Zadatak 1. (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 porizvođača pripadajuće mrežne kartice korištenjem web-tražilice.

Uzmimo npr. adresu **C8:4C:75:00:00:00**.

OUI ove adrese su prva 3 okteta, tj. **C8:4C:75**.
NIC ove adresu su druga 2 okteta, tj. **00:00:00**.

Preko OUI možemo saznati proizvođača mrežne kartice jer je OUI unikatan za jednog proizvođača. Internet pretragom doznajemo da OUI C8:4C:75 pripada proizvođaču **Cysco Systems, Inc**. Za pretragu sam koristio *https://www.wireshark.org/tools/oui-lookup.html*

Zadatak 2. (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-stranice: *http://wiki.wireshark.org/Ethernet*)

Označio sam isti paket kao u prvo zadatku. Otišao sam na Ethernet II tab u kojem je Wireshark izbrojao 14 okteta. Prvih 6 okteta su odredište paketa (**33:33:00:00:00:09**), drugih 6 okteta su izvor paketa (**C8:4C:75:00:00:00**), a zadnjih 2 okteta govore o tipu internetskog protokola, u našem slučaju IPv6 (**86:DD**).

Zadatak 3.

Na koji način protokol Ethernet „pamti“ vrstu paketa koji se prenosi u podatkovnom dijelu Ethernet-okvira?

Vrstu paketa Ethernet zapisuje u zadnja 2 okteta zaglavlja Ethernet-okvira.

Zadatak 4. (Traceroute/traceroute.imn)

U emulatoru/simulatoru IMUNES, ispitajte način rada alata tracerout.

1. Započnite simulaciju.
2. Pokrenite alat *Wireshark* na sučelju *eth0* računala *pc1* i započnite snimanje mrežnog prometa.
3. Otvorite konzolu na računalu *pc1*.
4. Provjerite najvjerojatniji put (naredba *traceroute*) od računala *pc1* do poslužitelja *server* (10.0.8.10). Analizirajte odgovor koji je dobilo računalo *pc1* u sklopu izvršavanja naredbe *traceroute*. Provjerite IP-adrese sučelja na usmjeriteljima uključenim u usmjeravanje paketa generiranih alatom *traceroute*.
5. Otvorite konzolu na poslužitelju *server*.
6. Provjerite najvjerojatniji put (naredba *traceroute*) od poslužitelja *server* do računala *pc1* (10.0.0.21). Analizirajte odgovor koji je dobio poslužitelj server u sklopu izvršavanja naredbe *traceroute*. Provjerite IP-adrese sučelja na usmjeriteljima uključenim u usmjeravanje paketa generiranih alatom *traceroute*.
7. Usporedite rezultate dobivene izvršavanjem naredbe *traceroute* u koracima 4 i 6 te ih komentirajte.

Traceroute *pc1* -> *server* vraća sljedeće:

1	10.0.0.1	(10.0.0.1)	0.173 ms	0.022 ms	0.016 ms
2	10.0.1.1	(10.0.1.1)	29.433 ms	0.029 ms	0.016 ms
3	10.0.2.2	(10.0.2.2)	0.023 ms	0.022 ms	0.017 ms
4	10.0.3.1	(10.0.3.1)	0.024 ms	0.026 ms	0.020 ms
5	10.0.7.2	(10.0.7.2)	0.026 ms	0.026 ms	0.022 ms
6	10.0.8.10	(10.0.8.10)	0.043 ms	0.173 ms	0.039 ms

Vidimo da prilikom komunikacije sa 10.0.1.1 jedno od vremena između slanja poruke i primitka poruke o grešci je iznimno veliko. Ovo je zbog kreiranja ICMP poruke, tj. overheada na procesoru. Ponekad znamo imati i 10 puta veći delay od uobičajenog, vjerojatno zbog istog razloga. Vidimo da između *pc1* i *servera* imamo 5 međučvorova.

Finalno, ruta je *pc1* -> *router0* -> *router1* -> *router2* -> *router6* -> *router7* -> *server*

Traceroute server -> pc1 vraća sljedeće:

1	10.0.8.1	(10.0.8.1)	0.109 ms	0.083 ms	0.143 ms
2	10.0.7.1	(10.0.7.1)	0.076 ms	0.078 ms	0.204 ms
3	10.0.3.2	(10.0.3.2)	0.076 ms	0.187 ms	0.080 ms
4	10.0.2.1	(10.0.2.1)	0.080 ms	0.201 ms	0.078 ms
5	10.0.1.2	(10.0.1.2)	0.074 ms	0.081 ms	0.079 ms
6	10.0.0.21	(10.0.0.21)	0.083 ms	0.084 ms	0.084 ms

Vidimo da se u ovom slučaju ruta gotovo i ne razlikuje od obrnute, samo izlazi i ulazi u usmjeritelje imaju drukčije IP adrese. Ovog puta nema jako velikog odstupanja u delayu – sve vrijednosti su tu negdje. Razlikuje se eth sučelje, ali su usmjeritelji isti. Još uvijek imamo 5 međučvorova.

Ruta je *server -> router7 -> router6 -> router2 -> router1 -> router0 -> pc1*

Zadatak 5. (Traceroute/traceroute.imn)

Kojim protokolom se IP-paketi prenose između računala smještenih unutar jedne lokalne mreže? Čemu, pri tome, služi protokol ARP?

Prenose se TCP-em, ali i u posebnim slučajevima UDP-om. ARP pretvara IP adrese u MAC adrese. Kada pošiljatelj shvati da mu se primatelj ne nalazi u podmreži (nema isti prefiks), on pita mrežu tko ima adresu odredišta. Kada taj upit dođe do uređaja koji nema istu adresu, primatelj tog upita šuti, a kad se upit pošalje na adresu primatelja tog zahtjeva, on tada podnositelju zahtjeva, kojemu zna MAC adresu jer je priložena u zaglavlju, dojavljuje da je on onaj kojega traži.

Zadatak 6. (Ping/ping.imn)

Ponovite pokazni eksperiment s početka poglavlja 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 višeodredišne adrese u protokolu Ethernet? Koristi li ih protokol ARP?

Prilikom pinganja 10.0.8.10, 10.0.0.21 šalje ARP zahtjev kao broadcast (svima) da mu netko kaže tko ima IP 10.0.0.1, te se predstavlja kao 10.0.0.21. Nakon što 10.0.0.1 primi zahtjev, on kaže da mu je MAC adresa C8:4C:75:00:00:00. Sada pošiljatelj zna na koju adresu treba poslati datagram, a za ostalo će se pobrinuti usmjeritelj na adresi 10.0.0.1.

Višeodredišne adrese služe za slanje podataka skupu čvorova u mreži.

ARP ne koristi višeodredišne adrese jer je dizajniran za jednoodredišne adrese.

Zadatak 7. (Ping/ping.imn)

-c count – Određuje broj paketa prilikom slanja pinga

-G sweepmaxsize – Određuje 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 u sekundama proći između dva pinga.

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

-M mask | time – postavlja ICMP masku na echo ili reply, 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 odlazećim paketima koristi src_addr kao adresu pošiljatelja zahtjeva

-s packetsize – Određuje veličinu paketa.

-t timeout – Određuje vrijeme tijekom kojeg će se ping izvršavati prije nego što se automatski ugasi

-W waittime – vrijeme čekanja odgovora u milisekundama. Zakašnjeli odgovori se ne ispisuju, ali se kasnije smatraju dobivenima

-z tos – Određuje koji će se tip usluge koristiti

Zadatak 8. (Ping/ping.imn)

Utvrdite i objasnite što se događa pri slanju paketa alatom ping koji u polju TTL imaju vrijednost 3, a odredišno računalo je neko računalo udaljeno više od 3 „skoka“.

Ako je TTL = 3, a odredišno računalo je udaljeno više od 3 čvora, prije dolaskan odredišno računalo usmjeritelj shvaća da je TTL = 0, a kako nije on poslao zahtjev, ICMP će generirati odgovor originalnom podnositelju zahtjeva kako nije uspostavljen kontakt i kako je paket odbačen.

Zadatak 9. (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 je ping paket velik 10000 okteta, znamo da ćemo trebati dodati još 8 okteta za ICMP. Stoga, prava veličina paketa bit će 10008. Kako je MTU 1500, paket će se fragmentirati na 7 dijela. Maksimalna vrijednost veličine paketa prilikom izvršavanja naredbe ping je 65507, a time će veličina paketa biti 65515. To je $(2^{16} - 1) - 20$ okteta, tj. tih 20 okteta su IPv4 zaglavlje, a $(2^{16} - 1)$ je maksimalna veličina datagrama u protokolu IPv4.

Zadatak 10. (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 vremena 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 vrijeme se približno linearno povećava. Bez izravne veze imamo neke vrijednosti. Uz izravnu vezu, svaka od tih vrijednosti u prosječnom slučaju se znatno smanjuje. Možemo zaključiti da izrazvnom vezom između usmjeritelja *router0* i *router7* dobivamo kraći put do odredišta.

Zadatak 11. (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*).

Povećanje propagacijskog kašnjenja će uvelike povećati ping time. Za propagacijsko kašnjenje od samo 10 mikrosekundi, sve vrijednosti vremena se povećaju za oko 8 puta.

Zadatak 12. (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.

Prilikom prvog slanja paketa ping, koristi se protokol ARP koji će reći pc1 koja je MAC adresa pc2. Nakon što pc1 sazna adresu pc2, pc1 šalje ICMP poruku echo request pc2, koju pc2 zaprima, te šalje poruku echo reply. Svaka ICMP poruka koristi Ethernet sučelje a u datagramu je IPv4 protokol. ARP također koristi Ethernet sučelje. IPv4, ARP i ICMP pripadaju internet sloju, dok Ethernet pripada fizičkoj mreži.

Zadatak 13. (Ping/ping.imn)

Utvrdite što se sve mijenja u okviru protokola Ethernet kad se koristi naredba *ping* s različitim veličinama paketa koji se šalju.

Mijenjaju se stavke Frame Length (duljina okvira) i Capture Length (duljina ulovljenog). Trivijalno, mijenjaju se i vremenske vrijednosti, ali to vrijedi za svaki različiti paket.

Zadatak 14. (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.

Kada pingamo 127.0.0.1., na ethernetskom sučelju nema prometa. Ping u ovom slučaju šalje poruke lokalnom sučelju, tj. loopbacku.

Zadatak 15. (Ping/ping.imn)

Utvrdite kolike su minimalna i maksi9malna vrijednost MTU-a (*Maximum Transfer Unit*) na ethernetskom 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*.

Teoretski minimum i maksimum MTU su 46 i 1500. Međutim, ako pokušamo podesiti MTU u FreeBSD-u, minimalni MTU je 72 dok je maksimalni MTU 9018.

Zadatak 16. (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*).

Prvo, moguće je da se ruta promijeni. Ovo je vidljivo u Ispisu 3.1. u skripti, gdje TTL u jednom od pingova nije jednak kao u drugima. Analogno, ponekad će biti moguće dobiti kraću rutu. Kao drugo, moguće je ne naći rutu ako jedan od usmjeritelja blokira određene protokole. Ovo se riješava uporabom drugog protokola za slanje sondi. Također, moguće je da nešto sluša na defaultom portu, pa će prilikom slanja paketa biti poslana poruka ICMP PORT UNREACHABLE umjesto ICMP poruke da je TTL paketa istekao. Također, različiti tipovi usluge mogu promijeniti rutu paketa. Ovo nije nužno neispravan rezultat.

Zadatak 17. (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 temlji rad alata. Kako se koristi TTL-polje i protokol ICMP?

U prvo koraku šalje se ARP zahtjev za 10.0.0.1. Nakon što prvi usmjeritelj dojavi pc1 koja mu je MAC adresa, kreće UDP zahtjev koji šalje paket s TTL = 1 odredišnoj adresi. Paket umire kod 10.0.0.1, koji šalje ICMP poruku pošiljatelju zahtjeva. Ovo se poslalo na 3 različita porta, pa se stoga istovjetni paket šalje 3 puta, i svaki put mu 10.0.0.1 pošalje ICMP poruku da je paket umro. Zatim pc1 šalje paket sa TTL=2 prema odredišnoj adresi. Ovaj zahtjev se također šalje na 3 različita porta, tj. tri puta. Svaki korak se TTL povećava za jedan, dok ne dobijemo ICMP poruku od odredišta.

Zadatak 18. (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.

Zapisuje to u zaglavlje IP-datagrama. Neki od primjera su TCP i UDP, a postoje još i neki pomoćni protokoli koji se koriste rjeđe.

Zadatak 19. (Traceroute/traceroute.imn)

Utvrdite postoji li način da se iz primljenog IP-paketa očita put kojim je paket prošao kroz mrežu.

To nativno nije moguće jer IP-paket ne zapisuje rutu kojom je prošao. Alati kao što su npr. traceroute koriste se obilježjima mreže kojima mogu doznati najvjerojatniji put paketa, ali ni to nije sigurnost već jednostavno najvjerojatnija ruta. Ruta pojedinom paketa je, stoga, neodrediva čitanjem samog paketa.

Zadatak 20. (Ping/ping.imn)

Utvrdite postoji li način kojim protokol IP može ustanoviti da je poslani paket stvarno i primljen na odredištu.

Postoji, to se npr. postiže ICMP parom poruka echo request i echo reply.

Zadatak 21. (Ping/ping.imn)

Utvrdite utječe li fragmentacija na propusnost i kašnjenje te komentirajte dobivene rezultate.

Da – fragmentacija smanjuje propusnost i povećava kašnjenje.