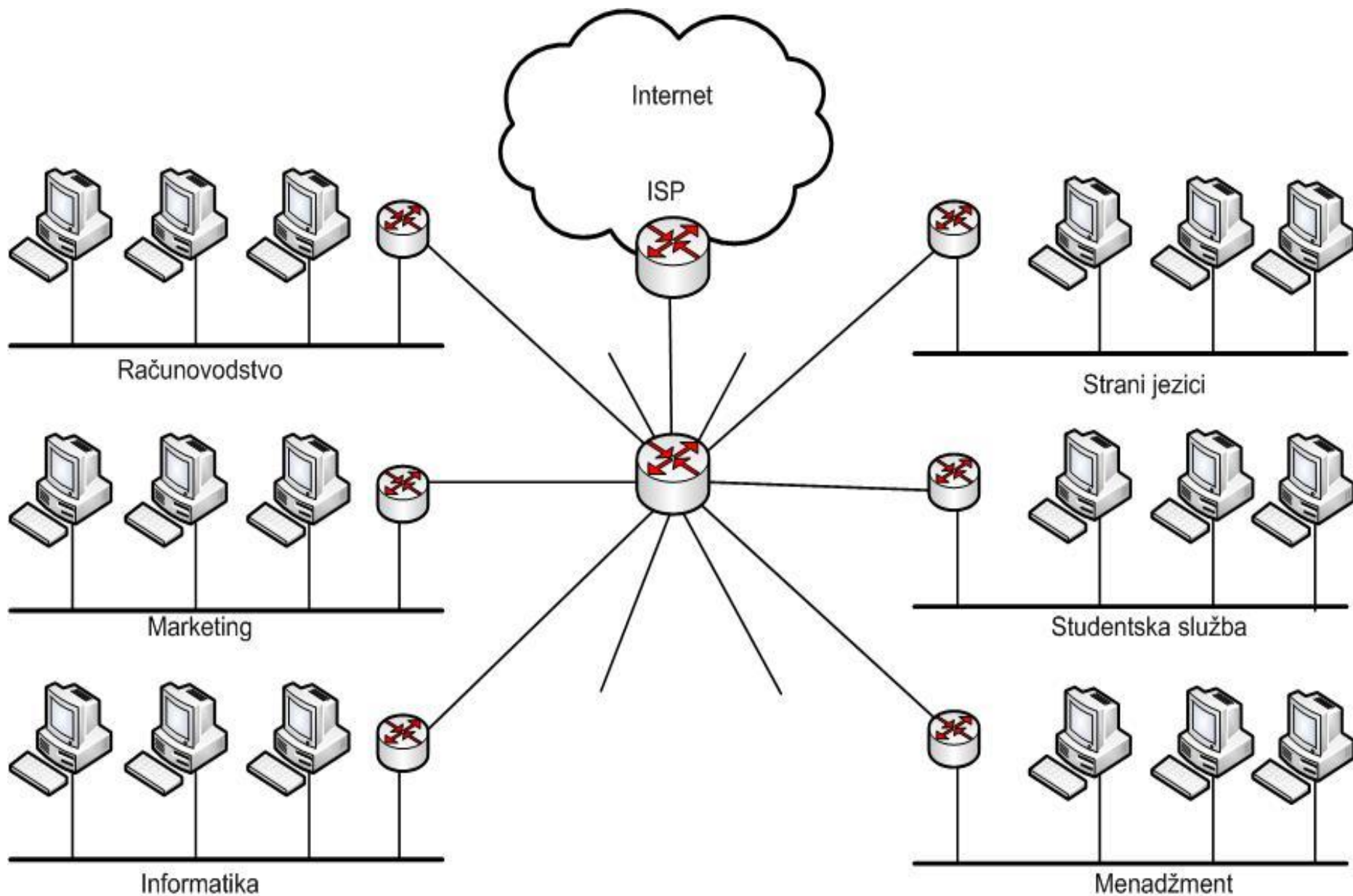


# IX - Povezivanje mreža

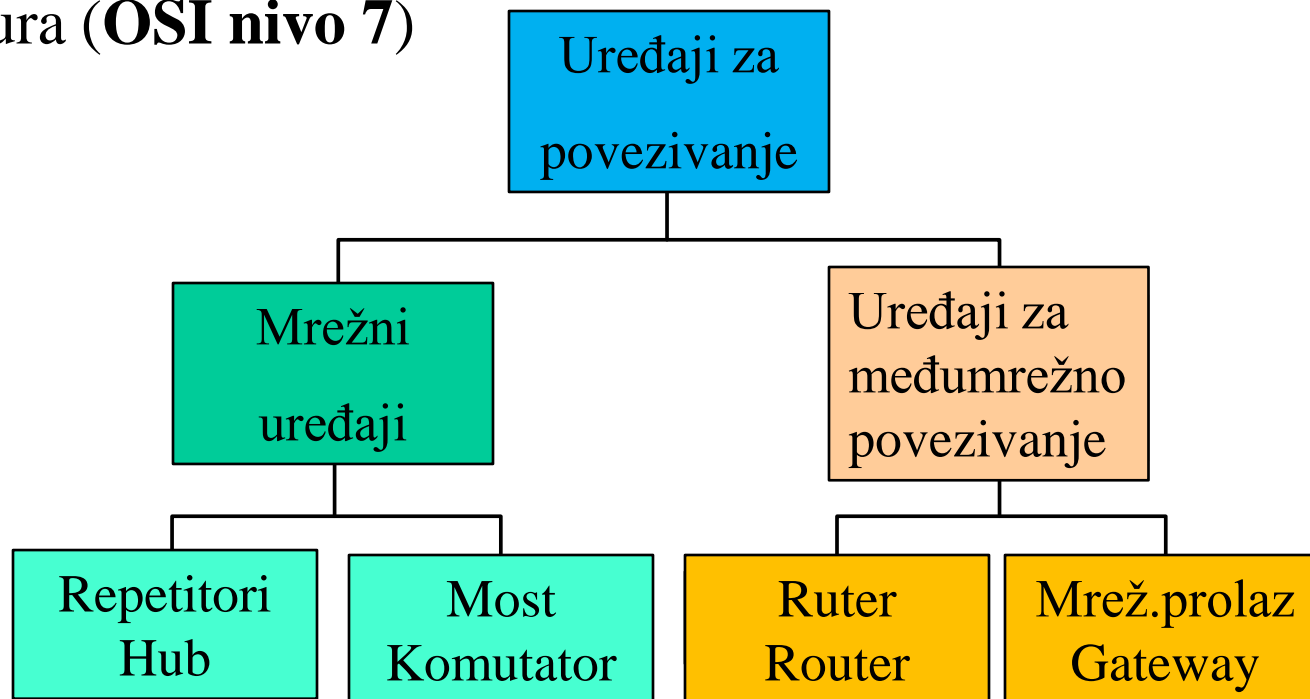
- **Povećava se broj uređaja** koji komuniciraju, zahtevaju se **velike brzine**.
- Ispunjavanje svih ovih zahteva nekada **nije moguće postići** samo protokolima koje smo opisali u prethodnim lekcijama - “**uska grla**”,
- Potreba da se LAN **podeli na više manjih povezanih LAN-ova**
- U jednoj organizaciji imate **više različitih celina**: proizvodno odeljenje, razvojno odeljenje, ekonomski sektor, marketing
- Svi oni imaju **različite potrebe u vidu komuniciranja** a zajedničko je za sve da svi moraju međusobno da komuniciraju **različitim protokolima i različitim brzinama** jer imaju različite uređaje.
- Gradivni blokovi savremenih računarskih mreža su različiti tipovi elektronskih uređaja čiji je osnovni zadatak **da primljenu informaciju proslede na korektnu destinaciju**.
- **Konvertori protokola** - uređaji koji uspostavljaju konekcije
- Postoje mnogi uređaji koji ove probleme rešavaju **u različitim OSI referentnim nivoima** u okviru koga se ta konekcija uspostavlja
- Delimo ih na uređaje koji **povezuju računare** (*host-ove*) i uređaje koji **povezuju LAN-ove**

# IX - Povezivanje mreža

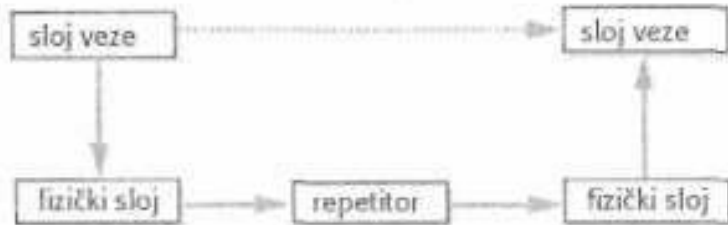


# IX – Uređaji za povezivanje

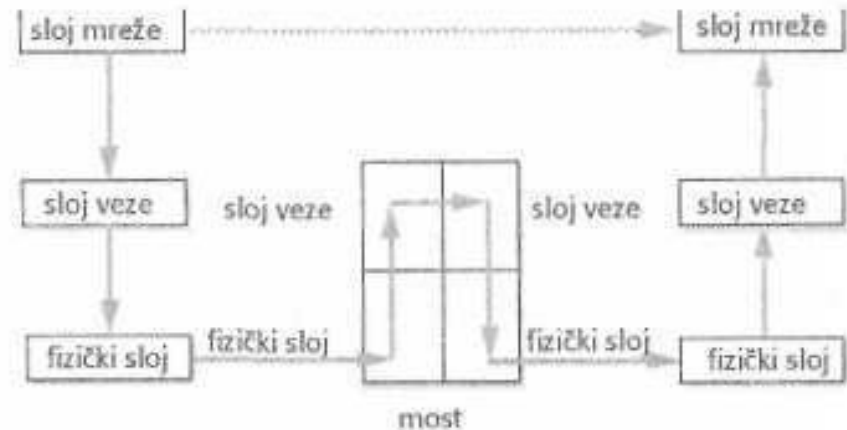
- Uređaji za mrežno povezivanje:
  - **Repetitor/Hab (*Repeater/Hub*)** – povećanje dometa signala, obnavljanje signala (**OSI nivo 1**)
  - **Most/Komutator (*Bridge/Switch*)** – fizička podela mreže na veći broj segmenata (**OSI nivo 2**)
- Uređaji za međumrežno povezivanje:
  - **Ruter (*Router*)** – povezivanje više LAN-ova (**OSI nivo 3**)
  - **Mrežni prolaz (*Gateway*)** – povezivanje mreža različitih arhitektura (**OSI nivo 7**)



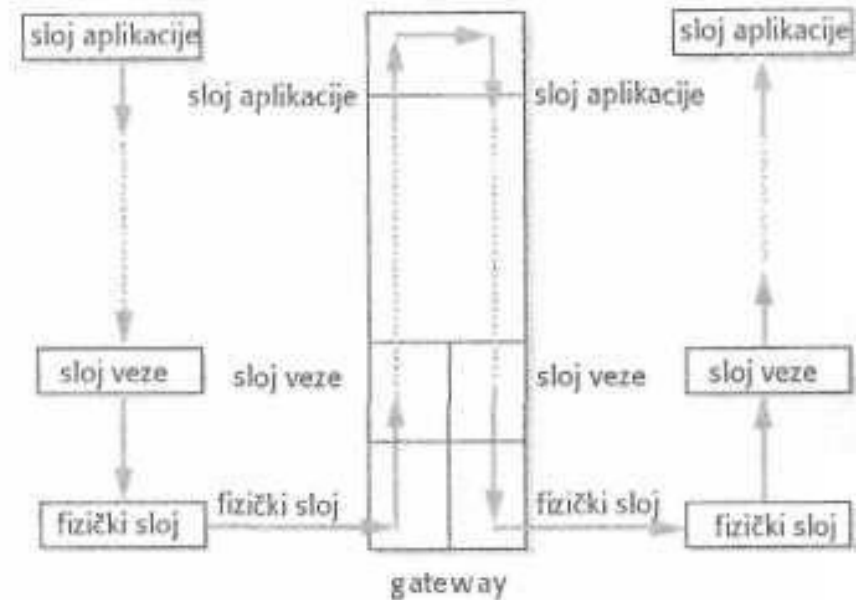
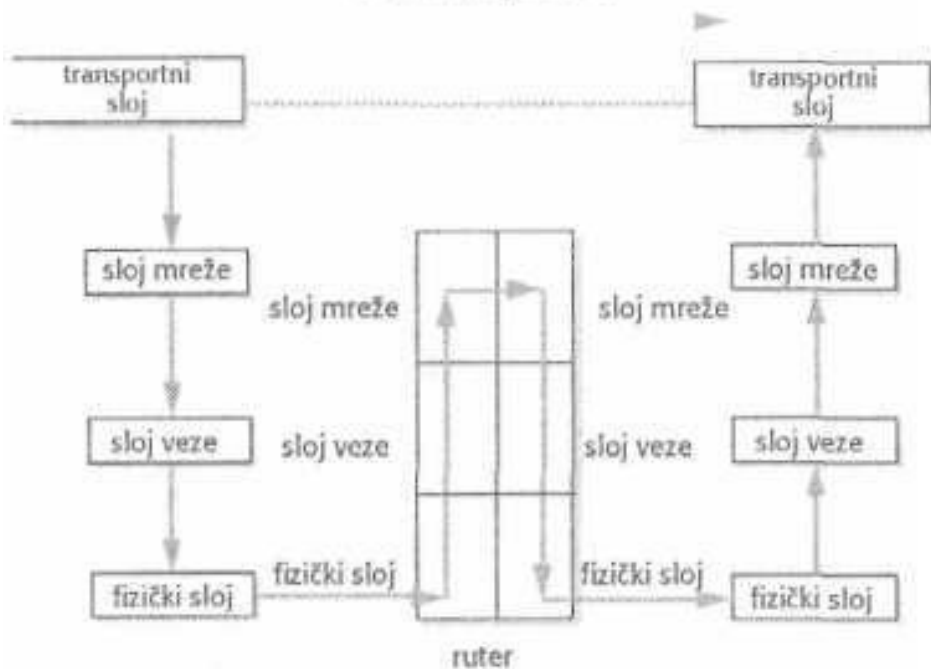
# IX - Povezivanje mreža



(a) Konekcija sloja 1

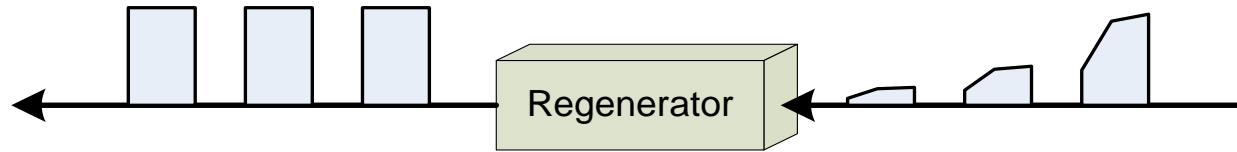


(b) Konekcija sloja 2



# IX - Repetitori

- Amplituda signala na prijemnom kraju **opada sa porastom rastojanja**.
- Repetitor ima zadatak **da podigne nivo signala**, a time i direktno obezbedi uslove za povećavanje rastojanja (dometa prostiranja signala).



- Ovi uređaji vrše **konekciju uređaja na prvom sloju** – fizičkom sloju.
- Repetitor **prihvata podatke/bitove** sa jednog i **prenosi** ga na drugi LAN.
- LAN-ovi koriste **iste protokole i iste formate okvira**.
- Svi uređaji koji su povezani sa repetitorima **vide sve pakete** i ako se detektuje **kolizija na nekom portu**, poruka o tome šalje se na sve ostale
- Uređaji **nisu svesni postojanja repetitora**
- Svi LAN-ovi koji koriste *Ethernet* nalaze se **u istom domenu kolizije**
- Sve što imamo na jednom portu **emituje se na svim ostalim portovima**
- Ne postoji potreba za **hardverskom adresom**
- Analogni signal se jednostavno **pojačava**, a ako je signal **digitalni on se samo regeneriše**.
- Mogu biti sa **dva ili više BNC portova**.

SP200 Series : 10 Mbps Ethernet Repeater



# IX - Habovi (hub)

- *Ethernet* je prvobitno zamišljen sa **topologijom tipa magistrala**.
- Ovakav pristup nije najpogodniji **kod povezivanja više računara** jer **bilo koji prekid na kabl** uzrokuje da sistem postane neoperativan.
- Koristi se ***Ethernet* tipa zvezde** koji se zasniva na UTP kablovima
- Povezivanje gde je **fizička topologija tipa zvezda a logička magistrala**.
- Centar zvezde naziva se **hub** - povezuje veći broj računara putem RJ45
- U suštini *hub* je **multiport repetitor** (broj portova može biti 4,8,16), pa se i on svrstava u uređaje koji **pripadaju istom domenu** kolizije.
- To znači da kada *host* predaje tada **hub** prema ostalim vezama deluje (ponaša se) kao **repetitor**, prosleđujući podatke po principu bit po bit.
- Ako dva računara pokušaju **istovremeno da šalju, dolazi do kolizije**.
- Repetitori i habovi **prvenstveno proširuju domet mreže**
- Svi uređaji **koriste isti medijum za prenos** pa se samim tim degradira ona ideja **da se mreža podeli na manje LAN-ove**.
- **Problem zaštite podataka** jer se svi podaci nalaze se na istom medijumu pa su **dostupni svim uređajima** koji su na tom medijumu.



# IX-Osobine Etherneta na fizičkom nivou

- Naziva se **deljeni Ethernet**.
- Sve radne stanice **dele isti propusni opseg**.
- Svi paketi se **prostiru i pojavljuju svuda na medijumu**.
- Svaki hab (*repetitor*) **unosí kašnjenje prilikom propagacije signala** i to dovodi do ograničenja u broju habova (*repetitor*) na Ethernet segmentu.

## **Problemi Etherneta sa repetitorima i habovima.**

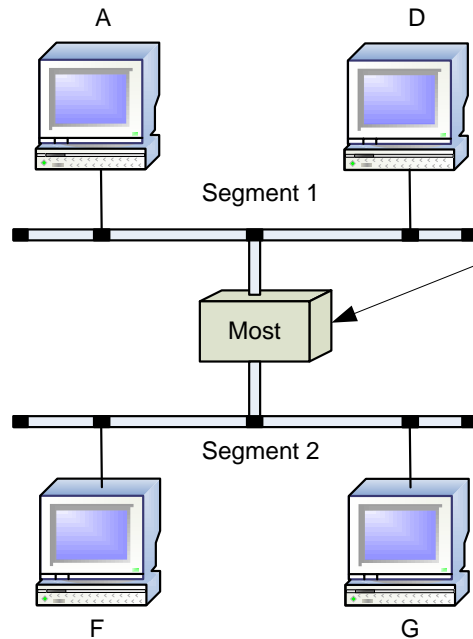
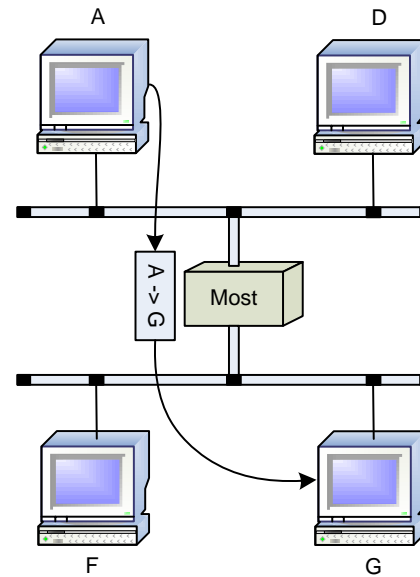
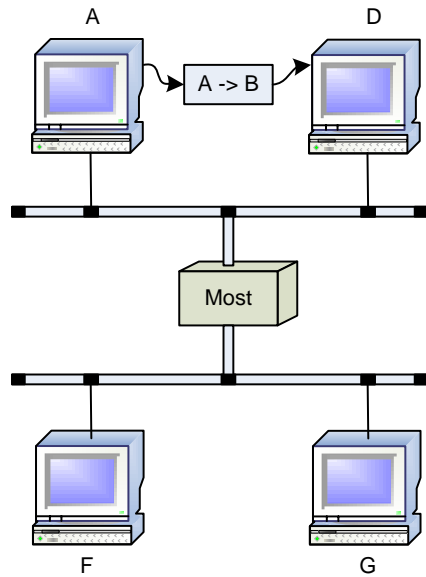
- Kod **povećanja broja računara performanse mreže osetno opadaju**.
- Veliki broj stanica **deli isti propusni opseg**.
- Verovatnoća da se javi kolizija raste i **kolizije se često događaju**.
- **Vreme odziva mreže**, sa porastom broja radnih stanica, **postaje nedozvoljeno veliko**.
- Rešenje problema je **upotreba *switch/komutatora* ili *bridge/mostova***

# IX - Mostovi (Bridge)

- Kod LAN mreža često se javlja potreba za prenosom paketa na **veća rastojanja od dozvoljenih**, ali pri tome treba naglasiti da dodatne uređaje nije moguće priključiti sa obzirom da je **specificirani (dozvoljeni) broj uređaja** po segmentu na samoj ivici da bude premašen
- Kod ovakvih situacija **poželjno je koristiti mostove (bridge)**.
- Sa gledišta OSI modela **most je operativan na nivou-veze (link level)**.
- Most prima okvire, **proverava njihovu adresu**, i na osnovu sadržaja svoje interne "**tabele prosleđivanja**" odlučuje da li će primljeni okvir **proslediti** prema sledećoj mreži ili će ga **izbaciti**.
- Obično se mostovi koriste **za spajanje dve meže** istog tipa, mada se oni mogu koristiti i **za spajanje Ethernet-a sa Token-Ring-om** na nivou veze
- Ovakvi mostovi nazivaju se **transitional bridge** jer mogu da konvertuju **bajt uređenu Token-Ring hardver.adresu na Ethernet adresu**, i obratno
- Obično se za potrebe upravljanja mostovima **može dodeliti IP adresa**.
- Na primer, SNMP (*Simple Network Management Protocol*) zahteva da uređaj ima IP adresu **kako bi se isti pozvao i istestirao njegov status**.

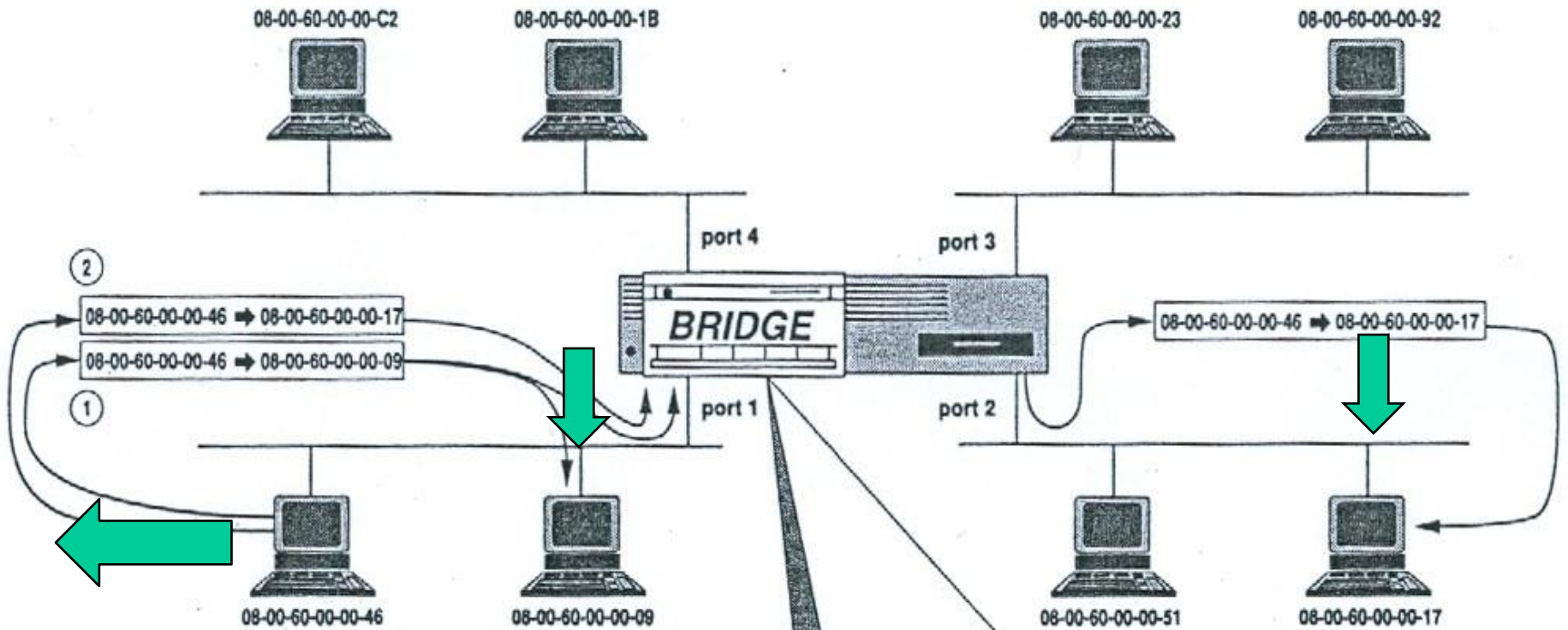


# IX - Mostovi (Bridge)



Fiz. adresa	Segment
A	1
D	1
F	2
G	2

# IX - Mostovi (Bridge)



Address Table

adr	port
08-00-60-00-00-09	1
08-00-60-00-00-C2	4
08-00-60-00-00-23	3
08-00-60-00-00-46	1
08-00-60-00-00-51	2
08-00-60-00-00-92	3
08-00-60-00-00-17	2
08-00-60-00-00-1B	4

# IX - Mostovi (Bridge)

- **jednostavni most** - najjednostavnija konfiguracija sa mostom koji povezuje dva *Ethernet* LAN-a. Ovakva konfiguracija se može ostvariti i pomoću repetitora sa izuzetkom da sav saobraćaj između DTE-ova u jednom LAN-u ne prolazi preko drugog LAN-a.
- **multiportni most** – most otvara paket, čita odredišnu MAC adresu i komutira paket ka odgovarajućem portu. Ovakvi mostovi su **samoučeći**. Da bi poslao paket, most obično analizira sve pakete koje prima, a u tabeli izvornih adresa pamti portove sa kojih je primio te poruke. Za slučaj da se odredišna adresa ne nalazi u tabeli izvornih adresa tada most koristi **tehniku plavljenja** (*flooding*) koja šalje paket ka svim portovima i otkriva tačan port. Tabela se **periodično čisti** (briše) od neaktivnih adresa.
- **multiportni multiprotokolarni most** - mostovi imaju ugrađenu mogućnost memorisanja-i-prosleđivanja (*store-and-forward*) paketa. To omogućuje mostovima da mogu ostvariti i konverziju protokola tj. mogućnost povezivanja dve mreže različitog tipa, *Ethernet* sa *Token Ring* mrežom, a konverzija protokola vrši se na nivou 2 OSI modela.

# IX - Vrste mostova (*Bridge*)

- 1. lokalni mostovi** - za povezivanje dva bliska segmenta LAN-a, vrši se manuelno konfigurisanje tabela u mostu
- 2. udaljeni mostovi** - za povezivanje udaljenih segmenata LAN-a, manuelna konfiguracija tabela
- 3. transparentni mostovi** - povezuju segmente s istim MAC-om (IEEE 802.1 D). Svaki most ima tablicu koja preslikava adrese stanica u priključke mosta
- 4. *multiport bridge*** – most s više od dva priključka, manuelna konfiguracija tabela
- 5. translacijski mostovi - komutator/switch**
  - povezuju segmente s različitim MAC-om
  - pretvaranje (translacija) okvira odvija se na LLC-u (OSI 2 nivo)
  - sposobnost učenja – autonomna konfiguracija tabela

# IX - Udaljeni mostovi

- Mostovi se uglavnom koriste kao mesta centralizovanog povezivanja kod lokalnih LAN konfiguracija.
- Jednostavni mostovi nazivaju se **lokalnim mostovima** (*local bridges*).
- Mostovi se mogu koristiti i za međusobno povezivanje **dve udaljene mreže preko WAN linka**
- Ovakvi mostovi nazivaju se **udaljeni mostovi** (*remote bridges*)
- Oni obavezno rade u **parovima** (jedan na jednom kraju, a drugi na drugom kraju WAN-a).
- LAN-ovi se mogu povezati preko mostova kako u slučajevima kada se **koristi topologija stabla** tako i kada se **koristi topologija rešetke**.

Mreže povezane preko mostova koriste a dva algoritma za rutiranje:

1. **spanning tree algorithm** kod povezivanja dva *Ethernet* LAN-a
2. **source routing algorithm** kod povezivanja *Token-Ring* LAN-ova.

# IX - Transparentni mostovi

- Svaki most **zna samo svog suseda** i transparentan je drugim mostovima
- Transparentni most, za potrebe rutiranja, koristi **spanning tree algorithm** koji određuje adrese memoriše u tabeli portova.
- **Sam se brine o svojim tabelama** koje su pre instalacije u mrežu **prazne**
- Za svaki primljeni segment **pretražuje tabele po određenoj adresi**
- Ako nađe adresu odredišta, **okvir prosleđuje na odgovarajući port**
- Most **ne zna gde se nalazi određeni LAN**, on samo zna **gde je povezan susedni čvor** za tu određenu adresu
- Most saznaje podatke o rutiranju putem **povratnog saznanja** (učenja).
- Kada paket pristigne na port, **on pamti izvornu adresu porta** i pridružuje adresu tog porta svojoj tabeli rutiranja.
- Zatim **prosleđuje paket ka odredištu**.
- Ako se **određena adresa ne nalazi u tabeli rutiranja**, transparentni most predaje **broadcast** poruku da bi pribavio (saznao) gde se nalazi (misli se na kom portu) ta adresa.
- Transparentni mostovi se **koriste kod topologije tipa stablo**, pri čemu **zatvaranje petlji nije dozvoljeno**.

# IX - Transparentni mostovi

## Problem:

**Neažurnost tabele** usled dinamičke rekonfiguracije mreže (zamena mesta čvorovima, priključen novi čvor, čvor je isključen sa mreže ...)

## Rešenje:

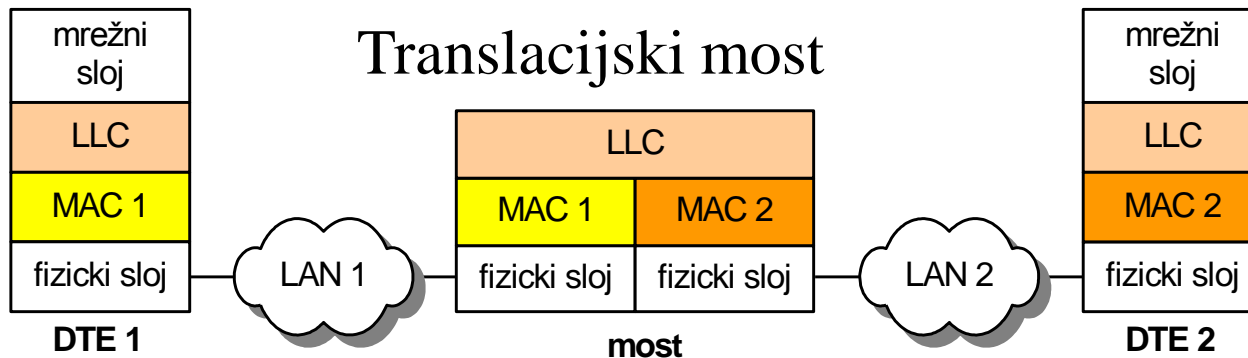
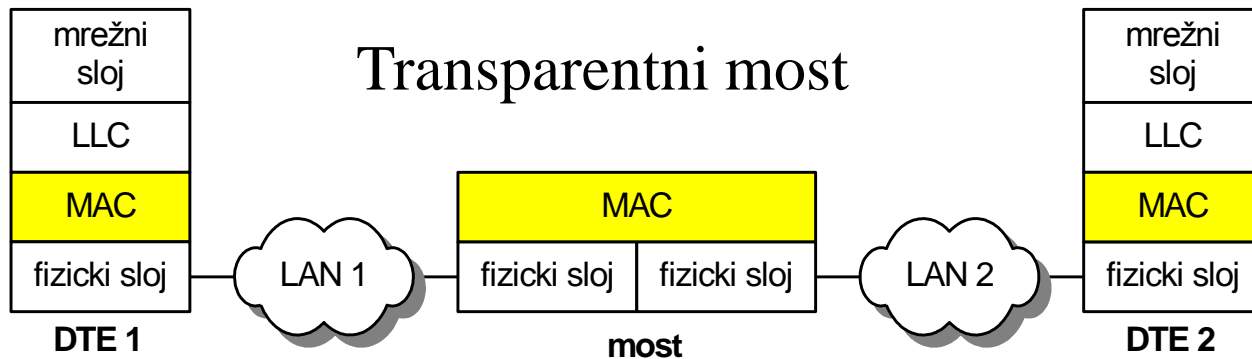
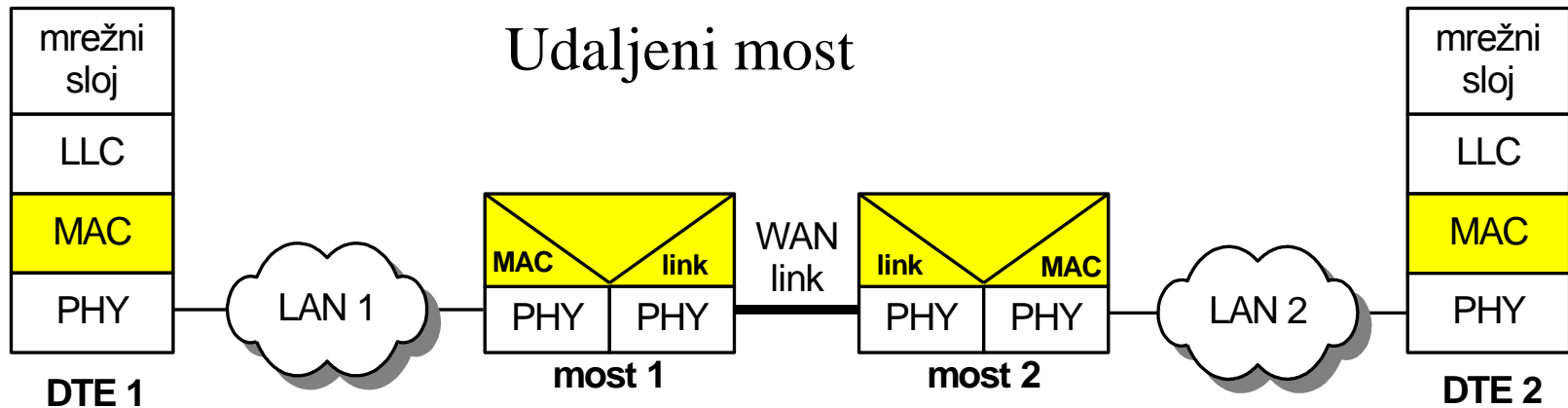
Svaka stavka u tabeli **ima ograničeno vreme trajanja**, koje se obnavlja uvek kada most primi okvir koji sadrži adresu datog čvora.

Tekuće vreme: **12:02**

Fiz. adresa	Segment	Važi do
A	1	12:10
D	1	12:23
F	2	11:50
G	2	12:05

Nevažeća stavka

# Arhitekture povezivanja putem mostova





# IX - Svič/Komutator (Switch)

- **Switch** upravlja protokom podataka između delova lokalne mreže.
- **Switch** deli mrežni saobraćaj i šalje ga na određena mesta, dok *hub* šalje podatke na sve uređaje koji su u mreži.
- Korisi se **za mreže srednje veličine** i daje računaru punu brzinu jedne konekcije, dok računari priključeni na *hub* dobijaju samo jedan deo
- Povezuje **radne stanice, habove i druge svičeve**
- Primljeni paket se **obavezno baferuje**
- Paket primljen sa jednog porta **emituje se na drugi port**
- **Proveri stanje na medijumu** i ako je medijum slobodan šalje paket
- Ako dođe do kolizije, **odustaje od slanja** i radi retransmisiju paketa
- Za prosleđivanje paketa koristi se **fizička tj. MAC adresa**.
- Gotovo svi današnji svičevi se **lako instaliraju** i sposobni su da **samostalno nauče i mapiraju MAC adrese** mrežnih uređaja na odgovarajući fizički port odnosno interfejs.



# IX - Svič/Komutator (Switch)

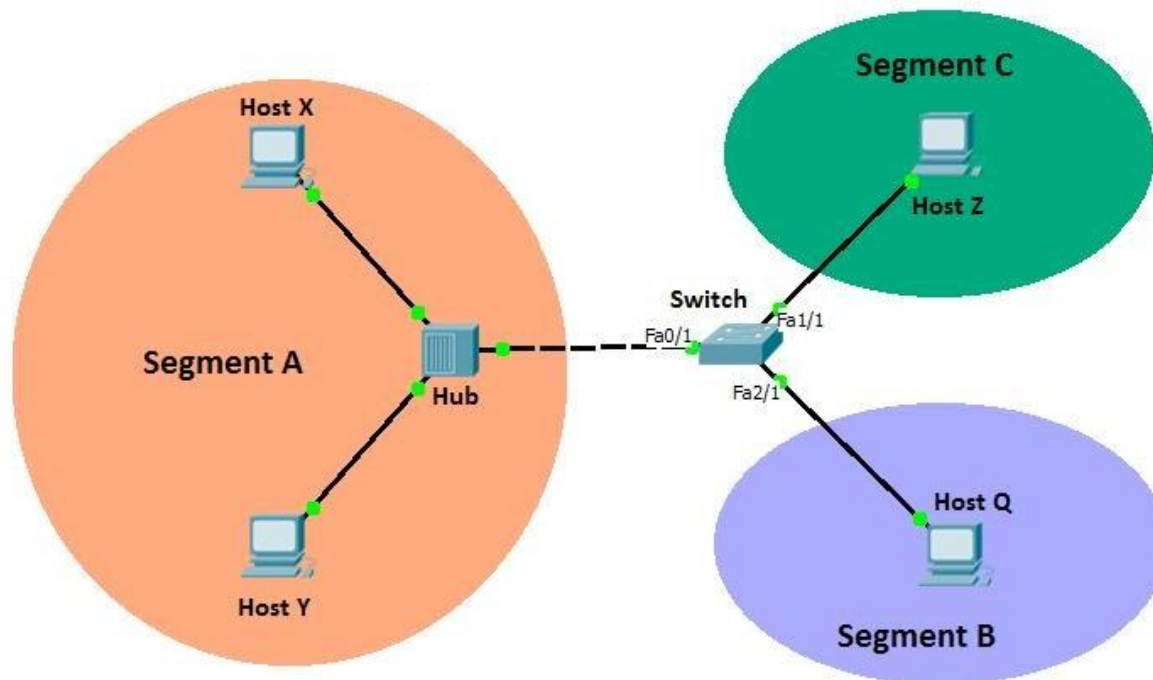
*Kako svič/komutator zna gde da uputi primljeni paket?*

- *Switch/komutator analizira sve okvire* i na osnovu polaznih **Ethernet (MAC) adresa** i određuje koji je računar priključen na **određeni port**.
- **Izvorna adresa** se koristi za učenje **o uređajima na mreži**, dok se **odredišna** koristi za **prosleđivanje paketa**.
- Tabelu sa adresom računara i brojem porta na koji je on priključen *switch/komutator čuva u memoriji*.
- Na osnovu **odredišne adrese iz okvira** i tabele *switch/komutator* zna na koji port treba da uputi paket.
- Postoje slučajevi kada se **primljeni okvir** sa jednog porta šalje na **sve portove switch/komutatora**
- **Brodkast paket** upućen na **brodkast adresu** prosleđuje se na **sve portove switch/komutatora**
- Paket upućen na adresu **koju komutator do tada nije "naučio"** (nema je u tabeli) **prosleđuje se na sve portove switch/komutatora**.

# IX - Svič/Komutator (Switch)

Na sledećem primeru može se videti kako klasičan komutator zaista radi:

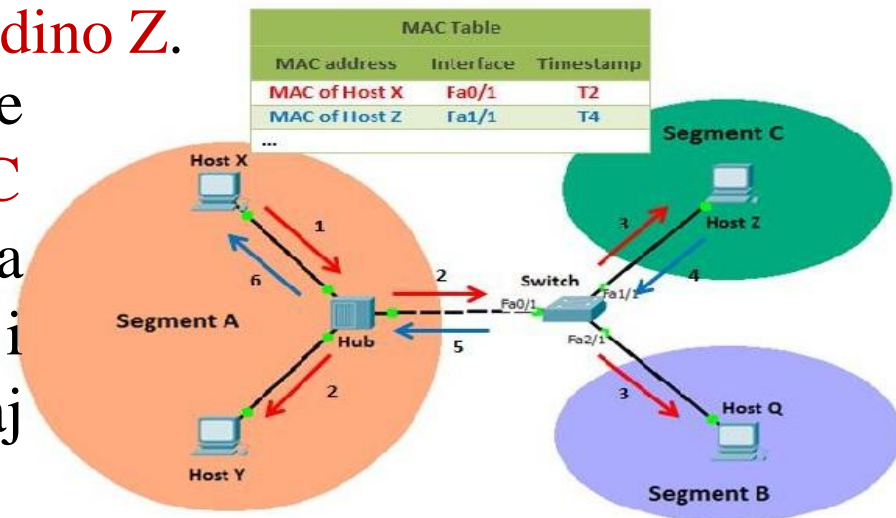
- Računari **X** i **Y** pripadaju segmentu **A**, dok računar **Q** i računar **Z** pripadaju segmentima **B** i **C** respektivno.
- U segmentu **A** nalazi se **jedan hub** uređaj na kome su mrežnim kablom povezani računari **X** i **Y**.
- Pretpostavimo da smo ovu mrežnu arhitekturu (sve segmente) povezali jednim **komutatorom** i na njega mrežnim kablom povezali hub, računare **Q**



# IX - Svič/Komutator (Switch)

## 1. Računar X (segment A) šalje paket računaru Z (segment C).

- Paket stiže do *hub* uređaja koji prosleđuje paket na svim portovima
- Kako su na istom segmentu, računar Y prima taj paket, ali ga odbacuje jer analizirajući MAC adresu zaključuje da paket nije namenjen njemu.
- Komutator analizira izvornu adresu, i u svojoj MAC tabeli koja je prazna, unosi zapis u obliku **<MAC address, port, timestamp>**.
- Tako komutator zna na kom se portu nalazi računar X (port Fa0/1).
- Ovo predstavlja proces učenja (*learning*).
- Kako trenutno komutator zna samo za računar X, paket se prosleđuje na svim interfejsima (*flooding*), osim na interfejs sa kog je stigao.
- Paket primaju Q i Z, ali potvrđuje jedino Z.
- Komutator saznaje na kom portu je računar Z i dodaje novi zapis u MAC
- Analizirajući tabelu komutator sada zna na kom je portu računar X i prosleđuje (*forwarding*) paket na taj port (nema *flooding*-a).



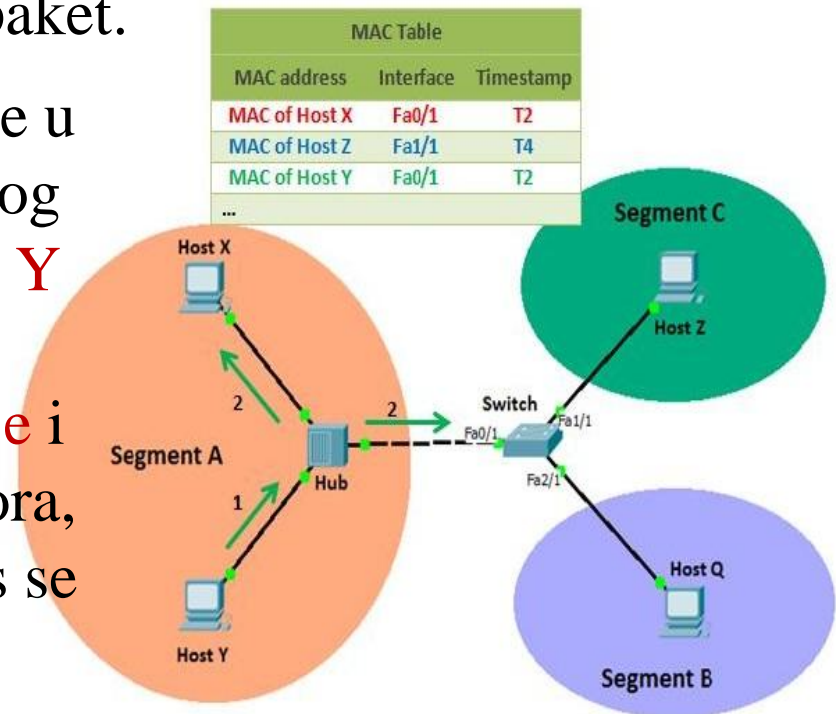
# IX - Svič/Komutator (Switch)

## 2. Računar Y (segment A) šalje paket računaru X (segment A).

- Kada paket stigne do hub uređaja, ovaj **prosleđuje** kao i u prvom primeru na sve portove.
- Kako su računari X i Y na istom mrežnom segmentu, **paket stiže do hosta X** i time je jednosmerna komunikacija završena.
- Međutim, **komutator to ne zna**, jer i on u istom vremenskom trenutku dobija paket namenjen računaru X, pa komutator sada **uči na kom se portu nalazi računar Y** koji je poslao paket.

- **Analizirajući odredišnu adresu** i zapise u tabeli za računare X i Y, na osnovu istog interfejsa (*Fa0/1*) zaključuje da su **X i Y na istom segmentu**.

- Zbog toga, primljeni **paket se odbacuje** i nema prosleđivanja od strane komutatora, tj. **nema dupliranja paketa**. Ovaj proces se naziva **filtriranje** (*filtering*)



# IX - Svič/Komutator (Switch)

## 3. Računar X šalje paket računaru Z koji trenutno nije aktivan.

- polje *timestamp* se koristi kako bi MAC tabela uvek bila ažurna.
- Uvek kada komutator analizira izvornu adresu dolaznog paketa, **ažurira se (aging) polje timestamp** trenutnim vremenom ili se kreira novi zapis
- Komutator **prolazi kroz sve zapise** u tabeli i proverava *timestamp* polje.
- Ako uređaj na mreži ne procesira pakete u određenom vremenskom intervalu onda se zapis za taj uređaj, u MAC tabeli **posmatra kao nevažeći i uklanja iz tabele.**

- **mac-table-aging-time** - interval posle kog se zapisi odbacuju kao zastareli

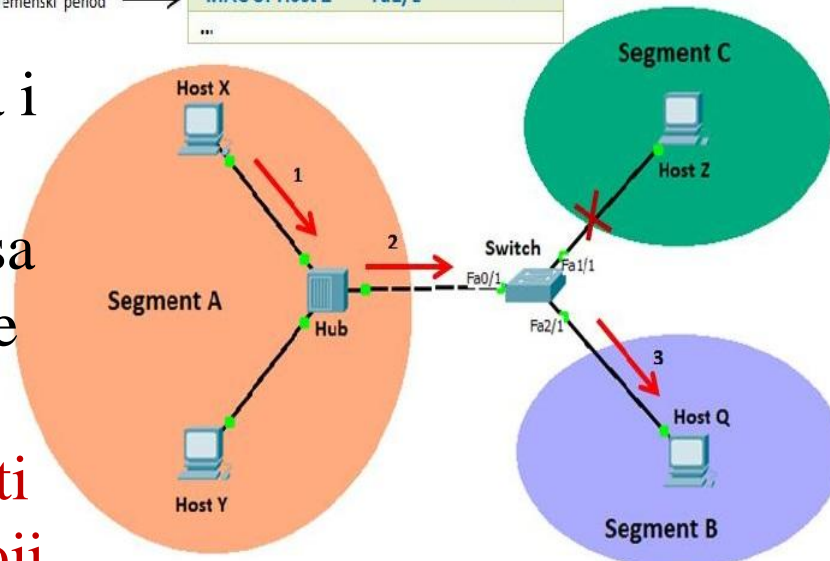
Uklanja se jer host Z nije bio aktivan određeni vremenski period →

MAC Table		
MAC address	Interface	Timestamp
MAC of Host X	Fa0/1	T2
MAC of Host Z	Fa1/1	-----
...		

- Zbog **ograničene memorije** komutatora i **broja zapisa** u tabeli.

- Ako računar X pokuša da komunicira sa računarem Z koji nije na mreži, onda se radi klasičan **flooding**

- Na primljeni paket **niko neće odgovoriti** pa se zaključuje da **uređaj više ne postoji.**



# IX - Tehnike prosleđivanja

Dve najpoznatije tehnike za prosleđivanje paketa kod komutatora su:

1. ***Cut-Through*** - zbog brzine, paket se prosleđuje odmah nakon analiziranja odredišne adrese,
  2. ***Store-and-Forward*** - paket se analizira u celini i prosleđuje jedino ako nema grešaka.
- Korišćenje prve tehnike **dobija se na brzini**, dok korišćenjem druge tehnike **smanjuje se kolizija i propagacija paketa s greškom** se ovde zaustavlja.
  - Danas je razlika između ove dve tehnike u **vremenu prosleđivanja postala zanemarljiva**, tako da se ***Store-and-Forward*** tehnika nameće kao bolje rešenje.

# IX - Svič/Komutator (Switch)

## Razlike habova i svičeva:

### Habovi:

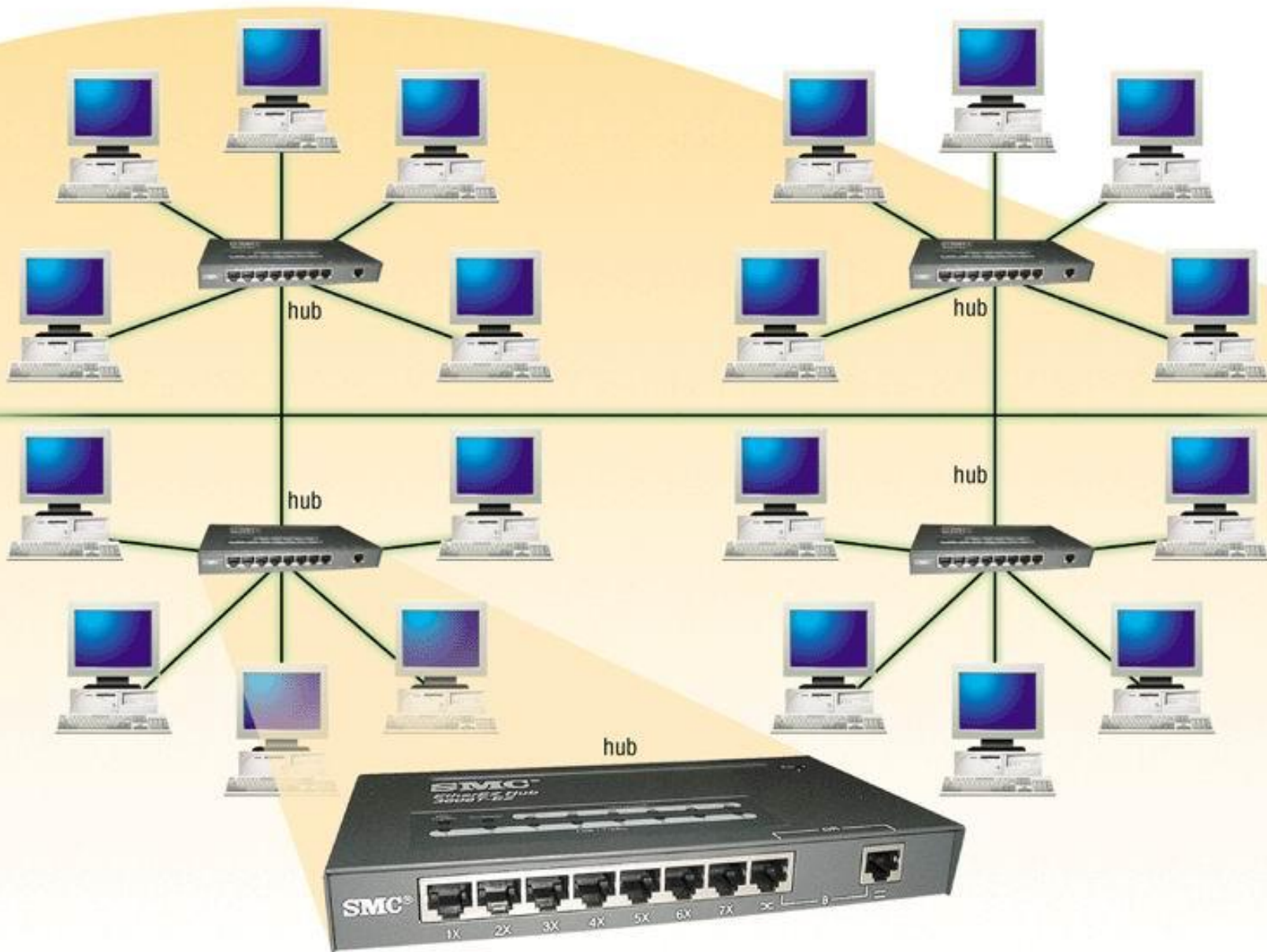
- sve tačke priključene na hab **dele isti propusni opseg** (10Mbps),
- paket koji se pojavi na nekom portu prosleđuje se **na sve ostale portove**
- rade u prvom OSI referentnom nivou i **istom domenu kolizije**

### Svičevi/komutatori:

- svaka tačka priključena na port *switch*/komutatora **koristi svoj propusni opseg** (10Mbps),
- paket koji se primi na neki port prosleđuje se **na tačno određeni port**
- rade na 2 OSI referentnom nivou i u **različitim domenima kolizije**
- Učenje se odvija korišćenjem **specijalnih tabela** koje su zbog brzine najčešće **hardverski implementirane** (*Application Specific Integrated Circuits ili skraćeno ASIC*).
- Ove tabele se sreću pod nazivom **CAM** odnosno **MAC** tabele (*Cisco koristi CAM terminologiju*).



# Tipčna mreža sa svičevima/habovima



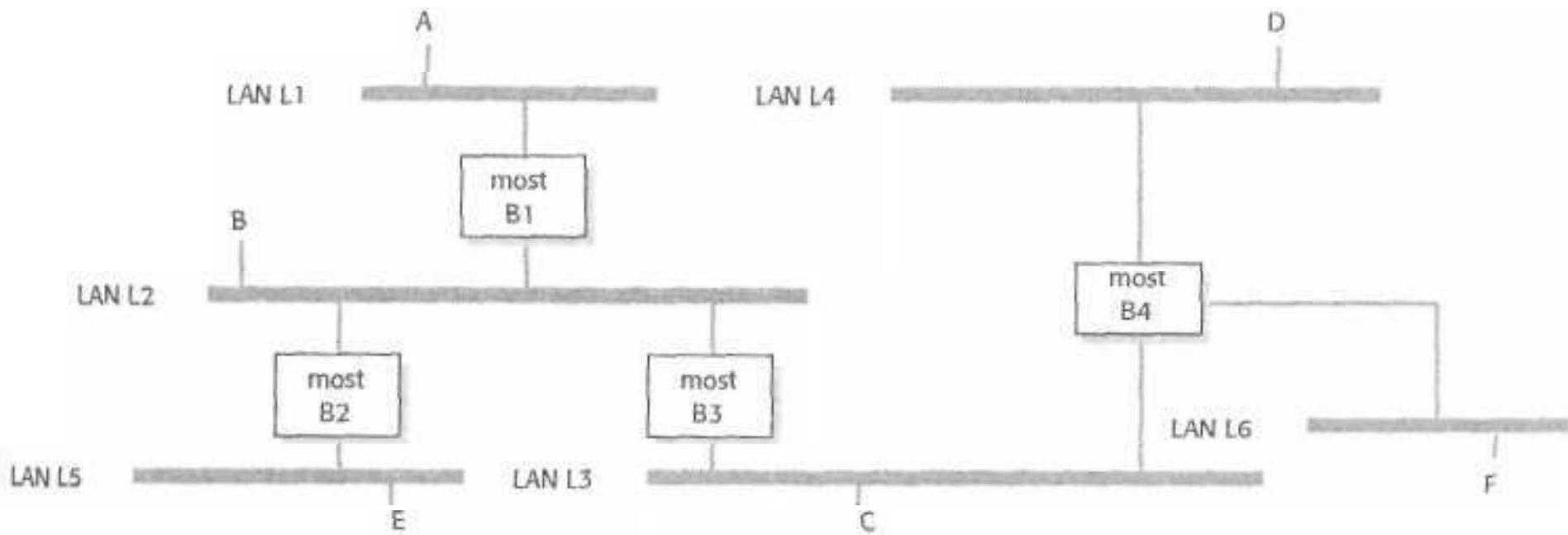
# IX - Tabele rutiranja

- Informacije koje pomažu mostovima da donesu **odluke o rutiranju**
- Ova tabela se nekada naziva **baza prosleđivanja** (*forwarding database*).
- Svaki most ima po jednu tabelu za svaki LAN na koji je povezan. Kada okvir dođe do mosta **on pogleda odredišnu adresu** u tabeli za taj LAN.
- Ulazi tabele definišu na koji LAN most treba da usmeri pristigli okvir.

## ***Kako most definiše svoju tabelu rutiranja ?***

- Da se unapred izvrši programiranje tih tabela – **fiksno rutiranje**.
- Ako želimo da koristimo mostove i komutatore u **dinamičkim mrežama**, potrebno je imati **dinamičko reprogramiranje** tih uređaja.
- Takve uređaje nazivamo **transparentnim uređajima** - IEEE 802.1d.
- Ova mogućnost ažuriranja tabela rutiranja naziva se **učenje adrese**.
- Javlja se problem **kada se uređaji premeste** a ne pošalju okvire.
- Kada se dobije okvir **za koji ne postoji zapis u tabeli rutiranja**, most se koristi **algoritmom plavljenja** (*flooding algorithm*).
- U određenim topologijama, kada imamo dva LAN-a spregnuta putem **više od 2 mosta**, dolazi do problema **beskonačno propagiranje okvira**
- **Broj okvira** koji se šalju na LAN-ove se uvećava (1-2-4-8-16...).

# IX - Usmeravanje paketa



*Proces donošenja odluke koji će se okviri proslediti i gde, naziva se **rutiranje pomoću mosta.***

# IX - Povezivanje mreža

**B1**

Izvor LAN L1		Izvor LAN L2	
Određište	Sledeći LAN	Određište	Sledeći LAN
A	—	A	L1
B	L2	B	—
C	L2	C	—
D	L2	D	—
E	L2	E	—
F	L2	F	—

**B2**

Izvor LAN L2		Izvor LAN L5	
Određište	Sledeći LAN	Određište	Sledeći LAN
A	—	A	L2
B	—	B	L2
C	—	C	L2
D	—	D	L2
E	L5	E	—
F	—	F	L2

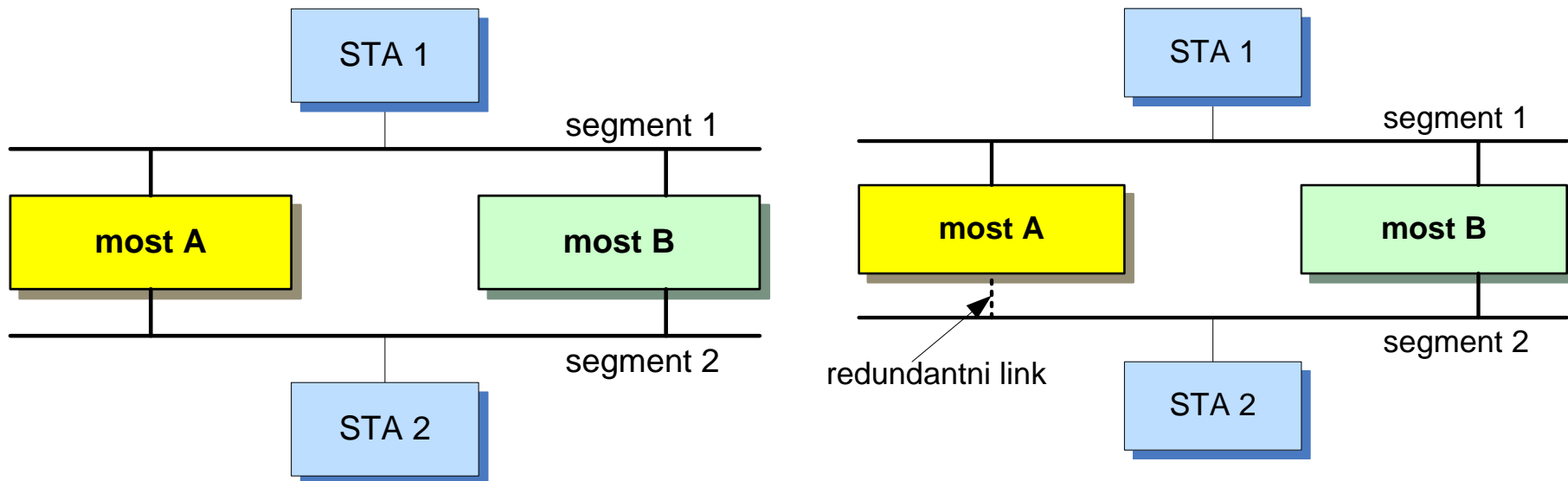
**B3**

Izvor LAN L2		Izvor LAN L3	
Određište	Sledeći LAN	Određište	Sledeći LAN
A	—	A	L2
B	—	B	L2
C	L3	C	—
D	L3	D	—
E	—	E	L2
F	L3	F	—

**B4**

Izvor LAN L3		Izvor LAN L4		Izvor LAN L6	
Određište	Sledeći LAN	Određište	Sledeći LAN	Određište	Sledeći LAN
A	—	A	L3	A	L3
B	—	B	L3	B	L3
C	—	C	L3	C	L3
D	L4	D	—	D	L4
E	—	E	L3	E	L3
F	L6	F	L6	F	—

# IX - Beskonačno propagiranje okvira



- Pretpostavimo da su **tabele rutiranja mostova A i B prazne**
- Računar STA1 **šalje okvir** do računara STA2.
- Pošto **ni jedan most nije "svestan" onog drugog**, svaki prihvata okvir i prosleđuje ga na LAN L2 (segment 2).
- Sada A vidi okvir iz B, a B vidi okvir iz A **ali sada sa L2** (segment 2).
- Pošto ni jedan ne zna gde se STA2 nalazi, oba mosta prihvataju okvir i prosleđuju ga na LAN L1 (segment 1).
- Oba mosta **vide okvir od onog drugog i ovoga puta ga prosleđuju nazad** na L2 (segment 2) - nastaje oluja slanja (*broadcast storm*).

# IX - Zagušenje i "samrtni zagrljaj"

*Ako dođe do otkaza nekih linkova, dolazi do još **intezivnijeg saobraćaja** na preostalim linkovima pa lako može doći do preteranog nagomilavanja paketa u jednom ili više mrežnih čvorova - **zagušenje mreže**.*

➤ Postoji nekoliko načina **za razrešavanje ovog problema**:

1. **Eliminisanje paketa** – podrazumeva odbacivanje nagomilanih paketa.
2. **Kontrola toka** – kontrolisanje dolaznih paketa i ograničavanje slanja.
3. **Dodela bafera** – unapred se rezerviše dovoljan broj bafera koji su zaduženi za prihvatanje podataka i njihovo dalje slanje - **virtuelna kola**.
4. **Prigušivanje**-nadgledanje aktivnosti na svojim portovima-**slanje Choke**

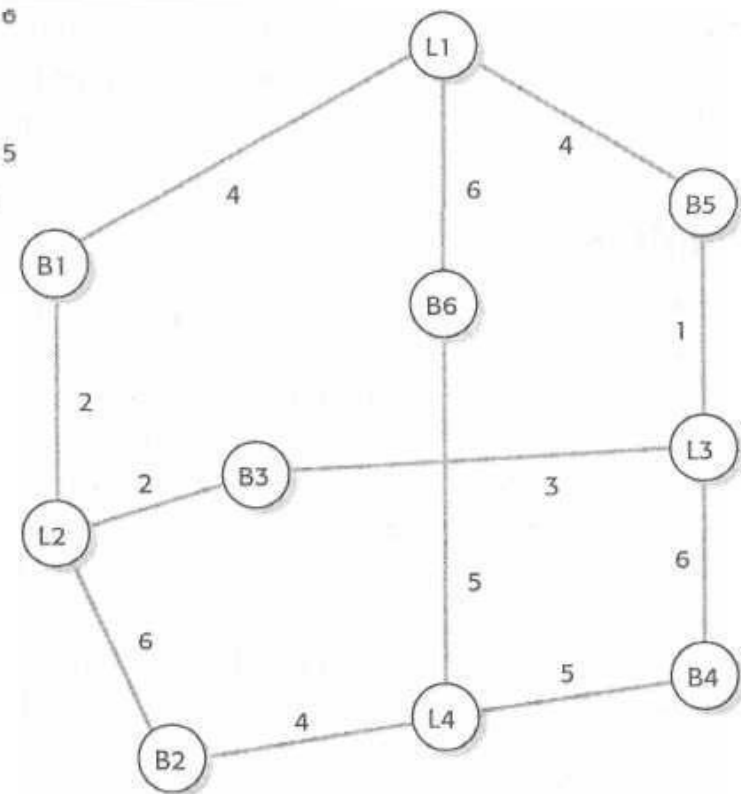
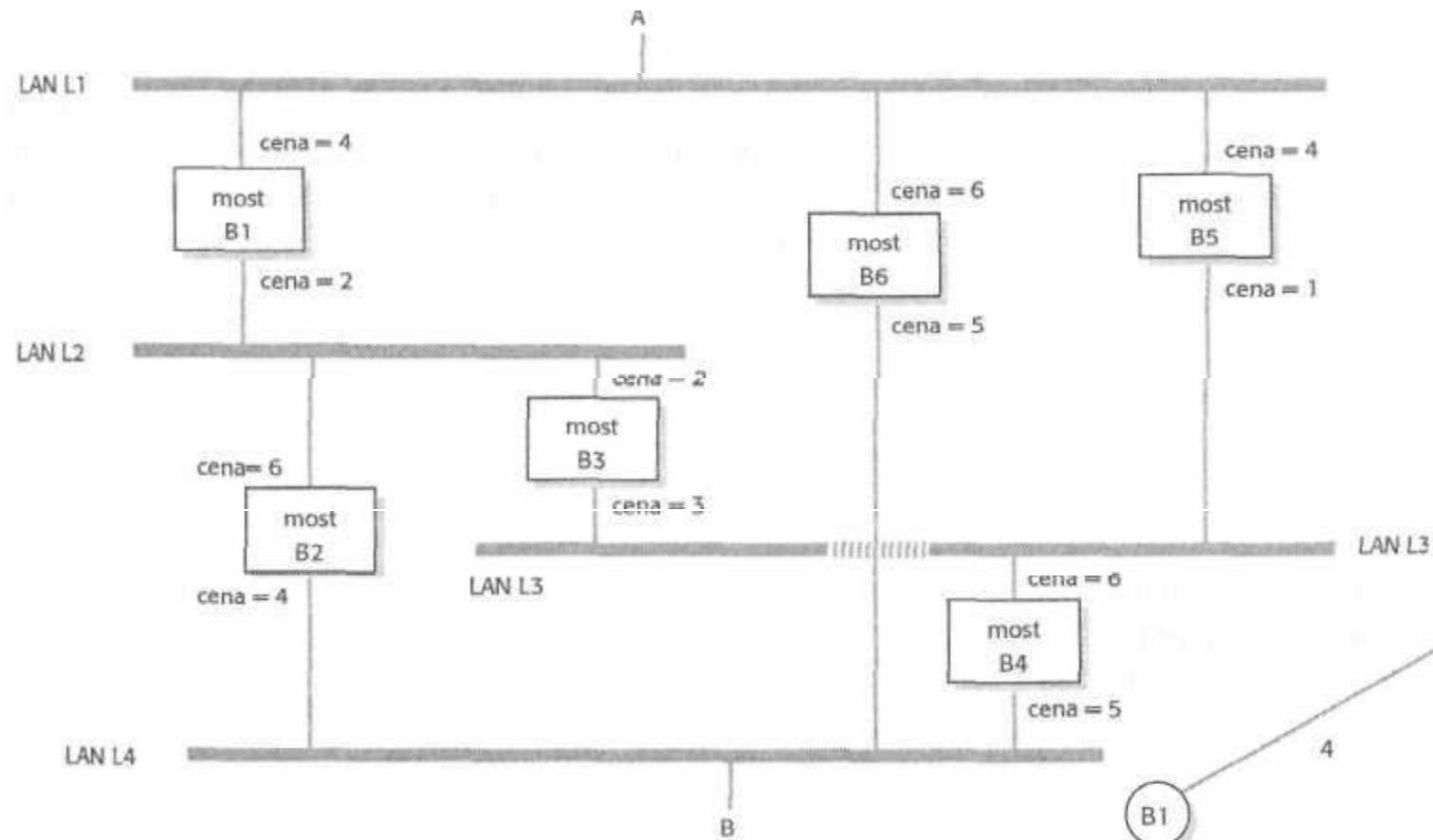
➤ U najgorem slučaju zagušenje može da dovede do **samrtnog zagrljaja**

Postoje više razloga koji dovode do ovog slučaja ali su **dva najčešća**:

1. **Smesti i prosledi** (*store-and-forward*) - **prepunjenost bafera tri čvora** koji su međusobno vezani i čekaju jedan drugog da se isprazni kako bi mogli da pošalju pakete.
2. **Ponovno sastavljanje paketa** (*reassembly deadlock*) – ovde se koriste zajednički baferi za prijem paketa od različitih čvorova. Može da se desi da stignu paketi 2,3,4,5 od dva čvora a da ne stigne paket 1.

# IX - Spanning tree algoritm

- Rešavanja nekontrolisanog propagiranja okvira je **eliminisanje petlji**.
- Izvršavanje **spanning tree algoritma** (stablo razapinjanja).
- Temenima grafa predstavljaju se **svi mostovi i LAN-ovi** u mreži
- Uzima se **minimalni podskup grana** koji će povezati sva temena grafa.
- Da bi algoritam funkcionisao potrebno je da se svakom portu mosta dodeli **cena konekcije**. Ovde se u ukupnu cenu računaju samo iznosi koji se odnose **na cenu konekcije od mosta ka LAN-u** gledano u smeru slanja okvira. **Cena slanja okvira od L1 do L4 iznosi 6 (M1-L2(2)+M2-L4(4))**
- Svaki **spanning tree** algoritam se sastoji iz 3 koraka tj. izbora: **1.root-korena stabla; 2.root porta i 3.označenog mosta (designated bridge)**
- Koristeći terminologiju struktura podataka **root će biti početak stabla**
- Obično za root se bira **most sa najnižim ID**.
- Mostovi biraju root **slanjem niza specijalnih okvira** poznatih kao **BPDU ( Bridge Protocol Data Units )** u pravilnim intervalima.
- Svaki od BPDU sadrži **ID mosta, ID porta** preko koga je okvir inicijalno poslat i **akumulirane cene** portova preko kojih je okvir poslat.





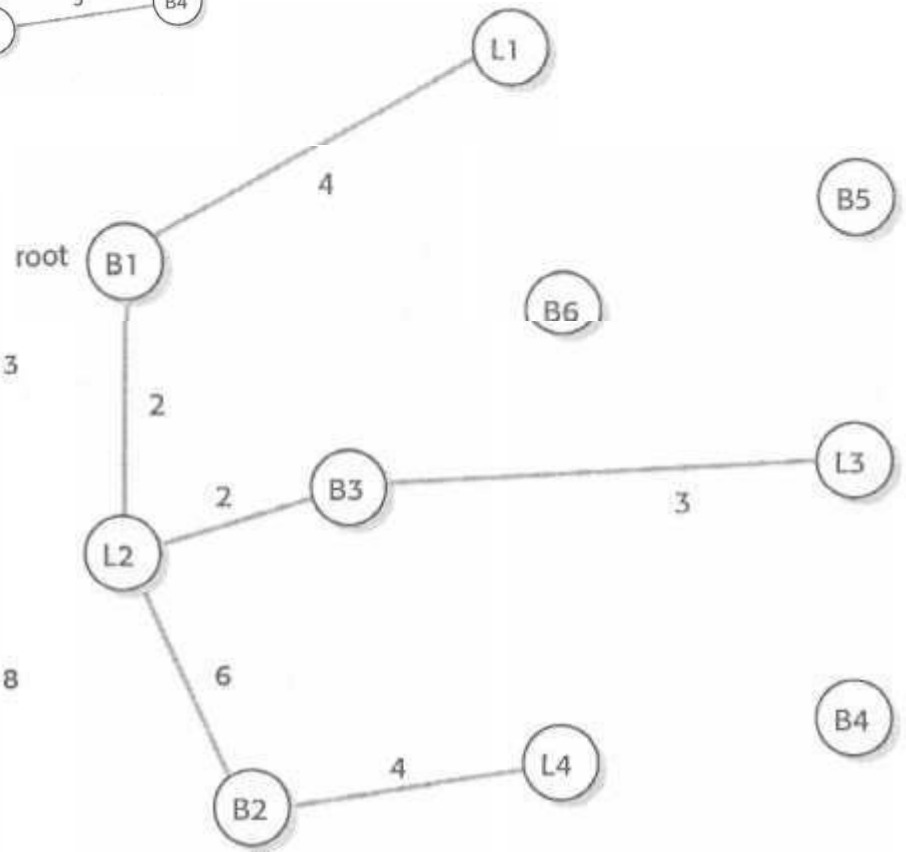
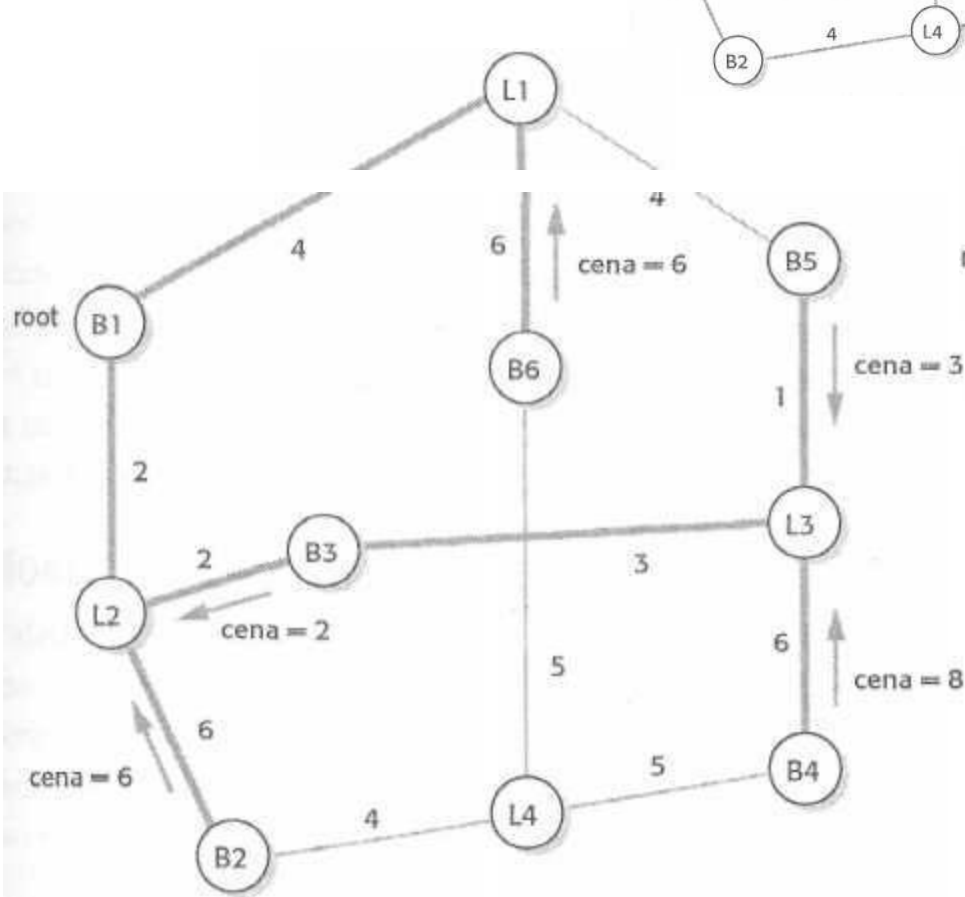
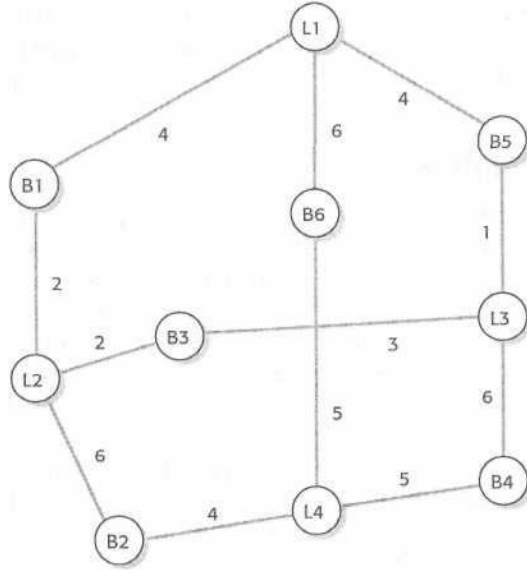
# IX - Spanning tree algoritm

1. Kada most primi BPDU, on **poredi ID izvornog mosta sa sopstvenim**. Ako most ima veći ID on zna **da neće biti root**. On onda ubeleži ID mosta pošiljaoca i cenu putanje do njega, **inkrementira putanju za cenu prijemnog porta i prosleđuje BPDU** kroz sve preostale portove a istovremeno **prestaje da šalje svoje sopstvene BPDU-e**. Ako je ID mosta **niži od onog koji je poslao BPDU** on onda neće proslediti primljeni BPDU jer most koji je poslao taj okvir **nikada neće biti root**. Na kraju, svaki most, osim onog sa najnižim ID-om **prestaje da šalje okvire**, zato što zna da neće biti izabran kao root. Posle određenog vremena kada okviri prestanu da stižu na taj most **on smatra da je on izabran za root**.
2. U ovom koraku svaki most utvrđuje svoj **root port**, a to je port koji **odgovara najjeftinijoj putanji ka root mostu**. Pošto svaki most prethodno beleži cene za svaki BPDU primljen preko svakog porta, jednostavno **traži najnižu cenu**. Tako je omogućena **najjeftinija komunikacija** sa root mostom preko root porta.

# IX - Spanning tree algoritm

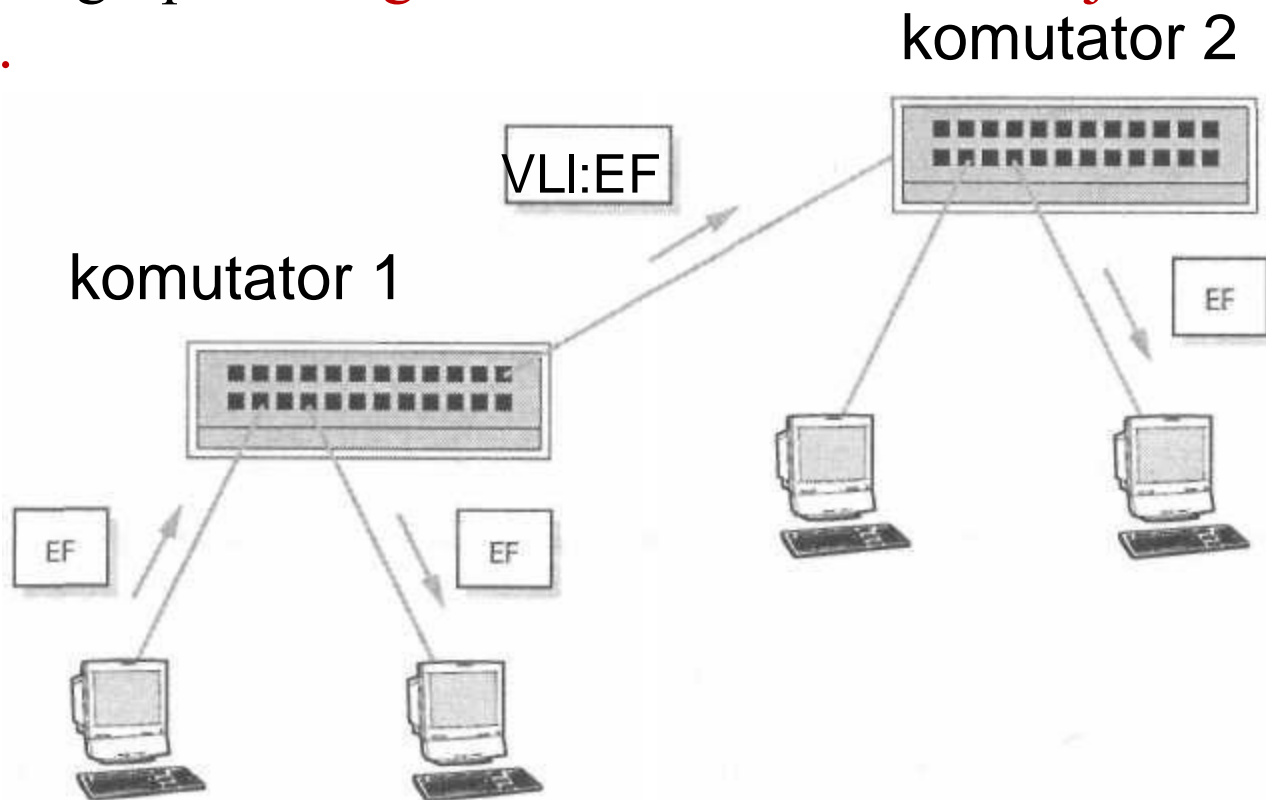
3. U poslednjem koraku utvrđuje se **označeni most** (*designated bridge*) za svaki LAN. To je **most koji eventualno prosleđuje okvire** sa tog LAN-a. Mostovi biraju označeni most **slanjem BPDU-a preko svakog LAN-a na koji su povezani**. Most neće poslati BPDU na LAN **preko ranije utvrđenog root porta**. U stvari, **root port određuje LAN u smeru ka root mostu**. Algoritam sada treba da utvrdi **da li postoje neki LAN-ovi u drugim smerovima**. On to radi tako što LAN prenosi BPDU-e od svojih mostova **koji zahtevaju da budu izabrani za označeni most**. Svaki BPDU **sadrži cenu od mosta koji ga je poslao do root mosta**. Kada most primi BPDU, on **poredi njegovu cenu sa svojom cenom do root mosta**. Ako je njegova cena veća, **zna da neće biti izabran** i odustaje od svojih zahteva. Na kraju, **samo jedan most ostaje sa najnižom cenom** i on postaje označeni most tog LAN-a. U slučaju da postoje dva mosta sa najnižom cenom **izbor se donosi na osnovu ID-a**. Niži ID pobeđuje.

- Nakon biranja označenih mostova za svaki LAN, algoritam je završen.
- Svaki LAN je povezan na svoj **označeni most**, a svaki most može da komunicira sa **root mostom preko svog root porta**.



# IX - Virtuelni LAN-ovi

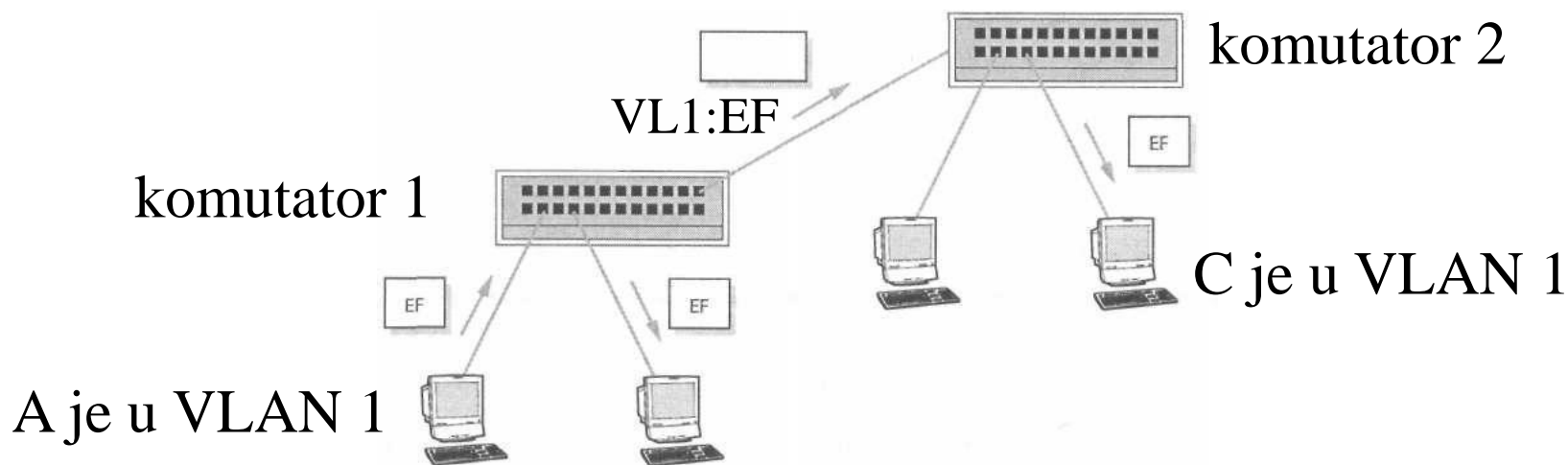
- Virtuelni LAN (VLAN) predstavlja posebno grupisanje određenog broja uređaja u posebne LAN-ove koji su nezavisni jedni od drugih.
- Saobraćaj se odvija preko više linkova pa se tako postiže bolja ravnoteža ukupnog saobraćaja na mreži.
- Uređaji se grupišu u logičke celine nezavisno od njihove fizičke lokacije.



A je u VLAN 1

C je u VLAN 1

# IX - Virtualni LAN-ovi



1. Ethernet okvir (označen kao EF) poslat je do komutatora 1. Komutator 1 **primećuje da okvir koji stigao preko porta ima VL1 oznaku**
2. Port povezan na **C** **takođe ima oznaku VL1**, tako da komutator šalje okvir na taj port.
3. Sledeći port povezuje komutator 1 sa komutatorom 2. Pošto postoje VL1 uređaji u tom smeru, **dodeljeni port takode ima VL1 oznaku.**
4. Komutator 1 **dodaje identifikator VLAN-a u Ethernet okvir** (prikazano kao VL1 :EF) i šalje ga na port komutatora 2.
5. Komutator 2 prima prošireni okvir i **šalje ga direktno do B**. Pošto okvir sadrži VLAN ID, komutator 2 zna kojem VLAN-u pripada.

# IX - Nedostaci mostova

- Mostovi se koriste kod **mreža specijalne namene**. U ostalim slučajevima, zbog razlika koje postoje u algoritmima za rutiranje, komunikacija između medija (mreža) **koji koriste različite protokole je otežana**. Tipično je to za komunikaciju između LAN-ova tipa *Ethernet*, *Token Ring* ili *FDDI*.
- **Odluka o prosleđivanju poruka** (posebno se to odnosi na *spanning tree*) **nije dovoljno sofisticirana**: u slučaju kada u tabeli rutiranja ne postoji zapis koji ukazuje na to gde treba usmeriti paket, **most će proslediti paket svim portovima** sa izuzetkom onog sa koga je paket primljen.
- *Spanning tree* protokol **ne dozvoljava formiranje petlji**.
- Mostovi se uglavnom koriste **za povezivanje mreža istog tipa**. Izuzetak su mostovi koji vrše transliranje.
- Imajući u vidu nabrojane nedostatke (posebno se to odnosi na povezivanje različitih tipova mreža, kakve su *Ethernet* i *Token Ring*) **logički se nameće zaključak** da je efikasnije u čvornim tačkama mreže ugrađivati **uređaje koji će raditi na višim OSI nivoima - ruter i gateway**

# IX - Ruteri

- Mreže šireg geografskog područja (WAN-*Wide Area Networks*) zahtevale su znatno sofisticiranije tehnike povezivanja.
- Sa povećanjem uređaja koji su bili na velikim rastojanjima rastao je i broj puteva – ruta koje su mogle da povežu dva uređaja.
- Ruter ima funkciju da za svaki paket podataka odredi rutu kojom treba taj paket da ide i da taj isti paket prosledi sledećem uređaju u nizu.
- Svaki ruter poseduje znanje koje se odnosi na moguće puteve koje paketi podataka treba da pređu na putu od izvorišta do odredišta.
- “Znanje” o usmeravanju paketa (rutiranju) ugrađuje se u ruting tabelu koja se periodično ažurira i memoriše u bazi podataka rutera.
- Proces u kome se definišu tabele rutiranja naziva se algoritam rutiranja
- Da bi odredio portove kojima se pridružuju odredišne adrese ruter koristi *broadcast* šemu koja koristi **ARP**(*Address Resolution Protocol*).
- Ruter može čitati sadržaje paketa podataka koji pristižu na dati port na osnovu čega može da odredi izvorišnu i odredišnu adresu, zatim tip podataka, kao i vreme prijema paketa.

# IX - Vrste rutiranja podataka

- 1. Centralizovano rutiranje** – podrazumeva da se sve informacije o međusobnim vezama generišu i održavaju **na jednoj centralnoj lokaciji**. Nakon toga se sa te lokacije informacije emituju do svih mrežnih čvorova, tako da svaki čvor može da definiše svoje tabele rutiranja. Jedini način za održavanje informacija o rutiranju na centralnoj lokaciji je preko matrice rutiranja.
- 2. Distribuirano rutiranje** – podrazumeva da nema centralne kontrole. **Svaki čvor mora nezavisno da utvrdi i da održava svoje informacije** o rutiranju. To se obično izvodi tako što zna ko su mu susedi, izračunava cenu puta do svog suseda i utvrđuje cenu za onog koji bi poslao podatke do željenog odredišta.
- 3. Statičko rutiranje** – svaki čvor utvrdi svoju tabelu rutiranja i **nakon toga je više ne menja**.
- 4. Adaptivno rutiranje** – dopušta da se tabele rutiranja **stalno menjaju** u zavisnosti od cene linka tj. stanja linka.



# IX - Algoritmi rutiranja podataka

- 1. Dijkstrin algoritam** – Algoritam učenja unapred koji može da se implementira kao strategija centralnog rutiranja. Može da se koristi i sa rutiranjem zasnovanim na strategiji linka.
- 2. Bellman-Fordov algoritam** – algoritam učenja unazad. Čvorovi uče najjeftinije rute do čvorova od svojih suseda, zajedno sa prvim čvorom na toj ruti. Koristi se na Internetu ili u svakoj velikoj mreži kod koje promene uslova na linkovima zahtevaju ažuriranje tabela rutiranja.
- 3. Rutiranje na osnovu stanja linka** – Čvorovi međusobno saraduju, razmenjujući pakete stanja linka u kojima se nalaze statusne informacije o susednim linkovima. Čvor može da sakupi sve pakete koje primi i utvrđuje mrežnu topologiju. Nakon toga, može da izvrši svoj algoritam za pronalaženje najkraće rute.
- 4. Hijerahijsko rutiranje** – ovo je metod deljenja čvorova na domene ili autonomne sisteme. Unutrašnji protokoli su zaduženi za rutiranje u okviru sistema, a spoljašnji za pronalaženje ruta između različitih sistema. Koristi na Internetu i svim mrežama sa većim brojem čvorova a nije praktičan u situacijama kada svi koriste istu strategiju rutiranja.

# IX – Protokoli rutiranja podataka

- Protokoli za rutiranje omogućuju ruterima **da među sobom razmenjuju podatke** na osnovu kojih mogu da kreiraju tabele za rutiranje.
- Podaci koji se razmenjuju između rutera **sadrže sledeće informacije**:
  - informacije o **topologiji mreže**,
  - informacije o **prekidima**,
  - informacije o **meri “troška” linka**, koja se utvrđuje na osnovu **broja “koraka”** između dve tačke ili na osnovu **propusnog opsega linka** ili **vremenu kašnjenja** na linku ili na osnovu kombinacije ovih parametara
  - informacije o **pouzdanosti linkova**.
- **RIP** (*Routing Information Protocol*) – koristi se za unutrašnje rutiranje sa manjim brojem skokova unutar neke konkretne mreže.
- **OSPF** (*Open Shortest Path First*) – služi za unutrašnje rutiranje; sličan rutiranju zasnovanom na stanju linka ali sa dodatnim karakteristikama koje omogućavaju bolje performanse i veću fleksibilnost.
- **BGP** (*Border Gateway Protocol*) – koristi se za spoljašnje rutiranje koji ruterima omogućava implementiranje specifičnih polisa, ili ograničenja

# IX – Prednosti rutera nad mostovima

- Ako se protokolom **specifira adresa na mrežnom nivou** (*network layer*) tada se taj paket može rutirati (tipično je to za TCP/IP, DECnet i dr.).
- Odluke o prosleđivanju paketa koje se donose od strane rutera variraju **od jednostavnih do jako sofisticiranih**.
- Kod mreža kod kojih se vrši rutiranje poruka **topološke strukture mreže tipa petlje su poželjne**. Nasuprot tome, kod mreža zasnovanih na mostovima, **petlje treba razbiti** (raskinuti) tako da saobraćaj ne prolazi kroz redundantne veze.
- Protokol rutiranja kod mreža zasnovanih na ruterima je taj koji **određuje kako će se saobraćaj usmeravati**. Kod ovih mreža mogu se javiti **rutirajuće petlje**. Određene mere kao što je **brojanje broja preskoka** poruke ili **vreme-života** smanjuju beskonačni kružni obilazak poruke
- **Broadcast** saobraćaj karakterističan za nivo veze ne prolazi kroz ruter.
- Pomoću rutera je **moguće spojiti različite tipove mreža**.
- Ruteri imaju svoje nedostatke koji se pre svega odnose na **smanjenje propusnosti i povećanje latentnosti** kod prenosa poruka.

# IX - Mrežni prolaz (Gateway)

- Postoji **veliki broj autonomnih mreža**-različit hardver i softver.
- Autonomne mreže razlikuju se po: **implementiranim protokolima, algoritmima za rutiranje, procedurama za administriranje, DNS** i dr.
- Korisnici jedne mreže često **žele da komuniciraju sa drugom mrežom**.
- Kada su dve ili veći broj mreža uključene u aplikaciji, **režim rada između ove dve mreže nazivamo umrežavanje** (*internetworking*).
- Često se pojam *Internet* ili *internetwork* koristi za složene mreže
- **Uređaj koji međusobno povezuje dve mreže** naziva se IS (*Intermediate System*) ili IWU (*InterWorking Unit*).
- S obzirom da uređaj **obezbeđuje vezu između dve autonomne mreže** on se još naziva i **gateway** ili mrežni prolaz (**hardver i/ili softverski paket**).
- Omogućava **komunikaciju između različitih arhitektura i okruženja**.
- Vrš **prepakivanje i pretvaranje podataka** koji se razmenjuju između potpuno različitih mreža, tako da svaka od njih može razumeti podatke
- Mrežni prolaz je **obično namenski računar**, koji mora biti sposoban **da podrži oba okruženja koja povezuje** kao i proces prevođenja podataka
- Zahteva **značajne količine RAM-a** za čuvanje i obradu podataka.

Hvala na pažnji !!!



Pitanja

? ? ?