

UNIVERZITET CRNE GORE

# RAČUNARSKE MREŽE

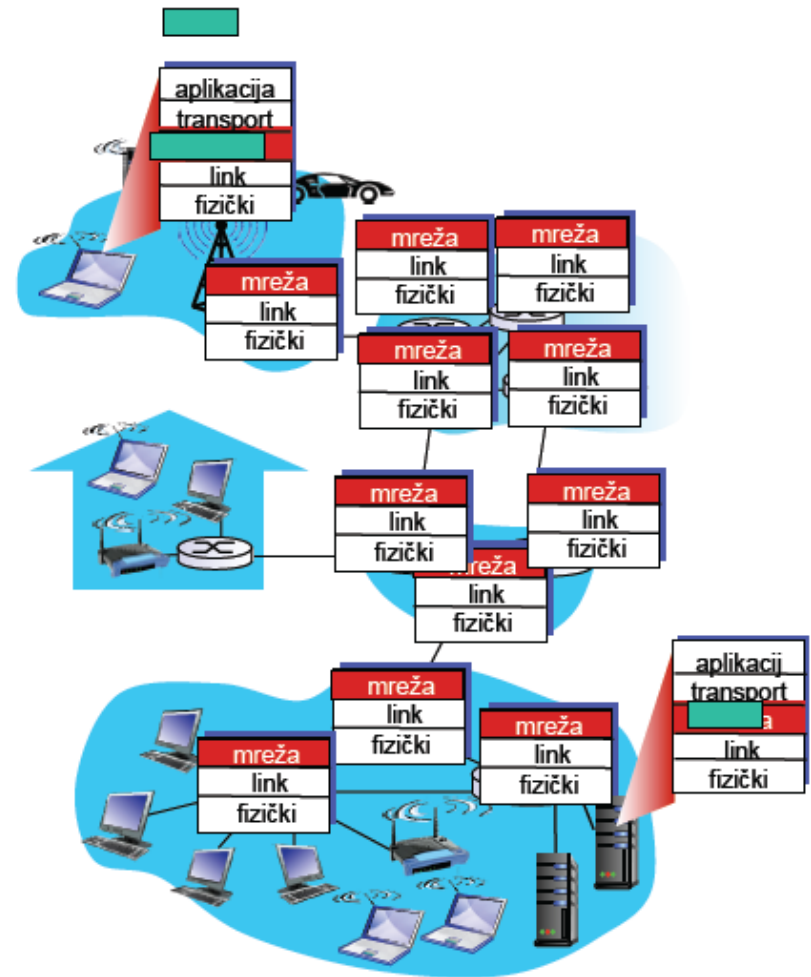
**Doc. dr Uglješa Urošević**

ugljesa@ucg.ac.me

Nivo mreže

# Mrežni nivo

- ❑ Prenos segmenta od pošiljaoca do odredišta
- ❑ Na strani koja šalje enkapsuliraju se segmenti u datagrame
- ❑ Na strani prijema predaja segmenata transportnom nivou
- ❑ Protokoli mrežnog nivoa su implementirani u svakom hostu, ruteru
- ❑ Ruter ispituje polja zaglavlja svakog IP datagrama kojeg prosleđuje



# Ključne funkcije mrežnog nivoa

- *prosleđivanje*:  
pomjeranje paketa sa ulaza rutera na odgovarajući izlaz
- *rutiranje*: izbor rute kojom se paketi prenose od izvora do destinacije.
  - *Algoritmi rutiranja*

## analogija:

- *rutiranje*: proces planiranja putovanja
- *prosleđivanje*: proces prolaska kroz jednu raskrnicu

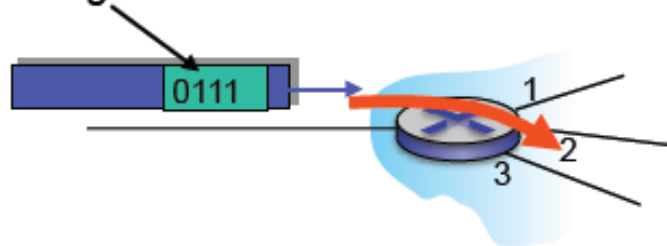
# Mrežni nivo: ravan podataka, ravan kontrole

## Ravan podataka

- Lokalna funkcija rutera
- Determiniše kako se datagram koji dolazi na ulazni port rutera prosleđuje na izlazni port

- Funkcija prosleđivanja

Vrijednosti u zaglavlju datagrama

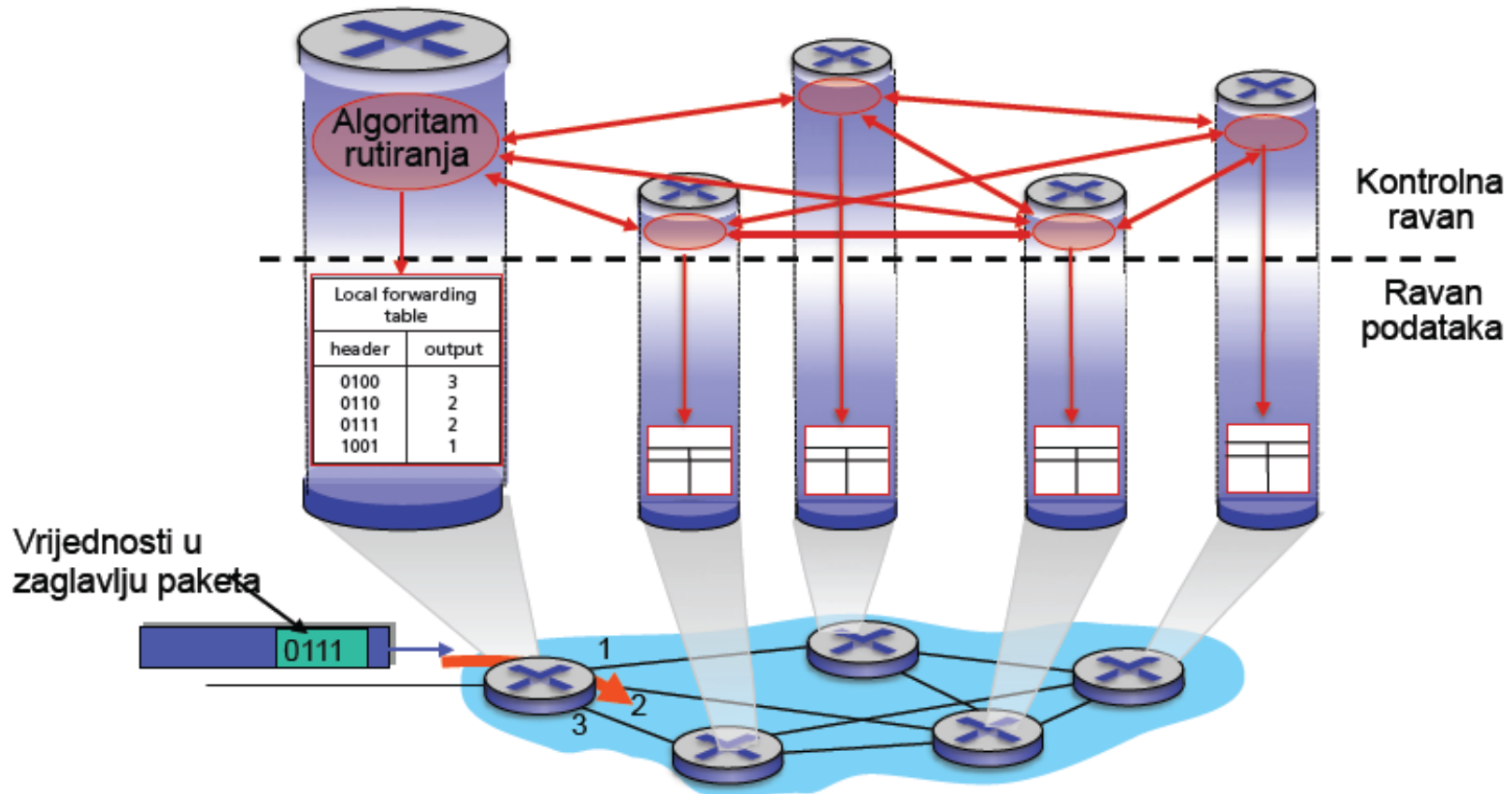


## Kontrolna ravan

- Mrežna logika
- Određuje kako se datagram rutira duž putanje od kraja do kraja od izvorišnog do odredišnog hosta
- Dva pristupa:
  - *Tradicionalni algoritmi rutiranja*: implementirani u ruterima
  - *software-defined networking (SDN)*: implementirani u udaljenim serverima

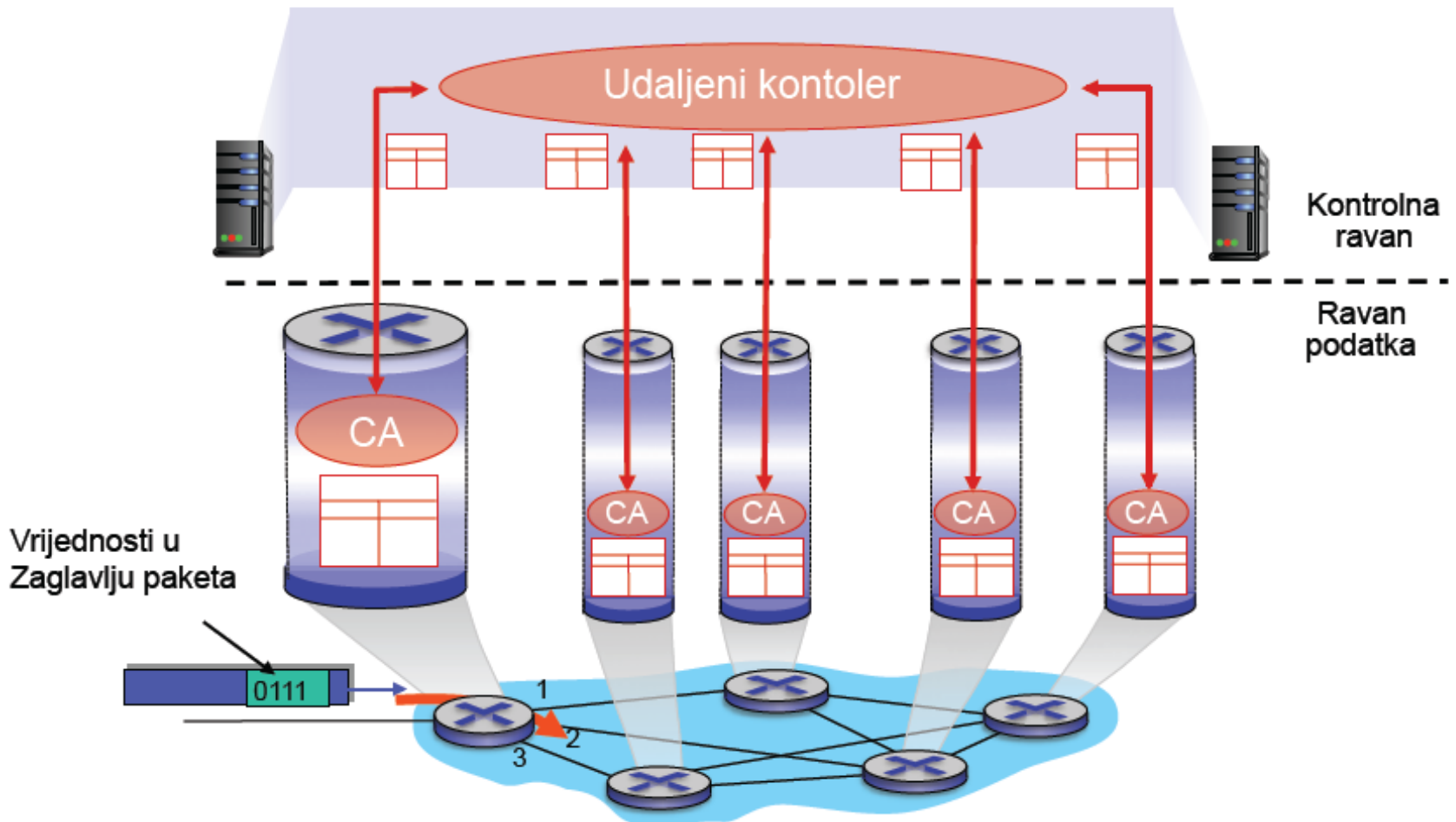
# Distribuirana kontrolna ravan

Individualni algoritmi rutiranja se izvršavaju samostalno *u svakom ruteru* i interaguju u kontrolnoj ravni



# Centralizovana kontrolna ravan

Udaljeni kontroler interaguje sa lokalnim kontrolnim agentima (CAs)



# Mrežni servisni model

**Pitanje:** Koji *servisni model* nudi “kanal” koji transportuje datagrame od pošiljaoca do prijemnika?

## Primjer servisa za individualne datagrame:

- Garantovana predaja
- Garantovana predaja sa kašnjenjem manjim od određene vrijednosti (recimo 40ms)

## Primjer servisa za tok datagrama:

- Redosledna predaja datagrama
- Garantovani minimalni protok toka
- Ograničene promjene u međupaketskim intervalima
- Nivo zaštite



# Modeli servisa mrežnog nivoa:

Mrežna Arhitektura	Model Servisa	Opseg	Garantovani ?			“Congestion Feedback”
			Gub.	Red.	Tajm.	
Internet	best effort	bez	ne	ne	ne	ne (preko gubitaka)
ATM	CBR	konstantna brzina	da	da	da	nema zagušenja
ATM	VBR	garantov. brzina	da	da	da	nema zagušenja
ATM	ABR	garantov. minimum	ne	da	ne	da
ATM	UBR	bez	ne	da	ne	ne

- Internet model se proširuje sa: Intserv, Diffserv

# Konektivni i nekonektivni servis

- Datagram mreža obezbeđuje nekonektivni servis nivoa mreže
- VC mreža obezbeđuje konektivni servis nivoa mreže (VC - Virtuelna kola)
- Analogno servisima nivoa transporta, ali:
  - **Servis:** host-to-host
  - **Bez izbora:** mreža obezbeđuje jedan ili drugi
  - **Implementacija:** na okosnici

# Virtuelna kola

“Veza izvor-destinacija” se skoro ponaša kao telefonska veza

- Vodi računa o performansama
- Mreža učestvuje u uspostavljanju veze

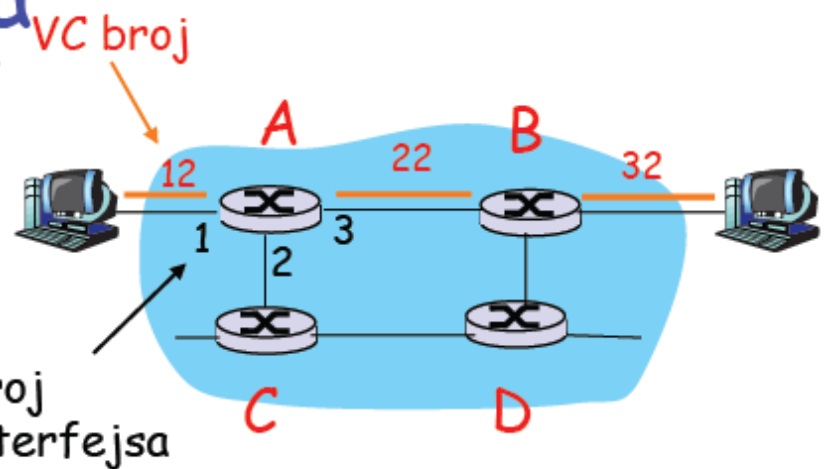
- Uspostavljanje i raskidanje veze se obavlja prije početka prenosa podataka
- Svaki paket u zaglavlju nosi VC identifikator (ne adresu destinacije)
- *Svaki* ruter na putanji vodi računa o “stanju” svake konekcije koja prolazi kroz njega
- Resursi linka i rutera (kapacitet, baferi) mogu biti *dodijeljeni* VC (dodijeljeni resursi = predviđenim servisima)

# VC implementacija

VC se sastoji od:

1. Puta od izvora do destinacije
  2. VC identifikatora, po jedan za svaki link na putu
  3. Sadržaja tabela prosleđivanja u ruterima duž puta
- Paket koji pripada VC nosi VC identifikator, a ne destinacionu adresu.
  - VC identifikator se može mijenjati na svakom linku.
    - Novi VC identifikator se dobija iz tabele prosleđivanja

# Tabela prosleđivanja



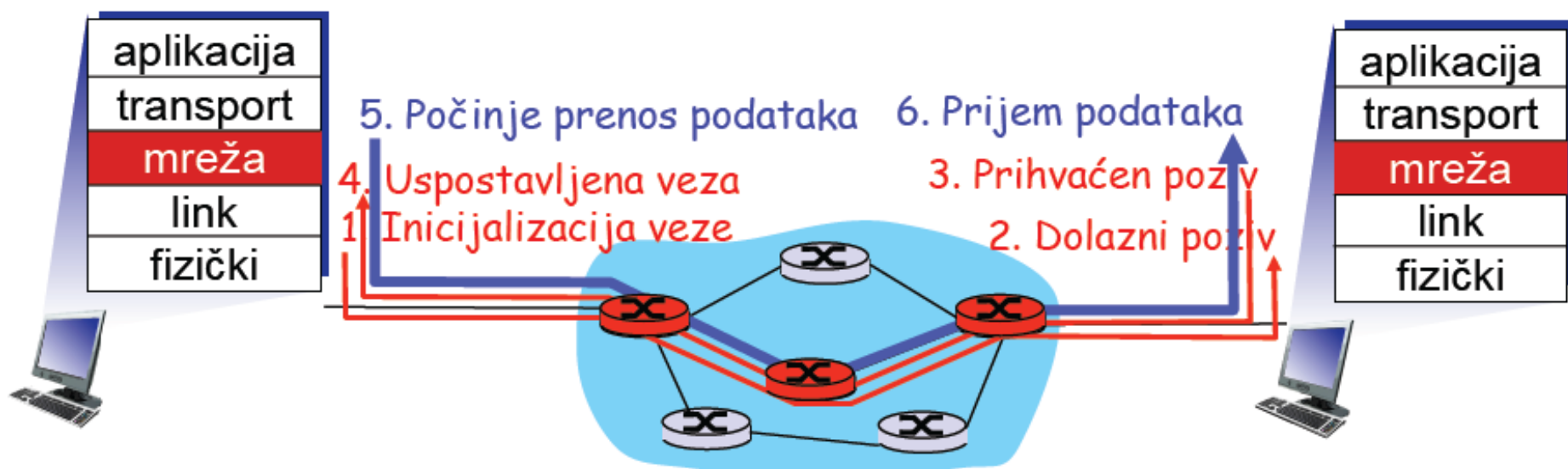
## Tabela prosleđivanja rutera A:

Dolazni interfejs	Dolazni VC #	Odlazni interfejs	Odlazni VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Ruteri moraju nadzirati informaciju o stanju veze!

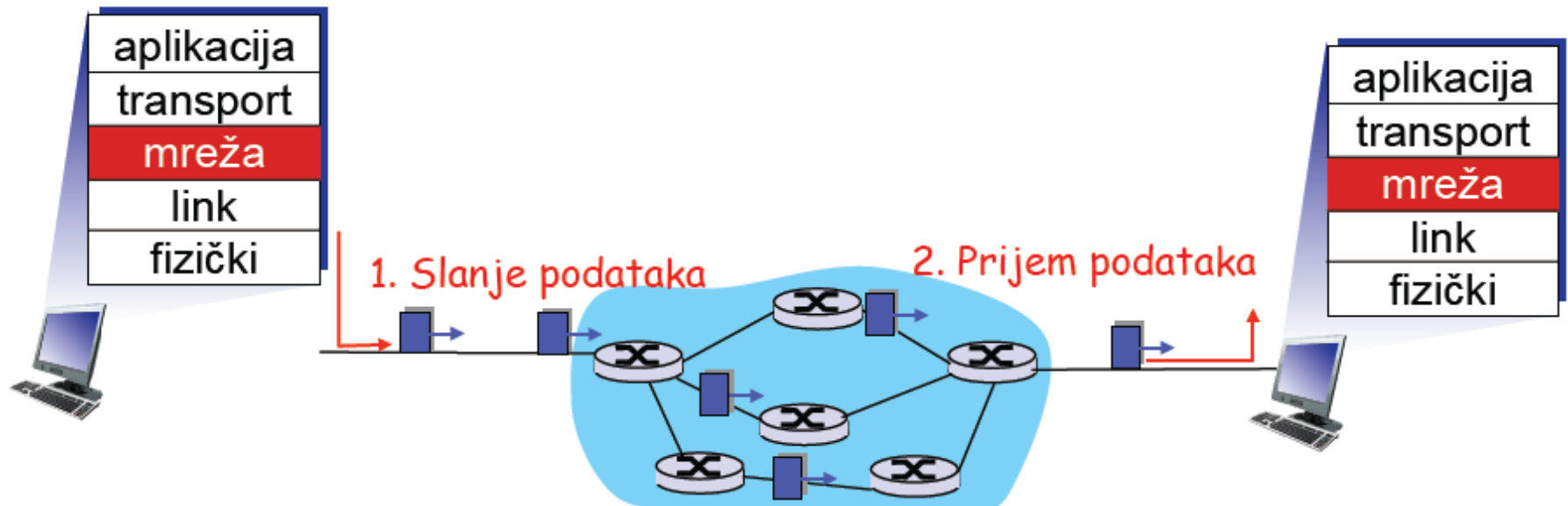
# Virtuelna kola: protokoli signalizacije

- Koriste se za uspostavljanje, održavanje i raskidanje VC
- Koristi se u ATM, frame-relay, X.25
- Ne koristi se u ovakvom u obliku na današnjem Internetu



# Datagram mreže: Internet model

- Nema uspostavljanja veze na nivou mreže
- ruteri nemaju informaciju o vezama od kraja do kraja
  - Na mrežnom nivou nema koncepta “veze”
- Paketi se prosleđuju korišćenjem adrese destinacionog hosta
  - Paketi između istog para izvor-destinacija mogu se prenositi različitim putanjama



# Tabela prosleđivanja

4 milijarde  
mogućih adresa  
(IPv4)

Opseg destinacionih adresa	<u>Link Interfejs</u>
11001000 00010111 00010000 00000000 do 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 do 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 do 11001000 00010111 00011111 11111111	2
ostalo	3



# Izbor najdužeg prefiksa

<u>Prefiks</u>	<u>Link Interfejs</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
inače	3

## Primjeri

DA: 11001000 00010111 00010110 10100001

Koji interfejs?

DA: 11001000 00010111 00011000 10101010

Koji interfejs?

# Datagram ili mreža sa virtuelnim kolima?

## Internet (datagram)

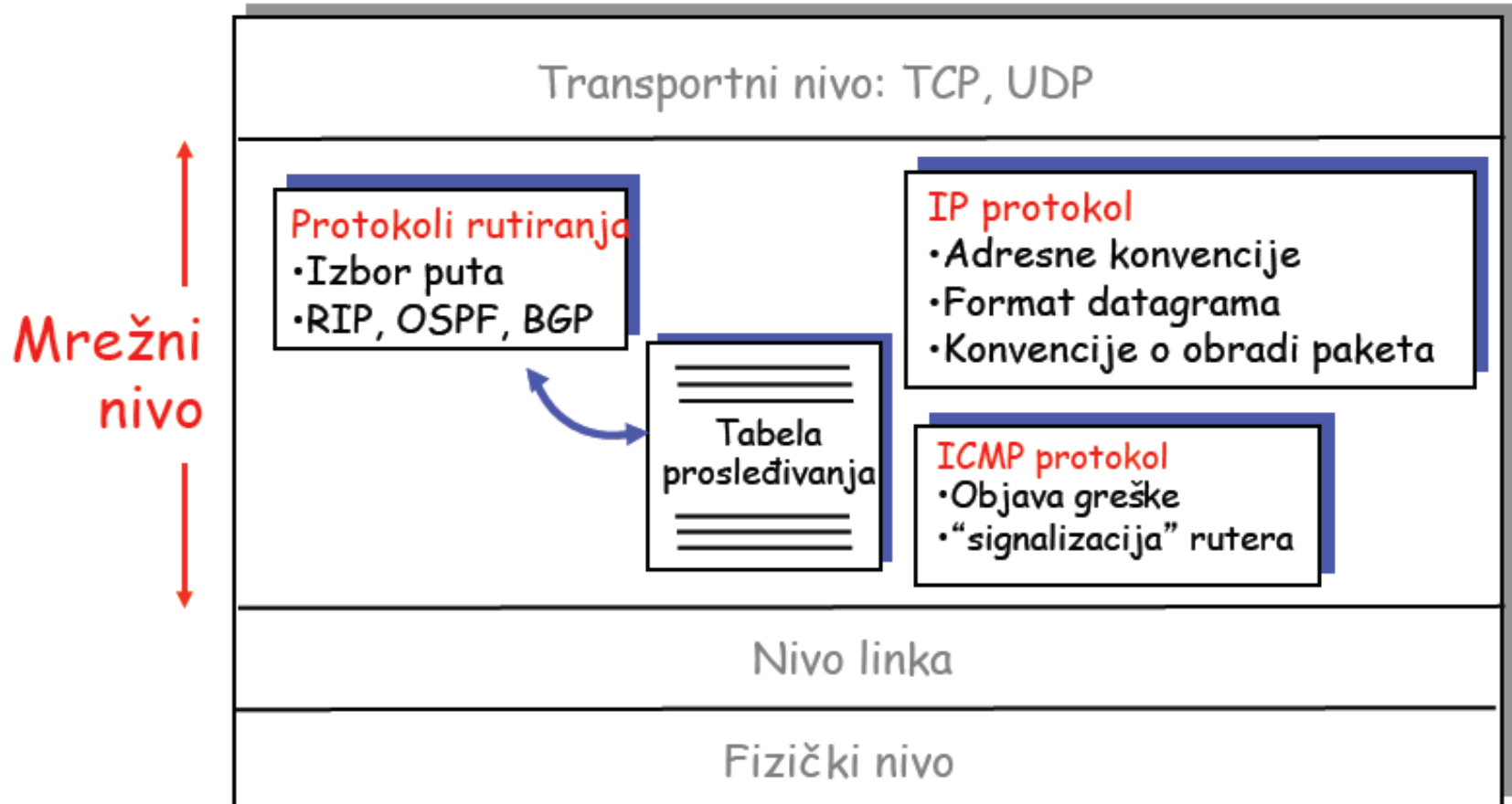
- Podaci se razmjenjuju između računara
  - “elastičan” servis, bez striktnih tajming zahtjeva
- “pametni” krajnji sistemi (računari)
  - Mogu se adaptirati, obavljati kontrolu, oporavljati od greške
  - Jednostavna mreža, kompleksnost na “ivici”
- Mnogo tipova linka
  - Različite karakteristike
  - Teškoće uniformnog servisa

## ATM (VC)

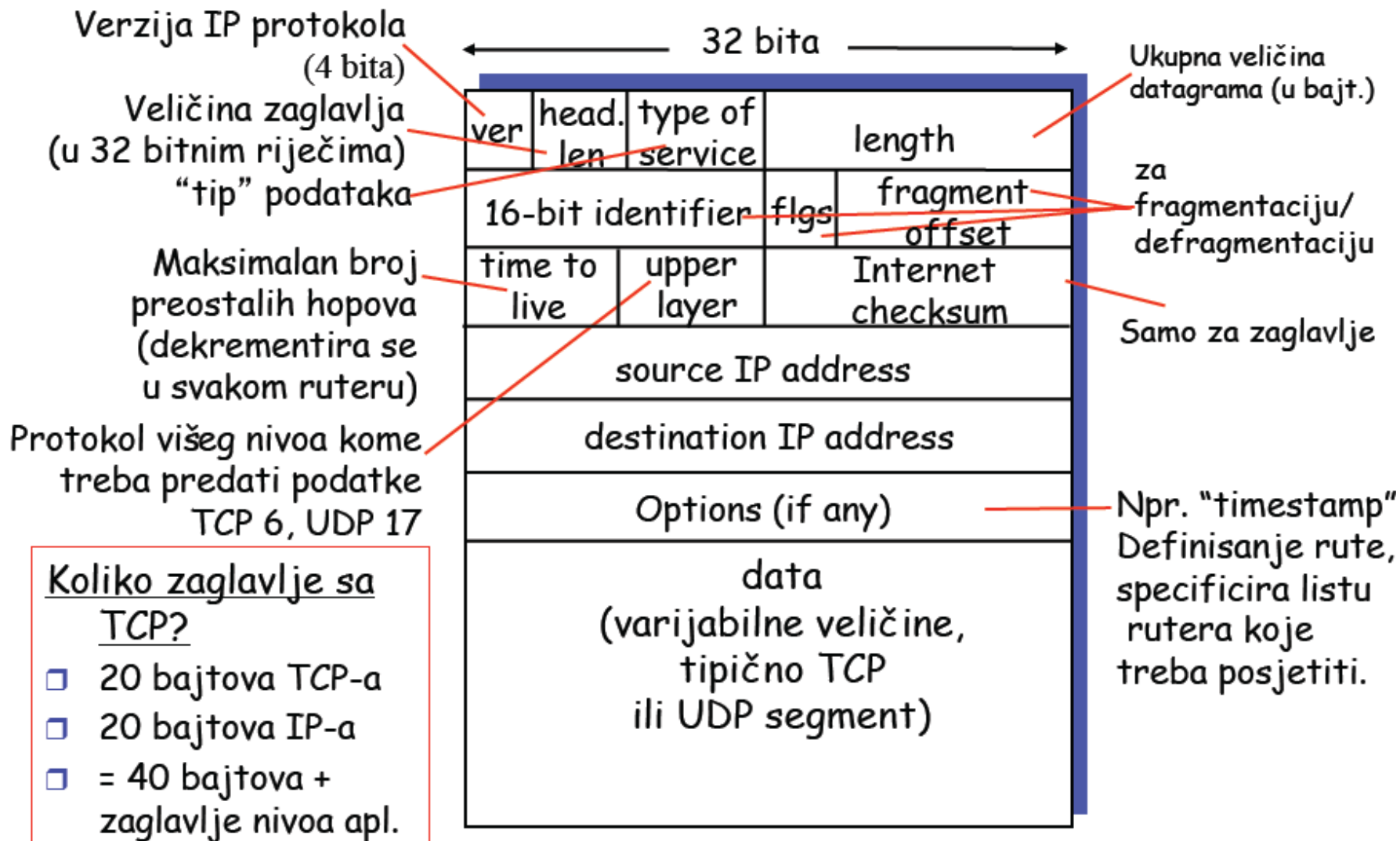
- Razvijen iz telefonije
- Ljudska konverzacija:
  - striktan tajming, zahtjevi za pouzdanošću
  - Potreba za garantovanim servisom
- “glupi” krajnji sistemi
  - telefoni
  - kompleksnost unutar mreže

# Internet mrežni nivo

Host, ruter funkcije mrežnog nivoa:

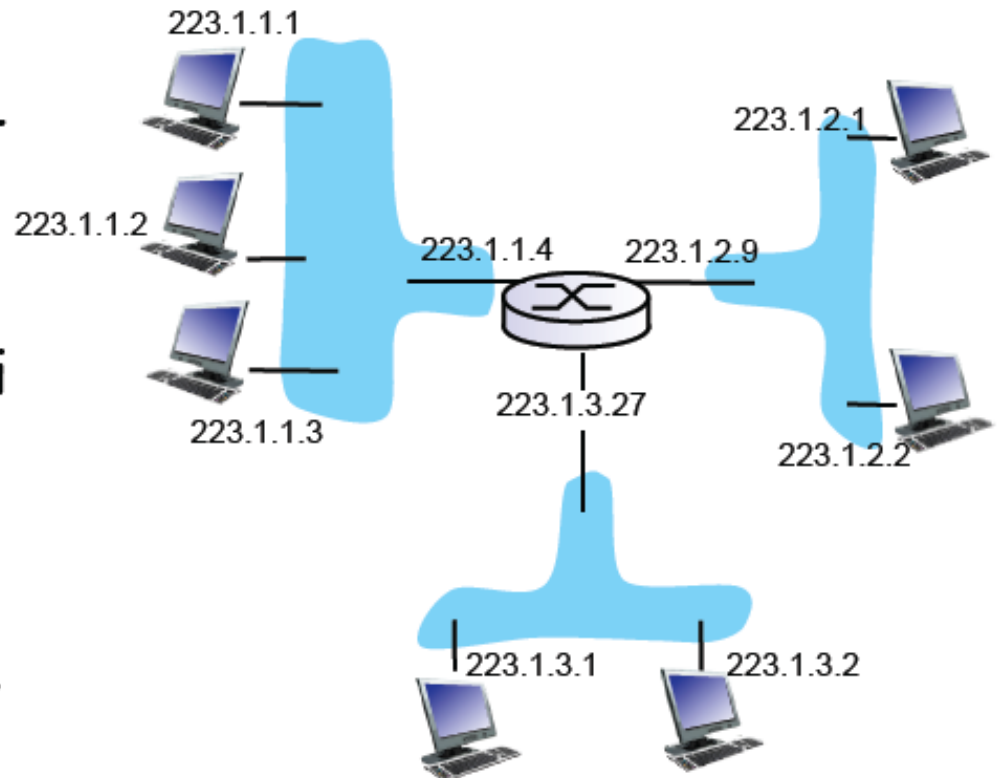


# Format IP datagrama



# IP Adresiranje: uvod

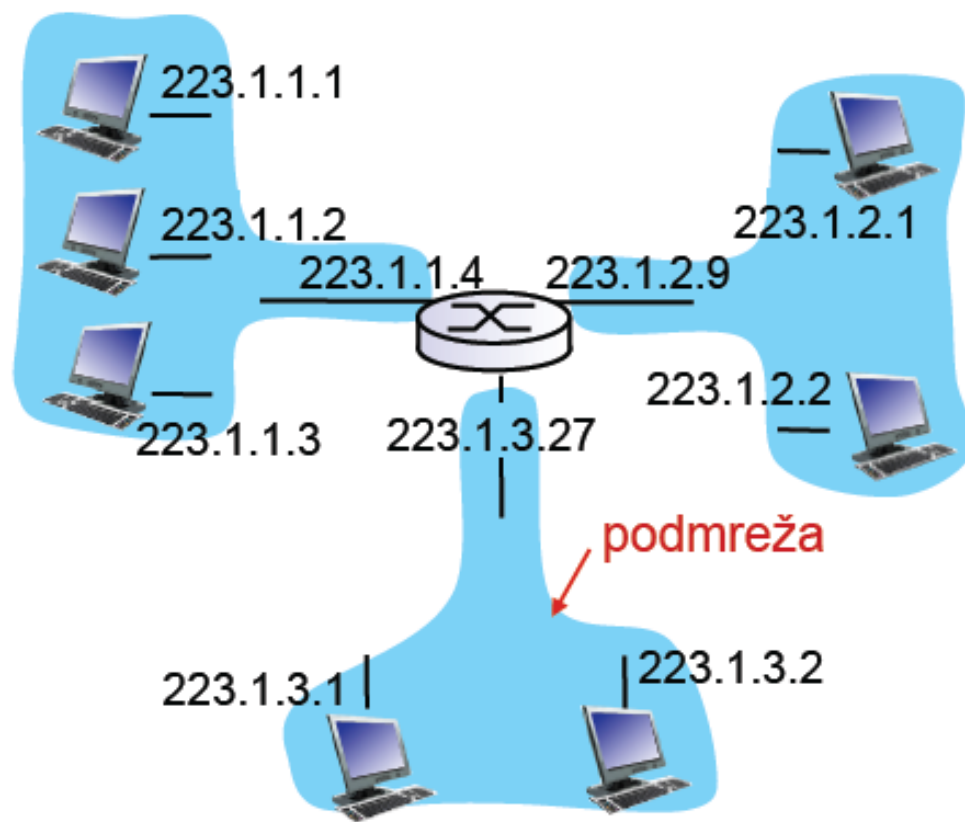
- IP adresa: 32-bitni identifikator za host ili ruter *interfejs*
- *interfejs*: veza između host/rutera i fizičkog linka
  - Ruteri imaju više od jednog interfejsa
  - i host može imati više interfejsa
  - IP adrese su vezane za svaki interfejs



$$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_1 \underbrace{00000001}_1 \underbrace{00000001}_1$$

# IP Adresiranje

- IP adresiranje:
  - Mrežni dio (biti višeg reda)
  - Dio hosta (biti nižeg reda)
- Šta je mreža? (iz perspektive IP adrese)
  - Interfejsi uređaja sa istim mrežnim dijelom IP adrese
  - mogu fizički dosegnuti jedni druge bez učešća rutera

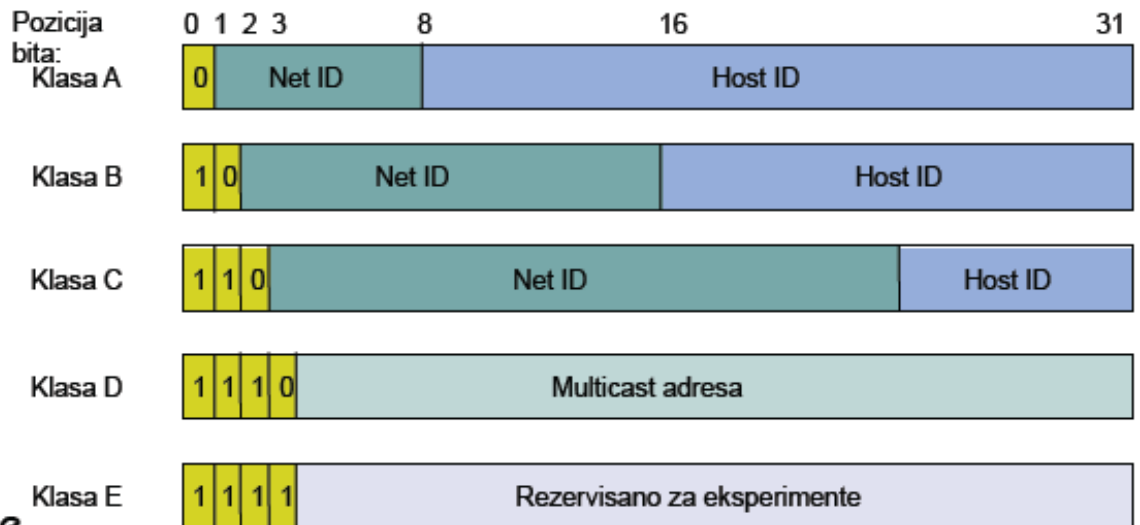


Mreža se sastoji od 3 IP podmreže (prvih 24 bita su mrežna adresa)

# Classful IP Adresiranje

- IPver4 adresna struktura je podijeljena na pet adresnih klasa: A, B, C, D i E, identifikacijom najznačajnijih bita adrese kao što je prikazano na slici.

- Klasa A ima 7 bita za mrežni ID i 24 bita za host ID, što znači  $2^7 - 2 = 126$  mreža i  $2^{24} - 2 = 16777214$  hostova. U klasu A spadaju adrese čiji je prvi bit uvijek 0. Ova klasa je namijenjena velikim organizacijama. Opseg validnih mrežnih adresa klase A je od 1.0.0.0 do 126.0.0.0.



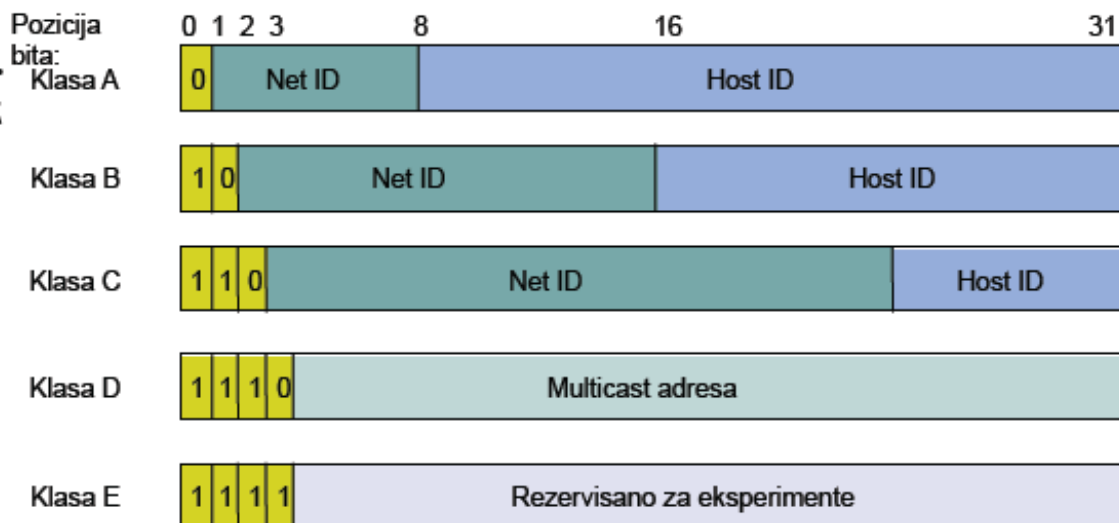


# Classful IP Adresiranje

Klasa B ima 14 bita za mrežni

- ID, što znači  $2^{14}-2=16382$  mreža i  $2^{16}-2=65534$  hostova. U klasu B spadaju adrese čija su prva dva bita uvijek 10. Ova klasa je namijenjena organizacijama srednje veličine. Opseg validnih mrežnih adresa klase B je od 128.1.0.0 do 191.254.0.0.

- Klasa C ima 21 bit za mrežni ID i 8 bita za host ID, što znači  $2^{21}-2=2097150$  mreža i  $2^8-2=254$  hostova. U klasu C spadaju adrese čija su prva tri bita uvijek 110. Ova klasa je namijenjena malim organizacijama. Opseg validnih mrežnih adresa klase C je od 192.0.1.0 do 223.255.254.0.

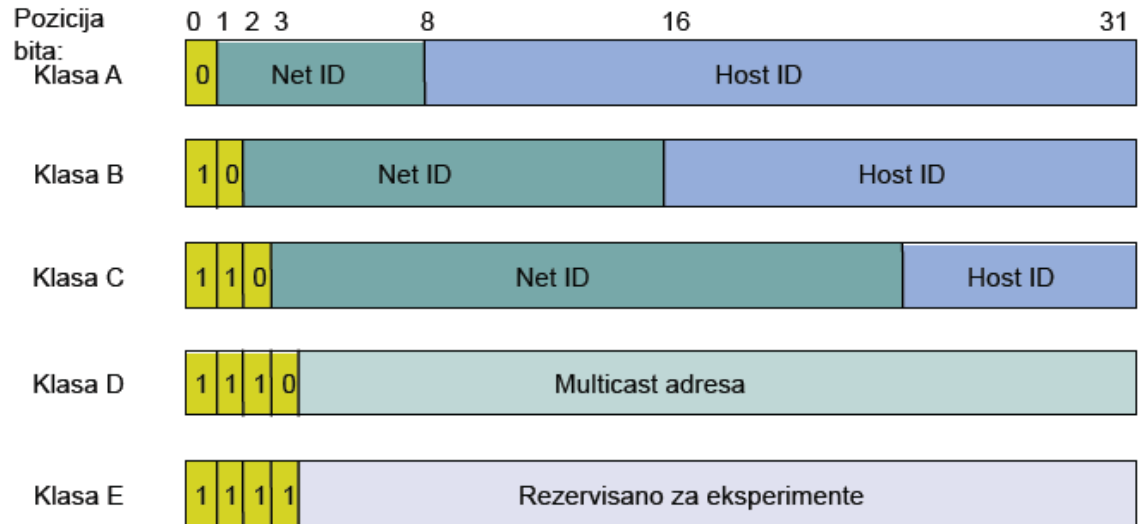




# Classful IP Adresiranje

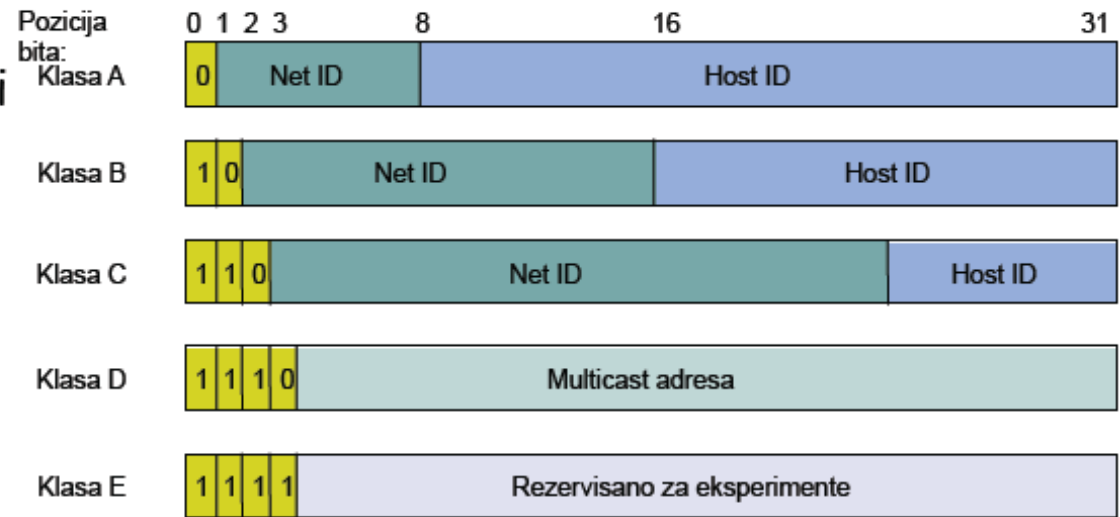
□ Klasa D se koristi za multikast servis koji omogućava da host šalje paket grupi hostova koji pripadaju istoj multikast grupi. U klasu D spadaju adrese čija su prva četiri bita uvijek 1110. Ova klasa je namijenjena za multicast grupe. Opseg adresa koji pripadaju ovoj klasi je od 224.0.0.0 do 239.255.255.255. Ove adrese nijesu za komercijalnu upotrebu.

□ Klasa E je rezervisana za eksperimente. U klasu E spadaju adrese čiji su prvih pet bita uvijek 11110. Ova klasa je namijenjena za multicast grupe. Opseg adresa koji pripadaju ovoj klasi je od 240.0.0.0 do 254.255.255.255. Ove adrese takođe nijesu za komercijalnu upotrebu.



# Classful IP Adresiranje

- ID koji imaju sve jedinice i sve nule imaju specijalnu namjenu.
- Host ID koji se sastoji od svih jedinica znači da se paket *broadcast*-uje svim hostovima mreže čiji je mrežni ID specificiran.
- Ako se mrežni ID sastoji od svih jedinica to znači da se paket *broadcast*-uje svim hostovima lokalne mreže. Svim podmrežama.
- Host ID koji se sastoji od svih 0 odgovara adresi mreže.



# Classful IP Adresiranje

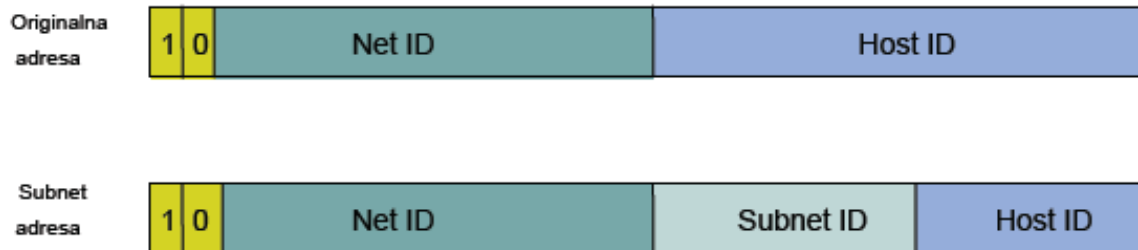
- IP adrese se najčešće pišu u formi tačka-decimalnog zapisa koji je pogodan za korišćenje od strane čovjeka. Adresa se dijeli na četiri bajta, pri čemu svaki bajt predstavlja decimalni broj, koji su razdvojeni tačkama. Na primjer adresa
- 10000000            10000111            01000100            00000101  
    ○ 128            .            135            .            68            .            5
- Klasa adrese se lako određuje ispitivanjem prvog okteta adrese. U IP adresi 128.135.68.5 prvi oktet je 128. Kako 128 pada između 128 i 191, jasno je da je ovo IP adresa klase B.

# Classful IP Adresiranje

- ❑ Određeni opsezi adresa su namijenjeni za privatne mreže (RFC1918).
- ❑ Ove adrese se koriste unutar mreža koje se ne vezuju direktno na Internet ili u mrežama u kojima je implementiran NAT.
- ❑ Ove adrese nijesu registrovane i ruteri na Internetu moraju odbacivati pakete sa ovakvim adresama. Opsezi privatnih adresa su: 10.0.0.0 - 10.255.255.255 (A klasa), 172.16.0.0 - 172.31.255.255 (B klasa) i 192.168.0.0 - 192.168.255.255 (C klasa - najčešće se primjenjuje u kućnim mrežama)

# Classful IP Adresiranje

- ❑ Opisano IP adresiranje ima više nedostataka.
- ❑ Ovo adresiranje može biti vrlo neefikasno. Na primjer, dodjela B klase jednoj akademskoj instituciji koja ima jednu ili više lokalnih računarskih mreža je besmislena.
- ❑ Rješenje ovog problema je razvijeno 1980-tih kada je usvojen koncept podmreže (subnetting) kada sa dodaje još jedan hijerarhijski nivo subnet (podmreža).
- ❑ Sjajna stvar ovog koncepta je njena transparentnost na Internetu. Naime, Internet "vidi" i dalje samo dva nivoa hijerarhije. Unutar intraneta mrežnom administratoru se ostavlja mogućnost kombinovanja veličina subnet i host polja.



# Classful IP Adresiranje

- To znači da dodijeljena mrežna adresa može biti podijeljena na više podmreža. Tako na primjer, 172.16.1.0, 172.16.2.0 i 172.16.3.0 predstavljaju podmreže mreže 171.16.0.0.
- Adresa podmreže se dobija "posuđivanjem" bita iz dijela koji se odnosi na host i njihovo dodjeljivanje podmreži.
- Broj "posuđenih" bita iz dijela koji se odnosi na host varira i zavisi od maske podmreže (subnet mask).
- Maska podmreže ima isti format i koncepciju kao i IP adrese. Razlika je u tome što sve jedinice označavaju polja koja pripadaju mreži i podmreži, dok 0 specificiraju polje adrese koje pripada hostu.







# Classful IP Adresiranje

172	16	125	1	zadata adresa u decimalnom formatu
10101100	00010000	01111101	00000001	zadata adresa u binarnom formatu
255	255	0	0	default maska pod mreže u decimalnom formatu
11111111	11111111	00000000	00000000	default maska pod mreže u binarnom formatu
10101100	00010000	0	0	adresa mreže u binarnom formatu
<b>172</b>	<b>16</b>	<b>0</b>	<b>0</b>	<b>adresa mreže u decimalnom formatu</b>
<hr/>				
172	16	125	1	zadata adresa u decimalnom formatu
10101100	00010000	01111101	00000001	zadata adresa u binarnom formatu
255	255	224	0	zadata maska pod mreže u decimalnom formatu
11111111	11111111	11100000	00000000	zadata maska pod mreže u binarnom formatu
10101100	00010000	01100000	00000000	adresa pod mreže u binarnom formatu
172	16	96	0	<b>adresa pod mreže u decimalnom formatu</b>

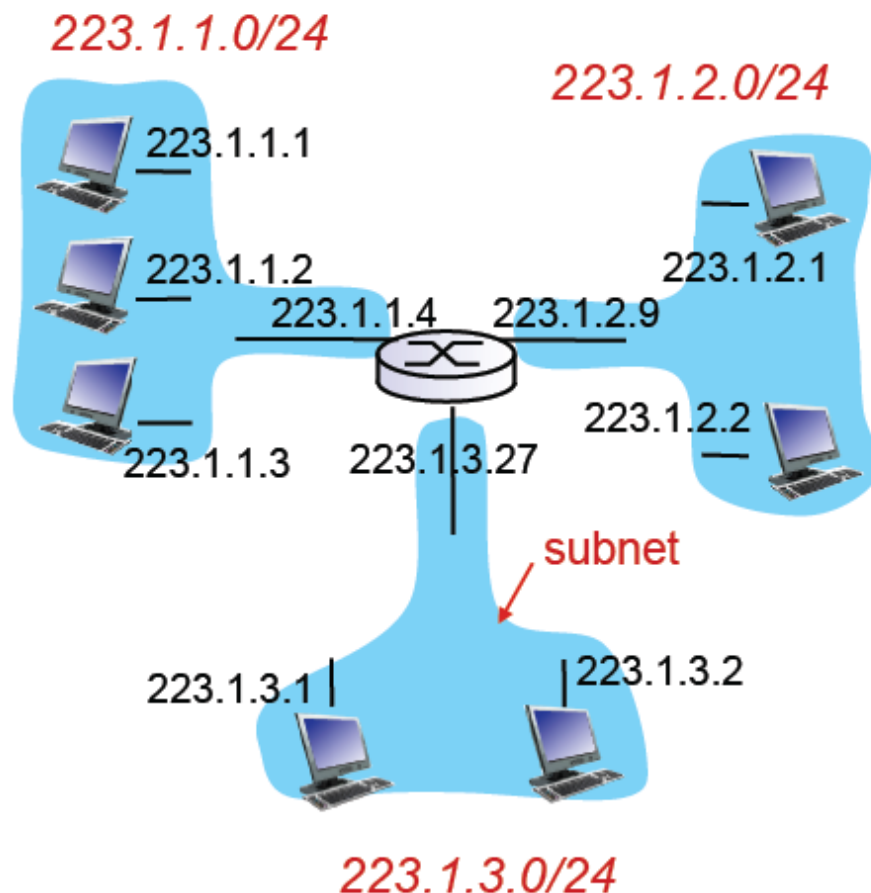
Broadcast adresa u ovoj pod mreži je 172.16.127.255 (10101100.00010000.01111111.11111111). Opseg adresa koje pripadaju ovoj pod mreži je od 172.16.96.1 (10101100.00010000.01100000.00000001) do 172.16.127.254 (10101100.00010000.01111111.11111110).



# Podmreža

## Napomena

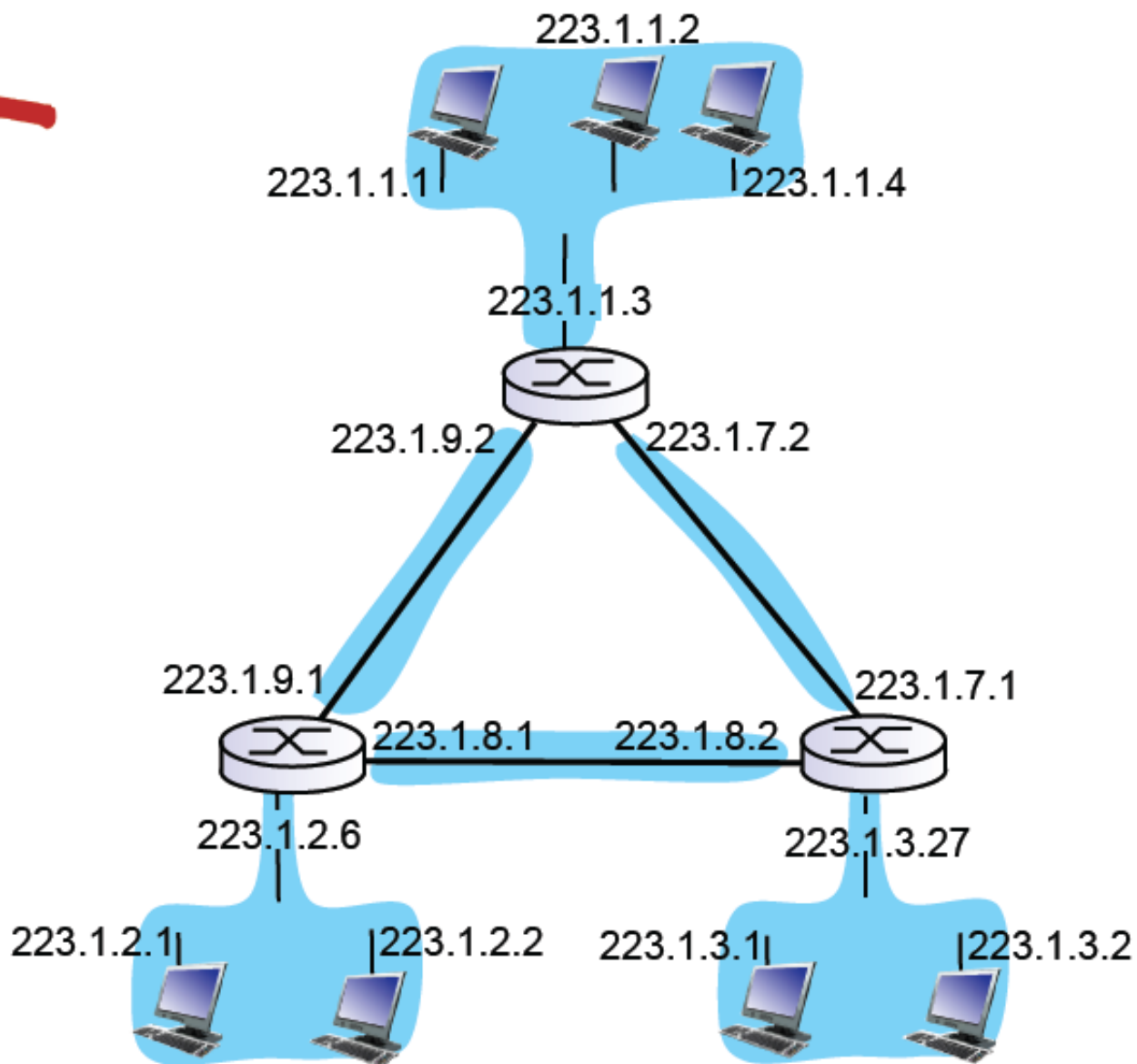
- Da bi odredili podmreže, treba razdvojiti svaki interfejs od njegovog hosta ili rutera, kreirajući ostrva izolovanih mreža. Svaka izolovana mreža se zove **podmreža**.



Maska podmreže: /24

# Podmreže

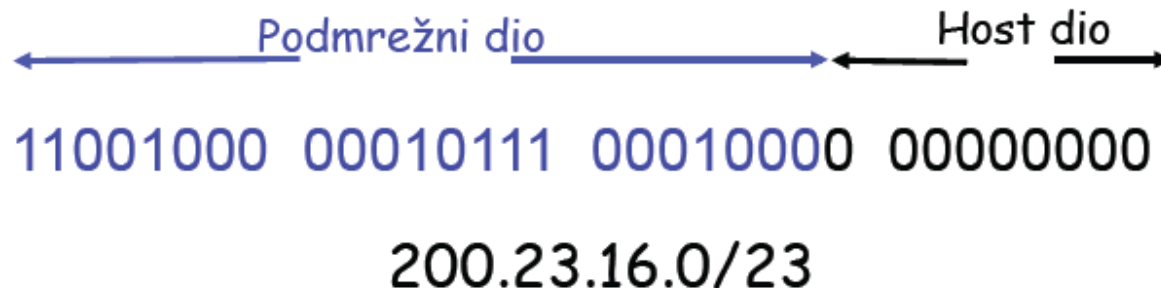
Koliko ih je?



# IP adresiranje: CIDR

## □ CIDR: Classless InterDomain Routing

- Podmrežni dio adrese je proizvoljne veličine
- Format adrese: **a.b.c.d/x**, gdje je x broj bita u mrežnom dijelu adrese



# IP adrese: kako dobiti IP adresu?

P: Kako *host* dobija IP adresu?

- "hard-coded" od strane sistem administratora u fajlu
  - Win1: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- **DHCP: Dynamic Host Configuration Protocol:** dinamički dobija adresu sa servera
  - "plug-and-play"

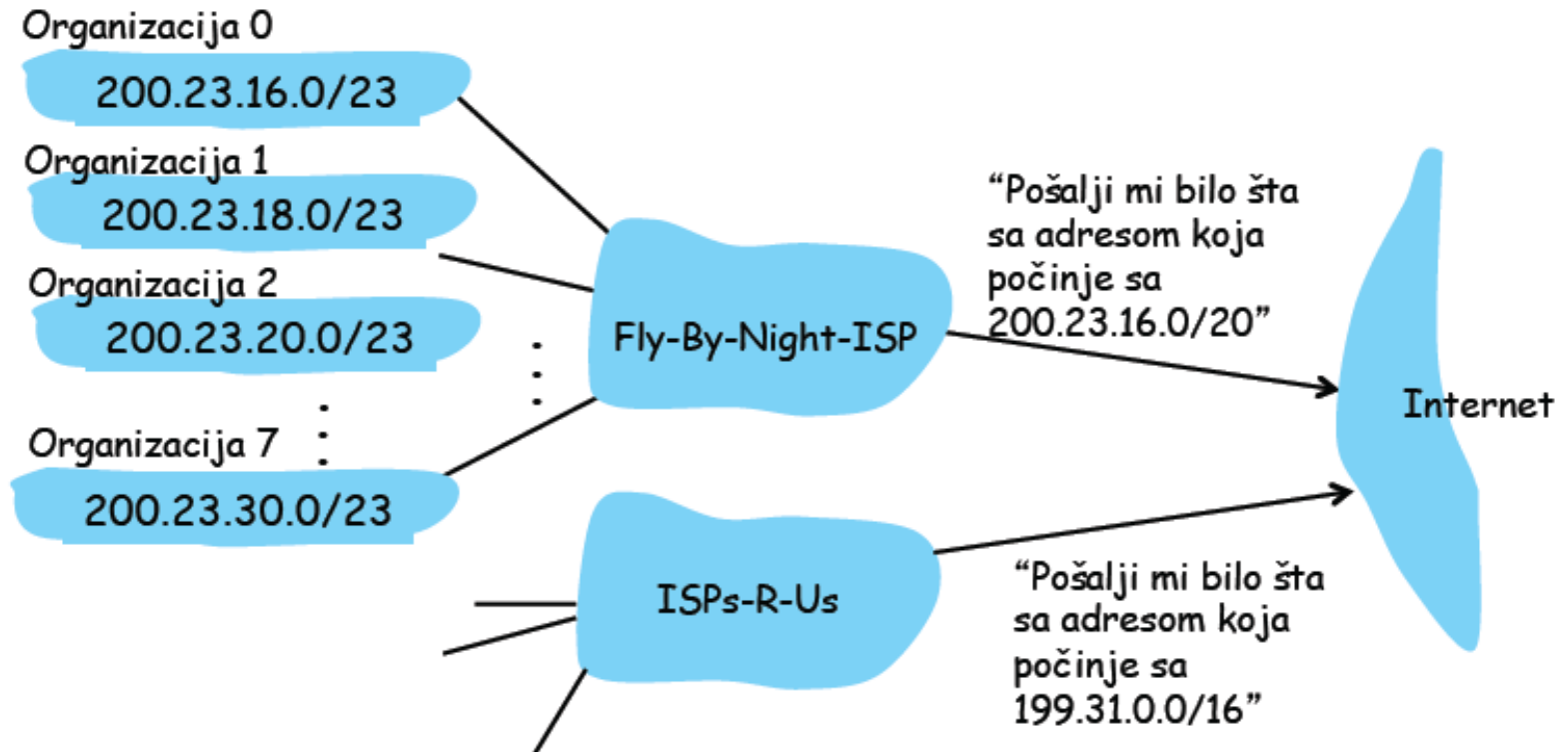
# IP adrese: kako dobiti IP adresu?

**P:** Kako mreža dobija podmrežni dio IP adrese?

ISP-ov blok	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Organizacija 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Organizacija 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Organizacija 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...	.....		....	....	
Organizacija 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

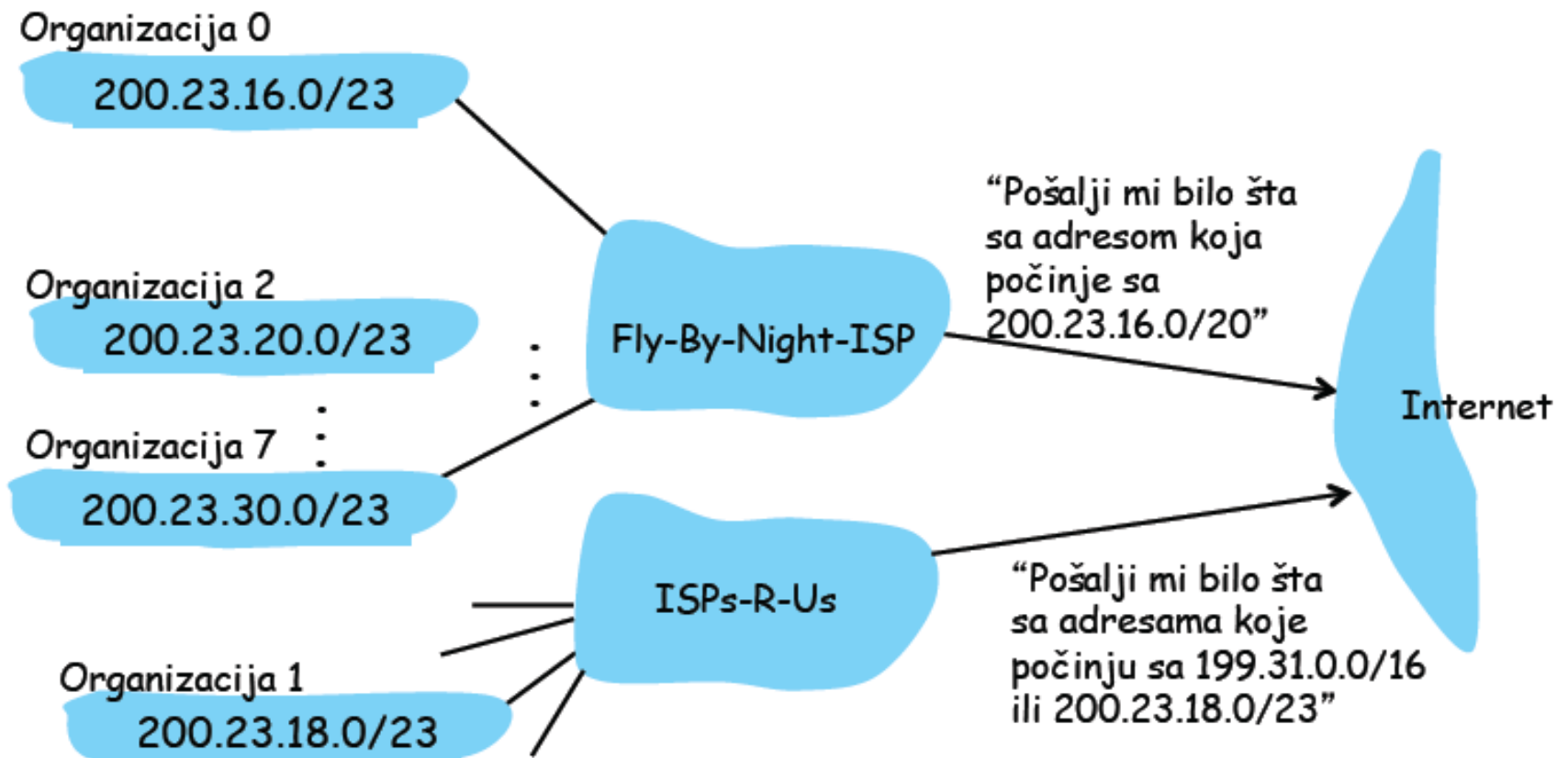
# Hijerarhijsko adresiranje: agregacija ruta

Hijerarhijsko adresiranje dozvoljava efikasno oglašavanje informacije potrebne za rutiranje:



# Hijerarhijsko adresiranje: specifičnije rute

ISPs-R-Us ima više specifičnih ruta do Organizacije 1



## IP adresiranje:

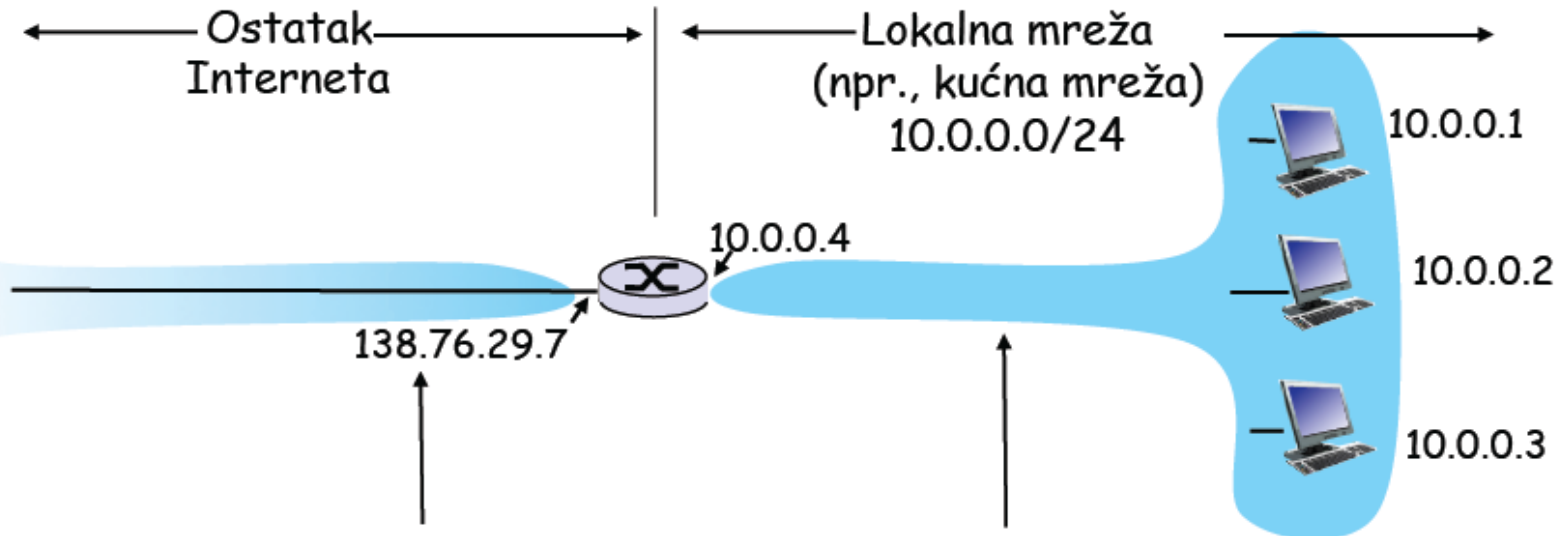
P: Kako ISP dobija svoj blok adresa?

O: **ICANN**: Internet Corporation for Assigned Names and Numbers

- Dodjeljuje adrese
- Upravlja DNS
- Dodjeljuje imena domena, razrješava sporove
- Dodjeljuje adrese lokalnim regionalnim Internet registrima (ARIN, RIPE, APNIC, LACNIC i AFRINIC)



# NAT: Network Address Translation



*Svi* datagrami *napuštaju* lokalnu mrežu imajući *istu* jedinstvenu izvorišnu adresu NAT IP: 138.76.29.7, Različiti brojevi izvorišnih portova

Datagrami sa izvorima ili destinacijama u ovoj mreži imaju 10.0.0.0/24 adresu za izvor, destinaciju (kao što je uobičajeno)

# NAT: Network Address Translation

- **Motivacija:** lokalna mreža koristi samo jednu IP adresu:
  - Nema potrebe za dodjelu opsega adresa od strane ISP (samo jedna IP adresa se koristi za sve uređaje)
  - Mogu mijenjati adrese uređaja u lokalnim mrežama bez obavještenja "ostatku svijeta"
  - Mogu mijenjati ISP bez mijenjanja adresa uređaja u lokalnim mrežama
  - Uređaji unutar mreže se eksplicitno ne adresiraju, na vidljiv način "ostatku svijeta" (plus u smislu zaštite).

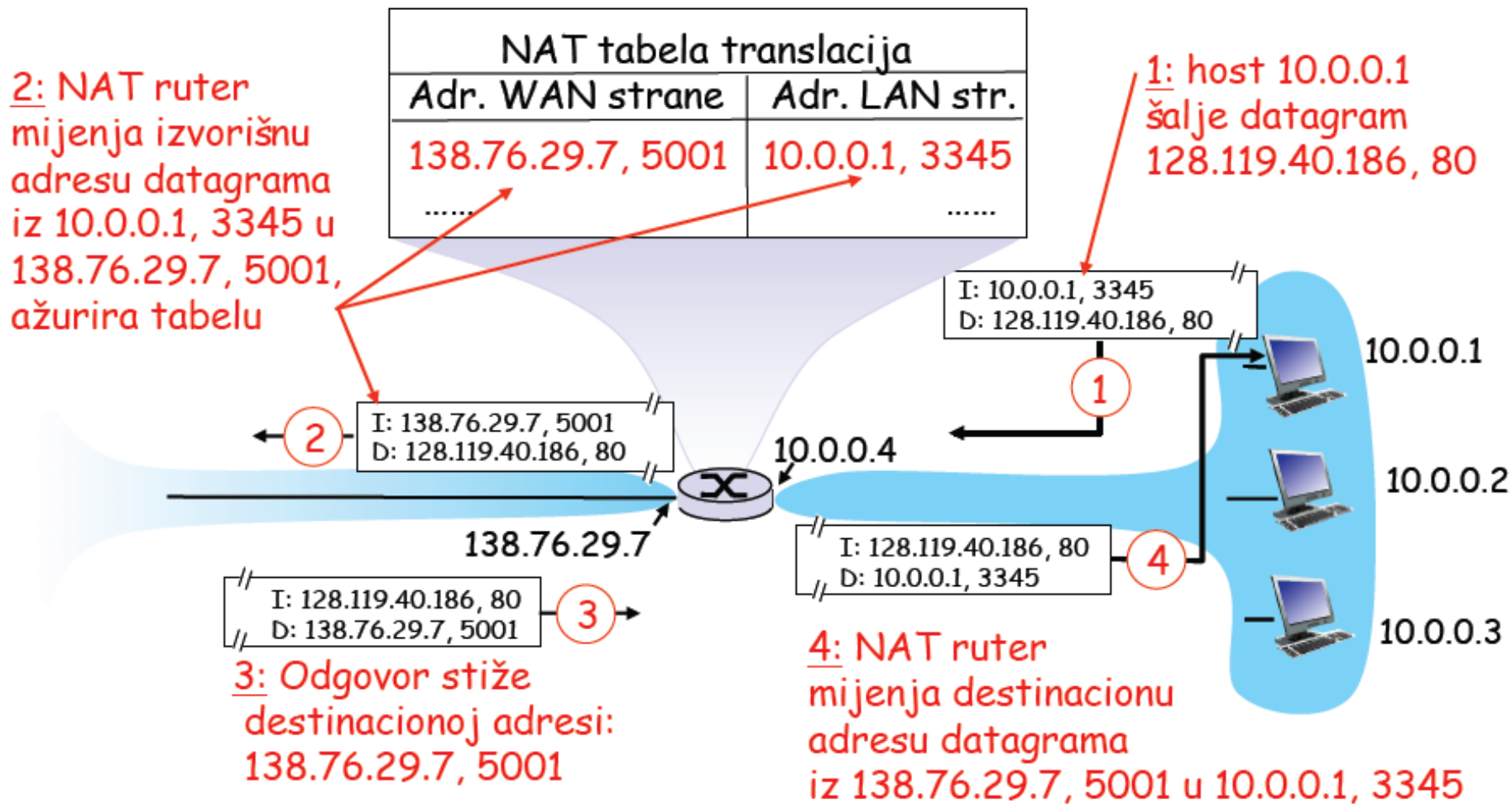
# NAT: Network Address Translation

---

## Implementacija: NAT ruter mora:

- *odlazni datagrami: zamijeniti* (izvorišnu IP adresu, broj port) svakog odlaznog datagrama sa (NAT IP adresom, novim brojem porta)
  - ... udaljeni klijenti/serveri će odgovoriti korišćenjem (NAT IP adrese, novi broj porta) kao adrese destinacije.
- *zapamtiti (u NAT tabeli translacija)* svaki (izvorišna IP adresa, broj porta) i (NAT IP adresa, novi broj porta) u vidu translacionog para
- *dolazeći datagrami: zamijeniti* (NAT IP adresu, novi broj porta) u polju destinacije svakog dolaznog datagrama sa odgovarajućim (izvorišna IP adresa, broj porta) smještenim u NAT tabeli

# NAT: Network Address Translation



# NAT: Network Address Translation

---

- ❑ 16-bitno polje broja porta:
  - 65536 simultanih veza sa jednom adresom sa LAN strane!
- ❑ NAT je kontraverzan:
  - Ruteri bi trebali vršiti obradu samo do nivoa 3
  - Narušava prirodu od kraja do kraja
    - NAT mora biti uzet u obzir od strane dizajnera aplikacija, npr., P2P aplikacija
  - Oskudica adresa se može ublažiti i prije upotrebe IPv6
  - Broj porta se posredno koristi za adresiranje računara

# IPv6

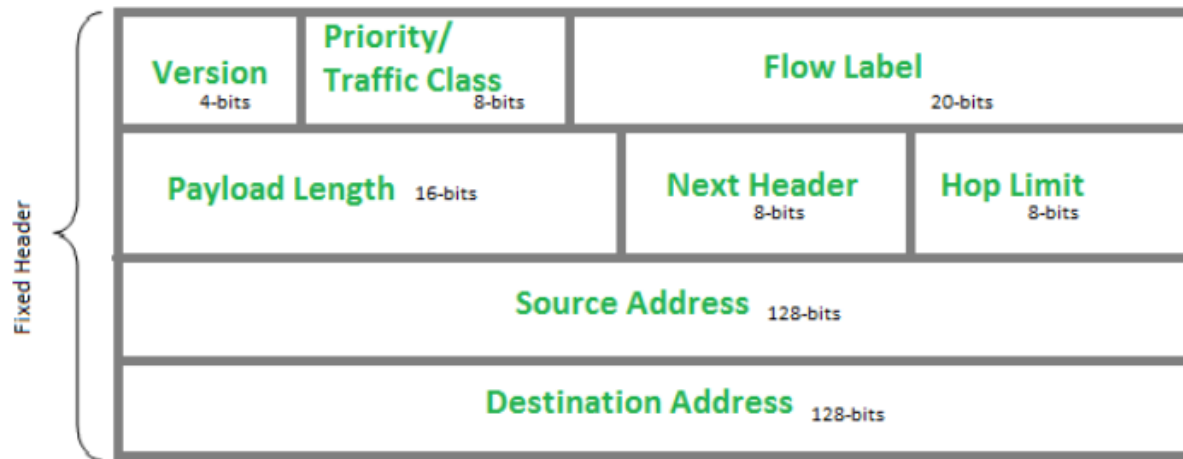
- ❑ **Inicijalna motivacija:** 32-bitni adresni prostor će vrlo brzo u potpunosti biti dodijeljen.
- ❑ **Dodatna motivacija:**
  - Format zaglavlja pomaže obradi/prosleđivanju
  - Promjene zaglavlja uključuju QoS
- ❑ **IPv6 format datagrama:**
  - Zaglavlje fiksne-dužine od 40B
  - Nije dozvoljena fragmentacija

# IPv6 zaglavlje (nastavak)

**Priority:** identifikuje prioritet između datagrama u “toku”

**Traffic class:** identifikuje datagrame u istom “toku”.  
(koncept “toka” nije precizno definisan).

**Next header:** identifikuje protokola višeg nivoa za podatke





# IPv6 zaglavlje (nastavak)

*Priority*: identifikuje prioritet između datagrama u "toku"

Od 0 do 7 prioriteti za *Congestion controlled* saobraćaj, a od 8 do 15 prioriteti za *Uncontrolled* saobraćaj. Mogu ruteru da mijenjaju prioritet

Priority	Meaning
0	No Specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic



# IPv6 zaglavlje (nastavak)

**Flow label** (20-bit): Polje oznake protoka izvor koristi za označavanje paketa koji pripadaju istom toku kako bi se zahtijevalo posebno rukovanje od *intermediate* IPv6 rutera, kao što je *ne-default* kvalitet servisa ili servisa u realnom vremenu.

Da bi razlikovao tok, *intermediate* ruter može koristiti izvornu adresu, adresu odredišta i oznaku toka paketa.

Između izvora i odredišta može postojati više tokova jer se mnogi procesi mogu istovremeno izvoditi. Ruteri ili host koji ne podržavaju funkcionalnost polja oznake toka, kao i za podrazumijevano rutiranje, polje oznake toka je postavljeno na 0.

Tokom postavljanja oznake toka izvor također treba da navede *lifetime* protoka.

# IPv6 zaglavlje (nastavak)

**Payload Length** (16-bit): To je 16-bitno polje koje označava ukupnu veličinu payload-a koje ruterima govori o količini informacija koji određeni paket sadrži u svom payload-u.

**Next Header** (8-bits): Indikator protokola višeg nivoa.

# Druge izmjene u odnosu na IPv4

- ❑ **Checksum:** potpuno uklonjena kako bi se smanjila obrada na svakom hopu
- ❑ **Options:** dozvoljene, ali van zaglavlja, indicirano sa “Next Header” poljem
- ❑ **ICMPv6:** nova verzija ICMP
  - dodatni tipovi poruka, npr. “Packet Too Big”
  - funkcija upravljanja multicast grupama

# IPv6 adresiranje

Format:

- ❑ RFC 4291 (Februar 2006)
- ❑ 128 bita
- ❑ Predstavlja se u vidu 8 grupa po četiri heksadecimalna broja
- ❑ X:X:X:X:X:X:X:X
- ❑ 1111111000011010 0100001010111001 0000000000011011  
0000000000000000 0000000000000000 0001001011010000  
0000000001011011 0000011010110000
- ❑ FE1A:42B9:001B:0000:0000:12D0:005B:06B0
- ❑ FE1A:42B9:001B:0:0:12D0:005B:06B0 (grupa od četiri 0 se može prikazati jednom 0)
- ❑ FE1A:42B9:1B::12D0:5B:6B0 (više susjednih grupa od četiri 0 se prikazuje sa ::, koja se može pojaviti samo jednom)
- ❑ 2001:4C::50:0:0:741
- ❑ 2001:004C::0050:0000:0000:0741
- ❑ 2001:004C:0000:0000:0050:0000:0000:0741

# IPv6 adresiranje

## IPv6 prefiks:

- ❑ Slično kao kod IPv4: IPv6adresa/dužina prefiksa
- ❑ 200C:001b:1100:0:0:0:0:0/40 ili 200C:1b:1100::/40
- ❑ Koristi se CIDR rutiranje

# IPv6 adresiranje

Tri tipa adresa:

- ❑ *unicast* - označava adresu jednog interfejsa na uređaju
- ❑ *multicast* - označava grupu interfejsa (uglavnom na različitim računarima) tako da paket poslat na ovu adresu stiže **do svih** adresiranih interfejsa koji pripadaju istom multicast stablu
- ❑ *anycast* - paket poslat na *anycast* adresu stiže **do jednog od interfejsa** opisanih ovom adresom (po pravilu najbližeg definisano pojmom rastojanja u protokolu rutiranja)

Nema više broadcast adrese. Njenu funkciju preuzima multicast adresa, čime se stvara mogućnost korišćenja adresa koje se sastoje od svih nula i jedinica.

# IPv6 adresiranje

## Dodjela IPv6 adresa:

- Kombinacija alokacije i automatske dodjele.
- Prvih nekoliko bita (Format prefiks) se koriste za alokaciju adresa.

Tip adrese -----	Binarni prefiks -----	IPv6 notacija -----
Unspecified	00...0 (128 bita)	::/128
Loopback	00...1 (128 bita)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	(sve ostalo)	

# IPv6 adresiranje

## *Unspecified Adresa*

- ❑ Adresa sa svim nulama: 0:0:0:0:0:0:0:0.
- ❑ Označava da IPv6 adresa nije definisana za interfejs. Datagrame sa ovom odredišnom adresom ne prosleđuje IPv6 ruter.

## *Loopback Adresa*

- ❑ IPv6 loopback adresa je 0:0:0:0:0:0:0:1.
- ❑ Njeno korišćenje je slično korišćenju IPv4 loopback adrese 127.0.0.1.

## *IPv4 mapirane adrese:*

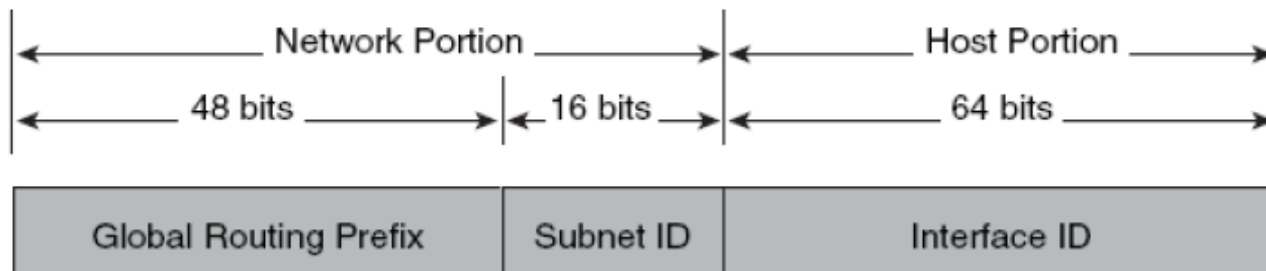
- ❑ Prvih 80 bita su nule
- ❑ Sledećih 16 bita su jedinice
- ❑ Ostalih 32 bita su jednaki bitima odgovarajuće IPv4 adrese
- ❑ 100.1.1.1 = 01100100 00000001 00000001 00000001=6401:0101
- ❑ 0000:0000:0000:0000:0000:FFFF:6401:0101 ili
- ❑ 0:0:0:0:0:FFFF:6401:0101 ili
- ❑ ::FFFF:6401:0101 ili čak ::FFFF:100.1.1.1



# IPv6 adresiranje

## *IPv6 globalna adresa*

- ❑ Koristi se za povezivanje na javnu mrežu.
- ❑ Ove unicast adrese su jedinstvene i na bazi njih ruteri mogu prosleđivati pakete.
- ❑ RFC 2374, RFC 3587
- ❑ Globalni prefiks rutiranja (generalno je dužine 48 bita), identifikator subneta (dužine 16 bita) i identifikator interfejsa (dužine 64 bita)



# IPv6 adresiranje

## IPv6 *link-local* adresa

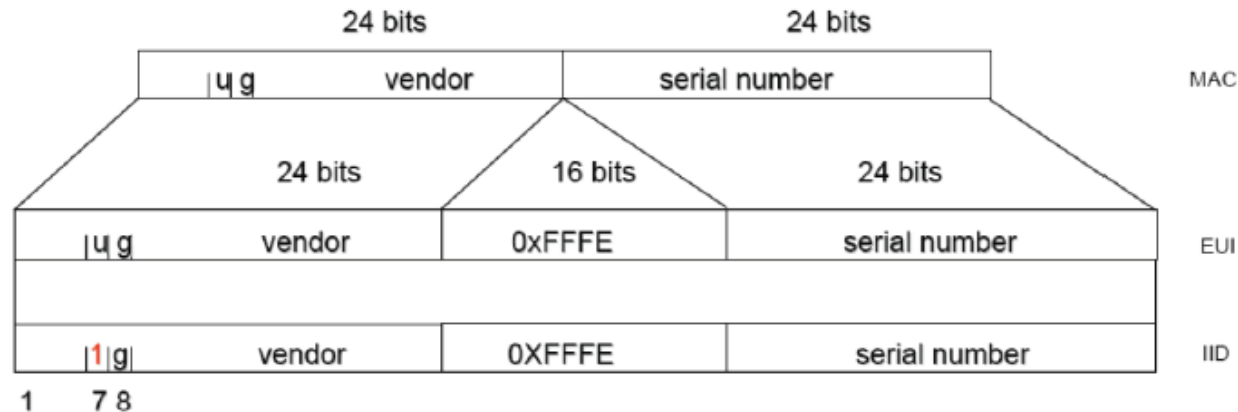
- ❑ Koriste se za adresiranje na jednom linku (mrežni segment bez rutera, npr. LAN).
- ❑ Značajne samo za čvorišta u okviru jedne LAN mreže.
- ❑ Ruteri ne prosleđuju pakete sa ovim izvorišnim ili odredišnim adresama van LAN-a.
- ❑ Koriste se za automatsko dodjeljivanje adresa, otkrivanje susjeda ili kada nema rutera u mreži.
- ❑ Ove adrese su identifikovane sa FE8 heksadecimalnim brojevima (10 bita) na početku.
- ❑ Konfiguriraju se automatski ili manuelno.
- ❑ 1111111010 + 54 nule i 64-bitni identifikator interfejsa.
- ❑ Identifikator interfejsa se dobija automatski, komunikacijom sa drugim čvorištem na linku.

10 bits	54 bits	64 bits
1111111010	0	Interface Identifier

# IPv6 adresiranje

## Identifikator interfejsa

- U modifikovanom EUI-64 formatu
- Jedinствен unutar jedne podmreže



- $u=1$ , adresa se formira na bazi MAC adrese (global scope)
- $u=0$ , adresa se formira na slučajan način (local scope)

# IPv6 adresiranje

## IPv6 *multicast* adresa

- Ista funkcija kao IPv4 multicast adresa

11111111	<b>Flag</b>	<b>Scope</b>	<b>Group ID</b>
8 bits	4 bits	4 bits	112 bits

FF00::/8 addresses are multicast addresses

**Flag bits:** 0 R P T

**T = 0** permanent addresses (managed by IANA)

**T = 1** transient multicast addresses

- **P = 1** derived from unicast prefix (RFC3306)

- **R = 1** embedded RP addresses (I-D)

### **Scope**

**0** : Reserved

**1** : Interface-local

**2** : Link-local

**3** : Subnet-local

**4** : Admin-local

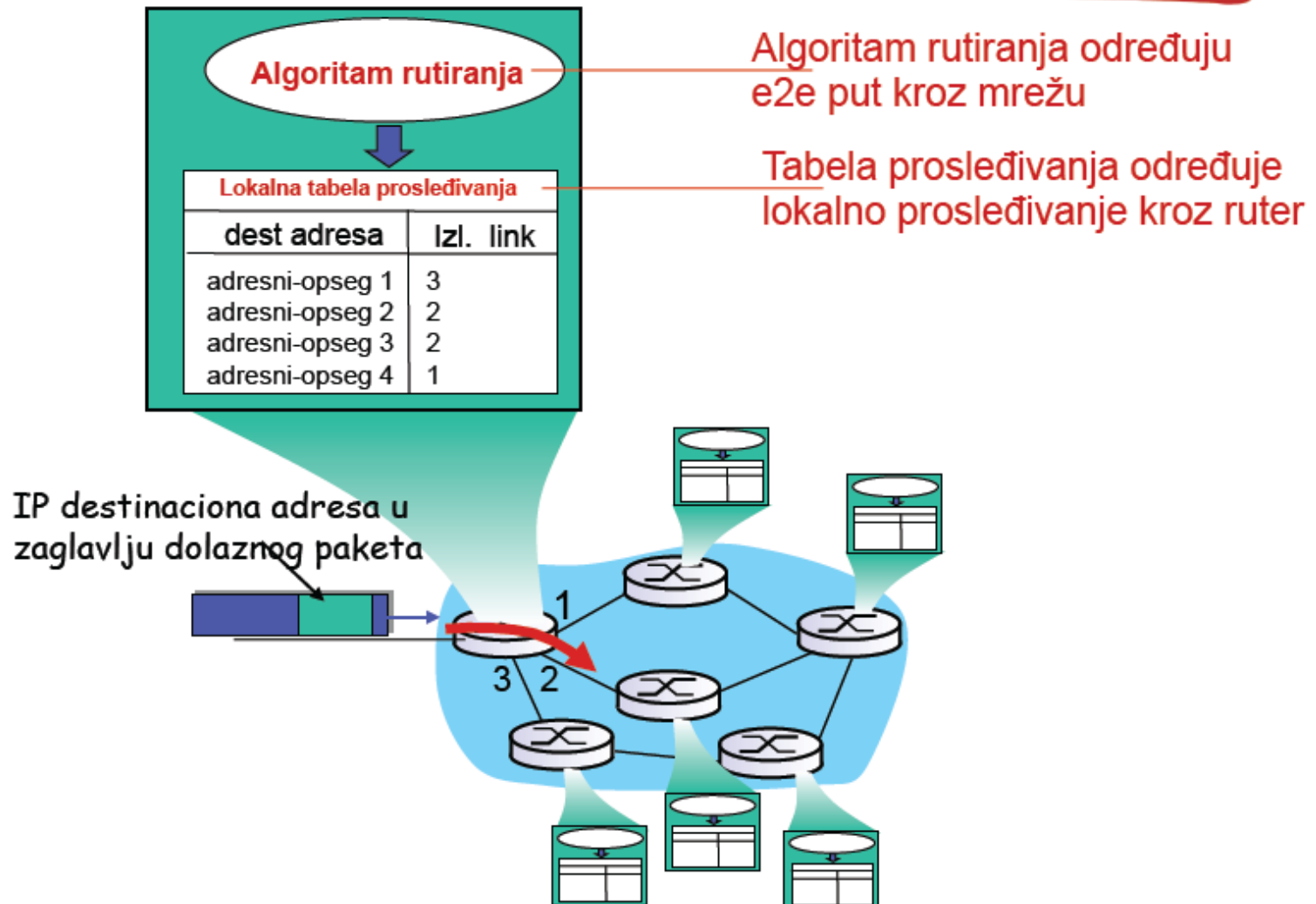
**5** : Site-local

**8** : Organization-local

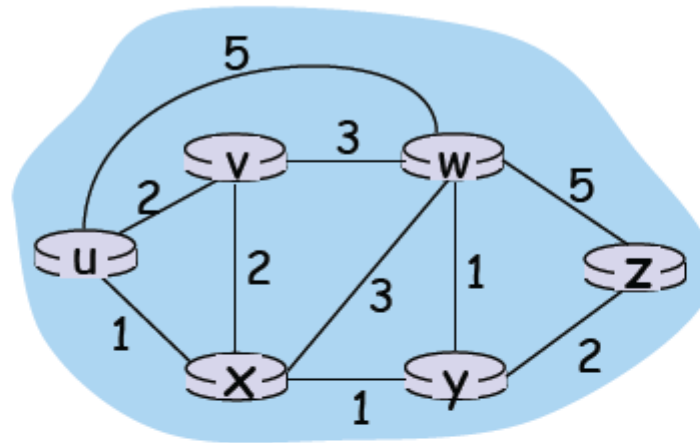
**E** : Global

**F** : Reserved

# Međuveza prosleđivanja i rutiranja



# Apstrakcija pomoću grafa



Graf:  $G = (N,E)$

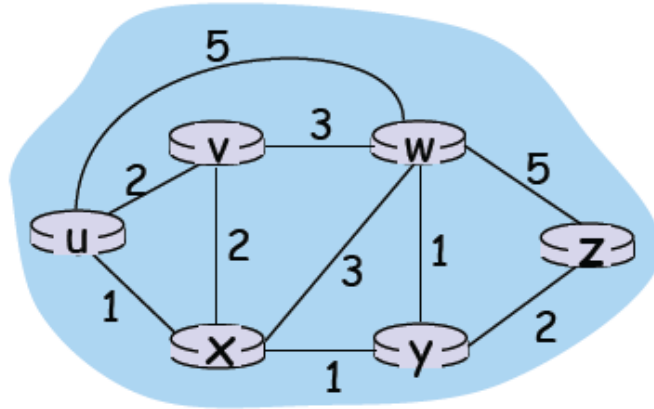
$N = \text{skup rutera} = \{ u, v, w, x, y, z \}$

$E = \text{skup linkova} = \{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Napomena: Abstrakcija pomoću grafa je korisna i u drugim mrežnim kontekstima.

Primjer: P2P, gdje je  $N$  skup peer-ova, a  $E$  skup TCP konekcija

# Apstrakcija pomoću grafa: troškovi



- $c(x,x')$  = težinski faktor (cost) linka  $(x,x')$ 
  - npr.,  $c(w,z) = 5$
- težinski faktor može biti uvijek 1, ili recipročan protoku, ili recipročan zagušenju

Težinski faktor puta  $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

Pitanje: Koji je put sa najmanjim težinskim faktorom između u i z ?

Algoritam rutiranja: algoritam koji pronalazi put sa najmanjim težinskim faktorom

# Klasifikacija algoritama rutiranja

---

## Globalna ili decentralizovana informacija?

### Globalna:

- ❑ svi ruteri posjeduju kompletnu topologiju mreže sa informacijama o težinskim faktorima linkova
- ❑ “link state” algoritmi

### Decentralizovani:

- ❑ ruter poznaje fizički povezane susjede i težinske faktore linkova do susjeda
- ❑ iterativni proces izračunavanja, razmjena informacija sa susjedima
- ❑ “distance vector” algoritmi

## Statički ili dinamički?

### Statički:

- ❑ Rute se sporo mijenjaju

### Dinamički:

- ❑ Rute se mijenjaju mnogo brže
  - periodični update
  - kao odgovor na promjene težinskih faktora linkova



# "Link-State" Algoritam Rutiranja

## Dijkstra algoritam

- Mrežna topologija, težinski faktori linkova poznati svim čvorištima
  - Dobijeno preko "link state broadcast"
  - Sva čvorišta imaju istu informaciju
- Proračunava puteve najmanjih težinskih faktora od jednog čvorišta ("izvor") do svih ostalih čvorova
  - generiše tabelu rutiranja za to čvorište
- iterativni: poslije k iteracija, poznat je put sa najmanjim težinskim faktorom do k destinacija

## Notacija:

- $c(A,B)$ : težinski faktor linka od čvorišta  $x$  do  $y$  su beskonačni ukoliko čvorišta nijesu susjedi
- $D(A)$ : trenutna vrijednost težinskog faktora puta od izvorišta do destinacije  $A$
- $p(A)$ : sledeće čvorište duž puta od izvorišta do čvorišta  $A$ , koje je susjed  $A$
- $N'$ : skup čvorišta čiji su najniži težinski faktori puta poznati

# Dijsktra Algoritam (na čvoru u)

## 1 **Inicijalizacija:**

2  $N' = \{u\}$

3 Za sva čvorišta A

4 Ako je A susjedno čvorište u

5 tada  $D(A) = c(u,A)$

6 else  $D(A) = \infty$

7

## 8 **Petlja**

9 Pronaći B koje nije u  $N'$  tako da je  $D(B)$  minimalno

10 dodati B skupu  $N'$

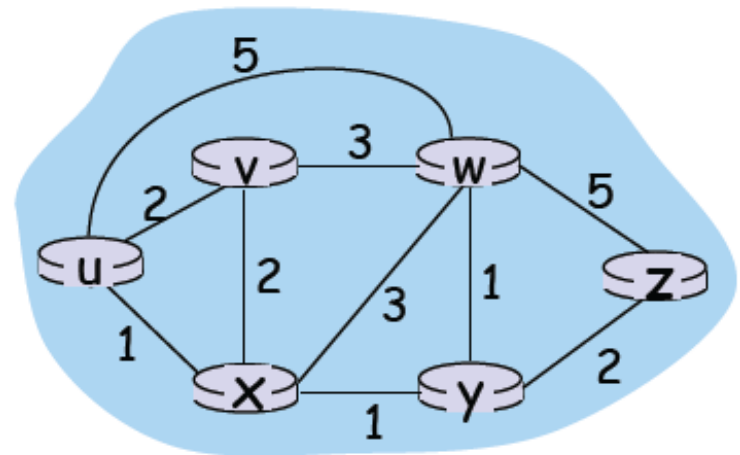
11 update  $D(A)$  za sve A susjede B koji nijesu u  $N'$  :

12  $D(A) = \min( D(A), D(B) + c(B,A) )$

13 /\* novi težinski faktori za A su ili stari težinski faktori za A ili su poznati

14 najmanji težinski faktori puta do B plus težinski faktori od B do A \*/

15 **dok sva čvorišta ne budu u  $N'$**

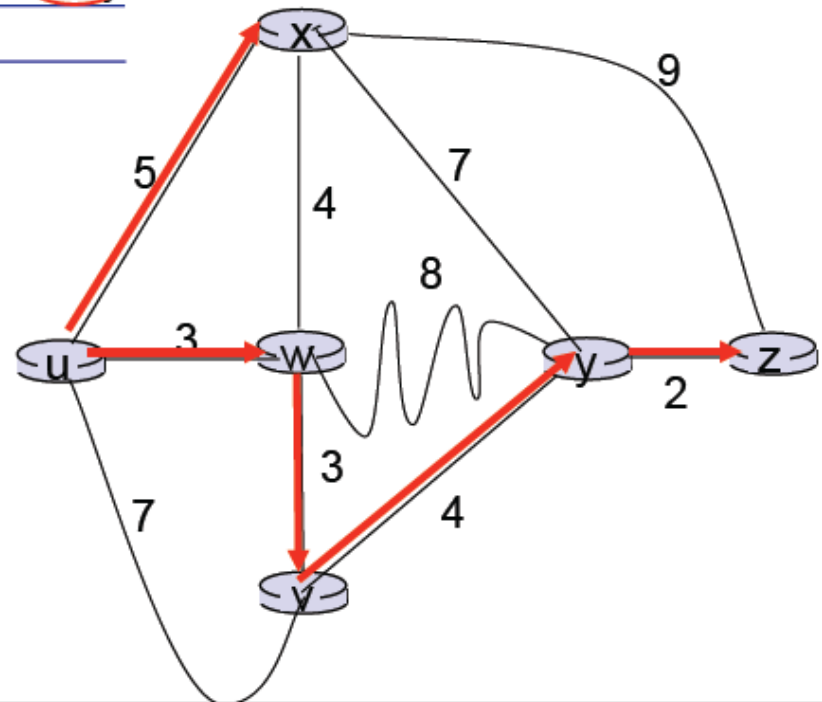


# Dijkstra algoritam: primjer

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	$\infty$	$\infty$
1	uw	6,w		5,u	11,w	$\infty$
2	uwx	6,w			11,w	14,x
3	uwxv				10,y	14,x
4	uwxvy					12,y
5	uwxvyz					

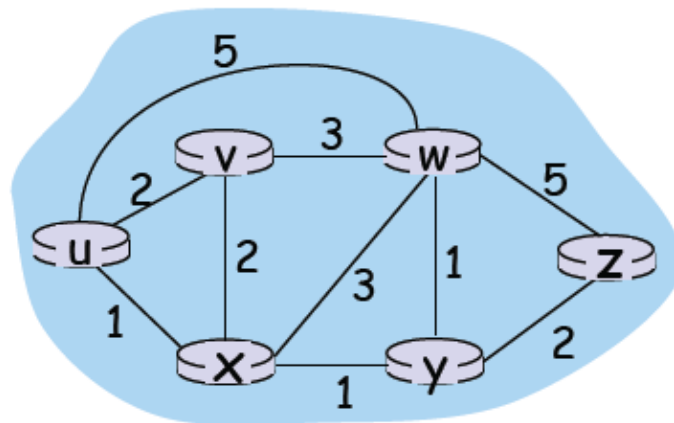
## Napomene:

- ❖ Konstruisati najkraći put praćenjem prethodnih čvorišta
- ❖ Linkovi mogu biti prekinuti



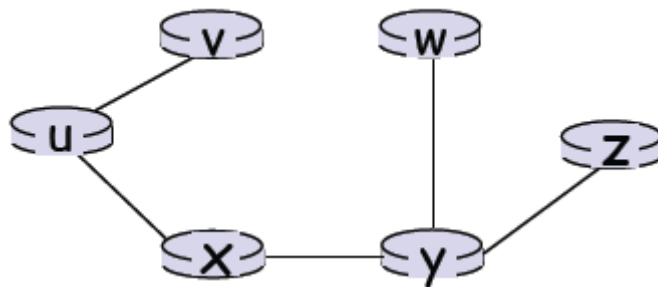
# Dijkstra algoritam: primjer čvorište U

Korak	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	$\infty$	$\infty$
1	ux	2,u	4,x	<del>1,u</del>	2,x	$\infty$
2	uxy	2,u	3,y	<del>4,x</del>	<del>2,x</del>	4,y
3	uxyv	<del>2,u</del>	3,y	<del>3,y</del>	<del>4,y</del>	<del>4,y</del>
4	uxyvw	<del>2,u</del>	<del>3,y</del>	<del>3,y</del>	<del>4,y</del>	4,y
5	uxyvwz	<del>2,u</del>	<del>3,y</del>	<del>3,y</del>	<del>4,y</del>	<del>4,y</del>



# Dijkstra algoritam: primjer (2)

Rezultujuća shortest-path topologija iz čvorišta u:



Rezultujuća tabela prosleđivanja u čvorištu u:

destinacija	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

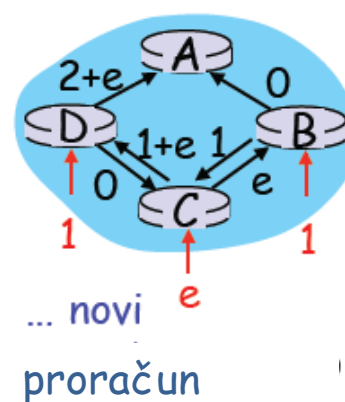
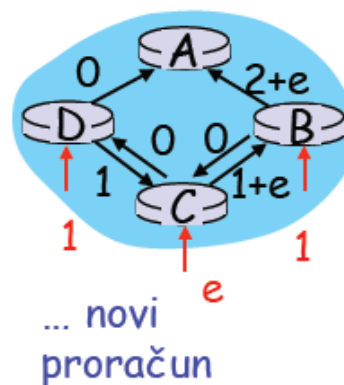
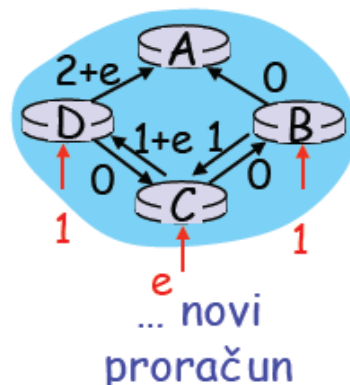
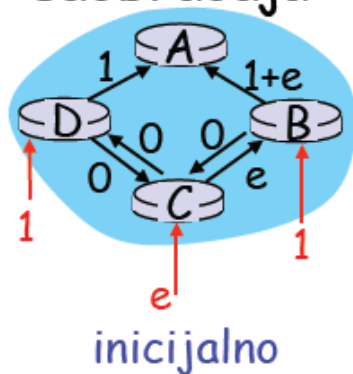
# Dijkstra's algoritam, diskusija

**Kompleksnost algoritma:**  $n$  čvorišta

- Svaka iteracija: potrebno da provjeri sva čvorišta,  $B$ , koja nijesu u  $N'$
- $n*(n-1)/2$  komparacija:  $O(n^2)$
- Moguće su efikasnije implementacije:  $O(n \log n)$

**Moguće su oscilacije:**

- npr., težinski faktor linka = količina prenešenog saobraćaja



# Distance Vector Algoritam (1)

## Bellman-Ford jednačina (dinamičko programiranje)

Definišimo

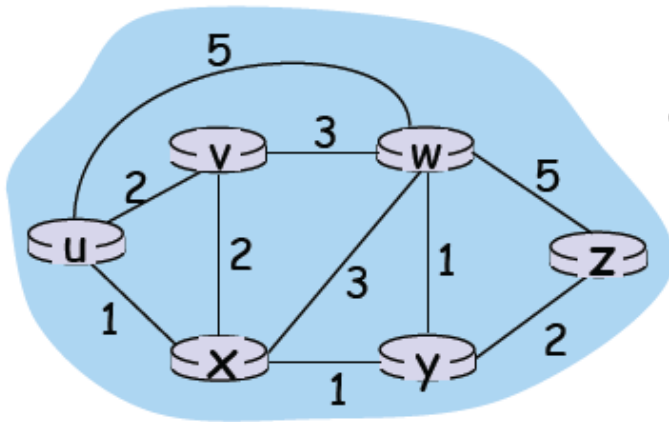
$d_x(y)$  := težinski faktor puta sa najmanjim troškovima od  $x$  do  $y$

Tada

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

Gdje se  $\min_v$  uzima u odnosu na sve susjede  $x$

# Bellman-Ford primjer (2)



Jasno,  $d_v(z) = 5$ ,  $d_x(z) = 3$ ,  $d_w(z) = 3$

B-F jednačina kaže:

$$\begin{aligned} d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\quad c(u,x) + d_x(z), \\ &\quad c(u,w) + d_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

Čvorište koje dostigne minimum je sledeći korak (hop) u najkraćem putu → tabela prosleđivanja



# Distance Vector Algoritam (3)

- $D_x(y)$  = estimira najmanji težinski faktor od  $x$  do  $y$
- *Distance vector*:  $D_x = [D_x(y): y \in N]$
- Čvorište  $x$  poznaje težinske faktore do svakog svog susjeda  $v$ :  $c(x,v)$
- Čvorište  $x$  nadzire  $D_x = [D_x(y): y \in N]$
- Čvorište  $x$  takođe nadzire *distance vector*-e svojih susjeda
  - Za svakog susjeda  $v$ ,  $x$  nadzire  $D_v = [D_v(y): y \in N]$

# Distance vector algoritam (4)

---

## Osnovna ideja:

- Svako čvorište periodično šalje estimaciju svog distance vector-a svojim susjedima
- Kada čvorište  $x$  primi novu DV estimaciju od svog susjeda  $v$ , update-je svoj sopstveni DV korišćenjem B-F jednačine:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \quad \text{za svako čvorište } y \in N$$

- U većem broju slučajeva, pod normalnim okolnostima, estimat  $D_x(y)$  konvergira stvarnom najmanjem težinskom faktoru  $d_x(y)$

# Distance Vector Algoritam (5)

---

## Iterativni, asinhron:

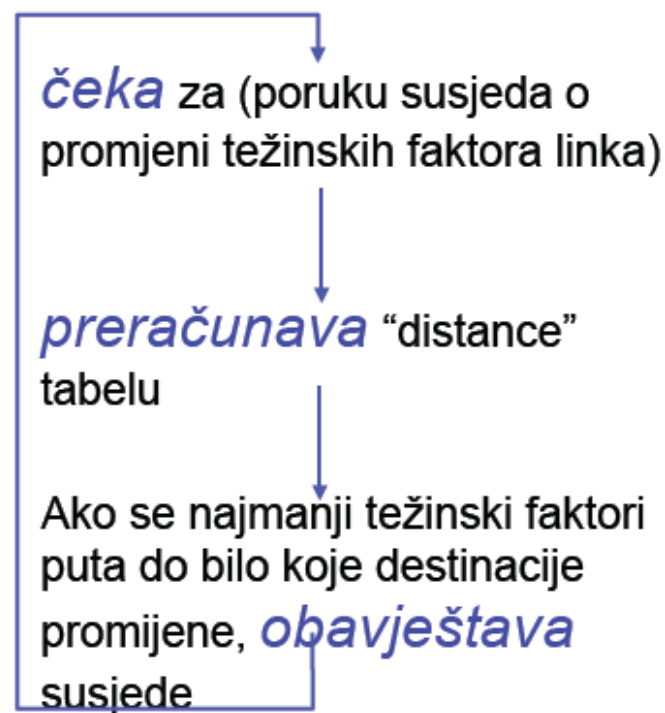
svaka lokalna iteracija je uzrokovana:

- lokalni težinskim faktorima linka koji su promjenljivi
- porukama od susjeda: najmanji težinski faktori puta su promijenjeni

## Distribuiran:

- svako čvorište obavještava susjeda *samo* kada se njegov put sa najmanjim težinskim faktorom promijeni
  - susjedi informišu susjede ako je to potrebno

## Svako čvorište:



$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

**Tabela čvorišta x**

		Tež. fakt. do		
		x	y	z
od	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		Tež. fakt. do		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	7	1	0

		Tež. fakt. do		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0

**Tabela čvorišta y**

		Tež. fakt. do		
		x	y	z
od	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

		Tež. fakt. do		
		x	y	z
od	x	0	2	7
	y	2	0	1
	z	7	1	0

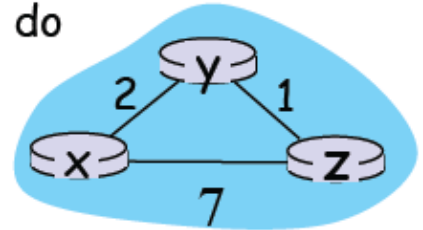
		Tež. fakt. do		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0

**Tabela čvorišta z**

		Tež. fakt. do		
		x	y	z
od	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		Tež. fakt. do		
		x	y	z
od	x	0	2	7
	y	2	0	1
	z	3	1	0

		Tež. fakt. do		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0



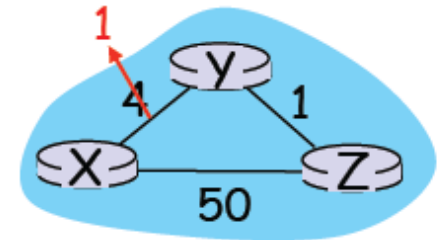
→ vrijeme

## "Distance Vector": promjene težinskih faktora linka

---

### Promjene težinskih faktora linka:

- ❑ Čvorište detektuje lokalne promjene težinskih faktora linka
- ❑ ažuriranje "distance" tabele
- ❑ Ako se težinski faktori promijene u putu sa najmanjim težinskim faktorom, obavijesti susjede



U trenutku  $t_0$ , y detektuje promjenu težinskog faktora linka, ažurira njegov DV, i informiše susjede.

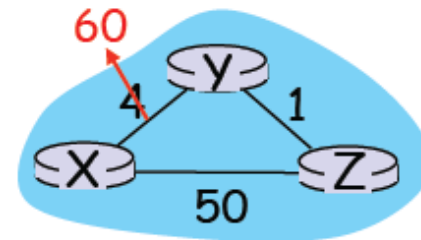
**"dobre vijesti brzo putuju"** U trenutku  $t_1$ , z prima update od y i update-je svoju tabelu. Izračunava novi najmanji težinski faktor do x i šalje svojim susjedima svoj DV.

U trenutku  $t_2$ , y prima z-ove update i ažurira svoju distance tabelu. Y-ov najmanji težinski faktor se ne mijenja i stoga y ne šalje nikakvu poruku ruteru z.

# “Distance Vector”: promjene težinskih faktora linka

## Promjene troškova linka:

- ❑ dobre vijesti se brže prenose
- ❑ loše vijesti se sporije prenose - problem “brojanje do  $\infty$ ”!
- ❑ 44 iteracije do stabilizacije algoritma



## Tehnika split-horizon

- ❑ Update ruta šalje težinske faktore ruta koje se mogu doseći preko drugih portova.

## Poisoned reverse (lažno rastojanje):

- ❑ Ako Z rutira preko Y do X :
  - Z govori Y da je njegova (Z-ova) udaljenost do X beskonačna (tako da Y ne bi rutirala do X preko Z)
- ❑ Da li će to riješiti problem brojanja do beskonačnosti?



# Poređenje LS i DV algoritama

---

## Kompleksnost poruke

- LS: sa  $n$  čvorišta,  $E$  linkova,  $O(nE)$  poruka šalje svaki čvor
- DV: razmjena samo između susjeda
  - Konvergencija varira u vremenu

## Brzina konvergencija

- LS:  $O(n^2)$  algoritam zahtijeva  $O(nE)$  poruka
  - Mogu imati oscilacije
- DV: konvergencija varira u vremenu
  - Može biti petlji
  - Problem brojanja do  $\infty$

## Robustnost: šta se dešava kada ruter otkaže?

### LS:

- Čvorište može objaviti težinski faktor neispravnog *linka*
- Svako čvorište proračunava svoju sopstvenu tabelu

### DV:

- DV čvorište može objaviti težinski faktor neispravnog *linka*
- Tabelu svakog čvorišta koriste drugi
  - Greška se prenosi kroz mrežu

# Hijerarhijsko rutiranje

---

Prethodna analiza je bila - idealizacija

- ❑ Svi ruteri su identični
- ❑ “flat” mreža

*... praksa je drugačija*

**veličina:** nekoliko stotina miliona destinacija:

- ❑ ne mogu se sve destinacije smjestiti u tabele rutiranja!
- ❑ razmjena tabela rutiranja može oboriti linkove!
- ❑ LS može izazvati potiskivanje saobraćaja na račun broadcasta tabela
- ❑ DV teško može konvergirati

**administrativna autonomija**

- ❑ internet = mreža svih mreža
- ❑ svaki mrežni administrator želi
  - kontrolu rutiranja u svojoj sopstvenoj mreži
  - Sakriti mrežnu organizaciju od ostalih



# Hijerarhijsko rutiranje

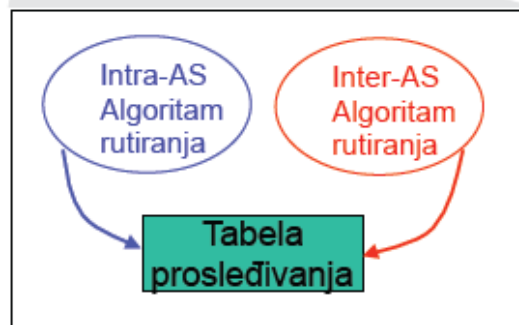
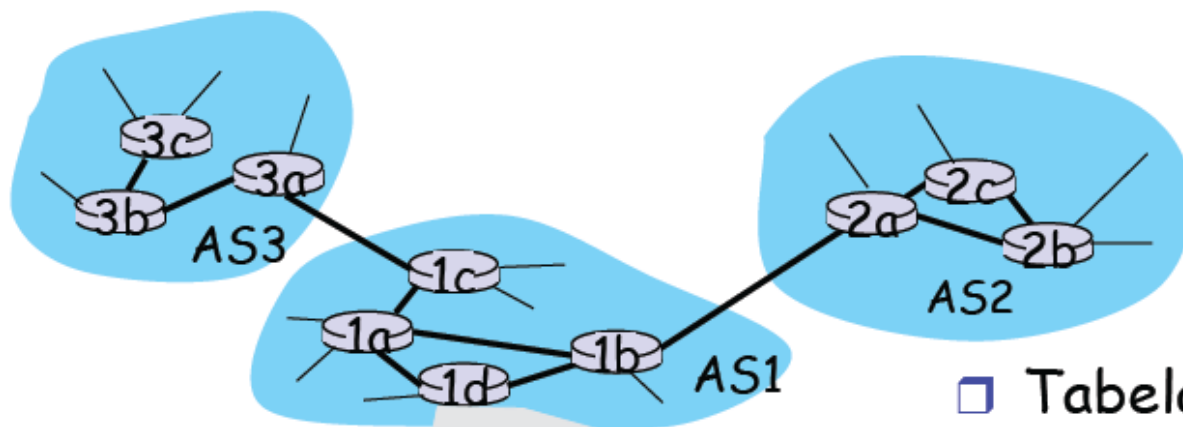
---

- grupiše rutere u regione, “autonomni sistemi” (AS)
- ruteri u istom AS izvršavaju isti protokol rutiranja
  - “intra-AS” protokol rutiranja se slično ponaša objašnjenim idealizovanim modelima
  - ruteri u različitim AS mogu izvršavati različite intra-AS protokole rutiranja

## Gateway ruter

- Prosleđuje datagrame van AS

# Međupovezivanje AS-ma



- Tabela prosleđivanja se konfigurira i sa intra- i sa-AS algoritmom rutiranja
  - Intra-AS setuje sadržaje za interne destinacije
  - Inter-AS & Intra-AS setuje sadržaje za eksterne destinacije

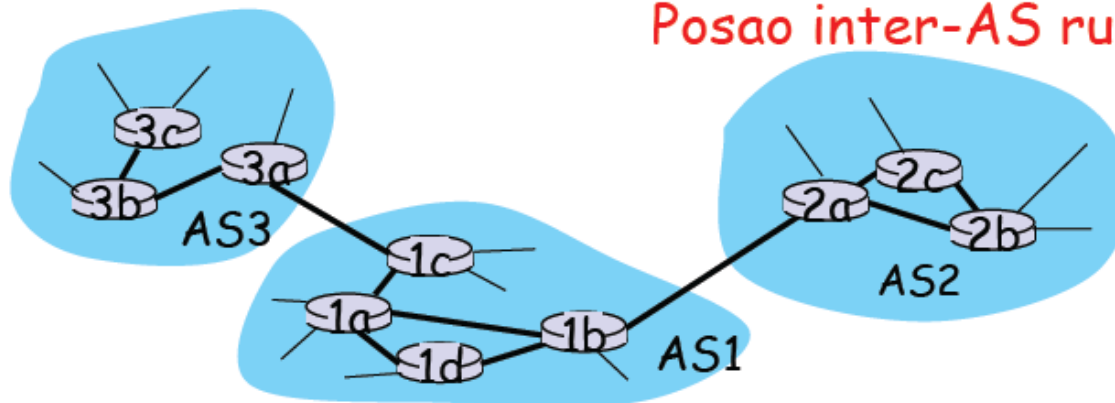
# Inter-AS zadaci

- Pretpostavimo da ruter u AS1 primi datagram za koji je destinacija van AS1
  - Ruter bi trebao proslijediti paket prema gateway ruteru ali kojem?

## AS1 treba:

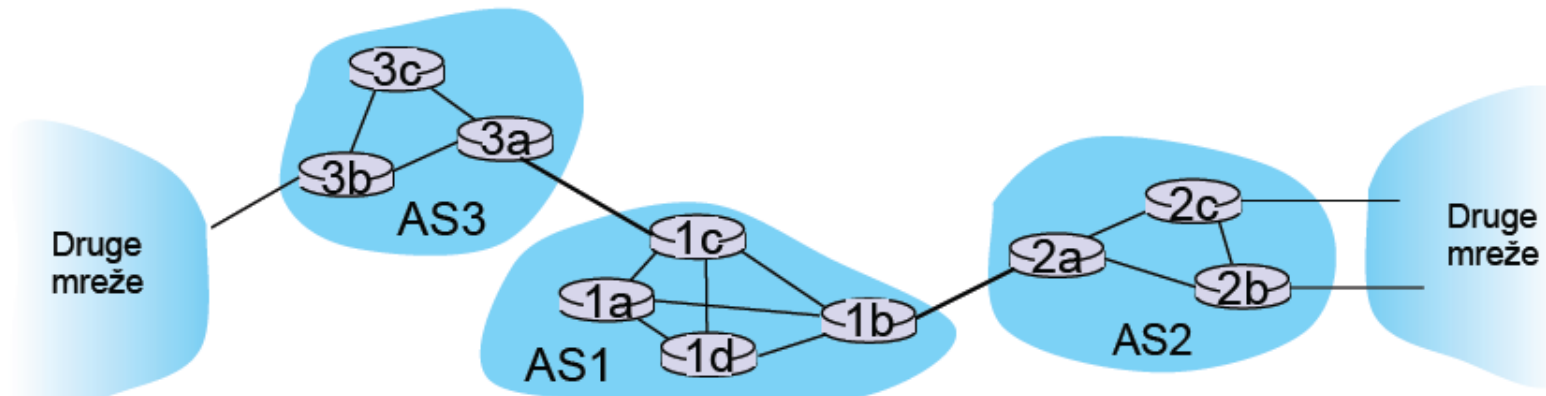
1. Da nauči koje su destinacije dostižne preko AS2, a koje preko AS3
2. Da proslijedi tu informaciju o mogućnosti doseganja do svih rutera u AS1

## Posao inter-AS rutiranja!



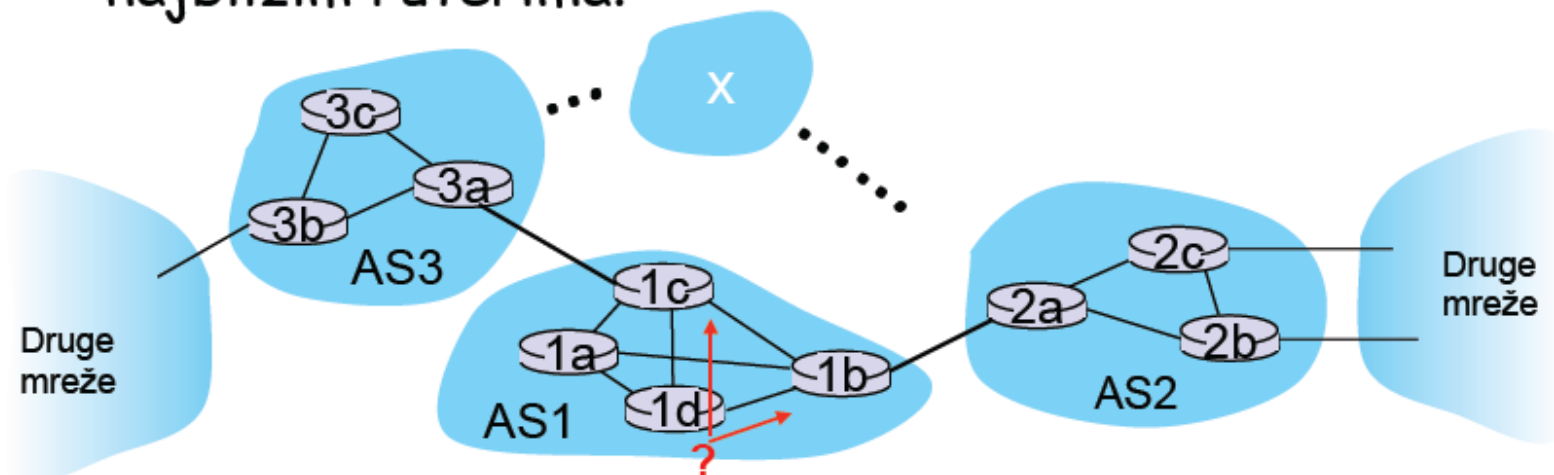
## Primjer: Setovanje tabele prosleđivanja u ruteru 1d

- ❑ Pretpostaviti da AS1 uči od inter-AS protokola da je pod mreža **x** dostižna preko AS3 (gateway 1c) ali ne i preko AS2.
- ❑ Inter-AS protokol šalje informaciju o mogućnosti dostizanja do svih internih rutera.
- ❑ Ruter 1d odlučuje na bazi intra-AS informacije rutiranja da je njegov interfejs **I** na putu sa najmanjim troškovima do 1c.
  - Upisuje u tabelu prosleđivanja sledeći sadržaj (**x,I**).



# Primjer: Izbor između više AS-ma

- ❑ Pretpostavimo da AS1 uči od inter-AS protokola da se podmreža **x** može doseći i od AS3 i od AS2.
- ❑ Da bi konfigurisao tabelu prosleđivanja, ruter 1d mora odlučiti prema kojem gateway bi trebala prosljediti pakete za destinaciju **x**.
- ❑ Ovo je takođe posao inter-AS protokola rutiranja!
- ❑ **Rutiranje "vrući krompir"**: slanje paketa prema najbližim ruterima.



# Intra-AS Rutiranje

- ❑ Poznato kao **Interior Gateway Protocols (IGP)**
- ❑ Najpoznatiji Intra-AS protokoli rutiranja:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol  
(vlasništvo kompanije Cisco)
  - IS-IS: Intermediate system to intermediate system

# OSPF (Open Shortest Path First)

---

- ❑ Interior Gateway Protocol (IGP)
- ❑ “open”: javno dostupan
- ❑ Verzija 2 (RFC 2328) iz 1998
- ❑ Verzija 3 (RFC2740) iz 1999 podržava IPv4 i IPv6
- ❑ Koristi se u velikim kompanijskim mrežama zbog brze konvergencije, rješavanja problema petlji i balansiranja saobraćaja, dok operatori koriste IS-IS koji je pogodan za stabilne mreže
- ❑ Koristi “Link State” algoritam
  - LS širenje paketa
  - Mapa topologije na svakom čvorištu
  - Proračun rute korišćenjem Dijkstra algoritma
  - Broadcast svakih 30min
- ❑ OSPF oglašavanja nose po jednu informaciju po susjednom ruteru
- ❑ Širenje oglašavanja preko **čitavog** AS (“flooding”)
  - Nose se u OSPF porukama direktno preko IP ( a ne preko TCP ili UDP) pri čemu potrebne kontrole obavlja OSPF
- ❑ Radi smanjenja saobraćaja može se koristiti koncept DR (designated router) i multicasta tabele.



## OSPF “advanced” karakteristike (ne u RIP)

---

- ❑ **Sigurnost:** za sve OSPF poruke se mora znati izvor (prevencija malicioznih aktivnosti) pri čemu se koriste lozinke ili MD5 kodiranje
- ❑ **Više** puteva sa istim troškovima je dozvoljeno (samo jedan put u RIP)
- ❑ Za svaki link, više metrika troškova za različiti **TOS** (npr., troškovi satelitskog linka su podešeni na “nisko” za best effort; visoko za servis u realnom vremenu)
- ❑ Integrisana uni- i **multicast** podrška:
  - Multicast OSPF (MOSPF) koristi istu bazu podataka o topologiji kao OSPF
- ❑ **Hijerarhijski** OSPF u velikim domenima.



# IS-IS (Intermediate system to intermediate system)

---

- ❑ Interior Gateway Protocol (IGP)
- ❑ Koristi "Link State" algoritam
  - LS širenje paketa
  - Mapa topologije na svakom čvorištu
  - Proračun rute korišćenjem Dijkstra algoritma
- ❑ OSI referentni model (protokol mrežnog nivoa)
- ❑ Poslednjih nekoliko godina je potisnuo OSPF iz operatorskih mreža
- ❑ Multicast prenos LSA
- ❑ CIDR adresiranje
- ❑ Ne koristi usluge IP tako da je samim tim indiferentan u odnosu na verzije IP protokola
- ❑ Zbog jednostavnosti generiše manji saobraćaj od OSPF tako da je pogodan za velike mreže
- ❑ Integrated IS-IS je predložen u TCP/IP arhitekturi
- ❑ IS-IS ruter pripada samo jednoj oblasti (Level 1, Level 2 i Level1-2)
- ❑ Nema okosnice

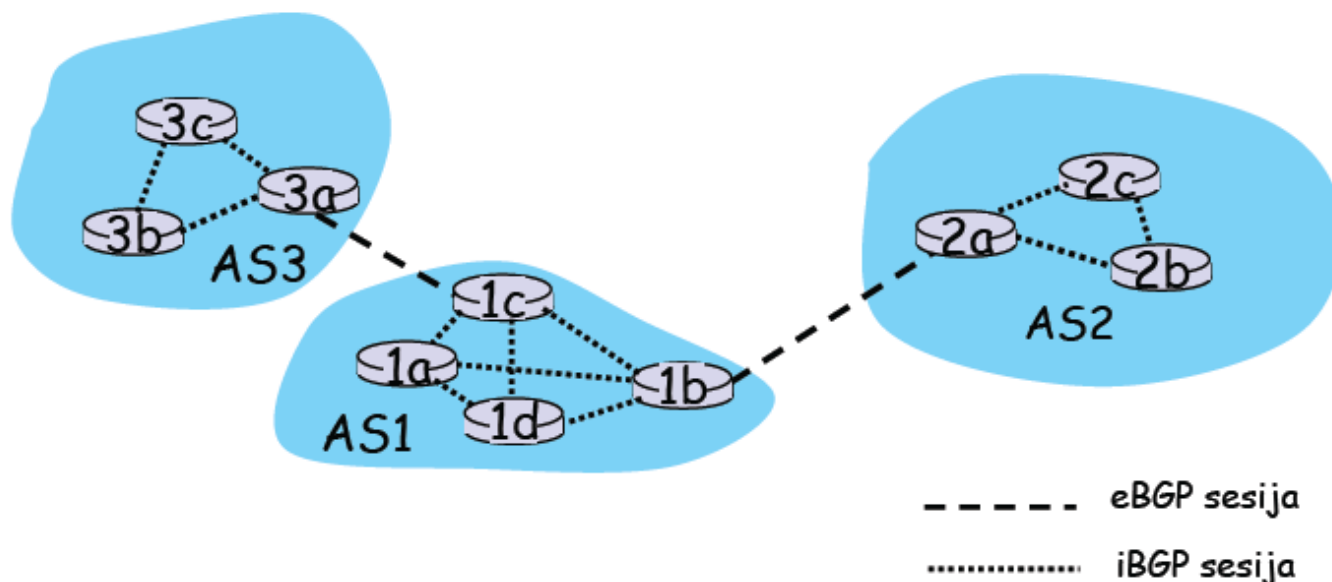
# Internet inter-AS rutiranje: BGP

---

- ❑ **BGP (Border Gateway Protocol):** *de facto* standard
- ❑ Verzija 4 (RFC1771) iz 1994 je doživjela preko 20 korekcija, pri čemu je zadnja RFC4271 (iz 2006)
- ❑ CIDR i agregacija ruta
- ❑ Naslijedio EGP čime je napravljena potpuna decentralizacija Interneta
- ❑ Mogu ga koristiti i kompanije kada OSPF nije dovoljno dobar i kada se radi o multihomed mreži (bolja redundansa).
- ❑ BGP omogućava svakom AS:
  1. Dobijanje informacije o dostižnosti sa susjednih AS-ova.
  2. Prosleđivanje prethodne informacije svim ruterima u okviru AS.
  3. Utvrđivanje “dobre” rute do podmreža baziranih na informaciji o dostižnosti i politici.
- ❑ Dozvoljava podmreži oglašavanje svog prisustva ostatku Interneta: *“Ovdje sam”*

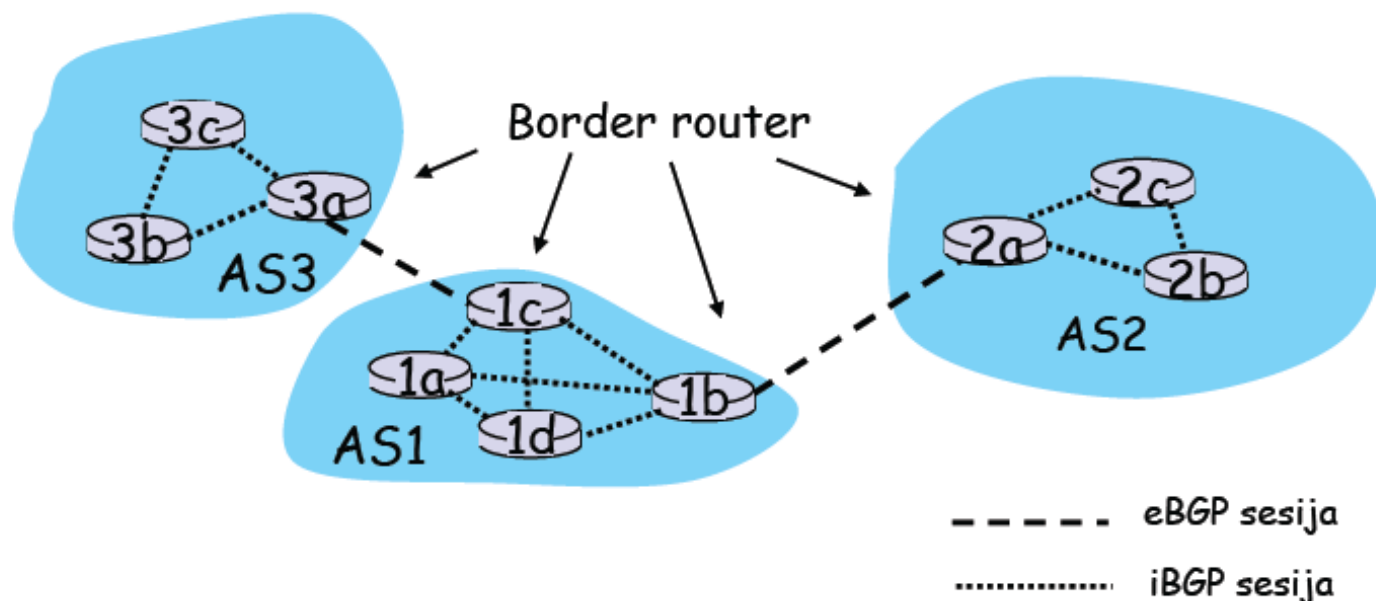
# BGP osnove

- Parovi rutera (BGP peer-ovi) razmjenjuju informaciju rutiranja preko semi-permanentne TCP konekcije (port 179): **BGP sesije**
- Svakih 60s šalje **keep alive** poruku
- **Napomena:** BGP sesije ne odgovaraju fizičkim linkovima.
- Kada AS2 oglasi prefiks do AS1, AS2 **obećava** da će proslijediti bilo koji datagram koji je adresiran do tog prefiksa preko sebe.
  - AS2 može agregirati prefikse u oglašavanjima

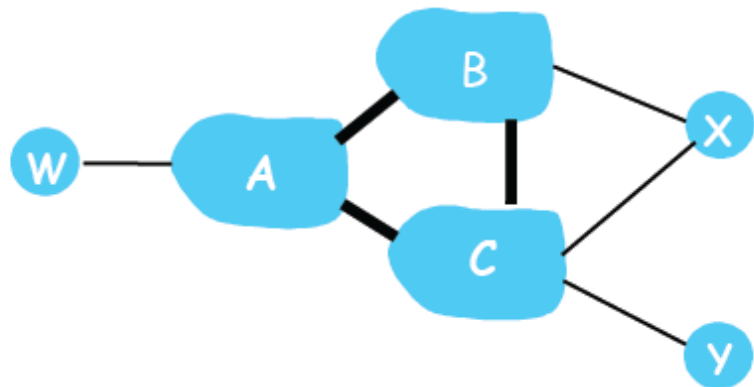


# Distribuirana informacija o dostižnosti

- Sa eBGP sesijom između 3a i 1c, AS3 šalje informaciju o dostižnosti prefiksa do AS1.
- 1c može tada koristiti iBGP za distribuciju ove nove informacije o dostizanju prefiksa do svih rutera u AS1
- 1b može tada ponovo oglasiti novu informaciju o dostizanju do AS2 preko 1b-2a eBGP sesije
- Kada ruter stekne znanje o novom prefiksu, kreira sadržaj za taj prefiks u tabeli rutiranja.

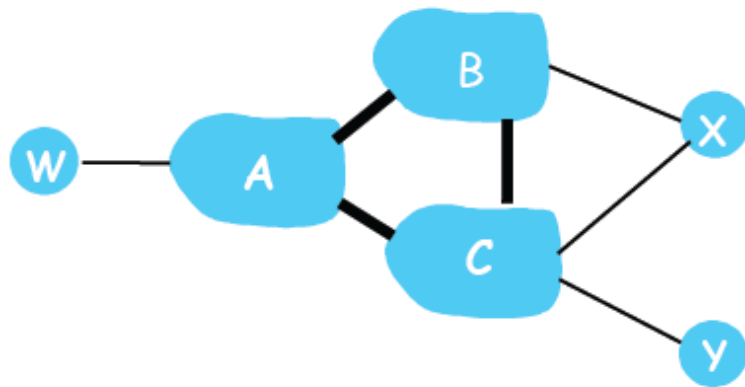


# BGP politika rutiranja



- ❑ A, B, C su mreže provajdera
- ❑ x, w, y su korisnici (mreža provajdera)
- ❑ x je "dual-homed": povezan na dvije mreže
  - x ne želi da se saobraćaj rutira od B preko x do C
  - .. tako x neće oglašavati B rutu do C

## BGP: kontroliše ko rutira do tebe



- ❑ A oglašava B put Aw
- ❑ B oglašava X put BAw
- ❑ Dali će B oglašavati C put BAw?
  - Nema šanse! B ne dobija “profit” za rutiranje CBAw pošto w i C nisu B-ovi korisnici
  - B želi da prinudi C da rutira do w preko A
  - B želi da rutira *samo* do/od njegovih korisnika!

## Zašto različito Intra- i Inter-AS rutiranje ?

---

### Politika:

- ❑ Inter-AS: administrator želi kontrolu nad načinom rutiranja saobraćaja i time ko rutira kroz njegovu mrežu.
- ❑ Intra-AS: jedan administrator, nema potrebe za političkim odlukama

### Veličina:

- ❑ hijerarhijsko rutiranje čuva veličinu tabele, smanjuje saobraćaj koji se odnosi na ažuriranje

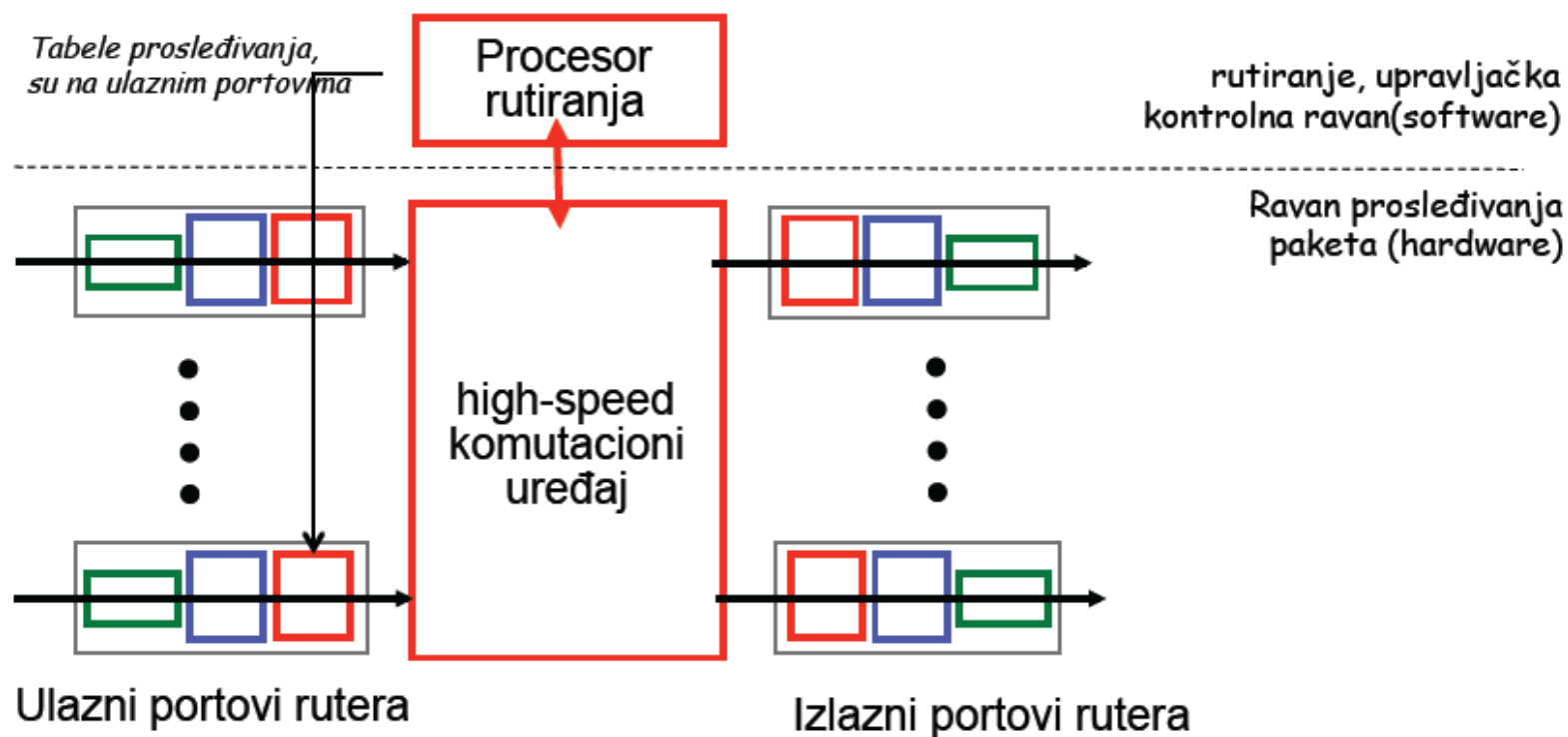
### Performanse:

- ❑ Intra-AS: može se fokusirati na performanse
- ❑ Inter-AS: politika može dominirati u odnosu na performanse

# Pregled arhitekture rutera

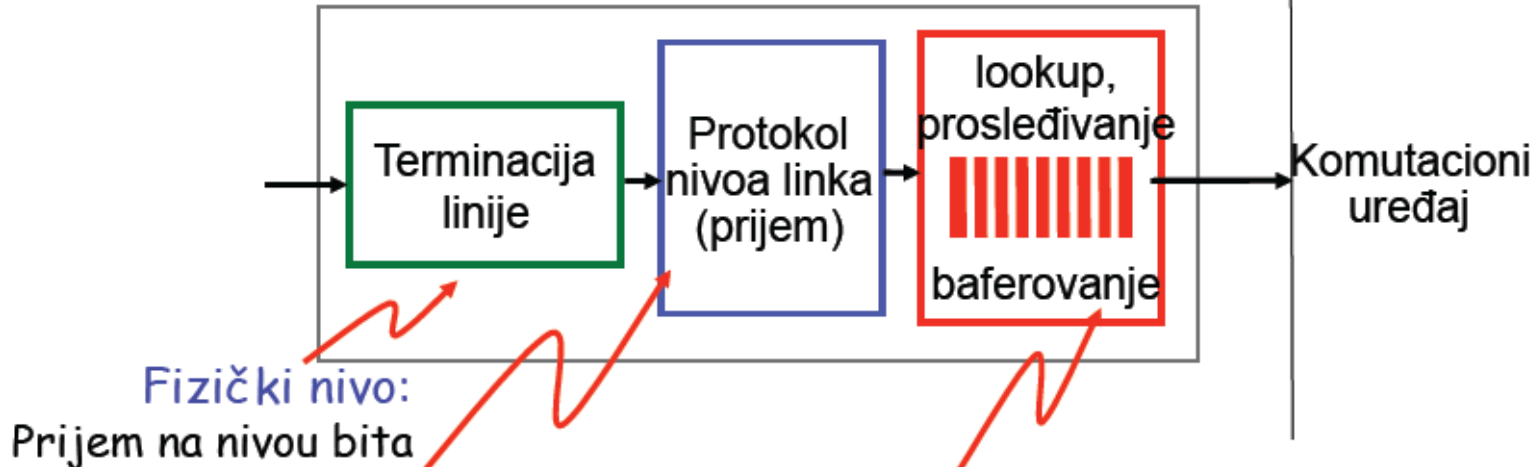
Dvije ključne funkcije rutera:

- Izvršava algoritme/protokole rutiranja (RIP, OSPF, BGP)
- *Prosleđuje (komutira)* datagrame sa ulaznog na izlazni link





# Funkcije ulaznog porta



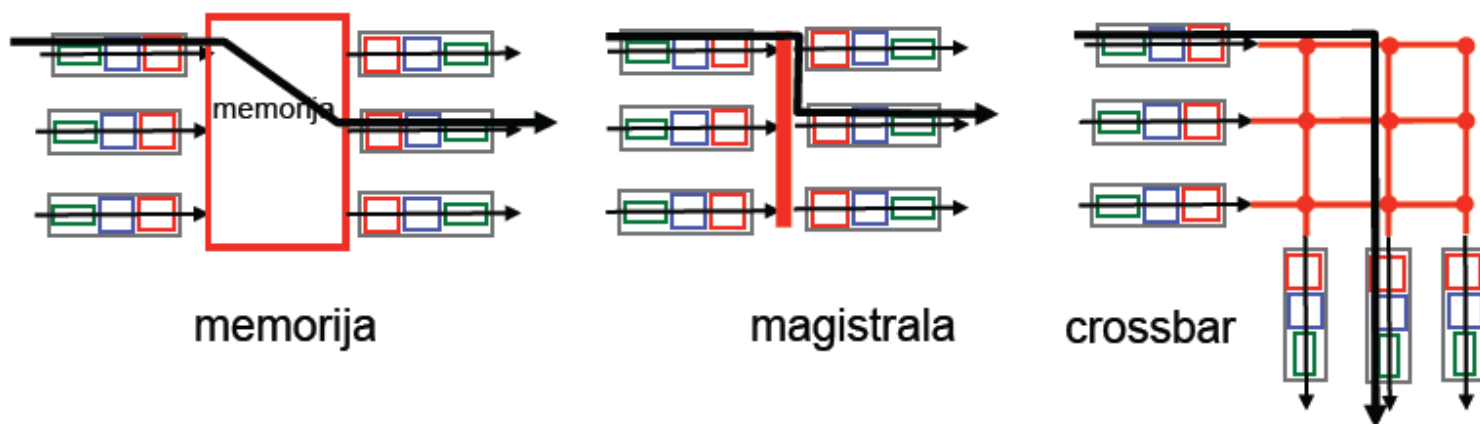
Fizički nivo:  
Prijem na nivou bita

Nivo linka:  
npr., Ethernet

## Decentralizovana komutacija:

- ❑ Na osnovu destinacije datagrama, traži izlazni port korišćenjem tabele rutiranja u memoriji ulaznog porta
- ❑ Cilj: kompletirati obradu na ulaznom portu u skladu sa brzinom na linku
- ❑ Red čekanja (bafer): ako datagrami pristižu brže nego što je brzina prosleđivanja u komutacionom uređaju

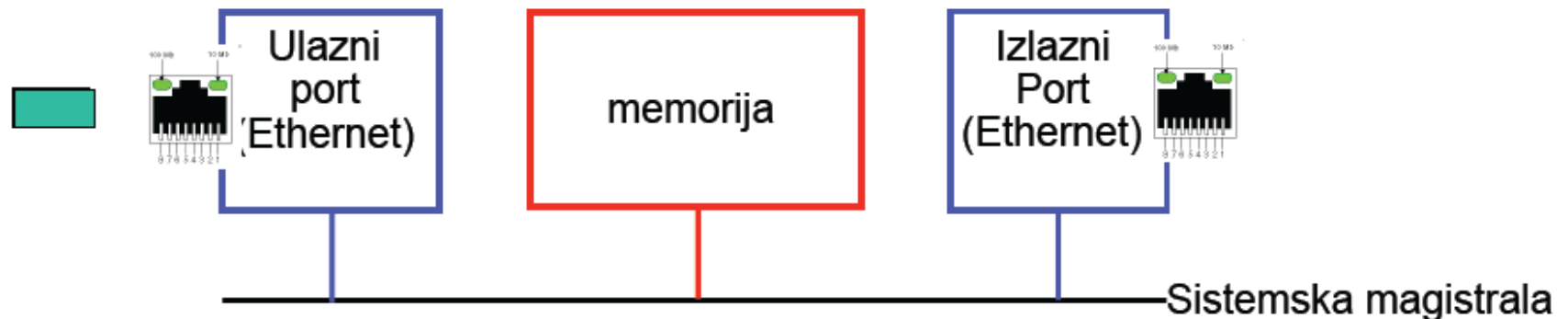
# Tri tipa komutacionih uređaja



# Komutacija preko zajedničke memorije

## Prva generacija rutera:

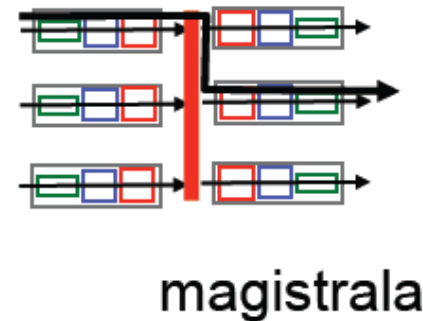
- ❑ tradicionalni računari sa komutacijom pod direktnom kontrolom CPU
- ❑ paketi se smještaju u memoriju sistema
- ❑ brzina ograničena brzinom memorije (svaki datagram se mora dva puta prenijeti preko magistrale)
- ❑ Cisco Catalyst switchevi serije 8500 (specifično rješenje)



# Komutacija preko zajedničke magistrale

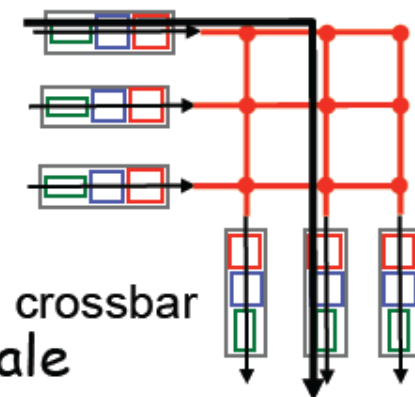
---

- Datagram se sa memorije ulaznog porta do memorije izlaznog porta prenosi preko zajedničke magistrale bez učešća procesora
- **Kolizija na magistrali:** brzina komutacije je ograničena kapacitetom magistrale
- 32 Gb/s magistrala, Cisco 5600: dovoljna brzina za pristupne i kompanijske rutere (neregionalne ili na okosnici)



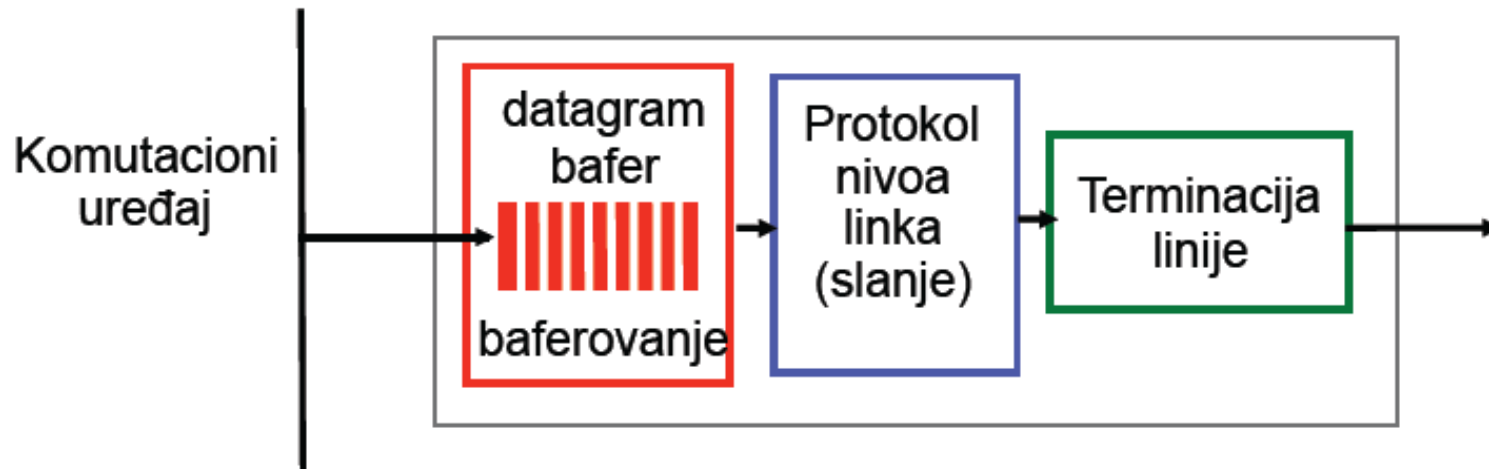
# Prostorni komutatori

---



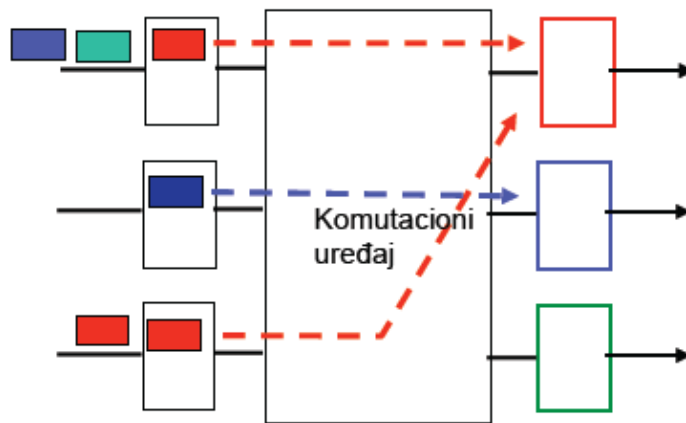
- ❑ Prevazilazi ograničenja kapaciteta magistrale
- ❑ Nudi više puteva između skupa ulaza i skupa izlaza
- ❑ Crossbar topologija
- ❑ Napredan dizajn: fragmentacija datagrama u ćelije fiksne dužine, komutiranje ćelija kroz uređaj.
- ❑ Cisco 12000: komutira do 60Gb/s kroz komutacionu matricu
- ❑ Banyan, Clos, paralelni...
- ❑ Komutacione strukture su inicijalno razvijene za povezivanje procesora u multiprocesorsku arhitekturu
- ❑ Komutiraju pakete fiksne dužine

# Izlazni portovi

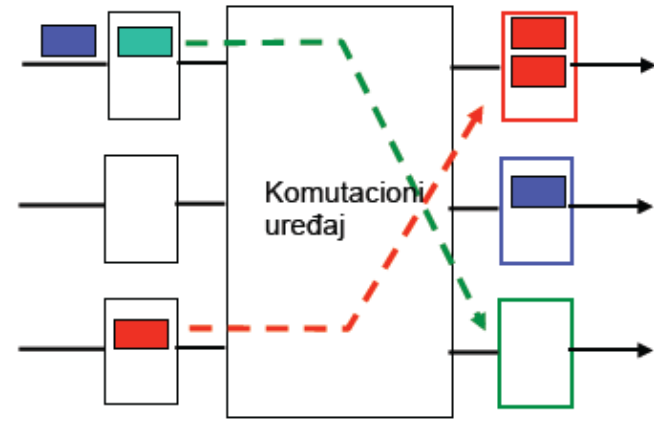


- ❑ *Baferovanje* se zahtijeva kada datagrami stižu iz uređaja većom brzinom nego što je brzina prenosa
- ❑ *Disciplina raspoređivanja "Scheduling"* bira za prenos datagrame u redovima čekanja

# Izlazno baferovanje



U slotu  $t$ , više paketa  
prema jednom izlazu



U narednom slotu

- ❑ Baferovanje kada je dolazna brzina veća od odlazne brzine
- ❑ *baferovanje (kašnjenje) i gubici zbog prepunog bafera na izlaznom portu!*

# Veličina bafera?

- RFC 3439 (rule of thumb) pravilo: srednja veličina bafera je jednaka “prosječno” RTT (npr 250ms) pomnoženo sa kapacitetom linka  $C$

$$RTT \cdot C$$

- npr.,  $C = 10\text{Gb/s}$  link: 298MB

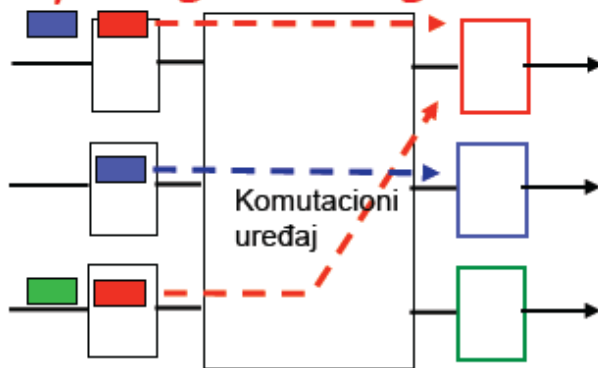
- Određene preporuke ukazuju da su moguće i manje memorije: za  $N$  tokova, potrebna veličina bafera je

$$\frac{RTT \cdot C}{\sqrt{N}}$$

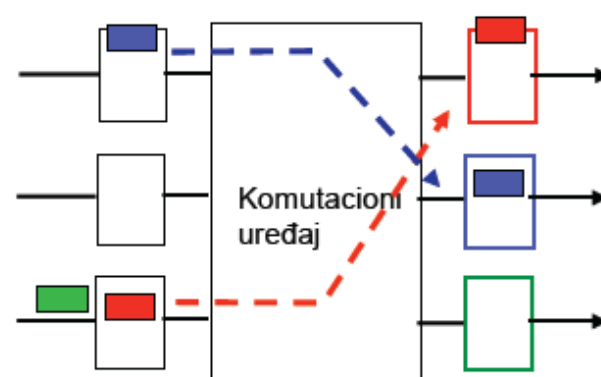


## Red čekanja na ulaznom portu

- ❑ Uređaj je sporiji od ulaznih portova -> redovi čekanja se mogu pojaviti na ulazima
- ❑ **Head-of-the-Line (HOL) blokiranje:** smještanje datagrama u redove čekanja ispred sprečava druge u redovima čekanja da se prosleđuju dalje
- ❑ **Kašnjenje u redovima čekanja i kašnjenje zbog prepunog ulaznog bafera!**



Izlazna kolizija u trenutku  $t$  - samo jedan crveni paket može biti proslijeđen



Zeleni paket doživljava HOL blokiranje