

Students:

Vasylyshyn Danylo 256711

Nykonchuk Illia 245693

We have presented the solution for the first two tasks on the last lesson (we are group 1), so here are solutions for task 3 and 4:

Task 3:

My ip: 156.17.237.32

Ethernet adapter Ethernet:

```
Connection-specific DNS Suffix . : 5-6.t19.ds.pwr.wroc.pl
Link-local IPv6 Address . . . . . : fe80::8cc2:e86d:d0d4:1ef9%17
IPv4 Address. . . . . : 156.17.237.32
Subnet Mask . . . . . : 255.255.255.128
Default Gateway . . . . . : 156.17.237.126
```

Partners ip: 156.17.237.52

Адаптер Ethernet Ethernet:

```
DNS-суффикс подключения . . . . . : 5-6.t19.ds.pwr.wroc.pl
Локальный IPv6-адрес канала . . . : fe80::68c8:f357:3962:e72c%17
IPv4-адрес. . . . . : 156.17.237.52
Маска подсети . . . . . : 255.255.255.128
Основной шлюз. . . . . : 156.17.237.126
```

- When looking into the arp table I can find the address of my partner there:

```
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: 192.168.56.1 --- 0x10

Internet Address	Physical Address	Type
192.168.56.255	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

Interface: 156.17.237.32 --- 0x11

Internet Address	Physical Address	Type
156.17.237.33	98-da-c4-2b-cb-23	dynamic
156.17.237.52	88-d7-f6-1e-12-2b	dynamic
156.17.237.53	74-da-88-32-7b-df	dynamic
156.17.237.75	20-25-64-87-84-84	dynamic
156.17.237.109	18-d6-c7-ec-d2-e5	dynamic
156.17.237.126	00-15-f9-7b-2f-c7	dynamic
156.17.237.127	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

C:\Users\danyl>

- Now cleaning the arp table and viewing it shows:

```
C:\Windows\system32>arp -d *

C:\Windows\system32>arp -a

Interface: 25.30.154.246 --- 0xd
    Internet Address      Physical Address        Type
    224.0.0.22            01-00-5e-00-00-16      static

Interface: 192.168.56.1 --- 0x10
    Internet Address      Physical Address        Type
    224.0.0.22            01-00-5e-00-00-16      static

Interface: 156.17.237.32 --- 0x11
    Internet Address      Physical Address        Type
    156.17.237.126        00-15-f9-7b-2f-c7      dynamic
    224.0.0.22            01-00-5e-00-00-16      static

C:\Windows\system32>
```

It's much shorter

- Pinging goes just fine:

```
C:\Windows\system32>ping 156.17.237.32

Pinging 156.17.237.32 with 32 bytes of data:
Reply from 156.17.237.32: bytes=32 time<1ms TTL=128
Reply from 156.17.237.32: bytes=32 time<1ms TTL=128
Reply from 156.17.237.32: bytes=32 time<1ms TTL=128
Reply from 156.17.237.32: bytes=32 time<1ms TTL=128

Ping statistics for 156.17.237.32:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\system32>
```

But if we clean the table again and capture the packets in wireshark we will see:

arp						
No.	Time	Source	Destination	Protocol	Length	Info
1588	11.869944	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.42? Tell 156.17.237.126
949	7.044949	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.43? Tell 156.17.237.126
1009	7.445390	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.45? Tell 156.17.237.126
430	3.581936	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.48? Tell 156.17.237.126
1385	10.271898	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.49? Tell 156.17.237.126
1840	13.929912	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.49? Tell 156.17.237.126
1721	12.967019	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.4? Tell 156.17.237.126
1101	8.151507	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.51? Tell 156.17.237.126
930	6.995512	ASUSTekC_4e:c3:77	Broadcast	ARP	42	Who has 156.17.237.52? Tell 156.17.237.32
596	4.583386	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.54? Tell 156.17.237.126
2081	15.829306	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.61? Tell 156.17.237.126
100	0.925917	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.63? Tell 156.17.237.126

> Frame 930: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{C2BC553A-428C-4D0B-9C70-B838291}

> Ethernet II, Src: ASUSTekC_4e:c3:77 (b0:6e:bf:4e:c3:77), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Address Resolution Protocol (request)

The packet that I selected is an ARP packet sent from my computer.

This is used to find the host's nic's MAC address when knowing the ip. The packet is first send to broadcast (everybody), and when the needed host recieves this packet it replies directly to the initial computer(my computer), as here:

arp						
No.	Time	Source	Destination	Protocol	Length	Info
1433	10.659448	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.15? Tell 156.17.237.126
1637	12.311632	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.16? Tell 156.17.237.126
1232	9.135691	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.17? Tell 156.17.237.126
1006	7.412888	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.21? Tell 156.17.237.126
829	6.320065	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.23? Tell 156.17.237.126
1139	8.386668	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.28? Tell 156.17.237.126
888	6.704909	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.30? Tell 156.17.237.126
1442	10.716604	ASUSTekC_1e:12:2b	ASUSTekC_4e:c3:77	ARP	60	Who has 156.17.237.32? Tell 156.17.237.52
2110	16.096720	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.34? Tell 156.17.237.126
1763	13.315161	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.36? Tell 156.17.237.126
1588	11.869944	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.42? Tell 156.17.237.126
949	7.044949	Cisco_7b:2f:c7	Broadcast	ARP	60	Who has 156.17.237.43? Tell 156.17.237.126

> Frame 1442: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{C2BC553A-428C-4D0B-9C70-B838291D2C69}, id 0

> Ethernet II, Src: ASUSTekC_1e:12:2b (88:d7:f6:1e:12:2b), Dst: ASUSTekC_4e:c3:77 (b0:6e:bf:4e:c3:77)

> Destination: ASUSTekC_4e:c3:77 (b0:6e:bf:4e:c3:77)

> Source: ASUSTekC_1e:12:2b (88:d7:f6:1e:12:2b)

> Type: ARP (0x0806)

> Padding: 00000000000000000000000000000000

> Address Resolution Protocol (request)

The selected packet is a reply in which the other host shares it's MAC address. We can see from the ethernet header fields the MAC addresses of the two machines:

My mac address: b0:6e:bf:4e:c3:77

Partner's mac address: 88:d7:f6:1e:12:2b

The address resolution answer contains:

▼ Address Resolution Protocol (request)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

Sender MAC address: ASUSTekC_1e:12:2b (88:d7:f6:1e:12:2b)

Sender IP address: 156.17.237.52

Target MAC address: ASUSTekC_4e:c3:77 (b0:6e:bf:4e:c3:77)

Target IP address: 156.17.237.32

B)

We can ping google.pl or some other site on external network, but when sending the packets to the website, the mac (physical address) is only used to connect to the local router, and the router de-encapsulates the part with physical address and changes it. Now the destination address becomes the next router and the source is our router. And the same thing is done throughout the next hops. So even when the remote website sends us the packet the source ip is the ip of the website, but the source physical address is the address of the router we are connected to.

1466	10.018786	156.17.237.32	216.58.215.78	ICMP	74 Echo (ping) request	id=0x0001, seq=437/46337, ttl=128 (reply in 1469)
1469	10.027638	216.58.215.78	156.17.237.32	ICMP	74 Echo (ping) reply	id=0x0001, seq=437/46337, ttl=117 (request in 1466)
1590	11.024045	156.17.237.32	216.58.215.78	ICMP	74 Echo (ping) request	id=0x0001, seq=438/46593, ttl=128 (reply in 1592)
1592	11.035513	216.58.215.78	156.17.237.32	ICMP	74 Echo (ping) reply	id=0x0001, seq=438/46593, ttl=117 (request in 1590)
1713	12.029124	156.17.237.32	216.58.215.78	ICMP	74 Echo (ping) request	id=0x0001, seq=439/46849, ttl=128 (reply in 1716)
1716	12.038360	216.58.215.78	156.17.237.32	ICMP	74 Echo (ping) reply	id=0x0001, seq=439/46849, ttl=117 (request in 1713)
1825	13.033520	156.17.237.32	216.58.215.78	ICMP	74 Echo (ping) request	id=0x0001, seq=440/47105, ttl=128 (reply in 1828)
1828	13.042963	216.58.215.78	156.17.237.32	ICMP	74 Echo (ping) reply	id=0x0001, seq=440/47105, ttl=117 (request in 1825)

>	Frame 1469: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{C2BC553A-428C-4D0B-9C70-B838291D2C69}, id 0
▼	Ethernet II, Src: Cisco_7b:2f:c7 (00:15:f9:7b:2f:c7), Dst: ASUSTekC_4e:c3:77 (b0:6e:bf:4e:c3:77)
	> Destination: ASUSTekC_4e:c3:77 (b0:6e:bf:4e:c3:77)
	> Source: Cisco_7b:2f:c7 (00:15:f9:7b:2f:c7)
	Type: IPv4 (0x0800)
>	Internet Protocol Version 4, Src: 216.58.215.78, Dst: 156.17.237.32
>	Internet Control Message Protocol

The Physical address in the ethernet header is still out routers address.

For that reason arp table doesn't contain google.pl address.

Task 4:

We have debian linux installed on virtual machine, so we ping the real computer from the virtual machine:

```
stud@deb10:~$ ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data:
64 bytes from 192.168.56.1: icmp_seq=1 ttl=128 time=0.558 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=128 time=0.555 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=128 time=0.340 ms
64 bytes from 192.168.56.1: icmp_seq=4 ttl=128 time=0.285 ms
64 bytes from 192.168.56.1: icmp_seq=5 ttl=128 time=0.325 ms
64 bytes from 192.168.56.1: icmp_seq=6 ttl=128 time=0.448 ms
64 bytes from 192.168.56.1: icmp_seq=7 ttl=128 time=0.368 ms
^C
--- 192.168.56.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 169ms
rtt min/avg/max/mdev = 0.285/0.411/0.558/0.103 ms
stud@deb10:~$ _
```

We captured the traffic sent from linux in wireshark installed on windows machine (the one which we pinged from linux):

4	5.621285	192.168.56.101	192.168.56.1	ICMP	98 Echo (ping) request	id=0x01b4, seq=1/256, ttl=64 (reply in 5)
5	5.621361	192.168.56.1	192.168.56.101	ICMP	98 Echo (ping) reply	id=0x01b4, seq=1/256, ttl=128 (request in 4)
6	6.632195	192.168.56.101	192.168.56.1	ICMP	98 Echo (ping) request	id=0x01b4, seq=2/512, ttl=64 (reply in 7)
7	6.632340	192.168.56.1	192.168.56.101	ICMP	98 Echo (ping) reply	id=0x01b4, seq=2/512, ttl=128 (request in 6)
8	7.660364	192.168.56.101	192.168.56.1	ICMP	98 Echo (ping) request	id=0x01b4, seq=3/768, ttl=64 (reply in 9)
9	7.660434	192.168.56.1	192.168.56.101	ICMP	98 Echo (ping) reply	id=0x01b4, seq=3/768, ttl=128 (request in 8)
10	8.679487	192.168.56.101	192.168.56.1	ICMP	98 Echo (ping) request	id=0x01b4, seq=4/1024, ttl=64 (reply in 11)
11	8.679552	192.168.56.1	192.168.56.101	ICMP	98 Echo (ping) reply	id=0x01b4, seq=4/1024, ttl=128 (request in 10)
12	9.726150	192.168.56.101	192.168.56.1	ICMP	98 Echo (ping) request	id=0x01b4, seq=5/1280, ttl=64 (reply in 13)
13	9.726220	192.168.56.1	192.168.56.101	ICMP	98 Echo (ping) reply	id=0x01b4, seq=5/1280, ttl=128 (request in 12)
16	10.781068	192.168.56.101	192.168.56.1	ICMP	98 Echo (ping) request	id=0x01b4, seq=6/1536, ttl=64 (reply in 17)
17	10.781214	192.168.56.1	192.168.56.101	ICMP	98 Echo (ping) reply	id=0x01b4, seq=6/1536, ttl=128 (request in 16)

> Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{AA5496CF-A557-4674-959C-526102899113}, id 0

> Ethernet II, Src: PcsCompu_64:ff:a6 (08:00:27:64:ff:a6), Dst: 0a:00:27:00:00:10 (0a:00:27:00:00:10)

> Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.1

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 84

Identification: 0x9bb0 (39856)

> Flags: 0x4000, Don't fragment

Fragment offset: 0

Time to live: 64

Protocol: ICMP (1)

Header checksum: 0xad41 [validation disabled]

0000	0a 00 27 00 00 10 08 00	27 64 ff a6 08 00 45 00	..'. 'd....E-
0010	00 54 9b b0 40 00 00 01	ad 41 c0 a8 38 65 c0 a8	-T-@-@-A-8e..
0020	38 01 08 00 d7 a3 01 b4	00 01 3c 3c 99 5f 00 00	8-.....<<_..
0030	00 00 84 38 06 00 00 00	00 00 10 11 12 13 14 15	...8-.....
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25!""#\$\$%
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	&'()*+,-./012345
0060	36 37	67	

- These are the ethernet header fields:

```
> Ethernet II, Src: PcsCompu_64:ff:a6 (08:00:27:64:ff:a6), Dst: 0a:00:27:00:00:10 (0a:00:27:00:00:10)
> Destination: 0a:00:27:00:00:10 (0a:00:27:00:00:10)
> Source: PcsCompu_64:ff:a6 (08:00:27:64:ff:a6)
Type: IPv4 (0x0800)
```

Their size In bytes are:

Destination: 6 bytes,

Source: 6 bytes,

Type: 2 bytes

All together : 14 bytes

- Because the total size of header fields is 14 bytes and the whole frame is 98 bytes, data is 98 – 14 = 84 bytes:

```
> Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{AA5496CF-A557-4674-959C-526102899113}, id 0
> Ethernet II, Src: PcsCompu_64:ff:a6 (08:00:27:64:ff:a6), Dst: 0a:00:27:00:00:10 (0a:00:27:00:00:10)
> Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.1
> Internet Control Message Protocol
```

- Contents of data field:

00	00	84	38	06	00	00	00	00	00	10	11	12	13	14	15	..	.8....
16	17	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23	24	25	!"#\$%
26	27	28	29	2a	2b	2c	2d	2e	2f	30	31	32	33	34	35	&'()	*+, -	./012345
36	37															67		

Looks alike the contents of asccii table.

- Source and destination mac addresses:

```
> Destination: 0a:00:27:00:00:10 (0a:00:27:00:00:10)
> Source: PcsCompu_64:ff:a6 (08:00:27:64:ff:a6)
Type: IPv4 (0x0800)
```