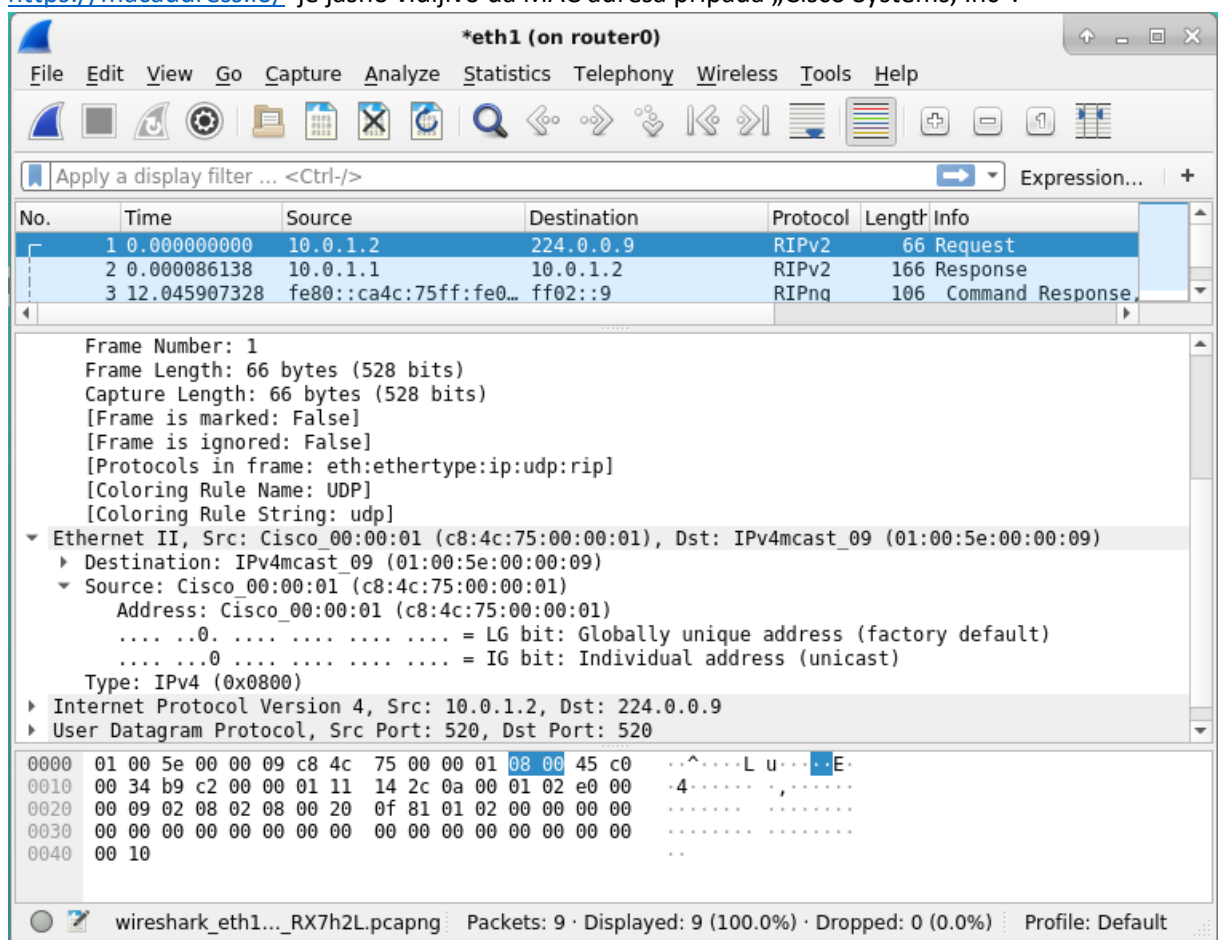


## PRVA LABORATORIJSKA VJEŽBA

**Zadatak 1.** (Topologija Ping/ping.imn) Analizirajte izvorišnu MAC-adresu iz proizvoljno odabranog Ethernet-okvira te odredite dijelove adrese koji se odnose na organizacijski jednoznačni identifikator (OUI) i identifikator mrežnog sučelja (NIC). Za odabranu MAC-adresu pokušajte utvrditi proizvođača pripadajuće mrežne kartice korištenjem web-tražilice.

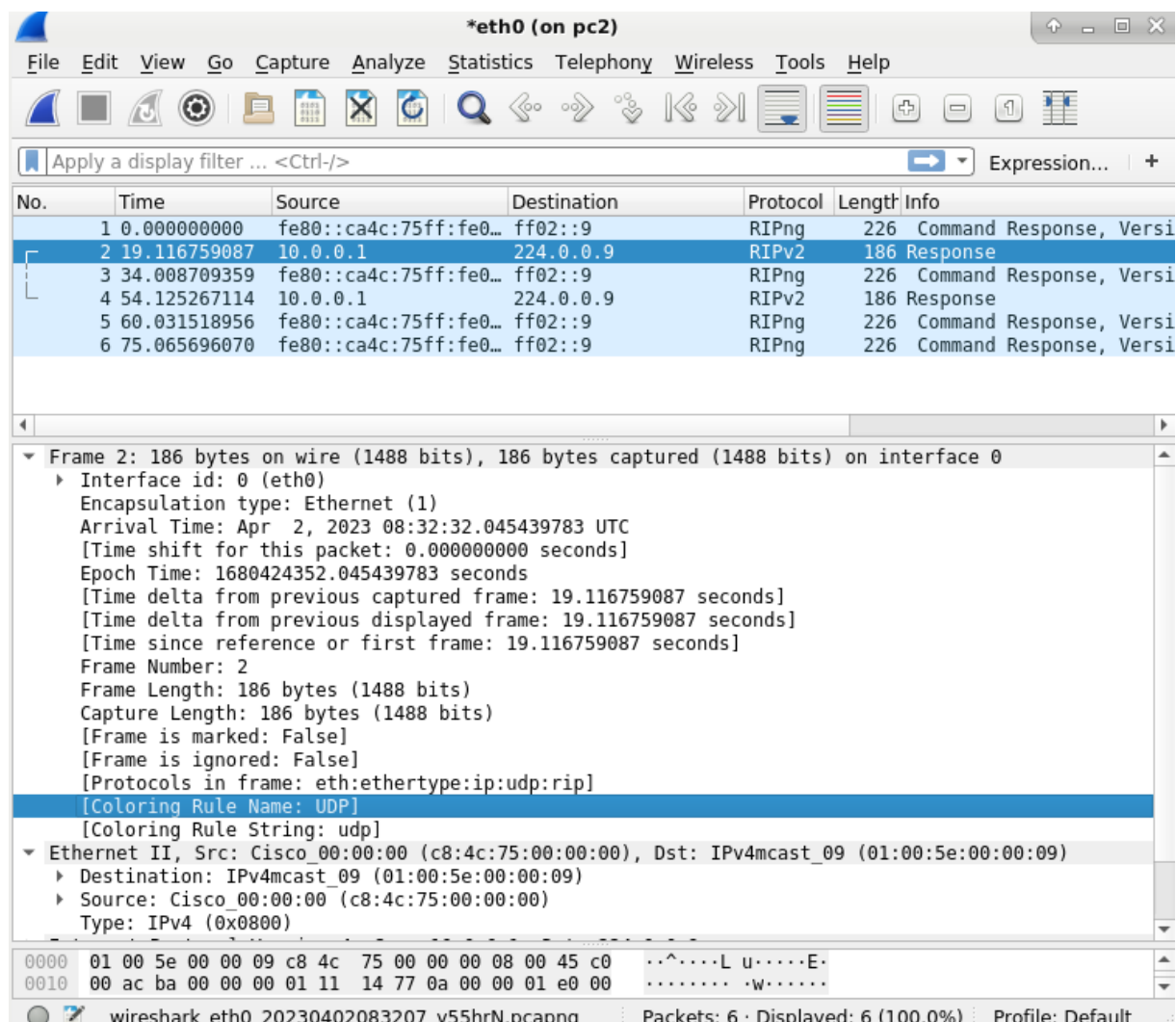
- Eth1 MAC adresa = c8:4c:75:00:00:01

Prva tri okteta(c8:4c:75) se odnose na organizacijski jednoznačni identifikator(OUI) dok se druga tri (00:00:01) odnose na identifikator mrežnog sučelja(NIC). Na stranici <https://macaddress.io/> je jasno vidljivo da MAC adresa pripada „Cisco Systems, Inc“.



**Zadatak 2.** (Topologija Ping/ping.imn) Proizvoljno odaberite jedan Ethernet-okvir i utvrdite veličinu njegovog zaglavlja. Skicirajte strukturu Ethernet-okvira te ju usporedite s prikazom odabranog okvira u alatu Wireshark. Koja polja prikazanog okvira prepoznajete? (Za pojašnjenje prikaza okvira u alatu Wireshark, koristite web-stranicu: <http://wiki.wireshark.org/Ethernet>)

Okvir 01:00:5e:00:00:09:c8:4c:75:00:00:00:08:00 govori u prvih 6 okteta koje je odredište paketa(01:00:5e:00:00:09), u drugih 6 okteta govori koji je izvor paketa(c8:4c:75:00:00:00), te zadnja dva okteta 08:00 govore koji je tip internetskog protokola. Nama govore da je veličine Ethernet zaglavlja 14 okteta.



**\*eth0 (on pc2)**

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::ca4c:75ff:fe0...	ff02::9	RIPng	226	Command Response, Versi
2	19.116759087	10.0.0.1	224.0.0.9	RIPv2	186	Response
3	34.008709359	fe80::ca4c:75ff:fe0...	ff02::9	RIPng	226	Command Response, Versi
4	54.125267114	10.0.0.1	224.0.0.9	RIPv2	186	Response
5	60.031518956	fe80::ca4c:75ff:fe0...	ff02::9	RIPng	226	Command Response, Versi
6	75.065696070	fe80::ca4c:75ff:fe0...	ff02::9	RIPng	226	Command Response, Versi

▼ Frame 2: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface 0

- Interface id: 0 (eth0)
- Encapsulation type: Ethernet (1)
- Arrival Time: Apr 2, 2023 08:32:32.045439783 UTC
- [Time shift for this packet: 0.000000000 seconds]
- Epoch Time: 1680424352.045439783 seconds
- [Time delta from previous captured frame: 19.116759087 seconds]
- [Time delta from previous displayed frame: 19.116759087 seconds]
- [Time since reference or first frame: 19.116759087 seconds]
- Frame Number: 2
- Frame Length: 186 bytes (1488 bits)
- Capture Length: 186 bytes (1488 bits)
- [Frame is marked: False]
- [Frame is ignored: False]
- [Protocols in frame: eth:ethertype:ip:udp:rip]
- [Coloring Rule Name: UDP]
- [Coloring Rule String: udp]
- ▼ Ethernet II, Src: Cisco\_00:00:00 (c8:4c:75:00:00:00), Dst: IPv4mcast\_09 (01:00:5e:00:00:09)
  - Destination: IPv4mcast\_09 (01:00:5e:00:00:09)
  - Source: Cisco\_00:00:00 (c8:4c:75:00:00:00)
  - Type: IPv4 (0x0800)

0000 01 00 5e 00 00 09 c8 4c 75 00 00 00 08 00 45 c0 ..^....L u.....E.  
0010 00 ac ba 00 00 00 01 11 14 77 0a 00 00 01 e0 00 .....w.....

wireshark eth0 20230402083207 v55hrN.pcapng Packets: 6 · Displayed: 6 (100.0%) Profile: Default

**Zadatak 3.** (Topologija Ping/ping.imn) Na koji način protokol Ethernet „pamti“ vrstu paketa koji se prenosi u podatkovnom dijelu Ethernet-okvira?

- Ethernet protokol pamti vrstu paketa kojeg prenosi u zadnja dva okteta zaglavlja

**Zadatak 4.** U emulatoru/simulatoru IMUNES, ispitajte način rada alata traceroute na mreži iz primjera Traceroute/traceroute.imn (Slika 3.2).

```
IMUNES: pc2 (console) bash
[root@pc2 ~]# traceroute 10.0.8.10
traceroute to 10.0.8.10 (10.0.8.10), 64 hops max, 40 byte packets
 1  10.0.0.1 (10.0.0.1)  0.395 ms  0.244 ms  0.161 ms
 2  10.0.1.1 (10.0.1.1)  36.990 ms  0.464 ms  0.232 ms
 3  10.0.4.1 (10.0.4.1)  0.167 ms  0.146 ms  0.136 ms
 4  10.0.3.1 (10.0.3.1)  0.152 ms  0.167 ms  0.139 ms
 5  10.0.7.2 (10.0.7.2)  0.144 ms  0.152 ms  0.146 ms
 6  10.0.8.10 (10.0.8.10)  0.207 ms  0.161 ms  0.150 ms
[root@pc2 ~]#
```

Put od PC2 do servera je : PC2 – router0 – router1 – router5 – router6 – router7 – server

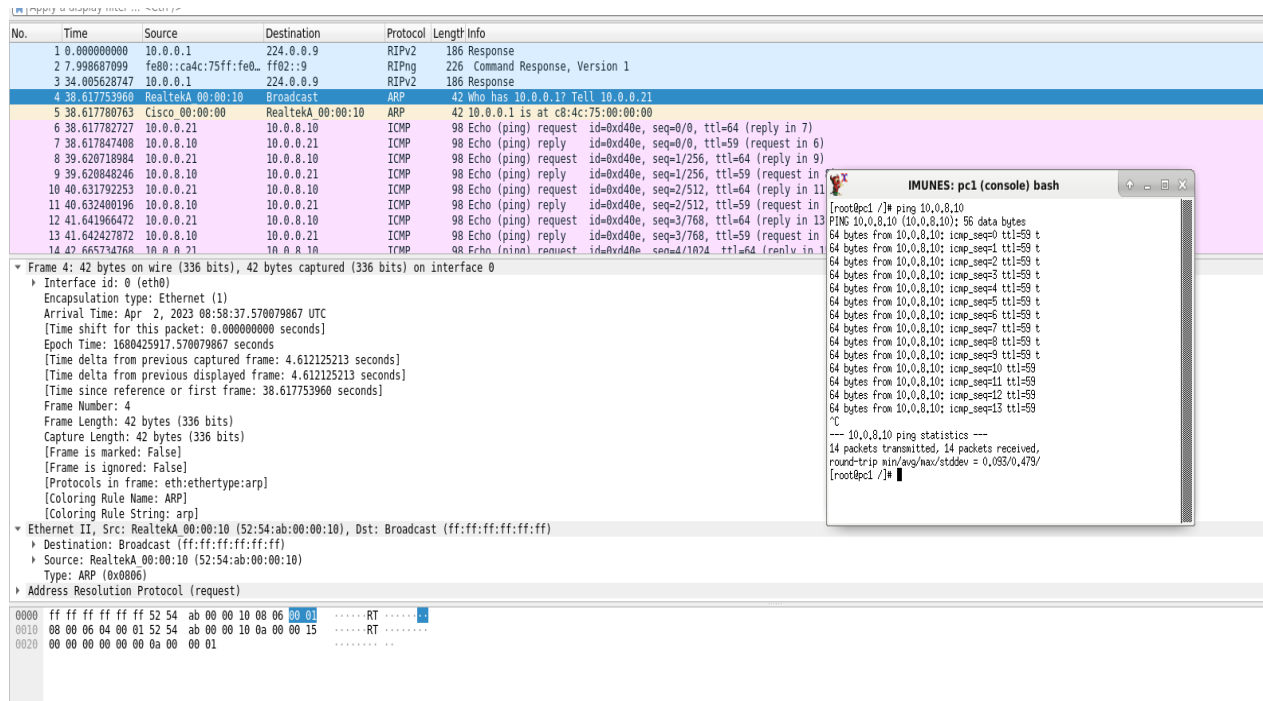
```
IMUNES: server (console) bash
[root@server ~]# traceroute 10.0.0.20
traceroute to 10.0.0.20 (10.0.0.20), 64 hops max, 40 byte packets
 1  10.0.8.1 (10.0.8.1)  0.315 ms  0.094 ms  0.132 ms
 2  10.0.7.1 (10.0.7.1)  0.083 ms  0.088 ms  0.132 ms
 3  10.0.3.2 (10.0.3.2)  0.086 ms  0.050 ms  0.153 ms
 4  10.0.4.2 (10.0.4.2)  0.095 ms  0.097 ms  0.138 ms
 5  10.0.1.2 (10.0.1.2)  0.094 ms  0.100 ms  0.142 ms
 6  10.0.0.20 (10.0.0.20)  0.106 ms  0.122 ms  0.183 ms
[root@server ~]#
```

Put od servera do PC2 daje drugačiji put : server – router7 – router6 – router2 – router1 – router0 – PC2

**Zadatak 5.** (Topologija Traceroute/traceroute.imn) Kojim protokolom se IP-paketi prenose između računala smještenih unutar jedne lokalne mreže? Čemu, pri tome, služi protokol ARP?

- IP paketi se prenose TCP protokolom, a ARP služi za pronalazak MAC adrese za određenu IP adresu.

**Zadatak 6.** (Topologija Ping/ping.imn) Ponovite pokazni eksperiment s početka poglavlja (Ping/ping.imn) te snimite promet koji pripada protokolu ARP. Skicirajte i objasnite način rada protokola ARP, a posebnu pozornost obratite na IP-adresu koja se navodi u ARP-zahtjevu. Kojem čvoru odgovara ta IP-adresa? Objasnite. Čemu služe višedredišne adrese u protokolu Ethernet? Koristi li ih protokol ARP?

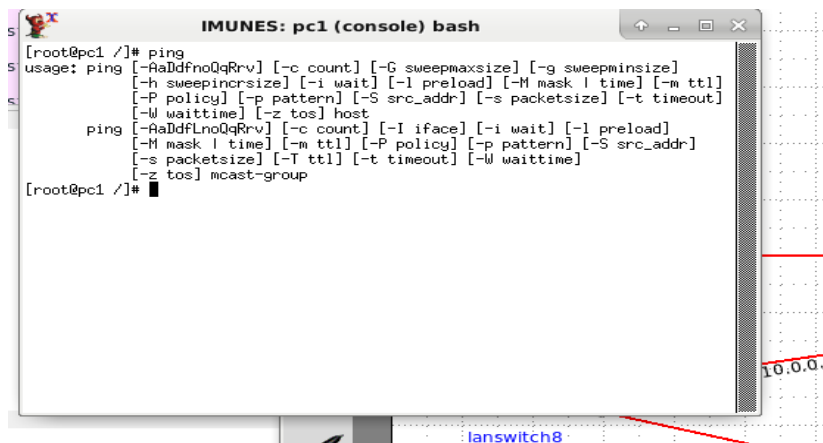


The image shows a Wireshark packet capture of network traffic. The main pane displays a list of packets, with packet 4 (ARP) highlighted. The packet details pane shows the ARP request from 10.0.0.21 to 10.0.0.1. The packet bytes pane shows the raw data. On the right, a terminal window titled 'IMUNES: pc1 (console) bash' shows the output of the ping command, indicating that the ping is successful.

Iz simulacije možemo vidjeti kako se šalje broadcast upit gdje izvor (10.0.21) pita(Who has 10.0.0.1) tko ima adresu 10.0.0.1 te onaj čija adresa odgovara 10.0.0.1 šalje svoju MAC adresu te ju izvor uparuje sa IP adresom. U ARP zahtjevu se navodi adresa routera0(10.0.0.1) jer jedino nju vidi pc1, zna mu IP adresu, ali ne zna MAC adresu, te prilikom preusmjeravanja router0 vidi da se traži njegova MAC adresa te ju šalje.

Višedredišne adrese u protokolu Ethernet služe za slanje podataka čvorovima, njih protokol ARP ne koristi.

**Zadatak 7.** (Topologija Ping/ping.imn) Proučite utjecaj raznih parametara, koje je moguće proslijediti naredbi ping, na sadržaj paketa koji se šalju. Parametri se mogu dobiti izvođenjem naredbe ping bez argumenata ili na stranici s uputama koja se dobiva izvršavanjem naredbe man ping. Komentirajte parametre ukratko.



c- broj paketa koji se šalju

G- maksimalna veličina ICMP datagrama prilikom sweepa

g-minimalna veličina ICMP datagrama prilikom sweepa

h-određuje koliko se veličina ICMP datagrama povećava nakon svake iteracije sweepa

i-interval između slanja pingova

m-postavljanje TTL vrijednosti paketa na određeni iznos

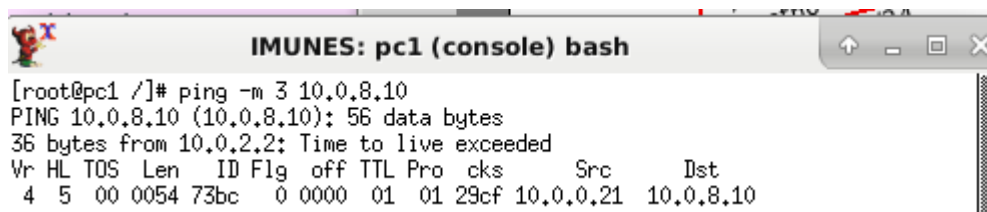
s-veličina paketa koji se šalje

t-određuje vrijeme izvođenja ping-a, kada dođe do tog vremena automatski se gasi

W-vrijeme čekanja na odgovor(ms)

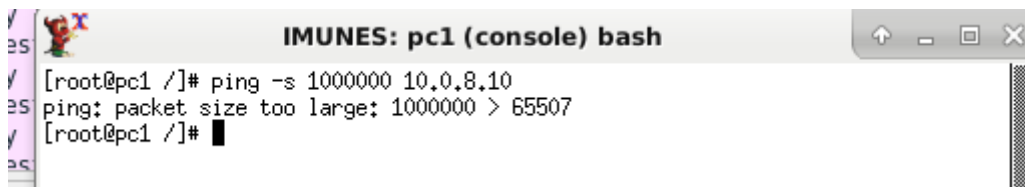
**Zadatak 8.** (Topologija Ping/ping.imn) Utvrdite i objasnite što se događa pri slanju paketa alatom ping koji u polju TTL imaju vrijednost 3, a određišno računalo je neko računalo udaljeno više od 3 „skoka“.

Svakim skokom će se TTL smanjivati te će se smanjiti na 0 , čvor će odbaciti paket te vratiti poruku (“Time to live exceeded”).



Vr	HL	TOS	Len	ID	Flg	off	TTL	Pro	cks	Src	Dst
4	5	00	0054	73bc	0	0000	01	01	29cf	10.0.0.21	10.0.8.10

**Zadatak 9.** (Topologija Ping/ping.imn) Utvrdite i objasnite što se događa kad je ping paket koji se šalje velik 10000 okteta. Kolika je maksimalna moguća veličina paketa koji se može postaviti prilikom izvršavanja naredbe ping? O čemu ona ovisi?



Kada se šalje paket ping veličine 10000 okteta dodaje se još 8 okteta za ICMP protokol.

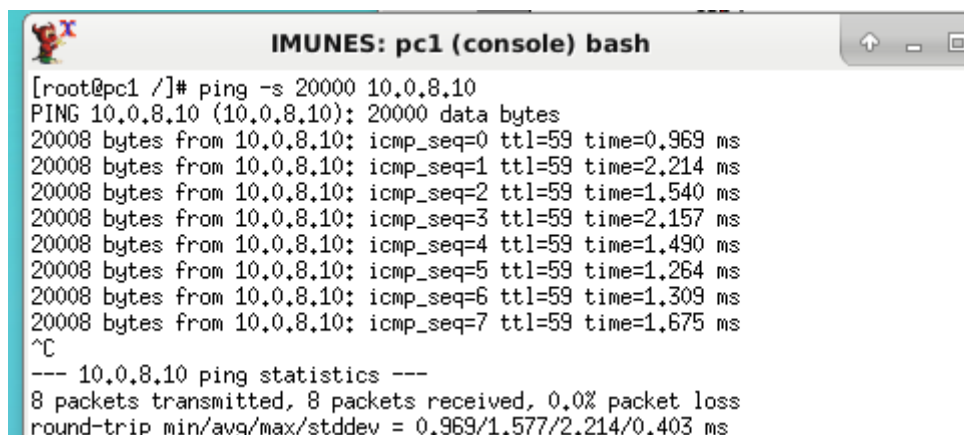
Kako je MTU 1500 okteta paket će se fragmetirati na  $\text{ceiling}(10000/1500)$ , odnosno na 7 fragmenata. Maksimalna veličina paketa naredbe ping je 65507 na što se dodaje još 8 okteta pa ispadne 65515.

Ona ovisi o veličini zaglavlja u IPv4(20 okteta) ->  $65515 = 2^{16} - 1 - 20$

**Zadatak 10.** (Topologija Traceroute/traceroute.imn) Utvrdite i objasnite kako veličina paketa koji se šalje utječe na vrijeme koje prijavljuje alat ping (tzv. ping time). Ispitajte kako se mijenjaju vrijednosti koje vraća alat ping, ako se u mreži izravno spoje dva usmjeritelja koja prije nisu bila izravno povezana (npr. računalo pc1 provjerava dostupnost poslužitelja server bez i uz postojanje izravne veze između usmjeritelja router0 i router7)?

Povećavanjem veličine paketa povećava se i ping time.

Slučajevi bez izravne veze:



Prosječno vrijeme(20000 okteta) : 1.577 ms

```
[root@pc1 /]# ping -s 50000 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 50000 data bytes
50008 bytes from 10.0.8.10: icmp_seq=0 ttl=59 time=0.859 ms
50008 bytes from 10.0.8.10: icmp_seq=1 ttl=59 time=0.855 ms
50008 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=2.025 ms
50008 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=6.488 ms
50008 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=1.655 ms
50008 bytes from 10.0.8.10: icmp_seq=5 ttl=59 time=2.653 ms
50008 bytes from 10.0.8.10: icmp_seq=6 ttl=59 time=2.586 ms
50008 bytes from 10.0.8.10: icmp_seq=7 ttl=59 time=2.195 ms
^C
--- 10.0.8.10 ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.855/2.414/6.488/1.670 ms
[root@pc1 /]#
```

Prosječno vrijeme(50000 okteta) : 2.414 ms

Slučajevi sa izravnom vezom :

```
[root@pc1 ~]# ping -s 20000 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 20000 data bytes
20008 bytes from 10.0.8.10: icmp_seq=0 ttl=62 time=0.245 ms
20008 bytes from 10.0.8.10: icmp_seq=1 ttl=62 time=1.180 ms
20008 bytes from 10.0.8.10: icmp_seq=2 ttl=62 time=1.540 ms
20008 bytes from 10.0.8.10: icmp_seq=3 ttl=62 time=1.881 ms
20008 bytes from 10.0.8.10: icmp_seq=4 ttl=62 time=1.470 ms
20008 bytes from 10.0.8.10: icmp_seq=5 ttl=62 time=1.702 ms
20008 bytes from 10.0.8.10: icmp_seq=6 ttl=62 time=1.492 ms
20008 bytes from 10.0.8.10: icmp_seq=7 ttl=62 time=1.014 ms
^C
--- 10.0.8.10 ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.245/1.316/1.881/0.478 ms
```

Prosječno vrijeme(20000 okteta) : 1.316 ms

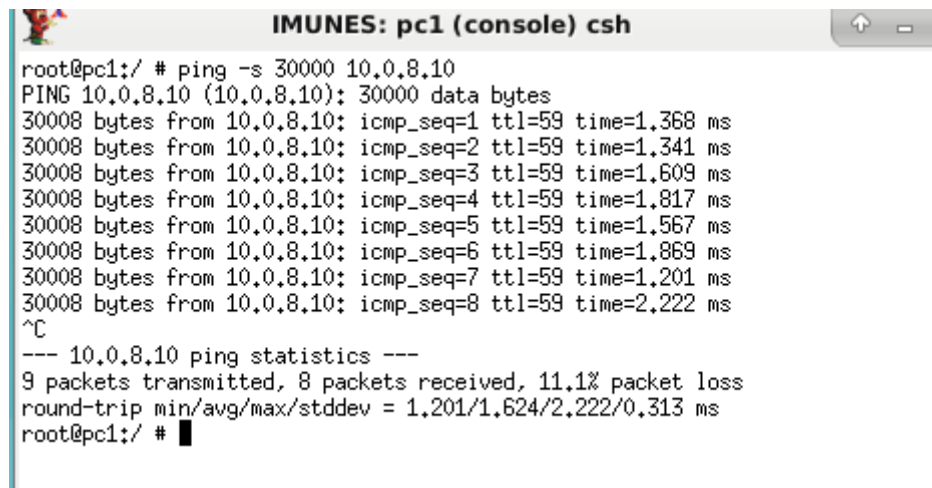
```
[root@pc1 ~]# ping -s 50000 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 50000 data bytes
50008 bytes from 10.0.8.10: icmp_seq=0 ttl=62 time=0.465 ms
50008 bytes from 10.0.8.10: icmp_seq=1 ttl=62 time=1.157 ms
50008 bytes from 10.0.8.10: icmp_seq=2 ttl=62 time=1.614 ms
50008 bytes from 10.0.8.10: icmp_seq=3 ttl=62 time=2.154 ms
50008 bytes from 10.0.8.10: icmp_seq=4 ttl=62 time=1.082 ms
50008 bytes from 10.0.8.10: icmp_seq=5 ttl=62 time=2.243 ms
50008 bytes from 10.0.8.10: icmp_seq=6 ttl=62 time=2.398 ms
50008 bytes from 10.0.8.10: icmp_seq=7 ttl=62 time=0.995 ms
50008 bytes from 10.0.8.10: icmp_seq=8 ttl=62 time=0.461 ms
^C
--- 10.0.8.10 ping statistics ---
9 packets transmitted, 9 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.461/1.397/2.398/0.699 ms
```

Prosječno vrijeme(50000 okteta) : 1.397 ms

**Zadatak 11.** (*Topologija Traceroute/traceroute.imn*) Utvrdite i objasnite kako propagacijsko kašnjenje utječe na vrijeme koje prijavljuje alat ping (tzv. ping time). Ispitajte kako se mijenjaju vrijednosti vremena koje vraća alat ping, ako se u mreži promijeni propagacijsko kašnjenje između računala i ethernetskog komutatora (npr. računalo pc1 provjerava dostupnost poslužitelja server uz različito podešeno propagacijsko kašnjenje između računala pc1 i ethernetskog komutatora lanswitch8)?

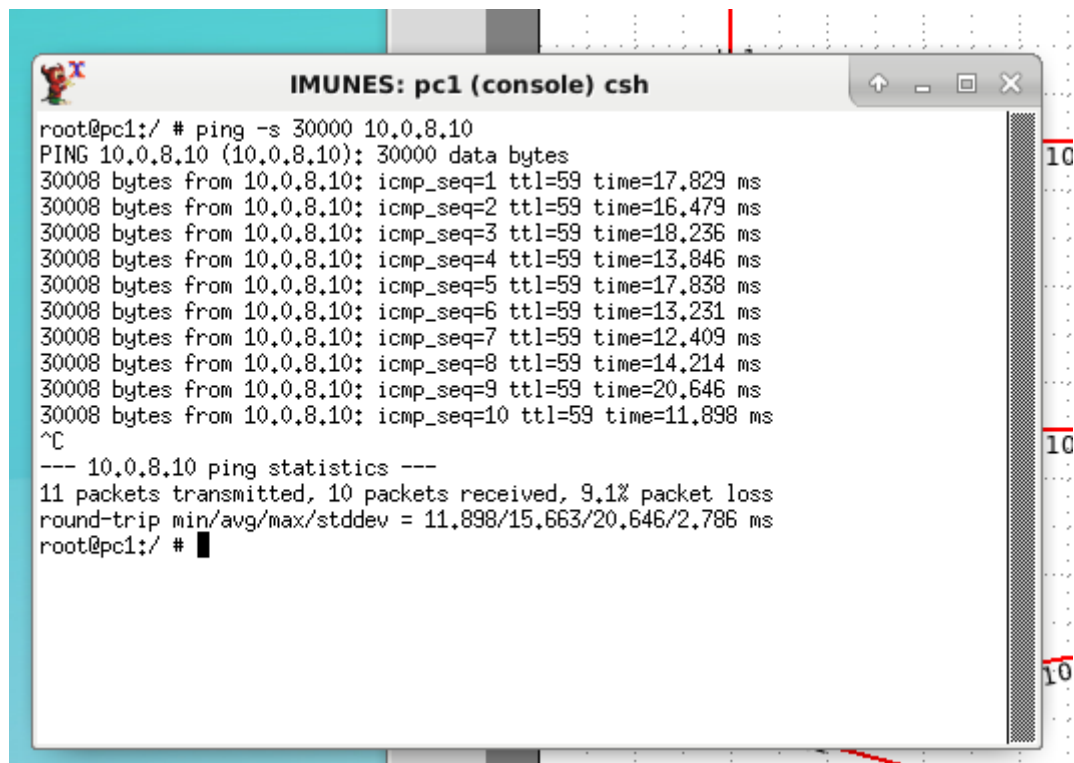
Bez kašnjenja :





```
root@pc1:/ # ping -s 30000 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 30000 data bytes
30008 bytes from 10.0.8.10: icmp_seq=1 ttl=59 time=1.368 ms
30008 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=1.341 ms
30008 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=1.609 ms
30008 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=1.817 ms
30008 bytes from 10.0.8.10: icmp_seq=5 ttl=59 time=1.567 ms
30008 bytes from 10.0.8.10: icmp_seq=6 ttl=59 time=1.869 ms
30008 bytes from 10.0.8.10: icmp_seq=7 ttl=59 time=1.201 ms
30008 bytes from 10.0.8.10: icmp_seq=8 ttl=59 time=2.222 ms
^C
--- 10.0.8.10 ping statistics ---
9 packets transmitted, 8 packets received, 11,1% packet loss
round-trip min/avg/max/stddev = 1,201/1,624/2,222/0,313 ms
root@pc1:/ #
```

Sa kašnjenjem (15 us) :



```
root@pc1:/ # ping -s 30000 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 30000 data bytes
30008 bytes from 10.0.8.10: icmp_seq=1 ttl=59 time=17.829 ms
30008 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=16.479 ms
30008 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=18.236 ms
30008 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=13.846 ms
30008 bytes from 10.0.8.10: icmp_seq=5 ttl=59 time=17.838 ms
30008 bytes from 10.0.8.10: icmp_seq=6 ttl=59 time=13.231 ms
30008 bytes from 10.0.8.10: icmp_seq=7 ttl=59 time=12.409 ms
30008 bytes from 10.0.8.10: icmp_seq=8 ttl=59 time=14.214 ms
30008 bytes from 10.0.8.10: icmp_seq=9 ttl=59 time=20.646 ms
30008 bytes from 10.0.8.10: icmp_seq=10 ttl=59 time=11.898 ms
^C
--- 10.0.8.10 ping statistics ---
11 packets transmitted, 10 packets received, 9,1% packet loss
round-trip min/avg/max/stddev = 11,898/15,663/20,646/2,786 ms
root@pc1:/ #
```

Iz priloženih slika se vidi da propagacijsko kašnjenje povećava vrijeme izvođenja ping-a.

**Zadatak 12.** (*Topologija Ping/ping.imn*) U emulatoru/simulatoru IMUNES proučite i detaljno analizirajte uhvaćeni slijed paketa koji je generirao alat ping između različitih računala u mreži. Utvrdite koji su sve protokoli iskorišteni kao posljedica izvođenja naredbe ping i koji je odnos među njima (tj., koje druge protokole svaki pojedini protokol koristi). Navedite kojem sloju TCP/IP-modela svaki od tih protokola pripada.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	10.0.0.1	224.0.0.9	RIPv2	66	Request
2	9.879533989	10.0.0.1	224.0.0.9	RIPv2	186	Response
3	10.267278643	RealtekA 00:00:10	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.21
4	10.267288792	Cisco 00:00:00	RealtekA 00:00:10	ARP	42	10.0.0.1 is at c8:4c:75:00:00:00
5	10.267297228	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=0/0, ttl=64 (reply in 6)
6	10.267361555	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=0/0, ttl=59 (request in 5)
7	11.269509622	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=1/256, ttl=64 (reply in 8)
8	11.269623831	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=1/256, ttl=59 (request in 7)
9	12.277423582	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=2/512, ttl=64 (reply in 10)
10	12.277728891	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=2/512, ttl=59 (request in 9)
11	13.290253647	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=3/768, ttl=64 (reply in 12)
12	13.291148306	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=3/768, ttl=59 (request in 11)
13	14.301840061	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=4/1024, ttl=64 (reply in 14)
14	14.303306972	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=4/1024, ttl=59 (request in 13)
15	15.310481285	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=5/1280, ttl=64 (reply in 16)
16	15.311118416	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=5/1280, ttl=59 (request in 15)
17	16.329429160	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=6/1536, ttl=64 (reply in 18)
18	16.329520848	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=6/1536, ttl=59 (request in 17)
19	17.339826348	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=7/1792, ttl=64 (reply in 20)
20	17.340367160	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=7/1792, ttl=59 (request in 19)
21	18.340781499	10.0.0.21	10.0.8.10	ICMP	98	Echo (ping) request id=0x5621, seq=8/2048, ttl=64 (reply in 22)
22	18.350155307	10.0.0.10	10.0.0.21	ICMP	98	Echo (ping) reply id=0x5621, seq=8/2048, ttl=59 (request in 21)
23	31.968659934	Fe80::ca4c:75ff:fe0... ff02::9		RIPng	226	Command Response, Version 1

Vidimo da su na slici RIP protokol, ICMP, i ARP. RIP se koristi za prijenos IP datagrama kroz mrežu.

ARP šalje broadcast čime saznaje MAC adresu, u ovom slučaju sučelja router0. Prilikom slanja paketa koristi se ICMP protokol sa Echo request i Echo reply. Echo request šalje paket sa izvora na odredište, a Echo reply sa odredišta na izvor. Svi ovi protokoli pripadaju mrežnom sloju.

**Zadatak 13. (Topologija Ping/ping.imn)** Utvrdite što se sve mijenja u okviru protokola Ethernet kad se koristi naredba ping s različitim veličinama paketa koji se šalju.

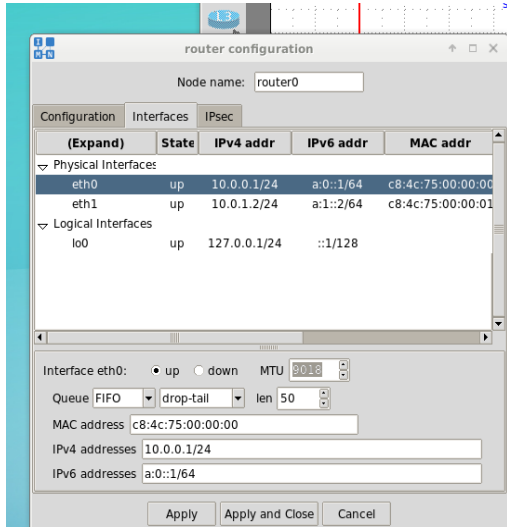
Mijenja se ping-time (što je veći paket veći je ping-time), postoji mogućnost fragmentacije te se mijenjaju Frame Length i Capture Length.

**Zadatak 14. (Topologija Ping/ping.imn)** Utvrdite kakav se promet generira na ethernetskom sučelju računala kad se provjerava dostupnost (naredba ping) adrese 127.0.0.1. Komentirajte rezultat.

The screenshot displays two windows from a network simulation. The left window is Wireshark, capturing traffic on the 'Loopback: lo0 (on pc1)' interface. It shows a series of ICMP Echo (ping) requests and replies between 127.0.0.1 and 127.0.0.1. The right window is IMUNES, showing a network diagram with two PCs, 'pc1' and 'pc2', connected to a switch 'lanswitch9'. The switch is connected to a server 'ser'. The console window for 'IMUNES: pc1 (console) csh' shows the execution of a 'ping 127.0.0.1' command, displaying the results of 10 ping attempts, including packet size, sequence number, and round-trip time.

Adresa 127.0.0.1 je loopback adresa te se paketi koji se pošalju na ovu adresu ne šalju dalje već se vraćaju nazad u mrežnom sloju.

**Zadatak 15.** (*Topologija Ping/ping.imn*) Utvrdite kolike su minimalna i maksimalna vrijednost MTU-a (Maximum Transfer Unit) na ethernetском sučelju. Pokušajte podesiti MTU i veličinu ping paketa tako da ostvarite što veći broj fragmenata. Način podešavanja MTU-a pronađite u uputama naredbe `ifconfig(8)`, dakle, izvršenjem naredbe `man ifconfig`.



Maksimalna veličina MTU je 9018, a minimalna 256. Najveći broj fragmenata se dobije za najmanji MTU(256) te najveću veličinu paketa(65507).

**Zadatak 16.** (*Topologija Traceroute/traceroute.imn*) Utvrdite neke od mogućih situacija u kojima alat `traceroute` može proizvesti rezultat koji nije ispravan (naputak: pogledajte što piše u uputama alata – izvršite naredbu `man traceroute`).

- Može se ruta promijeniti, postoji mogućnost da neki router blokira određeni protokol te se ruta ne može naći. Ako se dogodi da netko prisluškuje port može doći do greške `PORT UNREACHABLE`.

**Zadatak 17.** (*Topologija Traceroute/traceroute.imn*) U emulatoru/simulatoru IMUNES, ispitajte način rada alata `traceroute` na mreži iz primjera `Traceroute/traceroute.imn`. Potrebno je snimati mrežni promet na pojedinim sučeljima i utvrditi mehanizam na kojem se temelji rad alata. Kako se koristi TTL-polje i protokol ICMP?

3	10.972174047	10.0.0.1	10.0.8.10	UDP	54	42767	→	33435	Len=12
4	10.972211004	10.0.0.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
5	10.973681698	10.0.0.1	10.0.8.10	UDP	54	42767	→	33436	Len=12
6	10.973689071	10.0.0.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
7	10.973788774	10.0.0.1	10.0.8.10	UDP	54	42767	→	33437	Len=12
8	10.973792071	10.0.0.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
9	10.973935905	10.0.0.1	10.0.8.10	UDP	54	42767	→	33438	Len=12
10	10.973952876	10.0.1.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
11	10.974312083	10.0.0.1	10.0.8.10	UDP	54	42767	→	33439	Len=12
12	10.974321590	10.0.1.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
13	10.974425369	10.0.0.1	10.0.8.10	UDP	54	42767	→	33440	Len=12
14	10.974434346	10.0.1.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
15	10.974586102	10.0.0.1	10.0.8.10	UDP	54	42767	→	33441	Len=12
16	10.974606329	10.0.2.2	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
17	10.974943684	10.0.0.1	10.0.8.10	UDP	54	42767	→	33442	Len=12
18	10.974957740	10.0.2.2	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
19	10.975039080	10.0.0.1	10.0.8.10	UDP	54	42767	→	33443	Len=12
20	10.975049519	10.0.2.2	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
21	10.975184869	10.0.0.1	10.0.8.10	UDP	54	42767	→	33444	Len=12
22	10.975209575	10.0.3.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
23	10.975566457	10.0.0.1	10.0.8.10	UDP	54	42767	→	33445	Len=12
24	10.975585051	10.0.3.1	10.0.8.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)			
25	10.975666611	10.0.0.1	10.0.8.10	UDP	54	42767	→	33446	Len=12

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0  
Ethernet II, Src: 42:00:00:00:00:00 (42:00:00:00:00:00), Dst: 08:00:00:00:00:00 (08:00:00:00:00:00)

Primjer na sučelju `router0`. Prvo se šalje paket sa `TTL = 1`, koji staje na `10.0.0.1`, taj paket se šalje sa tri porta te je zato u `Wireshark`-u ispisano 3 puta. Nakon toga `ICMP` javlja „Time to live exceeded“ te se

paket odbacuje. Ponovno slanje paketa je sa TTL = 2 koji staje na 10.0.1.1 te tamo javlja TTL exceeded. Svaki novi put se TTL postepeno povećava dok ICMP ne javi da je poruka stigla.

**Zadatak 18.** (*Topologija Traceroute/traceroute.imn*) Na koji način protokol IP „pamti“ vrstu paketa koji se prenosi u podatkovnom dijelu IP-datagrama? Navedite primjere različitih paketa iz podatkovnog dijela IP-datagrama.

- Protokol IP pamti vrstu paketa tako što u zaglavlju IP datagrama se zapisuje vrsta paketa koji se prenosi IP podatkovnim dijelom. Npr. TCP i UDP.

**Zadatak 19.** (*Topologija Traceroute/traceroute.imn*) Utvrdite postoji li način da se iz primljenog IP-paketa očita put kojim je paket prošao kroz mrežu.

- Ne postoji, zato što IP paket ne pamti put kojim je prošao kroz mrežu, to se radi naredbom traceroute.

**Zadatak 20.** (*Topologija Traceroute/traceroute.imn*) Utvrdite postoji li način kojim protokol IP može ustanoviti da je poslani paket stvarno i primljen na odredištu.

- ICMP šalje Echo reply izvoru i time signalizira da je paket primljen.

**Zadatak 21.** (*Topologija Ping/ping.imn*) Utvrdite utječe li fragmentacija na propusnost i kašnjenje te komentirajte dobivene rezultate.

- Fragmentacija smanjuje propusnost te povećava kašnjenje.