HOMEWORK FOR NETWORKING

Exercise 1 – Basic network stuff.

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

```
stan@stan-VirtualBox:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   52:54:00:12:35:02   C                     enp0s3
stan@stan-VirtualBox:~$
```

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

```
stan@stan-VirtualBox:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
stan@stan-VirtualBox:~$
```

Use the traceroute command on your system and observe the hops to Google's DNS, 8.8.8.8.

The traceroute command is used to trace the path that packets take from your system to a destination IP address.

Paste the full output from the command bellow showing all the hops from your system to 8.8.8.8.

```
stan@stan-VirtualBox:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.317 ms  0.291 ms  0.274 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
stan@stan-VirtualBox:~$
```

Why would you need to use the ping command? Answer:

The ping command is used to test connectivity between your system and another device on the network. It sends packets to the specified device and measures the time it takes to receive a response. The ping command is useful for troubleshooting network connectivity issues and verifying that a device is reachable on the network.

Write down the TCP/UDP ports of the most commonly used services bellow in the form of TCP[PORT] or UDP[PORT].

- HTTP - TCP80

- SNMP - UDP161

- HTTPS - TCP443

- DNS client - UDP53

- DNS zone transfer - TCP53

- SMTP - TCP25

- SSH - TCP22

- FTP - TCP21

- Telnet - TCP23

- MSSQL - TCP1433

- MySQL - TCP3306

- PostgreSQL - TCP5432

- RDP (Remote Desktop Protocol) - TCP3389

- NTP - UDP123

- NFS - TCP2049 (UDP can also be used for NFS)

Exercise 2 – TCP/IP Basics.

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 package look like at this stage?

- SRC IP : 100.20.30.10/24
- DST IP : 80.70.60.100/24
- 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 IFWAN. What would the packet look like at this stage?

- SRC IP : 100.20.30.10/24

- DST IP : 80.70.60.100/24
- SRC MAC : BB:BB:BB:11:11:01
- DST MAC : BB:BB:BB:11:11:02

3. RTR2 receives the packet on its IF-WAN interface, prepares it accordingly and forwards it out via IFLAN. What would the packet look like at this stage?

- SRC IP : 100.20.30.10/24
- DST IP : 80.70.60.100/24
- SRC MAC : CC:CC:CC:22:22:02
- DST MAC : CC:CC:CC:22:22:01

4. The web server receives the packet and prepares a response packet back. What would the packet

look like at this stage?

- SRC IP : 80.70.60.100/24
- DST IP : 100.20.30.10/24:
- SRC MAC : DD:DD:DD:77:77:77
- DST MAC : CC:CC:CC:22:22:02

  The most probable transport layer protocol to be used is TCP.

   Since we are talking about web traffic (www), the most probable transport layer protocol that will be used is TCP.

   When the laptop sends the packet, we can expect to see a random high-numbered source port

  50000 and destination port 80 (HTTP) or 443(HTTPS).

   When the web server sends a response packet back, we can expect to see source port 80 (HTTP) or 443(HTTPS) and a random high-numbered destination port.

   There are four broadcast domains in the exhibit shown: one for RTR1's IF-LAN interface and one for RTR2's IF-LAN interface and 2 more.
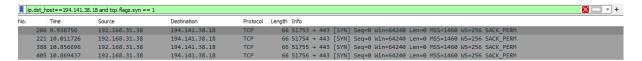
Exercise 3 – Traffic analysis and identifying the OSI layers of the network packets.
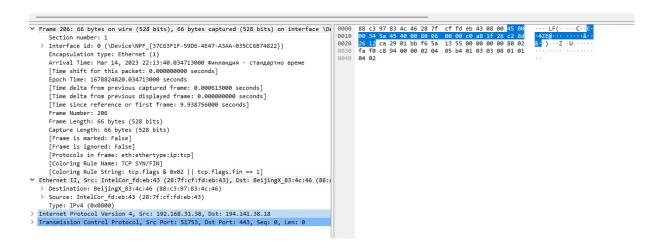
Analyze the TCP's three-way handshake and using screenshots from the Wireshark window answer the questions bellow:

1. What is the source IP (of the initiating host): 192.168.31.38

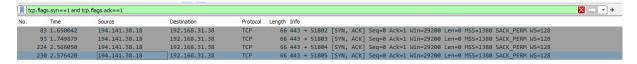**2. What is the destination IP? (target website): 194.141.38.18**

Identify the Network Interface (Layer 1 & 2) section of the SYN packet and paste a screenshot from it:
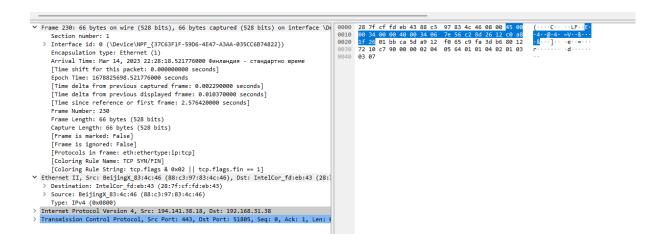
No.	Time	Source	Destination	Protocol	Length	Info
206	9.938756	192.168.31.38	194.141.38.18	TCP	66	51753 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
221	10.011726	192.168.31.38	194.141.38.18	TCP	66	51754 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
388	10.856696	192.168.31.38	194.141.38.18	TCP	66	51755 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
405	10.869437	192.168.31.38	194.141.38.18	TCP	66	51756 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

ip.dst_host==194.141.38.18 and tcp.flags.syn == 1

```
∨ Frame 206: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \De
    Section number: 1
  > Interface id: 0 (\Device\NPF_{37C63F1F-59D6-4E47-A3AA-035CC6B74822})
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 14, 2023 22:13:40.034713000 Финландия - стандартно време
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1678824820.034713000 seconds
    [Time delta from previous captured frame: 0.000613000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 9.938756000 seconds]
    Frame Number: 206
    Frame Length: 66 bytes (528 bits)
    Capture Length: 66 bytes (528 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: TCP SYN/FIN]
    [Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]
∨ Ethernet II, Src: IntelCor_fd:eb:43 (28:7f:cf:fd:eb:43), Dst: BeijingX_83:4c:46 (88:
  > Destination: BeijingX_83:4c:46 (88:c3:97:83:4c:46)
  > Source: IntelCor_fd:eb:43 (28:7f:cf:fd:eb:43)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.31.38, Dst: 194.141.38.18
> Transmission Control Protocol, Src Port: 51753, Dst Port: 443, Seq: 0, Len: 0
```

```
0000  88 c3 97 83 4c 46 28 7f  cf fd eb 43 08 00 45 00   ····LF(· ···C··E·
0010  00 34 5a 45 40 00 80 06  00 00 c0 a8 1f 26 c2 8d   ·4ZE@··· ·····&··
0020  26 12 ca 29 01 bb f6 5a  13 55 00 00 00 00 80 02   &··)···Z ·U······
0030  fa f0 c8 94 00 00 02 04  05 b4 01 03 03 08 01 01   ················
0040  04 02                                              ··
```

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


```
tcp.flags.syn==1 and tcp.flags.ack==1
```

No.	Time	Source	Destination	Protocol	Length	Info
83	1.650042	194.141.38.18	192.168.31.38	TCP	66	443 → 51802 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM WS=128
93	1.749879	194.141.38.18	192.168.31.38	TCP	66	443 → 51803 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM WS=128
224	2.566050	194.141.38.18	192.168.31.38	TCP	66	443 → 51804 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM WS=128
230	2.576420	194.141.38.18	192.168.31.38	TCP	66	443 → 51805 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM WS=128

```
> Frame 230: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \De
    Section number: 1
  > Interface id: 0 (\Device\NPF_{37C63F1F-59D6-4E47-A3AA-035CC6B74822})
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 14, 2023 22:28:18.521776000 Финляндия - стандартно време
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1678825698.521776000 seconds
    [Time delta from previous captured frame: 0.002290000 seconds]
    [Time delta from previous displayed frame: 0.010370000 seconds]
    [Time since reference or first frame: 2.576420000 seconds]
    Frame Number: 230
    Frame Length: 66 bytes (528 bits)
    Capture Length: 66 bytes (528 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: TCP SYN/FIN]
    [Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]
> Ethernet II, Src: BeijingX_83:4c:46 (88:c3:97:83:4c:46), Dst: IntelCor_fd:eb:43 (28:7
  > Destination: IntelCor_fd:eb:43 (28:7f:cf:fd:eb:43)
  > Source: BeijingX_83:4c:46 (88:c3:97:83:4c:46)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 194.141.38.18, Dst: 192.168.31.38
> Transmission Control Protocol, Src Port: 443, Dst Port: 51805, Seq: 0, Ack: 1, Len: (
```

```
0000  28 7f cf fd eb 43 88 c3  97 83 4c 46 08 00 45 00   (····C·· ··LF··E·
0010  00 34 00 00 40 00 34 06  7e 56 c2 8d 26 12 c0 a8   ·4··@·4· ~V··&···
0020  1f 26 01 bb ca 5d a9 12  f0 65 c9 fa 3d b6 80 12   ·&···]·· ·e··=···
0030  72 10 c7 90 00 00 02 04  05 64 01 01 04 02 01 03   r······· ·d·····
0040  03 07                                              ··
```

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


```
ip.dst_host==194.141.38.18 and tcp.flags.ack==1
```

No.	Time	Source	Destination	Protocol	Length	Info
121	2.048290	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=1356 Ack=26241 Win=131072 Len=0
126	2.365498	192.168.31.38	194.141.38.18	TLSv1.2	1120	Application Data
127	2.367892	192.168.31.38	194.141.38.18	TLSv1.2	1031	Application Data
140	2.505764	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=40041 Win=131072 Len=0
145	2.513714	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=45561 Win=131072 Len=0
154	2.514147	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=56601 Win=131072 Len=0
161	2.517184	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=64881 Win=131072 Len=0
166	2.522995	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=70401 Win=131072 Len=0
181	2.523751	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=89721 Win=131072 Len=0
190	2.524254	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=100761 Win=131072 Len=0
195	2.525675	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=106281 Win=131072 Len=0
204	2.526042	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=117321 Win=131072 Len=0
211	2.531552	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=125601 Win=131072 Len=0
217	2.531786	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=132501 Win=131072 Len=0
221	2.532486	192.168.31.38	194.141.38.18	TCP	54	51802 → 443 [ACK] Seq=2422 Ack=136324 Win=131072 Len=0
223	2.557230	192.168.31.38	194.141.38.18	TLSv1.2	1024	Application Data
226	2.566132	192.168.31.38	194.141.38.18	TCP	54	51804 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
227	2.566289	192.168.31.38	194.141.38.18	TLSv1.2	577	Client Hello
231	2.576504	192.168.31.38	194.141.38.18	TCP	54	51805 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
232	2.576890	192.168.31.38	194.141.38.18	TLSv1.2	577	Client Hello
234	2.578908	192.168.31.38	194.141.38.18	TLSv1.2	1170	Application Data
236	2.580321	192.168.31.38	194.141.38.18	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
237	2.580738	192.168.31.38	194.141.38.18	TLSv1.2	1128	Application Data
243	2.583430	192.168.31.38	194.141.38.18	TCP	54	51803 → 443 [ACK] Seq=1552 Ack=7038 Win=131072 Len=0

```
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.31.38
    Destination Address: 194.141.38.18
v Transmission Control Protocol, Src Port: 51802, Dst Port: 443, Seq: 2422, Ack: 566
    Source Port: 51802
    Destination Port: 443
    [Stream index: 0]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 2422    (relative sequence number)
    Sequence Number (raw): 1096289654
    [Next Sequence Number: 2422    (relative sequence number)]
    Acknowledgment Number: 56601    (relative ack number)
    Acknowledgment number (raw): 2459323699
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 512
    [Calculated window size: 131072]
    [Window size scaling factor: 256]
    Checksum: 0xc888 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
```

```
0000  88 c3 97 83 4c 46 28 7f  cf fd eb 43 08 00 45 00   ····LF(· ···C··E·
0010  00 28 5a fc 40 00 80 06  00 00 c0 a8 1f 26 c2 8d   ·(Z·@··· ·····&··
0020  26 12 ca 5a 01 bb 41 58  0d 76 92 96 4d 33 50 10   &··Z··AX ·v··M3P·
0030  02 00 c8 88 00 00                                  ······
```

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.

No.	Time	Source	Destination	Protocol	Length	Info
121	2.048290	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
126	2.365498	192.168.31.38	194.141.38.18	TLSv1.2	1120	Applicati
127	2.367892	192.168.31.38	194.141.38.18	TLSv1.2	1031	Applicati
140	2.505764	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
145	2.513714	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
154	2.514147	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
161	2.517184	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
166	2.522995	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
181	2.523751	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
190	2.524254	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
195	2.525675	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
204	2.526042	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
211	2.531552	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
217	2.531786	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
221	2.532486	192.168.31.38	194.141.38.18	TCP	54	51802 → 4
223	2.557230	192.168.31.38	194.141.38.18	TLSv1.2	1024	Applicati
226	2.566132	192.168.31.38	194.141.38.18	TCP	54	51804 → 4
227	2.566289	192.168.31.38	194.141.38.18	TLSv1.2	577	Client He
231	2.576504	192.168.31.38	194.141.38.18	TCP	54	51805 → 4
232	2.576890	192.168.31.38	194.141.38.18	TLSv1.2	577	Client He
234	2.578908	192.168.31.38	194.141.38.18	TLSv1.2	1170	Applicati
236	2.580321	192.168.31.38	194.141.38.18	TLSv1.2	105	Change Ci
237	2.580738	192.168.31.38	194.141.38.18	TLSv1.2	1128	Applicati
243	2.583430	192.168.31.38	194.141.38.18	TCP	54	51803 → 4

```
> Frame 154: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \De
∨ Ethernet II, Src: IntelCor_fd:eb:43 (28:7f:cf:fd:eb:43), Dst: BeijingX_83:4c:46 (88:
    > Destination: BeijingX_83:4c:46 (88:c3:97:83:4c:46)
    > Source: IntelCor_fd:eb:43 (28:7f:cf:fd:eb:43)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.31.38, Dst: 194.141.38.18
> Transmission Control Protocol, Src Port: 51802, Dst Port: 443, Seq: 2422, Ack: 56601
```

Filter: `ip.dst_host==194.141.38.18 and tcp.flags.ack==1`

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

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : lan
   Description . . . . . . . . . . . : Intel(R) Wireless-AC 9560 160MHz
   Physical Address. . . . . . . . . : 28-7F-CF-FD-EB-43
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::ec9:3505:304b:92d9%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.31.38(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 14 март 2023 г. 18:58:07
   Lease Expires . . . . . . . . . . : 15 март 2023 г. 6:58:07
   Default Gateway . . . . . . . . . : 192.168.31.1
   DHCP Server . . . . . . . . . . . : 192.168.31.1
   DHCPv6 IAID . . . . . . . . . . . : 103317455
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-65-8D-8C-00-68-EB-7B-E0-22
   DNS Servers . . . . . . . . . . . : 192.168.31.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Exercise 4 – Hacking mockup (for Bonus points).

Use Wireshark to capture the packet's application layer data and discover the implications

of using unencrypted communication over a network.

As a proof of competition for this exercise paste in bellow a screenshot of the application layer data containing visible username and password.

Using TELNET:

Using HTTP: