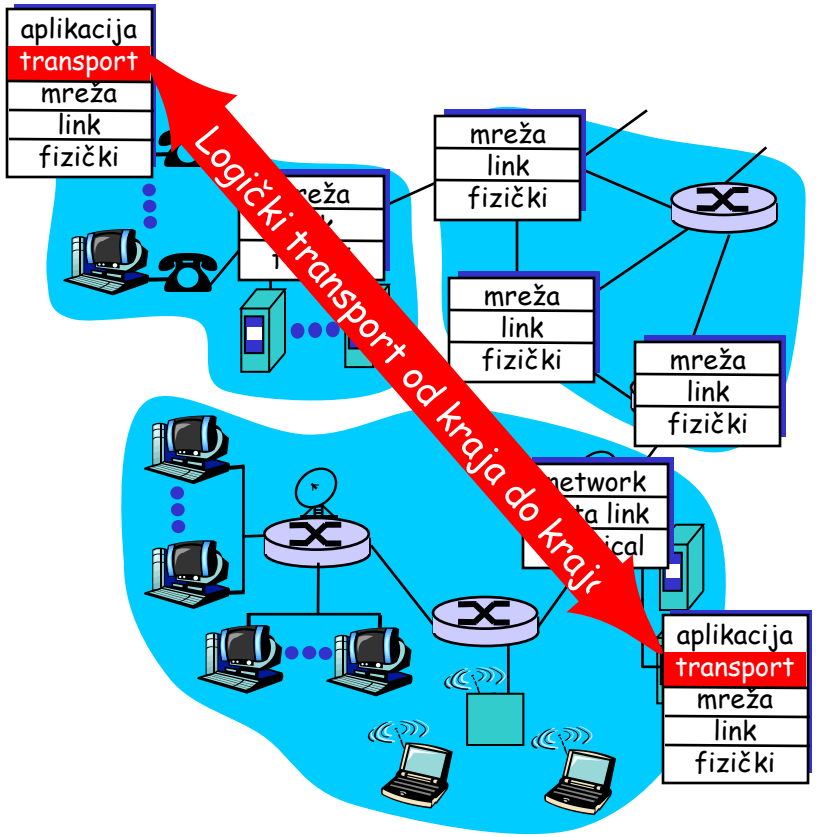
Transportni servisi i protokoli

- obezbjeđuju *logičku komunikaciju* između aplikacija koje se odvijaju na različitim hostovima
- transportni protokoli se implementiraju na krajnjim sistemima
  - Predajna strana: dijeli poruke u segmente, prosleđuje ih mrežnom nivou
  - Prijemna strana: desegmentira u poruke, i prosleđuje ih nivou aplikacije
- Više od jednog transportnog protokola je na raspolaganju aplikacijama
  - Internet: TCP i UDP



Transportni vs. Mrežni nivo

- *Mrežni nivo:* logička komunikacija između hostova
- *Transportni nivo:* logička komunikacija između procesa
  - Oslanja se na, poboljšava, servise mrežnog nivoa

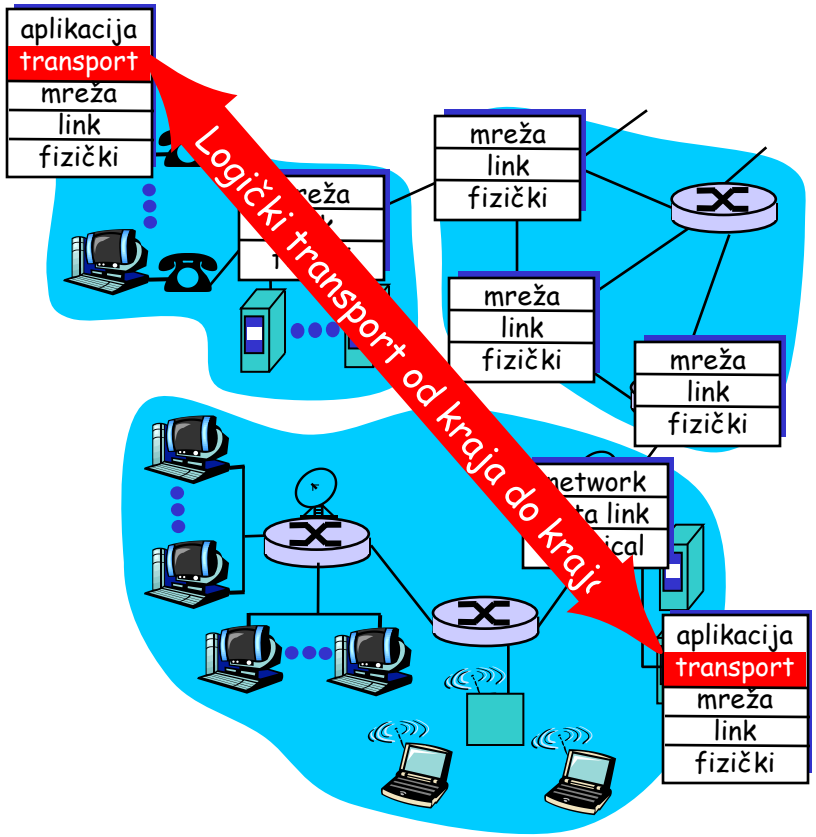
Analogija:

12 ljudi šalje pisma za 12 ljudi

- procesi = ljudi
- poruke = poruke u kovertama
- hostovi = kuće u kojima ljudi žive
- transportni protokol = zapis na koverti
- Mrežni protokol = poštanski servis

Internet protokoli transportnog nivoa

- nepouzdan, neredosledna isporuka: UDP
  - Bez unapređenja"best-effort" pristupa IP
- pouzdan, redosledna isporuka (TCP)
  - Kontrola zagušenja
  - Kontrola protoka
  - Uspostavljanje veze
- Servisi koji se ne pružaju:
  - Garantovano kašnjenje
  - Garantovana propusnost



UDP: User Datagram Protocol [RFC 768]

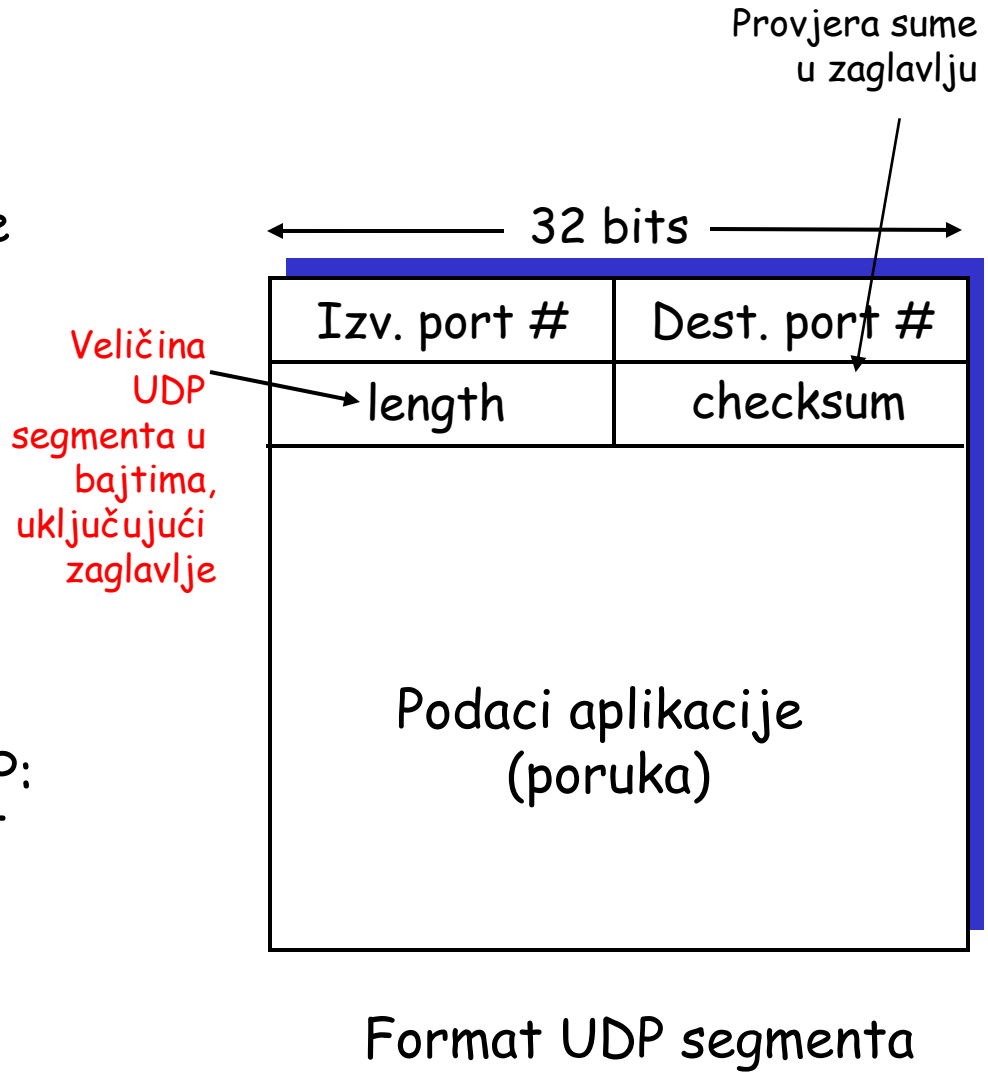
- Nema poboljšanja koji se nude Internet transport protokolu
- "best effort" servis, UDP segmenti mogu biti:
  - izgubljeni
  - predani neredosledno
- *nekonektivni:*
  - Nema uspostavljanja veze (handshaking) između UDP pošiljaoca i prijemnika
  - svaki UDP segment se tretira odvojeno od drugih

Zašto onda UDP?

- Nema uspostavljanja veze (koja povećava kašnjenje)
- jednostavnije: ne vodi se računa o stanju veze
- Manje zaglavlje segmenta
- Nema kontrole zagušenja: UDP može slati podatke onom brzinom kojom to aplikacija želi

UDP: više

- Često se koristi za "streaming" multimedijalne aplikacije
  - Tolerantne u odnosu na gubitke
  - Osjetljive na brzinu prenosa
- drugi UDP korisnici
  - DNS
  - SNMP
- Pouzdani prenos preko UDP: mora se dodati pouzdanost na nivou aplikacije
  - Oporavak od greške na nivou aplikacije



Format UDP segmenta

UDP checksum-a

Cilj: detekcija greške u prenošenom segmentu

Pošiljac:

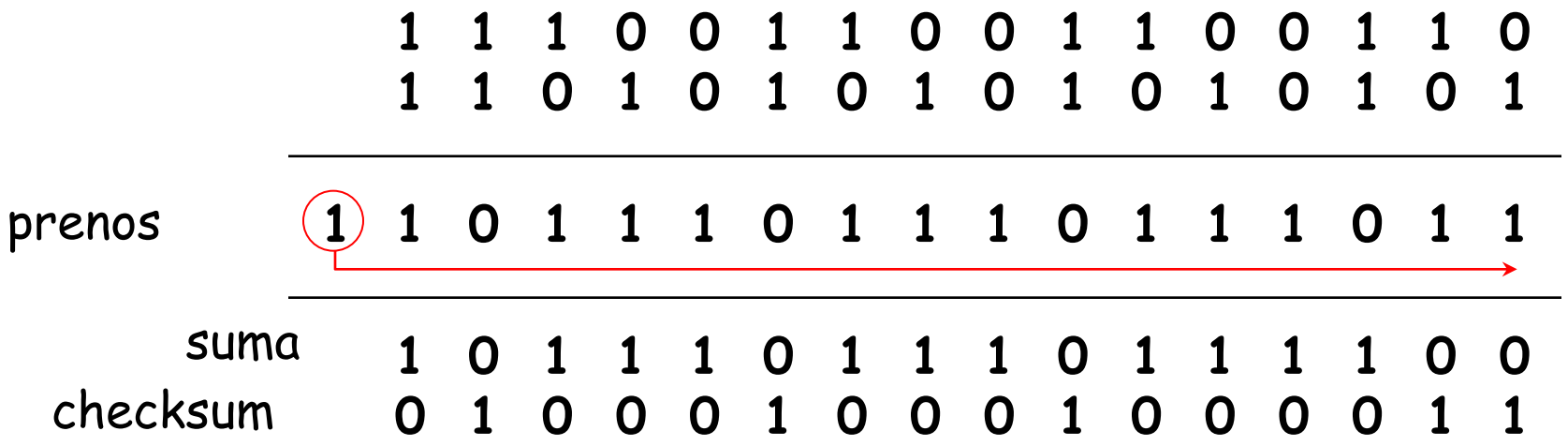
- Tretira sadržaj segmenta kao sekvence 16-bit prirodnih brojeva
- checksum: dodaje (suma 1 komplementa) informaciju segmentu
- Pošiljac postavlja vrijednost checksum -e u odgovarajuće polje UDP segmenta

Prijemnik:

- Proračun checksum-e primljenog segmenta
- Provjera da li je izračunata checksum-a jednaka vrijednosti u odgovarajućem polju:
  - NE – detektovana greška
  - Da – nema greške. *Da li ste sigurni?*

Internet Checksum-a primjer

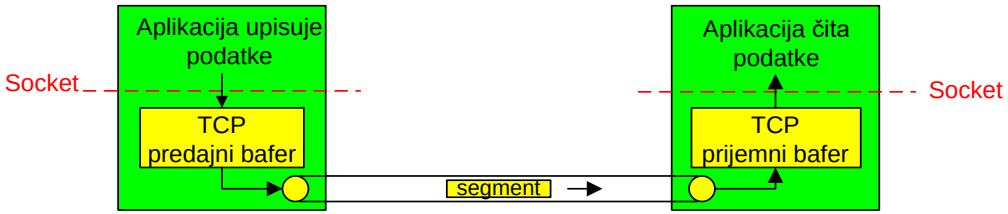
- Napomena
  - Kada se sabiraju brojevi, prenos sa najznačajnijeg bita se dodaje rezultatu



TCP: Pregled

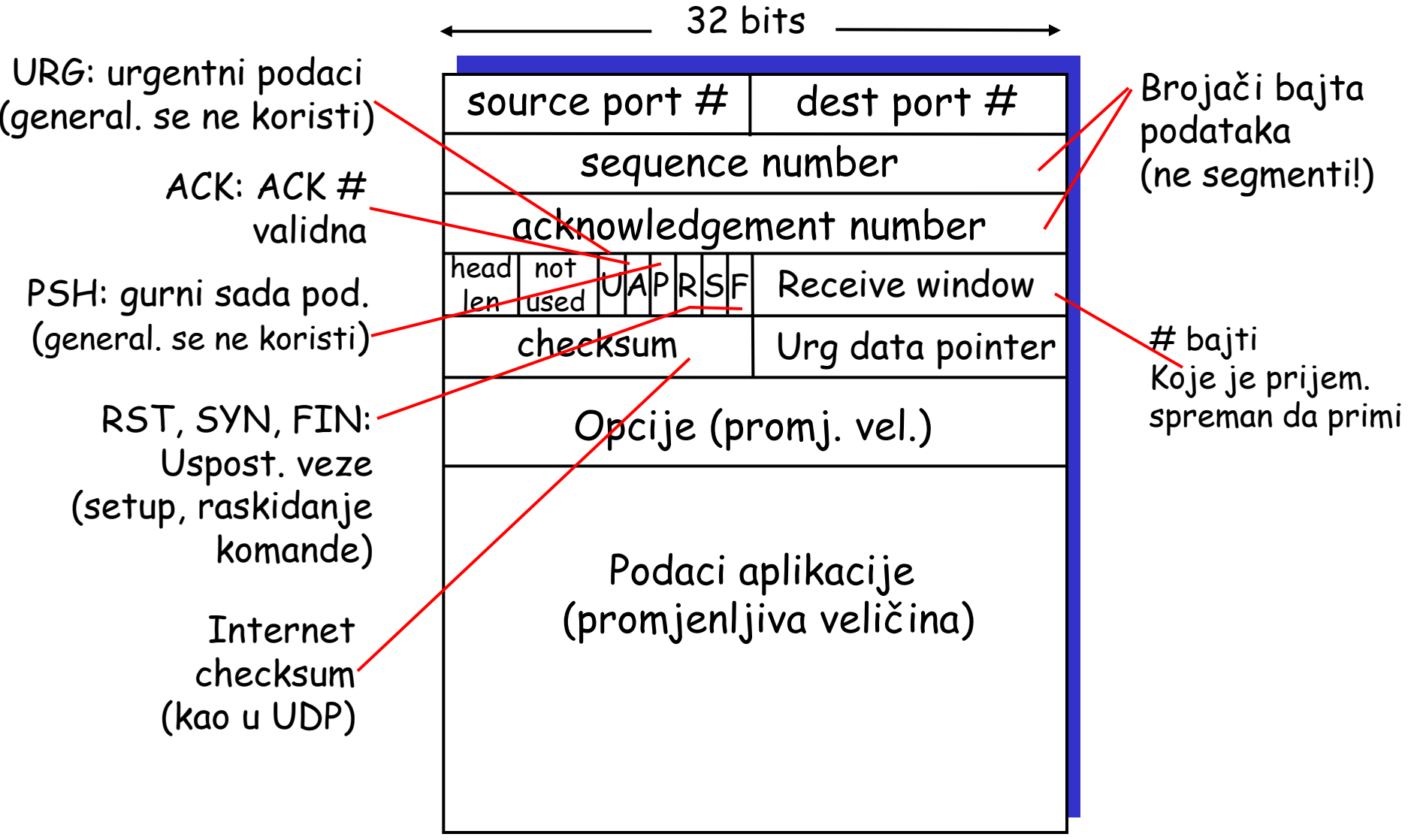
RFCovi: 793, 1122, 1323, 2018, 2581

- tačka-tačka:
  - Jedan pošilj, jedan prij.
- pouzdan, redosledan prenos bajta:
  - nema "granica poruka"
- "pipelined":
  - TCP kontrola zagušenja i protoka podešava veličinu prozora
- *Baferi za slanje & prijem*



- "full duplex" prenos:
  - U istoj vezi prenos u dva smjera
  - MSS: maksimalna veličina segmenta (1460B, 536B, 512B)
- konektivan:
  - "handshaking" (razmjena kontrolnih poruka) inicira je pošiljalac, razmjenjuje stanja prije slanja
- Kontrolisani protok:
  - Pošiljalac ne može "zagušiti" prijemnika

TCP struktura segmenta



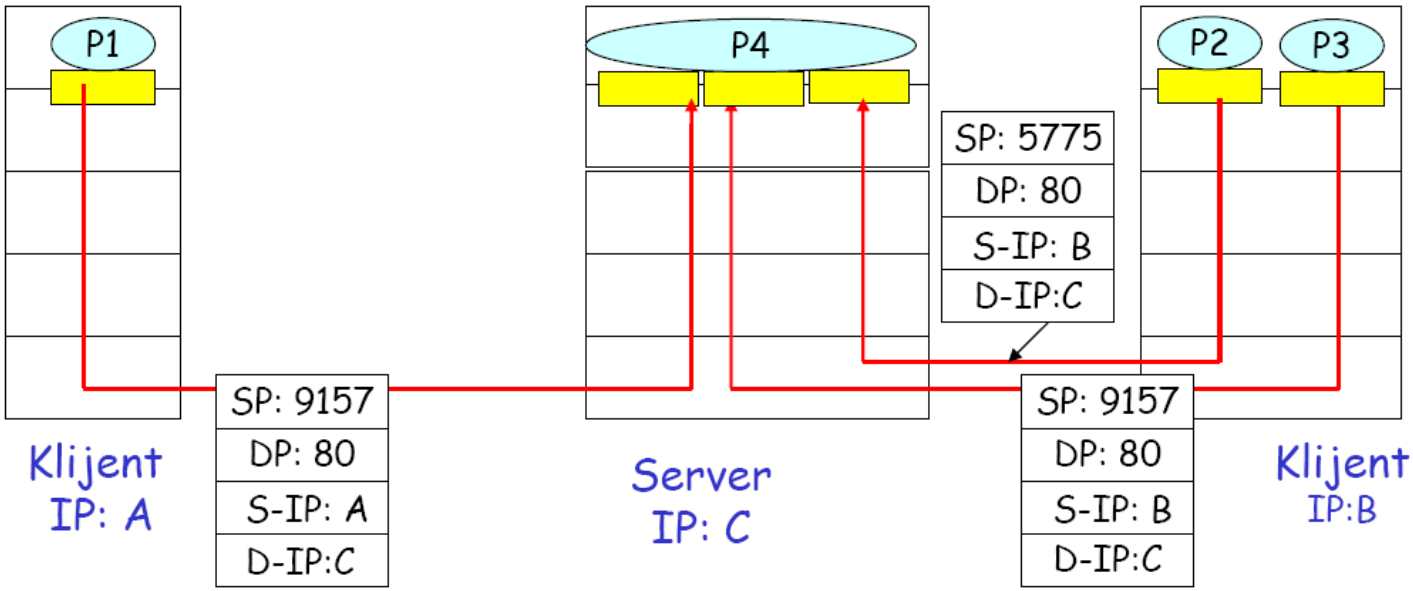
TCP pouzdani prenos podataka

- TCP kreira pouzdani servis po IP nepouzdanom servisu
- "Pipelined" segmenti
- Kumulativne potvrde
- TCP koristi jedan retransmisioni tajmer
- Retransmisije su trigerovane sa:
  - timeout događajima
  - duplim ack-ovima
- Na početku razmotrimo pojednostavljenog TCP pošiljaoca:
  - Ignorišu se duplirani ack-ovi
  - Ignorišu se kontrole protoka i zagušenja

Primjer TCP demultipleksiranja

TCP veza je jedinstveno identifikovana sa četiri parametra:

- izvorišne IP adresa
- izvorišni broj porta
- destinaciona IP adresa i
- destinacioni broj porta.

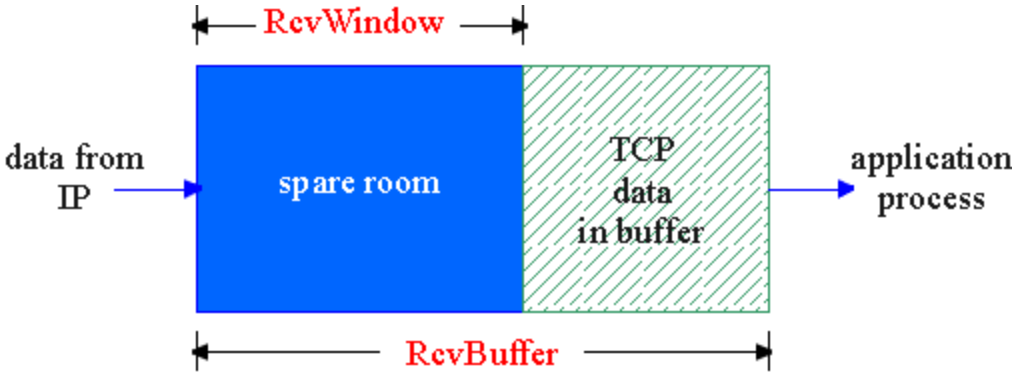


TCP kontrola protoka

Kontrola protoka

Pošiljalac neće "zagušitit" prijemnikov bafer šaljući previše podataka

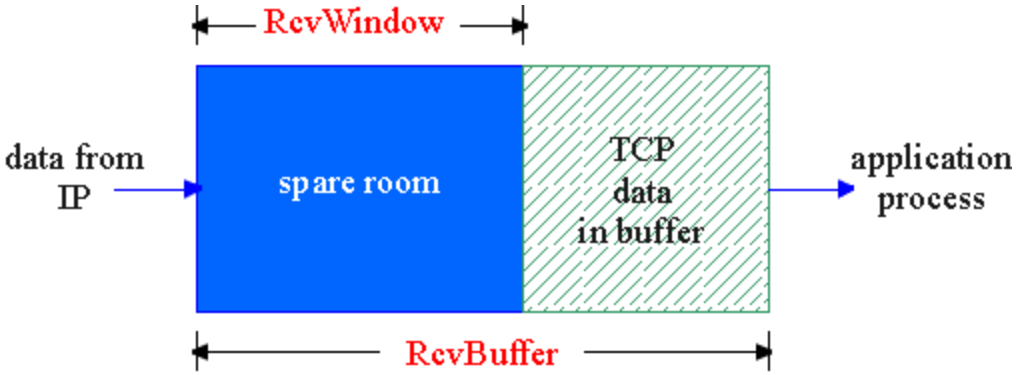
- Prijemna strana TCP veze ima prijemni bafer:



- Servis podešavanja brzine: podešavanje brzine slanja brzini pražnjanja bafera

- Aplikacioni proces može sporo čitati podatke iz bafera

TCP kontrola protoka: kako funkcioniše



(Pretpostavimo da TCP prijemnik odbacuje out-of-order segmente)

- Slobodan dio bafera

= **RcvWindow**

= **RcvBuffer-[LastByteRcvd - LastByteRead]**

- Prijemnik oglašava slobodan dio bafer ubacujući vrijednost **RcvWindow** u segmente
  - Garancija da se prijemni bafer neće prepuniti

TCP Upravljanje vezom

Ponovimo: TCP pošiljalac i prijemnik uspostavljaju "vezu" prije razmjene segmenata sa podacima

- Inicijalizuju se TCP variable:
  - Broj u sekvenci
  - baferi, info o kontroli protoka (npr. **`RcvWindow`**)
- *klijent:* inicijator veze

"Three way handshake":

Korak 1: Host klijenta šalje TCP SYN segment serveru
- specificira inicijalni broj u sek.
- nema podataka

Korak 2: Host servera prima SYN, odgovara sa SYNACK segmentom
- server dodjeljuje bafer
- specificira inicijalni broj sekvence servera

Korak 3: klijent prima SYNACK, odgovara sa ACK segmentom, koji može sadržati podatke

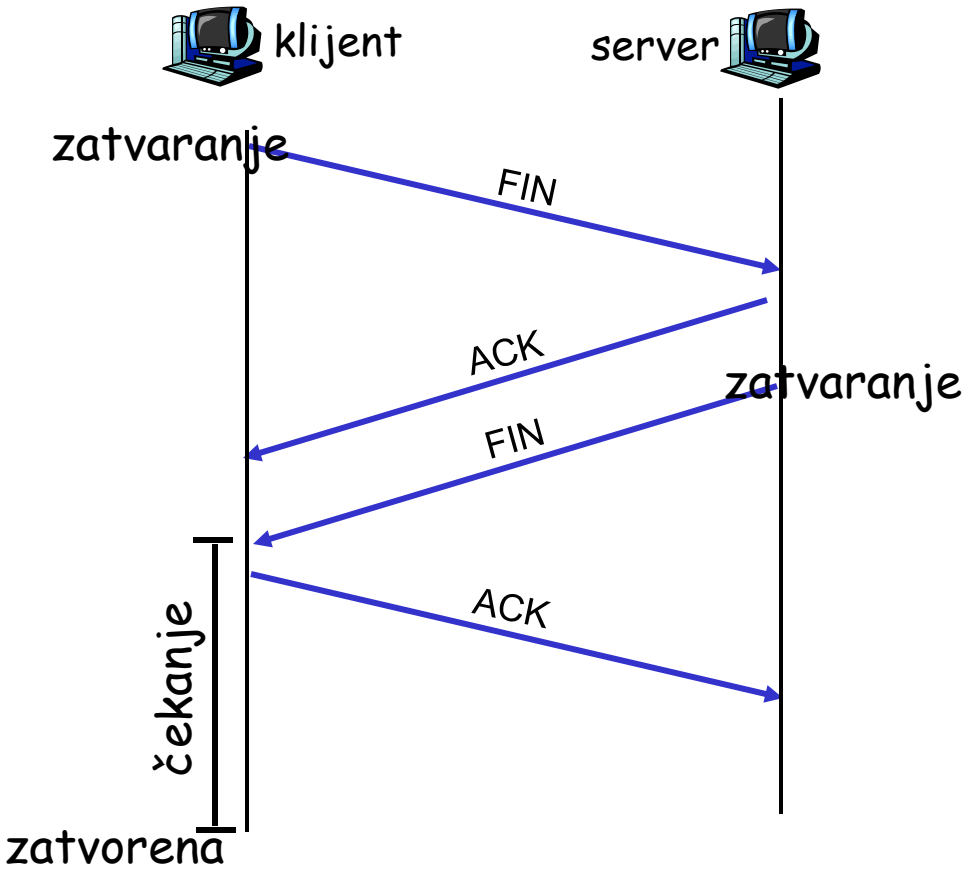
TCP Upravljanje vezom(nastavak)

Raskidanje veze:

klijent zatvara socket:
`clientSocket.close();`

Korak 1: klijent krajnjeg sistema šalje TCP FIN kontrolni segment serveru

Korak 2: server prima FIN, odgovara sa ACK. Zatvara vezu, šalje FIN.



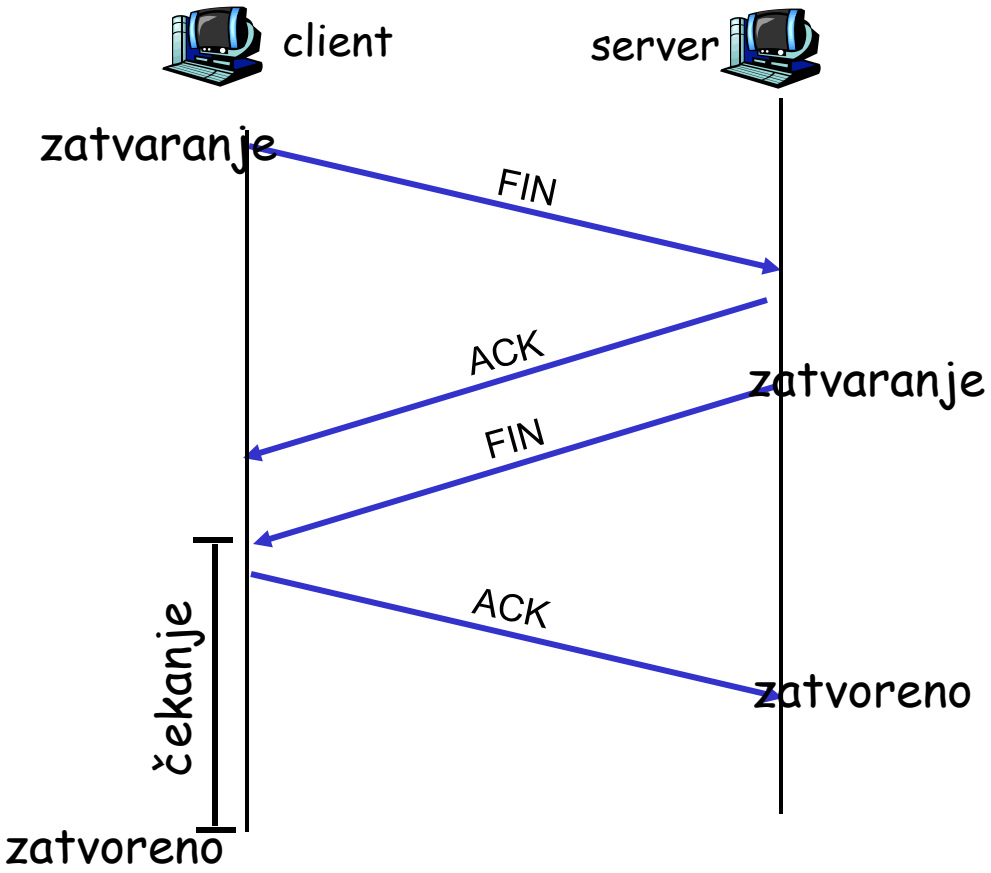
TCP Upravljanje vezom(nastavak)

Korak 3: klijent prima FIN, odgovara sa ACK.

- Unosi "čekanje" – odgovoriće sa ACK na primljene FIN-ove

Korak 4: server, prima ACK. Veza se raskida.

Napomena: sa malim modifikacijama može podržati više simultanih FIN-ova.



TCP kontrola zagušenja

- Kontrola od kraja do kraja (bez učešća mreže)
- Pošiljalac ograničava slanje:

  LastByteSent-LastByteAcked

  ≤ CongWin

- Približno,

$$\text{brzina} = \frac{\text{CongWin}}{\text{RTT}} \quad \text{B/s}$$

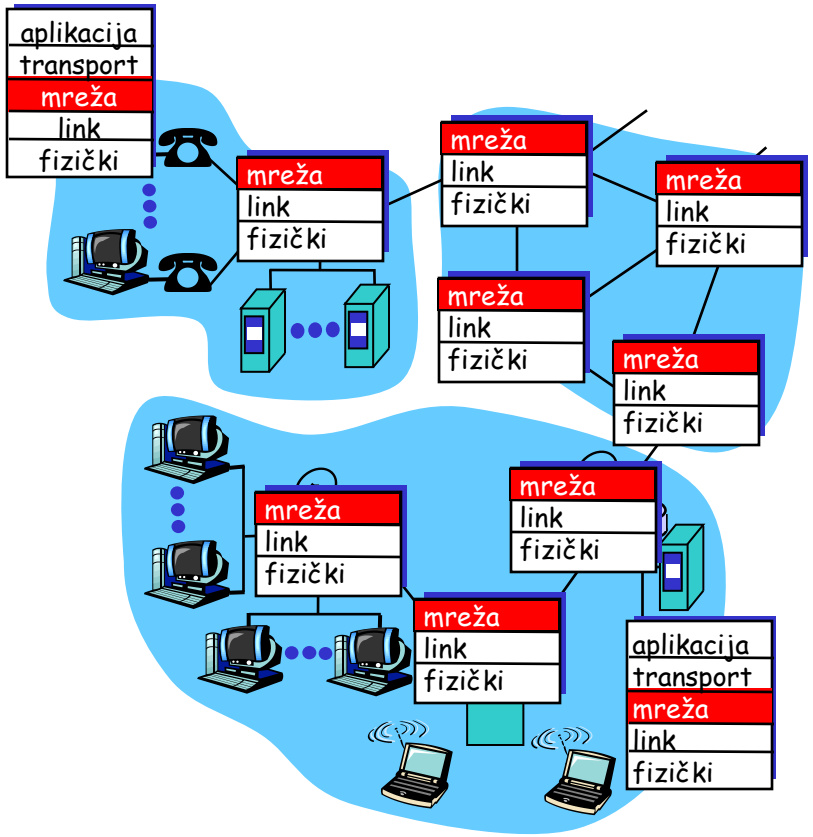
- **CongWin** je dinamička funkcija detekcije zagušenja mreže

Kako pošiljac otkriva zagušenje?

- gubitak = timeout *ili* 3 duplirane potvrde
- TCP pošiljalac smanjuje brzinu (**CongWin**) poslije gubitka

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



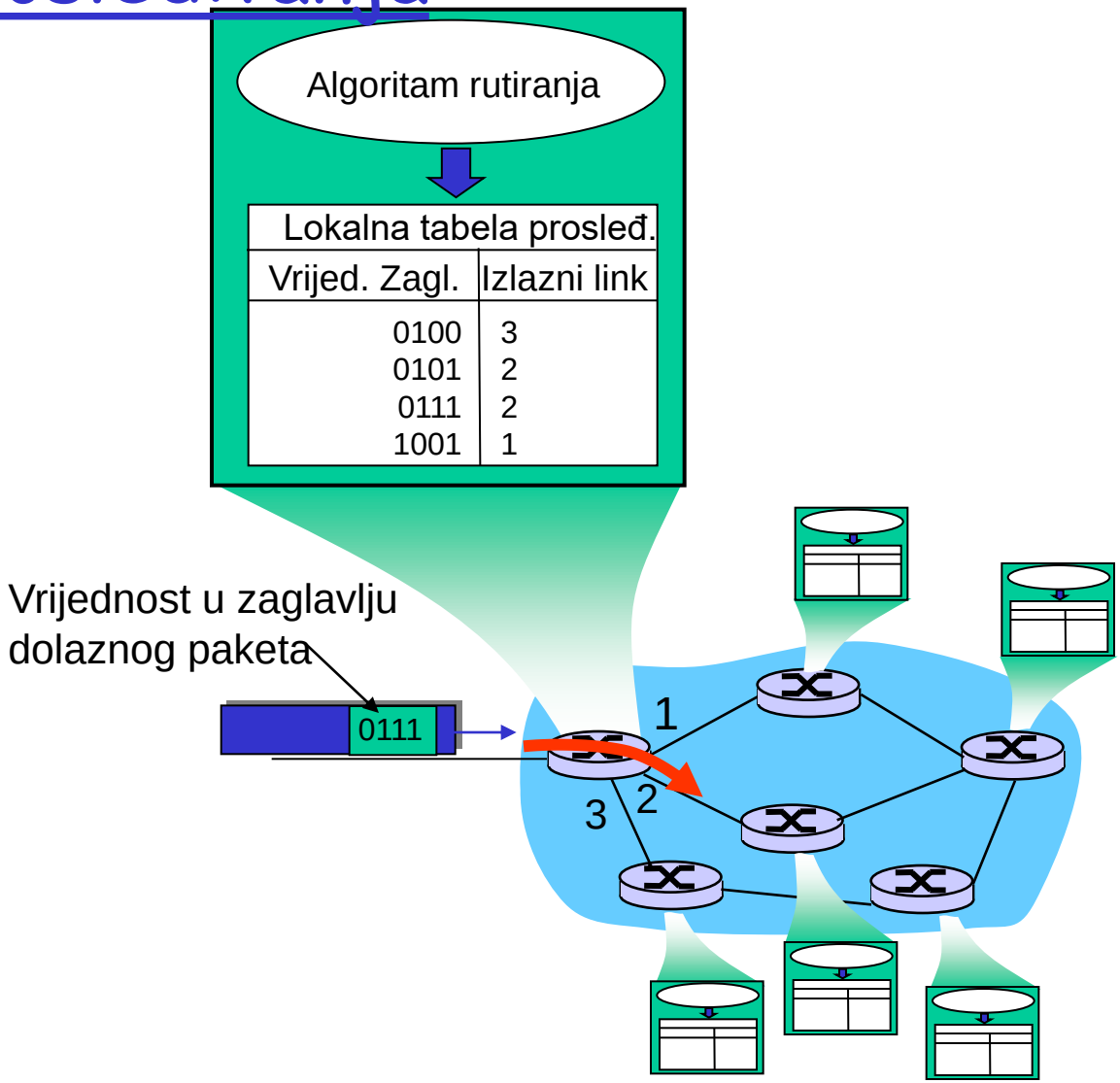
Ključne funkcije mrežnog nivoa

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

analogija:

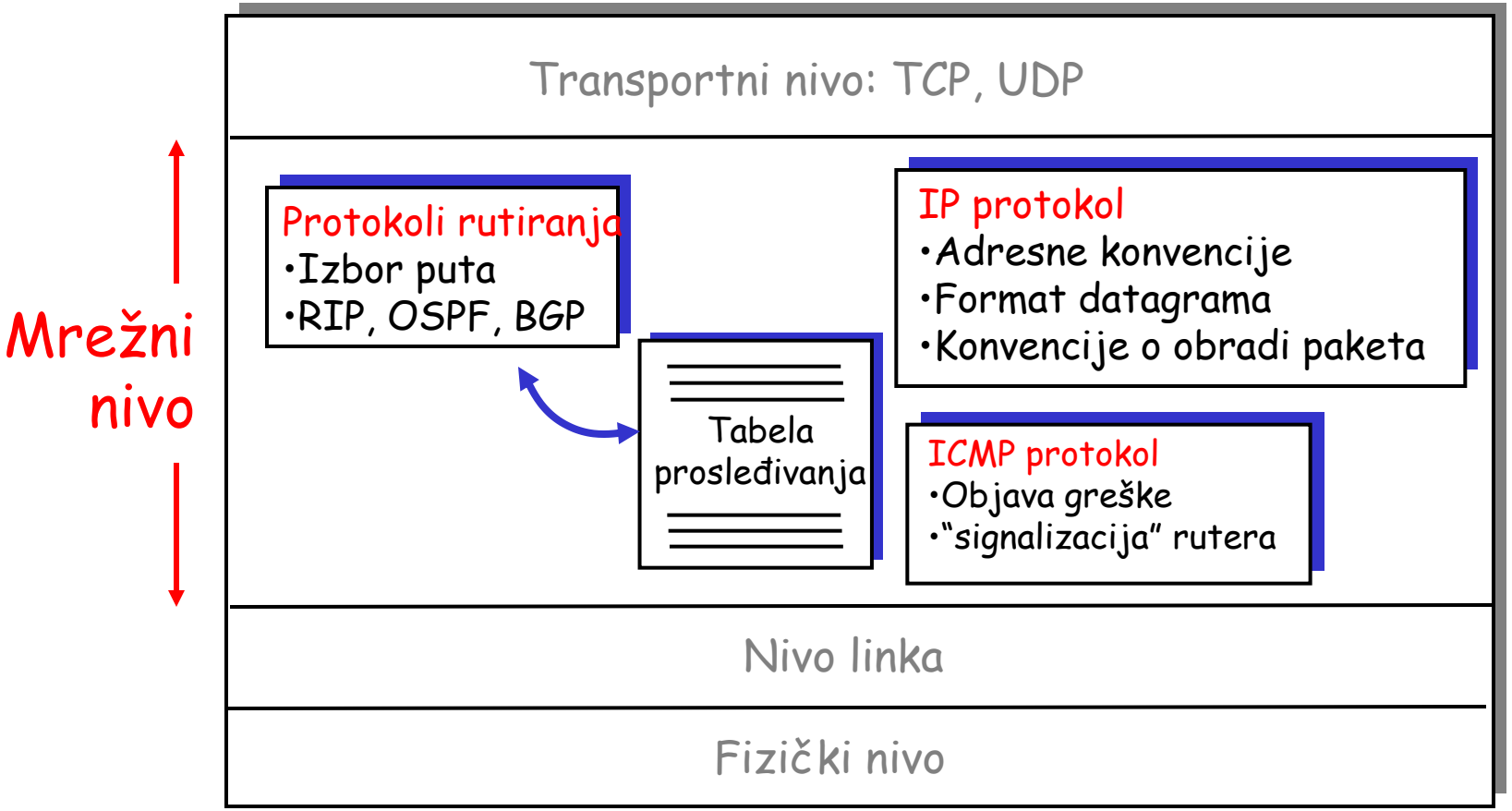
- rutiranje: proces planiranja putovanja
- prosleđivanje: proces prolaska kroz jednu raskrsnicu

Međusobna povezanost rutiranja i prosleđivanja

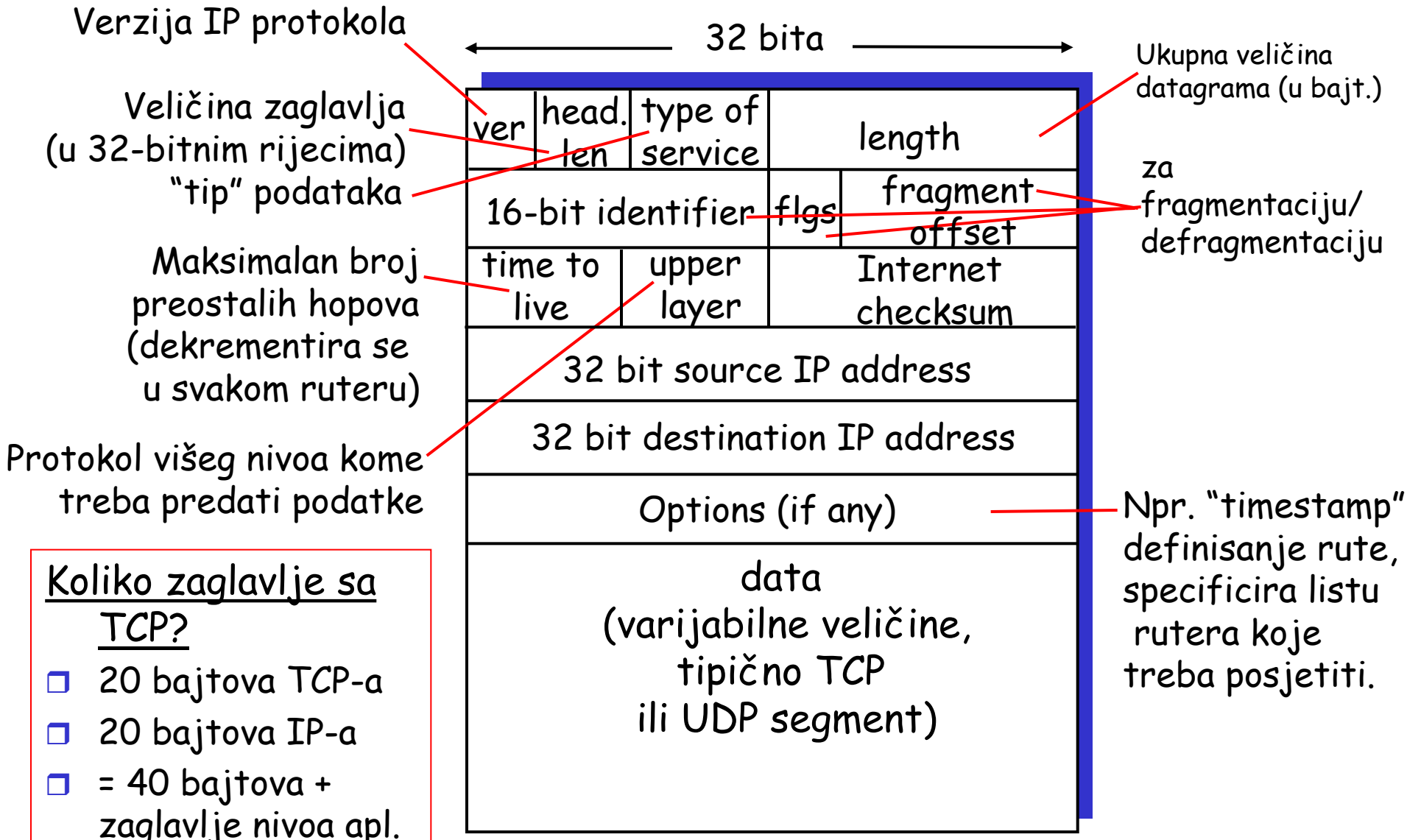


Internet mrežni nivo

Host i ruter funkcije mrežnog nivoa:



Format IP datagrama

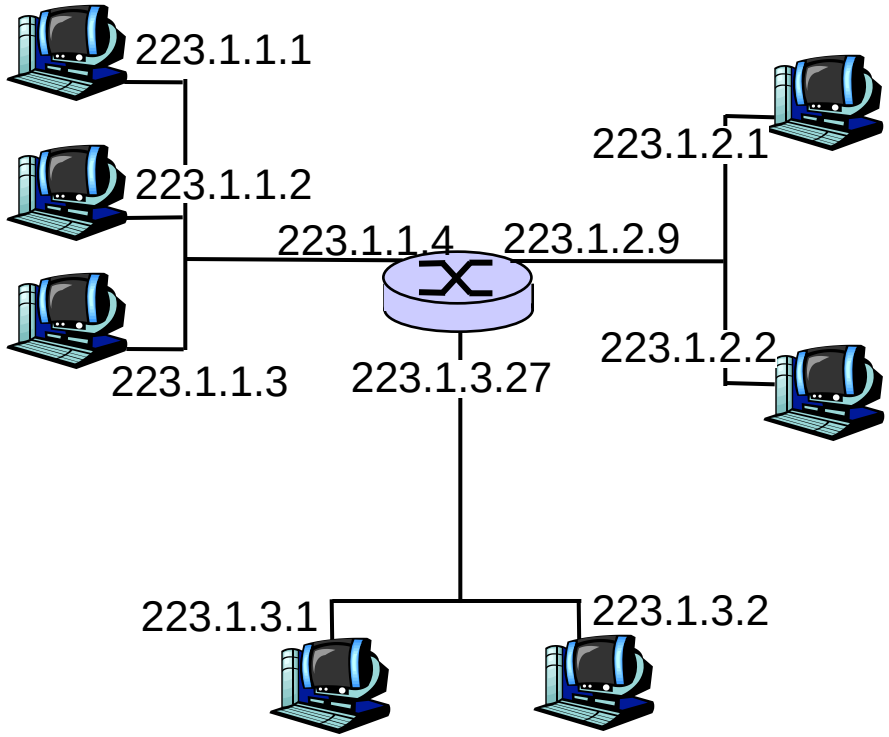


Koliko zaglavlje sa TCP?

- 20 bajtova TCP-a
- 20 bajtova IP-a
- = 40 bajtova + zaglavlje nivoa apl.

IP Adresiranje: uvod

- IP adresa: 32-bitni identifikator za host, ruter *interfejs*
- *interfejs*: veza između host/rutera i fizički link
  - ruteri tipično imaju više interfejsa
  - i host može imati više interfejsa
  - IP adrese su vezane za svaki interfejs

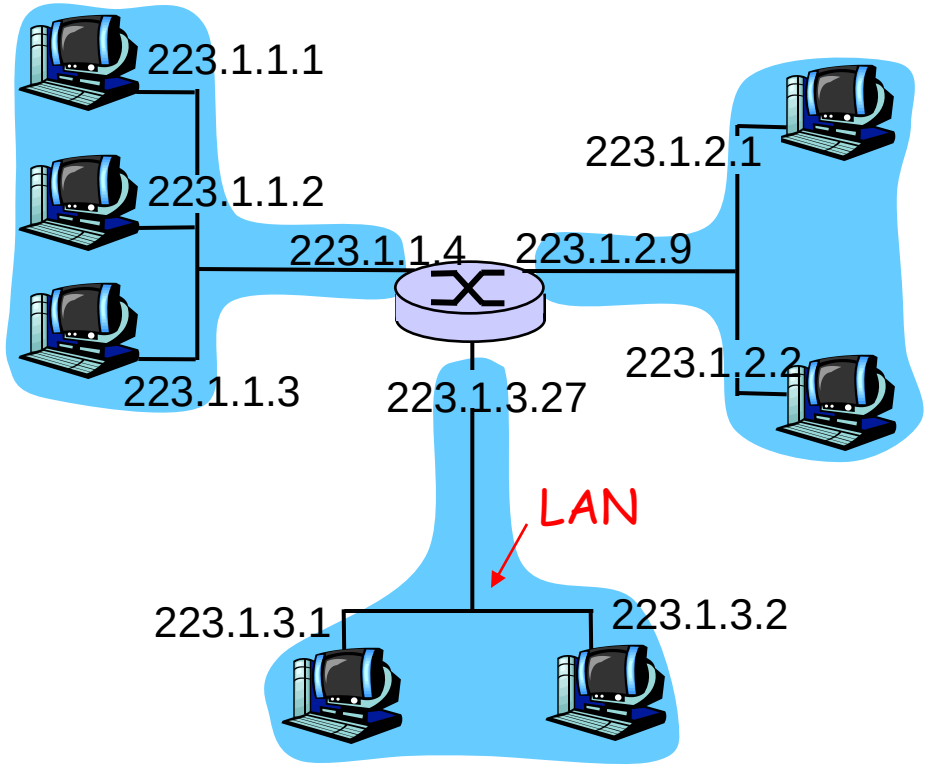


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 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

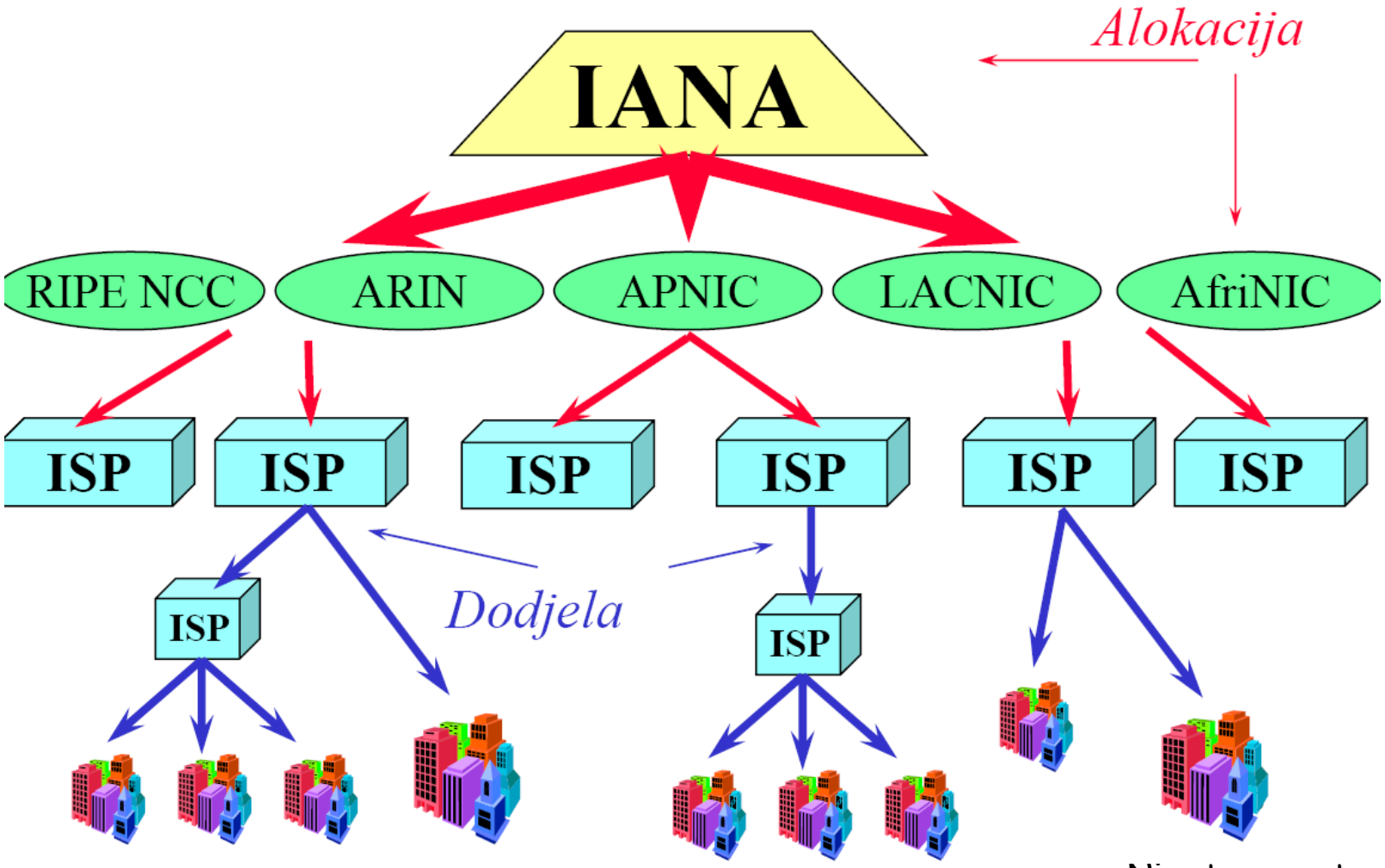


IPv4

Javne IP adrese

- Internet Assigned Numbers Authority (IANA) upravlja njihovom dodjelom
- Korisnicima IP adrese dodjeljuju Internet operatori (Internet service providers - ISP).
- Internet operatori dobijaju opsege IP adresa od odgovarajućeg Regionalnog Internet Registra (RIR)

Dodjela IPv4 adresa



IPv4

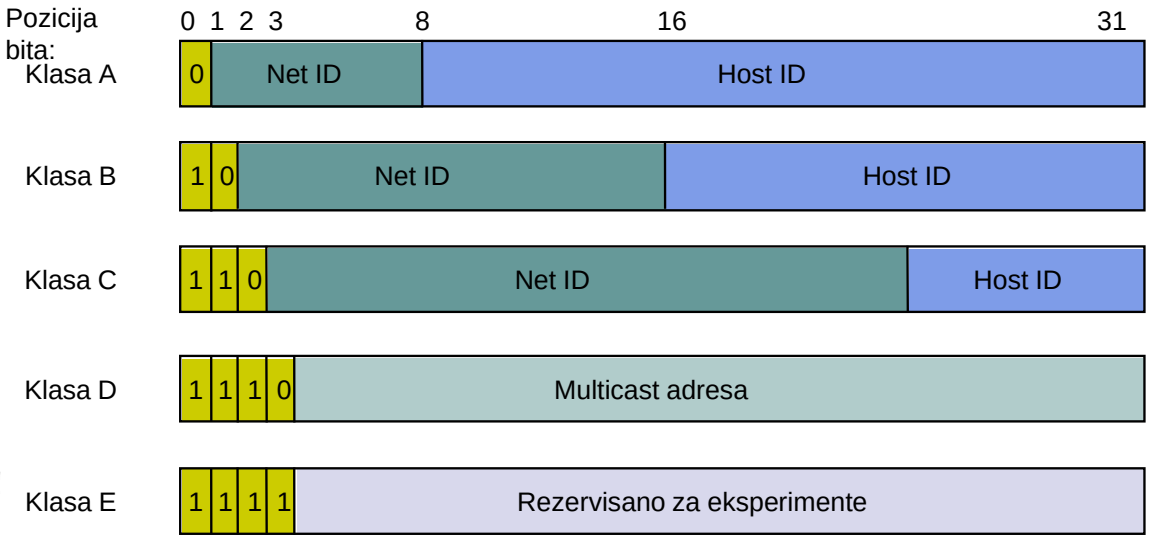
Regionalni Internet Registri (RIR)

- APNIC (Asia Pacific Network Information Centre) – Region Azije i Pacifikan
- ARIN (American Registry for Internet Numbers) – Sjeverna Amerika i države Afrike južno od Ekvatoraca
- LACNIC (Regional Latin-American and Caribbean IP Address Registry) – Latinska Amerika i neka Karibska ostrva
- RIPE NCC (Réseaux IP Européens) – Evropa, Bliski Istok, Centralna Azija

IPv4: Regionalni registri

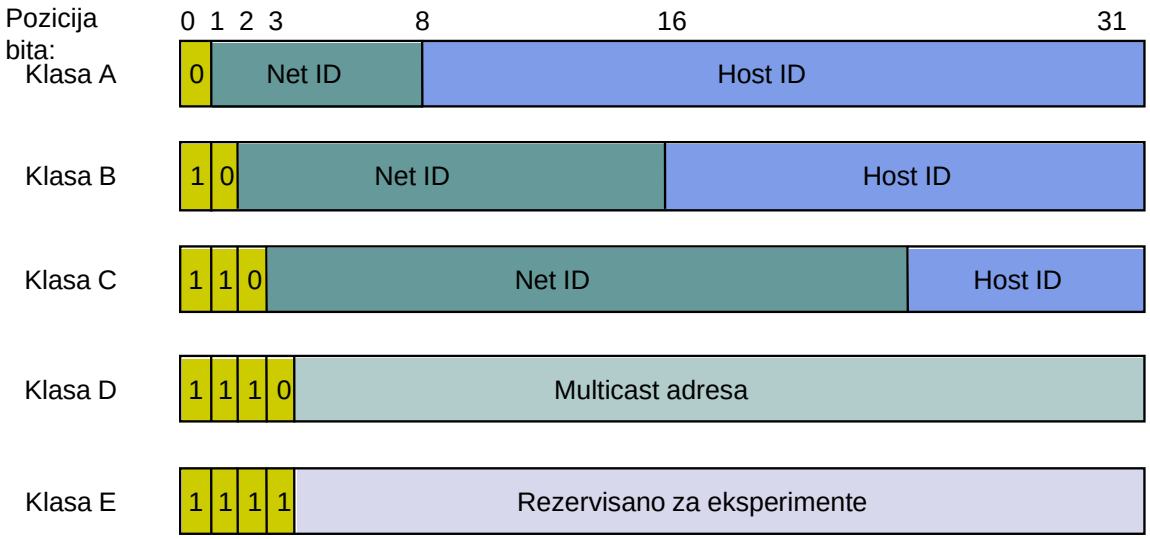
Classful IP Adresiranje

- 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.



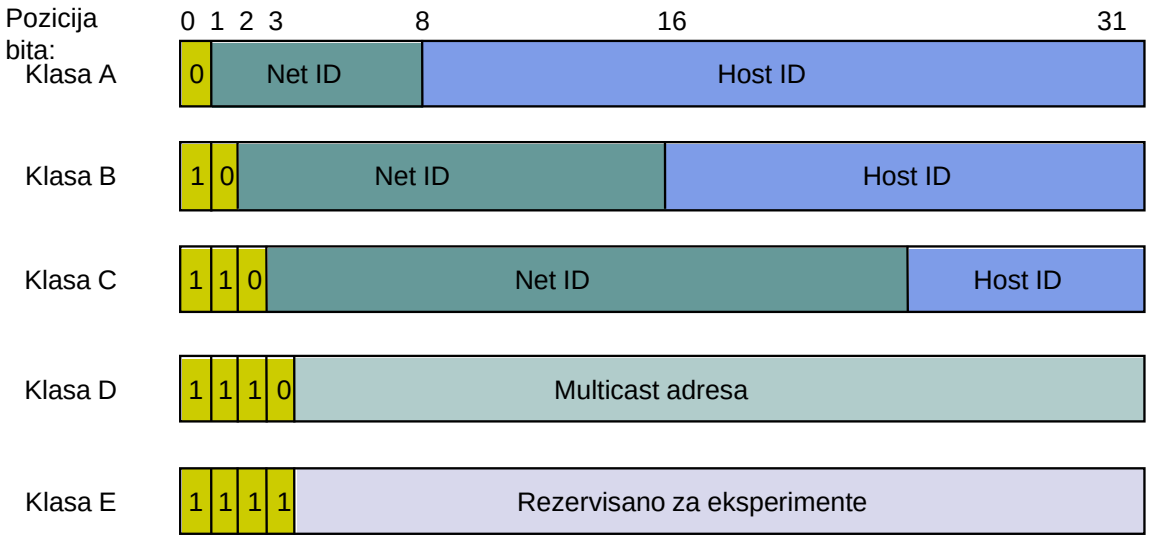
Classful IP Adresiranje

- Klasa B ima 14 bita za mrežni ID i 16 bita za host 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.



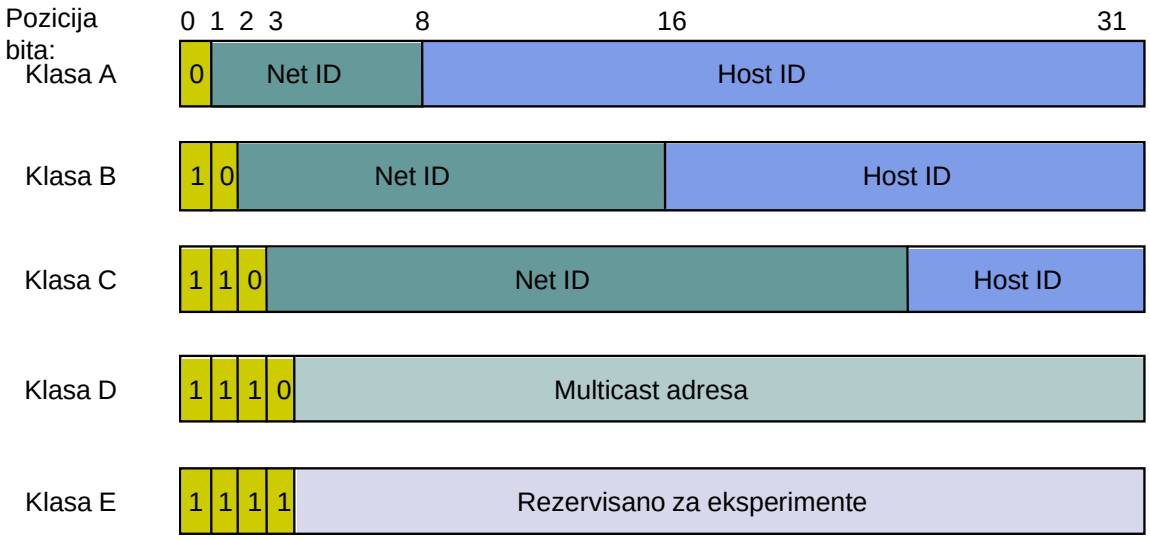
Classful IP Adresiranje

- 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.
  - Host ID koji se sastoji od svih 0 odgovara adresi mreže.



Classful IP Adresiranje

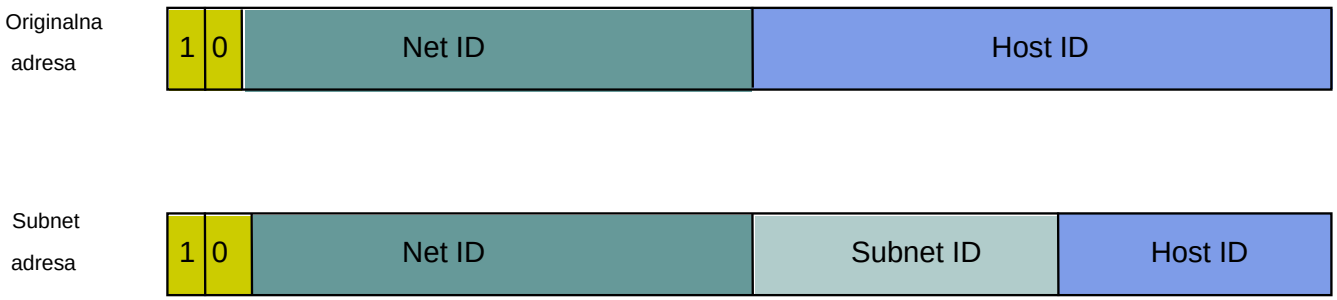
- IP adrese se najčešće pišu u formi tačka-decimalnog zapisa koji je pogodan za korišćenje od strane čovjeka. Adresa se dijeli na četiri bajta, pri čemu svaki bajt predstavlja decimalni broj, koji su razdvojeni tačkama. Na primjer adresa
- 10000000 10000111 01000100 00000101
  - 128 . 135 . 68 . 5
- 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)

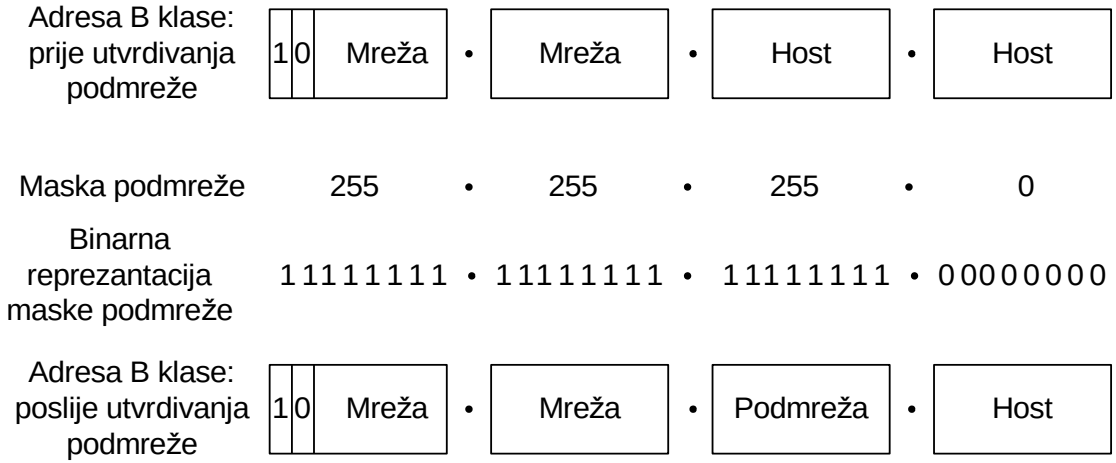
Classful IP Adresiranje

- Opisano IP adresiranje ima više nedostataka.
- Ovo adresiranje može biti vrlo neefikasno. Na primjer, dodjela B klase jednoj akademskoj instituciji koja ima nekoliko lokalnih računarskih mreža je besmislena.
- Rješenje ovog problema je nađeno 1980-tih kada je usvojen koncept podmreže (subnetting), odnosno dodavanje još jednog hijerarhijskog nivoa - 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

U tabeli je prikazana je veza između binarne i decimalne reprezentacije maske podmreže.

128	64	32	16	8	4	2	1	
1	0	0	0	0	0	0	0	128
1	1	0	0	0	0	0	0	192
1	1	1	0	0	0	0	0	224
1	1	1	1	0	0	0	0	240
1	1	1	1	1	0	0	0	248
1	1	1	1	1	1	0	0	252
1	1	1	1	1	1	1	0	254
1	1	1	1	1	1	1	1	255

- Default maske podmreža su:
- 255.0.0.0 (A klasa)
- 255.255.0.0 (B klasa)
- 255.255.255.0 (C klasa)

Classful IP Adresiranje

172	16	125	1	zadata adresa u decimalnom formatu

IP adresiranje: CIDR

- CIDR: Classless InterDomain Routing
  - 1993
  - Mrežni dio adrese je proizvoljne veličine
  - Format adrese: a.b.c.d/x, gdje je x broj bita u mrežnom dijelu adrese

11001000 00010111 00010000 00000000

200.23.16.0/23

IP adrese: Podjela besklasnog IP adresnog prostora

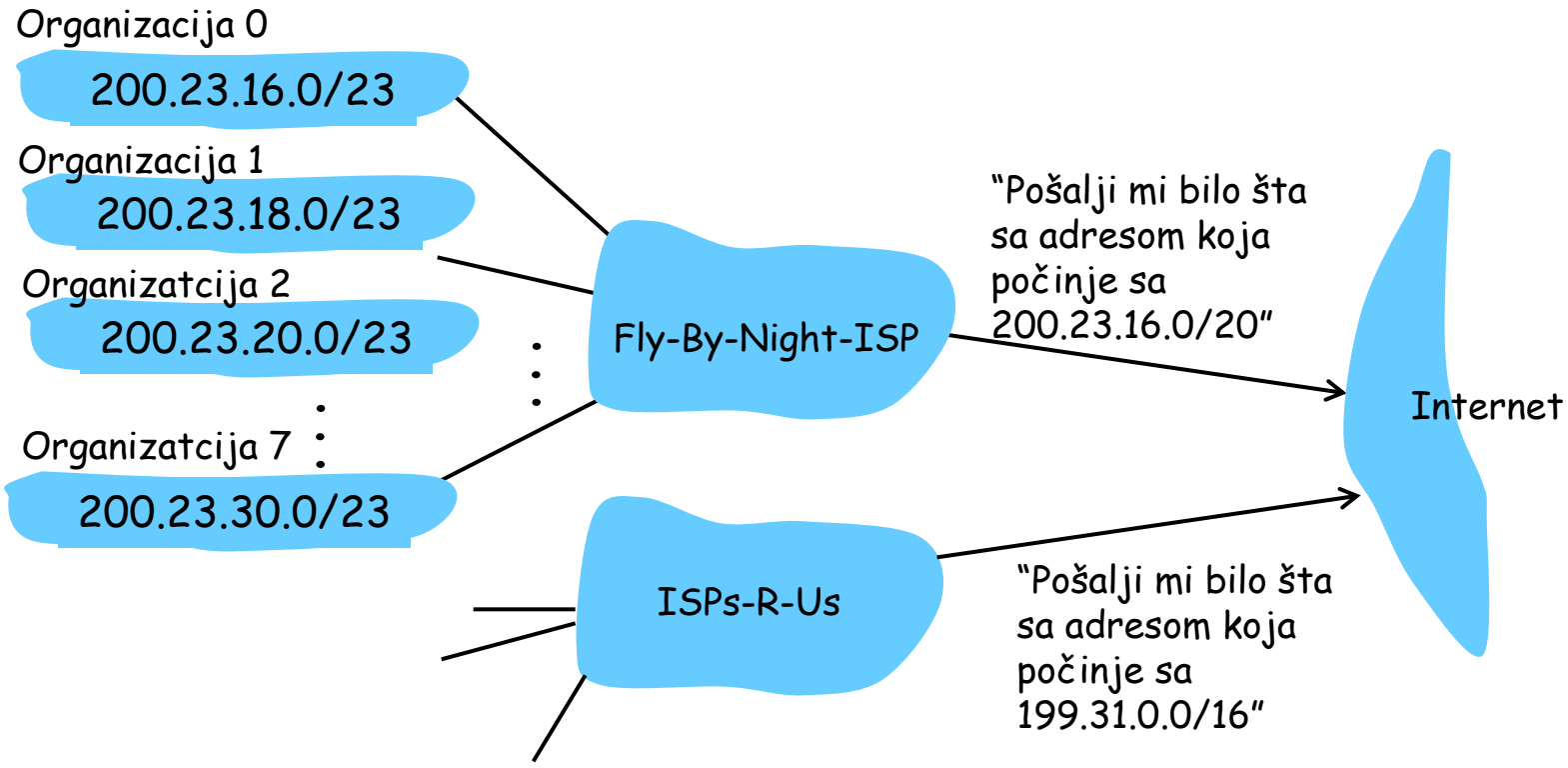
P: Kako podijeliti adresni prostor na 8 podmreža?

O: Produžiti mrežni prefiks za 3 bita (2^3)

ISP-ov blok	11001000 00010111 00010000	00000000	200.23.16.0/20
Organizacija 0	11001000 00010111 0001000**0**	00000000	200.23.16.0/23
Organizacija 1	11001000 00010111 0001001**0**	00000000	200.23.18.0/23
Organizacija 2	11001000 00010111 0001010**0**	00000000	200.23.20.0/23
...	.....	....	....
Organizacija 7	11001000 00010111 0001111**0**	00000000	200.23.30.0/23

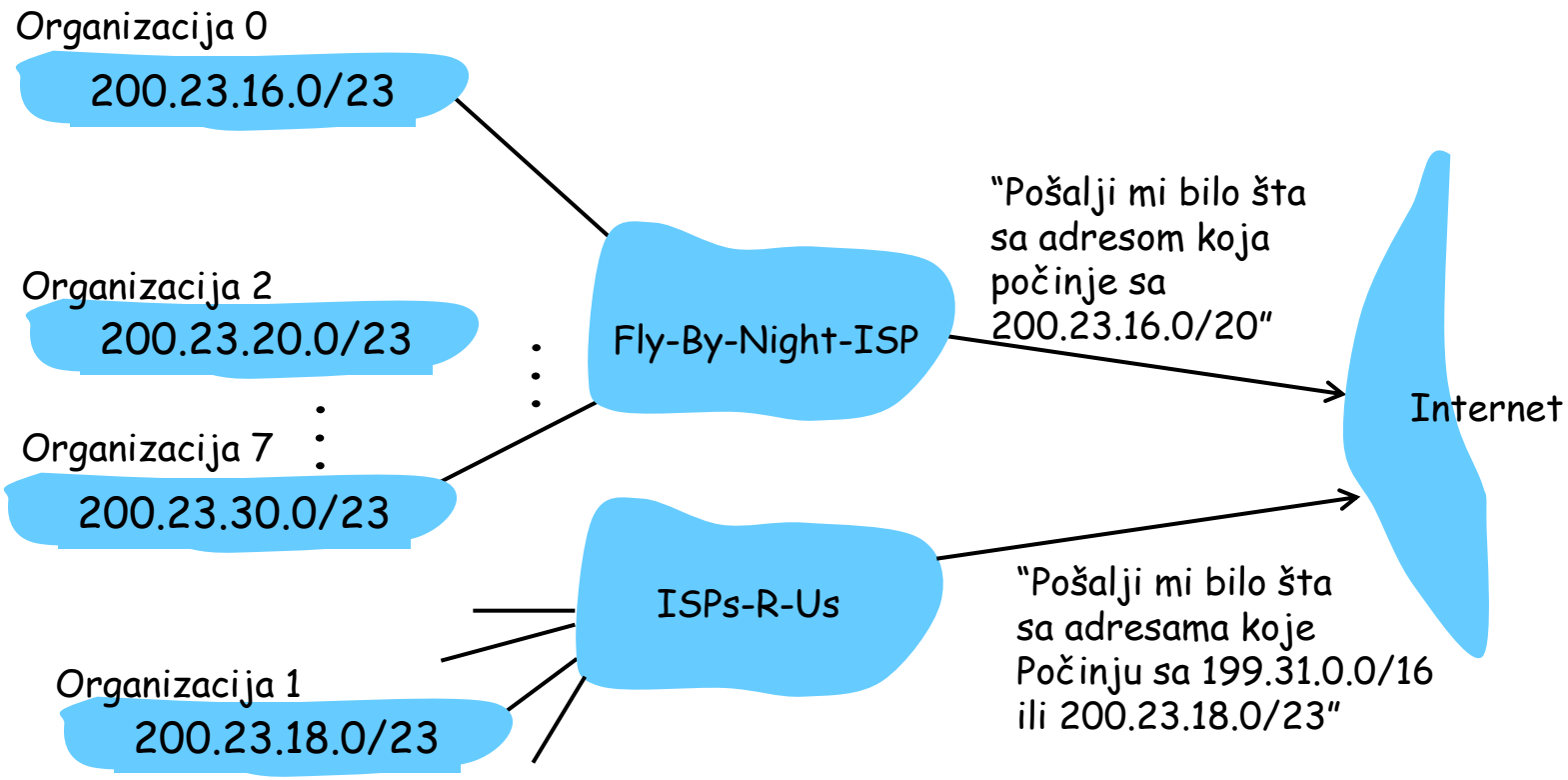
Hijerarhijsko adresiranje: agregacija ruta

Hijerarhijsko adresiranje dozvoljava efikasno oglašavanje informacije potrebne za rutiranje:



Hijerarhijsko adresiranje: specifičnije rute

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



IP adrese: kako dobiti IP adresu?

P: Kako *host* dobija IP adresu?

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