MREŽNI SLOJ INTERNETA

Stjepan Groš

07. 09. 2006.

Sadržaj

1. Uvod

Internet je vrlo popularna mreža čija upotreba za zabavu ili posao predstavlja svakodnevicu mnoštva ljudi. To, zajedno s vrlo brzim rastom, čini ju vrlo bitnom komponentom ljudskog društva i njega razvoja te predstavlja dobar motiv za njeno proučavanje i razumijevanje.

Temeljom Interneta može se smatrati mrežni sloj a posebice IP protokol, na osnovu kojega se zatim gradi niz ostalih protokola. Slika 1 prikazuje arhitekturu TCP/IP porodice protokola, odnosno, Interneta. Na toj slici označen je sloj za koji se može reći da predstavlja početak svih Internet protokola – Mrežni sloj.

Slika 1. Arhitektura TCP/IP porodice protokola

Ovaj dokument detaljno se bavi mrežnim slojem Interneta i pojašnjava način na koji rade protokoli u tom sloju. U tom smislu dan je relativno detaljan opis protokola, a zatim je izrađen i niz pojednostavljenih primjera koji pokušavaju ilustrirali principe svakog protokola.

Primarni izvor informacija za ovaj dokument čine RFC dokumenti koje su normativne za sve protokole Interneta. U slučaju da postoji neslaganje između opisa protokola danog u ovom tekstu i odgovarajuće specifikacije u RFC dokumentu, tada treba uzeti RFC dokument kao ispravan.

1.1. Izrada osnovne IP mreže

Prije detaljnog opisa protokola koji čine Internet pogledati ćemo kako ostvariti jednostavnu mrežu od nekoliko računala, primjerice mrežu kakvu je moguće ostvariti doma. Recimo da je zadatak spojiti tri računala u mrežu na kojoj je moguće razmjenjivati dokumente i dijeliti pisač. Kako bi se omogućila zadana funkcionalnost potrebno je podesiti parametre koji su vezani uz Internet. Na tom primjeru uvesti će se osnovni pojmovi koji se kroz daljnji tekst dodatno produbljuju.

Ostvarivanje zadane mreže započeti ćemo fizičkim povezivanjem računala. To je moguće obaviti putem bakrenih vodiča ili bežičnom vezom. Ti detalji opisani su u poglavlju koje se bavi fizičkim i podatkovnim slojem, odnosno Ethernet lokalnom mrežom. Dakle, nakon fizičkog povezivanja recimo da ćemo imati sljedeću situaciju:

SLIKA TRI RAČUNALA S PREKLOPNIKOM U SREDINI!

Kako bi računala mogla komunicirati na mrežnom sloju potrebno im je dati adrese. Naime, kada računala međusobno razmjenjuju podatke tada na sve podatke dodaju adresu kome su namijenjeni. Na Internetu adresa je 32 bitni broj koji se radi lakšeg manipuliranja i pamćenja piše u posebnom obliku. Primjerice, adresa 0xC0A8000A piše se u sljedećem obliku:

C0	A8	00	0A
192	168	0	10

tj.

192.168.0.10

Bez dodatnog ulaženja u način raspodjele i dodjele IP adresa pretpostavimo da su za naša tri računala određene sljedeće adrese:

Računalo A 192.168.0.10
Računalo B 192.168.0.11
Računalo C 192.168.0.12

Primjetimo kako preklopniku koji povezuje računala međusobno nismo dodijelili IP adresu. Naime, računala neće slati podatke preklopniku, niti preklopnik generira ikakve podatke koje šalje računalima. Dakle, preklopnik je potpuno transparentan za računala, ili preciznije rečeno za mrežni sloj, i ona nisu svjesna njegova postojanja.

Prilikom upisivanja IP adrese svaki operacijski sustav zatražiti će i *mrežnu masku*! Mrežna maska dijeli adresu na dva dijela: mežni i računalni. Najčešća mrežna maska je 255.255.255.0 i u toj varijanti mrežni dio sastoji se od prva tri okteta, a računalni od zadnjeg okteta.

Nakon što su računalima dodjeljene adrese i definirana mrežna maska ona će moći komunicirati međusobno. Međutim, ta komunikacija je dosta ograničena budući da nije definiran niz usluga koje čine Internet tako popularnom mrežom niti je definiran način na koji se pristupa Internetu.

PATH MTU discovery RFC1191, RFC1435, RFC1981, RFC2923, RFC1435

2. Adresiranje na internetu

2.1. IPv4 adrese

Kako bi uređaji na Internetu bili dostupni moraju imati adresu na koju će im se upućivati podaci. Adresa na Internetu je 32-bitni broj. Radi lakšeg pamćenja i manipuliranja tim adresama one se pišu u sljedećem obliku:

```
aaa.bbb.ccc.ddd
```

`aaa`, `bbb`, `ccc` i `ddd` su decimalne vrijednosti pojedinog okteta 32-bitne riječi pri čemu je oktet `aaa` najveće težine, a `ddd` oktet najmanje težine. Očito su vrijednosti pojedinih elemenata adrese u rasponu od 0 do 255. Primjerice, adresu računala koje ima heksadecimalni oblik:

```
0xA135410B
```

pišemo u sljedećem obliku:

```
161.53.65.11
```

Prilikom dodjeljivanja adrese nekom računalu taj broj se ne može odabrati proizvoljno već postoje strogo određena pravila po kojima se on bira, a ta pravila diktira mreža na koju se računalo priključuje.

Uz sam broj, tj. internet adresu, vezan je još jedan parametar koji se zove *mrežna maska*. Radi lakšeg povezivanja Interneta svaka internet adresa sastoji se od *mrežnog* i *računalnog dijela*. Mrežni dio identificira neku određenu mrežu na Internetu, dok računalni dio određuje pojedino računalo u toj mreži. I opet kao i kod same internet adrese i za mrežnu masku postoje određena pravila po kojima se ona određuju te se ne može odabrati proizvoljno. Uzmimo primjerice adresu `161.53.65.11`, ona priprada Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva. Zbog te činjenice mrežna dio je predodređen i čini ga gornjih 24 bita, tj. sve od 31. bita do uključivo 8. bita, dok se računalni dio sastoji od donjih osam bitova, tj. od 7. do 0. bita. Ta činjenica može se pisati na dva ekvivalentna načina:

```
161.53.65.11/255.255.255.0
```

ili

```
161.53.65.11/24
```

Očito je da se u prvom primjeru mrežni dio označava pomoću broja čiji oblik je sličan samoj adresi pri čemu su u tom broju pojedine binarne znamenke postavljene na 1 ako pripadaju mrežnom dijelu, a na 0 ako pripadaju dijelu koje određuje pojedino računalo. U drugom slučaju, koji je daleko kompaktniji i češće se koristi, upotrebljava se zapis kod kojega se nakon kose crte piše broj bitova koji pripadaju mrežnom dijelu pri čemu se podrazumijeva da se ti bitovi nalaze na najvišim mjestima 32-bitne riječi. U navedenom primjeru dakle, radi se o lokalnoj mreži čija adresa je 161.53.65.0 dok računalo ima broj 11 unutar te mreže.

Veličina računalnog dijela u adresi određuje maksimalan broj računala koji je moguće priključiti na mrežu. Na mrežu čija je adresa 161.53.65.0/24 moguće je priključiti 254 računala. Naime, za računalni dio na raspolaganju je 8 bitova (od 7. bita do 0. bita) što daje ukupno $2^8 = 256$ različitih

kombinacija. Međutim, dvije kombinacije po mreži se ne mogu upotrijebiti za adrese računala ili drugih mrežnih uređaja. Prva kombinacija koja se ne može upotrijebiti sadrži sve nule u računalnom dijelu. Ona se upotrebljava za označavanje mreže. Druga kombinacija koja se ne može upotrijebiti sadrži sve jedinice u računalnom dijelu. To je *difuzna adresa* (engl. broadcast) te će svaki paket koji se pošalje na tu adresu biti isporučen svim računalima koja se nalaze u navedenoj mreži.

Svaka mrežna kartica u računalu, a općenito i bilo koji mrežni uređaj uz pomoć kojega se računalo priključuje na internet, mora imati barem jednu adresu i odgovarajuću mrežnu masku no može ih imati i više, tj. jedna mrežna kartica može imati više IP adresa. U tom slučaju govori se o *alias* adresama.

Konfiguracija uređaja (mrežne kartice) obavlja se ili automatski ili ručno, što ovisi o postavkama mreže na koju se priključuje.

Postoji određeni niz adresa koje imaju posebno značenje [RFC3330]. Rezervirane adrese koje se češće javljaju su:

1. Svako računalo ima poseban "uređaj" koji se zove *loopback* i koje ima posebnu adresu, 127.0.0.1/8. To je virtualni uređaj u smislu da fizički ne postoji, tj. operacijski sustav emulira njegovo ponašanje. Nadalje, mreža 127.0.0.0/8 je rezervirana i ne smije se pojaviti na Internetu!
2. Mrežne adrese 10.0.0.0/8, 172.16.0.0/12 i 192.168.0.0/16 su tzv. privatne adrese i nikada se ne smiju pojaviti na Internetu, no smiju se pojaviti na internetu ili na intranetu! Drugim riječima namijenjene su za lokalne mreže koje nikada neće biti priključene na Internet ili su na njega vezane indirektno.

Popis svih rezerviranih IP adresa zajedno s referentnim izvorom dan je u tablici 1 koja je preuzeta iz [RFC3330].

Tablica 1. Popis svih mreža posebne namjene [RFC3330]

Mreža	***Namjena***	***Referenca***
0.0.0.0/8	"This" Network	[RFC1700, page 4]
10.0.0.0/8	Za privatne mreže	[RFC1918]
14.0.0.0/8	Javne podatkovne mreže	[RFC1700, page 181]
24.0.0.0/8	Cable Television Networks	
39.0.0.0/8	Rezervirano s mogućnošću alokacije	[RFC1797]
127.0.0.0/8	Loopback	[RFC1700, page 5]
128.0.0.0/8	Rezervirano s mogućnošću alokacije	
169.254.0.0/16	Adrese vezane uz sučelja	
172.16.0.0/12	Za privatne mreže	[RFC1918]
191.255.0.0/16	Rezervirano s mogućnošću alokacije	
192.0.0.0/24	Rezervirano s mogućnošću alokacije	
192.0.2.0/24	Test-Net	
192.88.99.0/24	6to4 Relay Anycast	[RFC3068]
192.168.0.0/16	Za privatne mreže	[RFC1918]
198.18.0.0/15	Network Interconnect Device Benchmark Testing	[RFC2544]
223.255.255.0/24	Rezervirano s mogućnošću alokacije	
224.0.0.0/4	Difuzija u grupi	[RFC3171]
240.0.0.0/4	Rezervirano za buduću upotrebu	[RFC1700, page4]

Dodjeljivanje brojeva mrežama se obavlja preko odgovarajućih institucija koje su hijerarhijski organizirane. Tako npr. CARNet je dobio mrežnu adresu 161.53.0.0/16 koja je dodatno podijeljena od strane samog CARNet-a u blokove od po 256 adresa koje se dodjeljuju pojedinim akademskim institucijama i zavodima. Primjerice, ZEMRIS ima već spomenutu mrežnu adresu 161.53.65.0/24, dok Računski centar FER-a ima dodijeljenu adresu 161.53.73.0/24. Dodijeljivanje brojeva pojedinim računalima obavlja mrežni administrator unutar odgovarajuće administrativne jedinice.

Nekada su sve IP adrese bile klasificirane u grupe na osnovu prvih nekoliko bitova 32-bitne adrese. Svaka grupa imala je unaprijed određenu veličinu mrežnog i računalnog dijela. Tablica 2 prikazuje tu podjelu.

Tablica 2. Podjela IP adresa na klase mreža

Početni bitovi	***Klasa***	***Broj raspoloživih mreža***	***Broj mogućih računala po mreži***
0	A	126	16777214
10	B	16384	65532
110	C	2,097,152	254
1110	D	24	
1111	E	8	

Svaki puta kada je neka organizacija zatražila blok IP adresa alokacija je bila obavljana iz gore navedenih raspona. To je značilo da je bilo moguće dobiti isključivo kombinaciju mreža koje mogu sadržavati 2^{24}, 2^{16} ili 2^{8} mreža. Međutim takva podjela je izuzetno nefleksibilna. Najveći problem predstavljala je B klasa adresa koja je brzo nestajala, a za njom je bila najveća potreba. Osim tih problema javljala su se još dva problema, oba posljedica brzog rasta Interneta. Prvi problem manifestirao se je u usmjernicima, tj. u uređajima koji povezuju Internet u jednu cjelinu, gdje su zbog nefleksibilne podjele adresa eksponencijalno rasli zahtijevi za memorijom. Drugi problem bio je potencijalni nedostatak IP adresa.

Kako bi se riješio taj problem ukinute su klase te se dodjela blokova IP adresa ravna prema potrebama. Primjerice, ako se pretpostavlja da nekoj organizaciji neće nikada trebati više od 1000 adresa koje će sigurno iskoristiti, tada će ona dobiti mrežnu masku duljine 20 bita. Besklasno adresiranje također znači da poznavanjem IP adrese više nije moguće znati koji dio predstavlja mreža, a koji računalni dio. Zbog toga se mrežni dio mora eksplicitno navoditi u slučajevima kada to zahtijeva situacija.

RFC1338, RFC1519

2.2. IPv6

Adresa opisane u prethodnom potpoglavlju se označavaju i kao IPv4 adrese. U suštini, naziv IPv4 sastoji se od dva dijela. Prvi dio je skraćenica IP koja potiče od izraza *Internet Protocol*, a drugi dio označava verziju koja je u ovom slučaju 4 – slovo v označava riječ *Version*. Problem s tim vrstama adresa je njihov mali broj, odnosno neefikasna iskorištenost. Iako je teoretski moguće imati 2^{32} adresa (ili brojeva, što je ekvivalentno), dobar dio tih adresa je neupotrebljiv iz različitih razloga. Internet brzo raste, i na njega se osim stolnih računala sve više priključuju i drugi uređaji (ručna računala, kućanski uređaji, itd.) a i kompletna telekomunikacijska infrastruktura prelazi na Internet (VoIP telefoni, mobilni telefoni, itd). Očito je problem manjka adresa vrlo ozbiljan iako ne i jedini koji IPv4 ima. Zahvaljujući raznoraznim poboljšanjima i dodacima nestanak IP adresa odgođen je do daljnjega, no ipak rješenje postoji i zove se IPv6, tj. *Internet Protocol Version 6.* Sve više uređaja sadrži podršku za tu verziju protokola i na pojedinim dijelovima Interneta se polako prelazi na taj protokol.

Taj protokol i adrese vezanje uz njega donose određen broj promjena u odnosu na IPv4. Najbitnija promjena je da se adrese sastoje od 128 bita umjesto dosadašnjih 32. Mrežna maska koristi se na isti način kao i kod IPv4 adresa s tim da se zapisuje isključivo u skraćenom obliku, tj. obliku po kojemu se navodi samo broj bitova mrežnog dijela. Primjer IPv6 adrese je:

```
fe80:0000:0000:0000:02c0:dfff:fe22:4b85/10
```

Umjesto decimalnog zapisa koristi se heksadecimalni zapis pri čemu se svih 128 bita dijeli u grupe od po 16 bitova. Grupe se međusobno razdvajaju dvotočkom. S lijeve strane se nalaze teži bitovi. Zbog velikog broja nula uveden je i skraćeni način zapisivanja IPv6 adrese. Dva su pravila koja se pri tome koriste i koja je potrebno poštivati:

1. Vodeće nule se u pojedinoj grupi mogu izbaciti. Tada prethodna adresa iz primjera ima sljedeći oblik:

```
fe80:0:0:0:2c0:dfff:fe22:4b85/10
```

2. Nadalje, uzastopni niz grupa koje su u cijelosti nula može se ispustiti i na to mjesto se stavljaju dvije uzastopne dvotočke. To je moguće napraviti isključivo jednom u cijelom broju! Primjenom tog pravila prethodna IPv6 adresa se svodi na sljedeći oblik:

```
fe80::2c0:dfff:fe22:4b85/10
```

Iduća razlika u odnosu na IPv4 adrese odnosi se na kategorije. Naime, postoje tri kategorije:

- *Unicast addresses*. Ova vrsta adresa označava točno određeno sučelje na Internetu.
- *Anycast address*. Adrese ove kategorije su novost u odnosu na IPv4. Naime, adrese iz ove grupe označavaju skupinu sučelja, pri čemu se paket s tom adresom isporučuje "najbližem" sučelju. Najbliže sučelje odabire usmjernik na osnovu svojih tablica usmjeravanja.
- *Multicast address*. Adrese ove grupe označavaju grupu sučelja i sva sučelja koja pripadaju toj grupi primit će kopiju paketa.

Dakle, novost je Anycast grupa adresa, a također, nema više difuznih adresa. Pretpostavka je da će se difuzne adrese implementirati upotrebom adresa za difuziju u grupi.

IPv6 adrese podijeljene su u tipove, a određen broj adresa je kao i kod IPv4 rezerviran [RFC3513]. U tablici 3 navedeni su tipovi adresa.

Tablica 3. Tipovi IPv6 adresa

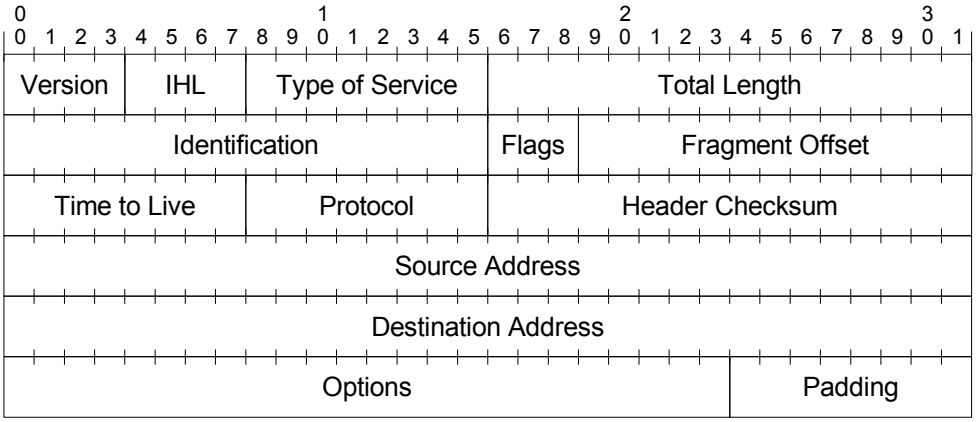
Tip adrese	***Binarni prefiks***	***IPv6 notacija***
Nespecificirana	00....00 (128 bita)	::/128
Loopback	00...01 (128 bita)	::1/128
Difuzija u grupi	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Site-local unicast	1111111011	FEC0::/10
Global unicast	Sve ostalo	

Anycast adrese biraju se iz unicast grupe adresa i međusobno se ne razlikuju.

3. IP paket

3.1. IPv4

Na slici 3 prikazana je struktura zaglavlja IP paketa verzije 4 [RFC0791]. Nakon zaglavlja dolaze podaci koje paket prenosi. Zaglavlje se sastoji od dva dijela, obaveznog i opcionalnog. Obavezni dio čine sva polja do opcija, dok se same opcije mogu ali i ne moraju pojaviti u IP paketu. Veličina zaglavlja IP paketa mora biti višekratnik 32 bita te u slučaju da kraj zadnje opcije ne pada na zadanu granicu koristi se ispuna (padding). Tijekom prijenosa bitovi se čitaju s lijeva na desno i odozgo prema dolje. Dakle, prvo ide 0. bit 0. riječi, zatim 1. bit 0. riječi i tako dalje do 31. bita nulte riječi. Nakon toga ide 0. bit prve riječi, 1. bit prve riječi itd. U pojedinim višebitnim poljima teži bit nalazi se s lijeve strane i ima manji indeks.



Slika 2. Struktura IPv4 paketa

IP paket započinje poljem *Version* koje sadrži oznaku verzije IP paketa. To polje uvedeno je s namjerom da se na Internetu mogu koristiti IP paketi različitih verzija. Dugo vremena na Internetu je korištena samo verzija 4 IP paketa. U novije vrijeme razvijena je i verzija 6 koja je opisana kasnije u ovom poglavlju.

Polje *IHL* (engl. Internet Header Length) sadrži veličinu zaglavlja IP paketa mjereno u 32-bitnim riječima. Kako obavezni dio zaglavlja IP paketa ima 5 32-bitnih riječi to je minimalna vrijednost ovog polja 5. Maksimalna vrijednost polja, zbog njegove veličine od 4 bita, je 15. To znači da je maksimalna veličina zaglavlja IP paketa, uključivši i opcije, 60 okteta.

Uz pomoć polje *Type of Service* definira se željeni tretman paketa. Naime, Internet je mreža *najbolje namjere* (engl. best effort) što znači da će pokušati isporučiti paket na odredište što je ranije moguće. Međutim mreža ne daje nikakve garancije, čak ni garancije da će paket stići, ili ako stigne da će biti ispravan. Više o ovom polju biti će riječi u poglavlju koje se bavi kvalitetom usluge na Internetu.

U polju *Total Length* nalazi se upisana veličina IP paketa zajedno s zaglavljem mjereno u oktetima. Kako je to polje veličine 16 bita to znači da je maksimalna veličina IP paketa 65536 okteta.

Polje *identification* postavlja usmjernik koji dijeli IP paket na fragmente. Svaki fragment u ovo polje ima upisanu identičnu vrijednost. Kod paketa koji nisu fragmentirani to polje nama funkciju.

Zastavice u IP paketu (polje *Flags*) koriste se za rukovanje fragmentima. Prva zastavica (bit 16) se ne koristi i mora biti postavljena na nulu. Druga zastavica (bit 17) onemogućava fragmentaciju paketa u slučaju da je postavljena na 1 (DF – Don't Fragment). Ako paket naiđe na usmjernik koji ne može poslati cijeli paket zbog njegove prevelike veličine on će biti odbačen, a usmjernik će generirati odgovarajuću ICMP poruku. Konačno, treća zastavica (bit 18) se koristi kod IP paketa koji su

podijeljeni na fragmente. Svi fragmenti, osim zadnjega, tu zastavicu imaju postavljenu na 1, dok je kod zadnjeg fragmenta postavljena na nulu.

Fragment Offset

Internet vrlo dinamočna mreža i lako se može desiti da određen broj paketa počne kružiti po Internetu. Bez posebnih mjera ne bi bilo načina da se takvi paketi otkriju i uklone, a bez uklanjanja bi u jednom trenutku preplavili mrežu te bi ona zbog toga postala neupotrebljiva. Da se spriječi takav scenario uvedeno je polje *Time to Live* uz pomoć kojega se upravlja zastarijelim paketima. Naziv polja potiče od prvotne ideje da se vrijednost tog polja umanjuje svake sekunde. Međutim, ispostavilo se da je to teško izvesti te da je puno pogodnije umanjiti vrijednost prilikom prolaska kroz usmjernike. Svaki usmjernik kroz koji paket prođe umanjuje vrijednost za 1 i u trenutku kada ona dosegne nulu usmjernik odbacuje paket. Na taj način sprečava se nekontrolirano kruženje paketa po mreži jer će u jednom trenutku biti odbačen.

Protokol koji IP paket prenosi u podatkovnom dijelu naznačen je u polju *Protocol*.

Header Checksum

Adresa na koju je IP pakete upućen nalazi se upisana u polju *Source Address*, dok se izvorište s kojega dolazi paket nalazi upisano u polju *Destination Address*.

Options

3.1.1. Zapisi IP paketa

Prilikom pisanja IP paketa u daljnjem tekstu i svim primjerima koristit će se sljedeći zapis:

```
IP (odredišna adresa, izvorišna adresa, TTL, koristan teret)
```

U određenim situacijama nisu bitni svi podaci, primjerice tijekom jednostavne analize proslijeđivanja paketa kroz neki usmjernik u većini slučajeva bitne su samo odredišna i izvorišna adresa. U tom slučaju koristi se skraćeni zapis:

```
IP (odredišna adresa, izvorišna adresa)
```

3.2. IPv6 paketi

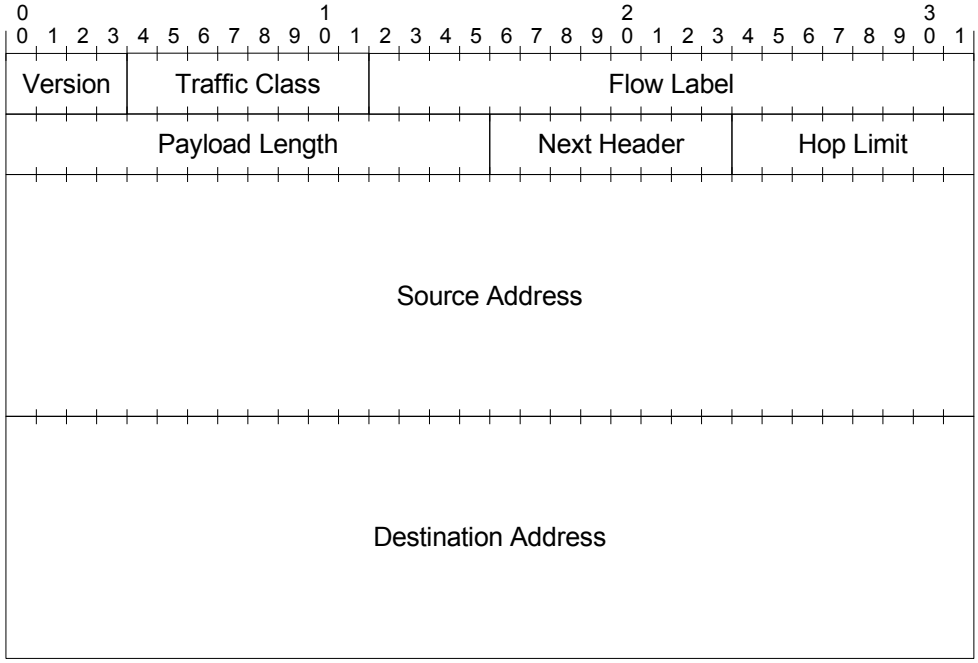
3.2.1. Zapisi IPv6 paketa

IPv6 paketi koriste 128-bitne adrese te su iz tog razloga paketi nešto veći. Međutim, određena polja koja su se tijekom korištenja IPv4 protokola iskazala kao problematična, ili slabo korištena, izbačena su ili preimenovana.

Zaglavlje IPv6 paketa prikazano je na slici 3, počinje oznakom verzije protokola (4 bita). Za IPv6 u to polje mora biti upisan broj 6. Nakon verzije dolazi oznaka klase prometa te oznaka toka kojemu paket pripada. Oba navedena polja podložna su daljnjim specifikacijama. Zatim dolazi oznaka duljine paketa bez zaglavlja, ali uključivo s eventualnim opcijama koje nailaze nakon zaglavlja. Polje *Next Header* je ekvivalentno polju *Protocol* iz IPv4 pri čemu to polje može sadržavati i oznaku opcije koja dolazi nakon zaglavlja. Naime ideja je da se u to polje upiše kod opcije koja dolazi nakon zaglavlja, ili oznaka idućeg protokola koji se prenosi u IPv6 paketu. Sve oznake koje se mogu pojaviti u ovom

polju definira IANA te se mogu pronaći na Internetu [FIXME:IANA]. Iduće polje je *Hop Limit* koje ima ekvivalentu ulogu kao i polje TTL iz IPv4 paketa. Naime, prvotna ideja polja TTL bila je da se umanjuje za 1 svake sekunde. Međutim, u praksi se je vrijednost polja umanjivala za 1 nakon svakog usmjernika, tj. svakog *skoka* koji bi paket napravio. Iz tog razloga se to polje sada naziva *Hop Limit*, tj. maksimalan broj skokova koje paket može napraviti. Na kraju dolaze 128-bitne adrese odredišta i izvorišta.

U odnosu na IPv4 uklonjena su neka polja. Za početak uklonjena su sva polja koja su se koristila za fragmentaciju IP paketa. Naime, fragmentacija je tijekom korištenja IPv4 protokola iskazala niz nedostataka. Primjerice, kada bi se neki IP paket razdijelio u nekoliko fragmenata svi fragmenti su morali stići na odredište. Ako bi koji fragment nedostajao nije se mogao složiti paket, a istovremeno su se trošili resursi za prijenos svih fragmenata i njihovo privremeno pohranjivanje do ponovnog sastavljanja IP paketa. Budući da je u određenim slučajevima fragmentacija ipak nužna, jer lako se može desiti da je paket prevelik za podatkovni sloj, rješenje koje je usvojeno pretpostavlja otkrivanje maksimalne veličine paketa u prijenosu. Također, uklonjeno je polje s kontrolnom sumom.



Slika 3. Format zaglavlja IPv6 paketa

Prilikom pisanja IP paketa u tekstu koristiti će se sljedeći zapis:

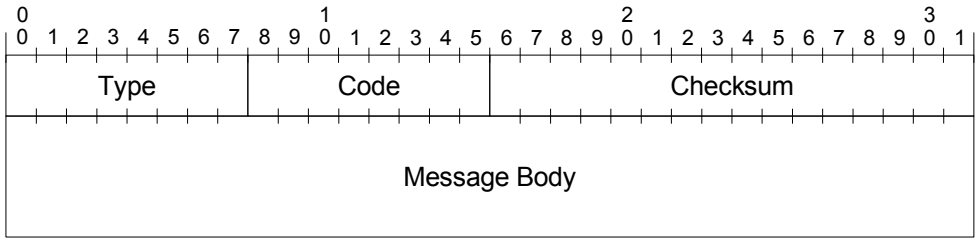
```
IPv6 (odredišna adresa, izvorišna adresa, TTL, koristan teret)
```

U određenim situacijama nisu bitni svi podaci, primjerice tijekom jednostavne analize proslijeđivanja paketa kroz neki usmjernik u većini slučajeva bitne su samo odredišna i izvorišna adresa. U tom slučaju koristi se skraćeni zapis:

```
IPv6 (odredišna adresa, izvorišna adresa)
```

3.3. ICMP

Upravljačke informacije mrežnog sloja Interneta prenose se uz pomoć protokola ICMP (engl. Internet Control Message Protocol) [RFC0792]. Protokol ICMP definira određen broj poruka i njihovo značenje, a uvijeti u kojima te poruke nastaju definirane su dodatnim specifikacijama [RFC1812,RFC1122]. ICMP za prijenos koristi IP protokol, a identificiran je kao protokol broj 1. Drugim riječima, svaki IP paket koji nosi ICMP poruku u. polju *Protocol* zaglavlja IP paketa sadrži vrijednost 1. Nakon zaglavlja IP paketa dolazi zaglavlje ICMP protokola s podacima. Svako ICMP zaglavlje započinje s zajedničkom 32-bitnom riječi čiji format je prikazan na slici 4.



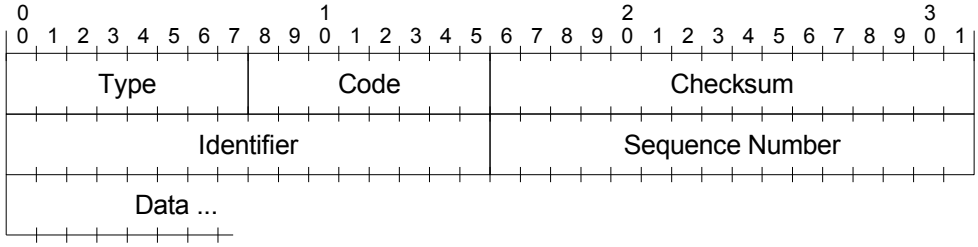
Slika 4. Zajednički dio svih ICMP poruka

U prvih osam bitova (polje *Type*) upisan je tip ICMP poruke koja se prenosi. U iduće polje (*Code*)upisan je podtip poruke. Konačno, polje *Checksum* sadrži kontrolnu vrijednost cijele ICMP poruke. U tijelu poruke nalaze se upisani podaci specifični za pojedinu vrstu poruke.

```
IP (odredišna adresa, izvorišna adresa, ICMP(Tip, ...))
```

3.3.1. *Echo* i *Echo Reply* poruke

Echo poruka prepoznaje se po upisanoj vrijednosti 8 u polje *Type*. Podtip se ne koristi i mora imati upisanu vrijednost 0. Potpuni format Echo poruke prikazan je na slici 5.



Slika 5. Format Echo poruke

Po primitku Echo poruke odredište treba generirati odgovor, poruku tipa *Echo reply*. Tip poruke Echo reply je 0, dok je podtip također 0. U osnovi odgovor se kreira koristeći Echo poruku kojoj se zamijene izvorište, odredište i tip poruke te ju se potom vratiti pošiljatelju. Na taj način sačuvani su svi podaci koji su poslani, a koji obično uključuju vrijeme kada je Echo poruka poslana. Format poruke Echo reply identičan je formatu poruke Echo prikazane na slici 5.

Za zapisivanje Echo poruke u zadacima i prilikom analize mrežnog prometa koristiti će se sljedeća notacija:

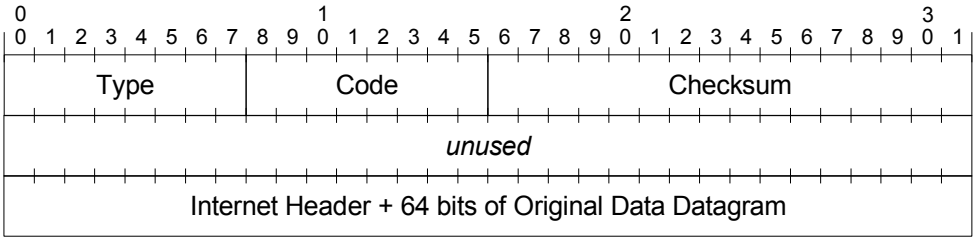
```
IP (odredišna adresa, izvorišna adresa, ICMP(Echo, Sequence, Data))
```

Za zapisivanje Echo Reply poruke koristiti će se sljedeći zapis:

```
IP (odredišna adresa, izvorišna adresa, ICMP(EchoReply, Sequence,
Data))
```

3.3.2. Poruka isteka vremenskog ograničenja

Poruka isteka vremenskog ograničenja (engl. Time Exceeded Message) šalje se kao informacija da je prilikom obrade IP paketa isteklo neko vremensko ograničenje. Format poruke prikazuje slika 6.



Slika 6. Format poruke isteka vremenskog ograničenja

Tip poruke vremenskog ograničenja je 11, dok se u podtipu pobliže definira koje vremensko ograničenje je isteklo. Ako se u podtipu nalazi upisana vrijednost 0 tada je isteklo vrijeme života paketa, tj. polje TTL je doseglo nulu. U takvim slučajevima usmjernik odbacuje paket i generira navedenu ICMP poruku isteka vremenskog ograničenja.

Kada je u polje podtipa upisana vrijednost 1 tada prilikom defragmentiranja IP paketa neki od fragmenata nisu stigli, a u međuvremenu je isteko unaprijed definirano vremensko ograničenje u usmjerniku koji je obavljao defragmentaciju paketa.

3.3.3. Poruka nedostižnog odredišta

3.3.4. Poruka redirekcije

3.3.5. Poruka indikacije problema s parametrima

3.4. ICMPv6

3.5. Kompresija zaglavlja IP paketa

RFC3173

3.6. Enkapsulacija IP paketa u mrežnim nižim slojevima

RFC894, RFC3718

4. Usmjernici i struktura Interneta

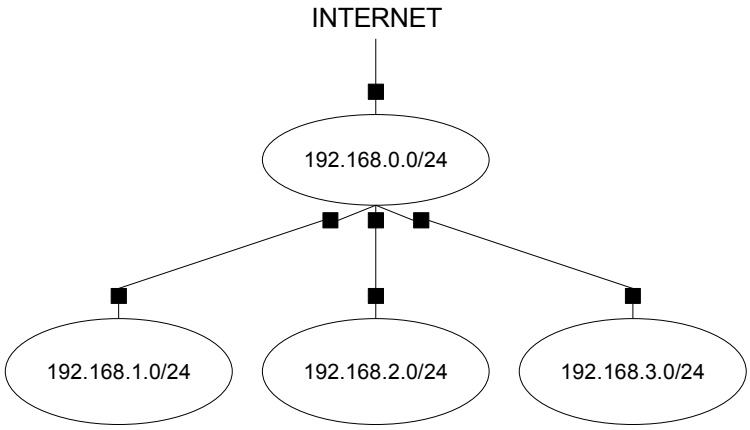
RFC3768 VRRP, RFC3746 ForCES

Usmjernici su uređaji koji se također mogu nazvati i "ljepilo" Interneta. Naime, oni povezuju Internet, ili preciznije rečeno, mreže koje čine Internet, u jednu cjelinu i omogućavaju razmjenu podataka između bilo koja dva računala. Zbog izuzetno velikog broja uređaja na Internetu te vrlo dinamičkih uvjeta u kojima stalno mnoštvo uređaja nestaje i pojavljuje se, usmjernici ne vide Internet kao računala s IP adresama, već kao mreže! To pravilo vrijedi za sve usmjernike kroz koje prolazi paket, osim zadnjega usmjernika. Naime, zadnji usmjernik direktno isporučuje paket računalu te zbog toga on mora znati kako mu pristupiti.

U ovom poglavlju pozabaviti ćemo se prvo strukturom Interneta, a potom ćemo definirati model usmjernika i njegovu funkcionalnost te se pozabaviti radom usmjernika.

4.1. Struktura Interneta

Kako bi opisali strukturu Interneta krenut ćemo od slike 7. Na slici su prikazane neke fiktivne mreže pri čemu se jedna mreža koristi za pristup ostatku Interneta. Takva mreža, koja prenosi podatke za druge mreže, naziva se *tranzitna mreža*. Općenitije rečeno, svaka mreža koja prihvaća podatke čije odredište je u nekoj drugoj mreži, te koja šalje podatke koji ne potiču iz nje same naziva se tranzitna mreža. Ta mreža može biti, primjerice, pružatelj Internet usluga (ISP, Internet Service Provider). Preostale tri mreže mogu pripadati neki tvrtkama i one su *krajnje mreže* (stub networks). Na slici su s crnim kvadratima označeni usmjernici.



Slika 7. Pojednostavljeni primjer dijela Interneta

Pružatelji Internet usluga obično dobiju blok IP adresa od svog nadređenog pružatelja mrežnih usluga (NSP, Network Service Provider) ili od neke institucije zadužene za dodjelu IP adresa (ref na RIPE, ARIN, ...). Kako bi pružatelj Internet usluga mogao spajati mreže korisnika na svoju infrastrukturu on mora podijeliti svoj blok adresa između sebe i svojih korisnika. Ako s ponovo vratimo na sliku 7 možemo pretpostaviti kako je naš pružatelj Internet usluga dobio blok adresa 192.168.0.0/16. Taj blok adresa podijeljen je u podblokove čija mrežna maska ima 24 bita. S tim je omogućeno spajanje 256 mreža (192.168.0.0/24, 192.168.1.0/24, ..., 192.168.255.0/24), pri čemu 0. mrežu koristi sam ISP, a preostalih 255 stoji na raspolaganju korisnicima.

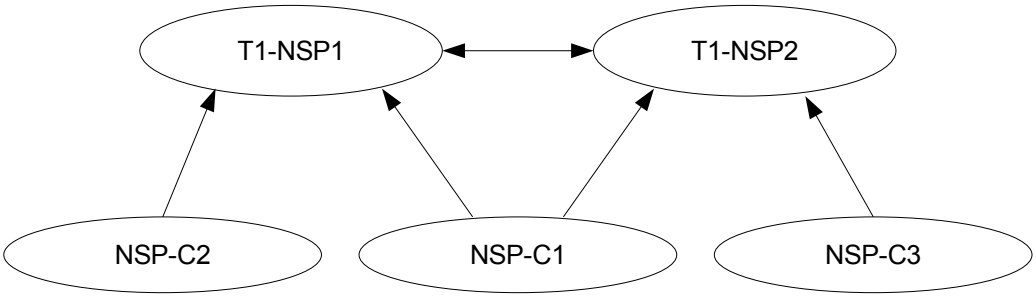
Pri opisu navedene slike uveli smo pojmove *pružatelja Internet usluga* te *pružatelja mrežnih usluga*. U osnovi, korisnici pružatelja Internet usluga su pojedinci te male tvrtke. Također se pružatelji Internet usluga rasprostiru na relativno malom zemljopisnom području. Za razliku od njih, korisnici pružatelja mrežnih usluga su uglavnom drugi pružatelji mrežnih usluga kao i pružatelji Internet

usluga. Dodatna karakteristika NSP-ova je da imaju infrastrukturu koja se rasprostire na vrlo velikom području, čak i po cijeloj Zemlji. Dakle, oni posjeduju vrlo skupu opremu i prekooceanske optičke kablove.

Internet međutim nema pravilnu strukturu kao što je to prikazano na slici 7. Za početak, ne postoji vrhovni NSP već na Internetu ima nekoliko davatelja usluga čija infrastruktura povezuje kontinente i koji isključivo prodaju svoje usluge. To su primjerice UUNet, Sprint, MCI, itd. Ti NSP-ovi, koje ćemo zvati NSP-ovi prvog reda (Tier 1) međusobno su povezani na bazi partnerskih ugovora (peering agreements).

Partnerskim ugovorima NSP-ovi osiguravaju razmjenu podataka s ostalim NSP-ovima kako bi njihovi klijenti mogli pristupiti svakom računalu na Internetu. Također, svi NSP-ovi se obavezuju da će prihvaćati promet od svojih partnerskih NSP-ova te ih prosljeđivati svojim klijentima.

Nadalje, na spomenute NSP-ove spajaju se regionalni pružatelji mrežnih usluga. Pri tome, regionalni NSP-ovi mogu istovremeno koristiti više NSP-ova prvog reda. Primjer takvog povezivanja prikazuje slika 8. Na toj slici prikazan je NSP-C1 koji je istovremeno vezan na T1-NSP1 i T1-NSP2, dok su navedena dva NSP-a prvog reda međusobno povezana partnerskim ugovorom. Osim NSP-C1, na slici su prikazana još dva NSP-a koja koriste usluge NSP-a prvog reda. To su NSP-C2 i NSP-C3.



Slika 8. Primjer povezivanja NSP-ova

Zbog toga što NSP-C1 ima više pristupa na ostatak Interneta kažemo da je *višeprisutan* (multihome). Više je razloga za višeprisutnost:

- *povećanje pouzdanosti*

  Povećanje pouzdanosti je očita. Ako veze koje povezuju NSP-C2 ili NSP-C3 i njihove NSP-ove prvog reda prekinu, navedeni NSP-ovi su odsječeni od ostatka Interneta. Za razliku od njih, ako NSP-C1 razmjenjuje podatke s NSP-C2 putem T1-NSP1 te mu lijevi link prestane funkcionirati, tada se promet može preusmjeriti na desni link na T1-NSP2, potom na T1-NSP1 i konačno do NSP-C2.

- *smanjenje troškova*

  Primjerice, moguće je da NSP-C2 plaća manje za promet koji ostaje unutar T1-NSP1, a više za promet koji mora ići van navedenog NSP-a prvog reda. Očito je da u takvim slučajevima NSP-C2 manje plaća za komunikaciju s NSP-C1, a više u slučaju komunikacije s NSP-C3. Ako se prenosi puno prometa između NSP-C2 i NSP-C3 u određenom trenutku isplati se zakupiti poseban vod prema T1-NSP2 kako bi se smanjila cijena usluga pristupa Internetu.

- *povećanje propusnosti*

- *politički i sigurnosni razlozi*

Međutim, mogućnosti povezivanja tu ne prestaju budući da bilo koja mreža na Internetu može ući u partnerski ugovor s bilo kojom drugom mrežom što se dosta često i radi. Primjerice, ako NSP-C2 i NSP-C3 međusobno razmjenjuju puno podataka u približno jednakim omjerima tada oni mogu ući u partnerski ugovor (partnerski odnos). Na taj način smanjuju trošak budući da taj promet sada ne ide preko NSP-ova koji naplaćuju prema količini prometa. Dosta često uvjeti pod kojima dva entiteta stupaju u partnerski odnos nisu javno poznati te su zaštićeni kao poslovna tajna.

Do sada su spominjane samo mreže kao način grupiranja više IP adresa. Međutim, na Internetu se i računalne mreže grupiraju u *autonomne sustave*. Autonomni sustavi identificiraju se uz pomoć 16 bitnog broja, a alokacija tih brojeva obavlja se uz pomoć regionalnih institucija.

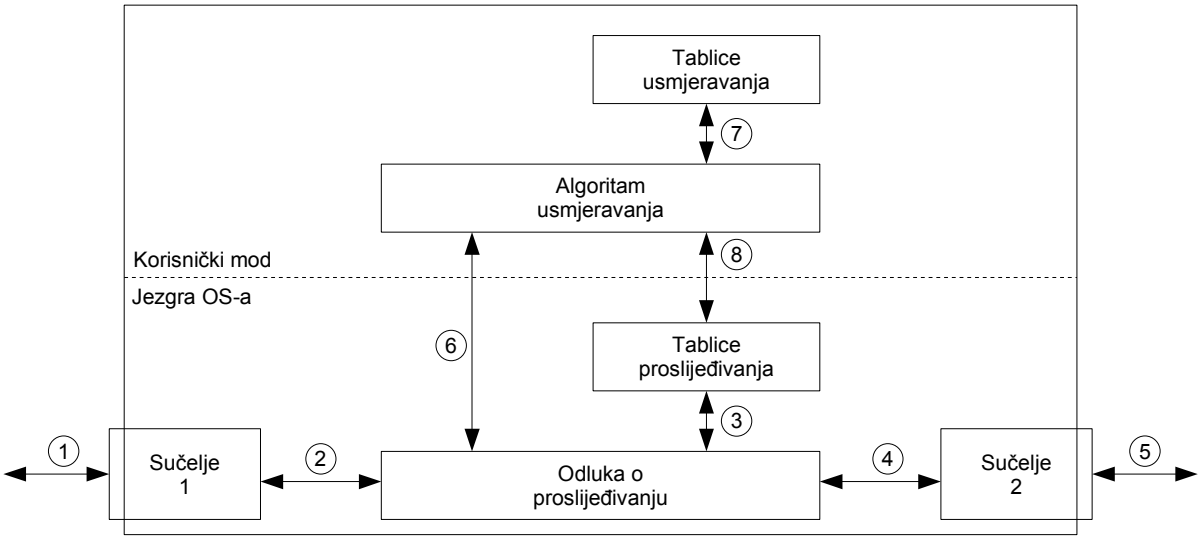
Multihoming: RFC2260, RFC4116

Internet: RFC1958, RFC3439

Dodjela IP adresa: RFC1518, RFC1519, RFC2008

4.2. Model usmjernika

Na slici 9 prikazan je pojednostavljeni model usmjernika koji posjeduje dva sučelja. Sličan model vrijedi za usmjernik s bilo kojim brojem sučelja. Treba primjetiti kako usmjernik na svako svoje sučelje može i slati i primati okvire!



Slika 9. Model usmjernika s dva sučelja

Na usmjerniku se paralelno odvijaju dva procesa. Prvi je proces *proslijeđivanja* (engl. forwarding), a drugi je proces izgradnje *tablice proslijeđivanja* (engl. forwarding information base – FIB). Proces izgradnje tablice proslijeđivanja naziva se *usmjeravanje* (engl. routing) te se tijekom procesa usmjeravanja izgrađuje *tablica usmjeravanja* (engl. routing information base – RIB).

Kako bi opisali proces proslijeđivanja pretpostavimo da računalo koje se nalazi na 1. sučelju šalje paket računalu koje se nalazi priključeno na sučelje 2. Redoslijed koraka koje taj paket prolazi je sljedeći:

1. Okvir pristiže na 1. sučelje koje ga prihvaća, provjerava integritet okvira i smješta njegov sadržaj (*payload*) u privremenu memoriju. Sadržaj okvira je IP paket!
2. Slijedi prihvat paketa od strane *sustava proslijeđivanja* (engl. forwarding engine).
3. Sustav proslijeđivanja pretražuje tablicu proslijeđivanja kako bi odredio što učiniti sa paketom.

4. Po tablici usmjeravanja paket treba poslati kroz sučelje 2, te se paket prebacuje u privremenu memoriju 2. sučelja.
5. Sučelje šalje okvir sa paketom na mrežu.

Kao što je rečeno dva su paralelna procesa u usmjerniku. Drugi proces je izgradnja tablice proslijeđivanja. To se može obavljati statički ili dinamički. Statički način je prihvatljiv za računala i na usmjernicima koji se nalaze u određenim jednostavnim mrežnim konfiguracijama. Dinamički način se koristi u složenijim situacijama budući da se osvježavanje tablica vrši automatski bez intervencije čovjeka.

Cilj dinamičkih algoritama je da pronađu što bolji put između bilo koja dva čvora u mreži, a također da brzo reagiraju u slučaju promjena u topologiji. Korišteni algoritmi usmjeravanja na Internetu su RIP, OSPF, BGP, ISIS, IGRP, EIGRP i još nekoliko drugih. Karakteristika tih algoritama je da se periodički izmjenjuju poruke o stanju mreže te se na osnovu tih poruka računaju najbolji putevi kroz mrežu. U slučaju modela usmjernika, prikazanog na slici 9, kada takav paket pristigne u usmjernik tijekom analize odlučuje se što treba učiniti sa paketom. Ta odluka donosi se u sustavu za proslijeđivanje. Taj sustav će vidjeti da je taj paket namijenjen lokalnom računalu. U tom slučaju odvija se sljedeći niz događaja:

6. Sustav za proslijeđivanje predaje taj paket lokalnom procesu koji je u ovom našem slučaju algoritam usmjeravanja.
7. Algoritam usmjeravanja osvježava stanje u tablici usmjeravanja i preračunava novo stanje nastalo tom promjenom.
8. U slučaju da je došlo do kakve promjene algoritam usmjeravanja modificira tablicu proslijeđivanja.

Očito je osnovna funkcionalnost usmjernika dosta jednostavna, no ipak uprkos toj jednostavnosti usmjernici imaju čitavo mnoštvo dodatnih funkcija koje usporavaju njihov rad. To je pogotovo problem kod usmjernika u jezgri Interneta (*core*). Kako bi se riješio problem brzine usmjernicima se dodaje specijalizirano sklopovlje koje im omogućava vrlo brz rad. To specijalizirano sklopovlje dolazi u vidu ASIC čipova koji se mogu nabaviti na tržištu, a veći proizvođači usmjernika ih razvijaju sami. Dodatna prednost korištenja sklopovlja dostupnog na tržištu je brži izlazak u prodaju i smanjenje konačne cijene uređaja.

4.3. Proslijeđivanje

Kao što je već rečeno u uvodnom dijelu ovog poglavlja, temeljna zadaća usmjernika je proslijeđivanje paketa mrežnog sloja. Na Internetu to su IP paketi verzije 4 i verzije 6. Tijekom odluke o daljnjoj sudbini paketa sustav proslijeđivanja konzultira tablicu proslijeđivanja. U nastavku će biti opisana pojednostavljena verzija te tablice.

Tablica proslijeđivanja sastoji se od po jednog zapisa za svaku mrežu koju usmjernik poznaje. Zapis se inače naziva i *ruta*. Svaki taj zapis sadrži adresu odredišne mreže, idući čvora kojemu je potrebno proslijediti paket (*next hop*) i izlazno sučelje. Primjer tablice proslijeđivanja dan je u tablici 4.

Tablica 4. Primjer pojednostavljene tablice proslijeđivanja

Odredišna mreža	***Idući čvor***	***Izlazno sučelje***
192.168.2.0/30	192.168.2.150	eth1
161.53.65.0/24		eth0
192.168.2.0/24		eth1
161.53.0.0/16	161.53.65.1	eth0

U toj tablici prvi stupac predstavlja adresu odredišne mreže. Adresa je pisana u CIDR obliku, tj. računalni dio zamijenjen je nulama, a dodana je mrežna maska koja određuje koliko bitova s lijeva čini mrežni dio, tj. mrežnu adresu. Drugi stupac tablice prestavlja idući čvor kojemu je potrebno proslijediti paket. Ako za neku mrežu u tom stupcu nije upisano ništa tada se ta mreža nalazi direktno povezana s usmjernikom. Konačno, treći stupac određuje izlazno sučelje kroz koje je potrebno proslijediti paket kako bi on pristigao do odredišta.

Pretraživanje tablice obavlja se po *najduljem preklapanju mrežne adrese*. Da bi objasnili točno što se misli pod tim primjetimo prvo kako su retci u tablici poredani po sve manjem mrežnom prefiksu. Na vrhu se nalazi mreža koja u svojoj adresi posjeduje 30 bitova, a na dnu se nalazi mreža koja posjeduje samo 16 bitova u mrežnom dijelu. Pretraživanje kreće od prvog retka i prestaje u trenutku kada se utvrdi poklapanje. Za svaki redak računa se slijedeći izraz:

$$IP_{net} == IP_{dest} \,\&\, ((2^n - 1) << (32 - n))$$

U tom izrazu IP_{net} je mrežna adresa iz tablice proslijeđivanja. IP_{dest} je odredišna adresa koja se uzima iz IP paketa, a n je broj bitova u mrežnoj adresi. U slučaju da je vrijednost tog izraza točno tada se koriste podaci ostatka zapisa, tj. sljedeći čvor i izlazno sučelje.

U tablicama se često nalazi i jedna posebna ruta. Ona je posebna po tome što joj je duljina mrežne maske nula pa prema tome ima sljedeći oblik:

0.0.0.0/0

Osim tog oblika koristi se i kraći oblik 0/0, te riječ *default*. Jedna specifičnost te rute je da se provjerava zadnja (budući da je to najmanja mrežna maska koja postoji). Druga specifičnost je da uvijek postoji poklapanje. Naime, provjera kreće sa pretpostavkom pozitivnog ishoda, a kako se zahtijeva provjera nula bitova to znači da je konačan rezultat pozitivan i da se dotična ruta koristi za proslijeđivanje paketa.

Budući da pretraživanje tablica usmjeravanja može biti vrlo dugotrajan proces koriste se posebne podatkovne strukture uz pomoć kojih se cijeli postupak značajno ubrzava.

4.3.1. Podatkovne strukture za pohranu tablica proslijeđivanja

4.3.2. Proslijeđivanje u sklopovlju

5. Algoritmi usmjeravanja

5.1. RIP

RIP (Routing Information Protocol) je najstariji i najjednostavniji protokol usmjeravanja. Protokol za izračun tablica proslijeđivanja koristi algoritme vektora udaljenosti (*distance vector algorithm*).

Postoje dvije verzije RIP algoritma. Verziji 1 [RFC1058] nastala je u vrijeme podjele IP adresa u klase te se u toj verziji implicitno podrazumijevaju mrežne maske. Uvođenjem bezklasnog načina adresiranja bilo je nužno promijeniti protokol kako bi uključio i razmjenu mrežnih maski. Ta modifikacija učinjena je u verziji 2 protokola [RFC2453] koja se danas se isključivo koristi na Internetu i intranetima.

RIP protokol periodički razmjenjuje

5.2. OSPF

5.3. BGP

RFC1773, RFC1997

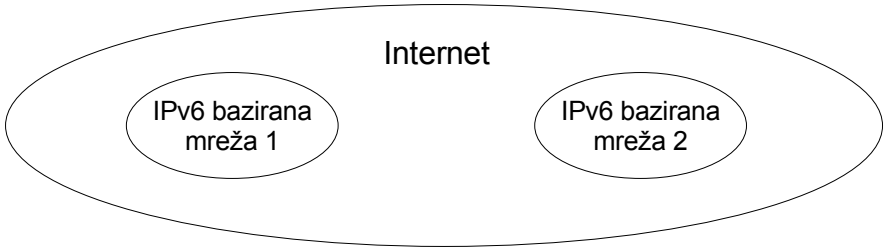
5.4. ISIS

6. Naprednije tehnike mrežnog sloja

6.1. Tuneliranje

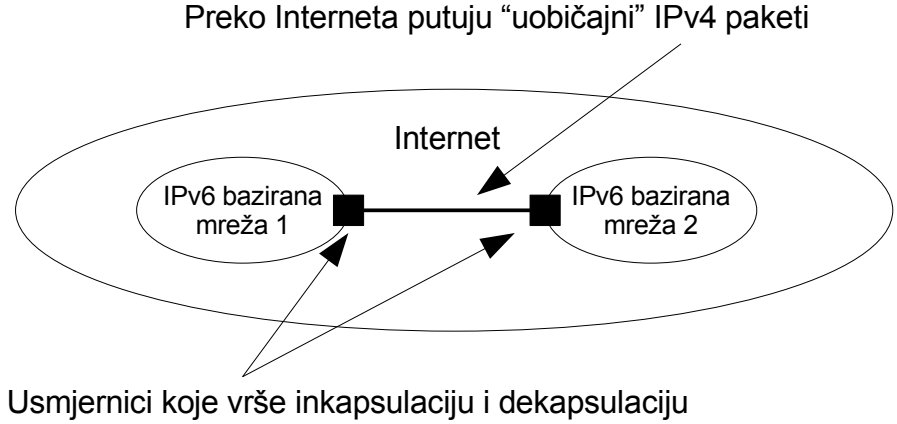
Općenito je tuneliranje definirano kao inkapsulacija protokola istog ili nižih slojeva. Kada razmatramo tuneliranje na nivou mrežnog sloja tada to znači da se unutar IP paketa prenose IP paketi. Primjena tuneliranja je mnogostruka; moguće je korištenjem infrastrukture Interneta povezati mreže koje inače ne bi mogle međusobno komunicirati, zatim tuneliranje se koristi prilikom realizacije virtualnih privatnih mreža, a također u slučajevima kada je potrebno utjecati na put što ga paket prolazi na Internetu, itd.

Uzmimo za primjer potrebu povezivanja dvije mreže koje putem Interneta ne mogu direktno komunicirati, slika 10. Konceptualno ta slika prikazuje dvije zamišljene mreže – dva otoka – unutar cijelog Interneta. Specifičnost tih mreža je da se baziraju na IPv6 mrežnom protokolu, dok se cijeli Internet bazira na IPv4 mrežnom protokolu. Za povezivanje te dvije mreže nije moguće koristiti IPv6 pakete budući da IPv4 bazirana infrastruktura Interneta ne zna proslijeđivati te pakete. Rješenje je da se koristi tuneliranje, tj. da se unutar IPv4 stavljaju IPv6 paketi.



Slika 10. Primjer mreža "otoka" na Internetu

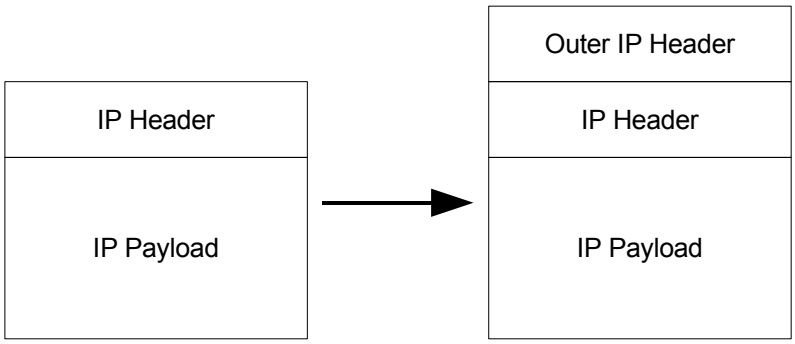
Slika 11 prikazuje situaciju u kojoj se koristi tuneliranje za prijenost IPv6 paketa preko IPv4 baziranog Interneta. U suštini ideja je sljedeća. Kada neki IPv6 paket iz mreže 1 treba otići do mreže 2 tada ga usmjernik mreže 1 upakira u IPv4 paket. Potom tako nastali paket prepusti Internet infrastrukturi da ga prenese do usmjernika druge mreže. Čim pristigne usmjernik druge mreže ukloni IPv4 zaglavlje i dobiveni IPv6 paket prebaci u mrežu 2.



Slika 11. Primjena tuneliranja za povezivanje dvije IPv6 mreže

Postoji više normi koje propisuju način inkapsulacije različitih protokola unutar IP paketa, primjerice [RFC1701, RFC2003].

Osnovna inkapsulacija definira način upisivanja IP paketa unutar IP paketa [RFC2003]. Suštinu inkapsulacije prikazuje slika 12. U osnovi, na IP paket dodaje se još jedno IP zaglavlje koje se naziva *vanjsko IP zaglavlje* (engl. Outer IP header).



Slika 12. Inkapsulacija IP paketa unutar IP paketa [RFC2003]

Polja vanjskog IP zaglavlja postavljaju se vrijednosti definirane u tablici .

Polje	***Vrijednost***
Version	4
IHL	Veličina vanjskog zaglavlja mjerena u 32-bitnim riječima.
TOS	Prekopiran od unutarnjeg zaglavlja.
Odredišna adresa	Drugi kraj tunela
Izvorišna adresa	Polazni kraj tunela

6.2. NAT

rfc2663, rfc3022, rfc2766

6.3. Difuzija u grupi

6.4. Mobilni IP protokol

6.5. MPLS

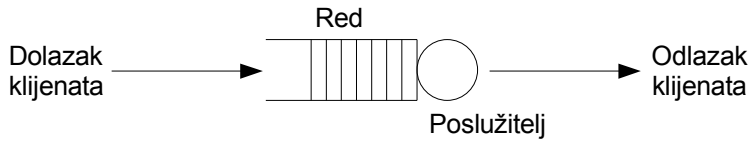
6.6. GMPLS

7. Kvaliteta usluge

Zbog sve veće popularnosti Interneta javljaju se aplikacije koje postavljaju nove zahtijeve na računalnu mrežu. Istovremeno stare aplikacije postaju sve bitnije u svakodnevnom radu. Primjerice, sve je veća upotreba Interneta za distribuciju video i audio sadržaja, zatim poslovanje poduzeća sve više ovisi o ispravnom pristupu Internetu, a velika telekomunikacijska poduzeća polako migriraju svoju cjelokupnu infrastrukturu prema Internetu i njegovim tehnologijama. Zajednička karakteristika svih navedenih primjera, a i niza drugih aplikacija, je da očekuju nekakvu garanciju od mreže, tj. određenu kvalitetu usluge (*Quality of Service*, *QoS*).

U svojim počecima, a dobrim dijelom i tijekom svog cjelokupnog postojanja, Internet je zamišljen i izgrađivan kao *best-effort* mreža. To znači da mreža ne daje nikakve garancije da će isporučeni paket stići na odredište, da će stići u određenom vremenu ili da će doći samo jedna kopija paketa. Mreža samo jamči da će učiniti sve što može da taj paket, neoštećen, što ranije isporuči na odredište. Međutim, dešava se da paketi nemaju isto kašnjenje, a u slučaju slanja više uzastopnih paketa ni njihov redoslijed nije očuvan. Neki od tih problema riješeni su na višim protokolima, kao primjerice pouzdanost isporuke i sačuvan redoslijed u slučaju TCP protokola. Međutim dobar dio problema nije moguće riješiti bez značajnijih zahvata u samom mrežnom sloju.

Svaki usmjernik koji nema mogućnosti pružanja kvalitete usluge možemo modelirati upotrebom reda za svako izlazno sučelje koje posjeduje. Model reda zajedno s izlazom je prikazan na slici 13. Na slici umjesto *sučelje* piše *poslužitelj*, a umjesto *paketa* koristi se izraz *klijent*. Razlozi za takvu terminologiju bit će opisani kasnije.



Slika 13. Model jednog sučelja usmjernika

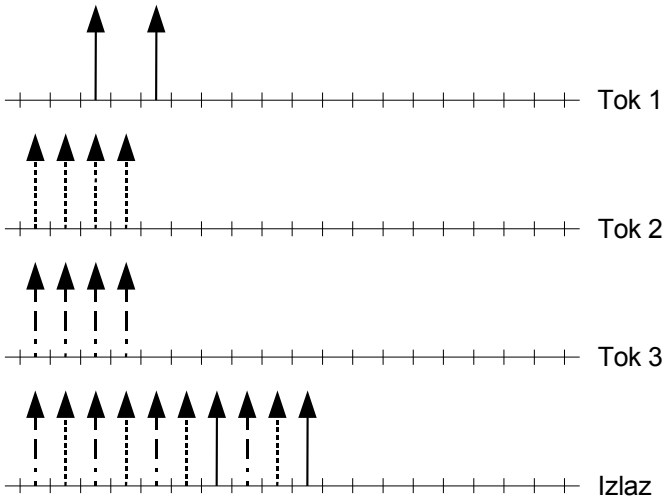
Svaki paket koji pristigne u usmjernik, nakon odluke o proslijeđivanju, šalje se na izlazno sučelje. Međutim, kada je izlazno sučelje zauzeto on se stavlja u red, tj. mora se privremeno pohraniti u memoriju. Kada se sučelje oslobodi, šalje se najstariji paket koji se nalazi u redu. Upravo opisani model usmjernika koristi najjednostavniji način raspodjele izlazne veze, tzv. FIFO (engl. First In First Out). Umjesto skraćenice FIFO često se koristi i skraćenica FCFS (engl. First Come First Serve). U slučaju takvog načina posluživanja paket koji prvi stigne biti će i prvi poslužen. Takvi načini raspoređivanja očito nisu pretjerano efikasni, odnosno pravedni. Ipak, ako je kapacitet komunikacijske veze znatno veći od potreba tada su ti algoritmi raspoređivanja dovoljno dobri.

Kako bi lakše pratiti materiju vezanu uz kvalitetu usluge potrebno je prvo definirati terminologiju. Osnovni pojam predstavlja *tok* (engl. flow). Svaki tok sastoji se od niza *paketa*, s određenim zajedničkim karakteristikama. Primjerice, u slučaju TCP veza zajedničke karakteristike su parovi IP adresa i pristupa na svakom kraju veze, zatim prilikom izvršavanja `ping` naredbe to su IP adrese krajeva i tipovi ICMP poruke. Dakle, kroz usmjernike u svakom trenutku prolazi određen broj tokova.

Nisu svi tokovi jednake važnosti i u tome leži temelj kvalitete usluge. To više dolazi do izražaja što je veće opterećenje sustava, tj. više paketa treba poslati preko istog sučelja. U tom slučaju raste red i poslužitelj treba iz reda odabrati klijenta kojeg će obraditi. Neki klijenti imaju veći prioritet, bilo zbog svoje važnosti ili zbog spremnosti korisnika da plati za bolju uslugu.

Na primjeru prikazanom na slici 14 biti će objašnjeni problemi koji mogu nastati upotrebom raspoređivanja paketa po najjednostavnijoj FCFS metodi. Slika prikazuje tri toka. Za prvi tok je specificirano vrijeme između nailaska dva uzastopna paketa dvije vremenske jedinice, za drugi tok je

to 5 jedinica i za treći tok je također 5 jedinica. Očito je kako drugi i treći tok ne poštuju svoje specifikacije te šalju podatke daleko bržim ritmom od navedenog. Dolazak svakog paketa simbolički je prikazan strelicom. Za sve pakete pretpostavlja se kako su iste veličine, te da njihov prijenos traje točno jednu vremensku jedinicu.



Slika 14. Prikaz FIFO/FCFS posluživanja dolaznih paketa

Zadnja linija slike prikazuje izazni mrežni promet iz poslužitelja. Očito je da iako tok 1 poštuje svoju specifikaciju zbog drugih tokova ne dobija traženu uslugu. Kako bi se izbjegla takva situacija, ali i niz drugih koje dovode do sličnih nepravilnosti, koriste se algoritmi koji na osnovu specifikacija toka, njegova ponašanja u prošlosti i zauzeća poslužitelja odlučuju o redoslijedu posluživanja.

Prvi prijedlog modifikacije jednog reda odnosi se na poslužitelje s beskonačnom količinom memorije. Naime, za svaki tok koji prolazi kroz usmjernik, tj. poslužitelj, rezervira se po jedan FIFO spremnik. Paketi se potom ciklički šalju iz svakog spremnika. U slučaju da neki od tokova pojača intenzitet slanja to neće utjecati na druge tokove.

Navedeni algoritmi implementirani su u raspoređivačima (engl. schedulers) u trenutku kada je odlazna veza slobodna odabiru idući paket koji treba poslati. Raspoređivanje je proces koji se koristi u svakoj situaciji u kojoj više korisnika (klijenata) upotrebljava ograničen broj resursa. U slučaju raspoređivanja paketa ograničeni resurs je sučelje preko kojega se šalju paketi budući da je istovremeno moguće slati samo jedan paket. Općeupotrebljavani izraz za dijeljeni resurs je *poslužitelj*. Klijenti dolaze u sustav i koriste uslugu poslužitelja. Poslužitelj odlučuje da li će obraditi pristiglog klijenta. Ako iz bilo kojeg razloga klijent ne može biti obrađen u tom trenutku tada se on stavlja u red čekanja. Nakon što je klijent obrađen on izlazi iz sustava.

Osim problema s raspodjelom poslužitelja, problem je i memorijski prostor u koji se pohranjuju paketi. U određenim situacijama može se desiti da više nema prostora u memoriji i tada se pristigli klijente odbacuje. Kako bi se odredio paket koji će biti odbačen tako da se ne kažnjavaju tokovi koji se pridržavaju svojih specifikacija koristi se poseban skup algoritama za upravljanjem redovima. Primjeri takvih algoritama u računalnim mrežama su RED (engl. Random Early Detect), RIO, itd.

U stvarnim situacijama poslužitelj u određenim situacijama može imati različit kapacitet posluživanja klijenata. Primjerice, u računalnim mrežama odlazna komunikacijska veza može biti potpuna podmreža čiji kapacitet zbog zakrčenja varira. Također je moguće da procesor u mrežnom elementu nema dovoljno kapaciteta za obradu svih pristiglih paketa. Takve varijacije modeliraju se upotrebom FC (engl. Fluctuation Constrained) ili EBF (engl. Exponentially Bounded Fluctuation) modela. Kada

se poslužitelj modelira upotrebom FC modela to znači da njegov kapacitet nikada ne pada ispod određene razine, tj. radi se o deterministčkom modelu. Ako se poslužitelj modelira EBF modelom tada vjerojatnost da je kapacitet pao ispod određene razine eksponencijalno pada sa padom kapaciteta. U tom slučaju radi se o vjerojatnosnom modelu.

7.1. Karakterizacija prometa

Tijekom analize pojedinih algoritama raspoređivanja potrebno je na neki način karakterizirati dolazni promet. U literaturi koja se bavi tradicionalnom teorijom redova i posluživanja koriste se stohastički procesi, primjerice Poissonov proces, koji je ujedno najčešće upotrebljavan, on-off procesi, Markovljevi modeli itd. Od početka 90-tih koristi se i teorija samosličnosti budući da je otkriveno kako Poissonov proces ne opisuje najbolje podatkovne komunikacije.

U novije vrijeme predloženi su modeli čiji pristup je pronalaženje ograničenja na promet, a ne njegova točnog karakteriziranja. Primjer tih modela su (X_{min}, X_{ave}, I, S_{max}), (σ,ρ), (r,T) i D-BIND. Promet se pokorava (X_{min}, X_{ave}, I, S_{max}) modelu ako je međuvrijeme dolazaka (*inter-arrival time*) između bilo koja dva paketa više od X_{min}, prosječno vrijeme između dolazaka na intervalu I je barem X_{ave} i maksimalna veličina paketa je S_{max}. U modelu prometa (σ,ρ) u bilo kojem intervalu *u* broj bitova je manji od $\sigma+\rho u$. Slično je u modelu (r,T) kod kojega u intervalu T nema više od rT bitova. Model D-BIND je nešto drugačiji od prethodna tri jer se kod njega ne pokušava koristiti jedno ograničenje već se koristi niz ograničenja.

U svakom od navedenih modela nije poznat točan uzorak prometa koji se modelira, ali su poznata ograničenja na taj promet. To je sasvim dovoljno algoritmima koji se koriste za upravljanje prometom budući da na osnovu ograničenja znaju rezervirati potrebnu količinu mrežnih resursa.

7.2. Arhitekture za postizanje kvalitete usluge

Za postizanje kvalitete usluge nije dovoljno korištenje samo dobrih algoritama raspoređivanja, već se različite tehnike postizanja kvalitete usluga moraju koristiti od aplikacije na jednom računalu do aplikacije na drugom računalu. Primjerice, za objavljivanje audio sadržaja potrebno je da aplikacija koje vrši emitiriranje dobija podatke sa diska ili nekog drugog medija u točno određenim intervalima, zatim da dobija dovoljnu količinu procesora, da u svakom nivou ISO/OSI referentnog modela postoje odgovarajući mehanizmi koji garantiraju kvalitetu usluge, da mreža tijekom slanja ne gubi pakete, niti da oni previše kasne u mreži, da aplikacija koja prima pakete također dobija dovoljnu količinu procesora i da je izlazni uređaj dovoljno raspoloživ kako bi se nesmetano obavljala reprodukcija. Sve to, ili barem većina toga, definirana je u sklopu arhitekture za postizanje kvalitete usluge.

Broj predloženih arhitektura za postizanje kvalitete usluge dosta je velik i šarolik. Od istraživačkih eksperimenata do standardiziranih modela. U ovom seminarskom radu bitne su dvije arhitekture koje se koriste na Internetu i uglavnom se bave Internetom. U osnovni radi se o dvije međusobno konkurentne arhitekture. Obje imaju svojih prednosti i mana.

7.2.1. Integrirane usluge

Prva je arhitektura *integriranih usluga*. U toj arhitekturi, prije korištenja računalne mreže potrebno je rezervirati odgovarajući kapacitet u mreži. Rezervacija se obavlja upotrebom RSVP protokola. Tijekom rezervacije može se desiti da mreža odbije zahtijev zbog pomanjkanja slobodnih resursa. Prednost ove arhitekutre je velika fleksibilnost, ali je njen problem skalabilnost. Drugim riječima za svaki tok koji prolazi kroz mrežni uređaj potrebno je držati odgovarajuću evidenciju i trošiti određeno vrijeme na obradu što je u jezgrenim usmjernicima Interneta značajan problem jer kroz njih prolazi vrlo velik broj tokova, vrlo velikim brzinama.

7.2.2. Diferencirane usluge

Druga arhitektura je arhitektura *diferenciranih usluga*. Ta arhitektura dijeli sve tokove u nekoliko grupa, pri čemu svaka grupa dobija određenu kvalitetu usluge. Kojoj grupi pripaka pojedini tok, odnosno paket pojedinog toka, naznačava se korištenjem nekoliko bitova u zaglavlju samog paketa. Prema tome, izvođenje zahtijevnijeg dijela obrade potrebno je vršiti samo u graničnim usmjernicima, dok svi ostali usmjernici promatraju samo te bitove. Dakle, količina informacija koju je potrebno čuvati više ne ovisi o broju tokova već samo o broju klasa usluga. Klase usluga nisu univerzalno definirane već svaki administrativni dio Interneta može definirati vlastite klase i provoditi ih na ulazima u svoju mrežu.

7.3. Algoritmi za upravljanje redovima

7.4. Algoritmi raspoređivanja

U ovom poglavlju detaljnije je obrađen određen broj algoritama za raspoređivanje paketa. Algoritmi se mogu podijeliti u dvije skupine. U prvoj skupini su algoritmi kod kojih dok god ima paketa za slanje poslužitelj neće biti slobodan, to su tzv. *work conserving algorithms*. U drugu grupu spadaju algoritmi kod kojih kada ima paketa za slanje, ako je algoritmom predviđeno slanje u budućnosti poslužitelj, tj. odlazna linija, će biti besposlena dok god ne dođe trenutak kada je predviđeno slanje. Ta grupa algoritama na engleskom se naziva *non-work conserving algorithms*.

Svi algoritmi iz prve grupe, koriste sličan mehanizam poredanog reda s prioritetima (engl. sorted priority queue mechanism). Pripadnici te grupe su WFQ, WF^2Q, SCFQ. U tim mehanizmima sa svakim tokom povezana je varijabla stanja koja se koristi za nadzor i upravljanje prometom tog toka. Nakon što pristigne pojedini paket nekog toka varijabla stanja se ažurira prema (a) rezervaciji koju je načinio tok prije početka slanja, i (b) povijesti prometa tog toka. Paketu se tada dodjeljuje oznaka koja je u direktnoj vezi sa izračunatom vrijednošću varijable stanja. Ta oznaka se koristi kao indeks prioriteta paketa, te se paketi poslužuju prema rastućem indeksu.

Kvaliteta pojedinih algoritama mjeri se s nekoliko parametara. Jedan od bitnijih parametara je i *pravednost* (*fairness*). Taj pojam može se definirati na više načina, ali su dva često upotrebljavana. U prvom načinu definiranja pravednost je vezana uz referentni raspoređivač koji je u potpunosti pravedan (FFQ, opisan kasnije) te što je manje odstupanje pojedinog algoritma od to referentnog raspoređivača, to je algoritam pravedniji. Drugi način definicije je pomoću sljedećeg izraza:

$$\left| \frac{W_f(t_1, t_2)}{r_f} - \frac{W_m(t_1, t_2)}{r_m} \right|$$

U tom izrazu $W_f(t_1, t_2)$ i $W_m(t_1, t_2)$ predstavljaju količinu propusnosti koju su tokovi (procesi) f i m dobili u vremenskom intervalu (t1, t2). r_f i r_m su težine tokova. Što je navedena veličina za pojedini algoritam manja to je algoritam pravedniji. Gornji izraz vrijedi samo u slučaju ako su oba toka u navedenom vremenskom intervalu imali pakete za slanje.

Osim pravednosti za algoritme su bitna i maksimalna i prosječna vremenska kašnjenja, *jitter* i granica na *jitter*.

7.4.1. Algoritam virtualnog sata

Algoritam raspoređivanja nazvan Virtualni sat (*Virtual Clock*) [ZHAN90] pokušava emulirati prijenos paketa upotrebom vremenskog multipleksa (TDM). Svakom pristiglom paketu dodjeljuje se virtualno vrijeme slanja. To vrijeme je trenutak kada bi paket bio poslan da se stvarno koristi TDM. Paketi se potom šalju po rastućem dodijeljenom vremenu.

Temelj algoritma predstavlja varijabla stanja označena s *auxVC*, skraćeno od *auxiliary Virtual Clock*. Ta varijabla se za svaki paket toka koji dođe ažurira po sljedećem pravilu:

$$auxVC_{i,j}^{k} \leftarrow max\{a_{i,j}^{k}, auxVC_{i,j}^{k}\} + Vtick_{i,j}$$

U tom izrazu indeks *i* označava preklopnik u kojemu se obavlja ažuriranje varijable, indeks *j* označava tok na koji se varijabla odnosi i *k* označava pojedini paket toka. $a_{i,j}^{k}$ je vrijeme dolaska *k*-tog paketa toka *j* do preklopnika *i*. $Vtick_{i,j}$ se izračunava na osnovu prosječne količine prometa koju generira *j*-ti tok.

Da bi ilusrirali korištenje danog izraza, uzeti ćemo prethodni primjer, opisan kvalitativno, i za njega proračunati dane izraze. Tok 1 je specificirao prosječno 0.5 paketa/s pa je $Vtick_{i,j}=2$, a tokovi 2 i 3 su specificirali 0.2 paketa/s pa je njihov $Vtick_{i,j}=5$. Inicijalno svi imaju *auxVC* jednak nuli. Sada dolazak i odlazak paketa možemo pratiti uz pomoć tablice 5. U nastavku neće u primjerima biti korišten indeks *j* budući da promatramo izvršavanje algoritma u samo jednom preklopniku pa ne može doći do zabune.

U prvoj koloni tablice vodi se evidencija o vremenskim trenucima. Vrijeme počinje u nekom nultom trenutku i u tom trenutku, kao što je to već rečeno, svi *auxVC* su jednaki nuli. U svakoj koloni tablice sa zaglavljem F_i vodi se evidencija o prispjeću paketa za tok *i*. Primjerice, kada je jedinica u retku koji pripada trenutku 2 i stupcu sa zaglavljem F_1, to znači da je u 2. trenutku pristigao paket koji pripada prvom toku i da je paket veličine 1 jedinice.

Nakon što pristigne svaki paket, računa se novi *auxVC* te se njegova vrijednost pridodjeljuje paketu. U tablici nisu posebno vođene *auxVC* vrijednosti za pojedini paket budući da se oni lako iščitaju. Potrebno je samo za odgovarajući paket pogledati ćeliju desno od njega i to je vrijednost *auxVC*-a za taj paket. Za prvi paket 1. toka, koji dolazi u 3. trenutku, vrijednost varijable a_1^1 je prema tome 3. Pa je prema dakle $auxVC_1^1$:

$$auxVC_1^1 \leftarrow max\{a_1^1, auxVC_1^1\} + Vtick_1 = max\{3,0\} + 2 = 5$$

Tablica 5. Vrijednosti varijabli stanja za algoritam virtalnog sata

T	F_1	$auxVC_1$	F_2	$auxVC_2$	F_3	$auxVC_3$	Izlaz
0		0		0		0	
1		0	1	6	1	6	2(1)
2		0	1	11	1	11	3(1)
3	1	5	1	16	1	16	1(3)
4		5	1	21	1	21	2(2)
5	1	7		21		21	1(5)
6		7		21		21	3(2)
7		7		21		21	2(3)
8		7		21		21	3(3)
9		7		21		21	2(4)
10		7		21		21	3(4)
11		7		21		21	

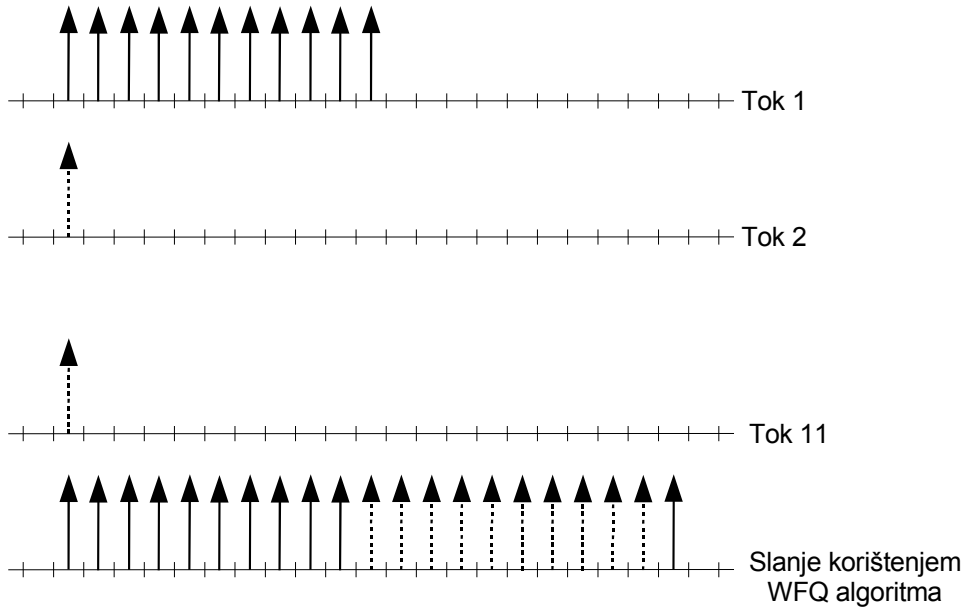
Analogan postupak računanja primjenjen je i za svaki drugi pristigli paket svih tokova. Kada preklopnik treba poslati neki paket on pregledava sve pakete i šalje paket koji ima najmanji *auxVC*. U slučaju da je više takvih paketa, proizvoljno je odabrano da se šalje onaj paket čiji indeks toka je manji. U koloni za izlaz iz mrežnog uređaja koristi se notacija *i*(*T*). *i* je indeks toka čiji paket se šalje, a u zagradi je vrijeme kada je taj paket pristikao u preklopnik. Primjerice, u 7. trenutku šalje se paket 2(3), tj. šalje se paket 2. toka koji je u preklopnik pristigao u 3. trenutku.

7.4.2. WFQ

Weighted Fair Queuing (WFQ) algoritam pokušava aproksimirati algoritam poznat kao *Fair Fluid Queuing* (FFQ). Taj algoritam je skoro istodobno opisan u dva međusobno nevezana rada, pa se zbog toga za njega koristi i ime *Packetized Generalized Processor Sharing*. Sam FFQ se javlja i pod imenom *Generalized Processor Sharing* (GPS). FFQ je malo poopćena inačica algoritma HOL-PS koji je opisao i analizirao Kleinrock. Kod HOL-PS algoritma svaki dolazni tok ima svoj FIFO spremnik. Tijekom bilo kojeg vremenskog intervala u kojemu točno N redova sadrži podatke za slanje poslužitelj obrađuje tih N redova. Svih N paketa poslano je brzinom koja odgovara N-tom dijelu raspoloživog kapaciteta odlazne veze. Za razliku od HOL-PS algoritma, FFQ omogućava da tokovi imaju različitu težinu i shodno tome da dobijaju odgovarajući dio izlaznog kapaciteta.

WFQ je najpoznatija aproksimacija opisanog FFQ algoritma. U trenutku τ kada poslužitelj treba poslati paket on odabire između svih paketa koji čekaju onaj koji bi prvi završio slanje u aproksimiranom FFQ sustavu. U tom trenutku pretpostavlja se da do završetka slanja tog paketa neće pristići niti jedan drugi paket. Manja modifikacija WFQ algoritma je tzv. FQS odnosno *Fair Queuing based on Start Time*, kod kojega se raspoređivanje obavlja po početnom vremenu, a ne po završnom. Za tu modifikaciju pokazano je kako ima prednosti u slučaju raspoređivanja procesa, ali u slučaju raspoređivanja paketa nema nikakve dobiti.

Promotrimo sliku 15 kao primjer raspoređivanja paketa korištenjem WFQ algoritma.



Slika 15. Slanje paketa korištenjem WFQ algoritma raspoređivanja

Jednostavnosti radi pretpostavljeno je kao i u dosadašnjim primjerima da svi paketi imaju jediničnu veličinu, te da je brzina komunikacijske veze 1. Nadalje, pretpostavka je da tok 1 ima zajamčenu brzinu kojoj pristižu paketi (*rate*) 0.5 što znači da šalje po jedan pakete svake dvije vremenske jedinice. Istovremeno svi ostali tokovi, od 2 do 11, imaju zajamčenu brzinu 0.05 što znači da šalju po

jedan paket svakih 20 vremenskih jedinica. Na slici je prikazana situacija kada je tok 1 poslao 11 paketa, jedan za drugim, dok su svi preostali tokovi poslali po 1 paket.

U FFQ sustavu bi za slanje svakog od prvih deset paketa toka 1 bile potrošene 2 vremenske jedinice, a za jedanaesti paket bi bila potrošena 1 vremenska jedinica. Za pakete ostalih jedanest tokova trebalo bi po 20 vremenskih jedinica. Označimo k-ti paket toka j sa p_j^k, tada početno vrijeme prijenosa k-tog paketa, k=1...10, iznosi 2(k-1), a završno je 2k. Početno vrijeme prijenosa paketa p_1^{11} je u 20. vremenskoj jedinici, a završetak je u 21. vremenskoj jedinici. Početak prijenosa paketa p_j^1, j=2...11, su u 0. vremenskoj jedinici, a završetci su u 20. vremenskoj jedinici. Na osnovu tako izračunatih vremena obavlja se raspoređivanje prijenosa paketa u poslužitelju koji koristi WFQ algoritam raspoređivanja. Kao što je rečeno WFQ upotrebljava isključivo završna vremena iz FFQ sustava, pa kada treba poslati prvi paket odlučuje se za prvi paket toka 1 jer on ima najmanje vrijeme završetka. Po tom istom načinu odlučivanja WFQ će poslati prvih 10 paketa toka 1 jer oni imaju najmanja vremena završetka od svih preostalih paketa u trenutku donošenja odluke. Ovdje se prilikom slanja 10-tog paketa koristimo činjenicom da ako dva paketa imaju isto vrijeme završetka tada se uzima onaj čiji indeks toka je manji. U trenutku kada je poslano prvih 10 paketa 1. toka odluka je između 11. paketa toka 1 i paketa svih ostalih tokova. Sada 11. paket toka 1 ima veće vrijeme završetka, 21, pa se počinju slati paketi svih ostalih tokova, a tek nakon njih se šalje i zadnji paket toka 1. Konačni izlaz prikazan je na zadnjoj liniji slike 15.

Varijabla stanja se u sustavu koji koristi WFQ algoritam izračunava na sljedeći način:

$$F_{i,j}^k \leftarrow max\{V_i(a_{i,j}^k), F_{i,j}^{k-1}\} + \frac{L_j^k}{\phi_{i,j}}$$

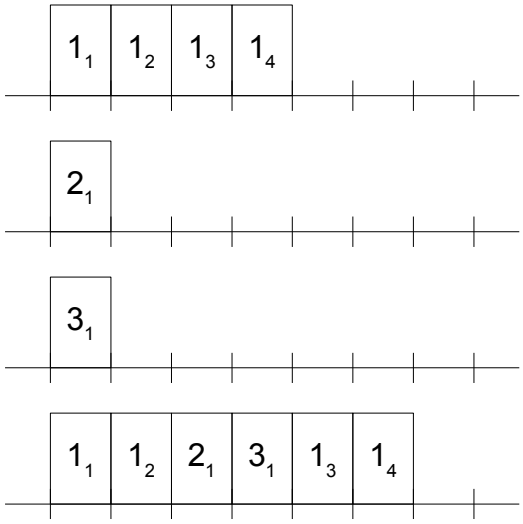
Kao i u slučaju algoritma virtualnog sata i u ovom slučaju se indeks *i* odnosi na preklopnik, *j* na tok i *k* na paket. Varijabla *F* se naziva virtualno vrijeme završetka prijenosa (*virtual finish time*). V(t) je virtualno vrijeme sistema u trenutku t, L_j^k je veličina paketa u bitovima, $a_{i,j}^k$ je vrijeme kada je u preklopnik *i* pristigao *k*-ti paket toka *j*. Fukcija virtualnog vremena zadana je sljedećim izrazom:

$$V(t) = \int_0^t \frac{C}{\sum_{j \in B(t)} w_j} dt$$

U tom izrazu *C* predstavlja kapacitet izlaznog linka, a *B(t)* je skup svih tokova koji u trenutku t imaju pakete u redu za slanje.

Korištenje navedenih izraza biti će prikazano za slučaj tri toka. Prvi tok ima periodu od 2 vremenske jedinice i odgovarajuću težinu $\phi_1 = 0.5$. Druga dva toka su periode 5 vremenskih jedinica i jednake težine, $\phi_2 = \phi_3 = 0.2$, duljine paketa su 1 i brzina veze je C = 1. Na slici 16 prikazani su dolazni tokovi te izlaz za kompletno izvršavanje algoritma.

U trenutku nula virtualno vrijeme je nula te se računaju završna vremena svih paketa. Za prvi paket prvog toka, p_1^1, to je



Slika 16. Način rada WFQ algoritma

$$F_1^1 \leftarrow max\{V(0),0\} + \frac{1}{0.5} = 0 + 2 = 2$$

U tom izrazu za F_1^0 korištena je vrijednost 0. Za prvi paket 2. i 3. toka vrijednost završetka prijenosa prvog paketa je identična i iznosi:

$$F_2^1 = F_3^1 \leftarrow max\{V(0),0\} + \frac{1}{0.2} = 5$$

Kako paket prvog toka ima ranije vrijeme završetka to se on prvi šalje. Njegovo slanje je gotovo u trenutku 1, ali je odmah spreman i drugi paket tog toka te je potrebno za njega izračunati vrijeme završetka slanja u FFQ sustavu. Virtualno vrijeme u vrijeme dolaska paketa iznosi:

$$V(a_1^2) = V(1) = \frac{C}{\phi_1 + \phi_2 + \phi_3} = \frac{1}{0.5 + 0.2 + 0.2} = \frac{10}{9}$$

Sada se može izračunati vrijeme završetka slanja drugog paketa:

$$F_1^2 \leftarrow max\{V(1),2\} + \frac{1}{0.5} = 2 + 2 = 4$$

Od svih raspoloživih paketa u trenutku 1 (p_1^2, p_2^1, p_2^1) paket p_1^2 ima najmanje vrijeme završetka te se on prvi šalje. Nakon što je poslan, dolazi treći paket tog toka te se za njega proračunava vrijeme završetka slanja. Virtualno vrijeme u trenutku dolaska tog paketa iznosi:

$$V(a_1^3) = V(2) = V(0) + \frac{C}{\phi_1 + \phi_2 + \phi_3} = 2\frac{1}{0.5 + 0.2 + 0.2} = \frac{20}{9}$$

pa je prema tome vrijeme završetka:

$$F_1^3 \leftarrow max\{V(2),4\} + \frac{1}{0.5} = 4 + 2 = 6$$

Sada najmanje vrijeme završetka slanja imaju paketi 2. i 3. toka te se proizvoljno može odabrati bilo koji od njih. Da se razriješi ta situacija na jedinstven način koristiti će se pravilo da se šalju paketi veze koja ima manji indeks, što u ovom slučaju znači da se šalje paket 2. toka.

Nakon što je poslan paket p_2^1 dolazi još jedan paket prvog toka te se za njega izvračunava završno vrijeme. Virtualno vrijeme iznosi:

$$V(a_1^4)=V(2)=V(1)+\frac{C}{\phi_1+\phi_3}=\frac{20}{9}+\frac{1}{0.5+0.2}=\frac{320}{63}\sim 5.07$$

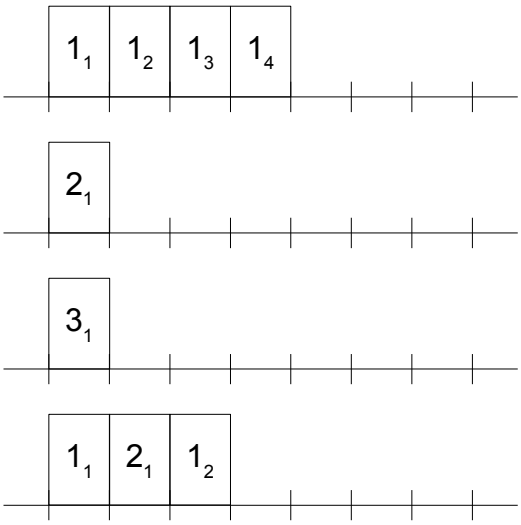
Treba primjetiti kako ovom prilikom nije uračunata težina 2. toka budući da on nema ni jedan paket za slanje! Sada se može izračunati završno vrijeme slanja paketa p_1^4:

$$F_1^4\leftarrow max\{V(3),6\}+\frac{1}{0.5}=6+2=8$$

Kako paket p_3^1 ima manje završno vrijeme on se prvi šalje. Nakon toga ostala su samo dva paketa prvog toka te se oni šalju i izvršavanje algoritma je za zadanu situaciju završeno.

7.4.3. WF²Q

Za razliku od WFQ algoritma, WF²Q algoritam uzima u obzir i vrijeme početka slanja paketa. Drugim riječima, kada se donosi odluka da li poslati određeni paket dodatno se provjerava i da li je vrijeme početka slanja tog paketa manje ili jednako trenutnom vremenu. Na slici 17 prikazan je isti primjer korišten u slučaju WFQ algoritma, slika 16.



Slika 17. Slanje upotrebom WF²Q algoritma

Izraz za završetak prijenosa paketa u referentnom FFQ sustavu korišten u WFQ algoritmu biti će zapisan u nešto drugačijem obliku. Završetak se računa kao vrijeme početka prijenosa plus trajanje samog prijenosa, tj.

$$F_{i,j}^k\leftarrow S_{i,j}^k+\frac{L_j^k}{\phi_{i,j}}$$

U tom izrazu $S_{i,j}^k$ je vrijeme početka prijenosa definirano sa:

$$S_{i,j}^k\leftarrow max\{V_i(a_{i,j}^k),F_{i,j}^{k-1}\}$$

Virtualno vrijeme u izrazu računa se na isti način kao i u WFQ algoritmu. Koristeći ta dva izraza računaju se vremena završetka prijenosa paketa p_1^1, p_2^1, p_3^1. Ta vremena je jednostavno izračunati budući da se za vrijeme početka prijenosa uzima 0. trenutak:

$$F_1^1 \leftarrow 0 + \frac{1}{0.5} = 0 + 2 = 2$$

$$F_2^1 = F_3^1 \leftarrow 0 + \frac{1}{0.2} = 5$$

Kao što se može primjetiti vrijednosti su identične WFQ sustavu, pa će prvi paket koji se šalje biti p_1^1. Nakon što je poslan taj paket u trenutku 1 pristiže paket p_1^2 za kojeg se računaju vremena početka prijenosa i završetka prijenosa u referentnom FFQ sustavu. Prvo je potrebno odrediti virtualno vrijeme u trenutku 0:

$$V(a_1^2) = V(1) = \frac{C}{\phi_1 + \phi_2 + \phi_3} = \frac{1}{0.5 + 0.2 + 0.2} = \frac{10}{9}$$

$$S_1^2 \leftarrow max\{V(1), 2\} = 2$$

$$F_1^2 \leftarrow 2 + \frac{1}{0.5} = 2 + 2 = 4$$

Ovdje nastupa prva razlika između WFQ i WF²Q. Iako p_1^2 ima ranije vrijeme završetka prijenosa njegovo početno vrijeme je u budućnosti, te se zbog toga prenosi paket p_2^1. Nakon što je prenesen taj paket dolazi paket p_1^3 za kojega treba izračunati početak i završetak prijenosa u referentnom FFQ sustavu. Opet se započinje sa virtualnim vremenom:

$$V(a_1^3) = V(2) = V(1) + \frac{C}{\phi_1 + \phi_3} = \frac{10}{9} + \frac{1}{0.5 + 0.2} = \frac{160}{63} \sim 2.53$$

$$S_1^3 \leftarrow max\{V(2), 4\} = 4$$

$$F_1^3 \leftarrow 4 + \frac{1}{0.5} = 4 + 2 = 6$$

U 2. vremenskoj jedinici moguće je poslati pakete p_1^2 i p_3^1 jer su njihova vremena početka prijenosa manja ili jednaka trenutnom vremenu. Od ta dva paketa p_1^2 ima ranije vrijeme završetka pa će on biti prenesen u 2. vremenskoj jedinici. U 3. vremenskoj jedinici pristiže paket p_1^4 i njegovo vrijeme početka i završetka prijenosa je:

$$V(a_1^3) = V(3) = V(2) + \frac{C}{\phi_1 + \phi_3} = \frac{160}{63} + \frac{1}{0.5 + 0.2} = \frac{250}{63} \sim 3.96$$

$$S_1^4 \leftarrow max\{V(3), 6\} = 6$$

$$F_1^4 \leftarrow 6 + \frac{1}{0.5} = 6 + 2 = 8$$

Prvi paket koji se može slati je p_1^3, a u idućoj vremenskoj jedinici je to p_1^4.

WFQ u svom radu neće nikada kasniti za više od jednog paketa, ali može ići dosta u naprijed. Za razliku od njega, WF²Q nikada ne kasni niti ne rani za više od jednog paketa u odnosu na referentni FFQ sustav. Problem oba algoritma je što su računski zahtijevni što nije dobro budući da se upotrebljavaju u situacijama gdje treba što manje trošiti na računanje kako bi paketi imali što manje kašnjenje. Iz tog razloga razvijeni su jednostavniji algoritmi.

7.4.4. Self-Clocked Fair Queuing

I WFQ i WF^{2}Q moraju tijekom svog rada emulirati referentni FFQ sustav što je računski dosta zahtijevno. Jedno od predloženih rješenja je i ovaj algoritam.

Temeljna ideja ovog algoritma je da se virtualno vrijeme može procijeniti koristeći informacije o paketu koji se trenutno šalje. Međutim, povećavajući jednostavnosti algoritma uvedena je i veća nepreciznost te su moguća znatna odstupanja od referentnog FFQ algoritma. Virtualno vrijeme definirano je na sljedeći način:

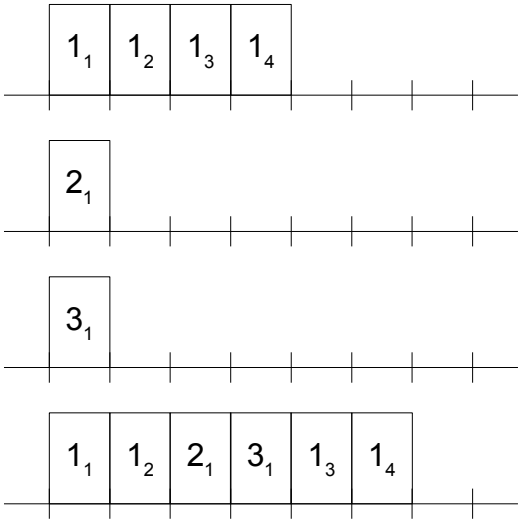
$$\hat{V}(t) = F^p \; za \; s^p < t \leqslant f^p$$

Riječima izrečeno, virtualno vrijeme jednako je završnom vremenu prijenosa paketa koji se trenutno šalje. Izrazi za početak slanja i kraj slanja ostaju slični kao i u WF^{2}Q i WFQ algoritmima:

$$F_{i,j}^k \leftarrow S_{i,j}^k + \frac{L_j^k}{\phi_{i,j}}$$

$$S_{i,j}^k \leftarrow max\{\hat{V}_i(a_{i,j}^k), F_{i,j}^{k-1}\}$$

Za demonstraciju izvršavanja algoritma može se uzeti primjer korišten za algoritme WFQ i WF^{2}Q.



Slika 18. Primjer raspodjele paketa upotrebom SCFQ algoritma

U početnom trenutku virtualno vrijeme ima vrijednost nula, a također su završna vremena prijenosa svih paketa prije prvog jednaka nuli. Zbog toga su vrijednosti početka i kraja prijenosa svih paketa isti kao u prethodnim algoritmima, tj.

$$F_1^1 \leftarrow 0 + \frac{1}{0.5} = 0 + 2 = 2$$

$$F_2^1 = F_3^1 \leftarrow 0 + \frac{1}{0.2} = 5$$

Prvi paket koji se šalje je p_1^1 budući da njegovo vrijeme završetka u referentnom FFQ sustavu dolazi prvo. Tijekom prijenosa tog paketa, i neposredno nakon završetka virtualno vrijeme ima vrijednost:

$$\hat{V} = F_1^1 = 2$$

Nakon što je završeno slanje tog paketa dolazi 2 paket tog istog toka, p_1^2, te se za njega izračunava vrijeme početka prijenosa i kraja prijenosa:

$$S_1^2 \leftarrow max\{\hat{V}(1), 2\} = 2$$

$$F_1^2 \leftarrow 2 + \frac{1}{0.5} = 2 + 2 = 4$$

Završno vrijeme tog paketa je 4 što je ujedno i najmanje završno vrijeme pa SCFQ bira taj paket za slanje. Tijekom slanja tog paketa virtualno vrijeme iznosi 4, te se nakon prijenosa računa završno vrijeme za 3. paket toka 1, tj. za p_1^3:

$$S_1^3 \leftarrow max\{\hat{V}(2), 4\} = 4$$

$$F_1^3 \leftarrow 4 + \frac{1}{0.5} = 4 + 2 = 6$$

Sada, budući da p_2^1 i p_3^1 imaju ranije vrijeme završetka prijenosa oni se prvi šalju. Koristeći pravilo da paket sa manjim indeksom toka ide prvi raspoređivač šalje prvo p_2^1.Tijekom slanja tog paketa virtualno vrijeme je postavljeno na vrijednost:

$$\hat{V} = F_2^1 = 5$$

U trenutku kada dođe zadnji paket prve veze potrebno je izračunati vrijeme prijenosa za njega:

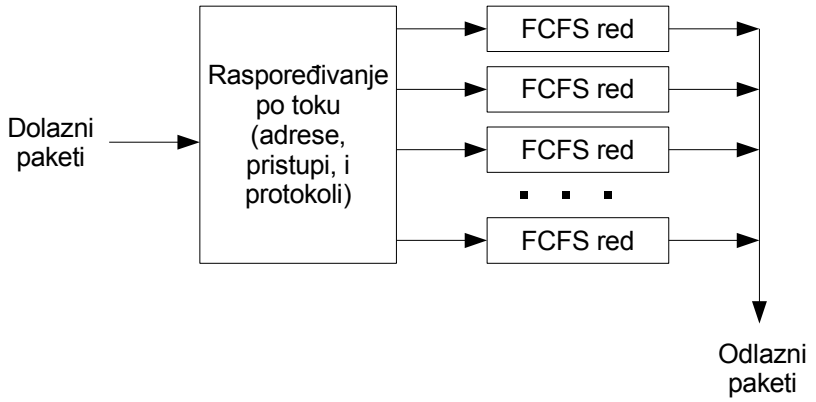
$$S_1^4 \leftarrow max\{\hat{V}(3), 6\} = 6$$

$$F_1^4 \leftarrow 6 + \frac{1}{0.5} = 6 + 2 = 8$$

U idućem trenutku šalje se paket p_3^1, a nakon njega i dva preostala paketa prve veze budući da nema novih dolazaka paketa.

7.4.5. Stochastic Fair Queuing

Svi prethodni algoritmi, WFQ, WF²Q, SCFQ su uprkos raznim poboljšanjima i pojednostavljenjima računski dosta zahtijevni. Da bi se shvatila osnovna ideja SFQ algoritma [5] potrebno je prvo razumijeti osnovu FFQ poslužitelja. Shematski, poslužitelj je prikazan na slici 19.



Slika 19. Arhitektura FFQ poslužitelja

U osnovi u FFQ poslužitelju se paketi različitih tokova međusobno razdvajaju u zasebne redove. To razdvajanje obavlja se na osnovu odredišne i izvorišne adrese, pristupa na odredištu i na izvorištu te na osnovu protokola. Redovi u koje su smješteni tokovi, po jedan red za svaki tok, kružno se

poslužuju bit po bit (*round robin*). Raspoređivanje u redove je u vrlo brzim mrežnim uređajima dosta teško izvesti jer je vrlo malo vremena na raspolaganju za obradu svakog paketa. To još više dolazi do izražaja sa porašću broja tokova. Za svaki tok je potrebno voditi evidenciju, kada se pojavi potrebno je smjestiti podatke o toku u memoriju, a nakon nekog vremena – ako se ne pojave novi paketi – izbrisati podatke o toku iz tablica. Ideja SFQ algoritma je da se koristi manji broj redova od broja tokova te se potom koriste funkcije raspršivanja (*hash*) koje raspoređuju svaki tok u neki od tih redova. Ako dva toka dođu u isti red tada će morati dijeliti izlaznu liniju što može dovesti do nepravednosti. Kako bi se ublažio taj problem funkcija raspršivanja se periodički mijenja te se na taj način tokovi raspoređuju u različite redove.

Ovaj algoritam implementiran je u Linux operacijskom sustavu za raspoređivanje paketa, a također se razvija inačica koja bi radila posluživanje zahtijeva za pristupom čvrstim diskovima.

Matematička analiza ovog algoritma oslanja se na analizu funkcija raspršivanja te kao odgovor daje broj tokova sa kojima neki tok može očekivati dijeljenje reda. Očekivani broj tokova u jednom redu može se prikazati na sljedeći način:

$$EC = \alpha + 1$$

dok se varijanca očekivanog broja tokova računa na sljedeći način:

$$VC = \frac{\alpha^2}{6} + \alpha$$

Parametar α je omjer broja tokova i broja redova. Primjerice, ako su redovi prazni tada je parametar α nula, te je očekivani broj tokova sa kojima će dijeliti red 1 i varijanca je 0. To drugim riječima znači da tok sigurno dobija svoj red koji ne dijeli sa ni sa kim. Ako sa druge strane SFQ poslužitelj sadrži broj tokova jednak broju redova tada je parametar α=1, očekivani broj tokova u redu koji dobija novi tok EC = 2, a varijanca je 1.17.

7.4.6. Start-time Fair Queuing

Prednosti ovog algoritma u odnosu na prethodne su (*i*) izračunski efikasan, i (*ii*) pravična raspodjela bez obzira na varijacije u propusnosti poslužitelja. Kompletan algoritam sastoji se od sljedećih koraka:

1. Po nailasku *j*-tog paket toka F, p_f^j, označava se s početnom oznakom, $S(p_f^j)$, koja se izračunava na sljedeći način:

   $$S(p_f^j) = max\{v(A(p_f^j)), F(p_f^{j-1})\}$$

   pri čemu je $F(p_f^j)$ završno vrijeme paketa p_f^j definirano na sljedeći način:

   $$F(p_f^j) = S(p_f^j) + \frac{l_f^j}{r_f}; \qquad j \geq 1$$

   i $F(p_f^0) = 0$, a r_f je težina toka *f*.

2. Na početku je virtualno vrijeme poslužitelja 0. Tijekom perioda rada u trenutku t, virtualno vrijeme v(t) jednako je početnoj oznaci paketa koji se poslužuje u trenutku t. Na kraju perioda rada, v(t) se postavlja na maksimalnu završnu oznaku koju je imao bilo koji posluživani paket do trenutka t.
3. Paketi se poslužuju po rastućim početnim oznakama; u slučaju istih početnih oznaka proizvoljno se bira paket.

Prednosti ovog algoritma su sljedeće:

- U odnosu na WFQ, SFQ daje manje srednje vrijeme kašnjenja po paketu i ukupno manje maksimalno vrijeme kašnjenja svakgo paketa.
- Postiže veću *pravednost* u odnosu na WFQ kod poslužitelja promijenjivog kapaciteta.
- Manja je maksimalna vrijednost kašnjenja u odnosu na SCFQ, te oba imaju istu mjeru pravednosti i jednostavnost implementacije.
- Bolji je od FQS jer je manje kompleksnosti i postiže pravednost kod poslužitelja različita kapaciteta bez povećanja maksimalnog kašnjenja paketa.

Način rada algoritma može se demonstrirati upotrebom tokova prikazanih na slici 18. Tok 1 ima najmanji prioritet, 0.2, potom tok 2 ima prioritet 0.3 i konačno, tok 3 ima najviši prioritet 0.5. U nultom trenutku do poslužitelja dolaze prvi paketi svih tokova te on za njih računa početna vremena i završna vremena prijenosa:

$$S(p_1^1) = max\{v(A(p_1^1)), F(p_1^0)\} = max\{v(0), 0\} = 0$$

$$F(p_1^1) = S(p_1^1) + \frac{l_1^1}{r_1} = 0 + \frac{1}{0.2} = 5$$

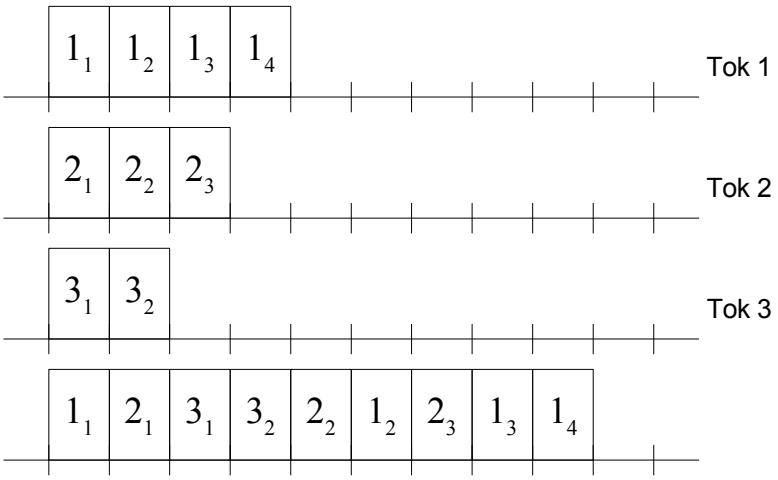
$$S(p_2^1) = max\{v(A(p_2^1)), F(p_2^0)\} = max\{v(0), 0\} = 0$$

$$F(p_2^1) = S(p_2^1) + \frac{l_2^1}{r_2} = 0 + \frac{1}{0.3} = \frac{10}{3}$$

$$S(p_3^1) = max\{v(A(p_3^1)), F(p_3^0)\} = max\{v(0), 0\} = 0$$

$$F(p_3^1) = S(p_3^1) + \frac{l_3^1}{r_3} = 0 + \frac{1}{0.5} = 2$$

Budući da svi paket imaju isto početno vrijeme prijenosa proizvoljno se odabire prvi paket, ali primjernom načela da se u takvim slučajevima odabire tok sa najnižim indeksom, prvi paket koji se šalje je p_1^1. Osim načela najmanjeg indeksa moguće je primjeniti i pravilo po kojemu se paket toka najvećeg prioriteta prvi šalje.



Slika 20. Primjer raspoređivanja korištenjem SFQ raspoređivača

Nakon što je poslan prvi paket dolaze tri nova paketa, po jedan od svakog toka te je potrebno ponovo preračunavati početna i završna vremena slanja. Virtualno vrijeme ostaje nula budući da je početno vrijeme paketa koji je poslan bilo nula. Računanjem se dobijaju sljedeća vremena:

$$S(p_1^2) = max\{v(A(p_1^2)), F(p_1^1)\} = max\{v(1), 5\} = max\{0, 5\} = 5$$

$$F(p_1^2) = S(p_1^2) + \frac{l_1^2}{r_1} = 5 + \frac{1}{0.2} = 10$$

$$S(p_2^2) = max\{v(A(p_2^2)), F(p_2^1)\} = max\{v(1), \frac{10}{3}\} = \frac{10}{3}$$

$$F(p_2^2) = S(p_2^2) + \frac{l_2^2}{r_2} = \frac{10}{3} + \frac{1}{0.3} = \frac{20}{3}$$

$$S(p_3^2) = max\{v(A(p_3^2)), F(p_3^1)\} = max\{v(1), 2\} = 2$$

$$F(p_3^2) = S(p_3^2) + \frac{l_3^2}{r_3} = 2 + \frac{1}{0.5} = 4$$

Od svih paketa u sustavu najmanje početno vrijeme prijenosa imaju paketi p_2^1 i p_3^1 pa se po već navedenom načelu prenosi paket p_2^1. Nakon što je taj paket prenesen virtualno vrijeme je još uvijek nula, ali u sustav pristižu dva nova paketa, p_1^3 i p_2^3. Za te pakete također je potrebno odrediti početna i završna vremena prijenosa:

$$S(p_1^3) = max\{v(A(p_1^3)), F(p_1^2)\} = max\{v(2), 10\} = max\{0, 10\} = 10$$

$$F(p_1^3) = S(p_1^3) + \frac{l_1^3}{r_1} = 10 + \frac{1}{0.2} = 15$$

$$S(p_2^3) = max\{v(A(p_2^3)), F(p_2^2)\} = max\{v(2), \frac{20}{3}\} = \frac{20}{3}$$

$$F(p_2^3) = S(p_2^3) + \frac{l_2^3}{r_2} = \frac{20}{3} + \frac{1}{0.3} = 10$$

Nakon tog računanja odlučuje se o paketu koji će biti poslan. Najmanje početno vrijeme slanja ima paket p_3^1, te se on šalje. Nakon slanja tog paketa u sustav dolazi još jedan novi paket, p_1^4, te se za njega računaju početna i završna vremena slanja. Virtualno vrijeme je još uvijek nula budući da je početno vrijeme slanja zadnjeg paketa bilo 0:

$$S(p_1^4) = max\{v(A(p_1^4)), F(p_1^3)\} = max\{v(3), 15\} = max\{0, 15\} = 15$$

$$F(p_1^4) = S(p_1^4) + \frac{l_1^4}{r_1} = 15 + \frac{1}{0.2} = 20$$

Od tog trenutka na dalje novi paketi ne pristižu u sustav te se samo šalju drugi. Tijekom tog slanja povećava se virtualno vrijeme u skladu sa navedenim algoritmom. Dakle, prvo se šalje paket p_3^2 i tijekom njegova slanja virtualno vrijeme je 2. Nakon toga šalje se paket p_2^2 i tijekom njegova slanja virtualno vrijeme je 3.33. Onda se šalje p_1^2 i virtualno vrijeme je 5. Taj postupak traje dok se ne pošalju svi paketi. Cijeli postupak može se prikazati i tablično, kako je to prikazano u tablici 6.

Tablica 6. Tablični prikaz izvođenja SFQ algoritma

	p_1^1	p_2^1	p_3^1	p_1^2	p_2^2	p_3^2	p_1^3	p_2^3	p_1^4	
S	0	0	0	5	10/3	2	10	20/3	15	
F	5	10/3	2	10	20/3	4	15	10	20	
n	1	2	3	6	5	4	8	7	9	
v	0	0	0	5	10/3	2	10	20/3	15	20

U toj tablici redak sa oznakom S sadrži početna vremena prijenosa svih paketa, F označava kraj prijenosa, n označava redni broj paketa u prijenosu i konačno v označava virutalno vrijeme. Tablica se tijekom izvršavanja algoritma proširuje na desno pri čemu se u zaglavlja stupaca upišu pristigli paketi, a potom se izračunavaju S i F. Na osnovu S određuje se redni broj paketa, a paket u prijenosu određuje i vrijednost virutalnog vremena.

8. Sigurnost na mrežnom sloju

8.1. IPSec

RFC2207, RFC2410, RFC2709, RFC3104, RFC3193, RFC3456, RFC3457, RFC3554, RFC3566, RFC3585, RFC3602, RFC3686, RFC3715, RFC3776, RFC3884,

8.2. VPN

8.3. Firewall

9. Primjeri

PRIMJER 1. IP adrese

Za zadane mreže odrediti alternativne zapise, difuzne adrese i maksimalan broj uređaja u mreži.

a) 10.0.0.0/24

b) 192.168.0.0/255.255.0.0

c) 172.16.128.0/17

d) 192.168.255.252/255.255.255.252

Rješenje

PRIMJER 2. Proces proslijeđivanja

Za zadanu tablicu proslijeđivanja:

Odredišna mreža	*Idući čvor*	*Izlazno sučelje*
161.53.65.0/24		eth0
161.53.0.0/16	161.53.65.1	eth0
192.168.2.0/24		eth1
192.168.2.0/30	192.168.2.150	eth1

odredite kamo će biti proslijeđeni slijedeći paketi:

e) IP(161.53.65.15, 161.53.65.14)

f) IP(161.53.17.16, 161.53.65.18)

g) IP(161.54.8.8, 192.168.2.117)

h) IP(192.168.2.1, 192.168.2.117)

Rješenje

Kao što je rečeno u uvodnom dijelu, pretraživanje u tablici prosljeđivanja uvijek ide od zapisa koji imaju najdulju mrežnu masku prema zapisu s najkraćom mrežnom maskom. Na prvo preklapanje se prestaje s pretraživanjem!

Isključivo radi lakšeg rješavanja zadatka, tablicu proslijeđivanja ćemo prikazati ponovo, ali ovaj puta poredanu po opadajućoj duljini mrežne maske.

Odredišna mreža	*Idući čvor*	*Izlazno sučelje*
192.168.2.0/30	192.168.2.150	eth1
192.168.2.0/24		eth1
161.53.65.0/24		eth0
161.53.0.0/16	161.53.65.1	eth0

a) IP(161.53.65.15, 161.53.65.14)

U ovom slučaju odredišna IP adresa 161.53.65.15 se provjerava slijedno s odredišnom mrežom svakog zapisa u tablici. Koristeći izraz dan u uvodu za usporedbu IP adrese s mrežom, za prvi redak imamo sljedeći slučaj:

$$IP_{net} == IP_{dest} \,\&\, ((2^n - 1) << (32 - n))$$

$$192.168.2.0 == 161.53.65.15 \,\&\, ((2^{30} - 1) << (32 - 30))$$

$$192.168.2.0 == 161.53.65.15 \,\&\, 0xFFFFFFFC$$

$$192.168.2.0 == 161.53.65.12$$

Prema tome rezultat izračunavanja tog izraza nije točan te se prelazi na drugi redak tablice. Za drugi redak tablice imamo sljedeći slučaj:

$$IP_{net} == IP_{dest} \,\&\, ((2^n - 1) << (32 - n))$$

$$192.168.2.0 == 161.53.65.15 \,\&\, ((2^{24} - 1) << (32 - 24))$$

$$192.168.2.0 == 161.53.65.15 \;\&\; 0xFFFFFF00$$

$$192.168.2.0 == 161.53.65.0$$

Opet ne postoji preklapanje te se pretraživanje nastavlja s trećim retkom tablice:

$$IP_{net} == IP_{dest} \;\&\; ((2^n - 1) << (32 - n))$$

$$161.53.65.0 == 161.53.65.15 \;\&\; ((2^{24} - 1) << (32 - 24))$$

$$161.53.65.0 == 161.53.65.15 \;\&\; 0xFFFFFF00$$

$$161.53.65.0 == 161.53.65.0$$

Ovaj puta izraz je točan. Kako je drugi stupac tog retka prazan usmjernik zaključuje kako se odredišno računalo nalazi direktno priključeno na njegovo sučelje eth0 te mu može proslijediti paket direktno.

b) IP(161.53.17.16, 161.53.65.18)

I za ovaj paket usporedba kreće od prvog retka tablice (retka koji ima najdulju mrežnu adresu) te se postupno nastavlja prema sve kraćim mrežim adresama. Za prvi redak tablice imamo sljedeću vrijednost usporedbe:

$$IP_{net} == IP_{dest} \;\&\; ((2^n - 1) << (32 - n))$$

$$192.168.2.0 == 161.53.17.16 \;\&\; ((2^{30} - 1) << (32 - 30))$$

$$192.168.2.0 == 161.53.65.16 \;\&\; 0xFFFFFFFC$$

$$192.168.2.0 == 161.53.65.16$$

Prvi redak tablice očito ne odgovara, pa se nastavlja s drugim retkom:

$$IP_{net} == IP_{dest} \;\&\; ((2^n - 1) << (32 - n))$$

$$192.168.2.0 == 161.53.17.16 \;\&\; ((2^{24} - 1) << (32 - 24))$$

$$192.168.2.0 == 161.53.17.16 \;\&\; 0xFFFFFF00$$

$$192.168.2.0 == 161.53.17.0$$

Ni drugi redak ne odgovara. Treći redak:

$$IP_{net} == IP_{dest} \;\&\; ((2^n - 1) << (32 - n))$$

$$161.53.65.0 == 161.53.17.16 \;\&\; ((2^{24} - 1) << (32 - 24))$$

$$161.53.65.0 == 161.53.17.16 \;\&\; 0xFFFFFF00$$

$$161.53.65.0 == 161.53.17.0$$

Ni treći redak ne odgovara. Četvrti redak:

$$IP_{net} == IP_{dest} \;\&\; ((2^n - 1) << (32 - n))$$

$$161.53.0.0 == 161.53.17.16 \;\&\; ((2^{16} - 1) << (32 - 16))$$

$$161.53.0.0 == 161.53.17.16 \;\&\; 0xFFFF0000$$

$$161.53.0.0 == 161.53.0.0$$

Četvrti redak odgovara pa će se koristiti za odluku o daljnjoj sudbini paketa. Budući da drugi stupac nije prazan, već se u njemu nalazi IP adresa to znači da paket treba proslijediti na tu IP adresu preko sučelja eth0.

c) IP(161.54.8.8, 192.168.2.117)

Opet se ponavlja kompletan postupak usporedbe. Prvi redak:

$$IP_{net} == IP_{dest} \,\&\, ((2^n - 1) << (32 - n))$$
$$192.168.2.0 == 161.54.8.8 \,\&\, ((2^{30} - 1) << (32 - 30))$$
$$192.168.2.0 == 161.54.8.8 \,\&\, 0xFFFFFFFC$$
$$192.168.2.0 == 161.54.8.8$$

Dakle, prvi redak ne odgovara. Drugi redak:

$$IP_{net} == IP_{dest} \,\&\, ((2^n - 1) << (32 - n))$$
$$192.168.2.0 == 161.54.8.8 \,\&\, ((2^{24} - 1) << (32 - 24))$$
$$192.168.2.0 == 161.54.8.8 \,\&\, 0xFFFFFF00$$
$$192.168.2.0 == 161.54.8.0$$

Ni drugi redak ne odgovara. Treći redak:

$$IP_{net} == IP_{dest} \,\&\, ((2^n - 1) << (32 - n))$$
$$161.53.65.0 == 161.54.8.8 \,\&\, ((2^{24} - 1) << (32 - 24))$$
$$161.53.65.0 == 161.54.8.8 \,\&\, 0xFFFFFF00$$
$$161.53.65.0 == 161.54.8.0$$

Četvrti redak:

$$IP_{net} == IP_{dest} \,\&\, ((2^n - 1) << (32 - n))$$
$$161.53.0.0 == 161.54.8.8 \,\&\, ((2^{16} - 1) << (32 - 16))$$
$$161.53.0.0 == 161.54.8.8 \,\&\, 0xFFFF0000$$
$$161.53.0.0 == 161.54.0.0$$

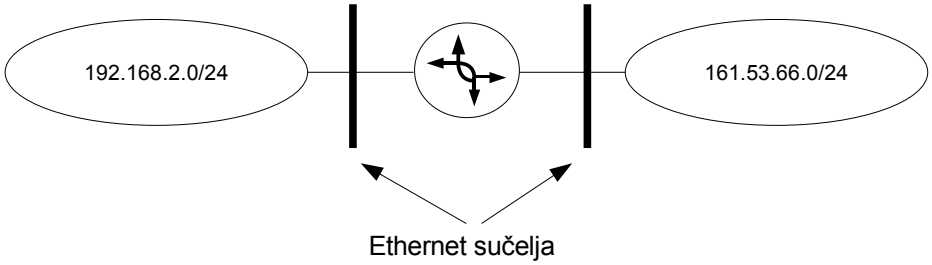
Nakon pregleda kompletne tablice usmjernik ustanovljava kako niti jedan redak ne odgovara. U tom trenutku on odbacuje paket i eventualno šalje nazad upozorenje o nepoznatoj mreži!

d) IP(192.168.2.1, 192.168.2.117)

U ovom slučaju, nakon pretraživanja analognog pretraživanju u prethodnim podzadacima, usmjernik će proslijediti ovaj paket čvoru s IP adresom 192.168.2.150 koji se nalazi priključen na sučelje eth0.

PRIMJER 3. ***Izrada tablica proslijeđivanja usmjernika***

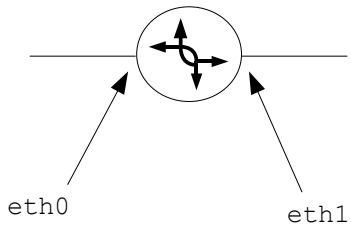
Za mrežnu konfiguraciju prikazanu na slici složiti odgovarajuću tablicu usmjeravanja usmjernika. Na mjestima gdje je to potrebno odabrati odgovarajuće IP adrese i imena sučelja poštivajući uobičajena pravila.



Rješenje

Prvi korak prilikom slaganja tablica usmjeravanja za neku mrežnu konfiguraciju je definiranje imena svih sučelja i IP adresa. Sve to potrebno je obaviti iz perspektive samog usmjernika za kojega ćemo slagati tablice usmjeravanja.

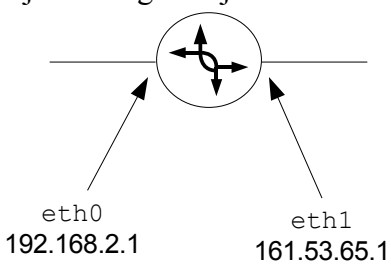
U ovom zadatku javljaju se isključivo Ethernet mreže. Sučelja koja se spajaju na Ethernet mrežu uvijek imaju imena oblika `eth`*`n`* pri čemu se slovo n zamjenjuje s brojem koji počinje od 0. U slučaju ovog zadatka usmjernik je priključen na dvije ethernet mreže te zbog toga mora imati dva ethernet sučelja. Ta sučelja će biti označena s `eth0` i `eth1`. Koje sučelje će dobiti oznaku `eth0`, a koje `eth1` u ovakvim slučajevima nije bitna. Nakon označavanja sučelja nova situacija je prikazana na sljedećoj slici:



Na toj slici nisu više prikazane mreže na koje je spojen usmjernik, ali se one podrazumijevaju. Nakon što su definirana imena sučeljima potrebno im je dodijeliti IP adrese. IP adrese koje će biti dodijeljene sučeljima moraju se odabrati s mreže na koju je sučelje spojeno.

Uzmimo na primjer sučelje `eth0`. To sučelje spaja usmjernik na mrežu čija mrežna IP adresa je 192.168.2.0/24, prema tome IP adresa koju će dobiti to sučelje mora biti iz tog raspona. Običaj je da se sučeljima usmjernika na lokalnim mrežama pokuša dati što niži broj. U ovom slučaju će odabrana IP adresa biti 192.168.2.1. Analognim postupkom će sučelje `eth1` dobiti IP adresu 161.53.66.1.

Dakle, usmjernik s označenim sučeljima i odgovarajućim IP adresama prikazan je na sljedećoj slici:



Nakon što su označena sva sučelja i postavljene sve potrebne IP adrese pristupa se izradi tablice usmjeravanja.

Postupak za generiranje tablice je sljedeći:

- Za svaku mrežu koja postoji na zadanoj slici napravi se jedan redak tablice.
- U prvi stupac piše se adresa mreže.
- Potrebno je provjeriti kroz koje sučelje se stiže do te mreže. To sučelje se piše u treći stupac tablice.

U ovom konkretnom slučaju postoje dvije mreže (192.168.2.0/24 i 161.53.65.0/24) te će tablica proslijeđivanja imati dva retka. Mreža 192.168.2.0/24 dostupna je preko sučelja `eth0` pa se prema tome paket proslijeđuje na to sučelje i u tablici se za tu mrežu u treći stupac piše `eth0`. Slično se za drugu mrežu u treći stupac piše `eth1` budući da je ta mreža dostupna preko tog sučelja!

Prema tome, tablica proslijeđivanja ima sljedeći sadržaj:

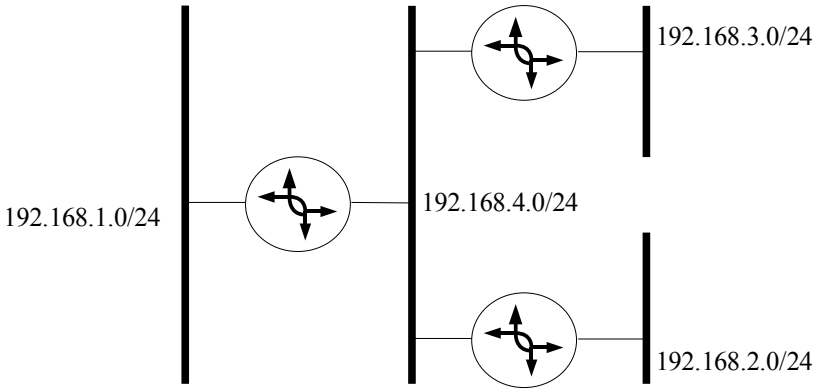
Odredišna mreža	***Idući čvor***	***Izlazno sučelje***
192.168.2.0/24		eth0
161.53.65.0/24		eth1

Napomena

Prilikom realizacije računalne mreže ne treba posebno postavljati tablicu usmjeravanja za direktno spojene mreže. Tijekom dodjele IP adrese sučelju i aktivacije tog sučelja Linux kernel automatski postavlja navedene rute! U nastavku, kao i u ovom zadatku, će se ignorirati ta činjenica, no ona ne mijenja ništa na ispravnosti dobijenih tablica!

PRIMJER 4. ***Izrada tablica proslijeđivanja usmjernika***

Odrediti sve tablice proslijeđivanja konfiguracije usmjernika prikazane na slici.



Rješenje

Kako bi se mogle pisati tablice proslijeđivanja usmjernika potrebno je odrediti nazive sučelja usmjernika i njihove IP adrese. Na prikazanoj slici sva sučelja su Ethernet

PRIMJER 5. ***Promet na jednostavnoj međumreži (internetwork)***

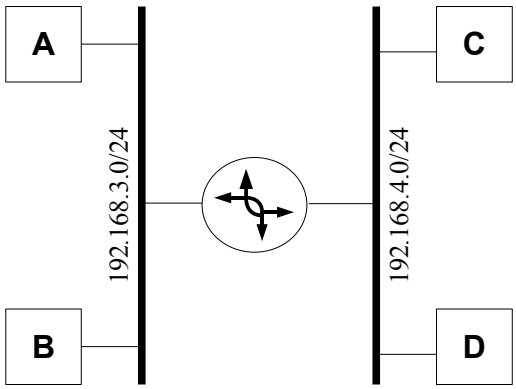
Za lokalnu mrežu prikazanu na slici odrediti tablice proslijeđivanja svih uređaja. Prikazati promet IP i ARP paketa na mreži. Detaljno navesti način upotrebe tablica proslijeđivanja za sljedeće slučajeve:

a) Računalo A šalje IP paket računalu B

b) Računalo A šalje IP paket računalu D

c) Računalo A šalje IP paket računalu C

d) Računalo B šalje IP paket računalu C



Rješenje

Da bi se zadatak mogao riješiti nužno je znati IP i Ethernet adrese svih sučelja. Kako te adrese nisu zadane potrebno ih je odrediti. Adrese nije moguće u potpunosti odabrati proizvoljno. Mrežni dio je predodređen i zadan na slici. Primjerice, lijeva mreža ima adresu 192.168.3.0/24. To znači da svako sučelje spojeno na lijevu mrežu mora imati IP adresu s identičnom mrežnom adresom. Usmjernik ima dva mrežna sučelja. Po jedno na lijevoj i desnoj mreži. U sljedećoj tablici navedene su odabrane adrese.

Uređaj	*Ethernet adresa*	*IP adresa*
A	a_e	192.168.3.2
B	b_e	192.168.3.3
C	c_e	192.168.4.2
D	d_e	192.168.4.1
Usmjernik, lijevo sučelje	l_e	192.168.3.1
Usmjernik, desno sučelje	r_e	192.168.3.2

Svako od računala na lijevoj mreži ima sljedeću tablicu usmjeravanja:

Mreža	*Sljedeći čvor*	*Izlazno sučelje*
192.168.3.0/24		eth0
default	192.168.3.1	eth0

Sličnu tablicu sadrže i računala koja se nalaze na desnoj mreži:

Mreža	*Sljedeći čvor*	*Izlazno sučelje*
192.168.3.0/24		eth0
default	192.168.3.1	eth0

Po uvijetima zadatka računalo A treba poslati paket računalu B, tj. sljedeći IP paket

IP (192.168.3.3, 192.168.4.2)

PRIMJER 6. P*romet na mreži generiran upotrebom naredbe `ping`*

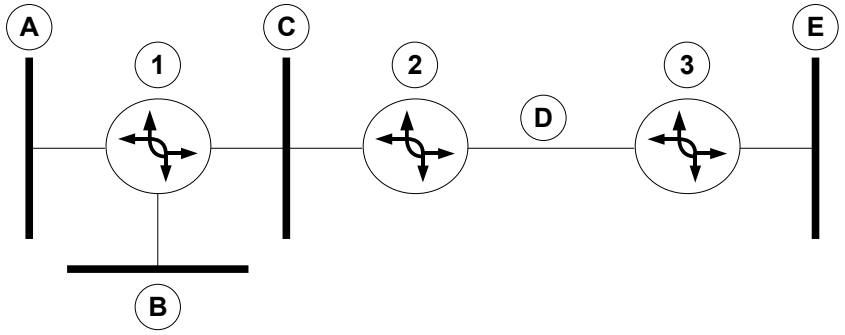
Pretpostavite da se na lokalnoj mreži nalaze tri računala: A, B i C. Svako od tih računala ima po jedno Ethernet sučelje čije su MAC adrese a_e, b_e i c_e a IP adrese a_{IP}, b_{IP} i c_{IP}. Prikazati promet **samo** ARP paketa za sljedeći niz događaja: na računalu A izvršava se naredba `ping` b_{IP}. Neposredno nakon toga na računalu C izvršava se naredba `ping` b_{IP}. Koristiti notaciju s auditornih vježbi!

Rješenje

Eth(BROADCAST, a_e, ARPReq(b_{IP}, a_{IP}))
Eth(a_e, b_e, ARPReq(b_{IP}, a_{IP}))
Eth(BROADCAST, c_e, ARPReq(b_{IP}, c_{IP}))
Eth(c_e, b_e, ARPReq(c_{IP}, a_{IP}))

PRIMJER 7. ***Podjela IP mreže na manje cjeline (subnetting)***

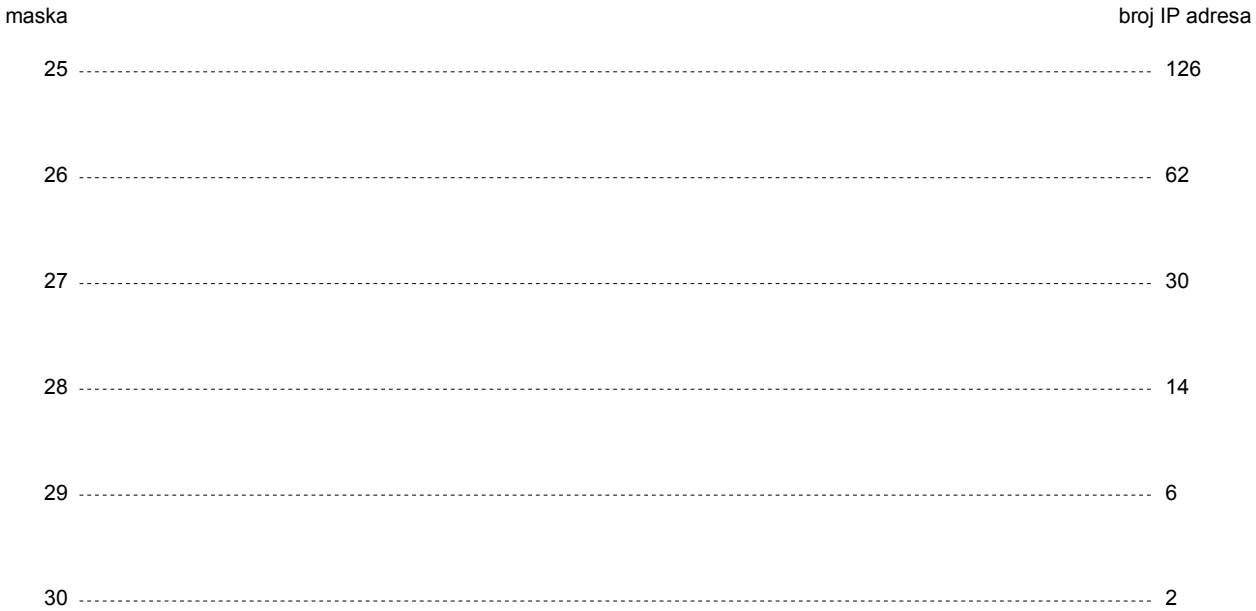
Za mrežnu konfiguraciju prikazanu na slici dodijeljen je blok IP adresa 192.168.0.0/25. Potrebno je taj blok optimalno raspodijeliti na mreže u zadanoj mrežnoj konfiguraciji. Koliki je maksimalan broj računala u danom bloku IP adresa? Koliko IP adresa je moguće imati nakon podjele zadanog bloka na podmreže? Pretpostavite da je maksimalan planirani broj računala na A mreži 16, na B mreži 12, na C mreži 10, i na E mreži 5. Ti brojevi ne uključuju i sučelja usmjernika!



Rješenje

Budući da planirani broj računala po mreži ne uključuje sučelje usmjernika to znači da je svaki od njih potrebno uvećati za 1. D mreža je posebna. To je serijska veza koja povezuje isključivo 2 usmjernika i zbog toga za tu mrežu nije naveden maksimalan planirani broj računala.

Raspodjela IP adresa vršiti će se korištenjem dijagrama prikazanog na sljedećoj slici:

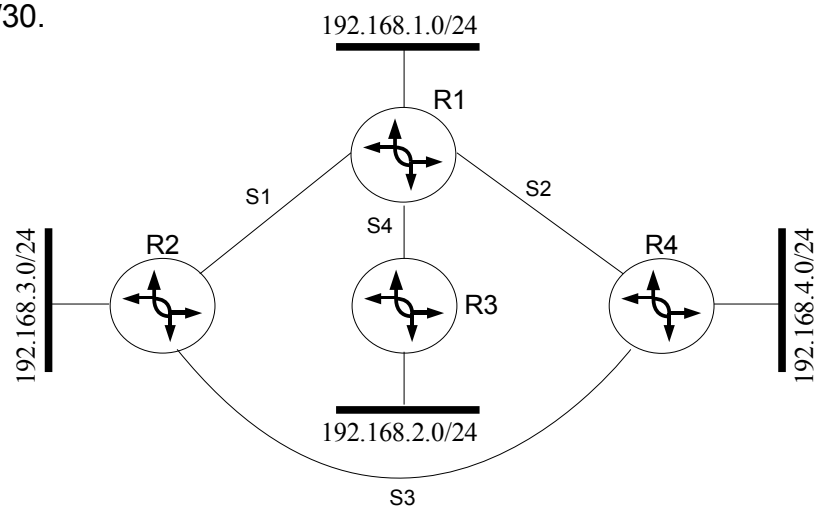


Na tom dijagramu broj horizontalnih linija odgovara broju bitova računalnog dijela koji nam je ostavljen na raspolaganju. Sa lijeve strane označeni su indeksi bitova računalnog dijela. Indeksiranje bitova kreće od 0. i to je najteži bit 32-bitne IP adrese koji se nalazi sa lijeve strane. U ovom zadatku mrežna maska ima 25 bitova. Budući da zbog toga bitovi od 0 do 24 pripadaju mrežnom dijelu dijagram se crta od 25. bita. Crtanje završava sa 30. bitom budući da je to najveća mrežna maska koja se može pojaviti.

Sa desne strane naveden je broj raspoloživih IP adresa koji je moguće ostvariti za zadanu veličinu maske. Kao primjer možemo uzeti odmah prvu liniju. U tom slučaju 25 bitova pripada mrežnoj maski, a ostalih 32 – 25 = 7 pripada računalnom dijelu. To znači da je moguće imati najviše $2^7 – 2$, tj. 126 IP adresa. Ako za mrežnu masku uzmemo 30 bitova, to znači da računalnom dijelu priprada 32 – 30 = 2 bita, te je moguće imati najviše $2^2 – 2$ računala. Broj 2 se oduzima zbog dvije rezervirane IP adrese. To su adrese koje u računalnom dijelu imaju sve nule i sve jedinice, kako je to objašnjeno u uvodnom dijelu.

PRIMJER 8. ***Izrada tablica usmjeravanja za mreže sa serijskim vezama***

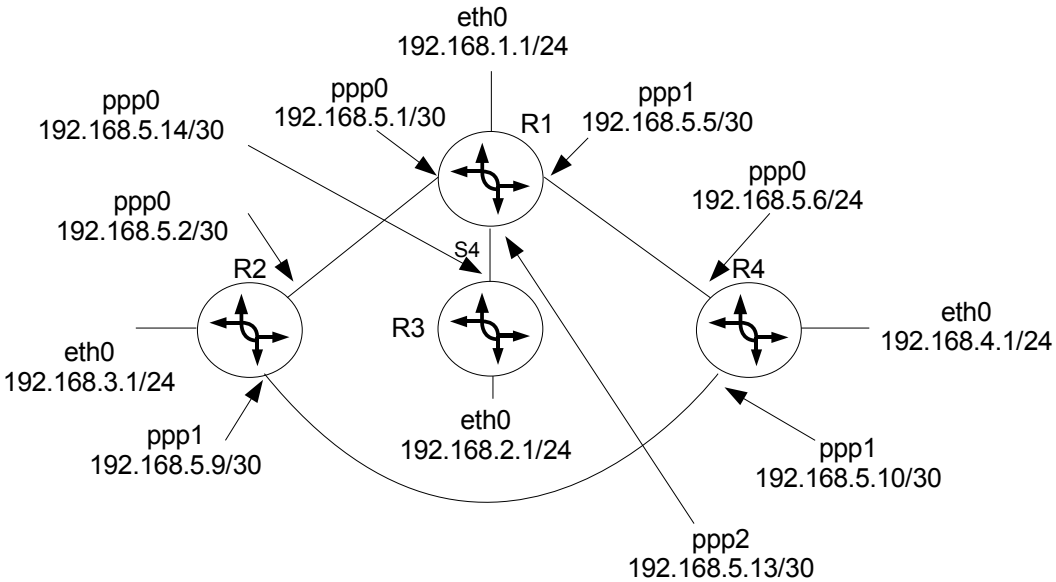
Usmjernicima prikazanim na slici potrebno je složiti tablice proslijeđivanja. Pretpostaviti da serijka sučelja nisu globalno dostupna. Tablice proslijeđivanja moraju biti optimalne s obzirom na broj usmjernika koje paketi prolaze do odredišta. Mrežne adrese dodijeljene serijskim sučeljima su sljedeće: S1:192.168.5.0/30, S2: 192.168.5.4/30, S3: 192.168.5.8/30 i S4: 192.168.12/30.



Rješenje

U zadatku je zadan uvijet da serijska sučelja nisu globalno dostupna. To znači da IP adresama pojedinih serijskih sučelja mogu pristupiti isključivo usmjernici na koje je serijsko sučelje spojeno. Primjerice, serijskom sučelju S1, čija mrežna adresa je 192.168.5.0/30, mogu pristupiti isključivo usmjernici R1 i R2, dok usmjernici R3 i R4 nemaju u svojim tablicama usmjeravanja te mreže i zbog toga im ne mogu pristupiti.

Radi lakšeg pisanja tablica proslijeđivanja svih usmjernika preporučljivo je prvo na zadanoj slici naznačiti imena sučelja svih usmjernika i njihove pripadne IP adrese. To je učinjeno na sljedećoj slici:



Slika 21. Usmjernici iz zadatka sa označenim sučeljima i odgovarajućim IP adresama

Tablica 7. Tablica proslijeđivanja usmjernika R1

Mreža	***Sljedeći čvor***	***Izlazno sučelje***
192.168.1.0/24		eth0
192.168.5.0/30		ppp0
192.168.5.4/30		ppp1
192.168.5.12/30		ppp2
192.168.2.0/24	192.168.5.14	ppp2
192.168.3.0/24	192.168.5.2	ppp0
192.168.4.0/24	192.168.5.6	ppp1

Tablica 8. Tablica proslijeđivanja usmjernika R2

Mreža	***Sljedeći čvor***	***Izlazno sučelje***
192.168.3.0/24		eth0
192.168.5.0/30		ppp0
192.168.5.8/30		ppp1
192.168.1.0/24	192.168.5.1	ppp0
192.168.2.0/24	192.168.5.1	ppp0
192.168.4.0/24	192.168.5.10	ppp1

Tablica 9. Tablica proslijeđivanja usmjernika R3

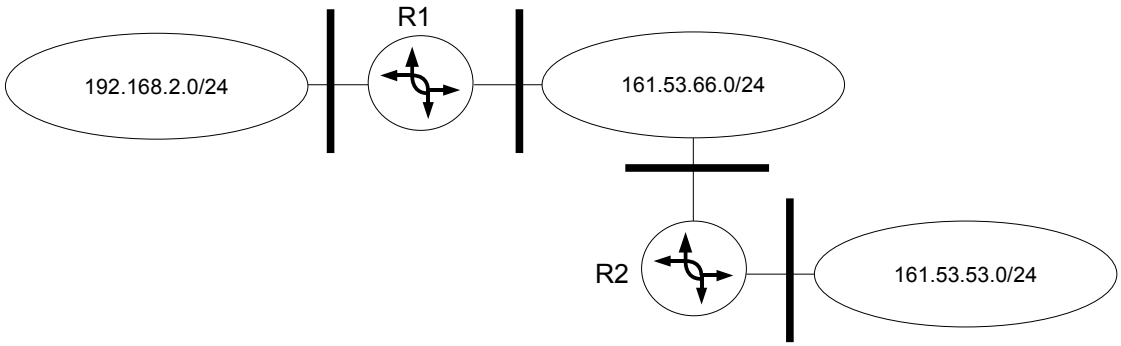
Mreža	***Sljedeći čvor***	***Izlazno sučelje***
192.168.2.0/24		eth0
192.168.5.12/30		ppp0
192.168.1.0/24	192.168.5.13	ppp0
192.168.3.0/24	192.168.5.13	ppp0
192.168.4.0/24	192.168.5.13	ppp0

Tablica 10. Tablica proslijeđivanja usmjernika R4

Mreža	***Sljedeći čvor***	***Izlazno sučelje***
192.168.4.0/24		eth0
192.168.5.4/30		ppp0
192.168.5.12/30		ppp1
192.168.1.0/24	192.168.5.5	ppp0
192.168.2.0/24	192.168.5.5	ppp0
192.168.3.0/24	192.168.5.13	ppp1

PRIMJER 9. Izrađivanje tablica proslijeđivanja

Za mrežnu konfiguraciju prikazanu na slici složiti odgovarajuću tablicu usmjeravanja usmjernika. Na mjestima gdje je to potrebno odabrati odgovarajuće IP adrese i imena sučelja poštivajući uobičajena pravila.

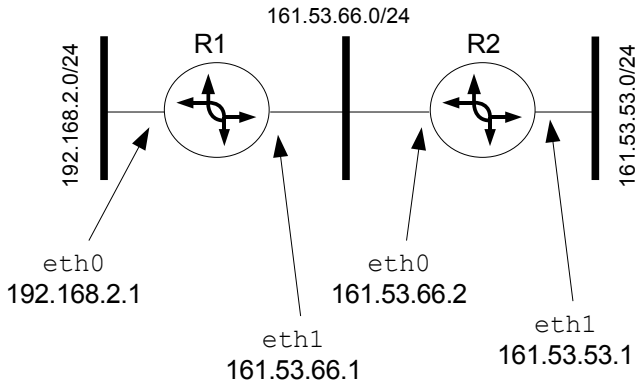


Rješenje

Opet kao i u prethodnom zadatku potrebno je prije kreiranja same tablice označiti sučelja i dodijeliti im IP adrese. Oba usmjernika imaju po dva ethernet sučelja. Oznake na usmjerniku R1 su `eth0` i `eth1` i isto, na usmjerniku R2 su oznake sučelja `eth0` i `eth1`. Treba primjetiti kako možemo koristiti više puta indeks 0 sučelja, ali pod uvijetom da su sučelja na različitim usmjernicima!

Odabir IP adresa identičan je prethodnom zadatku s jednom malom razlikom. U ovom slučaju oba usmjernika spojena su na istu Ethernet mrežu (mreža s adresom 161.53.66.0/24) te u tom slučaju treba pripaziti prilikom odabira IP adrese. Jedan usmjernik će dobiti adresu 161.53.65.1, dok će drugi dobiti sljedeću najnižu, tj. 161.53.65.2.

Usmjernici, s oznakama sučelja i dodijeljenim adresama, prikazani su na sljedećoj slici:



Sada je na osnovu slike potrebno napisati tablice proslijeđivanja. Postupak kreiranja tablice usmjeravanja sličan je postupku opisanom u prethodnom zadatku osim što sadrži nekoliko dodatnih koraka zbog složenije mreže. Postupak za generiranje tablice treba provesti za **svaki** usmjernik, a sastoji se od sljedećih koraka:

- Za svaku mrežu koja postoji na zadanoj slici napravi se jedan redak tablice.
- U prvi stupac piše se adresa mreže.

- Potrebno je provjeriti kroz koje sučelje se stiže do te mreže. To sučelje se piše u treći stupac tablice.
- Ako mreža nije direktno spojena na usmjernik tada je potrebno pronaći sučelje usmjernika kome treba predati paket. Adresu tog sučelja upisati u drugi stupac tablice. Pripaziti da se za sučelje odaberu samo direktno spojena sučelja!

U ovom zadatku postoje dva usmjernika te će biti dvije tablice proslijeđivanja, za svaki usmjernik po jedna. Budući da su ovom zadatku zadane tri mreže (192.168.2.0/24, 161.53.66.0/24 i 161.53.53.0/24) to će svaka tablica proslijeđivanja imati tri retka.

Uzmimo primjer usmjernika R1 i mreže 161.53.53.0/24. Ta mreža dostupna je tom usmjerniku preko sučelja `eth1`. Ali, kako se ne radi o mreži direktno spojenoj na sam usmjernik to proizlazi da će se paket morati predati idućem usmjerniku. Taj usmjernik, prema slici, je R2, tj. na njegovo sučelje `eth0` koje ima IP adresu 161.53.66.2. Prema tome prvi redak tablice proslijeđivanja usmjernika R1 ima sljedeći oblik:

Odredišna mreža	***Idući čvor***	***Izlazno sučelje***
161.53.66.0/24	161.53.66.2	eth1

Provođenjem sličnog postupka i za ostale dvije mreže dobijamo potpunu tablicu proslijeđivanja usmjernika R1:

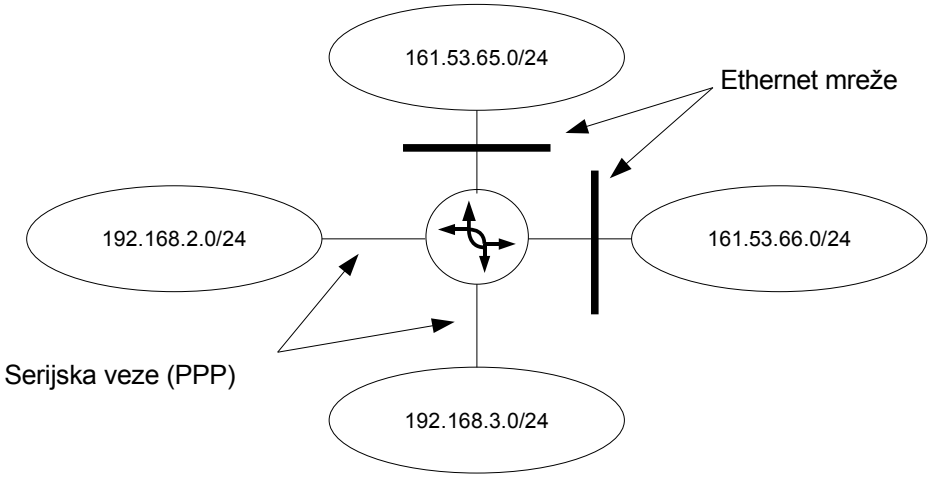
Odredišna mreža	***Idući čvor***	***Izlazno sučelje***
192.168.2.0/24		eth0
161.53.66.0/24		eth1
161.53.53.0/24	161.53.66.2	eth1

Cijeli postupak se potom ponavlja za usmjernik R2, te se na kraju odbija sljedeća tablica proslijeđivanja:

Odredišna mreža	***Idući čvor***	***Izlazno sučelje***
192.168.2.0/24	161.53.66.1	eth0
161.53.66.0/24		eth1
161.53.53.0/24		eth1

PRIMJER 10. ***Izrađivanje tablica proslijeđivanja***

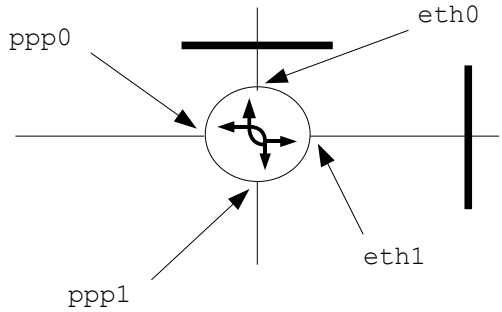
Za mrežnu konfiguraciju prikazanu na slici složiti odgovarajuću tablicu proslijeđivanja usmjernika R1. Na mjestima gdje je to potrebno odabrati odgovarajuće IP adrese i imena sučelja poštivajući uobičajena pravila.



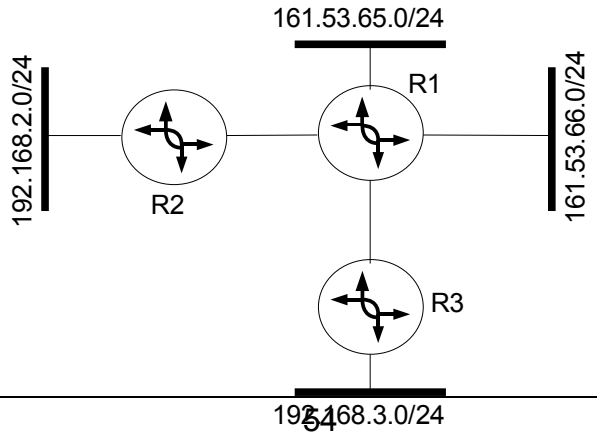
Rješenje

Razlika između ovog zadatka i prethodnih je u tome što se u ovom slučaju javljaju serijska sučelja. Budući da je postupak za ethernet sučelja identičan kao u prethodnim zadacima to će ovdje biti navedena pojašnjena samo za serijska sučelja.

Serijska sučelja na kojima se vrti PPP dobijaju imena čiji oblik je `ppp`*n*. Kao i kod Ethernet sučelja umjesto n piše se broj koji počinje od 0. Kako u ovom zadatku usmjernik ima dva serijska sučelja to će njihove oznake biti: `ppp0` i `ppp1`. To je prikazano na sljedećoj slici:



Kod dodjele IP adresa treba pripaziti. Naime, serijska sučelja su specifična budući da jedno takvo sučelje može spojiti isključivo dva usmjernika. To je razlika u odnosu na Ethernet na kojemu se može nalaziti više računala! Na sljedećoj slici prikazana je jedna moguća izvedba mreže iz zadatka.



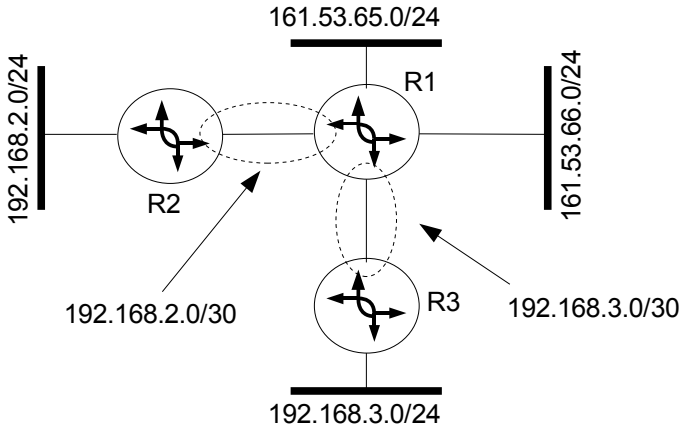
Razlog zašto je to **samo jedna moguća** izvedba je činjenica da zadatkom nije definirana točna struktura mreža 192.168.3.0/24 i 192.168.2.0/24! Postoje i druge mrežne konfiguracije koje bi odgovarale generičkom obliku zadanom u zadatku. Ipak, ta moguća konfiguracija će dobro poslužiti da se pojasni odabir IP adresa za serijska sučelja!

Primjetite kako su usmjernici R1 i R2 spojeni serijskom vezom, te su također usmjernici R1 i R3 spojeni serijskom vezom! Serijska veza omogućava da se direktno spoje isključivo dva usmjernika, za razliku od primjerice Ethernet mreže na koju je moguće spojiti (skoro) proizvoljan broj usmjernika i oni će moći **direktno** komunicirati!

Posljedica toga je potreba da se pažljivo odaberu IP adrese i njihove mrežne maske za serijsku vezu. Kako bi vidjeli zašto je to tako pogledajmo koliko računala se može nalaziti na mreži čija mrežna adresa je 192.168.2.0/24. Na toj mreži ostavljeno je 8 bita za broj računala, prema tome teoretski maksimum je 256 ($2^{32 - 24}$). Međutim, ne mogu se koristiti IP adrese koje na mjestu koje pripada računalu imaju sve nule ili sve jedinice. To u ovom slučaju znači da se ne mogu koristiti adrese 192.168.2.0 i 192.168.2.255, budući da one imaju posebno značenje. Prema tome, ostaje na raspolaganju 254 adrese (256 – 2) i mogućnost priključivanja isto toliko računala (namjerno ignoriramo činjenicu kako i usmjernik uzima jednu IP adresu!).

Sada je jasno da na serijskoj vezi odabirom 24 bita za mrežnu masku ostavljamo mogućnost spajanja 254 računala, ali serijska veza omogućava maksimalno dva računala! Iz tog razloga se za serijske veze odabire mrežna maska 30! Mreža koja spaja usmjernike R1 i R2 biti će podmreža mreže 192.168.2.0/24 i mreža koja spaja usmjernike R1 i R3 će biti podmreža mreže 192.168.3.0/24.

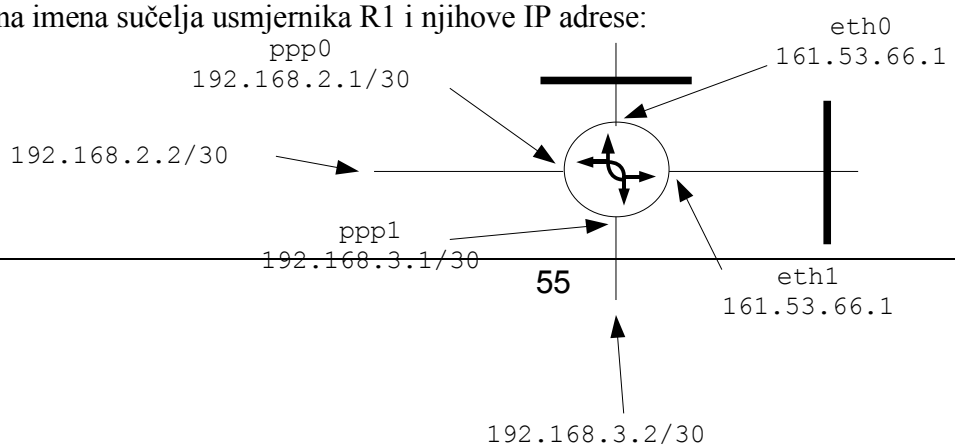
Te mreže su sada naznačene na sljedećoj slici:



Kako je rečeno, adrese sučelja biraju se tako da pripadaju mreži na kojoj se sučelje nalazi. Mrežna maska /30 dopušta isključivo dvije adrese, te se samo treba odlučiti kojem sučelju će biti dodjeljena koja od dvije raspoložive adrese. Oznake sučelja R2 i R3 nisu bitna budući da njihove tablice proslijeđivanja nije potrebno slagati.

Primjetite kako se sada na slici nalazi 6 mreža! To znači da se u tablici proslijeđivanja usmjernika R1 mora nalaziti 6 zapisa!

Kako R2 i R3 nisu bitni za rješavanje zadatka to su oni uklonjeni i na sljedećoj slici prikazane su odabrana imena sučelja usmjernika R1 i njihove IP adrese:



Imena sučelja usmjernika R2 i R3 nisu bitni i to je naznačeno na prethodnoj slici. Definirane su IP adrese (adrese 192.168.2.2/30 i 192.168.3.2/30) ali uz njih nisu navedena imena sučelja!

Na prethodnoj slici može se primjetiti još jedna specifičnost. Ethernet sučelja nemaju navedenu mrežnu masku uz IP adrese, dok ih ppp sučelja imaju! Razlog je u tome da Ethernet sučelja zadržavaju mrežnu masku mreže (/24) te ju nije potrebno posebno navoditi. Za razliku od njih, ppp sučelja imaju drugačiju mrežnu masku (/30) od zadane za mrežu te se ona zbog toga eksplicitno navodi.

Algoritam izgradnje tablice proslijeđivanja sličan je već opisanom postupku osim što u slučaju serijske veze treba uzeti u obzir kako se do mreže na suprotnoj strani dolazi preko IP adrese spojene na drugi kraj serijskog sučelja. Primjerice, do mreže 192.168.2.0/24 se dolazi kroz sučelje ppp0, ali je paket potrebno proslijediti drugom kraju, tj. IP adresi 192.168.2.2.

U potpunosti ponavljajući algoritam izgradnje tablice proslijeđivanja dan u prethodnim zadacima, uzimajući u obzir specifičnosti serijskih sučelja, dobijamo sljedeću tablicu usmjeravanja usmjernika R1:

Odredišna mreža	***Idući čvor***	***Izlazno sučelje***
161.53.65.0/24		eth0
161.53.66.0/24		eth1
192.168.2.0/24	192.168.2.2	ppp0
192.168.3.0/24	192.168.3.2	ppp1
192.168.2.0/30		ppp0
192.168.3.0/30		ppp1

PRIMJER 11. ***Crtanje mreže na osnovu zadanih tablica proslijeđivanja***

Na nekakvom internetu nalaze se četiri usmjernika s sljedećim tablicama proslijeđivanja. Nacrtati shemu mreže na osnovu zadanih tablica.

Usmjernik 1

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24		eth0
192.168.1.0/24		eth1
192.168.2.0/24	192.168.1.2	eth1
192.168.3.0/24	192.168.0.1	eth0
192.168.4.0/24	192.168.1.2	eth1
192.168.5.0/24	192.168.1.2	eth1

Usmjernik 2

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24	192.168.2.1	eth0
192.168.1.0/24	192.168.2.1	eth0
192.168.2.0/24		
192.168.3.0/24	192.168.2.1	eth0
192.168.4.0/24		
192.168.5.0/24	192.168.2.1	eth0

Usmjernik 3

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24		
192.168.1.0/24		
192.168.2.0/24		
192.168.3.0/24		
192.168.4.0/24		
192.168.5.0/24		

Usmjernik 4

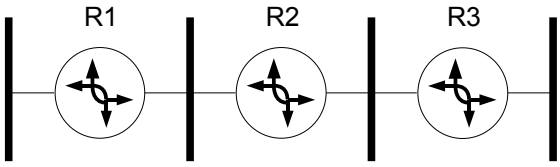
Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24		
192.168.1.0/24		
192.168.2.0/24		
192.168.3.0/24		
192.168.4.0/24		
192.168.5.0/24		

Rješenje

Razlika između ovog zadatka i prethodnih je u tome što se u ovom slučaju javljaju serijska sučelja. Budući da je postupak za ethernet sučelja identičan kao u prethodnim zadacima to će ovdje biti navedena pojašnjena samo za serijska sučelja.

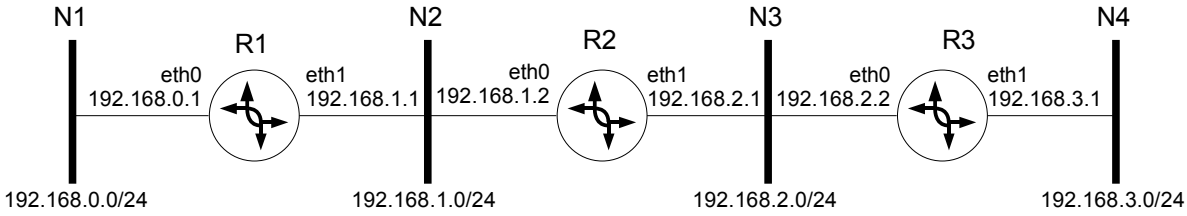
PRIMJER 12. Princip izvršavanje naredbe `ping` na složenoj mreži

Na slici je prikazana konfiguracija usmjernika. Prikazati generirani promet na mreži kada se na usmjerniku R1 pokrene naredba ping s odredišnom adresom desnog sučelja usmjernika R3.



Rješenje

Kako bi mogli pratiti promet na mrežama moraju biti definirane sve adrese mreža, oznake sučelja svih usmjernika, adrese sučelja i tablice usmjeravanja svakog od usmjernika. Na sljedećoj slici prikazane su sve potrebne oznake za sliku iz zadatka. Dodatno su sve mreže označene sa slovom N i rednim brojem kako bi označavanje prometa na pojedinim mrežama bilo jednostavnije.



Tablice proslijeđivanja svih usmjernika su:

Usmjernik R1

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24		eth0
192.168.1.0/24		eth1
0.0.0.0/0	192.168.1.2	eth1

Usmjernik R2

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24	192.168.1.1	eth0
192.168.1.0/24		eth0
192.168.2.0/24		eth1
192.168.3.0/24	192.168.2.2	eth0

Usmjernik R3

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.2.0/24		eth0
192.168.3.0/24		eth1
0.0.0.0/0	192.168.2.1	eth0

Po uvjetima zadatka i na osnovu odabranih parametara mreže, na usmjerniku R1 pokreće se sljedeća naredba:

```
$ ping 192.168.3.1
```

Za svoj rad naredba `ping(8)` koristi dvije ICMP poruke. Prva poruka je *Echo request*, a druga je *Echo response*. Odmah nakon pokretanja naredba kreira paket koji sadrži Echo request ICMP poruku koju prosljeđuje mrežnom sloju usmjernika R1. Usmjernik R1 prosljeđuje paket čvoru 192.168.1.2 kroz sučelje eth1 kako mu je to određeno tablicom proslijeđivanja. Dakle, na mreži N2 vidi se sljedeći paket:

N2: IP (192.168.3.1, 192.168.1.1, ICMP (Echo, 1))

Navedeni IP paket sadrži ICMP poruku koju ćemo na dalje označavati s Echo. Osim oznake tipa poruke ICMP teret sadrži i još nekoliko dodatnih podataka. Bitan podatak o kojemu će se voditi evidencija tijekom rješavanja primjera je slijedni broj poruke koji monotono raste od 1. Osim slijednog broja u stvarnosti ping naredba koristi i identifikator, te vrijeme kada je poruka poslana.

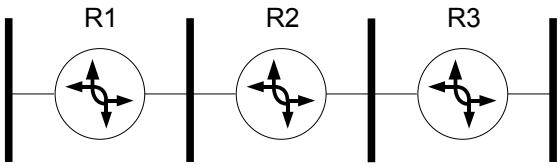
Kada navedeni paket pristigne do R2 on na osnovu svoje tablice proslijeđivanja predaje ga do čvora 192.168.2.2 preko sučelja eth1. Prema tome, na mreži N3 opet se vidi isti paket. Usmjernik R3 utvrđuje da je IP paket namijenjen njemu, tj. jednom od njegovih sučelja te uzima teret iz paketa. Prema Internet specifikacijama [RFC????] svako računalo ili usmjernik na Internetu koje primi ICMP Echo poruku mora odgovoriti na nju porukom Echo Response. U odgovoru treba zamijeniti izvorišnu i odredišnu adresu te sačuvati sve podatke koji su pristigli u ICMP Echo poruci. To znači da će R3 kreirati odgovor i proslijediti ga do R2. Dakle, na mreži N3 vidjeti će se sljedeći IP paket:

N3: IP (192.168.1.1, 192.168.3.1, ICMP (EchoResponse, 1))

R2 u skladu sa svojim tablicama proslijeđivanja predaje taj paket usmjerniku R1 koji utvrđuje kako je paket namijenjen njemu te ga predaje naredbi ping. Naredba ping na osnovu primljenog odgovora ispisuje informacijski redak. Taj redak sadrži slijedni broj odgovora i proteklo vrijeme od slanja upita pa do primitka odgovora.

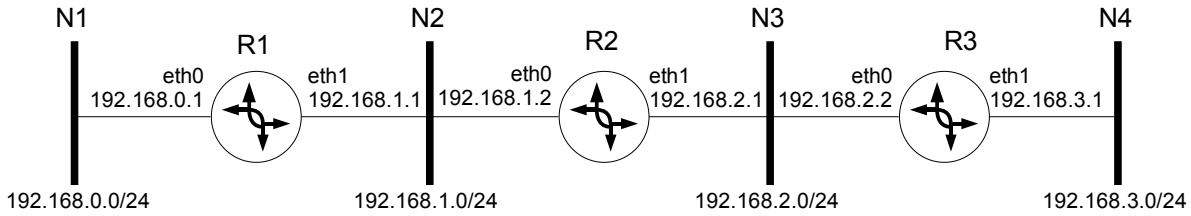
PRIMJER 13. Princip izvršavanja naredbe `traceroute` na složenoj mreži

Na slici je prikazana konfiguracija usmjernika. Prikazati generirani promet na mreži kada se na usmjerniku R1 pokrene naredba traceroute do desnog sučelja usmjernika R3.



Rješenje

Kao i u slučaju prethodnog primjera potrebno je prvo označiti sve parametre na mreži kako bi se moglo zapisivati pakete koji prolaze po mrežama. Sljedeća slika prikazuje odabrane parametre na zadanoj mreži:



Naredba `traceroute(8)` za svoj rad koristi UDP pakete i dvije vrste ICMP poruka, o isteku vremenskog ograničenja te nedostižnom odredištu.

Prvi paket koji naredba šalje nakon pokretanja je:

N2: IP (192.168.3.1, 192.168.1.1, 1, UDP())

Paket je vidljiv na mreži N2, odredište mu je prema uvjetima zadatka desno sučelje usmjernika R3 koje ima adresu 192.168.3.1, a šalje se sa sučelja 192.168.1.1. U slučajevima kada računalo ili usmjernik ima više IP adresa tada se prilikom slanja koristi adresa sučelja kroz koje paket izlazi. Treći parametar IP paketa sadrži vrijednost TTL vrijednost iz zaglavlja IP paketa. Nakon što taj paket pristigne do R2, on prvo provjerava da li je paket namijenjen lokalnom računalu te ako nije umanjuje TTL. Budući da je umanjeni TTL 0 to prema specifikaciji IP protokola znači da je paketu isteklo vremensko ograničenje i mora biti odbačen. Nakon odbacivanja paketa usmjernik generira ICMP poruku kojom obaviještava pošiljatelja da je isteklo vremensko ograničenje. Prema tome, na N2 mreži sada se vidi sljedeći paket:

N2: IP (192.168.1.1, 192.168.1.2, ICMP(TimeExceeded))

Tu obavijest prihvaća program traceroute te nakon slanja još dva paketa ispisuje minimalno, maksimalno i srednje vrijeme potrebno do prvog čvora. Nakon toga traceroute šalje sljedeći IP paket:

N2: IP (192.168.3.1, 192.168.1.1, 2, UDP())

Sada je TTL uvećan za 1. Nakon što taj paket pristigne do R2 on ponavlja identičan postupak kao i ranije, ali ovaj puta nakon umanjivanja TTLa za jedan on nije nula te ga na osnovu tablica usmjeravanja šalje do R3. Prema tome, na N3 mreži vidljiv je sljedeći paket:

N3: IP (192.168.3.1, 192.168.1.1, 1, UDP())

Taj paket prihvaća R3 i nakon što ustanovi da je namijenjen njemu isporučuje UDP višim slojevima jezgre operacijskog sustava. Kako na pristupu navedenom u UDP paketu ne čeka niti jedna aplikacija,

to se generira ICMP poruka o nedostižnom odredištu koja se potom vraća nazad pošiljatelju. Zbog toga se na mreži N3 vidi sljedeći paket:

N3: IP (192.168.1.1, 192.168.2.1, ICMP(Unreachable, port))

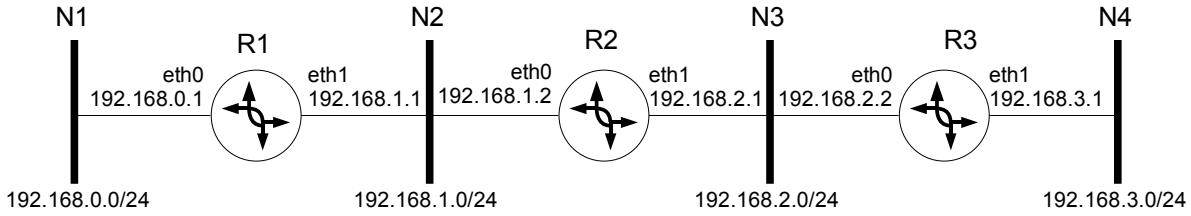
Taj paket prihvaća R2 koji ga potom proslijeđuje do R1 te se na mreži N2 prilikom prijenosa od R2 do R1 vidi sljedeći paket:

N3: IP (192.168.1.1, 192.168.2.1, ICMP(Unreachable, port))

Na R1 tu poruku prihvaća tcpdump koji nakon što kompletnu proceduru ponovi još dva puta ispisuje konačni čvor na putu do odredišta.

PRIMJER 14. Izvršavanja naredbe `traceroute` na složenoj mreži

Na slici je prikazana konfiguracija usmjernika. Na usmjerniku R1 pokrenuti naredbu traceroute do desnog sučelja usmjernika R3. Prikazati i rastumačiti snimljeni promet na svim



mrežama korištenjem programa tcpdump.

Rješenje

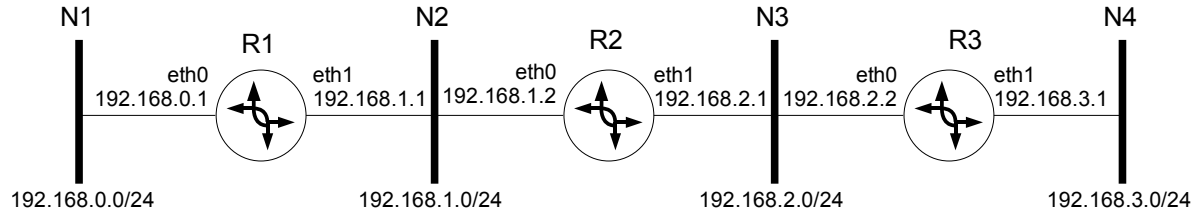
Kao i u slučaju prethodnog primjera potrebno je prvo označiti sve parametre na mreži kako bi se moglo zapisivati pakete koji prolaze po mrežama. Sljedeća slika prikazuje odabrane parametre na zadanoj mreži:

PRIMJER 15. Tuneliranje

Na slici je prikazana konfiguracija usmjernika zajedno sa svim relevantnim mrežnim parametrima. Uz pretpostavku da su mreže N1 i N4 povezane putem tunela prikazati tablice proslijeđivanja i promet na svim mrežama za slijedeći niz paketa:

N1: IP(192.168.3.5, 192.168.0.3)

N4: IP(192.168.0.3, 192.168.3.5)



Rješenje

Tablice proslijeđivanja usmjernika imaju sljedeći oblik:

Usmjernik R1

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24		eth0
192.168.1.0/24		eth1
192.168.3.0/24		tun0
0.0.0.0/0	192.168.1.2	eth1

Usmjernik R2

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.1.0/24		eth0
192.168.2.0/24		eth1

Usmjernik R3

Odredišna mreža	Idući čvor	Izlazno sučelje
192.168.0.0/24		tun0
192.168.2.0/24		eth0
192.168.3.0/24		eth1
0.0.0.0/0	192.168.2.1	eth0

Dakle, tablice su ponešno modificirane. Za početak, usmjernik R2 nezna za postojanje mreža N1 i N4 budući da je pretpostavka kako se radi o privatnim i izvana dostupnim mrežana. Zatim, usmjernik R1 sadrži informacije o mreži N4 isto kao što i usmjernik R3 sadrži informacije o mreži N1. Međutim, te mreže definirane su kao direktno spojene putem uređaja pod nazivom `tun0`.

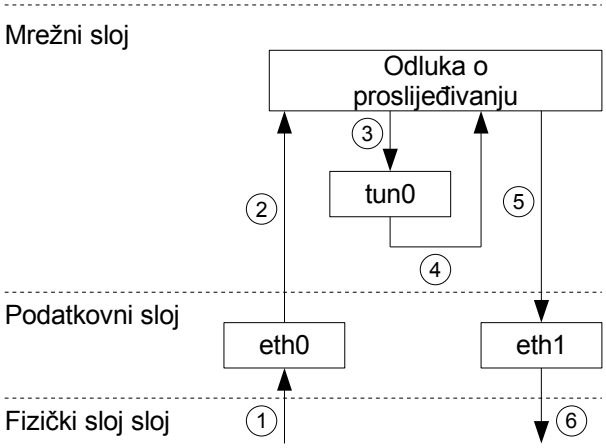
Uređaj tun0 predstavlja tunel, tj. virtualni mrežni uređaj koji je posebno podešen. Kada neki IP paket dođe tom uređaju za slanje on ga upakira u novi paket i potom ponovo proslijedi mrežnom sloju operacijskog sustava. Shematski je put koji prolazi taj paket u jezgri operacijskog sustava prikazan na slici 22.

Pogledajmo na primjeru točan put što ga prolazi paket. U zadatku je rečeno kako neko računalo na mreži N1 generira sljedeći IP paket:

N1: IP (192.168.3.5, 192.168.0.3)

Taj paket pristiže do sustava za proslijeđivanje usmjernika R1 (koraci 1 i 2). Taj sustav pretraživanjem svojih tablica odlučuje paket proslijediti virtualnom mrežnom uređaju tun0 (korak 3). Po primitku paketa virtualni mrežni uređaj tun0 upakira podatke u novi IP paket sljedećeg oblika:

IP (192.168.2.2, 192.168.1.1, IP (192.168.3.5, 192.168.0.3))



Slika 22. Princip izvedbe tuneliranja u jezgri operacijskog sustava

Dakle, paket koji treba otići u mrežu N4 upakiran je u novi IP paket čija odredišna adresa je vanjsko (lijevo) sučelje usmjernika R3, a izvorišna adresa je također vanjsko (desno) sučelje usmjernika R1. Nakon što je iskreiran novi paket se predaje na odluku o proslijeđivanju na **istom** usmjerniku (korak 4). Ovaj puta odredišna adresa 192.168.2.2 je prema tablici proslijeđivanja usmjernika R1 dostupna preko čvora s adresom 192.168.1.2 na sučelju eth0. Prema tome, sada paket odlazi preko sučelja eth0 (korak 5) do usmjernika R2 (korak 6). To znači da je na mreži N2 vidljiv sljedeći paket:

N2: IP (192.168.2.2, 192.168.1.1, IP (192.168.3.5, 192.168.0.3))

Ta sučelja i njihove adrese je nužno upotrebljavati budući da su njihove IP adrese poznate svim vanjskim usmjernicima. Točne adrese izvorišta i odredišta vanjskog IP paketa definiraju se prilikom podešavanja tunela korištenjem odgovarajućih alata.

Po primitku paketa, R2 na osnovu odredišne adrese i svojih tablica proslijeđivanja, šalje paket do R3, te se prema tome na mreži N3 vidi sljedeći paket:

N3: IP (192.168.2.2, 192.168.1.1, IP (192.168.3.5, 192.168.0.3))

Nakon što R3 prihvati paket on u jezgri operacijskog sustava prolazi put prikazan na slici 23. Paket pristiže preko sučelja eth0 (korak 1), te se potom proslijeđuje mrežnom sloju na donošenje odluke o proslijeđivanju (korak 2). Kako je paket namijenjen samom usmjerniku uzima se njegov sadržaj. Sadržaj paketa je novi IP paket koji se ponovo proslijeđuje mrežnom sloju na odluku o proslijeđivanju (korak 3). Rezultat odluke je da se paket proslijedi kroz sučelje eth1 (korak 4) direktno računalu kojemu je namijenjeno (korak 5). Prema tome, na mreži N4 vidljiv je sljedeći promet:

N4: IP (192.168.3.5, 192.168.0.3)

Po primitku tog paketa, računalo 192.168.3.5 šalje odgovor, tj. sljedeći paket:

N4: IP (192.168.0.3, 192.168.3.5)

Tak paket se proslijeđuje usmjerniku R3 koji na osnovu svoje tablice proslijeđivanja koristi za slanje virtualni uređaj tun0. Taj uređaj na osnovu postavki upakira paket u novi IP paket sljedećeg oblika:

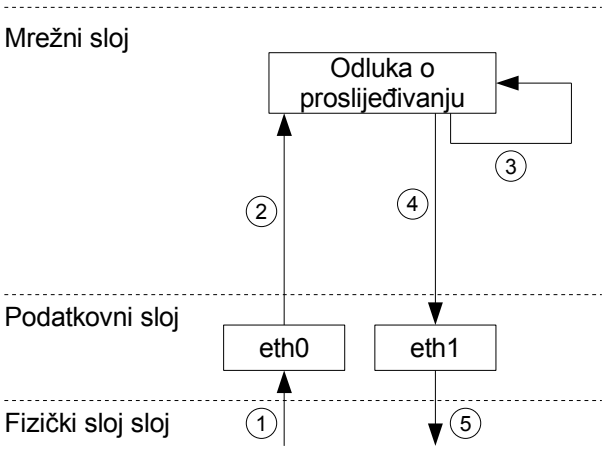
IP (192.168.1.1, 192.168.2.2, IP (192.168.0.3, 192.168.3.5))

Te ga ponovo proslijedi mrežnom sloju. Mrežni sloj na osnovu odredišne adrese i tablica proslijeđivanja šalje paket do R2 pa se prema tome na mreži N3 vidi sljedeći promet:

N3:IP (192.168.1.1, 192.168.2.2, IP (192.168.0.3, 192.168.3.5))

R2 ga nakon primitka proslijeđuje do R1 te je na mreži N2 viđen isti promet kao i na mreži N3. Konačno, R1 po primitku paketa vadi unutarnji IP paket te ga potom šalje do odredišnog računala. Prema tome, na mreži N1 viđen je sljedeći promet:

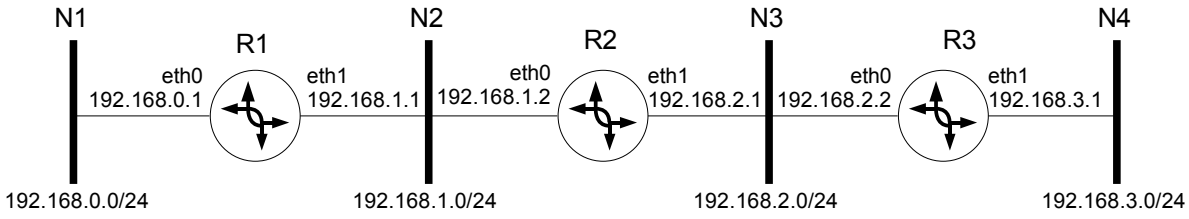
N1: IP (192.168.0.3, 192.168.3.5)



Slika 23. Slijed tuneliranog paketom u jezgri operacijskog sustava

PRIMJER 16. Prevođenje odredišnih mrežnih adresa

Slika prikazuje mrežnu konfiguraciju u kojoj je N1 privatna mreža čije IP adrese se ne smiju pojaviti na ostalim mrežama. Uz pretpostavku da računala s te mreže ipak moraju imati pristup do mreža N2, N3 i N4 prikazati način na koji se to može postići te generiran mrežni promet za nekoliko primjera.



Rješenje

Mreže se općenito mogu podijeliti u dvije grupe: javne i privatne. Karakteristika javnih mreža je da koriste javne IP adrese. Javne IP adrese moraju se zatražiti od odgovarajućih institucija i mogu se koristiti na cijelom Internetu. Primjerice, paket nastao u jednoj od takvih mreža šalje se na odredište i tijekom cijelog svog puta izvorišna adresa ostaje nepromijenjena. S druge strane, privatne mreže koriste IP adrese koje se ne smiju pojaviti na Internetu, tj. neke od mreža koje pripadaju rezerviranim IP blokovima, primjerice 10.0.0.0/24. Kada bilo koji usmjernik na Internetu prihvati paket koji u izvorišnoj ili odredišnoj adresi sadrži IP adresu koja pripada nekom od rezerviranih blokova on će odbaciti taj paket kao nelegalan. Prema tome bez nekakvih posebnih mjera privatne mreže ne bi mogle slati pakete na javne mreže niti bi s javnih mreža mogli prihvaćati pakete.

Za potrebe primjera mreža N1 je definirana kao privatna mreža koja koristi privatne IP adrese, dok su N2, N3 i N4 javne mreže s javnim IP adresama. Kako bi paketi s mreže N1 mogli doći do mreža N2, N3 i N4 nužno je podesiti usmjernik R1 da prikrije privatne adrese. Umjesto privatnih adresa usmjernik R1 će koristiti adresu svog javnog sučelja, je 192.168.1.1. Naime, to mora napraviti iz dva razloga. Prvi, jer je to jedina javna adresa koju posjeduje, i drugi razlog, kada se vraćaju odgovori oni se moraju vratiti usmjerniku R1 kako bi im on potom mogao vratiti prave adrese iz privatnog raspona.

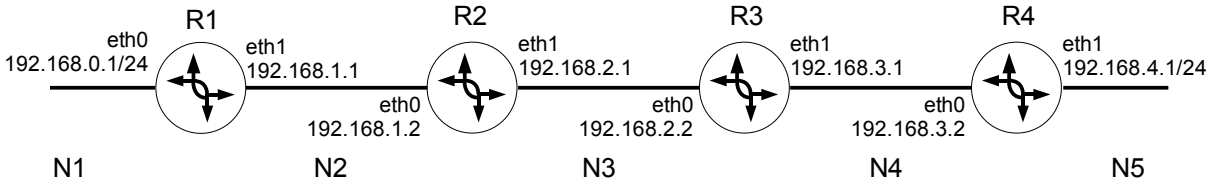
Uzmimo primjer kada računalo s adresom 192.168.0.10 šalje paket računalu 192.168.3.10, tj. na mreži N1 vidljiv je sljedeći paket na putu do R1:

N1: IP (192.168.3.10, 192.168.0.10)

Po primitku tog paketa usmjernik R1 nakon pregleda svoje tablice proslijeđivanja odlučuje da će paket poslati

PRIMJER 17. Algoritam usmjeravanja RIP

Slika prikazuje mrežnu konfiguraciju u kojoj svi usmjernici koriste protokol RIP za razmjenu informacija usmjeravanja. Uz pretpostavku da su usmjernici na mreži tek uključeni prikazati promet koji generira protokol RIP, izračun tablica prosljeđivanja i konačni izgled tablica proslijeđivanja nakon što stabilizira stanje u mreži.



Rješenje

Promet na mreži dosta ovisi o različitim parametrima, primjerice kada su usmjernici uključeni, kada su dodavane pojedine mreže ili brzini procesora pojedinih usmjernika. Uzmimo primjerice slučaj u kojemu su svi usmjernici potpuno identični te su istovremeno uključeni. U trenutku uključenja algoritmi usmjeravanja imaju informacije isključivo o direktno povezanim mrežama. Oni odmah podatke o tim mrežama oglašavaju na sve direktno spojene mreže. Pri tom oglašavanju na pojedinu mrežu ne šalju podatke da su spojeni na nju. Primjerice, usmjernik R1 neće na mrežu N2 slati informaciju da mu je ta mreža dostupna.

Izvršavanje RIP algoritma pratiti ćemo uz pomoć tablice sljedeća oblika:

Rbr.	*R1*	*R2*	*R3*	*R4*

Dakle, tablica sadrži kolonu za svaki usmjernik na mreži, a također kolonu u kojoj se vodi informacija o svakom koraku algoritma (Rbr.). Budući da u tekstu primjera ništa nije rečeno o vremenima slanja informacija pojedinih usmjernika, dodatno ćemo pojednostavili praćenje izvršavanja algoritma korištenjem pretpostavke da svi usmjernici rade sinkrono. Algoritam je iterativan pri čemu se svaki korak sastoji od dvije faze. U prvoj fazi razmjenjuju se tablice usmjeravanja, dok u drugoj fazi svaki usmjernik na osnovu primljenih informacija modificira svoju tablicu usmjeravanja.

Dakle, odmah nakon uključenja usmjernika oni su u inicijalnom stanju u kojemu su im poznate samo njihove direktno povezane mreže:

Rbr.	*R1*	*R2*	*R3*	*R4*
0.	N1, 0 N2, 0	N2, 0 N3, 0	N3, 0 N4, 0	N4, 0 N5, 0

Mreže u tablicama navodimo simboličkim imenima umjesto IP adresama radi manje količine pisanja. Svaka direktno povezana mreža ima definiranu udaljenost 0.

Nakon inicijalizacije kreće razmjena podataka o mrežama. Na mreži N1 usmjernik R1 šalje svoje podatke, tj. sljedeći IP paket:

N1: IP (RIP2, 192.168.0.1, UDP(RIP((192.168.1.0/24, 1))))

Taj IP paket, kao i svaki drugi koji sadrži podatke RIP protokola, šalje se na posebnu adresu koja je u ovom slučaju označena kao RIP2. To je adresa koja označava sve RIP usmjernike na mreži. Podaci RIP protokola upakirani su u UDP paket, a sastoje se od liste parova. Svaki par sadrži odredišnu mrežu i cijenu slanja do navedene mreže. Slično, na ostalim mrežama vide se sljedeći IP paketi:

N2: IP (RIP2, 192.168.1.1, UDP(RIP((192.168.0.0/24, 1))))
N2: IP (RIP2, 192.168.1.2, UDP(RIP((192.168.2.0/24, 1))))

N3: IP (RIP2, 192.168.2.1, UDP(RIP((192.168.1.0/24, 1))))
N3: IP (RIP2, 192.168.2.2, UDP(RIP((192.168.3.0/24, 1))))

N4: IP (RIP2, 192.168.3.1, UDP(RIP((192.168.2.0/24, 1))))
N4: IP (RIP2, 192.168.3.2, UDP(RIP((192.168.4.0/24, 1))))

N5: IP (RIP2, 192.168.4.1, UDP(RIP((192.168.3.0/24, 1))))

Prihvat navedenih paketa od strane pojedinih usmjernika možemo pratiti i u tablici:

Rbr.	*R1*	*R2*	*R3*	*R4*
0.	N1, 0 N2, 0	N2, 0 N3, 0	N3, 0 N4, 0	N4, 0 N5, 0
1.	N3, 1, R2	N1, 1, R1 N4, 1, R3	N2, 1, R2 N5, 1, R4	N3, 1, R3

U toj tablici dodan je redak za prvu iteraciju algoritma, pri čemu je izvršena samo prva faza razmjene podataka. Za svaki primljeni paket dodan je jedan zapis usmjerniku koji ga je prihvatio. Primjerice, R1 je prihvatio paketa:

N2: IP (RIP2, 192.168.1.2, UDP(RIP((192.168.2.0/24, 1))))

koji je u tablici zapisao kao uređena trojka (N3, 1, R2). Skraćeno je zapisano kako je R2 poslao informaciju o mreži N3 čija udaljenost je 1 i do koje se može doći korištenjem R2 usmjernika.

U drugoj fazi svakog koraka algoritma vrši se analiza pristiglih podataka na osnovu kojih se ažuriraju tablice usmjeravanja. Stanje nakon analize prikazano je u sljedećoj tablici:

Rbr.	*R1*	*R2*	*R3*	*R4*
0.	N1, 0 N2, 0	N2, 0 N3, 0	N3, 0 N4, 0	N4, 0 N5, 0
1.	N3, 1, R2	N1, 1, R1 N4, 1, R3	N2, 1, R2 N5, 1, R4	N3, 1, R3
	N1, 0 N2, 0 N3, 1, R2	N1, 1, R1 N2, 0 N3, 0 N4, 1, R3	N2, 1, R2 N3, 0 N4, 0 N5, 1, R4	N3, 1, R3 N4, 0 N5, 0

Očito je kako sada R1 zna za mreže N1, N2 i N3, zatim R2 zna za mreže N1, N2, N3 i N4, itd. U idućem koraku algoritma ponovo se razmjenjuju podaci. Ovaj puta dolazi do izražaja pravilo *raspodijeljenog horizonta* (engl. split horizon). Naime, usmjernik ne šalje podatke o mreži usmerniku od kojega je prihvatio informaciju. Primjerice, R1 ima informaciju od mreži N3 do koje može doći preko usmjernika R2. On neće slati tu informaciju nazad do R2 kako bi se izbjegao problem odbrojavanja do beskonačnosti.

Rbr.	*R1*	*R2*	*R3*	*R4*
1.	N3, 1, R2	N1, 1, R1 N4, 1, R3	N2, 1, R2 N5, 1, R4	N3, 1, R3
	N1, 0 N2, 0 N3, 1, R2	N1, 1, R1 N2, 0 N3, 0 N4, 1, R3	N2, 1, R2 N3, 0 N4, 0 N5, 1, R4	N3, 1, R3 N4, 0 N5, 0
2.	N3, 1, R2 N4, 2, R2	N1, 1, R1 N4, 1, R3 N5, 2, R3	N1, 2, R2 N2, 1, R2 N5, 1, R4	N2, 2, R3 N3, 1, R3

U idućoj fazi analizira se pristigli promet te je stanje pojedinih usmjernika prikazano u sljedećoj tablici:

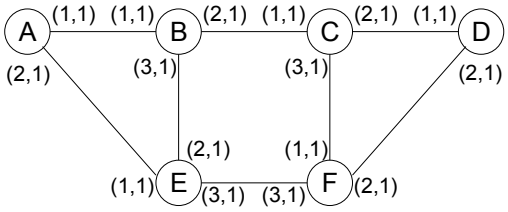
Rbr.	*R1*	*R2*	*R3*	*R4*
2.	N3, 1, R2 N4, 2, R2	N1, 1, R1 N4, 1, R3 N5, 2, R3	N1, 2, R2 N2, 1, R2 N5, 1, R4	N2, 2, R3 N3, 1, R3
	N1, 0 N2, 0 N3, 1, R2 N4, 2, R2	N1, 1, R1 N2, 0 N3, 0 N4, 1, R3 N5, 2, R3	N1, 2, R2 N2, 1, R2 N3, 0 N4, 0 N5, 1, R4	N2, 2, R3 N3, 1, R3 N4, 0 N5, 0

Konačno, slijedi zadnja razmjena i analiza pristiglih podataka te se nakon toga mreža stabilizira.

Rbr.	*R1*	*R2*	*R3*	*R4*
1.	N3, 1, R2 N4, 2, R2	N1, 1, R1 N4, 1, R3 N5, 2, R3	N1, 2, R2 N2, 1, R2 N5, 1, R4	N2, 2, R3 N3, 1, R3
	N1, 0 N2, 0 N3, 1, R2 N4, 2, R2	N1, 1, R1 N2, 0 N3, 0 N4, 1, R3 N5, 2, R3	N1, 2, R2 N2, 1, R2 N3, 0 N4, 0 N5, 1, R4	N2, 2, R3 N3, 1, R3 N4, 0 N5, 0
2.	N3, 1, R3 N4, 2, R3 N5, 3, R3	N1, 1, R1 N4, 1, R3 N5, 2, R3	N1, 2, R2 N2, 1, R2 N5, 1, R4	N1, 3, R3 N2, 2, R3 N3, 1, R3
	N1, 0 N2, 0 N3, 1, R3 N4, 2, R3 N5, 3, R3	N1, 1, R1 N2, 0 N3, 0 N4, 1, R3 N5, 2, R3	N1, 2, R2 N2, 1, R2 N3, 0 N4, 0 N5, 1, R4	N1, 3, R3 N2, 2, R3 N3, 1, R3 N4, 0 N5, 0

PRIMJER 18. Pouzdano preplavljivanje na računalnoj mreži

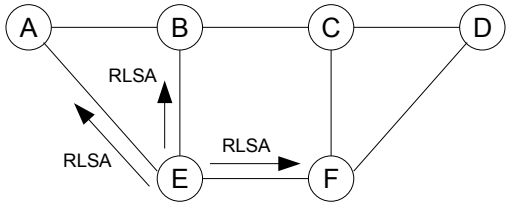
Na mreži prikazanoj na slici sve veze su tipa od točke-do-točke (point to point). Uz svako sučelje usmjernika naveden je uređen par: prvi član para predstavlja indeks sučelja, a drugi član cijenu slanja preko tog sučelja. Uz pretpostavku da se na svim usmjernicima izvršava OSPF algoritam usmjeravanja prikazati razmjenu informacija između usmjernika te konačni rezultat razmjene.



Rješenje

Protokol OSPF koristi pouzdano preplavljivanje (engl. reliable flooding) za distribuciju podataka o računalnoj mreži. Svaki usmjernik šalje usmjerničke LSA (engl. router LSA) poruke koje sadrže podatke o njegovim sučeljima te usmjernicima i mrežama spojenim na njih. Nakon nekog vremena svi usmjernici u mreži imat će identičnu bazu koja opisuje topologiju mreže. Na osnovu te baze računaju se najkraće udaljenosti u mreži te se podešavaju tablice proslijeđivanja usmjernika.

Kako bi opisali način na koji se razmjenjuju podaci o usmjernicima i mrežama, uzmimo primjer kada usmjernik E šalje svoj LSA. Jednostavnosti radi pretpostaviti ćemo kako je vrijeme slanja pojedinih LSA paketa, kao i vrijeme obrade u usmjernicima identično. S tom pretpostavkom u trenutku T=1 imamo sljedeću situaciju:

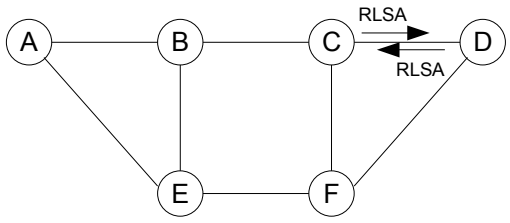


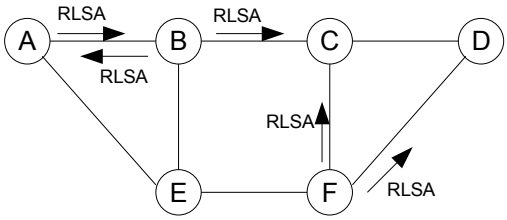
Dakle, usmjernik E šalje svoj LSA koji za potrebe primjera ima sljedeći oblik:

RLSA (E, 1, (1, 1, A), (2, 1, B), (3, 1, F))

LSA se prenosi direktno u IP paketu međutim to je u ovom primjeru zanemareno. Nadalje, svaki LSA nosi dosta više informacija što je također zanemareno. Nisu zanemareni podaci o usmjerniku koji je kreirao LSA (prvi parametar, usmjernik E), slijednom broju LSA (drugi parametar, slijedni broj 1), te podaci o sučeljima i priključenim usmjernicima. Kako je usmjernik E spojen na tri druga usmjernika to se nose informacije o tip spojevima u tri uređene trojke. Prvi član je indeks sučelja usmjernika, drugi član je cijena slanja preko tog sučelja i konačno, treći član predstavlja usmjernik koji se nalazi na drugom kraju sučelja.

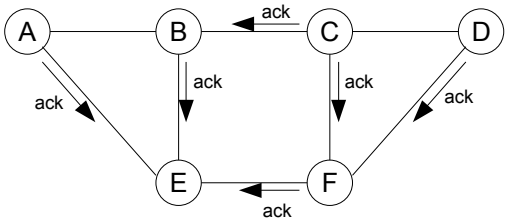
Nakon što su LSA prihvatili A, B i F, oni ga šalju dalje pri čemu ne šalju na sučelje na koje su prihvatili LSA. Dakle, u T=2 imamo sljedeći promet na mreži:





Dakle, u trećem koraku imamo sljedeću situaciju:

Konačna faza razmjene informacija je potvrda o primitku. Prilikom generiranja potvrde usmjernik šalje potvrdu na sva sučelja na koja je prihvatio LSA, međutim ako je na nekom sučelju primio identičan LSA tada neće slati potvrdu. Dakle, imamo sljedeću situaciju:



Nakon što svi usmjernici pošalju svoje LSAove i razmjene podatke, svaki usmjernik će imati sljedeću bazu:

RLSA (A, 1, (1, 1, B), (2, 1, E))

RLSA (B, 1, (1, 1, A), (2, 1, C), (3, 1, E))

RLSA (C, 1, (1, 1, B), (2, 1, D), (3, 1, F))

RLSA (D, 1, (1, 1, C), (2, 1, F))

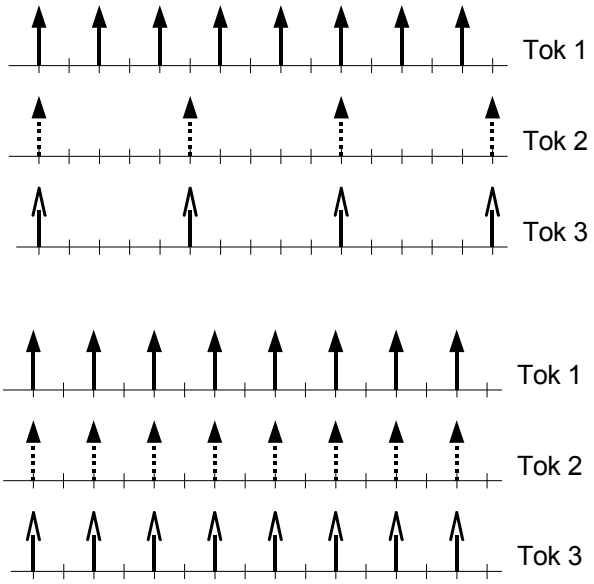
RLSA (E, 1, (1, 1, A), (2, 1, B), (3, 1, F))

RLSA (F, 1, (1, 1, C), (2, 1, D), (3, 1, E))

Na osnovu te baze usmjernik računa najkraće puteve u mreži i potom podešava tablice proslijeđivanja.

PRIMJER 19. Raspoređivanje paketa upotrebom FIFO metode

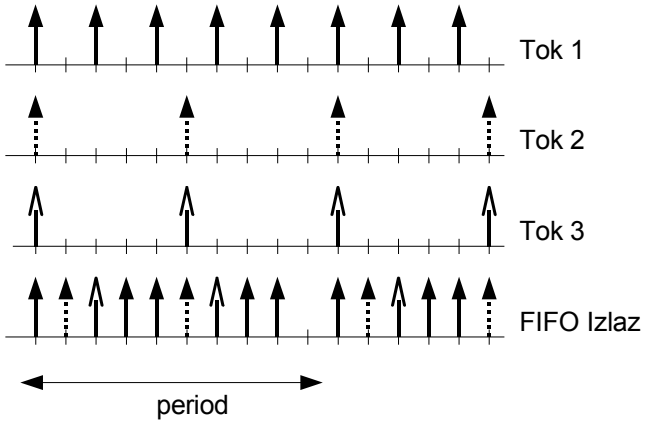
Prikazati izlazni promet usmjernika koji upotrebljava jedan red i raspoređivanje paketa upotrebom FIFO metode za dvije varijante prometa koji generiraju tri toka. Tok broj 1 je specificirao dvije vremenske jedinice između dva uzastopna paketa, a druga dva toka specificirala su pet vremenskih jedinica između dva uzastopna paketa. Pretpostaviti da je za prijenos jednog paketa potrebna jedna vremenska jedinica.



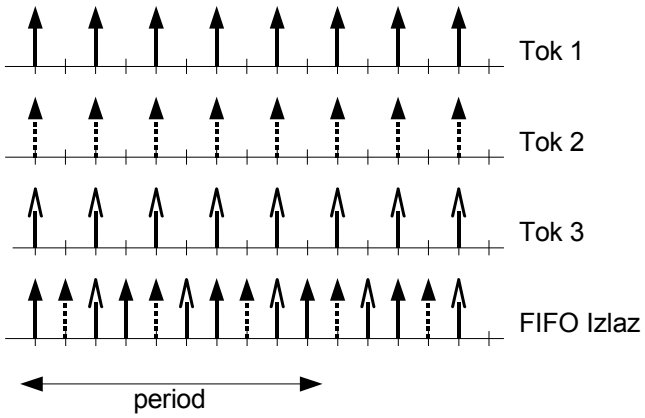
Rješenje

Od tri toka, prvi je rezervirao, tj. najavio, 0.5 paketa po vremenskoj jedinici, a druga dva su najavila 0.2 paketa po jedinici vremena. Kako je ukupan tok što ga generiraju sva tri toka 0.9 paketa u jedinici vremena, tj. manje od 1 to znači da će usmjernik biti u stanju opsluživati navedene tokove.

Kada se svi tokovi pridržavaju svoje specifikacije redoslijed dolazaka paketa i njihov odlazak prikazan je na sljedećoj slici:



Dakle, javlja se određeno kašnjenje no ipak svakom toku poslan je paket prije nailaska idućeg paketa. Međutim, u slučaju kada se neki od tokova ne pridržava svoje specifikacije FIFO metoda slanja pokazuje svoje mane. Na idućoj slici prikazana je situacija kada tokovi 2 i 3 počnu slati pakete brže od navedene specifikacije:

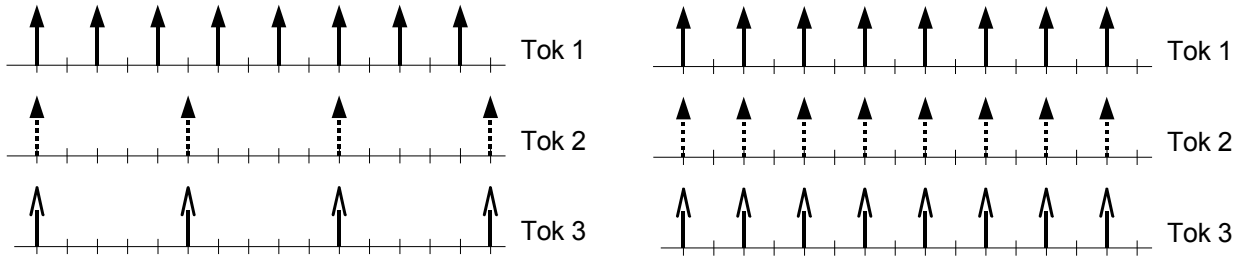


Očito je kako tijekom jednog perioda raspodjela više nije “pravična”, tj. tokovi 2 i 3 počinju brže slati pakete i zahvaljujući tome zauzimaju više izlaznog kapaciteta usmjernika na štetu toka 1.

Dodatna posljedica nepoštivanja specifikacija tokova 2 i 3 je porast reda u spremniku usmjernika koji će u jednom trenutku dovesti do njegova prepunjenja i odbacivanja dolaznih paketa. Ukratko, u usmjerniku nastaje zakrčenje.

PRIMJER 20. Raspoređivanje paketa upotrebom algoritma virtualnog sata

Prikazati raspoređivanje paketa upotrebom algoritma virtualnog sata za tri toka. Tokovi generiraju promet kako je to prikazano na slikama, s tim da na lijevoj slici svi tokovi poštivaju svoje specifikacije, dok na desnoj slici tokovi 2 i 3 ih ne poštivaju. Pretpostaviti da je za prijenos jednog paketa potrebna jedna vremenska jedinica.



Rješenje

Algoritam virtualnog sata pokušava emulirati sustav u kojemu se za prijenos upotrebljava TDM. U slučaju takvog sustava paketi prikazani na slikama bi imali slijedeći redoslijed slanja:

Algoritam virtualnog sata za svaki pristigli paket računa pridodjeljenu varijablu stanja. Ta varijabla stanja je u stvari vrijeme kada paket treba napustiti sustav u kojemu se upotrebljava TDM. Varijabla stanja se računa na osnovu sljedećeg izraza:

$$auxVC_{i,j}^{k} \leftarrow max\{a_{i,j}^{k}, auxVC_{i,j}^{k-1}\} + Vtick_{i,j}$$

auxVC, skraćeno od *auxiliary Virtual Clock*, je vrijednost koja se potom pridodjeljuje svakom paketu. U izrazu indeks *i* označava usmjernik u kojemu se obavlja ažuriranje varijable, indeks *j* označava tok na koji se varijabla odnosi i *k* označava pojedini paket toka. $a_{i,j}^{k}$ je vrijeme dolaska *k*-tog paketa toka *j* do preklopnika *i*. $Vtick_{i,j}$ se izračunava na osnovu prosječne količine prometa koju generira *j*-ti tok.

Kako se u ovom primjeru prati izvršavanje algoritma samo unutar jednog usmjernika nije nužno voditi evidenciju o usmjerniku, te se gornji izraz malo pojednostavljuje:

$$auxVC_{j}^{k} \leftarrow max\{a_{j}^{k}, auxVC_{j}^{k-1}\} + Vtick_{j}$$

Prema uvijetima danim u primjeru prvi tok je specificirao prosječno 0.5 paketa/s pa je $Vtick_1 = 2$, a tokovi 2 i 3 su specificirali 0.2 paketa/s pa je njihov $Vtick_2 = Vtick_3 = 5$. Inicijalno svi imaju *auxVC* jednak nuli. Sada dolazak i odlazak paketa možemo pratiti uz pomoć tablice 11.

Prva kolona tablice sadrži informaciju o vremenskom trenutku na koji se redak odnosi. Dodatno se u tablici nalaze podaci za svaki pojedini tok – nailazak pojedinog paketa (kolona F) i izračunati $auxVC_{j}^{k}$. Vrijeme počinje u nekom nultom trenutku i sve varijable *auxVC* prije tog trenutka postavljamo na nulu. U svakoj koloni tablice sa zaglavljem F_i vodi se evidencija o prispjeću paketa za tok *i*. Primjerice, kada je jedinica u retku koji pripada trenutku 2 i stupcu sa zaglavljem F_1, to znači da je u 2. trenutku pristigao paket koji pripada prvom toku i da je paket veličine 1 jedinice.

Nakon što pristigne svaki paket, računa se novi *auxVC* te se njegova vrijednost pridodjeljuje paketu. U tablici nisu posebno vođene *auxVC* vrijednosti za pojedini paket budući da se oni lako iščitaju. Potrebno je samo za odgovarajući paket pogledati ćeliju desno od njega i to je vrijednost *auxVC*-a za taj paket. Za drugi paket 1. toka, koji dolazi u 2. trenutku, vrijednost varijable a_1^2 je 2. Pa je prema tome $auxVC_1^2$:

$$auxVC_1^2 \leftarrow max\{a_1^2, auxVC_1^1\} + Vtick_1 = max\{2,2\} + 2 = 4$$

Analogan postupak računanja primjenjen je i za svaki drugi pristigli paket svih tokova. Kada preklopnik treba poslati neki paket on pregledava sve pakete koji su tog trenutka u spremniku i šalje paket koji ima najmanji *auxVC*. U slučaju da je više takvih paketa, proizvoljno je odabrano da se šalje onaj paket čiji indeks toka je manji. U koloni za izlaz iz mrežnog uređaja koristi se notacija *i*(*T*). *i* je indeks toka čiji paket se šalje, a u zagradi je vrijeme kada je taj paket pristikao u preklopnik. Primjerice, u 7. trenutku šalje se paket označen s 3(5), tj. šalje se paket 3. toka koji je u preklopnik pristigao u 5. trenutku.

Tablica 11. Izlazi iz usmjernika kada svi tokovi poštivaju svoje specifikacije

T	***F₁***	***auxVC₁***	***F₂***	***auxVC₂***	***F₃***	***auxVC₃***	***Izlaz***
		0		0		0	
0	1	2	1	5	1	5	1(0)
1							2(0)
2	1	4					1(2)
3							3(0)
4	1	6					1(4)
5			1	10	1	10	2(5)
6	1	8					1(6)
7							3(5)
8	1	10					1(8)
9							
10	1	12	1	15	1	15	1(10)
11							2(10)
12	1	14					1(12)
13							3(10)
14	1	16					1(14)
15			1	20	1	20	2(15)
16	1	18					1(18)

U slučaju kada tokovi 2 i 3 ne poštivaju svoju specifikaciju, tj. šalju pakete daleko brže no što su naveli, tada izvršavanje algoritma daje izlaz iz usmjernika prikazan u tablici 12.

Tablica 12. Izlazi iz usmjernika kada tokovi 2 i 3 ne poštivaju svoje specifikacije

T	***F_1***	***$auxVC_1$***	***F_2***	***$auxVC_2$***	***F_3***	***$auxVC_3$***	***Izlaz***
		0		0		0	
0	1	2	1	5	1	5	1(0)
1							2(0)
2	1	4	1	10	1	10	1(2)
3							3(0)
4	1	6	1	15	1	15	1(4)
5							2(2)
6	1	8	1	20	1	20	1(6)
7							3(2)
8	1	10	1	25	1	25	1(8)
9							2(6)
10	1	12	1	30	1	30	1(10)
11							3(4)
12	1	14	1	35	1	35	1(12)
13							2(6)
14	1	16	1	40	1	40	1(14)
15							3(6)
16	1	18	1	45	1	45	1(16)

PRIMJER 21. Raspoređivanje paketa upotrebom algoritma WFQ

Rješenje

PRIMJER 22. Raspoređivanje paketa upotrebom algoritma WF^2Q

Rješenje

PRIMJER 23. Raspoređivanje paketa upotrebom algoritma SCFQ

Rješenje

PRIMJER 24. Raspoređivanje paketa upotrebom algoritma SFQ

Rješenje

PRIMJER 25. Raspoređivanje paketa upotrebom algoritma STFQ

Rješenje

10. Literatura

[RFC0791] Postel, John, *Internet protocol – Darpa Internet Protocol Specification*, RFC791, IETF, September 1981.
http://www.ietf.org/rfc/rfc0791.txt?number=791

[RFC0792] Postel, John, *Internet Control Message Protocol – Darpa Internet Program Protocol Specification*, RFC792, IETF, September 1981.
http://www.ietf.org/rfc/rfc0792.txt?number=792

[RFC1058] Hedrick, C. L., *Routing Information Protocol*, RFC1058, IETF, June 1988.
http://www.ietf.org/rfc/rfc1058.txt?number=1058

[RFC1631] Egevang, K., Francis, P., *The IP Network Address Translator (NAT)*, RFC1631, IETF, May 1994.
http://www.ietf.org/rfc/rfc1631.txt?number=1631

[RFC1701] Hanks, S., Li, T., Farinacci, D., Traina, P., *Generic Routing Encapsulation*, RFC1701, IETF, October 1994.
http://www.ietf.org/rfc/rfc1701.txt?number=1701

[RFC1702] Hanks, S., Li, T., Farinacci, D., Traina, P., *Generic Routing Encapsulation over IPv4 Networks*, RFC1702, IETF, October 1994.
http://www.ietf.org/rfc/rfc1702.txt?number=1702

[RFC2003] Perkins, C., *IP Encapsulation within IP*, RFC2003, IETF, October 1996.
http://www.ietf.org/rfc/rfc2003.txt?number=2003

[RFC2453] Malkin, G., *RIP Version 2*, RFC2453, IETF, November 1998.
http://www.ietf.org/rfc/rfc2453.txt?number=2453

[RFC2460] Deering, S., Hinden, R., *Internet Protocol, Version 6 (IPv6) – Specification*, RFC2460, IETF, December 1998.
http://www.ietf.org/rfc/rfc2460.txt?number=2460

[RFC3513] Hinden, R., Deering, S., *Internet Protocol Version 6 (IPv6) Addressing Architecture*, RFC3513, IETF, April 2003.
http://www.ietf.org/rfc/rfc3513.txt?number=3513

[ZHAN95] Zhang, Hui, *Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks*, Proc. of the IEEE, Vol. 83, No. 10, October 1995, pp 1374-1396.

[ZHAN90] Zhang, Lixia, *VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks*, 1990.

[MICH98] Michele, Clark, *Comparison of Real-Time CPU Scheduling and Proportional Share Packet Scheuduling*, COMP291, April 30, 1998.

[BENN96] Bennet, J. C. R., Zhang, H., *WF2Q: Worst-case Fair Weighted Fair Queueing*, INFOCOM, 1996, pp. 120-128.

[MCKE91] McKenny, P. E., *Stochastic Fairness Queuing*, 1991.

[GOYA96] Goyal, P., Vin, H. M., Cheng, H., *Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks*, Technical Report TR-96-02, Department of Computer Sciences, University of Texas at Austin.

[GOYA??] Goyal, P., Guo, X., Vin, H. M., *A Hierarchical CPU Scheduler for Multimedia Operating Systems*.

[AURR??] Aurrecoechea, C., Campbell, A., Hauw, L., *A Survey of Quality of Service Architectures*, Columbia University,

[RFC2391] Load Sharing using IP Network Address Translation (LSNAT). P. Srisuresh, D. Gan. August 1998. (Format: TXT=44884 bytes) (Status: INFORMATIONAL)

[RFC2428] FTP Extensions for IPv6 and NATs. M. Allman, S. Ostermann, C. Metz. September 1998. (Format: TXT=16028 bytes) (Status: PROPOSED STANDARD)

[RFC2663] IP Network Address Translator (NAT) Terminology and Considerations. P. Srisuresh, M. Holdrege. August 1999. (Format: TXT=72265 bytes) (Status: INFORMATIONAL)

[RFC2709] Security Model with Tunnel-mode IPsec for NAT Domains. P. Srisuresh. October 1999. (Format: TXT=24552 bytes) (Status: INFORMATIONAL)

[RFC2766] Network Address Translation - Protocol Translation (NAT-PT). G. Tsirtsis, P. Srisuresh. February 2000. (Format: TXT=49836 bytes) (Updated by RFC3152) (Status: PROPOSED STANDARD)

[RFC2993] Architectural Implications of NAT. T. Hain. November 2000. (Format: TXT=74136 bytes) (Status: INFORMATIONAL)

[RFC3022] Traditional IP Network Address Translator (Traditional NAT). P. Srisuresh, K. Egevang. January 2001. (Format: TXT=37675 bytes) (Obsoletes RFC1631) (Status: INFORMATIONAL)

[RFC3235] Network Address Translator (NAT)-Friendly Application Design Guidelines. D. Senie. January 2002. (Format: TXT=29588 bytes) (Status: INFORMATIONAL)

[RFC3489] STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy. March 2003. (Format: TXT=117562 bytes) (Status: PROPOSED STANDARD)

[RFC3519] Mobile IP Traversal of Network Address Translation (NAT) Devices. H. Levkowetz, S. Vaarala. May 2003. (Format: TXT=80522 bytes) (Status: PROPOSED STANDARD)

[RFC3715] IPsec-Network Address Translation (NAT) Compatibility Requirements. B. Aboba, W. Dixon. March 2004. (Format: TXT=43476 bytes) (Status: INFORMATIONAL)