

## Exercise 1 – Basic network stuff

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

arp - stands for Address resolution protocol is used to resolve ip address to to MAC addresses. MAC address is physical address of a device. Whenever a device needs to communicate with another device on a local area network it needs MAC address for that device. arp stores MAC addresses in his local cache ARP cache. ARP Table is used to keep the record of the IP address and MAC address of the devices source and destination device for the communication between two devices

ds@ds-HP-ProBook-440-G6:~\$ arp -a

```
? (192.168.100.236) at 00:21:b7:2f:bf:4f [ether] on enp2s0
? (192.168.100.149) at 00:21:b7:7e:d1:11 [ether] on enp2s0
? (192.168.100.137) at 00:21:b7:d5:f2:e1 [ether] on enp2s0
? (192.168.100.141) at 00:21:b7:d5:f2:b1 [ether] on enp2s0
70-100-168-192.ds11-erie.roc.ny.frontiernet.net (192.168.100.70) at 58:20:b1:4e:bc:23 [ether] on enp2s0
? (192.168.100.231) at e0:70:ea:f9:2c:10 [ether] on enp2s0
node-81s.pool-1-1.dynamic.totinternet.net (192.168.1.1) at 00:1f:33:28:81:80 [ether] on wlp0s20f3
? (192.168.100.147) at 00:21:b7:e5:37:ce [ether] on enp2s0
? (192.168.100.1) at 04:76:b0:26:5a:74 [ether] on enp2s0
nothing.attdns.com (192.168.100.135) at e0:70:ea:f9:2c:5a [ether] on enp2s0
? (192.168.100.44) at e0:70:ea:f9:2c:b4 [ether] on enp2s0
ETH-240-ML3471ND.kultur.uni-hamburg.de (192.168.100.134) at 00:21:b7:d5:f6:c6 [ether] on enp2s0
```

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

```
**ip route**
```

```
default via 192.168.100.1 dev enp2s0 proto static metric 100
default via 192.168.1.1 dev wlp0s20f3 proto dhcp metric 600
169.254.0.0/16 dev wlp0s20f3 scope link metric 1000
192.168.1.0/24 dev wlp0s20f3 proto kernel scope link src 192.168.1.2 metric 600
192.168.100.0/24 dev enp2s0 proto kernel scope link src 192.168.100.33 metric 100
```

- 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.

- o traceroute is command line

### traceroute 8.8.8.8

```
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 192.168.100.1 (192.168.100.1)  1.097 ms  1.012 ms  0.960 ms
 2 192.168.22.137 (192.168.22.137)  4.066 ms  4.105 ms  4.790 ms
 3 62.162.85.14 (62.162.85.14)  3.421 ms  3.759 ms  3.746 ms
 4 62.162.85.13 (62.162.85.13)  6.402 ms  6.323 ms  2.673 ms
 5 host-41.152.158.95.etisalat.com.eg (95.158.152.41)  4.961 ms  4.911 ms  *
 6 95.158.188.213 (95.158.188.213)  7.625 ms  9.867 ms  10.643 ms
 7 ecs-1-92-251-142.compute.hwclouds-dns.com (142.251.92.1)  5.136 ms bras-base-mtrlpq427b
 8 209.85.243.245 (209.85.243.245)  4.727 ms 142.250.60.187 (142.250.60.187)  8.121 ms 108
 9 dns.google (8.8.8.8)  8.011 ms  7.957 ms  7.904 ms
```


- Why would you need to use the ping command?

Answer:

ping command is used to check connectivity between two hosts.

```
ds@ds-HP-ProBook-440-G6:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
 64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=7.82 ms
 64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=8.13 ms
 64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=7.75 ms
 64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=7.76 ms
 64 bytes from 8.8.8.8: icmp_seq=5 ttl=116 time=7.72 ms
 64 bytes from 8.8.8.8: icmp_seq=6 ttl=116 time=7.79 ms
 64 bytes from 8.8.8.8: icmp_seq=7 ttl=116 time=7.49 ms
```

## Exercise 2 – TCP/IP Basics

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.

1. The laptop initiates communication with the web server and prepares a packet. What

```

SRC IP      100.20.30.10
DST IP      80.70.60.100
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 80.70.60.100  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 80.70.60.100  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 80.70.60.100  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

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: • DST PORT: 80 / 443

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

- SRC PORT: 80 / 443 • DST PORT:

=====

- 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 443 port
- DNS client - DNS uses port 53
- DNS zone transfer - use TCP port 53
- SMTP
- SSH -os using 22
- FTP - os using 21
- Telnet - 23 or 2323
- MSSQL - 1433
- MySQL - 3306
- PostgreSQL - 5432
- RDP (Remote Desktop Protocol) - 3389
- NTP port 123 is used for NTP server communication and NTP clients use port 1023
- NFS - 2049

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

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:

\*enp2s0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

not arp and not udp and not ssdp and not quic

No.	Time	Source	Destination	Protocol	Length	Info
22	0.75099243	192.168.100.33	20.199.120.182	TCP	74	55580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2387088436 TSecr=0 WS=128
23	0.799357575	20.199.120.182	192.168.100.33	TCP	66	443 → 55580 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=1 SACK_PERM=1
24	0.799493178	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
25	0.800240297	192.168.100.33	20.199.120.182	TLSPv1.2	571	Client Hello
28	0.847512375	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
29	0.847619146	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
30	0.847645862	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1461 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
31	0.847676051	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=2921 Win=63488 Len=0
32	0.847736339	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=2921 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
33	0.847753289	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=4381 Win=62592 Len=0
34	0.847866391	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=4381 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
35	0.847877135	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5841 Win=64128 Len=0
36	0.866823435	fe80::9084:c1b2:81a...ff02::1:fff6f:5387	ff02::1:fff6f:5387	ICMPv6	86	Neighbor Solicitation for fe80::3664:a9ff:fe6f:5387 from b8:ac:6f:30:18:bb
38	0.887812908	20.199.120.182	192.168.100.33	TLSPv1.2	141	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
39	0.887873877	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5928 Win=64128 Len=0
40	0.893678577	192.168.100.33	20.199.120.182	TLSPv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
42	0.93595463	20.199.120.182	192.168.100.33	TLSPv1.2	105	Change Cipher Spec, Encrypted Handshake Message
43	0.938136641	192.168.100.33	20.199.120.182	TLSPv1.2	329	Application Data

Frame 22: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp2s0, id 0  
 Ethernet II, Src: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d), Dst: Cisco\_26:5a:74 (04:76:b0:26:5a:74)  
 Internet Protocol Version 4, Src: 192.168.100.33, Dst: 20.199.120.182  
 Transmission Control Protocol, Src Port: 55580, Dst Port: 443, Seq: 0, Len: 0

\*enp2s0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

not arp and not udp and not ssdp and not quic

No.	Time	Source	Destination	Protocol	Length	Info
22	0.75099243	192.168.100.33	20.199.120.182	TCP	74	55580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2387088436 TSecr=0 WS=128
23	0.799357575	20.199.120.182	192.168.100.33	TCP	66	443 → 55580 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=1 SACK_PERM=1
24	0.799493178	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
25	0.800240297	192.168.100.33	20.199.120.182	TLSPv1.2	571	Client Hello
28	0.847512375	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
29	0.847619146	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
30	0.847645862	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1461 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
31	0.847676051	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=2921 Win=63488 Len=0
32	0.847736339	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=2921 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
33	0.847753289	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=4381 Win=62592 Len=0
34	0.847866391	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=4381 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
35	0.847877135	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5841 Win=64128 Len=0
36	0.866823435	fe80::9084:c1b2:81a...ff02::1:fff6f:5387	ff02::1:fff6f:5387	ICMPv6	86	Neighbor Solicitation for fe80::3664:a9ff:fe6f:5387 from b8:ac:6f:30:18:bb
38	0.887812908	20.199.120.182	192.168.100.33	TLSPv1.2	141	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
39	0.887873877	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5928 Win=64128 Len=0
40	0.893678577	192.168.100.33	20.199.120.182	TLSPv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
42	0.93595463	20.199.120.182	192.168.100.33	TLSPv1.2	105	Change Cipher Spec, Encrypted Handshake Message
43	0.938136641	192.168.100.33	20.199.120.182	TLSPv1.2	329	Application Data

Frame 23: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp2s0, id 0  
 Ethernet II, Src: Cisco\_26:5a:74 (04:76:b0:26:5a:74), Dst: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
 Internet Protocol Version 4, Src: 20.199.120.182, Dst: 192.168.100.33  
 Transmission Control Protocol, Src Port: 55580, Seq: 0, Ack: 1, Len: 0

\*enp2s0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

not arp and not udp and not ssdp and not quic

No.	Time	Source	Destination	Protocol	Length	Info
22	0.75099243	192.168.100.33	20.199.120.182	TCP	74	55580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2387088436 TSecr=0 WS=128
23	0.799357575	20.199.120.182	192.168.100.33	TCP	66	443 → 55580 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=1 SACK_PERM=1
24	0.799493178	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
25	0.800240297	192.168.100.33	20.199.120.182	TLSPv1.2	571	Client Hello
28	0.847512375	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
29	0.847619146	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
30	0.847645862	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1461 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
31	0.847676051	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=2921 Win=63488 Len=0
32	0.847736339	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=2921 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
33	0.847753289	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=4381 Win=62592 Len=0
34	0.847866391	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=4381 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
35	0.847877135	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5841 Win=64128 Len=0
36	0.866823435	fe80::9084:c1b2:81a...ff02::1:fff6f:5387	ff02::1:fff6f:5387	ICMPv6	86	Neighbor Solicitation for fe80::3664:a9ff:fe6f:5387 from b8:ac:6f:30:18:bb
38	0.887812908	20.199.120.182	192.168.100.33	TLSPv1.2	141	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
39	0.887873877	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5928 Win=64128 Len=0
40	0.893678577	192.168.100.33	20.199.120.182	TLSPv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
42	0.93595463	20.199.120.182	192.168.100.33	TLSPv1.2	105	Change Cipher Spec, Encrypted Handshake Message
43	0.938136641	192.168.100.33	20.199.120.182	TLSPv1.2	329	Application Data

Frame 24: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp2s0, id 0  
 Ethernet II, Src: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d), Dst: Cisco\_26:5a:74 (04:76:b0:26:5a:74)  
 Internet Protocol Version 4, Src: 192.168.100.33, Dst: 20.199.120.182  
 Transmission Control Protocol, Src Port: 55580, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

1. What is the source IP (of the initiating host): 192.168.100.33
2. What is the destination IP? (target website): 20.199.120.182

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

The image displays a Wireshark packet capture analysis. The packet list on the left shows a series of packets, with the selected packet (No. 22) being a TCP segment. The packet details pane on the right shows the structure of the selected packet, including the Ethernet II header, Internet Protocol Version 4 header, and Transmission Control Protocol header. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
22	0.06099243	192.168.100.33	20.199.120.182	TCP	74	55580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2387088436 TSecr=0 WS=128
23	0.06193178	192.168.100.33	192.168.100.33	TCP	60	443 → 55580 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1440 WS=1 SACK_PERM=1
24	0.79949318	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
25	0.800240297	192.168.100.33	20.199.120.182	TLV51.2	571	Client Hello
28	0.847512375	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
29	0.847619146	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
30	0.847645862	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1461 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
31	0.847676951	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=2921 Win=63488 Len=0
32	0.847736339	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=2921 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
33	0.847753289	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=4381 Win=62592 Len=0
34	0.847866391	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=4381 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
35	0.847877135	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5841 Win=64128 Len=0
36	0.866823435	fe80::9084:c1b2:81a...ff02::1:ffef:5387	fe80::3664:a9ff:feef:5387	ICMPv6	86	Neighbor Solicitation for fe80::3664:a9ff:feef:5387 from b8:ac:6f:30:18:bb
38	0.887812908	20.199.120.182	192.168.100.33	TLV51.2	141	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
39	0.887873877	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5928 Win=64128 Len=0
40	0.893678577	192.168.100.33	20.199.120.182	TLV51.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
42	0.935995463	20.199.120.182	192.168.100.33	TLV51.2	185	Change Cipher Spec, Encrypted Handshake Message
43	0.938136641	192.168.100.33	20.199.120.182	TLV51.2	329	Application Data

Frame 22: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp2s0, id 0  
 Ethernet II, Src: HewlettP\_62:77:0d (b8:0c:d1:62:77:0d), Dst: Cisco\_26:5a:74 (04:76:b0:26:5a:74)  
 Destination: Cisco\_26:5a:74 (04:76:b0:26:5a:74)  
 Source: HewlettP\_62:77:0d (b8:0c:d1:62:77:0d)  
 Address: HewlettP\_62:77:0d (b8:0c:d1:62:77:0d)  
 ....0. .... : LG bit: Globally unique address (factory default)  
 ....0. .... : IG bit: Individual address (unicast)  
 Type: IPv4 (0x0000)  
 Internet Protocol Version 4, Src: 192.168.100.33, Dst: 20.199.120.182  
 Transmission Control Protocol, Src Port: 55580, Dst Port: 443, Seq: 0, Len: 0

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

**[not arp and not udp and not sctp and not icmp]**

No.	Time	Source	Destination	Protocol	Length	Info
0	0.267283869	192.168.100.34	192.168.100.34	TCP	60	60 60 60:60:16:21:45:3e IN/16_128
22	0.750829243	192.168.100.33	192.168.100.182	TCP	74	55580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2387088436 TSecr=0 WS=128
24	0.799493178	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
25	0.806240297	192.168.100.33	20.199.120.182	TLsv1.2	571	Client Hello
26	0.847512375	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
29	0.847619146	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
30	0.857645862	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1461 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
31	0.847676951	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=2921 Win=63488 Len=0
32	0.847736339	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=2921 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
33	0.847753289	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=4381 Win=62592 Len=0
34	0.847860391	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=4381 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
35	0.847877135	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5641 Win=64128 Len=0
36	0.862834355	f680::9084:c1b2:81a:f02:1::ffff:5387	f680::9084:c1b2:81a:f02:1::ffff:5387	ICMPv6	86	Neighbor Solicitation for f680::3064:aaff:feef:5387 from b8:ac:6f:30:18:bb
38	0.887812908	20.199.120.182	192.168.100.33	TLsv1.2	141	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
39	0.887873877	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5928 Win=64128 Len=0
40	0.893678577	192.168.100.33	20.199.120.182	TLsv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
42	0.935095463	20.199.120.182	192.168.100.33	TLsv1.2	185	Change Cipher Spec, Encrypted Handshake Message
43	0.938136641	192.168.100.33	20.199.120.182	TLsv1.2	329	Application Data
46	0.980723535	20.199.120.182	192.168.100.33	TLsv1.2	333	Application Data
47	0.982214728	192.168.100.33	20.199.120.182	TLsv1.2	288	Application Data
50	1.022717170	20.199.120.182	192.168.100.33	TLsv1.2	157	Application Data
51	1.023882534	192.168.100.33	20.199.120.182	TLsv1.2	538	Application Data
54	1.104912134	Cisco 20:5a:08	CDP/VTP/DTP/PagP/UDLD..DTP		60	Dynamic Trunk Protocol
55	1.126510673	20.199.120.182	192.168.100.33	TCP	60	443 → 55580 [ACK] Seq=6361 Ack=1589 Win=8192 Len=0
61	1.428847535	20.199.120.182	192.168.100.33	TLsv1.2	407	Application Data
62	1.430472922	192.168.100.33	20.199.120.182	TLsv1.2	277	Application Data

Source: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
Address: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
.....0 = LG bit: Globally unique address (factory default)  
.....0 = IG bit: Individual address (unicast)  
Type: IPv4 (0x000)

Internet Protocol Version 4, Src: 192.168.100.33, Dst: 20.199.120.182

Transmission Control Protocol, Src Port: 55580, Dst Port: 443, Seq #: 0, Len: 0  
Source Port: 55580  
Destination Port: 443  
[Stream index: 0]  
[Conversation completeness: Incomplete, DATA (15)]  
[TCP Segment Len: 0]  
Sequence Number: 0 (relative sequence number)  
Sequence Number (raw): 1208980315  
[Next Sequence Number: 1 (relative sequence number)]  
Acknowledgment Number: 0  
[Window size: 0]  
[Options: none]

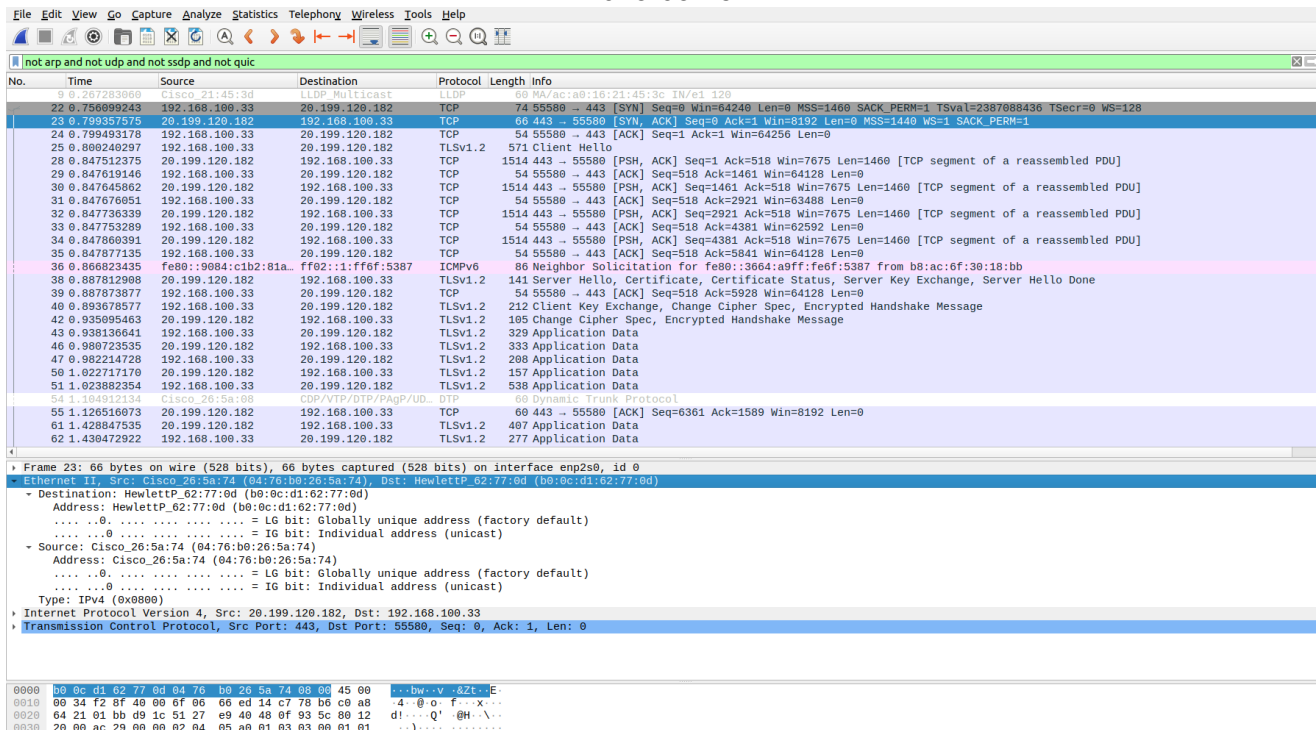
```

0020  75 b6 d9 1c 01 bb 48 0f 93 5b 00 00 00 a0 a0  x-----H-[.....
0030  fa f0 b2 75 00 00 02 04 05 b4 04 02 08 0a 8e 48  -u-----H[.....
0040  14 34 00 00 00 00 01 03 03 07                                     .-----

```

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

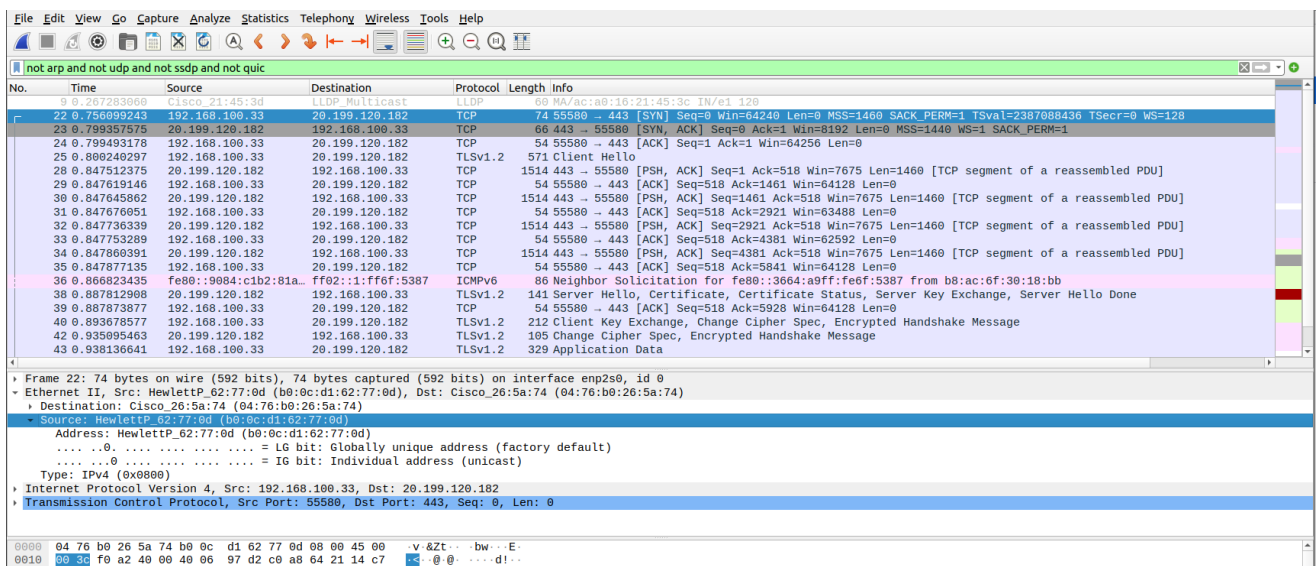




No.	Time	Source	Destination	Protocol	Length	Info
22	0.75699243	192.168.100.33	192.168.100.33	TCP	74	55580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2387088436 TSecr=0 WS=128
23	0.799357575	20.199.120.182	192.168.100.33	TCP	66	443 → 55580 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=1 SACK_PERM=1
24	0.799493178	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
25	0.800240297	192.168.100.33	20.199.120.182	TLSv1.2	571	Client Hello
26	0.847512375	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
27	0.847619146	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
28	0.847645862	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1461 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
29	0.847676951	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=2921 Win=63488 Len=0
30	0.847736339	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=2921 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
31	0.847753289	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=4381 Win=62592 Len=0
32	0.847860391	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=4381 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
33	0.847877135	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5841 Win=64128 Len=0
34	0.866823435	fe80::9084:c1b2:81a...ff02::1:ff6f:5387	ff02::1:ff6f:5387	ICMPv6	86	Neighbor Solicitation for fe80::3664:a9ff:fe6f:5387 from b8:ac:6f:30:18:bb
35	0.887812908	20.199.120.182	192.168.100.33	TLSv1.2	141	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
36	0.887873877	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5928 Win=64128 Len=0
37	0.893678577	192.168.100.33	20.199.120.182	TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
38	0.935955463	20.199.120.182	192.168.100.33	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
39	0.938136641	192.168.100.33	20.199.120.182	TLSv1.2	329	Application Data
40	0.980723535	20.199.120.182	192.168.100.33	TLSv1.2	333	Application Data
41	0.982214728	192.168.100.33	20.199.120.182	TLSv1.2	208	Application Data
42	0.102271710	20.199.120.182	192.168.100.33	TLSv1.2	157	Application Data
43	0.102388254	192.168.100.33	20.199.120.182	TLSv1.2	538	Application Data
44	0.118433333	20.199.120.182	192.168.100.33	TCP	60	443 → 55580 [ACK] Seq=6361 Ack=1589 Win=8192 Len=0
45	0.126516073	20.199.120.182	192.168.100.33	TLSv1.2	407	Application Data
46	0.142884753	20.199.120.182	192.168.100.33	TLSv1.2	277	Application Data
47	0.143847292	192.168.100.33	20.199.120.182	TLSv1.2	277	Application Data

Frame 23: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp2s0, id 0  
Ethernet II, Src: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d), Dst: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
Destination: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
Address: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
Type: IPv4 (0x0800)  
Internet Protocol Version 4, Src: 20.199.120.182, Dst: 192.168.100.33  
Transmission Control Protocol, Src Port: 443, Dst Port: 55580, Seq: 0, Ack: 1, Len: 0

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



No.	Time	Source	Destination	Protocol	Length	Info
22	0.75699243	192.168.100.33	192.168.100.33	TCP	74	55580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2387088436 TSecr=0 WS=128
23	0.799357575	20.199.120.182	192.168.100.33	TCP	66	443 → 55580 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=1 SACK_PERM=1
24	0.799493178	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
25	0.800240297	192.168.100.33	20.199.120.182	TLSv1.2	571	Client Hello
26	0.847512375	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
27	0.847619146	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
28	0.847645862	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=1461 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
29	0.847676951	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=2921 Win=63488 Len=0
30	0.847736339	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=2921 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
31	0.847753289	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=4381 Win=62592 Len=0
32	0.847860391	20.199.120.182	192.168.100.33	TCP	1514	443 → 55580 [PSH, ACK] Seq=4381 Ack=518 Win=7675 Len=1460 [TCP segment of a reassembled PDU]
33	0.847877135	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5841 Win=64128 Len=0
34	0.866823435	fe80::9084:c1b2:81a...ff02::1:ff6f:5387	ff02::1:ff6f:5387	ICMPv6	86	Neighbor Solicitation for fe80::3664:a9ff:fe6f:5387 from b8:ac:6f:30:18:bb
35	0.887812908	20.199.120.182	192.168.100.33	TLSv1.2	141	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
36	0.887873877	192.168.100.33	20.199.120.182	TCP	54	55580 → 443 [ACK] Seq=518 Ack=5928 Win=64128 Len=0
37	0.893678577	192.168.100.33	20.199.120.182	TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
38	0.935955463	20.199.120.182	192.168.100.33	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
39	0.938136641	192.168.100.33	20.199.120.182	TLSv1.2	329	Application Data

Frame 22: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp2s0, id 0  
Ethernet II, Src: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d), Dst: Cisco\_26:5a:74 (04:76:b0:26:5a:74)  
Destination: Cisco\_26:5a:74 (04:76:b0:26:5a:74)  
Source: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
Address: HewlettP\_62:77:0d (b0:0c:d1:62:77:0d)  
Type: IPv4 (0x0800)  
Internet Protocol Version 4, Src: 192.168.100.33, Dst: 20.199.120.182  
Transmission Control Protocol, Src Port: 55580, Dst Port: 443, Seq: 0, Len: 0

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?

## 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. It is recommended that you use your own Linux Virtual Machine on your system on which you need to configure 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.

