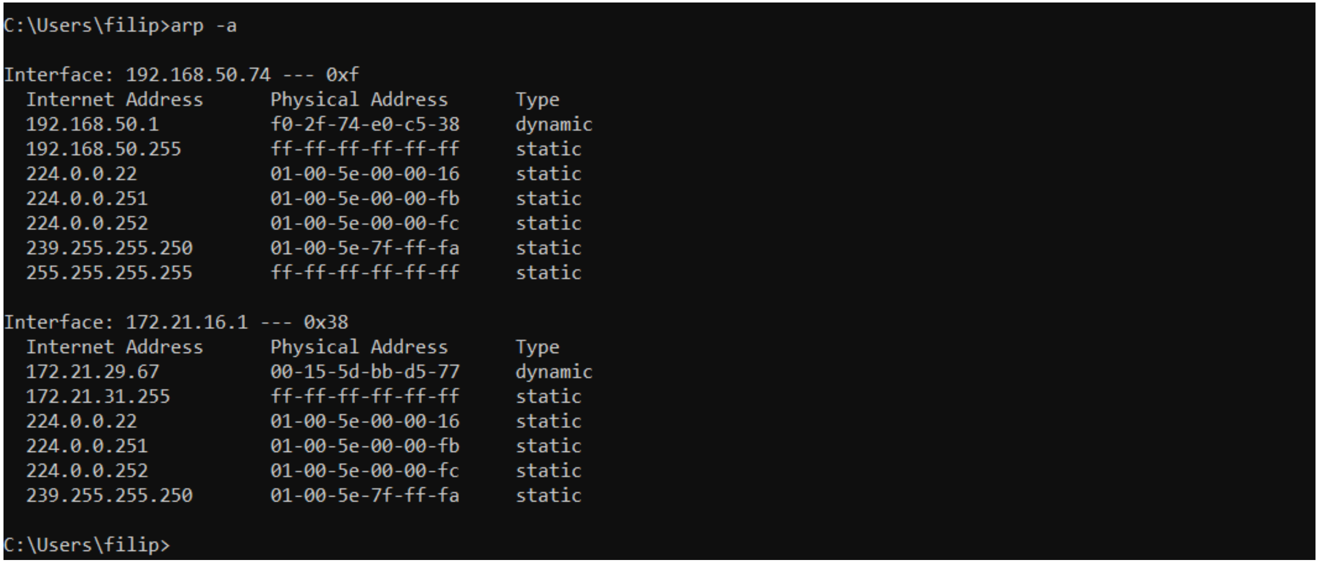
Exercise 1 – Basic network stuff

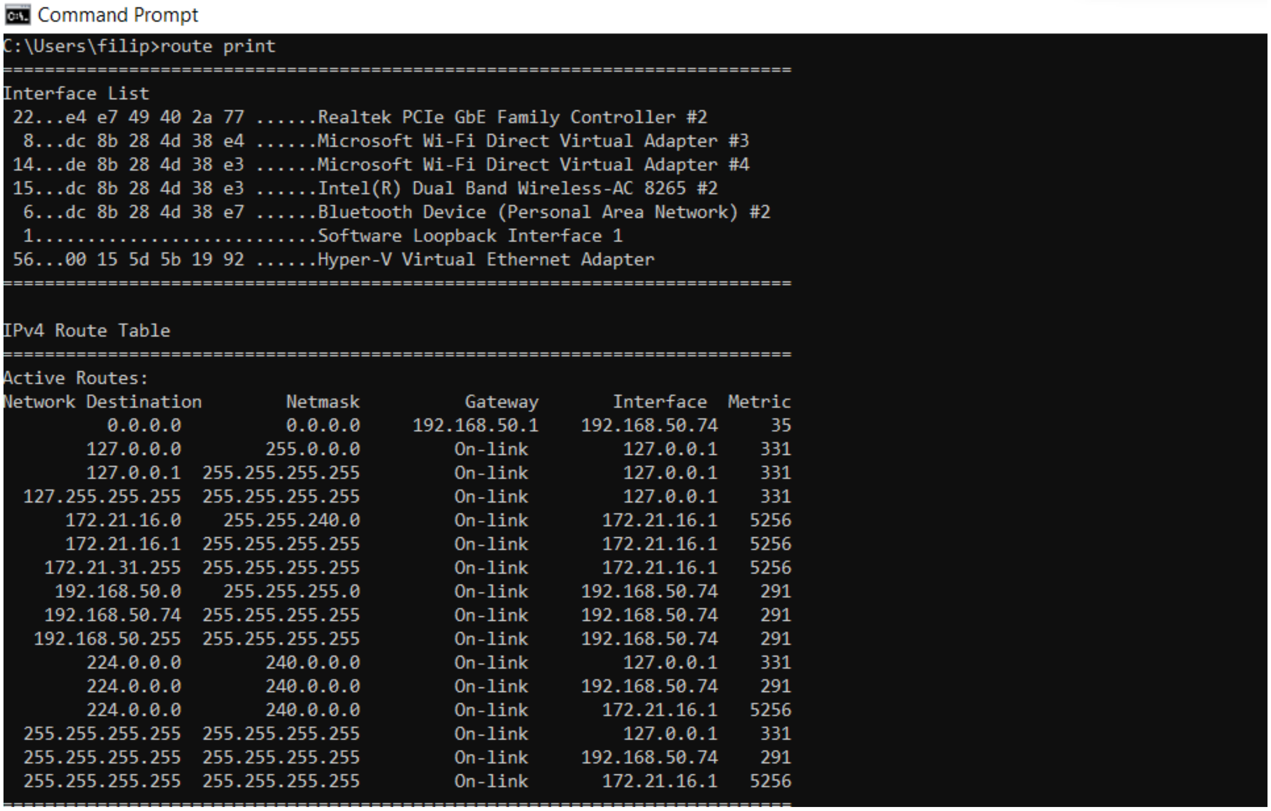
Difficulty: **Easy**

Use the `arp` command and paste the output from the arp table on your system:

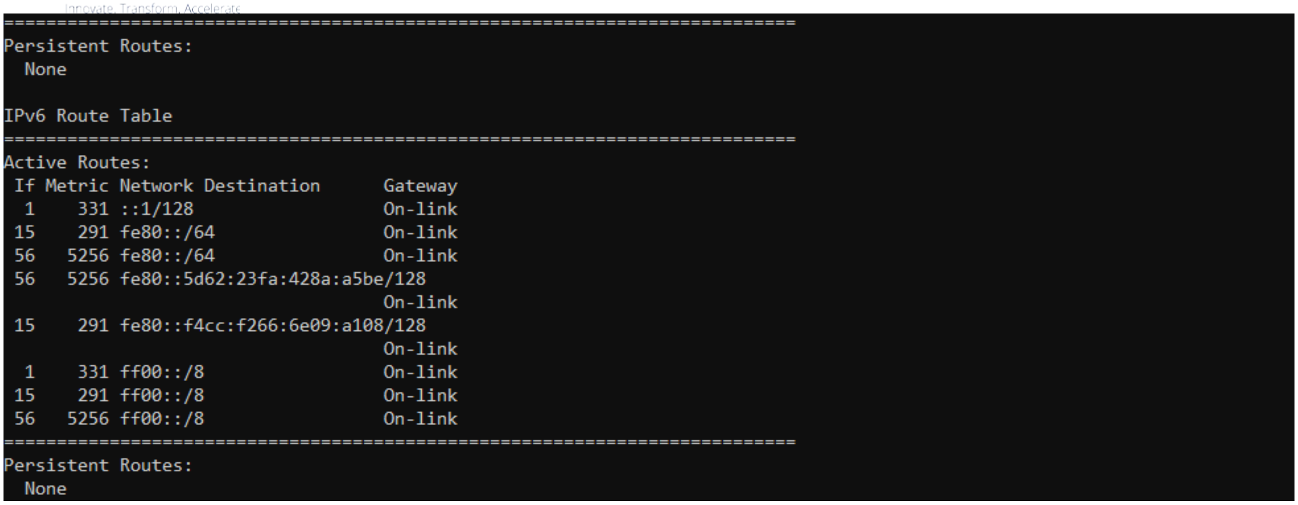


```
C:\Users\filip>arp -a

Interface: 192.168.50.74 --- 0xf
  Internet Address      Physical Address      Type
  192.168.50.1          f0-2f-74-e0-c5-38     dynamic
  192.168.50.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 172.21.16.1 --- 0x38
  Internet Address      Physical Address      Type
  172.21.29.67          00-15-5d-bb-d5-77     dynamic
  172.21.31.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

C:\Users\filip>
```

Use the `route` command and paste the output from the routing table on your system:

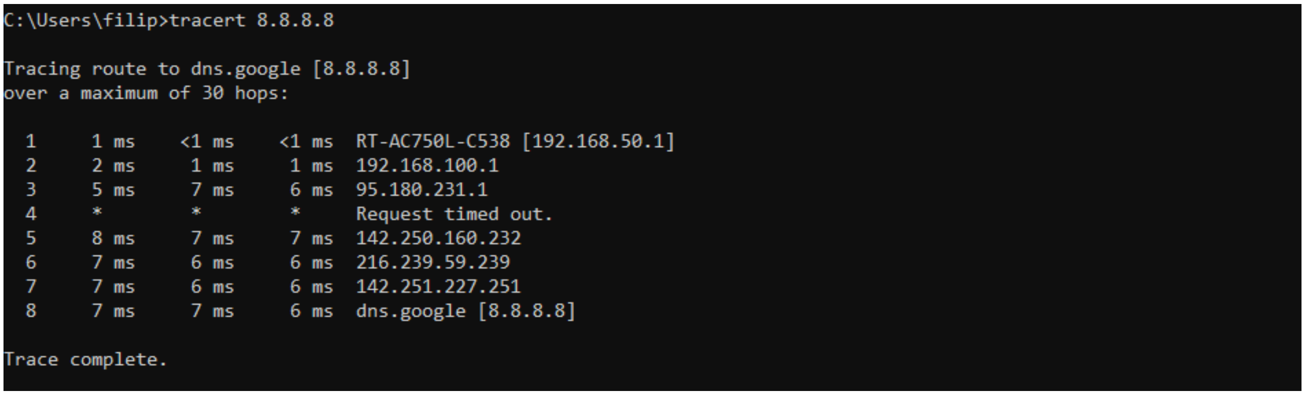


```
C:\Users\filip>route print
===========================================================================
Interface List
 22...e4 e7 49 40 2a 77 ......Realtek PCIe GbE Family Controller #2
  8...dc 8b 28 4d 38 e4 ......Microsoft Wi-Fi Direct Virtual Adapter #3
 14...de 8b 28 4d 38 e3 ......Microsoft Wi-Fi Direct Virtual Adapter #4
 15...dc 8b 28 4d 38 e3 ......Intel(R) Dual Band Wireless-AC 8265 #2
  6...dc 8b 28 4d 38 e7 ......Bluetooth Device (Personal Area Network) #2
  1...........................Software Loopback Interface 1
 56...00 15 5d 5b 19 92 ......Hyper-V Virtual Ethernet Adapter
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0     192.168.50.1    192.168.50.74     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
       172.21.16.0    255.255.240.0         On-link       172.21.16.1   5256
       172.21.16.1  255.255.255.255         On-link       172.21.16.1   5256
     172.21.31.255  255.255.255.255         On-link       172.21.16.1   5256
      192.168.50.0    255.255.255.0         On-link     192.168.50.74    291
     192.168.50.74  255.255.255.255         On-link     192.168.50.74    291
    192.168.50.255  255.255.255.255         On-link     192.168.50.74    291
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link     192.168.50.74    291
        224.0.0.0        240.0.0.0         On-link       172.21.16.1   5256
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link     192.168.50.74    291
  255.255.255.255  255.255.255.255         On-link       172.21.16.1   5256
===========================================================================
```



```
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
 15    291 fe80::/64                On-link
 56   5256 fe80::/64                On-link
 56   5256 fe80::5d62:23fa:428a:a5be/128
                                    On-link
 15    291 fe80::f4cc:f266:6e09:a108/128
                                    On-link
  1    331 ff00::/8                 On-link
 15    291 ff00::/8                 On-link
 56   5256 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

Use the `traceroute` command on your system and observe the hops to Google’s DNS, 8.8.8.8. Paste the full output from the command bellow showing all the hops from your system to 8.8.8.8.



```
C:\Users\filip>tracert 8.8.8.8

Tracing route to dns.google [8.8.8.8]
over a maximum of 30 hops:

  1     1 ms    <1 ms    <1 ms  RT-AC750L-C538 [192.168.50.1]
  2     2 ms     1 ms     1 ms  192.168.100.1
  3     5 ms     7 ms     6 ms  95.180.231.1
  4     *        *        *     Request timed out.
  5     8 ms     7 ms     7 ms  142.250.160.232
  6     7 ms     6 ms     6 ms  216.239.59.239
  7     7 ms     6 ms     6 ms  142.251.227.251
  8     7 ms     7 ms     6 ms  dns.google [8.8.8.8]

Trace complete.
```

Why would you need to use the ping command?
Answer: The ping command is a network tool that is commonly used to test the connectivity between two devices on a network. When you use the ping command, your computer sends a little packet of information to another computer on the internet or on your local network. Then the other computer sends the packet back to you, and your computer measures how long it took for the response to come back. So, the ping command is a tool that helps us check if two devices can communicate with each other and how fast they can do it.

Write down the TCP/UDP ports of the most commonly used services bellow in the form of TCP[PORT] or UDP[PORT].
As an example, the first two answers have been filled in:

- HTTP – TCP80
- SNMP – UDP161
- HTTPS - TCP443
- DNS client – UDP 53

- DNS zone transfer – TCP 53
- SMTP – TCP 25
- SSH – TCP 22
- FTP - TCP 21
- Telnet – TCP 23
- MSSQL – TCP 1443
- MySQL – TCP 3306
- PostreSQL - TCP 5432
- RDP (Remote Desktop Protocol) – TCP 3389
- NTP – TCP 123
- NFS – TCP 2049

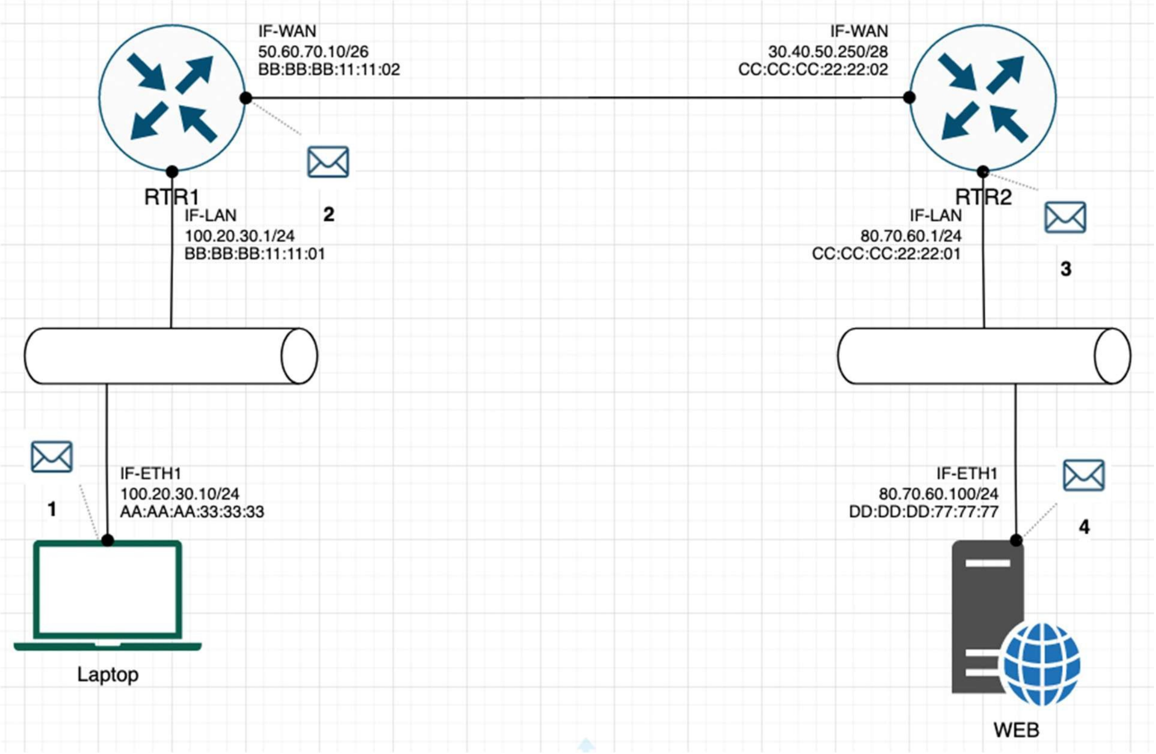
Exercise 2 – TCP/IP Basics

Difficulty: **Medium**

Refer to the exhibit and answer the questions below.

The letter symbol ✉, represents the IP packet as it travels across the network.
In the example shown, the laptop attempts to communicate with the web server in question.
During its travel the packet will be forwarded across the network nodes and will eventually end up across six network interfaces before it reaches the web server. Each

packet as part of the TCP/IP Stack contains fields for the source and destination MAC Address, IP Address and the TCP/UDP Port.



For each of the packet locations shown, 1 to 4 write down the source and destination MAC addresses of the packet as it travels across the network interfaces.

1. The laptop initiates communication with the web server and prepares a packet. What would the packet look like at this stage?
   - SRC IP   100.20.30.10
   - DST IP   80.70.60.10
   - SRC MAC AA:AA:AA:33:33:33
   - DST MAC BB:BB:BB:11:11:01

2. RTR1 receives the packet on its IF-LAN interface, prepares it accordingly and forwards it out its IF- WAN. What would the packet look like at this stage?
   - SRC IP   100.20.30.10
   - DST IP 80.70.60.10
   - SRC MAC BB:BB:BB:11:11:02
   - DST MAC CC:CC:CC:22:22:02

3. RTR2 receives the packet on its IF-WAN interface, prepares it accordingly and forwards it out via IF- LAN. What would the packet look like at this stage?
   - SRC IP 100.20.30.10
   - DST IP 80.70.60.10
   - SRC MAC CC:CC:CC:22:22:01
   - DST MAC DD:DD:DD:77:77:77

4. The web server receives the packet and prepares a response packet back. What would the packet look like at this stage?
   - SRC IP 80.70.60.10
   - DST IP 100.20.30.10
   - SRC MAC DD:DD:DD:77:77:77
   - DST MAC CC:CC:CC:22:22:01

Since we are talking about web traffic (www) in the example, which transport layer protocol will most probably be used?

- ❑ **TCP – Is most probably to be used**
- ❑ UDP

If we do a traffic analysis with a network packet monitoring tool like WireShark, what can we expect to see for the source and destination ports when the laptop sends the packet?

- SRC PORT: 1023 (for example) Random port
- DST PORT: 443(HTTPS)

Similarly, and vice versa, what can we expect to see as destination ports when the Web server sends a response packet back?

- SRC PORT: 443(HTTPS)
- DST PORT: Same port that was used to send the initial packet.

How many broadcast domains are there in the exhibit shown?

There are 3 broadcast domains.

Each LAN that is connected to a Router is a broadcast domain , and each connection between two Routers is a broadcast domain.

Exercise 3 – Traffic analysis and identifying the OSI layers of the network packets

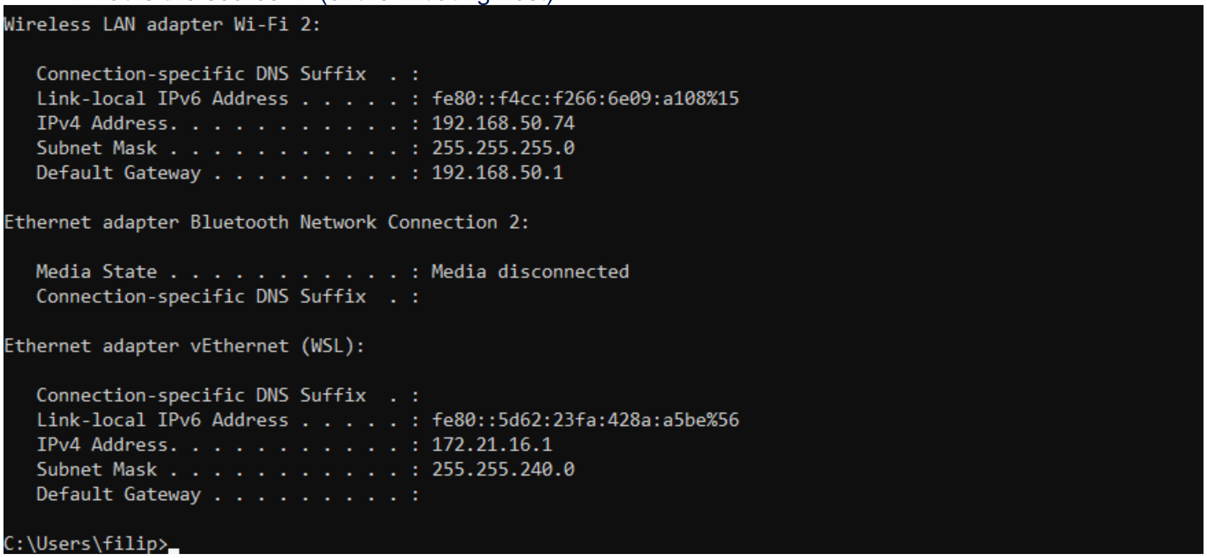
Difficulty: **Hard**

Prerequisite:
Search online and get familiar with the TCP’s three-way handshake. Learn how to capture the three way handshake using Wireshark.

Install Wireshark on your computer and use it to capture traffic against a website or a server or your choice. It is recommended that you capture traffic against a simple website. Name and the IP address of the website you plan to capture traffic:

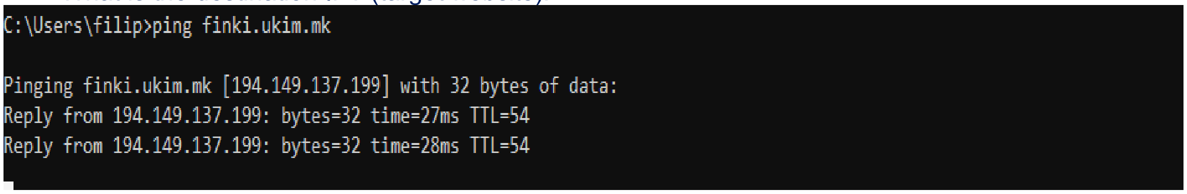
Analyze the TCP's three-way handshake and using screenshots from the Wireshark window answer the questions bellow:

1. What is the source IP (of the initiating host):



```
Wireless LAN adapter Wi-Fi 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f4cc:f266:6e09:a108%15
   IPv4 Address. . . . . . . . . . . : 192.168.50.74
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.50.1

Ethernet adapter Bluetooth Network Connection 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter vEthernet (WSL):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5d62:23fa:428a:a5be%56
   IPv4 Address. . . . . . . . . . . : 172.21.16.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :

C:\Users\filip>
```

2. What is the destination IP? (target website):

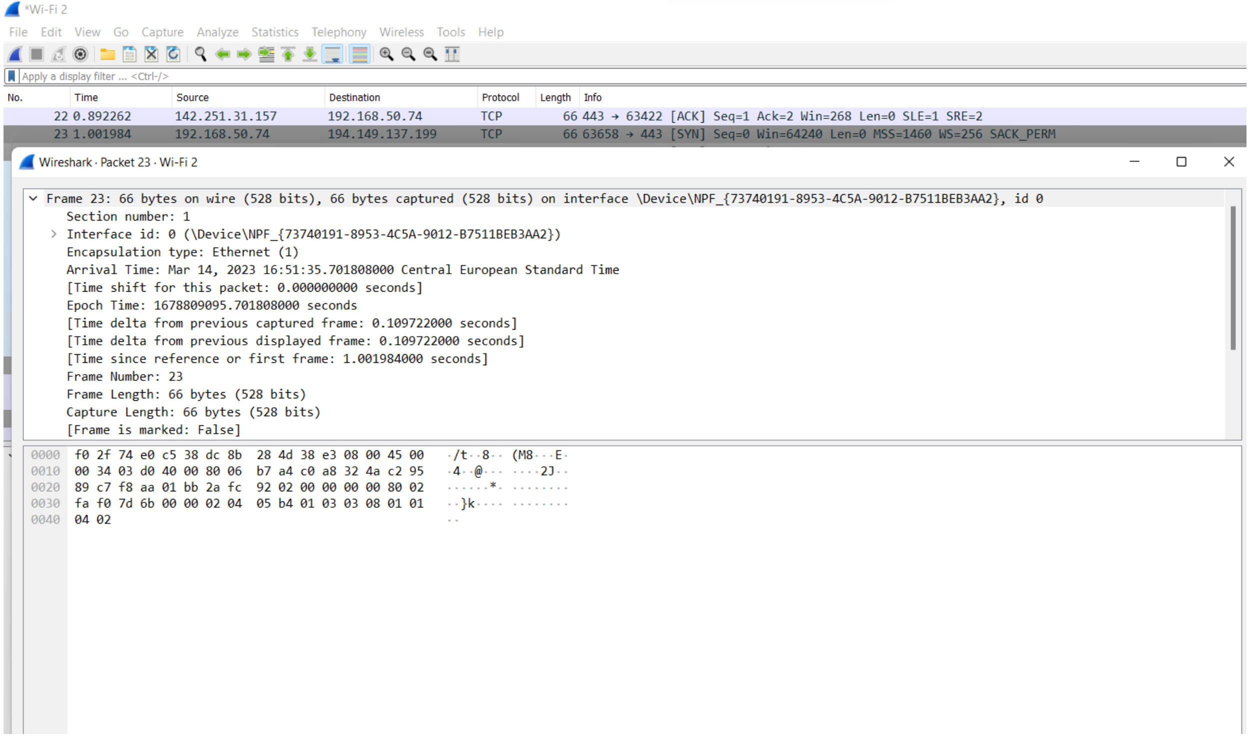


```
C:\Users\filip>ping finki.ukim.mk

Pinging finki.ukim.mk [194.149.137.199] with 32 bytes of data:
Reply from 194.149.137.199: bytes=32 time=27ms TTL=54
Reply from 194.149.137.199: bytes=32 time=28ms TTL=54
```

Identify the Network Interface (Layer 1 & 2) section of the SYN packet and paste a screenshot from it:

<- Paste a screenshot of the Layer 2 details section here →

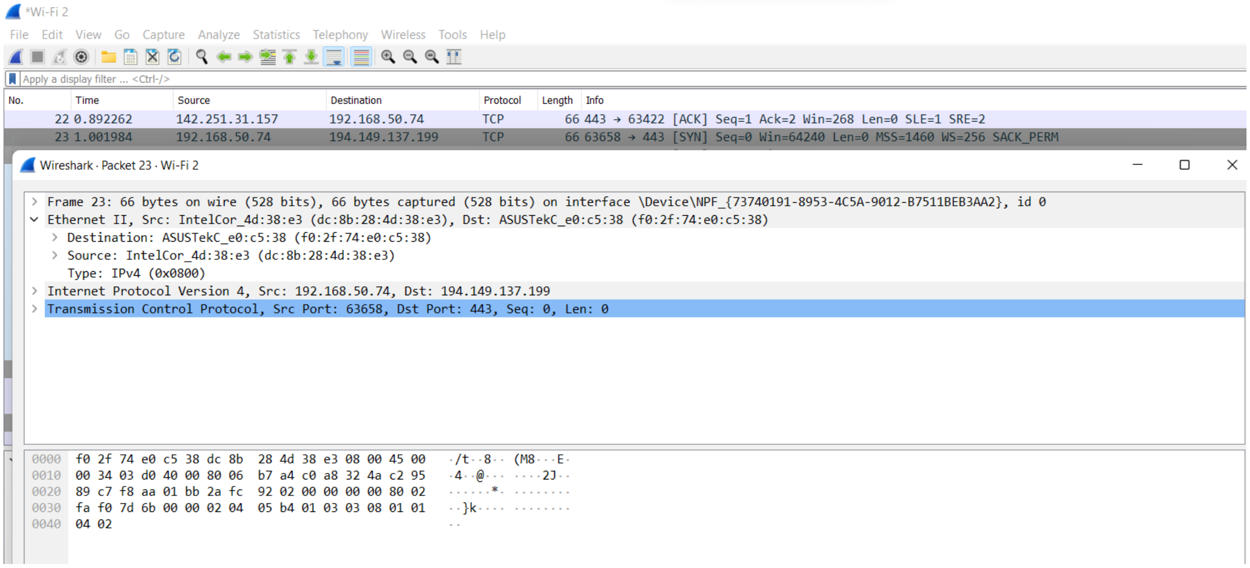
Physical and Datalink


*Wi-Fi 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Wireshark · Packet 23 · Wi-Fi 2

```
No.   Time        Source           Destination       Protocol  Length  Info
   22 0.892262    142.251.31.157   192.168.50.74     TCP         66 443 → 63422 [ACK] Seq=1 Ack=2 Win=268 Len=0 SLE=1 SRE=2
   23 1.001984    192.168.50.74    194.149.137.199   TCP         66 63658 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
```

```
⌄ Frame 23: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{73740191-8953-4C5A-9012-B7511BEB3AA2}, id 0
    Section number: 1
  › Interface id: 0 (\Device\NPF_{73740191-8953-4C5A-9012-B7511BEB3AA2})
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 14, 2023 16:51:35.701808000 Central European Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1678809095.701808000 seconds
    [Time delta from previous captured frame: 0.109722000 seconds]
    [Time delta from previous displayed frame: 0.109722000 seconds]
    [Time since reference or first frame: 1.001984000 seconds]
    Frame Number: 23
    Frame Length: 66 bytes (528 bits)
    Capture Length: 66 bytes (528 bits)
    [Frame is marked: False]
```

```
0000  f0 2f 74 e0 c5 38 dc 8b  28 4d 38 e3 08 00 45 00   ·/t··8·· (M8···E·
0010  00 34 03 d0 40 00 80 06  b7 a4 c0 a8 32 4a c2 95   ·4··@··· ····2J··
0020  89 c7 f8 aa 01 bb 2a fc  92 02 00 00 00 00 80 02   ······*· ········
0030  fa f0 7d 6b 00 00 02 04  05 b4 01 03 03 08 01 01   ··}k···· ········
0040  04 02                                              ··
```


*Wi-Fi 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Wireshark · Packet 23 · Wi-Fi 2

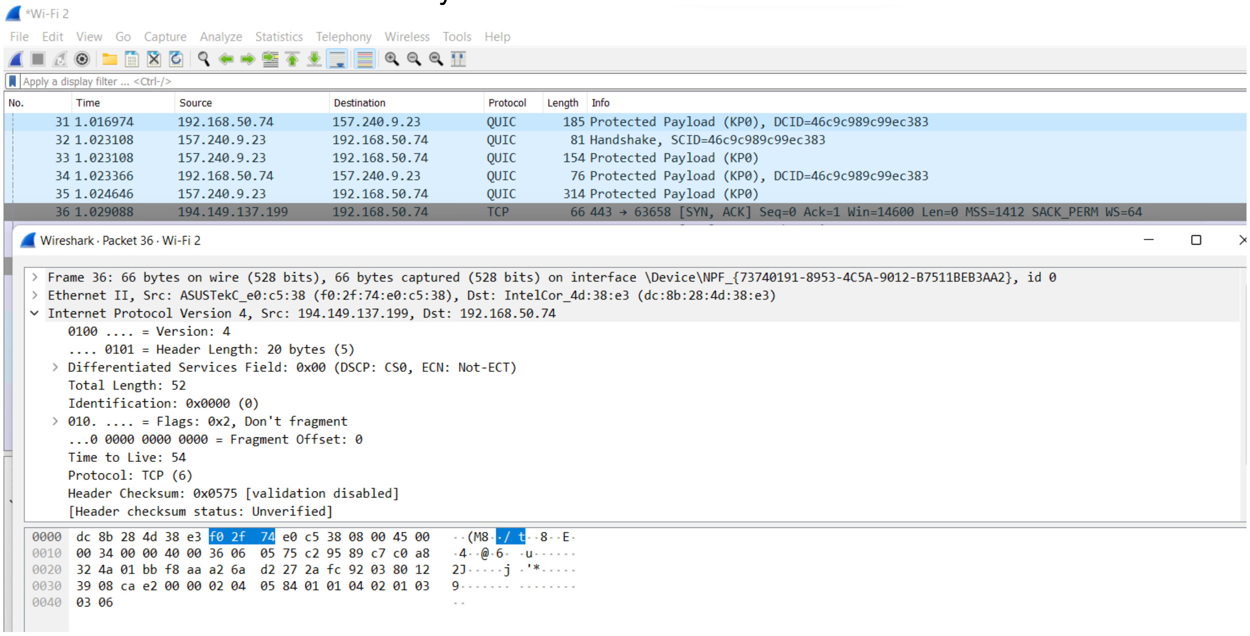
```
No.   Time        Source           Destination       Protocol  Length  Info
   22 0.892262    142.251.31.157   192.168.50.74     TCP         66 443 → 63422 [ACK] Seq=1 Ack=2 Win=268 Len=0 SLE=1 SRE=2
   23 1.001984    192.168.50.74    194.149.137.199   TCP         66 63658 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
```

```
› Frame 23: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{73740191-8953-4C5A-9012-B7511BEB3AA2}, id 0
⌄ Ethernet II, Src: IntelCor_4d:38:e3 (dc:8b:28:4d:38:e3), Dst: ASUSTekC_e0:c5:38 (f0:2f:74:e0:c5:38)
  › Destination: ASUSTekC_e0:c5:38 (f0:2f:74:e0:c5:38)
  › Source: IntelCor_4d:38:e3 (dc:8b:28:4d:38:e3)
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 192.168.50.74, Dst: 194.149.137.199
› Transmission Control Protocol, Src Port: 63658, Dst Port: 443, Seq: 0, Len: 0
```

```
0000  f0 2f 74 e0 c5 38 dc 8b  28 4d 38 e3 08 00 45 00   ·/t··8·· (M8···E·
0010  00 34 03 d0 40 00 80 06  b7 a4 c0 a8 32 4a c2 95   ·4··@··· ····2J··
0020  89 c7 f8 aa 01 bb 2a fc  92 02 00 00 00 00 80 02   ······*· ········
0030  fa f0 7d 6b 00 00 02 04  05 b4 01 03 03 08 01 01   ··}k···· ········
0040  04 02                                              ··
```

Identify the Network Layer 3 section of the SYN/ACK packet and paste a screenshot from it:

<- Paste a screenshot of the Layer 3 details section here → Network

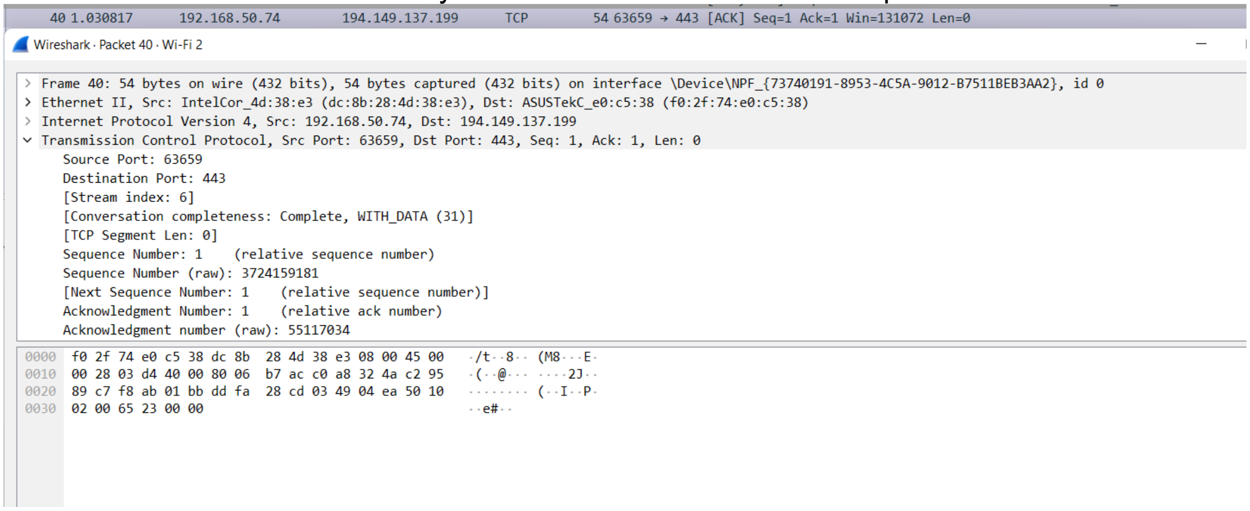

*Wi-Fi 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Wireshark · Packet 36 · Wi-Fi 2

```
No.   Time      Source           Destination     Protocol  Length Info
   31 1.016974  192.168.50.74    157.240.9.23    QUIC       185 Protected Payload (KP0), DCID=46c9c989c99ec383
   32 1.023108  157.240.9.23     192.168.50.74   QUIC        81 Handshake, SCID=46c9c989c99ec383
   33 1.023108  157.240.9.23     192.168.50.74   QUIC       154 Protected Payload (KP0)
   34 1.023366  192.168.50.74    157.240.9.23    QUIC        76 Protected Payload (KP0), DCID=46c9c989c99ec383
   35 1.024646  157.240.9.23     192.168.50.74   QUIC       314 Protected Payload (KP0)
   36 1.029088  194.149.137.199  192.168.50.74   TCP         66 443 → 63658 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1412 SACK_PERM WS=64
```

```
> Frame 36: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{73740191-8953-4C5A-9012-B7511BEB3AA2}, id 0
> Ethernet II, Src: ASUSTekC_e0:c5:38 (f0:2f:74:e0:c5:38), Dst: IntelCor_4d:38:e3 (dc:8b:28:4d:38:e3)
v Internet Protocol Version 4, Src: 194.149.137.199, Dst: 192.168.50.74
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x0000 (0)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 54
    Protocol: TCP (6)
    Header Checksum: 0x0575 [validation disabled]
    [Header checksum status: Unverified]

0000  dc 8b 28 4d 38 e3 f0 2f  74 e0 c5 38 08 00 45 00   ··(M8·./ t··8··E·
0010  00 34 00 00 40 00 36 06  05 75 c2 95 89 c7 c0 a8   ·4··@·6· ·u······
0020  32 4a 01 bb f8 aa a2 6a  d2 27 2a fc 92 03 80 12   2J·····j ·'*·····
0030  39 08 ca e2 00 00 02 04  05 84 01 01 04 02 01 03   9······· ········
0040  03 06                                              ··
```

Identify the Transport Layer 4 section of the ACK packet and paste a screenshot from it bellow:

<- Paste a screenshot of the Layer 4 details section here → Transport



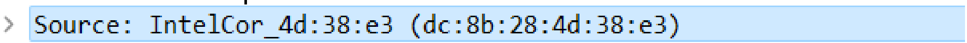
```
   40 1.030817  192.168.50.74    194.149.137.199   TCP        54 63659 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
```

```
> Frame 40: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{73740191-8953-4C5A-9012-B7511BEB3AA2}, id 0
> Ethernet II, Src: IntelCor_4d:38:e3 (dc:8b:28:4d:38:e3), Dst: ASUSTekC_e0:c5:38 (f0:2f:74:e0:c5:38)
> Internet Protocol Version 4, Src: 192.168.50.74, Dst: 194.149.137.199
v Transmission Control Protocol, Src Port: 63659, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
    Source Port: 63659
    Destination Port: 443
    [Stream index: 6]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 3724159181
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 55117034

0000  f0 2f 74 e0 c5 38 dc 8b  28 4d 38 e3 08 00 45 00   ·/t··8·· (M8···E·
0010  00 28 03 d4 40 00 80 06  b7 ac c0 a8 32 4a c2 95   ·(··@··· ····2J··
0020  89 c7 f8 ab 01 bb dd fa  28 cd 03 49 04 ea 50 10   ········ (··I··P·
0030  02 00 65 23 00 00                                  ··e#··
```

Look closely at the L2 section of the three-way handshake packet details. Each of them shows the source and destination MAC address of the packets.

Who is the owner of the destination MAC address of the SYN packet?

MAC address of the recipient device



```
> Source: IntelCor_4d:38:e3 (dc:8b:28:4d:38:e3)
```

Exercise 4 – Hacking mockup (for Bonus points)

Difficulty: **Very hard**

Use Wireshark to capture the packet's application layer data and discover the implications of using unencrypted communication over a network.

It is recommended that you use your own Linux Virtual Machine on your system on which you need to confiture a telnet server.

From your own system try to login with a Telnet on the target VM all while capturing the traffic with a Wireshark. As a proof of competition for this exercise paste in bellow a screenshot of the application layer data containing visible username and password.