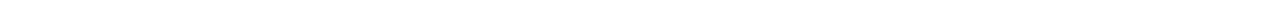


Sadržaj

1. Uvod.....	2
2. Ethernet.....	3
2.1. Osnove Ethernet mreža.....	4
2.1.1. Adresiranje u Ethernet mrežama.....	4
2.1.2. Ethernet okviri.....	5
2.1.3. Pretvaranje IP adresa u Ethernet adrese.....	6
2.2. Topologije Ethernet mreža.....	7
2.2.1. Sabirnička struktura.....	7
2.2.2. Zvezdasta struktura.....	7
2.3. Električni signali na Ethernet fizičkom mediju.....	7
2.4. Ethernet koncentratori.....	7
2.5. Premosnici.....	8
2.5.1. Virtualne lokalne mreže.....	10
2.5.2. Trunking.....	10
2.5.3. Protokol razapinjućeg stabla.....	10
2.5.4. Kontrola korištenja pristupa preklopnika.....	10
2.6. Pojednostavljeni model premosnika.....	11
3. Primjeri.....	14
PRIMJER 1. Prosljeđivanje okvira u premosnicima.....	14
PRIMJER 2. Tablice prosljeđivanja premosnika.....	15
PRIMJER 3. Izgradnja premosnog stabla.....	16
PRIMJER 4. Izgradnja premosnog stabla.....	17
4. Literatura.....	19

1. Uvod

Evolucija Interneta.



2. Ethernet

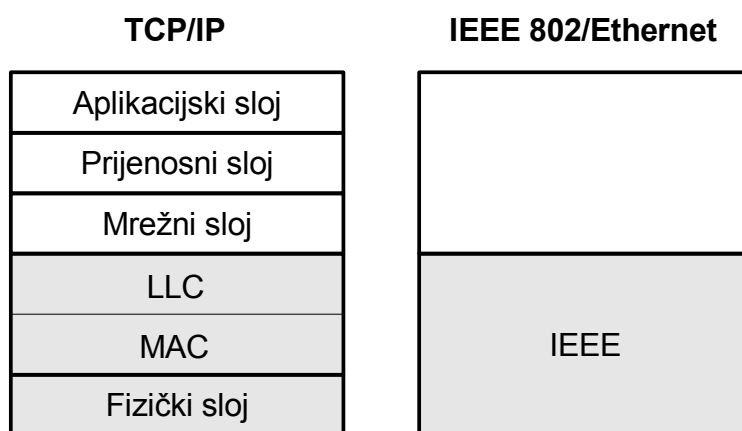
Ethernet je danas najzastupljenija računalna mreža koja se koristi za izgradnju lokalnih i gradskih mreža, a u novijim inačicama namjera je da se njena upotreba proširi i na izgradnju WAN mreža. To nije neobično kada se uzme u obzir raznolikost Ethernet mreža, primjerice brzine Ethernet mreža se kreću u rasponu od 10Mb/s do 10Gb/s, a na fizičkom sloju mogu se koristiti bakreni ili optički vodiči, odnosno radio valovi. Cilj ovog teksta je detaljno upoznavanje čitatelja s Ethernetom i s Ethernet tehnologijama.

Prije početka rasprave nužno je obratiti pozornost na terminologiju. Naime, postoje dvije međusobno nekompatibilne inačice Etherneta. Dominantna je inačica poznata pod imenom Ethernet II, proistekla iz specifikacije Etherneta poznatoj po imenu DIX Ethernet (DIX – DEC, Intel, Xerox). Osim DIX Etherneta dosta je poznata i korištena i IEEE inačica koja bi se trebala referencirati kao IEEE802.3. Ipak, u daljnjem tekstu će se koristiti izraz Ethernet za sve vrste te mreže, a u slučaju kada je to potrebno eksplicitno će biti naglašena razlika.

Na slici 1 prikazan je TCP/IP model i na njemu je označeno mjesto gdje se nalazi Ethernet. Ethernet zauzima prva dva sloja, tj. fizički i podatkovni sloj.

Na fizičkom sloju Ethernet standard propisuje priključke, fizikalne karakteristike vodiča i signala na njemu.

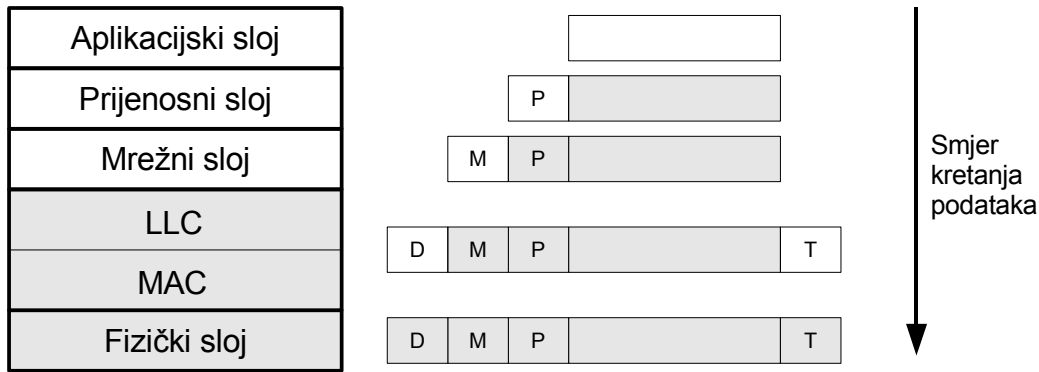
Zbog raznolikosti fizičkog sloja koje Ethernet podržava, podatkovni sloj podijeljen je u dva podsloja. Viši sloj je podsloj pristupa logičkom sloju (engl. Logical Link Control, LLC) i on je neovisan o mediju, niži podsloj je podsloj za pristup prijenosnom mediju (engl. Medium Access Control, MAC).



Slika 1. Mjesto IEEE802/Ethernet standarda u TCP/IP referentnom modelu

Tok podataka između slojeva može biti prema dolje ili prema gore, u ovisnosti da li se radi o slanju podataka (prema dolje), ili primanju podataka (prema gore). Slika 2 prikazuje situaciju tijekom slanja podataka. Aplikacijski sloj proslijedi podatke prijenosnom sloju. Prijenosni sloj na te podatke dodaje svoje kontrolne informacije, tj. svoje zaglavlje, i potom proslijedi te podatke mrežnom sloju. Mrežni sloj također dodaje svoje zaglavlje i predaje ga podatkovnom sloju. Podatkovni sloj osim zaglavlja dodaje i određenu kontrolnu informaciju na kraj svih podataka, te potom prepušta tako nastali niz bitova fizičkom sloju na stvarni prijenos do drugog računala.

Jedna od bitnih informacija u zaglavlju je odredišna adresa. U principu, svaki sloj može imati svoj način adresiranja neovisan o ostalim slojevima. Međutim, u stvarnosti određeni slojevi koriste adresiranje nižih slojeva te na to dodaju još neke svoje informacije.

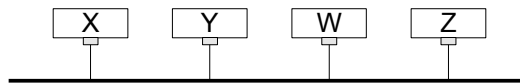


Slika 2. Kretanje podataka tijekom slanja u TCP/IP referentnom modelu

Tijek podataka prikazan na slici 2 predstavlja samo jedan mogući scenario. Naime, vrlo lako je moguće da aplikacijski sloj, ili bilo koji drugi sloj, proslijedi nižem sloju više podataka no što ga on može poslati u jednom navratu. U tom slučaju potrebno je raspodijeliti podatke u više jedinica koje se potom prepuštaju nižem sloju. Primjerice, neka je prijenosni sloj predao mrežnom sloju preveliku količinu podataka da bi on to mogao proslijediti u jednom paketu. U tom slučaju, slika , mrežni sloj razbija podatke prijenosnog sloja i prosljeđuje ih nižem sloju. Tijekom primanja podataka odigrava se suprotan proces. Mrežni sloj prikuplja razdvojene podatke, rekreira

2.1. Osnove Ethernet mreža

Za početak pretpostavimo da su nam na raspolaganju četiri računala na Ethernet mreži. Slika 3 simbolički prikazuje tu situaciju. Računala su označena slovima X, Y, W i Z.

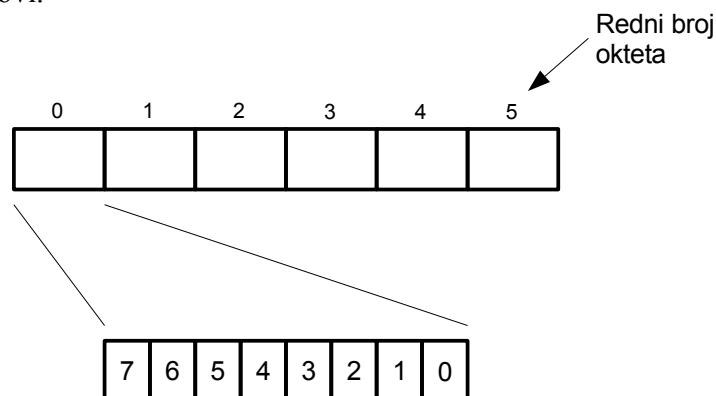


Slika 3. Simbolički prikaz Ethernet mreže sa četiri računala

Za prvo razmatranje zanemariti ćemo točnu implementaciju mreže, tj. njen fizički izgled. Također ćemo pretpostaviti da sve će što neko računalo pošalje biti i isporučeno. Ta pretpostavka je kriva, ali kasnije će biti ispravljena.

2.1.1. Adresiranje u Ethernet mrežama

U Ethernet mrežama koristi se adresa veličine 6 okteta, odnosno 48 bita. Shematski prikaz adrese dan je na slici 4. Tijekom prijenosa adrese preko mreže prvo se šalje oktet sa rednim brojem nula, a zadnji se šalje oktet sa rednim brojem 5. Tijekom slanja svakog okteta na 802.3 mreži prvo se šalje nulti bit, te redom svi ostali bitovi.



Slika 4. Shematski prikaz Ethernet adrese

Standardni način prikaza Ethernet adresa je po oktetima, u heksadecimalnoj bazi pri čemu su najčešće pojedini okteti razdvojeni dvotočkom. Primjer jedne Ethernet adrese je:

00:0C:30:4A:51:34

Navedeni simbolički prikaz Ethernet adrese nije jedinstven u smislu da se za razdvajanje okteta koriste i drugi simboli, točka, minus, itd. Također, određeni proizvođači samo u sredini adrese upotrebljavaju separator. Primjerice,

000C30-4A5134

Bez obzira na načine zapisa uvijek je u prikazu 12 heksadecimalnih znamenki.

Politika je da svako Ethernet sučelje ima svoju Ethernet adresu. U računalu s jednom mrežnom karticom postoji samo jedna Ethernet adresa vezana za tu karticu. U slučaju da postoje dvije kartice tada će biti dvije Ethernet adrese; po jedna za svaku karticu. Ethernet adrese su obično upisane u ROM na mrežnoj kartici, no bez obzira na to moguće ih je programski mijenjati što može stvoriti određene sigurnosne probleme.

Dodjeljivanje Ethernet adresa nije proizvoljno. Proizvođač mrežnog uređaja traži od IEEE-a grupu Ethernet adresa. Svaka grupa Ethernet adresa ima fiksirana prva tri okteta. Druga tri okteta proizvođač proizvoljno određuje, ali na takav način da ne postoje dva uređaja sa istom adresom. Jednom proizvođaču može biti dodjeljeno više grupa adresa, no jedna grupa adresa pripada isključivo jednom proizvođaču.

Između svim mogućih kombinacija Ethernet adresa određen broj ima posebno značenje. Prva adresa tog tipa je difuzna (engl. broadcast) adresa. Karakteristična je po tome da sadrži sve jedinice, drugim riječima, ta adresa ima sljedeći oblik:

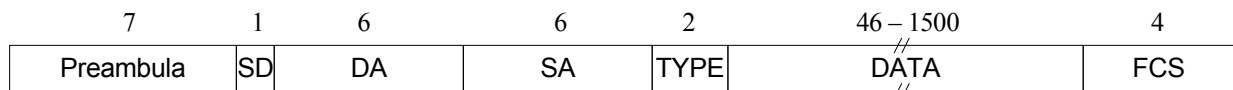
ff:ff:ff:ff:ff:ff

Svaki okvir poslan s tom odredišnom adresom isporučuje se svim računalima na mreži. Osim difuzne posebnu grupu predstavljaju i adrese za difuziju u grupi (multicast) te privatne adrese. Adresama koje se koriste za difuziju u grupi postavljen je najniži bit prvog okteta na jedinicu. Kada se određeni okvir šalje na tu adresu tada ga može primiti nijedna, jedna ili više stanica – ovisno o tome tko je u tom trenutku član te grupe. Privatne adrese prepoznaju se po postavljenom drugom bitu (odmah do bita za difuziju u grupu). Postavljenost tog bita znači da je adresa privatna, te da nije globalno dodijeljena od strane IEEE-a.

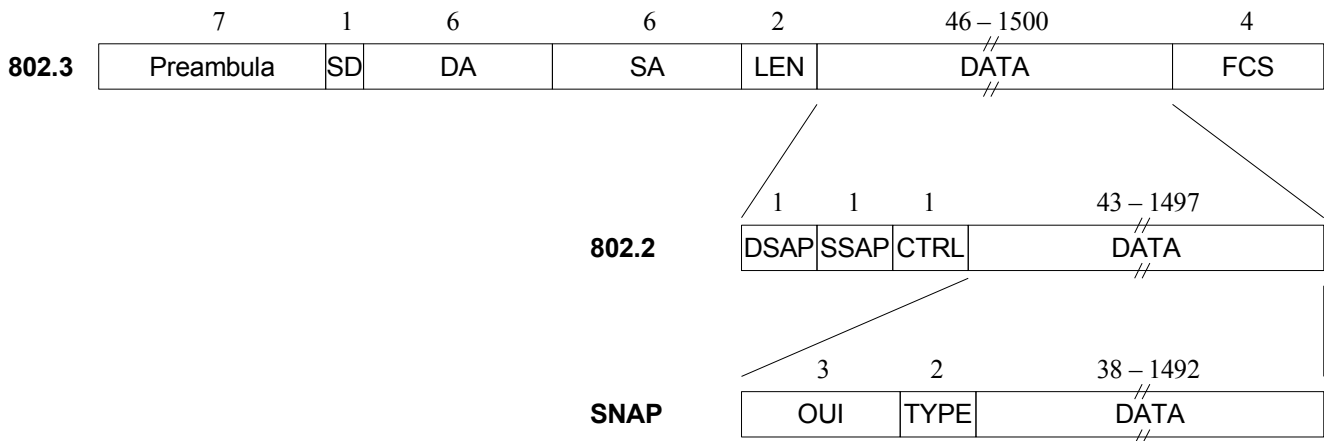
2.1.2. Ethernet okviri

Sva razmjena podataka na Ethernet mrežama obavlja se upotrebom okvira. Veličina okvira određena je nekim fizičkim karakteristikama upotrebene tehnologije i upotrebljenog protokola za pristup prijenosnom mediju. Za sada je dovoljno reći kako je minimalna ukupna veličina okvira 64 okteta, a maksimalna veličina 1518 okteta. Te veličine ne uzimaju u obzir preambulu, tj. početni dio okvira koji se koristi za sinkronizaciju prijemnika i predajnika.

Točan format okvira ovisi o standardu, iako oba formata mogu egzistirati na istoj mreži. Slika 5 prikazuje format Ethernet II okvira, a slika 6 IEEE802.3 format. Razlika je isključivo u polju TYPE odnosno LEN. Kod Ethernet II okvira informacija o protokolu koju se nalazi u podatkovnom polju upisana je u TYPE polje. Kod IEEE802.3 okvira umjesto protokola upisana je duljina okvira, a podatak o protokolu nalazi se na početku podataka zapisan u LLC zaglavlju.



Slika 5. Format Ethernet II okvira

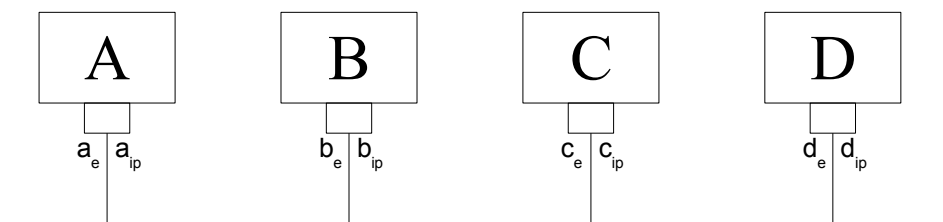


Slika 6. Format okvira prema IEEE802

2.1.3. Pretvaranje IP adresa u Ethernet adrese

U trećem sloju TCP/IP referentnog modela koriste se IP adrese. Kada mrežni sloj treba poslati neki paket on na njega doda svoje zaglavlje i prosljedi ga podatkovnom sloju. Podatkovni sloj treba tom paketu dodati svoja zaglavlja i preko LAN mreže prebaciti ga na odredište. Problem je: kako iz IP adrese dobiti Ethernet adresu?

Taj problem rješava ARP protokol. Namjena tog protokola je da za zadanu IP adresu odredi Ethernet adresu. To se radi uz pomoć difuzije. Uzmimo za primjer sliku 7.



Slika 7. Slika uz objašnjenje ARP protokola

Na slici su prikazana računala (A, B, C i D), i mrežna kartica svakog računala. Mrežna kartica ima dvije adrese. Prva je Ethernet adresa (ta adresa ima indeks e), a druga je IP adresa (indeks ip). Uzmimo da računalo A treba poslati nekakve podatke na adresu d_{ip} . Budući da se okvir ne može poslati dok nije poznata odredišna adresa računalo A mora saznati tu adresu. U tom slučaju računalo A šalje svim računalima okvir simbolički prikazan na sljedeći način:

$$\text{Eth}(\text{BROADCAST}, a_e, \text{ARPReq}(d_{ip}, a_{ip}))$$

Taj prikaz sadrži suštinu informacija koju računalo A šalje u okviru. Za početak, vidimo da se radi o Ethernet okviru. Taj Ethernet okvir upućen je na difuznu adresu (BROADCAST) što znači da ga

trebaju prihvatiti svi na mreži. Pošiljalac se je identificirao sa svojom Ethernet adresom (a_e). Adresa pošiljalca je nužna jer u suprotnom nije moguće poslati odgovor. Okvir je prikazan na slici .

Svako računalo na mreži koje primi taj upit provjerava da li se radi o njegovoj IP adresi. Samo računalo koje prepozna svoju adresu odgovara. Odgovor ćemo prikazivati na sljedeći način:

$$\text{Eth}(a_e, d_e, \text{ARPResp}(a_{ip}, d_{ip}))$$

Dakle, odgovor je upućen isključivo računalu koje je poslalo upit, te ostala računala ne mogu vidjeti odgovor.

2.2. Topologije Ethernet mreža

2.2.1. Sabirnička struktura

Prvotna Ethernet topologija bila je sabirnička struktura. Sva računala bila su spojena na jedan fizički vod. Shematski to je prikazano na sljedećoj slici:

Na toj slici shematski su prikazana četiri računala (X, Y, W i Z) spojena na jedan vodič. Prvotno je taj vodič bio debeli Ethernet, ili oznakama 10Base5. Kasnije je kao fizički vodič upotrebljavan tanki Ethernetom, tj. 10Base2. U oba slučaja kada neka stanica šalje okvir taj okvir vide sva računala priključena na kabel, ali ga preuzima samo ono kome je taj okvir namijenjen. Postoje dvije iznimke tog pravila. Prva iznimka su tzv. difuzni okviri (broadcast). Druga iznimka su kartice koje se nalaze u tzv. *promiskuitetnom* načinu rada.

Zbog činjenice da sva računala vide sav promet, te da kolizija dva računala onemogućava korištenje prijenosnog medija ostalim računalima, za takvu mrežu kaže se da se nalazi u jednoj kolizijskoj domeni.

Ta topologija iskazala je tijekom upotrebe niz problema. Neki od tih problema su:

- U jednom trenutku može slati isključivo jedno računalo, sva ostala računala moraju čekati da se medij oslobodi.
- U slučaju prekida kabla teško je odrediti mjesto gdje je kvar nastao.

Zbog tih razloga sabirnička topologija je napuštena i može se pronaći jedino kao ostatak prethodnih generacija mreža iako je moguće da se još uvijek upotrebljava za posebne namjene.

2.2.2. Zvezdasta struktura

U današnje vrijeme apsolutno je dominantna topologija bazirana na zvijezdi. U ovom slučaju možemo govoriti o dvije generacije zvezdastih topologija. Prva generacija u centru zvijezde imala je koncentrador, dok se u novijim generacijama isključivo upotrebljavaju preklopnici. Oba uređaja detaljnije su objašnjena u nastavku.

2.3. Električni signali na Ethernet fizičkom mediju

2.4. Ethernet koncentratori

Bitnu komponentu Ethernet lokalnih mreža zvezdaste topologije čine koncentratori (*hubs*) i prenosnici (*bridges*). Funkcionalnost tih dviju komponenti definirana je IEEE standardima. Ipak, zbog niza prednosti u odnosu na koncentratore prenosnici su danas dominantni i o njima će biti najviše riječi.

Koncentratori su vrlo jednostavni uređaji koji rade na fizičkom sloju ISO/OSI modela. Za njih su svi okviri samo niz električnih impulsa koji nemaju svoje značenje ni strukturu. Funkcionalnost tih uređaja svodi se na jednostavno povezivanje svih ulaza i izlaza, te oni ne nude skoro nikakve dodatne pogodnosti u odnosu na sabirničku strukturu prvotnih Ethernet mreža. Dakle, iako je sada fizička topologija mreže zvijezda, logički, ona je i dalje sabirnica.

Za mreže koje se baziraju na ovim uređajima vrijede pravila CSMA/CD pristupa mediju. Sva računala spojena putem koncentratora nalaze se u jednoj *domeni kolizije*. U jednoj domeni kolizije, kada dva ili više računala pokušaju istovremeno poslati okvir, dolazi do kolizije koju vide sva računala priključena na koncentrator.

Primjena ovih uređaja ipak je riješila određene probleme sabirničke topologije. Najznačajniji dobitak je veća robusnost na pogreške vodiča. Međutim i dalje ostaje problem da samo jedna stanica može slati dok ostale vide taj promet i čekaju da se medij oslobodi kako bi se natjecale za pristup tom mediju.

2.5. Premosnici

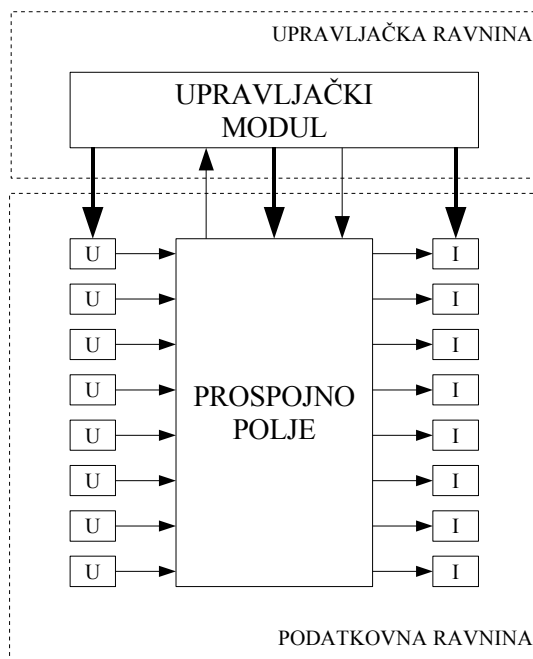
Za razliku od koncentratora, premosnici (engl. bridges) rade na 2. sloju ISO/OSI modela. Prema tome, ti uređaji poznaju Ethernet okvire i mogu vršiti optimizacije na osnovu analiziranog sadržaja. U računalnim mrežama koje se temelje na ovim uređajima ne koristi se više CSMA/CD pristup mediju, pa prema tome ne vrijede ni ograničenja na veličinu koja povlači taj protokol pristupa mediju! Također, uređaji sada mogu istovremeno i primati i slati podatke.

IEEE standard definira isključivo premosnike, dok proizvođači gotovo isključivo upotrebljavaju izraz preklopnik (engl. switch). Međutim, osim tog izraza koristi se i niz drugih. U vrijeme nastanka, premosnici su bili skupi uređaji s malom količinom pristupa, a kompletnu funkcionalnost obavljala je programska podrška. Razvojem mikroelektronike sve više funkcionalnosti prebacivano je u sklopovlje te su mogućnosti premosnika rasle, a cijena se smanjivala. Današnji premosnici mogu imati velik broj portova, 8, 12, 24 i više. Kako bi se ipak razlikovali ti premosnici od prvotnih za njih se koristi i u ovom radu će se koristiti izraz preklopnik.



Premosnici rade na takav način da povezuju isključivo računala koja međusobno komuniciraju i to povezivanje obavljaju na takav način da oba računala istovremeno mogu i primati podatke i slati. Takvim načinom rada efektivna brzina kroz svaki mrežni priključak je 200 Mb/s.

Na slici 8 prikazan je model preklopnika. Preklopnik se sastoji od dva dijela. Prvi dio je podatkovni (engl. data plane), dok je drugi dio upravljački (engl. control plane). Zahtjevi na ta dva dijela su potpuno oprečni. Podatkovni dio obrađuje svaki pristigli okvir pa se prema tome od njega traži što veća efikasnost kako bi preklopnik mogao raditi na što većim brzinama. Upravljački dio s druge strane prihvaća samo neke okvire i na osnovu njih odlučuje o radu preklopnika.

Podatkovna ravnina sastoji se od ulaza, prospojnog polja i izlaza. Iako ovdje ulaze i izlaze prikazujemo zasebno, u Ethernet preklopnicima oni idu u parovima. Drugim riječima, svaki Ethernet konektor sadrži po jedan ulaz i izlaz. Kada okvir dođe na neki od ulaza određuje se izlaz na koji ga je potrebno uputiti. Zadatak prospojnog polja je efikasno povezivanje ulaza i izlaza. Primjetimo kako to nije jednostavan zadatak. Moguće je da na dva ulaza dođu dva okvira koja trebaju ići na isti izlaz, u tom slučaju potrebno je jedan od njih propustiti a drugi privremeno pohraniti u nekakav spremnik. Također, moguća je situacija u kojoj na ulaz dolazi okvir koji je potrebno proslijediti na više različitih izlaza. Primjerice, okviri s difuznom adresom moraju se proslijediti na sve izlaze. Čak štoviše, od prospojnog polja se zahtijeva istovremeno prospajanje više ulaza i izlaza kako bi preklopnik bio što je moguće efikasniji.



Legenda

	Upravljačka sabirnica	U - ulaz
	Podatkovna sabirnica	I - izlaz

Slika 8. Model Ethernet preklopnika

Ovisno da li prenosnici prihvaćaju cijeli okvir prije njegova prosljeđivanja možemo ih podijeliti na *pohrani i prosljedi* (engl. store and forward) tip preklopnika, *cut-through* (engl. cut-through) ili adaptivne preklopnike. Pohrani i prosljedi tip preklopnika prvo prihvaćaju cijeli okvir, a tek potom ga prosljeđuju na izlaz. Zbog toga što preklopnik vidi cijeli okvir prije no što ga pošalje, može provjeriti njegovu ispravnost te je prednost te vrste preklopnika u slučajevima kada ima dosta neispravnih okvira na mreži. Tada preklopnik ne prosljeđuje okvire s pogreškama i mreža je manje opterećena. Nedostatak je velika i varijabilna latencija, tj. vrijeme zadržavanja u preklopniku koje je relativno visoko i ovisi o duljini okvira. Druga krajnost su preklopnici koji odmah nakon što vide određenu adresu (dakle nakon preambule još šest okteta određene adrese) započinju prosljeđivanje na izlazni pristup. To su tzv. cut-through tipovi preklopnika. Njihova prednost je konstantna i vrlo niska latencija, no nedostatak je da se na izlaze prosljeđuju i neispravni okviri što smanjuje propusnost mreže. Današnji preklopnici su skoro isključivo adaptivni, drugim riječima u slučaju velikog broja pogrešaka na mreži funkcioniraju kao pohrani i prosljedi tip preklopnika, a u odsustvu pogrešaka ili kada je njihov broj vrlo malen, ponašaju se kao cut-through preklopnici.

Implementacija prospojnog polja dosta varira. Primjerice, ako preklopnik pokušavamo ostvariti na PC računalo tada će prospojno polje biti PCI sabirnica računala, a odluka o izlazu će se donositi na centralnom procesoru. Međutim, PCI sabirnica vrlo brzo postaje usko grlo. Za početak, ograničen je maksimalan broj mrežnih priključaka tako izvedenog preklopnika. U najbolje slučaju moguće je dobiti oko 26 mrežnih priključaka ako se uzmu mrežne kartice s četiri Ethernet priključka te se koristi PC računalo s 6 PCI slotova i eventualno jednim ili dva mrežna priključka na samoj matičnoj ploči. Međutim, daleko prije nego se dosegne maksimalan broj priključaka problem postaje sporost cijelog sustava. Povećanjem brzine Ethernet mreže i dodavanjem sve više mrežnih kartica PCI sabirnica i CPU postaju usko grlo cijelog uređaja. Zbog toga se većina prospojnih polja danas implementira u sklopovlju i preklopnici se općenito izvode kao zasebni mrežni uređaji.

Upravljački modul preklopnika omogućava nadzor njegova rada. Primjerice, moguće je isključiti pojedine pristupe, vidjeti promet po pristupima, grupirati ih, itd. Tu funkcionalnost ne omogućavaju svi preklopnici već samo tzv. *upravljivi preklopnici* (engl. managed switches). Pod upravljive preklopnike spadaju preklopnici izvedeni s PC računalom (recimo na Linux operacijskom sustavu) i zasebni preklopnici više klase. Kod neupravljivih preklopnika ne postoji upravljački dio, ili je on integriran na istom čipu kao i prospojno polje te omogućava samo najosnovniju funkcionalnost potrebnu za rad preklopnika.

Upravljivi preklopnici izvršavaju neku verziju operacijskog sustava koja omogućava spajanje na preklopnik. Vrlo poznat operacijski sustav takve vrste je Ciscov IOS (Internet Operating System), ali svaki veći proizvođač ima svoj vlasnički operacijski sustav. Korisničko sučelje preklopnika je komandna linija, sustav menija u tekstualnom modu, a novije vrijeme sve veću popularnost stiže i Web sučelje. Preklopnici više klase uključuju sve te varijante, dok preklopnici niže klase ne nude komandnu liniju.

2.5.1. Virtualne lokalne mreže

2.5.2. Trunking

2.5.3. Protokol razapinjućeg stabla

Ethernet preklopnici se u svom radu automatski prilagođavaju uvjetima u kojima se nalaze i uređajima koji su priključeni na njih. To za sobom povlači bitan problem petlji. Naime, ako se prilikom povezivanja mreže stvore kružne petlje može se desiti da određeni okviri počnu beskonačno kružiti po mreži. Još gora situacija je u slučajevima kada se zbog konfiguracije mreže okviri uvišestručavaju. U tom slučaju lako se može desiti da mreža postane potpuno neupotrebljiva.

Rješenje po kojemu bi se prilikom izgradnje i upotrebe mreže pazilo da se ne ostvare petlje nije dobro jer taj zadatak može postati vrlo kompleksan budući da su današnje mreže sve veće i veće kompleksnosti. Dodatno, u određenim slučajevima se namjerno stvaraju petlje, najčešće kako bi se ostvarila nekakva varijanta visoke raspoloživosti.

Kako bi se adekvatno riješio navedeni problem u sve kvalitetnije preklopnike ugrađen je protokol razapinjućeg stabla (Spanning Tree Protocol, STP), i u novije vrijeme, ubrzanog razapinjućeg stabla (Rapid Spanning Tree Protocol, RSTP). Naime, preklopnici stalno oglašavaju svoju nazočnost na sve lokalne mreže na koje su spojeni. Kada se na istoj lokalnoj mreži pojave dva preklopnika oni će pokušati ustanoviti da li postoji petlja i ako postoji podesiti će se na takav način da uklone tu petlju.

Pojmovi RSTP-a

Cilj RSTP-a

Parametri

Stanja

Algoritam

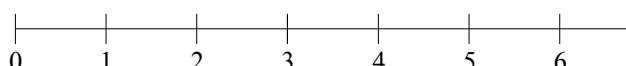
2.5.4. Kontrola korištenja pristupa preklopnika

802.1x

2.6. Pojednostavljeni model premosnika

Funkcionalnost koju preklopnici obavljaju relativno je jednostavna. Uprkos tome, preklopnici su vrlo složeni uređaji zbog uvijeta u kojima rade i zahtijeva koji se postavljaju na njih. Zbog toga nije moguće pratiti rad preklopnika u stvarnom vremenu, a također je to izuzetno teško učiniti precizno. Kako bi smo ipak opisali način na koji preklopnici rade, bez gubitka općenitosti, uvesti ćemo pojednostavljeni model.

Prvo pojednostavljenje koje ćemo uvesti je diskretizacija vremena. Naime, modeli koje ćemo opisivati obavljati će svoje funkcije isključivo u točno određenim vremenskim trenucima. Ako tijekom vremena prikažemo grafički kao na slici 9, tada se svi događaji odigravaju u trenucima označenim s



Slika 9. Vremenski trenuci kada se u preklopticima odvijaju sve operacije

brojevima 0, 1, 2, ... Između dva vremenska trenutka nema nikakvih događanja u mreži.

Kako bi mogli izolirati promet između računala preklopnici uče pristupe na kojima se nalaze pojedine Ethernet adrese. To drugim riječima znači da za svaki prihvaćeni okvir preklopnik pregledava izvorišnu adresu. Tu adresu potom upisuje u svoju memoriju i povezuje ju sa pristupom na koji je došao okvir. Pojednostavljena tablica u koju preklopnik pohranjuje viđene adrese ima sljedeći oblik:

Adresa	Pristup	Starost

U koloni *Adresa* nalazi se upisana viđena Ethernet adresa. U koloni *Pristup* je upisan redni broj pristupa na koji je pristigla adresa, a kolona *Starost* sadrži zadnji vremenski trenutak kada je prihvaćen okvir s navedenom adresom. Ako je primjerice, preklopnik u vremenskom trenutku 5 prihvatio Ethernet okvir s adresom A_e s pristupa 3, tada će u tablici, između ostalog, biti i sljedeći zapis:

Adresa	Pristup	Starost
A_e	3	5

Ako preklopnik u 7 vremenskoj jedinici prihvati okvir na isti pristup, i s istom izvorišnom adresom, tada će osvježiti zapis te će on tada glasiti:

Adresa	Pristup	Starost
A_e	3	7

Kolona *Starost* uvedena je kako bi preklopnik nakon nekog vremena “zaboravljao” naučene adrese. Zbog toga što je vrijeme diskretizirano javlja se problem redosljeda operacija u pojedinom vremenskom trenutku. Definirati ćemo proizvoljno da kada naiđe novi vremenski trenutak preklopnik prvo uklanja sve zastarjele zapise, a tek potom obavlja prosljeđivanje pristiglih okvira. Primjerice, ako je vrijeme zastare 3 vremenske jedinice, i do 10. trenutka nije prihvaćen više niti jedan okvir s izvorišnom adresom A_e , tada će u 10. trenutku preklopnik obrisati zapis za adresu A_e .

Dakle, na svaki označeni diskretni vremenski trenutak preklopnik prvo briše sve zastarijele zapise u tablici. Nakon toga prihvaća sve okvire, analizira njihov sadržaj i prosljeđuje ih na odgovarajuće izlaze.

Što se tiše protokola razapinjućeg stabla, uvesti ćemo sljedeća pojednostavljena. Kao prvo, prioritet svakog prenosnika biti će istovremeno i njegova oznaka odnosno identifikator. Prioriteti se određuju tako da se identifikatori leksički poredaju pa najviši prioritet ima prenosnik koji je prvi u tom poretku. Primjerice, ako imamo dva prenosnika s oznakama B1 i B2 tada veći prioritet ima prenosnik B1 budući da je on prvi po leksičkom poretku.

Nadalje, lokalne mreže ćemo označavati s velikim slovima abecede (A, B, C), a pojedine pristupe prenosnika s oznakom lokalne mreže na koju je spojen. Primjerice, ako prenosnik ima dva pristupa, od kojih je jedan spojen na mrežu A a drugi na mrežu C, tada će ti pristupi imati oznaku A i C. Ovim načinom označavanja ne možemo imati situacije u kojima prenosnik ima na jednu mrežu spojena dva pristupa.

Osim pretpostavki, postoji i određen broj pojmova koji se koristi. Ti pojmovi su:

- Glavni prenosnik. Glavni prenosnik ima ID najvećeg prioriteta.
- Cijena slanja preko pojedinog pristupa. To je administrativno postavljena vrijednost i često ovisi o brzini pristupa. U primjerima će se podrazumijevati da slanje na sve pristupe ima cijenu 1.
- Korijeni pristup. Svaki prenosnik na mreži je s jednim pristupom može dohvatiti korijeni prenosnik. Taj pristup naziva se korijeni pristup.
- Cijena do korijenog prenosnika. Ukupna cijena putovanja okvira od nekog prenosnika do korijenog prenosnika.
- Odabrani prenosnik/pristup. Pristup prenosnika koji prosljeđuje s i na određenu lokalnu mrežu naziva se odabrani pristup. Svaki prenosnik koji ima barem jedan odabrani pristup naziva se odabrani prenosnik.

Cilj algoritam je odrediti sljedeće:

- korijeni prenosnik i
- sve odabrane prenosnike i njihove pristupe.

Algoritam kreće od nekog inicijalnog stanja i izvršava se u sinkronim koracima. Prvo svi prenosnici šalju informacije na sve lokalne mreže, a zatim mijenjaju stanje na osnovu prihvaćenih informacija. Algoritam se izvršava dok se ne postigne stabilno stanje.

Analiza primljenih podataka teče sljedećim redoslijedom:

- Analiza korijenog pristupa, tj. prenosnik promatra da li koji od primljenih okvira sadrži prenosnik većeg prioriteta. Ako sadrži tada uzima taj prenosnik. U slučaju da ih više sadrži isti prenosnik tada uzima bliži korijenom prenosniku. Konačno, ako su svi isto udaljeni tada uzima prenosnik koji se nalazi na pristupu većeg prioriteta. U skladu s dobivenim informacijama ažurira svoje stanje.
- Nakon što je odredio korijeni pristup, prenosnik za sve ostale lokalne mreže na koje je spojen utvrđuje da li prosljeđuje podatke ili ne.

Kako bi mogli analizirati izvršavanje tog algoritma opet će se promatrati promjene u mreži sinkrono sa nekakvim zamišljenim davateljem takta. Prije opisa algoritma nužno je razjasniti određen broj pojmova i pretpostavki. Koristiti ćemo sljedeće pretpostavke:

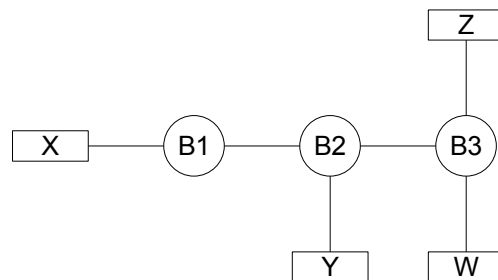
- Svaki prenosnik ima jedinstveni broj (ID). U stvarnosti taj broj nastaje od najmanje Ethernet adrese usmjernika i administrativno postavljenog prioriteta, no u zadacima ćemo za ID prenosnika koristiti njegovo ime. Leksički manje ime ima veći prioritet.
- Svaki pristup prenosnika ima svoj broj. U zadacima će se kao identifikator pristupa koristiti oznaka lokalne mreže na koju je spojen pristup prenosnika.
- Postoji posebna adresa, simboličke oznake ALLBRIDGES, koja adresira sve prenosnike na lokalnoj mreži. Također se okviri s tom adresom ne prosljeđuju na druge lokalne mreže.

3. Primjeri

PRIMJER 1. Prosljeđivanje okvira u premosnicima

Na slici su sa X, Y, Z i W označena računala, a s B1, B2 i B3 premosnici. U nekom inicijalnom trenutku sve tablice preklapanja premosnika su prazne. Odgovorite na sljedeća pitanja:

- a) X šalje paket Z-u. Koji premosnici uče gdje je X? Da li Y vidi paket?
- b) Z šalje paket X-u. Koji premosnici uče gdje je Z? Da li Y vidi paket?
- c) Y šalje paket X-u. Koji premosnici uče gdje je Y? Da li Z vidi paket?
- d) Z šalje paket Y-u. Koji premosnici uče gdje je Z? Da li W vidi paket?

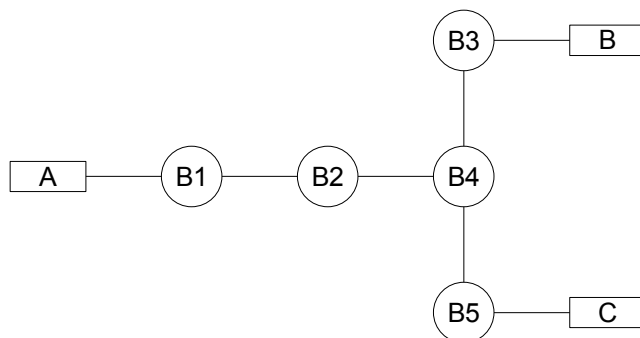


Rješenje

PRIMJER 2. Tablice prosljeđivanja prenosnika

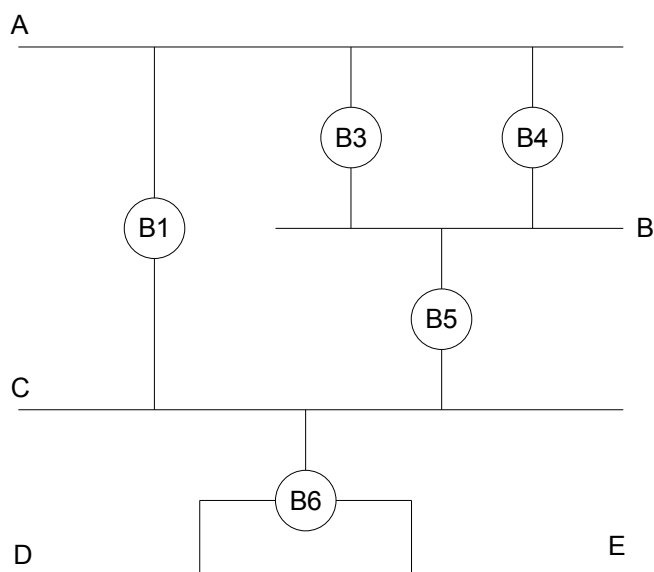
Na slici je zadana lokalna mreža sa prenosnicima B1 – B4 i računalima A, C i D. Vremensko ograničenje pojedinih zapisa u tablici je 2 vremenske jedinice. Uz pretpostavku da prenosnici započinju sa praznim tablicama preklapanja odredite sadržaj tablica nakon sljedećeg prometa:

- U 0. vremenskoj jedinici A šalje okvir C-u
- U 2. vremenskoj jedinici C šalje okvir A-u
- U 3. vremenskoj jedinici D šalje okvir C-u

**Rješenje**

PRIMJER 3. Izgradnja premosnog stabla

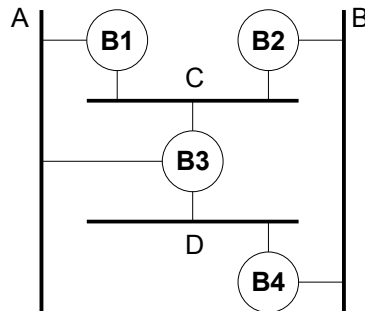
Za lokalnu mrežu prikazanu na slici odredite minimalno razapinjuće stablo koje formiraju preklopnici.



Rješenje

PRIMJER 4. Izgradnja prenosnog stabla

Za lokalnu mrežu prikazanu na slici odredite minimalno razapinjuće stablo koje formiraju preklopnici.

**Rješenje**

Odmah po uključenju prenosnici započinju s pretpostavkom kako je svatko od njih glavni prenosnik u mreži i da je prema tome zadužen za sve mreže koje su spojene na njega. Dakle, inicijalno stanje prikazano u tabličnom obliku je sljedeće:

<i>n</i>	B1	B2	B3	B4
0.	(B1, 0, -) A, C	(B2, 0, -) C, B	(B3, 0, -) A, C, D	(B4, 0, -) D, B

Prva iteracija algoritma započinje razmjenom informacija između prenosnika. Nakon obavljenje razmjene svaki od njih primio je sljedeće informacije:

<i>n</i>	B1	B2	B3	B4
1a.	(B2, 0, B2, C) (B3, 0, B3, C) (B3, 0, B3, A)	(B1, 0, B1, C) (B3, 0, B3, C) (B4, 0, B4, B)	(B1, 0, B1, A) (B1, 0, B1, C) (B2, 0, B2, C) (B4, 0, B4, D)	(B2, 0, B2, B) (B3, 0, B3, D)

Svaki od prenosnika potom traži minimalnu četvorku koju preuzima kao novi glavni prenosnik na mreži. U skup unutar kojega se traži minimalna četvorka uključeno je i trenutno stanje prenosnika. Dakle, traže se sljedeći minimum

$$B1: \min\{(B1, 0, B1, -), (B2, 0, B2, C), (B2, 0, B2, C)\} = (B1, 0, B1, -)$$

$$B2: \min\{(B2, 0, B2, -), (B1, 0, B1, C), (B3, 0, B3, C), (B4, 0, B4, B)\} = (B1, 0, B1, C)$$

$$B3: \min\{(B3, 0, B3, -), (B1, 0, B1, A), (B1, 0, B1, C), (B2, 0, B2, C), (B3, 0, B3, D)\} = (B1, 0, B1, A)$$

$$B4: \min\{(B4, 0, B4, -), (B2, 0, B2, B), (B3, 0, B3, D)\} = (B2, 0, B2, B)$$

Na osnovu dobivenih minimuma svaki prenosnik zapisuje svoje novo stanje, nova stanja prenosnika su:

<i>n</i>	B1	B2	B3	B4
1b.	(B1, 0, -)	(B1, 1, C)	(B1, 1, A)	(B2, 1, B)

Slijedi novi krug računanja minimuma kako bi odredio svoje stanje za sve preostale mreže, tj. mreže koje su ostale nakon što se izbacila mreža na kojoj je spojen na glavni prenosnik. Prilikom računanja novih minimuma svaki prenosnik više ne uzima u obzir svoje staro stanje, već novo, zatim ne uzima u obzir mrežu tj. pristup preko kojega je spojen na glavni prenosnik.

Dakle, za B2 preostala je mreža B. Minimum iz skupa svih paketa pristiglih s te mreže i njegova

vlastita stanja je dakle

$$B2.B: \min\{(B1, 1, B2, C), (B4, 0, B4, B)\} = (B1, 1, B2, C)$$

Njegovo stanje ispada minimalno te on odlučuje uzeti mrežu B. Slično razmatranje za prenosnik B3. B3 nakon što se ukloni pristup/mreža preko koje je spojen na glavni prenosnik mora odlučiti za mreže C i D. Za mrežu C imamo sljedeću situaciju:

$$B3.C: \min\{(B1, 1, B3, A), (B1, 0, B1, C), (B2, 0, B2, C)\} = (B1, 0, B1, C)$$

Dakle, minimalnu vrijednost ima BPDU koji je poslao prenosnik B1 pa B3 odustaje od mreže C. Za mrežu D imamo sljedeću situaciju:

$$B3.D: \min\{(B1, 1, B3, A), (B3, 0, B3, D)\} = (B1, 1, B3, A)$$

U ovom slučaju njegov BPDU je minimalan te on preuzima mrežu D. Konačno, isti postupak kada se ponovi za prenosnik B4:

$$B4.D: \min\{(B2, 1, B4, B), (B3, 0, B3, D)\} = (B2, 1, B4, B)$$

daje rezultat po kojemu B4 preuzima mrežu D jer je njegov BPDU minimalan.

Konačno stanje na kraju 1 iteracije algoritma je sljedeće:

<i>n</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>	<i>B4</i>
1.	(B1, 0, -) A,C	(B1, 1, C) B	(B1, 1, A) D	(B2, 1, B) D

U novoj iteraciji algoritma ponovo prenosnici razmjenjuju BPDU-ove te nakon razmjene svaki od njih ima sljedeći popis

<i>n</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>	<i>B4</i>
2a.		(B1, 0, B1, C)	(B1, 0, B1, A) (B1, 0, B1, C) (B2, 1, B4, D)	(B1, 1, B2, B) (B1, 1, B3, D)

B1 nije primio nikakvu novu informaciju te zadržava staro stanje. B2 je primio BPDU od prenosnika kojeg trenutno smatra glavnim u mreži te također ne mijenja stanje. B3 je primio obavijesti od B1 preko A i C mreža kao i u prethodnom koraku te ne mijenja stanje, a mrežu C otpušta. Konačno, dobio je novu obavijest na D mreži no ona je veća od njegove vlastite vrijednosti, tj.

$$B3.D: \min\{(B1, 1, B3, A), (B2, 1, B4, D)\} = (B1, 1, B3, A)$$

pa i dalje zadržava mrežu D. Konačno, prenosnik B4 dobije je dva potpuno nova BPDU te se za njega mora obaviti ispočetka kompletan izračun. Dakle, traži minimalni BPDU između svih pristiglih i vlastitog stanja kako bi odredio korijeni prenosnik:

$$B4: \min\{(B2, 1, B4, B), (B1, 1, B2, B), (B1, 1, B3, D)\} = (B1, 1, B2, B)$$

Dakle, mijenja stanje, a na osnovu novog stanja računa i svoje ponašanje na preostalim mrežama na koje je spojen, tj. na mreži D:

$$B4.D: \min\{(B1, 2, B4, B), (B1, 1, B3, D)\} = (B1, 1, B3, D)$$

Kako je BPDU drugog prenosnika manji prepušta mu mrežu D. Konačno stanje je:

<i>n</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>	<i>B4</i>
1.	(B1, 0, -) A,C	(B1, 1, C) B	(B1, 1, A) D	(B1, 2, B)

4. Literatura

- [RFC0791] Postel, John, *Internet protocol – Darpa Internet Protocol Specification*, RFC791, September 1981.
<http://www.ietf.org/rfc/rfc0791.txt?number=791>
- [RFC0792] Postel, John, *Internet Control Message Protocol – Darpa Internet Program Protocol Specification*, RFC792, ISI, September 1981.
<http://www.ietf.org/rfc/rfc0792.txt?number=792>
-