Protokol IP in DHCP

Komunikacije v avtomatiki – Laboratorijske Vaje

11. november 2013

Povzetek

V tej nalogi se boste ukvarjali za analizo protokola IP in njegovega podpornega protokola DHCP. Naloge so razdeljene v več segmentov, vsak segment vsebuje nekaj vprašanj. Na ta vprašanja odgovorite, in jih zapišite v dokument (npr., uporabite MS Word). Da boste lažje argumentirali svoje odgovore, naredite izrise Wiresharka (angl., print screen), iz katerih so razvidni podatki, ki ste jih uporabili za odgovor, in jih priložite v dokument. Nekateri deli te vaje črpajo iz knjige *Computer Networks – A top down approach* [1], ki je na voljo tudi v fakultetni knjižnici, ter prosojnic s predavanj prof. Stanislava Kovačiča. Snov za te vaje najdete v knjigi [1] v poglavjih 1.4.3 ter 3.4. Ker se določeni deli vaje dotikajo podrobnosti delovanja protokola, spodbujamo uporabo literature na spletu za iskanje odgovorov. S predavanj veste, da pakete IP imenujemo *datagrami*. V nalogi bomo uporabljali to izrazoslovje.

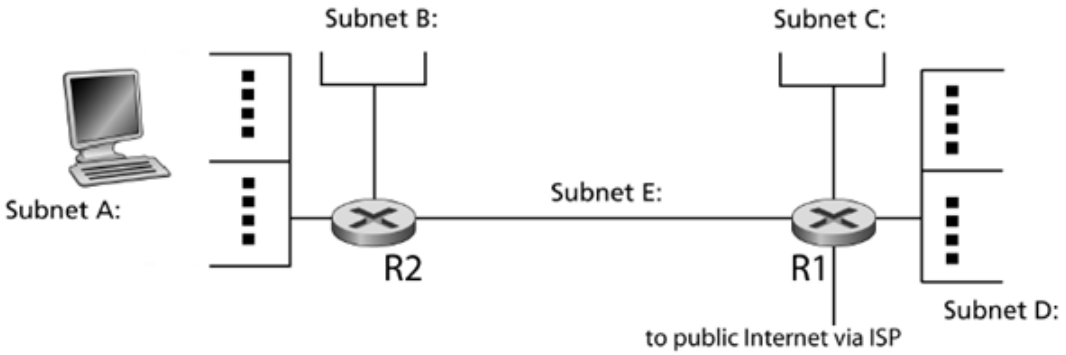
1 Številke IP, podmrežja in mrežne maske

Najprej rešite nalogo z določevanjem mrežnih številk in nastavljanjem mrežnih mask. Slika 1 prikazuje sistem podomrežji. Vsaka podmreža A-D vsebuje največ 28 vozlišč, podmreža E povezuje usmerjevalnika R_1 in R_2. Določite CIDR mrežne maske petim podmrežam (tipa x.y.z.w/a). Katero CIDR predpono usmerjevalnik R_1 oglašuje v javno omrežje?

2 Traceroute in zajem podatkov

V tej laboratorijski vaji boste morali generirati različne IP datagrame in jih poslati preko mrežnega vmesnika. V ta namen bomo demonstrirali uporabo programa Traceroute[1], ki omogoča pošiljanje IP datagramov specifični napravi X. Traceroute deluje tako, da najprej odda enega ali več IP datagramov, katerim nastavi značko TTL (time-to-live) na 1; nato pošlje več datagramov z značko TTL=2, TTL=3 in tako naprej. Ti datagrami so pravzaprav *ping* paketki, torej, paketki tipa "*ICMP echo request*", ki naslovniku sporočajo, naj se odzove tako, da nazaj pošlje

[1]Glejte, na primer, `http://en.wikipedia.org/wiki/Traceroute`



Slika 1: Primer sistema podomrežji. Slika je povzeta po [1].

isti paket. Ena od lastnosti usmerjevalnikov je ta, da vsakemu paketu, ki ga posredujejo, najprej znižajo vrednost TTL za ena (po RFC 791 mora usmerjevalnik znižati TTL za *vsaj* ena), in v primeru, da postane TTL=0, paket zavržejo. Zavrnitev paketa nato sporočijo izvornemu naslovu tako, da mu pošljejo paket ICMP tipa 11 (*ttl exceeded*). S tem preprečimo, da bi paketki plavali po mreži v nedogled. Kot rezultat bo usmerjevalnik, ki je za en hop oddaljen od pošiljatelja, prejel prvi paket s TTL=1, ga zavrnil, in poslal pošiljatelju ICMP sporočilo "ttl exceeded". Enako bo datagram s TTL=2 povzročil na za dva hopa oddaljenem usmerjevalniku nastanek sporočila ICMP in tako naprej. S takim postopkom lahko pošiljatelj, ki uporablja Traceroute, izve identitete vseh usmerjevalnikov na poti do naslovnika preprosto tako, da pogleda izvorno številko prispelih datagramov ICMP s "ttl exceeded". V nadaljevanju bomo uporabili program Traceroute za pošiljanje datagramov različnih velikosti.

Program Traceroute zaženete iz ukazne vrstice, oziroma iz terminala. Format klica je naslednji

```
traceroute -I -N 1 -z 1 -q 3 [naslov_IP_naprave] [dolžina_paketov]
```

Namesto IP naslova (npr. 8.8.8.8) lahko podate tudi ime IP naprave oziroma strežnika (npr. www.google.com). Parameter `[dolžina_paketov]` določa, kako velike pakete bo Traceroute pošiljal iz vašega računalnika, in je izražen v bajtih. Privzeta vrednost je odvisna od omrežnega vmesnika in parametrov povezave, s katero je vaš računalnik povezan v IP omrežje. Bodite zelo pozorni kaj ta številka pomeni. Dolžina pomeni dolžino *celega IP datagrama*: dolžino IP glave + dolžino podatkov. Podatki so v našem primeru cel paket ICMP!

Ostali parametri programa Traceroute v zgornjem primeru imajo naslednji pomen: `-I` pomeni, da bo Traceroute pošiljal ICMP paketke (možnosti sta še `-U` za UDP in `-T` za TCP), `-N 1` določa, naj pošilja paketke zaporedoma (brez tega parametra bo pošiljal veliko količino paketkov vzporedno, delovanje bo hitrejše, a lahko se zgodi, da vas kakšno vozlišče na poti do cilja začne ignorirati, in vam ne vrne ICMP paketka *ttl exceeded*, in vaši zajem ne bi bil popoln. To se sicer še vedno lahko zgodi, a če omrežja ne zasuvamo s paketki prehitro, je za to precej manj možnosti. Parameter `-z 1` določa, da med pošiljanjem paketkov čaka eno sekundo, razlogi pa so enaki kot za parameter `-N 1`. Parameter `-q 3` pa določi, da vsako zahtevo ponovi trikrat, saj se lahko ali vaši paketki ali pa odgovori nanje po poti preprosto izgubijo.

2.1 Preizkusite program *Traceroute*

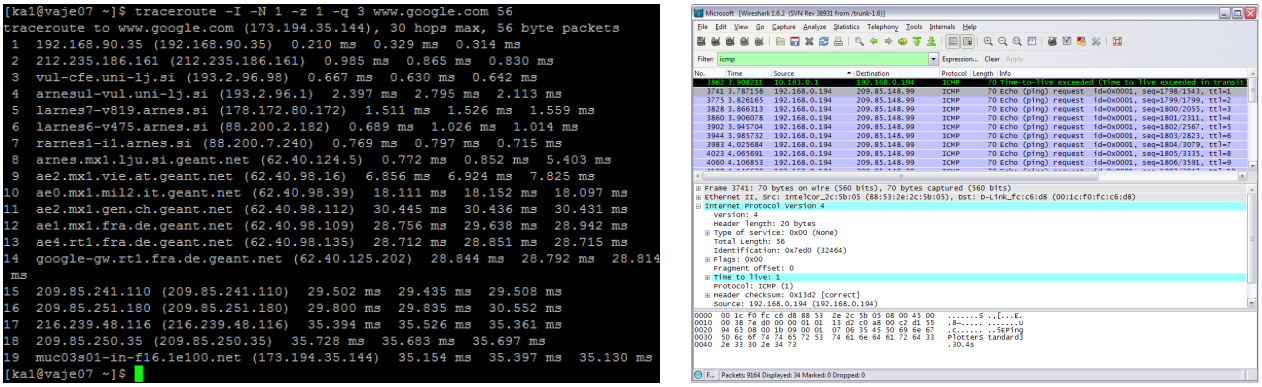
Izberite velikost datagrama 56 bajtov, in preizkusite s programom Traceroute naciljati naslov `www.google.com`:

```
traceroute -I -N 1 -z 1 -q 3 www.gooogle.com 56
```

Če ste vse izvedli pravilno, bi morali videti sliko podobno Sliki 2. Program vam izpiše vsa imena usmerjevalnikov na poti do `www.google.com`, oziroma, njihove številke IP, desno od njih pa čase odzivov ustreznega usmerjevalnika skupaj s standardno deviacijo meritve. Včasih lahko pri teh meritvah naletite na navidez čudne pojave: čas potovanja paketa do prvega usmerjevalnika je lahko daljši od potovanja nekega drugega paketa do na primer osmega usmerjevalnika. Razlog je v tem, da smo meritev za prvi usmerjevalnik dobili preko prvega poslanega paketa (s TTL=1), meritev za osmi pa preko kasnejšega paketa (s TTL=8). Medtem se je lahko promet v mreži spremenil, spremenile so se zakasnitve in v tem primeru so se zakasnitve znižale, med izvajanjem meritev za osmi usmerjevalnik. Rezultat je v tem, da je prvi usmerjevalnik porabil več časa za prejem prvega paketa in njegovo procesiranje, kot celotna pot kasnejšega paketa do osmega usmerjevalnika (skupaj s časom za procesiranje na osmem usmerjevalniku).

2.2 Priprava eksperimenta

Sedaj, ko poznate Traceroute, in razumete njegovo delovanje, se lahko lotimo eksperimenta z Wiresharkom. V eksperimentu bomo pošiljali različno velike datagrame in jih spremljali z Wiresharkom. Nekateri datagrami bodo večji od največje možne dolžine paketa IP, kar nam



(a) (b)

Slika 2: Primer izpisa programa Traceroute (a) in primer Wiresharka z urejenimi paketi ICMP (b).

bo omogočalo analizo fragmentacije v protokolu IP. Zaženite Wireshark in aktivirajte zajem paketov preko aktivnega mrežnega vmesnika. Medtem, ko Wireshark zajema promet, ponovite eksperiment s Traceroute s ciljnim naslovom `www.google.com`. Ko Traceroute konča, ponovite eksperiment še z 2000 bajti velikimi IP datagrami in nato s 3500 bajti velikimi datagrami. Ko se Traceroute še zadnjič konča, ustavite zajem podatkov v Wiresharku. Wiresharkov izpis paketov spravite na disk, da boste lahko z nalogo nadaljevali tudi doma.

2.3 Analiza zajetih podatkov

V Wiresharkovem izpisu bi zdaj morali videti množico sporočil tipa *ICMP echo request* (če bi Traceroute zagnali s parametrom `-U` ali pa brez parametra `-I`, bi videli UDP segmente), ki jih je poslal vaš računalnik, ter prejete pakete *ICMP ttl exceeded*. V Wiresharkov filter vpišite `icmp` za boljši pregled.

1. Postavite se na prvi ICMP echo request, ki ga je poslal vaš računalnik in v srednjem oknu Wiresharka razširite zavihek za Internet Protocol. Kakšen je naslov IP vašega računalnika in pod katero značko ste ga odčitali?
2. V glavi IP preberite značko, ki vam pove, kateri protokol višjega sloja nosi IP datagram. Kateri protokol ustreza vrednosti te značke?
3. Koliko bajtov je dolga glava IP datagrama? Koliko bajtov zavzamejo *podatki* (angl. payload) v IP datagramu?
4. Kaj predstavlja značka `Total length`? Če za vaš paket poznate vrednost `Total length`, kako bi iz tega izračunali koliko bajtov so dolgi podatki v IP datagramu?
5. Z analizo značk v `Flags` preverite ali je bil ta paket fragmentiran. Razložite pomen treh značk, ki jih najdete v `Flags` (lahko si pomagate, na primer z Wikipedijo ali s stranjo `http://www.cs.rpi.edu/academics/courses/spring06/netprog/c04.html`).

Sedaj uredite vse pakete v Wiresharku po izvornem naslovu IP. To storite tako, da z miško kliknete na ime `source` v *zgornjem* oknu Wiresharka. Če v polju puščica kaže navzgor, kliknite še enkrat, da bo kazala navzdol. Če ste vse storili prav, potem vaš Wireshark izgleda podobno kot v Sliki 2b. Postavite se na prvi *ICMP echo request* paket, ki ste ga poslali iz vašega računalnika na `www.google.com`.

6. Za naslednjih pet paketov si izpišite polja: izvorno številko IP, ponorno številko IP, Identification, Total length in Time to live. Razložite njihov pomen in njihove vrednosti.

2.4 Fragmentacija v IP

Ker se bomo v tej podnalogi ukvarjali z analizo fragmentacije v IP, naprej v Wiresharku spremenimo nastavitve tako, da bomo prikazovali tudi fragmentirane pakete IP. Najprej izbrišite filter `icmp` v Wiresharku s pritiskom na gumb *clear* poleg polja s filtri. Nato v *Edit→Preferences→Protocols→IPv4*

izklopite vse kljukice razen pri *Support packet-capture from IP TSO-enabled hardware*, in pritisnite *OK*. Izklopite prikaz protokola ICMP tako, da v *Analyze→Enabled protocols* izklopite kljukico ob *ICMP* in kliknite *OK*. V polje s filtri vpišite `ip`. Sedaj vidite IPv4 datagrame, ki jih je poslal Tracerizte, med njimi pa IP datagrame, ki so jih pošiljali ter prejemali drugi procesi z vašega računalnika.

1. Premaknite se navzdol po spisku paketov in poiščite prvega od paketov, ki jih je Traceroute poslal in so bili fragmentirani. Paket boste prepoznali kot zaporedje dveh IPv4 paketov. Za ta dva paketa izpišite vrednosti naslednjih značk: izvorni naslov IP, ponorni naslov IP, Header Length, Total length, Identification, Flags, in dolžino podatkov v paketu. Razložite vrednost značke Identification pri obeh datagramih.
2. Veste, da ste Traceroute nastavili vrednost IP datagrama na 2000. Ali je ta vrednost enaka vsoti bajtov v podatkih zgornjih fragmentiranih paketov? Razložite svoj odgovor.
3. Sedaj poiščite fragmentirane IP datagrame, ki ustrezajo paketu *ICMP echo request* velikosti 3500, ki ste ga poslali s Traceroute. Koliko fragmentov je nastalo iz tega datagrama?
4. Recimo, da nam Ethernet omejuje velikost IP datagrama na 1500 bajtov. Naš sistem tvori IP datagram, ki je velik 4000 bajtov, z izvorno številko IP1 in ponorno številko IP2 . Koliko fragmentiranih IP datagramov bo tvorila fragmentacija? Za vsakega napišite njegovo dolžino, izvorno in ponorno IP številko, značko `Identification`, značko `Flags` in značko `Total length`.

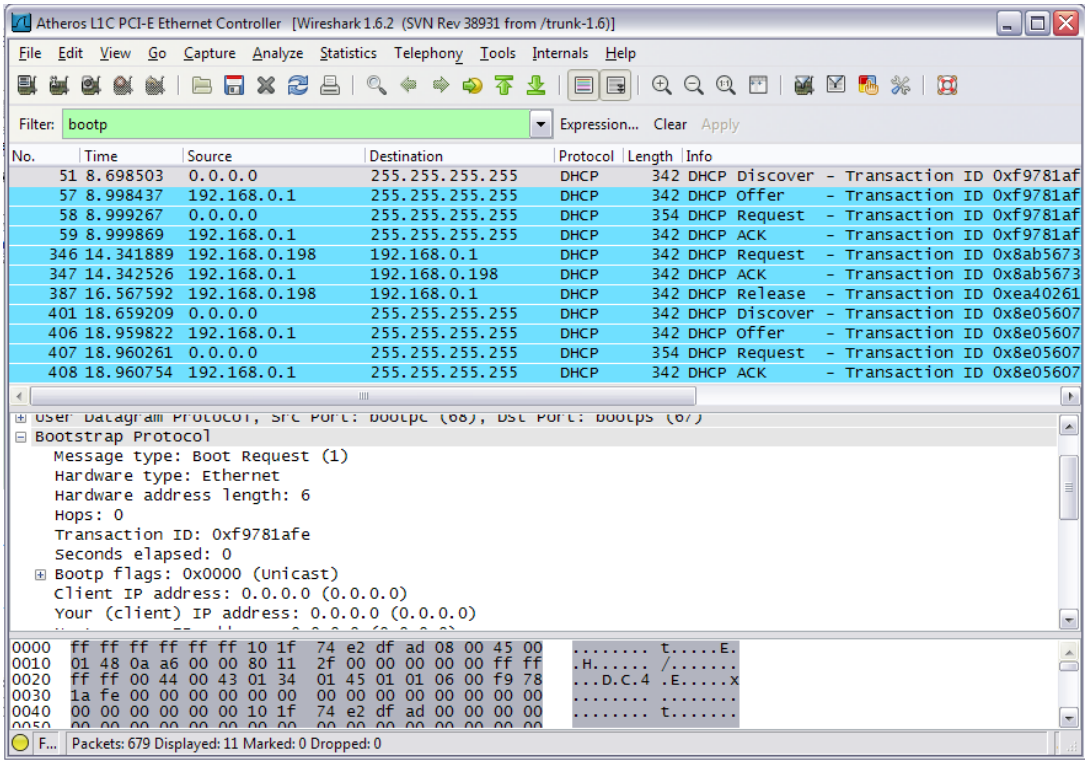
3 IP in protokol DHCP

V tem delu naloge si boste na kratko pogledali delovanje protokola DHCP. O protokolu DHCP si lahko preberete na prosojnicah prof. Stanislava Kovačiča ali pa v knjigi [1] v poglavju 4.4.2. Spomnite se, da je naloga protokola DHCP dinamično dodeljevanje številke IP napravam, ki se vklapljajo v mrežo, kakor tudi konfiguracija nekaterih drugih parametrov mreže.

Pomembno. Protokol DHCP je najlažje testirati tako, da dejansko spreminja IP naslov vašega računalnika. Pod operacijskim sistemom Windows lahko to vajo izvedete s pomočjo ukaza `ipconfig`, in sicer tako da mu podate parameter `/renew` ali `/release` (torej `ipconfig /renew` in `ipconfig /release`). To lahko poskusite doma, vendar pazite, da na koncu ne boste ostali brez omrežne povezljivosti – po koncu takšnega eksperimenta boste morali še enkrat nastaviti parametre omrežne povezave.

Zato, da računalnikov na vajah ne bi pustili brez omrežne povezljivosti, bomo uporabili testno verzijo DHCP odjemalca, programček `dhtest`. Ta program izvede enake operacije, kot bi jih izvedel vaš računalnik, če bi želel od DHCP strežnika dobiti IP naslov, vendar pa ostane dejanski IP naslov računalnika nespremenjen, namesto tega pa se rezultati komunikacije z DHCP strežnikom shranijo v tekstno datoteko, ki jo lahko pregledate. Vajo bomo zato zastavili na naslednji način:

- Najprej odprite terminal, da pridete do komandne vrstice – če ga že nimate odprtega iz prejšnje vaje.
- Poženite ukaz `ifconfig` in v izpisu najdite ime vašega omrežnega vmesnika (mrežne kartice) in njegov Ethernet (MAC) naslov.
- Zaženite zajemanje podatkov z Wiresharkom.
- Sedaj se vrnite v terminal in izvedite naslednje zaporedje ukazov:
  1. `dhtest -V -i [omrežni_vmesnik] -m [Ethernet naslov]`. S tem bo vaš računalnik poiskal DHCP strežnik, in od njega dobil IP naslov ter nekaj ostalih podatkov, ter jih zapisal v datoteko, katere ime je enako Ethernet naslovu vaše kartice.
  2. `dhtest -V -i [omrežni_vmesnik] -m [Ethernet naslov] --release`. S tem bo vaš računalnik strežniku sporočil, da IP naslova ne potrebuje več, datoteka s podatki o IP nastavitvah pa bo izginila.
- Ko se zadnji ukaz v konzoli izvrši, ustavite zajem paketov v Wiresharku. Izpis paketov iz Wiresharka shranite na disk, da boste lahko z nalogo nadaljevali tudi doma.



Slika 3: Primer Wiresharka z urejenimi paketi protokola DHCP.

Postavite se v Wireshark in njegov filter nastavite na `bootp` (protokol DHCP je izpeljanka njegovega predhodnika BOOTP). Če ste postopek izvedli pravilno, bi moral vaš Wireshark izgledati podobno kot Slika 3. Iz te slike vidimo, da je prvi ukaz `dhtest` povzročil prve štiri paketke: DHCP discover, DHCP offer, DHCP request in DHCP ACK. Odgovorite na spodnja vprašanja.

1. S pregledom značk v paketih preverite ali so sporočila DHCP poslana preko UDP datagramov ali preko TCP segmentov.
2. Skicirajte časovni potek pošiljanja paketkov med vašim računalnikom in strežnikom DHCP za prve štiri izmenjane paketke. Za vsak paket si izpišite izvorno in ponorno številko porta. Ali so te številke tudi v splošnem take za katerikoli DHCP klient in strežnik?
3. Spomnite se na Ethernet naslov vaše vaše mrežne kartice. Kje ga najdete v zajetih podatkih?
4. Izpišite vrednosti *Transaction-ID* za vsako od štirih DHCP sporočil (Discover/Offer/Request/ACK DHCP). Kaj je namen številke *Transaction-ID*?
5. Klient uporablja DHCP zato, da med drugim pridobi številko IP. Vendar je številka IP potrjena šele na koncu izmenjave štirih sporočil. Torej, če klient dobi svojo številko IP šele na koncu izmenjave, pred izmenjavo pa niti ne pozna številko IP DHCP strežnika, katere številke potem uporablja med procesom komunikacije s strežnikom? Za vsako od štirih sporočil (Discover/Offer/Request/ACK DHCP) napišite izvorno in ponorno številko IP.
6. Katera številka IP pripada vašemu DHCP strežniku?
7. Katero številko IP vam strežnik nudi v paketu *DHCP offer*? Priložite izpis iz Wiresharka, ki prikazuje te podatke.
8. Razložite namen značk *router* in *subnet mask* v paketu *DHCP offer*.
9. V primeru v Sliki 3 klient zahteva številko IP, ki mi jo je ponudil DHCP strežnik. Kaj se dogaja v vašem primeru?
10. Razložite namen najemnjega časa (angl., lease time). Kako dolg je ta čas v vašem experimentu?

11. Kaj je namen sporočila *DHCP release*? Ali DHCP strežnik pošlje potrdilo na klientovo sporočilo *DHCP release*? Kaj bi se zgodilo, če bi se sporočilo *DHCP release* izgubilo na poti in ne bi nikoli prispelo do DHCP strežnika?
12. Podatke, ki ste jih pridobili s pomočjo Wiresharka, primerjajte s tistimi, ki jih je izpisal programček

Literatura

[1] J.F. Kurose and K.W Ross, *Computer networking – a top-down approach*, Addison Wesley, 2009.