Komunikacijske mreže

Kristo Palić

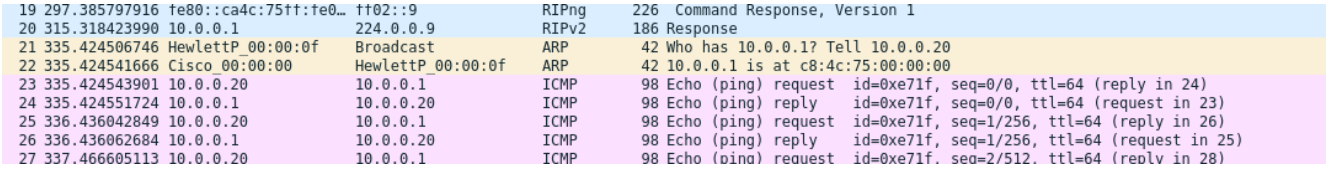
0246074767

1. Laboratorijska vježba

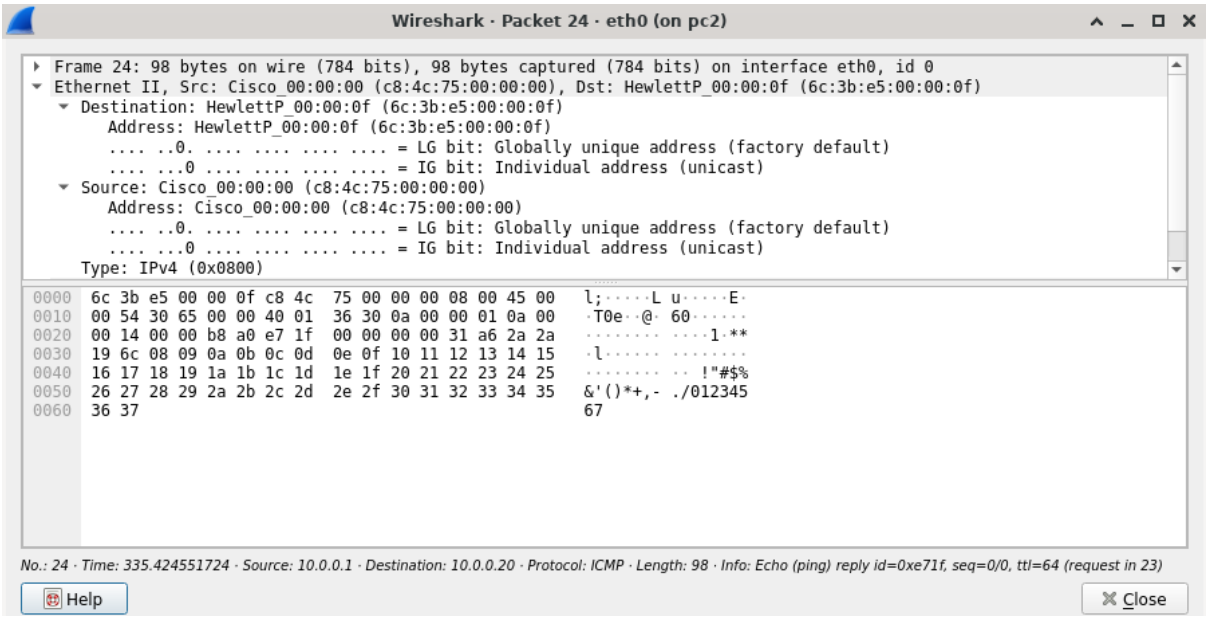
1. zadatak

Učitamo Ping/ping.imn u IMUNES i stisnemo execute. Pokrenemo snimanje u alatu Wireshark. Otvorimo terminal za pc2 i naredbom ping 10.0.0.1 pošaljemo ICPM request prema pc1

```
root@pc2:/ # ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1): 56 data bytes
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=0.074 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.084 ms
```



```
19 297.385797916 fe80::ca4c:75ff:fe0… ff02::9             RIPng     226  Command Response, Version 1
20 315.318423990 10.0.0.1             224.0.0.9           RIPv2     186 Response
21 335.424506746 HewlettP_00:00:0f    Broadcast           ARP        42 Who has 10.0.0.1? Tell 10.0.0.20
22 335.424541666 Cisco_00:00:00       HewlettP_00:00:0f   ARP        42 10.0.0.1 is at c8:4c:75:00:00:00
23 335.424543901 10.0.0.20            10.0.0.1            ICMP       98 Echo (ping) request  id=0xe71f, seq=0/0, ttl=64 (reply in 24)
24 335.424551724 10.0.0.1             10.0.0.20           ICMP       98 Echo (ping) reply    id=0xe71f, seq=0/0, ttl=64 (request in 23)
25 336.436042849 10.0.0.20            10.0.0.1            ICMP       98 Echo (ping) request  id=0xe71f, seq=1/256, ttl=64 (reply in 26)
26 336.436062684 10.0.0.1             10.0.0.20           ICMP       98 Echo (ping) reply    id=0xe71f, seq=1/256, ttl=64 (request in 25)
27 337.466605113 10.0.0.20            10.0.0.1            ICMP       98 Echo (ping) request  id=0xe71f, seq=2/512, ttl=64 (reply in 28)
```



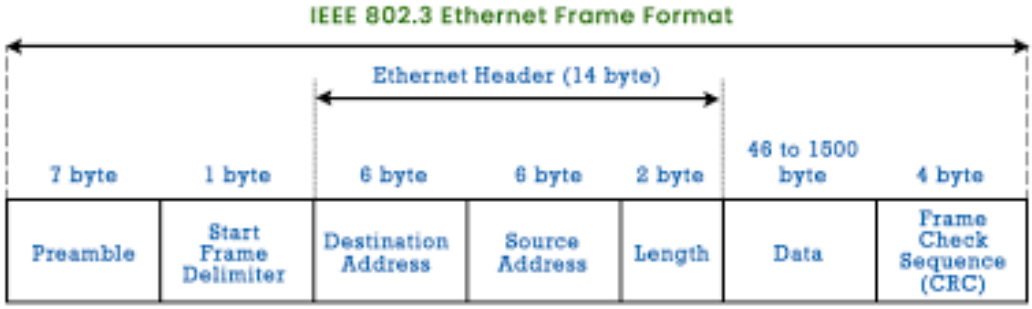
```
Wireshark · Packet 24 · eth0 (on pc2)

Frame 24: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
Ethernet II, Src: Cisco_00:00:00 (c8:4c:75:00:00:00), Dst: HewlettP_00:00:0f (6c:3b:e5:00:00:0f)
  Destination: HewlettP_00:00:0f (6c:3b:e5:00:00:0f)
      Address: HewlettP_00:00:0f (6c:3b:e5:00:00:0f)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: Cisco_00:00:00 (c8:4c:75:00:00:00)
      Address: Cisco_00:00:00 (c8:4c:75:00:00:00)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)

0000  6c 3b e5 00 00 0f c8 4c  75 00 00 00 08 00 45 00   l;·····L u·····E·
0010  00 54 30 65 00 00 40 01  36 30 0a 00 00 01 0a 00   ·T0e··@· 60······
0020  00 14 00 00 b8 a0 e7 1f  00 00 00 00 31 a6 2a 2a   ········ ····1·**
0030  19 6c 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   ·l······ ········
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

No.: 24 · Time: 335.424551724 · Source: 10.0.0.1 · Destination: 10.0.0.20 · Protocol: ICMP · Length: 98 · Info: Echo (ping) reply id=0xe71f, seq=0/0, ttl=64 (request in 23)
Help                                                                                                  Close
```

OUI – prva tri para heksadecimalnih brojeva : C8:4C:75 (Cisco systems inc.)

NIC – zadnja tri para heksadecimalnih brojeva : 00:00:00

2. zadatak

```
6c 3b e5 00 00 0f c8 4c  75 00 00 00 08 00 45 00
00 54 30 65 00 00 40 01  36 30 0a 00 00 01 0a 00
00 14 00 00 b8 a0 e7 1f  00 00 00 00 31 a6 2a 2a
19 6c 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15
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
36 37
```



Ukupna veličina zaglavlja Ethernet okvira u Wiresharku je 14 bajtova (6 bajtova odredišna adresa + 6 bajtova izvorišna adresa + 2 bajta tip/dužina). Prepoznajemo zaglavlje koje sam već opisao i 46 bajtova namjenjenih za podatke.

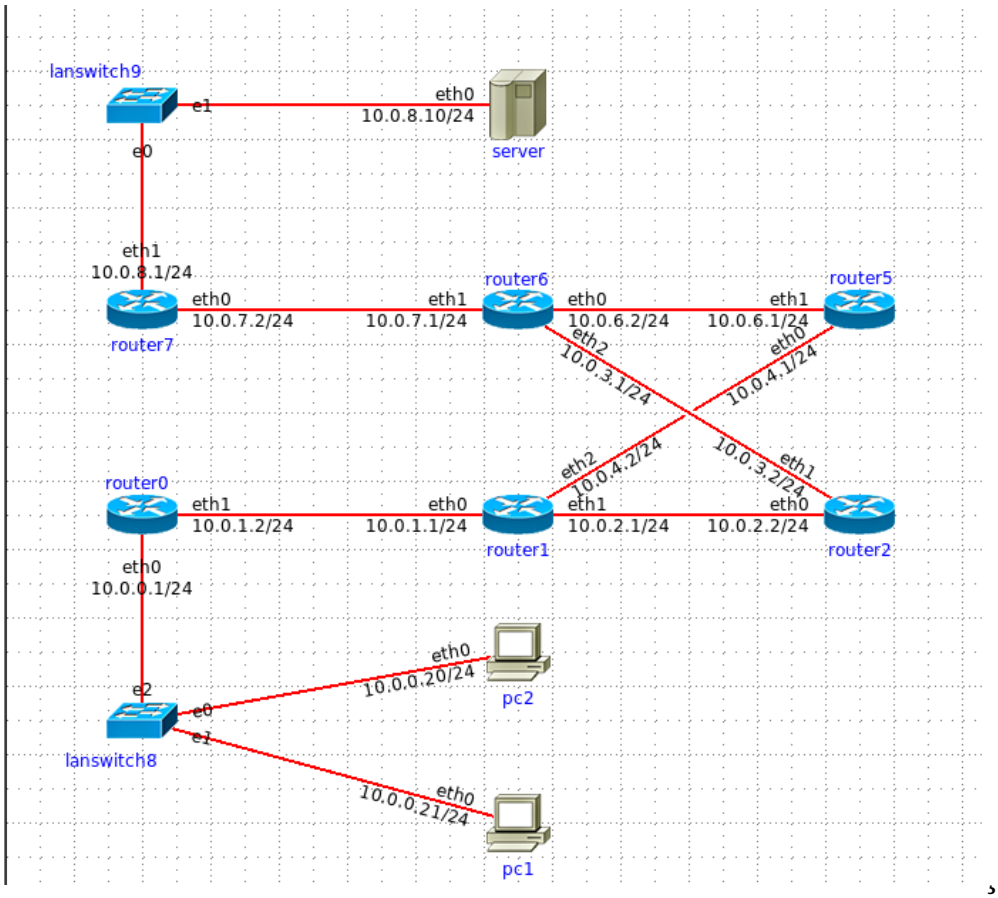
Preambula, SFD i CRC nisu prikazani u Wiresharku jer se oni obrađuju na razini fizičkog sloja (Layer 1).

3. zadatak

Pomoću polja Type u Ethernet zaglavlju. Dugo je dva bajta i nalazi se između izvorišne MAC adrese te podatkovnog polja okvira.

Polje Type sadrži Ethernet kod koji identificira protokol višeg sloja. (IP, ARP, VLAN itd.)

4. zadatak




```
root@pc1:/ # traceroute 10.0.8.10
traceroute to 10.0.8.10 (10.0.8.10), 64 hops max, 40 byte
 1  10.0.0.1 (10.0.0.1)  0.088 ms  0.047 ms  0.036 ms
 2  10.0.1.1 (10.0.1.1)  0.034 ms  0.032 ms  0.017 ms
 3  10.0.4.1 (10.0.4.1)  0.023 ms  0.060 ms  0.036 ms
 4  10.0.6.2 (10.0.6.2)  0.043 ms  0.090 ms  0.069 ms
 5  10.0.7.2 (10.0.7.2)  0.047 ms  0.099 ms  0.041 ms
 6  10.0.8.10 (10.0.8.10)  0.129 ms  0.060 ms  0.027 ms
```

```
root@server:/ # traceroute 10.0.0.21
traceroute to 10.0.0.21 (10.0.0.21), 64 hops max, 40 byte
 1  10.0.8.1 (10.0.8.1)  0.169 ms  0.059 ms  0.027 ms
 2  10.0.7.1 (10.0.7.1)  0.037 ms  0.060 ms  0.101 ms
 3  10.0.6.1 (10.0.6.1)  0.073 ms  0.061 ms  0.091 ms
 4  10.0.4.2 (10.0.4.2)  0.103 ms  0.101 ms  0.075 ms
 5  10.0.1.2 (10.0.1.2)  0.098 ms  0.063 ms  0.036 ms
 6  10.0.0.21 (10.0.0.21)  0.058 ms  0.082 ms  0.055 ms
```

Putevi su simetrični.

5. zadatak

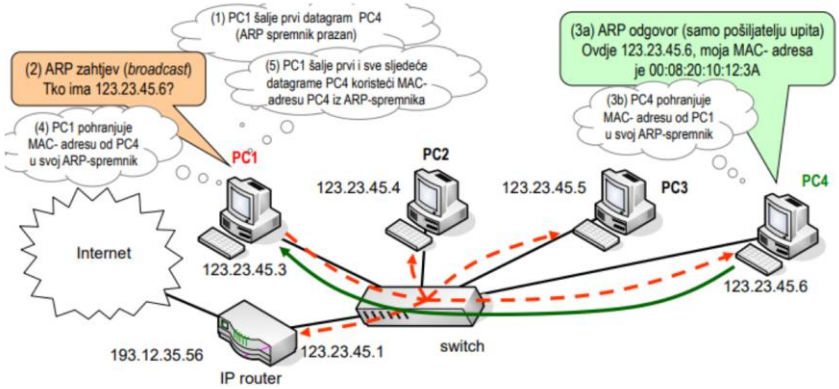
IP paketi se prenose Ethernet protokolom unutar lokalne mreže na podatkovnom sloju OSI modela i koristi MAC adrese kako bi pravilno usmjerio pakete unutar lokalne mreže.

Protokol ARP (Address Resolution Protocol) igra ključnu ulogu u tom procesu. Koristi se za mapiranje IP adresa (sloj 3) na odgovarajuće MAC adrese (sloj 2) uređaja unutar lokalne mreže. ARP šalje upit za IP-adresu mreži, tj. svim sučeljima. Svi primaju upit, no odgovor šalje samo uređaj s traženom IP adresom, no ne svima već samo prozivatelju. ARP spremnik pohranjuje uparene IP-MAC adrese na temelju navedenih upita i odgovora te ih u daljnoj komunikaciji koristi kako se ne bi ponavljao upit cijeloj mreži.

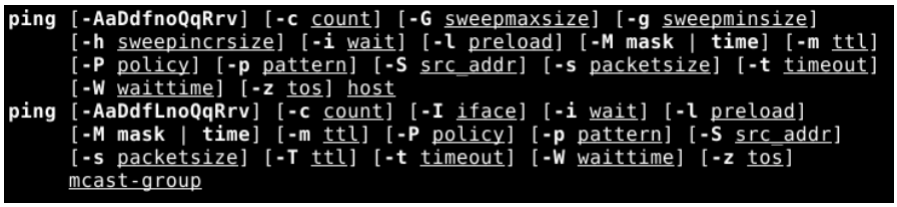
6. zadatak

```
2 3.429221032   RealtekA_00:00:10   Broadcast           ARP        42 Who has 10.0.0.1? Tell 10.0.0.21
3 3.429258467   Cisco 00:00:00       RealtekA 00:00:10   ARP        42 10.0.0.1 is at c8:4c:75:00:00:00
```

- Kada uređaj želi poslati podatke drugom uređaju unutar iste lokalne mreže, ali zna samo njegovu IP adresu, šalje ARP zahtjev.
- ARP zahtjev sadrži IP adresu traženog uređaja i šalje se kao broadcast, što znači da ga svi uređaji u lokalnoj mreži primaju.
- Uređaj s traženom IP adresom prepoznaje zahtjev i šalje ARP odgovor, koji sadrži njegovu MAC adresu.
- Početni uređaj prima ARP odgovor i ažurira svoju ARP tablicu s MAC adresom traženog uređaja. Sada može poslati podatke na odgovarajuću MAC adresu.



7. zadatak



-c count – Određuje broj paketa prilikom slanja pinga

-G sweepmaxsize – Odrežuje maksimalnu veličinu ICMP datagrama prilikom pročešljavanja

-g sweepminsize – Određuje minimalnu veličinu ICMP datagrama prilikom pročešljavanja

-h sweepincrsize – Određuje koliko se veličina ICMP datagrama povećava nakon svake iteracije

-i wait – Određuje koliko će vremena proći između dva pinga

-l preload – Ako je specifičan argument, ping šalje taj broj paketa dok se ne vrati u normalno ponašanje

-M mask | time – Postavlja ICMP oznaku na echo ili repl, time je vrijeme slanja, primanja i transmisije

-m ttl – Postavlja Time To Live paketa

-P policy – Postavlja odredbe policy za odgovarajuću ping sjednicu

-p pattern – Puni pakete sa do 16 bajtova sadržanih u pattern

-S src_addr – u odlaznim paketima koristi srd_addr kao adresu pošiljatelja zahtjeva

-s packetsize – Određuje veličinu paketa

-t timeout – Određuje vrijeme tijekom kojeg će se ping izvršavati

-W waittime – vrijeme čekanja odgovora u milisekundama. Zakašnjeli odgovori se ne ispisuju

-z tos – određuje koji će se tip usluge koristiti

8. zadatak

```
root@pc1:/ # ping -m 3 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 56 data bytes
92 bytes from 10.0.2.2: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 27ce   0 0000  01  01 75bd 10.0.0.21  10.0.8.10

92 bytes from 10.0.2.2: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 16be   0 0000  01  01 86cd 10.0.0.21  10.0.8.10

92 bytes from 10.0.2.2: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 27cf   0 0000  01  01 75bc 10.0.0.21  10.0.8.10

92 bytes from 10.0.2.2: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 27d0   0 0000  01  01 75bb 10.0.0.21  10.0.8.10
```

TTL je dopušten broj skokova. S obzirom da je traženi server udaljen za više od 3 skoka, vraća nam se error poruka ttl exceeded

9. zadatak

```
242 1023.4196896… 10.0.0.21   10.0.8.10   ICMP   1162 Echo (ping) request  id=0xa40c, seq=10/2560, ttl=64 (reply in 249)
243 1023.4199156… 10.0.8.10   10.0.0.21   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8cd3) [Reassembled in #249]
244 1023.4199226… 10.0.8.10   10.0.0.21   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8cd3) [Reassembled in #249]
245 1023.4199248… 10.0.8.10   10.0.0.21   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=8cd3) [Reassembled in #249]
246 1023.4199265… 10.0.8.10   10.0.0.21   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=4440, ID=8cd3) [Reassembled in #249]
247 1023.4199285… 10.0.8.10   10.0.0.21   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=5920, ID=8cd3) [Reassembled in #249]
248 1023.4199301… 10.0.8.10   10.0.0.21   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=7400, ID=8cd3) [Reassembled in #249]
249 1023.4199318… 10.0.8.10   10.0.0.21   ICMP   1162 Echo (ping) reply    id=0xa40c, seq=10/2560, ttl=59 (request in 242)
```

Maksimalna transmisijska jedinica MTU iznosi 1500 okteta i to je najveća veličina koja se može prenijeti u jednom komadu. Tada je maksimalni IP datagram koji možemo poslati 1472 okteta + 8 okteta ICMP + 20 okteta zaglavlja = 1500.

Kada je ping velik 10000 okteta prvo mu se doda 8 za ICMP te iznosi 10008. Zatim se fragmentira na :

1. 1480 + 20
2. ...

7. 80 + 20 + 8

```
root@pc1:/ # ping -s 70000 10.0.8.10
ping: packet size too large: 70000 > 65507
root@pc1:/ #
```

Najveći broj paketa je 65507 + 8 = 65515

10. zadatak

- veliki paket

```
root@pc1:/ # ping -s 10000 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 10000 data bytes
10008 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=0.349 ms
10008 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=0.826 ms
10008 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=0.472 ms
10008 bytes from 10.0.8.10: icmp_seq=5 ttl=59 time=0.333 ms
10008 bytes from 10.0.8.10: icmp_seq=6 ttl=59 time=0.995 ms
```

- neizravno spojeni

```
root@pc1:/ # ping  10.0.8.10
PING 10.0.8.10 (10.0.8.10): 56 data bytes
64 bytes from 10.0.8.10: icmp_seq=0 ttl=59 time=0.128 ms
64 bytes from 10.0.8.10: icmp_seq=1 ttl=59 time=0.246 ms
64 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=0.205 ms
64 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=0.244 ms
64 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=0.117 ms
64 bytes from 10.0.8.10: icmp_seq=5 ttl=59 time=0.092 ms
64 bytes from 10.0.8.10: icmp_seq=6 ttl=59 time=0.601 ms
^C
--- 10.0.8.10 ping statistics ---
7 packets transmitted, 7 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.092/0.233/0.601/0.161 ms
```

- izravno spojeni

```
root@pc1:/ # ping 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 56 data bytes
64 bytes from 10.0.8.10: icmp_seq=0 ttl=62 time=0.113 ms
64 bytes from 10.0.8.10: icmp_seq=1 ttl=62 time=0.518 ms
64 bytes from 10.0.8.10: icmp_seq=2 ttl=62 time=0.309 ms
64 bytes from 10.0.8.10: icmp_seq=3 ttl=62 time=0.454 ms
64 bytes from 10.0.8.10: icmp_seq=4 ttl=62 time=0.219 ms
64 bytes from 10.0.8.10: icmp_seq=5 ttl=62 time=0.306 ms
64 bytes from 10.0.8.10: icmp_seq=6 ttl=62 time=0.556 ms
64 bytes from 10.0.8.10: icmp_seq=7 ttl=62 time=2.211 ms
64 bytes from 10.0.8.10: icmp_seq=8 ttl=62 time=0.608 ms
^C
--- 10.0.8.10 ping statistics ---
9 packets transmitted, 9 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.113/0.588/2.211/0.594 ms
```

U nekom normalnom slučaju bi vrijeme izravno spojenih trebalo bit manje nego kod neizravno spojenih, ali je moje računalo odlučilo protestirat. TTL se promijeni jer postoji izravni skok sa r0 na r7

11. zadatak

Uz kašnjenje

```
root@pc1:/ # ping 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 56 data bytes
64 bytes from 10.0.8.10: icmp_seq=0 ttl=59 time=32.454 ms
64 bytes from 10.0.8.10: icmp_seq=1 ttl=59 time=13.433 ms
64 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=4.227 ms
64 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=14.296 ms
64 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=13.882 ms
64 bytes from 10.0.8.10: icmp_seq=5 ttl=59 time=12.809 ms
^C
--- 10.0.8.10 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 4.227/15.183/32.454/8.461 ms
root@pc1:/ #
```

Bez kašnjenja

```
root@pc1:/ # ping 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 56 data bytes
64 bytes from 10.0.8.10: icmp_seq=0 ttl=59 time=0.094 ms
64 bytes from 10.0.8.10: icmp_seq=1 ttl=59 time=0.499 ms
64 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=0.565 ms
64 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=0.278 ms
64 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=0.451 ms
^C
--- 10.0.8.10 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.094/0.377/0.565/0.171 ms
root@pc1:/ #
```

Propagacijsko kašnjenje ima veliki utjecaj na vrijeme prijenosa. U ovim primjerima smo dodali delay između pc1 i lanswitch8

12. zadatak

2	6.134062011	RealtekA_00:00:10	Broadcast	ARP	42 Who has 10.0.0.1? Tell
3	6.134103357	Cisco_00:00:00	RealtekA_00:00:10	ARP	42 10.0.0.1 is at c8:4c:7
4	6.134107268	10.0.0.21	10.0.8.10	ICMP	98 Echo (ping) request i
5	6.134224043	10.0.8.10	10.0.0.21	ICMP	98 Echo (ping) reply i
6	7.143816323	10.0.0.21	10.0.8.10	ICMP	98 Echo (ping) request i
7	7.143977237	10.0.8.10	10.0.0.21	ICMP	98 Echo (ping) reply i
8	8.155641011	10.0.0.21	10.0.8.10	ICMP	98 Echo (ping) request i
9	8.155718953	10.0.8.10	10.0.0.21	ICMP	98 Echo (ping) reply i
10	9.180199160	10.0.0.21	10.0.8.10	ICMP	98 Echo (ping) request i
11	9.180525737	10.0.8.10	10.0.0.21	ICMP	98 Echo (ping) reply i
12	10.200660699	10.0.0.21	10.0.8.10	ICMP	98 Echo (ping) request i
13	10.201030578	10.0.8.10	10.0.0.21	ICMP	98 Echo (ping) reply i

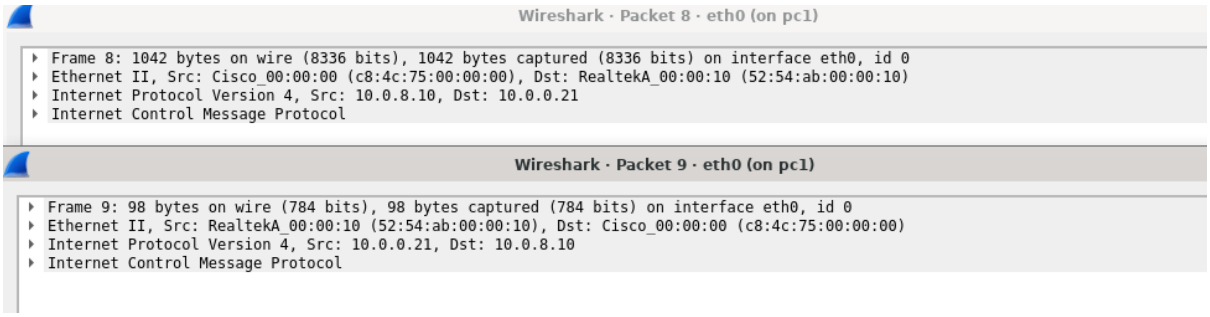
ICMP (Internet Control Message Protocol) – alat ping koristi ovaj protokol za slanje Echo Request poruka i primanje Echo Reply poruka. Pripada mrežnom sloju (sloj 3) u TCP/IP modelu.

IP – ICMP koristi IP za usmjeravanje između izvorišnog i odredišnog računala. Također pripada mrežnom sloju (sloj 3) u TCP/IP modelu

Ethernet – služi za prijenos IP paketa unutar LAN mreže između računala i mrežnih uređaja. Pripada sloju pristupa mreži (Data Access Layer) sloj 2

ARP – (Address Resolutin Protocol) – kada računalo treba poslati paket unutar LAN koristi protokol ARP kako bi pronašlo odgovarajuću MAC adresu za odredište. Pripada sloju pristupa mreži (sloj 2)

13. zadatak



Mijenjaju se: polje podataka, duljina, padding ako je polje podataka premalo i FCS

14. zadatak

Kada pingamo loopback adresu (127.0.0.1), promet se ne šalje na ethernetsko sučelje računala. Umjesto toga, promet ostaje unutar lokalnog računala. ICMP Echo Request i ICMP Echo Reply poruke generirane naredbom ping neće napustiti računalo, već će se obraditi unutar IP sloja lokalnog računala. Kao rezultat toga, promet na ethernetskom sučelju računala ostaje nepromijenjen i neće biti vidljiv u alatu Wireshark ili drugim alatima za praćenje mrežnog prometa.

15. zadatak

```
root@pc1:/ # ifconfig eth0 mtu /0
ifconfig: ioctl SIOCSIFMTU (set mtu): Invalid argument
root@pc1:/ # ifconfig eth0 mtu 72
root@pc1:/ # ifconfig eth0 mtu 9020
ifconfig: ioctl SIOCSIFMTU (set mtu): Invalid argument
root@pc1:/ # ifconfig eth0 mtu 9019
ifconfig: ioctl SIOCSIFMTU (set mtu): Invalid argument
root@pc1:/ # ifconfig eth0 mtu 9018
root@pc1:/ # ifconfig eth0 mtu 1500
root@pc1:/ #
```

Teoretski min i max za MTU su 46 i 1500, ali ako pokušamo podesiti vidimo da su rezultati 72 i 9018.

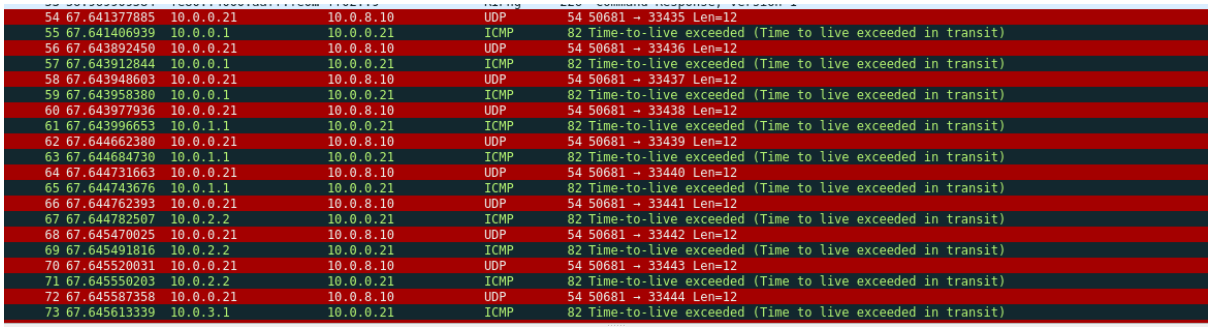
16. zadatak

```
TRACEROUTE(8)          FreeBSD System Manager's Manual          TRACEROUTE(8)

NAME
     traceroute - print the route packets take to network host

SYNOPSIS
     traceroute [-adDeFISnrvx] [-f first_ttl] [-g gateway] [-M first_ttl]
                [-m max_ttl] [-P proto] [-p port] [-q nqueries] [-s src_addr]
                [-t tos] [-w waittime] [-A as_server] [-z pausemsecs] host
                [packetlen]

DESCRIPTION
     The Internet is a large and complex aggregation of network hardware,
     connected together by gateways.  Tracking the route one's packets follow
     (or finding the miscreant gateway that's discarding your packets) can be
     difficult.  traceroute utilizes the IP protocol `time to live' field and
     attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along
     the path to some host.

     The only mandatory parameter is the destination host name or IP number.
     The default probe datagram length is 40 bytes, but this may be increased
     by specifying a packet length (in bytes) after the destination host name.
```

Alat traceroute može proizvesti neispravne rezultate u nekoliko situacija:

1. **Loše rutiranje ili petlje**: TTL exceeded bez dostizanja odredišta
2. **Promjene na putanji usmjeravanja tijekom izvođenja traceroutea**: Ako se putanja paketa između izvora i odredišta promijeni tijekom izvođenja traceroutea (npr. zbog promjena u mrežnoj topologiji ili dinamičkog usmjeravanja), rezultati mogu biti neispravni ili zastarjeli.
3. **Korištenje drugih protokola**: Traceroute obično koristi ICMP poruke za dobivanje informacija o putanji, ali u nekim implementacijama može koristiti i druge protokole kao što su UDP ili TCP. Ovi protokoli mogu biti podložni različitim politikama filtriranja i usmjeravanja u mreži, što može dovesti do neispravnih ili nepotpunih rezultata.

17. zadatak



54	67.641377885	10.0.0.21	10.0.8.10	UDP	54 50681 → 33435 Len=12
55	67.641406939	10.0.0.1	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
56	67.643892450	10.0.0.21	10.0.8.10	UDP	54 50681 → 33436 Len=12
57	67.643912844	10.0.0.1	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
58	67.643948603	10.0.0.21	10.0.8.10	UDP	54 50681 → 33437 Len=12
59	67.643958380	10.0.0.1	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
60	67.643977936	10.0.0.21	10.0.8.10	UDP	54 50681 → 33438 Len=12
61	67.643996653	10.0.1.1	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
62	67.644662380	10.0.0.21	10.0.8.10	UDP	54 50681 → 33439 Len=12
63	67.644684730	10.0.1.1	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
64	67.644731663	10.0.0.21	10.0.8.10	UDP	54 50681 → 33440 Len=12
65	67.644743676	10.0.1.1	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
66	67.644762393	10.0.0.21	10.0.8.10	UDP	54 50681 → 33441 Len=12
67	67.644782507	10.0.2.2	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
68	67.645470025	10.0.0.21	10.0.8.10	UDP	54 50681 → 33442 Len=12
69	67.645491816	10.0.2.2	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
70	67.645520031	10.0.0.21	10.0.8.10	UDP	54 50681 → 33443 Len=12
71	67.645550203	10.0.2.2	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)
72	67.645587358	10.0.0.21	10.0.8.10	UDP	54 50681 → 33444 Len=12
73	67.645613339	10.0.3.1	10.0.0.21	ICMP	82 Time-to-live exceeded (Time to live exceeded in transit)

Mehanizam rada alata traceroute:

1. Traceroute šalje pakete s početnom vrijednošću TTL-a postavljenom na 1.
2. Svaki usmjeritelj koji primi paket smanjuje vrijednost TTL-a za 1. Ako TTL padne na 0, usmjeritelj ne šalje paket dalje i umjesto toga šalje ICMP Time Exceeded poruku natrag na izvorno računalo.
3. Kad izvorno računalo primi ICMP Time Exceeded poruku, može zaključiti da je paket stigao do usmjeritelja s TTL-om jednakim 1. Računalo tada povećava TTL za 1 (na 2) i šalje novi paket.
4. Proces se ponavlja, a svaki puta kad izvorno računalo primi ICMP Time Exceeded poruku, povećava TTL za 1 i šalje novi paket.
5. Konačno, paket stiže do odredišnog računala. Odredišno računalo šalje ICMP Echo Reply poruku natrag na izvorno računalo umjesto ICMP Time Exceeded poruke.
6. Traceroute se zaustavlja kad izvorno računalo primi ICMP Echo Reply poruku ili kad se dostigne maksimalna vrijednost TTL-a.

18. zadatak

Protokol IP "pamti" vrstu paketa koji se prenosi u podatkovnom dijelu IP-datagrama pomoću polja Protocol (ili polja Next Header u slučaju IPv6) unutar IP zaglavlja. Polje Protocol je 8-bitno polje koje sadrži identifikacijski broj koji označava vrstu protokola koji se koristi u podatkovnom dijelu IP-datagrama. Npr. TCP, UDP, ICMP

19. zadatak

Nije moguće. IP protokol je dizajniran da bude neovisan o načinu prolaska kroz mrežu, što znači da ne postoji način da rekonstruiramo put kojim je paket prošao. To možemo jedino pomoću alata traceroute koji nije 100% siguran.

20. zadatak

IP protokol sam po sebi ne pruža mehanizam za potvrdu primitka paketa na odredištu. IP protokol je nesiguran protokol, što znači da ne jamči isporuku paketa, redoslijed paketa ili ispravnost podataka u paketu. Jedino uz pomoć viših protokola kao što su TCP ili ICMP

21. zadatak

Što je veći broj paketa to je veće vrijeme čekanja da se pristigli podatci procesiraju. Veća je mogućnost gubitka paketa te je veći dodatni protokolni overhead. Svaki paket ima svoj IP header što rezultira povećanjem poslanih bitova za istu količinu informacije.