USING WIRESHARK TO OBSERVE THE TCP THREE-WAY HANDSHAKE

CSE1004(NETWORK AND COMMUNICATION)LAB:L53-L54

MARCH 24, 2022

ANIRUDH VADERA

20BCE2940

Using Wireshark to Observe the TCP Three-way Handshake

Objectives

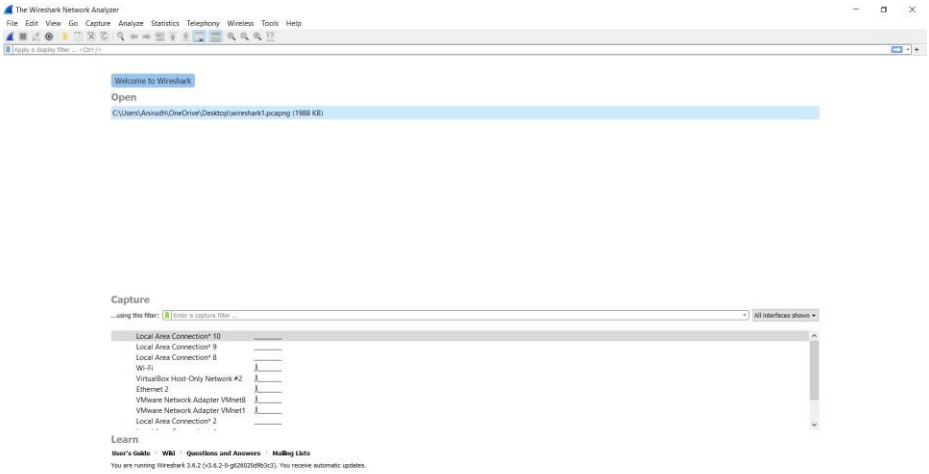
- Use Wireshark to monitor an Ethernet interface for recording packet flows
- Generate a TCP connection using a web browser
- Observe the initial TCP/IP three-way handshake

Task 1: Prepare Wireshark to Capture Packets

Step 1: Start Wireshark.

Double-click the Wireshark icon, which is located on the desktop.

THE INTERFACE LOOKS LIKE THIS


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Open
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

C:\Users\Anirudh\OneDrive\Desktop\wireshark1.pcapng (1988 KB)

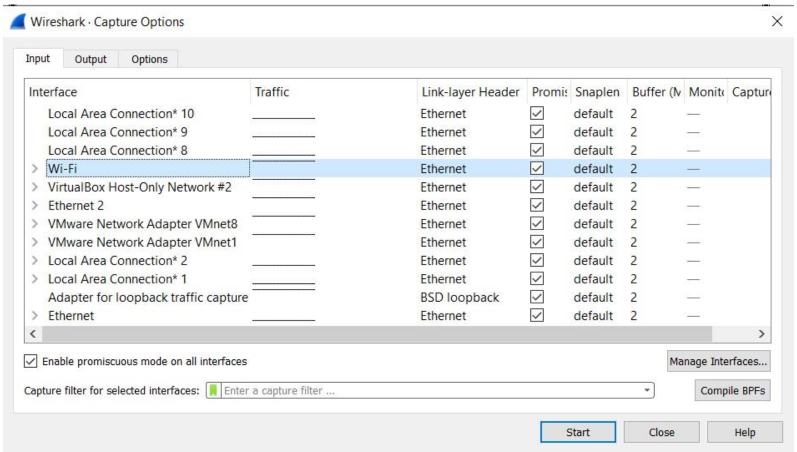
Local Area Connection* 10
Local Area Connection* 9
Local Area Connection* 8
Wi-Fi
VirtualBox Host-Only Network #2
Ethernet 2
VMware Network Adapter VMnet8
VMware Network Adapter VMnet1
Local Area Connection* 2

You are running Wireshark 3.6.2 (v3.6.2-0-g626020d9b3c3). You receive automatic updates.

Step 2: Select an interface to use for capturing packets.

a. From the Capture menu, choose Options.

Select your connection type



Interface	Traffic	Link-layer Header	Promis	Snaplen	Buffer (M	Monit	Captur
Local Area Connection* 10		Ethernet	☑	default	2	—	
Local Area Connection* 9		Ethernet	☑	default	2	—	
Local Area Connection* 8		Ethernet	☑	default	2	—	
Wi-Fi		Ethernet	☑	default	2	—	
VirtualBox Host-Only Network #2		Ethernet	☑	default	2	—	
Ethernet 2		Ethernet	☑	default	2	—	
VMware Network Adapter VMnet8		Ethernet	☑	default	2	—	
VMware Network Adapter VMnet1		Ethernet	☑	default	2	—	
Local Area Connection* 2		Ethernet	☑	default	2	—	
Local Area Connection* 1		Ethernet	☑	default	2	—	
Adapter for loopback traffic capture		BSD loopback	☑	default	2	—	
Ethernet		Ethernet	☑	default	2	—	

☑ Enable promiscuous mode on all interfaces

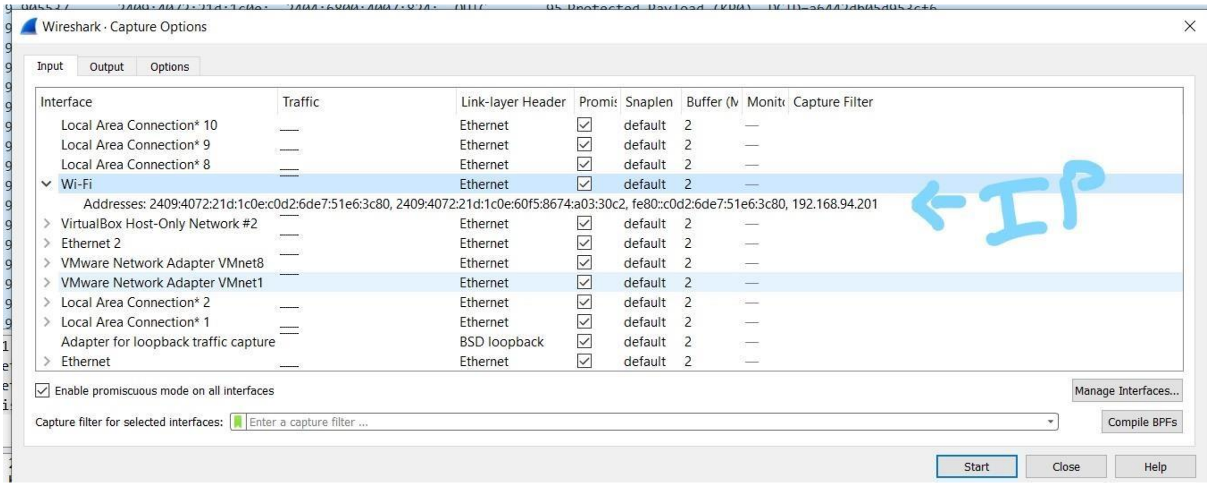
Capture filter for selected interfaces: Enter a capture filter ...

Step 3: Start a network capture.

a. Choose the WIFI for capturing network traffic. Click the Start button of the chosen interface.

b. Write down the IP address associated with the selected Ethernet adapter, because that is the source IP address to look for when examining captured packets.

The host IP address: 192.168.94.201



Interface	Traffic	Link-layer Header	Promis	Snaplen	Buffer (M	Monit	Capture Filter
Local Area Connection* 10		Ethernet	☑	default	2	—	
Local Area Connection* 9		Ethernet	☑	default	2	—	
Local Area Connection* 8		Ethernet	☑	default	2	—	
Wi-Fi		Ethernet	☑	default	2	—	
Addresses: 2409:4072:21d:1c0e:c0d2:6de7:51e6:3c80, 2409:4072:21d:1c0e:60f5:8674:a03:30c2, fe80::c0d2:6de7:51e6:3c80, 192.168.94.201							
VirtualBox Host-Only Network #2		Ethernet	☑	default	2	—	
Ethernet 2		Ethernet	☑	default	2	—	
VMware Network Adapter VMnet8		Ethernet	☑	default	2	—	
VMware Network Adapter VMnet1		Ethernet	☑	default	2	—	
Local Area Connection* 2		Ethernet	☑	default	2	—	
Local Area Connection* 1		Ethernet	☑	default	2	—	
Adapter for loopback traffic capture		BSD loopback	☑	default	2	—	
Ethernet		Ethernet	☑	default	2	—	

☑ Enable promiscuous mode on all interfaces

Capture filter for selected interfaces: Enter a capture filter ...

Task 2: Generate and Analyse Captured Packets

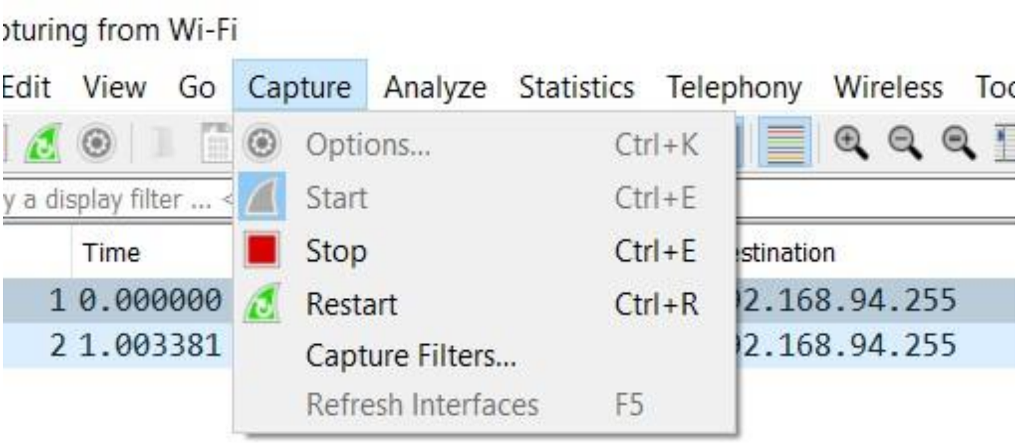
Step 1: Open a browser and access a website.

a. Go to www.google.com. Minimize the Google window, and return to Wireshark. You should see captured traffic similar to shown below

WEBSITE NAME: google.com,youtube.com

b. The capture Windows are now active. Locate the Source, Destination, and Protocol columns on the Wireshark display screen. The HTTP data that carries web page text and graphics uses TCP for reliability **Step 2: Stop the capture.**

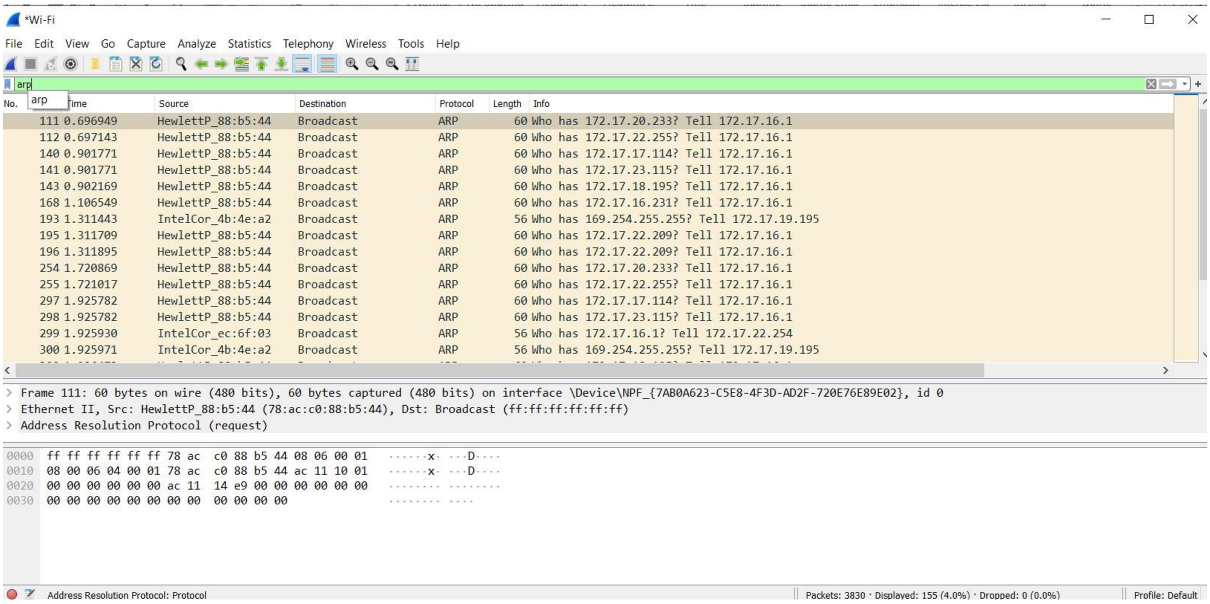
From the Wireshark Capture menu, choose Stop.



Step 3: Analyse the captured output.

If the computer was recently started and there has been no activity in accessing the Internet, you can see the entire process in the captured output, including ARP, DNS, and the TCP three-way handshake.

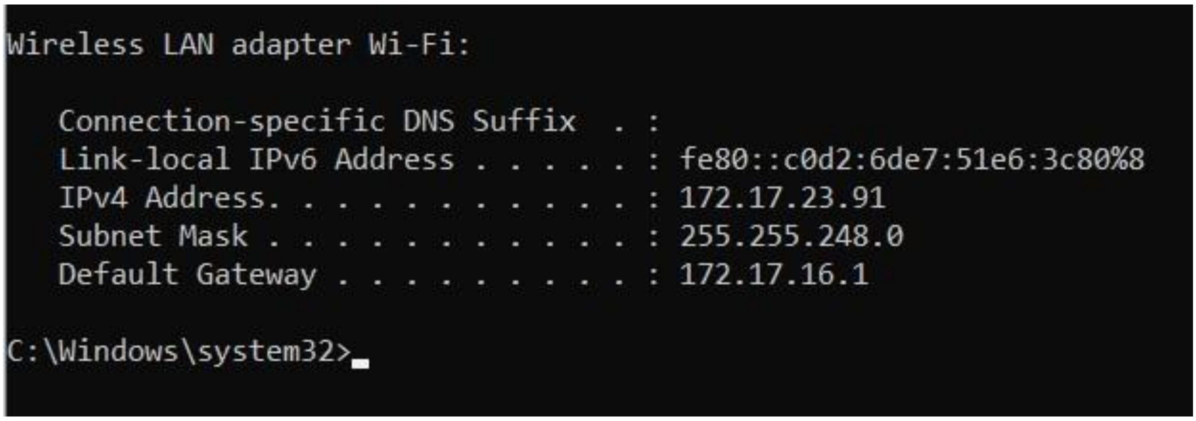
The capture screen in Task 2, Step 1 shows all the packets the computer needs to get to a website, starting with the initial ARP for the gateway router interface MAC address. (Your screen capture may vary.)



*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp

```
No.   Time        Source              Destination   Protocol  Length Info
  111 0.696949    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.20.233? Tell 172.17.16.1
  112 0.697143    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.22.255? Tell 172.17.16.1
  140 0.901771    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.17.114? Tell 172.17.16.1
  141 0.901771    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.23.115? Tell 172.17.16.1
  143 0.902169    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.18.195? Tell 172.17.16.1
  168 1.106549    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.16.231? Tell 172.17.16.1
  193 1.311443    IntelCor_4b:4e:a2   Broadcast     ARP       56 Who has 169.254.255.255? Tell 172.17.19.195
  195 1.311709    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.22.209? Tell 172.17.16.1
  196 1.311895    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.22.209? Tell 172.17.16.1
  254 1.720869    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.20.233? Tell 172.17.16.1
  255 1.721017    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.22.255? Tell 172.17.16.1
  297 1.925782    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.17.114? Tell 172.17.16.1
  298 1.925782    HewlettP_88:b5:44   Broadcast     ARP       60 Who has 172.17.23.115? Tell 172.17.16.1
  299 1.925930    IntelCor_ec:6f:03   Broadcast     ARP       56 Who has 172.17.16.1? Tell 172.17.22.254
  300 1.925971    IntelCor_4b:4e:a2   Broadcast     ARP       56 Who has 169.254.255.255? Tell 172.17.19.195

> Frame 111: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{7AB0A623-C5E8-4F3D-AD2F-720E76E89E02}, id 0
> Ethernet II, Src: HewlettP_88:b5:44 (78:ac:c0:88:b5:44), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 78 ac  c0 88 b5 44 08 06 00 01   ······x· ···D····
0010  08 00 06 04 00 01 78 ac  c0 88 b5 44 ac 11 10 01   ······x· ···D····
0020  00 00 00 00 00 00 ac 11  14 e9 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····

Address Resolution Protocol: Protocol        Packets: 3830 · Displayed: 155 (4.0%) · Dropped: 0 (0.0%)        Profile: Default
```

a. In the screen capture, the process starts with frame 1, which is an ARP broadcast from the source computer to determine the MAC address of the router default gateway. The gateway is the local LAN Fast Ethernet interface on the router. The computer needs to resolve the default gateway IP address to the interface MAC address before it can send the first frame or packet to the router.



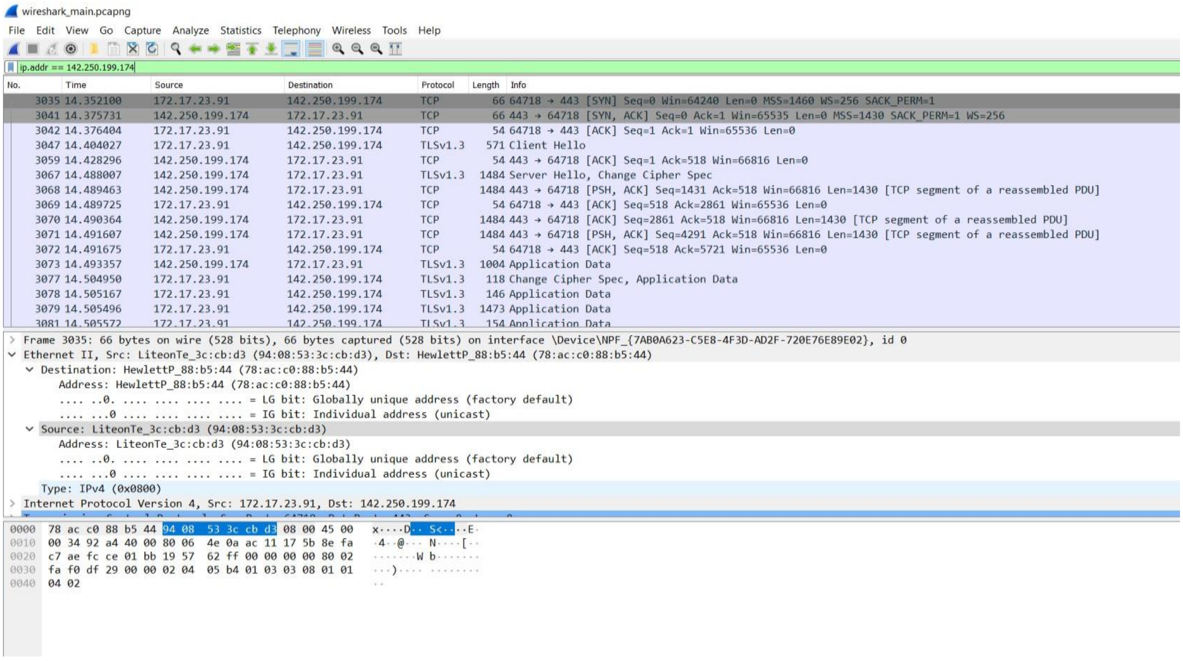
```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c0d2:6de7:51e6:3c80%8
   IPv4 Address. . . . . . . . . . . : 172.17.23.91
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Default Gateway . . . . . . . . . : 172.17.16.1

C:\Windows\system32>_
```

What is the IP address of the router default gateway? 172.17.16.1

b. The second frame is the reply from the router telling the computer the MAC address of its Fast Ethernet interface.

Ethernet II, Src: HewlettP_88:b5:44 (78:ac:c0:88:b5:44), Dst: LiteonTe_3c:cb:d3

What is the MAC address? 78:ac:c0:88:b5:44


wireshark_main.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 142.250.199.174

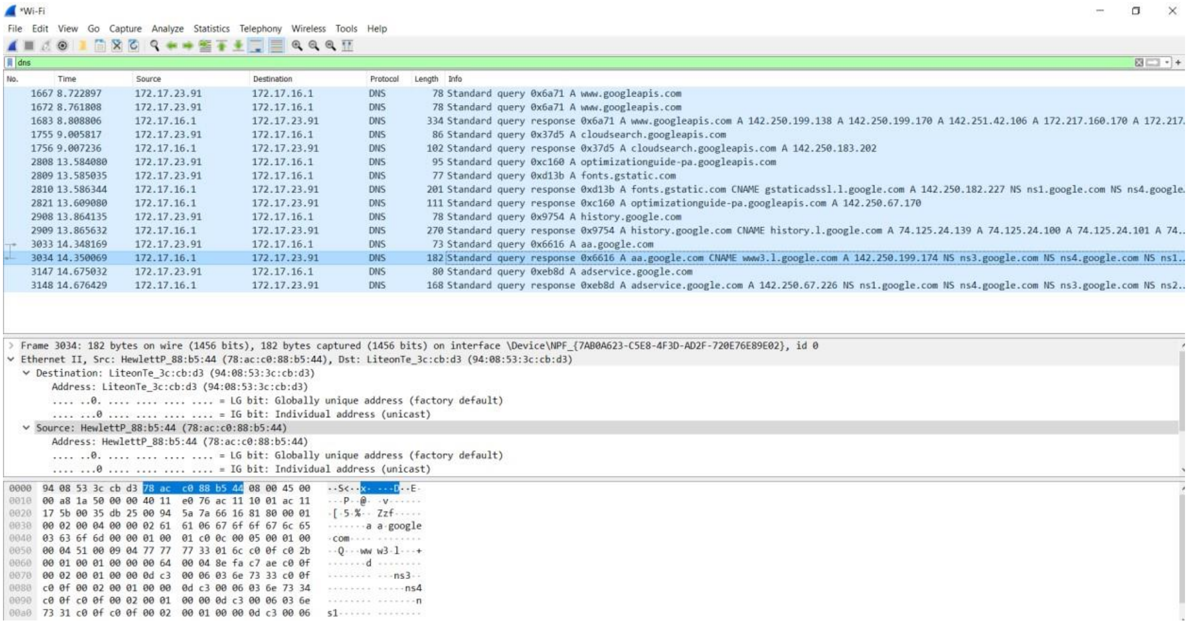
```
No.   Time       Source          Destination     Protocol Length Info
 3035 14.352100  172.17.23.91    142.250.199.174 TCP        66 64718 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
 3041 14.375731  142.250.199.174 172.17.23.91    TCP        66 443 → 64718 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 WS=256
 3042 14.376404  172.17.23.91    142.250.199.174 TCP        54 64718 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0
 3047 14.404027  172.17.23.91    142.250.199.174 TLSv1.3   571 Client Hello
 3059 14.428296  142.250.199.174 172.17.23.91    TCP        54 443 → 64718 [ACK] Seq=1 Ack=518 Win=66816 Len=0
 3067 14.488007  142.250.199.174 172.17.23.91    TLSv1.3  1484 Server Hello, Change Cipher Spec
 3068 14.489463  142.250.199.174 172.17.23.91    TCP      1484 443 → 64718 [PSH, ACK] Seq=1431 Ack=518 Win=66816 Len=1430 [TCP segment of a reassembled PDU]
 3069 14.489725  172.17.23.91    142.250.199.174 TCP        54 64718 → 443 [ACK] Seq=518 Ack=2861 Win=65536 Len=0
 3070 14.490364  142.250.199.174 172.17.23.91    TCP      1484 443 → 64718 [ACK] Seq=2861 Ack=518 Win=66816 Len=1430 [TCP segment of a reassembled PDU]
 3071 14.491607  142.250.199.174 172.17.23.91    TCP      1484 443 → 64718 [PSH, ACK] Seq=4291 Ack=518 Win=66816 Len=1430 [TCP segment of a reassembled PDU]
 3072 14.491675  172.17.23.91    142.250.199.174 TCP        54 64718 → 443 [ACK] Seq=518 Ack=5721 Win=65536 Len=0
 3073 14.493357  142.250.199.174 172.17.23.91    TLSv1.3  1004 Application Data
 3077 14.504950  172.17.23.91    142.250.199.174 TLSv1.3   118 Change Cipher Spec, Application Data
 3078 14.505167  172.17.23.91    142.250.199.174 TLSv1.3   146 Application Data
 3079 14.505496  172.17.23.91    142.250.199.174 TLSv1.3  1473 Application Data
 3081 14.505572  172.17.23.91    142.250.199.174 TLSv1.3   154 Application Data
```

```
> Frame 3035: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{7AB0A623-C5E8-4F3D-AD2F-720E76E89E02}, id 0
v Ethernet II, Src: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3), Dst: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
  v Destination: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
      Address: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
      Address: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 172.17.23.91, Dst: 142.250.199.174
```

```
0000  78 ac c0 88 b5 44 94 08  53 3c cb d3 08 00 45 00   x····D·· S<····E·
0010  00 34 92 a4 40 00 80 06  4e 0a ac 11 17 5b 8e fa   ·4··@··· N····[··
0020  c7 ae fc ce 01 bb 19 57  62 ff 00 00 00 00 80 02   ·······W b·······
0030  fa f0 df 29 00 00 02 04  05 b4 01 03 03 08 01 01   ···)···· ········
0040  04 02                                              ··
```

c. The frame is a DNS query from the computer to the configured DNS server, attempting to resolve the domain name www.google.com to the IP address of the web server. The computer must have the IP address before it can send the first frame to the web server.

What is the IP address of the DNS server that the computer queried? 172.17.23.91


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

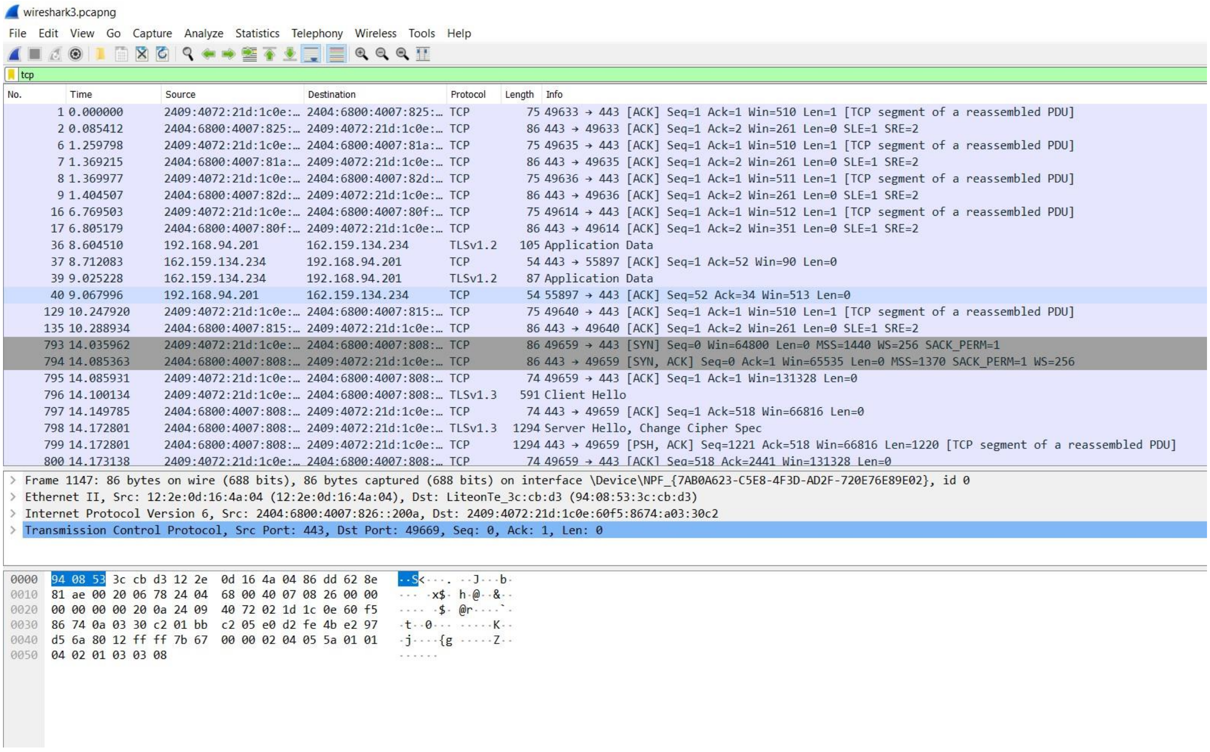
```
No.   Time       Source        Destination   Protocol Length Info
 1667 8.722897   172.17.23.91  172.17.16.1   DNS        78 Standard query 0x6a71 A www.googleapis.com
 1672 8.761808   172.17.23.91  172.17.16.1   DNS        78 Standard query 0x6a71 A www.googleapis.com
 1683 8.808806   172.17.16.1   172.17.23.91  DNS       334 Standard query response 0x6a71 A www.googleapis.com A 142.250.199.138 A 142.250.199.170 A 142.251.42.106 A 172.217.160.170 A 172.217.
 1755 9.005817   172.17.23.91  172.17.16.1   DNS        86 Standard query 0x37d5 A cloudsearch.googleapis.com
 1756 9.007236   172.17.16.1   172.17.23.91  DNS       102 Standard query response 0x37d5 A cloudsearch.googleapis.com A 142.250.183.202
 2808 13.584080  172.17.23.91  172.17.16.1   DNS        95 Standard query 0xc160 A optimizationguide-pa.googleapis.com
 2809 13.585035  172.17.23.91  172.17.16.1   DNS        77 Standard query 0xd13b A fonts.gstatic.com
 2810 13.586344  172.17.16.1   172.17.23.91  DNS       201 Standard query response 0xd13b A fonts.gstatic.com CNAME gstaticadssl.l.google.com A 142.250.182.227 NS ns1.google.com NS ns4.google.
 2821 13.609080  172.17.16.1   172.17.23.91  DNS       111 Standard query response 0xc160 A optimizationguide-pa.googleapis.com A 142.250.67.170
 2908 13.864135  172.17.23.91  172.17.16.1   DNS        78 Standard query 0x9754 A history.google.com
 2909 13.865632  172.17.16.1   172.17.23.91  DNS       270 Standard query response 0x9754 A history.google.com CNAME history.l.google.com A 74.125.24.139 A 74.125.24.100 A 74.125.24.101 A 74..
 3033 14.348169  172.17.23.91  172.17.16.1   DNS        73 Standard query 0x6616 A aa.google.com
 3034 14.350069  172.17.16.1   172.17.23.91  DNS       182 Standard query response 0x6616 A aa.google.com CNAME www3.l.google.com A 142.250.199.174 NS ns3.google.com NS ns4.google.com NS ns1..
 3147 14.675032  172.17.23.91  172.17.16.1   DNS        80 Standard query 0xeb8d A adservice.google.com
 3148 14.676429  172.17.16.1   172.17.23.91  DNS       168 Standard query response 0xeb8d A adservice.google.com A 142.250.67.226 NS ns1.google.com NS ns4.google.com NS ns3.google.com NS ns2..
```

```
> Frame 3034: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface \Device\NPF_{7AB0A623-C5E8-4F3D-AD2F-720E76E89E02}, id 0
v Ethernet II, Src: HewlettP_88:b5:44 (78:ac:c0:88:b5:44), Dst: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
  v Destination: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
      Address: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
      Address: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
```

```
0000  94 08 53 3c cb d3 78 ac  c0 88 b5 44 08 00 45 00   ··S<··x· ···D··E·
0010  00 a8 1a 50 00 00 40 11  e0 76 ac 11 10 01 ac 11   ···P··@· ·v······
0020  17 5b 00 35 db 25 00 94  5a 7a 66 16 81 80 00 01   ·[·5·%·· Zzf·····
0030  00 02 00 04 00 00 02 61  61 06 67 6f 6f 67 6c 65   ·······a a·google
0040  03 63 6f 6d 00 00 01 00  01 c0 0c 00 05 00 01 00   ·com···· ········
0050  00 04 51 00 09 04 77 77  77 33 01 6c c0 0f c0 2b   ··Q···ww w3·l···+
0060  00 01 00 01 00 00 00 64  00 04 8e fa c7 ae c0 0f   ·······d ········
0070  00 02 00 01 00 00 0d c3  00 06 03 6e 73 33 c0 0f   ········ ···ns3··
0080  c0 0f 00 02 00 01 00 00  0d c3 00 06 03 6e 73 34   ········ ·····ns4
0090  c0 0f c0 0f 00 02 00 01  00 00 0d c3 00 06 03 6e   ········ ·······n
00a0  73 31 c0 0f c0 0f 00 02  00 01 00 00 0d c3 00 06   s1······ ········
```

d. The next frame is the response from the DNS server with the IP address of www.google.com. You need to scroll to the right to see the IP address of the Google server in the DNS response, but you can see it in the next frame.

Ip address of www.google.com: 142.250.199.174


wireshark3.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2409:4072:21d:1c0e:…	2404:6800:4007:825:…	TCP	75	49633 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1 [TCP segment of a reassembled PDU]
2	0.085412	2404:6800:4007:825:…	2409:4072:21d:1c0e:…	TCP	86	443 → 49633 [ACK] Seq=1 Ack=2 Win=261 Len=0 SLE=1 SRE=2
6	1.259798	2409:4072:21d:1c0e:…	2404:6800:4007:81a:…	TCP	75	49635 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1 [TCP segment of a reassembled PDU]
7	1.369215	2404:6800:4007:81a:…	2409:4072:21d:1c0e:…	TCP	86	443 → 49635 [ACK] Seq=1 Ack=2 Win=261 Len=0 SLE=1 SRE=2
8	1.369977	2409:4072:21d:1c0e:…	2404:6800:4007:82d:…	TCP	75	49636 → 443 [ACK] Seq=1 Ack=1 Win=511 Len=1 [TCP segment of a reassembled PDU]
9	1.404507	2404:6800:4007:82d:…	2409:4072:21d:1c0e:…	TCP	86	443 → 49636 [ACK] Seq=1 Ack=2 Win=261 Len=0 SLE=1 SRE=2
16	6.769503	2409:4072:21d:1c0e:…	2404:6800:4007:80f:…	TCP	75	49614 → 443 [ACK] Seq=1 Ack=1 Win=512 Len=1 [TCP segment of a reassembled PDU]
17	6.805179	2404:6800:4007:80f:…	2409:4072:21d:1c0e:…	TCP	86	443 → 49614 [ACK] Seq=1 Ack=2 Win=351 Len=0 SLE=1 SRE=2
36	8.604510	192.168.94.201	162.159.134.234	TLSv1.2	105	Application Data
37	8.712083	162.159.134.234	192.168.94.201	TCP	54	443 → 55897 [ACK] Seq=1 Ack=52 Win=90 Len=0
39	9.025228	162.159.134.234	192.168.94.201	TLSv1.2	87	Application Data
40	9.067996	192.168.94.201	162.159.134.234	TCP	54	55897 → 443 [ACK] Seq=52 Ack=34 Win=513 Len=0
129	10.247920	2409:4072:21d:1c0e:…	2404:6800:4007:815:…	TCP	75	49640 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1 [TCP segment of a reassembled PDU]
135	10.288934	2404:6800:4007:815:…	2409:4072:21d:1c0e:…	TCP	86	443 → 49640 [ACK] Seq=1 Ack=2 Win=261 Len=0 SLE=1 SRE=2
793	14.035962	2409:4072:21d:1c0e:…	2404:6800:4007:808:…	TCP	86	49659 → 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM=1
794	14.085363	2404:6800:4007:808:…	2409:4072:21d:1c0e:…	TCP	86	443 → 49659 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1370 SACK_PERM=1 WS=256
795	14.085931	2409:4072:21d:1c0e:…	2404:6800:4007:808:…	TCP	74	49659 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
796	14.100134	2409:4072:21d:1c0e:…	2404:6800:4007:808:…	TLSv1.3	591	Client Hello
797	14.149785	2404:6800:4007:808:…	2409:4072:21d:1c0e:…	TCP	74	443 → 49659 [ACK] Seq=1 Ack=518 Win=66816 Len=0
798	14.172801	2404:6800:4007:808:…	2409:4072:21d:1c0e:…	TLSv1.3	1294	Server Hello, Change Cipher Spec
799	14.172801	2404:6800:4007:808:…	2409:4072:21d:1c0e:…	TCP	1294	443 → 49659 [PSH, ACK] Seq=1221 Ack=518 Win=66816 Len=1220 [TCP segment of a reassembled PDU]
800	14.173138	2409:4072:21d:1c0e:…	2404:6800:4007:808:…	TCP	74	49659 → 443 [ACK] Seq=518 Ack=2441 Win=131328 Len=0

```
> Frame 1147: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{7AB0A623-C5E8-4F3D-AD2F-720E76E89E02}, id 0
> Ethernet II, Src: 12:2e:0d:16:4a:04 (12:2e:0d:16:4a:04), Dst: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
> Internet Protocol Version 6, Src: 2404:6800:4007:826::200a, Dst: 2409:4072:21d:1c0e:60f5:8674:a03:30c2
> Transmission Control Protocol, Src Port: 443, Dst Port: 49669, Seq: 0, Ack: 1, Len: 0
```

```
0000  94 08 53 3c cb d3 12 2e  0d 16 4a 04 86 dd 62 8e   ··S<···. ··J···b·
0010  81 ae 00 20 06 78 24 04  68 00 40 07 08 26 00 00   ··· ·x$· h·@··&··
0020  00 00 00 00 20 0a 24 09  40 72 02 1d 1c 0e 60 f5   ···· ·$· @r····`·
0030  86 74 0a 03 30 c2 01 bb  c2 05 e0 d2 fe 4b e2 97   ·t··0··· ·····K··
0040  d5 6a 80 12 ff ff 7b 67  00 00 02 04 05 5a 01 01   ·j····{g ·····Z··
0050  04 02 01 03 03 08                                  ······
```

e. The next frame is the start of the TCP three-way handshake [SYN].

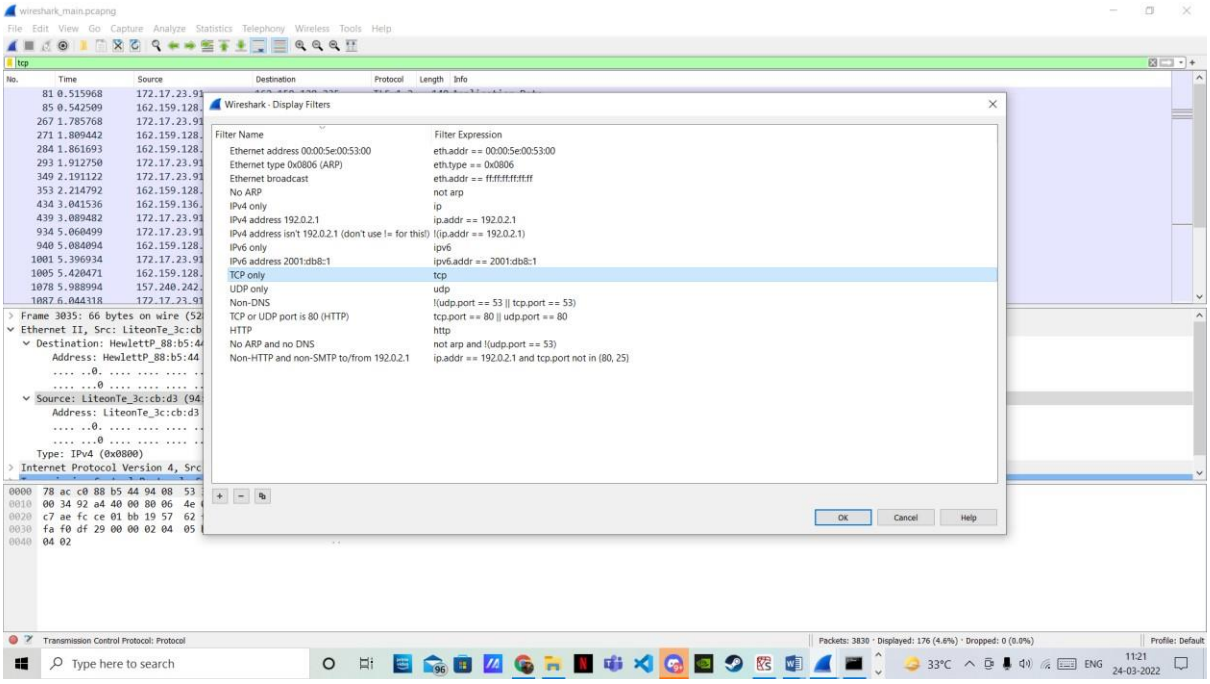
What is the IP address of the Google web server? 142.250.199.174

Step 4: Filter the capture to view only TCP packets

If you have many packets that are unrelated to the TCP connection, it may be necessary to use the Wireshark filter capability.

a. To use a preconfigured filter, click the Analyze menu option, and then click Display Filters.

b. In the Display Filter window, click TCP only, and then click OK.

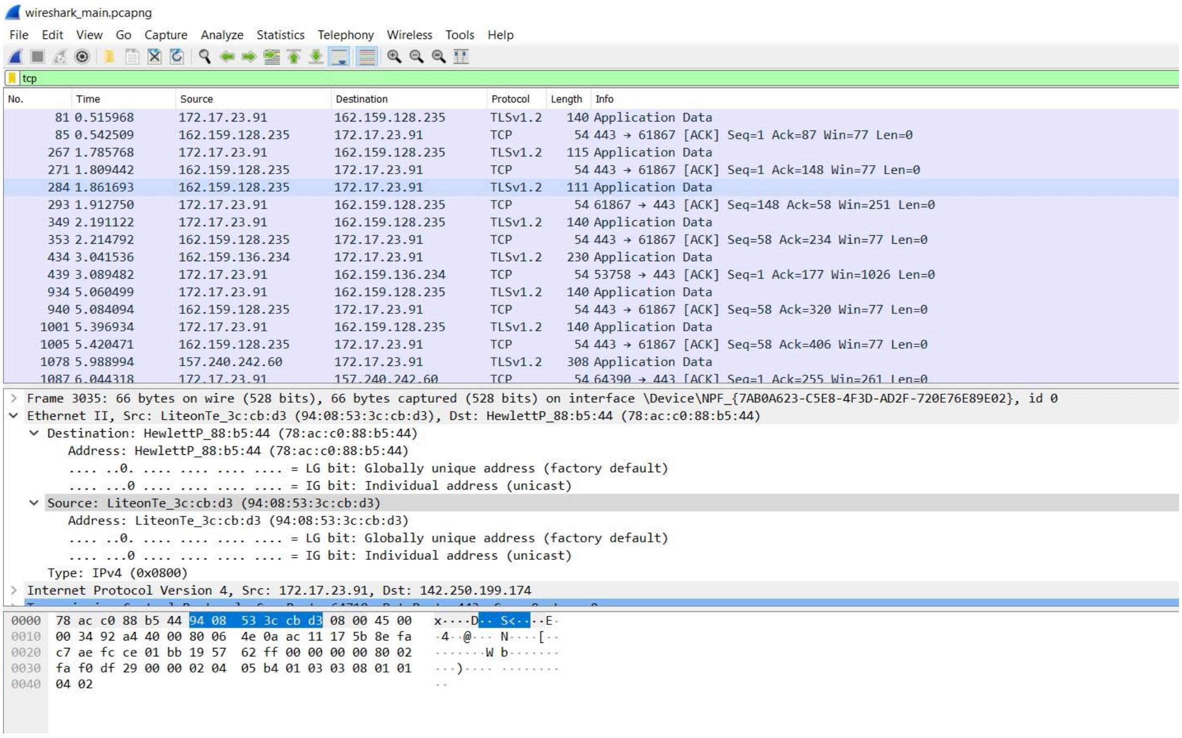


Wireshark · Display Filters

Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	eth.addr == 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	eth.type == 0x0806
Ethernet broadcast	eth.addr == ff:ff:ff:ff:ff:ff
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	ip.addr == 192.0.2.1
IPv4 address isn't 192.0.2.1 (don't use != for this!)	!(ip.addr == 192.0.2.1)
IPv6 only	ipv6
IPv6 address 2001:db8::1	ipv6.addr == 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	!(udp.port == 53 || tcp.port == 53)
TCP or UDP port is 80 (HTTP)	tcp.port == 80 || udp.port == 80
HTTP	http
No ARP and no DNS	not arp and !(udp.port == 53)
Non-HTTP and non-SMTP to/from 192.0.2.1	ip.addr == 192.0.2.1 and tcp.port not in {80, 25}

wireshark_main.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
OK
Cancel
Help
Transmission Control Protocol: Protocol
Packets: 3830 · Displayed: 176 (4.6%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
33°C
ENG
11:21
24-03-2022

The tcp packets are:



wireshark_main.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Filter: tcp

```
No.   Time      Source           Destination      Protocol Length Info
   81 0.515968  172.17.23.91     162.159.128.235  TLSv1.2  140 Application Data
   85 0.542509  162.159.128.235  172.17.23.91     TCP       54 443 → 61867 [ACK] Seq=1 Ack=87 Win=77 Len=0
  267 1.785768  172.17.23.91     162.159.128.235  TLSv1.2  115 Application Data
  271 1.809442  162.159.128.235  172.17.23.91     TCP       54 443 → 61867 [ACK] Seq=1 Ack=148 Win=77 Len=0
  284 1.861693  162.159.128.235  172.17.23.91     TLSv1.2  111 Application Data
  293 1.912750  172.17.23.91     162.159.128.235  TCP       54 61867 → 443 [ACK] Seq=148 Ack=58 Win=251 Len=0
  349 2.191122  172.17.23.91     162.159.128.235  TLSv1.2  140 Application Data
  353 2.214792  162.159.128.235  172.17.23.91     TCP       54 443 → 61867 [ACK] Seq=58 Ack=234 Win=77 Len=0
  434 3.041536  162.159.136.234  172.17.23.91     TLSv1.2  230 Application Data
  439 3.089482  172.17.23.91     162.159.136.234  TCP       54 53758 → 443 [ACK] Seq=1 Ack=177 Win=1026 Len=0
  934 5.060499  172.17.23.91     162.159.128.235  TLSv1.2  140 Application Data
  940 5.084094  162.159.128.235  172.17.23.91     TCP       54 443 → 61867 [ACK] Seq=58 Ack=320 Win=77 Len=0
 1001 5.396934  172.17.23.91     162.159.128.235  TLSv1.2  140 Application Data
 1005 5.420471  162.159.128.235  172.17.23.91     TCP       54 443 → 61867 [ACK] Seq=58 Ack=406 Win=77 Len=0
 1078 5.988994  157.240.242.60   172.17.23.91     TLSv1.2  308 Application Data
 1087 6.044318  172.17.23.91     157.240.242.60   TCP       54 64390 → 443 [ACK] Seq=1 Ack=255 Win=261 Len=0
```

```
> Frame 3035: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{7AB0A623-C5E8-4F3D-AD2F-720E76E89E02}, id 0
v Ethernet II, Src: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3), Dst: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
  v Destination: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
      Address: HewlettP_88:b5:44 (78:ac:c0:88:b5:44)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
      Address: LiteonTe_3c:cb:d3 (94:08:53:3c:cb:d3)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 172.17.23.91, Dst: 142.250.199.174
```

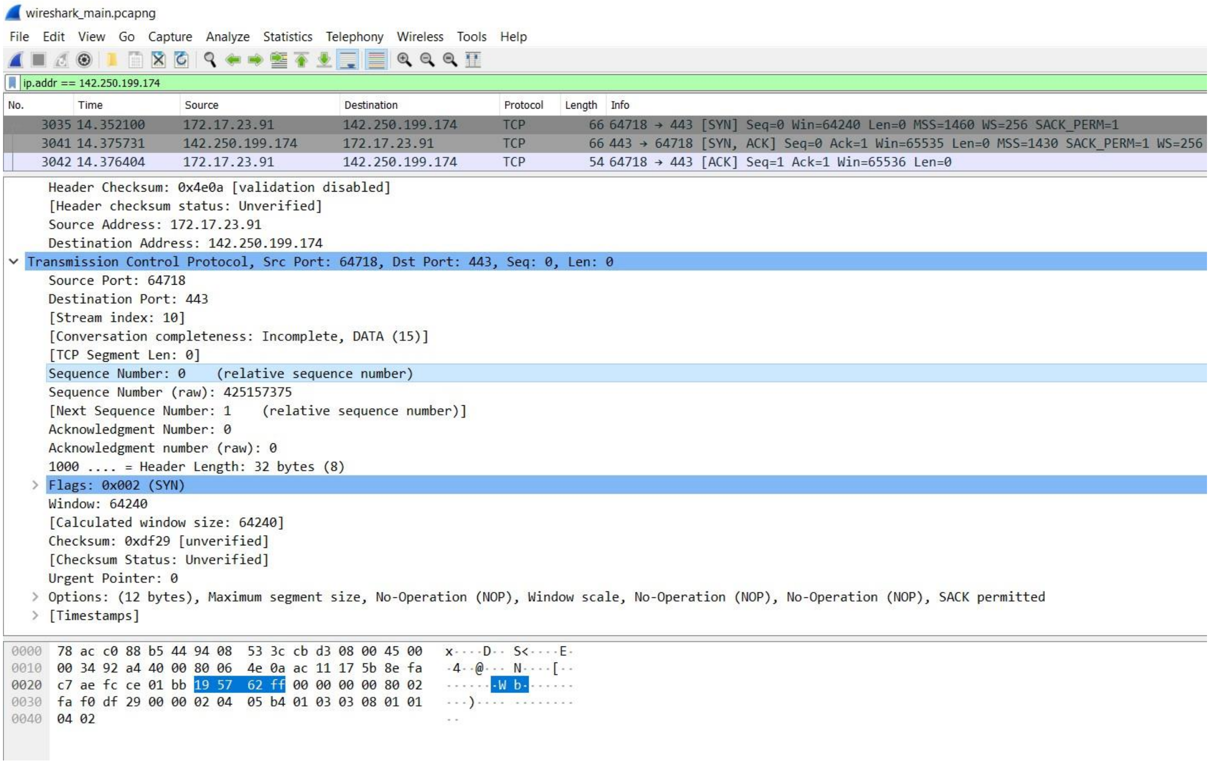
```
0000  78 ac c0 88 b5 44 94 08  53 3c cb d3 08 00 45 00   x····D·· S<····E·
0010  00 34 92 a4 40 00 80 06  4e 0a ac 11 17 5b 8e fa   ·4··@··· N····[··
0020  c7 ae fc ce 01 bb 19 57  62 ff 00 00 00 00 80 02   ·······W b·······
0030  fa f0 df 29 00 00 02 04  05 b4 01 03 03 08 01 01   ···)···· ········
0040  04 02                                              ··
```

e. In the Info column, look for three packets similar to the first three shown in the window above. The first TCP packet is the [SYN] packet from the initiating computer. The second is the [SYN, ACK] Page 4 of 7 responses from the web server. The third packet is the [ACK] from the source computer, which completes the handshake.

Step 5: Inspect the TCP initialization Sequence

a. In the top Wireshark window, click on the line containing the first packet identified in Step 4. This highlights the line and displays the decoded information from that packet in the two lower windows fill. Note: The Wireshark windows below were adjusted to allow the information to be viewed in a compact size. The middle window contains the detailed decoding of the packet.

b. Click the + icon to expand the view of the TCP information. To contract the view, click the – icon. c. Notice in the first TCP packet that the relative sequence number is set to 0, and the SYN bit is set to 1 in the Flags field.


wireshark_main.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

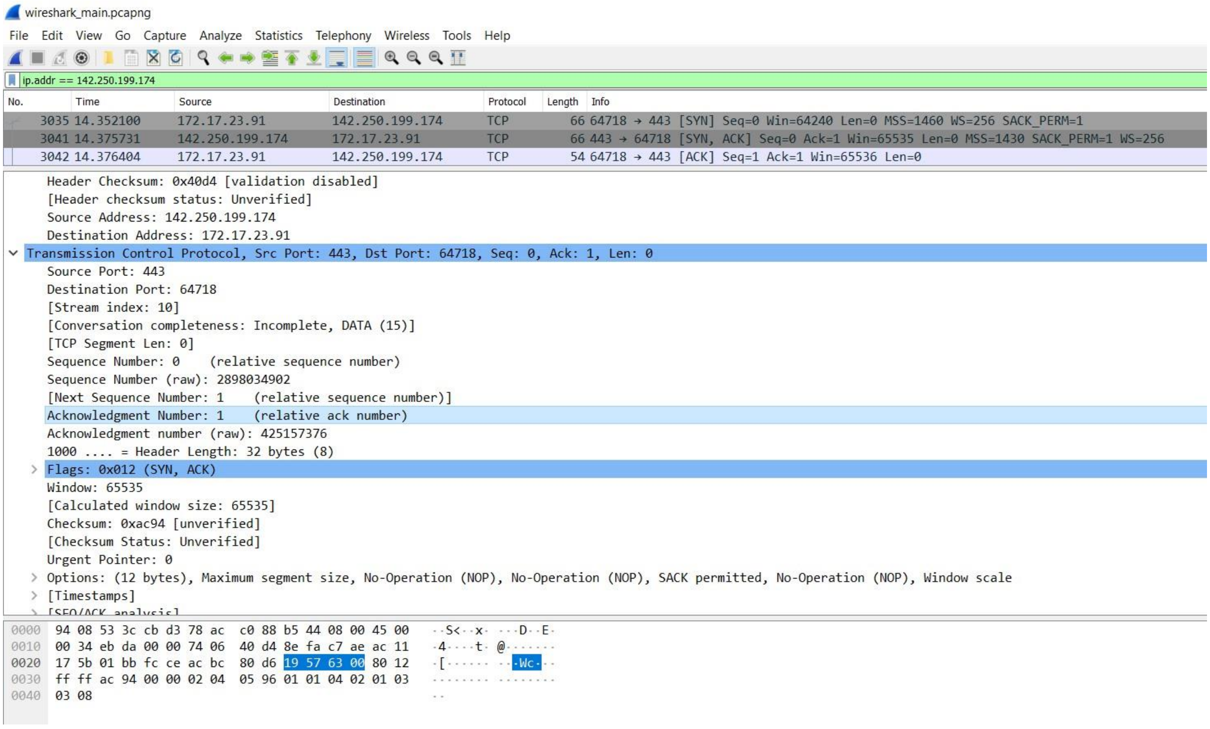
ip.addr == 142.250.199.174

```
No.     Time       Source           Destination      Protocol  Length  Info
   3035 14.352100  172.17.23.91     142.250.199.174  TCP          66 64718 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   3041 14.375731  142.250.199.174  172.17.23.91     TCP          66 443 → 64718 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 WS=256
   3042 14.376404  172.17.23.91     142.250.199.174  TCP          54 64718 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0
```

```
    Header Checksum: 0x4e0a [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.17.23.91
    Destination Address: 142.250.199.174
v Transmission Control Protocol, Src Port: 64718, Dst Port: 443, Seq: 0, Len: 0
    Source Port: 64718
    Destination Port: 443
    [Stream index: 10]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 425157375
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window: 64240
    [Calculated window size: 64240]
    Checksum: 0xdf29 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]
```

```
0000  78 ac c0 88 b5 44 94 08  53 3c cb d3 08 00 45 00   x····D·· S<····E·
0010  00 34 92 a4 40 00 80 06  4e 0a ac 11 17 5b 8e fa   ·4··@··· N····[··
0020  c7 ae fc ce 01 bb 19 57  62 ff 00 00 00 00 80 02   ·······W b·······
0030  fa f0 df 29 00 00 02 04  05 b4 01 03 03 08 01 01   ···)···· ········
0040  04 02                                              ··
```

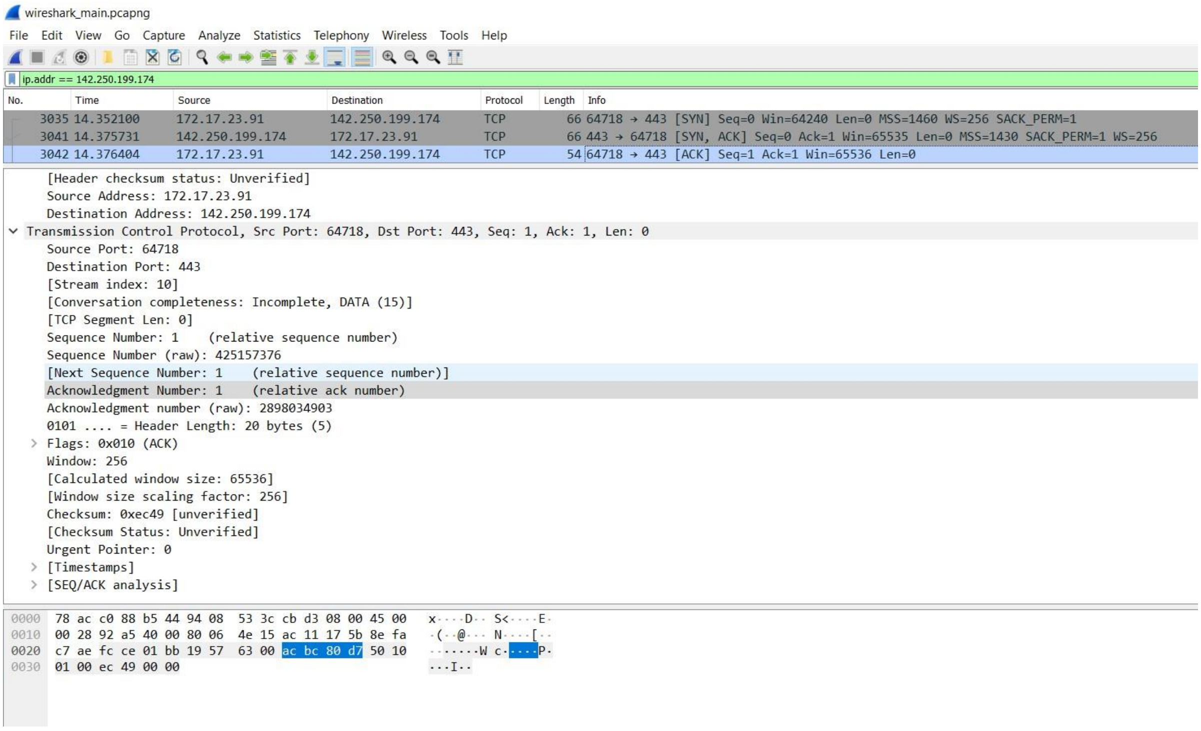
Notice in the second TCP packet of the handshake that the relative sequence number is set to 0, and the SYN bit and the ACK bit are set to 1 in the Flags field.



wireshark_main.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 142.250.199.174

No.   Time       Source          Destination     Protocol Length Info
3035 14.352100   172.17.23.91    142.250.199.174 TCP      66 64718 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3041 14.375731   142.250.199.174 172.17.23.91    TCP      66 443 → 64718 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 WS=256
3042 14.376404   172.17.23.91    142.250.199.174 TCP      54 64718 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0

    Header Checksum: 0x40d4 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 142.250.199.174
    Destination Address: 172.17.23.91
v Transmission Control Protocol, Src Port: 443, Dst Port: 64718, Seq: 0, Ack: 1, Len: 0
    Source Port: 443
    Destination Port: 64718
    [Stream index: 10]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 2898034902
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 425157376
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x012 (SYN, ACK)
    Window: 65535
    [Calculated window size: 65535]
    Checksum: 0xac94 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  > [Timestamps]
  > [SEQ/ACK analysis]

0000  94 08 53 3c cb d3 78 ac  c0 88 b5 44 08 00 45 00   ··S<··x· ···D··E·
0010  00 34 eb da 00 00 74 06  40 d4 8e fa c7 ae ac 11   ·4····t· @·······
0020  17 5b 01 bb fc ce ac bc  80 d6 19 57 63 00 80 12   ·[······ ··Wc····
0030  ff ff ac 94 00 00 02 04  05 96 01 01 04 02 01 03   ········ ········
0040  03 08                                              ··
```

In the third and final frame of the handshake, only the ACK bit is set, and the sequence number is set to the starting point of 1. The acknowledgement number is also set to 1 as a starting point. The TCP connection is now established, and communication between the source computer and the web server can begin.



wireshark_main.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 142.250.199.174

No.   Time       Source          Destination     Protocol Length Info
3035 14.352100   172.17.23.91    142.250.199.174 TCP      66 64718 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3041 14.375731   142.250.199.174 172.17.23.91    TCP      66 443 → 64718 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 WS=256
3042 14.376404   172.17.23.91    142.250.199.174 TCP      54 64718 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0

    [Header checksum status: Unverified]
    Source Address: 172.17.23.91
    Destination Address: 142.250.199.174
v Transmission Control Protocol, Src Port: 64718, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
    Source Port: 64718
    Destination Port: 443
    [Stream index: 10]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 425157376
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2898034903
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 256
    [Calculated window size: 65536]
    [Window size scaling factor: 256]
    Checksum: 0xec49 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]

0000  78 ac c0 88 b5 44 94 08  53 3c cb d3 08 00 45 00   x····D·· S<····E·
0010  00 28 92 a5 40 00 80 06  4e 15 ac 11 17 5b 8e fa   ·(··@··· N····[··
0020  c7 ae fc ce 01 bb 19 57  63 00 ac bc 80 d7 50 10   ·······W c·····P·
0030  01 00 ec 49 00 00                                  ···I··
```

f. Close Wireshark.

Task 3: Reflection

a. There are hundreds of filters available in Wireshark. A large network could have numerous filters and many different types of traffic. Which three filters in the list might be the most useful to a network administrator?

IP.addr == <"Ip Address">

Protocols such as HTTP,TCP,Arp,DNS

What other ways could Wireshark be used in a production network?

Wireshark is often used for security purposes for after-the-fact analysis of normal traffic or after a network attack. New protocols or services may need to be captured to determine what port or ports are used.