

Labos 1

Komunikacijske mreže

1. Na eth0 na pc2 pronašao sam izvorišnu MAC adresu **c8:4c:75:00:00:00**. OUI čine prva 3 okteta (**c8:4c:75**), a NIC čine druga 3 okteta (**00:00:00**). Pretragom interneta (<https://www.macvendorlookup.com>) ustanovio sam da se radi o Ciscovoj mrežnoj kartici.

Company	CISCO SYSTEMS, INC.
Address	170 W. TASMAN DRIVE M/S SJA-2 SAN JOSE CA 95134-1706 UNITED STATES
Range	C8:4C:75:00:00:00 - C8:4C:75:FF:FF:FF
Type	IEEE MA-L

2. Odabran Ethernet-okvir ima 14 okteta zaglavlja. Prvih 6 okteta predstavljaju **odredišnu MAC adresu** (01:00:5e:00:00:09). Drugih 6 predstavljaju **izvorišnu MAC adresu** (ista kao i u 1. odgovoru). **EtherType**, tj. zadnja 2 okteta (08:00) označuju protokol, u ovom slučaju IPv4. Na wiresharku nije prikazana preambula jer je ona služi fizičkom sloju.

```
0000 01 00 5e 00 00 09 c8 4c 75 00 00 00 08 00 45 c0
0010 00 ac 97 81 00 00 01 11 36 f6 0a 00 00 01 e0 00
0020 00 09 02 08 02 08 00 98 b0 78 02 02 00 00 00 02
0030 00 00 0a 00 01 00 ff ff ff 00 00 00 00 00 00 00
```

3. Protokol Ethernet zapisuje vrstu paketa u zadnja 2 okteta zaglavlja okvira.

4. Ispis naredbe **traceroute 10.0.8.10** na pc1:

```
[root@pc1 ~]# 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) 0.324 ms 0.272 ms 0.104 ms
 2 10.0.1.1 (10.0.1.1) 0.255 ms 0.136 ms 0.107 ms
 3 10.0.4.1 (10.0.4.1) 0.158 ms 6.659 ms 0.396 ms
 4 10.0.3.1 (10.0.3.1) 0.199 ms 0.074 ms 0.054 ms
 5 10.0.7.2 (10.0.7.2) 0.191 ms 0.101 ms 0.057 ms
 6 10.0.8.10 (10.0.8.10) 0.197 ms 0.077 ms 0.061 ms
```

Čitanjem IP adresa zaključujem da je najvjerojatnija ruta:

pc1 -> router0 -> router1 -> router5 -> router6 -> router7 -> server

Ispis naredbe **traceroute 10.0.0.21** na server:

```
[root@server ~]# traceroute 10.0.0.21
traceroute to 10.0.0.21 (10.0.0.21), 64 hops max, 40 byte packets
 1 10.0.8.1 (10.0.8.1) 0.282 ms 0.157 ms 0.148 ms
 2 10.0.7.1 (10.0.7.1) 0.201 ms 0.165 ms 0.130 ms
 3 10.0.3.2 (10.0.3.2) 0.205 ms 0.380 ms 5.219 ms
 4 10.0.4.2 (10.0.4.2) 0.382 ms 0.069 ms 0.055 ms
 5 10.0.1.2 (10.0.1.2) 0.092 ms 0.065 ms 0.067 ms
 6 10.0.0.21 (10.0.0.21) 0.091 ms 0.071 ms 0.060 ms
```

Čitanjem IP adresa zaključujem da je najvjerojatnija ruta:

server -> router7 -> router6 -> router2 -> router1 -> router0 -> pc1

5. Paketi se prenose protokolima **TCP** i **UDP**. Također se koristi protokol **RIP** za određivanje optimalnog puta paketa. Uz to, protokol **ARP** pretvara **IP** adrese u **MAC** adrese. Pošiljalatelj želi poslati okvir uređaju koji se ne nalazi u njegovoj podmreži. Kako primatelj nije u podmreži, pošiljalatelj ne može odmah uspostaviti vezu, već mora uređajima na mreži slati upit s **MAC**

adresom. Kada traženi uređaj (ili neki uređaj s njegove pod mreže) prepozna upit, on dojavljuje početnom pošiljatelju da je pronađen pomoću MAC adrese iz zaglavlja. Uređaji koji nisu traženi, jednostavno će ignorirati upit ili će ga proslijediti, sve dok upit ne dođe do pod mreže traženog.

6. Prilikom pinganja servera s pc2 (10.0.0.20) ARP zahtjev poslan je na **ff:ff:ff:ff:ff:ff**. To je oznaka da se upit šalje svima povezanim s tim pc-om (broadcast). Na upit je reagirao router0 (10.0.1.1) te uzvratio ARP odgovor sa svojom MAC adresom (c8:4c:75:00:00:00). pc2 sada zna da ako pinga server (10.0.8.10) zahtjev treba poslati na router0 koji će se pobrinuti da zahtjev zaista dođe do servera.

4	21.321675012	HewlettP_00:00:0f	Broadcast	ARP	42
5	21.321675151	Cisco_00:00:00	HewlettP_00:00:0f	ARP	42

Višeodrešne adrese služe za slanje istog paketa na više odredišta na mreži. Obzirom da ARP služi za određivanje pojedine adrese, on ne koristi višeodrešne adrese, već jednodrešne.

7. Naredba **ping** ima mnoštvo opcija, a ovo su neke:

Broj paketa za slanje: *-c*
Mod za debugging socketa: *-d*
Brzo slanje paketa: *-f*
Maksimalna veličina ICMP paketa: *-G*
Početna veličina ICMP paketa: *-g*
Količina inkrementacije ICMP paketa: *-h*
Interval čekanja između slanja pingova: *-i*
IP Time To Live odlazećih paketa: *-m*
Zastavica za završetak nakon prvog primljenog odgovora: *-o*
Uzorak za popunjavanje paketa: *-p*
Tihi način ispisa na standardni izlaz (bez ICMP error poruka): *-Q*
Zastavica za snimanje rute: *-R*
Adresa pošiljatelja: *-S*
Veličina (u bajtovima) paketa za slanje: *-s*
Vrijeme isteka za zahtjev: *-t*
Detaljan način ispisa na standardni izlaz: *-v*
Vrijeme čekanja na odgovor: *-w*

8. TTL s vrijednošću 3 omogućuje najviše 3 skoka za paket od izvorišta do odredišta. Dakle, usmjeritelj na kojem će umanjujući TTL doći do 0 će odbaciti paket. ICMP će poslati **izvješće o grešci** pošiljatelju.

9. Zadan je MTU od 1500 okteta. Stoga paket od 10000 okteta mora biti podijeljen na 7 dijelova. Maksimalna veličina paketa koju je moguće postaviti za ping je 65507 bajtova.

```
[root@pc2 /]# ping -s 65508 10.0.8.10
ping: packet size too large: 65508 > 65507
[root@pc2 /]# ping -s 65507 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 65507 data bytes
65515 bytes from 10.0.8.10: icmp_seq=0 ttl=59 time=2.511 ms
```

Konačni paket bit će veći za 8 bajtova. Također još preostaje 20 bajtova za IPv4 zaglavlje. To sveukupno čini **65535** ($= 2^{16} - 1$) bajtova, što je maksimalna veličina IPv4 datagrama (jer polje *size* ima veličinu 16 bitova pa ne može zapisati veći broj).

10. Veličina paketa produljuje ping time, no te dvije veličine nisu potpuno proporcionalne. Za manje brojeve (veličina < 1000) ping time je praktički konstantan na oko 0.2 ms. Na većim brojevima moguće je utvrditi **donekle lineranu zavisnost**. Primjerice za 10000 potrebno je 0.5 ms, za 20000 0.9 ms, a za 40000 1.5 ms. Ako izravno povežemo router0 i router7, ping time se smanjuje redom za 0.2, 0.5 te 0.7 ms. U slučaju s 40000 bajtova to je dvostruko manje vrijeme. To je i bilo očekivano jer smo smanjili broj skokova po paketu (za 3).

11. Čak i male vrijednosti propagacijskog kašnjenja znatno utječu na ping time. Ako na poveznici pc1 - lanswitch8 postavimo kašnjenje od 5 us, ping time poveća se preko 20 puta. To je posljedica toga što kašnjenje djeluje u oba smjera, a znatno su usporeni paketi koji su fragmentirani jer kašnjenje djeluje na svaki fragment.

12. Prilikom prvog pinga protokol ARP javlja pc1 koja je MAC adresa od pc2. ARP i MAC pripadaju sloju podatkovne poveznice. Zatim pc1 šalje *echo request* do pc2, a pc2 uzvraća *echo reply* do pc1. To se postiže protokolom ICMP koji je dio internetskog sloja. U pozadini sve prolazi Ethernet sučeljem koje je dio fizičkog sloja. Slanje paketa, odnosno datagram ostvaren je internetskim protokolom.

13. Kada povećavamo veličinu paketa, povećava se i veličina okvira, poslanog i primljenog. Naravno, povećavaju se i vremenske delte između okvira, a može doći i do fragmentacije paketa.

14. Kada ping šaljemo na 127.0.0.1, adresa poznata kao *localhost*, na ethernetskom sučelju nema prometa. pc zapravo šalje ping sam sebi, odnosno svom lokalnom sučelju, pri čemu ne kontaktira mrežu.

15. IPv4 standard zahtjeva da MTU bude minimalno 68 bajtova, no IMUNES ne dozvoljava MTU manji od 72. Teoretski, MTU nema gornju granicu, no IMUNES ne dopušta vrijednost veću od 9018. Naredba koju sam koristio jest `ifconfig etho0 mtu <broj>`.

16. Alat **tracert** može dati neispravan rezultat. Prilikom slanja protokolom koji nije UDP, zadnji odgovor se ignorira jer se za njega ne šalje ICMP poruka, te se čini da je paket izgubljen. S druge strane, tracert može dati neispravan rezultat zbog *bugova* u sustavu. Neki usmjeritelji mogu imati neispravan kernel te prosljeđivati pakete kojima je TTL nula. Također, neki usmjeritelji blokiraju pojedine protokole što onemogućava prolazak upita.

BUGS

When using protocols other than UDP, functionality is reduced. In particular, the last packet will often appear to be lost, because even though it reaches the destination host, there's no way to know that because no ICMP message is sent back. In the TCP case, **tracert** should listen for a RST from the destination host (or an intermediate router that's filtering packets), but this is not implemented yet.

The AS number capability reports information that may sometimes be inaccurate due to discrepancies between the contents of the routing database server and the current state of the Internet.

17. Pokretanjem **tracert** na pc1 događa se sljedeće: Prvo se šalje ARP zahtjev do 10.0.0.1 (router0). Usmjeritelj vraća MAC adresu. Zatim se šalje upit protokolom UDP do adrese zadane argumentom naredbe. Prvi upit ima TTL = 1. Nakon toga pc1 čeka na ICMP odgovor. Ukoliko je odgovor negativan, šalje se upit s **TTL-om za 1 većim**. Ovaj postupak se ponavlja sve dok pc1 ne dobije pozitivan odgovor, odnosno dok uređaj s traženom adresom nije pronađen.

18. Vrsta paketa zapisana je u zaglavlju IP paketa. To su na primjer TCP, UDP, RAW, ICMP, IGMP i tako dalje.

19. Putanja IP paketa ne zapisuje se na neko jedinstveno mjesto. Sam paket ne sadži takvu informaciju. Putanja bi se eventualno mogla rekonstruirati čitanjem logova na svakom usmjeritelju kojim paket potencijalno može proći.

20. Postoji mehanizam kojim protokol IP ustvrđuje je li paket primljen na odredištu. Za to postoji **Internet Control Message Protocol**. Kao što je već spomenuto do sada, koristi se dinamika *echo request* <--> *echo reply*.

21. Fragmentacija negativno utječe na performanse komunikacijske mreže. Obzirom da se poruka rascjepka te se svaki dio šalje zasebno, rezultat je **povećano kašnjenje**. Posljedično je manja ukupna propusnost poruka.