



**FER**

Diplomski studij

Informacijska i komunikacijska  
tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

2.

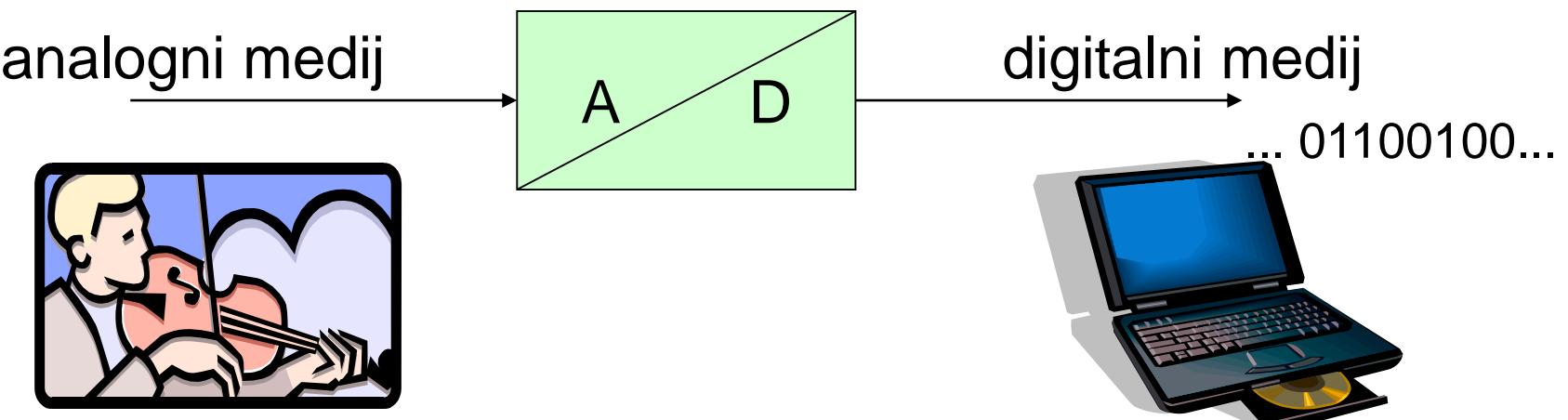
Osnove kodiranja i kompresije

# Osnove kodiranja i kompresije

- ◆ Uvod, definicije
  - Što je višemedijski sadržaj?
  - Definicije, podjela metoda kompresije
  - Hibridno kodiranje
  - Kompresija u sklopu komunikacijskog sustava
- ◆ Entropijsko kodiranje
  - Karakteristike izvora informacije
  - Vrste kodova i njihova svojstva
  - Metode entropijskog kodiranja
    - ◆ Huffmanovo kodiranje
    - ◆ Aritmetičko kodiranje
    - ◆ Metode rječnika (LZ77, LZ78, LZW)
    - ◆ Metode skraćivanja niza (potiskivanje nula, slijedno kodiranje)
- ◆ Izvorno kodiranje
  - Osnovna svojstva i principi
  - Osnovne metode izvornog kodiranja
    - ◆ Kvantizacija
    - ◆ Poduzorkovanje
    - ◆ Transformacijsko kodiranje
    - ◆ Diferencijalno (predikcijsko) kodiranje
    - ◆ Potpojasno kodiranje
    - ◆ Kodiranje zasnovano na modelu

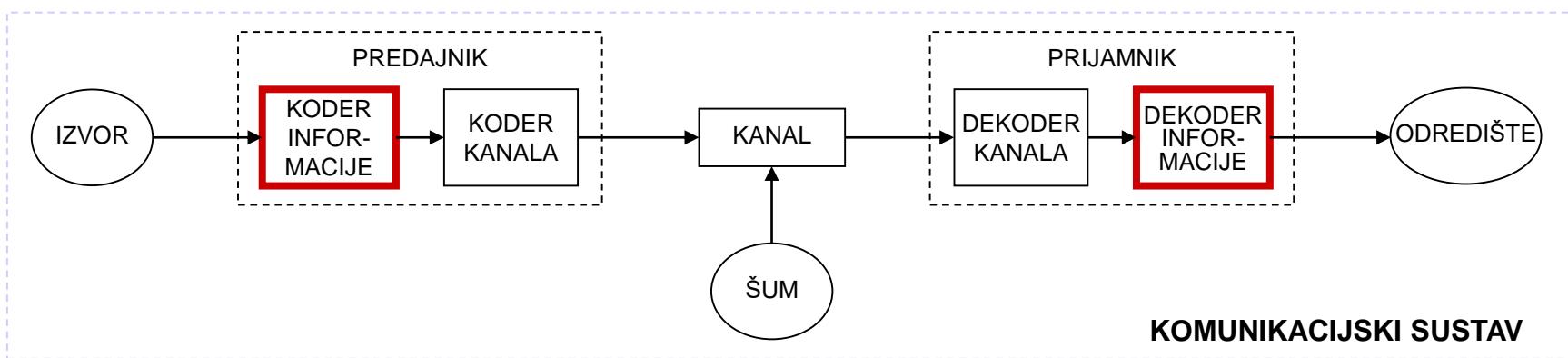
# Što je (više)medijski sadržaj?

- ◆ analogni medij → uzorkovanje → kvantizacija → kodiranje/kompresija → digitalni medij
- ◆ medij se pretvara u digitalni oblik, pogodan za obradu, pohranu ili prijenos



# Kodiranje i kompresija

- ◆ Kodiranje: dodjela kodnih riječi simbolima poruke
- ◆ Kompresija: kodiranje koje smanjuje broj bitova potreban za izražavanje poruke
- ◆ U jasnom kontekstu, koristimo ove pojmove kao sinonime
- ◆ Kompresija se vrši u koderu informacije

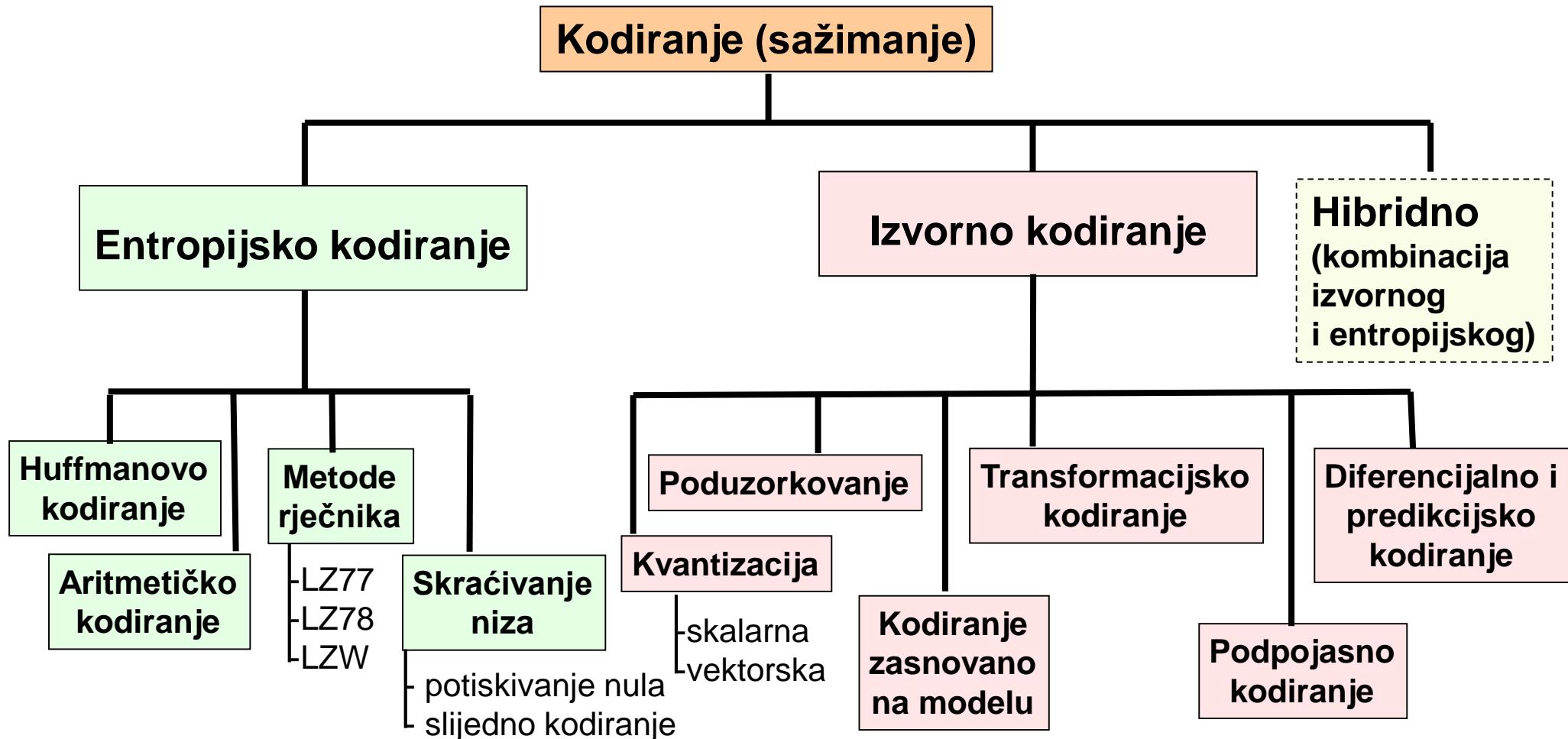


# Zašto je kompresija moguća?

- ◆ Postupci kompresije se uglavnom temelje na redundanciji u podacima:
  - vremenska redundancija (npr. korelacija uzastopnih uzoraka audia)
  - prostorna redundancija (npr. korelacija susjednih elemenata slike - pixela)
  - spektralna (npr. korelacija između boja ili svjetline u slici)
  - percepcijska (ljudski percepcijski sustav; primjena: slika i zvuk)

# Osnovna svojstva kompresije

- ◆ Kompresija **bez gubitaka**
  - Komprimirani podaci mogu se dekomprimiranjem rekonstruirati bez gubitka informacije (*reverzibilno*)
  - Primjene: npr. tekst, medicinske slike, satelitske snimke
- ◆ Kompresija **s gubicima**
  - Cilj je ili dobiti najbolju vjernost rekonstruiranih podataka za zadanu brzinu (bit/s) ili postići najmanju brzinu za zadanu granicu vjernosti
  - Primjene: npr. govor, slika, video
- ◆ Važan parametar je **omjer kompresije**
  - Omjer veličine komprimiranih i originalnih podataka, npr. 1:10



# Hibridno kodiranje

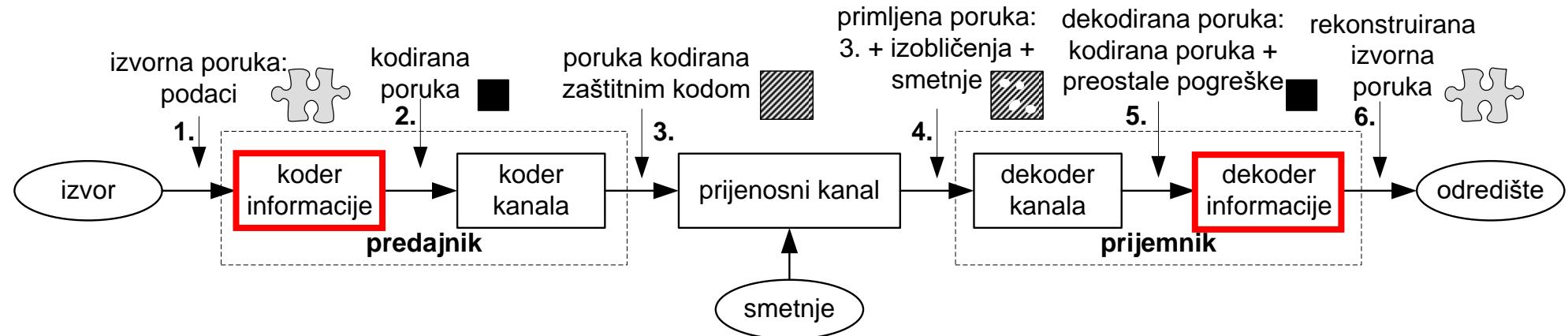
- ◆ Kombinacija izvornog i entropijskog kodiranja
- ◆ Primjene na razne vrste medija (slika, video, glazba...)
- ◆ Različite metode, uglavnom s gubicima (ljudima manje ili više neprimjetnima)
- ◆ Omjer kompresije ovisan o sadržaju i (subjektivnoj) kvaliteti

# Primjeri primjene hibridnog kodiranja

- ◆ Zvuk
  - PCM, ADPCM, MPEG audio,...
- ◆ Nepomična slika
  - GIF, JPEG, JPEG 2000,...
- ◆ Video
  - H.261, MPEG video...
- ◆ 3D modeli i animacija
  - MPEG-4

Odabране metode hibridnog kodiranja zvuka i slike obrađene su u predavanjima koja slijede.

# Kompresija u sklopu komunikacijskog sustava



# Entropijsko kodiranje

# Entropijsko kodiranje

- ◆ Uvod u entropijsko kodiranje
- ◆ Karakteristike izvora informacije
  - Stacionarni izvor, ergodički izvor, izvori s memorijom (Markovljevi)
- ◆ Vrste kodova i njihova svojstva
  - Singularni, nesingularni, jednoznačno dekodabilni, prefiksni kodovi
- ◆ Optimalno kodiranje
- ◆ Metode entropijskog kodiranja
  - Huffmanovo kodiranje
  - Aritmetičko kodiranje
  - Metode rječnika (LZ77, LZ78, LZW)
  - Metode skraćivanja niza (potiskivanje nula, slijedno kodiranje)

# Uvod u entropijsko kodiranje

- ◆ Osnovna ideja: skraćeno zapisati višestruko ili često ponavljane simbole ili nizove simbola
- ◆ Zajedničko svim metodama entropijskog kodiranja:
  - temelje se direktno na teoriji informacije
  - kodiranje bez gubitaka
  - omjer kompresije ovisi samo o statističkim svojstvima izvora informacije
  - poruka se promatra isključivo kao niz niz slučajnih vrijednosti, ne uzimaju se u obzir svojstva medija (za razliku od izvornog kodiranja)

# Karakteristike izvora informacije

- ◆ Izvor informacije promatramo kao stohastički proces, tj. niz slučajnih varijabli:

$$X_1, X_2, \dots, X_n$$

- ◆ Izvor u potpunosti opisan raspodjelom zdrženih vjerojatnosti pojavljivanja varijabli:

$$P\{(X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)\} = p(x_1, x_2, \dots, x_n)$$

- ◆ Općenito, moguća zavisnost među varijablama

# Stacionarni izvor

- ◆ Statistička svojstva se ne mijenjaju s vremenom

$$P\{(X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)\} = P\{(X_{1+l}, X_{2+l}, \dots, X_{n+l}) = (x_1, x_2, \dots, x_n)\},$$
$$\forall l, (x_1, x_2, \dots, x_n) \in X^n, n > 0$$

- ◆ Trivijalan primjer stacionarnog izvora:  
AEAEAEAEAEAEAEAE.....
- ◆ Trivijalan primjer nestacionarnog izvora:  
AEAAEEAAAEEEEAAAAEEEAAAAAEEEEEE...

# Ergodički izvor

- ◆ Izvor kao skup svih mogućih proizvedenih nizova
  - Prosjek po skupu: prosjek pojavljivanja simbola na nekom mjestu u nizu, gledano među svim nizovima
  - Prosjek po vremenu: učestalost pojavljivanja simbola unutar pojedinog niza
- ◆ Ergodičnost: prosjek po skupu = prosjek po vremenu
- ◆ Svaki proizvedeni niz ima ista svojstva i ona se ne mijenjaju u vremenu
- ◆ Za entropijsko kodiranje promatramo ergodičke izvore (aproksimacija stvarnih izvora)

# Ergodičnost izvora - primjer

- ◆ Izvor počinje  $1/3$  sa A,  $1/3$  B i  $1/3$  E
  - Ako počne sa A ili B ponavlja ih izmjenično
  - Ako počne sa E, ponavlja samo E
  - Skup mogućih nizova:

Niz 1: ABABABABABABAB...

Niz 2: BABABABABABABA...

Niz 3: EEEEEEEEEE...

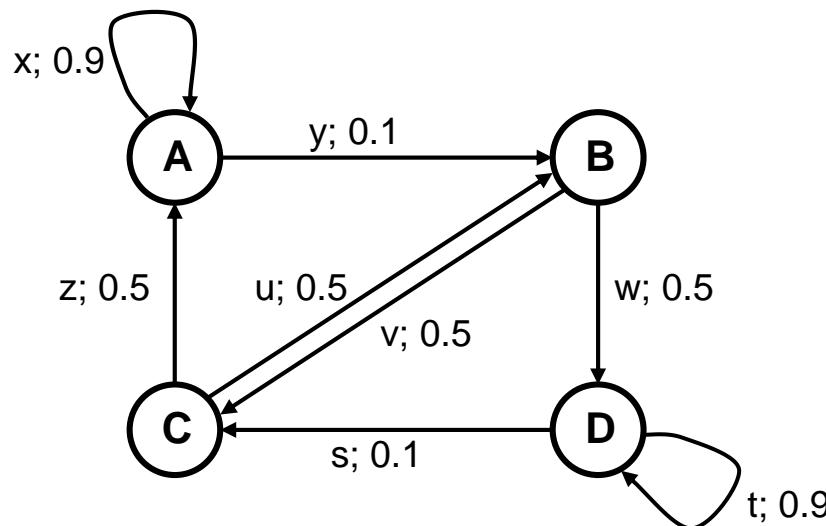
Simbol	Prosjek po vremenu za niz 1	Prosjek po vremenu za niz 2	Prosjek po vremenu za niz 3	Prosjek po skupu
A	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{3}$
B	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{3}$
E	0	0	1	$\frac{1}{3}$

# Izvori s memorijom

- ◆ Vjerojatnost pojavljivanja simbola je ovisna o jednom ili više prethodnih simbola
- ◆ Neki nizovi simbola vjerojatniji od drugih
- ◆ Većina prirodnih izvora su izvori s memorijom
  - Npr. iz slova u tekstu, zvuk govora, slika

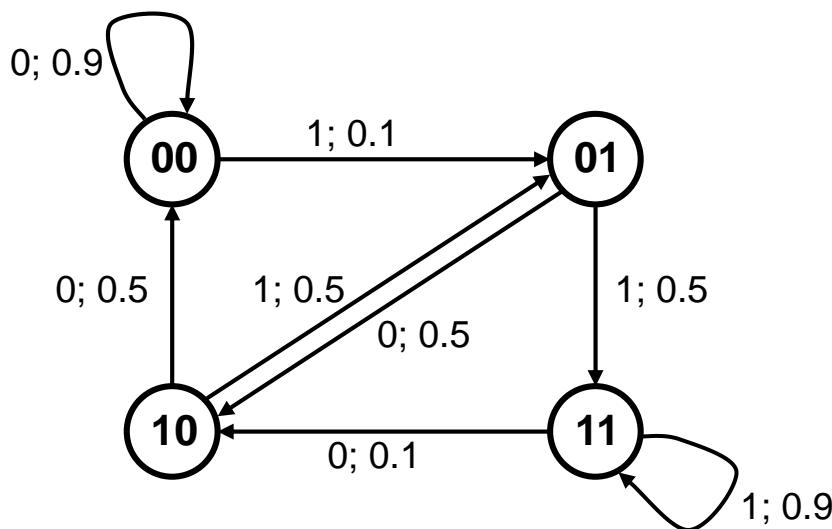
# Markovljevi informacijski izvori

- ◆ Izvori s memorijom često se mogu opisati pomoću Markovljevih
- ◆ Stanja, vjerojatnosti prijelaza
- ◆ Pri prijelazu stanja generira se simbol



# Primjer Markovljevog izvora

- ◆ Binarni Markovljev izvor s memorijom od dva simbola



- ◆ Tipičan izlaz:

0000000000000000111111111111111100001111111111000001111111111111...

# Kodiranje

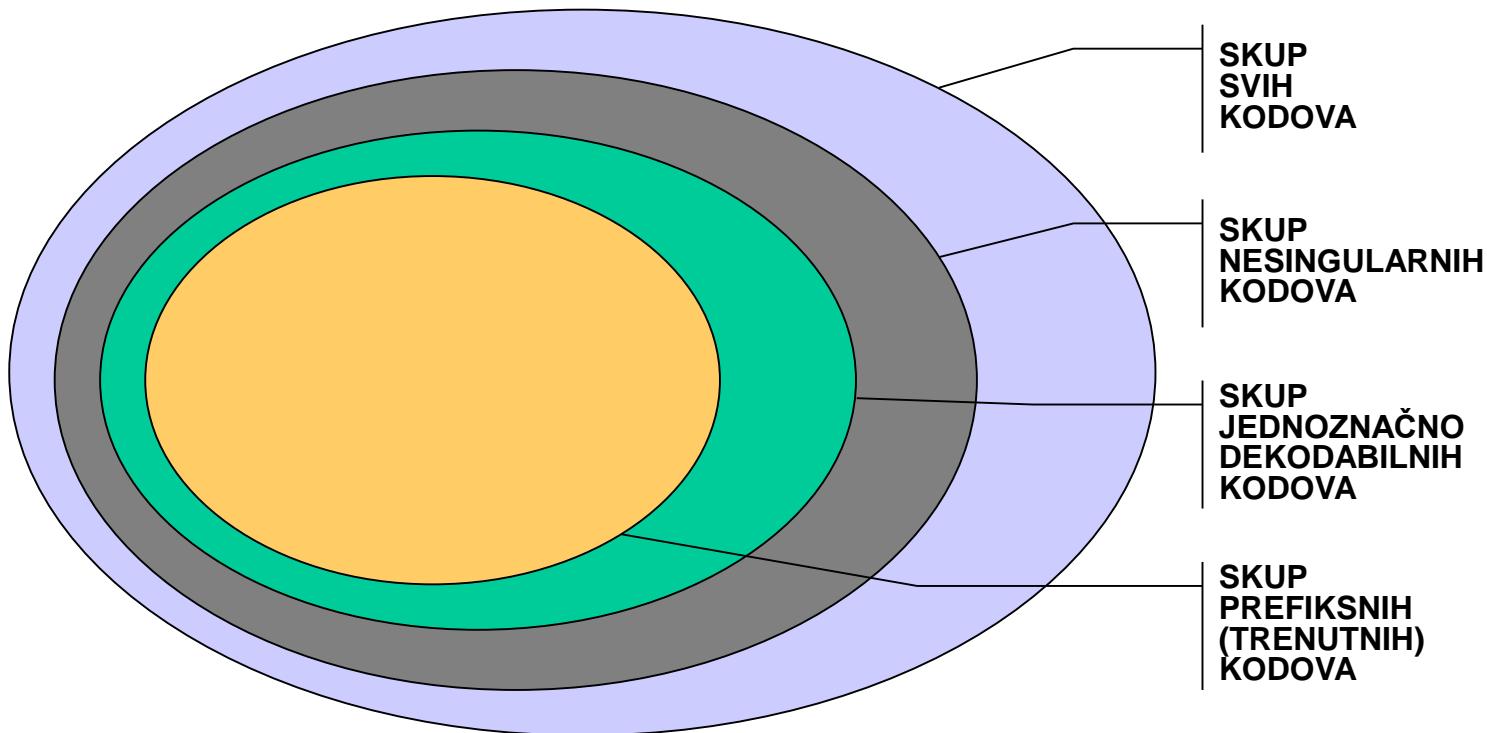
- ◆ Dodjela kodnih riječi simbolima poruke

$$X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$$

$$x_i \in X \xrightarrow{\text{KODIRANJE}} C(x_i)$$

$$C(x_i) \in D^*, D = \{a_1, a_2, \dots, a_d\},$$

- ◆ Kodiranje sa svojstvom sažimanja: kompresija
- ◆ U praksi gotovo uvijek binarna abeceda
  - $d = 2, D = \{0, 1\}$
  - Izlaz kodera: struja bitova (engl. *bitstream*)



# Optimalni kodovi

- ◆ Minimum prosječne duljine kodne riječi se dobiva za:

$$l_i^* = -\log_d p_i \Rightarrow L = -\sum_{i=1}^n p_i \log_d p_i = H(X)$$

- ◆ Ali  $l_i$  moraju biti cijeli brojevi, pa se ne može uvijek postići  $L=H$ :

$$L \geq H(X)$$

- ◆ Za optimalni kod, prosječna duljina kodne riječi je unutar jednog bita od entropije:  $H(X) \leq L < H(X) + 1$

- ◆ Efikasnost koda:

$$\varepsilon = \frac{H(X)}{L}$$

# Metode entropijskog kodiranja

- ◆ **Huffmanovo kodiranje**
  - optimalno kodiranje
  - binarno stablo
  - kraći zapis čestih znakova
- ◆ **Aritmetičko kodiranje**
  - poopćenje Huffmanovog kodiranja
  - cijela poruka se pretvara u jednu kodnu riječ
- ◆ **Metode rječnika**
  - isti rječnik kodnih riječi na strani pošiljatelja i primatelja
  - dinamička konstrukcija rječnika
  - Lempel-Ziv (LZ77, LZ78), Lempel-Ziv-Welch (LZW)
- ◆ **Metode skraćivanja niza**
  - potiskivanje nula, slijedno kodiranje

# Huffmanovo kodiranje

- ◆ D. A. Huffman, 1952. godine
- ◆ Kodira pojedinačne simbole kodnim riječima promjenjive duljine, ovisno o (poznatim!) vjerojatnostima njihova pojavljivanja
- ◆ Temelji se na dvije jednostavne činjenice:
  - (1) U optimalnom kodu, simboli s većom vjerojatnošću pojavljivanja imaju kraće kodne riječi od onih s manjom vjerojatnošću
  - (2) U optimalnom kodu, dva simbola s najmanjim vjerojatnostima imaju kodne riječi jednakog duljina (vrijedi za prefiksni kod)
- ◆ Ishod: sažetiji zapis (npr. tipičan tekst se sažima za 45%)

# Huffmanovo kodiranje: postupak

## ◆ Algoritam stvaranja koda:

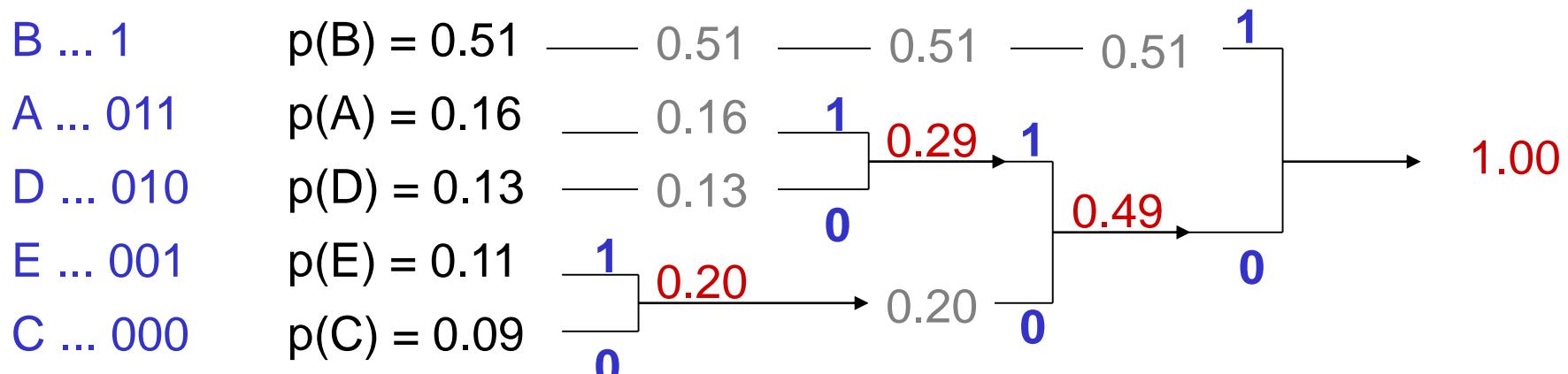
1. Sortiraj simbole po padajućim vjerojatnostima
2. Pronađi dva simbola s najmanjim vjerojatnostima
3. Jednom od njih dodijeli simbol "0", drugom "1"
4. Kombiniraj ta dva simbola u jedan nadsimbol (nadsimbol je novi simbol čija je vjerojatnost pojavljivanja jednaka zbroju vjerojatnosti pojavljivanja dvaju simbola od kojih je nastao) i zapiši ih kao dvije grane binarnog stabla, a nadsimbol kao račvanje iznad njih
5. Ponavljam 1-4 dok ne dobiješ samo jedan nadsimbol
6. Povratkom kroz stablo očitaj kodove

## ◆ Podatkovna struktura algoritma je binarno stablo

- ## ◆ Algoritam dekodiranja koristi isti postupak za gradnju stabla
- Dekoder mora znati vjerojatnosti pojavljivanja simbola

# Huffmanovo kodiranje: primjer

- ◆ Skup simbola  $\{A, B, C, D, E\}$  s vjerojatnostima pojavljivanja  $p(A) = 0.16, p(B) = 0.51, p(C) = 0.09, p(D) = 0.13, p(E) = 0.11$
- ◆ Za uniformni kod, prosječna duljina koda je **3 bit/simbol** (jer je  $2^2 \leq 5 \leq 2^3$ ).
- ◆ Entropija: **1.96 bit/simbol**



- ◆ Prosječna duljina dobivenog koda u našem slučaju je:

$$L = \sum_{x \in X} p_x l_x = 3 \times (0.09 + 0.11 + 0.13 + 0.51) + 0.09 = 1.98 \text{ bit/simbol}$$

# Huffmanovo kodiranje: svojstva

- ◆ kodiranje je idealno ako su vjerojatnosti  $1/2, 1/4, \dots, 1/2^n$
- ◆ u stvarnim slučajevima to obično nije slučaj, te rezultat ovisi o vjerojatnostima pojavljivanja simbola
- ◆ prednosti:
  - jednostavan za izvedbu
  - vrlo dobro kodiranje za „dobre“ vjerojatnosti pojavljivanja simbola
- ◆ nedostaci:
  - vjerojatnosti pojavljivanja simbola moraju biti poznate; ovise o primjeni (tekst, slika)
  - za “loše raspoređene” vjerojatnosti pojavljivanja dobiju se izrazito loši kodovi

# Primjer lošeg koda i prošireni Huffmanov kod

Simbol	Vjerojatnost	Kodna riječ
$a_1$	0.95	0
$a_2$	0.02	10
$a_3$	0.03	11

PROSIRENI KOD		
Simbol	Vjerojatnost	Kodna riječ
$a_1a_1$	0.9025	0
$a_1a_2$	0.0190	111
$a_1a_3$	0.0285	100
$a_2a_1$	0.0190	1101
$a_2a_2$	0.0004	110011
$a_2a_3$	0.0006	110001
$a_3a_1$	0.0285	101
$a_3a_2$	0.0006	110010
$a_3a_3$	0.0009	110000

- ◆ Entropija: 0.335 bit/simbol
- ◆ Prosječna duljina: 1.05 bit/simbol: **213% više od entropije!!**
- ◆ Prošireni kod:  $1.222 / 2 = 0.611$  bit/simbol: 72% više od entropije.
- ◆ Bolje je kodirati duže sekvence, ali tada broj kodnih riječi raste eksponencijalno

# Huffmanovo kodiranje: primjene

- ◆ Česta primjena unutar složenijih algoritama
- ◆ Primjeri:
  - standardi za telefaks (T.4, T.6)
  - standard za nepomičnu sliku JPEG

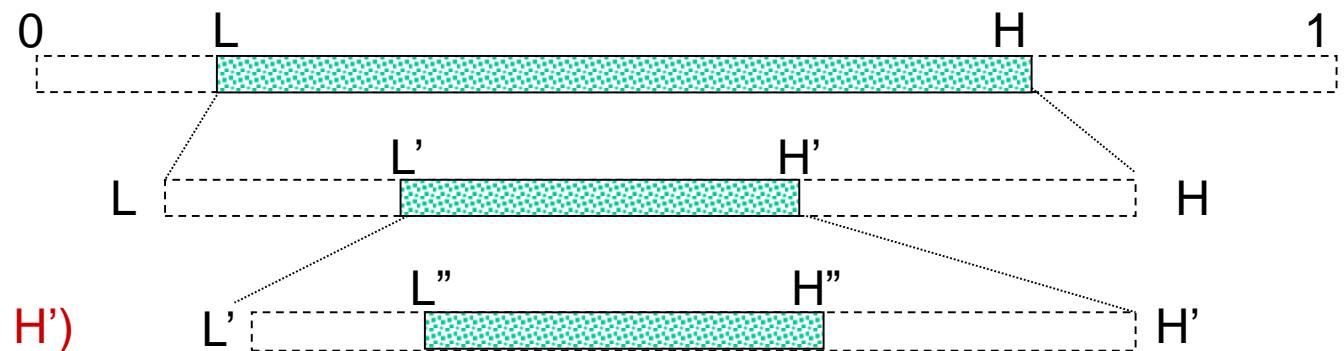
# Aritmetičko kodiranje

- ◆ Autori Pasco & Rissanen (nezavisno), 1976. godine
- ◆ Algoritam uzima kao ulaz cijele nizove simbola (“poruke”) i preslikava ih na realne brojeve, ovisno o (poznatim!) statističkim svojstvima

# Aritmetičko kodiranje: postupak

1. Podijeli interval  $[0, 1)$  u  $n$  podintervala koji odgovaraju simbolima iz abecede; duljina svakog podintervala proporcionalna vjerojatnosti odgovarajućeg simbola
2. Iz promatranih skupa podintervala, odaberi podinterval koji odgovara sljedećem simbolu u poruci
3. Podijeli taj podinterval u  $n$  novih podintervala, proporcionalno vjerojatnostima pojavljivanja simbola iz abecede; tako nastaje novi skup podintervala koji promatramo
4. Ponavljam korake 2 i 3 dok cijela poruka nije kodirana
5. Konačni kod za čitavu poruku je jedan broj iz intervala u binarnom obliku

početni interval  $[0, 1)$

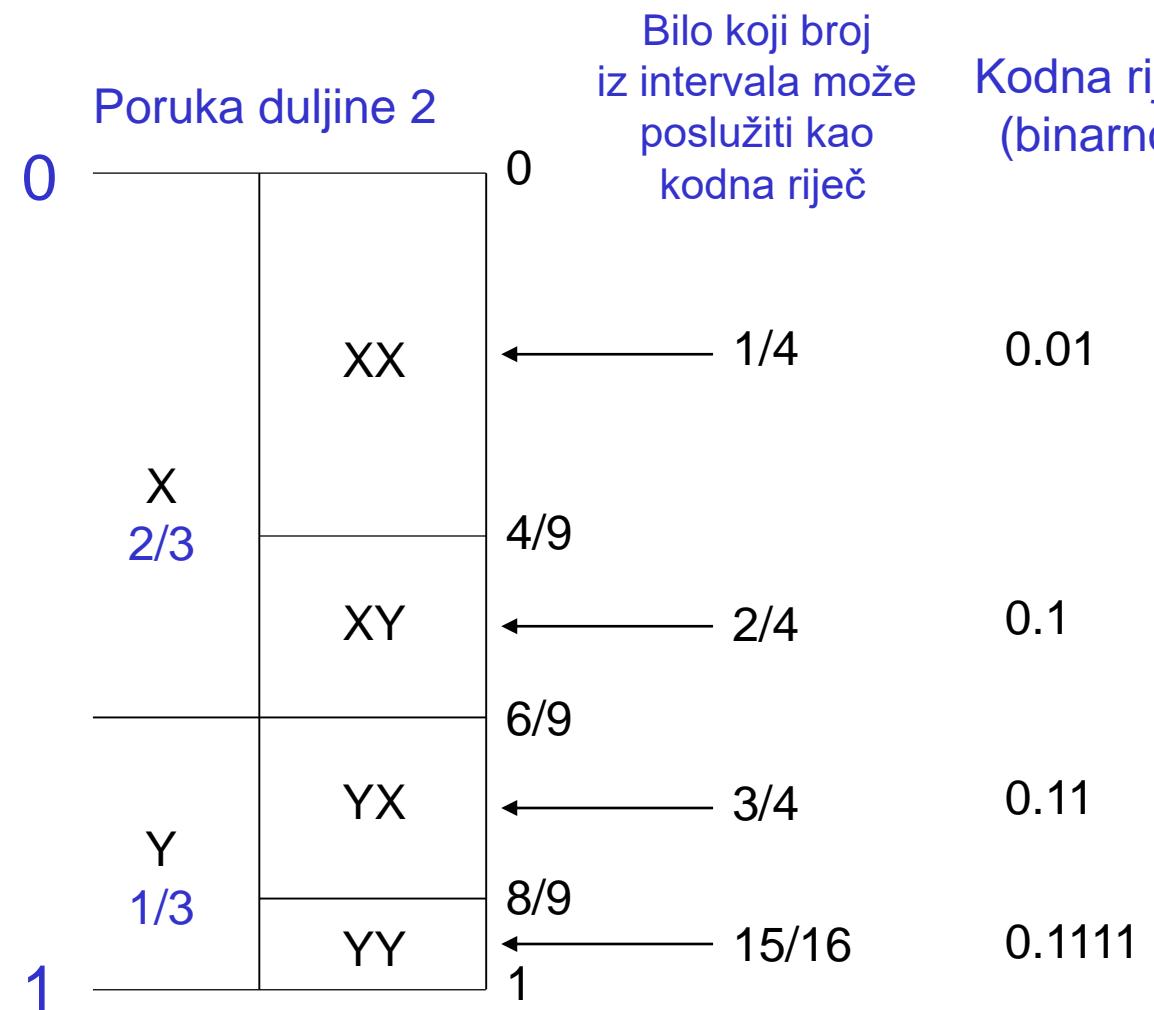


podinterval  $[L, H)$

novi podinterval  $[L', H')$

# Aritmetičko kodiranje: primjer (1)

- $M=2$
- simboli: X, Y
- $p(X) = 2/3$
- $p(Y) = 1/3$
- poruka duljine 2 (moguće poruke XX, XY, YX, YY) kodira se onim brojem bita dovoljnim za jedinstveno određivanje intervala (binarni razlomak!)



# Aritmetičko kodiranje: primjer (2)

Kodna riječ  
(binarno)

0.01

- primjer za poruku duljine 3

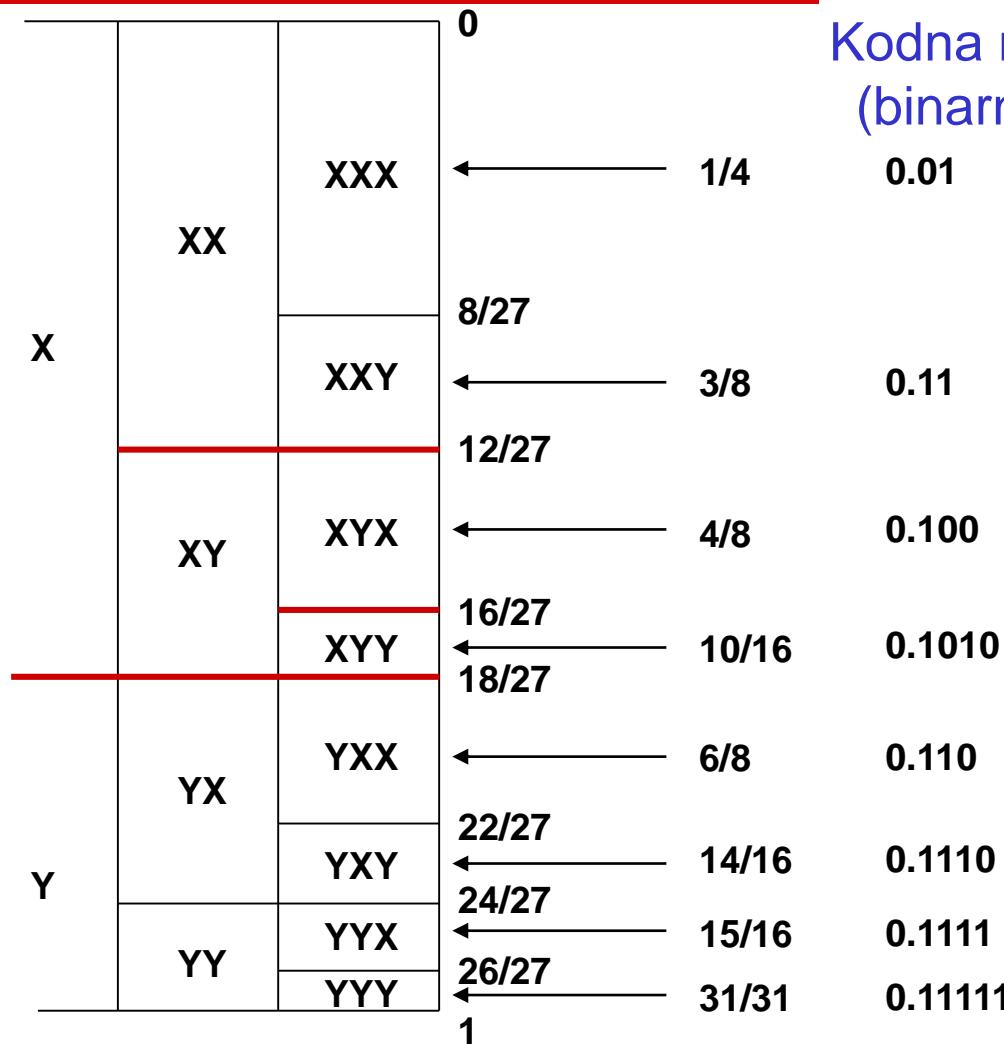
- $M=2$

- simboli:

$X, Y$

$$p(X) = 2/3$$

$$p(Y) = 1/3$$



# Postupak dekodiranja

1. Podijeli početni interval  $[0, 1)$  u podintervale po vjerojatnostima pojavljivanja simbola
2. Uzmi primljeni kod kao realni broj
3. Pronađi podinterval u kojem se nalazi broj (kod)
4. Zapiši simbol koji odgovara tom podintervalu
5. Podijeli taj podinterval u n novih podintervala, proporcionalno vjerojatnostima pojavljivanja simbola iz abecede; tako nastaje novi skup podintervala koji promatramo
6. Ponavljam korake 3-5 dok ne dođe kraj poruke

# Dekodiranje: primjer

- primjer za poruku duljine 3

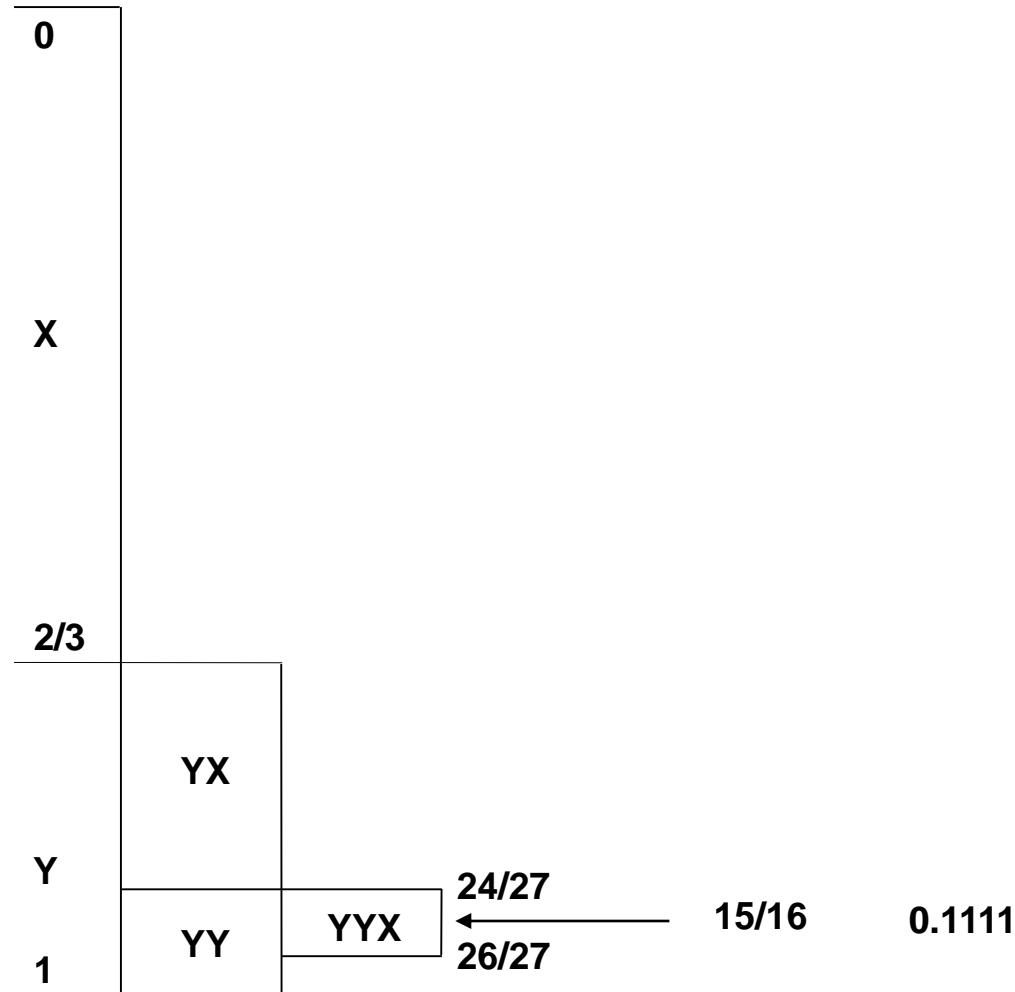
- $M=2$
- simboli:

X, Y

$$p(X) = 2/3$$

$$p(Y) = 1/3$$

- Primljeni kod 1111  
tj. 15/16



# Odabir koda

- ◆ Kojim brojem iz podintervala kodirati poruku?
- ◆ Može se uzeti bilo koja vrijednost iz podintervala
- ◆ Dovoljan broj znamenki:

$$l(x) = \left\lceil \log \frac{1}{P(x)} \right\rceil + 1 \text{ [bit]}$$

- ◆ Na ovakav način dobiva se uvijek prefiksni kod

# Implementacija

---

- ◆ Do sada opisani algoritam neupotrebljiv
  - Neprihvatljivo čekanje do kraja poruke
  - Algoritam podrazumijeva beskonačnu preciznost realnih brojeva – na računalu prikaz s pomičnim zarezom
  - Operacije s realnim brojevima su skupe
- ◆ Potreban je algoritam koji:
  - Koristi operacije sa cijelim brojevima
  - Koristi prikaz sa fiksnim brojem bitova
  - Proizvodi simbole koda tokom postupka kodiranja, a ne na kraju

# Aritmetičko kodiranje: praktičan postupak

- ◆ Osnovni postupak podjele na podintervale je isti
- ◆ Koristi se fiksni broj znamenki za prikaz intervala
- ◆ Kada je prva znamenka u prikazu gornje i donje granice ista, interval se *renormalizira*:
  - Prvih n znamenki se šalje na izlaz kodera
  - Znamenke se pomiću ulijevo za jedno mjesto
  - Desno se dodaje znamenka: 0 na donju, 1 na gornju granicu intervala (ako su znamenke binarne)

# Renormalizacija: primjer

Zavod za telekomunikacije

x	p(x)
RAZMAK	1/10
A	1/10
B	1/10
E	1/10
G	1/10
I	1/10
L	2/10
S	1/10
T	1/10

	GORNJA GRANICA	DONJA GRANICA	DULJINA INTERVALA	KUMULATIVNI IZLAZ
Početno stanje	99999	00000	100000	
Kodiraj B (0.2-0.3)	29999	20000		
Renormalizacija, izlaz: 2	99999	00000	100000	.2
Kodiraj I (0.5-0.6)	59999	50000		.2
Renormalizacija, izlaz: 5	99999	00000	100000	.25
Kodiraj L (0.6-0.8)	79999	60000	20000	.25
Kodiraj L (0.6-0.8)	75999	72000		.25
Renormalizacija, izlaz: 7	59999	20000	40000	.257
Kodiraj RAZMAK (0.0-0.1)	23999	20000		.257
Renormalizacija, izlaz: 2	39999	00000	40000	.2572
Kodiraj G (0.4-0.5)	19999	16000		.2572
Renormalizacija, izlaz: 1	99999	60000	40000	.25721
Kodiraj A (0.1-0.2)	67999	64000		.25721
Renormalizacija, izlaz: 6	79999	40000	40000	.257216
Kodiraj T (0.9-1.0)	79999	76000		.257216
Renormalizacija, izlaz: 7	99999	60000	40000	.2572167
Kodiraj E (0.3-0.4)	75999	72000		.2572167
Renormalizacija, izlaz: 7	59999	20000	40000	.25721677
Kodiraj S (0.8-0.9)	55999	52000		.25721677
Renormalizacija, izlaz: 5	59999	20000		.257216775
Renormalizacija, izlaz: 2				.2572167752
Renormalizacija, izlaz: 0				.25721677520

# Usporedba aritmetičko - Huffman

Huffman	Aritmetičko kodiranje
Kodira svaki simbol posebno	Kodira cijelu poruku jednim kodom: realni broj 0 - 1
Minimalno 1 bit/simbol	Moguće < 1 bit/simbol
Duljina poruke nije važna	Teoretski optimalno za dugačke poruke
Kodiranje niza simbola moguće samo proširenim Huffman kodom	Uvijek se kodira cijela poruka
Jednostavno za računanje	Zahtjevnije za računanje

# Aritmetičko kodiranje: primjene

- ◆ Primjena kao komponente u raznim standardima i za razne vrste medija
- ◆ Dokumenti
  - JBIG (Joint Bi-level Image Processing Group)
- ◆ Slika
  - JPEG
- ◆ Sintetički sadržaji/animacija
  - MPEG-4 FBA (Face and Body Animation)

# Metode rječnika

- ◆ Algoritmi kodiranja metodama rječnika uzimaju kao ulaz nizove simbola ("riječi") promjenjive duljine i kodiraju ih kodnim riječima stalne duljine iz rječnika
- ◆ Ne trebaju znati vjerojatnosti pojavljivanja simbola, nazivaju se i *univerzalni koderi*
- ◆ Koder i dekoder moraju imati isti rječnik
- ◆ Rječnik može biti statičan, no najčešće je prilagodljiv

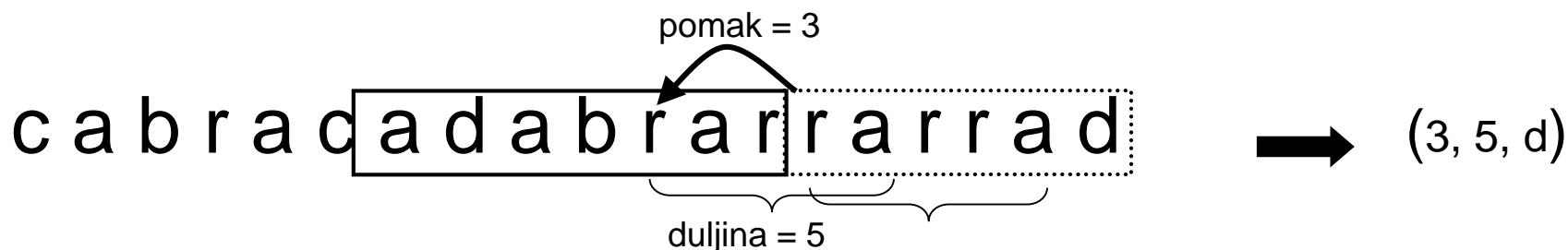
# Metode s prilagodljivim rječnikom

- ◆ Koder i dekoder dinamički grade rječnik
  - LZ77: Rječnik je posmični prozor
  - LZ78: riječi se grade dodavanjem slova na postojeće riječi (u početku rječnik je prazan)
  - Lempel-Ziv-Welch (LZW) algoritam
    - izvorni algoritam smislili Ziv i Lempel (1977 - LZ77, 1978 - LZ78), a Welch ga je doradio i poboljšao 1984 (zato **LZW**)
    - algoritam relativno jednostavan, iako složeniji od Huffmanovog
    - izvorni LZW algoritam koristi rječnik s 4K riječi, s tim da su prvih 256 riječi standardni ASCII kodovi

# Algoritam LZ77

- ◆ Rječnik je posmični prozor od  $N$  zadnjih simbola
- ◆ U svakom koraku traži se u rječniku najduži niz simbola jednak nadolazećim simbolima, te se kodira kao uređena trojka (*pomak, duljina, sljedeći\_simbol*)
- ◆ Nedostatak: “kratka” memorija

# LZ77: primjer kodiranja



# Algoritmi LZ78 i LZW

- ◆ Umjesto posmičnog prozora, zasebna memorija za rječnik
  - Rječnik je poredana lista riječi (nizova simbola)
  - Riječ se dovaća pomoću indeksa (rednog broja)
- ◆ LZ78
  - Rječnik u početku prazan
  - U svakom koraku šalje se (*indeks, idući simbol*)
    - Indeks pokazuje na najdulju riječ u rječniku jednaku nadolazećem nizu simbola
    - Rječnik se nadopunjava novim rijećima tijekom kodiranja

# LZW algoritam

- Algoritam kodiranja:

1. RadnaRiječ = slijedeći simbol sa ulaza
2. WHILE (ima još simbola na ulazu) DO
3. NoviSimbol = slijedeći simbol sa ulaza
4. IF RadnaRiječ+NoviSimbol postoji u rječniku THEN
5. RadnaRiječ = RadnaRiječ+NoviSimbol
6. ELSE
7. IZLAZ: kod za RadnaRiječ
8. dodaj RadnaRiječ+NoviSimbol u rječnik
9. RadnaRiječ = NoviSimbol
10. END IF
11. END WHILE
12. IZLAZ: kod za RadnaRiječ

# Kodiranje algoritmom LZW: primjer

Sadržaj rječnika na početku:

kodna riječ	znak
(1)	A
(2)	B
(3)	C

Niz znakova koje treba kodirati:

Mjesto	1	2	3	4	5	6	7	8	9
Simbol	A	B	B	A	B	A	B	A	C

LZW:

korak	mjesto	sadržaj rječnika	izlaz iz kodera
1.	1	(4) A B	(1)
2.	2	(5) B B	(2)
3.	3	(6) B A	(2)
4.	4	(7) A B A	(4)
5.	6	(8) A B A C	(7)
6.	9		(3)

# LZW kodiranje: primjer dekodiranja

Zavod za telekomunikacije

KORAK	RADNA RIJEČ	ULAZ DEKODERA	DEKODIRANI SIMBOLI	SADRŽAJ RJEČNIKA
1		(1)	A	
2	A	(2)	B	(4) AB
3	B	(2)	B	(5) BB
4	B	(4)	AB	(6) BA
5	AB	(7)	ABA	(7) ABA
6	ABA	(3)	C	

# Metode rječnika: primjene

- ◆ LZW
  - UNIX compress
  - GIF
  - Modem V.24 bis
- ◆ LZ77
  - ZIP

## Metode skraćivanja niza



ABCCCCCCCC  
8 okteta

DEFFFABC...  
3 okteta

**ABC!8**      **DEFFFABC...**  
3 okteta      3 okteta

← “isplati” se za 4+ znakova



- ◆ Primjena: prva generacija telefaksa, unutar JPEG-a

# Izvorno kodiranje

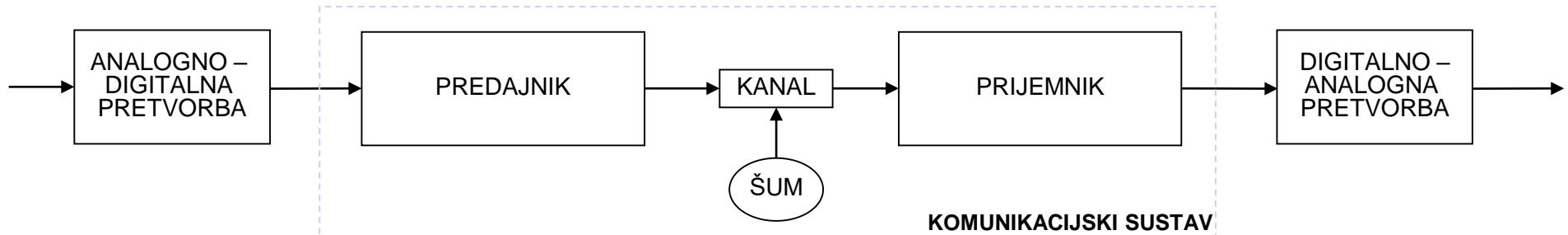
# Izvorno kodiranje

- ◆ Uvod
  - Svojstva metoda izvornog kodiranja
  - Analogni mediji u diskretnom kom. sustavu
  - Principi kompresije pri izvornom kodiranju
- ◆ Osnovne metode izvornog kodiranja
  - Kvantizacija
  - Poduzorkovanje
  - Transformacijsko kodiranje
  - Diferencijalno (predikcijsko) kodiranje
  - Potpojasno kodiranje
  - Kodiranje zasnovano na modelu

# Svojstva metoda izvornog kodiranja

- ◆ Najčešće sažimaju s gubicima
- ◆ Koriste semantiku izvora, tj. posebna svojstva pojedinih medija
- ◆ Koriste karakteristike ljudske percepcije medija za bolju kompresiju uz malo primjetnu pogrešku
- ◆ Omjer kompresije jako ovisan o sadržaju
- ◆ Koriste se u sklopu hibridnih metoda za kodiranje pojedinih medija
  - Obično prvo izvorno kodiranje, zatim entropijsko

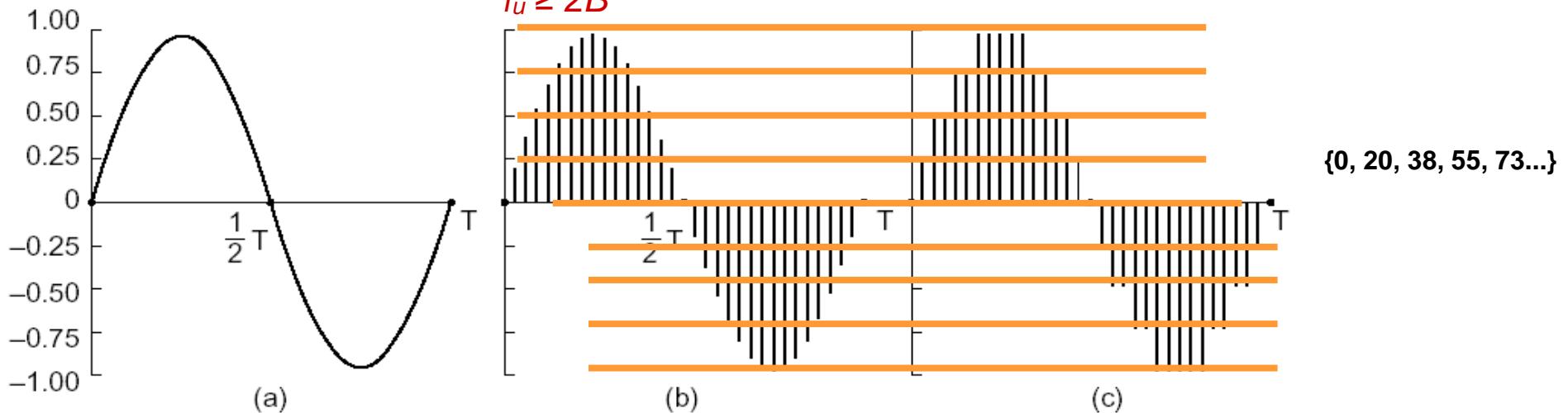
# Analogni mediji u diskretnom komunikacijskom sustavu



*izvorni signal* → uzorkovanje  $f_u$  → kvantizacija → poruka

Nyquistov kriterij:

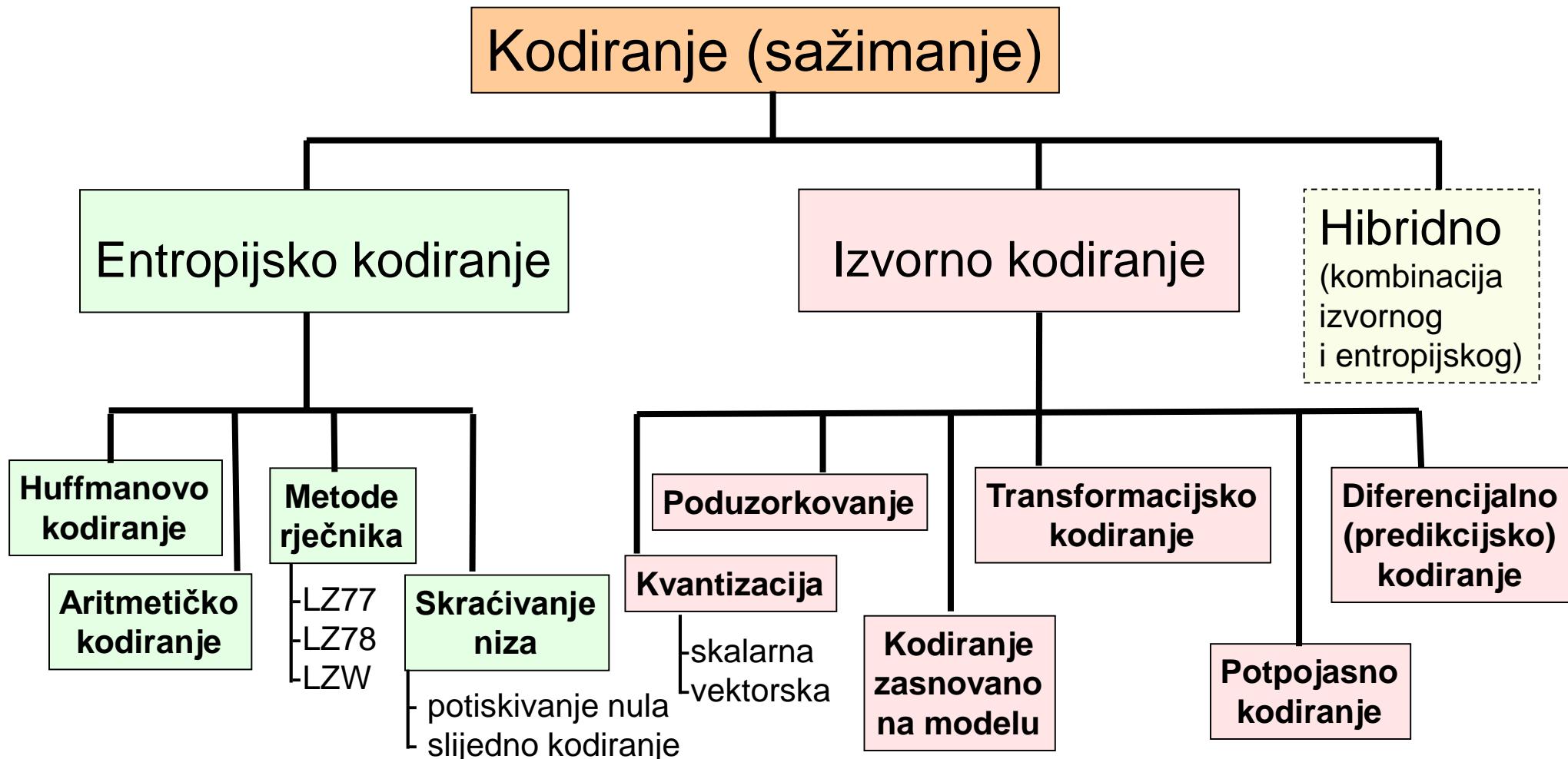
$$f_u \geq 2B$$



# Principi kompresije pri izvornom kodiranju

- ◆ Uklanjanje zalihosti (redundancije):
  - Vremenska zalihost (npr. korelacija uzastopnih uzoraka zvučnog signala)
  - Prostorna zalihost (npr. korelacija susjednih elemenata slike - pixela)
  - Spektralna zalihost (npr. korelacija između boja ili svjetline u slici)
- ◆ Uklanjanje irelevantnosti:
  - Granice ljudske precepcije
  - Spuštenje razine kvalitete reprodukcije

# Osnovne metode izvornog kodiranja

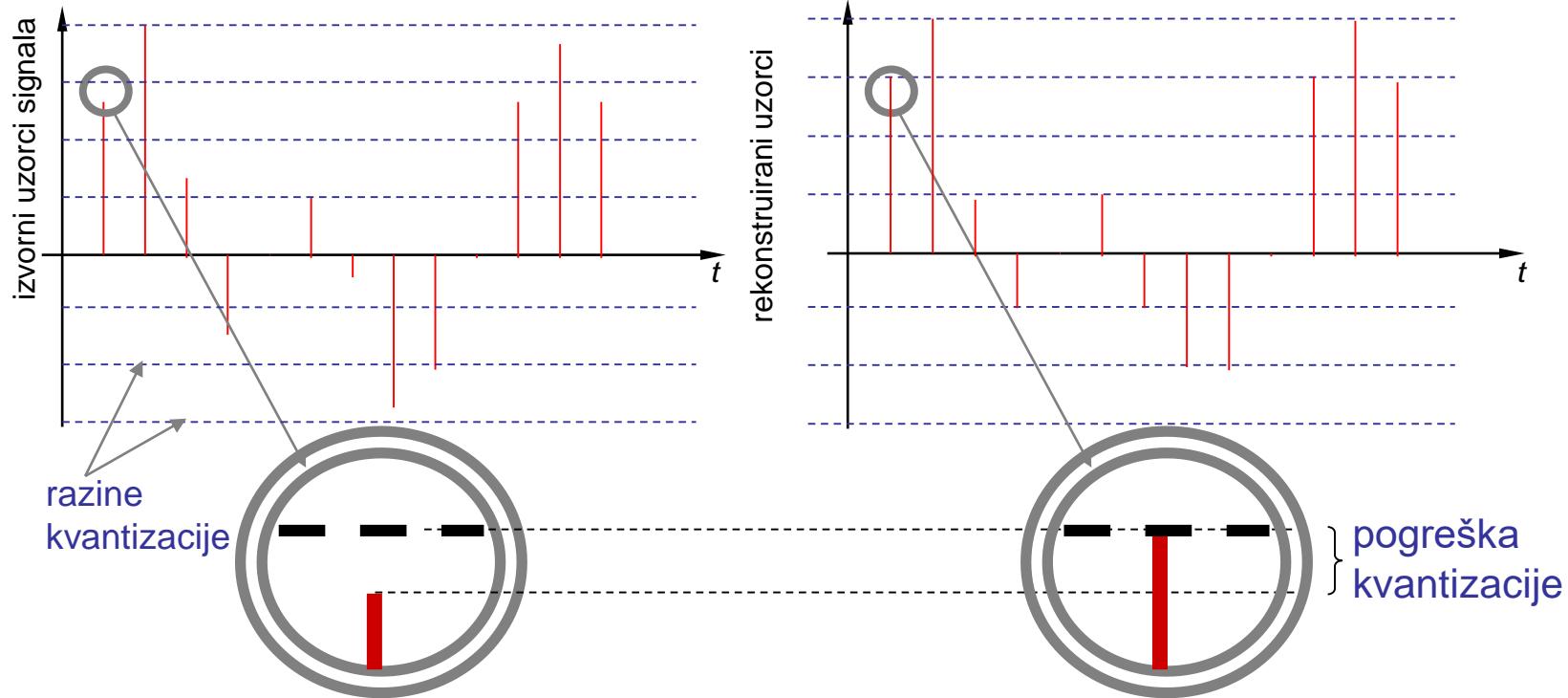


# Kvantizacija

- ◆ Aproksimacija signala konačnim skupom kodova
  - Raspon ulaza podijeljen na inervale
  - Sve vrijednosti u intervalu isti kod – nije reverzibilno!
  - Kod intervala dekodira se kao razina kvantizacije
  - Svaka vrijednost se svodi na jednu razinu kvantizacije
- ◆ A/D pretvorba uvijek uvodi kvantizaciju
- ◆ Izuzetno korisna kao metoda kompresije
  - Primjer: zvuk 16 bit (65536 razina)  $\rightarrow$  8 bit (256 razina)
- ◆ Linearna i nelinearna kvantizacija
- ◆ Skalarna i vektorska kvantizacija

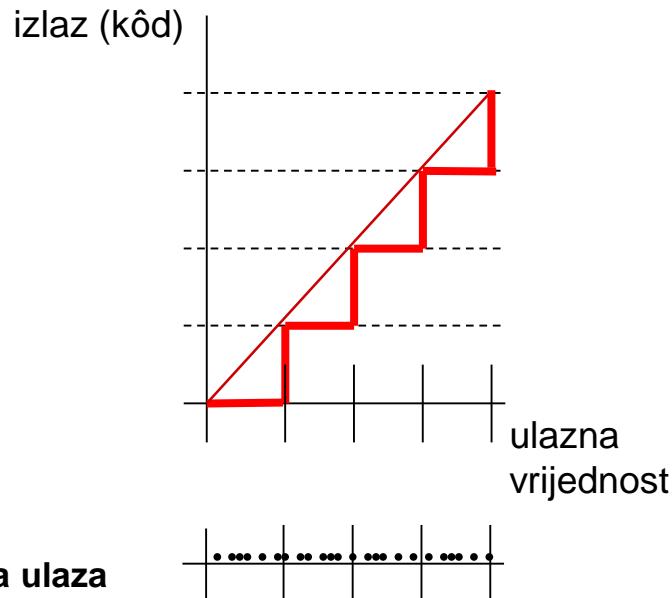


# Pogreška kvantizacije

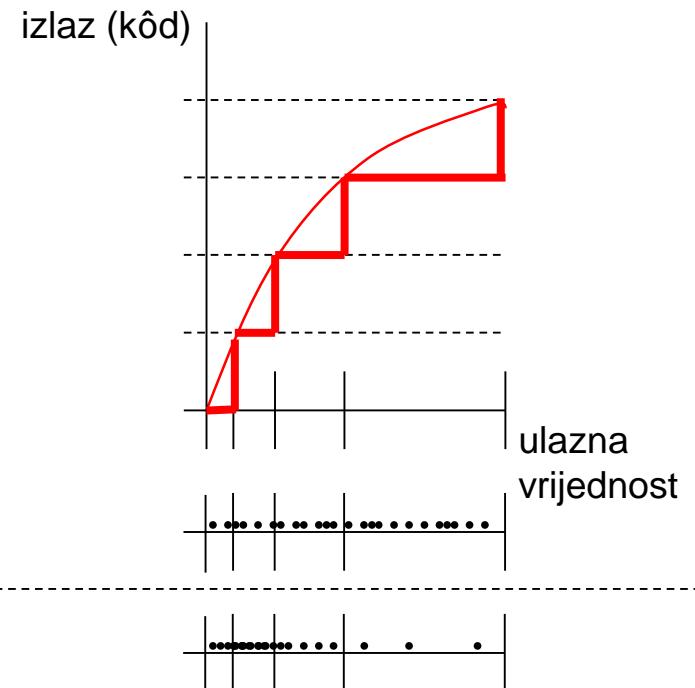


# Linearna i nelinearna kvantizacija

**a) Linearna kvantizacija**

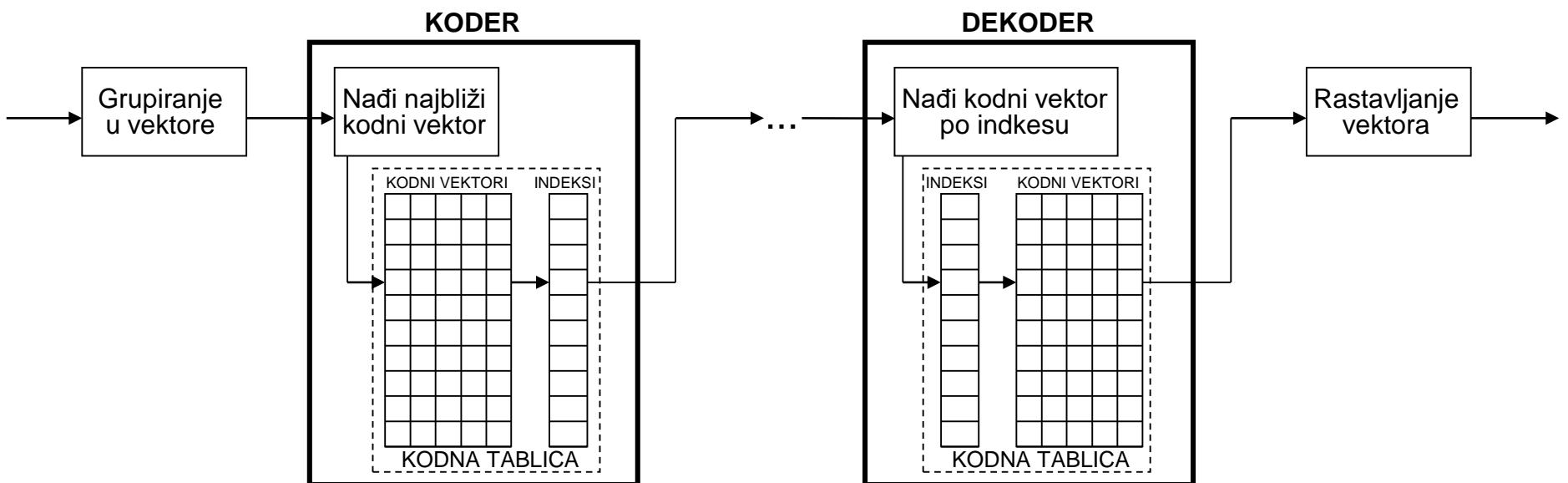


**b) Nelinearna kvantizacija**



# Vektorska kvantizacija

- ◆ Podaci se grupiraju u  $n$ -dimenzionalne vektore
- ◆ Za svaki vektor se u *kodnoj tablici* pronađe najблиži *kodni vektor*; njegov *indeks* je kod
- ◆ Za dekodiranje se koristi ista kodna tablica



# Primjer: 2D vektorska kvantizacija

◆ Kodna tablica:

- $i = 1: \quad y_1 = (0, 0)$
- $i = 2: \quad y_2 = (2, 1)$
- $i = 3: \quad y_3 = (1, 3)$
- $i = 4: \quad y_4 = (1, 4)$

◆ Poruka: 

◆ Kodirano: 

◆ Dekodirana poruka:

0      0      1      3      2      1

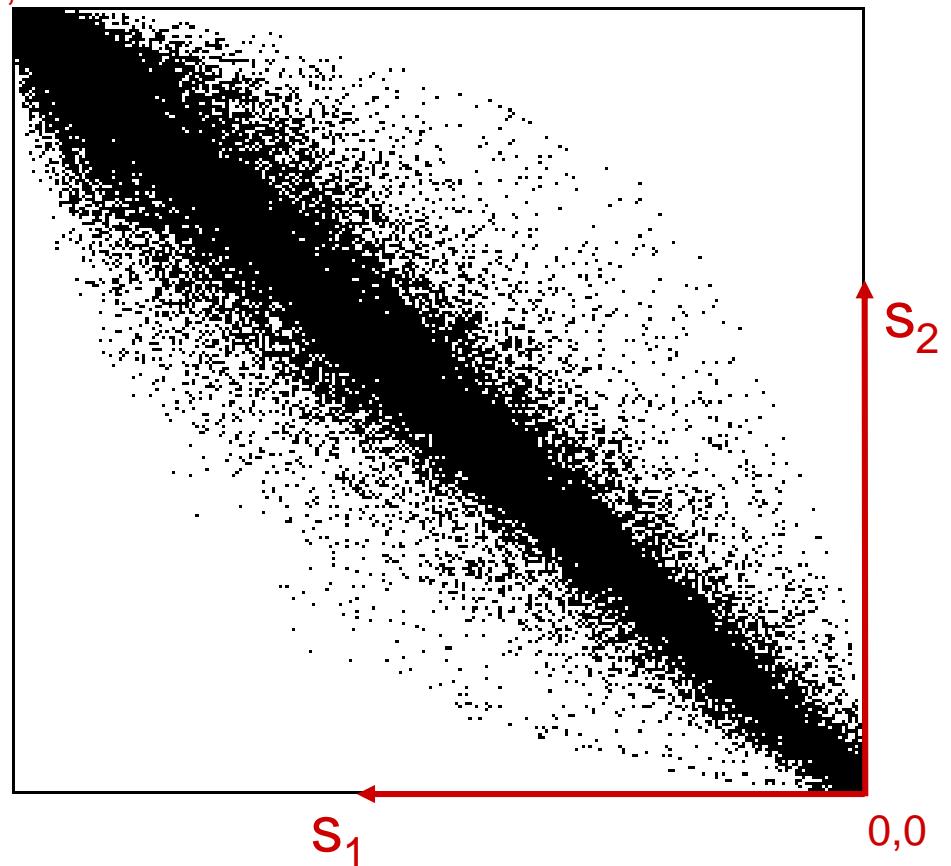
◆ Pogreška kvantizacije:

0      -1      -1      0      0      1

# Primjer: vektorska kvantizacija slike (1/2)



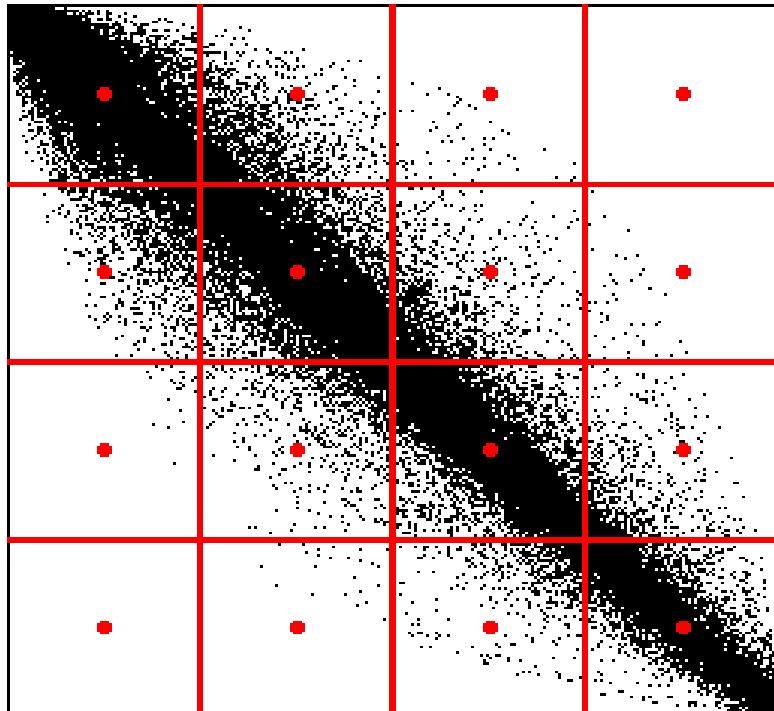
255,255



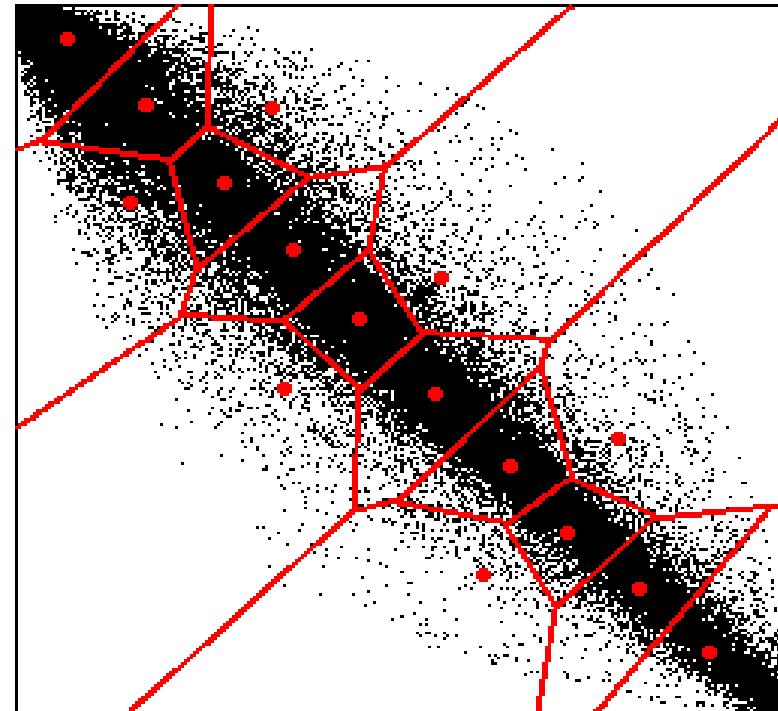
Izvorna slika sastoji se od  
točaka  $b_i$ ; svaka točka je  
vrijednost 0-255

Parovi svjetlina susjednih točaka slike

# Primjer: vektorska kvantizacija slike (2/2)



Skalarna kvantizacija  
prikazana u obliku  
vektorske kvantizacije



Vektorska kvantizacija

# Poduzorkovanje

- ◆ Smanjivanje frekvencije uzorkovanja
  - Slika: rezolucija
  - Zvuk: broj uzoraka zvuka u sekundi
- ◆ Smanjivanje broja uzoraka = sažimanje
- ◆ Nyquist: granica poduzorkovanja bez pogreške
- ◆ Svjesno unošenje pogreške
  - Uzima se u obzir:
    - ograničenja ljudske percepcije
    - posebnosti pojedine primjene
  - Prije poduzorkovanja niskopropusni filter

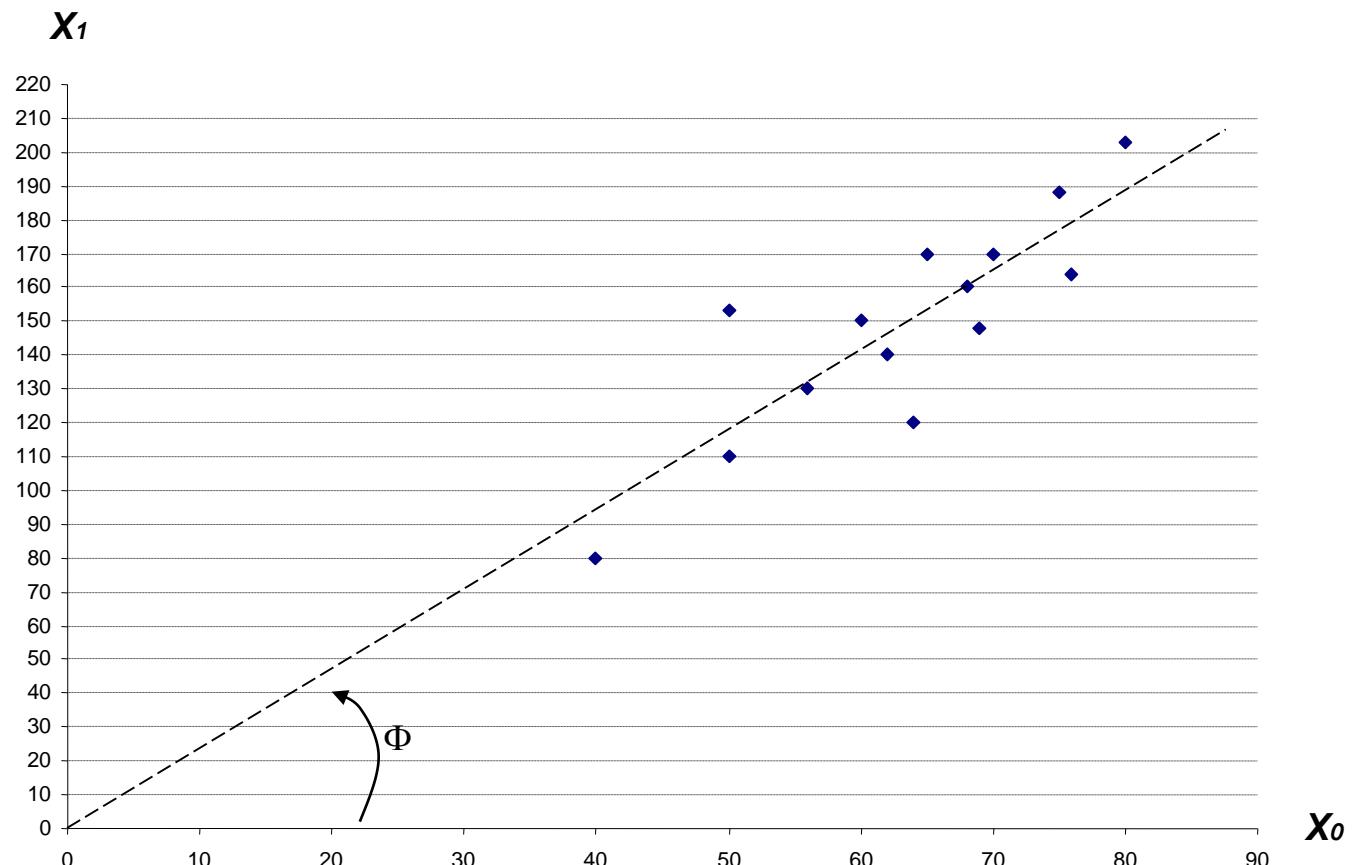
# Transformacijsko kodiranje

- ◆ Poruka se pretvara (*transformira*) u oblik pogodniji za kompresiju
  - npr. iz vremenske u frekvencijsku domenu
- ◆ Transformacija je reverzibilna i ne komprimira
- ◆ Kompresija u drugom koraku: odbacivanje i/ili kvantizacija
  - Transformacija omogućuje kompresiju, premda je izravno ne vrši

# Primjer: princip transf. kodiranja (1/3)

TEŽINA ( $X_0$ )	VISINA( $X_1$ )
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120

$$\mathbf{X} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

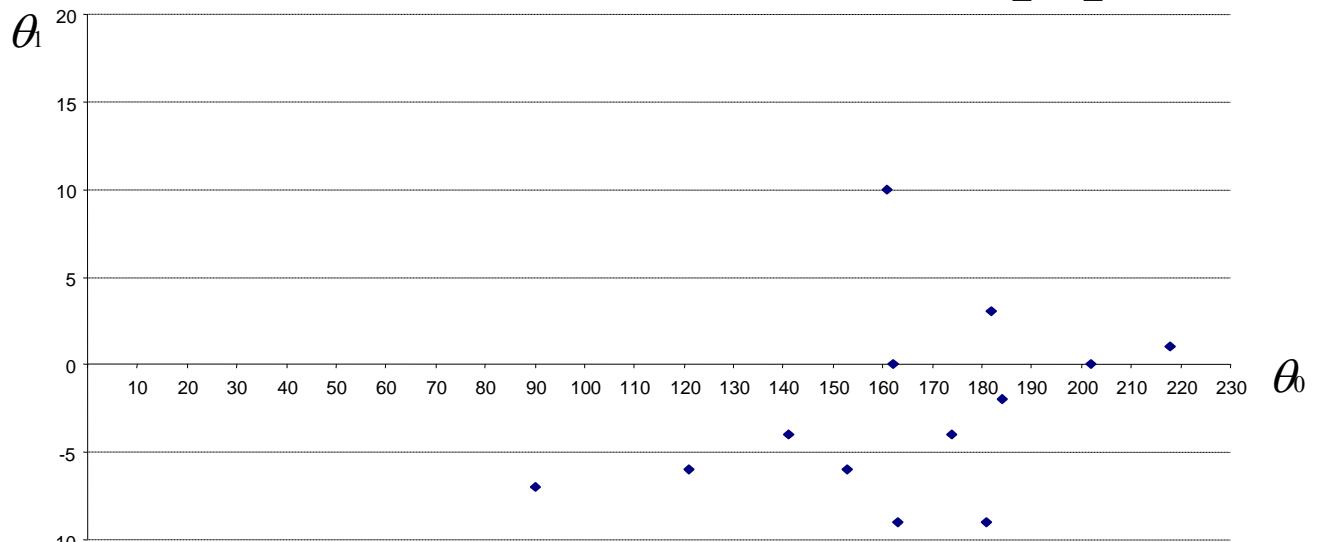


# Primjer: princip transf. kodiranja (2/3)

- ◆ Rotiramo točke grafa za  $\Phi$ :  $\Theta = AX$

$$A = \begin{bmatrix} \cos \Phi & \sin \Phi \\ -\sin \Phi & \cos \Phi \end{bmatrix} = \begin{bmatrix} 0.37139068 & 0.92847669 \\ -0.92847669 & 0.37139068 \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$\theta_0$	$\theta_1$
182	3
202	0
162	0
184	-2
141	-4
218	1
174	-4
121	-6
90	-7
161	10
163	-9
153	-6
181	-9
135	-15



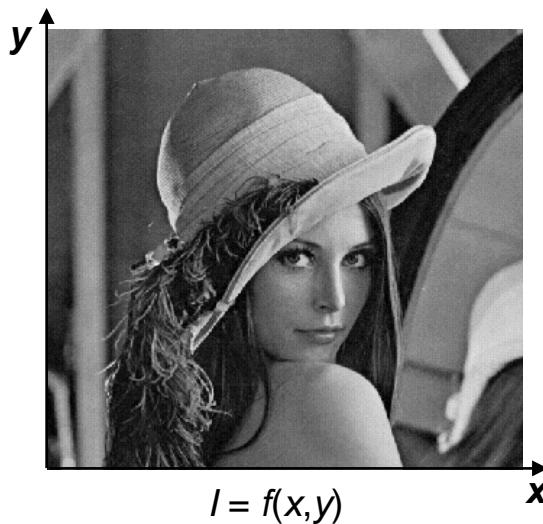
$$A^{-1} = \begin{bmatrix} \cos \Phi & -\sin \Phi \\ \sin \Phi & \cos \Phi \end{bmatrix} \quad X = A^{-1} \Theta$$

# Primjer: princip transf. kodiranja (3/3)

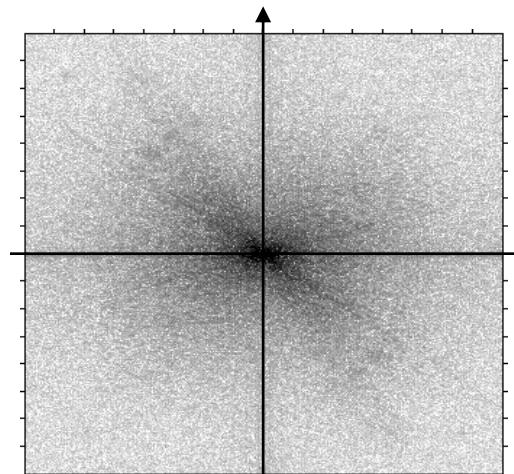
- ◆ Sve  $\theta_1$  izjednačimo s nulom  
(ne treba ih kodirati)
- ◆ Inverzna transformacija:  $\mathbf{X} = \mathbf{A}^{-1}\Theta$

Izvorni podaci		Rekonstruirani podaci	
TEŽINA ( $X_0$ )	VISINA( $X_1$ )	TEŽINA ( $X_0$ )	VISINA( $X_1$ )
65	170	68	169
75	188	75	188
60	150	60	150
70	170	68	171
56	130	53	131
80	203	81	203
68	160	65	162
50	110	45	112
40	80	34	84
50	153	60	150
69	148	61	151
62	140	57	142
76	164	67	168
64	120	50	125

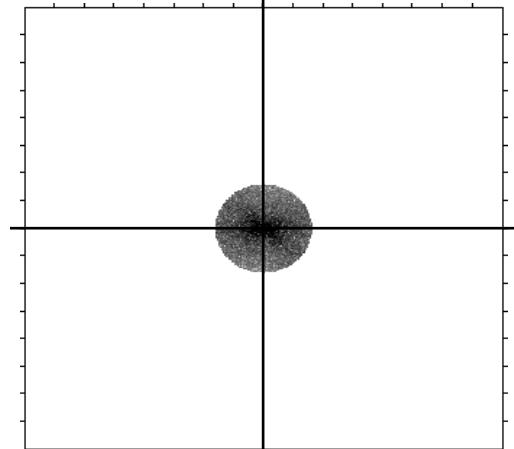
# Primjer 2: Fourierova transformacija slike



2D DFT  
→



↓ Odbacivanje

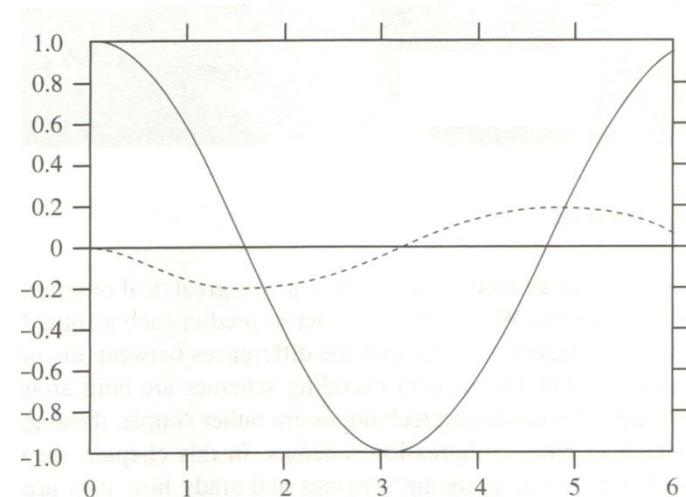


← 2D IDFT

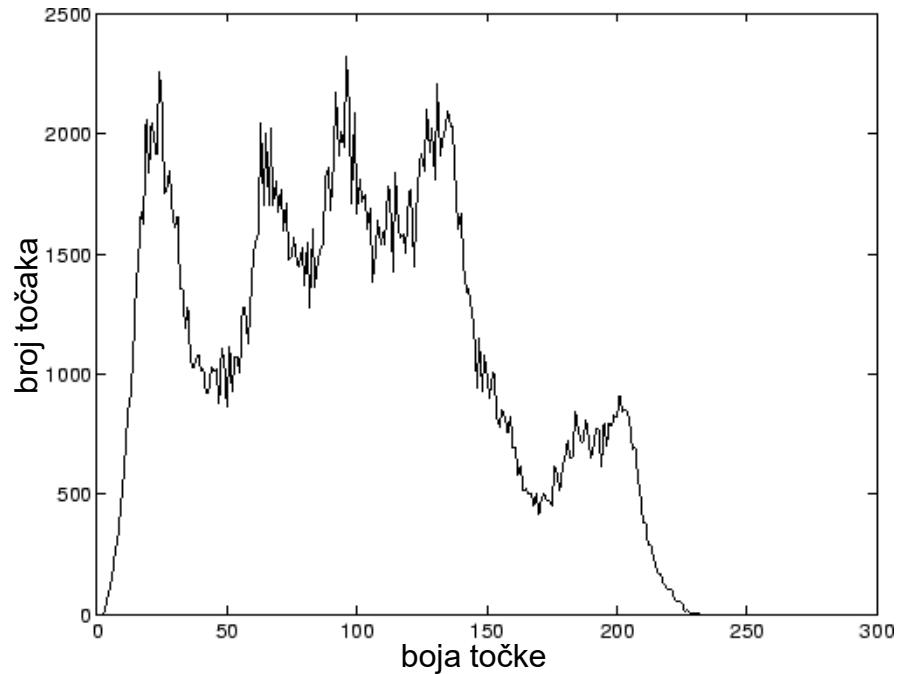


# Diferencijalno (predikcijsko) kodiranje

- ◆ Koristi korelaciju među susjednim uzorcima
  - U vremenu (npr. zvuka) ili prostoru (npr. slika)
- ◆ Svaki uzorak se predviđa iz prethodnih uzoraka
- ◆ Predviđena – stvarna vrijednost = ***signal razlike***
  - Manji raspon i promjena → bolje kodiranje
- ◆ Jednostavno predviđanje:  $x_n = x_{n-1}$ 
  - Signal razlike je razlika među susjednim uzorcima



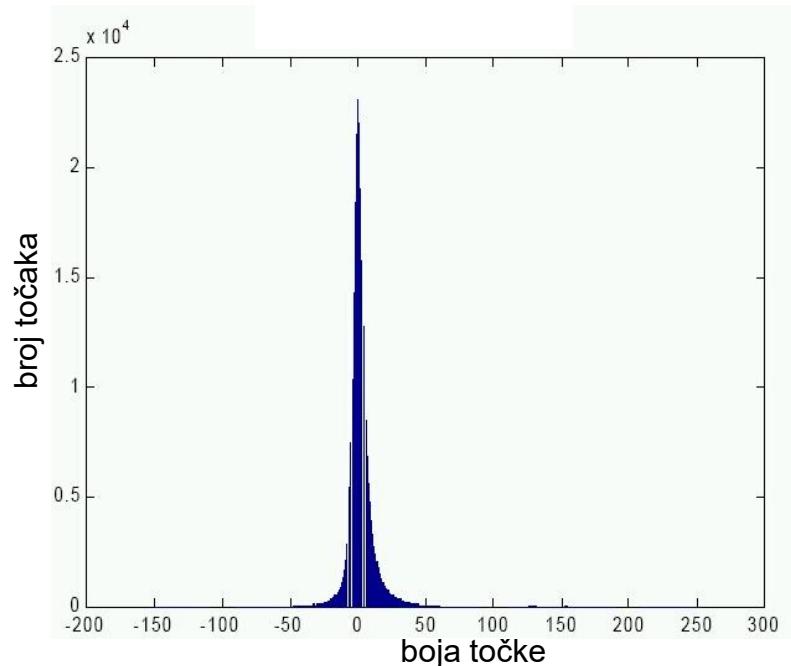
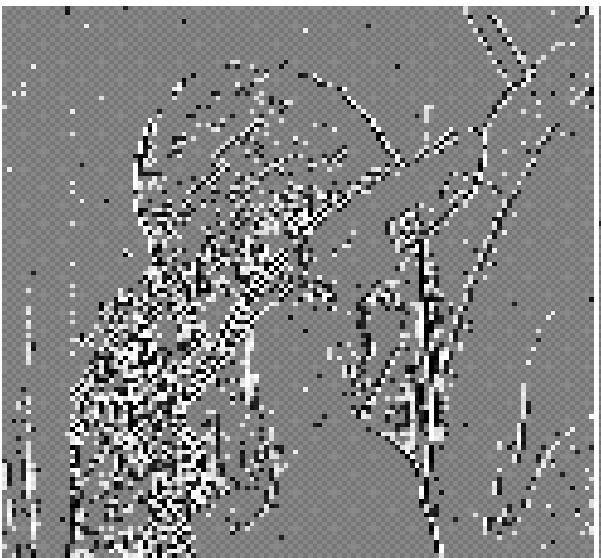
# Primjer: diferencijalno kodiranje slike



Direktnim entropijskim kodiranjem može se postići 7 bita po točki

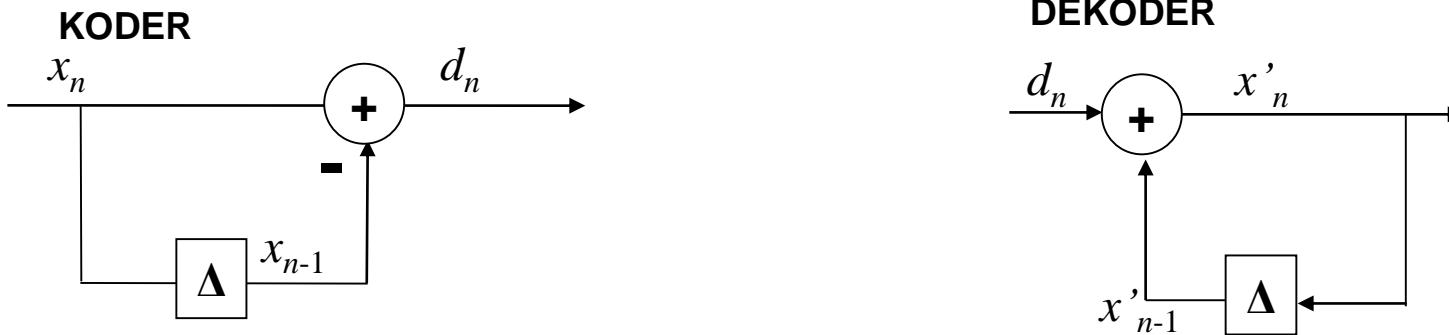
# Signal razlike

- ◆ Razlika susjednih pixela



- ◆ Signal se ujednačuje, koncentracija u malom broju vrijednosti, entropija 2.6 bita po točki

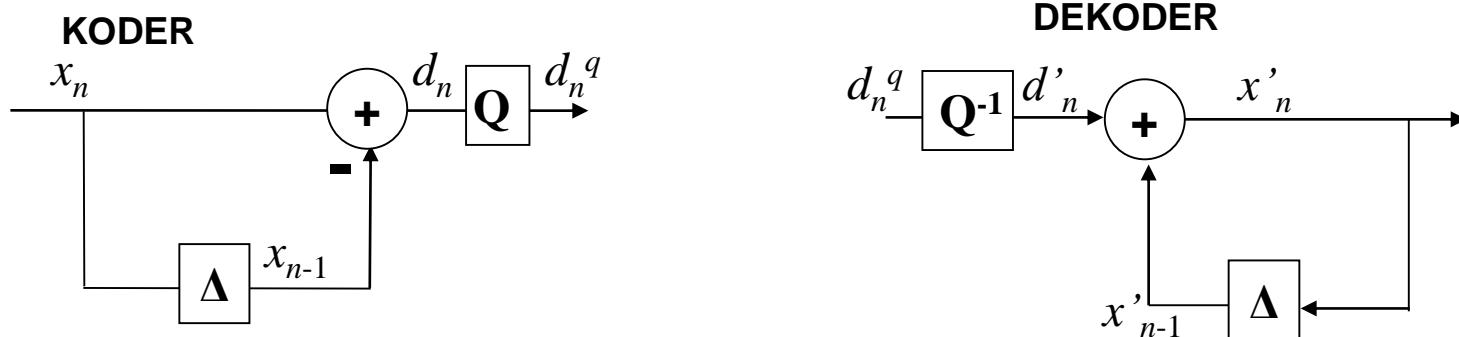
# Primitivni postupak diferencijalnog kodiranja



$$d_n = x_n - x_{n-1}$$

$$x'_n = d_n + x'_{n-1} = x_n - x_{n-1} + x'_{n-1} = x_n$$

- ◆  $\{x_n\}$ : 6.2 9.7 13.2 5.9 8 7.4 4.2 1.8
- ◆  $\{d_n\}$ : 6.2 3.5 3.5 -7.3 2.1 -0.6 -3.2 -2.4
- ◆  $\{x'_n\}$ : 6.2 9.7 13.2 5.9 8 7.4 4.2 1.8
- ◆ Pogreška: 0



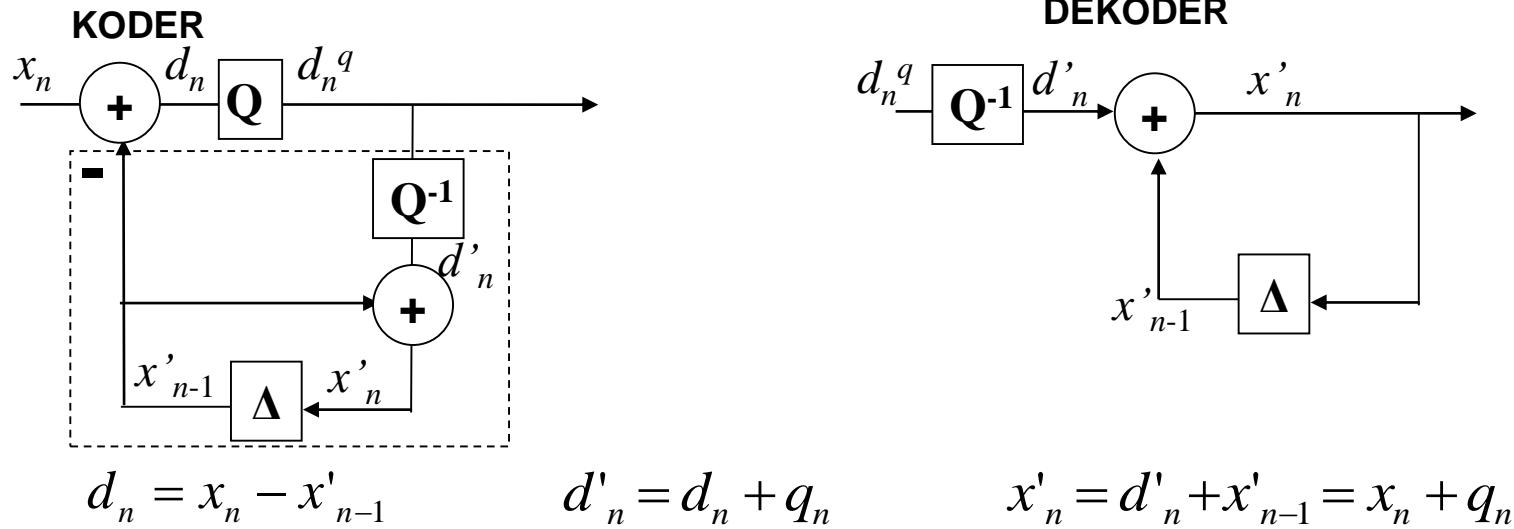
$$d_n = x_n - x_{n-1}$$

$$d'_n = d_n + q_n$$

$$x'_n = d'_n + x'_{n-1} = x_n + \sum_{k=1}^n q_k$$

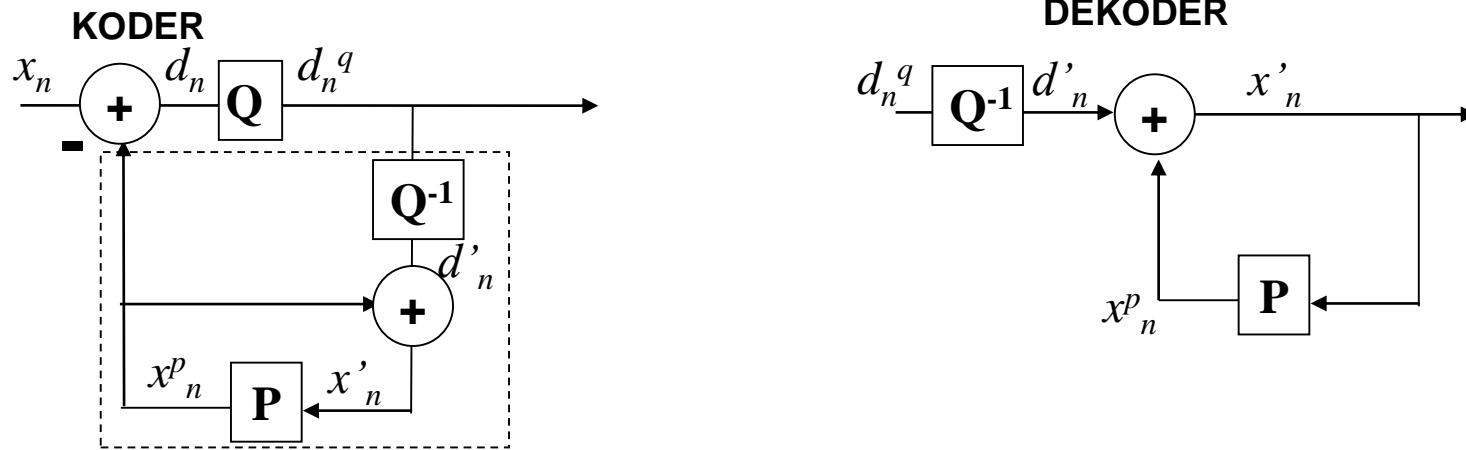
- ◆ Kvantizator sa 7 razina: -6, -4, -2, 0, 2, 4, 6
- ◆  $\{d'_n\}$ : 6 4 4 -6 2 0 -4 -2
- ◆  $\{x'_n\}$ : 6 10 14 8 10 6 4
- ◆  $\{\varepsilon_n\}$ : 0.2 -0.3 -0.8 -2.1 -2 -2.6 -1.8 -2.2
  - Pogreška kvantizacije se akumulira

# Izbjegavanje akumulacije kvant. pogreške



- ◆  $\{d'_n\}$ : 6 4 4 -6 0 0 -4 -2
- ◆  $\{x'_n\}$ : 6 10 14 8 8 8 4 2
- ◆  $\{\varepsilon_n\}$ : 0.2 -0.3 -0.8 -2.1 0 -0.6 0.2 -0.2

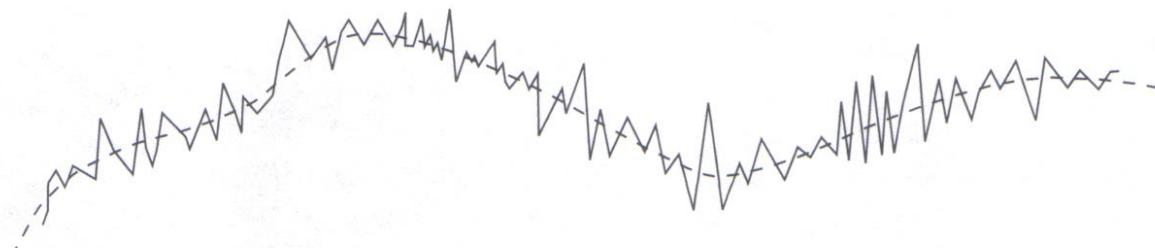
# Osnovni postupak diferencijalnog kodiranja



- ◆ Prediktor:  $x_n^p = P(x_{n-1}, x_{n-2}, \dots, x_{n-k})$
- ◆ Linearna predikcija:  $x_n^p = a_1 x_{n-1} + a_2 x_{n-2} + \dots + a_k x_{n-k}$

# Podpojasno kodiranje

- ◆ Signal se razdvaja na frekvencijske pojaseve
  - Dobiva se niz signala, po jedan za svaki frek. pojas
  - Svaki ima drugačije karakteristike i važnost
- ◆ Kodiranje svakog pojasa posebno
  - Više ili manje bitova s obzirom na važnost
  - Razne metode kodiranja s obzirom na karakteristike
- ◆ Primjeri primjene: kodiranje zvuka (MP3), slike (JPEG 2000)



# Primjer (1/5)

- ◆ Kodiramo niz  $\{x_n\}$

10 14 10 12 14 8 14 12 10 8 10 12

- ◆ Diferencijalno kodiranje; signal razlike

10 4 -4 2 2 -6 6 -2 -2 -2 2 2

- ◆ Dinamički raspon -6 do 6 = 12

- ◆ Koristimo  $m$  bitova;  $M = 2^m$  razina kvantizacije

- ◆ Kvantizacijski interval:  $\Delta = 12/M$

- ◆ Maksimalna pogreška kvantizacije:  $\Delta/2 = 6/M$

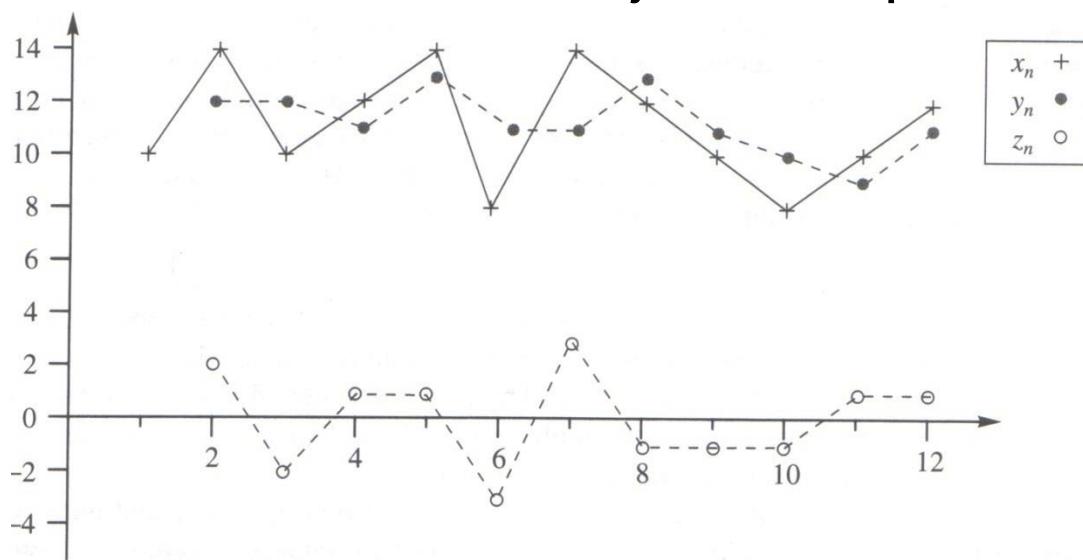
# Primjer (2/5)

- ◆ Rastavimo niz  $\{x_n\}$  na dva nova niza  $\{y_n\}$  i  $\{z_n\}$ 
  - $\{y_n\}$  je prosjek susjednih vrijednosti – niskofrekvenčna komponenta
  - $\{z_n\}$  je razlika susjednih vrijednosti – visokofrekvenčna komponenta

$$y_n = \frac{x_n + x_{n-1}}{2}$$

$$z_n = \frac{x_n - x_{n-1}}{2}$$

$$x_n = y_n + z_n$$



# Primjer (3/5)

- ◆ Kodiramo niz  $\{y_n\}$

10 12 12 11 13 11 11 13 11 10 9 11

- ◆ Diferencijalno kodiranje; signal razlike

10 2 0 -1 2 -2 0 2 -2 -1 -1 2

- ◆ Dinamički raspon -2 do 2 = 4

- ◆ Koristimo  $m$  bitova;  $M = 2^m$  razina kvantizacije

- ◆ Kvantizacijski interval:  $\Delta = 4/M$

- ◆ Maksimalna pogreška kvantizacije:  $\Delta/2 = 2/M$

# Primjer (4/5)

- ◆ Kodiramo niz  $\{z_n\}$

0 2 -2 1 1 -3 3 -1 -1 -1 1 1

- ◆ Signal razlike imao bi veću varijaciju od samog niza, stoga ne koristimo diferencijalno kodiranje nego direktno kvantiziramo
- ◆ Dinamički raspon  $-3 \text{ do } 3 = 6$
- ◆ Koristimo  $m$  bitova;  $M = 2^m$  razina kvantizacije
- ◆ Kvantizacijski interval:  $\Delta = 6/M$
- ◆ Maksimalna pogreška kvantizacije:  $\Delta/2 = 3/M$

# Primjer (5/5)

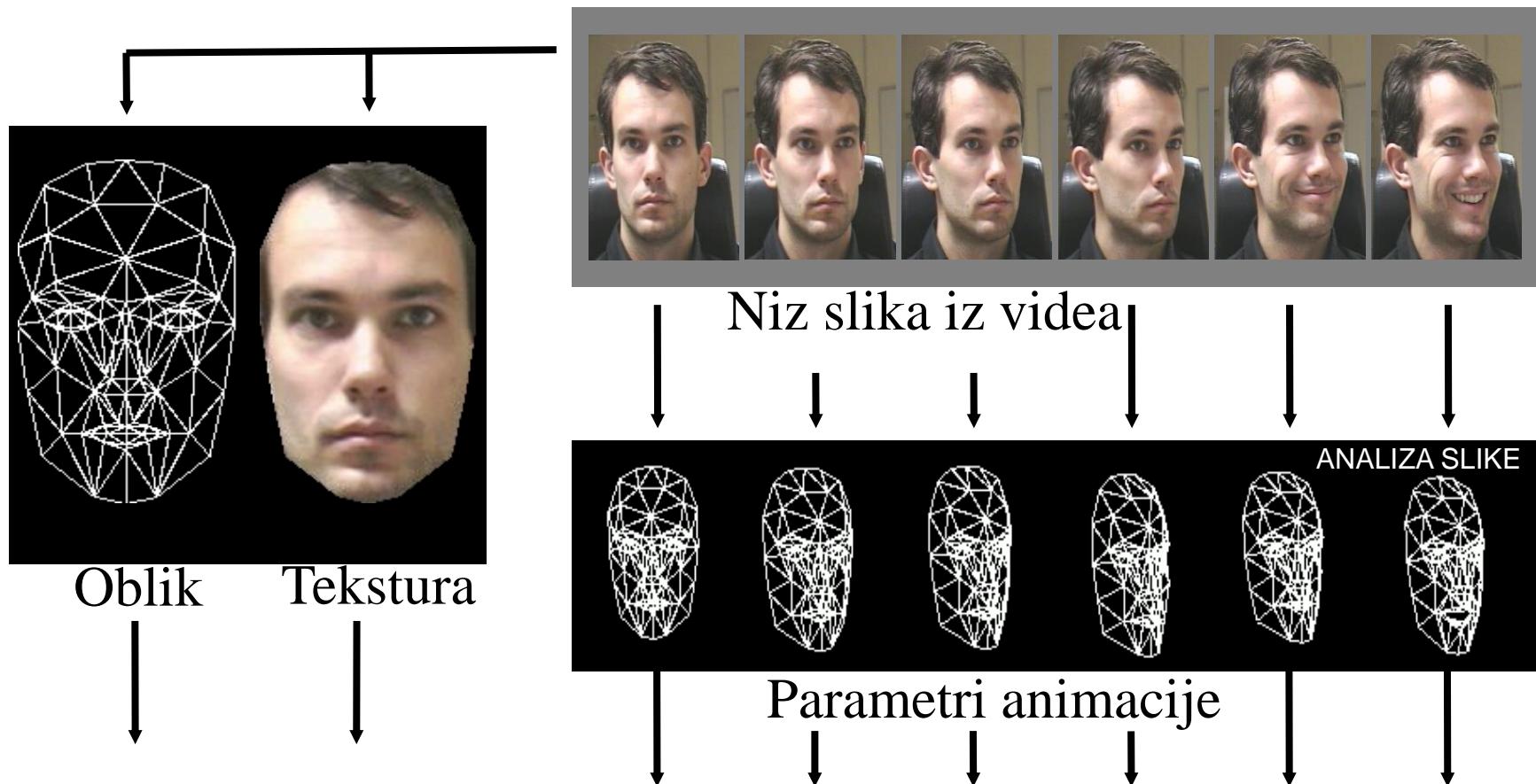
- ◆ Uz jednak broj bitova, max. pogreška kvantizacije
  - Za  $\{x_n\} \rightarrow 6/M$
  - Za  $\{y_n\} \rightarrow 2/M$
  - Za  $\{z_n\} \rightarrow 3/M$
- Rastavljanje na komponente = bolje kodiranje!
- ◆ Međutim, kodiramo dva niza umjesto jednog!?
  - Ne! Kodiramo samo svaki drugi član iz  $\{y_n\}$  i  $\{z_n\}$
  - Iz svakog drugog para  $y_n$  i  $z_n$  rekonstruiramo dva susjedna člana niza  $\{x_n\}$

$$x_n = y_n + z_n \quad x_{n-1} = y_n - z_n$$

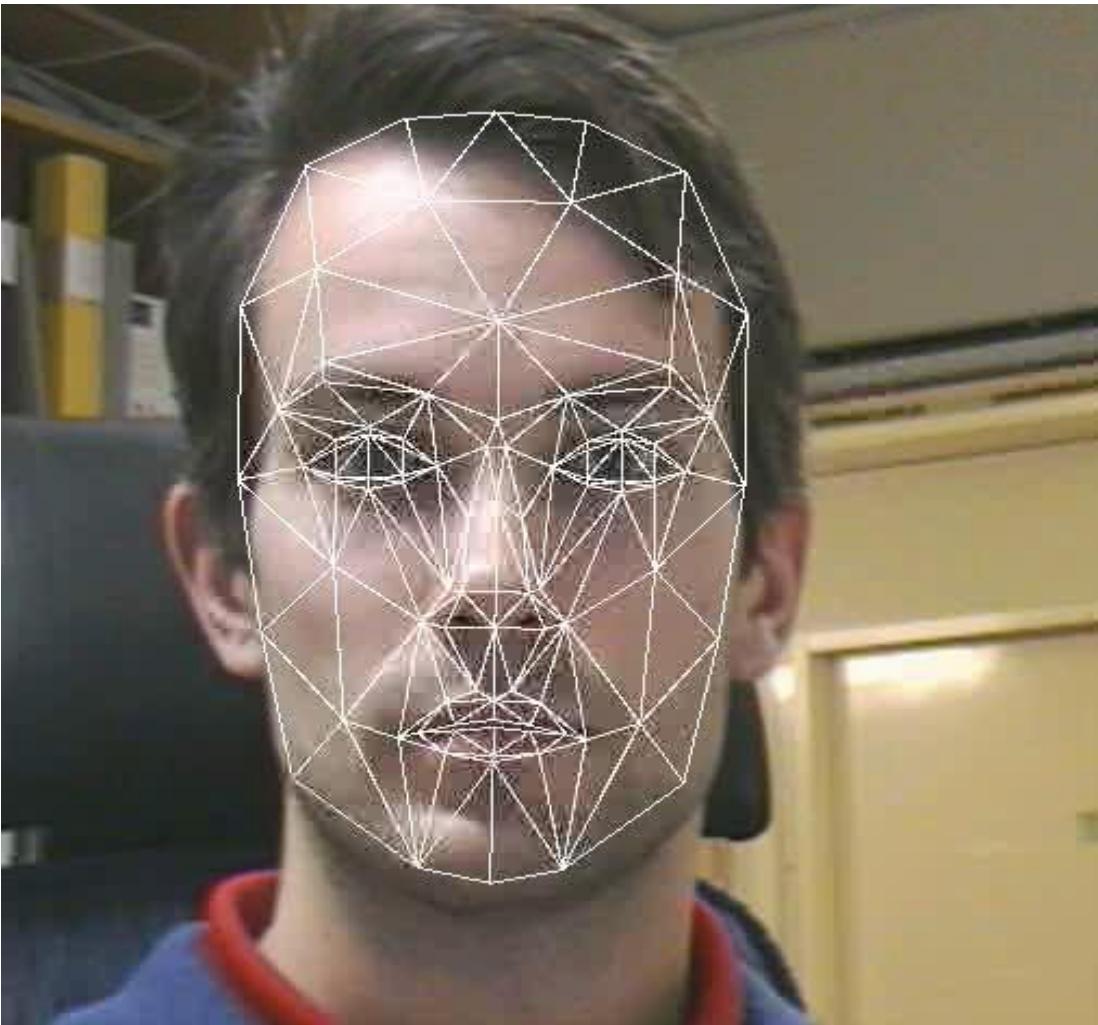
# Kodiranje zasnovano na modelu

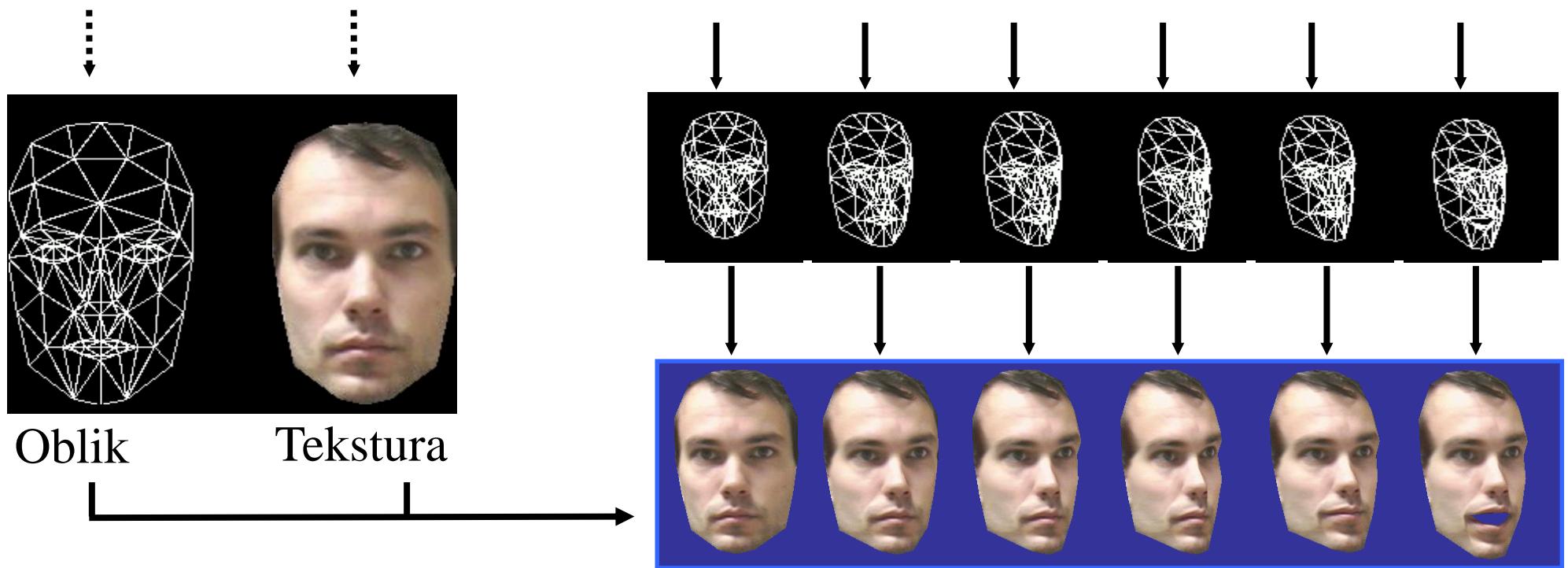
- ◆ Ne prenose se uzorci, nego parametri modela
- ◆ Na dekoderu se iz parametara pomoću modela sintetiziraju podaci slični izvornima
- ◆ Primjer
  - Koderi govora zasnovani na modelu (GSM)
  - Fraktalno kodiranje slike
  - Kodiranje videa ljudskog lica

# Primjer: video ljudskog lica



# Praćenje parametara u videu





# Rezultat dekodiranja na drugom licu



# Još jedan primjer





**FER**

Diplomski studij

Informacijska i komunikacijska  
tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

3.

Informacijska svojstva i  
kodiranje jezika

# Informacijska svojstva i kodiranje jezika

- ◆ Kolika je entropija prirodnog jezika?
- ◆ Kako kodirati tekst?

# Kolika je entropija hrvatskog jezika?

- ◆ Osnovni simbol: slovo
- ◆ 27 slova (uključujući razmak):  
$$H = \log 27 = 4,755 \text{ bit/simbol}$$

# Vjerojatnost pojavljivanja pojedinih slova

znak	rel.frekv.	kod	znak	rel.frekv.	kod	znak	rel.frekv.	kod
razmak	0.1700	000	t	0.0367	10100	b	0.0155	11100
a	0.0960	001	u	0.0364	10101	z	0.0144	111010
e	0.0770	0100	d	0.0319	10110	š	0.0086	111011
o	0.0754	0101	m	0.0313	10111	č	0.0084	111100
i	0.0742	0110	v	0.0306	11000	c	0.0067	111101
n	0.0464	0111	l	0.0306	11001	h	0.0065	1111100
j	0.0435	1000	k	0.0298	11010	ž	0.0052	1111101
s	0.0420	10010	p	0.0204	110110	ć	0.0049	1111110
r	0.0382	10011	g	0.0166	110111	f	0.0011	1111111

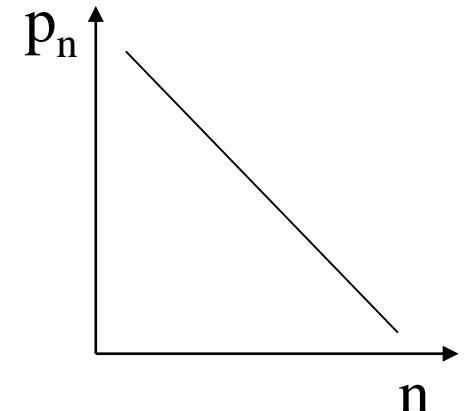
$$H = - \sum_{i=1}^{27} p(x_i) \log p(x_i) = 4,19$$

- ◆ Npr. iza samoglasnika vjerojatniji suglasnik
- ◆ Neodređenost je smanjena
- ◆ Promatramo po dva susjedna znaka:  $H = 3,59$
- ◆ Promatramo po tri susjedna znaka:  $H = 3,1$

# Entropija uz promatranje čitavih riječi

- ◆ Zipfov zakon:  $p_n = \frac{P}{n}$ 
  - n: redni broj riječi, počevši od najčešćih
  - $p_n$ : vjerojatnost pojavljivanja riječi n
  - P: konstanta
- ◆ Prosječni sadržaj informacije po riječi

$$\bar{I}_r = -\sum_{n=1}^R p_n \log p_n [\text{bit / riječ}]$$



- ◆ Dijelimo s prosječnim brojem slova po riječi; za engleski dobivamo 1,66 bit/simbol
- ◆ Korelacijske među riječima, gramatika...: **0,6 – 1.3 bit/simbol**

Međutim...

- ◆ Metoda kodiranja zasnovana na ovim razmatranjima bila bi složena
- ◆ Gramatika, riječi, slova, vjerojatnosti pojave riječi i slova: sve ovisi o jeziku
- ◆ Jednostavnost i univerzalnost važnija od moguće uštede
- ◆ Stoga: ASCII (8 bit), Unicode (16 bit)



Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

4.

Informacijska svojstva i  
kodiranje zvuka

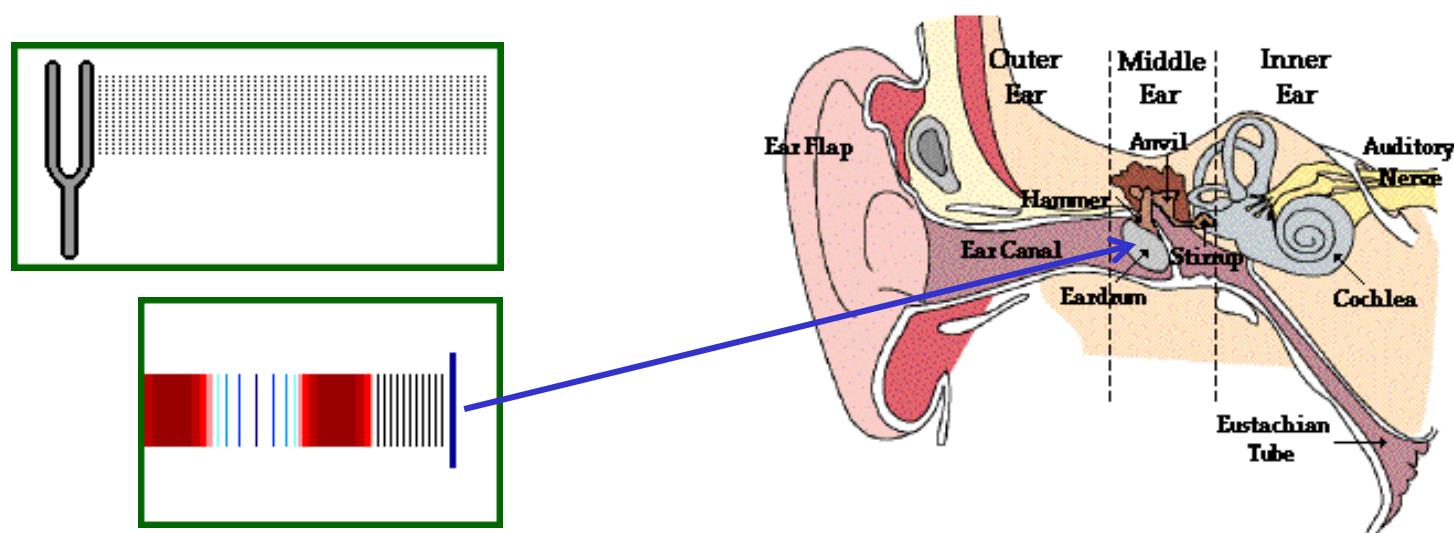
# Informacijska svojstva i kodiranje zvuka

- Fizikalne osnove, svojstva i percepcija zvuka
- Digitalni zvuk u umreženom računalnom sustavu
- Kriteriji usporedbe i mjerjenje kvalitete kodera
- Kodiranje govora
  - Karakteristike govora važne za kodiranje
  - Koderi govora
    - Koderi valnog oblika: PCM, ADPCM
    - Koderi zasnovani na modelu: LPC, CELP
- Koderi zvuka u frekvencijskoj domeni
  - Osnovni percepcijski koder
  - MPEG audio

# Fizikalne osnove zvuka

## • Longitudinalni val titranja čestica zraka

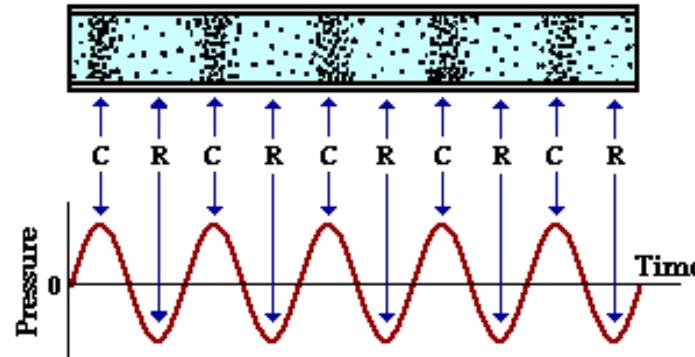
- Izvor zvuka stvara titranje čestica (promjene tlaka)
- Titranje se širi brzinom  $v=\lambda f \approx 340 \text{ m/s}$  (ovisno o temperaturi)
- Lokalno titranje tlaka uzrokuje titranje struktura uha
- Pužnica ima vise od 10000 „detektor“ osjetljivih na razne frekvencije; oni pretvaraju zvuk u žičane signale



# Fizikalna svojstva zvuka

- Vremenskoj domena: valni oblik

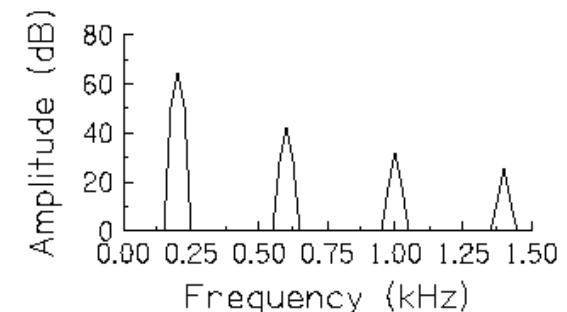
- amplituda
- faza
- frekvencija



- Frekvencijska domena: frekvencijski spektar

- harmonici (višekratnici osnovne frekvencije  $f$ ,  $2f$ ,  $3f$ , ...)
- za govor, formanti (rezonantne frekvencije govornog trakta), npr:

- a   F1 660   F2 1700 F3 2400 Hz
- e   F1 530   F2 1850 F3 2500 Hz
- u   F1 300   F2 870   F3 2250 Hz



# Percepcija zvuka

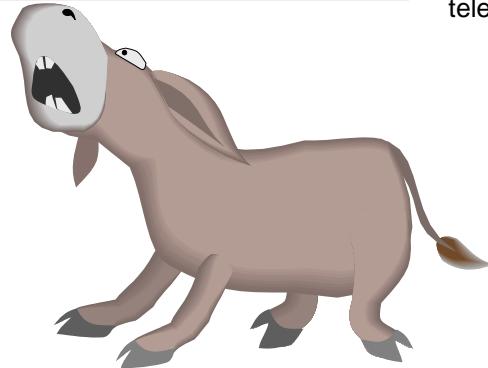
- Frekvencija ( $f$ ) [Hz] → **visina**:  $20 \log (f/f_r)$  [Mel]
  - **Jednake rastuće korake frekvencije primjećujemo kao sve manje i manje korake rasta visine zvuka**
    - Čujno područje 20 Hz do 20 kHz
- Amplituda (A) [Pa] → Intenzitet ( $I$ ) [ $\text{W/m}^2$ ] → **glasnoca**:  $20 \log (I/I_r)$  [dB] ( $I_r$  je granica čujnosti)
  - **Jednake rastuće korake amplitude (intenziteta) primjećujemo kao sve manje i manje korake rasta jačine odnosno, glasnoće zvuka**
    - granica čujnosti 0 dB ( $1*10^{-12} \text{ W/m}^2$ )
    - šapat 25 dB
    - uredska buka 50 dB
    - razgovor 60-65 dB
    - prosječni kućni stereo uređaj ~90 dB
    - granica boli 120 dB
    - ozljeda bubnjića 130 dB



dinamički raspon  
ljudskog sluha  
120 dB

# Izvori zvuka

- Prirodni - analogni
  - ljudski glas (govor, pjevanje)
  - glasanje životinja
  - glazbeni instrumenti
  - prirodni zvukovi, npr. šum mora, vjetar, grmljavina,...
- Sintetički – obično digitalni
  - oponašanje nekog od prirodnih zvukova, računalna glazba, ...
- Digitalni oblik je pogodan za pohranu, obradu i prijenos u računalnim sustavima
- Prirodni zvukovi se stoga digitaliziraju
- Svrha kodiranja digitalnog zvuka: što manji zapis bez osjetnog gubitka kvalitete



# Digitalni zvuk u umreženom računalnom sustavu

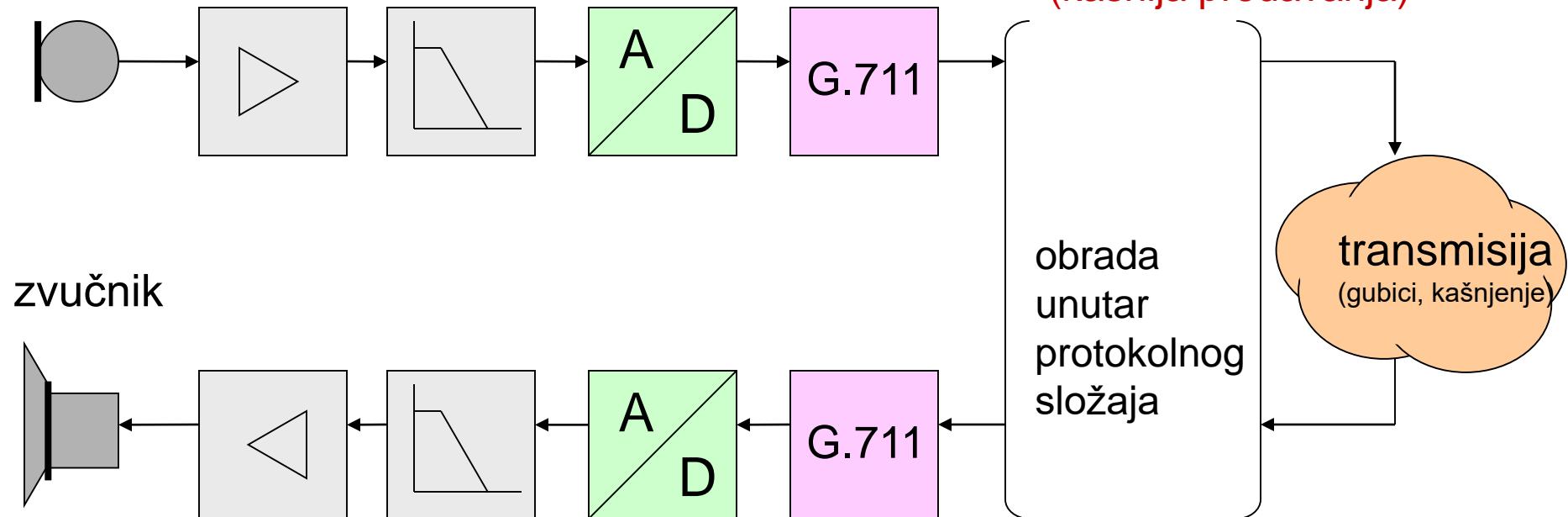
(ovo predavanje)

mikrofon      pojačalo      filter

A/D

codec

(kasnija predavanja)

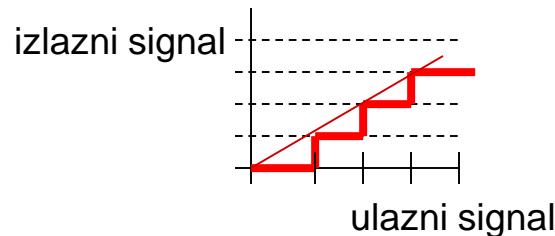


- **codec, coder + decoder** = uređaj koji kodira i dekodira, odn. komprimira i dekomprimira, audio i video

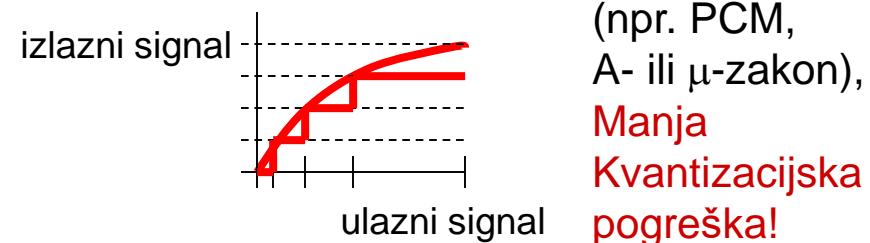
# Postupak digitalizacije zvuka

- 1. uzorkovanje
- 2. kvantizacija
  - više ulaznih vrijednosti preslikava se na istu izlaznu vrijednost čime se gubi mogućnost točne rekonstrukcije
  - razlika stvarne i kvantizirane vrijednosti je kvantizacijska pogreška
  - uz broj bita  $k$  dobiva se  $2^k$  razina kvantizacije
  - općenito vrijedi: veći broj bita po uzorku → manje izobličenje

*linearna* kvantizacija



*nelinearna* kvantizacija



kvantizacija može biti *fiksna* (zadane razine) ili *adaptivna* (adaptivno određene razine kvantizacije)

- 3. kodiranje

# Vrste kodera s obzirom na namjenu

- **uskopojasni** (telefonski kanal;  $B = 4 \text{ kHz}$ )
  - prikaz zvuka s visokom točnošću:  
 $f_u = 8 \text{ kHz}$ , 16 bita po uzorku (linearno)  $\rightarrow 128 \text{ kbit/s}$
  - analogna telefonija: 300 Hz – 3400 Hz  
 $f_u = 8 \text{ kHz}$ , 8 bita po uzorku (nelinarno, A/ $\mu$ )  $\rightarrow 64 \text{ kbit/s}$
- **širokopojasni** (AM radio, ISDN; videokonferencija;  $B = 7 \text{ kHz}$ )
- **zvuk visoke kvalitete** (FM radio, televizija;  $B = 15 \text{ kHz}$ )
  - kanal 30 Hz – 15 kHz  
 $f_u = 32 \text{ kHz}$ , 16 bita po uzorku
- **zvuk visoke vjernosti** (CD kvaliteta;  $B = 20 \text{ kHz}$ )
  - svaki kanal 20 Hz – 20 kHz  
 $f_u = 44.1 \text{ kHz}$ , 16 bita po uzorku

# Kriteriji usporedbe codec-a

- brzina, izlaz, *bitrate* (bit/s)
- kvaliteta
  - objektivna mjerila (“klasične” metode, izobličenje signala i SNR, nisu dobra mjerila za ljudsku percepciju rekonstruiranog signala)
  - subjektivna mjerila (često važnija od objektivnih!)
- kašnjenje
  - algoritamsko kašnjenje u koderu na izvoru - koliko traje kodiranje
  - kod dekodiranja - koliko traje dekodiranje?
  - sinkronizacija s ostalim medijima u višemedijskoj aplikaciji
- otpornost na gubitke
  - posebno važno za prijenos preko mreže
- primjena na ostale zvukove koji nisu govor, npr. fax i modemske signale, te glazbu
- složenost (hw/sw)
- cijena izvedbe

# Subjektivna mjerila kvalitete

- Opća ocjena kvalitete - mišljenje korisnika
  - Mean Opinion Score (MOS)
  - Degradation MOS (DMOS)

Ocjena	MOS	DMOS	Procjena napora razumijevanja
5	izvrsno	nečujno pogoršanje	bez napora
4	dobro (toll quality)	čujno pogoršanje, ali ne smeta	bez posebnog napora
3	prihvatljivo	primjetno pogoršanje, malo smeta	osrednji napor
2	slabo	podnošljivo, ali smeta	priličan napor
1	loše	izraženo pogoršanje, jako smeta	neprepoznatljivo bez izrazitog napora

- Ocjena razumljivosti govora
  - dijagnostički testovi s parovima riječi koje slično zvuče, 90% točnost smatra se “toll quality”
  - Dynamic Rhyme Test (DRT) – 96 parova engleskih riječi, npr. *dune/tune, chair/care, moon/noon, ..* slušačima se nudi popis i pita ih se koju riječ su čuli

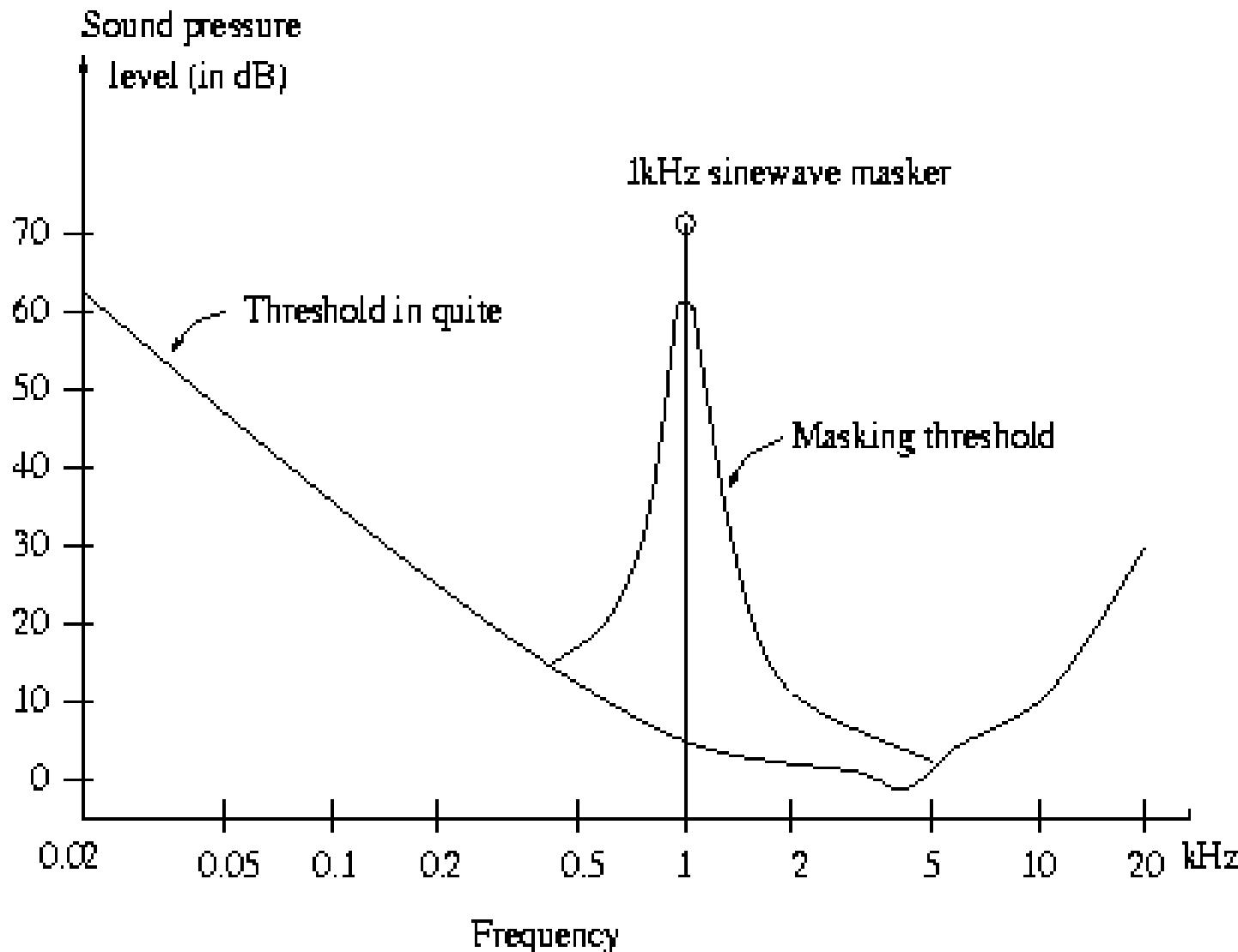
# Objektivne metode

- novije objektivne metode temelje se na poznavanju ljudskog slušnog sustava – računaju izobličenje zvučnog signala s *percepcijskim* težinskim faktorima
  - ideja: izobličenja koje uho “više čuje” ima veći težinski faktor od onog manje primjetnog ili neprimjetnog
- Perceptual Evaluation of Speech Quality (**PESQ**)
  - ITU-T preporuka P.862
  - metoda procjene subjektivne kvalitete govornih kodeka
  - algoritam predviđa subjektivnu ocjenu kvalitete degradiranog uzorka govora
  - izlaz iz algoritma je procijenjena vrijednost MOS
- Perceptual Evaluation of Audio Quality (**PEAQ**)
  - ITU-R preporuka BS.1387
  - algoritam za procjenu kvalitete audia

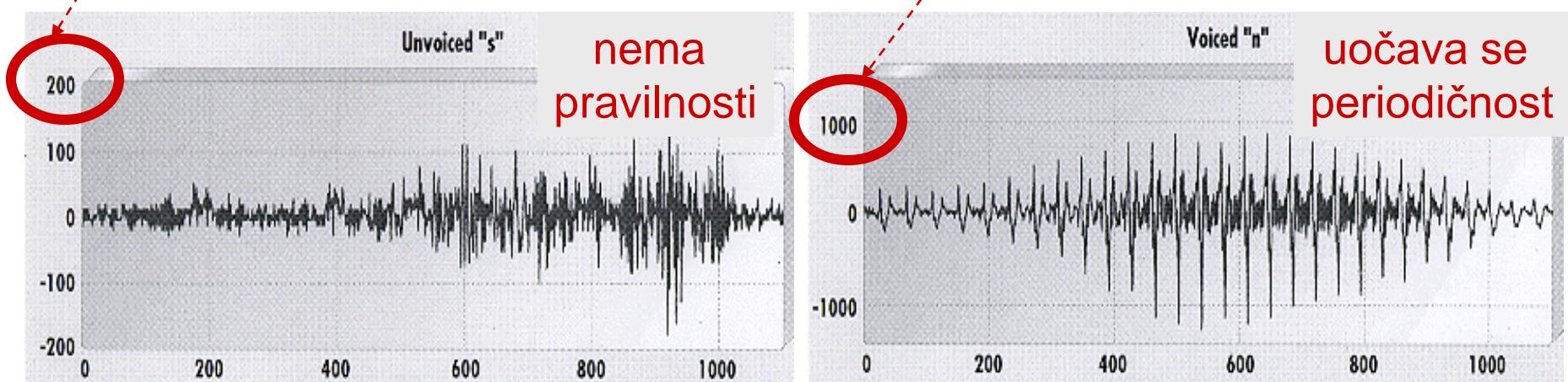
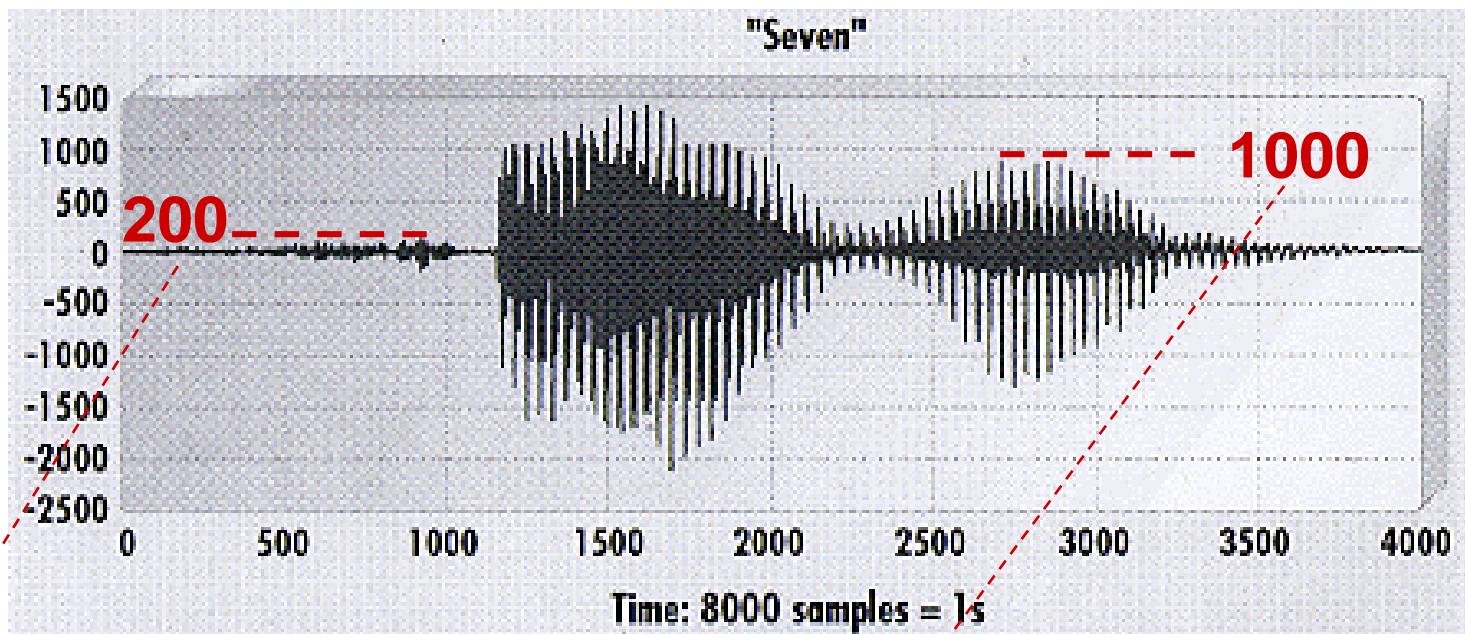
# Karakteristike govora važne za kodiranje

- Izvor: ljudski govorni organi
  - Zrak iz pluća prolaskom kroz govorne organe (dušnik, grkljan, glasnice, šupljine usta i nosa) stvara glas
    - **zvučni glasovi** (zrak izaziva titranje glasnica)
    - **bezvučni glasovi** (nema titranja glasnica)
  - Frekv. raspon 60 Hz - 8 kHz, dinamički raspon 40 dB
- Prijamnik: ljudski slušni organi
  - Čujno područje 20Hz – 20 kHz, dinamički raspon 120 dB
  - Za razumljivost najvažnije 2 - 5 kHz (500-2000Hz)
  - Kvaliteta se ocjenjuje subjektivno
  - Poznata anomalija: efekt *maskiranja*: jedan zvučni signal prekriva drugi (ovisno o relativnim glasnoćama i frekvencijama)

# Maskiranje zvuka



# Vremenska analiza govora

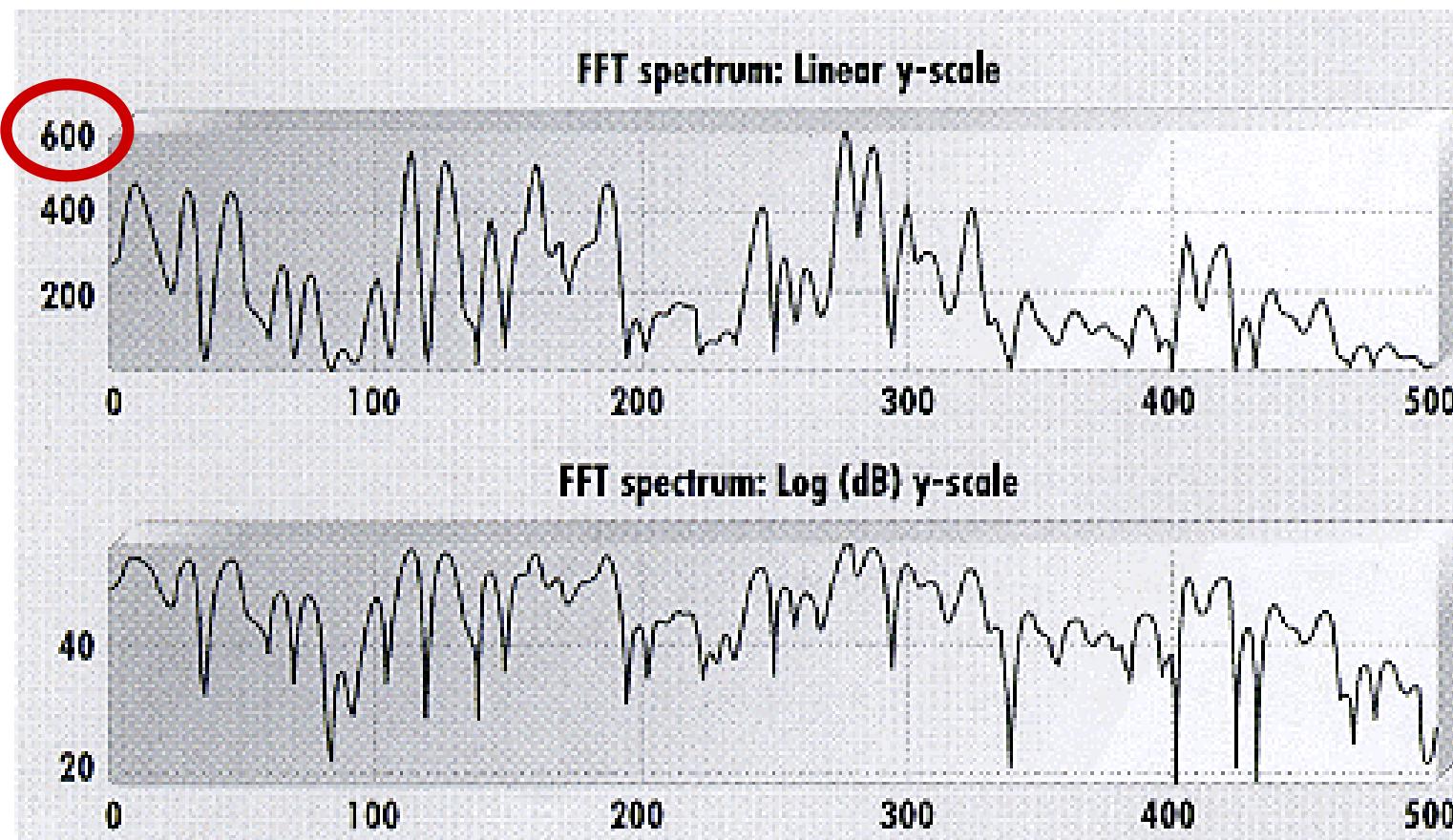


bezučni suglasnik "s"

zvučni suglasnik "n"

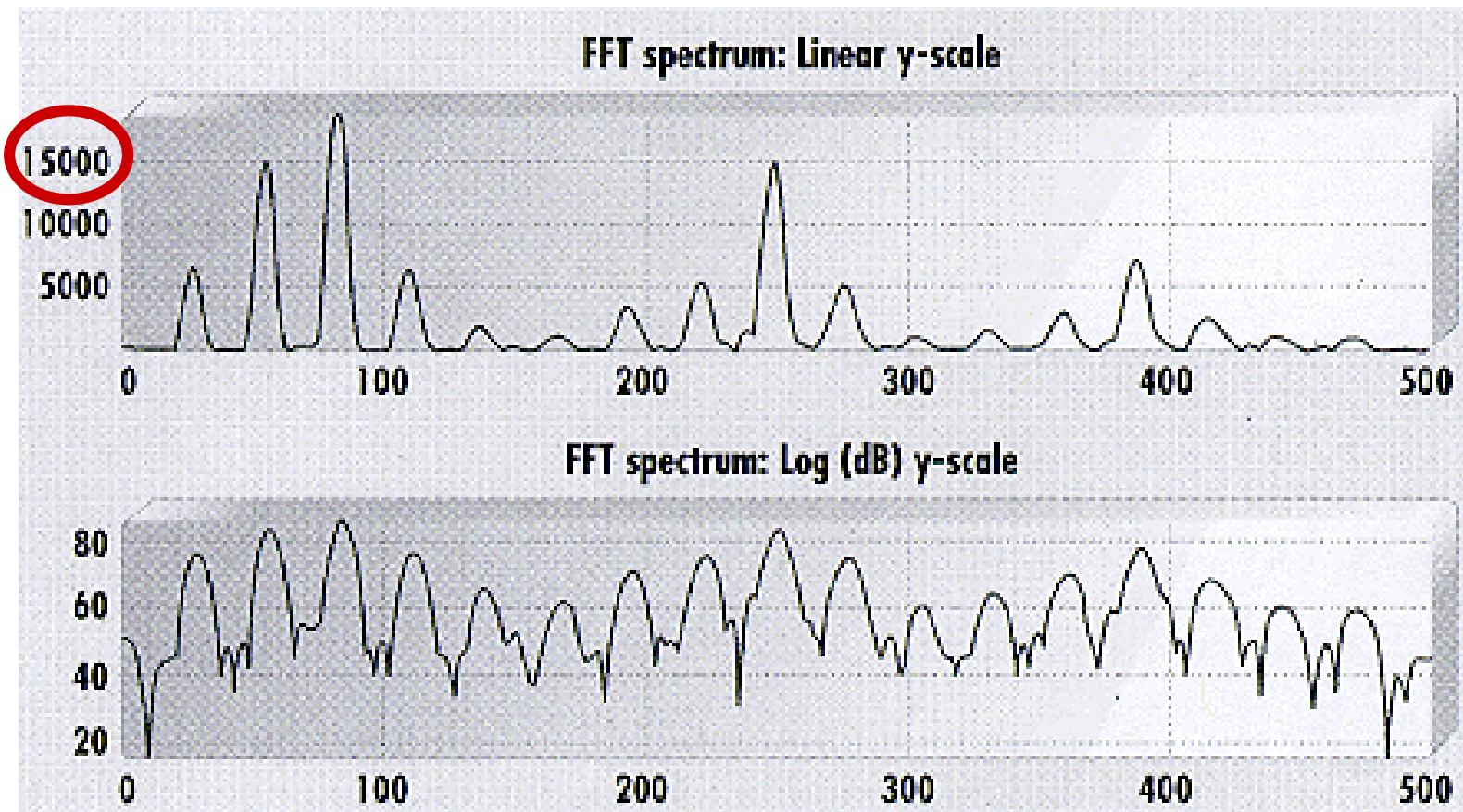
# Frekvenčijska analiza govora (1)

- Odsječak 30 ms bezvučni suglasnik "s"



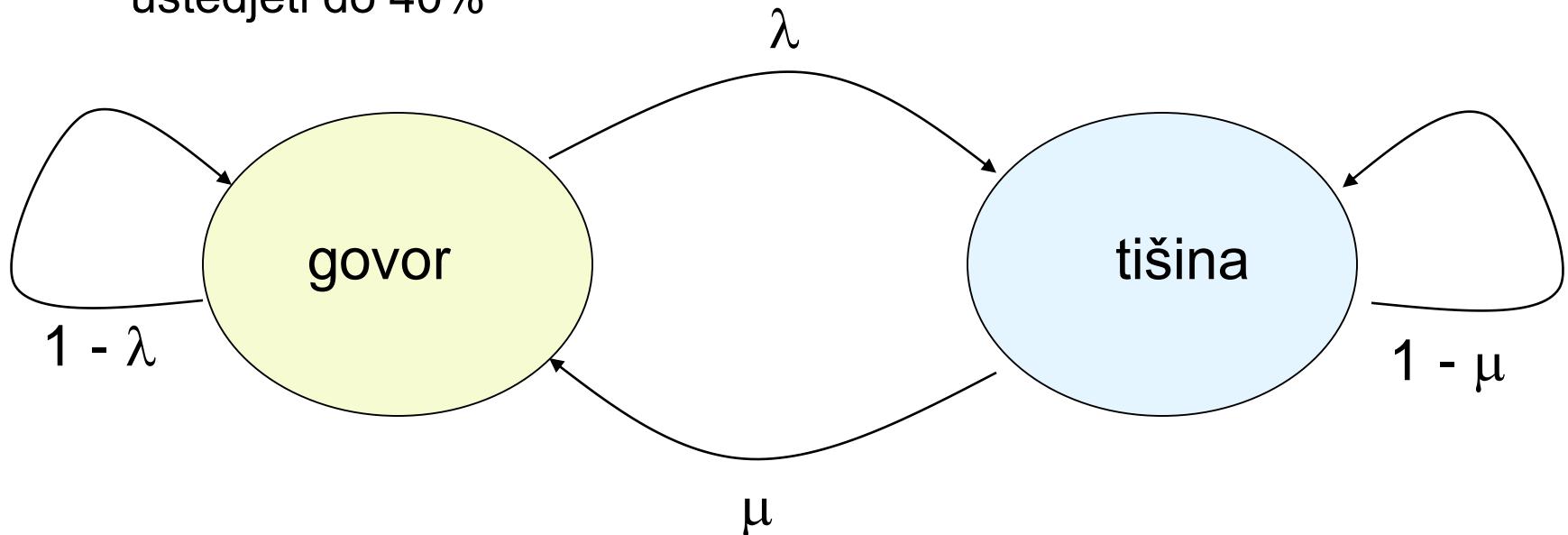
# Frekvenčijska analiza govora (2)

- Odsječak 30 ms zvučni suglasnik "n"
- Uočljive rezonantne frekvencije - **formanti**



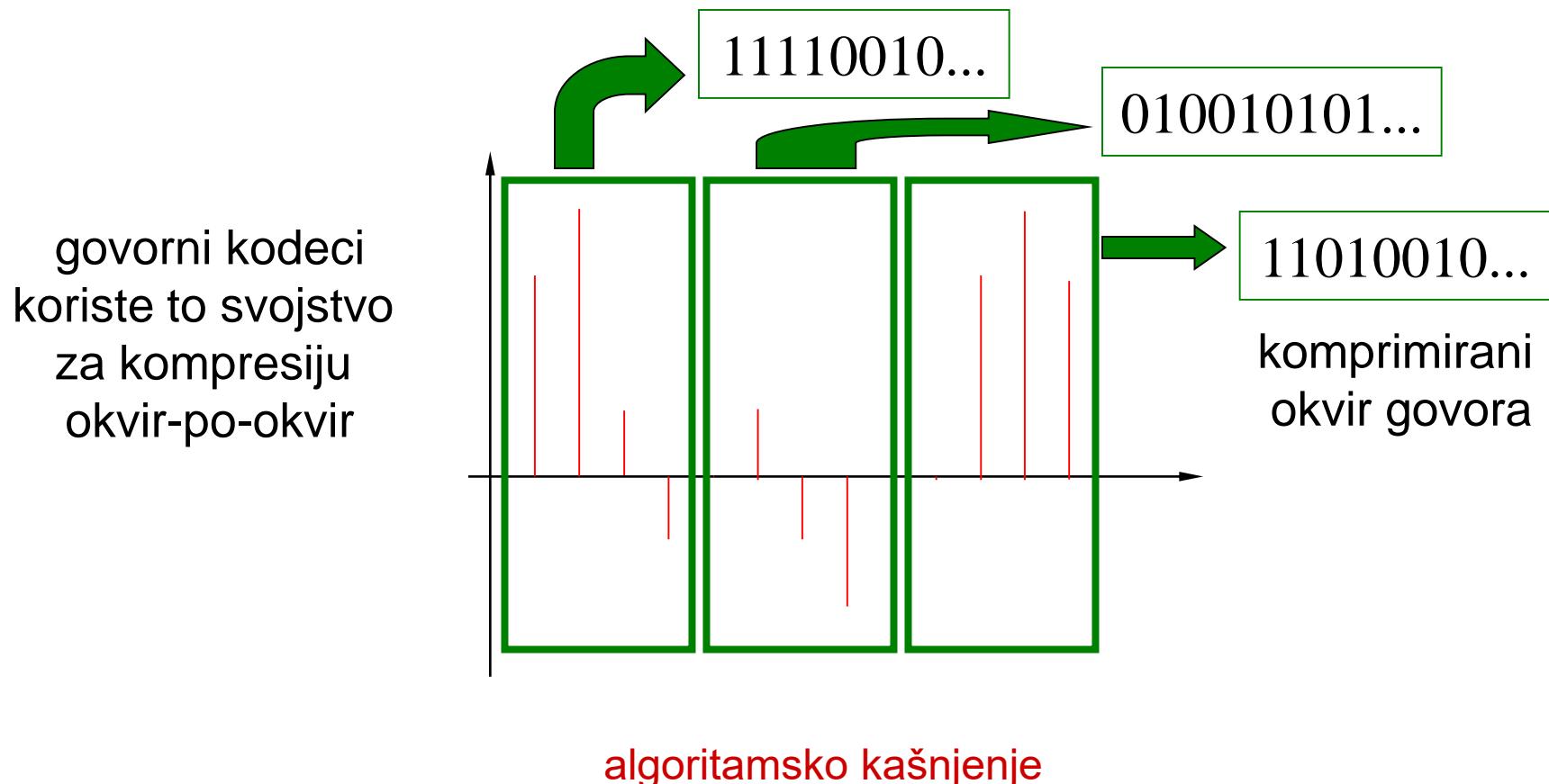
# Model govora u vremenu

- model govora u vremenu može se opisati “on-off” modelom
- intervali govora prosječno traju 800 ms - 1.2 s
- intervali tišine (između pojedinih glasova, riječi i rečenica) prosječno traju 1 – 1.6 s
  - ako se na izlaz kodera ne šalje ništa u intervalima tišine, može se uštedjeti do 40%



# Vremenska svojstva govornog signala

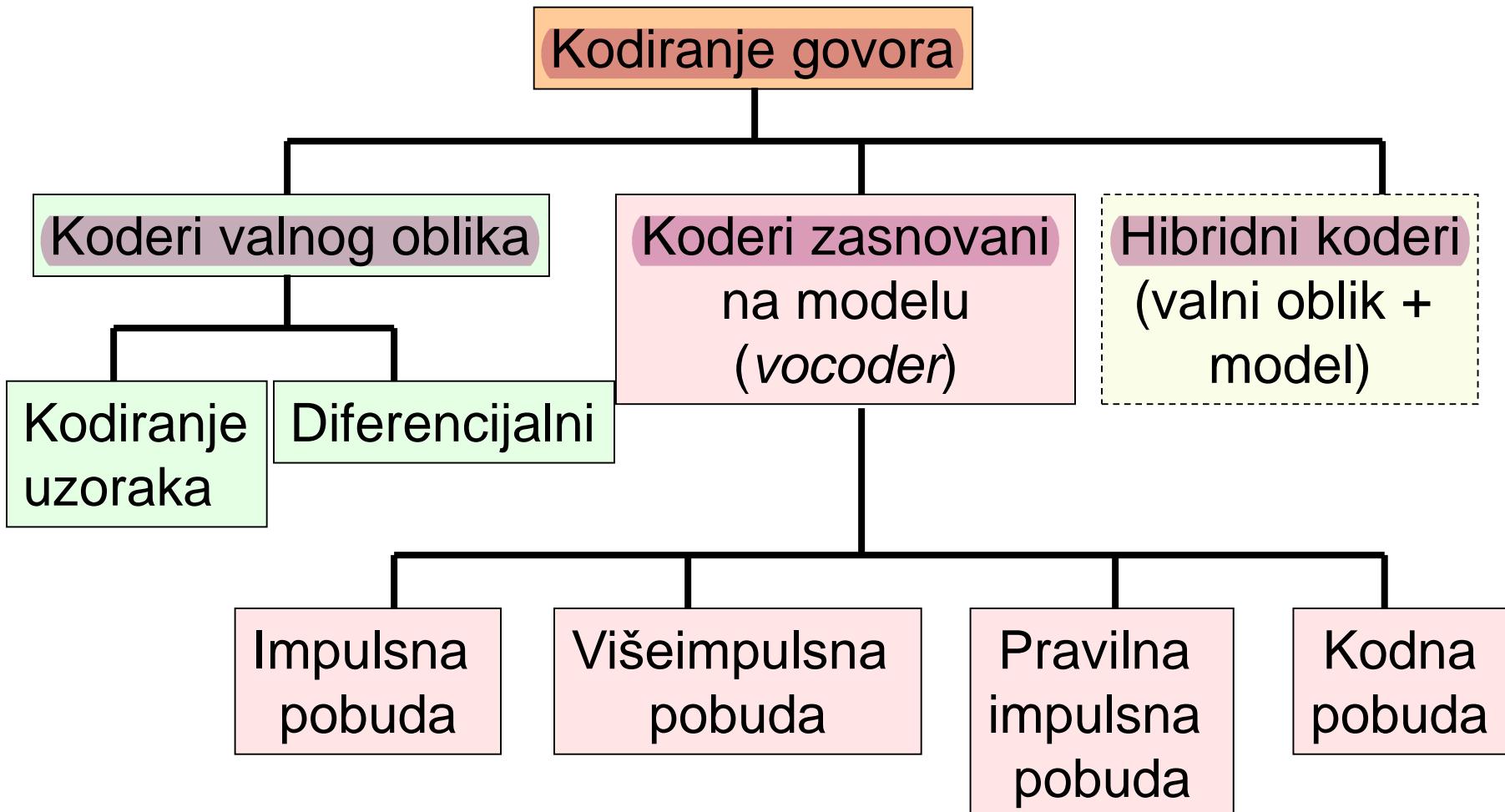
- govorni signal je nestacionaran (mijenja se u vremenu), ali u manjim vremenskim odsječcima (okvirima) od 20 - 30 ms može se promatrati kao stacionaran



# Ideje za konstrukciju codeca

- Svojstvo govornog signala je da ima veću vjerojatnost poprimanja manjih vrijednosti nego većih vrijednosti
  - Uniformna kvantizacija nije optimalna
  - “isplati se” točnije kodirati manje vrijednosti od većih - nelinearna kvantizacija daje bolju kvalitetu uz jednak broj bita po uzorku
- Postoji visoka korelacija između uzastopnih uzoraka i uzastopnih okvira
  - Uklanjanjem redundancije u signalu može se sažeti zapis
- Na temelju poznavanja svojstava govora, tj. fizioloških karakteristika govornog trakta, može se napraviti model
  - Parametri modela se računaju na temelju stvarnih uzoraka
  - Prenose se samo parametri, a govor se rekonstruira (sintetizira) na temelju modela
  - U najnovijim koderima (npr. MPEG-4 SA) ova ideja proširuje se i na druge zvukove

# Koderi govora

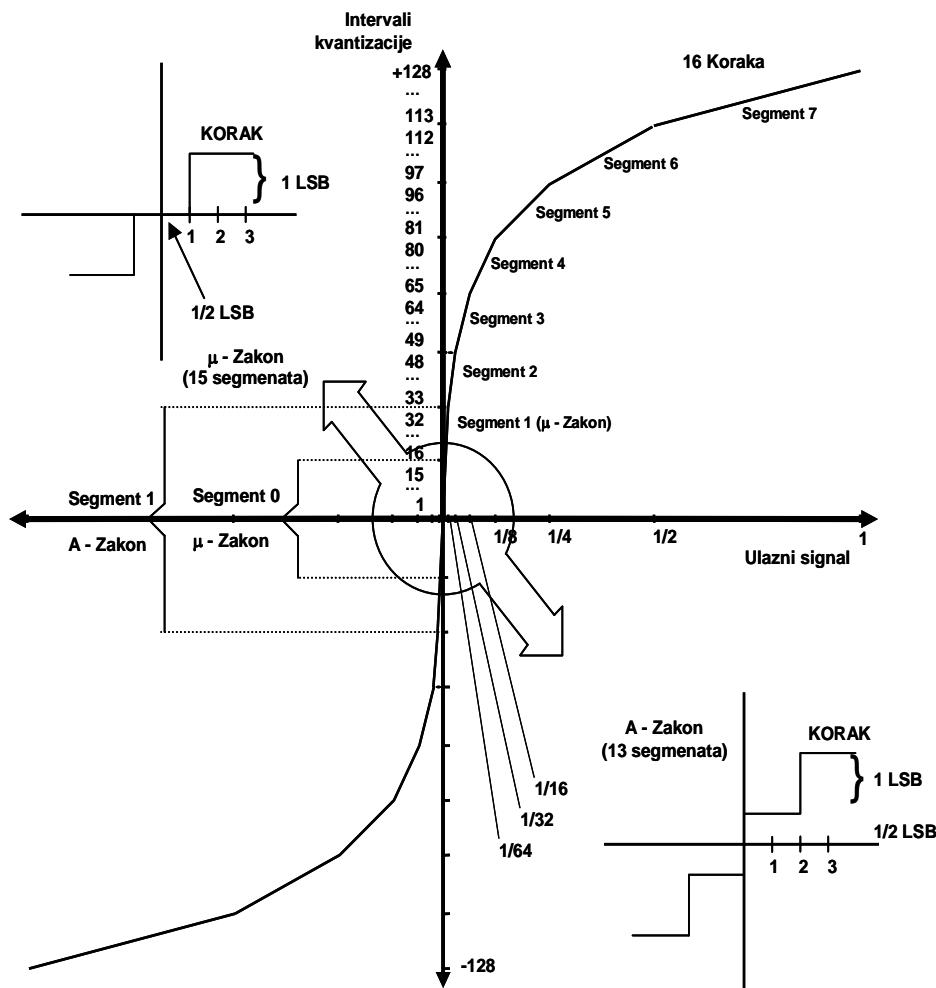


# Koderi valnog oblika

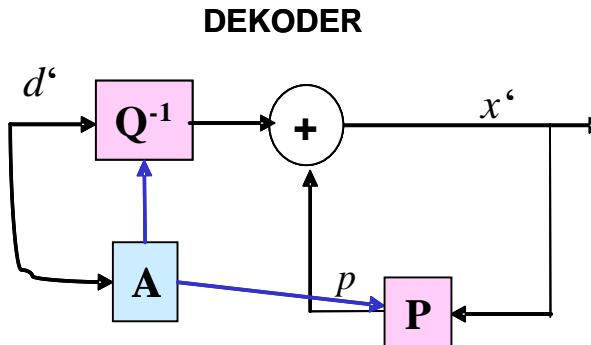
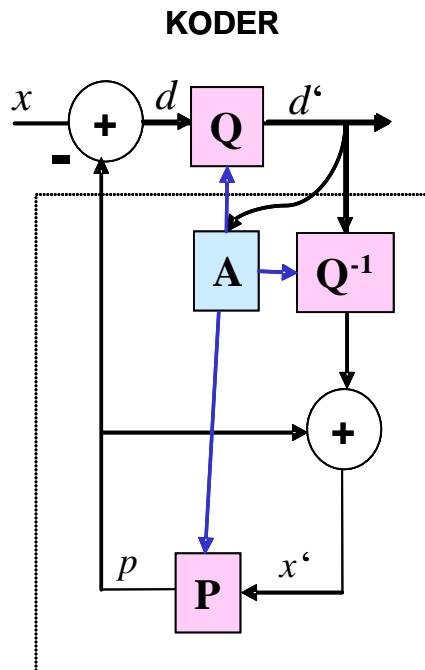
- Veće brzine, dobra kvaliteta, razvijeni za fiksnu i (kasnije dorađeni) za mobilnu telefoniju
- Pulsno-kodna modulacija (**PCM**)
  - preporuka *ITU-T G.711 Pulse Code Modulation for voice frequencies (PCM)*
- Adaptivni diferencijalni PCM (**ADPCM**)
  - preporuka *ITU-T G.726 Adaptive Differential Pulse Code Modulation (ADPCM)*; sadrži zastarjelu preporuku G.721 (originalna norma)
  - proširenje je *ITU-T G.727 5-, 4-, 3- and 2 bits per sample embedded Adaptive Differential Pulse Code Modulation (ADPCM)*

# PCM

- Uzorkovanje na 8 kHz, nelinearna kvantizacija po logaritamskoj karakteristici prema A-zakonu (Europa) ili  $\mu$ -zakonu (SAD, Japan)
- Prednosti:
  - jednostavan
  - visoka kvaliteta (MOS 4.3)
  - malo kašnjenje (1 uzorak)
- Nedostaci:
  - 64 kbit/s nije malo
  - nema mehanizme za kontrolu i ispravljanje pogrešaka (nije dobar kandidat za internetsku telefoniju)
- Primjena: već desetljećima u uporabi u fiksnoj telefonskoj mreži



# Princip diferencijalnog kodera (ADPCM)



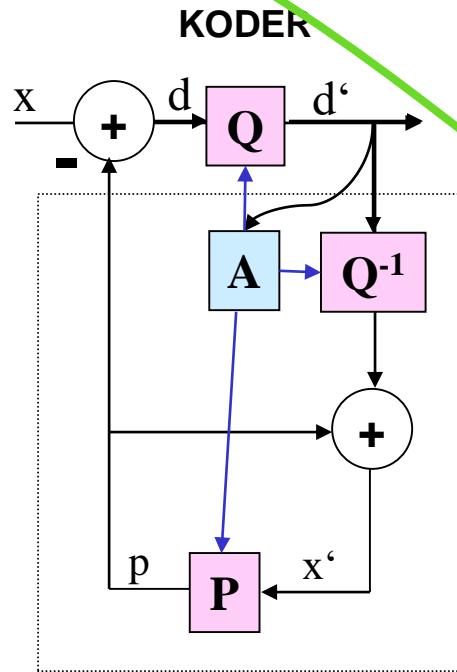
$x$  - ulazni signal  
 $d$  - signal razlike  
 $d'$  - kvantizirani signal razlike  
 $x'$  - rekonstruirani signal  
 $p$  - predvidjeni signal

$\mathbf{Q}$  – fiksna ili adaptivna kvantizacija  
 $\mathbf{Q}^{-1}$  – inverzna kvantizacija  
 $\mathbf{P}$  – linearna predikcija (fiksni ili adaptivni koeficijenti)  
 $\mathbf{A}$  – postupci adaptacije

$$p_n = a_1 x'_{n-1} + a_2 x'_{n-2} + \dots + a_k x'_{n-k}$$

U koderu je sadržan dekoder, te se računa razlika izmedju signala kojeg bi dekoder predviđao i stvarnog signala; ova razlika se kvantizira i šalje dekoderu.

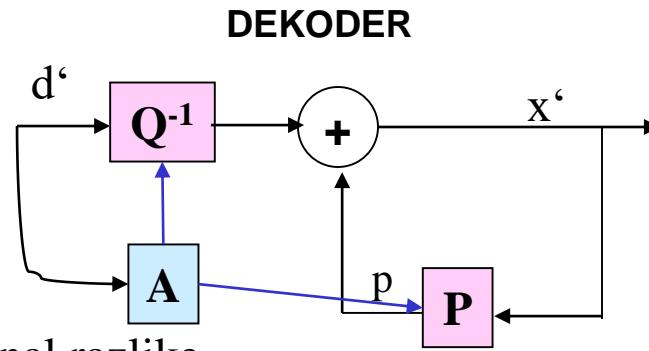
# Princip diferencijalnog kodera (ADPCM)



$x$  - ulazni signal  
 $d$  - signal razlike  
 $d'$  - kvantizirani signal razlike  
 $x'$  - rekonstruirani signal  
 $p$  - predvidjeni signal  
**Q** – fiksna ili adaptivna kvantizacija  
**Q<sup>-1</sup>** – inverzna kvantizacija  
**P** – linearna predikcija (fiksni ili adaptivni koeficijenti)  
**A** – postupci adaptacije

$$p_n = a_1 x'_{n-1} + a_2 x'_{n-2} + \dots + a_k x'_{n-k}$$

U koderu je sadržan dekoder, te se računa razlika izmedju signala kojeg bi dekoder predviđio i stvarnog signala; ova razlika se kvantizira i šalje dekoderu.



# ADPCM - svojstva

- osim tipične brzine od 32 kbit/s, ovisno o broju bita za kodiranje greške, norma specificira i brzine 40 kbit/s (5 bita), 24 kbit/s (3 bita), 16 kbit/s (2 bita)
- prednosti:
  - nema algoritamskog kašnjenja
  - prenosi i modemske i fax signale bez degradacije
- nedostaci:
  - visoka brzina (postoje bolja rješenja na manjim brzinama), osjetljiv na gubitke
- primjena: kućni bežični telefon - norma DECT (Digital European Cordless Telephony)
- primjena i u širokopojasnom koderu; preporuka *ITU-T G.722 Wideband (7 kHz) audio codec by Subband ADPCM (SB-ADPCM)*
  - 64 (56, 48) kbit/s
  - primjena: ISDN aplikacije, telekonferencija

# Koderi zasnovani na modelu

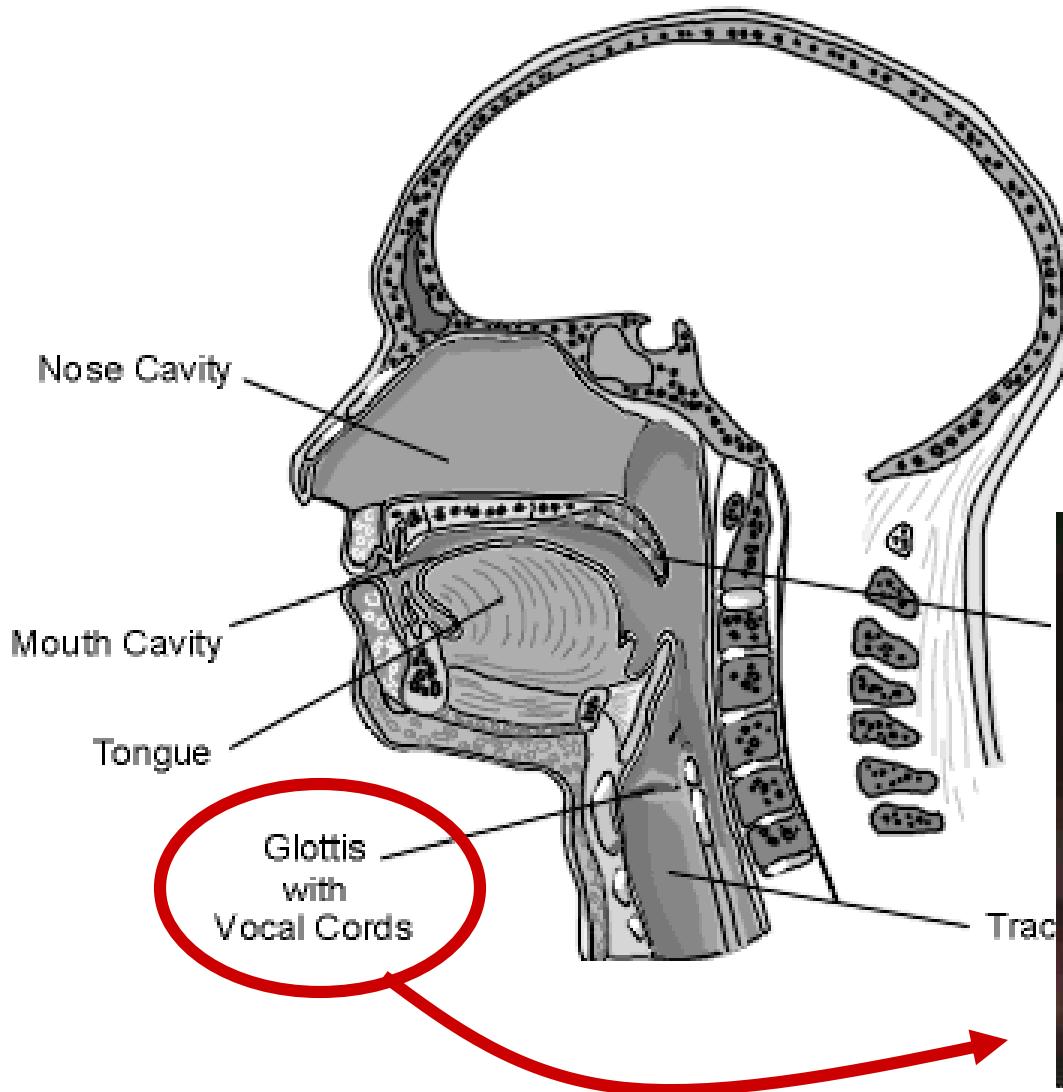
- Ideja: koder i dekoder imaju isti (parametrizirani) model govornog trakta
  - Parametri modela se računaju za okvire uzoraka govora
  - Dekoderu se prenose parametri modela (a ne uzorci govora) te se govor sintetizira na odredištu
  - Princip analize/sinteze
- postižu se vrlo male brzine
- prvi koderi, npr. LPC-10, su bili lošije kvalitete, razvijeni za sustave ograničene namjene, npr. robotika, sigurna telefonija
- noviji koderi, npr. CELP na malim brzinama postižu dobru kvalitetu, ali su računski složeniji

# Malo povijesti

- Kempelen Farkas / Wolfgang von Kempelen (1734-1804)
  - „govorni stroj“ – govorni trakt simuliran pomoću cijevi
  - Prvi zabilježeni pokušaj umjetne proizvodnje govora
- Homer Dudley, Bell Labs, 1939
  - „*Channel vocoder*“ (voice coder)
  - Model govornog trakta s filterima
  - Prethodnik današnjih kodera govora, postavlja osnovne principe

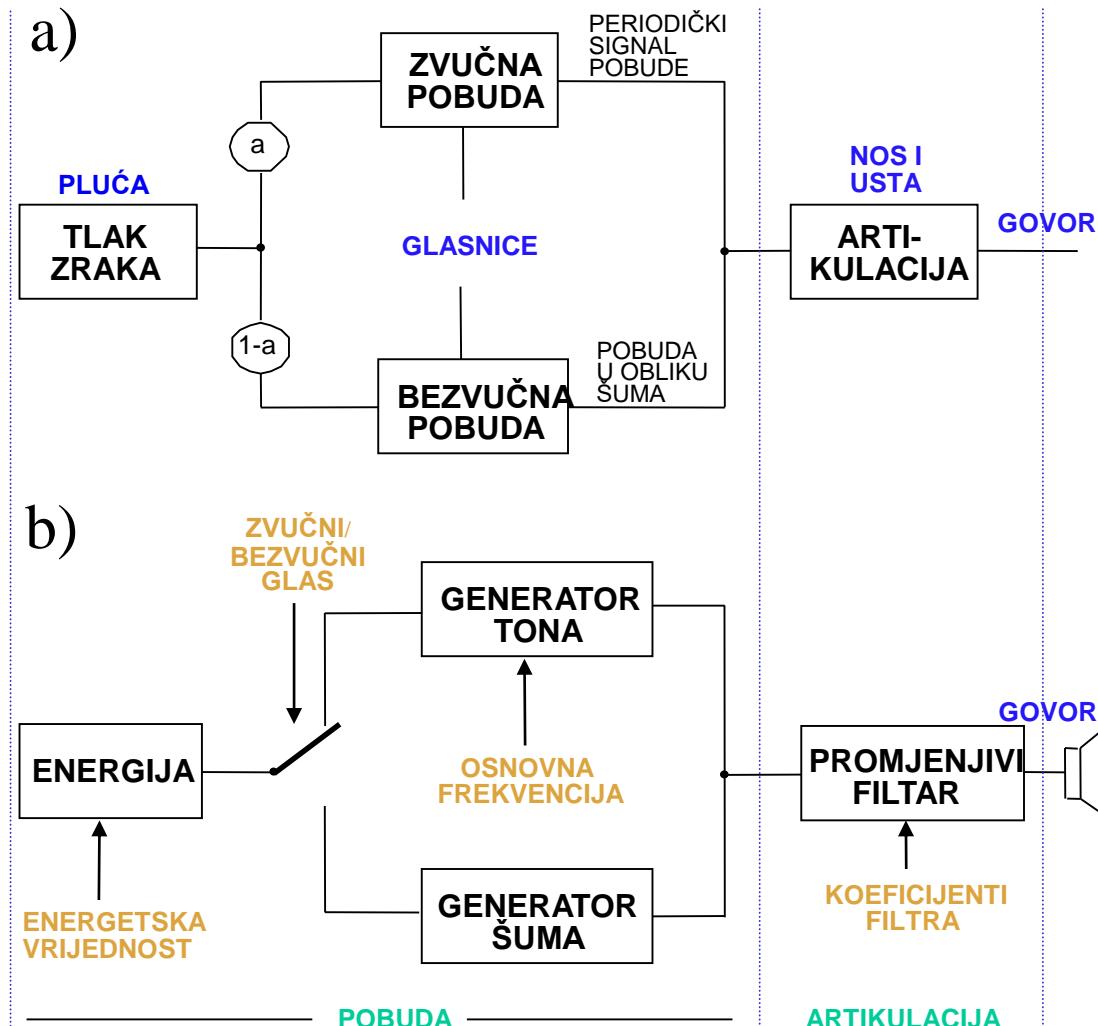


# Ljudski govorni organi



# Model proizvodnje govora

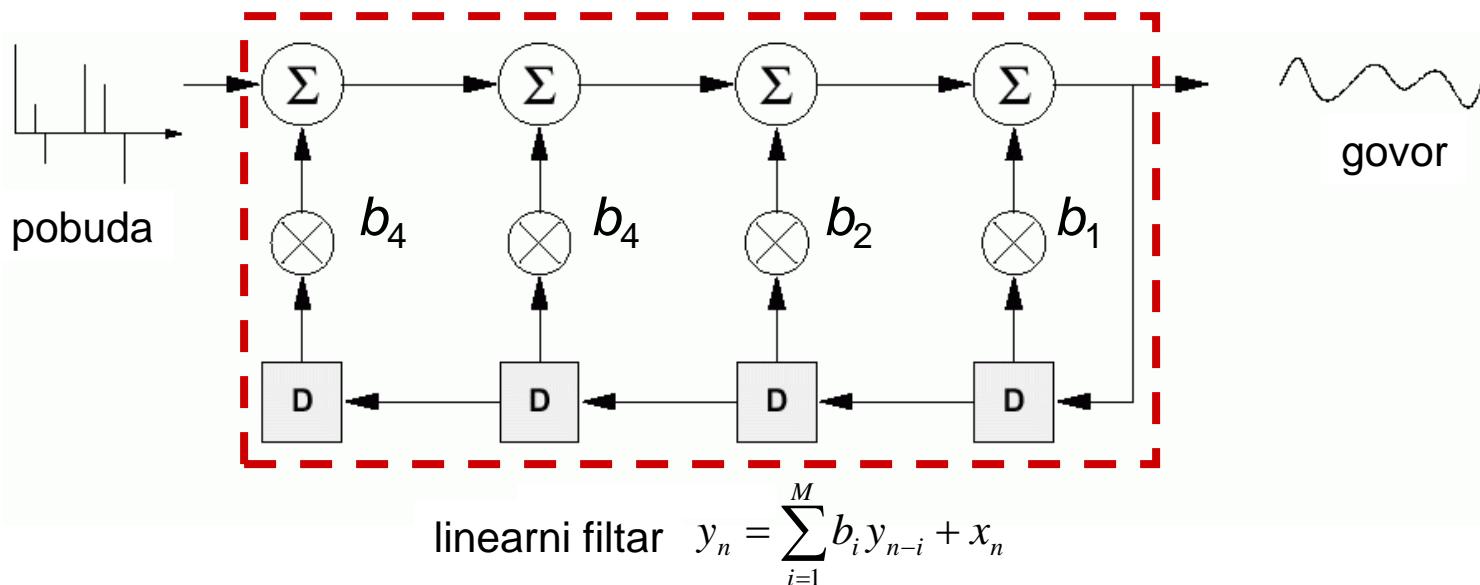
a) blok dijagram  
ljudskih  
govornih organa



(LPC - Linear  
Predictive Coder)

# Model govornog trakta

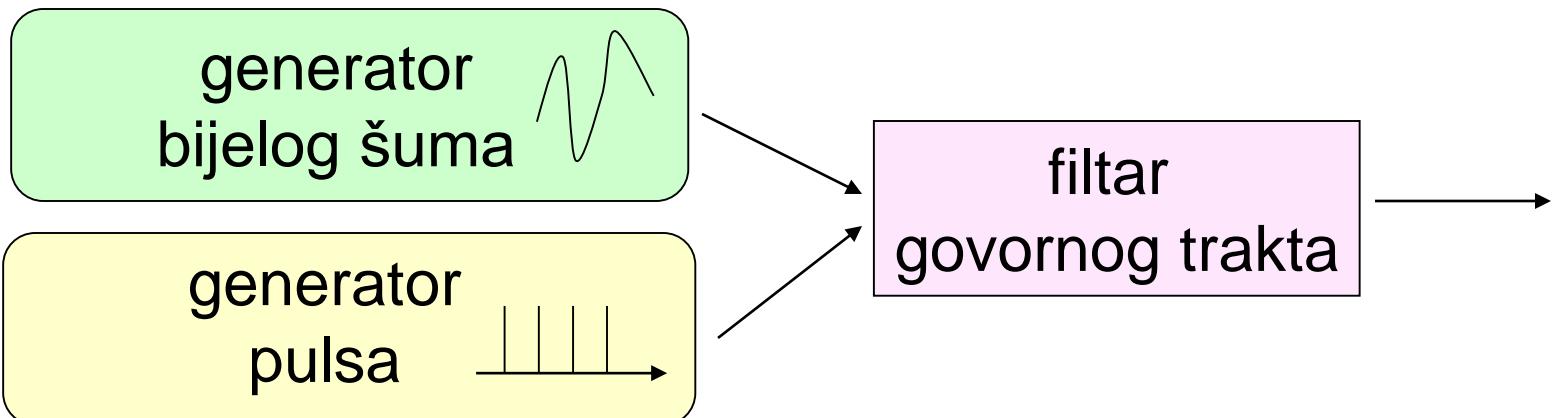
- Govorni trakt se modelira linearnim filtrom s nizom koeficijenata
- Signal se može prikazati kao izlaz linearog filtra uz zadanu *pobudu*



- Koder izračunava parametre filtra i pobude i šalje ih dekoderu

# Linear Predictive Coding (LPC)

bezvučni  
glasovi,  
tišina



- Pobuda: periodički impulsni signal

- Parametri modela:**

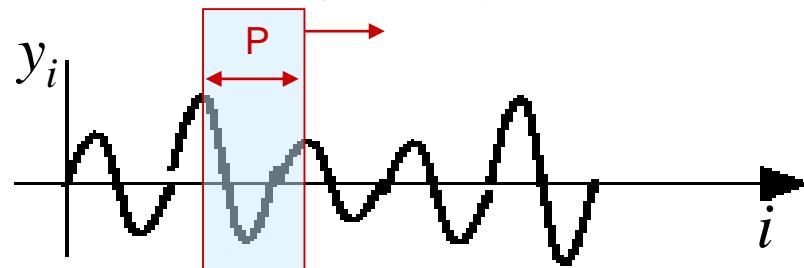
- Frekvencija pobude
  - Jačina pobude
  - Zvučni/bezvučni glas
  - Koeficijenti filtra
- |                            |
|----------------------------|
| 6 bita                     |
| 5 bita                     |
| 1 bit                      |
| 42 bita (10 koeficijenata) |

- Npr, **LPC-10** na **2.4 kbit/s**
  - Frekvencija uzorkovanja: 8kHz
  - Duljina okvira: 180 uzoraka = **22.5 ms**

# LPC - računanje parametara modela

- Frekvencija pobude
  - Average Magnitude Difference Function (AMDF)

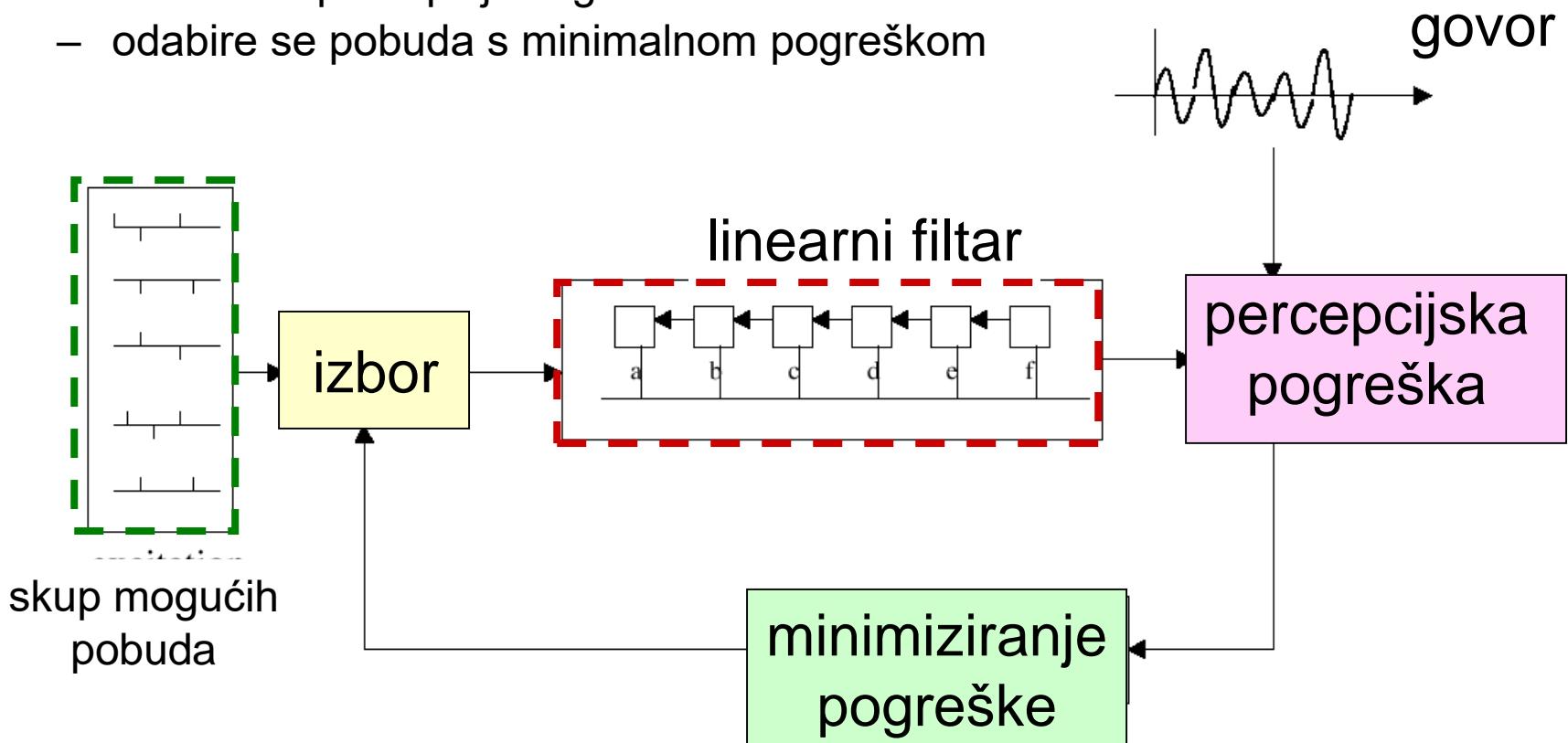
$$AMDF(P) = \frac{1}{N} \sum_{i=k_0+1}^{k_0+N} |y_i - y_{i-P}|$$



- AMDF izračunava prosjek razlika signala u zvučnom okviru udaljenih za neki period P
- Zvučni/bezvučni glas
  - AMDF nema jasnih minimuma za bezvučne glasove
  - Bezvučni glasovi: manja amplituda signala
- Jačina pobude
  - korijen srednje vrijednosti kvadrata signala
- Koeficijenti filtra
  - Metoda najmanjeg kvadrata pogreške

# Code Excited Linear Prediction (CELP)

- Zajednički rječnik kodova (code-book) u koderu i dekoderu
- Dekoderu se šalje indeks (kôd) pobude
- Analiza-sintezom služi za određivanje pobude
  - pretražuje se skup mogućih pobuda i za svaku provodi sinteza
  - računa se percepcijska greška
  - odabire se pobuda s minimalnom pogreškom



# Primjeri kodera zasnovanih na modelu (1)

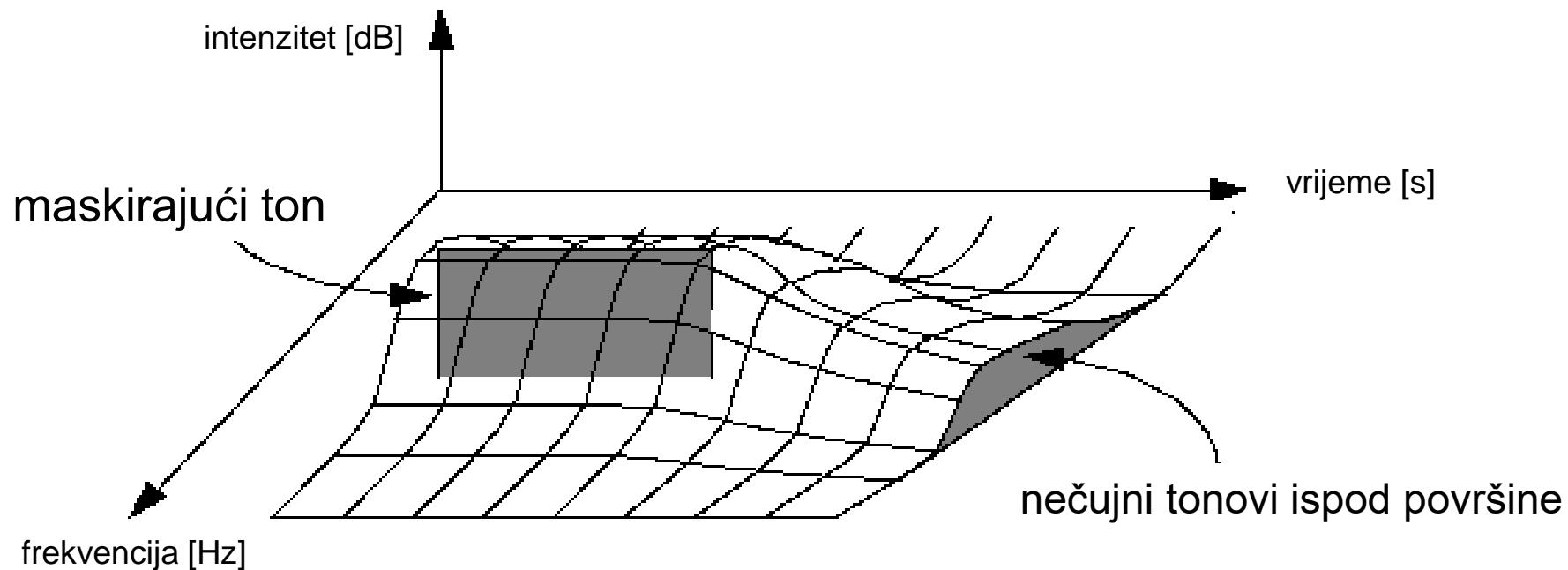
- preporuka *ITU-T G.728 Low Delay CELP (LD-CELP)*
  - 16 kbit/s, MOS 4, algoritamsko kašnjenje samo 0,625 ms
- preporuka *ITU-T G.729 Conjugate Structure Algebraic CELP (CS-ACELP)*
  - 8 kbit/s, MOS 4, kašnjenje 15 ms
- preporuka *ITU-T G.723.1 Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*
  - 5.3 i 6.3 kbit/s, MOS 3.8
  - pogodan za internetsku telefoniju

# Primjeri kodera zasnovanih na modelu (2)

- ETSI GSM 06.10: Full Rate codec na 13 kbit/s
  - u uporabi u većini GSM 900 and PCS 1800 mreža
  - Regular Pulse Excitation LPC with Long Term Prediction (RPE-LTP) koder
- ETSI GSM 06.60: GSM Enhanced Full Rate na 12.2 kbit/s
  - Algebraic Code-Excited Linear Prediction (ACELP)
  - osnova i za sjevernoamerički TDMA IS-136, kao i za ITU-T G.729
- ETSI GSM 06.20: GSM Half Rate na 5.6 kbit/s
  - Vector-Sum Excited Linear Prediction (VSELP)

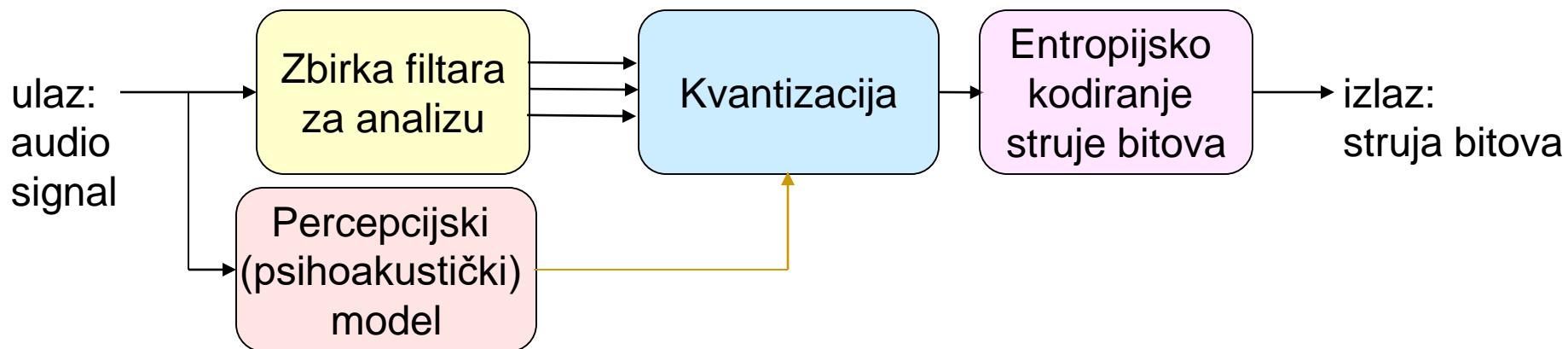
# Koderi zvuka u frekvencijskoj domeni

- Koderi zvuka u frekvencijskoj domeni nisu ograničeni na govor; imaju dobra svojstva za bilo kakve zvukove npr. glazbu
  - Koriste pod-pojasno kodiranje (podjela na frekvencijske pod-pojase prije kodiranja)
- Koriste efekt maskiranja
  - uz jaki signal na nekoj frekvenciji uho ne može čuti slabiji signal na bliskoj frekvenciji
  - osim u zadanim trenutku, maskiranje ima utjecaj i u vremenu (prije/poslije)

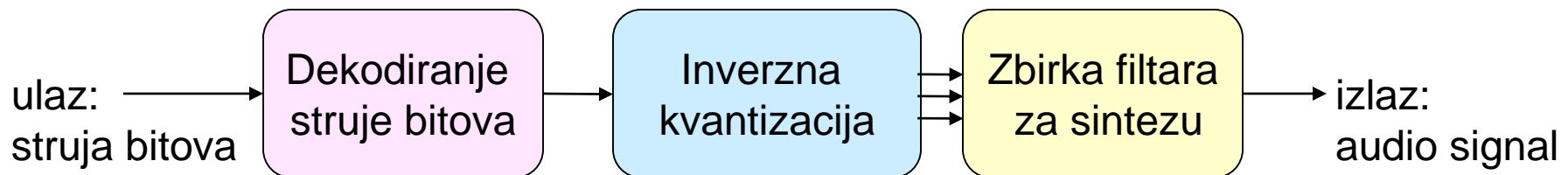


# Osnovni percepcijski koder

## Shema kodera



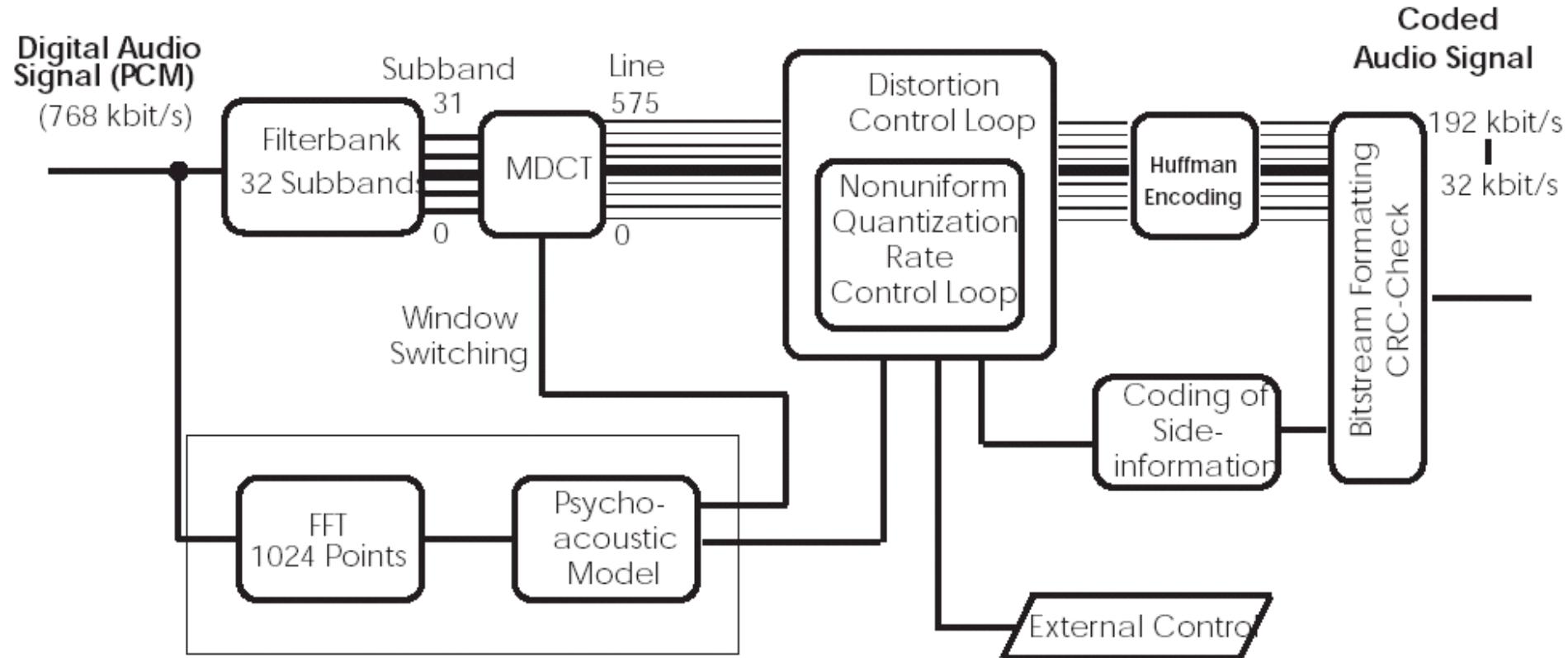
## Shema dekodera



# MPEG Audio

- MPEG - Moving Picture Expert Group ([ISO/IEC JTC1/SC29/WG11](#))
- **MPEG-1**
  - Dva audio kanala
  - $f_u = 44.1 \text{ kHz}$  (isto kao CD), 32 kHz, 48 kHz (isto kao DAT)
  - Brzine od 8-16 kbit/s do 320 kbit/s
  - MPEG Audio Layer I, II, III: razine kodiranja rastuće složenosti
  - Koristi se percepcijsko kodiranje
- **MPEG-2**
  - Isti osnovni koder kao MPEG-1
  - Pet audio kanala + niskofrekventni kanal
  - Uz MPEG-1 još i  $f_u = 16 \text{ kHz}$ , 22.05 kHz, 24 kHz
  - AAC – Advanced Audio Codec, dodan kasnije
- **MPEG-4**
  - AAC, dva kodera za govor, strukturirani audio, ...

# MPEG Audio Layer 3 – MP3 koder



- Literatura: K. Brandenburg “MP3 and AAC Explained”, AES 17th international conference on High Quality Audio Coding, Florence, Italy, September 1999.

# Pregled kodera govora i zvuka

<b>Kodiranje</b>	<b>kb/s</b>	<b>MOS</b>	<b>uporaba</b>
LPC-10	2.4	2.3	robotika, sigurna telefonija
G.711	64.0	4.5	telefonija (A, u-law)
G.722	64.0		7kHz codec (subband)
G.723.1	5.3/6.3	3.8	videotelefon (ostatak kanala za video)
G.726	16-40	ovisi	niska kompleksnost (ADPCM)
G.726	32	4.1	niska kompleksnost (ADPCM)
G.728	16.0	4.0	malo kašnjenje
G.729	8.0	4.0	mobilna telefonija
GSM	13.0	3.5	GSM (mobilna telefonija, Europa)
GSM EFR	12.2	4.0	GSM 2.5G
GSM HR	5.6	3.5	GSM 2.5G
IS 54/136	7.95	3.5	TDMA (Sj. Amerika mobile (stari std.))
IS 641	7.4	4.0	TDMA (Sj. Amerika mobile (novi std.))
MPEG L3	56-128.0	N/A	CD stereo
DVI	32.0	toll-quality	(Intel, Microsoft)
16 bit/44.1 kHz	1411		compact disc

# LPC demo

- Pokazati na predavanju



Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

5.

Informacijska svojstva i  
kodiranje nepomične slike

# Informacijska svojstva i kodiranje nepomične slike

- Izvori digitalne slike
- Ljudski vid i percepcija slike
- Računalni prikaz slike
- Principi kodiranja slike
- Kodiranje bez gubitaka
- **Kodiranje sa gubicima**
  - Diferencijalno kodiranje
  - Transformacijsko kodiranje: JPEG
  - Kodiranje valićima: JPEG 2000
  - Fraktalno kodiranje

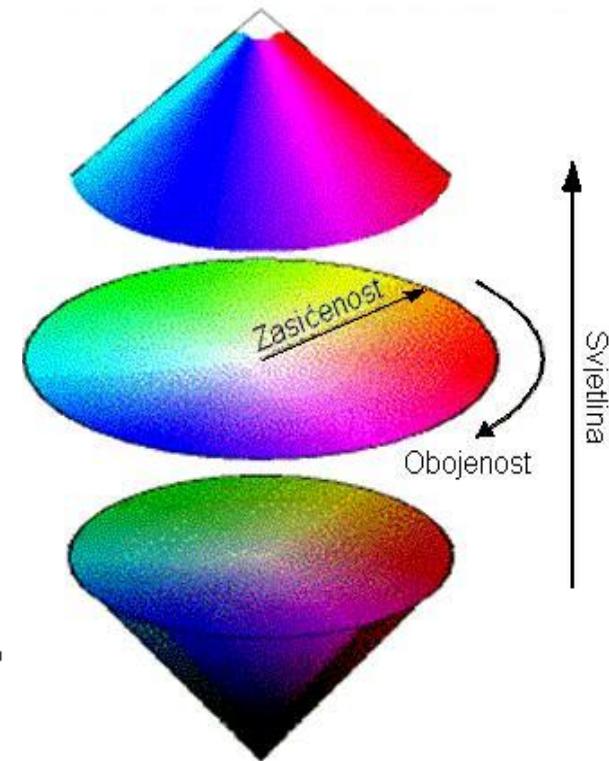
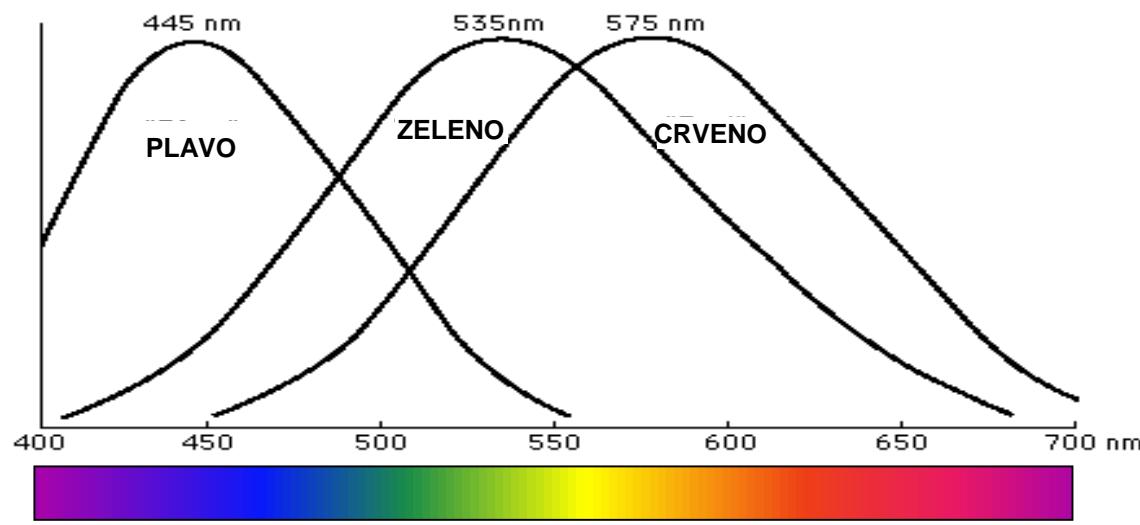
# Digitalna (nepomična, 2D) slika

---

- Slika iz stvarnog svijeta (u digitalnom obliku):
  - slika snimljena digitalnom kamerom
  - slika prenesena na računalo preko optičkog čitača (scanner)
  - ...
- Slika stvorena pomoću računala:
  - crtež (vektorska grafika)
  - slika (obrađena digitalna slika, bitmap, računalna grafika)
  - fraktalna slika
  - graf
  - vizualizacija
  - ...
- Slika kao jedinica unutar animiranog filma ili videa

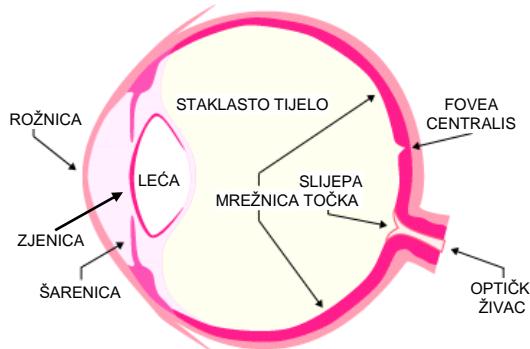
# Percepcija slike

- Vidljiva svjetlost = elektromagnetski val
  - $\lambda = 330 - 770 \text{ nm}$ ,  $c \approx 3 \times 10^8 \text{ m/s}$   
 (ovisno o gustoći medija)



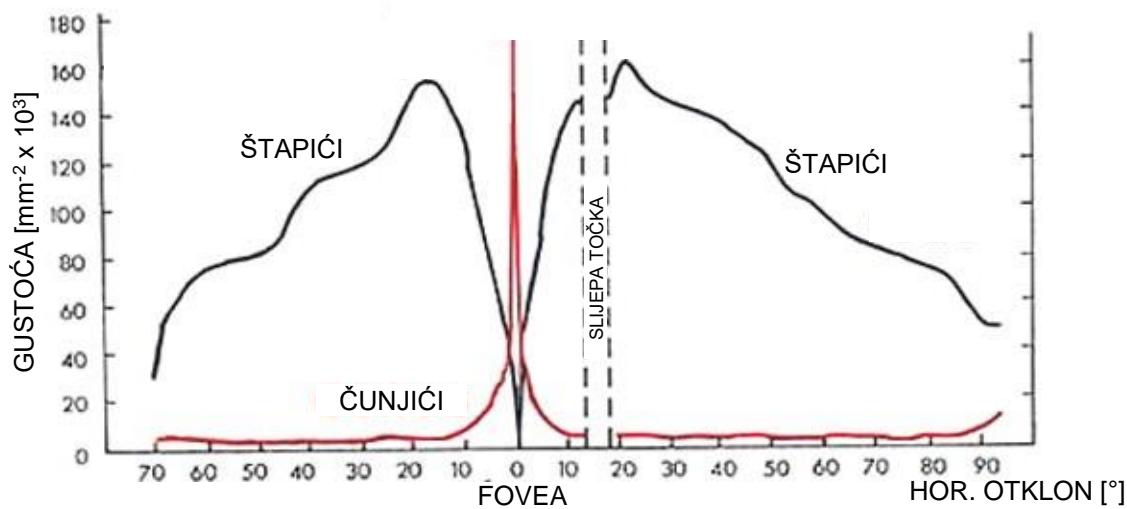
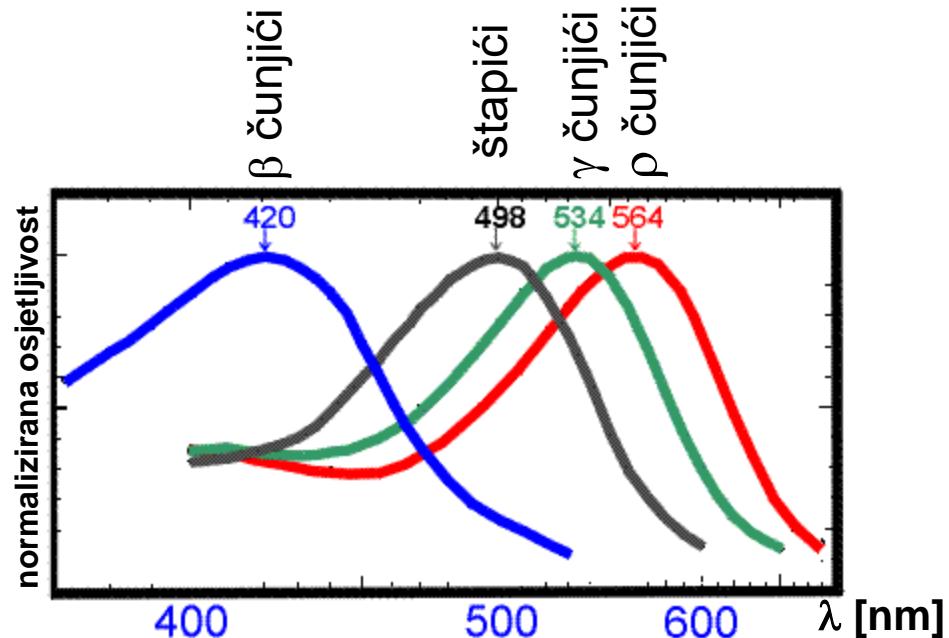
- tri svojstva boje koje ljudsko oko raspoznaće su **obojenost** (*hue*), **zasićenost** (*saturation*), i **svjetlina** (*brightness*)
  - to je osnova za HSB model boja u računalnoj grafici

# Ljudski vid



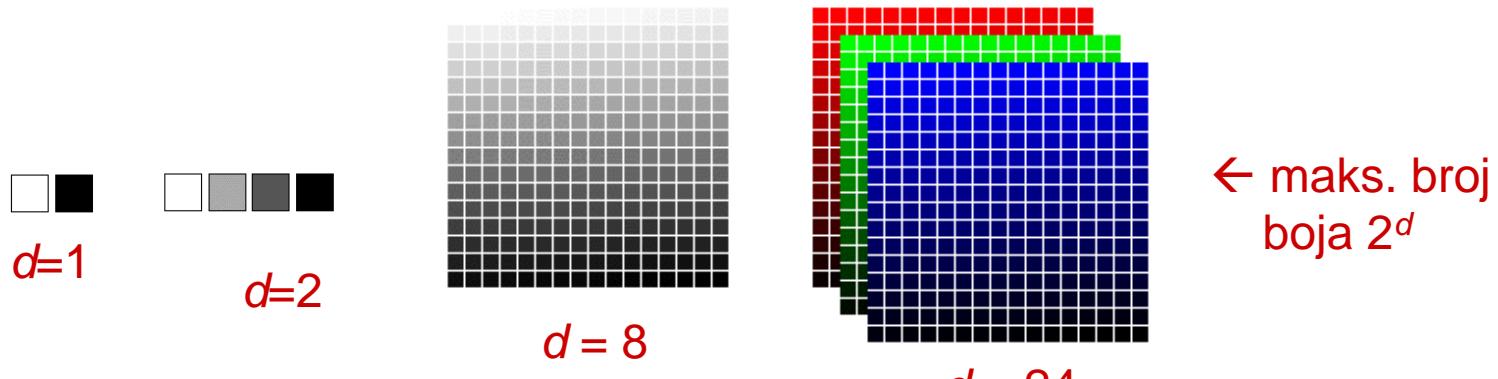
## receptori u ljudskom oku:

- štapićaste stanice (osjetljive na svjetlost)
- čunjaste stanice (osjetljive na boju)
  - crveno  $\rho$
  - plavo  $\beta$
  - zeleno  $\gamma$



# Računalni prikaz slike

- slika se promatra kao matrica obojanih točaka, odn. *pixela*
- pixel = *picture element*
  - razlikujemo *pixel slike* od “*pixela uređaja*” (*device pixel, dot*)!! Npr:
    - printer 600 dpi: kvadrat sa stranicom  $1/600$
    - video monitor 72 dpi: kvadrat sa stranicom  $1/72$
- rezolucija slike = dimenzije matrice pixela  $N_1 \times N_2$
- dubina slike = broj bita ( $d$ ) za opis pixela (odn. boju)



- boja = varijabla koja opisuje pixel
  - modeli boje za sliku: RGB, CMY, HSB, HSI, ...
  - modeli boje za video: YUV, YIQ, ...

# Modeli boje za sliku

- 

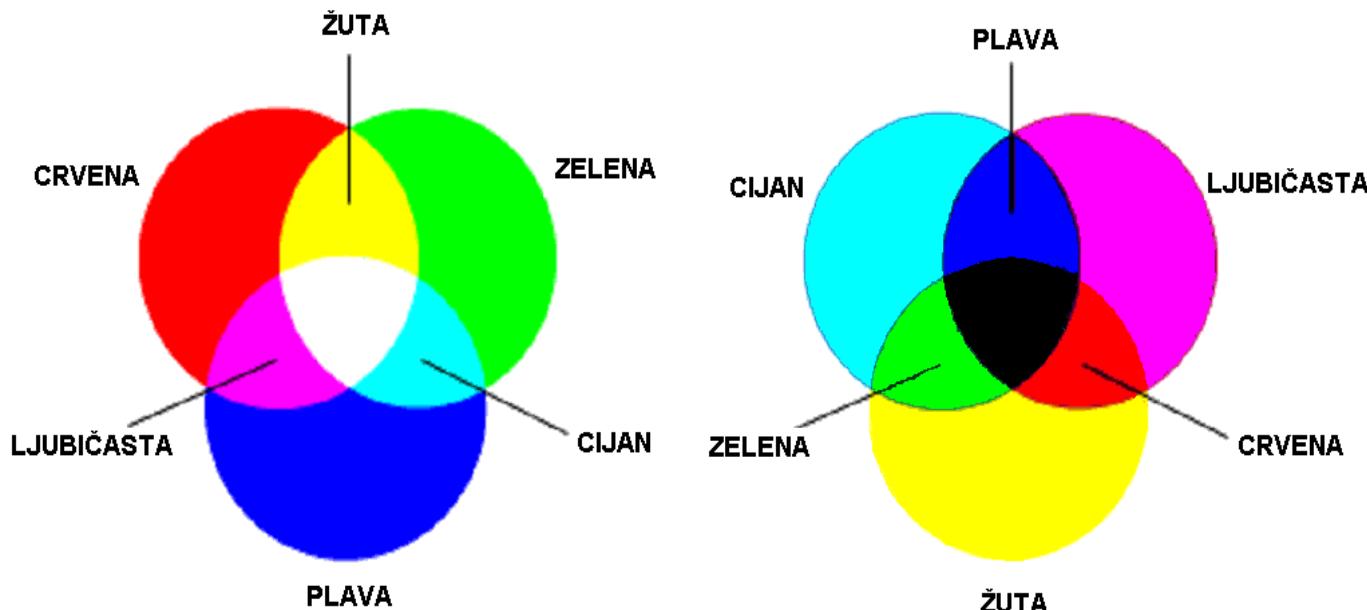
## RGB:

- Red – Green – Blue
- (Crvena – Zelena - Plava)
- svjetlo, zbrojive boje
- TV ekran, monitor u boji

- 

## CMY:

- Cyan, Magenta, Yellow
- (Cijan – Ljubičasta - Žuta)
- pigment, oduzimljive boje
- pisač



# Norme za kodiranje nepomične slike

---

- Brojne vlasničke i otvorene norme, npr:
  - BMP (Bitmap)
  - XBM (X11 Bitmap)
  - GIF (Graphics Interchange Format)
  - PNG (Portable Network Graphics)
  - TIFF (Tagged Image File Format)
  - **JPEG** (Joint Photographic Expert Group)
  - JPEG 2000

# Kodiranje slike zasniva se na:

---

- Statističkim karakteristikama slike: kodiranje bez gubitaka
- Karakteristikama ljudskog sustava vida: kodiranje s neprimjetnim gubicima
- Sažimanju manje važnih elemenata slike prema nekom kriteriju: kodiranje s vidljivim gubicima
- Obično se radi o kombinaciji ovih ideja

# Kodiranje bez gubitaka

- Koriste se metode entropijskog kodiranja (vidi prethodna predavanja)
- Slijedno kodiranje
  - Telefax (starija verzija)
- Huffman kodiranje
  - Telefax
- LZW metoda (metode rječnika)
  - GIF (Graphics Interchange Format) – bez gubitaka ako se koristi do 256 boja; pogodan za računalnu grafiku

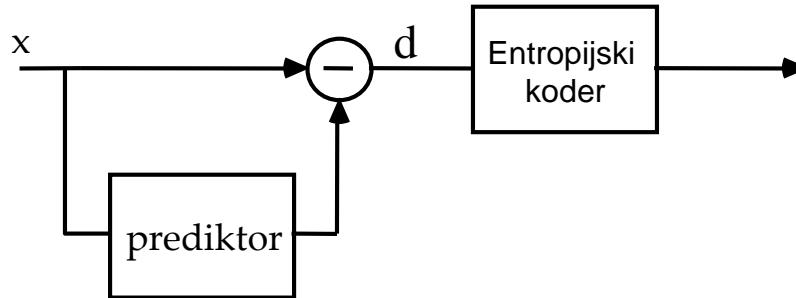
# Kodiranje sa gubicima

- Diferencijalno (prediktivno) kodiranje
- Transformacijsko kodiranje
  - JPEG
- Kodiranje valićima (wavelets)
  - JPEG 2000
- Fraktalno kodiranje
- Osnovna ideja:



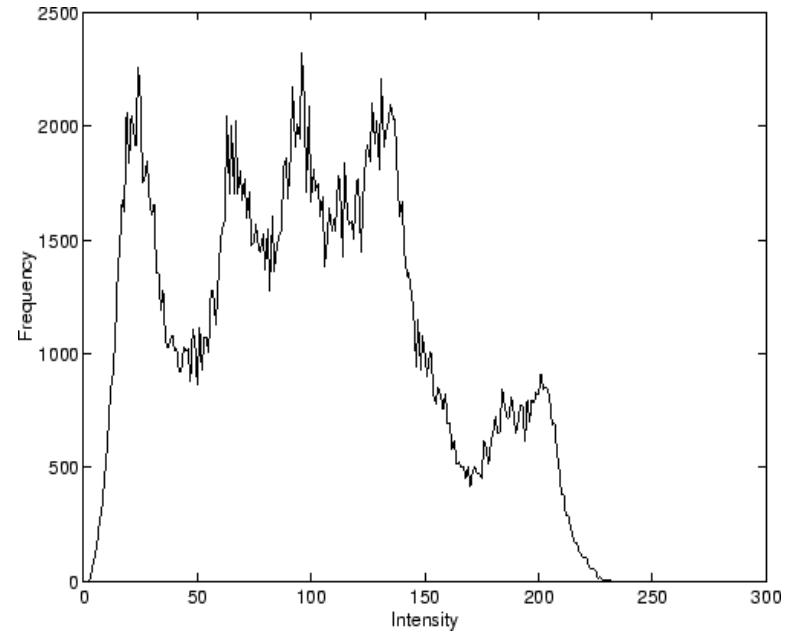
# Diferencijalno (prediktivno) kodiranje

- Princip: vrijednost slijedećeg signala (pixela) predviđa se iz dosadašnjih vrijednosti, te se kodira razlika stvarnog i predviđenog



- Raspon amplituda diferencijalnog signala je povoljniji za kodiranje od originalne slike

# Primjer: slika i njen histogram

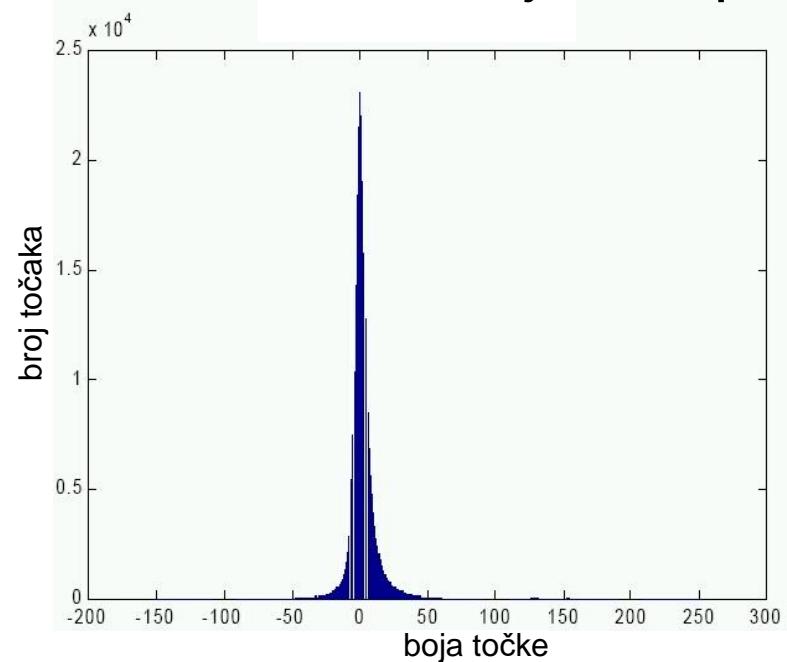
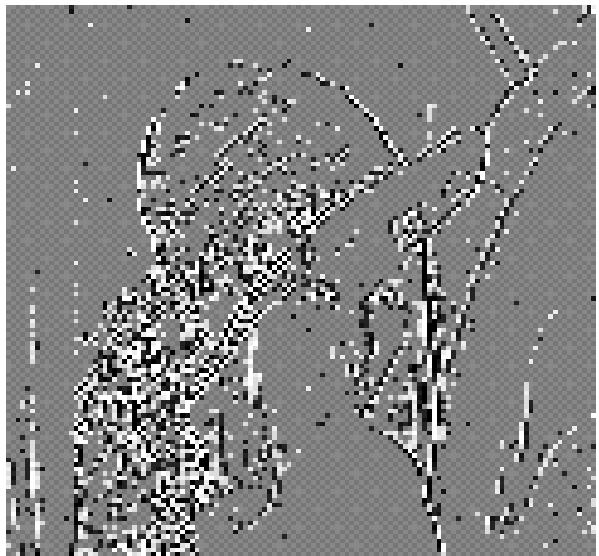


256\*256\*8 bita

Direktnim entropijskim kodiranjem može se postići 7 bit/pixel

# Signal razlike

- Prikazan je signal razlike za sliku iz primjera uz jedan stupanj predviđanja – svodi se na razliku susjednih pixela



- Signal se ujednačuje, povoljnije za entropijsko kodiranje
- Entropijskim kodiranjem dobiva se 2.6 bit/pixel!

# Svojstva diferencijalnog kodiranja

---

- + Jednostavna implementacija
- + Može biti bez gubitaka (ovisno o tome da li se upotrebljava kvantizacija)
- Postiže se relativno slaba kompresija

# Transformacijsko kodiranje

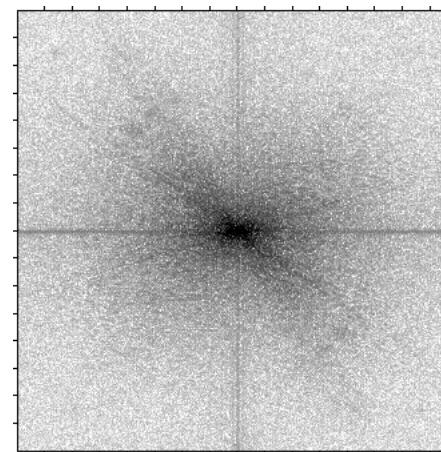
---

- Signal se transformira u prostorno frekvencijske komponente, te se one kodiraju
- Neke frekvencijske komponente slike pojavljuju se puno više od ostalih, što rezultira dobim kodiranjem
- Metoda razvijena 70-tih godina
- Široka primjena kroz normu JPEG
- JPEG = Joint Photographic Experts Group

# Primjer: Fourierova transformacija



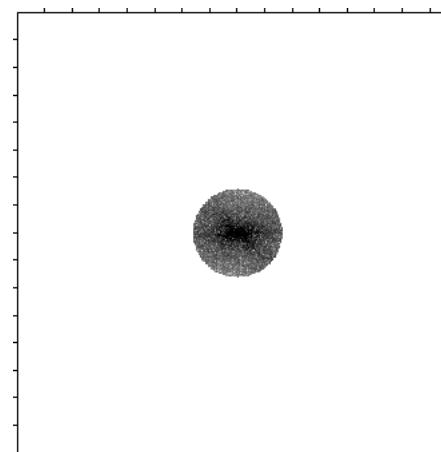
2D DFT  
→



↓ Reduction

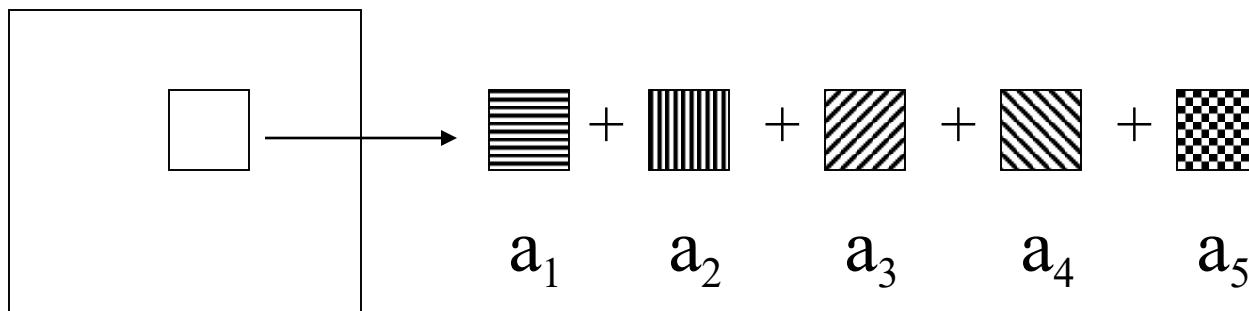


2D IDFT  
←



# Transformacijsko kodiranje u praksi: JPEG

1. Slika se dijeli na blokove veličine  $8 \times 8$  točaka
2. Svaki blok se aproksimira sumom osnovnih DCT blokova, svaki sa svojim koeficijentom doprinosa
3. Koeficijenti doprinosa svakog osnovnog bloka se kodiraju.



Za potpunu rekonstrukciju potrebno je onoliko osnovnih blokova koliko ima točaka u bloku:  $8 \times 8 = 64$

# Jednodimenzionalna DCT

- DCT – diskretna kosinusna transformacija
- Promatramo jednodimenzionalnu sliku kao vektor dimenzije  $N$  pixela
- Za pixel na mjestu  $x$ ,  $0 \leq x < N$ ,  $p(x)$  označava razinu sivog kodiranu s 8 bita (0 – bijelo, 255 – crno)



$$p = [ p(0) \ p(1) \ p(2) \ \dots \ p(x) \ \dots \ p(N-1) ]$$

- Promatranu sliku možemo prikazati kao zbroj DCT funkcija s određenim težinama

$$p(x) = \sum_{f=0}^{N-1} S(f) \cdot DCT_f(x) \quad [1]$$

# Osnovne DCT funkcije

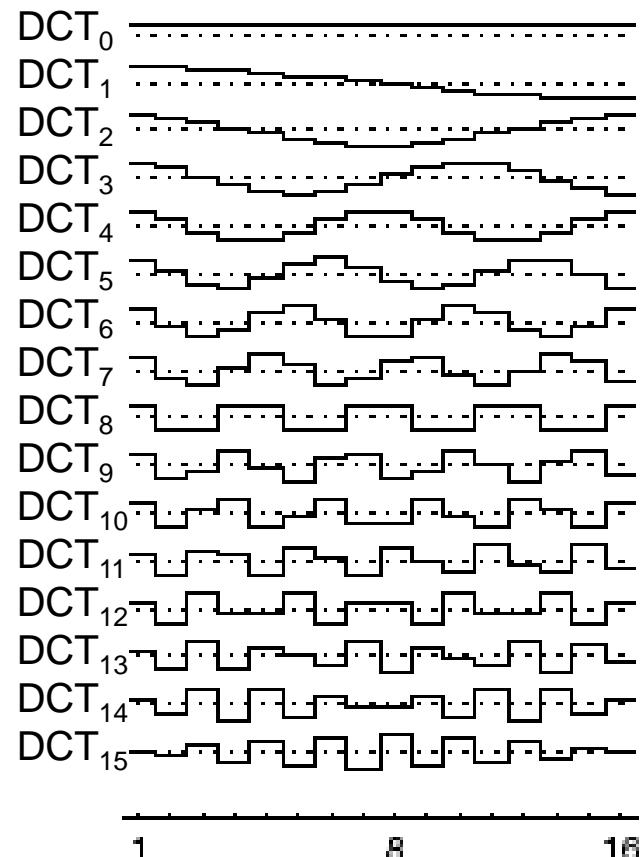
- U jednoj dimenziji:

$$DCT_f(x) = \sqrt{\frac{2}{N}} \cos\left[\frac{(2x+1) \cdot \pi \cdot f}{2N}\right], f > 0$$

$$DCT_0(x) = \sqrt{\frac{1}{N}}$$

- Primjer: N=16

- Uvrštavanjem u [1]:



$$p(x) = S(0) \sqrt{\frac{1}{N}} + \sum_{f=1}^{N-1} S(f) \cdot \sqrt{\frac{2}{N}} \cos\left[\frac{(2x+1) \cdot \pi \cdot f}{2N}\right] \quad [2]$$

# Svojstva osnovnih kosinusnih funkcija

---

- Cjelovitost
  - Težinski se zbroj ovih funkcija može naći za bilo koju kombinaciju od N piksela.
- Minimalnost
  - Niti jedna funkcija se ne može predstaviti zbrojem ostalih, tj. svih N je potrebno za cjelovitost.
- Jedinstvenost
  - Niti jedan drugi skup kosinus funkcija osim onih koje su u razmjeru s korištenim se ne može koristiti za opis bilo koje kombinacije od N pixela.

# Računanje koeficijenata

- Da bismo sliku prikazali DCT funkcijama, moramo izračunati koeficijente  $S(f)$  za  $0 \leq f \leq N-1$
- Prvi član u zbroju niza je srednja vrijednost, iz čega slijedi izraz za član  $S(0)$ :

$$\frac{S(0)}{\sqrt{N}} = \frac{1}{N} \sum_{x=0}^{N-1} p(x) \Rightarrow S(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} p(x)$$

- Kada to uvrstimo u prethodni izraz, dobivamo za koeficijente  $S(f)$ :

$$S(f) = \sqrt{\frac{2}{N}} \cdot C(f) \cdot \sum_{x=0}^{N-1} p(x) \cdot \cos \left[ \frac{(2x+1)\pi f}{2N} \right] \quad C(f) = \begin{cases} 1/\sqrt{2}; & f = 0 \\ 1; & f > 0 \end{cases} \quad [3]$$

- Matrica DCT koeficijenata  $S(f)$  opisuje sliku u frekvencijskoj domeni.

# Primjer (1/3)

- Promatramo konkretan primjer slike od  $N=8$  pixela



$$p(x) = [ 240 \ 210 \ 180 \ 150 \ 120 \ 90 \ 60 \ 30 ]$$

- Prije primjene DCT radi se posmak razina (*level shift*), tj. od svih vrijednosti u vektoru razina sivog oduzimamo 128 kako bi vrijednosti bile simetrične u odnosu na 0 (interval od -128 do 127) i tako dobivamo vektor  $p'$

$$p'(x) = [ 112 \ 82 \ 52 \ 22 \ -8 \ -38 \ -68 \ -98 ]$$

- Za  $N=8$ , koeficijente  $S(f)$  računamo prema izrazu [3] i dobivamo:

$$S(f) = [ 19.89 \ 193.27 \ 0 \ 20.20 \ 0 \ 6.03 \ 0 \ 1.52 ]$$

## Primjer (2/3)

- $S(f)$  zaokružujemo na cijele brojeve i dobivamo  $S'(f)$ :

$$S'(f) = [ \textcolor{blue}{20} \quad \textcolor{green}{193} \quad \textcolor{blue}{0} \quad \textcolor{green}{20} \quad \textcolor{blue}{0} \quad \textcolor{green}{6} \quad \textcolor{blue}{0} \quad \textcolor{green}{2} ]$$

- Za provjeru, izračunajmo sada  $p'(x)$  prema izrazu [2]:

$$p(x) = S(0) \sqrt{\frac{1}{N}} + \sum_{f=1}^{N-1} S(f) \cdot \sqrt{\frac{2}{N}} \cos \left[ \frac{(2x+1) \cdot \pi \cdot f}{2N} \right]$$

$$p'(x) = [ \textcolor{blue}{111.89} \quad \textcolor{green}{81.86} \quad \textcolor{blue}{52.29} \quad \textcolor{blue}{21.86} \quad \textcolor{blue}{-7.71} \quad \textcolor{red}{-38.15} \quad \textcolor{blue}{-67.72} \quad \textcolor{red}{-97.75} ]$$

- Nakon zaokruživanja i posmaka razina dobivamo originalnu funkciju  $p(x)$ .

$$p'(x) = [ \textcolor{blue}{112} \quad \textcolor{green}{82} \quad \textcolor{blue}{52} \quad \textcolor{green}{22} \quad \textcolor{blue}{-8} \quad \textcolor{green}{-38} \quad \textcolor{blue}{-68} \quad \textcolor{red}{-98} ]$$

$$p(x) = [ \textcolor{blue}{240} \quad \textcolor{red}{210} \quad \textcolor{blue}{180} \quad \textcolor{green}{150} \quad \textcolor{blue}{120} \quad \textcolor{green}{90} \quad \textcolor{blue}{60} \quad \textcolor{red}{30} ]$$

Ovisno o efektima zaokruživanja, DCT je samo približno reverzibilna!

## Primjer (3/3)

- Utjecaj nižih frekvencijskih komponenata:

A  $S'(f) = [ 20 \ 193 \ 0 \ 20 \ \underline{0 \ 6 \ 0 \ 2} ]$

$$p(x) = [ 240 \ 210 \ 180 \ 150 \ 120 \ 90 \ 60 \ 30 ]$$

B  $S'(f) = [ 20 \ 193 \ 0 \ 20 \ \underline{0 \ 0 \ 0 \ 0} ]$

$$p(x) = [ 238 \ 213 \ 179 \ 148 \ 122 \ 91 \ 57 \ 32 ]$$

- Vizualno:



# Svojstva slike u frekvencijskoj domeni

- Frekvencijske komponente na nižim frekvencijama su izraženije od onih na višim frekvencijama
  - Prva četiri koeficijenta u primjeru opisuju gotovo cijelu sliku
- Visoke frekvencije izražavaju zaista sitne detalje u slici
- Doprinos članova na višim frekvencijama je mali kada su razlike susjednih pixela relativno male, dakle kada slika ne sadrži puno detalja
- Za većinu slika, ovo je zaista slučaj, te se na tom svojstvu zasniva JPEG kompresija (i druge metode transformacijskog kodiranja)
- Premda smo ovo pokazali na primjeru 1D, ovo jednako važi za dvodimenzionalnu sliku

# Blok shema JPEG kodera

1. korak:  
pripreme slike

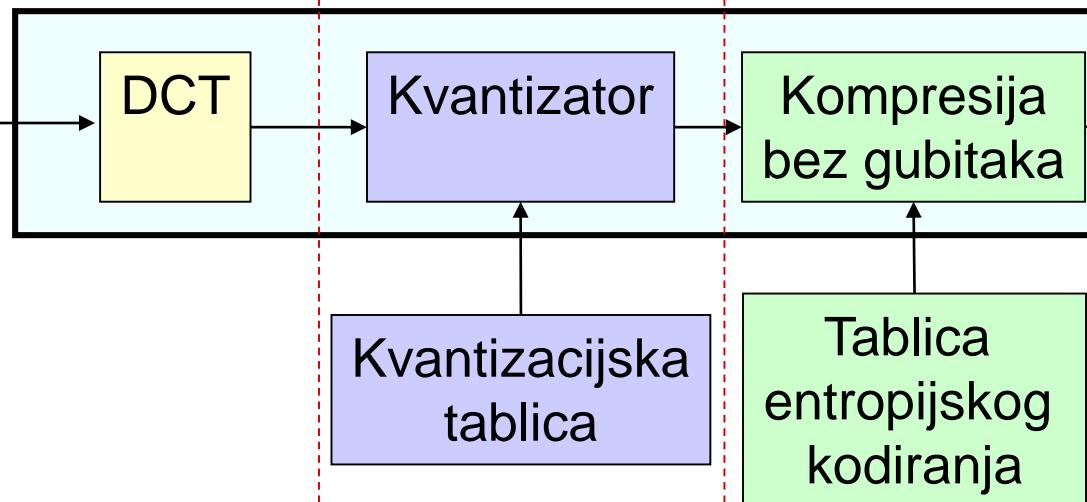
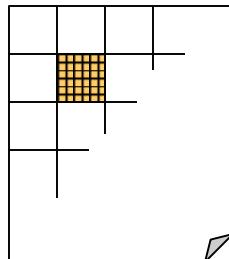
2. korak:  
obrada  
(transformacija)

3. korak:  
kvantizacija

4. korak:  
kompresija  
bez gubitaka

Koder zasnovan na DCT-u

blokovi izvorne  
slike  $8 \times 8$  pixela



$p(x, y)$

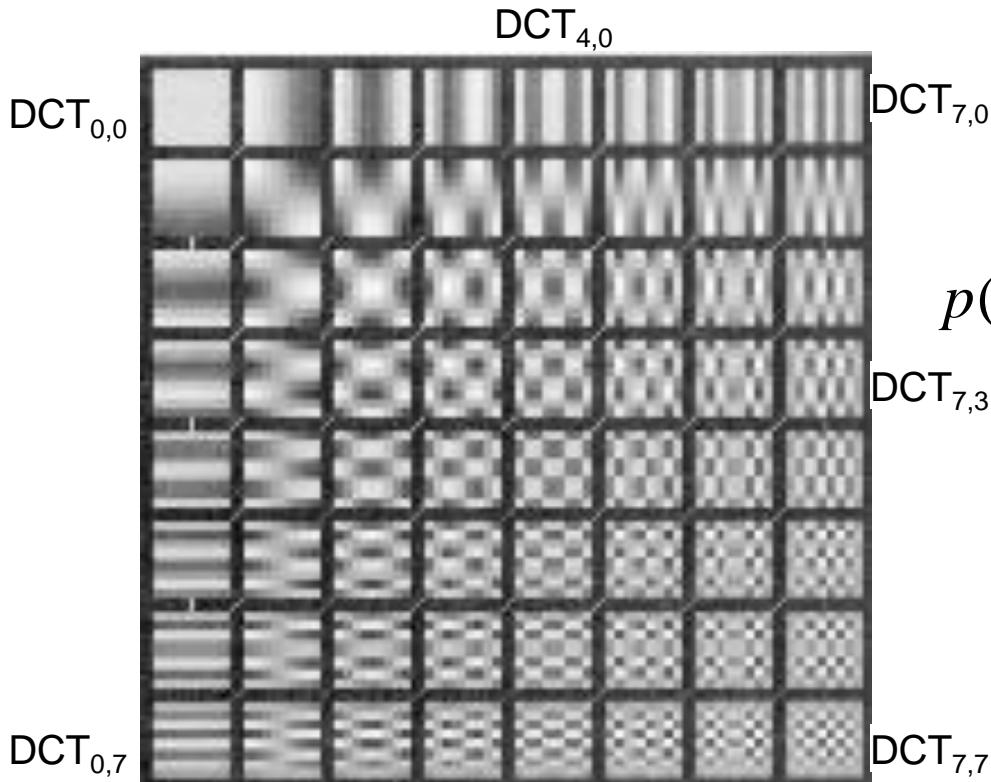
$S(u, v)$

$K(u, v)$

...100110..

# DCT transformacija slike (1/2)

- DCT funkcije u 2D dobivaju se množenjem 1D funkcija
- Osnovni blokovi DCT transformacije:



$$p(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} S(u, v) \cdot DCT_{u,v}(x, y)$$

# DCT transformacija slike (2/2)

- DCT transformacija vrši preslikavanje niza vrijednosti piksela u niz koeficijenata težine osnovnih blokova
- DCT koeficijenti za svaki blok 8x8 pixela se računaju prema formuli:

$$S(u, v) = \frac{C(u) \cdot C(v)}{4} \cdot \sum_{x=0}^7 \sum_{y=0}^7 p(x, y) \cdot \cos\left[\frac{(2x+1)\pi \cdot u}{16}\right] \cdot \cos\left[\frac{(2y+1)\pi \cdot v}{16}\right]$$

$$C(f) = \begin{cases} 1/\sqrt{2}; & f = 0 \\ 1; & f > 0 \end{cases}$$

- primjer:

x/y	0	1	2	3	4	5	6	7
0	79	75	79	82	82	86	94	94
1	76	78	76	82	83	86	85	94
2	72	Slične vrijednosti!					82	
3	74	Slične vrijednosti!					79	
4	73	70	75	67	78	78	79	85
5	69	63	68	69	75	78	82	80
6	76	76	71	71	67	79	80	83
7	72	77	78	69	75	75	78	78

Izvorišna matrica  $p(x, y)$

u/v	Najviša vrijednost $S(0, 0)$							Niske vrijednosti		
	0	1	2	3	4	5	6	7		
0	619	-29	8	2	1	-3	0	1		
1	22	-6	-4	0	7	0	-2	-3		
2	11	0	5	-4	-3	4	0	-3		
3	2	-10	5	0	0	7	3	2		
4	6	2	-1	-1	-3	0	0	8		
5	1	2	1	2	0	2	-2	-2		
6	-8	-2	-4	1	2	1	-1	1		
7	-3	1	5	-2	1	-1	1	-3		

DCT matrica  $S(u, v)$

# 3: kvantizacija

- DCT koeficijenti se kvantiziraju zaokruživanjem na cjelobrojnu vrijednost omjera matrice koeficijenata **S** i kvantizacijske matrice **Q**; nastaje kvantizirana DCT matrica **K**:

$$K(u, v) = \text{round} \left( \frac{S(u, v)}{Q(u, v)} \right) = \left\lfloor \frac{S(u, v)}{Q(u, v)} + 0.5 \right\rfloor$$

- Q** mora biti tako izabrana da kvantizacija rezultira visokom kompresijom, ali bez primjetnog gubitka kvalitete
  - norme ne određuju, ali se preporučuje skup kvantizacijskih matrica **Q**

u/v	0	1	2	3	4	5	6	7
0	16	11	10	16	24	40	51	61
1	12	12	14	19	26	58	60	55
2	14	13	16	24	40	57	69	56
3	14	17	22	29	51	87	80	62
4	18	22	37	56	68	109	103	77
5	24	35	55	64	81	104	113	92
6	49	64	78	87	103	121	120	101
7	72	92	95	98	112	100	103	99

Kvantizacijska matrica  $Q(u, v)$

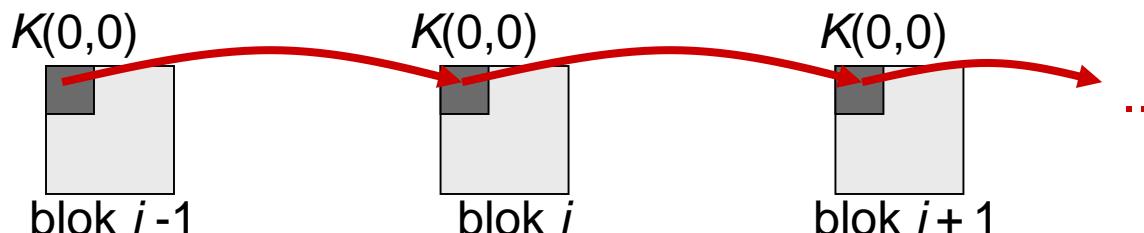
u/v	0	1	2	3	4	5	6	7
0	39	-3	1	0	0	0	0	0
1	2	-1	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0
3	0	-1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Kvantizirana DCT matrica  $K(u, v)$

Vrijednost 0

# 4: kompresija bez gubitaka

- Koeficijent  $K(0,0)$  (DC komponenta) redovito sadrži najveći dio ukupne vrijednosti bloka, tj. nosi najviše informacije o bloku kojeg predstavlja
- Korelacija između vrijednosti koeficijenata  $K(0,0)$  susjednih blokova u slici → koristi se diferencijalno kodiranje



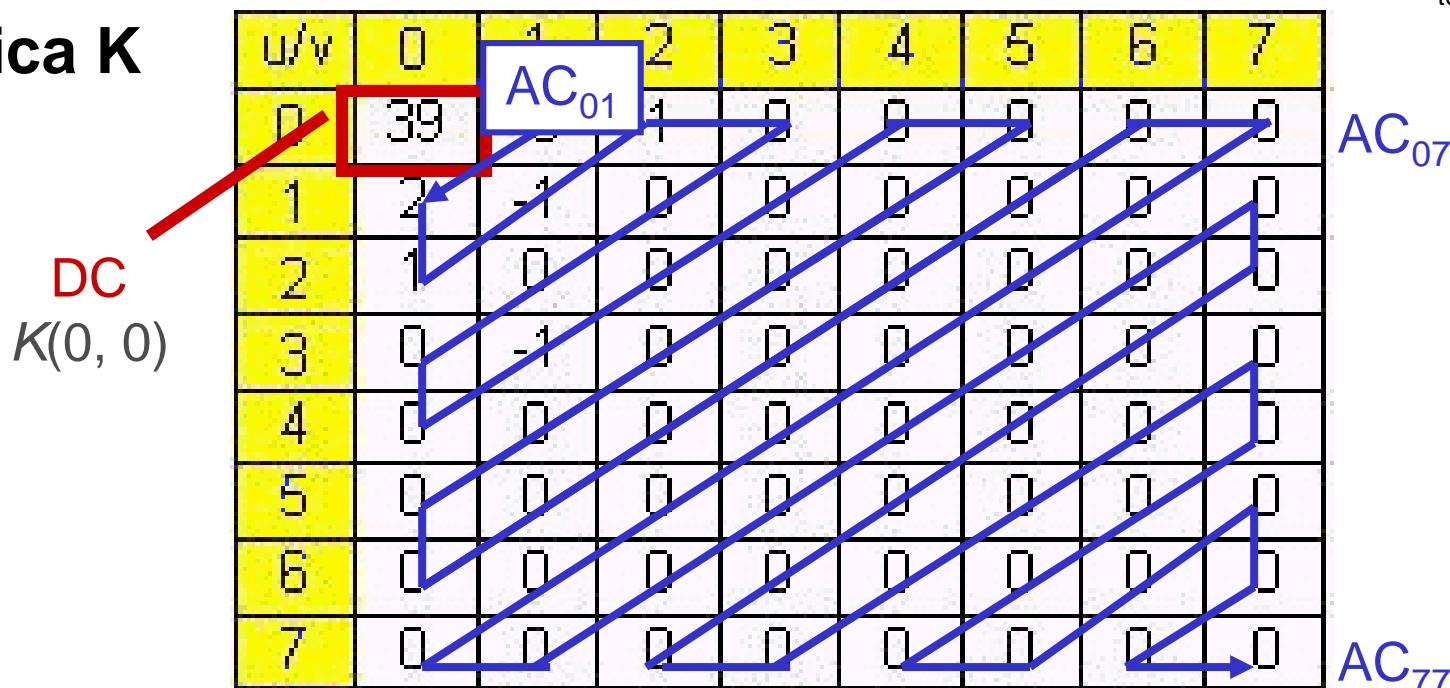
# 4: kompresija bez gubitaka

matrica K

$u/v$	0	1	2	3	4	5	6	7
0	39	-3	2	1	1	0	0	0
1	2	-1	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0
3	0	-1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

DC  
 $K(0, 0)$

$AC_{01}$        $AC_{07}$        $AC_{77}$



- Koeficijenti redom: 39 -3 2 1 1 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 ...,0, što se kraće zapisuje kao:  
 39 -3 2 1 1 0 0 0 0 0 -1 **EOB** (End of Block)
- Nakon takvog zapisa svih blokova u slici slijeva nadesno i odozgo prema dolje, slijedi **entropijsko kodiranje**
  - može se upotrijebiti Huffmanovo ili aritmetičko kodiranje

# Primjer: JPEG

- Slika  $516 \times 516$  točaka
- Kompresija 1:16



256 KB = 8 bit/pixel



16 KB = 0.5 bit/pixel

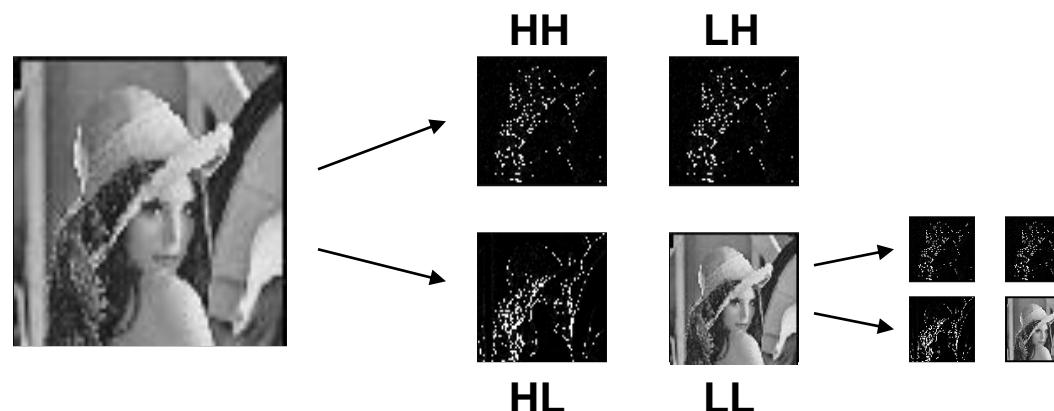
# Svojstva transformacijskog kodiranja

---

- + Visoka kompresija
  - (odlična kvaliteta slike do 0.25 bit/pixel)
- + Kvaliteta se može regulirati
- Pri većim kompresijama postaju vidljive granice blokova

# Kodiranje valićima (wavelet coding)

- Modernija metoda
- Slika se iterativno dijeli u prostorno-frekvencijske pojaseve u horizontalnom i vertikalnom smjeru
- Podjela se obavlja zbirkama filtara
- Prilikom kodiranja, manje bitova za komponente s manje energije



# Kodiranje valićima: svojstva

---

- Visoka kompresija
- „Prirodna“ distorzija (zamućenost)
- Prednosti u odnosu na transformacijsko kodiranje:
  - Manja složenost
  - Nema blok efekta
  - Mogućnost progresivnog primanja i stvaranja slike
- Osnova nove norme JPEG 2000

# JPEG 2000 vs JPEG (1/2)

---

- Veći dinamički raspon (16-32 bit/pixel)
- Bolja kompresija (25-30% manje podataka)
- Progresivno slanje podataka
- Kodiranje bez gubitaka prema područjima interesa (Region-of-interest, ROI)
- Međunarodna norma 2001

# JPEG 2000 vs JPEG (2/2)

- Usporedba rezultata uz jednaku kompresiju – 1:64



JPEG



JPEG2000

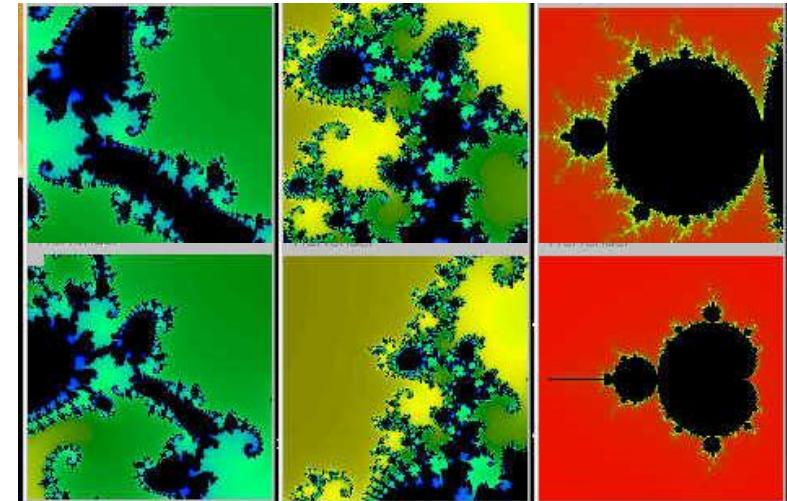
# Fraktalno kodiranje

---

- Princip analize i sinteze
  - Umjesto kodiranja samih podataka, kodiraju se parametri neke **funkcije** koja će generirati te podatke
  - Isti princip kao kod kodera govora zasnovanih na modelu
  - Nažalost, za općenitu sliku je teško pronaći model, odnosno funkciju koja je opisuje
- Ideja: samo-sličnost
  - Dijelovi slike međusobno slični, uz transformaciju
  - Pretpostavka je da se ovo svojstvo može iskoristiti za prikaz slike pomoću **fraktala**, koji pokazuju slična svojstva

# Fraktali

- Benoit Mandelbrot, 1975
- Fragmentirani, nepravilni geometrijski objekti koji pokazuju svojstvo samo-sličnosti
- Obično stvoreni rekurzivnim ponavljanjem određene funkcije
- U svakoj iteraciji objekt je transformirana verzija objekta iz prošle iteracije
- Proizvoljan nivo detalja



$$z := z^2 + c$$

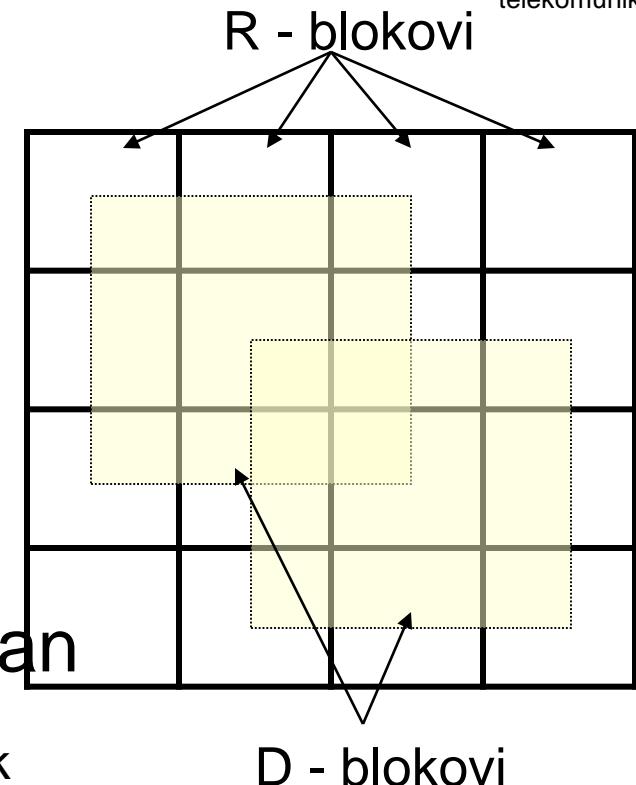
# Funkcija s fiksnom točkom

- $f(x) = x$ ; za točno jednu vrijednost  $x=x_0$  (vrijednost funkcije će biti ista kao onu koju uvrštavamo)
- Npr.  $F(x) = ax + b$ ; za  $a = 0.5$  i  $b = 1 \Rightarrow x_0 = 2$ ,  $F(2) = 2$
- Počevši od bilo koje početne vrijednosti, iteracijom ćemo uvijek stići do  $x_0 = 2$
- Vrijednost  $x_0$  možemo prikazati parom parametara  $a$ ,  $b$  ako smo prethodno fiksirali oblik funkcije  $F = ax + b$
- Za sliku (skup točaka)  $I$ ,  $F(I) = I$ 
  - Ako se parametri funkcije  $F$  mogu prikazati manjim brojem bita nego sama slika, postiže se kompresija

# Fraktalno kodiranje u praksi

- Arnaud Jacquin (1989)
- Podjela slike
  - R – blokovi uniformno pokrivaju čitavu sliku  $\bigcup_k R_k = I$
  - D – blokovi su veći, mogu biti bilo gdje unutar slike i ne pokrivaju je
- Za svaki R – blok, nalazi se jedan D – blok i funkcija  $f_k$ ,  $f_k(D_k) = R_k$
- Skup svih funkcija  $f_k$  definira čitavu sliku

$$\bigcup_k f_k = F \rightarrow F(\hat{I}) = \hat{I}$$



# Fraktalno kodiranje: svojstva

---

- Komplikirano
- Neujednačen stupanj kompresije
  - 4:1 do 100:1, jako ovisno o slici
- U početku se od fraktaala puno očekivalo, no druge metode (najviše valiči!) su uspješnije
- Princip je privlačan, moguće je da dođe do dalnjih napredaka i da metoda ipak zaživi u praksi



Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

6.

Informacijska svojstva i  
kodiranje videa

# Informacijska svojstva i kodiranje videa

---

- Osnovna svojstva videa
- Analogni i digitalni video, primjene
- Modeli boje za video, formati
- Postupak kompresije videa
  - Kompenzacija gibanja
  - Hibridno kodiranje videa
- Evolucija normi za kodiranje videa

# Video

---

- Video odn. “pokretna slika” se sastoji od niza nepomičnih slika (okvira), prikazanih *dovoljno brzo*
- Frekvencija promjene slike je brzina osvježavanja okvira (engl. *frame rate*) izražena u okvirima u sekundi [**fps**]
  - Opažamo gibanje kao neprekinuto ako je brzina osvježavanja slike **od 15 fps** (npr. za animaciju) **do 25-30 fps** (npr. za *full-motion video*)
  - Posebne primjene, npr. 3D simulacija leta, do **60-75 fps**

# Analogni video

---

- Video signal modulira jačinu elektronskog topa koji red po red prolazi po ekranu i ostavlja trag
  - Za mirnu sliku (bez titranja između uzastopnih okvira) minimum 50 Hz, stoga se koristi preplitanje (*interlace*)
- Norme za analogni video (TV)
  - **NTSC** (National Television System Committee), **30 fps**, u SAD i Japanu
  - **PAL** (Phase Alteration Line), **25 fps**, u Evropi, Kini i Australiji
  - **SECAM** (Séquentiel Couleur Avec Memoire), **25 fps**, u Francuskoj
  - **(HDTV)** (High Definition Television), **60 fps**

# Digitalni video

- Okvir videa → digitalna nepomična slika
- Video telefonija, telekonferencija
  - Simetrična primjena, koder i dekoder jednako složeni
  - Osjetljivost na kašnjenje
- Digitalna TV, filmovi, DVD
  - Asimetrična primjena, jedan koder služi puno dekodera, dakle može biti složeniji
  - Uključivanje u struju videa u bilo kojem trenutku
- Video preko interneta i mobilnih mreža
  - Relativno male brzine prijenosa
  - Osjetljivost na greške

# Modeli boje za video

- RGB (Red – Green – Blue)

- YUV
  - PAL video
  - Y – svjetlina (*luminance*)
  - U, V – boja (*chrominance*)

Veza YUV i RGB modela:

$$\begin{aligned}Y &= 0.299R + 0.587G + 0.114B \\U &= 0.493(B - Y) \\V &= 0.877(R - Y)\end{aligned}$$

- YIQ

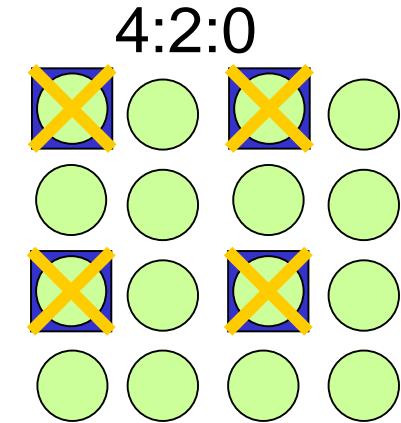
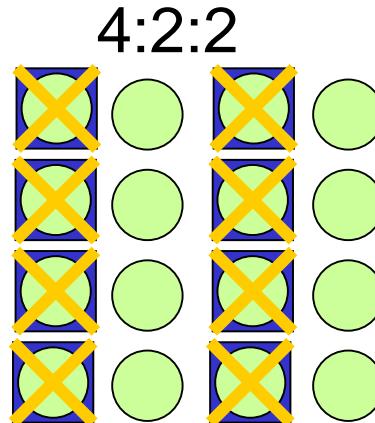
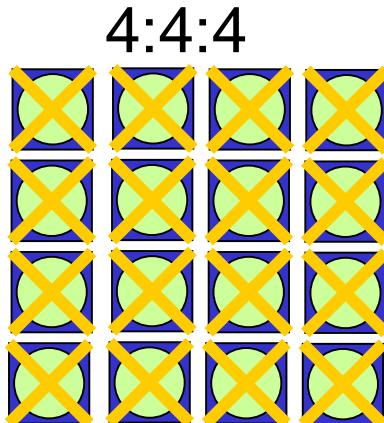
- NTSC televizija
  - Y – svjetlina; I, Q – boja

Veza YIQ i RGB modela:

$$\begin{aligned}Y &= 0.30R + 0.59G + 0.11B \\I &= 0.60R - 0.28G - 0.32B, \\Q &= 0.212R - 0.52G + 0.31B\end{aligned}$$

# Pod-uzorkovanje boje

- Dokazano je da je ljudsko oko manje osjetljivo na boju nego na svjetlinu
- Shodno tome, boja se može pod-uzorkovati da bi se smanjila količina podataka



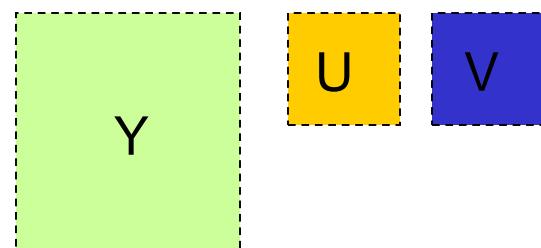
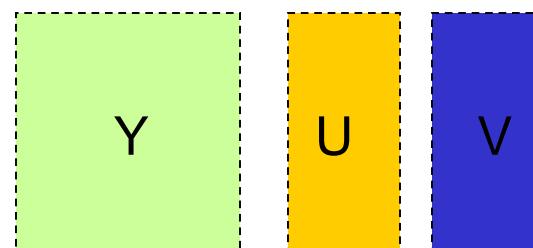
$Y$



$C_b$



$C_r$

# Formati za digitalni video

Format	Rezolucija ( $x \times y$ )	omjer $x:y$
NTSC	$640 \times 480$	4:3
PAL	$720 \times 576$	4:3
CIF	$352 \times 288$	4:3
MPEG-1	$360 \times 240$	3:2
MPEG-2	$720 \times 480$	3:2
HDTV	$1920 \times 1080$ (1280×720)	16:9
IMAX	$4096 \times 3072$	4:3

# Primjer: nekomprimirani video (1)

- Primjer 1:

Izračunajmo brzinu struje podataka za nekomprimirani CIF i QCIF (Quarter CIF) uz standardnu frekvenciju osvježavanja slike 30 fps.

CIF: Y (352 x 288); Cb, Cr (176x44)

$$30 \text{ [1/s]} * (352 * 288 + 2 * 176 * 144) * 8 \text{ [bit/pixel]} \approx \mathbf{36.495 \text{ Mbit/s}}$$

podijeljeno s 64 kbit/s  
(ISDN-B kanal)  $\approx 570$

QCIF: Y (176 x 144); Cb, Cr (88 x 72)

$$30 \text{ [1/s]} * (176 * 144 + 2 * 88 * 72) * 8 \text{ [bit/pixel]} \approx \mathbf{9.124 \text{ Mbit/s}}$$

podijeljeno s 64 kbit/s  
(ISDN-B kanal)  $\approx 142$

- Rješenja:

- smanjiti broj okvira u sekundi?

- npr. za 10 fps dobivamo **3.041 Mbit/s** što je još uvijek previše

- komprimirati podatke?

- npr. za **3.041 Mbit/s** i izlaz 64 kbit/s, trebamo omjer kompresije **47.5:1**  
(podsjetnik: JPEG tipično 10:1 do 50:1 za nepomičnu sliku)

# Primjer: nekomprimirani video (2)

- Primjer 2:

Izračunajmo podatke za nekomprimirani *full-motion* video uz format 640x480 pixela (NTSC), 24-bitnu boju i 30 fps – TV kvaliteta

jedan okvir uz rezoluciju 640 x 480: ..... **307 200 pixela**  
uz “true color” (24 bit/pixel): ..... **7 372 800 bit**  
@ 30 fps: ..... **221 184 000 bit/s**

Prostor za pohranu:

1 sekunda nekomprimiranog videa  $\approx$  **27 MB**

1 sat nekomprimiranog videa  $\approx$  **99 GB**



Postojeća rješenja:

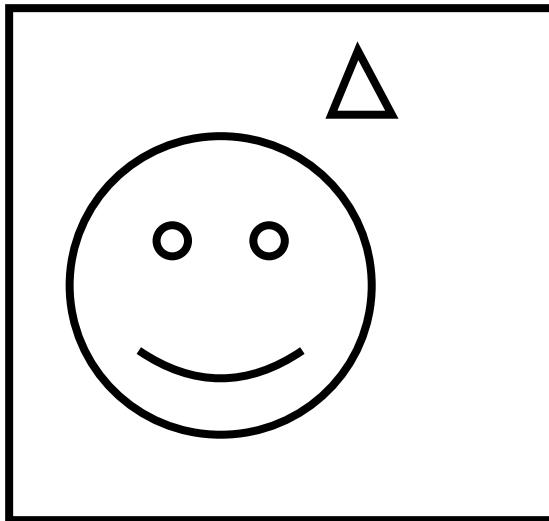
uz kompresiju 100:1 (npr. MPEG) 1 sat videa  $\approx$  **990 MB**

# Postupak kompresije videa

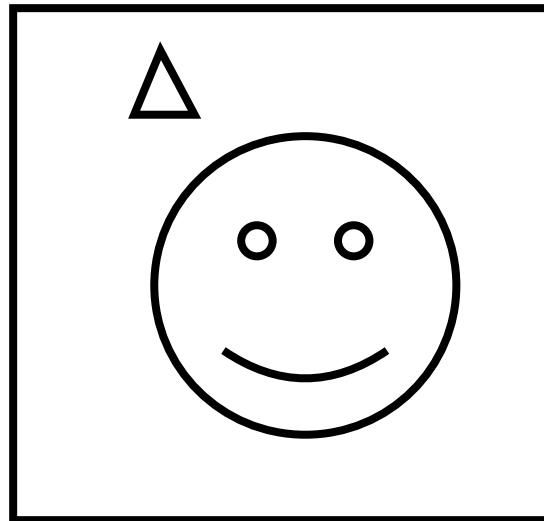
- Video sadrži prostornu i vremensku redundanciju
- Uklanjanje prostorne redundancije
  - Na razini svakog okvira/slike
  - Koristi se (najčešće) transformacijsko kodiranje
- Uklanjanje vremenske redundancije
  - U nizu uzastopnih okvira
  - Koristi se sličnost slika u nizu: diferencijalno kodiranje
- Pomak u slici je problem za diferencijalni koder
  - Koristi se kompenzacija gibanja (*motion compensation*)

# Kompenzacija gibanja (1)

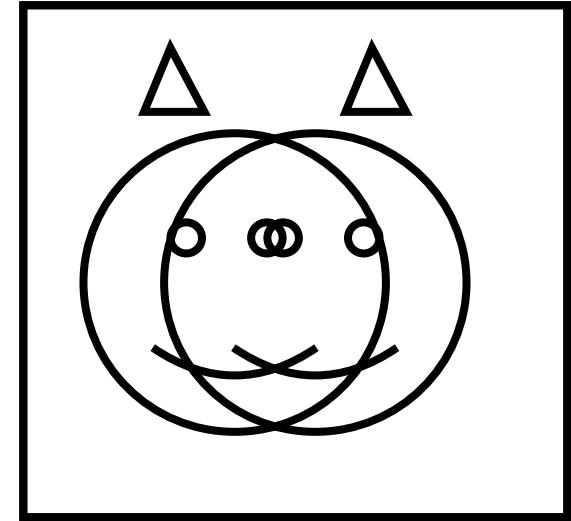
SLIKA  $n-1$



SLIKA  $n$



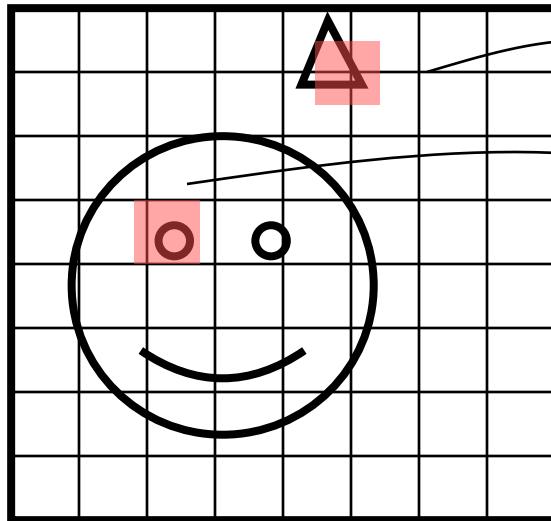
RAZLIKA



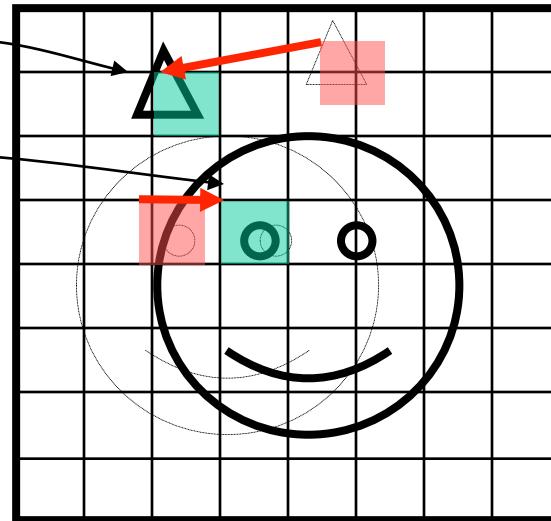
- Premda su slike vrlo slične, zbog pomaka ih ne možemo direktno diferencijalno kodirati: razlika sadrži više informacija od same slike!
- Različiti dijelovi slike imaju različite pomake

# Kompenzacija gibanja (2)

SLIKA n-1



SLIKA n



- Za svaki blok u slici, traži se **najsličniji blok** u prethodnoj slici
- **Razlika položaja** (u pixelima) između ova dva bloka je **vektor pomaka** →
- **Vektori pomaka** za sve blokove šalju se **dekoderu**; tako se konstruira slika slična prethodnoj, te se na njoj vrši **diferencijalno kodiranje**

# Kompenzacija gibanja (3)

Okvir  $n-1$

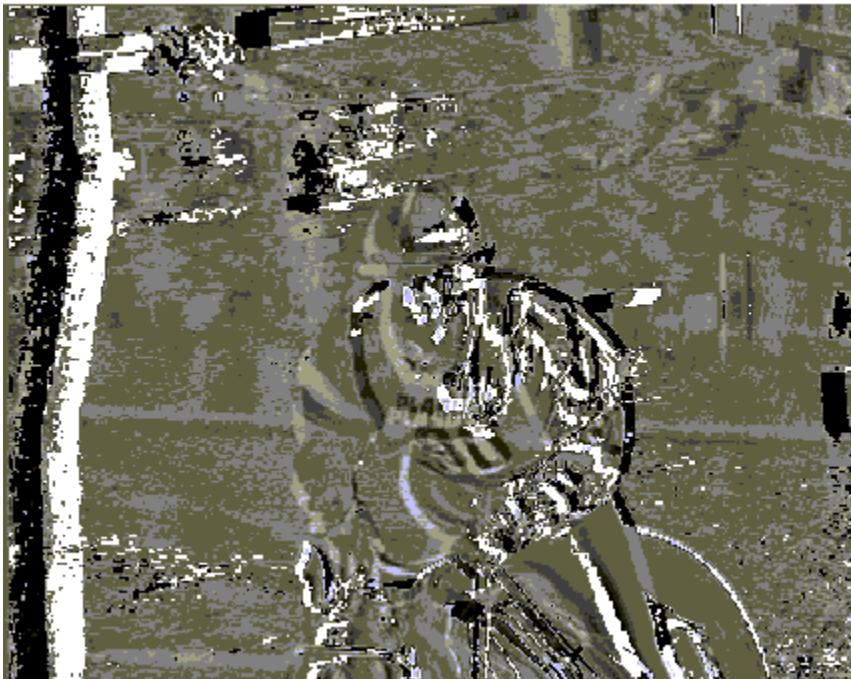


Okvir  $n$

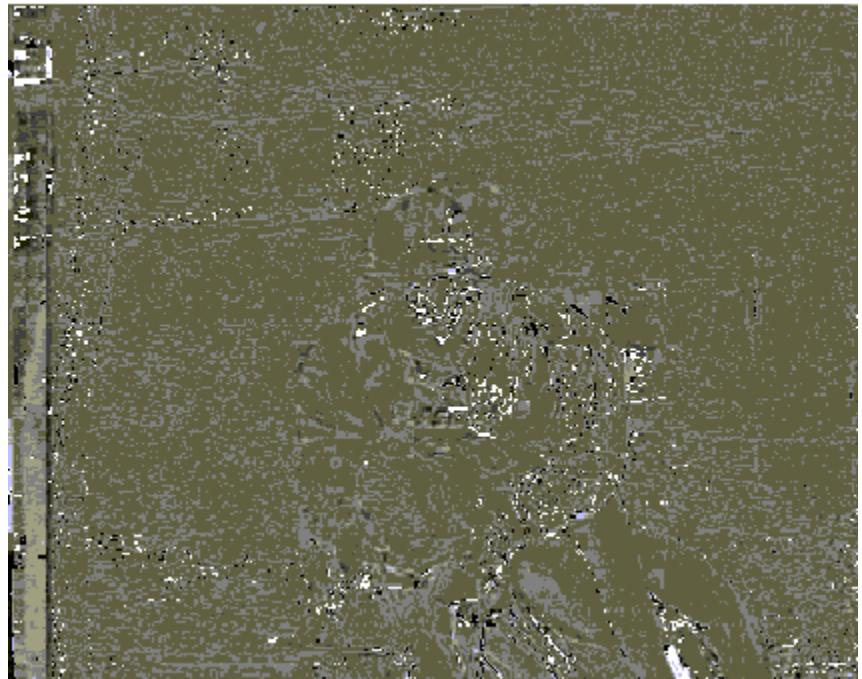


# Kompenzacija gibanja (4)

Razlika okvira  $n$  i  $n-1$



Pogreška predikcije s kompenziranim gibanjem

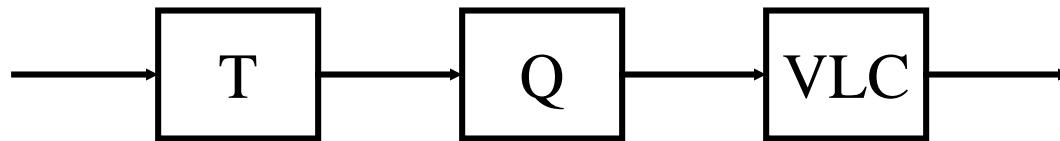


# Hibridno kodiranje videa

Diferencijalno kodiranje slika s kompenziranim gibanjem, te transformacijsko kodiranje signala razlike

- Ovo je najčešći princip kodiranja videa
  - Koristi se u svim normama koje ćemo spominjati
  - Ostalo su detalji ☺

# Transformacijsko kodiranje

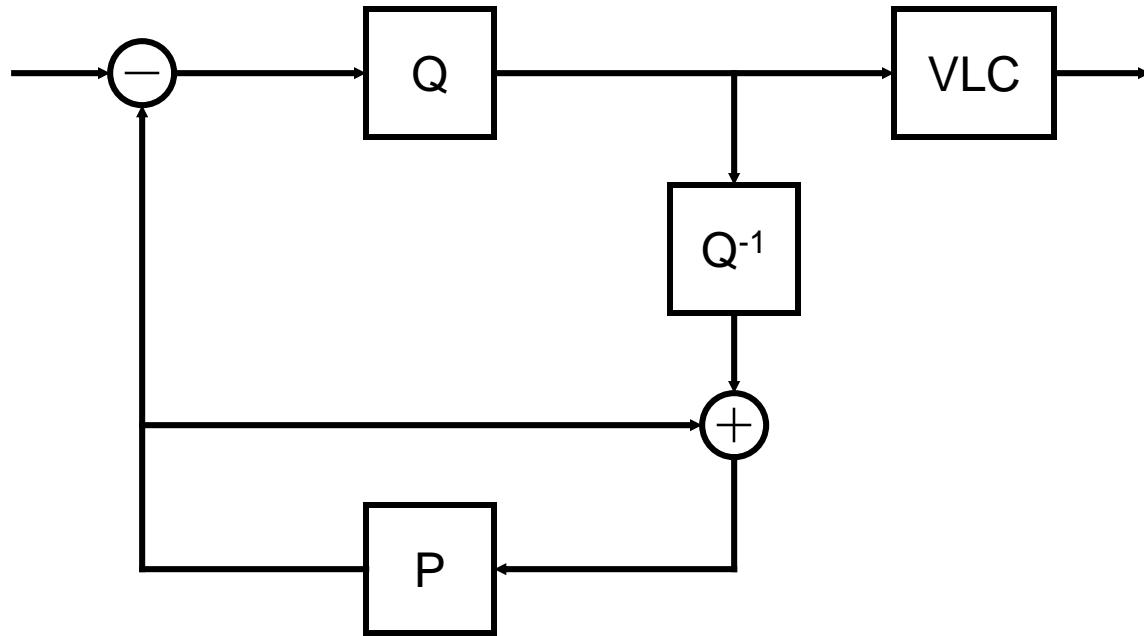


T: Transformacija

Q: Kvantizacija

VLC: Variable Length Coder (Entropijsko kodiranje)

# Diferencijalno kodiranje

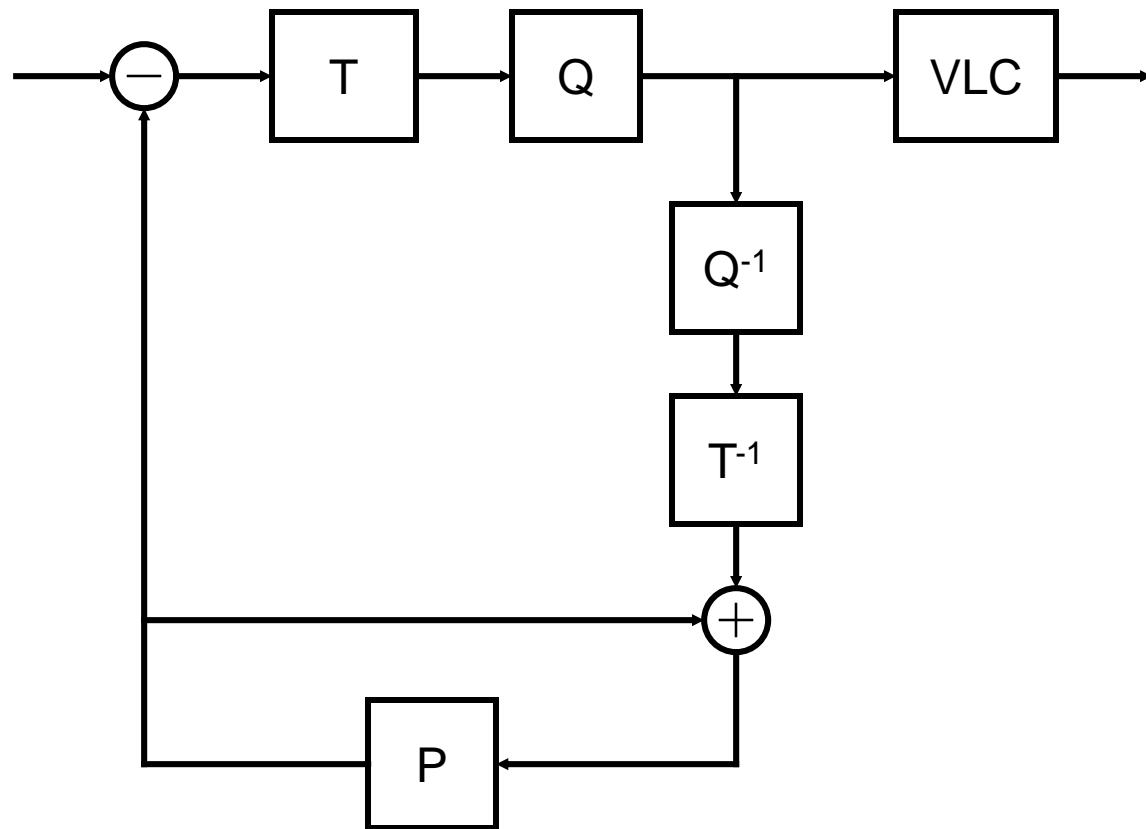


$Q$ : Kvantizacija

$Q^{-1}$ : Inverzna kvantizacija (rekonstrukcija)

$P$ : Predviđanje (predikcija)

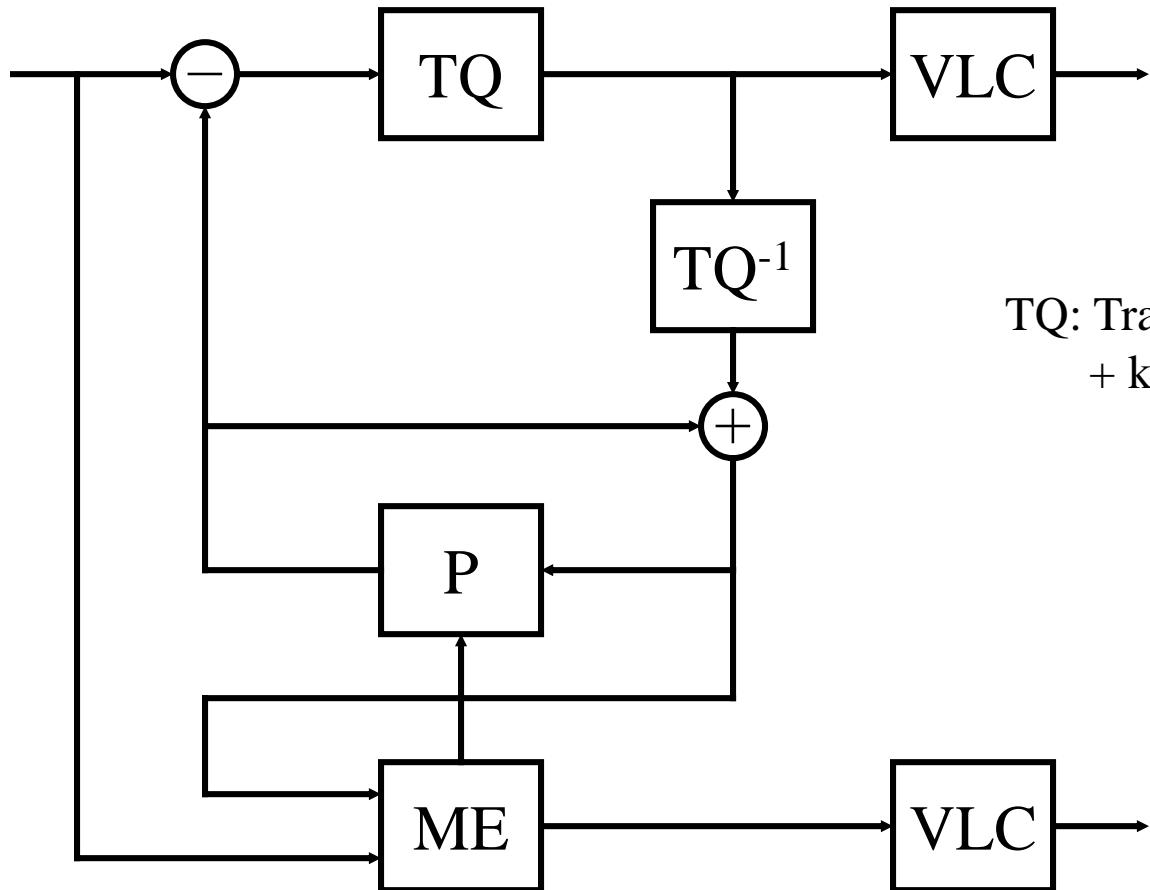
# Hibridno kodiranje



# Kompenzacija gibanja



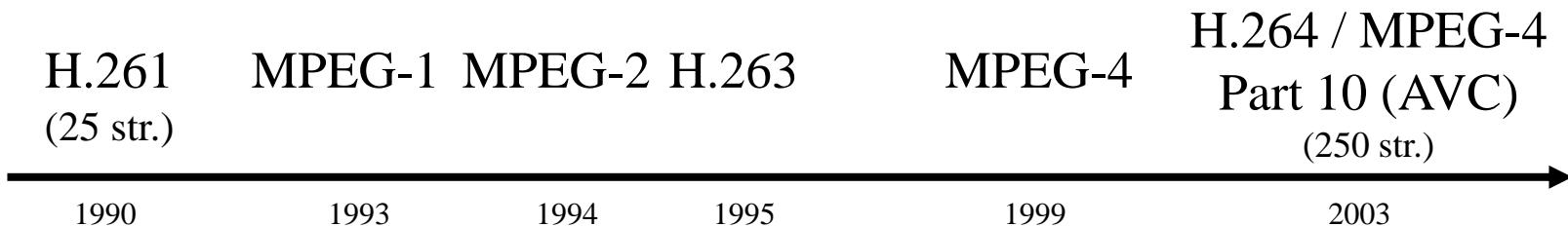
# Hibridno kodiranje s kompenzacijom gibanja



TQ: Transformacija  
+ kvantizacija

ME: Procjena gibanja (motion estimation)

# Norme za kodiranje videa



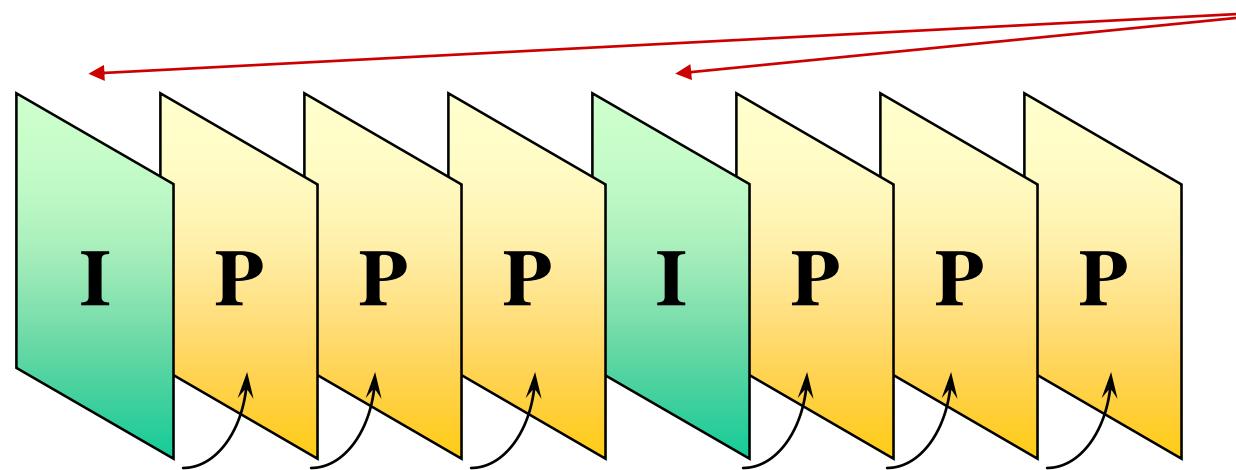
- Evolucija osnovnog principa na svim poljima
- Strategija predikcije (I, P, B okviri/blokovi)
- Veličina blokova (komp. gibanja, DCT)
- Veličina vektora pomaka
- Preciznost vektora pomaka
- Hijerarhijska podjela slika na jedinice
  - njihovo pakiranje u struju bitova
- (Dodatne funkcije: kodiranje oblika, sintetičkih objekata)

# Strategija predikcije (I, P, B okviri/blokovi)

- Diferencijalno (predikcijsko) kodiranje akumulira pogrešku
- Potrebno je s vremena na vrijeme poslati puni okvir, tzv. I-okvir (inter-kodiran)
- Prvi koderi uvode I- i P-okvire
- Bolja predikcija ako se koriste elementi iz prethodnih i idućih okvira
  - B-okviri s predikcijom/kompenzacijom gibanja u odnosu na prethodni i idući okvir
  - višestruke referentne slike (AVC/H.264)

# Kodiranje I- i P- okvira

- Kombinacija **intra-kodiranja** i **inter-kodiranja** omogućuje dovoljno brzu kompresiju za kompresiju i dekompresiju videa "u hodu", odn. u stvarnom vremenu



početni i svaki 132 okvir mora biti I-okvir radi izravnog pristupa, kao i oporavka od grešaka do kojih može doći u prijenosu

- Dvije vrste okvira:
  - I-okvir** – pojedinačno kodiran (intra-kodiran, neovisan o okvirima prije/poslije)
  - P-okvir** – predikcijski kodiran (inter-kodiran, relativno u odnosu na prethodni)

# Zašto B okviri?

Svojstva uzastopnih okvira:

- Neki makro-blokovi su **isti, ali pomaknuti** u odnosu na makro-blokove prethodne slike (objekti na slici)
- Neki makro-blokovi su **na istom mjestu** kao na prethodnoj slici (pozadina)
- Neki makro-blokovi su **novi** (npr. otkrivena pozadina)
  - otkriveni blokovi pojavljuju se u cijelosti u budućim slikama

koristi se predikcija “**unaprijed**”: reference iz prošlosti (mogu se iskoristiti iz prethodne slike)

koristi se predikcija “**unatrag**”: reference iz budućnosti  
- pogodno za otkrivenu pozadinu

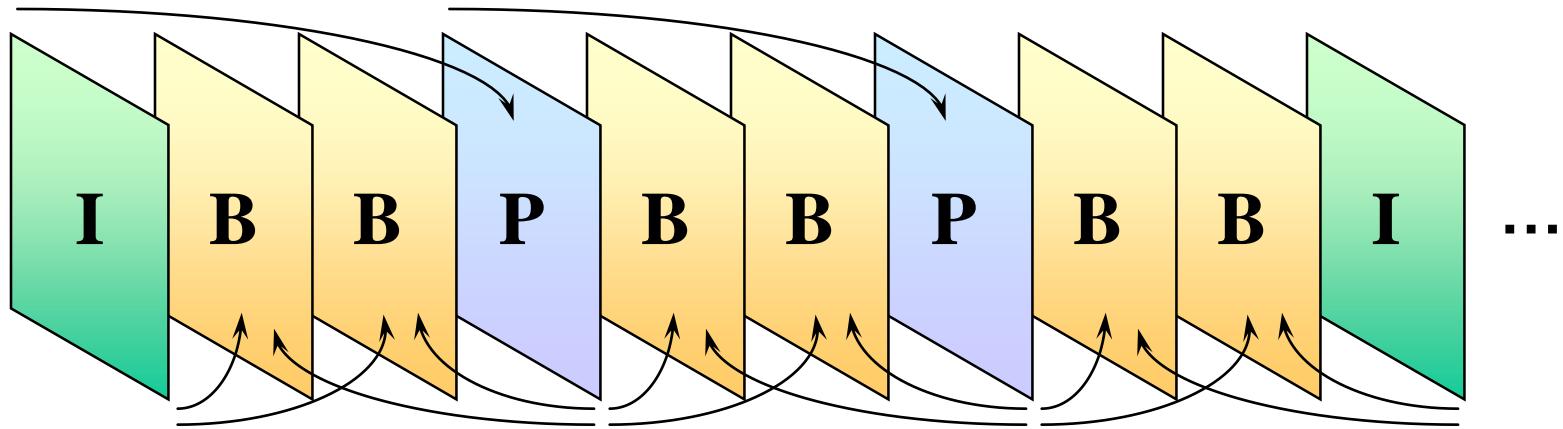
Dvosmjerna predikcija: reference iz prošlih i budućih okvira

# Kodiranje I-, P- i B-okvirima

- Tri vrste slika:
  - **I-slika** – pojedinačno kodirana (intra-kodirana, neovisana o slikama prije/poslije, koristi JPEG kompresiju)
  - **P-slika** – predikcijski kodirana (relativno u odnosu na prethodnu I- ili P sliku, postupak kao H.263, jer ga je H.263 preuzeo iz MPEG-a)
  - **B-slika** – dvosmjerno predikcijski kodirana (relativno u odnosu na prethodnu ili sljedeću I ili P sliku)
- Učinkovitost kompresije:
  - I-slika – JPEG kompresija
  - P-slika – bolja kompresija od I-slike, lošija od B-slike
  - B-slika – najbolja kompresija
  - tipični omjer:  $I = 3P = 5B$

# Redosljed okvira

- Redosljed okvira za prikaz (prirodni redosljed):



- B-okvir se izračunava na temelju prethodnog i sljedećeg I ili P okvira, tako da oni moraju biti poslani ranije
- Redosljed slanja i dekodiranja okvira je drukčiji od redosljeda prikaza

# Redosljed dekodiranja okvira

## Redosljed okvira za prikaz

Vrsta okvira: I B B P B B P B B I B B P  
Broj okvira: 1 2 3 4 5 6 7 8 9 10 11 12 13

## Redosljed slanja i dekodiranja

Vrsta okvira : I P B B P B B I B B P B B  
Broj okvira : 1 4 2 3 7 5 6 10 8 9 13 11 12

# Veličina blokova

---

- Veličina blokova za
  - kompenzaciju gibanja
  - DCT transformaciju
- Manji blokovi:
  - veća preciznost
  - bolji rezultati
  - Ali više računanja
- Novije norme dozvoljavaju manje blokove
  - komp. gibanja: H.261: 16x16; MPEG-4: 8x8;  
H.264/AVC: 4x4
  - DCT: H.261, MPEG-4: 8x8; H.264/AVC: 4x4

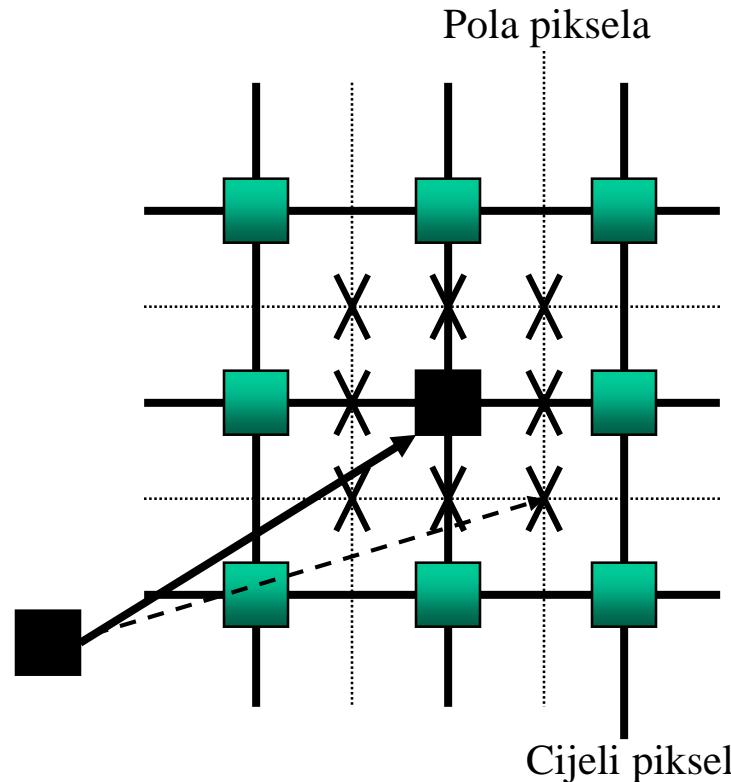
# Veličina vektora pomaka

---

- Određuje površinu pretrage za sličnim dijelom slike
  - Veći dozvoljeni vektor pomaka:
    - veća površina pretrage
    - veća vjerojatnost pronađenja dobrog uzorka slike
    - bolja kompenzacija gibanja
    - ALI više računanja (pretraga je skupa!)
  - Iznad neke granice više se ne isplati povećavati
    - stvarni pomak obično nije tako velik
    - u praksi se koriste vektori pomaka od 16-32 pixela

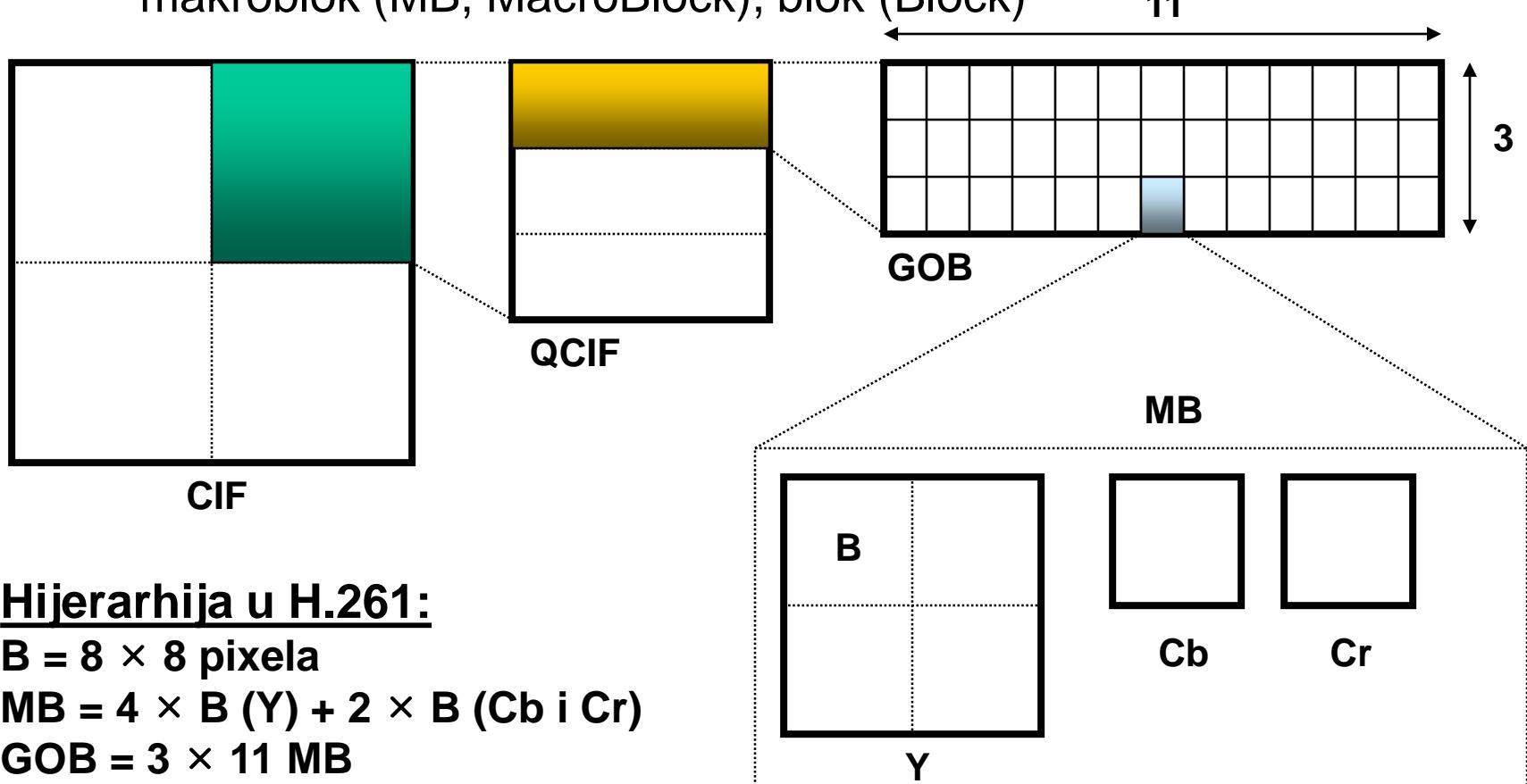
# Preciznost vektora pomaka

- Preciznost na “pola pixela” umjesto na “cijeli pixel”
- Koristi se interpolacija Y vrijednosti kako bi se umjetno stvorile dodatne točke



# Hijerarhijska podjela slika na jedinice

- Primjer: H.261
  - Slika (Picture), skupina blokova (GOB, Group Of Blocks), makroblok (MB, MacroBlock), blok (Block)



## Hijerarhija u H.261:

$$B = 8 \times 8 \text{ pixela}$$

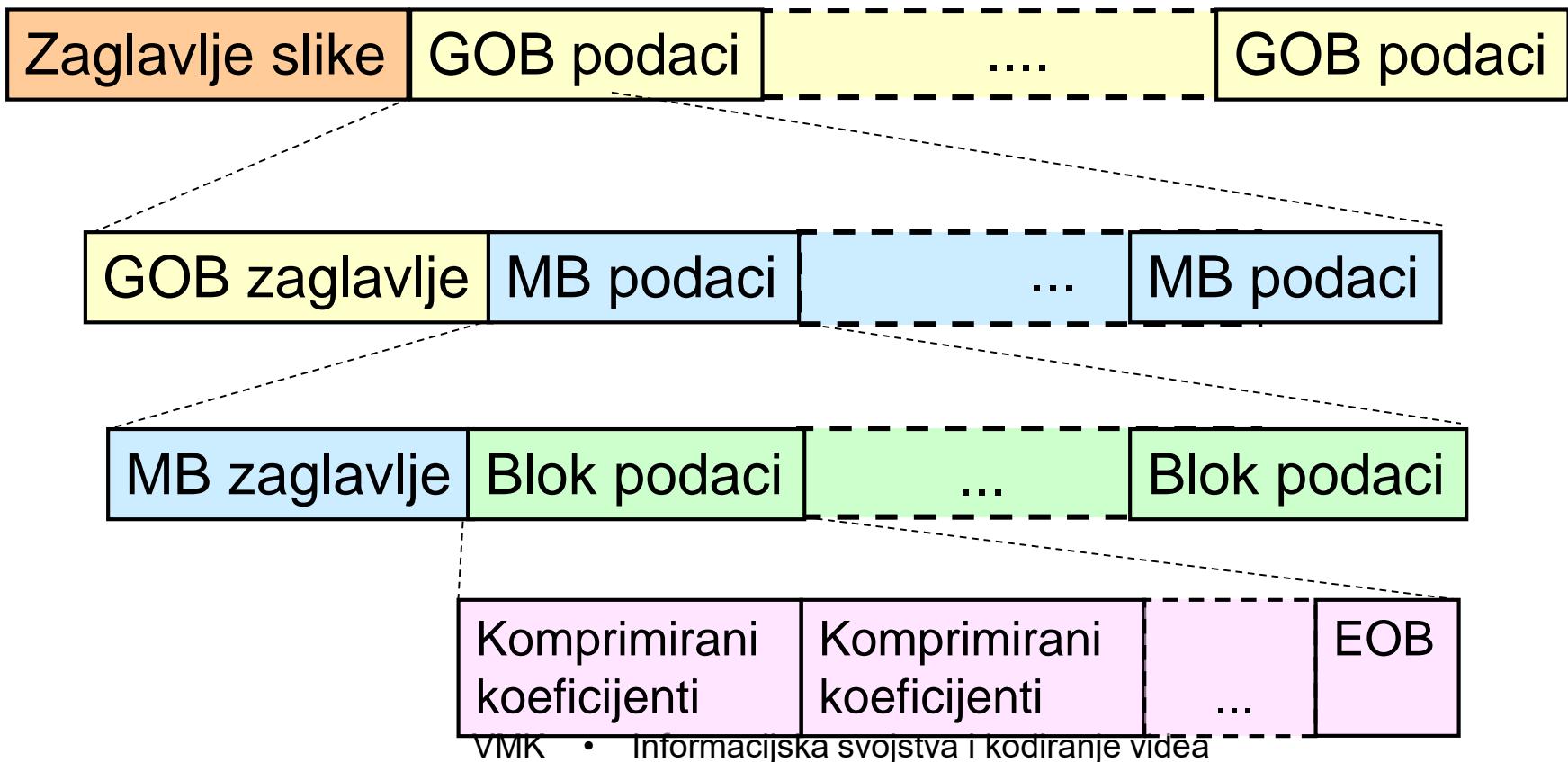
$$MB = 4 \times B (Y) + 2 \times B (Cb \text{ i } Cr)$$

$$GOB = 3 \times 11 MB$$

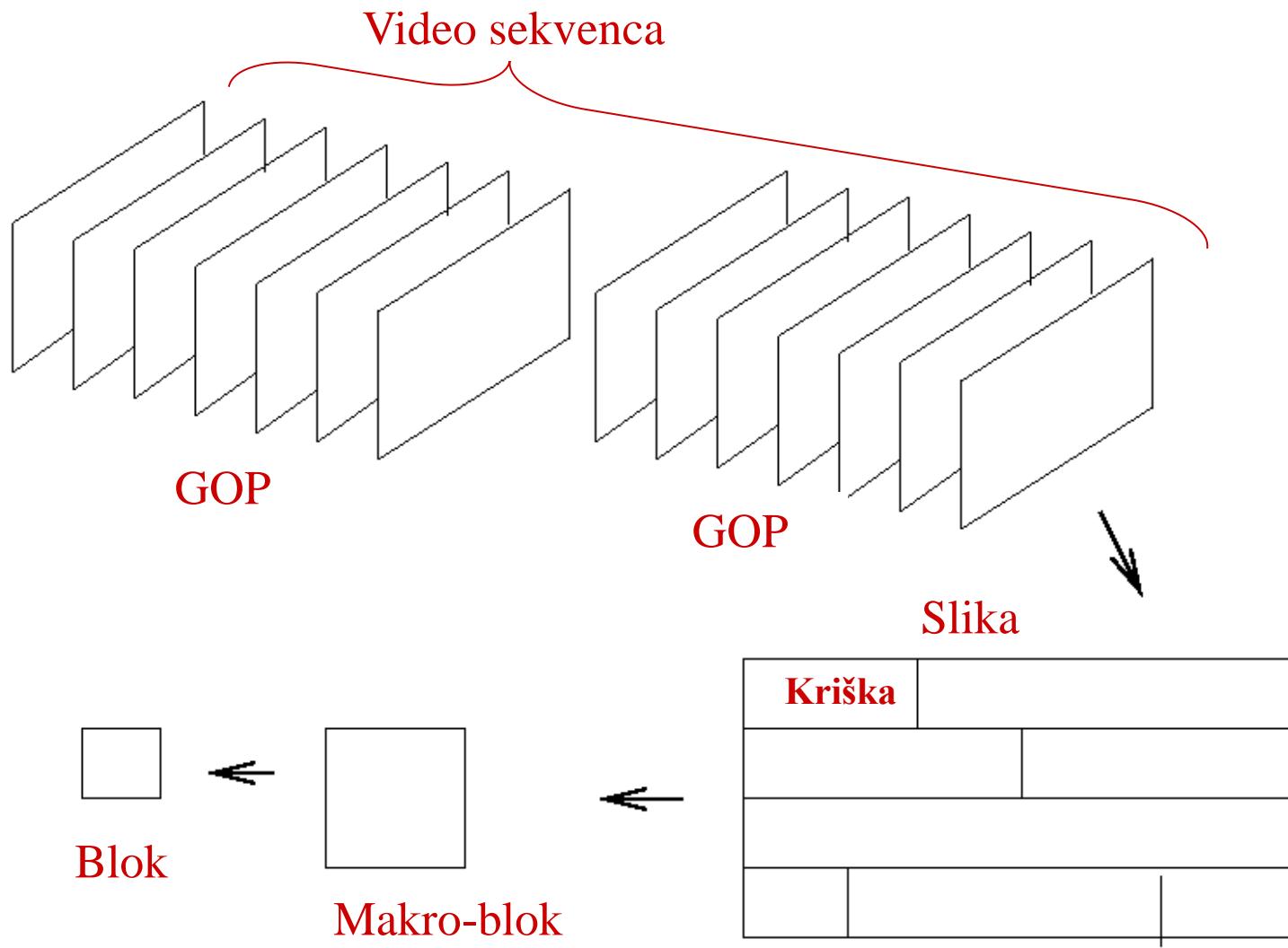
$$QCIF = 3 \times GOB$$

# Struja podataka u H.261

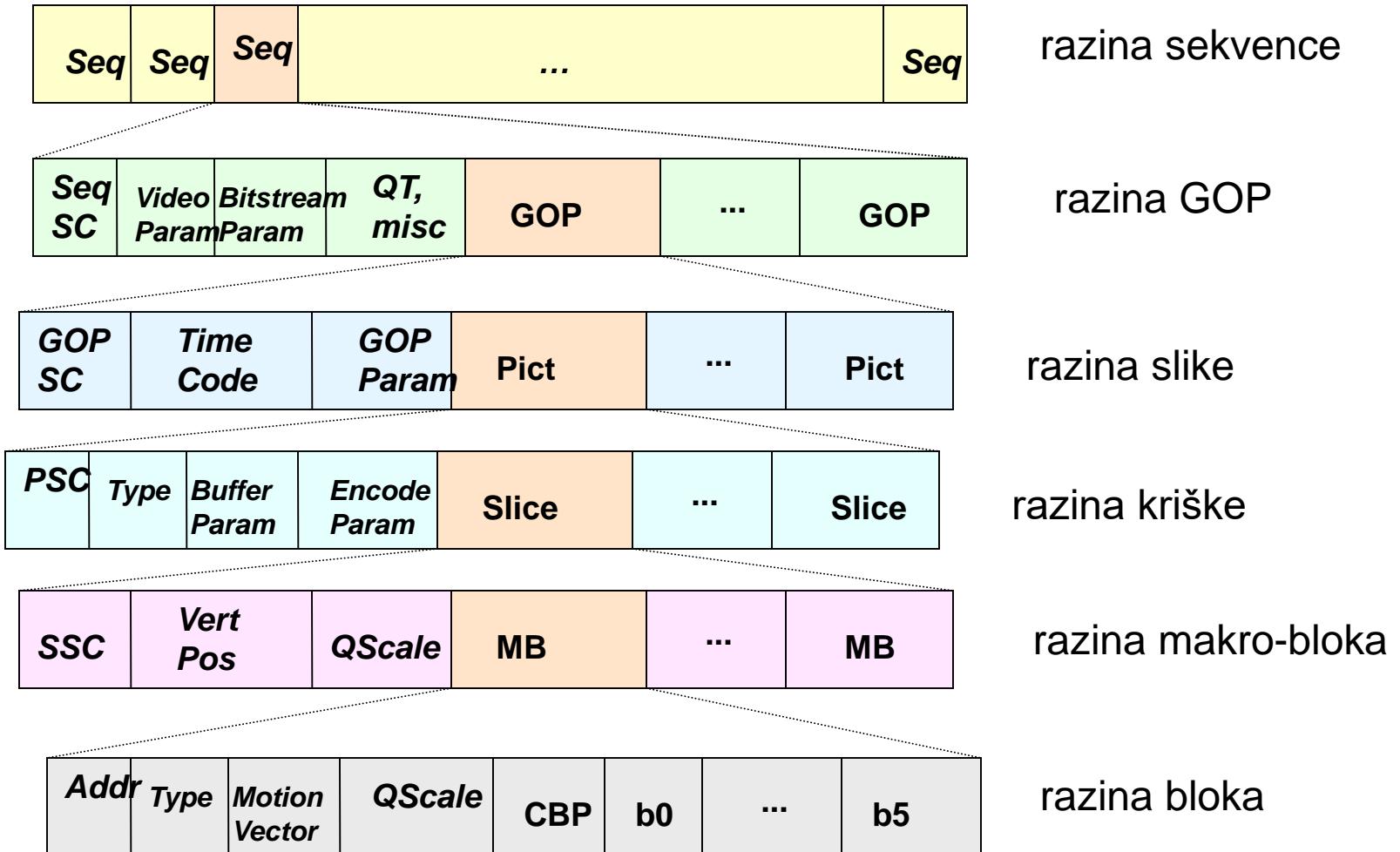
- Struja podataka formirana je hijerarhijski prema podjeli slike na jedinice podataka
- Zaglavljje s podacima koji važe za čitavu jedinicu
  - npr. radi li se o I- ili P-okviru; vektori pomaka; kvantizacijski faktori



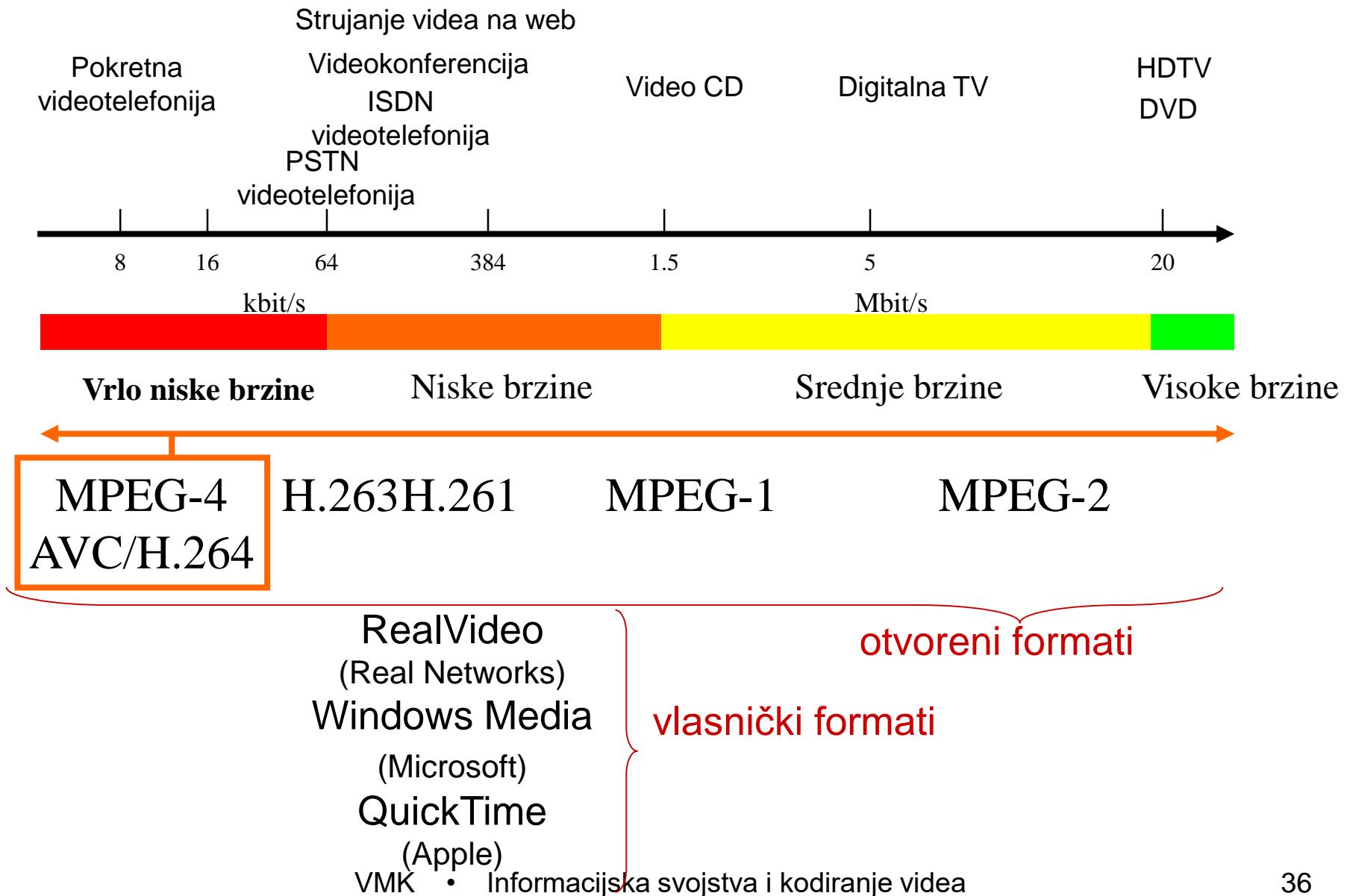
# Primjer: MPEG-2



# Primjer: MPEG-2



# Norme i njihove primjene





Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

7.  
Kodiranje sintetičkih  
sadržaja

# Kodiranje sintetičkih sadržaja

---

- Vrste sintetičkih sadržaja
- Kodiranje sintetičkih sadržaja u normi MPEG-4
- Primjer sintetičkog sadržaja: animacija ljudskog lica

# Sintetički sadržaji

---

- Stvoreni na računalu, a ne iz prirode
- 2D grafika & tekst
- 3D grafika
- Animacija ljudi
- Sintetički zvuk
  - Govor, instrumenti, 3D zvuk

# MPEG-4

---

- MPEG-4 V1 postaje IS 1999, V2 2000
- Norma za kodiranje višemedijskih objekata na raznim brzinama
- Fokus više nije na kompresiji, nego na novim funkcijama i sadržajima
- Jedinice za kodiranje nisu više okviri, nego **audio-vizualni objekti**
  - Video, audio, tekst, slika, grafika...
- AV objekti se na dekoderu slažu u scenu
  - Opis scene određuje prostorni i vremenski raspored

# MPEG-4 scena

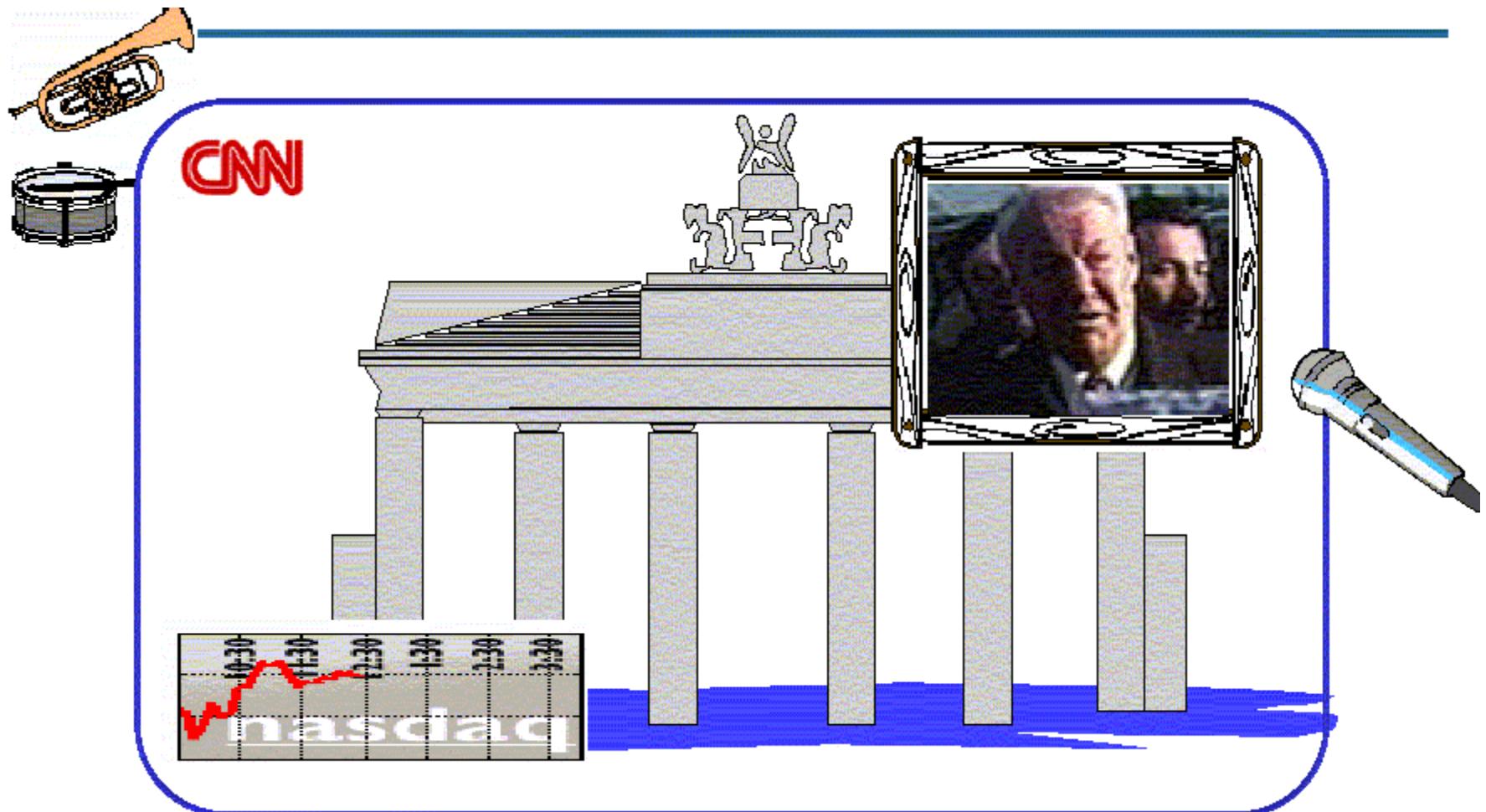


# Djelovi norme MPEG-4

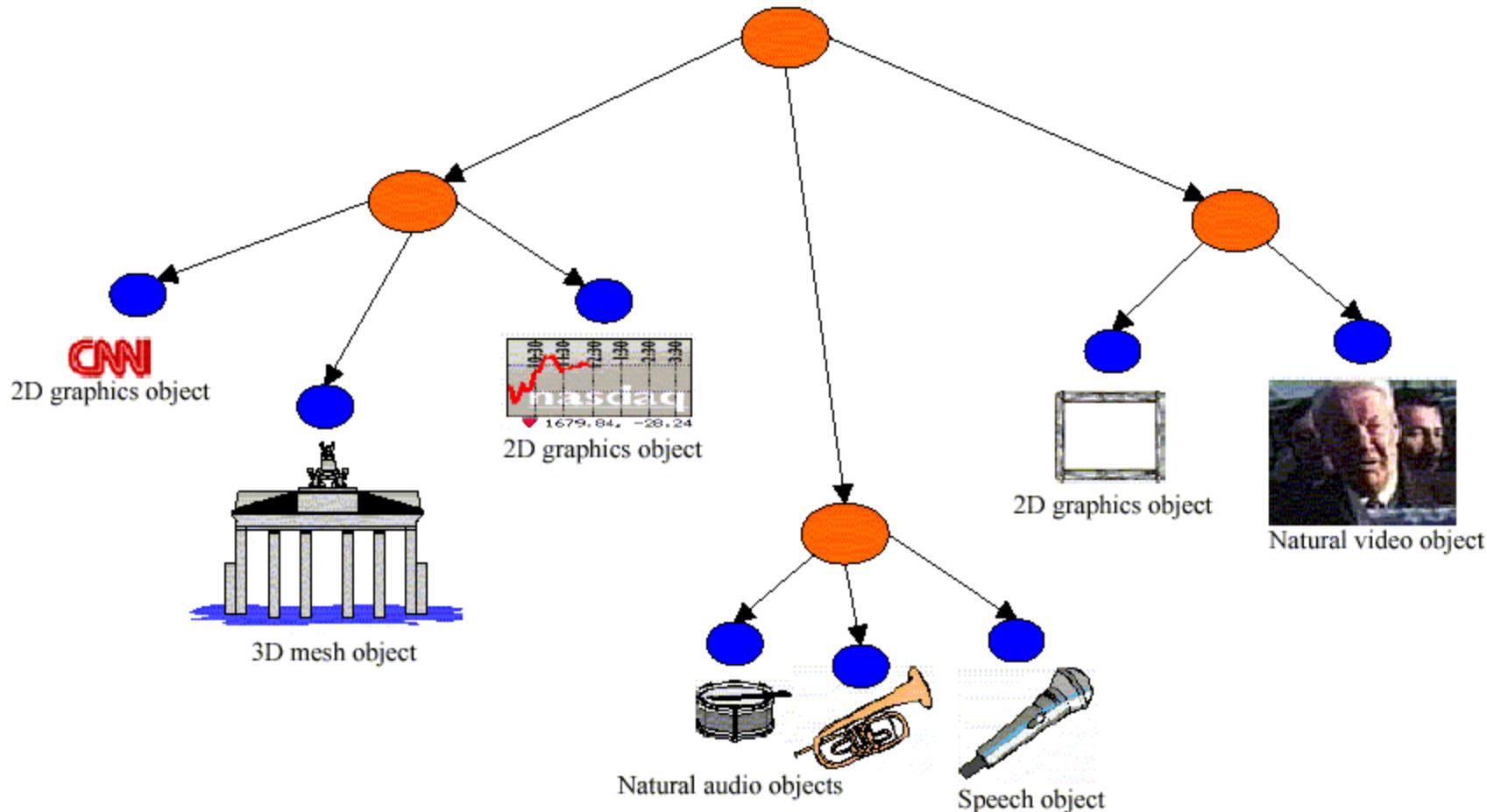
---

- Part 1, Systems (sustav)
  - „Jezik“ za opis scene – BIFS
  - Ispreplitanje (mux),...
- Part 2, Visual
  - Kodiranje videa
  - Kodiranje nepomične slike
  - Kodiranje texture
  - Kodiranje animacije lica i tijela
  - ...
- Part 3, Audio
  - Nekoliko audio codec-a za razne aplikacije

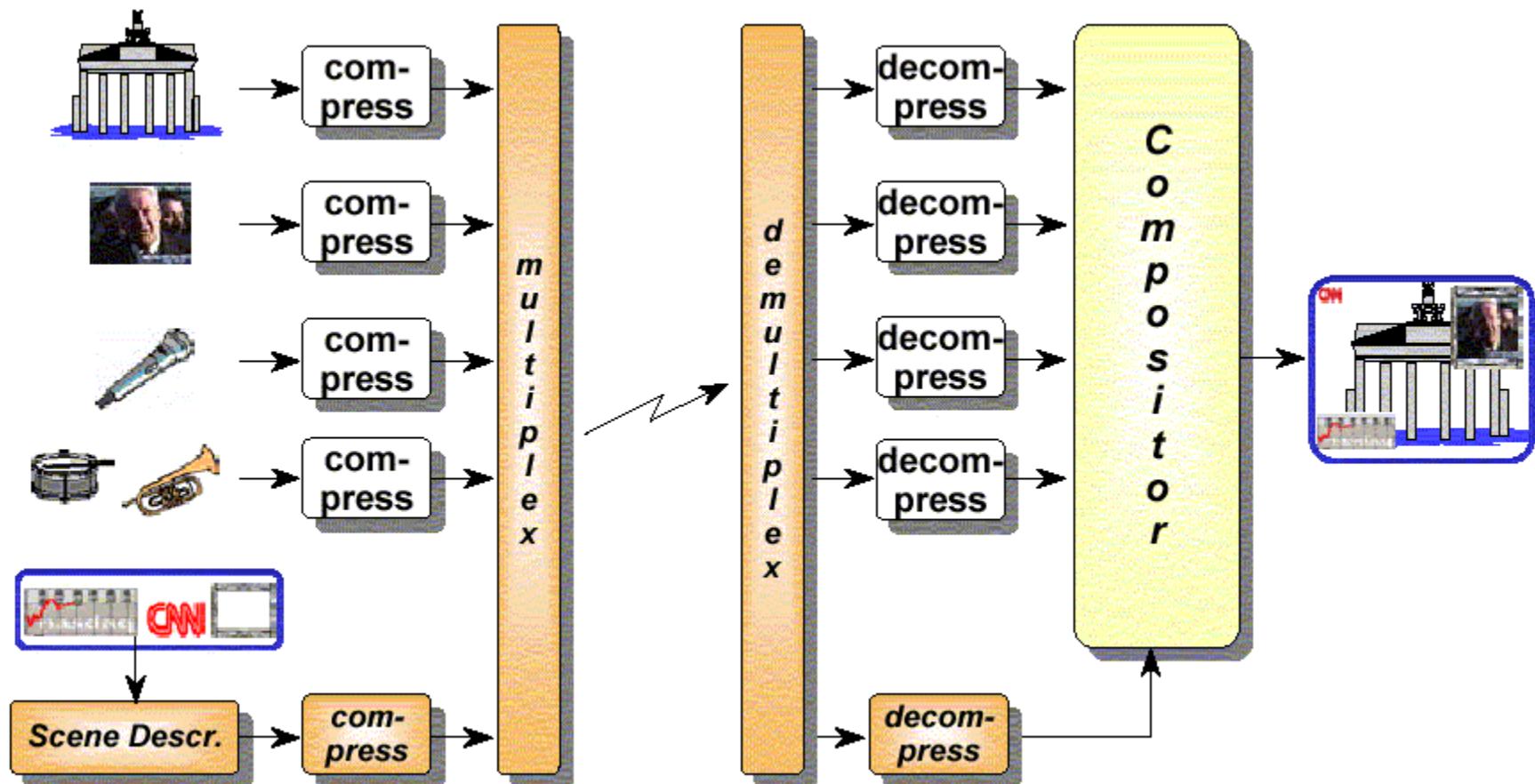
# Primjer kodiranja scene



# MPEG-4 opis scene



# MPEG-4 arhitektura



# Opis scene: BIFS

---

- Binary Format for Scene Description (BIFS)
- Opis scene je aciklički, usmjereni graf
  - Standardni način prikaza scene u 3D grafici
- Na završnim (medijskim) čvorovima su objekti za prikaz
- Ostali čvorovi: hijerarhijska i prostorna struktura
- Čvorovi imaju polja koja opisuju njihove značajke
  - Boolean, integer, float, color, ...

# Hijerarhijska i prostorna struktura

---

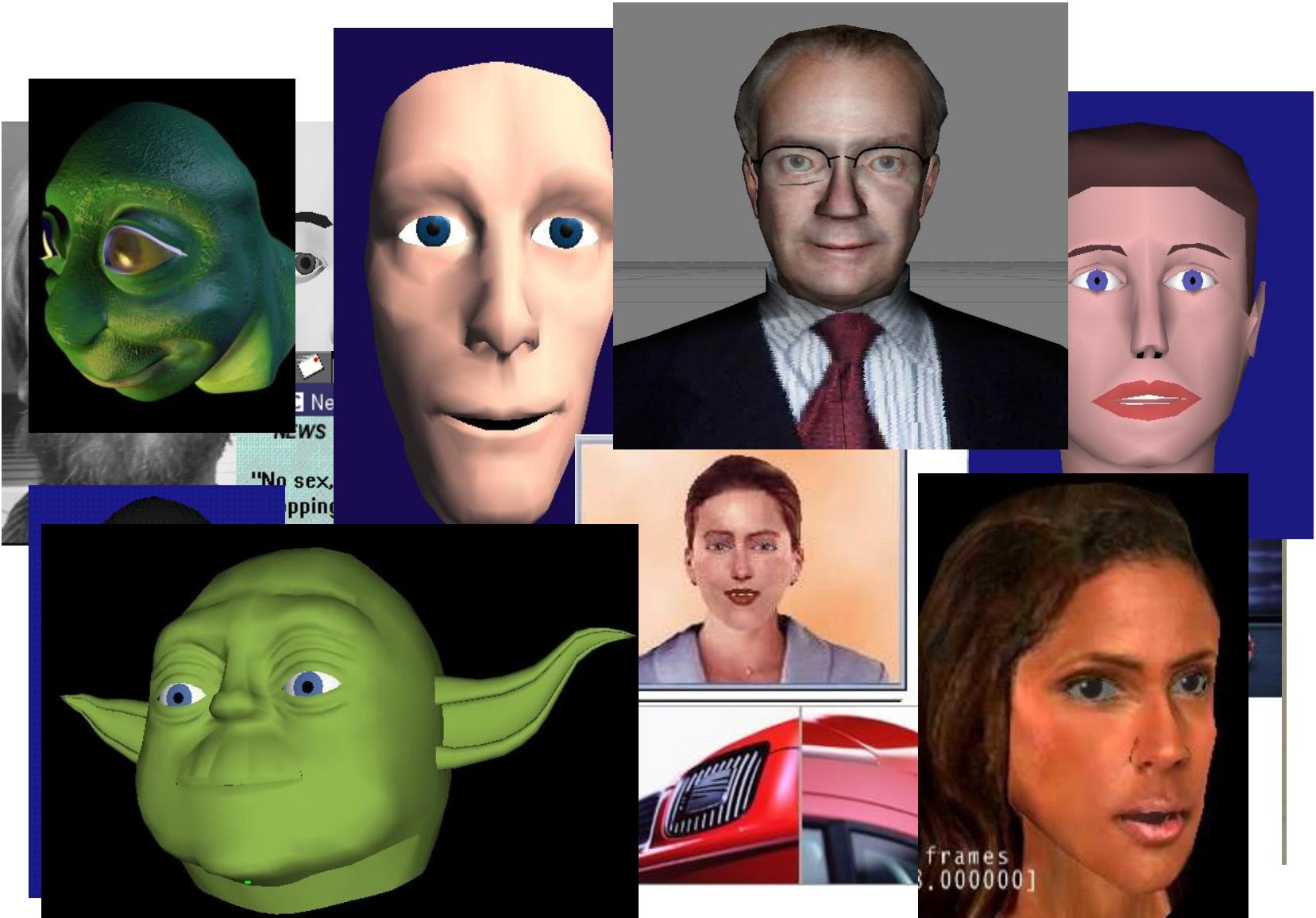
- Grupirajući čvorovi
- Transformacijski čvorovi (pomak, rotacija, veličina)
- Izborni čvorovi
- Čvorovi za izbor razine detalja
- Interaktivni čvorovi
  - Omogućavaju korisničku interakciju
- Čvorovi mogu biti animirani (BIFS Anim)
  - Npr. pozicija se mijenja u vremenu

# Tipovi medijskih čvorova

---

- Nepomična slika ili video
- Prirodni ili sintetički zvuk
- Jednostavni grafički objekti
  - Kvadar, kugla, cilindar...
- Mreže poligona (2D ili 3D)
  - Kodiraju se posebnim postupcima
- Animirano lice i tijelo

# Animacija lica

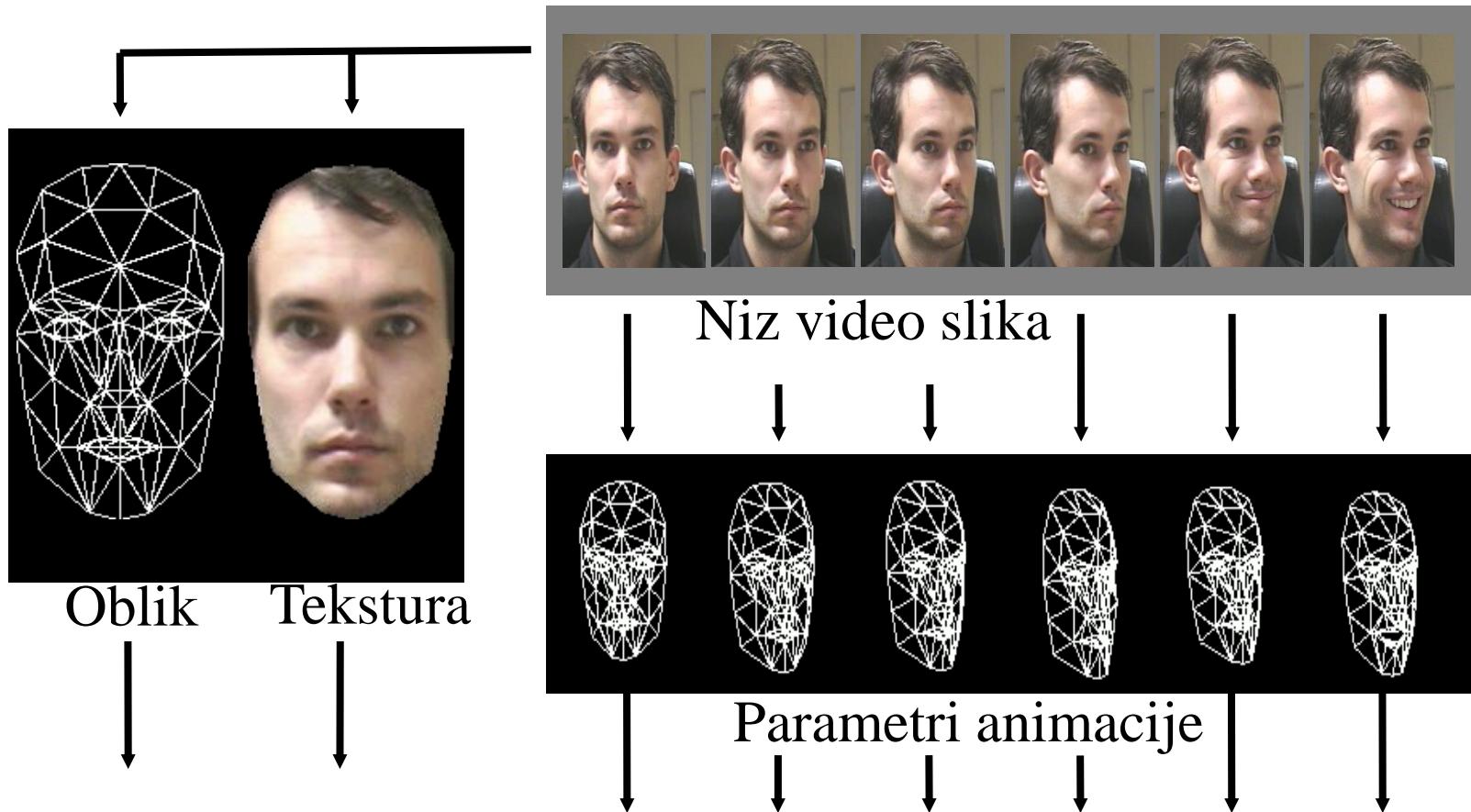


# Animacija lica

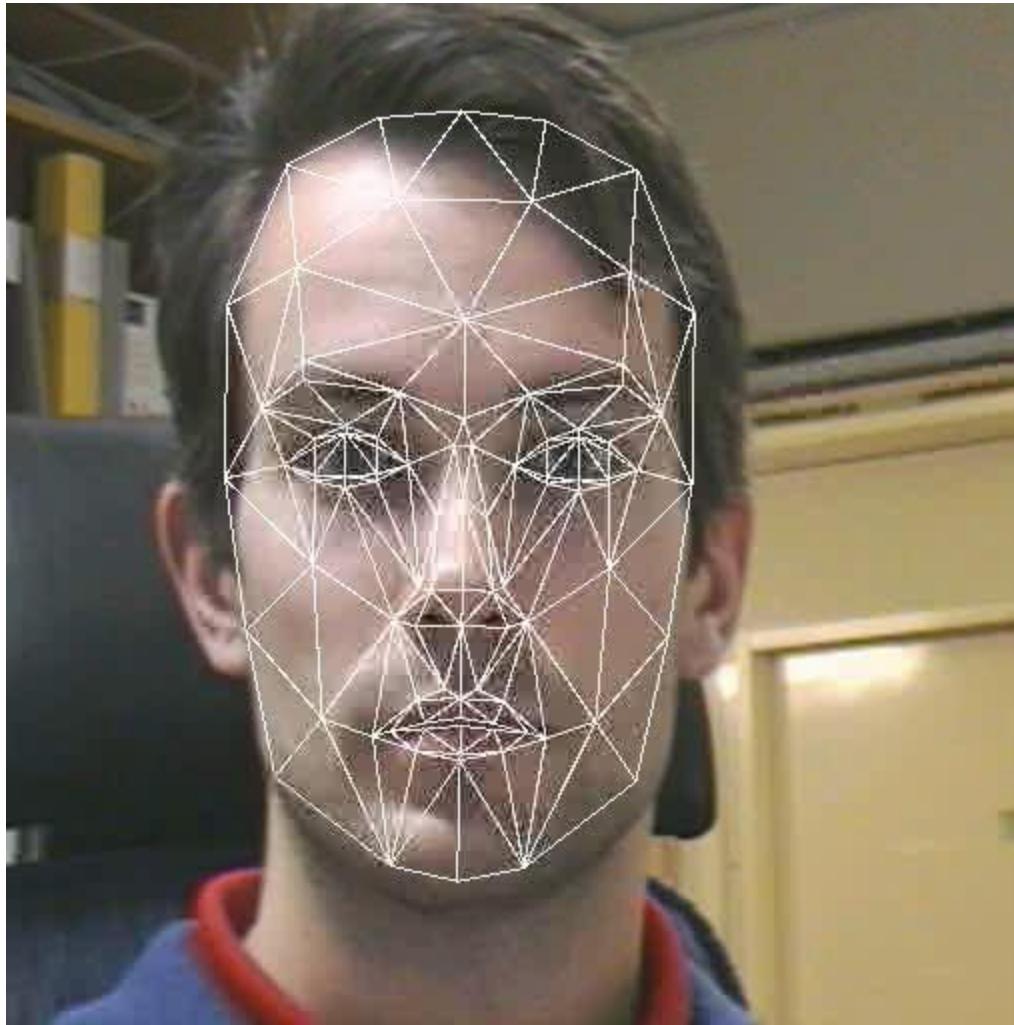
---

- Deformacija 3D modela lica
- Primjene
  - Film, TV, igre
  - Komunikacije (video-telefonija, poruke, vijesti...)
  - Napredno korisničko sučelje (računalo kao sugovornik)
- Izvori animacije lica
  - Sinteza govora
  - Analiza govora
  - Ručna animacija
  - Analiza videa

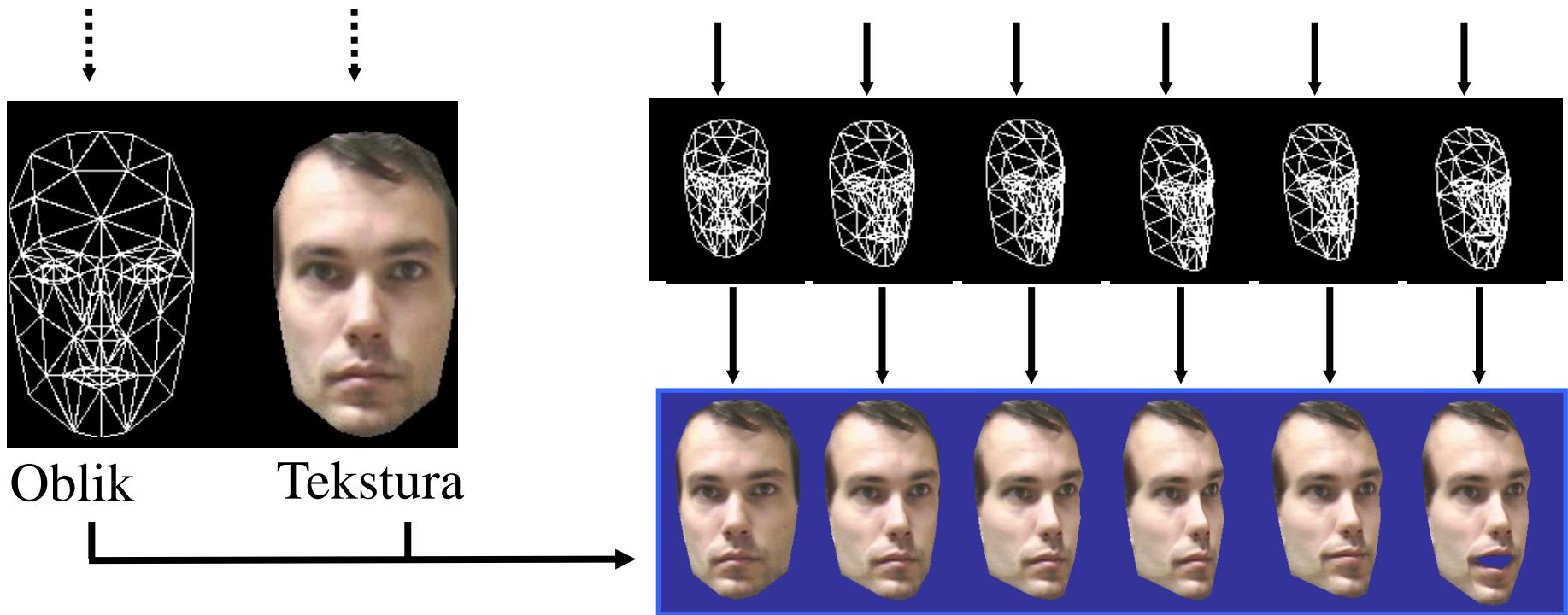
# Kodiranje zasnovano na modelu



# Praćenje parametara u videu



# Dekodiranje



# Rezultat dekodiranja na drugom licu



Zavod za  
telekomunikacije

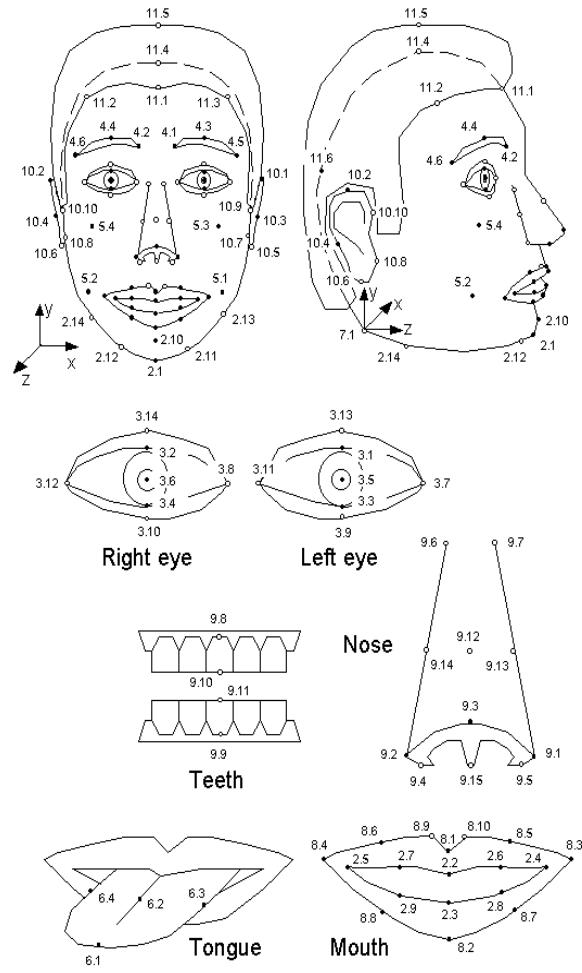


# Još jedan primjer



# MPEG-4 animacija lica

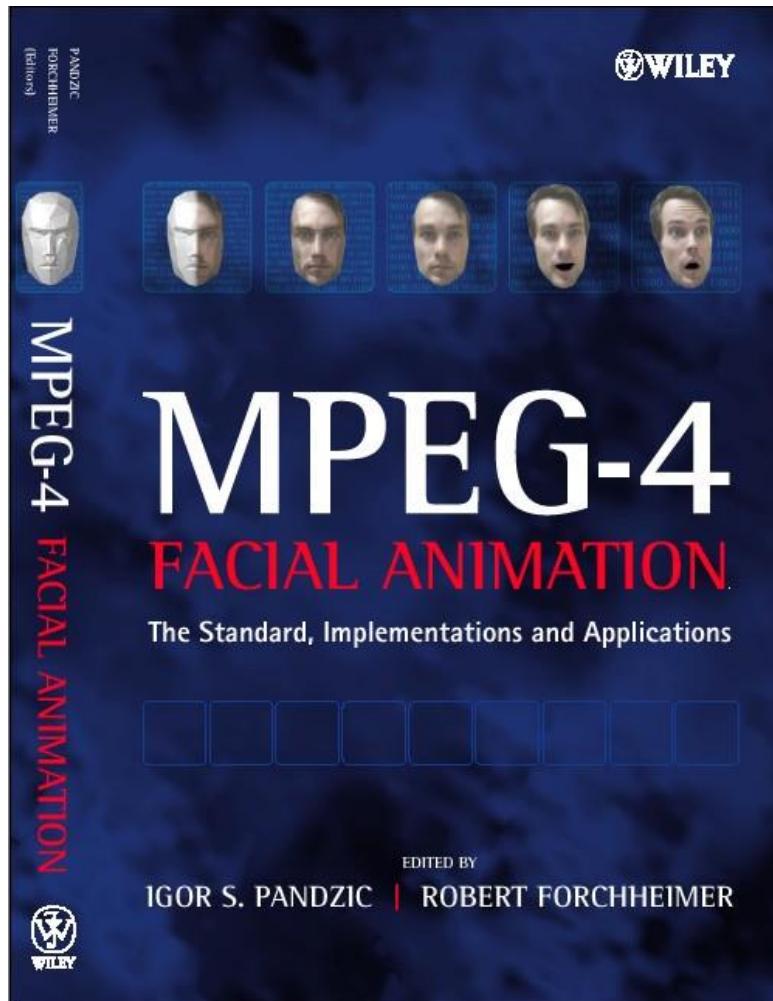
- 68 animacijskih parametara (FAP)
  - 2 parametra visokog nivoa: vizem i izraz
  - 66 parametara niskog nivoa: pomicu karakteristične točke lica
  - Parametri su normalizirani u odnosu na lice: primjena istih parametara na razna lica
- Kvantizacija, te aritmetičko ili DCT kodiranje slijeda parametara



• Feature points affected by FAPs

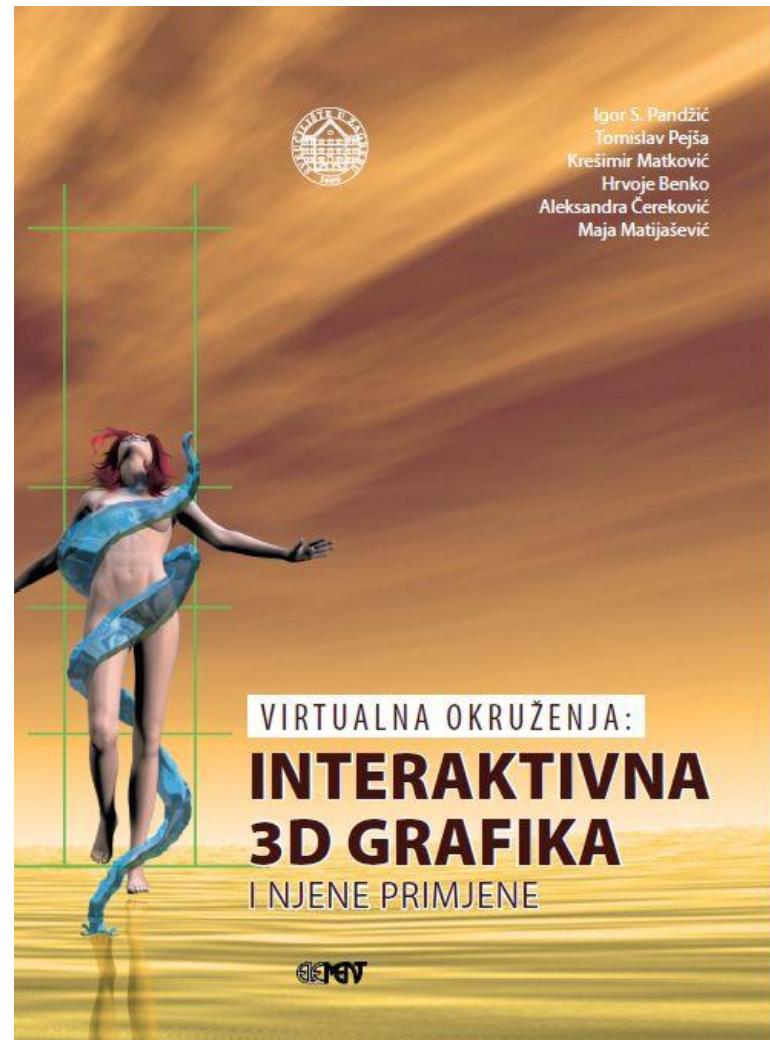
° Other feature points

# Više detalja



# Sintetički sadržaji - literatura

- Graf scene: poglavlje 6
- Norme: poglavlje 6
- Virtualni ljudi: poglavlje 14



# Kako nastaju MPEG norme?

---

- MPEG osnovan 1988, 25 eksperata
- Danas >350 eksperata iz 20 zemalja dolazi na sastanke
- Sastanci 3-5 puta godišnje širom svijeta
- Članovi iz industrije (uglavnom velike tvrtke), te sa sveučilista
- Članstvo kroz Nacionalna Tijela
  - Tvrtke dolaze iz interesa
  - Sveučilista financirana kroz projekte

# Od ideje do međunarodne norme

---

- Odluka o izradi nove norme, njezina svrha
- Poziv za prijedloge (Call for Proposals)
- Verification Model – prvi dokument; optimizacije
- Working Draft – grubi oblik norme; testovi
- Committee Draft – komentari Nacionalnih Tijela
- Final Committee Draft – glasanje NT
- Final Draft International Standard – da/ne glasanje
- International Standard

# Kako izgleda sastanak MPEG-a

---

- Obično 5 dana
  - 3 plenarna sastanka, ostalo po grupama
  - Ulazni i izlazni dokumenti
  - Pritisak je velik, posla je puno
  - Srijeda navečer: zabava!
- 
- Više o MPEG-u:

<http://www.chiariglione.org/mpeg/>

<http://www.mpegif.org>



SVEUČILIŠTE U ZAGREBU  
Fakultet  
elektrotehnike i  
računarstva

Diplomski studij

Informacijska i komunikacijska  
tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

7.

Višeodredišno razašiljanje  
IP multicast

Ak.g. 2020./2021.

# Pregled predavanja

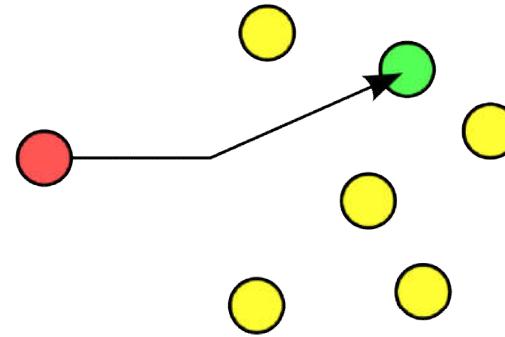
- ◆ pojам višeodredišnog razašiljanja
- ◆ višeodredišno adresiranje
- ◆ evidencija članstva u skupini
  - protokoli za otkrivanje članstva u skupini
    - IGMP, MLD
- ◆ višeodredišno usmjeravanje
  - tehnike izgradnje stabla usmjeravanja
  - protokoli višeodredišnog usmjeravanja
    - DVMRP, PIM-SM

# Višeodredišna komunikacija

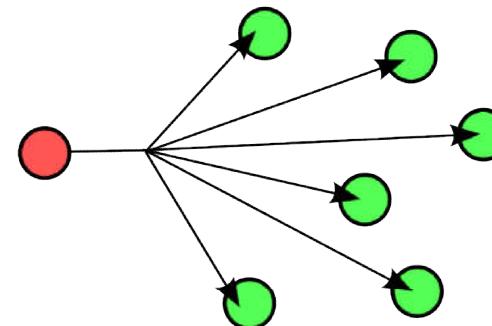
- ◆ višeodredišna komunikacija traži rješenje problema istovremenog slanja istih podataka na više od jednog primatelja
- ◆ posebno su zanimljive primjene gdje se na više krajnjih točaka šalju velike količine podataka, npr:
  - audio ili video konferencija
  - višekorisničke igre
  - računalom podržani zajednički rad
  - distribuirana simulacija
  - poslužitelji višemedijskih sadržaja
  - distribucija softvera

# Metode usmjeravanja u mreži

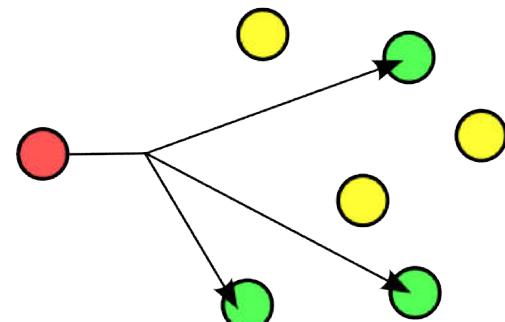
*Unicast 1:1*



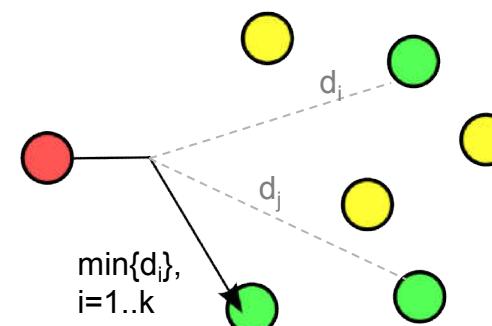
*Broadcast 1:n*



*Multicast 1:m ili m1:m2*

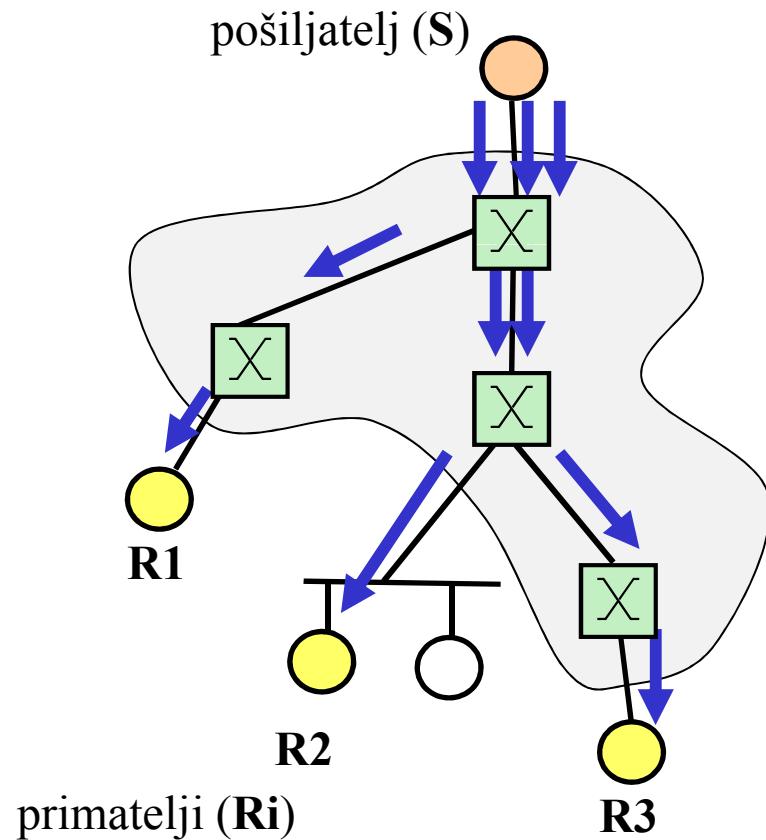


*Anycast 1:1(k)*

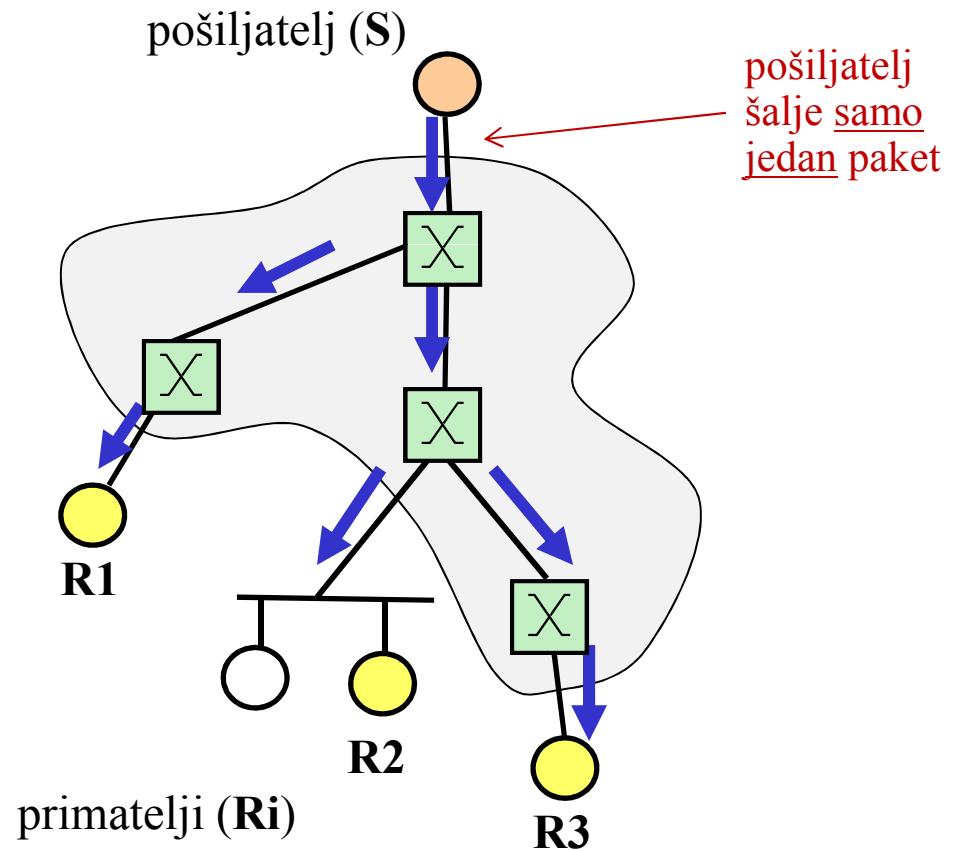


# Načini višeodredišnog razašiljanja

višestruko pojedinačno (unicast)

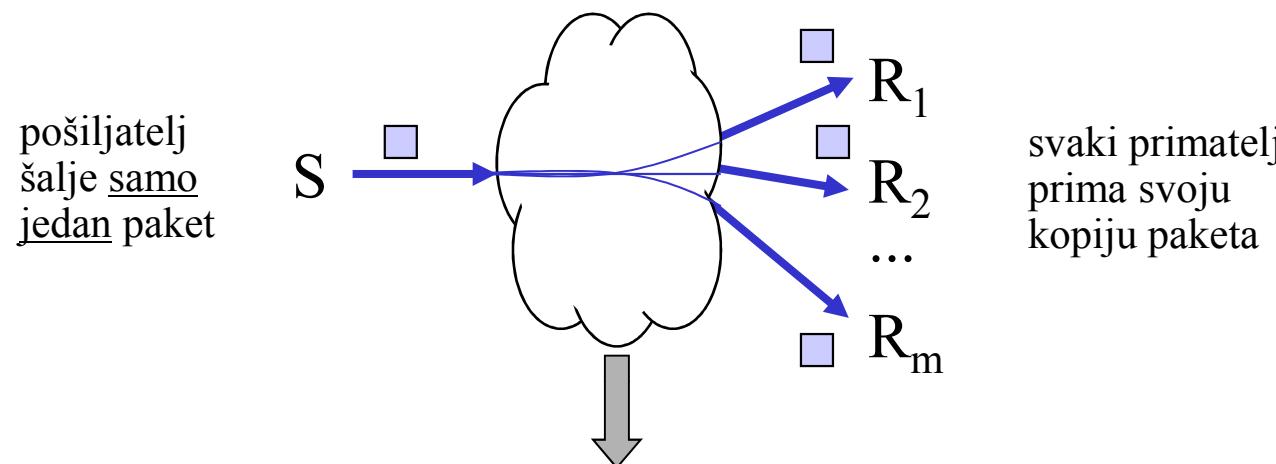


višeodredišno (multicast)

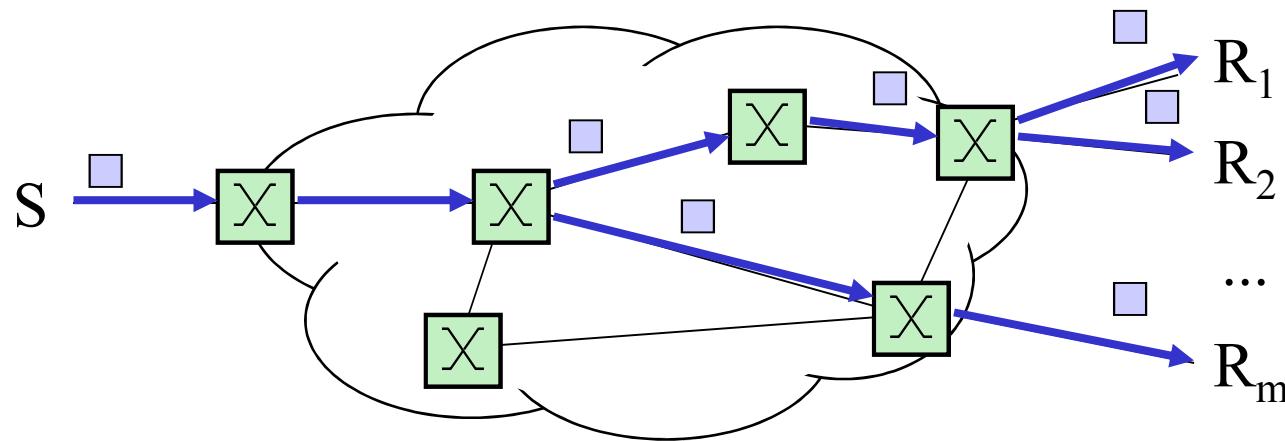


# Ideja i izvedba višeodredišnog razašiljanja

## Ideja



## Izvedba



ključna uloga usmjeritelja (mjesta "grananja" stabla) za uvišestručavanje paketa

# Višeodredišno razašiljanje u Internetu

- ◆ višeodredišno razašiljanje u Internetu (*IP multicast*) je slanje jednog IP datagrama skupini, koju čini (0 ili više) računala određenih **jednom IP odredišnom adresom**
- ◆ IP multicast ne mijenja osnovni (datagramske) način komunikacije u Internetu
- ◆ Jednostavan model usluge:
  - *primatelji* izražavaju interes
  - *pošiljatelji* šalju podatke
  - *usmjeritelji* surađuju kako bi dostavili podatke od pošiljatelja do svih zainteresiranih primatelja

# Elementi višeodredišnog razašiljanja

- ◆ Pitanja koja treba riješiti:

- 1. adresiranje skupine primatelja
  - IP adrese klase D
- 2. pitanje pripadnosti skupini primatelja
  - Internet Group Management Protocol (IGMP), ili Multicast Listener Discovery Protocol (MLDP)
  - podrška na krajnjim računalima i usmjeriteljima
- 3. višeodredišno usmjeravanje
  - protokoli višeodredišnog usmjeravanja
    - ▶ pronalaženje puta
    - ▶ prosljeđivanje paketa kroz mrežu

# Klasifikacija višeodredišnog razašiljanja

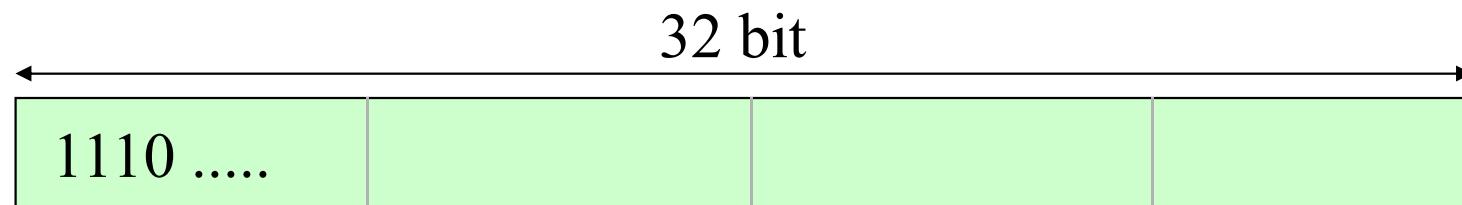
- ◆ bilo koji pošiljatelj unutar grupe (*Any Source Multicast*, skr. ASM)
  - komunikacija tipa *više-na-više*
  - datagram poslan s bilo koje izvorišne adrese na višeodredišnu IP-adresu G dostavlja se na svaki socket prijavljen za primanje prometa skupine G
  - ASM je pogodan za aplikacije gdje su čvorovi ujedno i primatelji i pošiljatelji (npr. audio konferencija, aplikacije za zajednički rad)
- ◆ zadani pošiljatelj (*Source Specific Multicast*, skr. SSM)
  - isključivo komunikacija tipa *jedan-na-više*
  - datagram poslan s izvorišnom IP-adresom S i odredišnom IP-adresom G u zadanom rasponu dostavlja se na svaki (i samo takav) socket koji je prijavljen za promet od S prema G
  - SSM je pogodan za distribucijske aplikacije gdje se zna tko su pošiljatelji prije nego što aplikacija započne (npr. distribucija a/v sadržaja)

# Klasifikacija višeodredišnog razašiljanja

- ◆ bilo koji pošiljatelj unutar grupe  
*Any Source Multicast*. skr. ASM
  - omogućuje komunikaciju tipa *više-na-više* (što pokriva i jedan-na-više)
  - datagram poslan s bilo koje izvorišne adrese na višeodredišnu IP-adresu G dostavlja se na svaki *socket* prijavljen za primanje prometa skupine G
  - ASM je pogodan za aplikacije gdje su čvorovi ujedno i primatelji i pošiljatelji (npr. audio konferencija, aplikacije za zajednički rad)
- ◆ zadani pošiljatelj  
*Source Specific Multicast*, skr. SSM
  - isključivo komunikacija tipa *jedan-na-više*
  - datagram poslan s izvorišnom IP-adresom S i odredišnom IP-adresom G u zadanom rasponu dostavlja se na svaki (i samo takav) *socket* koji je prijavljen za promet od S prema G
  - SSM je pogodan za distribucijske aplikacije gdje se zna tko su pošiljatelji prije nego što aplikacija započne (npr. distribucija a/v sadržaja)

# Višeodredišna (multicast) adresa

- ◆ višeodredišna adresa je IP adrese klase D

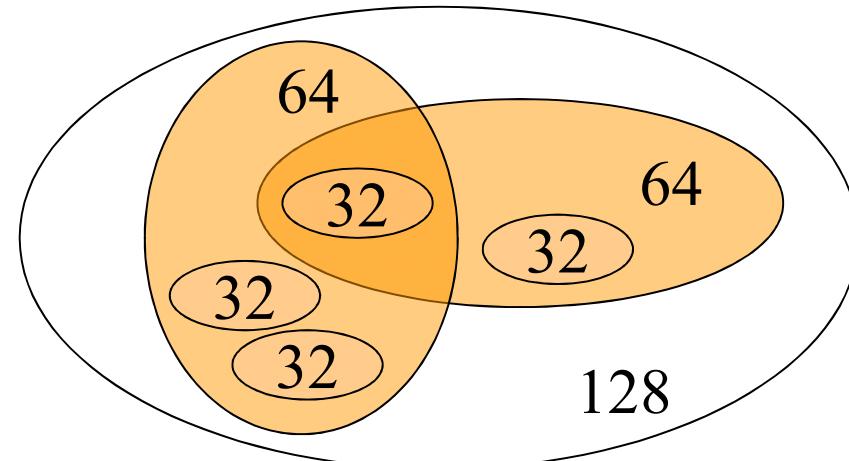


Adresni prostor: 224.0.0.0 -- 239.255.255.255

- ◆ vrste višeodredišnih adresa:
  - stalno dodijeljene i rezervirane adrese
  - adrese s administrativnim rasponom
  - ostale adrese (dodjeljuju se dinamički)

# Dinamička dodjela višeodredišnih adresa

- ◆ dinamička dodjela adresa je potpuno distribuirana
- ◆ adrese se dodjeljuju slučajnim izborom; prihvatljivo zbog relativno male vjerojatnosti preklapanja
- ◆ parametri mogućeg preklapanja adresa:
  - veličina adresnog prostora
  - trajanje
  - raspon (određen preko TTL-a ili administrativno)



Primjer raspona ograničenog putem TTL-a

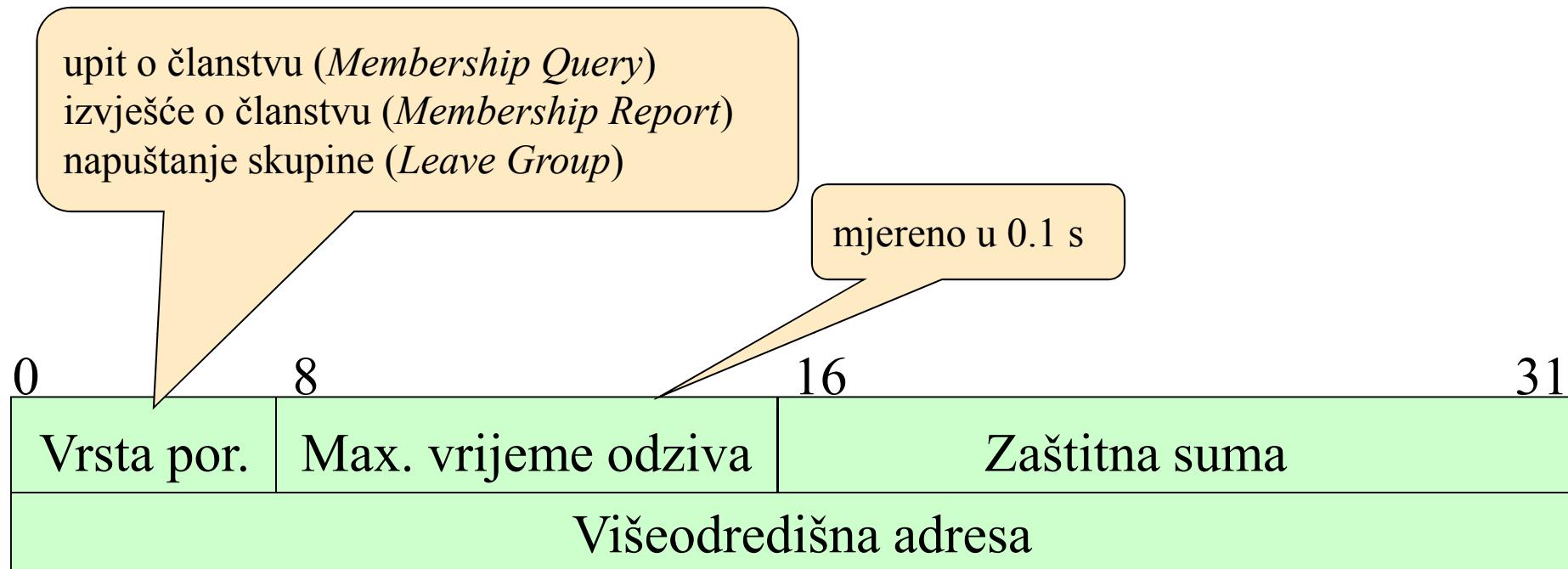
# Svojstva skupine primatelja

- ◆ skupina primatelja je određena jednom odredišnom IP adresom G
  - ◆ skupina može biti stalna ili privremena
  - ◆ višeodredišno razaslani datagram dostavlja se svakom članu skupine s istom razine usluge (“*best-effort*”) kao kod pojedinačnog razašiljanja
  - ◆ članstvo u skupini je dinamično
  - ◆ evidentiranje članova skupine se vrši putem protokola IGMP ili MLD (za IPv6)
-

# Internet Group Management Protocol (IGMP)

- ◆ **IGMPv1**
  - dvije vrste poruka:
    - upit o članstvu (*Membership Query*) ("ima li tko zainteresiran za multicast?")
    - prijava članstva (*Membership Report*) ("zanima me multicast skupina G")
  - odjava članstva implicitno, putem isteka vremenske kontrole (posljedica - spora prilagodba topologiji - promet nepotrebno teče još neko vrijeme)
- ◆ **IGMPv2**
  - dodatno, omogućuje upit o članstvu za pojedinu skupinu ("ima li koga zainteresiranog za multicast skupinu G?")
  - rješava problem spore prilagodbe topologiji uvođenjem poruke za objavu odlaska (*Leave Group*)
- ◆ **IGMPv3** aktualna verzija
  - dodatno, primatelj može ograničiti članstvo u skupini na odabrane pošiljatelje, čime se omogućuje filtriranje višeodredišnog prometa ("zanimaju me samo pošiljatelji Si iz multicast skupine G", ili "... svi osim pošiljatelja Si iz skupine G")

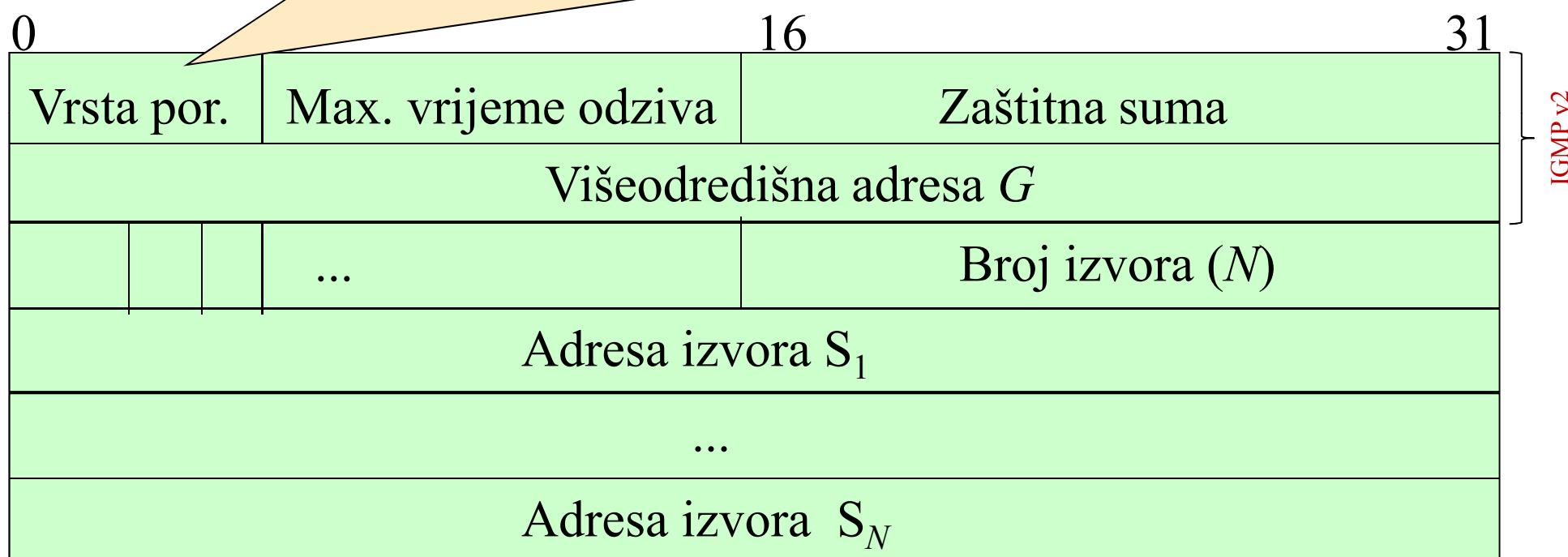
# Format IGMPv2 poruke



- ◆ IGMP poruka se prenosi IP datagramom uz TTL=1 i postavljenu IP Router Alert opciju

# Format IGMPv3 upita

upit o članstvu (*Membership Query*) - općeniti, za grupu G, za grupu G i izvor S  
izvješće o članstvu v3 (*Membership Report*)  
+ poruke IGMPv2 radi kompatibilnosti



# Uloge u komunikaciji IGMP-om

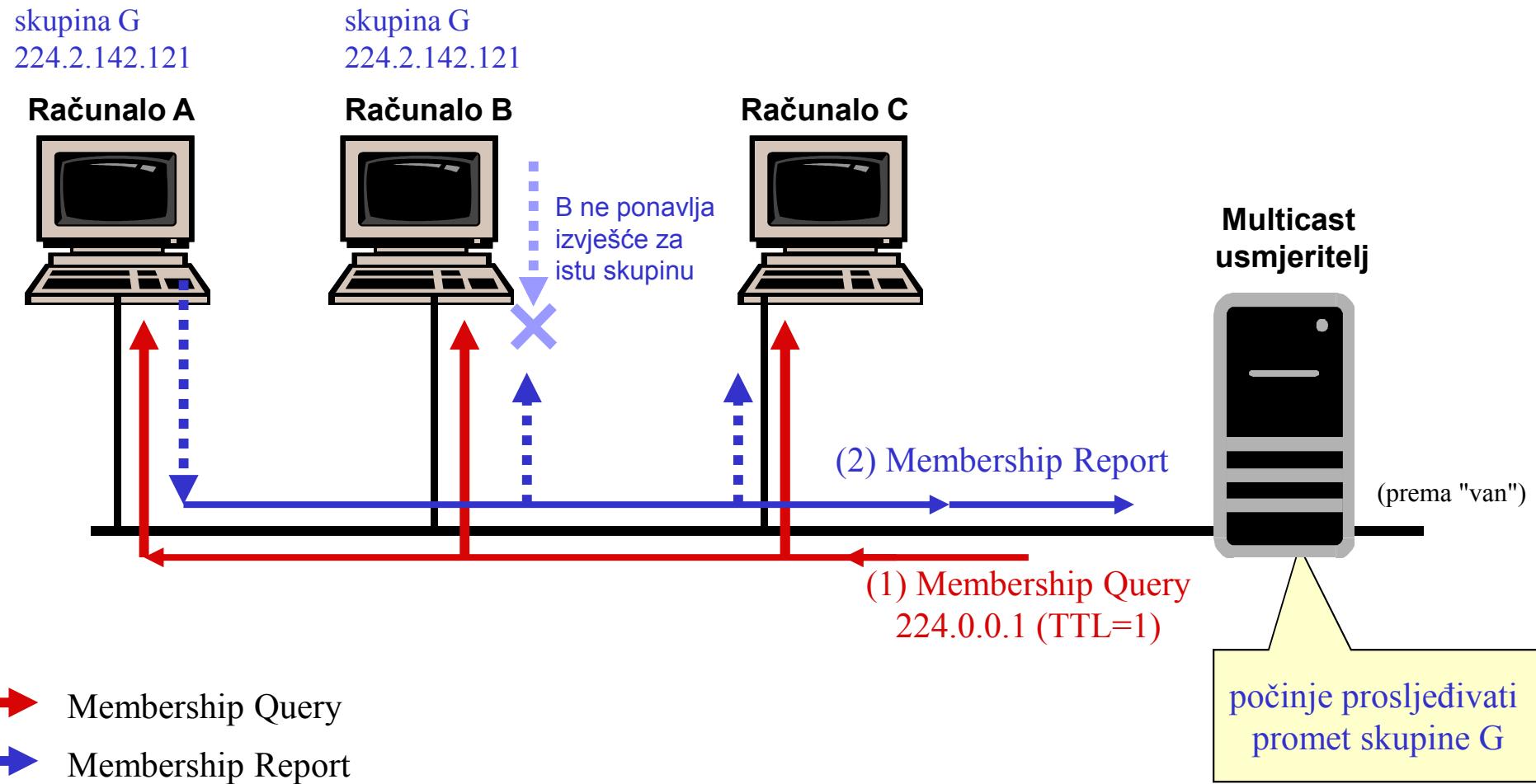
## ◆ član skupine (npr. računalo):

- prilikom pridruživanja skupini računalo programira svoje Ethernet sučelje za prijam i šalje IGMP izvješće o članstvu (*Membership Report*)
- prilikom napuštanja skupine računalo programira svoje Ethernet sučelje za prestanak prijama i šalje odgovarajuću IGMP poruku (*Leave Group*)

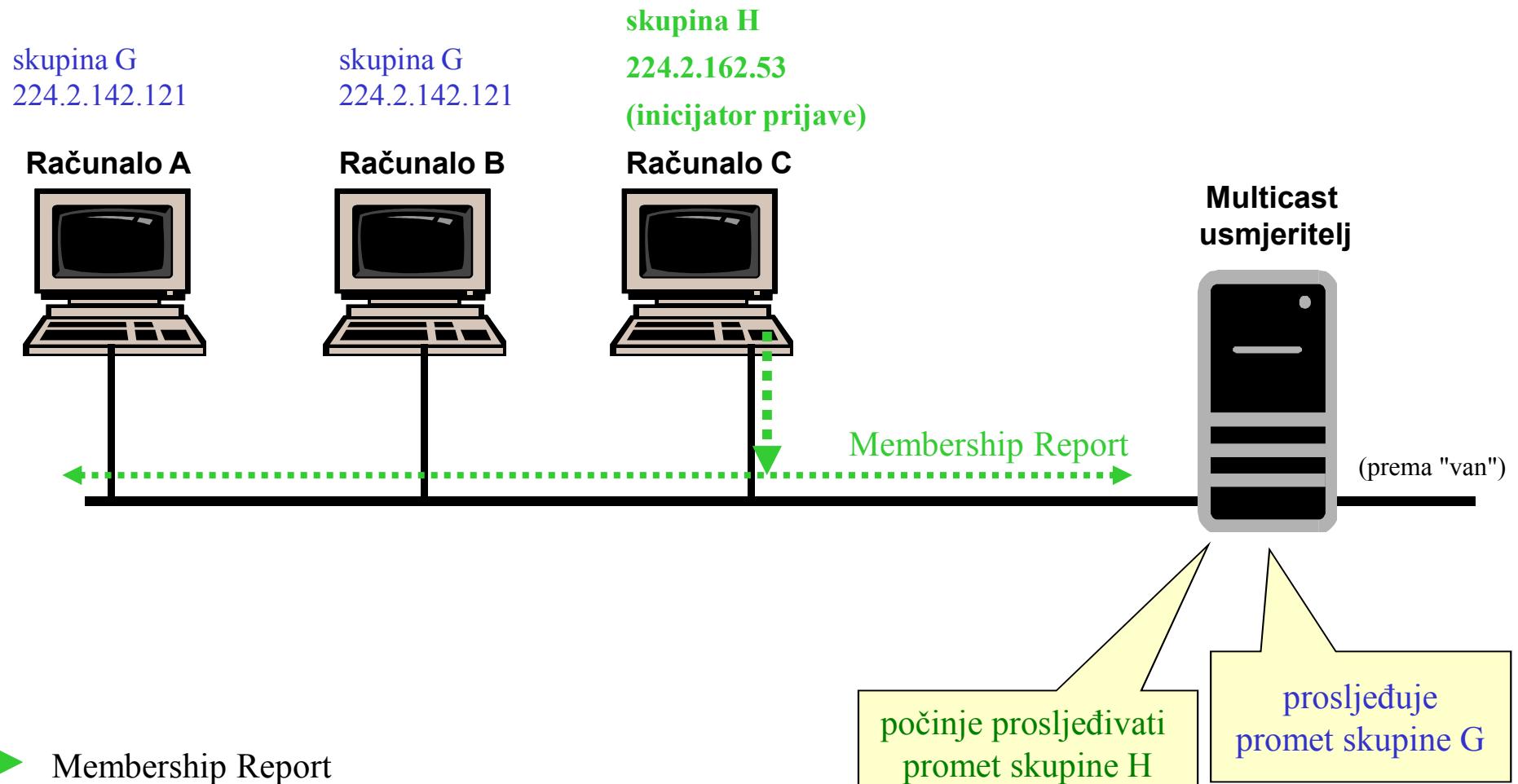
## ◆ usmjeritelj:

- na temelju izvješća o članstvu počinje prosljeđivati promet adresiran na skupinu
- na temelju primitka poruke o napuštanju skupine od posljednjeg člana prestaje prosljeđivati promet adresiran na skupinu
- povremeno provjerava ima li i dalje zainteresiranih primatelja; ako nema, prestaje prosljeđivati promet

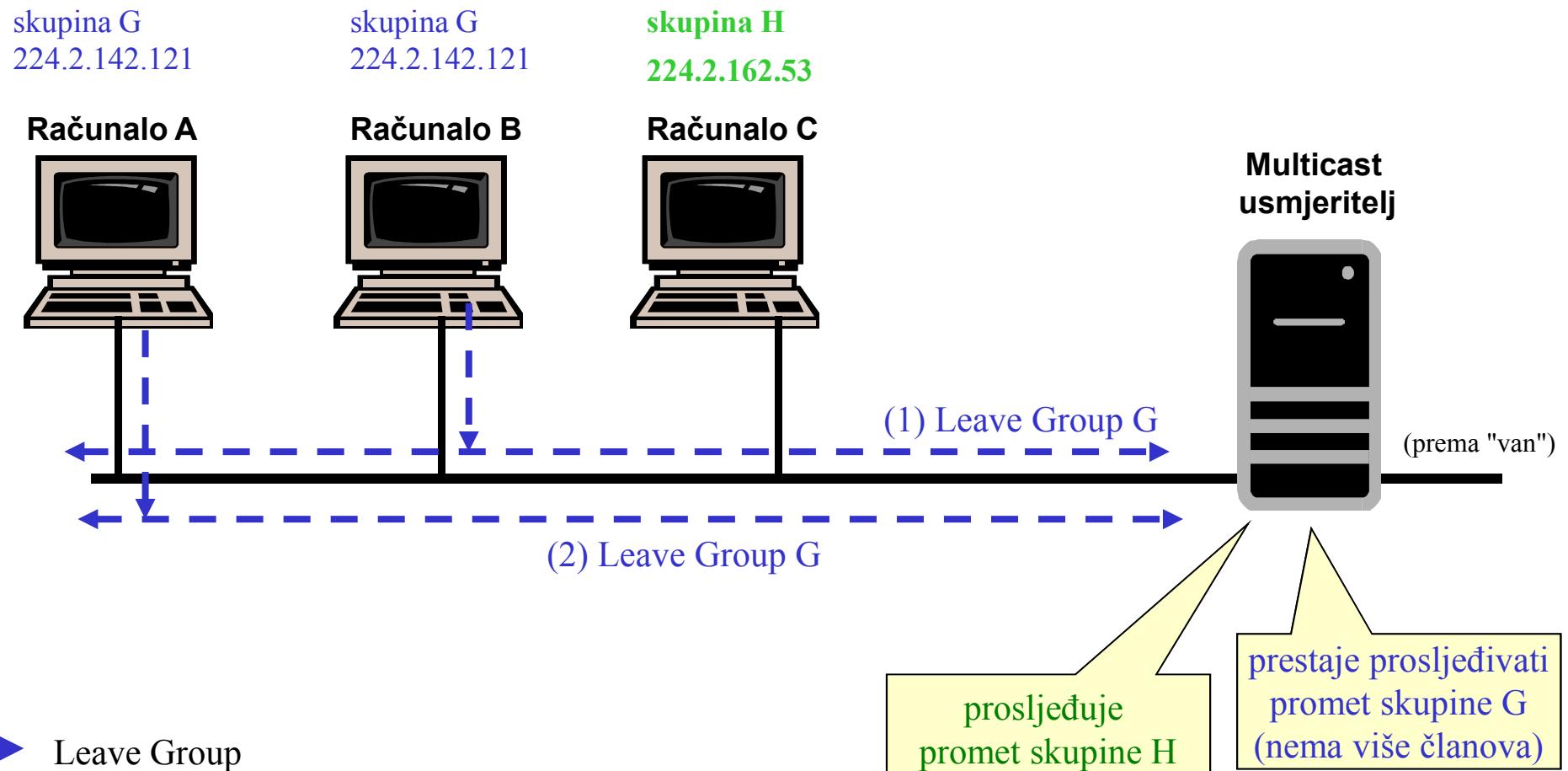
# IGMP: prijava u skupinu na upit o članstvu



# IGMP: samostalna prijava



# IGMP: odjava



# Protokol *Multicast Listener Discovery*

- ◆ Multicast Listener Discovery (MLD) je protokol za upravljanje višeodredišnim skupinama za IPv6
  - aktualna verzija je MLDv2
- ◆ definiran u okviru protokola ICMPv6
- ◆ tri vrste poruka (slično IGMPv3):
  - *Multicast Listener Query* (smisao *Membership Query*)
  - *Multicast Listener Report* (smisao *Membership Report*)
  - *Multicast Listener Done* (smisao *Leave Group*)

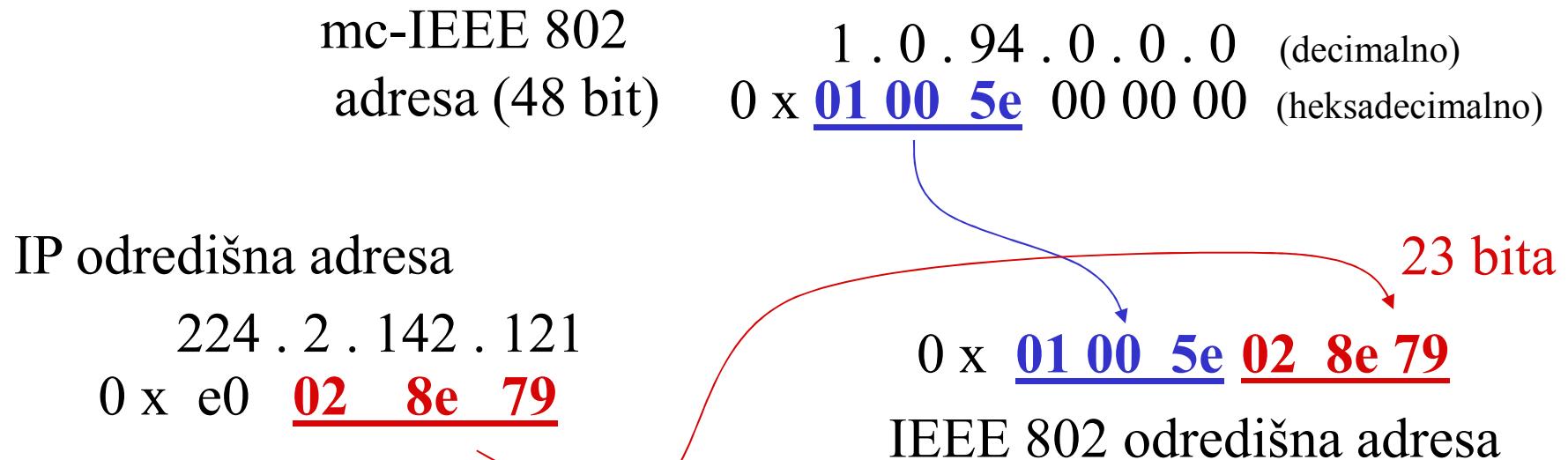
# Višeodredišno (*multicast*) usmjeravanje

# Višeodredišno (*multicast*) usmjeravanje

- ◆ usmjeritelji surađuju kako bi dostavili podatke od pošiljatelja do svih zainteresiranih primatelja
- ◆ svaki član skupine (pošiljatelj ili primatelj) komunicira sa sebi najbližim usmjeriteljem
- ◆ obrada datagrama:
  - TTL=1: **lokalno** višeodredišno razašiljanje
  - TTL>1: datagram se proslijeđuje drugim mrežama u kojima ima članova odredišne skupine -- mora postojati podrška u usmjeriteljima, odn. u mreži mora biti izведен **protokol višeodredišnog usmjeravanja**

# Lokalno višeodredišno razašiljanje

- ◆ izvedba lokalnog višeodredišnog razašiljanja ovisna je o mrežnoj tehnologiji
- ◆ primjer: IEEE 802.x mreže (npr. Ethernet)

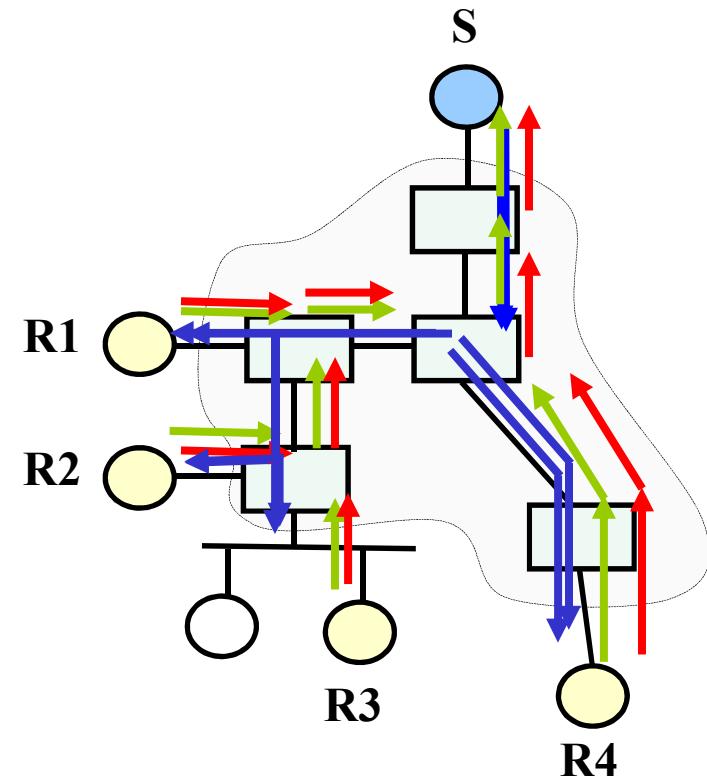


# Protokoli višeodredišnog usmjeravanja

- ◆ isto kao kod običnog usmjeravanja:
  - podjela s obzirom na područje djelovanja:
    - unutar autonomnog sustava (Interior Gateway Protocol, IGP)
    - između autonomnih sustava (Exterior Gateway Protocol, EGP)
  - podjela s obzirom na algoritam usmjeravanja
    - statički (neadaptivan)
    - dinamički (adaptivan)
      - algoritam vektora udaljenosti, algoritam stanja linka
- ◆ posebnosti višeodredišnog usmjeravanja:
  - zgusnuti (engl. *dense*) i rijetko popunjeni (engl. *sparse*) način rada (objašnjeni u nastavku)
  - sprega s protokolom jednoodredišnog usmjeravanja

# Podjela na “sparse” i “dense” način rada (1)

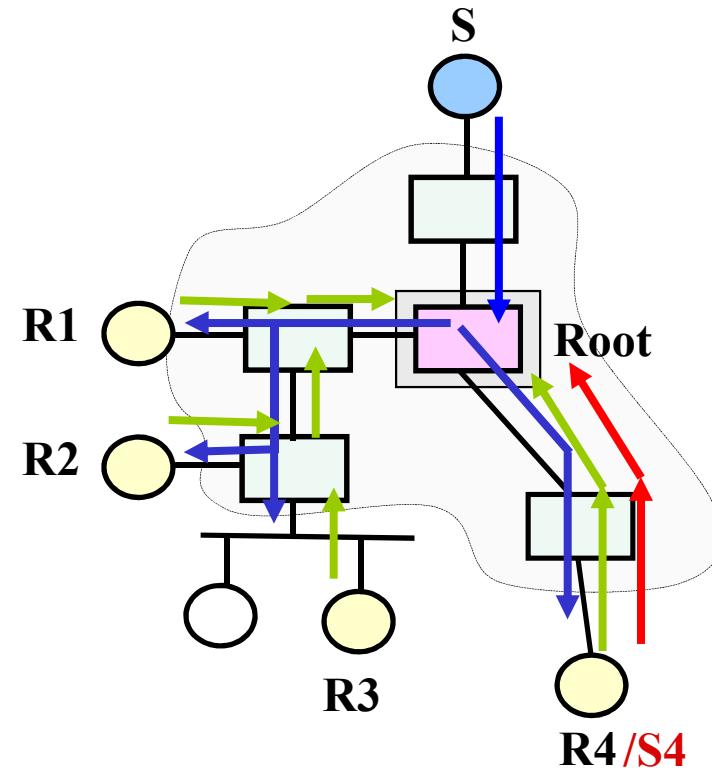
- ◆ dva načina rada, *sparse* (rijetki, raštrkani) i *dense* (gusti, zbijeni), ovisno o raspodjeljenosti primatelja i pošiljatelja u mreži
  - klasični protokoli su dizajnirani za “dense mode”, tj. grupirane primatelje i pošiljatelje
  - gradi se usmjereni stablo “ukorijenjeno” u pošiljatelju
  - osnovna ideja: preplavljivanje mreže višeodredišnim prometom i naknadno podrezivanje nepotrebnih grana



**Primjer: Dense-Mode Multicast**  
stvara se usmjereni stablo s  
korijenom kod pošiljatelja

# Podjela na “sparse” i “dense” način rada (2)

- ◆ “sparse mode” je noviji način
  - rješava problem učinkovitosti usmjeravanja za skupine koje nisu “zbijene”
  - pogodan za naširoko razasute, “raštrkane” primatelje
  - gradi se zajedničko stablo za sve primatelje i pošiljatelje
  - osnovna ideja: izgradnja stabla počinje od unaprijed zadane središnje točke
  - nema preplavljanja prometom mreža u kojima nema primatelja



**Primjer: Sparse-Mode Multicast stvara se zajedničko stablo sa središnjom točkom**

# Prosljeđivanje datagrama

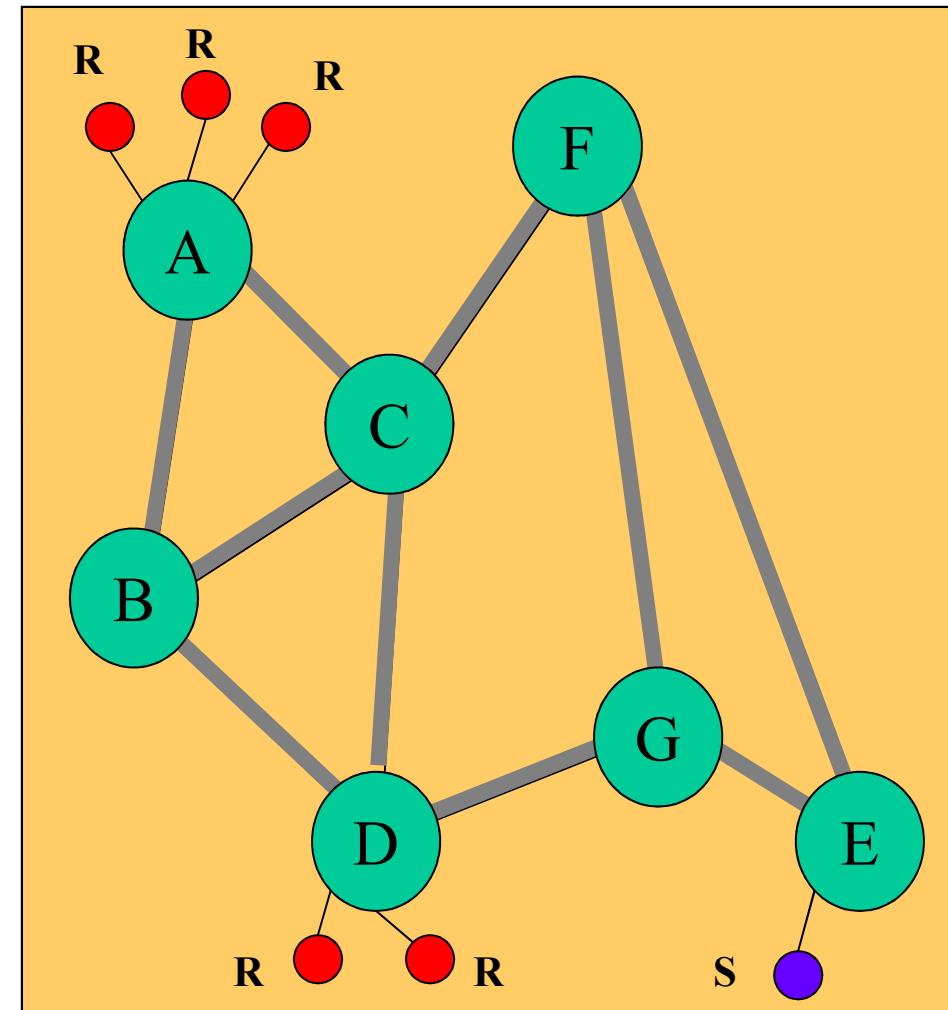
- ◆ glavna razlika između jednoodredišnog i višeodredišnog usmjeravanja je način prosljeđivanja datagrama
  - zašto? -- višeodredišna adresa definira skupinu sučelja - problem usmjeravanja proširuje se na skup puteva do više odredišta
- ◆ dvije skupine načina prosljeđivanja datagrama adresiranih na višeodredišnu adresu
  - jednostavnii načini
    - preplavljivanje mreže
    - ograničena primjena
  - složeni načini - stablo usmjeravanja
    - stablo za dostavu datagrama s korijenom na pošiljatelju (po jedno stablo za svakog pošiljatelja)
    - zajedničko stablo (jedno stablo za sve pošiljatelje)

# Tehnike korištene pri izgradnji stabla

- ◆ preplavljivanje  
(engl. *flooding*)
- ◆ razapinjuće stablo  
(engl. *spanning tree*)
- ◆ razapinjuće stablo od zadanog pošiljatelja  
(engl. *Reverse Path Broadcasting, RPB*)
- ◆ odrezano razapinjuće stablo od zadanog pošiljatelja  
(engl. *Truncated Reverse Path Broadcasting, TRPB*)
- ◆ višeodredišno usmjeravanje povratnom stazom  
(engl. *Reverse Path Multicasting, RPM*)

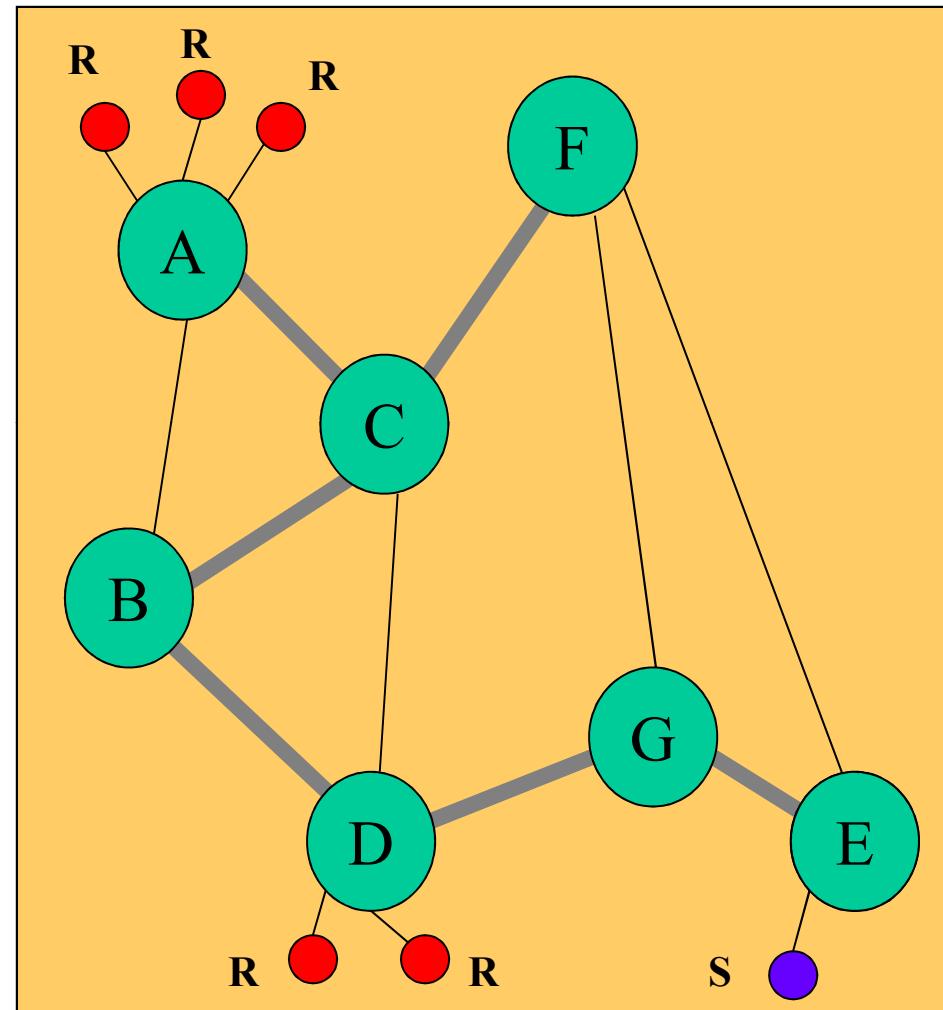
# Preplavljanje (*flooding*)

- ◆ kad usmjeritelj primi paket, provjerava je li ga “već viđao”; ako nije, prosljeđuje ga **na sva sučelja osim onog po kojem je primio paket**; inače ga odbacuje
- ◆ svaki usmjeritelj vodi računa o “već viđenim” paketima
- ◆ nedostatak: nepotrebno trošenje resursa jer promet preplavljuje cijelu mrežu



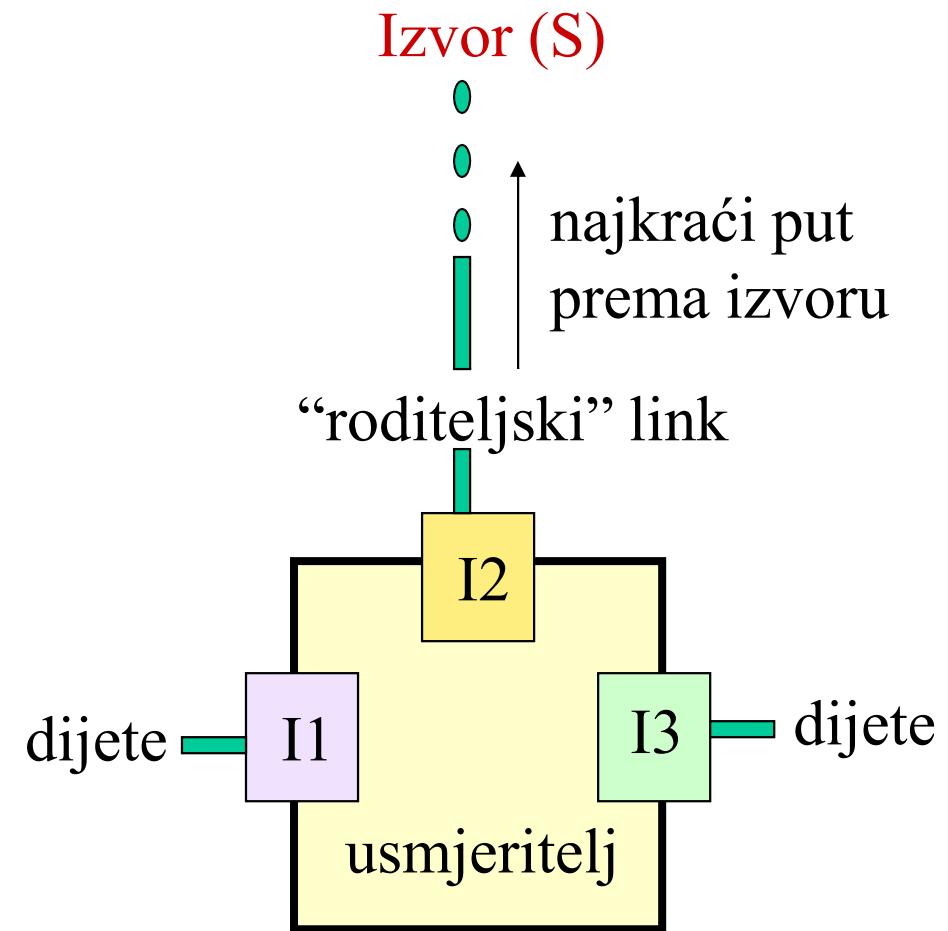
# Razapinjuće stablo (*spanning tree*)

- ◆ razapinjuće stablo je stablo koje osigurava minimalnu povezivost za bilo koji par (pošiljatelj, primatelj); ne mora biti optimalno
- ◆ kad usmjeritelj primi paket, prosljeđuje ga **samo na ona sučelja koja su dio razapinjućeg stabla** (opet, osim dolznog); inače ga odbacuje
- ◆ paket se uvišestručuje samo na mjestima gdje se stablo grana



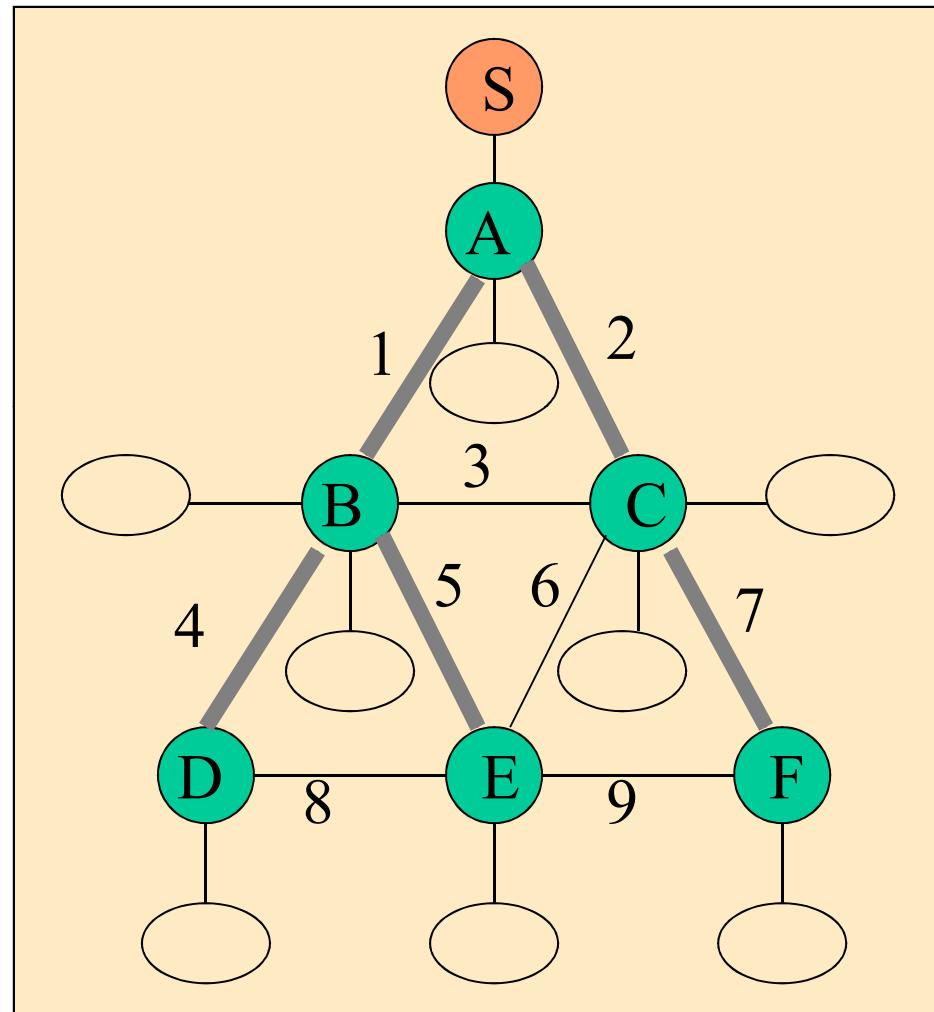
# Reverse Path Broadcasting, RPB

- ◆ Gradi se razapinjuće stablo sa svojstvom najkraćeg puta za svaki par (pošiljatelj, skupina)
- ◆ svaki usmjeritelj postavlja smjer prosljeđivanja tako da kao dolazni, tzv. "roditeljski link", uzima ono sučelje koje je na najkraćem putu "unatrag" tj. prema pošiljatelju
- ◆ koji je najkraći put "zna" se na temelju tablice jednoodredišnog usmjeravanja



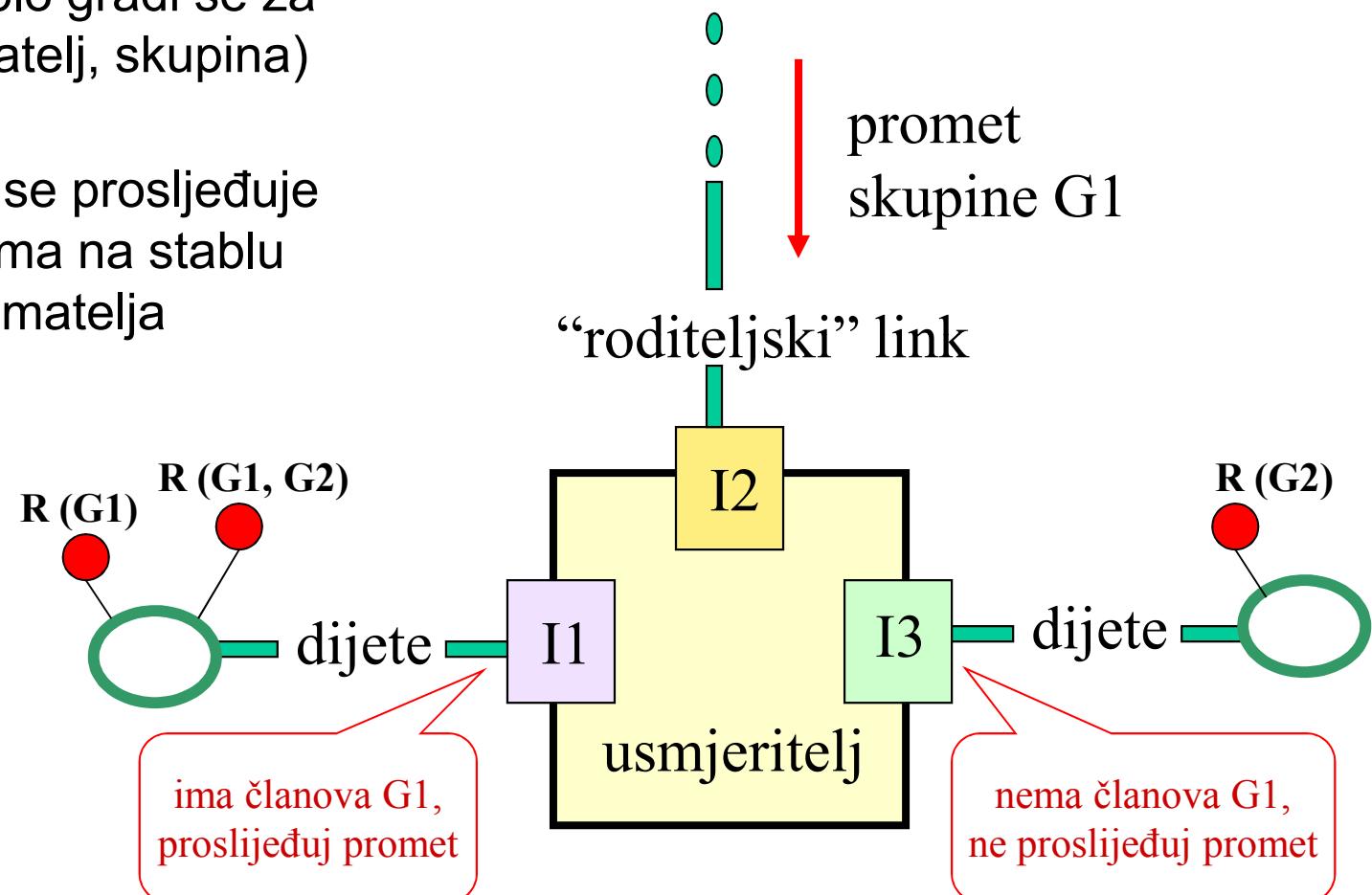
# RPB - primjer

- ◆ razapinjuće stablo ima korijen kod pošiljatelja
- ◆ nedostatak: ne vodi se računa o članstvu u skupinama, promet dolazi do svih usmjeritelja, neovisno o interesu primatelja



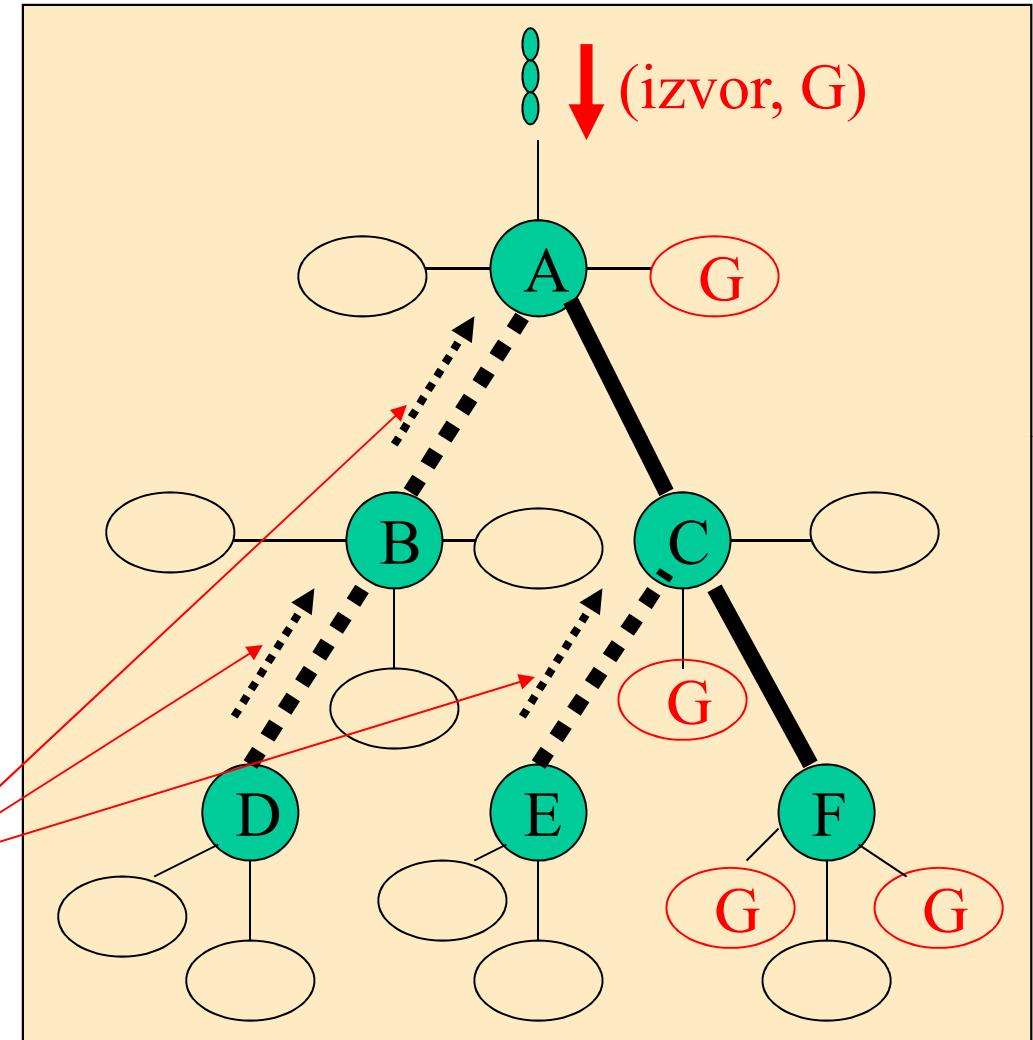
# Truncated Reverse Path Broadcasting, TRPB

- ◆ razapinjuće stablo gradi se za svaki par (pošiljatelj, skupina) kao i za RPB
- ◆ ali se datagram se prosljeđuje samo po sučeljima na stablu koja vode do primatelja
- ◆ za evidentiranje članova skupine koristi se **IGMP**

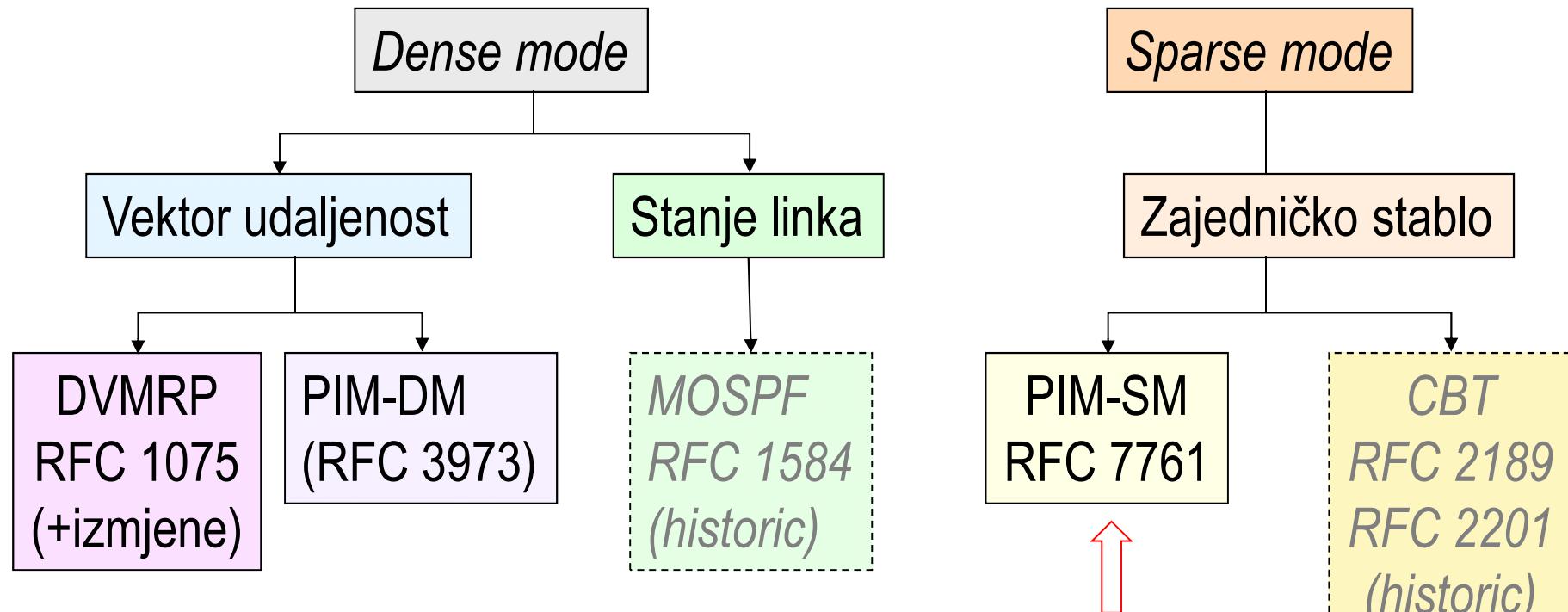


# Reverse Path Multicasting, RPM

- ◆ stvara se razapinjuće stablo sa sljedećim svojstvima:
  - sadrži mreže u kojima ima članova skupine
  - sadrži usmjeritelje koji se nalaze na najkraćem putu prema tim mrežama
  - grane na kojima nema primatelja se podrezuju
- ◆ način rada:
  - prvi paket se proslijeđuje koristeći TRPB
  - usmjeritelji koji nemaju članova šalju poruke o rezerviranju
  - stablo se periodički osvježava preko TRPB



# Pregled protokola višeodredišnog usmjeravanja



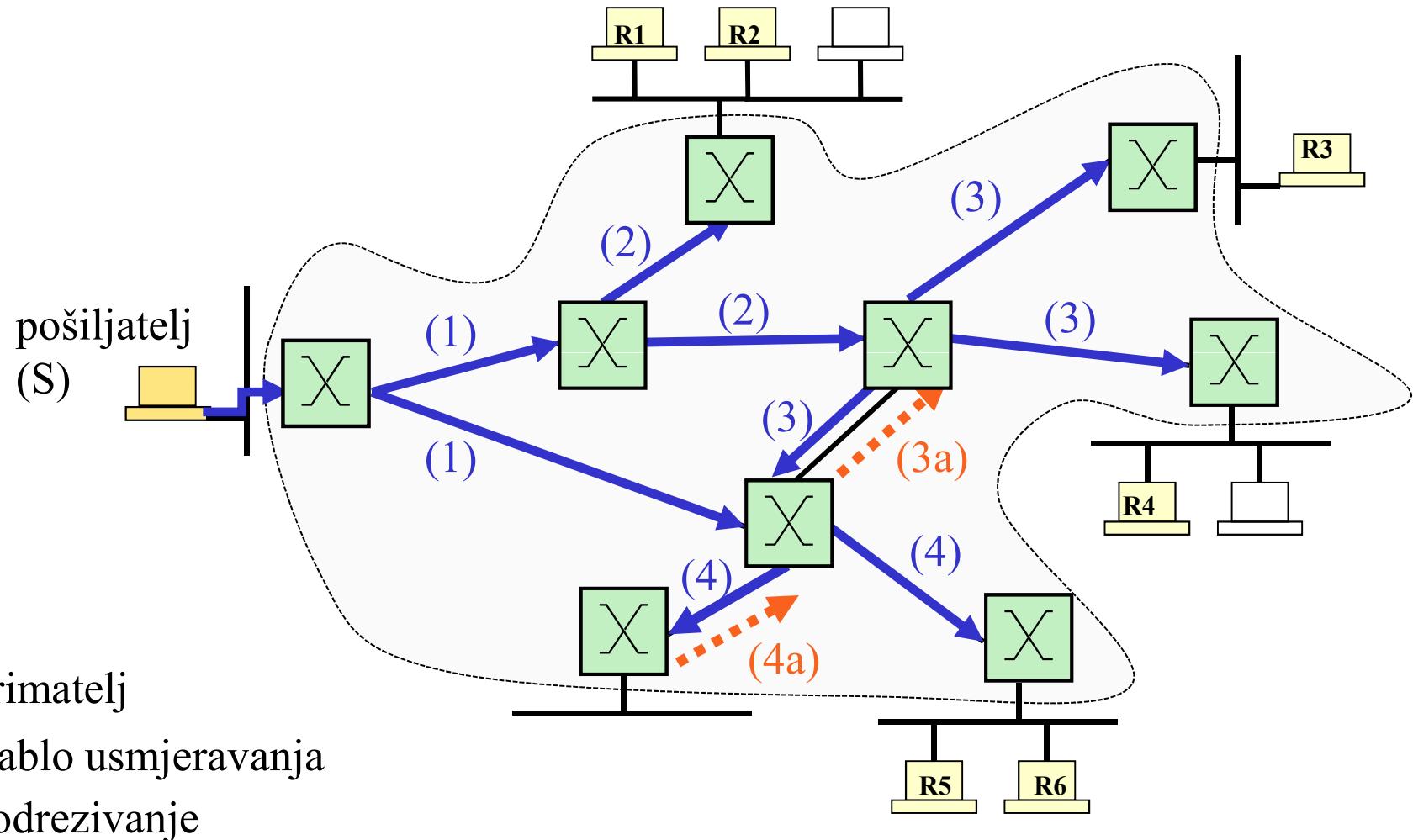
DVMRP	Distance Vector Multicast Routing Protocol
MOSPF	Multicast Extensions to Open Shortest Path First
PIM-SM	Protocol Independent Multicast - Sparse Mode
PIM-DM	Protocol Independent Multicast - Dense Mode
CBT	Core Based Trees (v2)

U primjeni je  
najzastupljeniji  
PIM-SM.

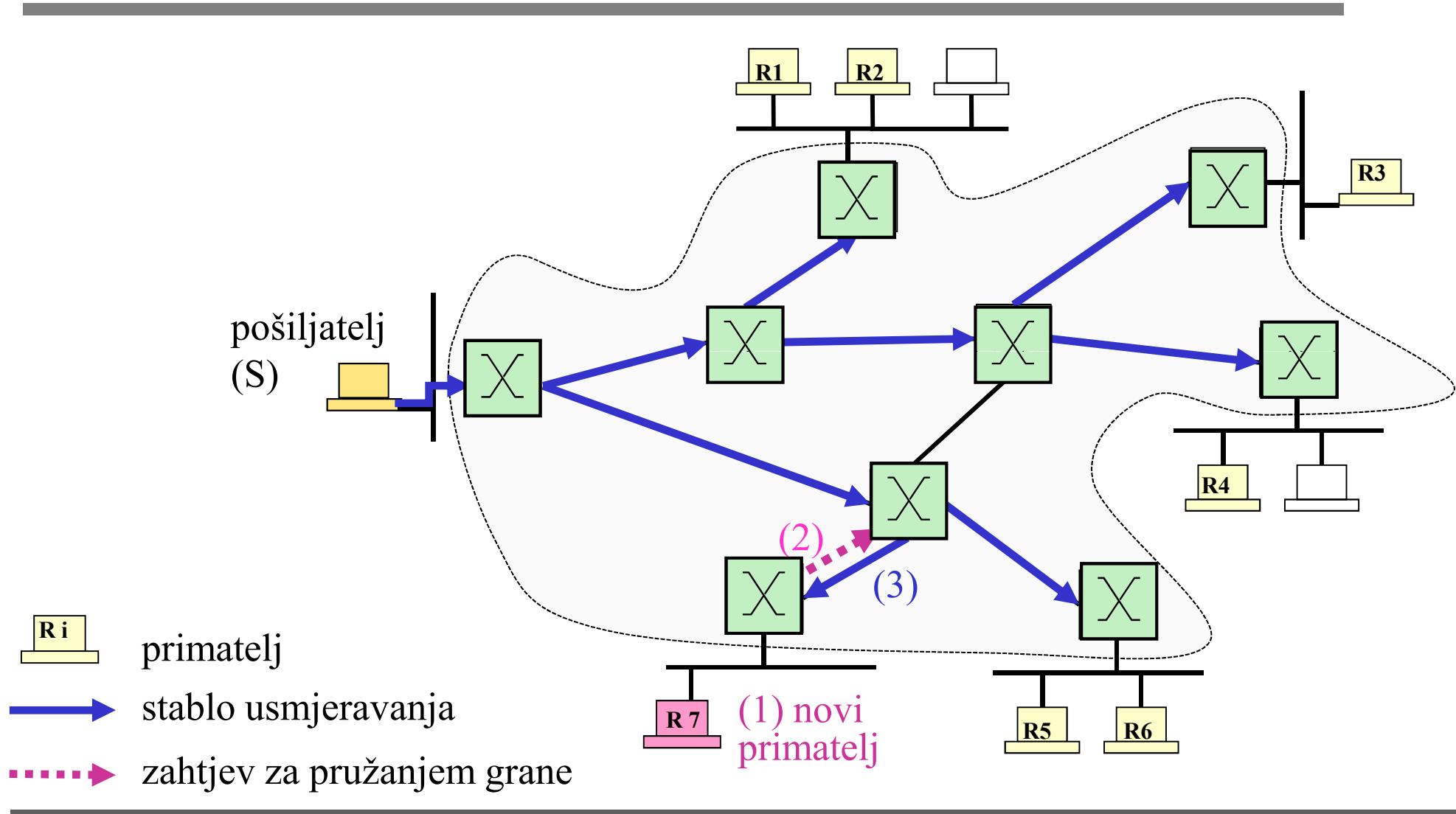
# Distance Vector Multicast Routing Protocol

- ◆ DVMRP spada u skupinu protokola koji se temelje na računanju vektora *udaljenosti*
- ◆ DVMRP radi u sprezi s protokolom *Routing Information Protocol* (RIP) za pojedinačno usmjeravanje
- ◆ DVMRP dinamički gradi stablo usmjeravanja sa svojstvom najkraćeg puta, za svakog pošiljatelja
  - prednosti: jednostavna izvedba, stablo najkraćeg puta
  - nedostaci: nije pogodan za vrlo velike mreže, periodičke *broadcast* i *prune* poruke, sporo prilagođavanje promjenama u topologiji

# DVMRP - preplavljanje i podrezivanje



# DVMRP - pružanje nove grane



- ◆ PIM-SM gradi zajedničko stablo usmjeravanja za svaku skupinu, s mogućnošću prelaska na stablo najkraćeg puta za svakog pošiljatelja
  - stablo se gradi od središnje točke, nazvane *Rendezvous Point (RP)*
  - uvodi se pojam odabranog usmjeritelja, *Designated Router (DR)*
    - DR "zastupa" računala na LAN-u prema ostalim PIM-SIM usmjeriteljima
- ◆ PIM-SM je neovisan o protokolu pojedinačnog usmjeravanja (može raditi s bilo kojim od njih)

# PIM-SM pojmovi i podatkovne strukture

- ◆ Pojmovi i podatkovne strukture:
    - Rendezvous Point (RP)
    - Designated Router (DR)
    - Multicast Routing Information Base (MRIB)
    - Reverse Path Forwarding (RPF) Neighbor
    - Tree Information Base (TIB)
-

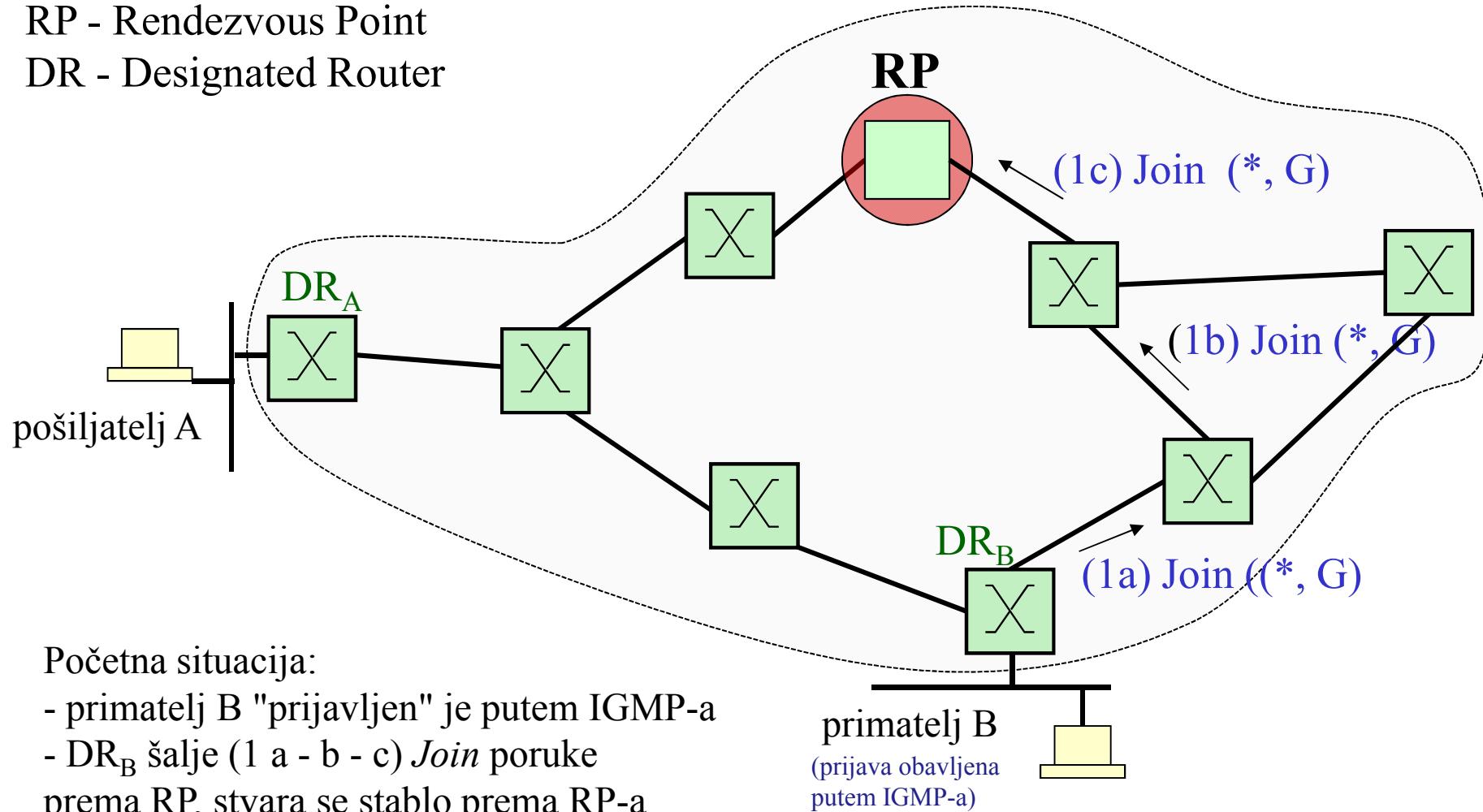
# Način rada protokola PIM-SM

- ◆ PIM-SM usmjerava podatkovne pakete od pošiljatelja prema primateljima
  - primatelji i pošiljatelji ne znaju unaprijed jedni za druge
  - primatelji i pošiljatelji mogu se pridružiti skupini G (ili otići) bilo kada
- ◆ način rada može se opisati kroz tri faze (mogu se odvijati i istovremeno)
  - 1. faza: uspostava stabla usmjeravanja "ukorijenjenoga" u RP-u, skr. "RP-stabla" (*RP Tree*)
  - 2. faza: zaustavljanje registracije (*Register-Stop*)
  - 3. faza: prelazak na stablo najkraćeg puta (*Shortest Path Tree*)

# Faza 1: uspostava RP-stabla (1/4)

RP - Rendezvous Point

DR - Designated Router

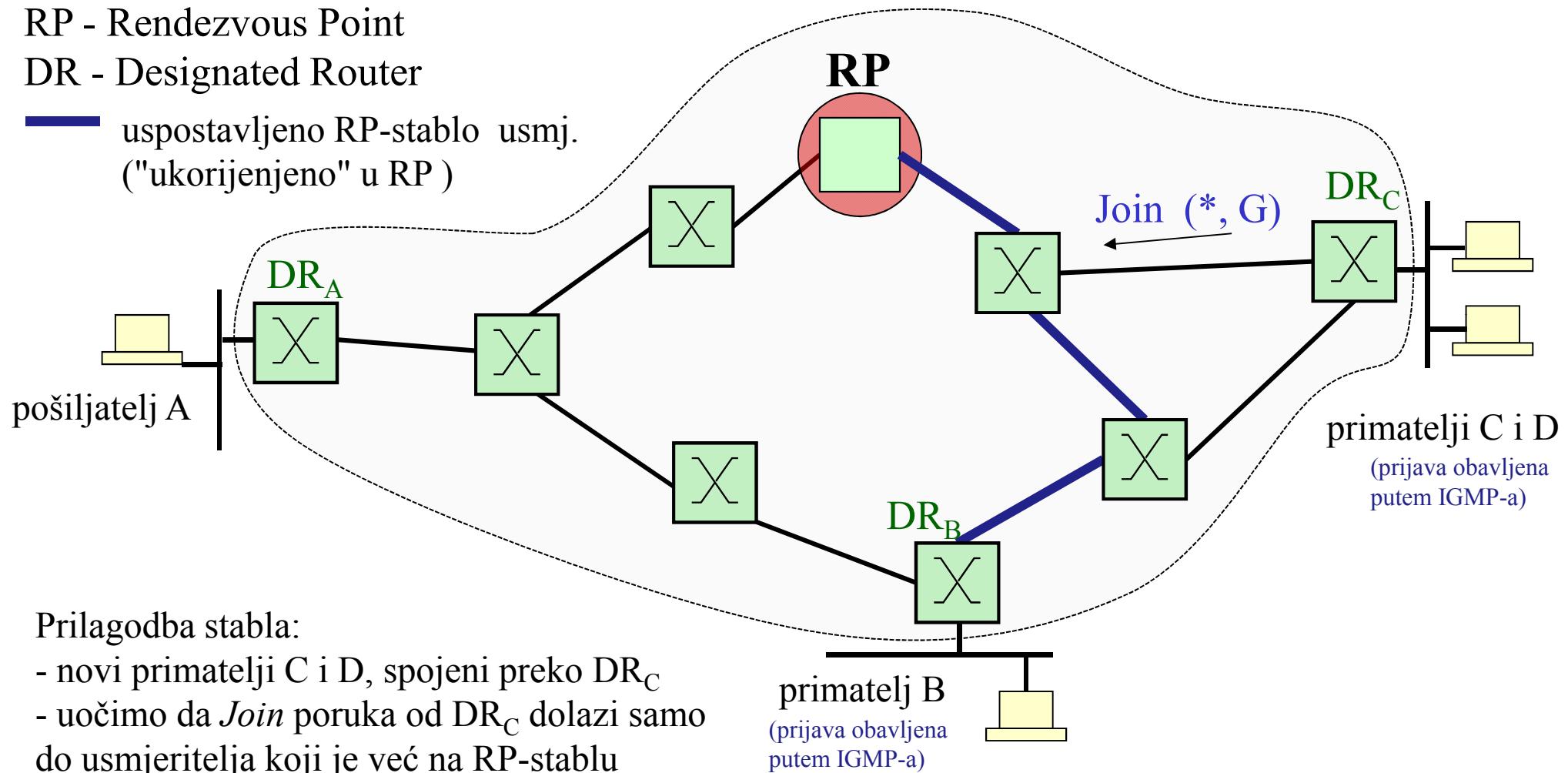


# Faza 1: uspostava RP-stabla (2/4)

RP - Rendezvous Point

DR - Designated Router

— uspostavljeno RP-stablo usmj.  
 ("ukorijenjeno" u RP )

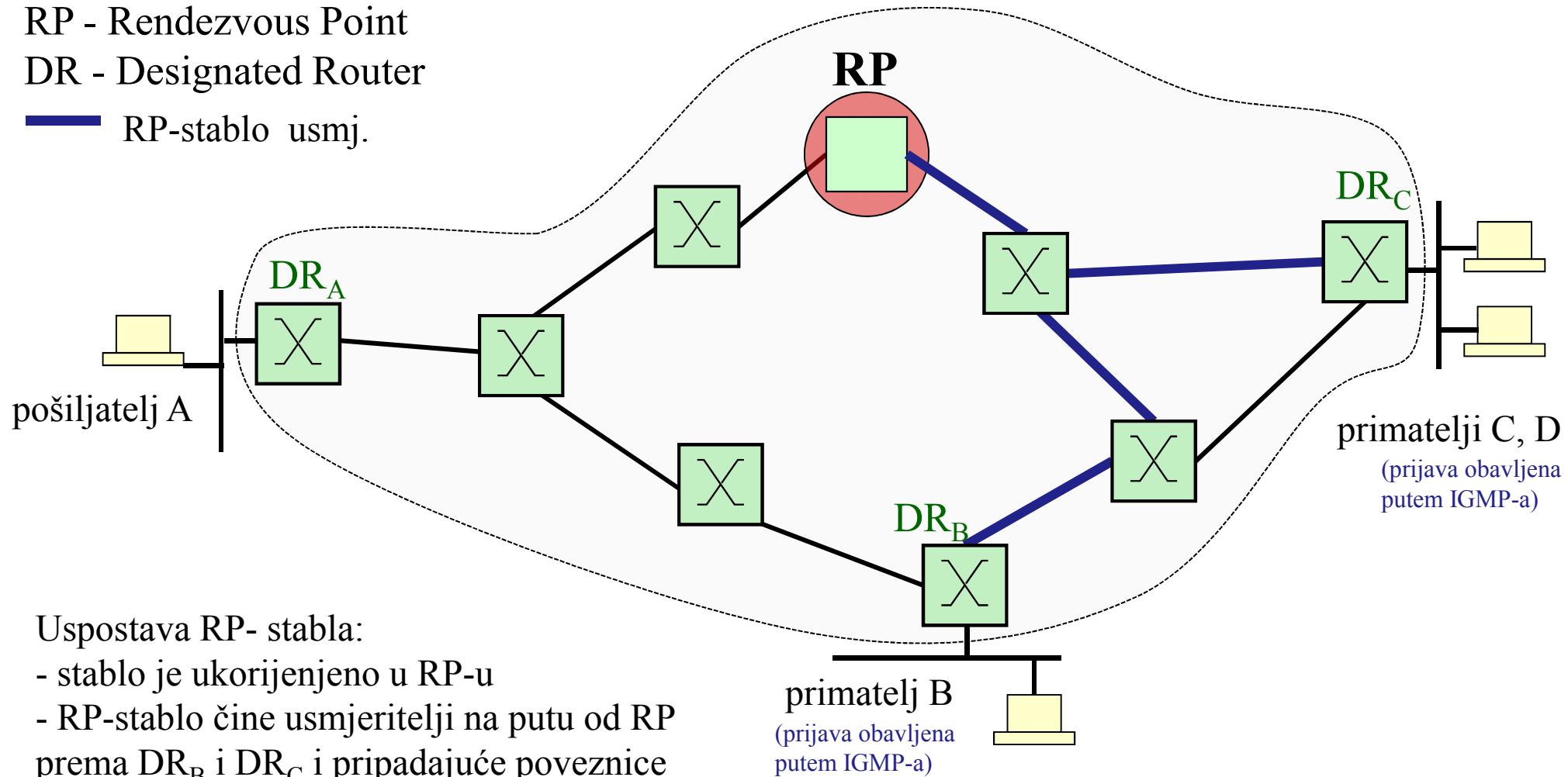


# Faza 1: uspostava RP-stabla (3/4)

RP - Rendezvous Point

DR - Designated Router

 RP-stablo usmj.

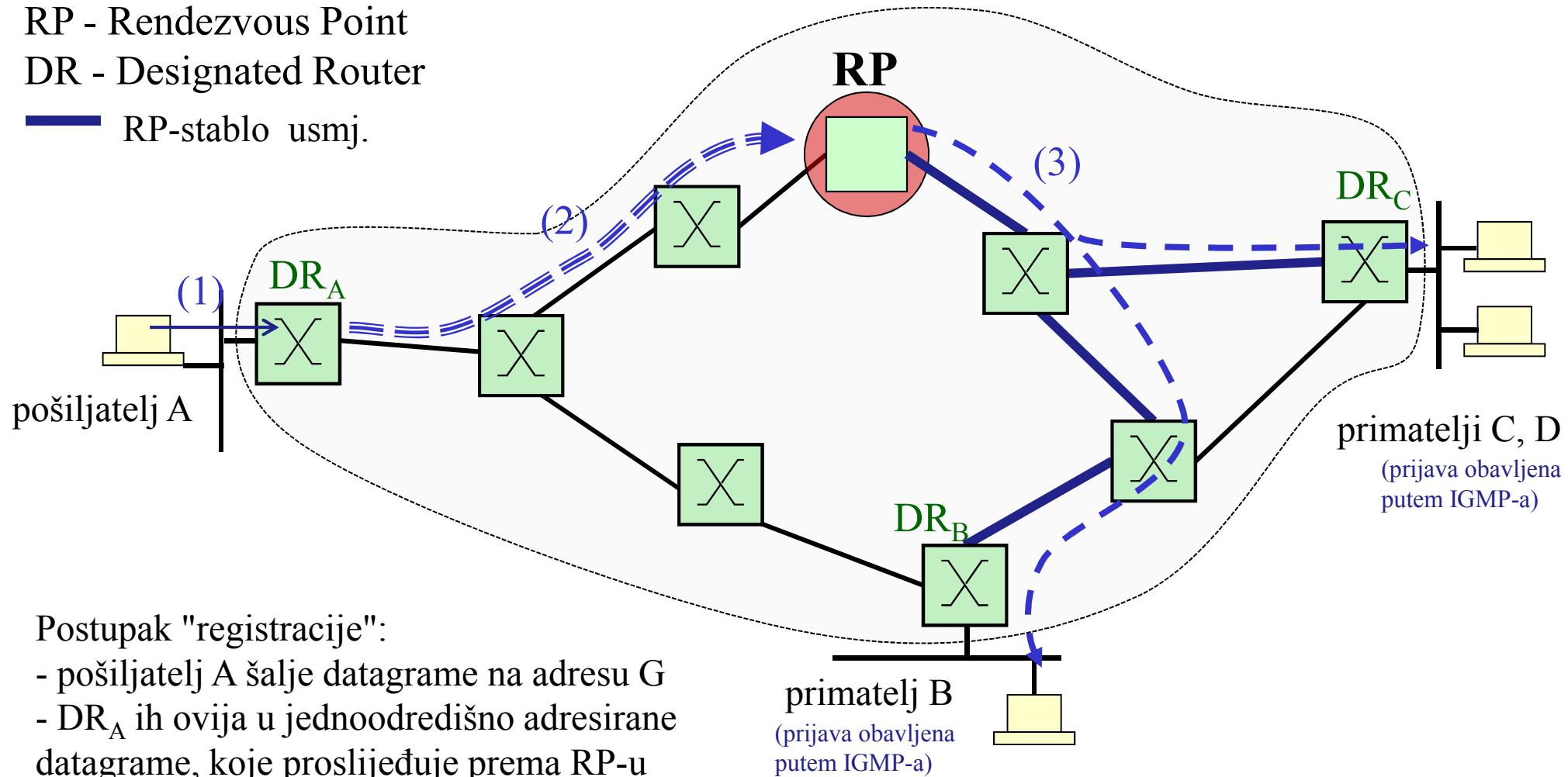


# Faza 1: uspostava RP-stabla (4/4)

RP - Rendezvous Point

DR - Designated Router

— RP-stablo usmj.

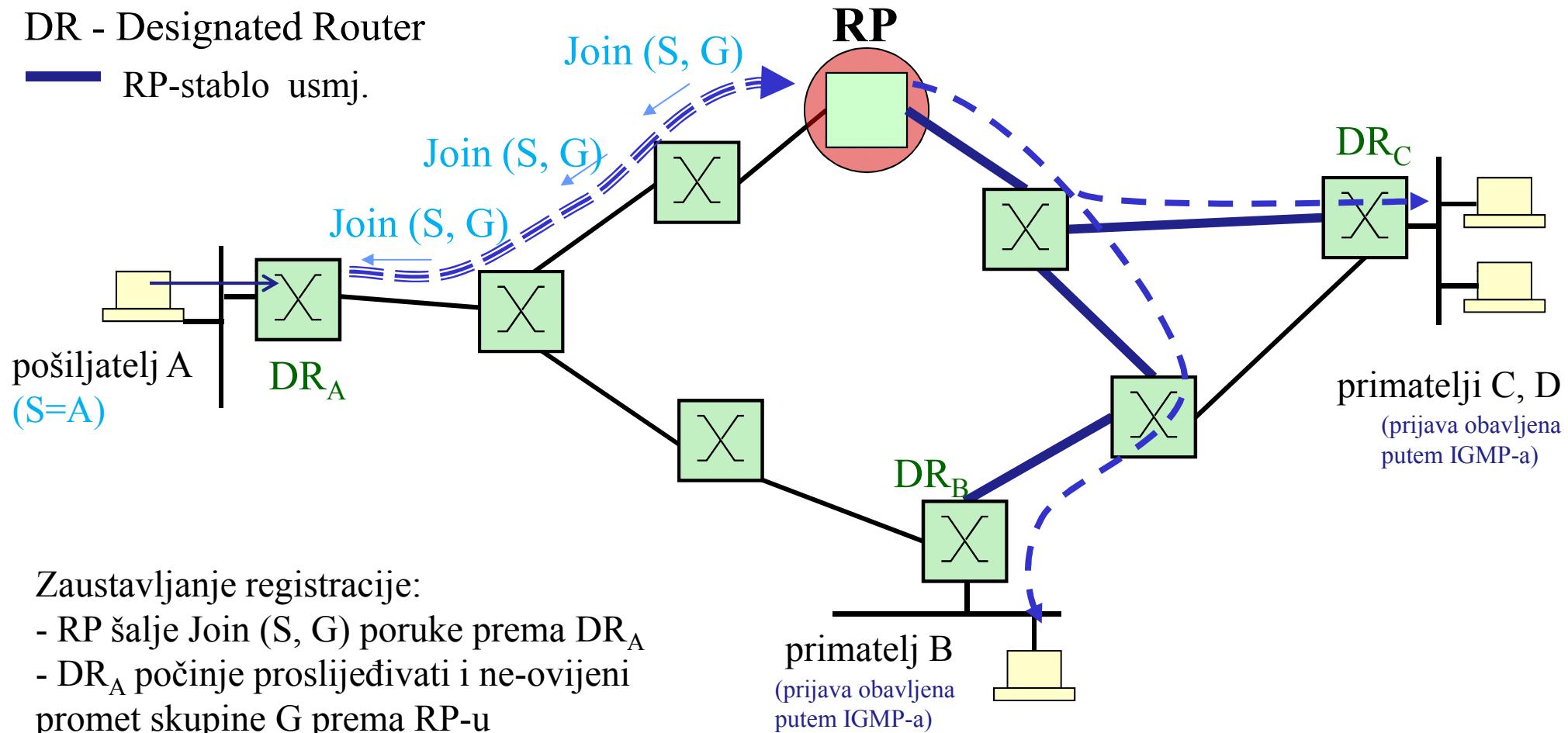


# Faza 2: zaustavljanje registracije (1/3)

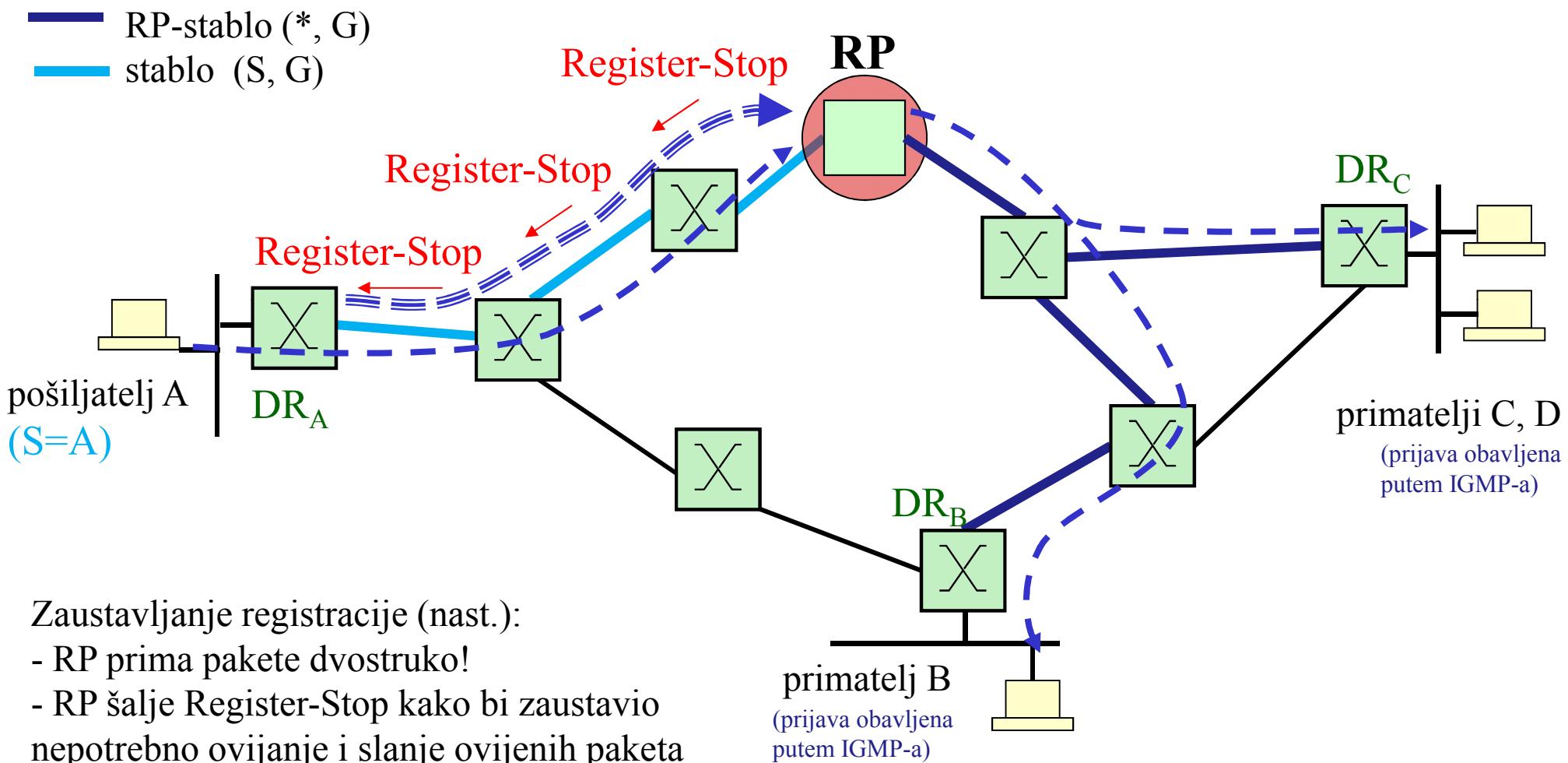
RP - Rendezvous Point

DR - Designated Router

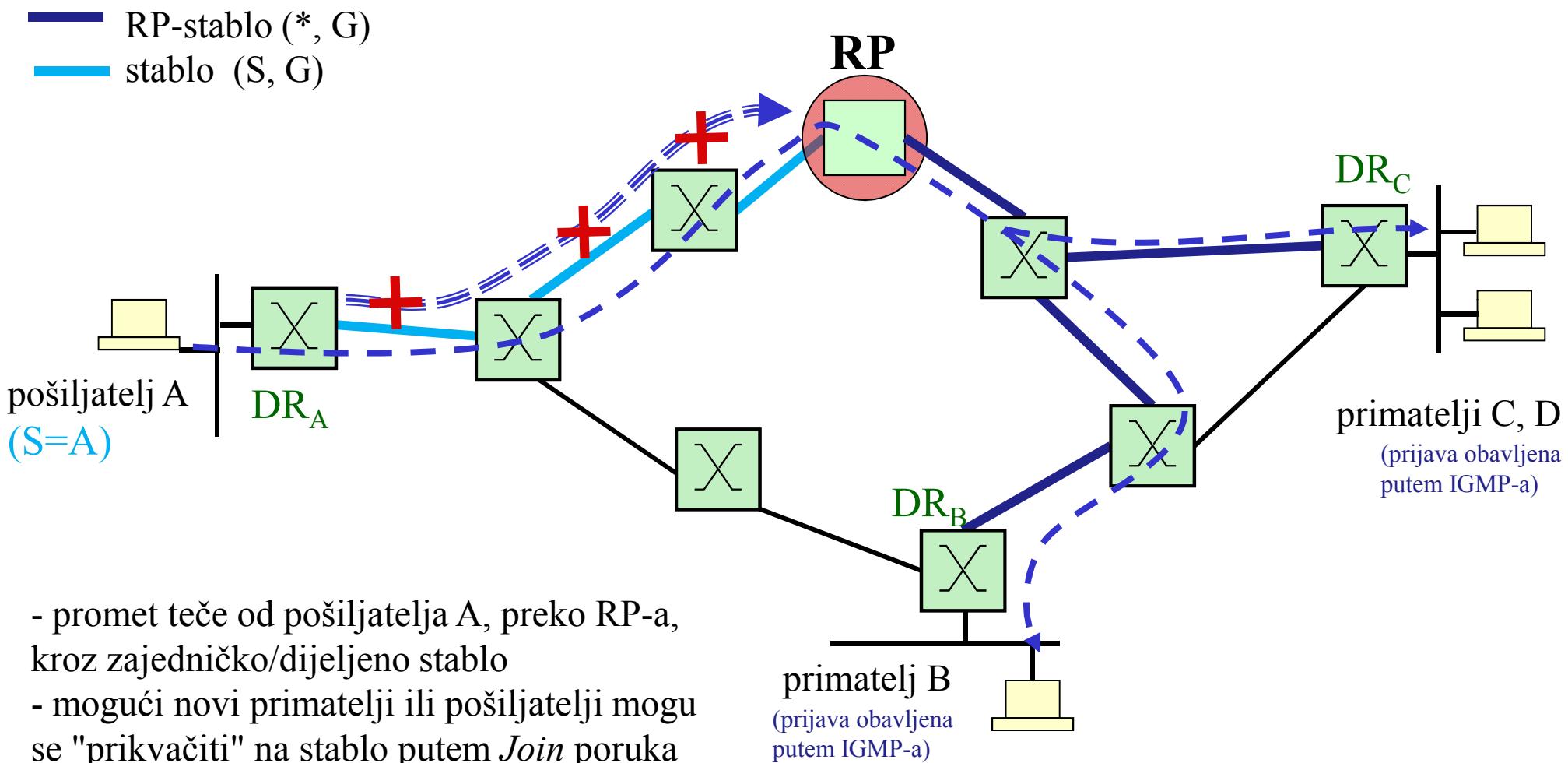
 RP-stablo usmj.



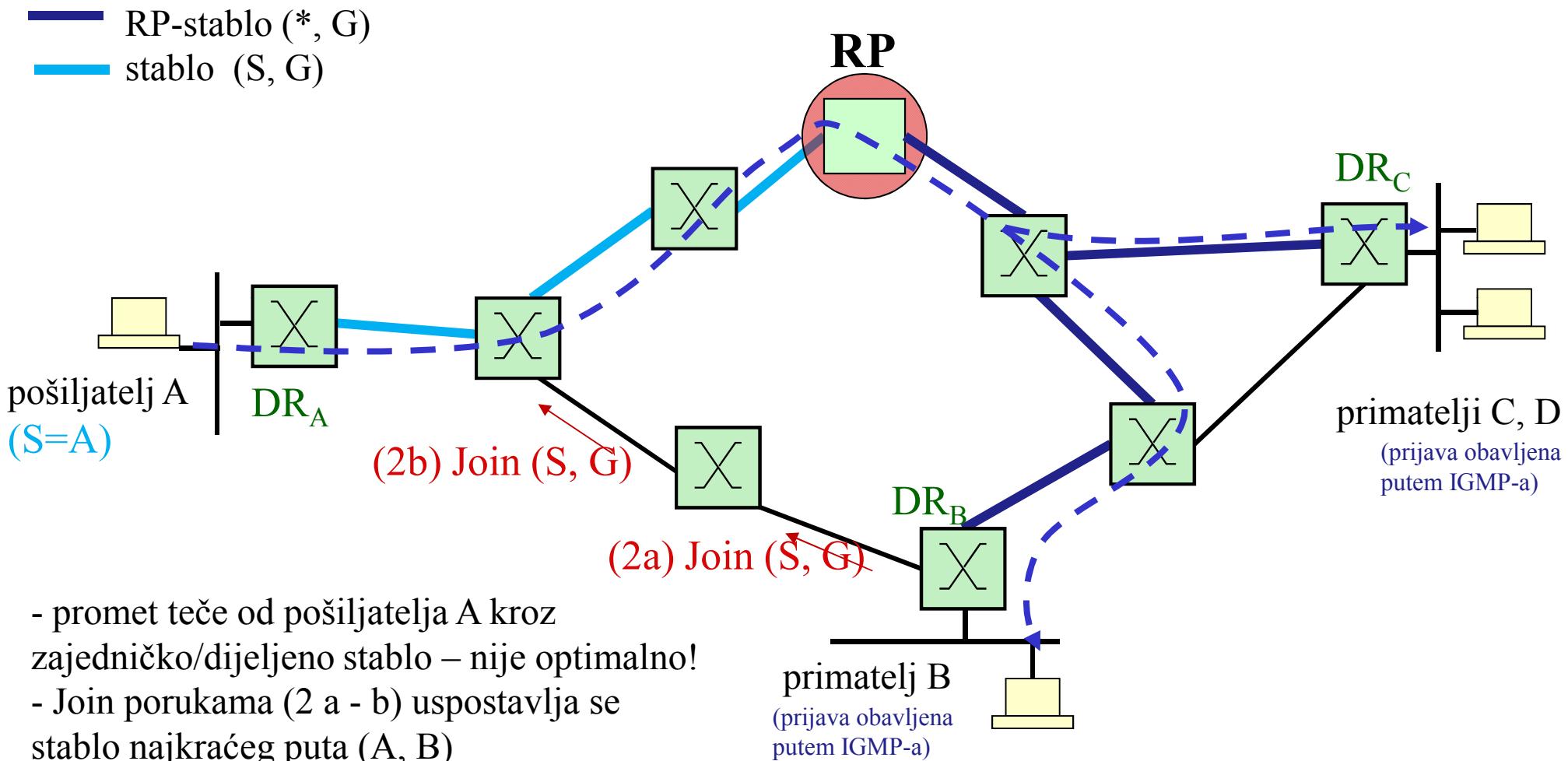
# Faza 2: zaustavljanje registracije (2/3)



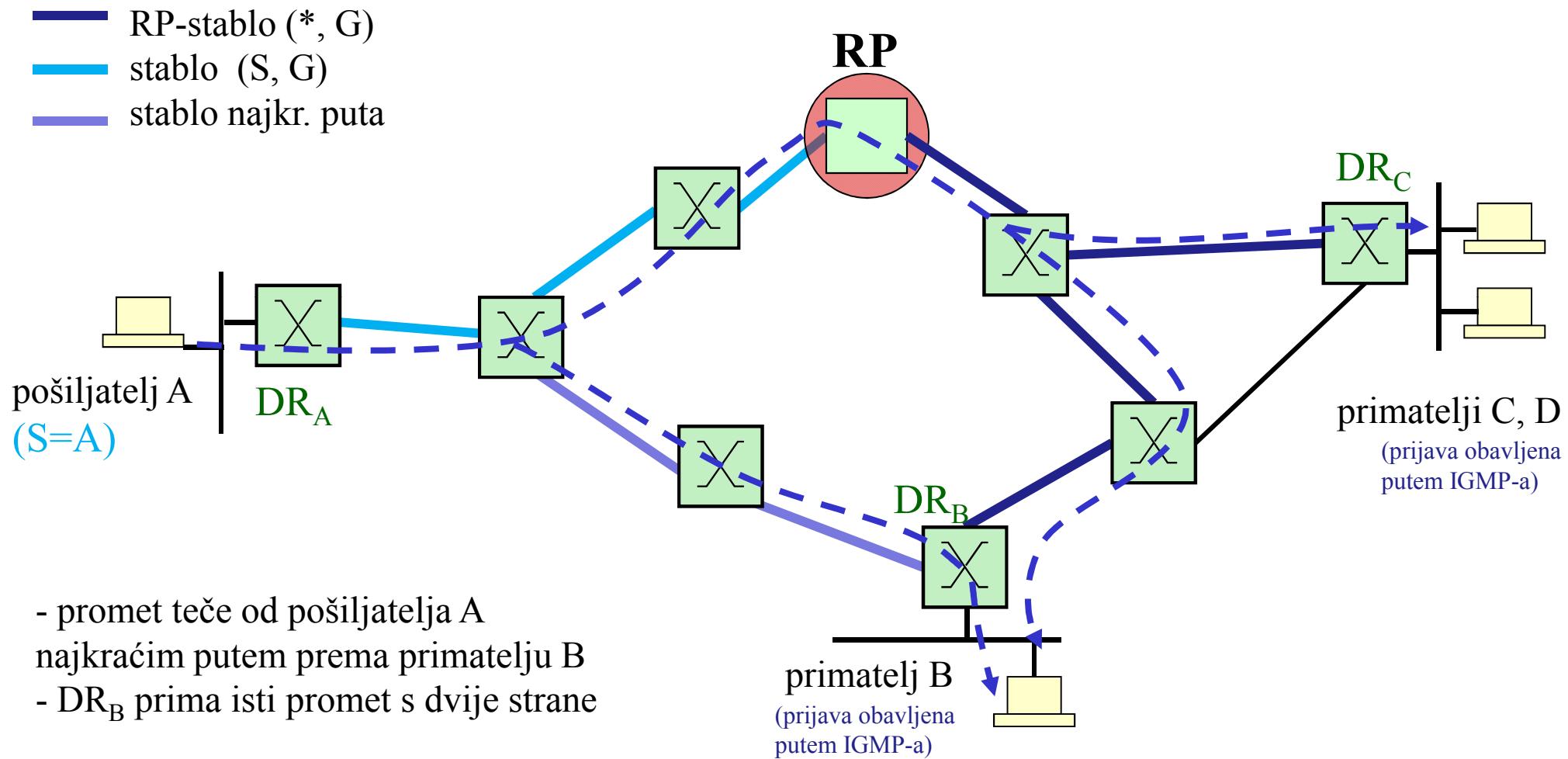
# Faza 2: zaustavljanje registracije (3/3)



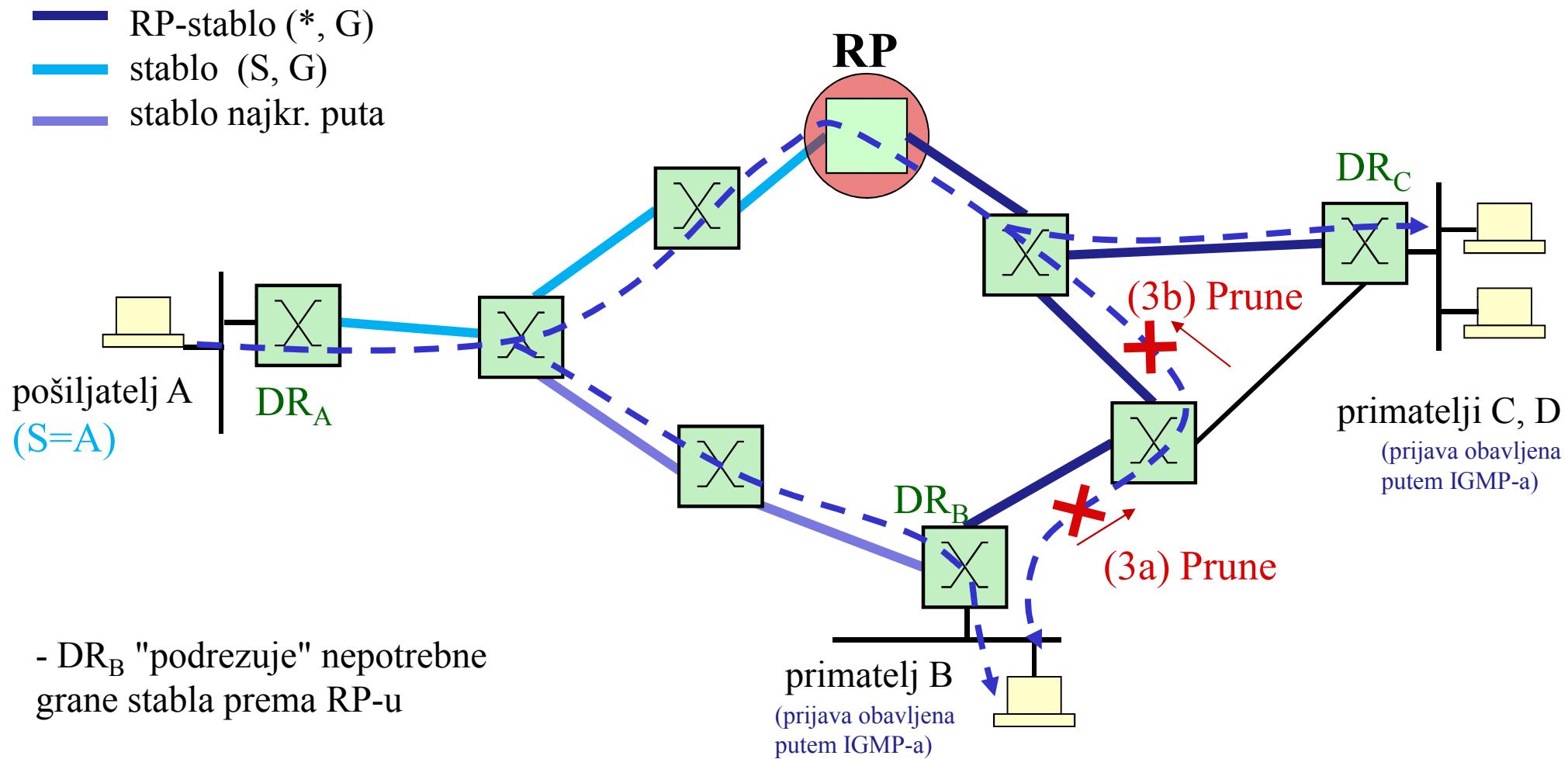
# Faza 3: prelazak na stablo najkraćeg puta (1/3)



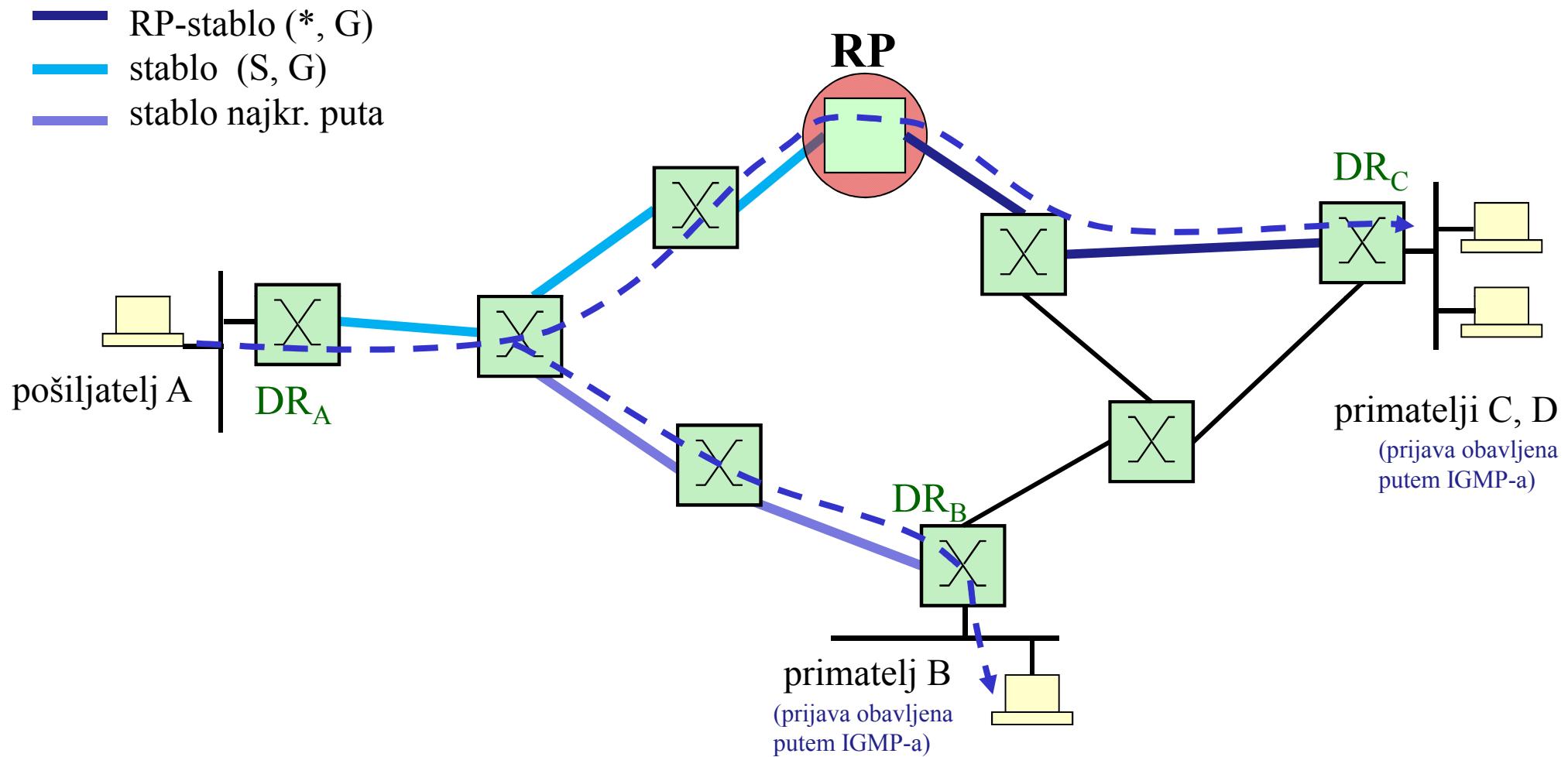
# Faza 3: prelazak na stablo najkraćeg puta (2/3)



# Faza 3: prelazak na stablo najkraćeg puta (3/3)



# Faza 3: prelazak na stablo najkraćeg puta (3/3)



# Sažetak svojstava protokola PIM-SM

## ■ prednosti:

- prilagodljivost veličini (bolja skalabilnost od, primjerice, DVMRP i sličnih protokola koji se temelje na preplavljuvanju)
- mogućnost raspoređivanja opterećenja, odn. prelaska na stablo najkraćeg puta

## ■ nedostaci:

- mora postojati mehanizam za preslikavanje višeodredišne adrese na adresu RP-a, ili se ona mora unaprijed odrediti (npr. staticki konfigurirati)
  - ovisnost o RP-u općenito: izbor smještaja središnje točke nije trivijalan problem (početno RP-stablo ne mora biti optimalno, ali to se rješava u fazama 2 i 3)
-



Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

8.

Prostorno i vremensko  
usklađivanje višemedijskog  
sadržaja

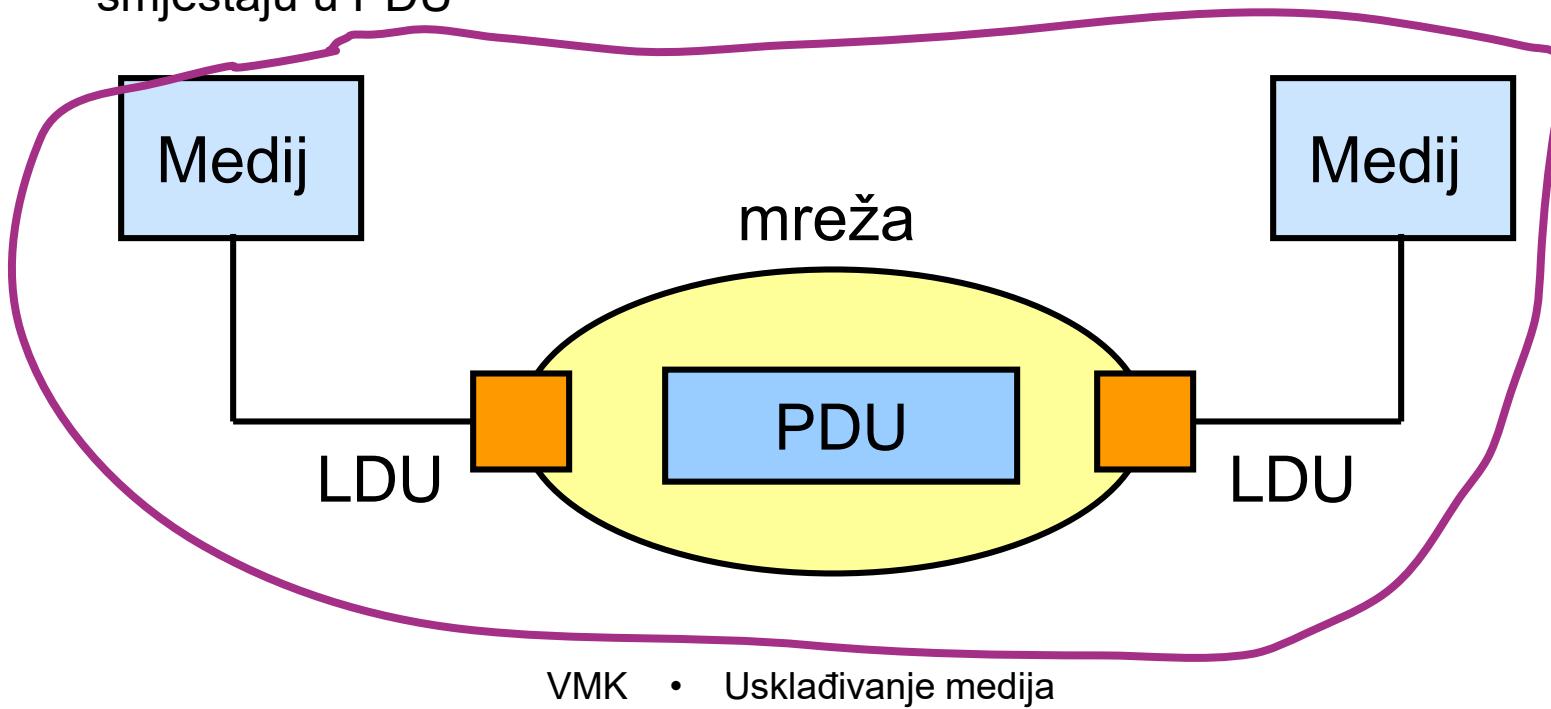
# Svojstva višemedijskog sadržaja

---

- broj medija (medijskih objekata)
- vrste medija
  - vremenski neovisni medij (*diskretni*)
    - valjanost podataka ne ovisi o vremenskim uvjetima tj. o trenutku pojavljivanja, trajanju i sl.
    - npr: tekst, nepomična slika
  - vremenski ovisni medij (*kontinuirani*)
    - vrijednost podataka se mijenja tijekom vremena, a valjanost ovisi o vremenskim uvjetima - obrada i komunikacija složena
    - npr: govor, glazba, video, animacija
- integracija

# Prijenos višemedija

- **LDU:** logička podatkovna jedinica - opisuje medij
  - može se promatrati na raznim razinama hijerarhije, npr. pixel, blok, okvir, kadar, film
- **PDU:** protokolna podatkovna jedinica – jedinica za transport, format prema komunikacijskom protokolu
  - tijekom komunikacije, LDU se cijeće, po dijelovima ili više njih zajedno smještaju u PDU

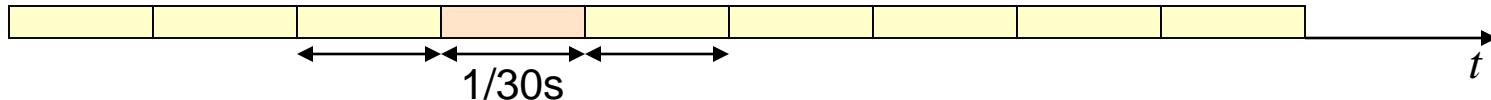


# Problemi usklađivanja medija

- **sadržajni odnos**
  - npr. prikaz podataka tablicom i grafom u Excelu
- **prostorni odnos**
  - npr. prevedeni tekst, u obliku "titlova" ispod slike; raspored slika i teksta na Web stranici
- **vremenski odnos (sinkronizacija)**
  - **sinkronizacija uživo**
    - npr. televizijski prijenos
  - **umjetna (sintetička) sinkronizacija**
    - npr. 3D animirani lik i govor
    - prezentacija uz slideove
- sinkronizacija se veže uz pojam logičke podatkovne jedinice (Logical Data Unit, LDU)
  - **otvorena LDU** (nepredvidivo trajanje)
  - **zatvorena LDU** (predvidivo trajanje)

# Primjeri i vrste LDU

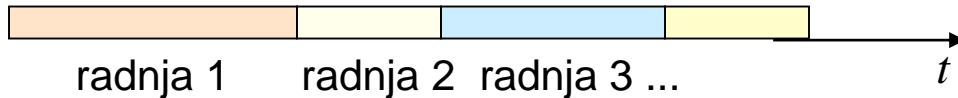
a) **digitalni video** - zatvorena LDU (slika; 1/30 s)



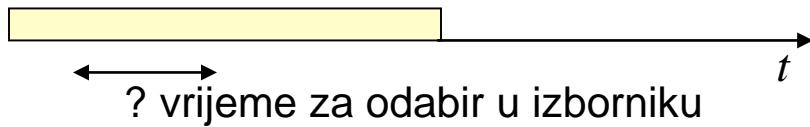
b) **digitalni audio** - zatvorena LDU (npr. 512 byte odgovara 0.064 s audia)

c) **animacija** - zatvorena LDU (15 – 30 slika u sekundi)

d) prikaz korisnikovih **radnji u grafičkom** sučelju – otvorena LDU (npr. kopiranje direktorija, trajanje ovisno o vrsti radnje)



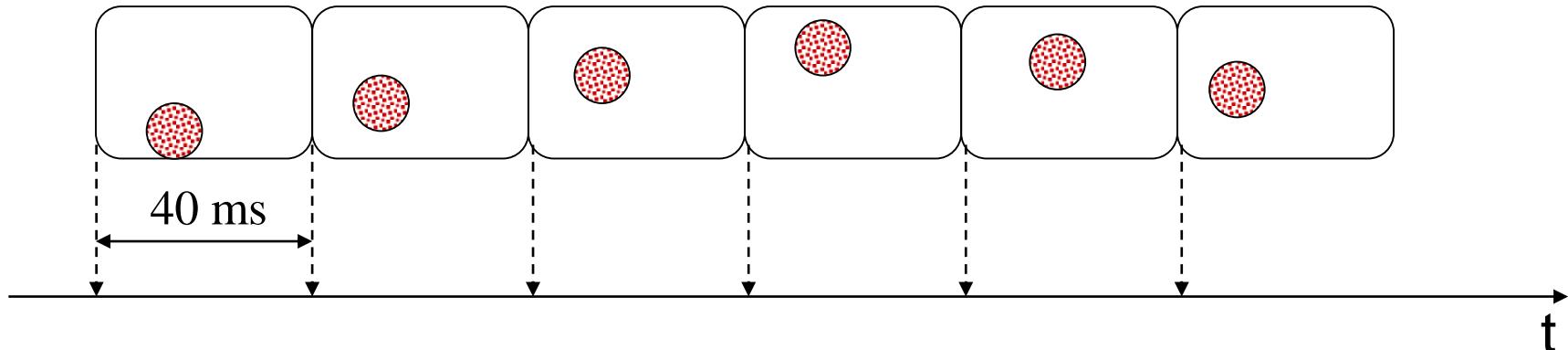
e) **korisnička interakcija** - otvorena LDU (nepoznato, eventualno ograničeno vremenskom kontrolom)



f) **mjerač vremena** - zatvorena i prazna LDU

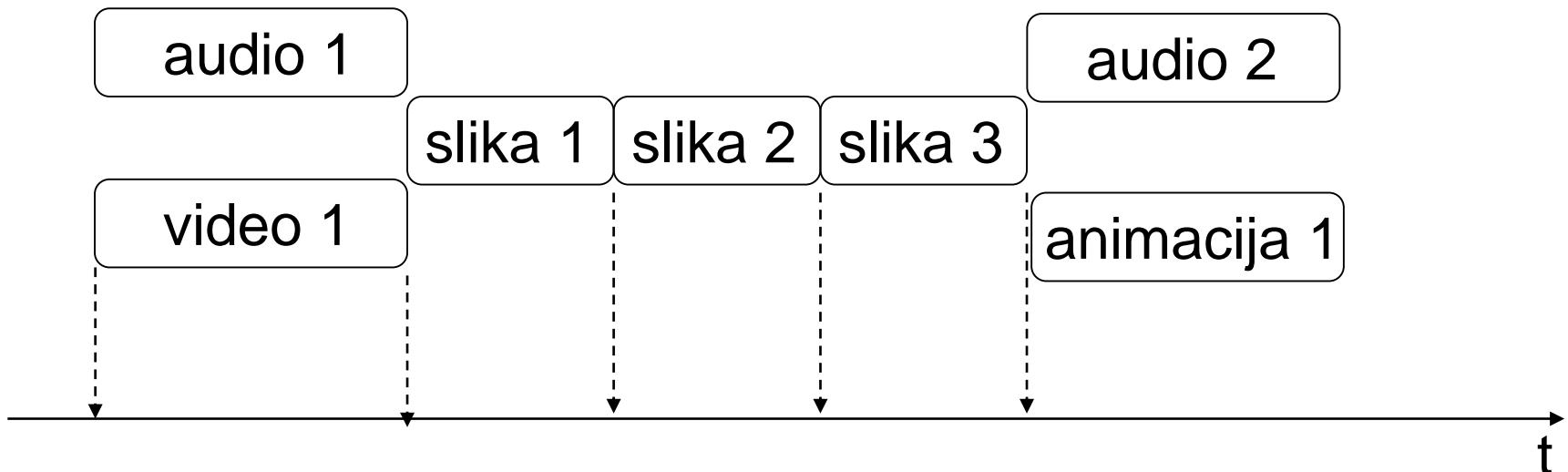
# Sinkronizacija unutar medijskog objekta

- sinkronizacija jedinica **unutar** samog medijskog objekta (engl. *intra-stream synchronization*)
- vremensko usklađivanje unutar struje vremenski ovisnog (kontinuiranog) medija
- primjer: okviri unutar videa



# Sinkronizacija između medijskih objekata

- sinkronizacija jedinica **između** različitih medijskih objekata (engl. *inter-stream synchronization*)
- primjer: vremensko usklađivanje unutar višemedijske prezentacije

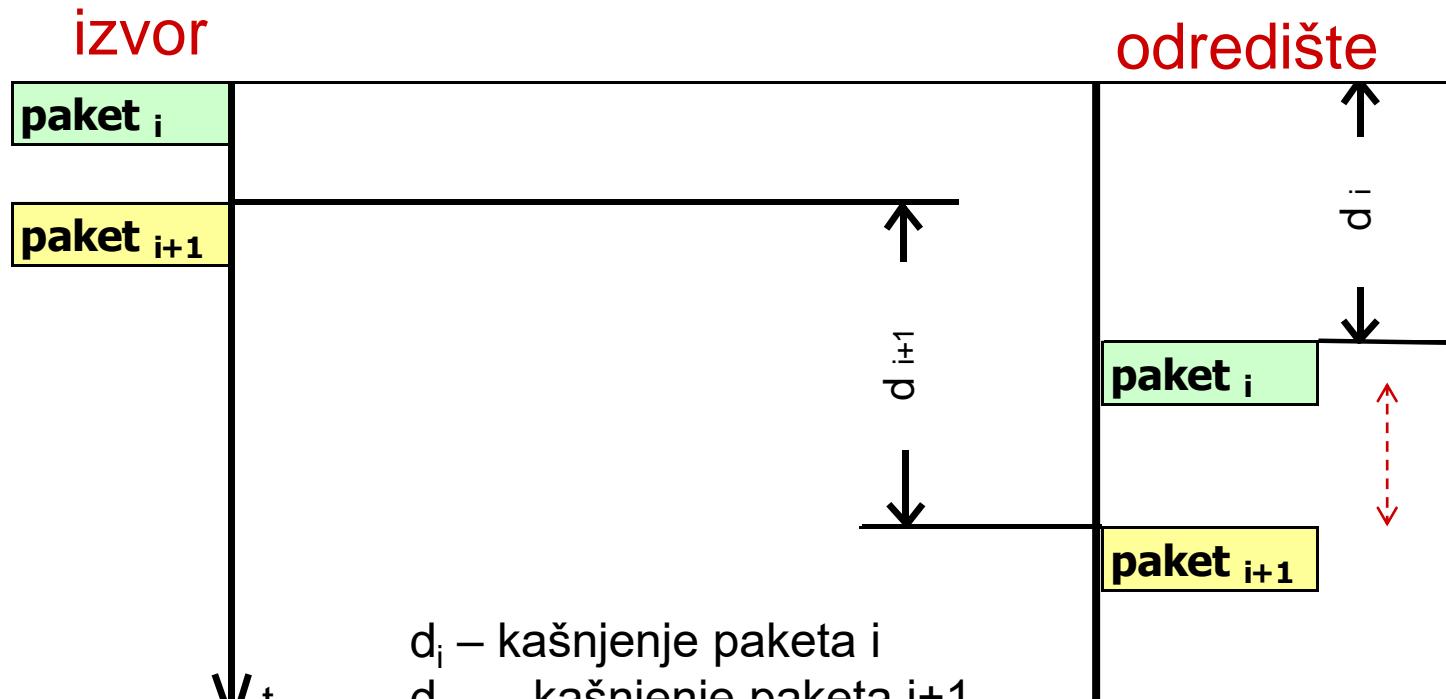


# Načini mrežnog prijenosa

- **Asinkroni prijenos**
  - Omogućava komunikaciju bez vremenskih ograničenja
  - Paketi pristižu na odredište “najbrže moguće”
- **Sinkroni prijenos**
  - Definira maksimalno kašnjenje s kraja na kraj za svaki paket unutar struje podataka
  - Može zahtijevati međuspremnik podataka
- **Izokroni prijenos**
  - definira maksimalno i minimalno kašnjenje s kraja na kraj za svaki paket unutar struje podataka.
  - kolebanje kašnjenja pojedinih paketa je ograničeno!
  - smanjena potreba za međuspremnikom podataka

# Problemi u umreženoj okolini (1)

- cilj: smanjiti učinak kolebanja kašnjenja (engl. *jitter*)  
 kolebanje kašnjenja može nastati tijekom prijenosa, kao i tijekom obrade



$$j = \max ( |d_i - d_{i+1}| ), i = 1, 2, \dots$$

# Problemi u umreženoj okolini (2)

- cilj: smanjiti učinak gubitka okvira
- gubitak okvira može nastati zbog prevelikog kašnjenja okvira, gubitka okvira (podataka) u mreži ili zbog greške nastale u prijenosu okvira



Proširenje ostalih okvira umjesto nedostajućeg okvira



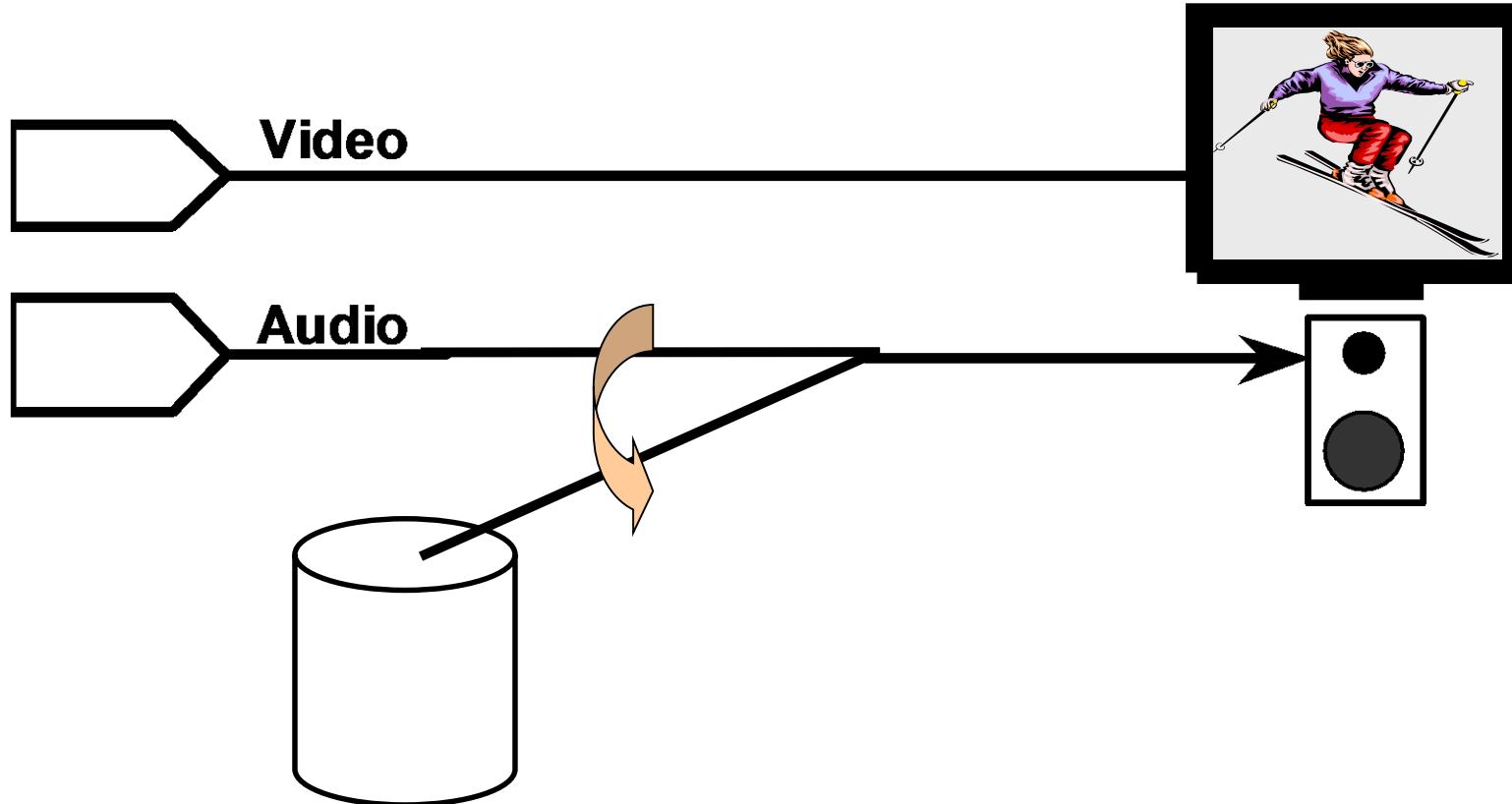
Udvostručenje prethodnog okvira umjesto nedostajućeg okvira



Preskakanje nedostajućeg okvira (bez intervencije)

# Problem premoštavanja jaza u prikazu

- zbog blokiranja struje podataka dolazi do jaza u prikazu
- problem se rješava alternativnim prikazom



**Alternativni prikaz**

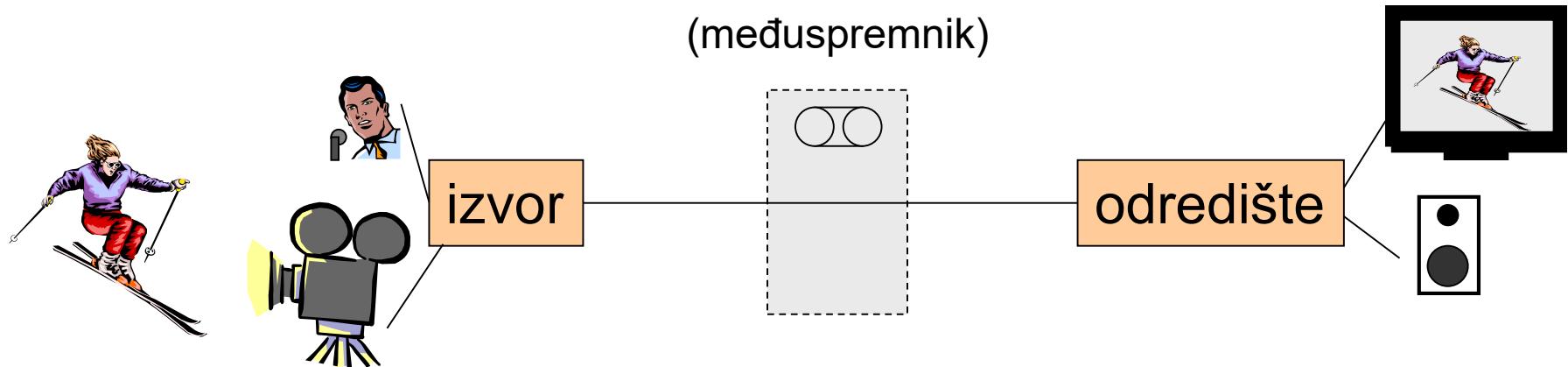
VMK • Usklađivanje medija

# Sinkronizacija u umreženoj okolini

- način transporta sinkronizacijske informacije
  1. slanje specifikacije sinkronizacije prije sadržaja
  2. odvojeni signalizacijski kanal
  3. multipleksirane struje - isprepletenost medija i sinkronizacijske informacije
- mjesto sinkronizacije
  - na izvoru
  - na odredištu
  - problem: više izvora
- višestruki odnosi izvor-odredište (1:1, 1:n, m:1, n:m)

# Sinkronizacija uživo

- cilj: rekonstrukcija vremenskih odnosa uspostavljenih na izvoru (prilikom snimanja medija)
- primjer primjene: konverzacijalne višemedijske usluge
- ne treba specifikacija sinkronizacije, budući da su vremenski odnosi implicitno zadani prilikom snimanja



- sinkronizacija uživo može se raditi:
  - bez međuspremnika (izravno)
  - s međuspremnikom (mogućnost kasnijeg prijenosa i obrade)

# Umjetna sinkronizacija

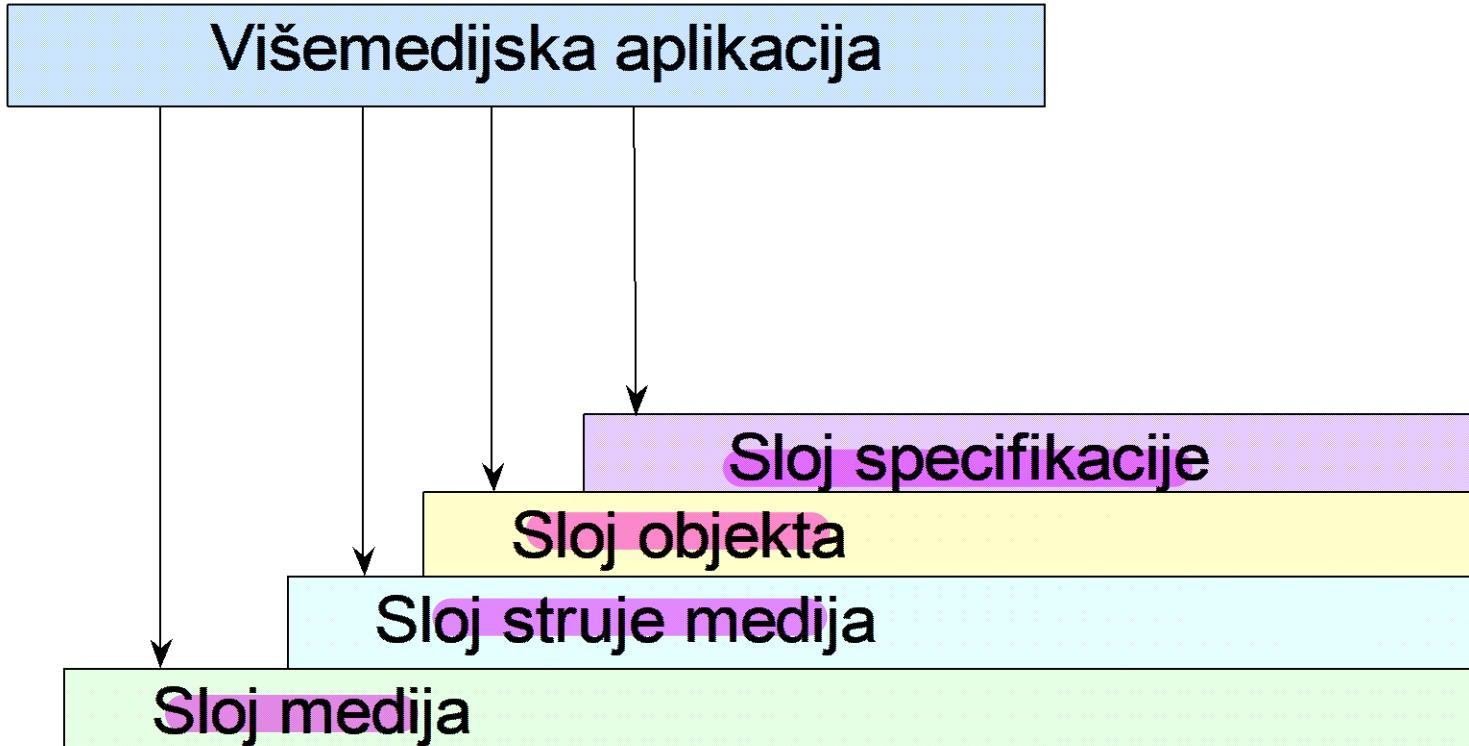
- cilj: vremenski odnosi između objekata ne postoje "sami po sebi", već se uvode eksplicitno, putem specifikacije
- primjer primjene: stvaranje novih višemedijskih objekata slaganjem različitih medijskih objekata (komponenata)
- razlikujemo dvije faze umjetne sinkronizacije:
  - **faza specifikacije** – definiranje vremenskih odnosa u modelu sinkronizacije
  - **faza prikaza** – prikaz medija korištenjem specificiranog modela sinkronizacije

# Specifikacija sinkronizacije

- pojам medijskog objekta
  - audio struja, video struja, animacija, ... (vremenski ovisni mediji )
  - nepomična slika, tekst, ... (vremenski neovisni mediji)
- specifikacija mora sadržavati:
  - specifikaciju **sinkronizacije unutar** medijskog objekta (intra-sinkronizacija)
    - npr. okviri videa
  - opis **kvalitete usluge** za sinkronizaciju **unutar** medijskog objekta
    - npr. 30 fps
  - specifikaciju **sinkronizacije između** dvaju ili više medijskih objekata (inter-sinkronizacija)
    - npr. animacija i zvuk
  - opis **kvalitete usluge** za sinkronizaciju **između** medijskih objekata
    - npr. razilaženje +/-80 ms

# Referentni model sinkronizacije

- referentni model sinkronizacije medija ima četiri sloja
- svaki sloj prikazuje višemedijsku aplikaciju na drugoj razini apstrakcije
- najviši sloj je sloj specifikacije, a najniži sloj medija



# Sloj medija

---

- sloj medija rukuje s jednom kontinuiranom strujom medija koju promatra kao niz LDU-ova
  - LDU - logička podatkovna jedinica
    - može biti bilo što, ovisno o razini
    - npr. u video: sekvenca, kadar, makroblok, blok, pixel
- sloj medija osigurava pristup LDUovima neovisan o uređaju
- apstrakcija na razini “čitaj LDU s uređaja”, “piši LDU na uređaj”
- daje garanciju za obradu jedne LDU
- Zadaci sloja medija: pristup datotekama i uređajima

# Sloj medija: primjer

```
window = open("VideoDevice")          \\ otvori video prozor
movie = open("File");                \\ otvori datoteku
while (not eof(movie)) {
    read(movie, &ldu);
    if (ldu.time == 20)              \\ čitaj LDU
        print("Subtitle1");
    else if (ldu.time == 26)         \\ sinkroniziraj titlove
        print("Subtitle2");
    write(window, ldu); }            \\ prikaži LDU
close(window);                      \\ zatvori prozor
close(movie);                       \\ zatvori datoteku
```

# Sloj struje medija

- sloj struje medija rukuje s kontinuiranim **strujama medija** i **skupinama struja medija**
- sve struje unutar skupine prikazuju se paralelno uz primjenu sinkronizacije između njih
- daje garanciju za sinkronizaciju između struja podataka u promatranoj skupini
- apstrakcija na razini “start”, “stop”, “stvori skupinu”, “pokreni skupinu”, “zaustavi skupinu”
- Zadaci sloja struje medija: rezervacija resursa i vremensko raspoređivanje obrade LDU-ova

# Sloj struje medija: primjer

```
open digitalvideo alias ex          \\ stvori video deskriptor
load ex video.avs                  \\ pridruži mu datoteku
setcuepoint ex at 20 return 1      ←
setcuepoint ex at 26 return 2      ←
setcuepoint ex on
play ex                            \\ počni reprodukciju
switch readevent() {
    case 1: display("Subtitle1")   \\ događaj 1 pokaži subtitle 1
    case 2: display("Subtitle2")   \\ događaj 2 pokaži subtitle 2
}
```

postavljanje događaja  
koji se šalje aplikaciji

# Sloj objekta

- sloj objekta rukuje s **medijskim objektima**
- objekti su apstrakcije koje potpuno skrivaju vrstu (diskretni/ kontinuirani) stvarnog medija kojeg uključuju
- sloj objekta **na temelju specifikacije sinkronizacije** stvara cjelokupni prikaz višemedijskog sadržaja
- **kvaliteta usluge (QoS)** promatrana za svaki medij pojedinačno
  - prihvatljivo razilaženje između struja medija
  - kolebanje kašnjenja
- Zadaci: **planiranje i usklađivanje raspoređivanja prikaza, pokretanje prikaza vremenski neovisnih medijskih objekata, pokretanje prikaza radnji za pripremu prikaza**

# Sloj objekta: primjer

(MHEG ISO/IEC JTC1/SC29/WG12 Multimedia and Hypermedia information coding Expert Group)

```
Composite {
    start-up link
    viewer start-up
    viewer list
    Viewer1:ref. To Component1
    Viewer2:ref. To Component2
    Viewer3:ref. To Component3
    Component1
        ref. to content "movie.avs"
    Component2
        ref. to content "Subtitle1"
    Component3
        ref. to content "Subtitle2"
    Link1 "when timestone=20
          then start Viewer2"
    Link2 "when timestone=26
          then start Viewer3"
}
```

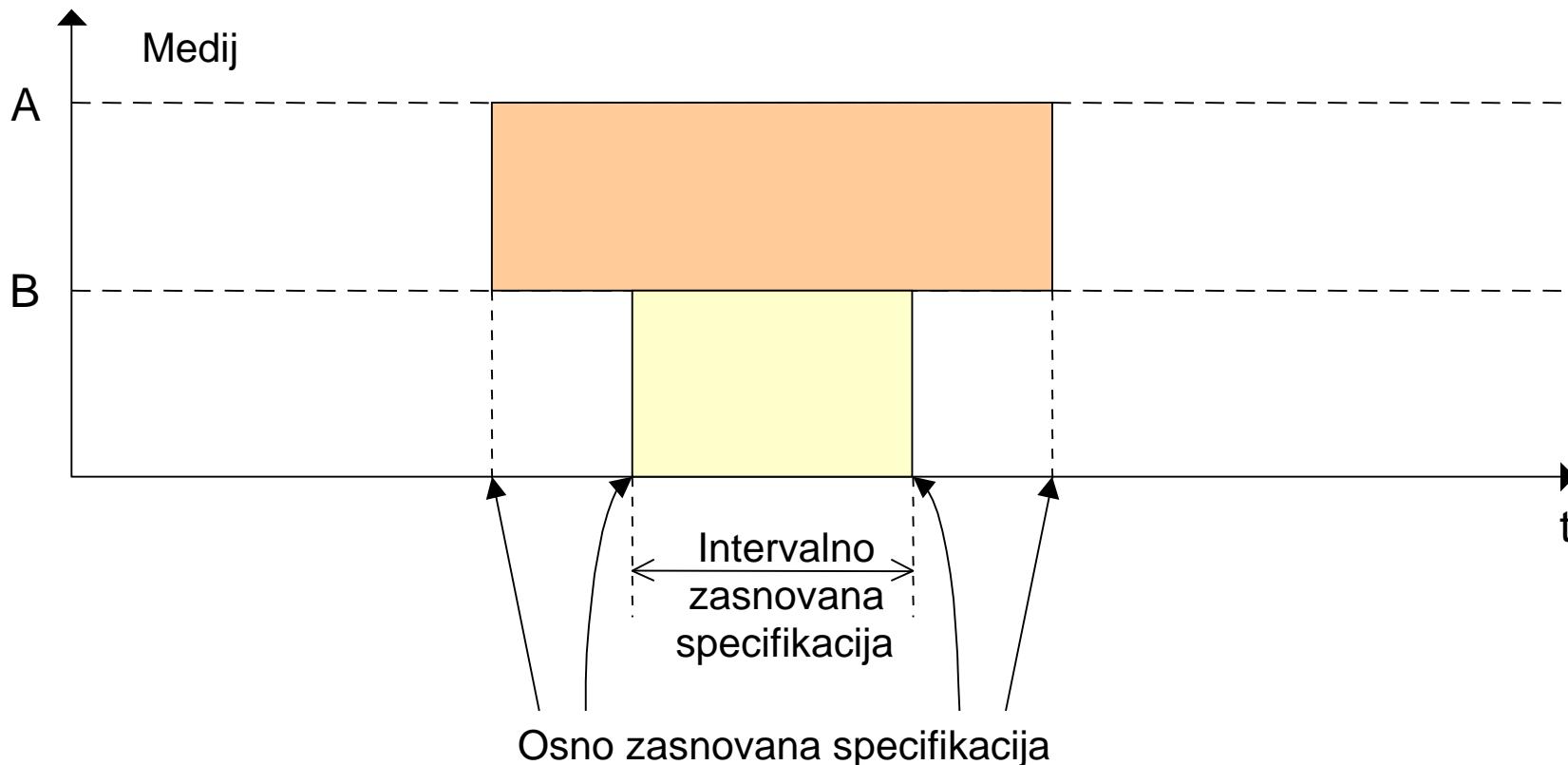
\ kompozitni objekt  
\ kako započeti prikaz  
\ virtualni pogled na komponentne objekte  
\ kompozitnog objekta  
\ komponentni objekt 1  
\ komponentni objekt 2  
\ komponentni objekt 3  
\ vremenski odnosi

# Sloj specifikacije

- otvoreno sučelje, sadrži alate za stvaranje specifikacija po odabranom standardu
- zadužen za preslikavanje QoS zahtjeva s korisničke razine na sloj objekta
- klasifikacija sinkronizacijskih specifikacija (4 vrste):
  - intervalna
    - zasniva se na specifikaciji vremenskih odnosa između *vremenskih intervala* prikaza medijskih objekata
  - osna
    - događaji u prikazu ravnaju se prema *osima* koje su zajedničke objektima u prezentaciji
  - temeljena na *kontroli toka prikaza*
    - višestruke niti prikaza se *sinkroniziraju* u *unaprijed zadanim sinkronizacijskim* točkama
  - temeljena *na događajima*
    - događaji *unutar prikaza medija* *okidaju druge prikaze* (start, stop, priprema prikaza)

# Ideja intervalne i osne specifikacije

- **intervalna specifikacija:** definira se **trajanje** i **uskladjenost** (međusobni odnos) vremenskih intervala prikaza medija
- **osna specifikacija:** na vremenskoj osi se definiraju **točke pokretanja i zaustavljanja** prikaza medija

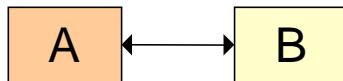


# Intervalna specifikacija

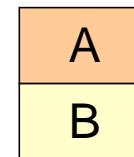
- trajanje prikaza objekta promatra se kao vremenski interval
- dva intervala A i B mogu se sinkronizirati na ukupno 13 različitih načina, pri čemu se kod nekih prikazi A i B mogu obrnuti, tako da postoji 7 jedinstvenih vremenskih odnosa (sljedeća slika)
- prednosti:
  - mogu se usklađivati vremenski ovisni i vremenski neovisni mediji
  - rukuje se s logičkim objektima, dobra apstrakcija sadržaja
  - mogu se usklađivati otvorene LDU (npr. korisnička interakcija)
- nedostaci:
  - složena specifikacija
  - ne obuhvaća specifikaciju razilaženja (engl. *skew*)
  - ne mogu se usklađivati pod-jedinice medijskih objekata

# Vremenski odnosi medija A i B

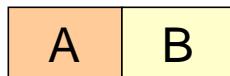
- 1) A prije B



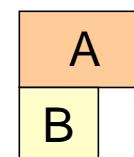
- 5) A i B jednaki



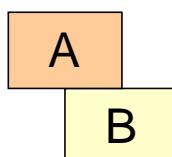
- 2) B neposredno iza A



- 6) B započinje s A

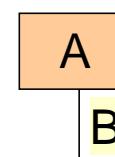
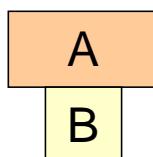


- 3) A se preklapa s B



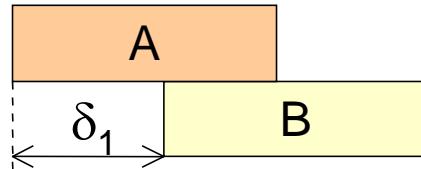
- 7) B završava s A

- 4) B za vrijeme A



# Parametri

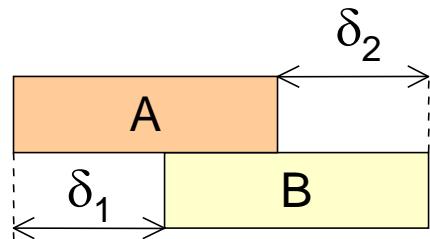
## a) Operacije s jednim parametrom



A prije B( $\delta_1$ )

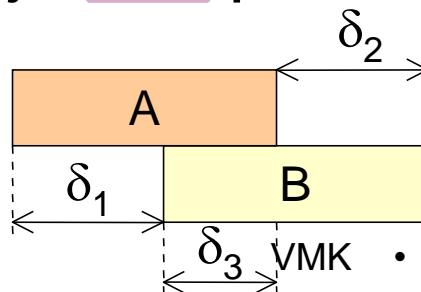
podvrste: A prije završetka B, a počinje prije B, A završava prije B, ...

## b) Operacije s dva parametra



B kasni za A( $\delta_1, \delta_2$ )

## c) Operacije s tri parametra



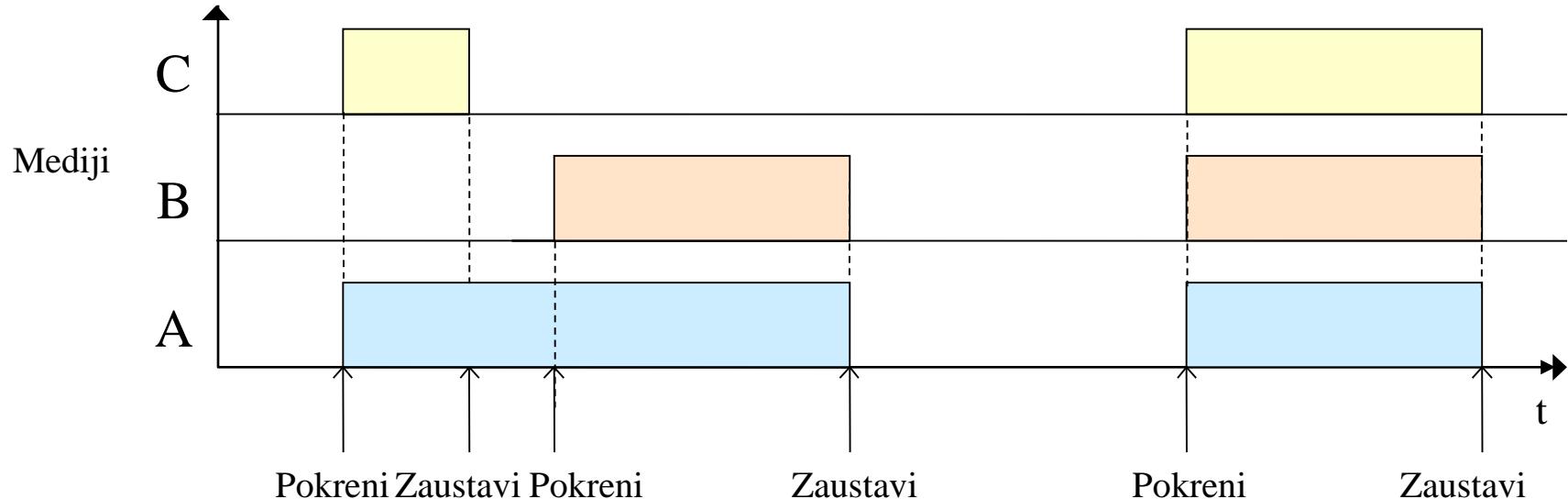
A se preklapa s B ( $\delta_1, \delta_2, \delta_3$ )

# Osna specifikacija

- **osna specifikacija:** na vremenskoj osi se definiraju **točke pokretanja zaustavljanja prikaza medija**
- **prednosti:**
  - jednostavnost
  - pogodan prikaz za sinkronizaciju unutar jednog medija i ugniježđenih medija
  - jasna hijerarhija, jasno upravljanje zbog međusobne neovisnosti medija
- **nedostaci:**
  - ne mogu se opisati otvorene LDU kod kojih trajanje nije poznato ili predvidivo (npr. korisnička interakcija)
  - ne mogu se opisati složeniji odnosi prikaza medijskih objekata koji ne ovise samo o vremenu
  - razilaženje se mora indirektno specificirati pomoću posebne zajedničke osi za promatrane medije

# Vrste osne specifikacije

- globalna vremenska os



- virtualne osi (primjer: glazbene note)
  - koordinatne osi s bilo kakvim mjernim jedinicama, npr. trajanje/visina tona

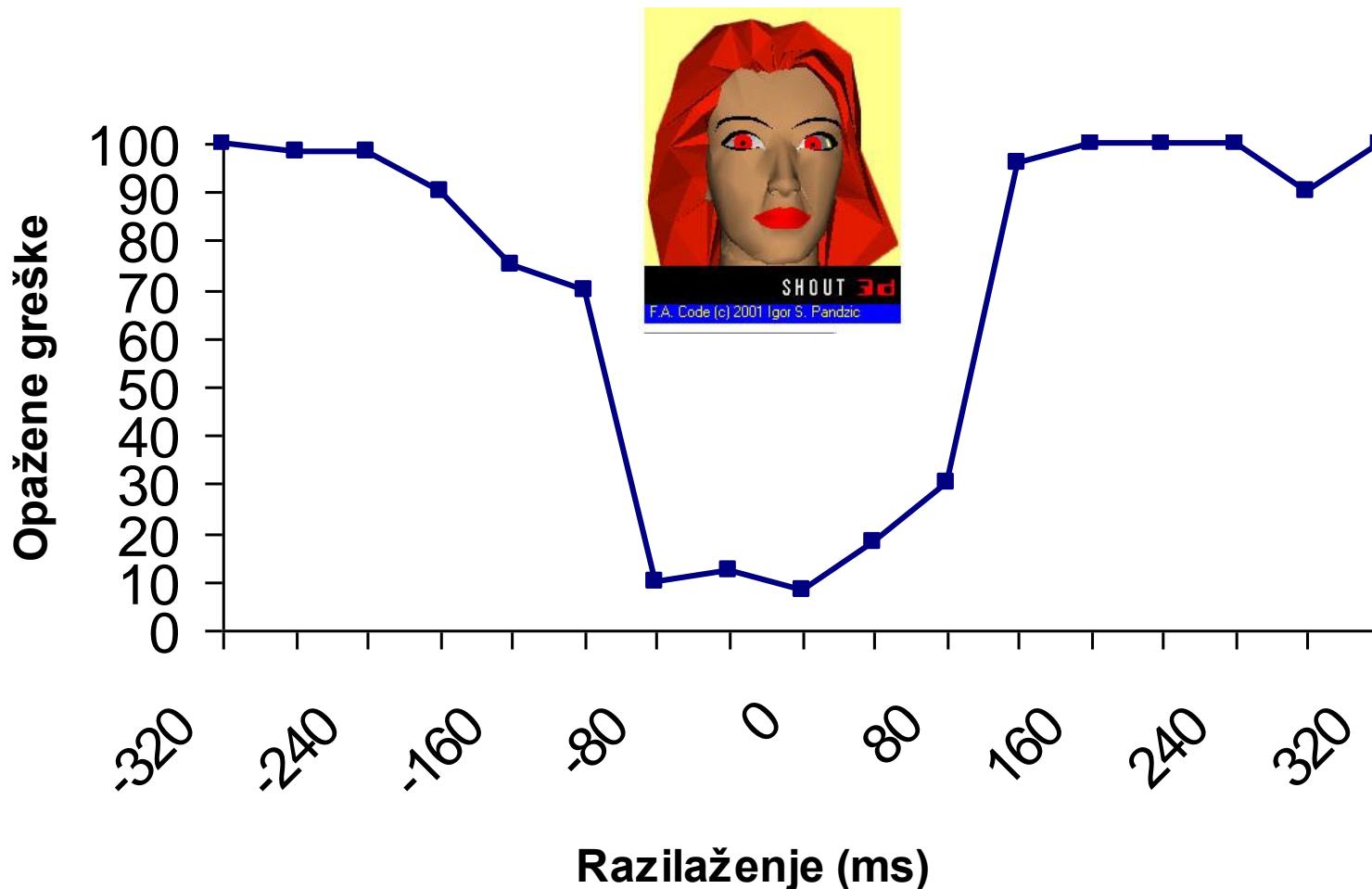


# Kvaliteta usluge

- osim specifikacije sinkronizacije unutar svakog i između zadatah medijskih objekata, specifikacija sinkronizacije mora sadržavati i **opis kvalitete usluge**
- 1. za **pojedini medijski objekt**
  - ovisi o vrsti medija i načinu kodiranja
  - objektivna i subjektivna mjerila (prema poznatim parametrima za pojedine medije)
  - npr. vremenski interval između LDU 1/30 s, dopušteno kolebanje +/- 2 ms i sl.
- 2. **između medijskih objekata**
  - kvaliteta usluge ovisi o uspješnosti usklađivanja međusobnog odnosa medija
  - npr. razilaženje +/- 80 ms

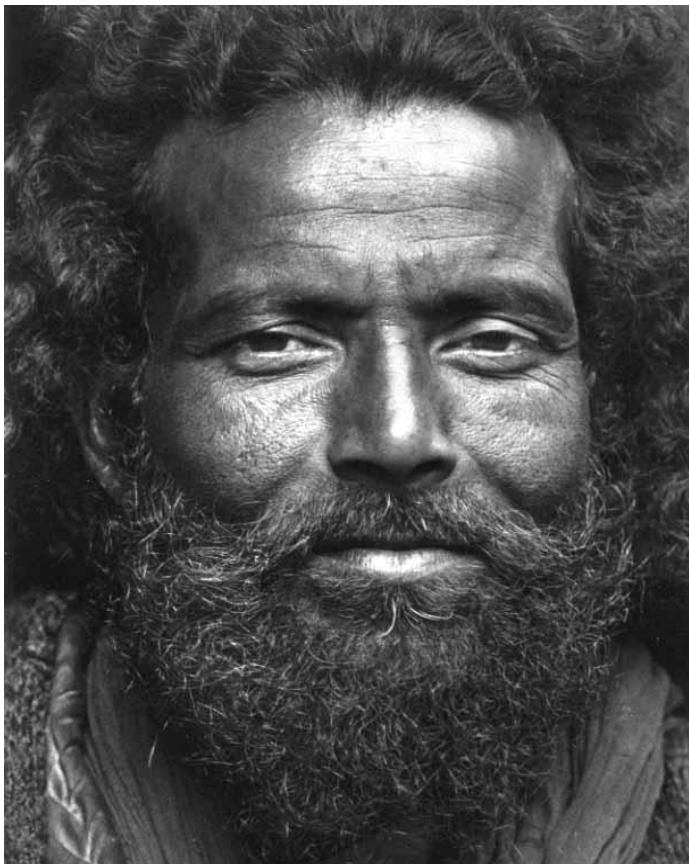
# Sinkronizacija između medijskih objekata (1)

- **Slučaj 1:** sinkronizacija usana i zvuka (govora)



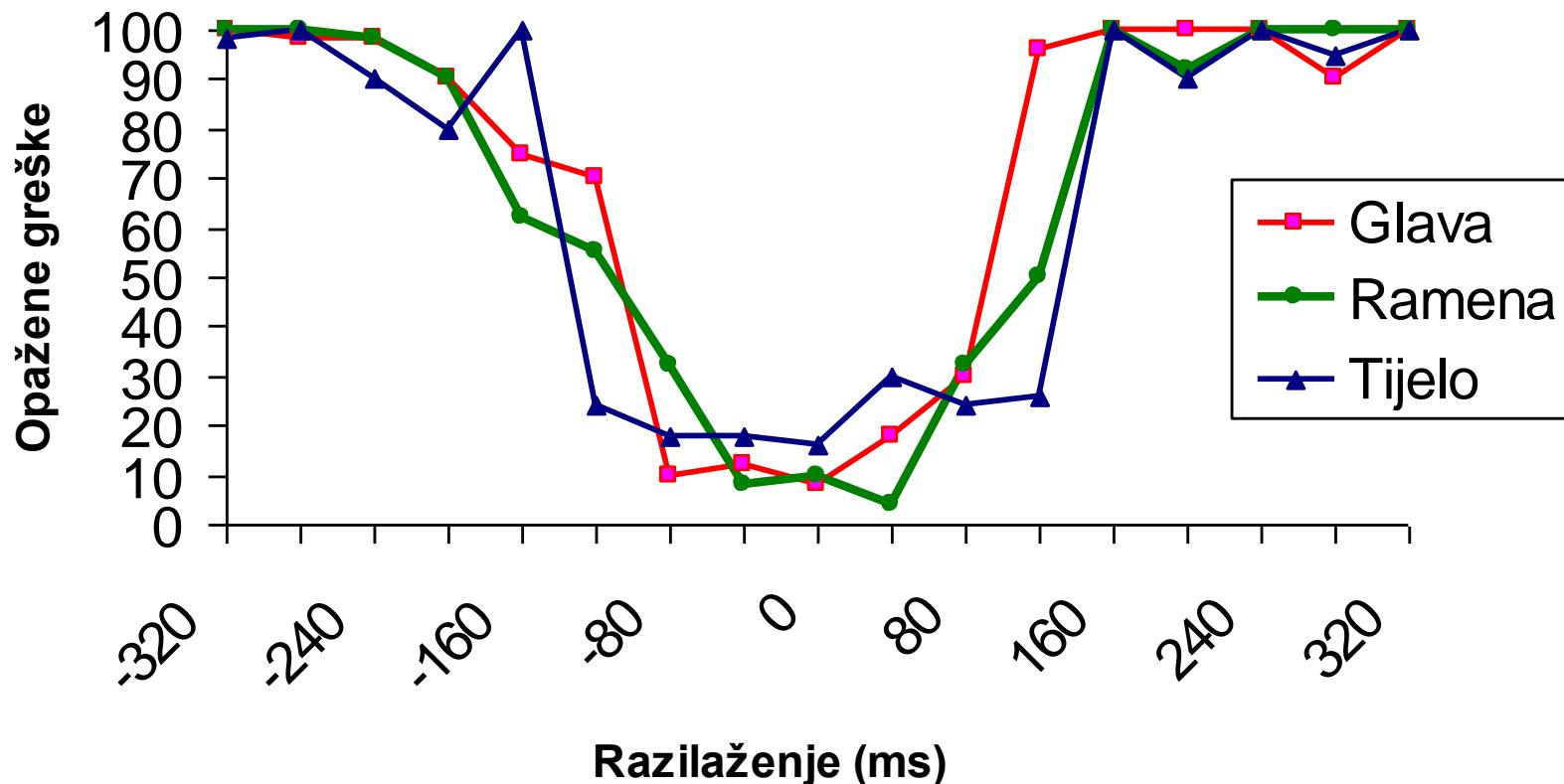
# Pogled: glava – ramena- tijelo

- percepcija sinkronizacije ovisi i o pogledu: nije svejedno promatramo li samo glavu odn. lice, glavu i ramena, ili cijelu osobu



# Sinkronizacija između medijskih objekata (2)

- **Slučaj 2:** sinkronizacija usana u ovisnosti o pogledu
- uočava se najveća osjetljivost na pogled glave (samo lice u kadru)

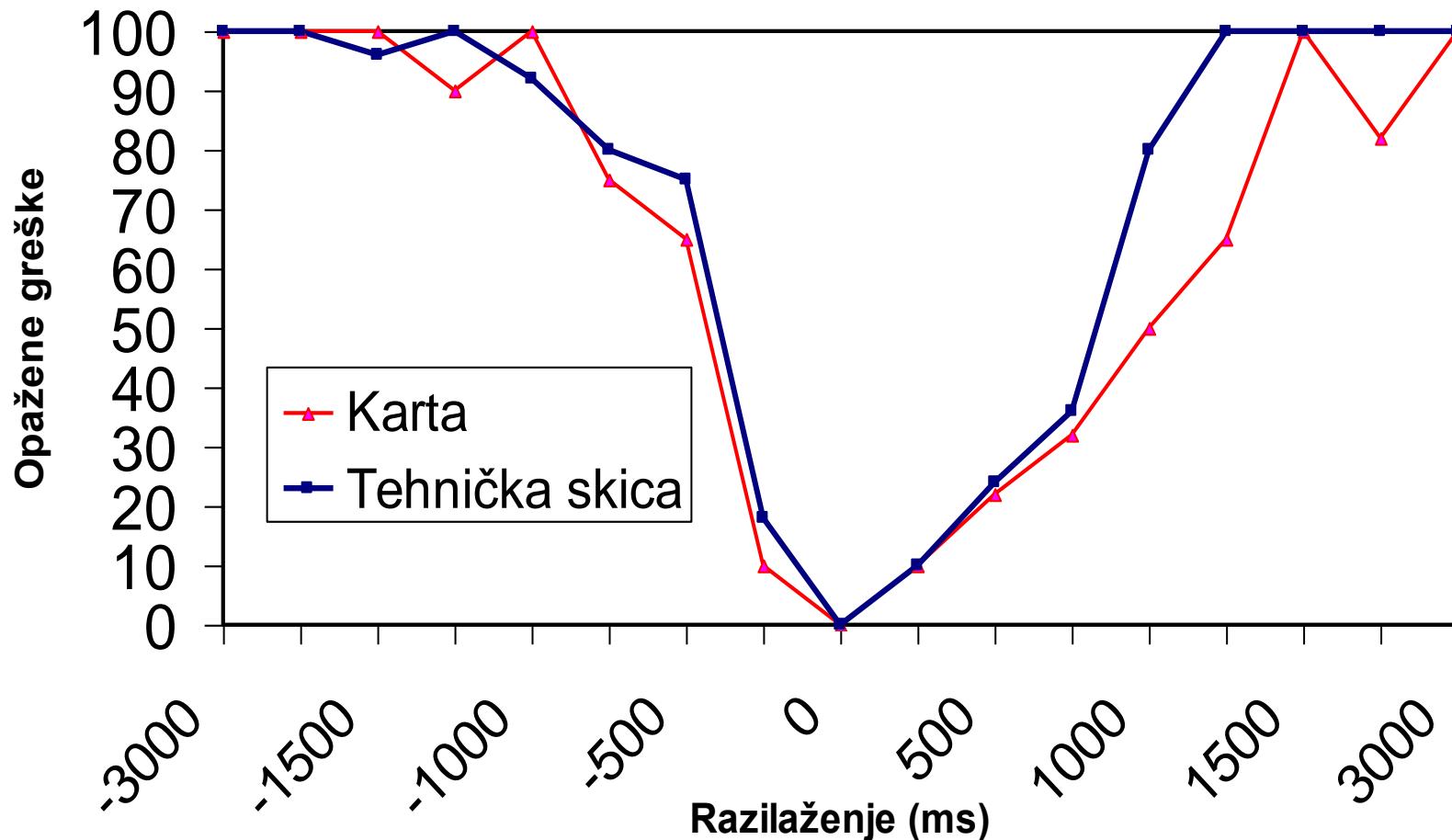


# Sinkronizacija između medijskih objekata (3)



Zavod za  
telekomunikacije

- **Slučaj 3:** pokazivač i zvuk (govor)



# Sinkronizacija između medijskih objekata (4)

- Još nekoliko tipičnih primjera:
  - Audio i animacija
    - plesni tečaj
      - plesni koraci objašnjeni uz takt glazbe
    - akcijski film
      - zvuk mora biti usklađen s trenutkom kad se vidi udarac
  - Dvije audio struje
    - jako povezane
      - npr. lijevi i desni (stereo) kanali snimljenog koncerta
    - slabo povezane
      - npr. pozadinska glazba i govor
- **Kvaliteta usluge** za sinkronizaciju između medija obično se izražava preko dopuštenog vremenskog **razilaženja** od “savršene sinkronizacije”

# Kvaliteta usluge (1)

- **video** u kombinaciji s ostalim medijima

Medij	Način, primjena	QoS
video	animacija	povezani
	audio	sinkronizacija usana
	nepomična	preklapajući
	slika	nepreklapajući
		preklapajući
	tekst	nepreklapajući

# Kvaliteta usluge (2)

- audio u kombinaciji s ostalim medijima

Medij		Način, primjena	QoS
audio	animacija	povezani događaji	+/- 80 ms
	audio	jako povezani	+/- 11 us
		slabo povezani	
		razni sudionici	+/- 120 ms
		slabo povezani	
		glazbena podloga	+/- 500 ms
	nepomična	jako povezani	
	slika	(npr. glazba i note)	+/- 5 ms
		slabo povezani	
		(npr. slide show)	+/- 500 ms
	tekst	komentari uz tekst	+/- 240 ms
	pokazivač	audio vezan za	-500 ms ...
		pokazani objekt	+750 ms

pokazivač prije audia 500ms, a  
 pokazivač poslije audia 750 ms

---

# Primjer specifikacije sinkronizacije: Synchronized Multimedia Integration Language (SMIL)

# SMIL

- W3C Recommendation: Synchronized Multimedia Integration Language (SMIL 2.0)
- <http://www.w3.org/AudioVideo/>
- SMIL je format za objedinjavanje i sinkronizaciju skupa neovisnih višemedijskih elemenata u zajedničku višemedijsku prezentaciju
  - deklarativni jezik - parovi atribut/vrijednost
  - medijski elementi (tekst, grafika, audio, video...) referiraju se preko Uniform Resource Identifiera
  - sinkronizacija
    - paralelno ili sekvencijalno izvođenje
- aplikacije - RealPlayer, Helio SOJA, GRiNS, QuickTime player, IE 5.5+

# Elementi višemedijske prezentacije

- **sadržaj** (medijske komponente)
- **prostorni raspored**
- **vremenski raspored**
  - <par> - medijski elementi se izvode u paraleli; redoslijed nije bitan
  - <seq> - medijski elementi se izvode u nizu; redoslijed je bitan
- semantičko pridruživanje
- veze (izvor i odredište)
- alternativni sadržaj

# Sadržaj (medijske komponente)

- pozadinska slika



e i c k \_ a r c t r a n . j p g

- slika 1



J e d a n . j p g

- slika 2



D v a . j p g

- slika 3



T r i . j p g

- audio



P i n k P a n t h e r T h e m e . a u

- tekst



T h a n k s . t x t

VMK

• Usklađivanje medija



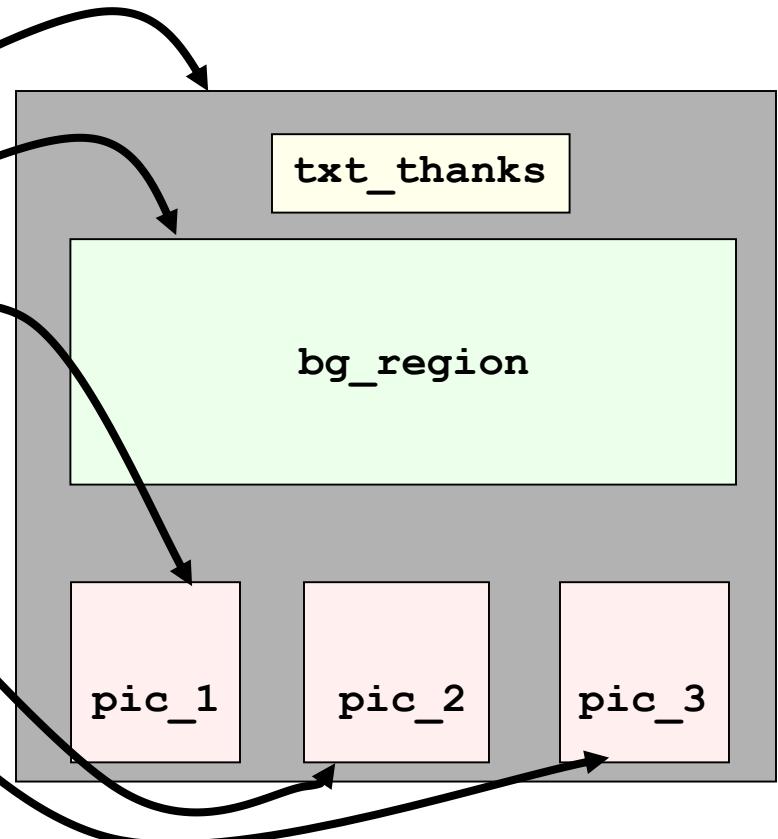
d e m o z z t . s m i l

# Prostorni raspored

```

<smil>
  <head>
    <layout>
      <root-layout width="512" height="397"
                   background-color="#000000" />
      <region id="bg_region" width="512"
              height="286" left="0" top="0"
              background-color="#000000" />
      <region id="pic_1" width="80"
              height="111" left="68"
              top="286" />
      <region id="pic_2" width="80"
              height="111" left="216"
              top="286" />
      <region id="pic_3" width="80"
              height="111" left="364"
              top="286" />
      <region id="txt_thanks" width="120"
              height="20" left="200" top="10"
              background-color="#000000" />
    </layout>
  </head>
  <body>
    .....
  </body>
</smil>

```

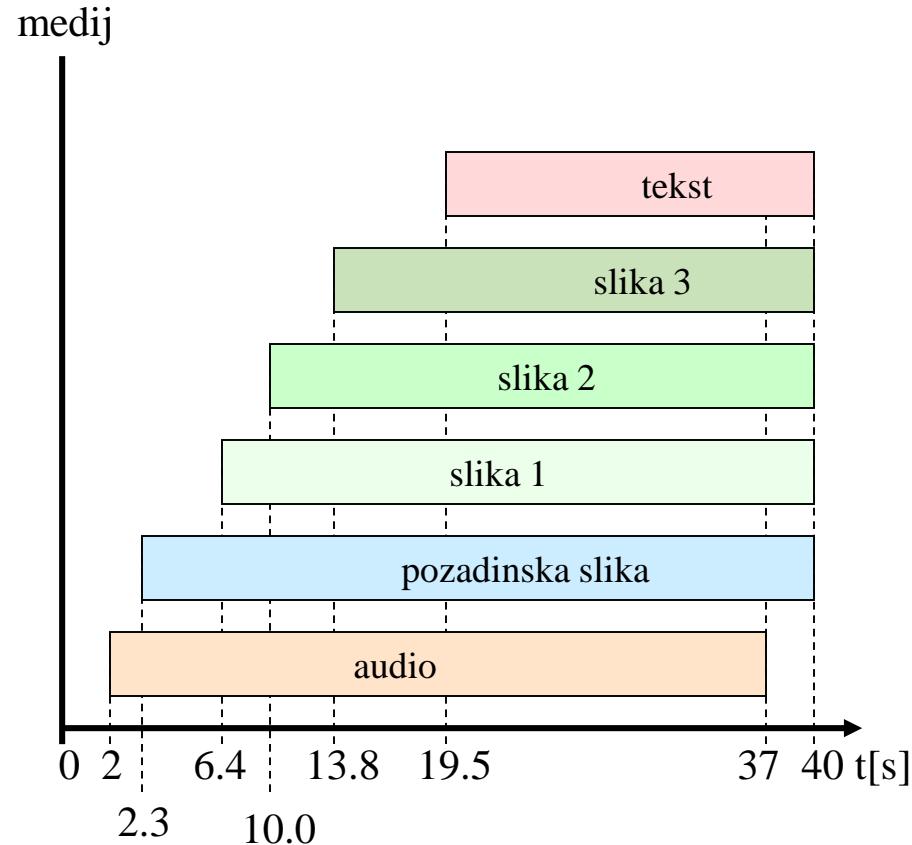


# Vremenski raspored

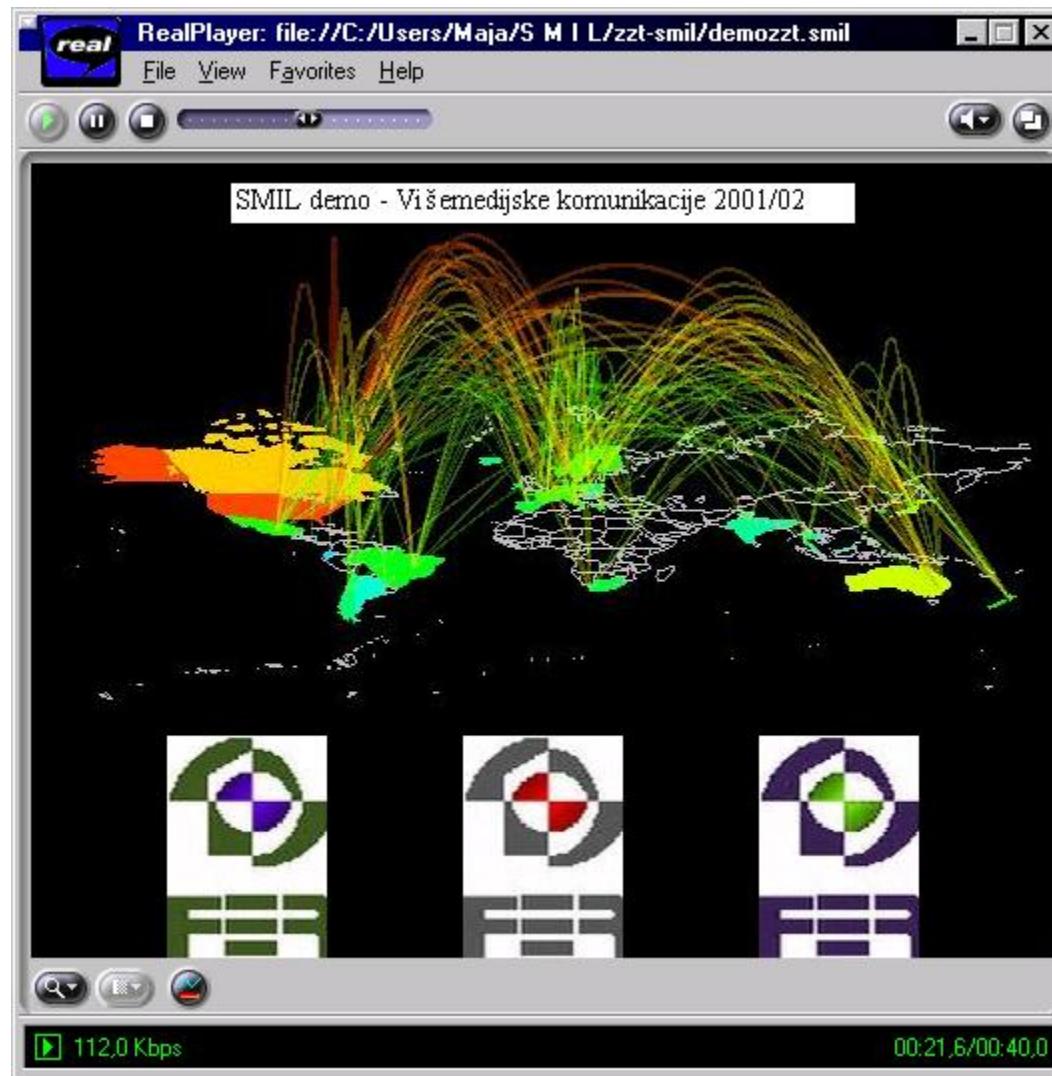
```

<smil>
  <head>
    <layout>
    .....
    </layout>
  </head>
  <body>
    <par dur="40s">
      <audio src="PinkPantherTheme.au"
        type="audio/x-auz" begin="2s"
        dur="35s"/>
      
      
      
      
      <text src="thanks.txt"
        region="txt_thanks" begin="19.5s"/>
    </par>
  </body>
</smil>

```



# Prikaz SMIL dokumenta





Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

9.

Arhitektura višemedijskog  
komunikacijskog sustava

# Pregled predavanja do kraja semestra

---

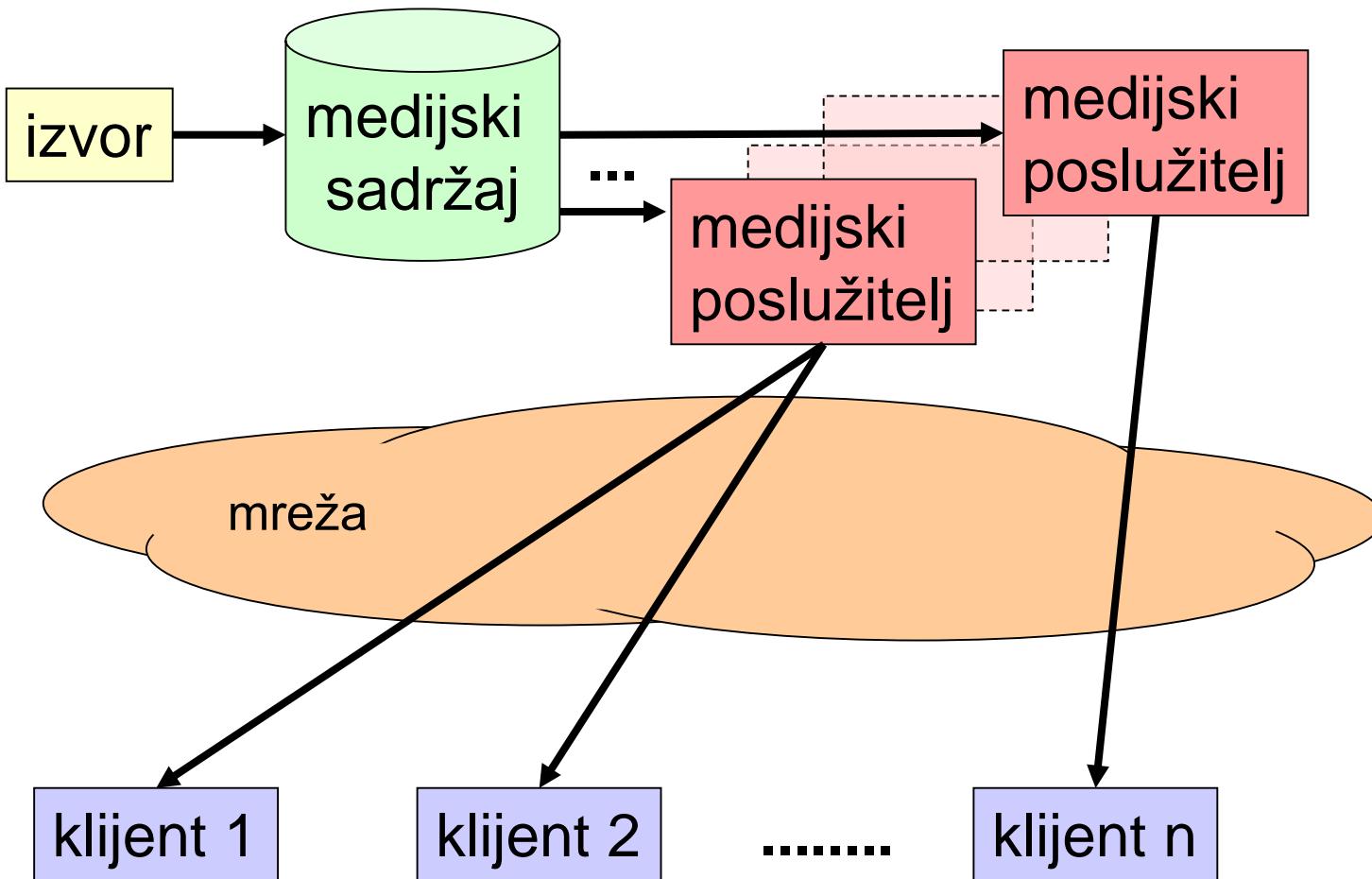
- arhitektura višemedijskog komunikacijskog sustava
  - arhitektura klijent – poslužitelj
  - višeodredišna arhitektura
- protokolna arhitektura
  - koncepcijski model
  - primjeri stvarnih arhitektura
- transportni podsustav
  - višeodredišna komunikacija
  - strujanje višemedijskog sadržaja
  - kvaliteta usluge
- aplikacijski podsustav
  - protokoli za podršku sjednice
  - kvaliteta usluge složene višemedijske usluge – umreženih igara

# Arhitekture višemedijskih komunikacijskih sustava

- **arhitektura klijent – poslužitelj**
  - “klasičan” pristup, primjena kod većine internetskih usluga
  - glavni problem: prilagodljivost veličini
    - količina podataka
    - broj istovremenih korisnika
  - jedno od rješenja: višeposlužiteljski sustavi
- **višeodredišna arhitektura**
  - primjene kod kojih više korisnika koristi isti sadržaj ili generira sadržaj koji se distribuirala svima ostalima
  - glavni problem: usklađivanje raspodijeljenih podataka
    - nužni posebni mehanizmi za održavanje konzistentnosti

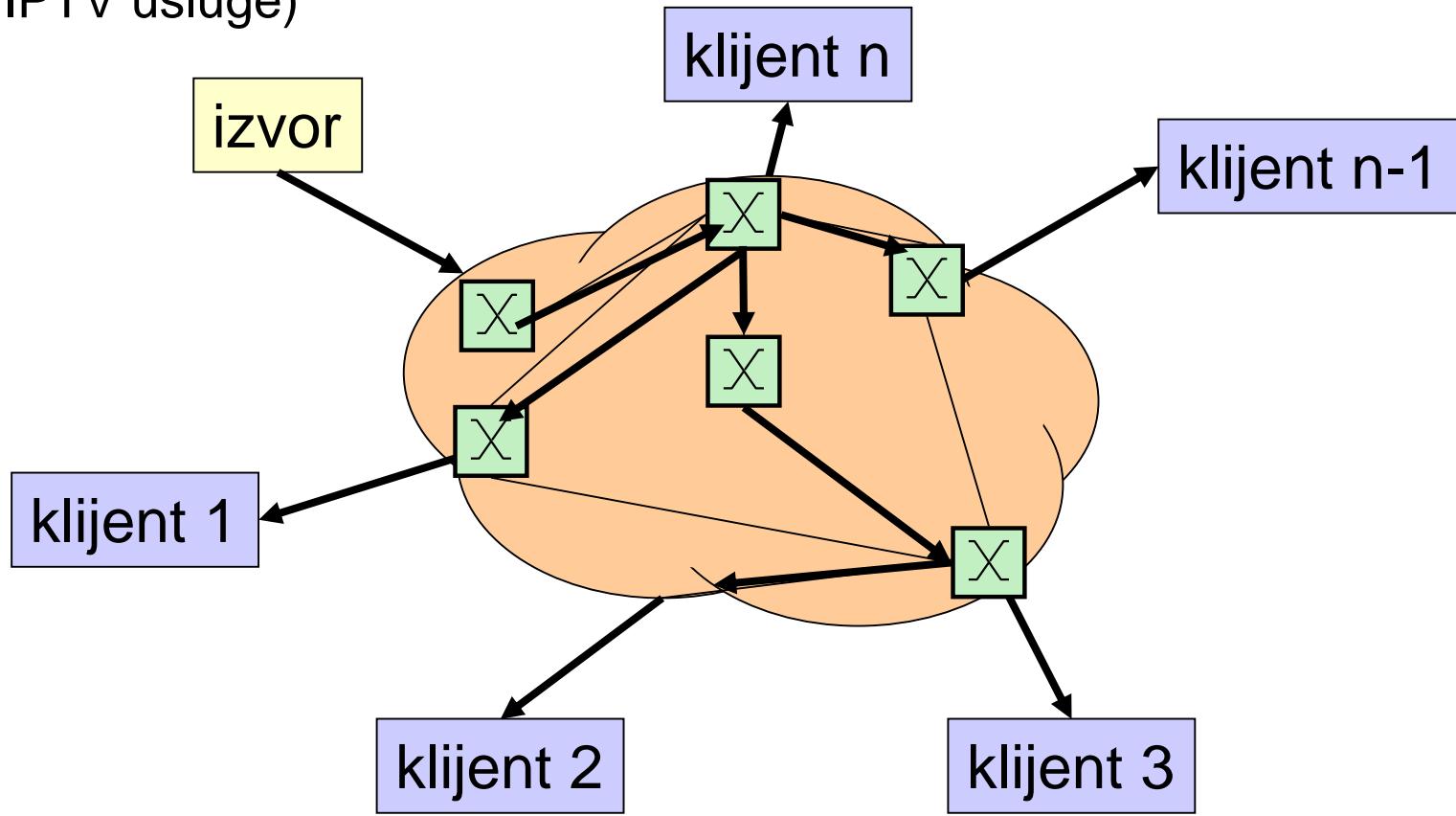
# Arhitektura klijent – poslužitelj

- uglavnom višeposlužiteljske arhitekture (više)medijskog sadržaja
- tipična primjena: distribucijske usluge



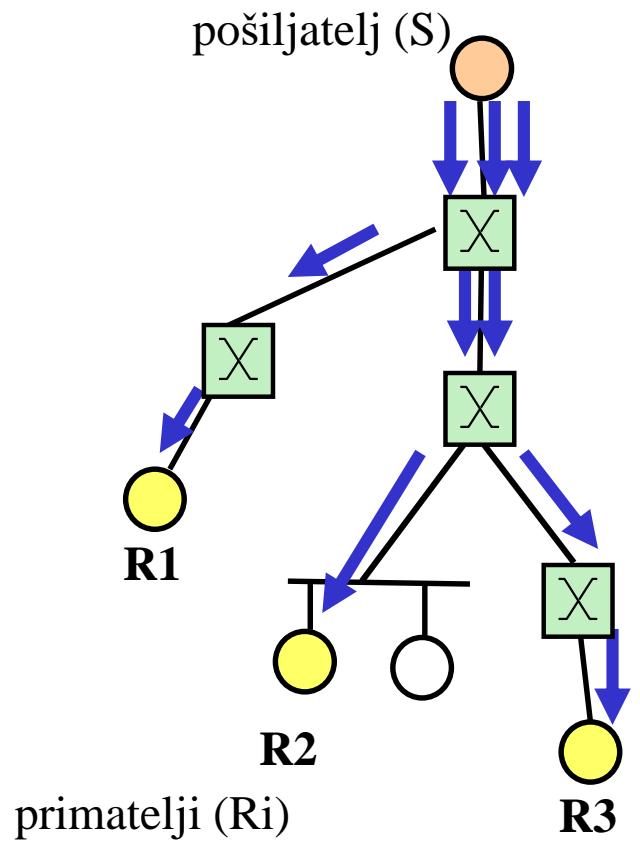
# Višeodredišna arhitektura

- višeodredišno razašiljanje (engl. *multicast*)
- komunikacija jedan-na-više
- tipična primjena: interaktivne usluge (npr. audio/video-konferencija, IPTV usluge)

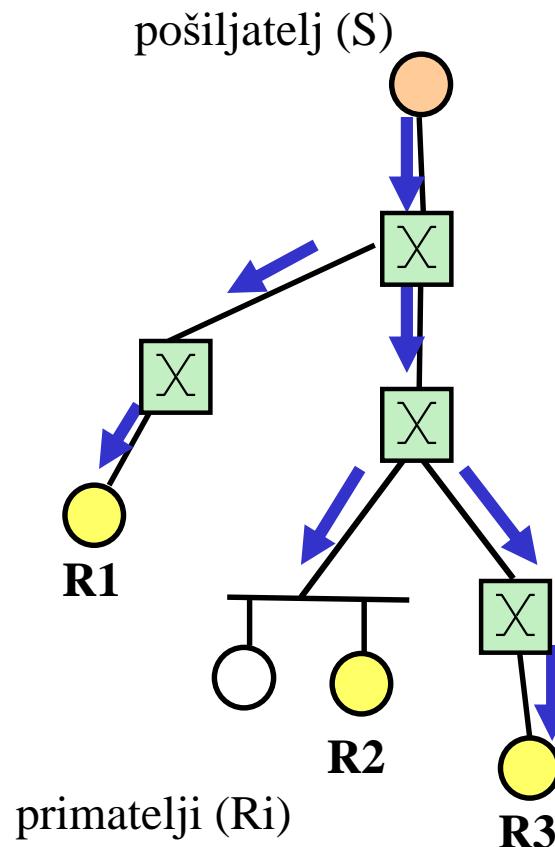


# Usporedba sa stajališta opterećenja

višestruko pojedinačno (unicast)



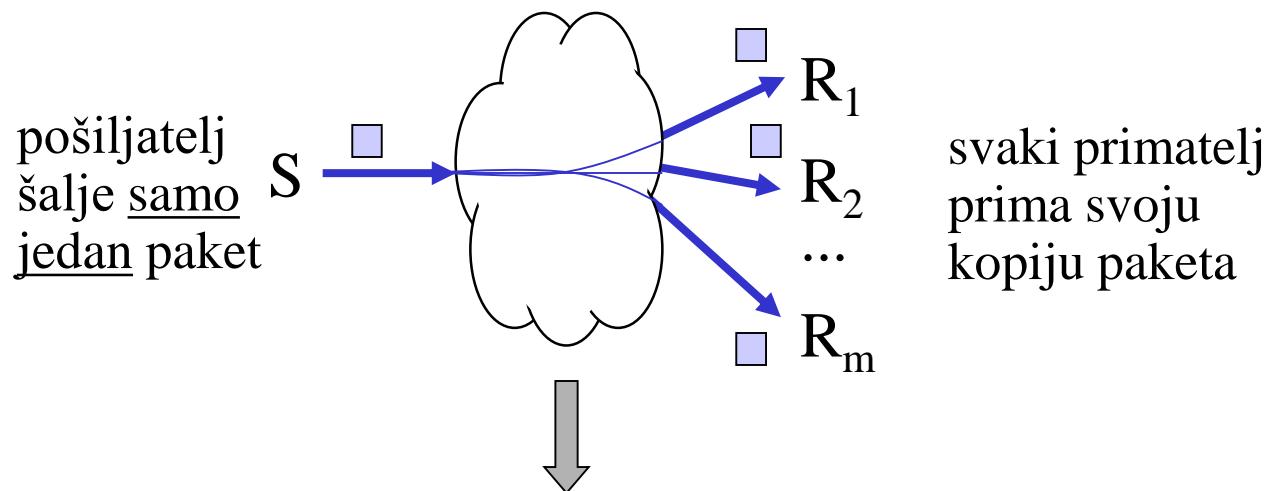
**višeodredišno (multicast)**



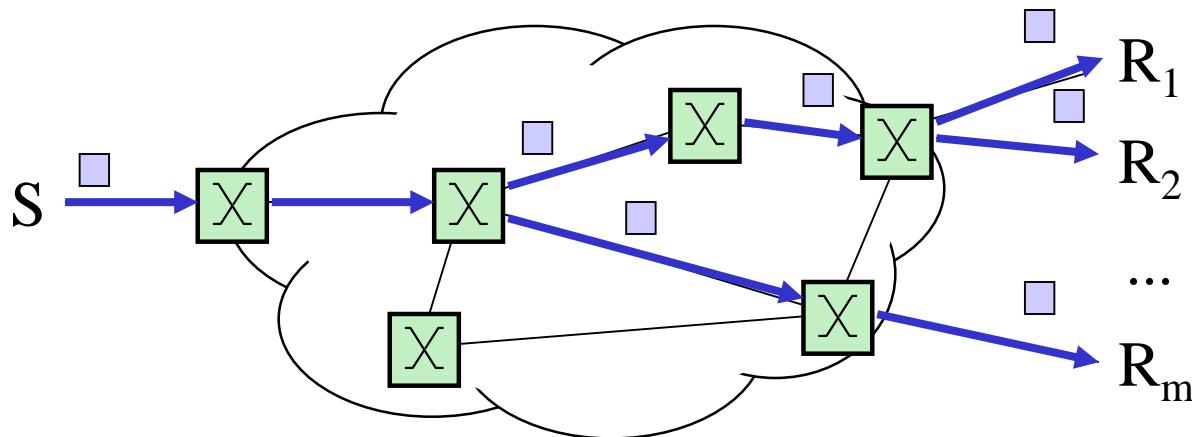
# Ideja i izvedba višeodredišnog razašiljanja

(tema sljedećeg predavanja)

## Ideja

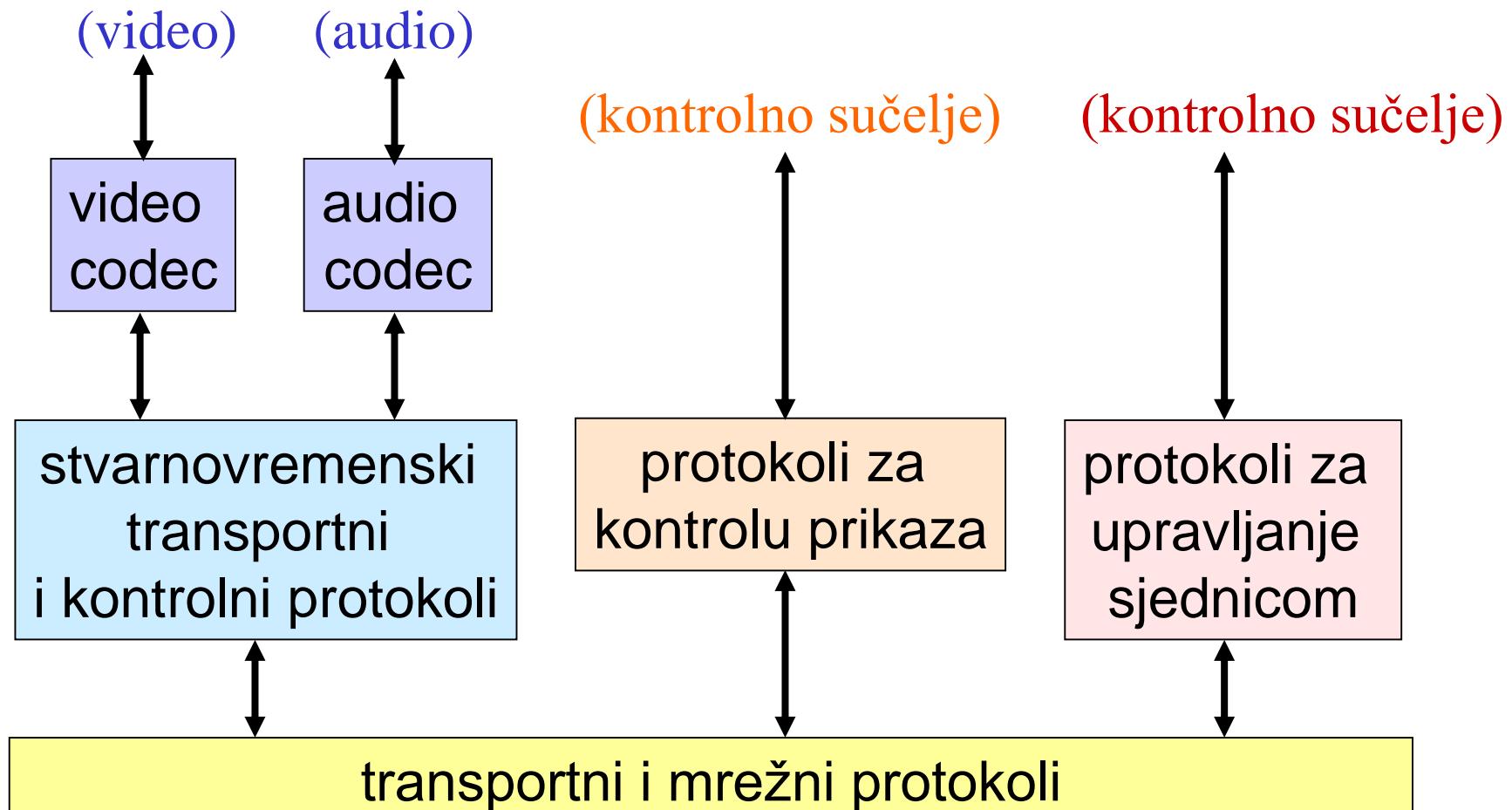


## Izvedba



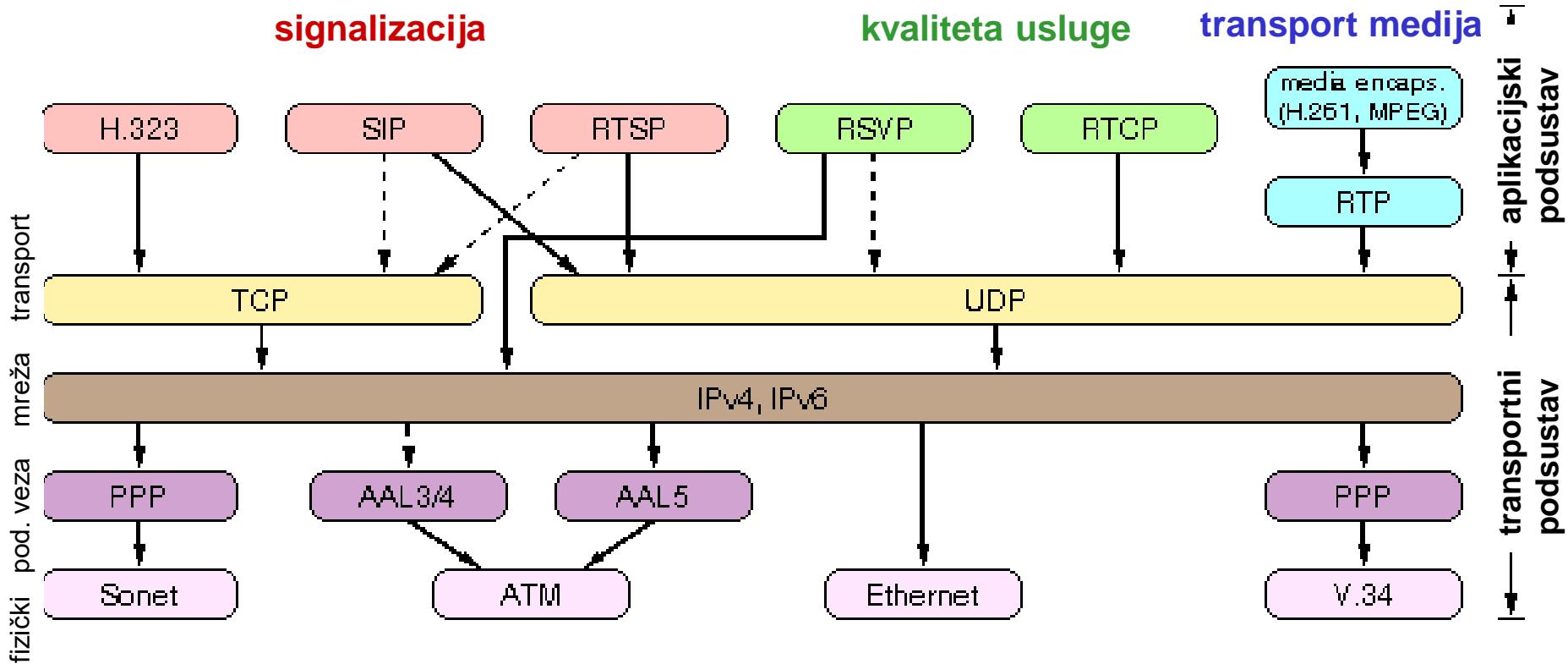
ključni elementi su usmjerenitelji!

# Protokolna arhitektura (ideja)



# Protokolna arhitektura: Internet (1)

Izvor: <http://www.cs.columbia.edu/~hgs/internet/>



Oznake:

RTP – Real-time Transport Protocol

RTCP – RTP Control Protocol

RTSP – Real Time Streaming Protocol

RSVP – Resource Reservation Protocol

SDP – Session Description Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

IP – Internet Protocol

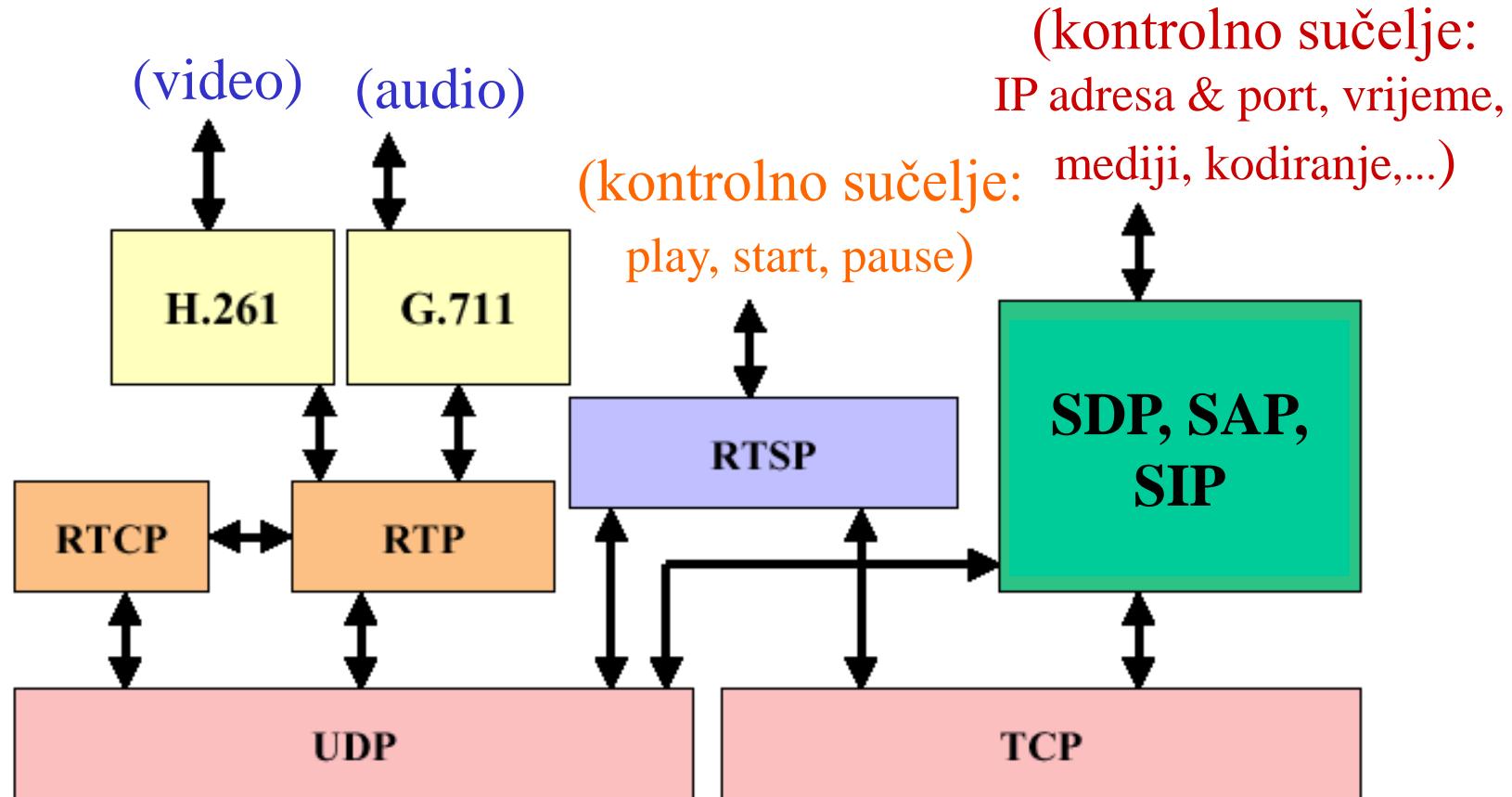
PPP – Point-to-Point Protocol

ATM – Asynchronous Transfer Mode

AAL – ATM Adaptation Layer

# Protokolna arhitektura: Internet (2)

- primjer protokolne arhitekture za audio-video konferenciju i strujanje



Oznake:

RTP – Real-time Transport Protocol  
RTCP – RTP Control Protocol  
RTSP – Real Time Streaming Protocol

SDP – Session Description Protocol  
SAP – Session Announcement Protocol  
SIP – Session Initiation Protocol

# Aplikacijski podsustav

- obuhvaća više slojeve (iznad sloja transporta) arhitekture višemedijskog komunikacijskog sustava
- zanimaju nas tri komponente:
  - podrška za suradničke aplikacije
    - skupni naziv za “računalno-podržani zajednički rad” (engl. *Computer Supported Collaborative Work, CSCW*)
    - npr. zajedničko uređivanje dokumenata, teksta, slike, dizajn
  - podrška za konferencijske aplikacije
    - npr. audio konferencija, video konferencija, distribuirane igre
  - podrška za upravljanje sjednicom

# Dimenzije suradnje

- parametri kategorizacije:

- vrijeme

- sinkrono (svi sudionici istovremeno prisutni)
    - asinkrono (svi sudionici nisu istovremeno prisutni)

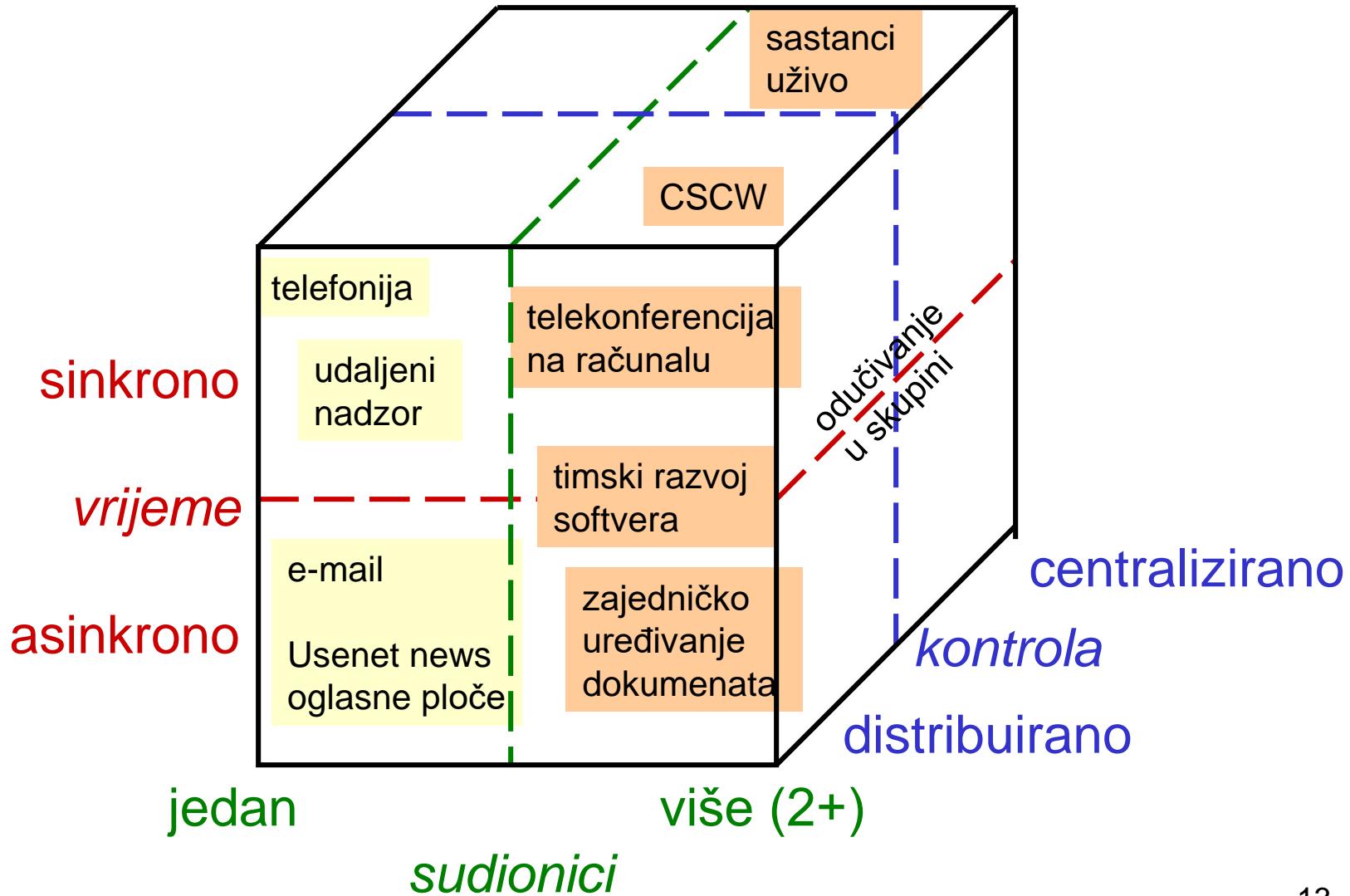
- sudionici

- broj sudionika: jedan, dva ili više
    - dinamika skupine: statičke ili dinamičke skupine
    - homogeno ili heterogeno članstvo
    - uloge (osnivač sjednice, voditelj, član, promatrač)

- način kontrole

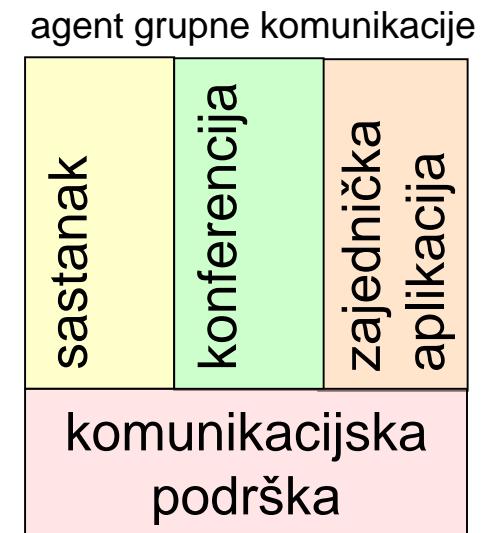
- centralizirano (jedan član skupine djeluje kao predsjedavajući i nadzire rad skupine)
    - distribuirano (svaki član skupine odgovoran za svoj dio posla, mora postojati protokol koji osigurava konzistentnost)

# Primjeri usluga u odnosu na dimenzije suradnje

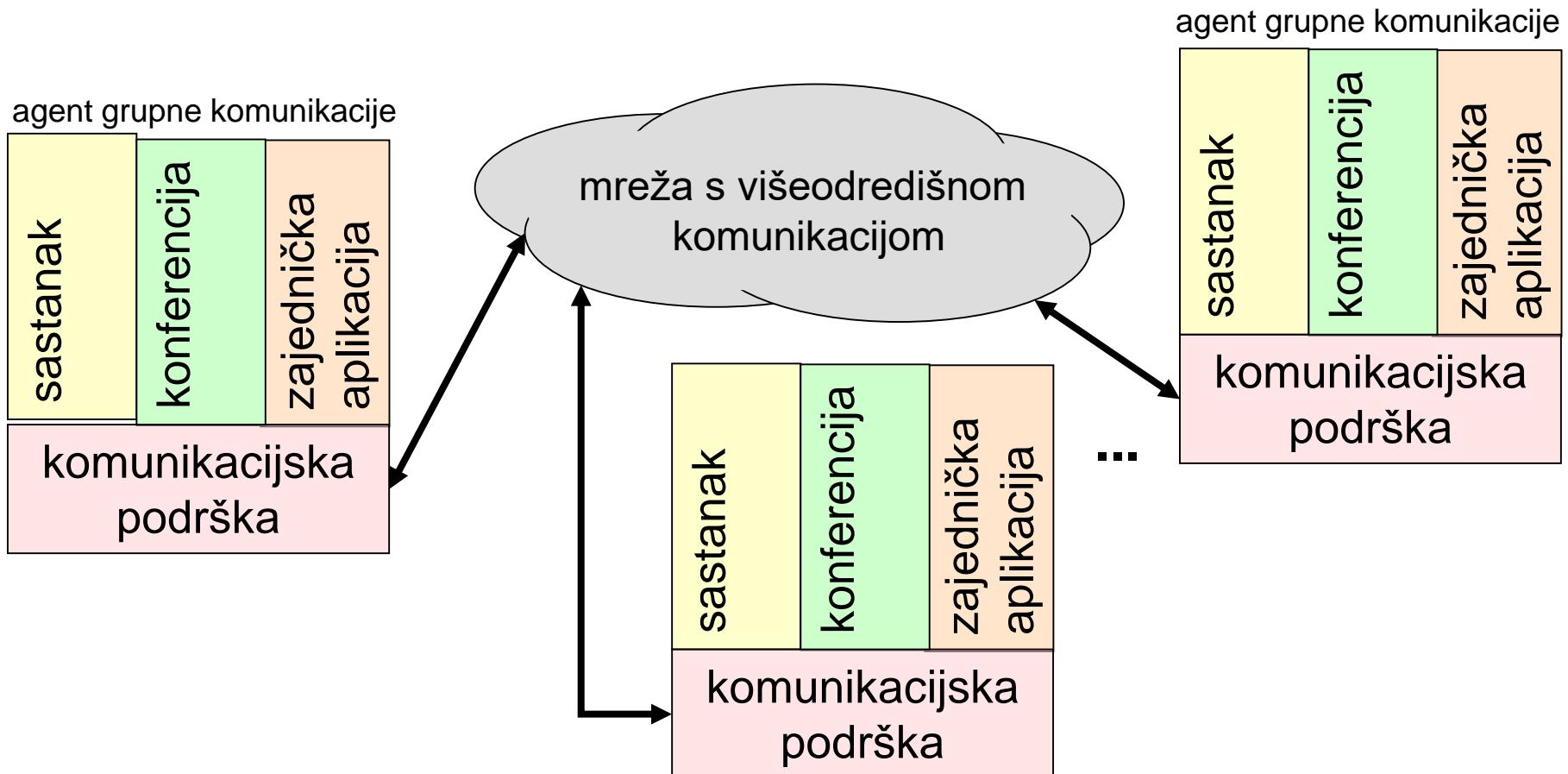


# Model podrške grupne komunikacije

- **grupna komunikacija:** sinkrona ili asinkrona komunikacija više korisnika s centraliziranoj ili distribuiranoj kontrolom
- model uključuje:
  - agente grupne komunikacije (po jedan za svakog sudionika), koji obuhvaća podršku za sastanak, konferenciju, zajedničku aplikaciju i višeodredišnu komunikaciju
  - mrežu s višeodredišnom komunikacijom



# Model podrške grupne komunikacije



# Podrška grupne komunikacije: sastanak

- mehanizmi sastanka obuhvaćaju razmjenu uputa i podataka o uspostavljanje sjednice i samom sastanku:
  - **definicija sjednice**
    - naziv
    - opis
    - kontaktni podaci pozivatelja ili organizatora
  - **gdje**
    - IP adresa, TCP/UDP port
  - **tko**
    - inicijator, višeodredišna adresa ili popis pozvanih
  - **kako**
    - vrste medija, način kodiranja, komunikacijski parametri
  - **kada**
    - UTC, GMT
  - **oglašavanje**

# Načini oglašavanja sjednice

---

- **sinkroni** načini oglašavanja
  - imenik sjednica (*session directory*) - omogućuje pozivanje skupine zainteresiranih preko višeodredišnog adresiranja
    - mogućnost asinkronog načina preko sačuvanog popisa sastanaka, sudionika i sl. u on-line kalendaru sjednica
  - izravno pozivanje (adresa pozvanog mora biti poznata!)
  - primjer: Mbone session directory, sdr
- **asinkroni** načini oglašavanja
  - osobni e-mail, mailing liste
  - bulletin board
  - WWW sjedište

# Podrška grupne komunikacije: konferencija

- konferencija je **interaktivna konverzacijska** usluga
- kontrola konferencije obuhvaća tri faze:
  - 1. uspostava konferencije
    - uloge korisnika (priključivanje skupini, npr. predsjedavajući, član, promatrač)
    - pravila ponašanja (izmjena prava riječi)
    - vrste podataka (audio, video, tekst, podaci ...) i način kodiranja
    - registracija, prijem
    - pregovaranje o medijima i formatima
  - 2. održavanje konferencije
    - razmjena podataka u stvarnom vremenu
    - dodavanje novih korisnika
    - odlazak postojećih
  - 3. zatvaranje konferencije

# Kontrola konferencije

---

- provodi se kontrola nad podacima i nad sudionicima
- kontrola nad podacima
  - vrste medija (audio, video, podaci ...), formati
  - postavljanje parametara prijenosa
  - upravljanje prijenosom
    - npr. hoću primati audio, neću video
    - npr. hoću a/v od izvora x, a samo video od izvora y
- kontrola nad sudionicima
  - operacije vezane za ponašanje članova skupine
  - npr. priključivanje, pozivanje, odlazak
- kontrola može biti centralizirana i distribuirana

# Centralizirana kontrola konferencije (1/2)

---

- inicijator započinje konferenciju pozivanjem članova (mora znati adrese!)
- svaki pozvani član odgovara na poziv priključivanjem u konferenciju
- sudionici mogu imati razne uloge (predsjedavajući, govornik, slušatelj, ...)

# Centralizirana kontrola konferencije (2/2)

- prednosti:
  - jednostavna kontrola nad podacima (svi na jednom mjestu)
  - garantirana konzistentnost
  - pouzdani protokol za razmjenu poruka
  - poznati su resursi po korisniku
- nedostaci:
  - kašnjenje zbog centraliziranog odlučivanja i obrade
  - novi sudionik mora preuzeti cijelo stanje konferencije
  - u slučaju ispada veze, teže obnavljanje stanja

# Distribuirana kontrola konferencije (1/2)

---

- naziva se i “slabo kontrolirani model”
- zasniva se na distribuiranom stanju konferencije
- inicijator konferencije započinje objavu postavljanjem višeodredišne adrese za konferenciju
- svi zainteresirani sudionici imaju uvid u podatke o načinu priključivanja i pridružuju se na postojeću konferenciju
- korisnici se priključuju kako tko hoće, nema ograničenja u odlascima niti dolascima (“radio kanal”)
- nema evidencije o članstvu
- nema čuvanja stanja, odn. evidencije o podacima/ resursima po sudioniku

# Distribuirana kontrola konferencije (2/2)

- **prednosti:**

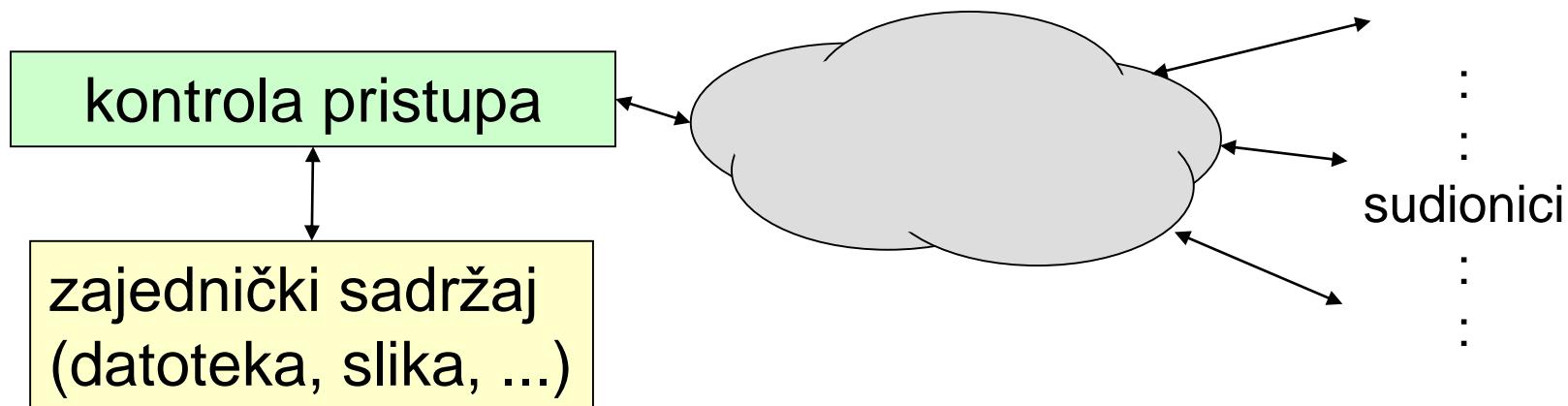
- jednostavan model, objava se može obaviti drugim putem, npr. preko podrške za sastanak
- nema dodatnog procesiranja; brže
- potencijalno veći broj sudionika
- jednostavno obnavljanje stanja u slučaju prekida veze

- **nedostaci:**

- moguća (privremena i/ili povremena) nekonzistentnost zbog nepouzdanog protokola za razmjenu poruka i kašnjenja u mreži
- nema garancije kvalitete (jer se ne kontrolira broj i aktivnost sudionika)

# Podrška grupne komunikacije: zajednička aplikacija

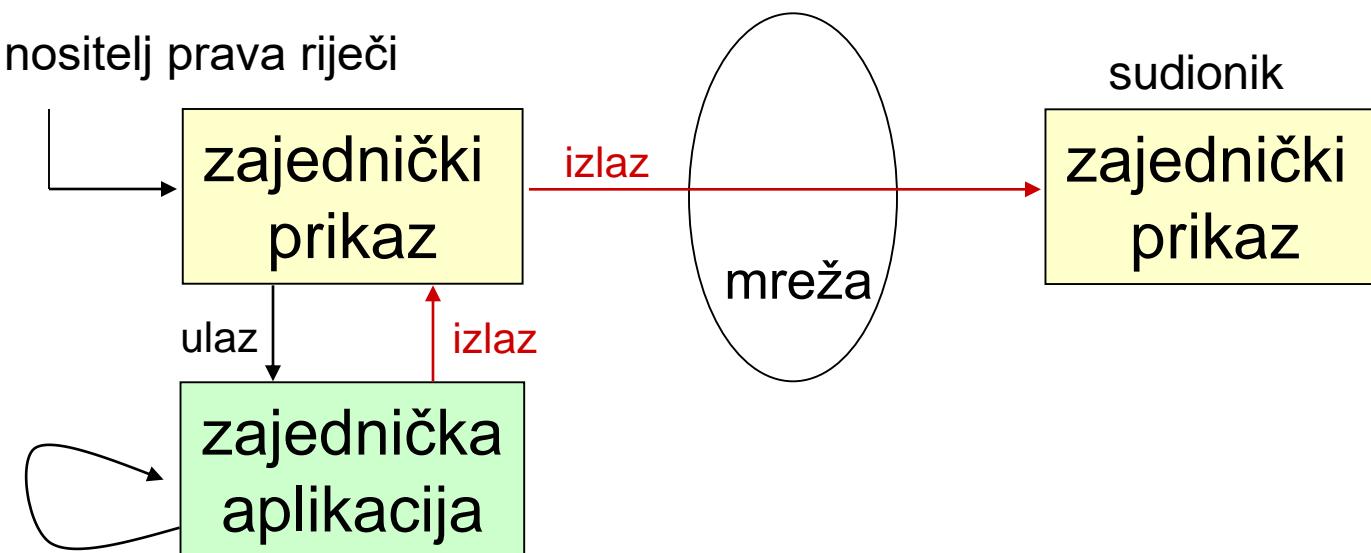
- zajednička aplikacija: obrada zajedničkog sadržaja uz istovremeni prikaz svim sudionicima
- sudionici unose promjene u zajednički sadržaj preko kontrole pristupa i ne mogu direktno mijenjati sadržaj
- unutar kontrole pristupa vrši se provjera prava, prioriteta i međusobna isključivost pristupa, čuva se vremenski slijed promjena



- osnovne arhitekture zajedničke aplikacije: **centralizirana** i **replicirana**

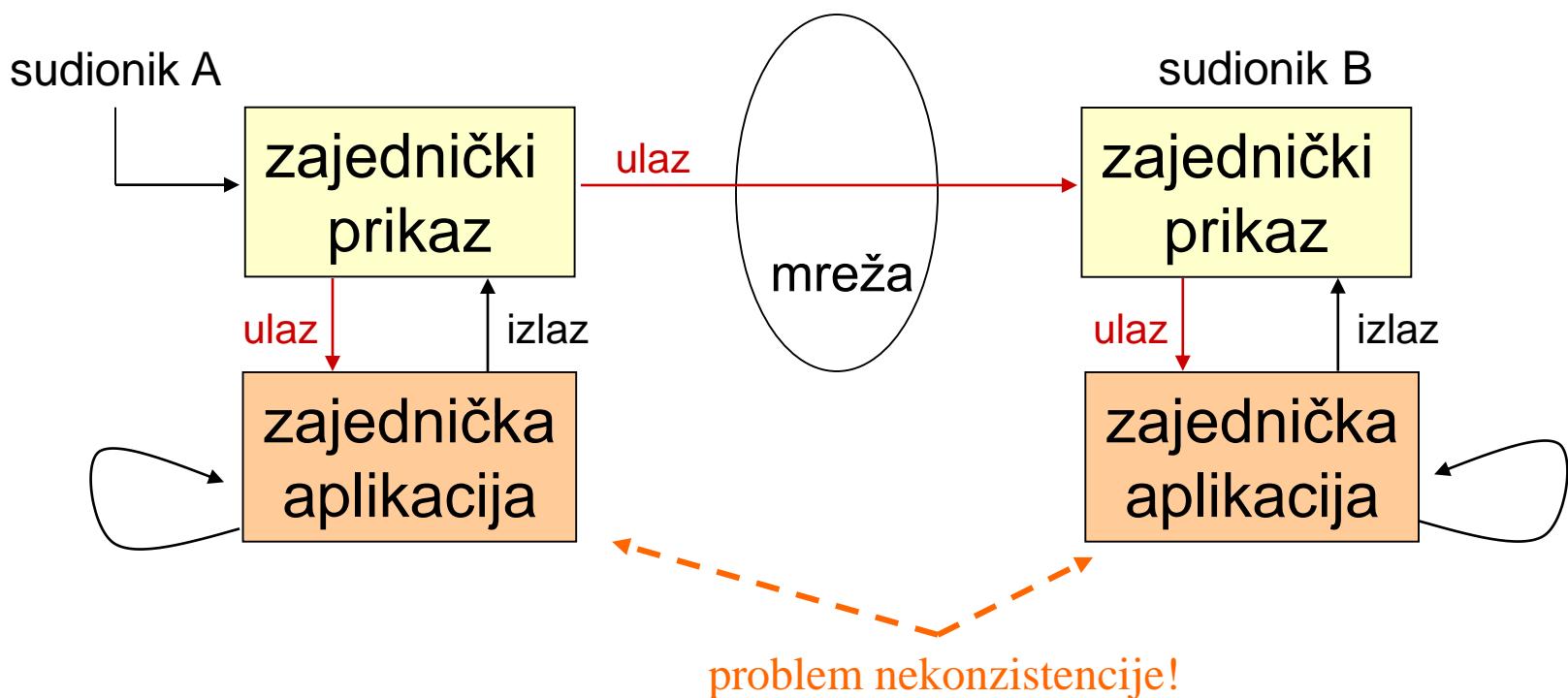
# Zajednička aplikacija s centraliziranim arhitekturom

- postoji samo jedna kopija zajedničke aplikacije na jednom mjestu; samo nositelj prava riječi može vršiti promjene
- svi ulazi se obrađuju lokalno, na jednom “centralnom” mjestu
- novo stanje se distribuira i prikazuje ostalim sudionicima



# Zajednička aplikacija s repliciranom arhitekturom

- postoji po jedna kopija (“replika”) zajedničke aplikacije za svakog sudionika
- svi ulazi se razašilju svim ostalim sudionicima
- obrada svih ulaza i prikaz vrše se lokalno, kod svakog sudionika

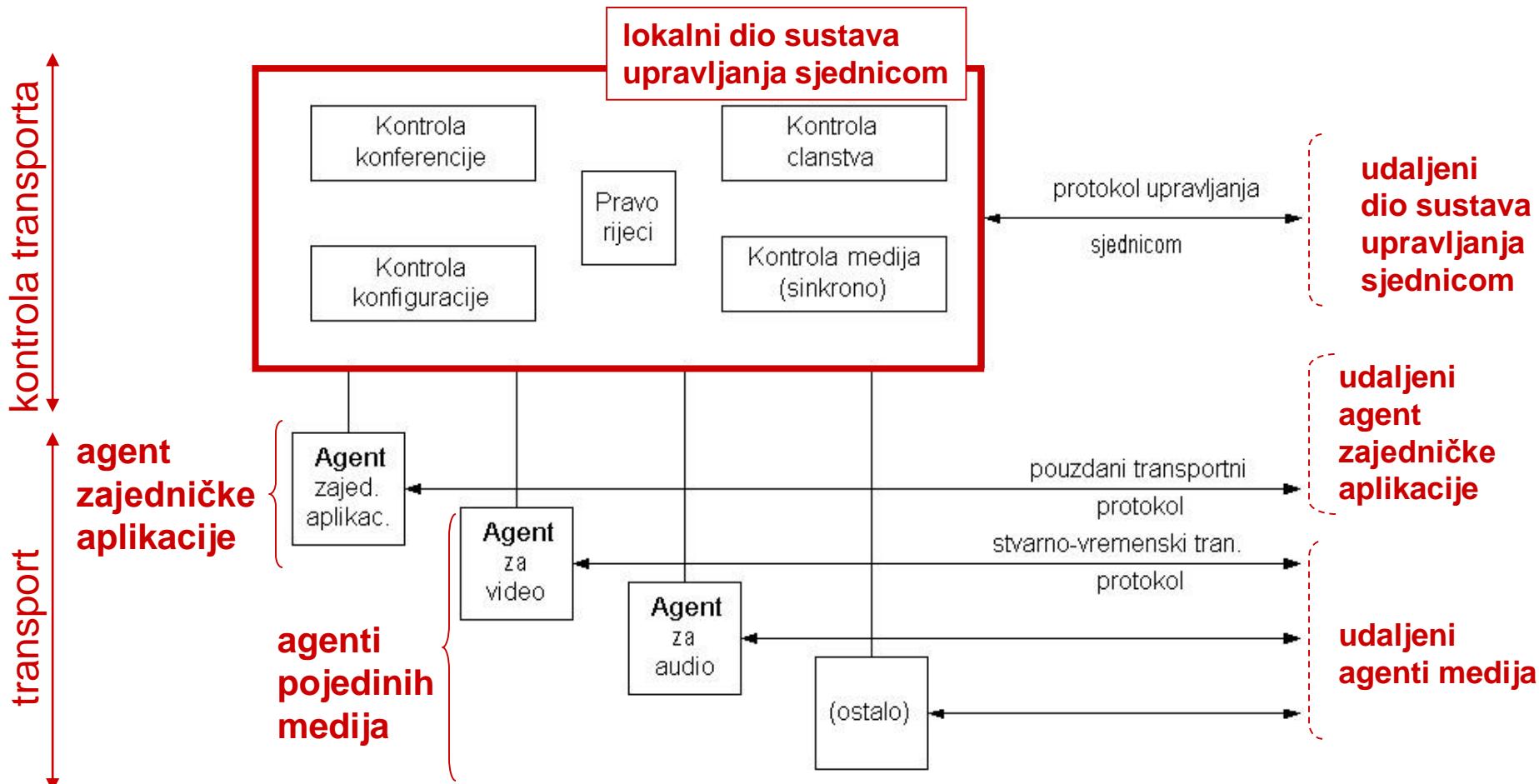


# Usporedba

centralna kontrola pristupa	replicirana kontrola pristupa
<ul style="list-style-type: none"><li>• prednost:<ul style="list-style-type: none"><li>– jednostavno održavanje konzistencije</li></ul></li><li>• nedostaci:<ul style="list-style-type: none"><li>– kašnjenje prikaza ovisi o trajanju ciklusa obrade i prijenosa</li><li>– veće opterećenje mreže jer se svaki put razašilje cjelokupno najnovije trenutno stanje</li></ul></li></ul>	<ul style="list-style-type: none"><li>• prednosti:<ul style="list-style-type: none"><li>– manji promet, jer se šalju samo ulazi (promjene), a ne cjelokupno stanje</li><li>– manje kašnjenje u prikazu</li></ul></li><li>• nedostaci:<ul style="list-style-type: none"><li>– složeno održavanje konzistencije</li><li>– rješava se posebnim protokolima, zaključavanjem, kontrolom vlasništva, otkrivanjem međuzavisnosti podataka i sl.)</li></ul></li></ul>

# Sustav upravljanja sjednicom

- dio sustava koji odvaja kontrolu nad transportom od samog transporta



# Komponente kontrole nad transportom

- kontrola konfiguracije
  - parametri zajednički za cijelu sjednicu, kvaliteta usluge (QoS)
- kontrola konferencije
  - otvaranje, modifikacije u tijeku, zatvaranje konferencije
- kontrola prava riječi
  - pristup zajedničkom sadržaju, pravo riječi u a/v konferenciji, predaja prava riječi ili prava pristupa
- kontrola članstva
  - pozivanje, prijava, odjava, postavljanje parametara
- kontrola usklađivanja medija (sinkronizacija)
  - parametri za usklađivanje svakog medija zasebno
- komunikacija lokalnog i udaljenog dijela se odvija putem protokola upravljanja sjednicom (npr. SIP, SAP, SDP)

# Komponente samog transporta

- agent zajedničke aplikacije
  - mijenjanje i distribucija zajedničkog sadržaja
  - koristi *pouzdani transportni protokol* (npr. TCP)
- agent pojedinog medija
  - npr. agent za audio, agent za video, agent za podatke
  - vrši kontrolu nad jednim medijem (pokretni, zaustavi, privremeno zaustavi,...)
  - koristi *stvarnovremenski transportni protokol* (npr. RTP)



Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

13.

Protokoli za podršku višemedijske  
sjednice (RTP/RTCP, SDP, SAP,  
RTSP)

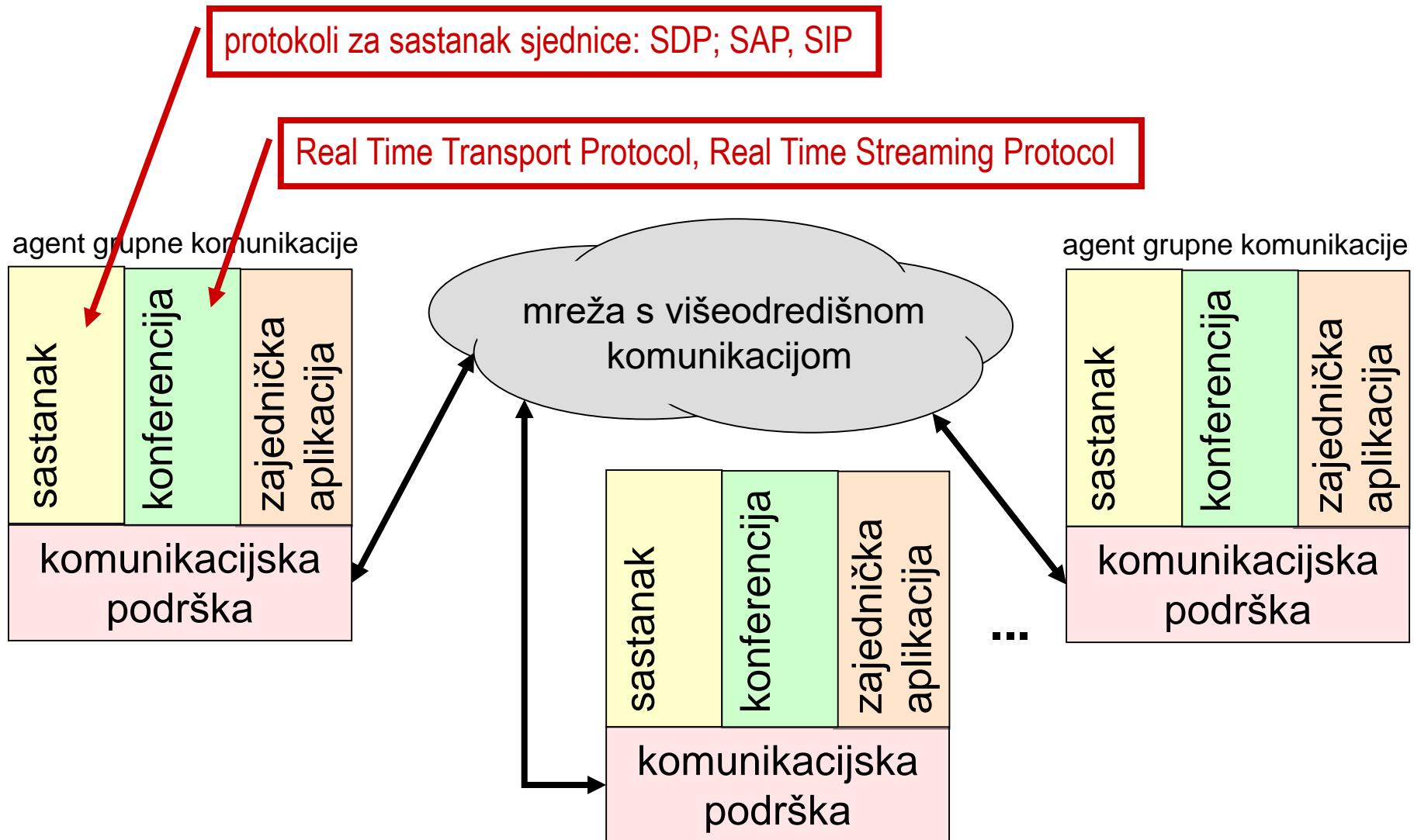
# Sadržaj predavanja

- Protokol za prijenos višemedijskih sadržaja:
  - Real-time Transport Protocol (RTP) i RTP Control Protocol (RTCP)
- Protokoli za opis i objavu sjednice:
  - Session Description Protocol (SDP)
  - Session Announcement Protocol (SAP)
- Protokol za kontrolu prijenosa višemedijskih sadržaja:
  - Real-Time Streaming Protocol (RTSP)

# Model podrške grupne komunikacije (podsjetnik)

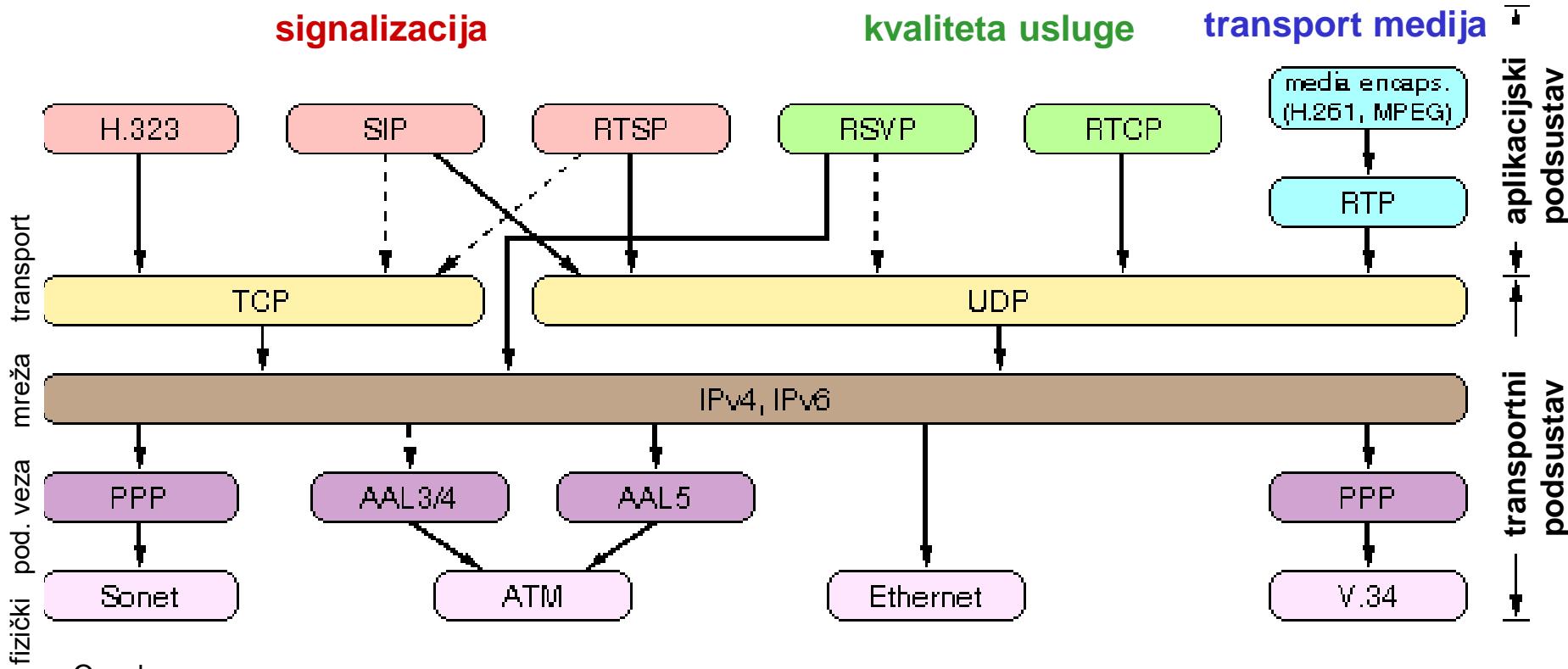


Zavod za telekomunikacije



# Protokolna arhitektura: Internet (1)

Izvor: <http://www.cs.columbia.edu/~hgs/internet/>



#### Oznake:

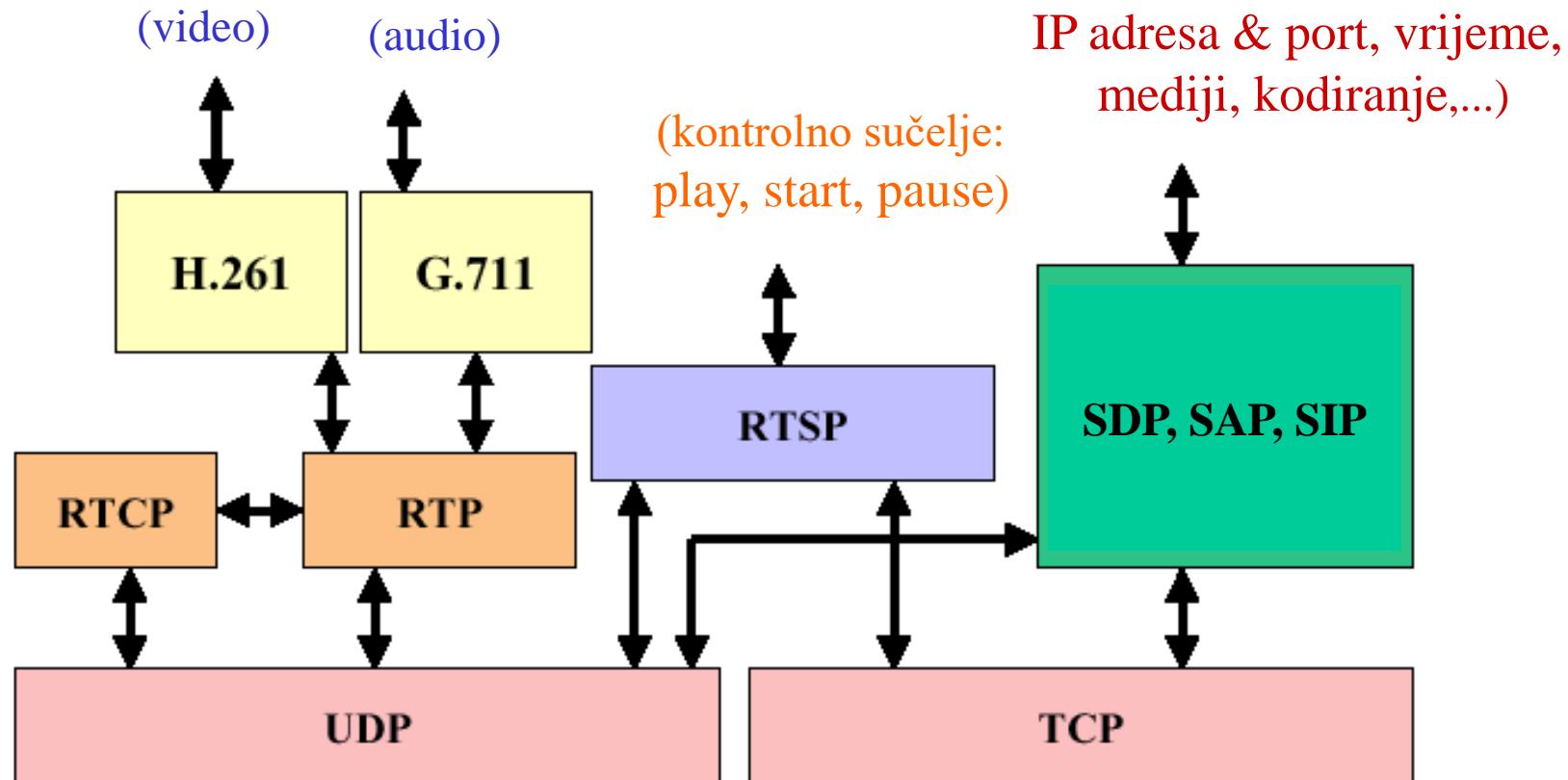
RTP – Real-time Transport Protocol  
RTCP – RTP Control Protocol  
RTSP – Real Time Streaming Protocol  
RSVP – Resource Reservation Protocol

SDP – Session Description Protocol  
TCP – Transmission Control Protocol  
UDP – User Datagram Protocol  
IP – Internet Protocol

PPP – Point-to-Point Protocol  
ATM – Asynchronous Transfer Mode  
AAL – ATM Adaptation Layer

# Protokolna arhitektura: Internet (2)

- primjer protokolne arhitekture za audio-video konferenciju i strujanje



## Oznake:

RTP – Real-time Transport Protocol  
RTCP – RTP Control Protocol  
RTSP – Real Time Streaming Protocol

SDP – Session Description Protocol  
SAP – Session Announcement Protocol  
SIP – Session Initiation Protocol

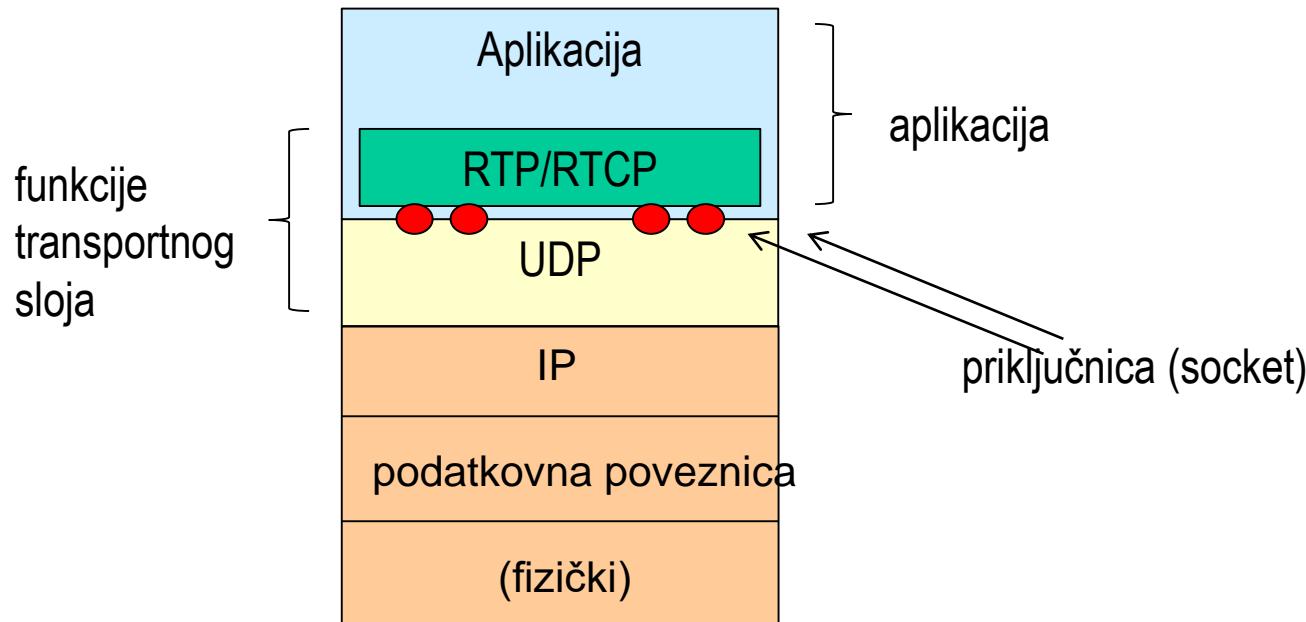
# RTP, RTCP

# Real-time Transport Protocol

- specifikacija u RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*, uključuje:
  - **Real-time Transport Protocol (RTP)**, koji pruža uslugu prijenosa podataka sa stvarno-vremenskim svojstvima (npr. audio i video) s kraja na kraj, koristeći pojedinačno (unicast) ili višeodredišno (multicast) razašiljanje na mrežnom sloju
    - RTP definira osnovni format paketa, ali ne i kontrolu
  - **RTP Control Protocol (RTCP)**, kontrolni protokol koji nadzire kvalitetu usluge i prenosi podatke o sudionicima u tekućoj sjednici

# Smještaj RTP-a u protokolnom složaju

- RTP je po smještaju u protokolnom složaju protokol aplikacijskog sloja
- RTP u nazivu ima “transportni protokol” s obzirom na svoju ulogu transporta s kraja na kraj – može se promatrati kao nadopuna transportnog sloja



# Svojstva RTP-a

- **Application Level Framing** (ALF) - koncept uokvirivanja na razini aplikacije: neki mehanizmi (npr. kontrola toka i strategija retransmisijske) koja su inače u nadležnosti transportnog protokola implementirana su na razini aplikacije
  - u duhu ALF-a definira se samo format paketa (PDU)
  - algoritmi za kontrolu toka i strategiju retransmisijsku definirani su unutar aplikacije
- RTP se oslanja na UDP (ili neki drugi transportni protokol) za multipleksiranje i zaštitnu sumu
- RTP nije pouzdan i **ne može garantirati isporuku u stvarnom vremenu** (to je uloga nižih slojeva!)

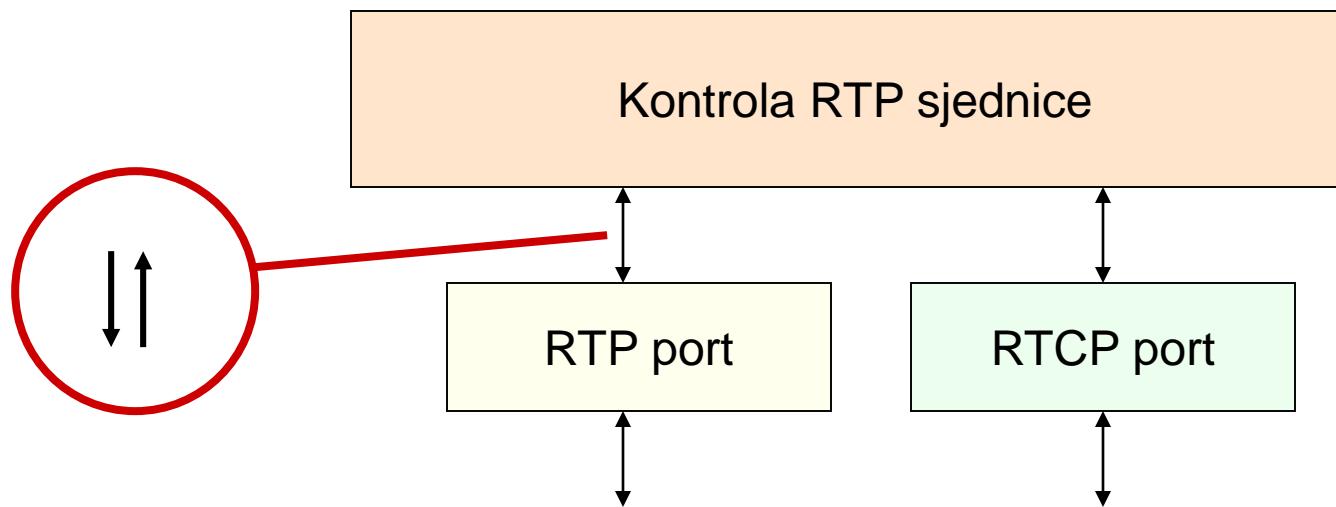
# Definicije (1)

- **RTP teret** - (višemedijski) sadržaj odn. podaci prenošeni u RTP paketu
- **RTP paket** - podatkovni paket koji se sastoji od fiksnog zaglavja, popisa doprinosećih izvora i samih podataka
- **RTCP paket** - kontrolni paket koji se sastoji od fiksnog zaglavja i struktturnih elemenata ovisnih o vrsti kontrolnog paketa
- **sinkronizirajući izvor** (synchronization source, SSRC) - izvor struje RTP paketa, s jedinstvenim identifikatorom
- **doprinoseći izvor** (contributing source, CSRC) - izvor struje RTP paketa koji je doprinio kombiniranoj struji koju generira **RTP mixer** (pojašnjeno kasnije!)

# Definicije (2)

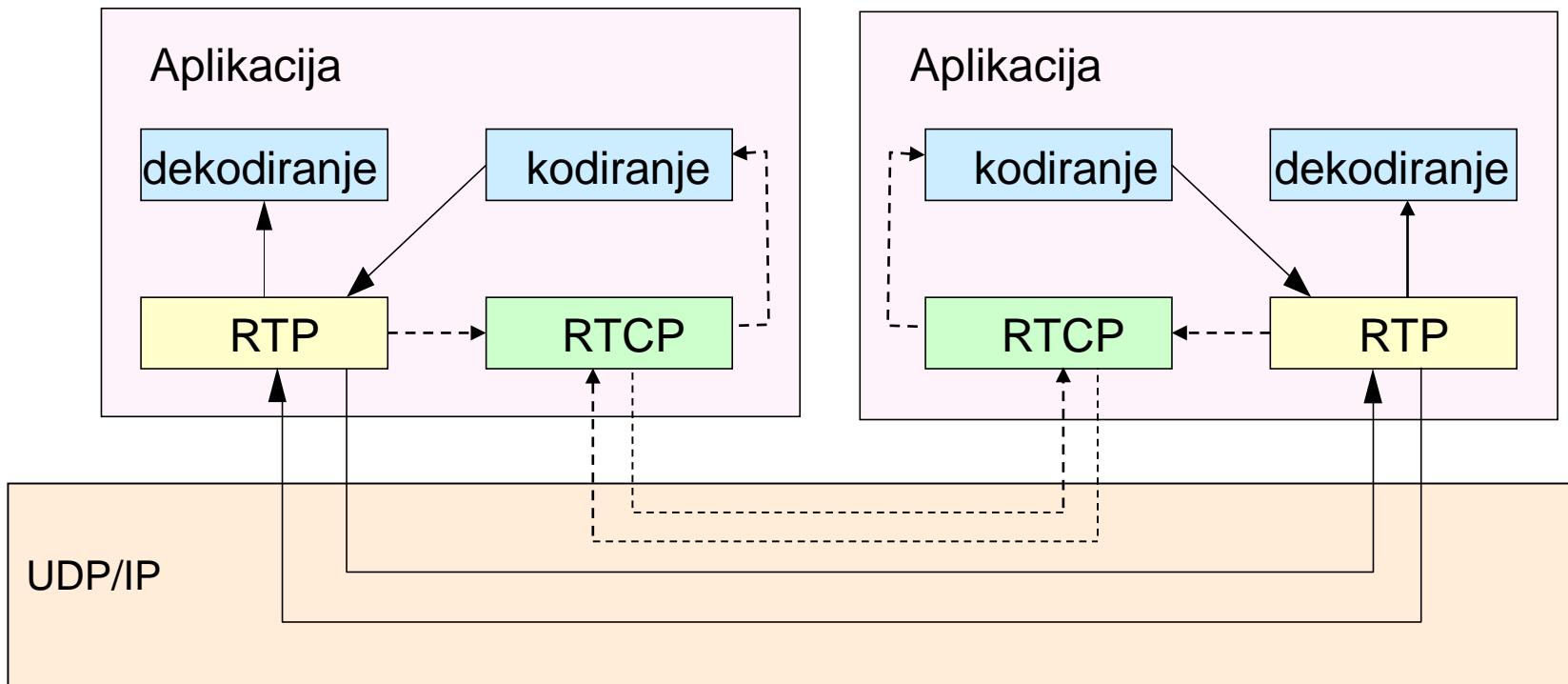
## RTP sjednica

- definira se preko dva para RTP/RTCP transportnih adresa
  - transportnu adresu čine mrežna adresa (IP adresa) i (TCP ili UDP) port
- par adresa za podatke (mrežna adresa, RTP port)
- par adresa za kontrolu (mrežna adresa, RTCP port)



# Prijenos podataka i kontrole - primjer

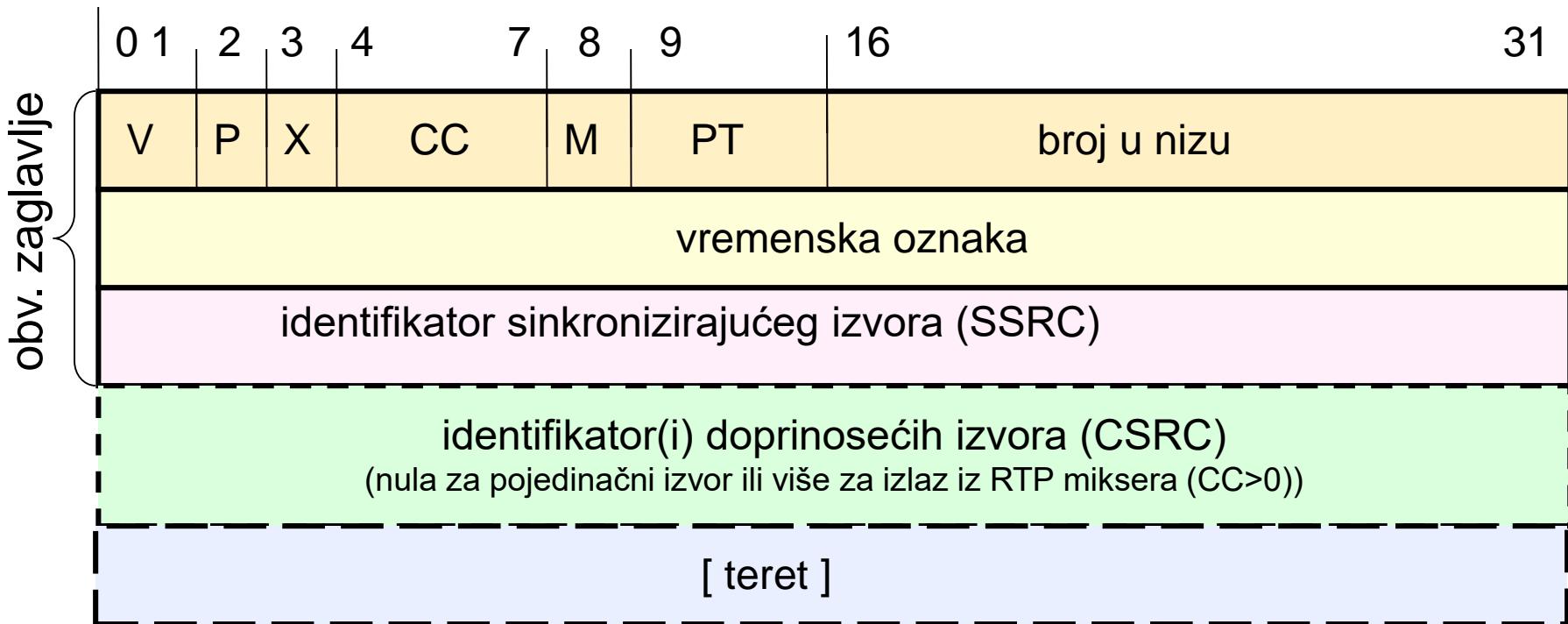
strana A



Oznake:

- solid line: podaci
- dashed line: kontrola

# Format RTP paketa



V - verzija (v=2)

P - ima/nema popunjavanje  
nulama do granice paketa

X - extension  
(ima/nema dodatnih zaglavja)

PT - vrsta tereta

CC - Contributor Count

(broj doprinosećih izvora, CRSC)

M - marker (interpretacija ovisi o profilu)

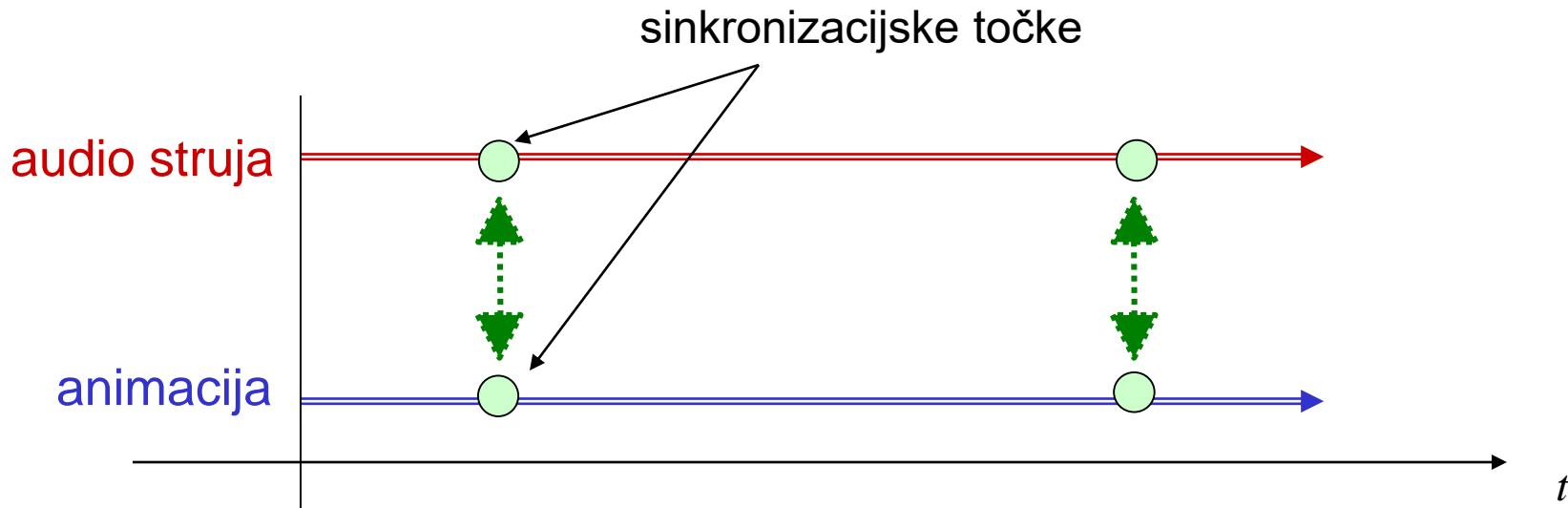
npr. za audio – početak/kraj perioda tišine,  
za video – početak okvira

# Polja u RTP zaglavlju

- **identifikacija vrste tereta** - oznaka u zaglavlju, omogućuje prenošenje različitih vrsta tereta (identifikacija putem Payload Type polja)
- **numeracija paketa** - polje “broj u nizu” definira redoslijed paketa
  - na primjer, za PCM audio (strogo periodička i regularna struktura podataka) dovoljno je znati redoslijed za ponovnu sinkronizaciju na odredištu
  - za neke medije, npr. MPEG-komprimirani video to nije dovoljno, odn. trebaju i vremenske oznake!
- **vremenska oznaka** - služi za definiranje vremenske ovisnosti
  - u polje se upisuje trenutak uzorkovanja prvog uzorka (npr. prvog audio uzorka ili video okvira)
  - vremenske oznake služe i za međusobnu sinkronizaciju različitih medija, npr. sinkronizaciju usana i audia

# Sinkronizacija RTP struja

- svi paketi/RTP struje od istog sinkronizirajućeg izvora (SSRC) (npr. izlaz iz mikrofona ili kamere) dijele istu vremensku os
- ponovna sinkronizacija na odredištu se postiže pomoću vremenskih oznaka i informacija dobivenih iz RTCP paketa



# Primjer: RTP paket (Wireshark)

**aaa.pcap - Wireshark**

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: rtp Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
624	1444.50909	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28590, Time=1240
625	1444.57904	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28591, Time=1400
626	1444.58257	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28592, Time=1560
627	1444.58824	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28593, Time=1720
628	1444.59035	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28594, Time=1880
629	1444.62516	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28595, Time=2040
630	1444.62706	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28596, Time=2200
631	1444.66468	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28597, Time=2360
632	1444.67172	192.168.1.2	212.242.33.36	RTP	PT=ITU-T G.711 PCMA, SSRC=0x3796CB71, Seq=28598, Time=2520

```
+ Frame 625: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
+ Ethernet II, Src: silicon_01:6e:bd (00:e0:ed:01:6e:bd), Dst: Castlrene_00:34:56 (00:30:54:00:34:56)
+ Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 212.242.33.36 (212.242.33.36)
- User Datagram Protocol, Src Port: 30000 (30000), Dst Port: 40392 (40392)
  Source port: 30000 (30000)
  Destination port: 40392 (40392)
  Length: 180
  Checksum: 0xbfbab [validation disabled]
- Real-Time Transport Protocol
  [Stream setup by SDP (frame 620)]
    [setup frame: 620]
    [Setup Method: SDP]
    10. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: ITU-T G.711 PCMA (8)
    Sequence number: 28591
    [Extended sequence number: 94127]
    Timestamp: 1400
    Synchronization Source identifier: 0x3796cb71 (932629361)
    Payload: 6e6815146a14141569606e6a6c6e646c6a16176c6e6d6066...
```

source: [http://wiki.wireshark.org/SampleCaptures#SIP\\_and\\_RTP](http://wiki.wireshark.org/SampleCaptures#SIP_and_RTP)

# Uloga RTCP-a

- **RTP Control Protocol (RTCP)**
- RTCP omogućuje nadzor kvalitete usluge i dijagnostiku
- pošiljatelj može pružiti više podataka o izvoru RTP prometa i samom prometu
- primatelji šalju izvješća o kvaliteti prijema natrag pošiljatelju
- svaki sudionik u sjednici periodički šalje RTCP kontrolne pakete svim ostalim sudionicima u sjednici, koristeći isti način slanja kao kod slanja podataka

# Vrste RTCP paketa

- izvješće primatelja
  - šalju ga svi primatelji, sadrži povratnu informaciju o kvaliteti prijema RTP paketa za svaki sinkronizirajući izvor
- izvješće pošiljatelja
  - šalju ga aktivni pošiljatelji
- opis izvora
  - kanonsko ime (CNAME) krajnjeg sustava, ime korisnika, e-mail, telefon, lokacija, naziv alata,..
- odlazak (bye)
  - označuje kraj sudjelovanja
- nestandardni
  - definirani unutar pojedinačne aplikacije

- U **izvješću primatelja** se za svaki sinkronizirajući izvor (SSRC) navodi statistika o podacima primljenim od prethodnog izvješća
  - neki parametri: % izgubljenih paketa, kumulativni broj izgubljenih paketa, kolebanje međudolaznog vremena, ...
- U **izvješću pošiljatelja** uz izvješće primatelja (kao gore) nalaze se i podaci o poslanim paketima (broj poslanih paketa) i Network Time Protocol vremenska oznaka koja omogućuje računanje Round Trip Time (RTT), odn. kašnjenja

# Primjer: RTCP paketi (Wireshark)

Sending\_Video\_and\_Audio\_to\_an\_EyeBeam\_Soft\_Phone.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: rtcp Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
4	-161846901	192.168.1.36	192.168.1.87	RTCP	Receiver Report Source description
126	908188888	192.168.1.87	192.168.1.36	RTCP	Sender Report Source description
166	-930792131	192.168.1.87	192.168.1.36	RTCP	Sender Report Source description
167	-904407704	192.168.1.36	192.168.1.87	RTCP	Sender Report Source description
168	-903818168	192.168.1.36	192.168.1.87	RTCP	Sender Report Source description
333	-178811330	192.168.1.36	192.168.1.87	RTCP	Receiver Report Source description
334	-178441570	192.168.1.36	192.168.1.87	RTCP	Receiver Report Source description
346	154514789	192.168.1.36	192.168.1.87	RTCP	Extended report (RFC 3611)
460	1996835972	192.168.1.87	192.168.1.36	RTCP	Sender Report Source description
477	-135644262	192.168.1.87	192.168.1.36	RTCP	Sender Report Source description
515	558369695	192.168.1.36	192.168.1.87	RTCP	Receiver Report Source description

Frame 515: 198 bytes on wire (1584 bits), 198 bytes captured (1584 bits)  
Ethernet II, Src: Micro-st\_15:44:8c (00:11:09:15:44:8c), Dst: Netgear\_c9:5a:fc (00:14:6c:c9:5a:fc)  
Internet Protocol, Src: 192.168.1.36 (192.168.1.36), Dst: 192.168.1.87 (192.168.1.87)  
User Datagram Protocol, Src Port: 50301 (50301), Dst Port: 5063 (5063)  
Real-time Transport Control Protocol (Receiver Report)  
[Stream setup by SDP (frame 1)]  
10. .... = Version: RFC 1889 version (2)  
..0. .... = Padding: False  
...0 0001 = Reception report count: 1  
Packet type: Receiver Report (201)  
Length: 7 (32 bytes)  
Sender SSRC: 0xb51468b (189875851)  
Source 1  
Identifier: 0x30d76cc2 (819424450)  
SSRC contents  
Fraction lost: 0 / 256  
Cumulative number of packets lost: 0  
Extended highest sequence number received: 45144  
Sequence number cycles count: 0  
Highest sequence number received: 45144  
Interarrival jitter: 224  
Last SR timestamp: 2225591991 (0x84a7d6b7)  
Delay since last SR timestamp: 40828 (622 milliseconds)  
Real-time Transport Control Protocol (source description)  
[Stream setup by SDP (frame 1)]  
10. .... = Version: RFC 1889 version (2)  
..0. .... = Padding: False  
...0 0001 = Source count: 1  
Packet type: Source description (202)  
Length: 30 (124 bytes)  
Chunk 1, SSRC/CSRC 0xb51468b  
Identifier: 0xb51468b (189875851)  
SDES items  
Type: CNAME (user and domain) (1)  
Length: 61  
Text: 0B390D3BE7CE4DF997E9B430EA847E00@unique.zC36F0F555B824185.org  
Type: PRIV (private extensions) (8)  
Length: 49  
Prefix length: 16  
Prefix string: x-rtp-session-id  
Text: 53B446587A59483B8F5A13E7CCA9C068

Source: [http://techtraces.com/sample\\_captures/](http://techtraces.com/sample_captures/)

# RTP profil za audio i video

- RTP se prilagođuje aplikaciji pomoću **profila** i specifikacije **formata vrste tereta**
- RFC 3551, RTP Audio/Video Profile definira *profil* za uporabu RTP-a i RTCP-a u **višekorisničkim audio i video konferencijama s minimalnom kontrolom**
  - interpretacija generičkih polja u RTP specifikaciji u pogodnih za audio i video konferenciju
  - veza pojedinih standardnih načina kodiranja i oznaka vrsta tereta (*payload type*) koje nosi RTP (formati tereta specificirani su odvojeno!)

# Vrste RTP tereta i kodiranja

- primjeri definiranih vrsta tereta za audio:  
0: PCMU, PCM  $\mu$ -zakon; MIME audio/basic (64 kbit/s)  
3: GSM (13 kbit/s)  
7: LPC (2,4 kbit/s)  
14: MPEG audio (varijabilno)
- primjeri definiranih vrsta tereta za video:  
26: Motion JPEG  
31: H.261  
32: MPV (MPEG-I i MPEG-II video kodiranje)
- otvorenost za nove vrste tereta!  
• broj za novu vrstu tereta se registrira s IANA-om  
[<http://www.iana.org/assignments/rtp-parameters>]

# Definirani formati RTP tereta (samo za ilustraciju!)

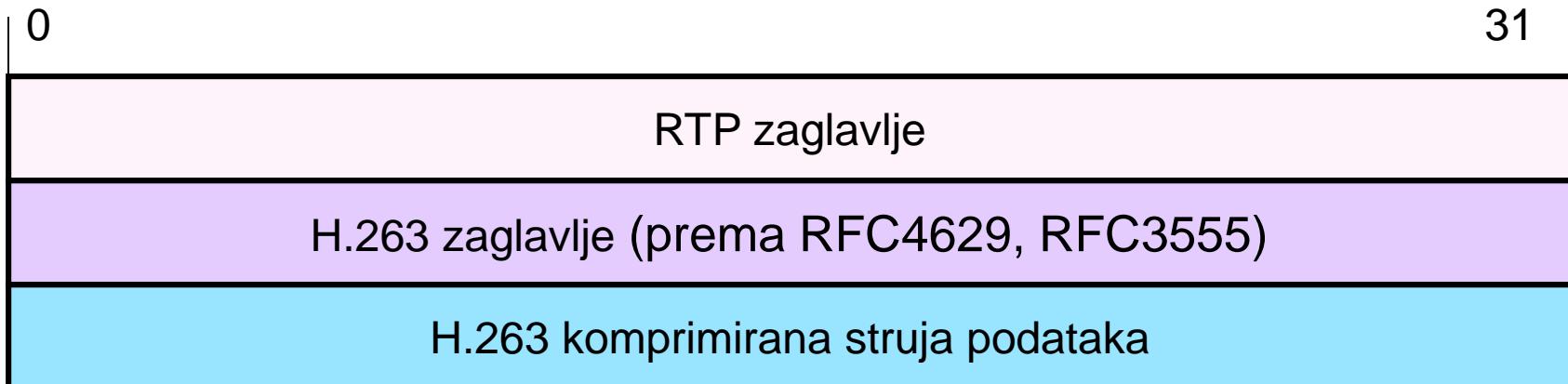
- Samo nekoliko primjera za svaki medij
- Trenutno definirano 130 vrsta tereta [4. 6.2 018.]

<b>Media Type</b>	<b>Subtype</b>	<b>Reference</b>
application	raptorfec	<a href="#">[RFC6682]</a>
application	rtx	<a href="#">[RFC4588]</a>
application	smpte336m	<a href="#">[RFC6597]</a>
application	ulpfec	<a href="#">[RFC5109]</a>
audio	VDVI	<a href="#">[RFC3551]</a> <a href="#">[RFC4856]</a>
audio	VMR-WB	<a href="#">[RFC4348]</a> <a href="#">[RFC4424]</a>
audio	vorbis	<a href="#">[RFC5215]</a>
audio	vorbis-config	<a href="#">[RFC5215]</a>
text	red	<a href="#">[RFC4102]</a>
text	rtx	<a href="#">[RFC4588]</a>
text	t140	<a href="#">[RFC4103]</a>
text	ulpfec	<a href="#">[RFC5109]</a>
video	rtx	<a href="#">[RFC4588]</a>
video	SMPTE292M	<a href="#">[RFC3497]</a>
video	ulpfec	<a href="#">[RFC5109]</a>
video	vc1	<a href="#">[RFC4425]</a>

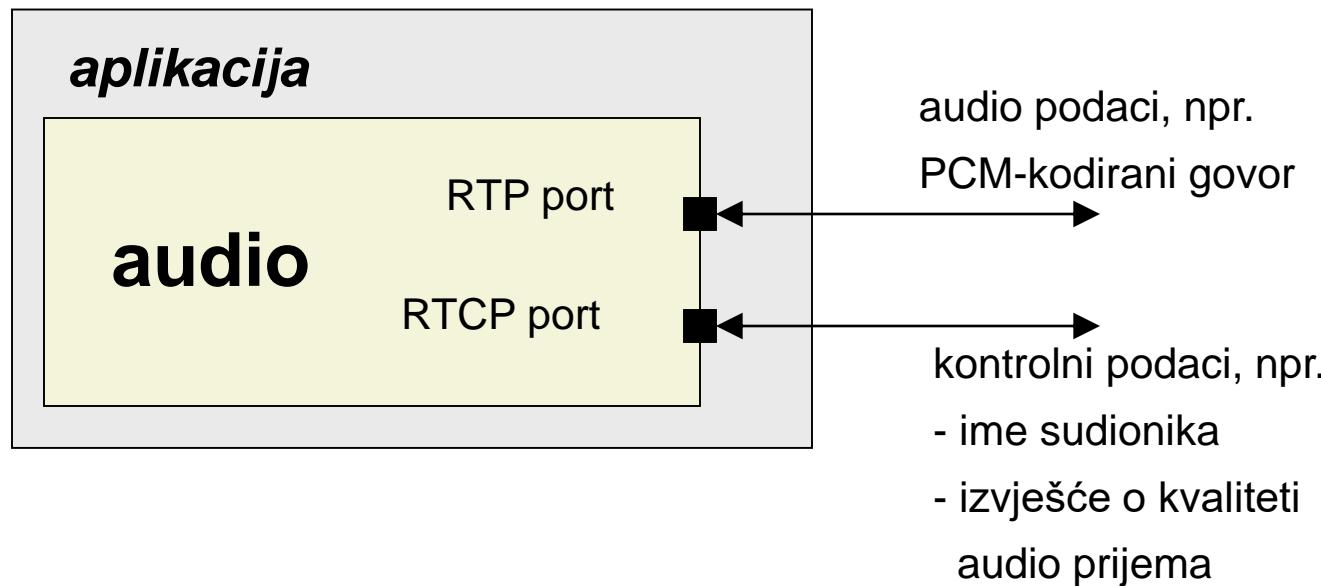
# Primjer: H.263 video preko RTP-a

Promatramo prenošenje struje video podataka kodiranih prema ITU-T H.263 standardu preko RTP-a.

- izlaz iz kodera se izravno paketizira; RTP paket = dio struje podataka
- RTP paket započinje fiksnim RTP zaglavljem
- koriste se samo tri polja zaglavlja: vrsta tereta (PT=34), vremenska oznaka i marker bit (M=1 za početak/kraj video okvira, kriške, GOB)
- slijedi zaglavljje H.263 tereta prema pripadajućoj specifikaciji tereta i nakon toga određeni broj okteta iz H.263 komprimirane struje podataka

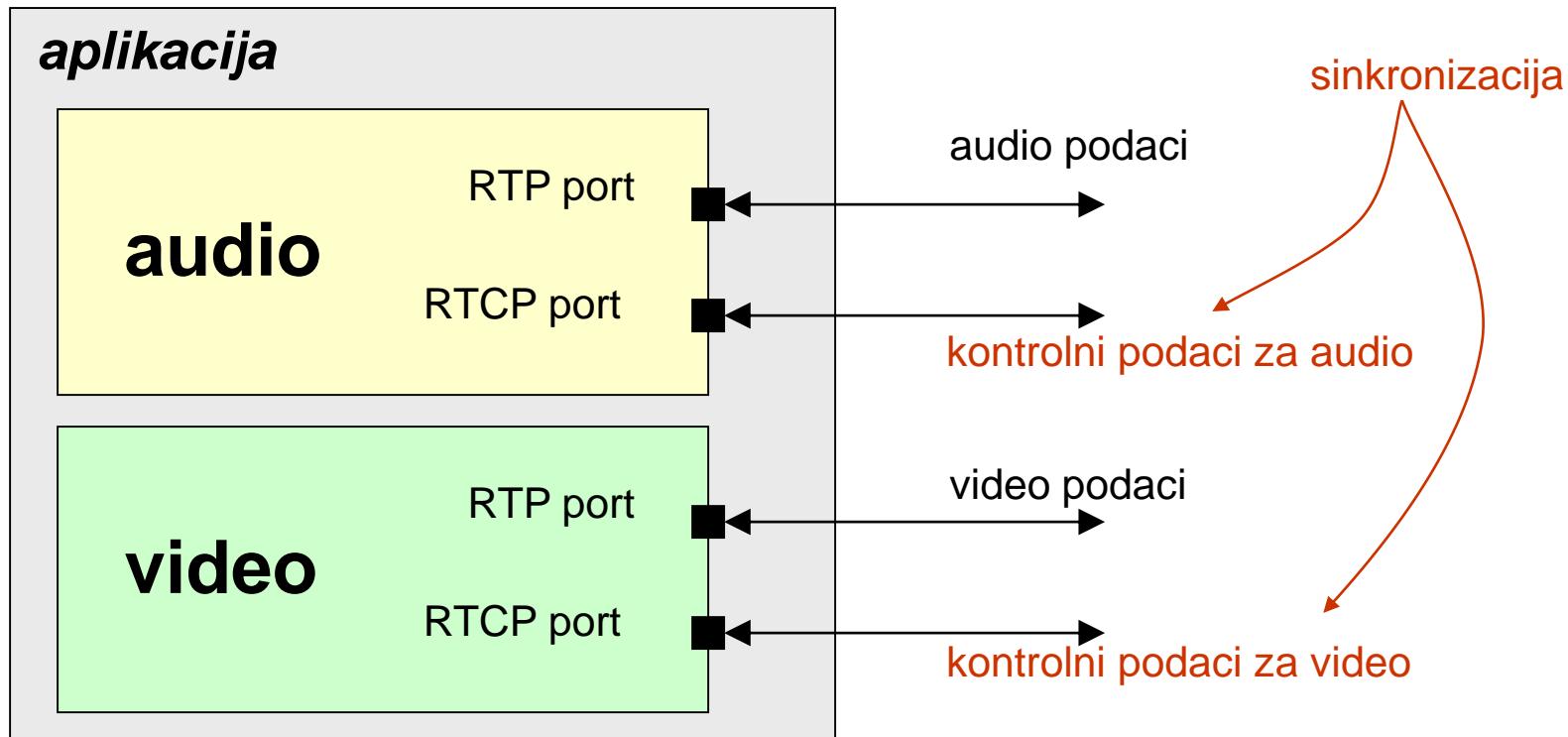


- ne postoji “well-known port”
- podaci o RTP sjednici (mrežna adresa i par portova) saznaju se na drugi način (npr. putem opisa sjednice u formatu SDP)

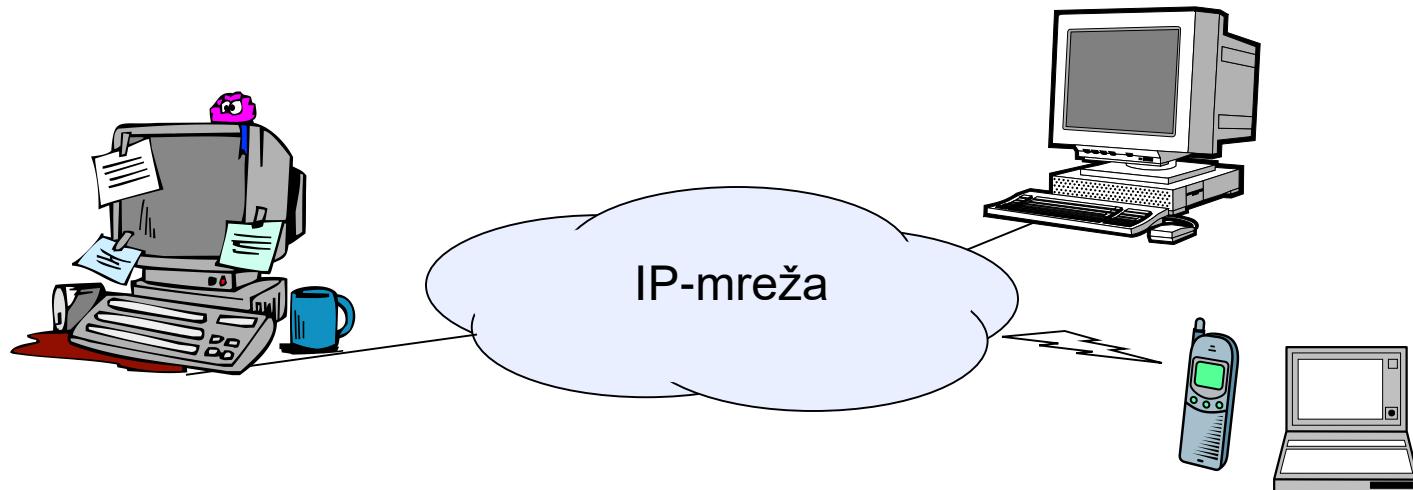


# Audio/video scenarij

- u višemedijskoj aplikaciji za svaki medij definira se posebna RTP sjednica



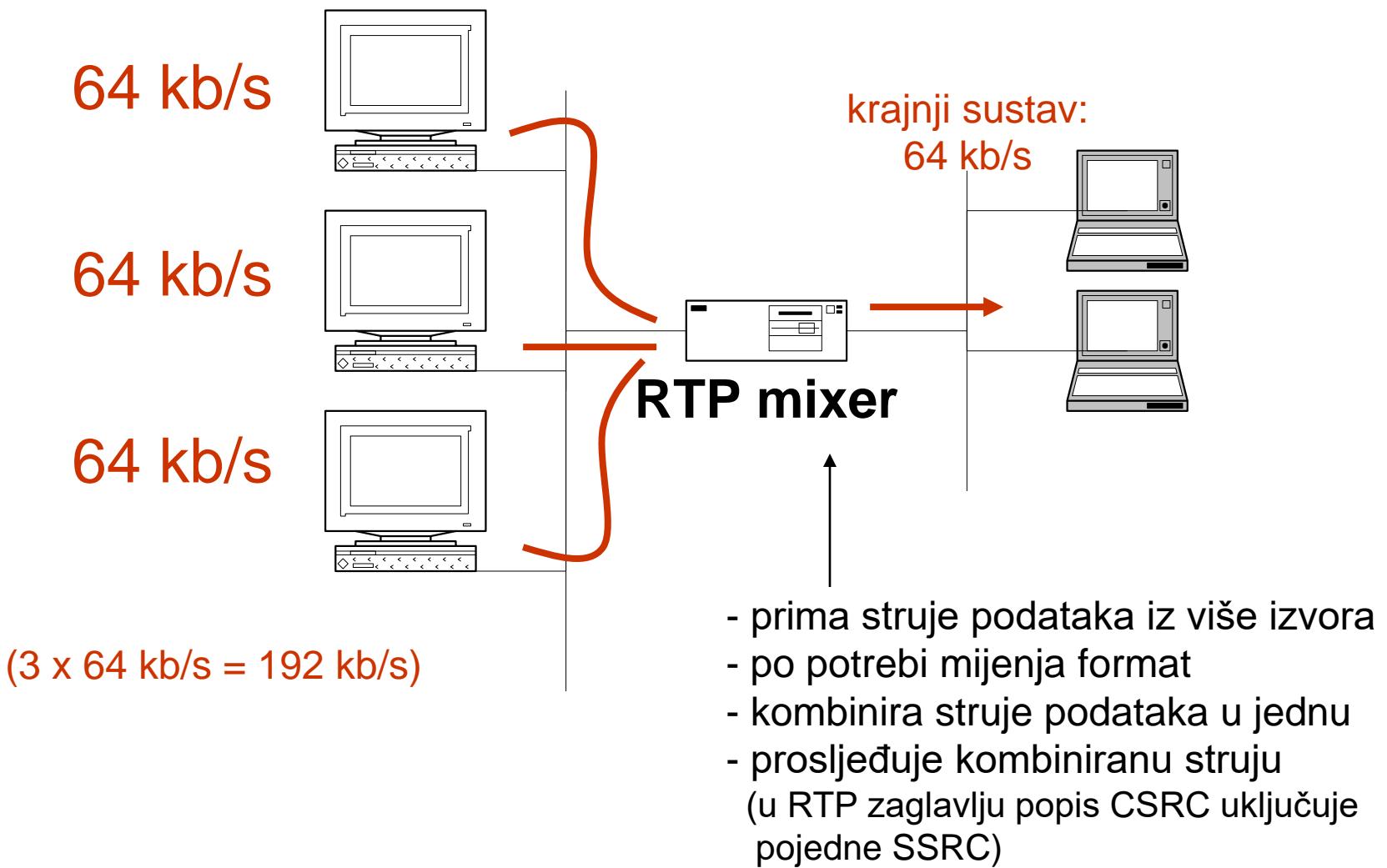
- zahvaljujući odijeljenim RTP sjednicama, svaki korisnik pojedinačno može birati koje medije (npr. audio-video, jedan ili više) želi primati



- mogući problemi u višekorisničkom pristupu:
  - svi korisnici ne moraju htjeti primati isti **format medija**
  - mogu postojati razlike u pogledu **pristupne mreže**
  - mogu postojati razlike u pogledu **krajnjeg sustava (terminala)**
- za prilagodbu se koriste RTP mixer i RTP translator

# Primjer 1: Ograničenja u mreži

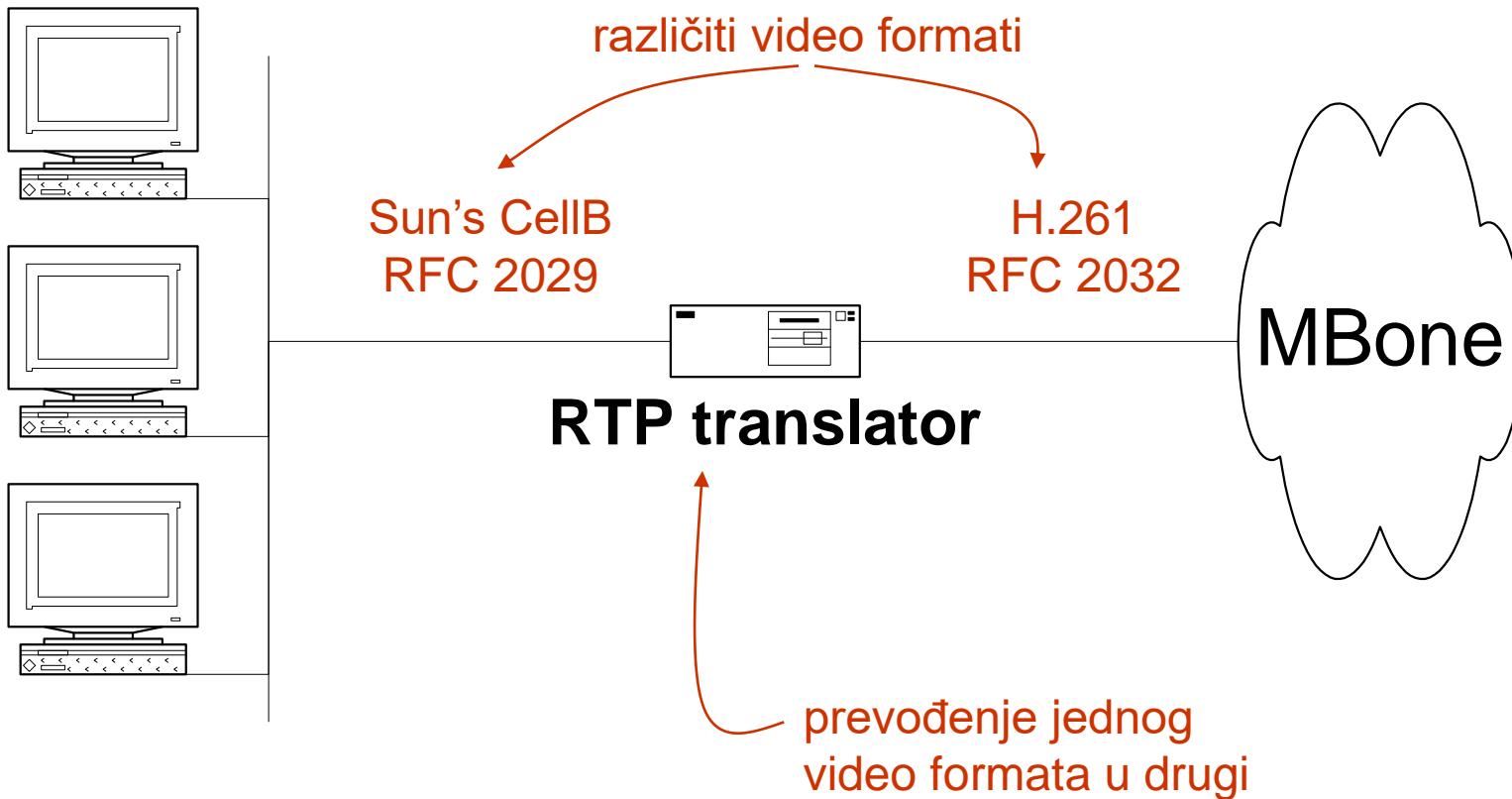
- promatramo slučaj kada je glavnina sudionika je mreži velike brzine, a neki sudionici su u dijelu mreže sa sporijom vezom
  - loše rješenje: svi sudionici koriste audio smanjene pojasne širine, tj. lošije kvalitete
  - bolje rješenje: prema sporijem dijelu mreže stavlja se **RTP mixer** (RTP prijenosnik), koji rekonstruira struje pojedinih audio izvora, resinkronizira ih i kombinira u jednu struju pogodniju za sporiju vezu
  - izlazna struja iz RTP mixera kodira se kao da je sinkronizirajući izvor mixer, a u zaglavljima su navedene doprinoće struje
- RTP mixer je pogodan samo za audio!



## Primjer 2: Ograničenja u mreži

- promatramo slučaj kada su svi sudionici u brzima mrežama, no koriste različite formate
- ovdje nije potrebno kombiniranje pojedinih struja u jednu jer je propusnost mreže dovoljna
- problem prilagodbe formata se rješava primjenom **RTP translatora**
- RTP translator vrši prekodiranje iz jednog formata u drugi, uz netaknutu oznaku sinkronizirajućeg izvora
- ostale primjene: propuštanje kroz vatrozid, šifriranje, unicast-multicast i sl.

# RTP translator



# **SDP, SAP**

# Protokoli za opis i objavu sjednice

- protokol za opis sjednice
  - propisani skup parametara koji služi za opis sjednice
  - standardni format za opis medija koji sudjeluju, podaci o protokolima i formatima koji će se koristiti u sjednici i sl.
  - Session Description Protocol, **SDP**
- protokol za objavu sjednice
  - služi za objavu sjednice svim zainteresiranim sudionicima (korištenje višeodredišnog razašiljanja preko poznate adrese)
  - Session Announcement Protocol, **SAP**

# SDP: Session Description Protocol

- specifikacija u IETF RFC 4566 “*SDP: Session Description Protocol*”
- SDP definira **format** za opis sjednice
- format je neovisan o transportu
  - npr. koriste se Session Announcement Protocol (SAP), Session Initiation Protocol (SIP), e-mail + MIME dodaci, Real Time Streaming Protocol (RTSP) ili HTTP
- opis sjednice dan SDP-om je kratak, strukturiran (parovi atributa i vrijednosti) i u obliku čistog teksta

# Opis sjednice putem SDP-a

- podaci koji se navode su:
  - naziv i svrha sjednice
  - vrijeme održavanja
  - podaci o primanju medija (protokoli, codeci, transportni parametri)
  - (izborne) dodatni podaci
- navedeni podaci moraju biti dostatni za donošenje odluke o eventualnom sudjelovanju u sjednici i za pokretanje pomagala (preglednika, aplikacija, i sl.) za sudjelovanje

# Format SDP-a

- Opis sjednice sadrži niz redaka teksta oblika:

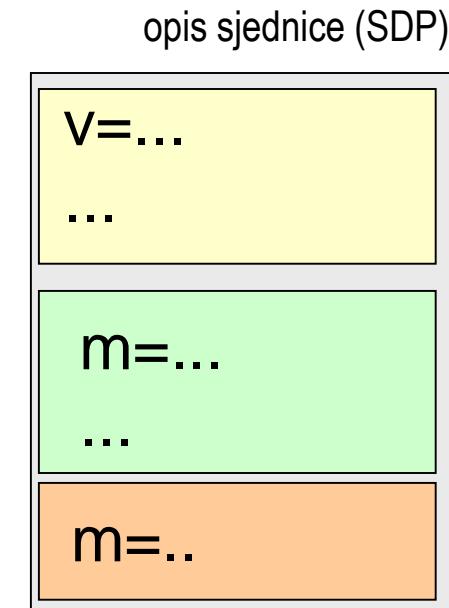
**<vrsta> = <vrijednost>**

gdje su:

<b>&lt;vrsta&gt;</b>	jedan znak koji označava atribut
<b>&lt;vrijednost&gt;</b>	niz znakova ( <i>string</i> ) čiji format ovisi o <vrsti>

- Opis sjednice sadrži, redom:

- parametre sjednice (od v= .. do prvog m=..)
  - vrijede za cijelu sjednicu
  - vrijede za sve struje medija, osim ako je parametar medija izričito definiran drugačije
- parametre medija (blok m=... )
  - po jedna skupina za pojedinačnu struju medija



# SDP: parametri sjednice

<u>znak</u>	<u>opis</u>	
v	<b>verzija protokola</b>	
o	<b>vlasnik/pokretač sjednice i identifikator sjednice</b>	
s	<b>naziv sjednice</b>	
i	podaci o sjednici (*)	
u	Universal Resource Identifier (URI) s opisom (*)	
e	e-mail adresa osobe za kontakt (*)	
p	telefonski broj osobe za kontakt (*)	
c	podaci o vezi (*)	
b	potrebna širina pojasa [kbit/s] (*)	
t	vrijeme aktivnosti (održavanja) sjednice	
r	vrijeme ponavljanja (*)	
z	usklađivanje vremenske zone (*)	
k	ključ šifriranja (*)	
a	atributi sjednice (*) [ može ih biti nula ili više ]	označeni (*) parametri nisu obavezni

# SDP: parametri medija

- za svaki medij ili struju medija zadaje se (pod)skup parametara medija

znak      opis

m	<b>naziv medija i transportna adresa</b>		- IP adresa (jedno- ili višeodredišna) - TCP ili UDP port
i	naslov medija (*)		
c	podaci o vezi (*)		
b	bandwidth (kb/s) (*)		
k	ključ šifriranja (*)		
a	atributi medija (*) [ može ih biti nula ili više ]		

označeni (\*) parametri  
nisu obavezni

# Primjer opisa sjednice SDP-om (1)

v=0                      vlasnikovo korisničko ime  
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5              identifikator sjednice (NTP)              verzija (NTP)  
s=SDP Seminar              naziv sjednice  
i=A Seminar on the session description protocol  
u=http://www.example.com/seminars/sdp.pdf  
e=j.doe@example.com (Jane Doe)  
c=IN IP4 224.2.17.12/127              Internet; IPv4; višeodredišna adresa/TTL  
t=2873397496 2873404696              vrijeme početka i kraja sjednice (NTP)  
a=recvonly              atribut sjednice; svi mediji “receive only”  
m=audio 49170 RTP/AVP 0              1. medij: PCM audio, RTP port 49170  
m=video 51372 RTP/AVP 99              2. medij: H.263 video, RTP port 51372  
a=rtpmap:99 h263-1998/90000

# Primjer opisa sjednice SDP-om (2)

v=0      vlasnikovo korisničko ime      identifikator sjednice (NTP)      verzija (NTP)      IN=Internet; IPv4; IP adresa izvora

**o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4**

**s=SDP Seminar**      naziv sjednice

**i=A Seminar on the session description protocol**

**u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps**

**e=mjh@isi.edu (Mark Handley)**

**c=IN IP4 224.2.17.12/127**      Internet; IPv4; višeodredišna adresa/TTL

**t=2873397496 2873404696**      vrijeme početka i kraja sjednice (NTP)

**a=recvonly**      atribut sjednice; svi mediji “receive only”

**m=audio 49170 RTP/AVP 0**      1. medij: PCM audio, RTP port 49170

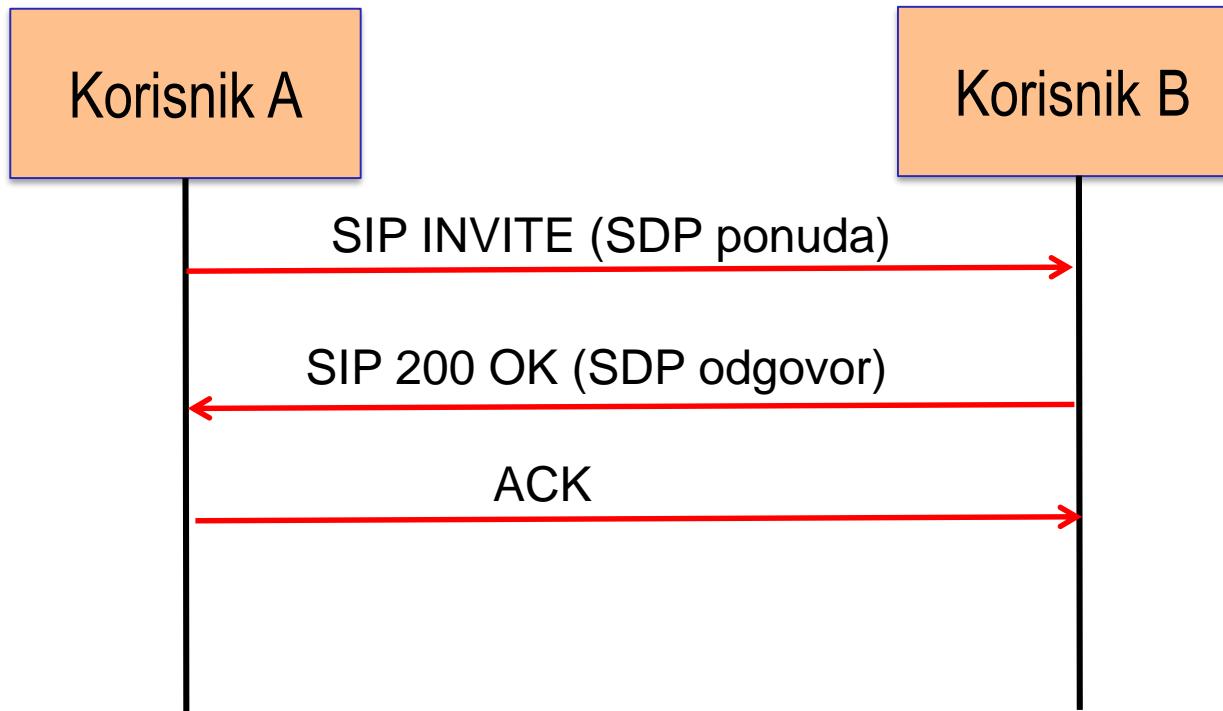
**m=video 51372 RTP/AVP 31**      2. medij: H.261 video, RTP port 51372

**m=application 32416 udp wb**      3. medij: whiteboard (aplikacija za zajedničko crtanje), UDP port 32416  
atribut: orijentacija papira (uspravno)

**a=orient:portrait**

# Razmjena mogućnosti (1/2)

- Razmjena višemedijskih mogućnosti između uređaja provodi se korištenjem formata SDP i protokola SIP po modelu *ponuda – odgovor* (OFFER/ANSWER) (RFC 3264)



# Razmjena mogućnosti (2/2)

zahtjev (INVITE)	odgovor (200 OK)
<b>v=0</b> <b>o=</b> <b>s=</b> <b>c=IN IP4 128.2.3.1</b> <b>t=</b> <b>m=video 4004 RTP/AVP 14 26</b> <b>a=rtpmap:14 MPA/90000</b> <b>a=rtpmap:26 JBEG/90000</b> <b>m=audio 5004 RTP/AVP 0 4</b> <b>a=rtpmap:0 PCMU/8000</b> <b>a=rtpmap:4 GSM/8000</b>	<b>v=0</b> <b>o=</b> <b>s=</b> <b>c=IN IP4 16.2.3.1</b> <b>t=</b> <b>m=video 0 RTP/AVP 14</b> <b>m=audio 6004 RTP/AVP 0</b> <b>a=rtpmap:0 PCMU/8000</b>

# Ograničenja SDP-a

- izvorno nije namijenjen za pregovaranje o sadržaju
  - ne zna opisati alternative (npr. “English or French”)
  - ne može ograničiti kodeke prema ograničenjima propusnosti i mogućnosti obrade; moguće su samo zadane kombinacije, npr. samo H.261 video + PCM audio po  $\mu$ -zakonu ili H.263 video + G.729 audio
  - ne omogućuje označavanje manje odn. više poželjnih mogućnosti
- nije dovoljno bogat za opis višemedijskih prezentacija
- objava vremena održavanja (NTP timestamp) nije u skladu s drugim standardima za definiranje i usklađivanje vremenskog rasporeda, kao npr. *vcalendar*

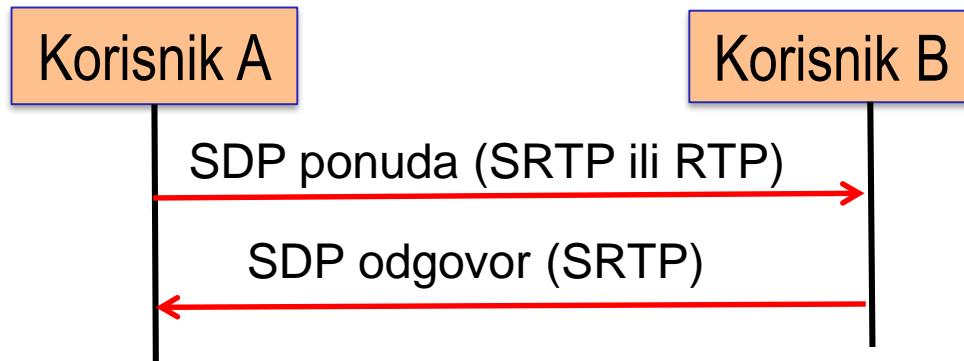
*neka ograničenja (djelomično) riješena novim RFC-ovima*



- RFC 5939 (“SDP Capability Negotiation”): proširuje osnovni model *ponuda-odgovor*
  - mogućnost pregovaranja alternativnih konfiguracija sjednice (npr. alternativnih RTP profila) i odgovarajućih parametara (SDP atributa)
  - Krajnje točke specificiraju:
    - podržane mogućnosti (npr. RTP/SAVP profil, PCMU kodek, itd.)
    - potencijalne konfiguracije (kombinacije podržanih mogućnosti) koje se mogu koristiti za sjednicu
    - aktualne konfiguracije sjednice i medijskih komponenata
  - Definira se **process pregovaranja**: uz aktualne konfiguracije, moguće je pregovarati i oko potencijalnih konfiguracija
  - proširenja u RFC 6971 (“Session Description Protocol (SDP) Media Capabilities Negotiation”)

# Podrška za pregovaranje o mogućnostima (2/2)

- Primjer: uz aktualnu konfiguraciju (RTP/AVP), nudi se alternativna potencijalna konfiguracija (Secure RTP: profil RTP/SAVP)



## SDP ponuda

```
v=0
o=- 25678 753849 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1 t=0 0
m=audio 53456 RTP/AVP 0 18
a=tcap:1 RTP/SAVP
a=acap:1 crypto:1
    AES_CM_128_HMAC_SHA1_80
    inline:WVNfX19zZW1jdGwgKCkgewky
    MjA7fQp9CnVubGVz|2^20|1:4
a=pcfg:1 t=1 a=1
```

## SDP odgovor

```
v=0
o=- 24351 621814 IN IP4 192.0.2.2
s=
c=IN IP4 192.0.2.2 t=0 0
m=audio 54568 RTP/SAVP 0 18
a=crypto:1
    AES_CM_128_HMAC_SHA1_80
    inline:PS1uQCveeCFCAnVmckj
    pPywJNWhcYD0mXXtxaVBR|2^
    20|1:4
a=acfq:1 t=1 a=1
```

# SDP podrška za grupiranje

- RFC 5888: omogućuje specificiranje odnosa između raznih medija u SDP-u
  - ➡ primjer: sinkornizacija usana i zvuka

```
v=0
o=Laura 289083124 289083124 IN IP4
one.example.com
c=IN IP4 192.0.2.1
t=0 0
a=group:LS 1 2
m=audio 30000 RTP/AVP 0
a=mid:1
m=video 30002 RTP/AVP 31
a=mid:2
```

“*a=group:*” grupira medijske tokove (referencirane putem ID-a) na osnovu grupne semantike (npr. LS: lip synchronization)

“*a=mid:*” media stream ID

# SAP: Session Announcement Protocol

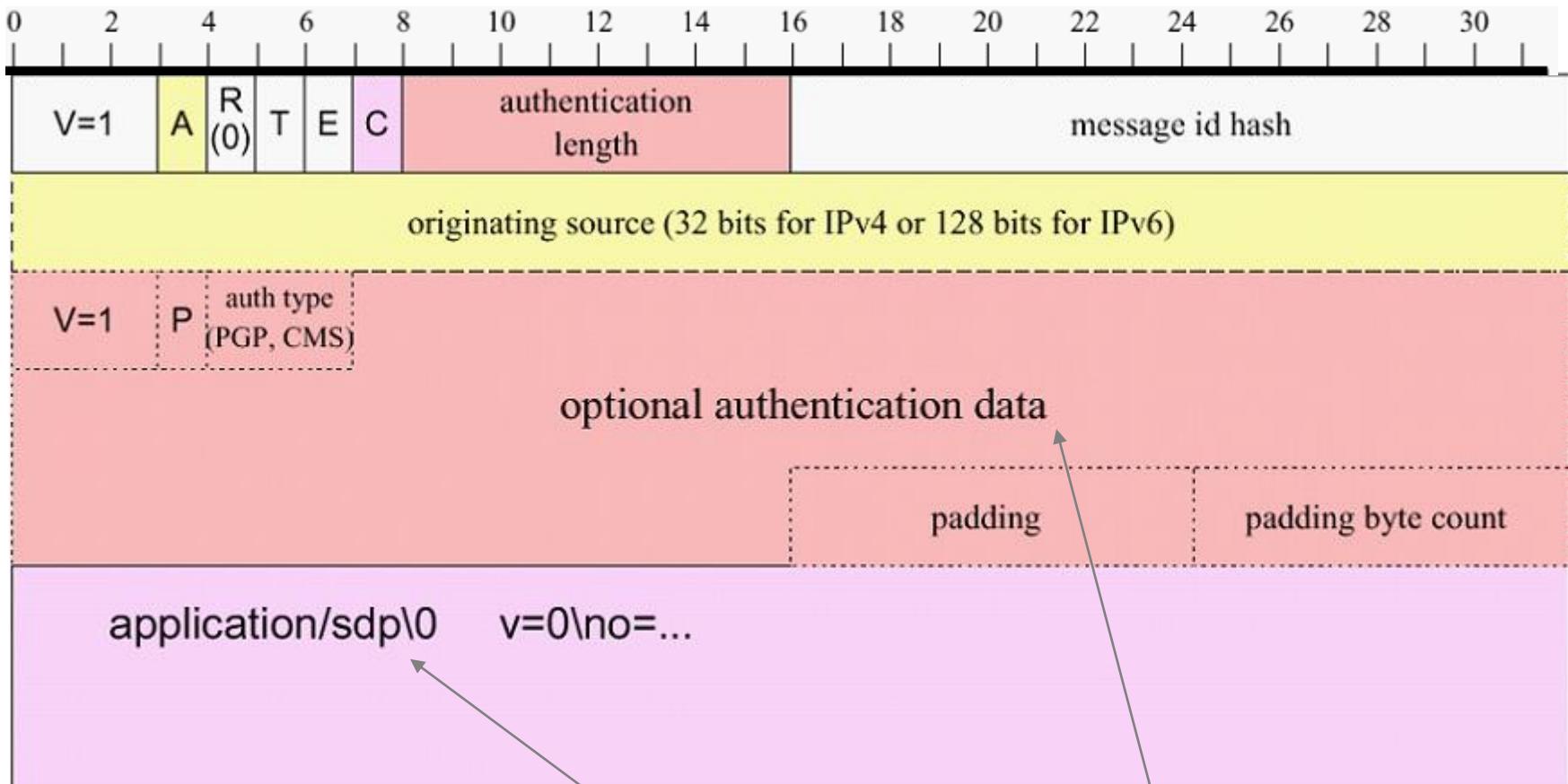


Zavod za telekomunikacije

- specifikacija u RFC 2974 “Session Announcement Protocol”, 10/2000
- služi za objavu višeodredišnih sjednica (standardni format objave koristi SDP za opis sjednice!)
- SAP periodički razašilje objavu na UDP port 9875, višeodredišnu adresu sap.mcast.net (224.2.127.254) ili administrativno određenu višeodredišnu adresu
- zainteresirani sudionici osluškuju objave i po želji se priključuju sjednici
- primjena: MBone

- ukupna širina pojasa za SAP objave je 4 kb/s, i to za sve objave (postiže se prilagodbom vremenskog razmaka među objavama, obično reda veličine nekoliko minuta)
- potencijalni problem: prilagodba veličini (engl. *scalability*) – problem nije u broju sudionika, već u broju objava sjednica!!
- SAP je pogodan za javne sjednice kod kojih se sudionici ne znaju unaprijed (inače bi ih bilo bolje pozvati SIP-om!)

# Format SAP paketa



# Primjer: SAP objava sjednice

```
+ User Datagram Protocol, Src Port: 1095 (1095), Dst Port: 9875 (9875)
  Session Announcement Protocol
    Flags: 0x20
    Authentication Length: 0
    Message Identifier Hash: 0x0
    originating Source: 161.53.19.124
  Session Description Protocol
    session Description Protocol version (v): 0
    Owner/Creator, Session Id (o): mmatijasevic 3248680638 3248680807 IN IP4 escher.zavod.tel.fer.hr
      session Name (s): Probna sjednica
      Session Information (i): ovo je probna sjednica za labos iz VK.
      URI of Description (u): http://www.tel.fer.hr
      E-mail Address (e): <smaja.matijasevic@fer.hr>
      Phone Number (p):
    Time Description, active time (t): 3248679600 3248686800
    Session Attribute (a): tool:sdr v3.0
    Session Attribute (a): type:test
    Media Description, name and address (m): audio 24930 RTP/AVP 0
    Connection Information (c): IN IP4 239.255.227.21/15
    Media Attribute (a): ptim:40
    Media Description, name and address (m): video 56422 RTP/AVP 31
    Connection Information (c): IN IP4 239.255.7.125/15
```

**SAP odredišni port: 9875**

SAP

SDP

# Real Time Streaming Protocol

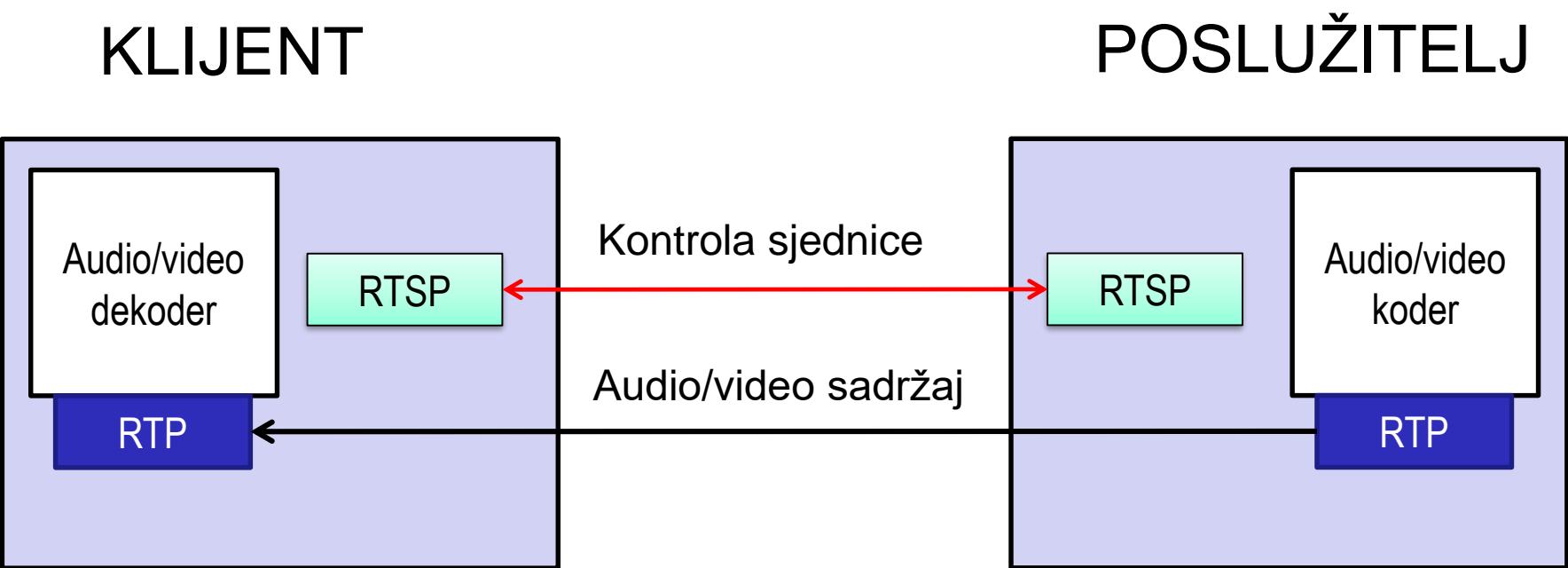
# Real Time Streaming Protocol

- specifikacija u RFC 2326, “Real Time Streaming Protocol (RTSP)”
- aplikacijski protokol za upravljanje dostavom podataka sa stvarno-vremenskim svojstvima
- izvori podataka: prijenos uživo ili već snimljeni podaci
- referenciranje podataka putem URL-a (rtsp:// ...)
- neovisan o transportnom protokolu; mogući izbori su npr. RTP, UDP/IP, TCP/IP, UDP/IP-multicast
- može se koristiti i za pojedinačne korisnike i za velike multicast grupe

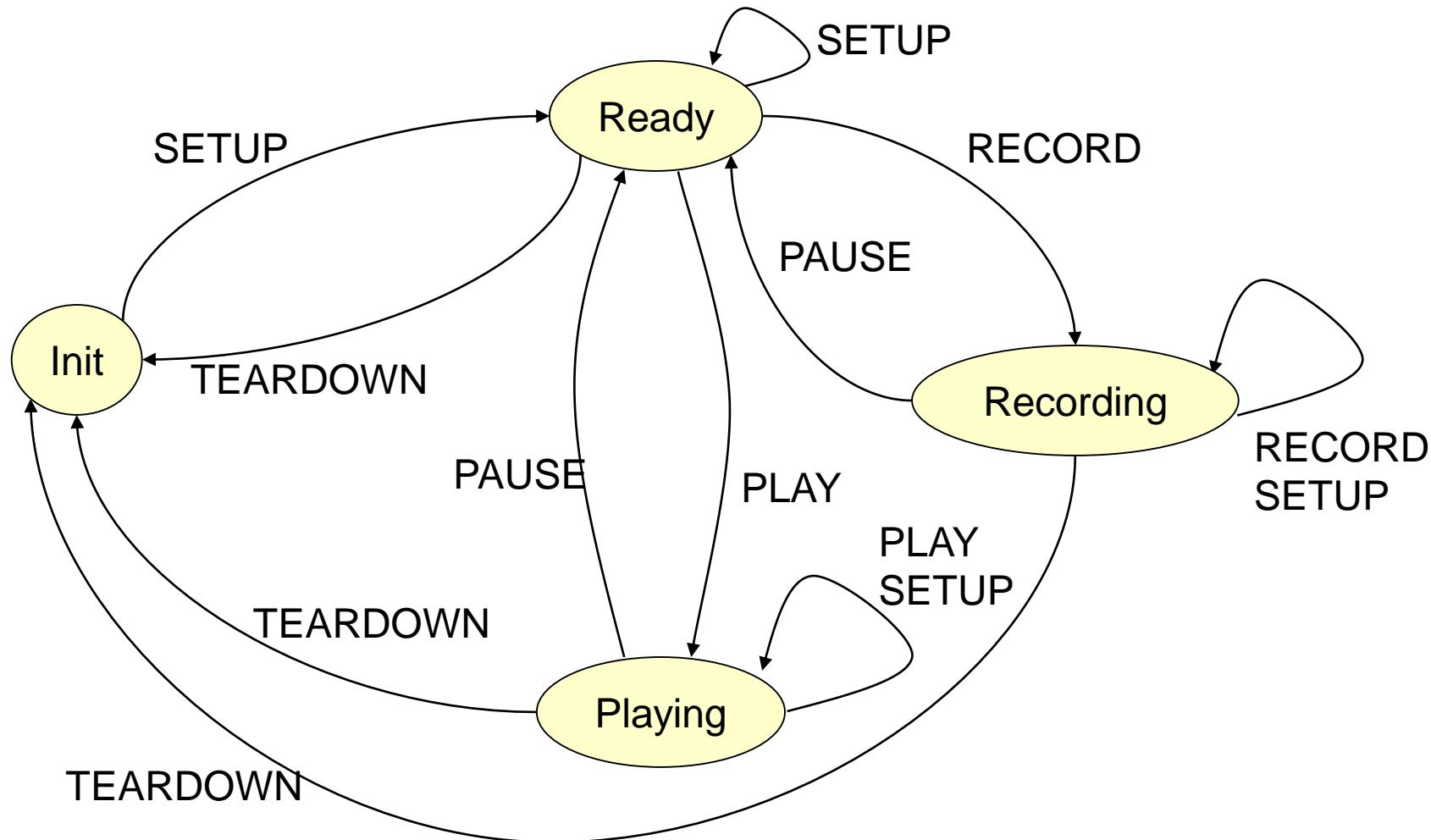
# Svojstva RTSP-a

- RTSP je aplikacijski protokol, sličan HTTP-u po sintaksi i načinu rada
- Glavne razlike:
  - RTSP ima nekoliko novih metoda i drugu identifikaciju protokola
  - RTSP čuva stanje (identifikator sjednice) za svaki prikaz u tijeku, za razliku od HTTP-a koji je *stateless*
  - i klijent i poslužitelj mogu slati zahtjeve
  - uz kontrolu koju vrši RTSP, sam prijenos podataka uglavnom vrši neki drugi protokol (npr. RTP)
- korisnikova kontrola prikaza slična daljinskom upravljaču na video uređaju

# Arhitektura klijent - poslužitelj



# Ponašanje klijenta i poslužitelja



# RTSP metode i odgovori

## Metode

OPTIONS

DESCRIBE

ANNOUNCE

SETUP

PLAY

PAUSE

TEARDOWN

GET\_PARAMETER

SET\_PARAMETER

REDIRECT

RECORD

## Odgovori / kodovi

2xx uspjeh

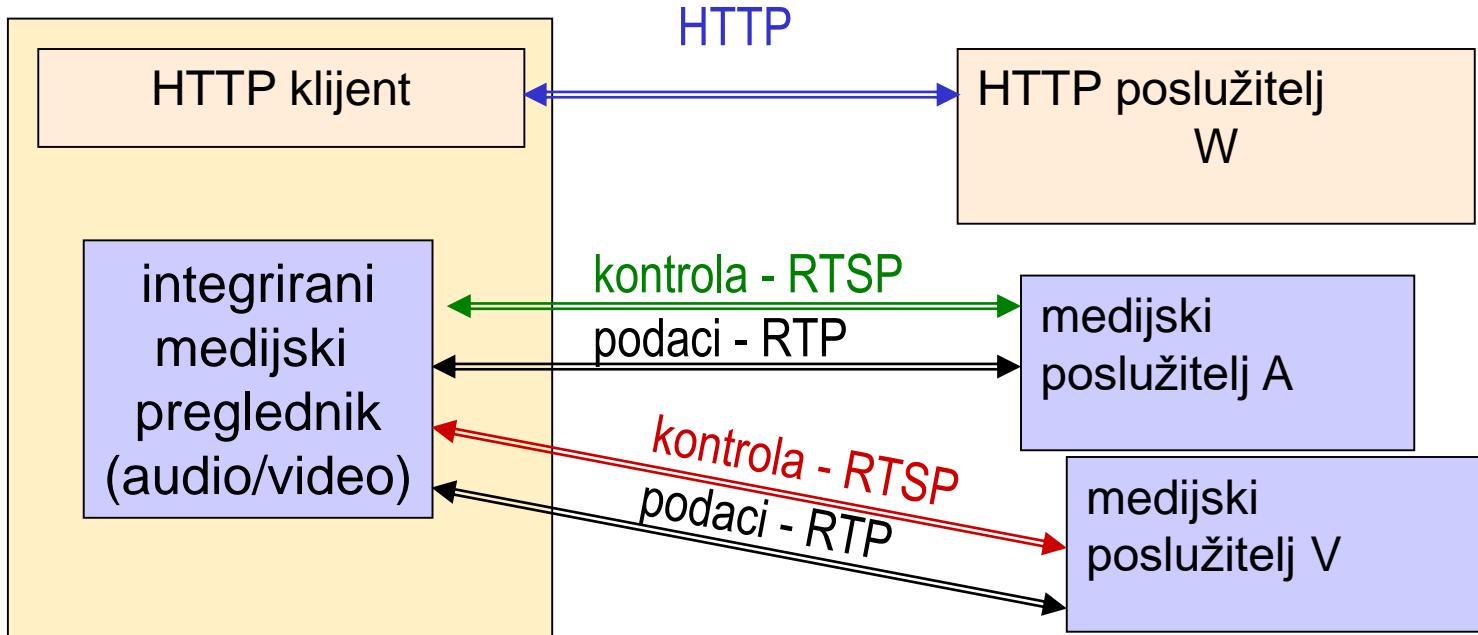
3xx preusmjeravanje

4xx pogreška klijenta

# Primjer aplikacije

- primjer iz RFC-a (RTSP-example-Media on Demand.pdf)

Web preglednik





Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

11.  
Uvođenje kvalitete usluge u  
Internet

# Pojam kvalitete usluge (Quality of Service – QoS) (1/2)

- kvaliteta usluge u najširem smislu je *stupanj zadovoljstva korisnika usluge* (ITU-T E.800)
- može se izraziti na tri razine:
  - aplikacija
  - sustav
  - mreža
- 1. kvaliteta na razini aplikacije (“korisnik” je čovjek)
  - uglavnom kvalitativni parametri
    - percepcijska kvaliteta pojedinog medija
    - odnos među pojedinim medijima, kvaliteta međusobne usklađenosti

# Pojam kvalitete usluge (Quality of Service – QoS) (2/2)

- 2. kvaliteta na razini sustava (“korisnik” je aplikacija)
  - kvalitativni i kvantitativni parametri
    - propusnost
    - vrijeme odziva
    - sustav posluživanja
    - raspoređivanje
- 3. kvaliteta na razini mreže (“korisnik” je sustav)
  - izražava se preko mjerljivih, kvantitativnih i kvalitativnih parametara kvalitete usluge
    - propusnost
    - kašnjenje
    - kolebanje kašnjenja
    - gubici
    - raspoloživost
    - blokiranje

# Iskustvena kvaliteta (Quality of Experience - QoE)

- Definicija [ITU-T P.10, 2007]:
  - Iskustvena kvaliteta je sveukupna prihvatljivost aplikacije ili usluge, subjektivno percipirana od strane krajnjeg korisnika
  - Pojam iskustvene kvalitete obuhvaća sve učinke sustava s kraja na kraj (klijent, terminal, mreža, itd.). Korisnikova očekivanja i kontekst mogu također utjecati na ukupnu prihvatljivost aplikacije ili usluge.

**Iskustvena kvaliteta je vezana uz subjektivnu percepciju kvalitete**

# Osnovne razlike: QoS i QoE

Karakteristike	QoS	QoE
<b>Područje primjene</b>	Uglavnom telekomunikacijske usluge	Šira primjena (nisu nužno u pitanju samo uslugu u mrežnom okruženju)
<b>Fokus</b>	Tehničke performanse sustava (mehanizmi kao npr. Diffserv)	ICT usluge ili aplikacije
<b>Metode mjerena</b>	Tehnološki orijentirane; empirijska ili simulacijska mjerena	Multi-disciplinaran pristup (razna subjektivna i objektivan mjerena)

izvor: Qualinet White Paper on definitions of QoE, 2012

# Upravljanje kvalitetom usluge

---

- **Tri uloge:**

- rezervacija i dodjela resursa od izvora do odredišta za vrijeme višemedijskog poziva/sjednice
- održavanje resursa prema specifikaciji zatražene kvalitete usluge
- prilagodba promjenama koje nastaju tijekom poziva

# Kvaliteta usluge u Internetu

- usluge u Internetu koriste **“best-effort” model**  
*Best-effort model: mreža će nastojati zadovoljiti korisnikove zahtjeve, ali bez ikakvih garancija da će tražena kvaliteta zaista biti pružena*
- “best-effort” usluga nije prihvatljiva za primjene u stvarnom vremenu i višemedijske primjene

# “Best-effort” model

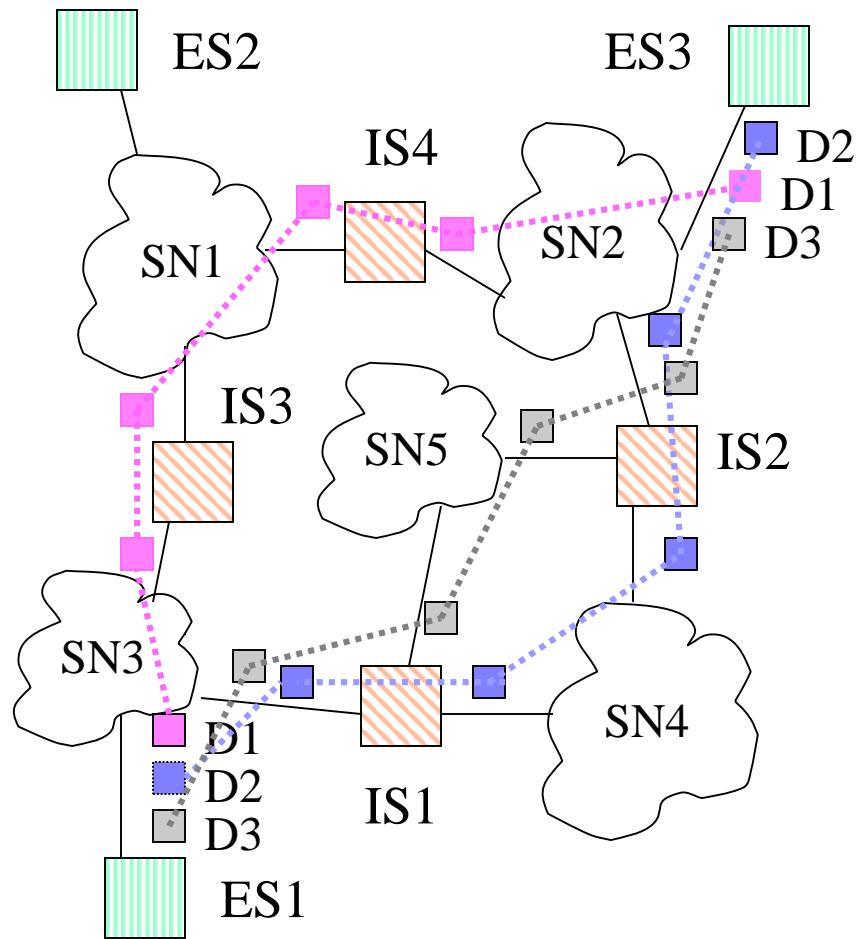
- posljedica datagramskog načina rada
  - komutacija paketa!
  - svaki paket usmjerava se neovisno o ostalima
  - različito kašnjenje i gubici po različitim putevima
- kašnjenje
  - propagacijsko kašnjenje (neizbjježno!)
  - vrijeme čekanja u usmjeriteljima (varira, uglavnom proporcionalno duljini repa čekanja, neovisno o vrsti prometa)
  - transmisijsko kašnjenje
- kolebanje kašnjenja - razlike u kašnjenju pojedinih paketa
- gubici kod zagušenja
  - u svakom usmjeritelju datagram ulazi na kraj repa čekanja s FIFO sustavom posluživanja
  - kad se rep čekanja napuni, odbacuju se paketi s kraja repa

# Datagramski način rada

- nema uspostave veze s kraja na kraj
- usmjeravanje datagrama se vrši za svaki datagram pojedinačno, na temelju odredišne adrese u zaglavljku
- svaki datagram putuje kroz niz mreža
- nema garancije, mreža daje "best effort" uslugu

## Primjer

- ES1 šalje ES3: D1, D2, D3
- datagrami putuju raznim stazama i uz različita kašnjenja i gubitke



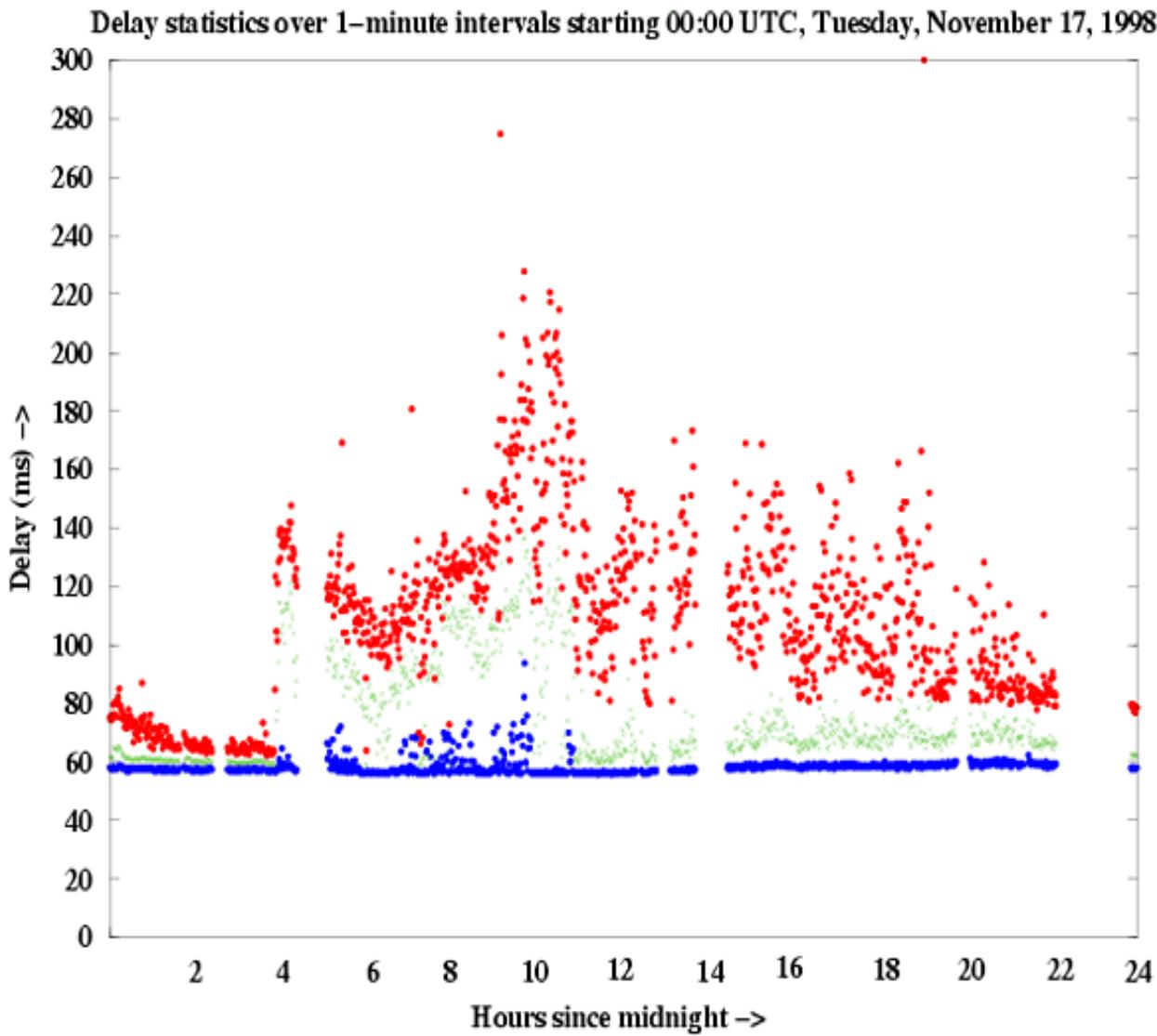
# Komponente kašnjenja

---

- transmisijsko kašnjenje - vrijeme potrebno za slanje paketa na mrežnom sučelju
  - relativno malo u modernim sustavima (red veličine  $\mu$ s)
  - npr. poznati rezultati za ispitne pakete duljine 40-bytne na T3 sučelju (44.736 Mbit/s) transmisijsko kašnjenje oko 7  $\mu$ s; isto na T1 sučelju (1.544 Mbit/s) oko 220  $\mu$ s
- propagacijsko kašnjenje - fizikalno ograničenje, zbog konačne brzine prostiranja signala
  - u LAN-ovima zanemarivo, ispod 1 ms
  - u WAN-ovima više desetaka ms, ovisno o udaljenosti
    - npr. ako se uzme propagacija kroz optičko vlakno približno 200 000 km/s, link preko sjevernoameričkog kontinenta (SAD) ima propagacijsko kašnjenje oko 20 ms
- kašnjenje zbog čekanja u repu - čekanje na obradu u usmjeriteljima

# Mjerenje na vezi SAD - Nizozemska, studeni 1998

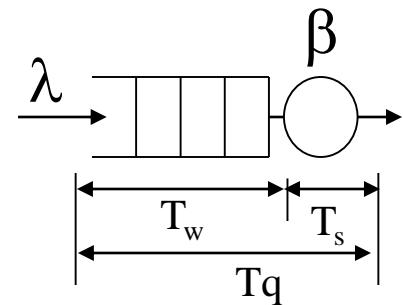
- minimum delay
- 50th percentile delay
- 90th percentile delay



The blue dots represent the minimums during each one minute interval, the green dots represent 50th percentile values, and the red dots represent 90th percentile values. When the 90th percentile is infinite (more than 10 percent packet loss), it is plotted as a red dot at the top of the graph. Gaps in the graph represent a period of no measurement activity. When the path is uncongested, the minimum observed delay in the given minute (blue) will be close to the propagation delay. We have found that even in a congested path, it is likely that at least one of the test packets will not experience congestion. So the blue points will still be the same and stay close to the propagation delay. So changes in the value of the blue points, typically, indicates a change in the path or an extreme case of congestion. The green and red points are indicators of congestion. When the path is uncongested, they are also close to the propagation delay and hence are close to the blue points. During periods of congestion, the green and red points differ from the blue points considerably. The distance between them is a measure of variance, and hence congestion.

# Gubici zbog zagušenja

- detaljno obrađeno u teoriji sustava posluživanja (Informacijske mreže, IV godina)
- zagušenje je neizbjegljivo kada je  $\lambda > \beta$   
 $(\lambda - \text{intenzitet nailazaka}; \beta - \text{intenzitet posluživanja})$   
 iz Littleove formule  $L_w = \lambda T_w$  mogu se izračunati parametri sustava posluživanja; vrijedi neovisno o vrsti ili granicama sustava, broju poslužitelja, svojstvima dolaznog toka, karakteristikama vremena posluživanja, disciplini posluživanja ili raspoređivanja, ALI NE VRIJEDI za sustav s gubicima!
- preusmjeravanje nije rješenje!
- neizvedivi “načini” izbjegavanja zagušenja:
  - savršena koordinacija
  - beskonačni međuspremnići
- izvedivi načini: ograničenje  $\lambda$  na ulazu u sustav, tj. kontrola prihvata



# Utjecaj transmisijskih pogrešaka

- Shannonova formula daje kapacitet **C** frekvencijski ograničenog kanala s aditivnim bijelim šumom (teorijski maksimum, nema grešaka)

$$C = B * \log_2 (1+S/N) \text{ [bit/s]}$$

C = kapacitet kanala [bit/s], B = pojasna širina [Hz], S/N = odnos signal - šum

- utjecaj šuma -> pogreške na razini bita (BER, bit error rate), tipično  $10^{-12}$  za optičko vlakno do  $10^{-2}$  za svemirske komunikacije
- kod paketske komunikacije, pogreška na razini bita "uništava" paket
  - pogreške se kompenziraju uvođenjem redundancije (kodovi za detekciju i korekciju pogrešaka) ili retransmisijom (primjer: TCP)

# Primjer (M/M/1)

IP usmjeritelj šalje pakete kroz 64 kbit/s link. Duljina paketa, sa svim zaglavljima, je eksponencijalna slučajna varijabla sa srednjom vrijednošću 400 byte. Međudolazni interval je eksponencijalna slučajna varijabla sa srednjom vrijednošću od 15 paketa u sekundi (pps). Kada se novi paket usmjerava kroz taj link, sljedeći nailazeći paket se sprema u FIFO rep.

Izračunajmo duljinu repa u usmjeritelju u paketima i prosječno vrijeme od kada paket nađe do trenutka kad je poslat zadnji bit tog paketa.

$$\lambda = 15 \text{ pps}$$

$$\beta = 64000 / (400 * 8) = 20 \text{ pps}$$

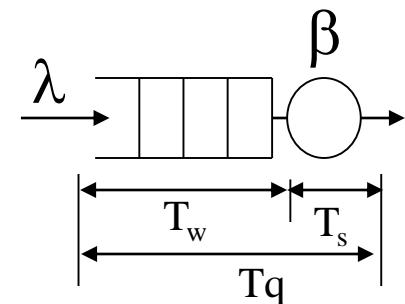
$$\text{opterećenje } \rho = \lambda / \beta = 15 / 20 = 0,75$$

$$\text{srednji broj paketa u sustavu } Lq = \rho / (1 - \rho) = 0,75 / 0,25 = 3$$

$$\text{srednje vrijeme zadržavanja } Tq = Lq / \lambda = 3 / 15 = 0,2 \text{ s}$$

$$\text{srednje vrijeme čekanja } Tw = Tq - Ts = Tq - 1 / \beta = 0,2 - 1/20 = 0,15 \text{ s}$$

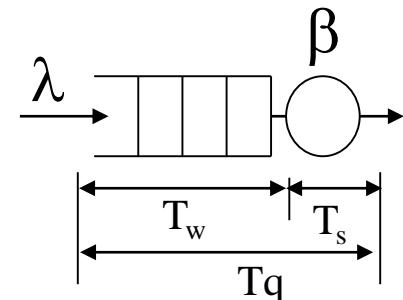
$$\text{srednji broj paketa u repu čekanja } Lw = \lambda Tw = 15 * 0,15 = 2,25$$



# Primjer (M/M/1) - nastavak

Ponovimo račun za  $\lambda = 19$  pps.

Ostavimo  $\beta = 20$  pps.



$$\text{opterećenje } \rho = \lambda / \beta = 19 / 20 = 0,95$$

$$\text{srednji broj paketa u sustavu } Lq = \rho / (1 - \rho) = 0,95 / 0,05 = 19$$

$$\text{srednje vrijeme zadržavanja } Tq = Lq / \lambda = 19 / 19 = 1 \text{ s}$$

$$\text{srednje vrijeme čekanja } Tw = Tq - Ts = Tq - 1 / \beta = 1 - 1/20 = 0,95 \text{ s}$$

$$\text{srednji broj paketa u repu čekanja } Lw = \lambda Tw = 19 * 0,95 = 18,05$$

# Primjer (M/M/1) - nastavak

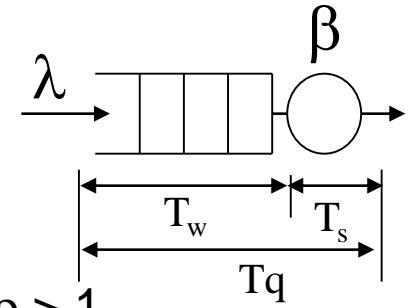
Ponovimo račun za  $\lambda = 25$  pps.

Ostavimo  $\beta = 20$  pps.

... više nema smisla opterećenje  $\rho = \lambda / \beta$  jer bismo dobili  $\rho > 1$

... više ne vijedi model M/M/1 jer dobivamo opterećenje  $\rho > 1$  i dolazi do gubitaka paketa

Kada se ulazni međuspremnik napuni, usmjeritelj će početi odbacivati pakete brzinom od  $25 - 20 = 5$  pps.



# Stvarni gubici u Internetu

Regions	Number	Entire day	Worst quarter of the day	Worst hour of the day
ALL PATHS	1084	1.06%	1.44%	4.88%
US-US	1060	1.04%	1.40%	4.79%
US-EUROPE	12	1.39%	1.52%	5.71%
US-NZ	10	2.58%	4.04%	11.15%
EUROPE-NZ	2	0.58%	1.51%	6.97%

Izvor:

<http://telesto.advanced.org/~kalidindi/papers/INET/inet99.html>

Sunil Kalidindi, Matthew J. Zekauskas

Advanced Network & Services, USA, June, 1999

*Surveyor: An Infrastructure for Internet Performance Measurements*

# Stvarni gubici u Internetu – ovisnost o veličini paketa

IPv4 packet size	all lost	10/10 received	9/10 received	8/10 received	1/10-7/10 received	number of probes
100	1.0%	95.6%	2.2%	0.6%	0.6%	3627
700	1.4%	95.0%	2.5%	0.7%	0.5%	3629
1000	1.4%	95.1%	2.2%	0.7%	0.6%	3626
1400	2.1%	96.0%	1.2%	0.4%	0.4%	3611
1401	2.4%	93.1%	3.2%	0.7%	0.7%	3616
1454	2.4%	92.6%	3.6%	0.6%	0.8%	3625
1455	2.5%	90.2%	5.3%	1.2%	0.8%	3626
1460	2.6%	92.9%	3.3%	0.5%	0.6%	3614
1461	2.8%	90.0%	5.8%	0.9%	0.6%	3624
1480	2.9%	89.1%	6.2%	1.0%	0.8%	3629
1481	3.0%	88.2%	7.3%	0.8%	0.7%	3631
1488	3.0%	88.9%	6.7%	0.9%	0.5%	3619
1489	3.1%	88.5%	6.8%	0.9%	0.7%	3624
1492	3.1%	88.2%	7.1%	0.9%	0.7%	3624
1493	5.1%	71.3%	<b>20.2%</b>	2.8%	0.7%	3619
1500	<b>5.2%</b>	71.0%	<b>20.8%</b>	2.3%	0.6%	3618
1501	<b>9.9%</b>	85.3%	2.7%	0.6%	1.5%	3620
1502	9.9%	84.8%	2.4%	1.0%	1.9%	3620
1600	9.8%	84.9%	2.7%	1.0%	1.5%	3626

Izvor:

<https://labs.ripe.net/Members/emileaben/ripe-atlas-packet-size-matters>

Emile Aben, Philip Homburg, 02 Oct 2013

# Podaci iz literature za “podnošljive” vrijednosti kašnjenja i gubitaka za višemedijske primjene

	Video telefonija	JPEG video transmisijska
Prosječno kašnjenje	0.25 s	0.2 s
Kolebanje kašnjenja	10 ms	5 ms
Max. BER	0.01	0.1
Max. PER	0.001	0.01

Oznake:

BER - Bit Error Rate

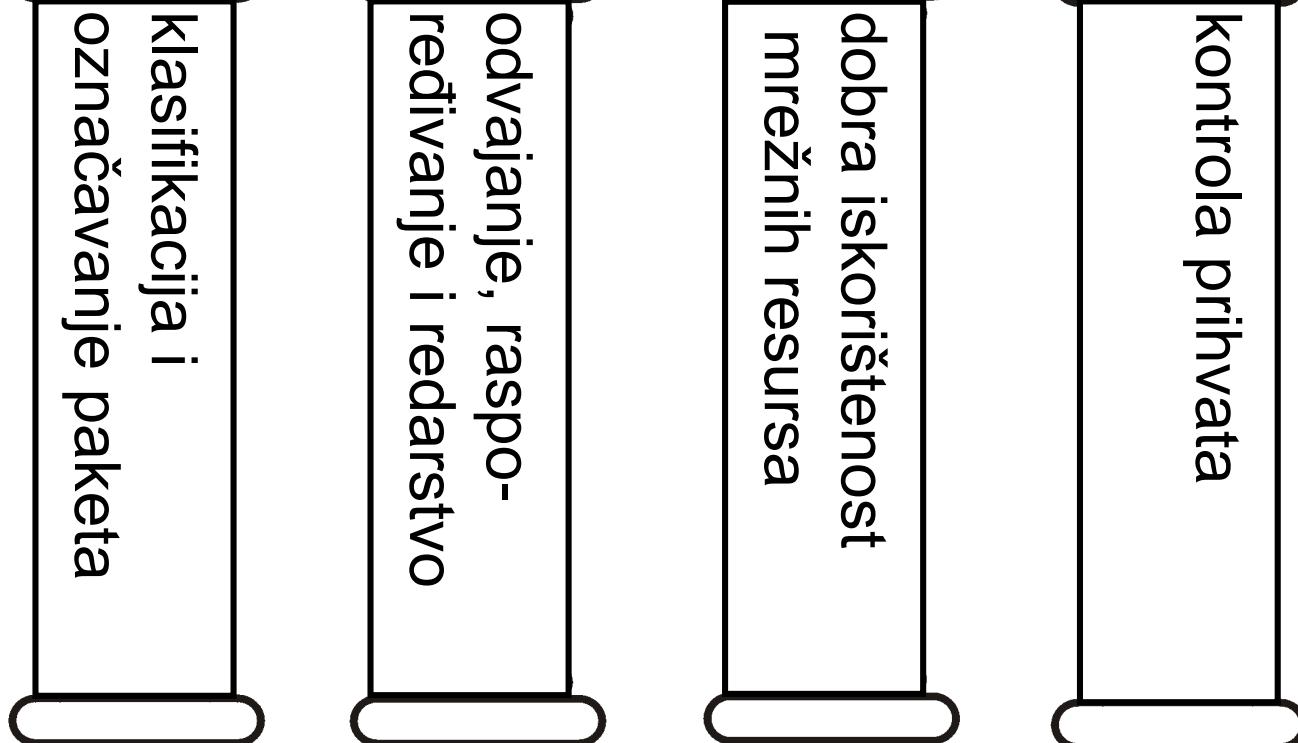
PER - Packet Error Rate

---

# Pregled mehanizama koji omogućuju uvodenje QoS-a u Internet

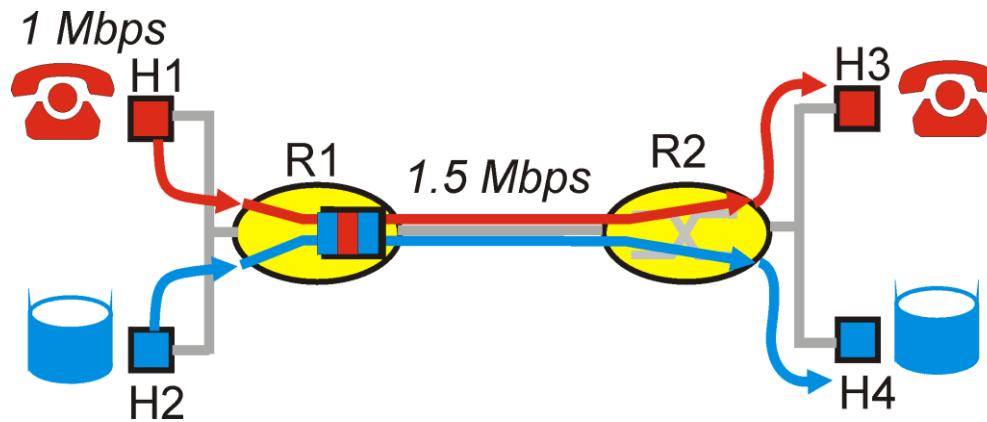
# Načela uvođenja QoS

## QoS za umrežene aplikacije



# Načelo 1: klasifikacija i označavanje paketa

- Primjer: 1 Mbps "IP telefon" i FTP dijele 1,5 Mbps link.
  - usnopljeni FTP može zagušiti usmjeritelj i uzrokovati gubitak govornih podataka
  - ideja je dati prednost govoru pred FTP-om

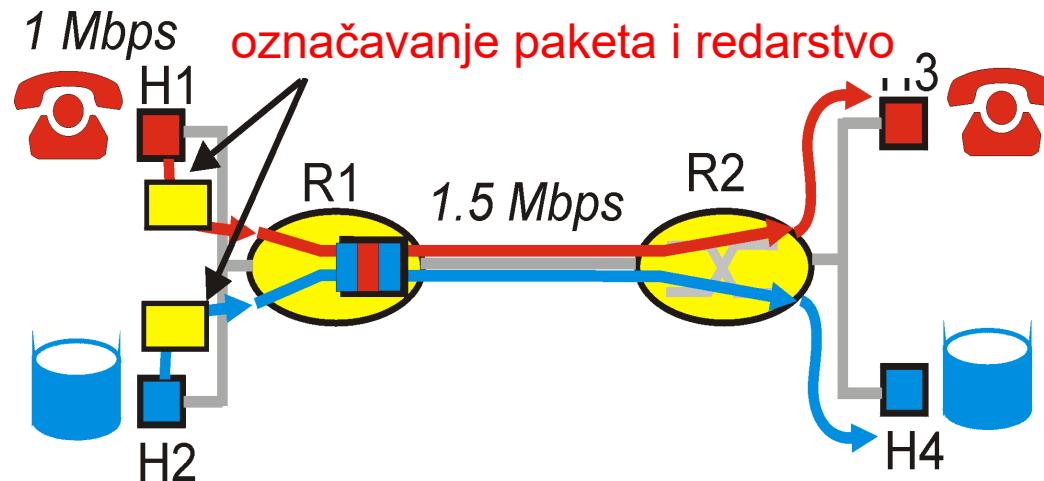


## Načelo 1

treba klasificirati, odnosno označiti pakete kako bi usmjeritelj mogao razlikovati pakete koji pripadaju različitim klasama te prilagoditi ponašanje usmjeritelja prema paketima koji zahtijevaju drugačije rukovanje

# Načelo 2: odvajanje, raspoređivanje, redarstvo

- pitanje: što ako aplikacije “varaju” šaljući većom brzinom od dogovorene/dodijeljene?
  - redarstvo (“policija”, engl. policing): prisiljava izvor da se ponaša u skladu s dodijeljenim kapacitetom
- označavanje i redarstvo na rubu mreže

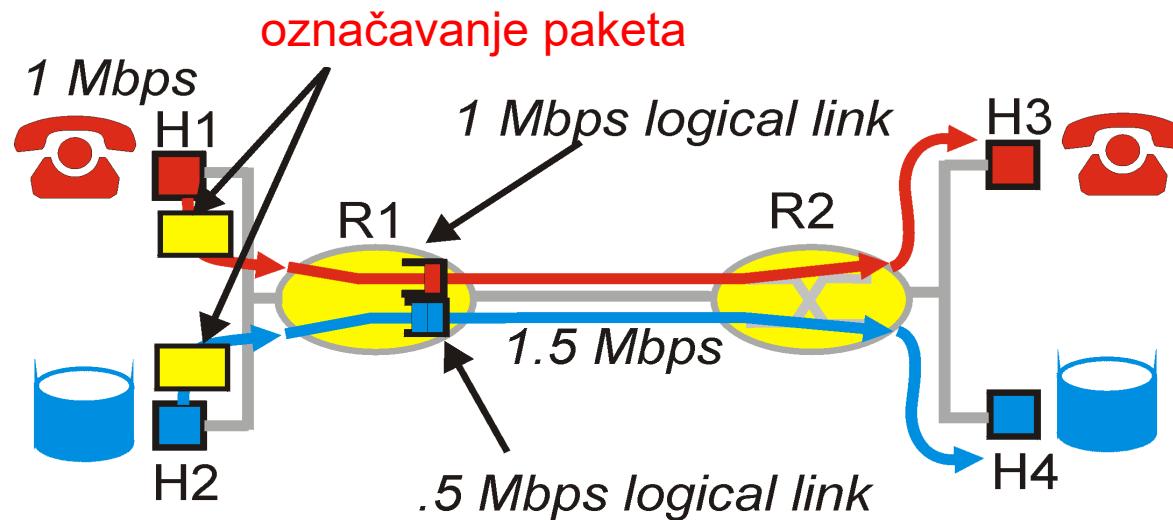


Načelo 2

osigurati zaštitu (odvajanje, izolaciju) jedne klase od ostalih

# Načelo 3: dobra iskorištenost mrežnih resursa

- pitanje: kako riješiti neučinkovito korištenje kapaciteta ako tokovi zapravo ne koriste dodijeljene im resurse (slučaj dodjele fiksnih (nedjeljivih) kapaciteta pojedinom toku)?

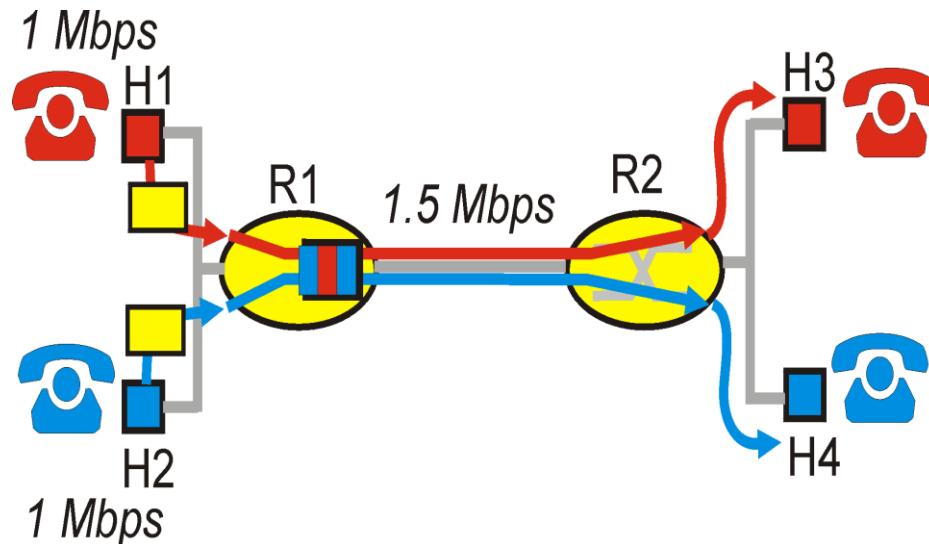


Načelo 3

uz odvajanje, poželjno je što bolje iskorištavanje mrežnih resursa

# Načelo 4: kontrola prihvata

- Poštovati fizičko ograničenje: ne može se udovoljiti zahtjevima većim od kapaciteta (ne smije biti  $\lambda > \beta$ )!

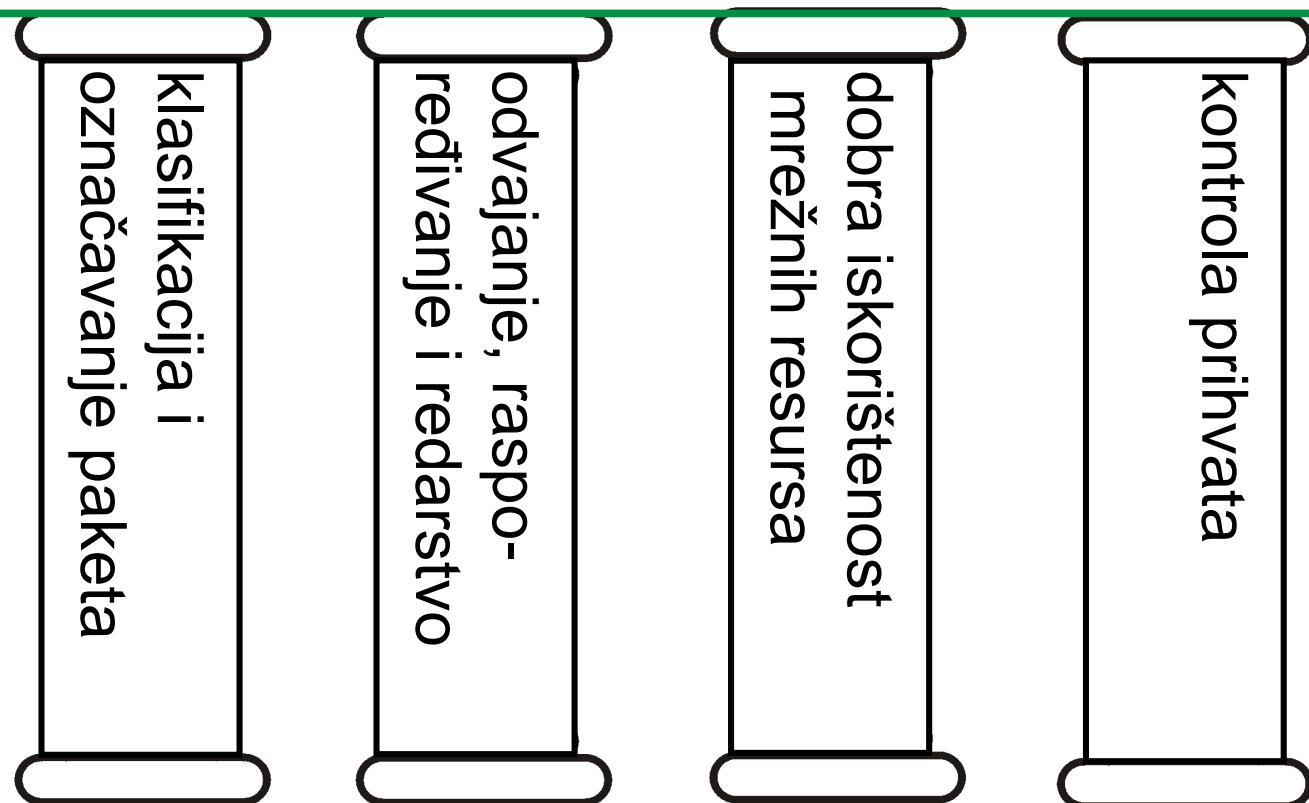


Načelo 4

**kontrola prihvata:** tok objavljuje svoj prometni zahtjev, a mreža može blokirati poziv ako ga ne može poslužiti

# Načela uvođenja QoS

## QoS za umrežene aplikacije

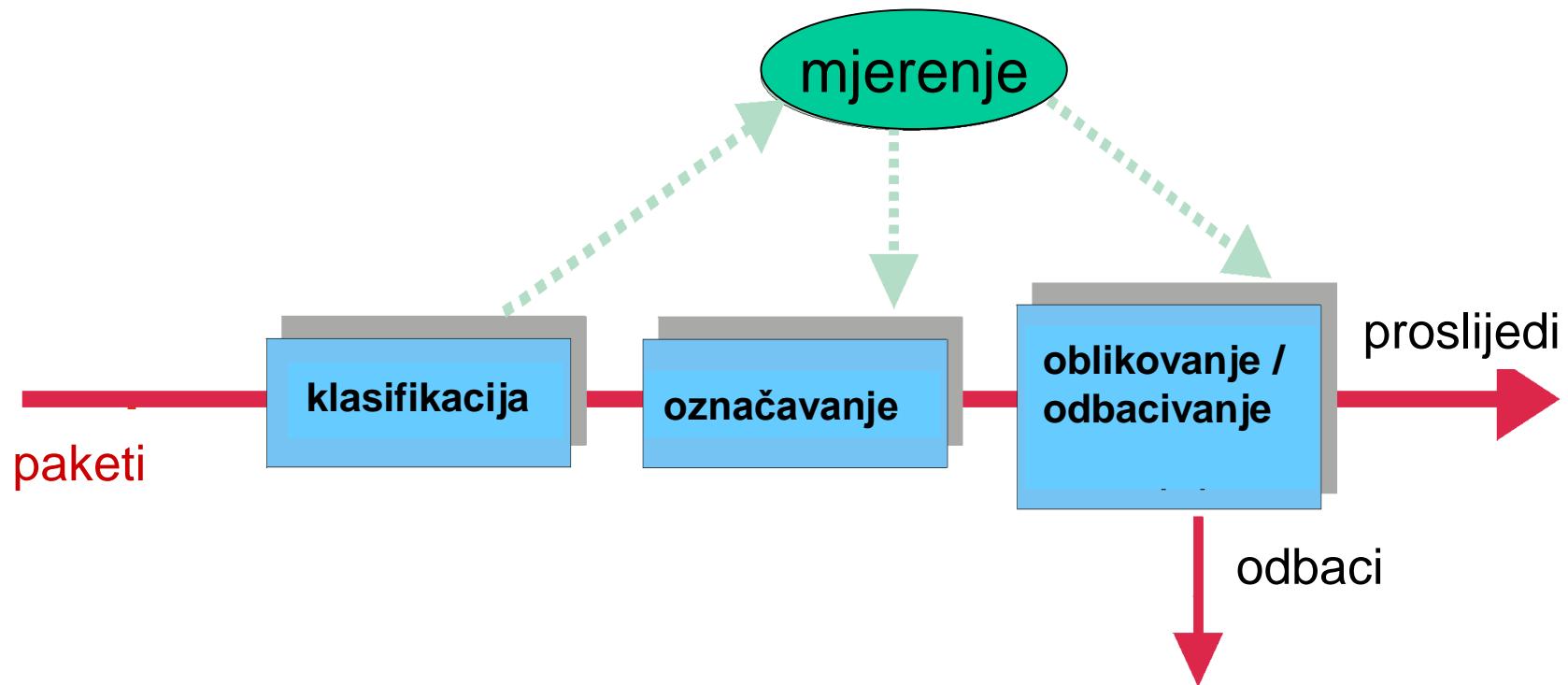


...slijedi pregled mehanizama koji to omogućuju...

# Klasifikacija i oblikovanje

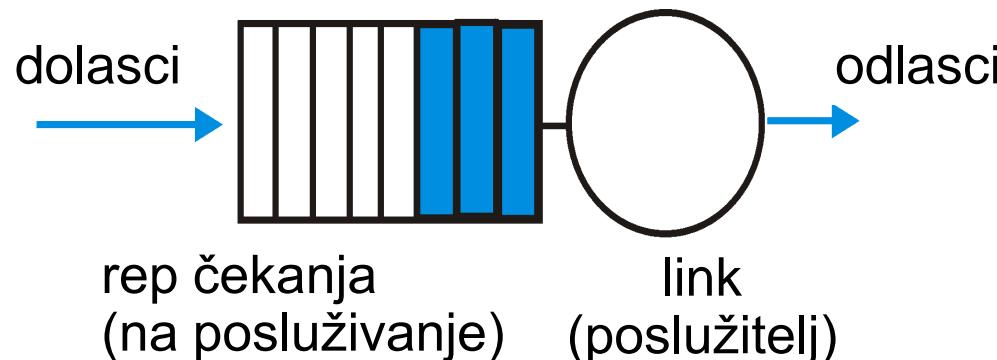
ponekad je potrebno ograničiti ulazni promet za neke klase:

- korisnik deklarira profil prometa (npr. brzina, usnopljenost)
- promet se mjeri i preoblikuje ili odbacuje ako ne odgovara deklaraciji



# Mehanizmi raspoređivanja i nadgledanja

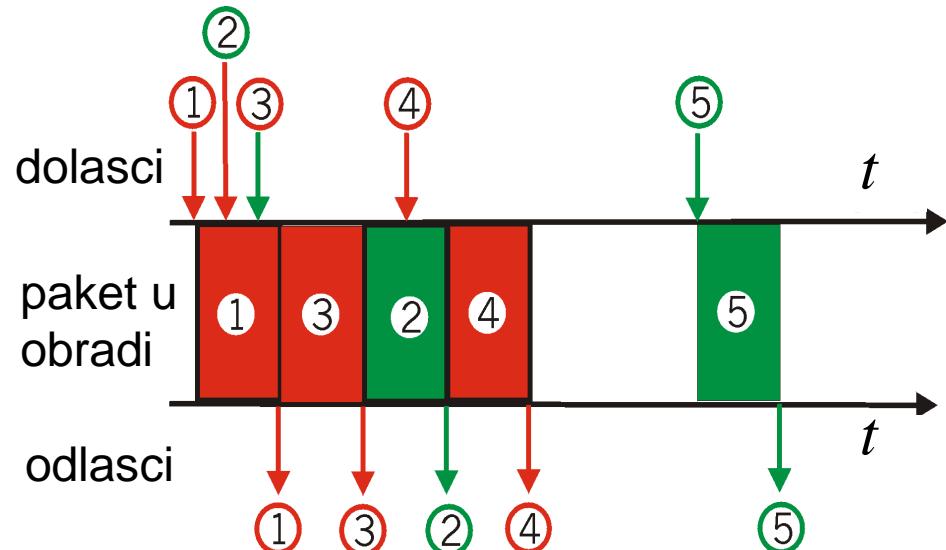
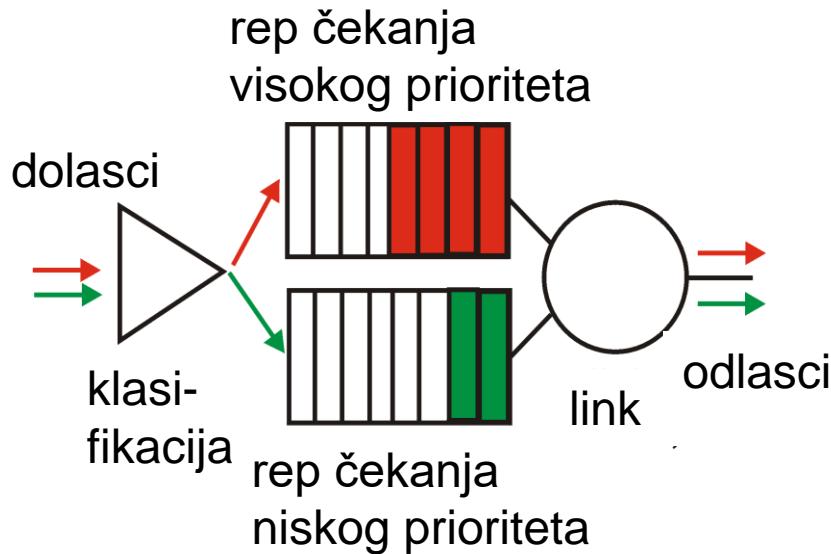
- **raspoređivanje:** određivanje sljedećeg paketa kojeg će se poslati na izlazno sučelje
- **FIFO (first in first out) raspoređivanje:** slanje u redosljedu dolazaka, odn. ulaska u rep čekanja
  - **politika odbacivanja:** ako paket dolazi u već puni rep – kojeg odbaciti?
    - odbacivanje zadnjeg (engl. *tail drop*): odbaci novoprdošli paket
    - prioritet: odbaci/obriši na osnovi prioriteta
    - slučajno: odbaci/obriši slučajnim odabirom



# Raspoređivanje po prioritetima

**Prioritetno raspoređivanje:** šalje se onaj paket iz repa čekanja koji ima najviši prioritet

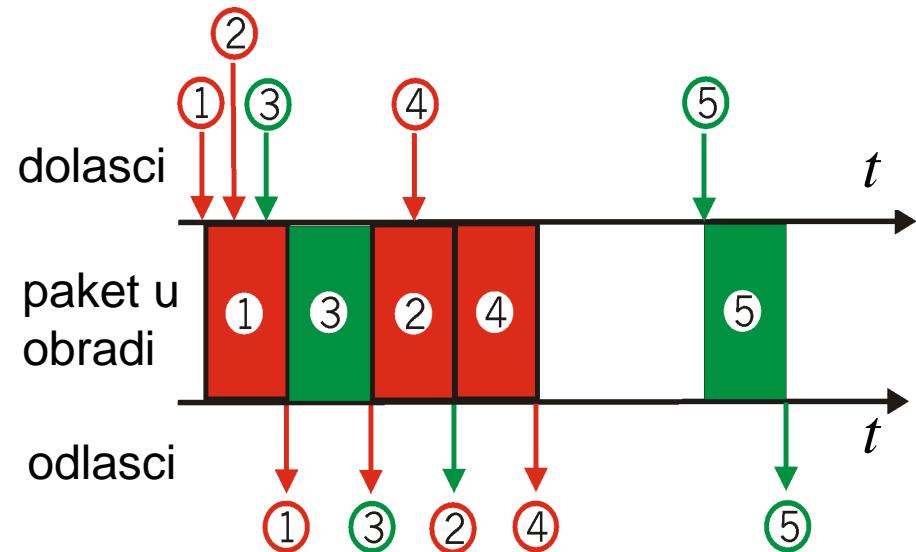
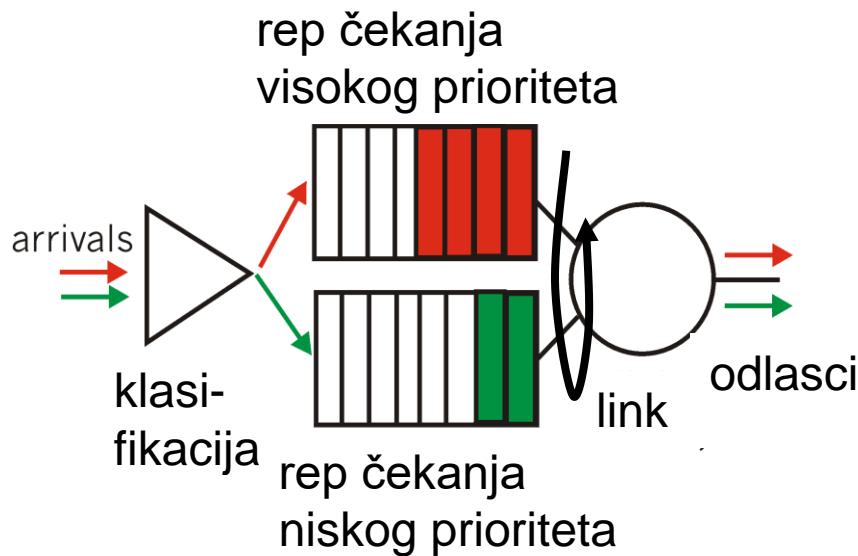
- više *klasa*, s različitom prioritetima
  - klasa može ovisiti o oznaci u paketu ili drugim podacima u zaglavljtu paketa (npr. IP izvorišna i odredišna adresa i portovi i sl.)



# Raspoređivanje unaokolo

## Raspoređivanje unaokolo (engl. round robin):

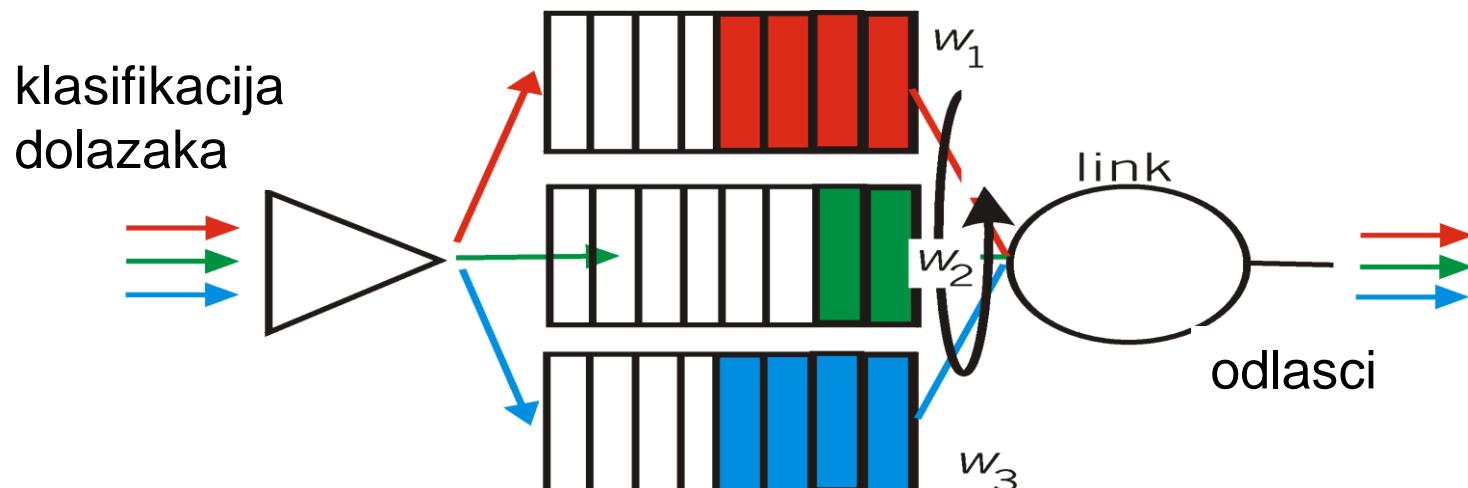
- više "klasa" (diffserv klase, tokovi, ...)
- ciklički obilazak repova po klasama, posluživanje po jednog paketa iz svake klase redom



# Težinski pravedno raspoređivanje

## Težinski pravedno raspoređivanje (Weighted Fair Queuing, WFQ):

- generalizacija posluživanja unaokolo
- svaka klasa dobiva težinski faktor i prema njemu primjerenu količinu posluživanja u svakom ciklusu

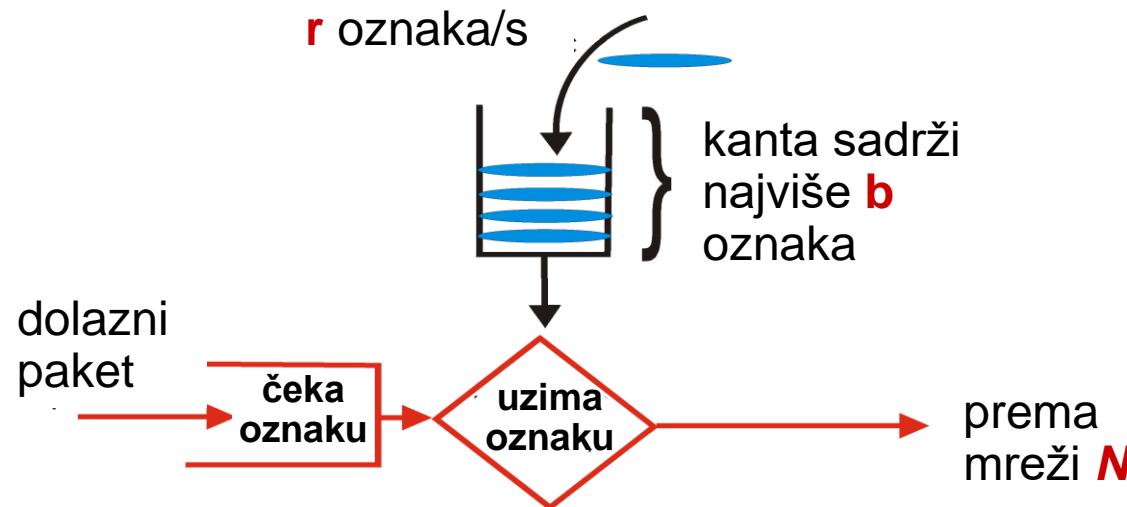


# Redarstveni mehanizmi

- Cilj: ograničiti promet kako ne bi prekoračio deklarirane parametre
- Tri često korištena kriterija:
  - **(Dugotrajna) srednja brzina:** koliko paketa se smije poslati u jedinici vremena (promatrano u dužem roku)
    - pitanje duljine intervala: 100 paketa u sekundi ili 6000 paketa u minuti imaju istu srednju vrijednost!
  - **Vršna brzina:** npr. 6000 paketa u minuti (ppm) srednja brzina, 15000 ppm vršna brzina
  - **Max. veličina snopa** (engl. *burst size*): max. broj paketa poslanih zaredom (bez razmaka)

# Redarstveni mehanizmi: Token bucket

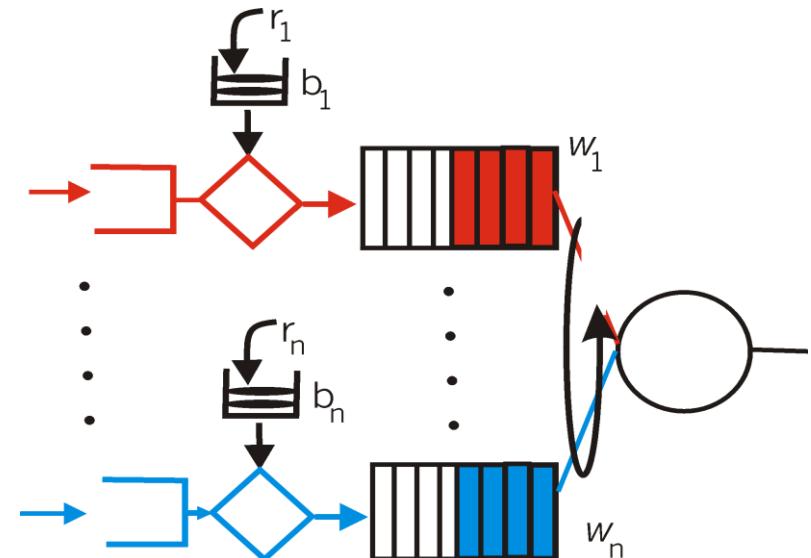
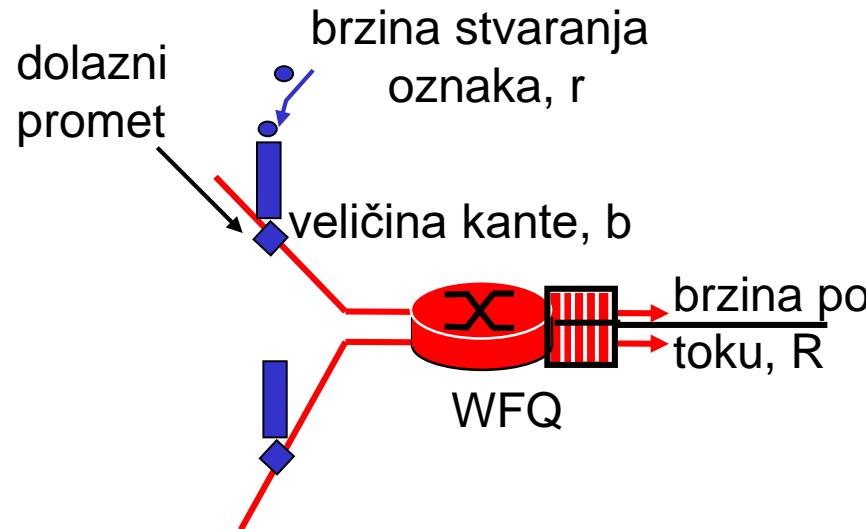
**Kanta s oznakama (Token Bucket):** ograničava ulaz na specificiranu veličinu snopa i srednju vrijednost



- ◆ “kanta” drži  $b$  oznaka
- ◆ oznake se generiraju brzinom  $r$  oznaka/s, osim ako je kanta puna
- ◆ u vremenskom intervalu duljine  $t$ , broj prihvaćenih paketa  $N \leq (r t + b)$ .

# Postizanje garantiranog QoS-a

- *token bucket* i WFQ u kombinaciji daju garantiranu gornju granicu kašnjenja, tj. *garantirani QoS!*



gornja granica kašnjenja:

$$D_{\max} = b/R$$

# Pristupi kvaliteti usluge u Internetu

---

- Integrirane usluge  
(Integrated Services, IntServ)
- Diferencirane usluge  
(Differentiated Services, DiffServ)

# Pristupi kvaliteti usluge u Internetu

- **Integrirane usluge**
  - ideja: IP za podršku usluga u stvarnom vremenu (*real-time*) kao i dosadašnjih (*non-real-time*) usluga
  - usmjeritelji pružaju uslugu za **tok** (*flow*), odn. niz paketa između aplikacija pošiljatelja i (jednog ili više) primatelja
  - ključni pojam: *rezervacija resursa*
- **Diferencirane usluge**
  - ideja: promatraju se **agregirani tokovi** ili **klase usluga**, a s paketima se u svakom usmjeritelju postupa prema obrascu *ponašanja za klasu* kojoj paketi pripadaju
  - u odnosu na integrirane usluge, postiže se bolja prilagodljivost veličini jer se klasifikacija paketa provodi samo na “rubu mreže”
  - ključni pojam: *prioritet klase*

---

# Integrirane usluge

# Integrirane usluge

- definirane su dvije vrste usluga, usmjerenе prema stvarno-vremenskim primjenama:
  - usluga kontroliranog opterećenja(*controlled load*)
    - tok podataka ima kvalitetu usluge kakvu bi imao pri malom opterećenju mreže u *best-effort* slučaju
  - garantirana (*guaranteed*)
    - garantirano kašnjenje (u matematički dokazivim granicama) zbog čekanja u usmjeriteljima i garantirana propusnost s kraja na kraj
- koristi se kontrola prihvata i rezervacija
- u pristupu integriranih usluga, za rezervaciju služi **Resource Reservation Protocol (RSVP)**

# Resource Reservation Protocol

- RSVP je signalacijski protokol koji vrši rezervaciju resursa, bez uspostave veze
- za promet koji zahtijeva uslugu, rezervacija se vrši za pojedinačni tok (*flow*)
- RSVP može raditi i s IPv4 i IPv6, a primjenjiv je kako za pojedinačno, tako i za višeodredišno razašiljanje (*multicast*)
- resursi se rezerviraju za svaki tok za koji se zahtijeva QoS u svakom mrežnom elementu (svakom usmjeritelju) na putu od izvora do odredišta

# Postupak rezervacije

- pošiljatelj (izvor toka) opisuje svojstva prometa
- primatelj traži željeni QoS od mreže za tok(ove) od nekog izvora
- rezervacijske poruke uvijek putuju prema pošiljatelju, odn. izvoru toka
- zadaće svakog mrežnog elementa:
  - rezervacija potrebnih resursa
  - prosljeđivanje zahtjeva “uzvodno” (prema pošiljatelju)
- *soft state* - stanje rezervacija po putu se osvježuje periodičkim rezervacijskim porukama

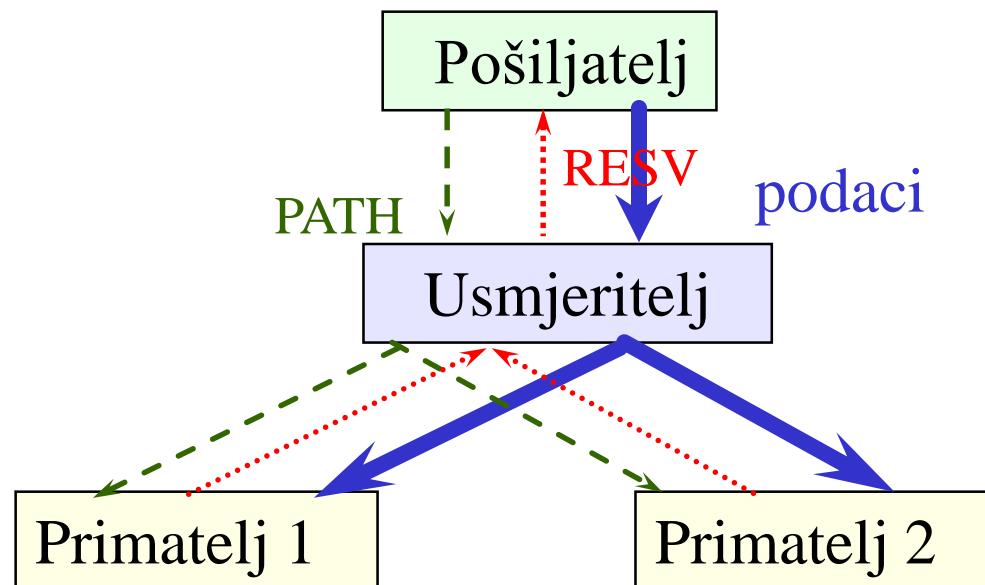
# Parametri rezervacije

---

- specifikacija toka (*flowspec*)
  - izbor usluge (*kontrolirano opterećenje ili garantirana*)
  - specifikacija prometa, **Tspec**, sadrži opis prometnih karakteristika
  - specifikacija zahtjeva, **Rspec**, sadrži željenu razinu kvalitete usluge za odabranu uslugu (po potrebi)
- specifikacija filtera (*filterspec*)
  - sadrži opis toka, odn. paketa za koje se traži rukovanje prema specifikaciji
  - razni načini, najjednostavniji: IP adresa i UDP/TCP port

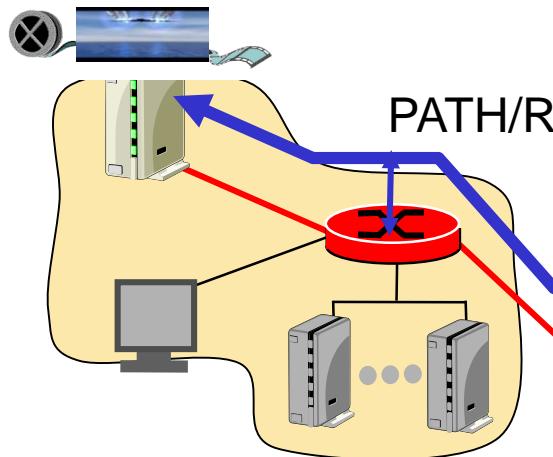
# RSVP poruke

- ◆ RSVP koristi dvije vrste poruka:
  - PATH poruke, za uspostavu staze
  - RESV poruke, za rezervaciju duž staze



# Postupak rezervacije

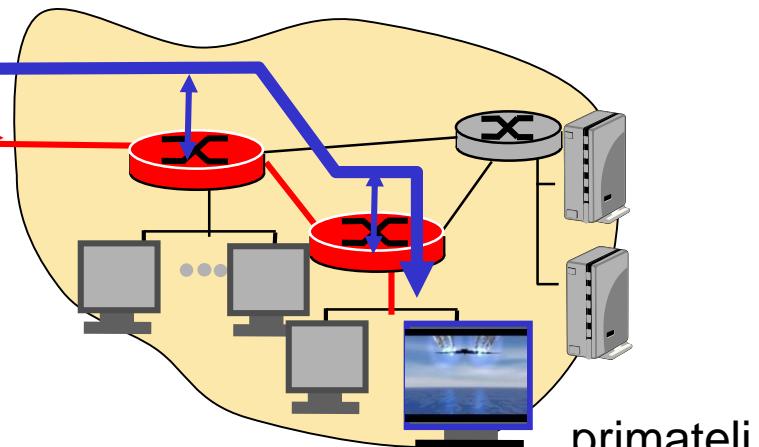
pošiljatelj (izvor)



- **Rezervacija resursa**

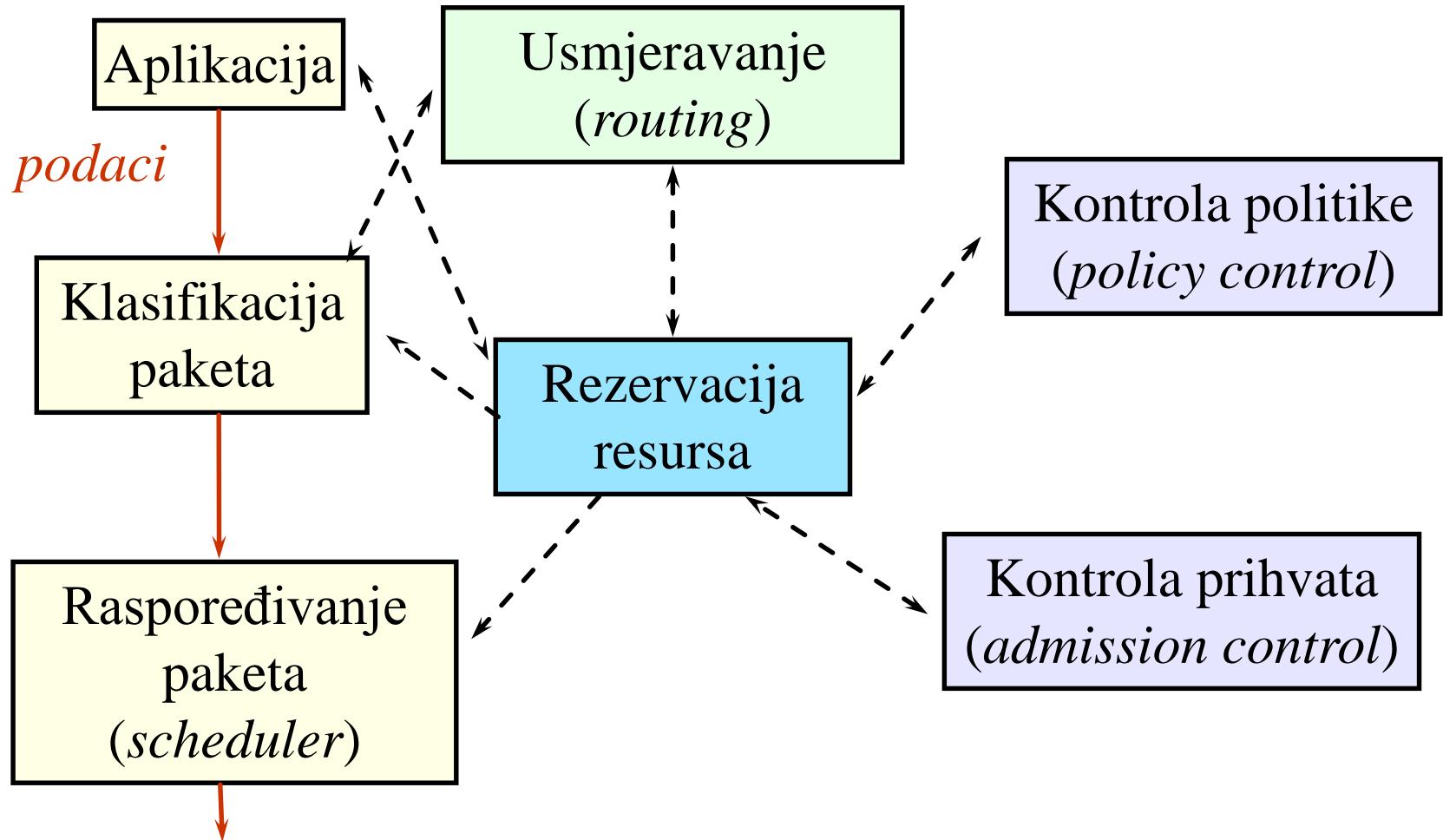
- uspostava poziva, signalizacija (RSVP)
- opis prometa, zahtjev za QoS
- kontrola prihvata u svakom mrežnom elementu

raspoređivanje  
prema QoS  
(npr, WFQ)



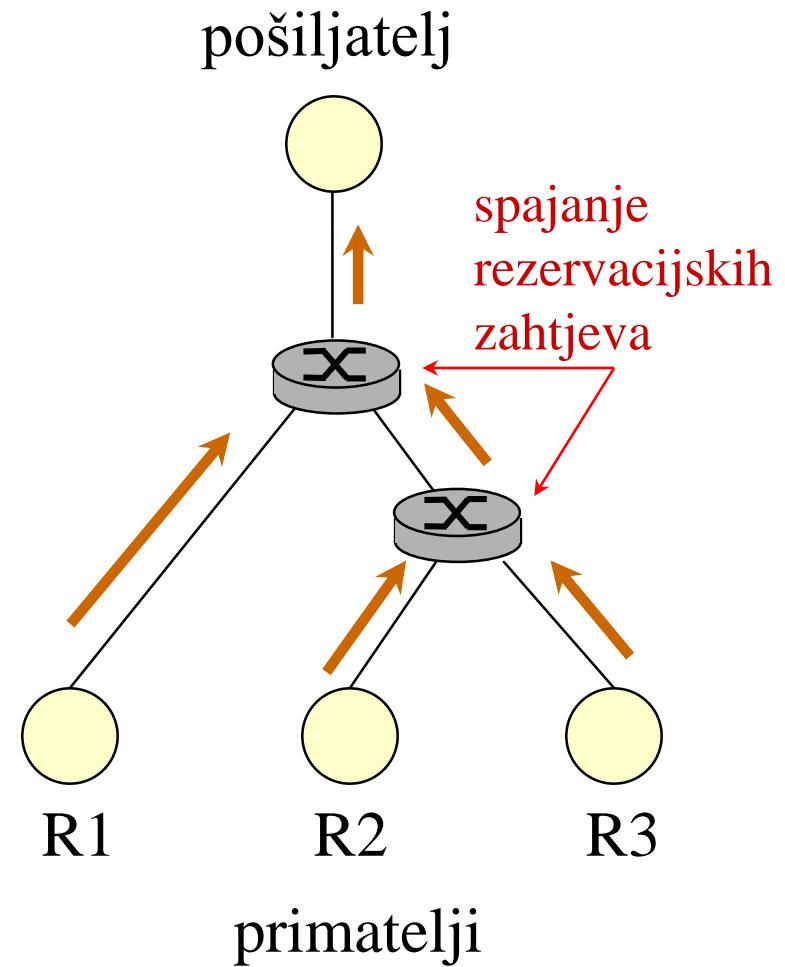
primatelj  
43

# Izvedba mrežnog elementa

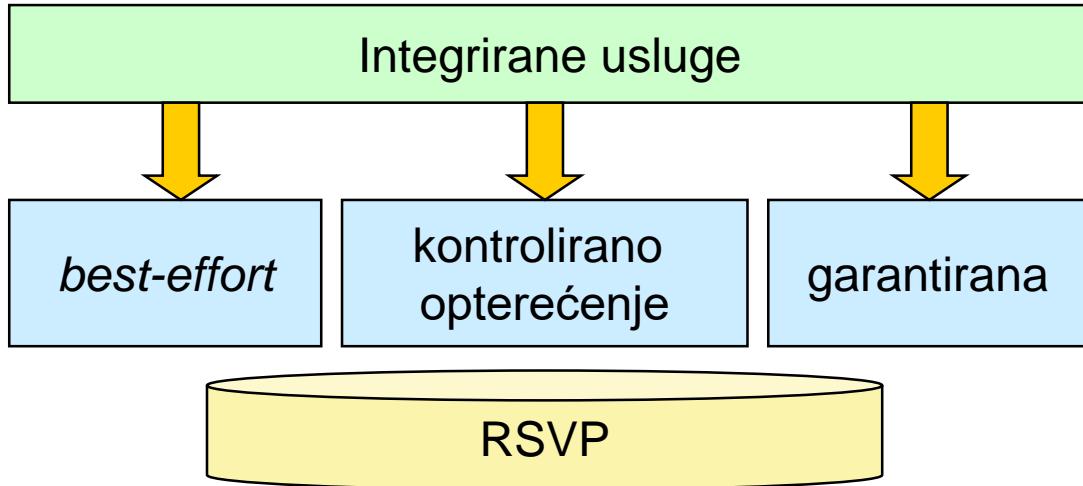


# RSVP i višeodredišno razašiljanje

- slučaj s više odredišta - svaki primatelj šalje svoje rezervacijske zahtjeve
- rezervacijski zahtjevi se spajaju kako putuju uz stablo višeodredišnog usmjerenja
- rezervacija temeljena na primatelju (*receiver-based reservation*) dobro se prilagođuje različitoj veličini skupine (scalability)



# Sažetak



- pristup zasnovan na **rezervaciji**
- aplikacije signaliziraju mreži svoje zahtjeve na QoS koristeći Resource Reservation Protocol (**RSVP**)
- svaki mrežni element (usmjerenitelj) na putu od izvora do odredišta određuje može li prihvatiti zahtjev ili ne, i, ako mreža prihvati novi tok (promatrane aplikacije), mrežni resursi se **rezerviraju za taj tok** po cijeloj stazi s kraja na kraj

# Prednosti i nedostaci

---

- + najveća prednost su rezervacije “po mjeri”, odn. prema prometnim svojstvima toka
- + ostale prednosti:
  - pouzdanost rezervacije
  - prilagodljivost rezervacije
  - moguća dinamička promjena puta kojim tok putuje i promjena rezervacije
  - pogodan za višeodredišne, heterogene skupine
- - najvažniji nedostatak je složenost i *overhead* zbog obrade u svakom usmjeritelju i za svaki tok pojedinačno

---

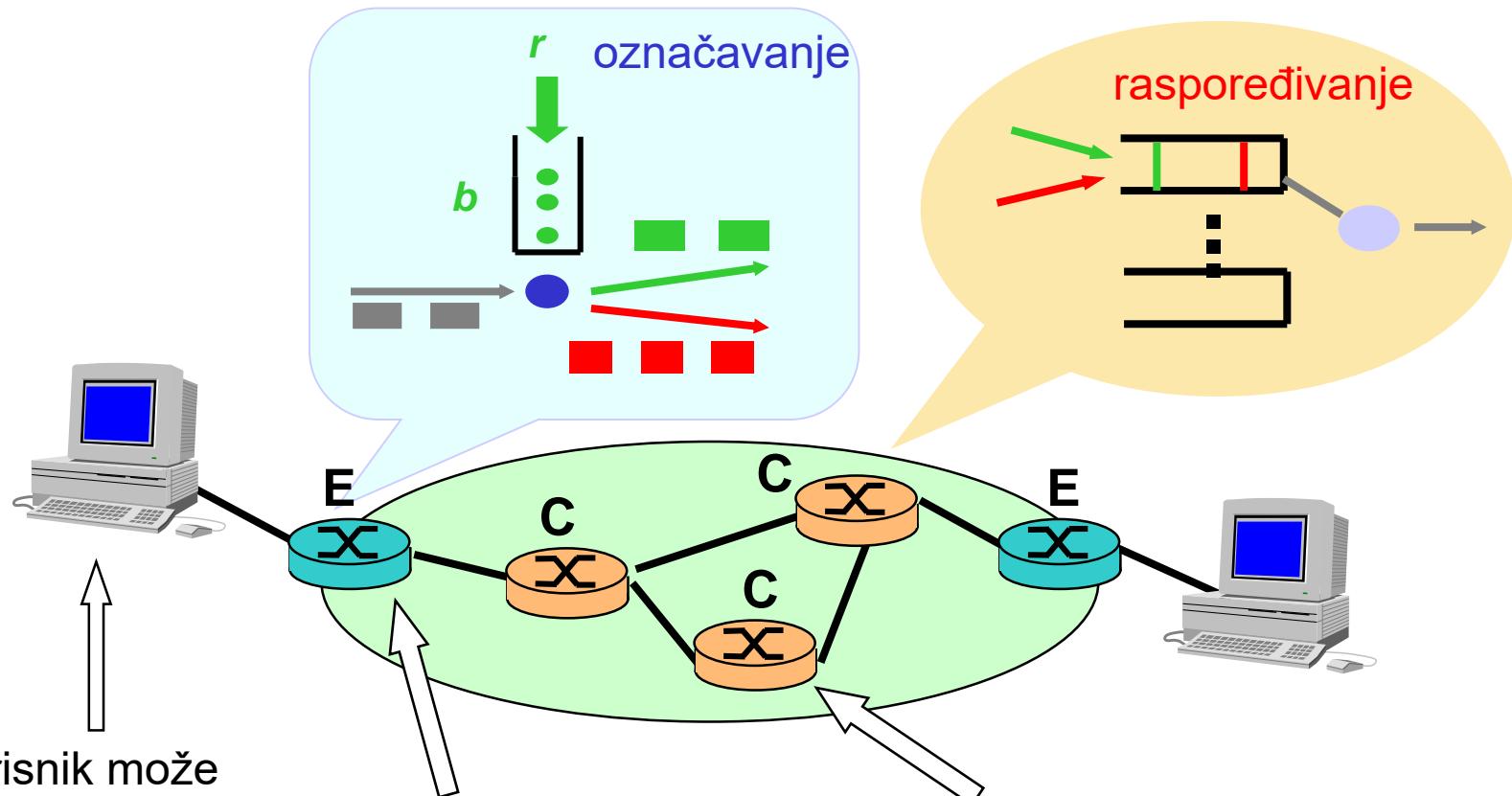
# Diferencirane usluge

# Diferencirane usluge

---

- klasificiraju se agregatni (objedinjeni) tokovi
- usmjeritelj na rubu mreže klasificira ulazni promet
- ponašanje unutar mreže propisuje se za klase prometa s različitim **prioritetima**
- u svakom usmjeritelju, s paketima se postupa prema zadanom ponašanju za klasu kojoj pripadaju (“*per-hop behavior*”, **PHB**).
- nema obrade po toku, tj. svakog paketa u svakom usmjeritelju
- politika korištenja usluge i prometni profil uključeni su u ugovor o razini usluge (Service Level Agreement)

# Arhitektura DiffServ mreže



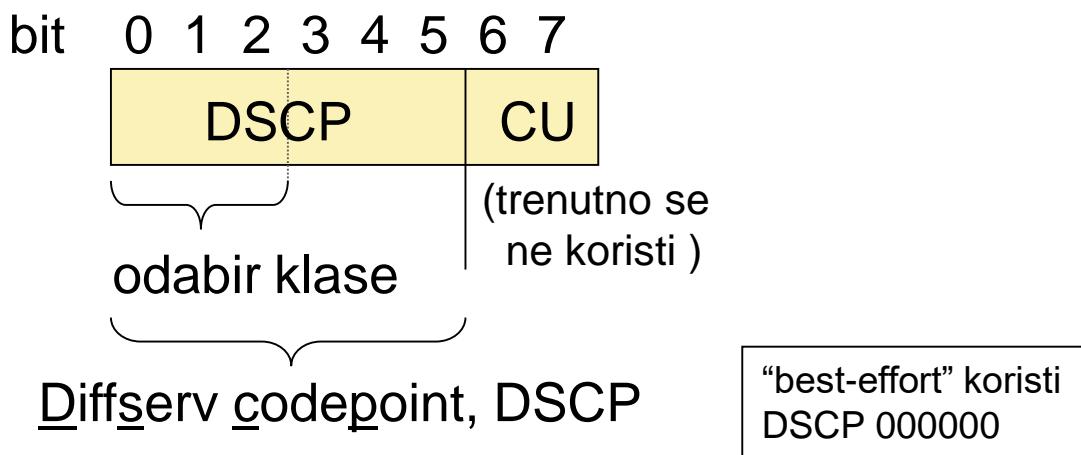
Korisnik može  
pred-označiti  
pakete

Rubni usmjeritelj (E)  
postavlja DS polje  
koje označuje PHB

Usmjeritelj u  
središnjoj mreži (C)  
rukaje s paketima  
prema PHB-u

# Označavanje paketa (1)

- oznaka u paketu (DSCP) definira postupak s paketom u usmjeritelju (*per-hop-behavior, PHB*)
- u svakom usmjeritelju, s paketima se postupa prema zadanom ponašanju za klasu kojoj pripadaju (“*per-hop behavior*”, *PHB*).
- dva definirana PHB su:
  - ubrzano prosljeđivanje (*Expedited Forwarding*)
  - osigurano prosljeđivanje (*Assured Forwarding*)



# Označavanje paketa (2)

- za IPv4 koristi se *Type of Service (TOS)* polje u zaglavlju IP datagrama
- ◆ za IPv6 koristi se *Traffic Class* polje u zaglavlju IPv6 datagrama

		Ver.	Len.	TOS	total length	
identification			flag	fragment offset		
TTL	protocol	checksum				
source IP address						
destination IP address						

		0	8	16	24
		Ver.	traffic class	flow label	
payload length		next Hdr.		hop limit	
source IP address					
destination IP address					

# Ubrzano prosljeđivanje (*Expedited Forwarding*)

- jedinstvena oznaka (*codepoint*)
  - preporučena vrijednost 101110
- osigurava EF paketima nisko kašnjenje i kolebanje kašnjenja i male gubitke
- suma brzina primljenih paketa na ulazu u čvor ne smije biti veća od minimalne brzine transmisije na izlazu → nema gomilanja!
- usluga garantira minimalnu vrijednost propusnosti za AF pakete
- promet koji se ne drži profila se pred-označava kao best effort

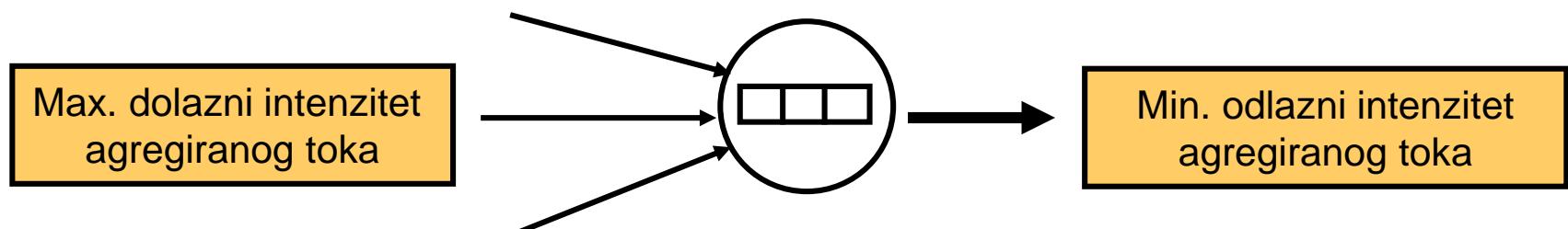
# Osigurano prosljeđivanje (Assured Forwarding)

- četiri klase i tri prioriteta odbacivanja (dvanaest oznaka)
- promet koji se ne drži profila se prosljeđuje s nešto nižom vjerojatnošću od onoga koji se drži profila, ali se ne odbacuje

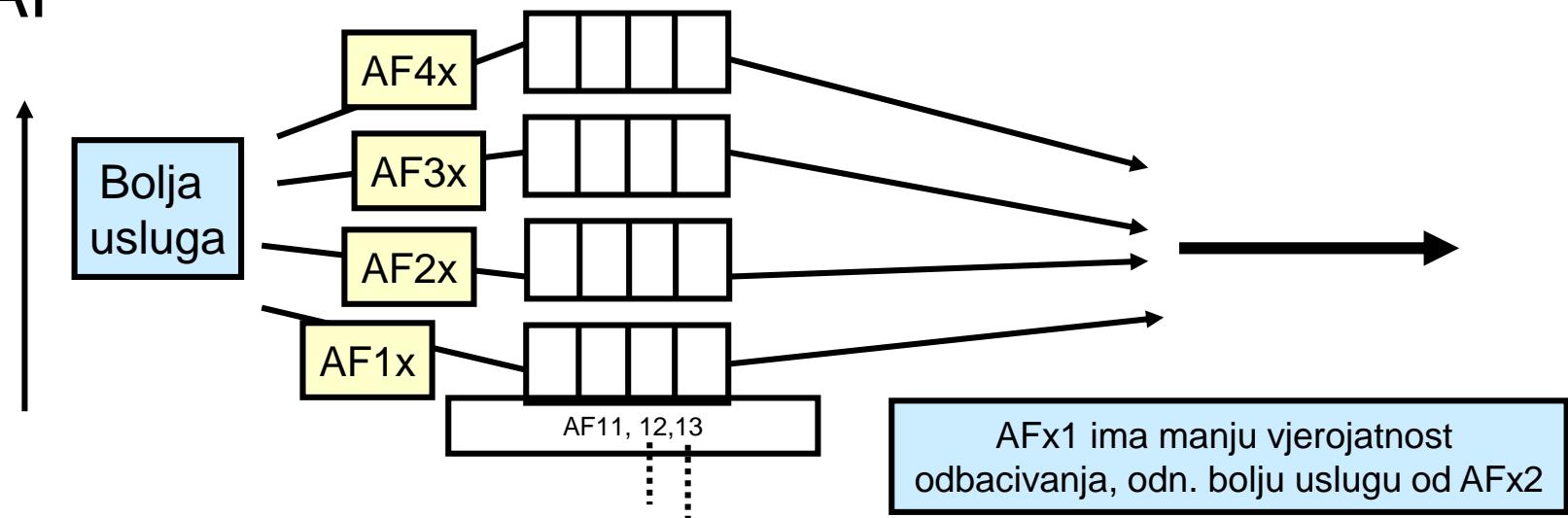
Prioritet odbacivanja	Klasa prosljeđivanja	Klasa AF1	Klasa AF2	Klasa AF3	Klasa AF4
Nizak		001010	010010	011010	100010
Srednji		001100	010100	011100	100100
Visok		001110	010110	011110	100110

# Ponašanje prema skoku

- EF

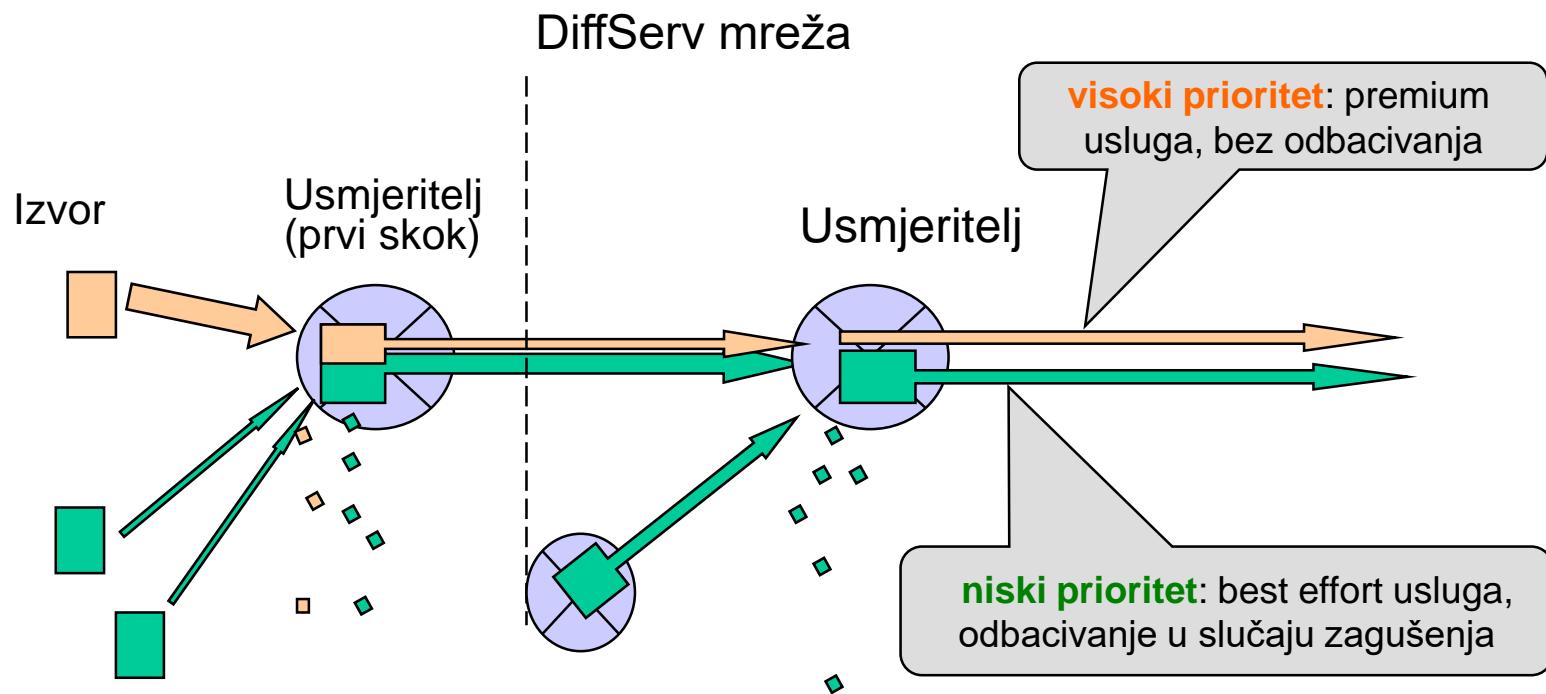


- ◆ AF



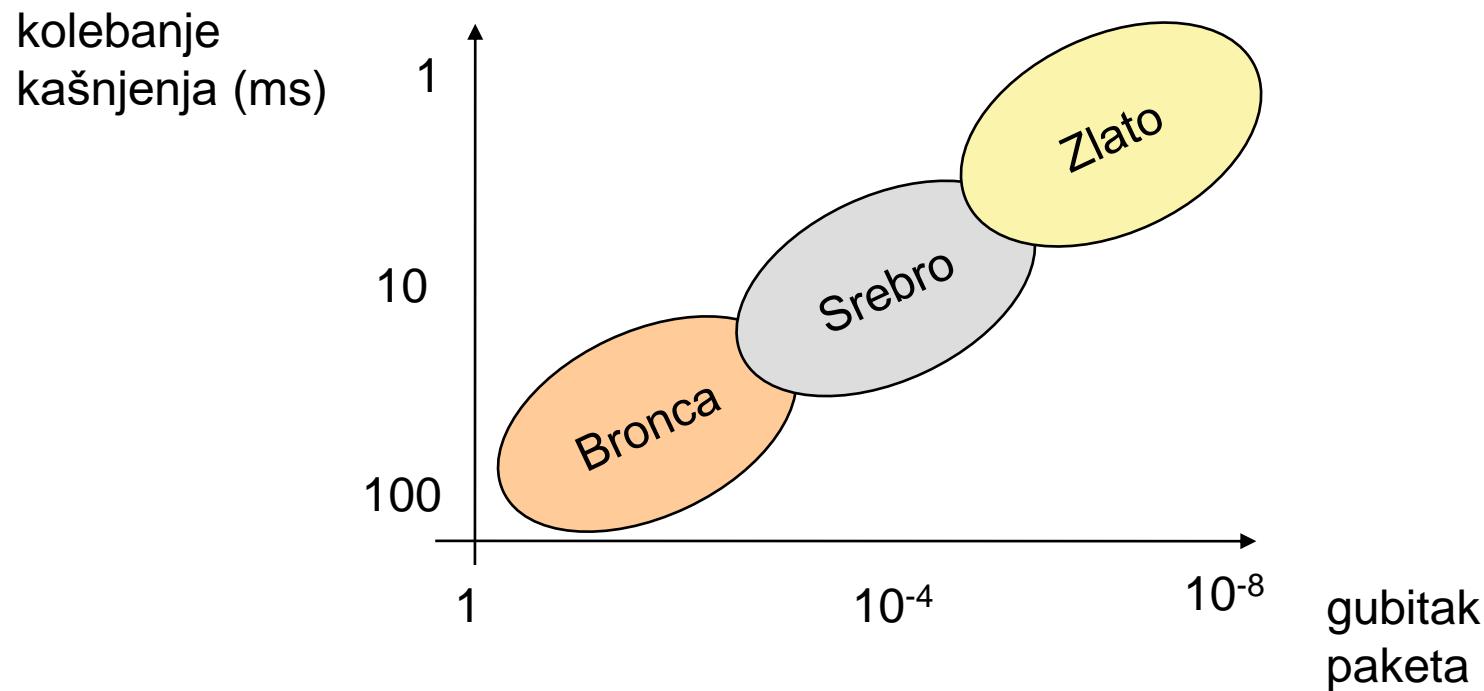
# Primjeri usluga (1)

- Usluga virtualne iznajmljene linije
  - usluga virtualne iznajmljene linije, poznata i kao “Premium service”, kvantificira se vršnom propusnošću
  - može se izvesti pomoću **EF PHB**



# Primjeri usluga (2)

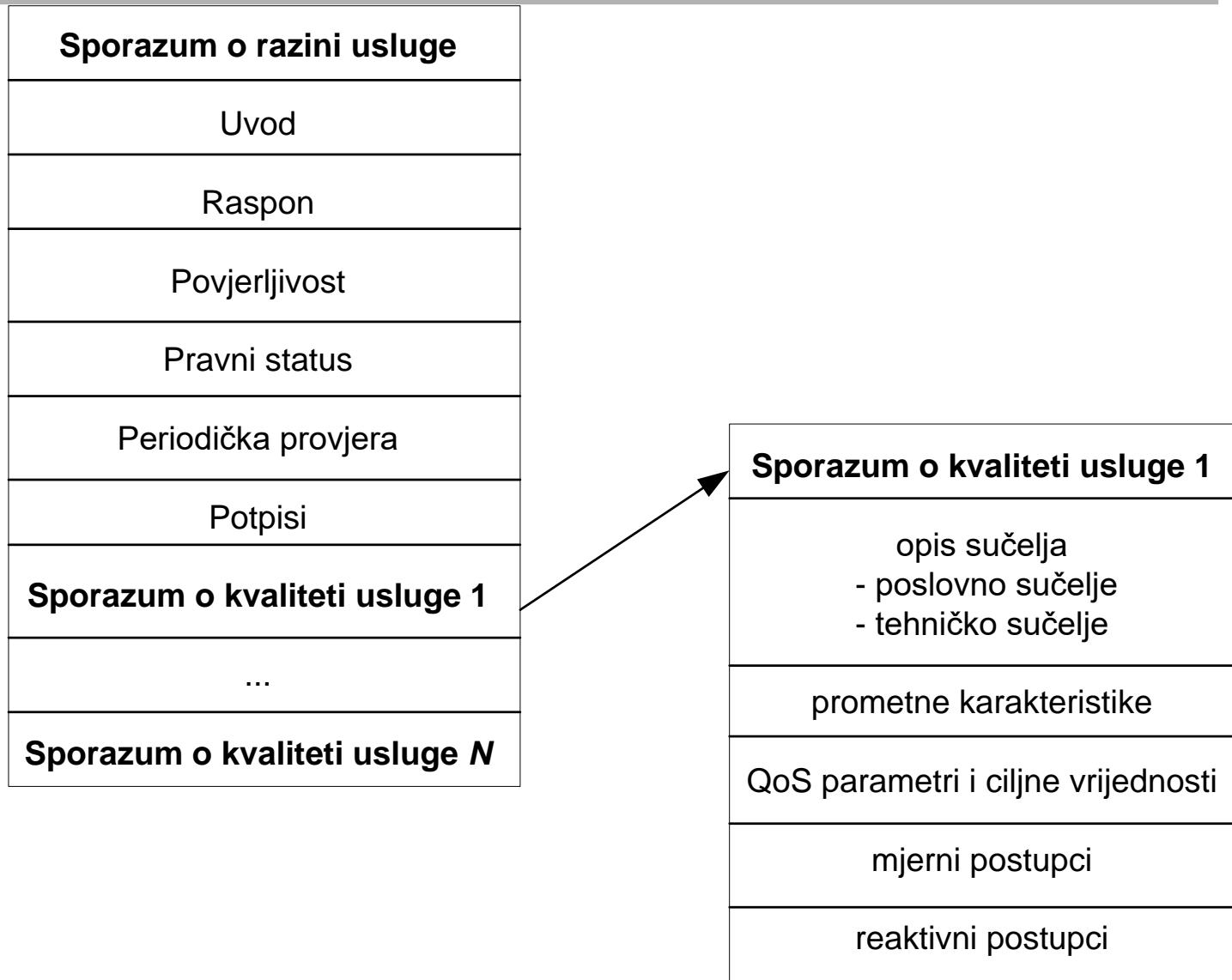
- “Olimpijske usluge”
  - tri klase usluga: zlatna, srebrna, brončana
  - mogu se izvesti pomoću AF PHB



# Pojam SLA

- *Service Level Agreement* (SLA) je ugovor kojeg potpisuju pružatelj internetske usluge i korisnik usluge
  - SLA opisuje i tehničke i administrativne parametre vezane uz uslugu koju korisnik prima
  - *Service Level Specification* (SLS) je skup parametara i njihovih vrijednosti koji zajednički opisuju uslugu koju prometni tok dobiva unutar DiffServ domene.
  - *Traffic Conditioning Specification* (TCS) je skup parametara i njihovih vrijednosti koji zajednički opisuju skup pravila klasifikatora u prometni profil. TCS je integralni dio SLS-a.

# Generička struktura SLA, ITU-T E.860

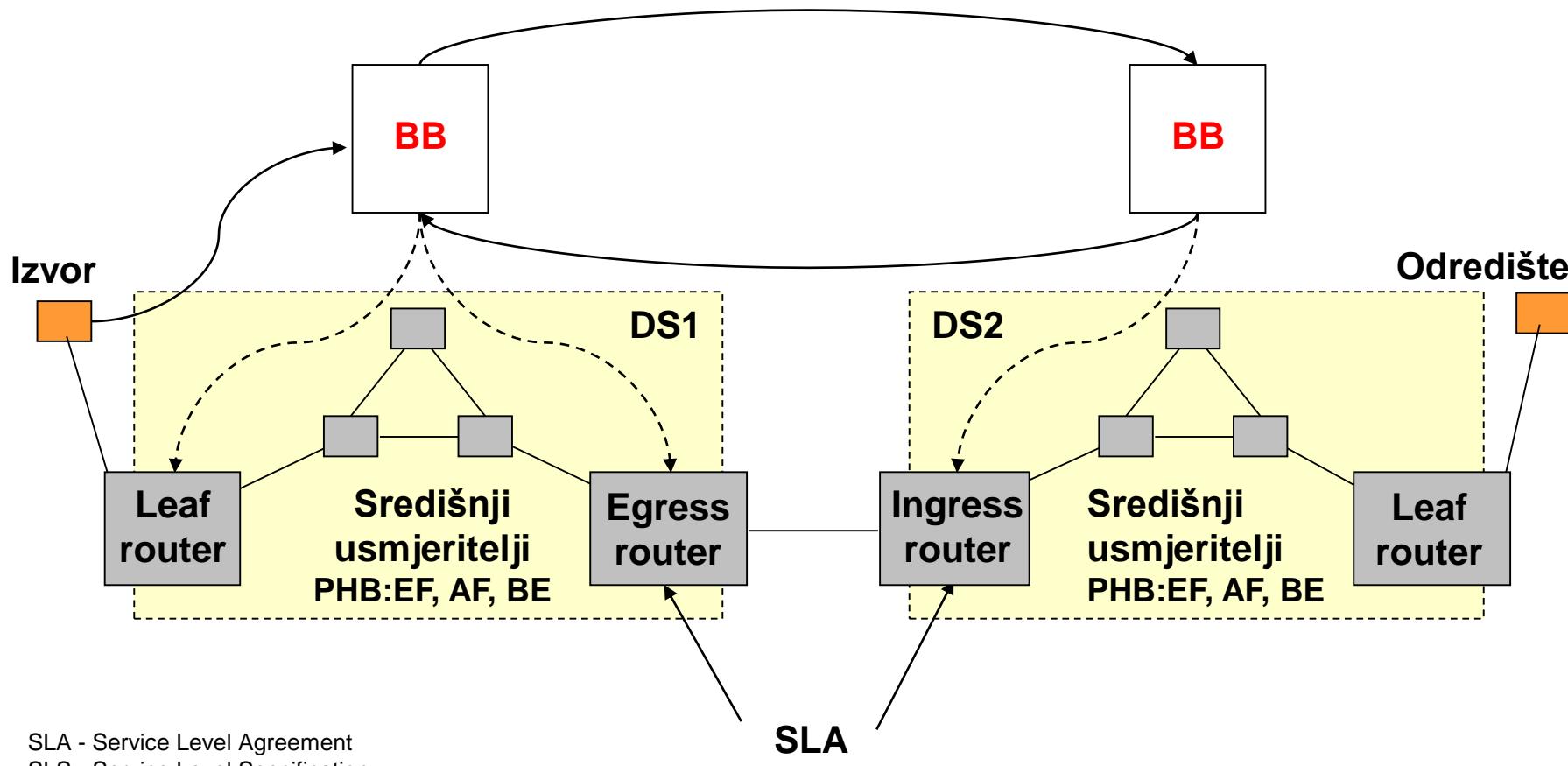


# Primjer SLA za uslugu VPN

Sučelja - poslovno - tehničko	Mjesto izvještavanja – Web stranica	
	SAP – IP adresa, port, broj telefona modemskog ulaza	
	Protokoli – IPv4, BGP4, SNMPPv3	
Karakteristike prometa	Propusnost 8-19 h	384 kbit/s
	Propusnost 19-8h	128 kbit/s
Parametri kvalitete usluge i ciljne vrijednosti	Raspoloživost	99 %
	Gubici	max. 5 %
	Vrijeme uspostave veze	max. 2 s
Mjerni postupci	Mjerne točke	- ulazni usmjeritelj
	Mjerne metode	- aktivne probe
	Alati	- ping, traceroute
	Granularnost	- 10 minuta
Reaktivni postupci	Popust	- ako davatelj usluge smanji raspoloživost, npr. ukoliko usluga nije raspoloživa između 1 h i 4 h, korisnik dobiva 1 besplatni dan korištenja
	Preoblikovanje prometa Alarm Upozorenje Aktiviranje politike	- ako korisnikov promet premaši 1 Mbit/s na ulaznom usmjeritelju, promet se odbacuje

# DiffServ arhitektura s više mreža

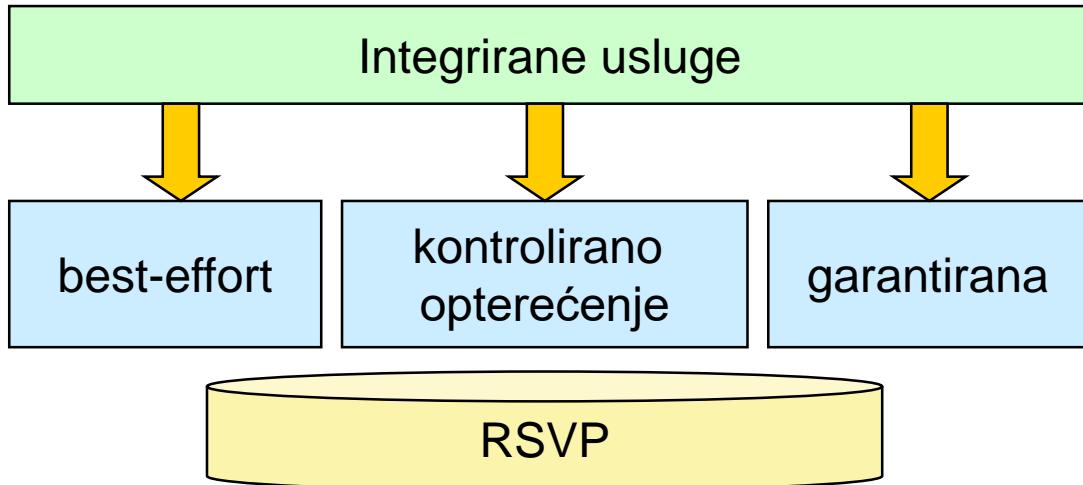
- BB – bandwidth broker – dogovor između DS mreža (domena)



---

# Sažetak i usporedba IntServ i DiffServ pristupa

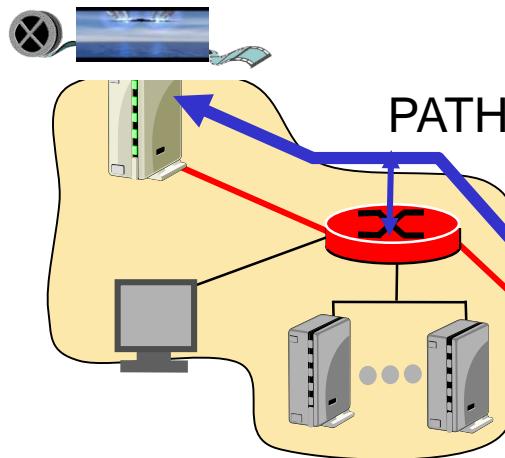
# Sažetak: Integrirane usluge (1)



- pristup zasnovan na **rezervaciji**
- aplikacije signaliziraju mreži svoje zahtjeve na QoS koristeći Resource Reservation Protocol (**RSVP**)
- svaki mrežni element (usmjerenitelj) na putu od izvora do odredišta određuje može li prihvatiti zahtjev ili ne, i, ako mreža prihvati novi tok (promatrane aplikacije), mrežni resursi se **rezerviraju za taj tok** po cijeloj stazi s kraja na kraj

# Sažetak: Integrirane usluge (2)

pošiljatelj (izvor)

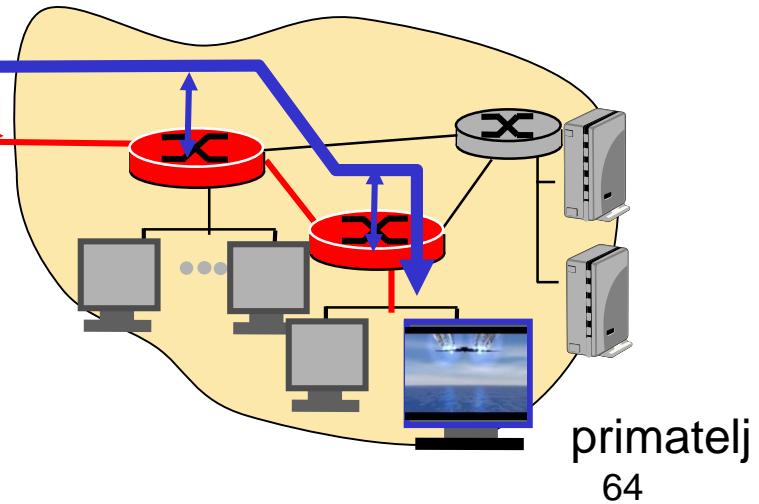


PATH/RESV

- **Rezervacija resursa**

- uspostava poziva, signalizacija (RSVP)
- opis prometa, zahtjev za QoS
- kontrola prihvata u svakom mrežnom elementu

raspoređivanje  
prema QoS  
(npr, WFQ)



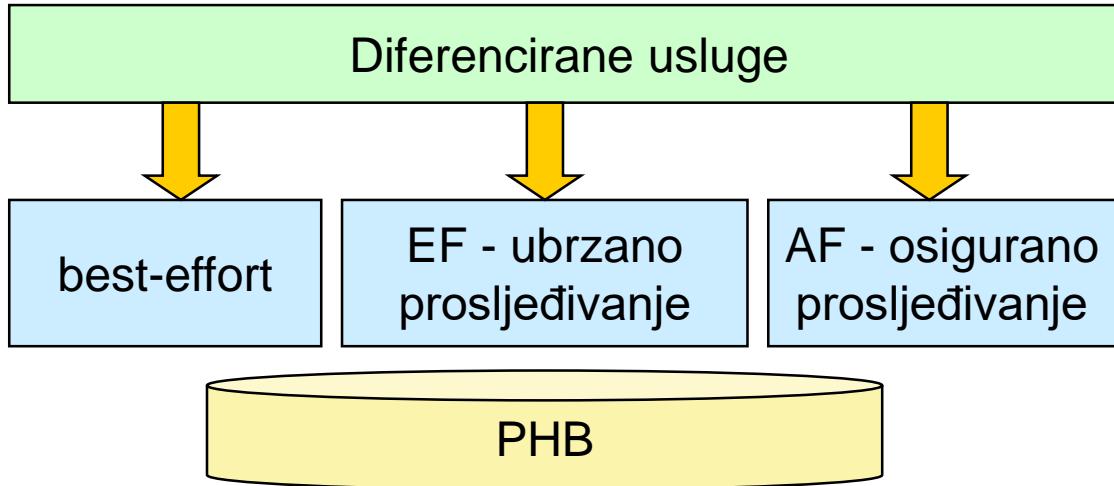
primatelj  
64

# Sažetak: Integrirane usluge (3)

---

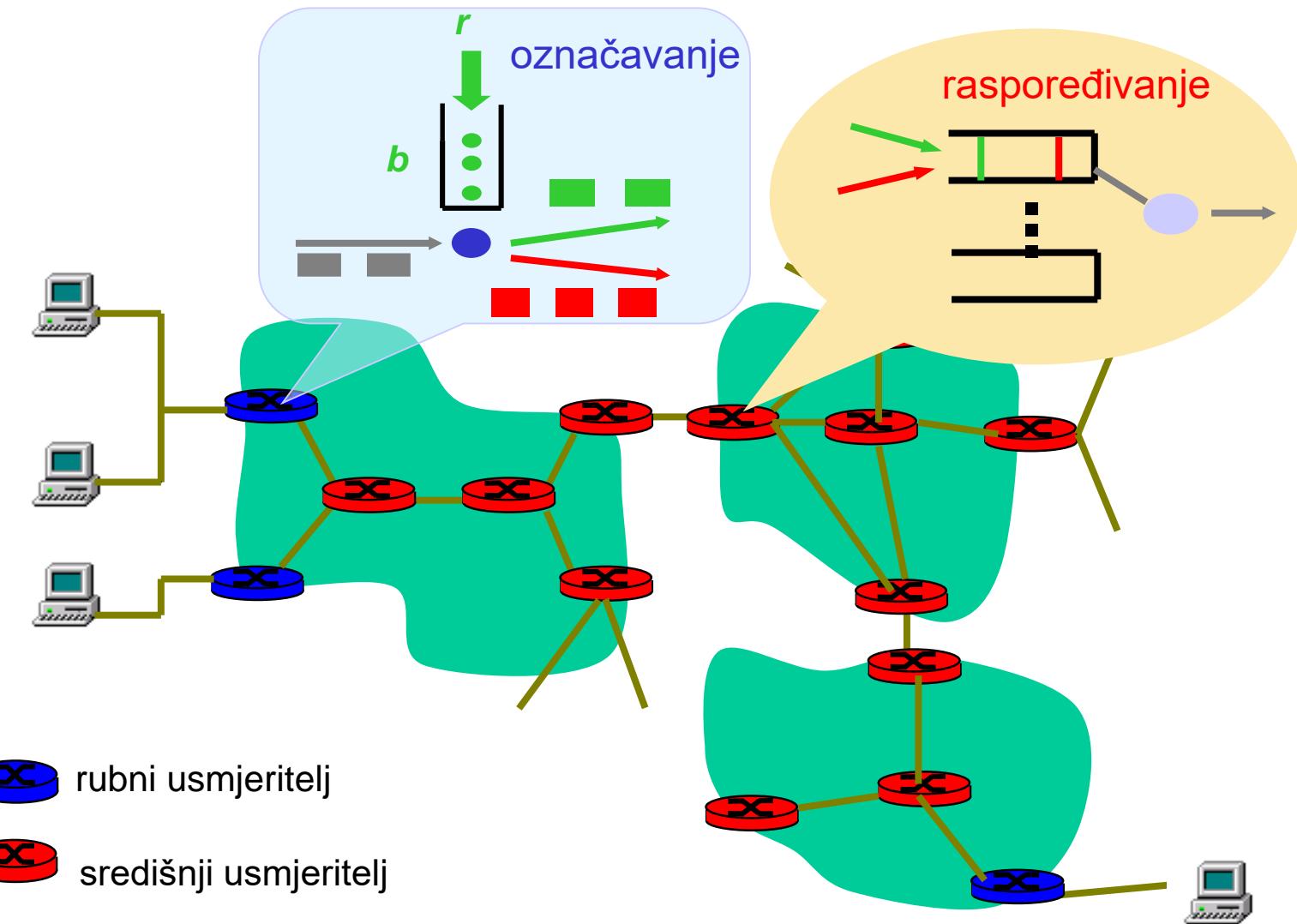
- podaci o stanju za svaki tok pohranjuju se i održavaju u svakom mrežnom elementu na stazi
  - nije prilagodljivo veličini, neizvedivo za cijeli Internet
  - može se primijeniti u poslovnim mrežama gdje je broj tokova za koje se traži rezervacija relativno mali
- vrlo složeni postupci
  - provjera politike, dozvola, naplata
  - kontrola prihvata
  - administracija politika

# Sažetak: Diferencirane usluge (1)



- grublja podjela na **uslužne klase** uz uvođenje **prioriteta**
- klasifikacija i označavanje pojedinih paketa promatrane aplikacije oznakom klase usluge kao Diffserv codepoint, DSCP
- korištenje proširenog TOS (IPv4) ili Class (IPv6) polja u zaglavljima IP paketa za zapis klase usluge za paket
- uređaji mrežnog sloja koriste DSCP paketa kao uputu za postupak s paketom

# Sažetak: Diferencirane usluge (2)

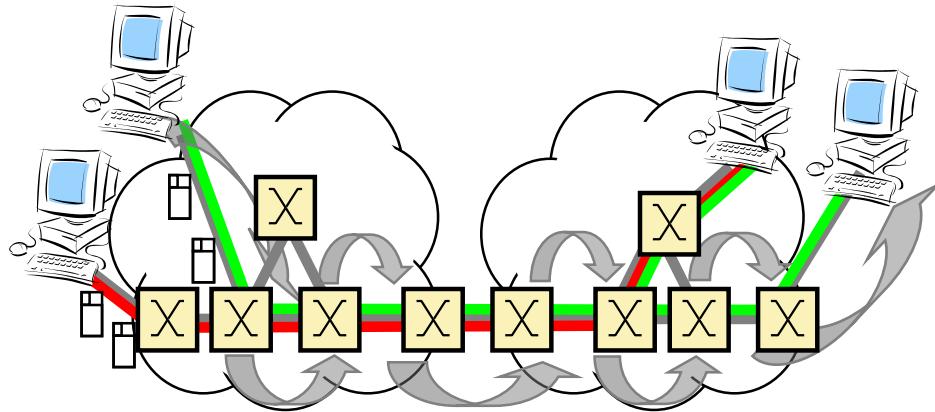


# Sažetak: Diferencirane usluge (3)

---

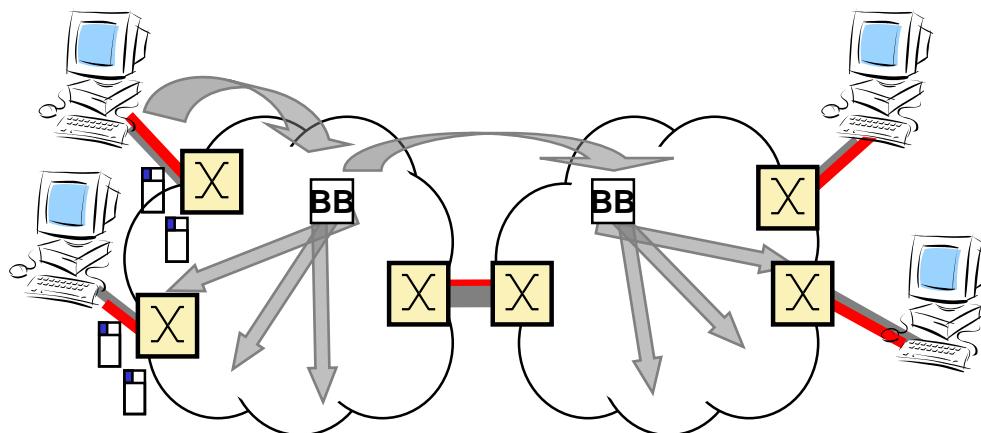
- kombinira malen broj klasa usluga s bogatim skupom mogućnosti upravljanja prometom
  - oblikovanje i redarstvo ovisno o klasi
  - postavljanje u rep prema prioritetu
  - Weighted Fair Queuing (WFQ)
  - mehanizmi za izbjegavanje zagušenja
- nema potrebe za čuvanjem stanja u svakom usmjeritelju kao ni signalizacijom
- pogodan za velike mreže, odn. veliki broj tokova (npr. Internet)

# Usporedba integriranih i diferenciranih usluga



## Integrirane usluge (RSVP)

- usluga za tok, obrada na svakom skoku
- problem s veličinom
- usmjerenost na višeodredišno razašiljanje



## Diferencirane usluge

- postupak s paketima prema klasi
- agregatni, a ne pojedinačni tokovi
- nadzor na rubovima mreže

# Kombiniranje IntServ i DiffServ pristupa

- IntServ s kraja na kraj nije prilagodljiv veličini zbog čuvanja podataka o toku u svakom usmjeritelju na stazi
- DiffServ nema dovoljno finu podjelu za klasifikaciju aplikacija
- Rješenje:
  - korištenje DiffServ u središnjoj mreži
  - korištenje IntServ na krajevima mreže
  - primijeniti principe upravljanja prometom i kontrolu pristupa kako bi se osigurala kvaliteta s kraja na kraj

# Priprema za iduće predavanje

---

- Pročitajte sljedeće članke

- Činjenice o industriji igara

[http://www.theesa.com/wp-content/uploads/2017/06/!EF2017\\_Design\\_FinalDigital.pdf](http://www.theesa.com/wp-content/uploads/2017/06/!EF2017_Design_FinalDigital.pdf)

- Što je cloud gaming?

<http://www.howtogeek.com/160851/htg-explains-what-is-cloud-gaming-and-is-it-the-future/>



# **Višemedijske komunikacije**

12. Višekorisničke umrežene okoline.

# Sadržaj predavanja



- ◆ Umrežena virtualna okruženja
- ◆ Umrežene višekorisničke igre

# Sadržaj predavanja

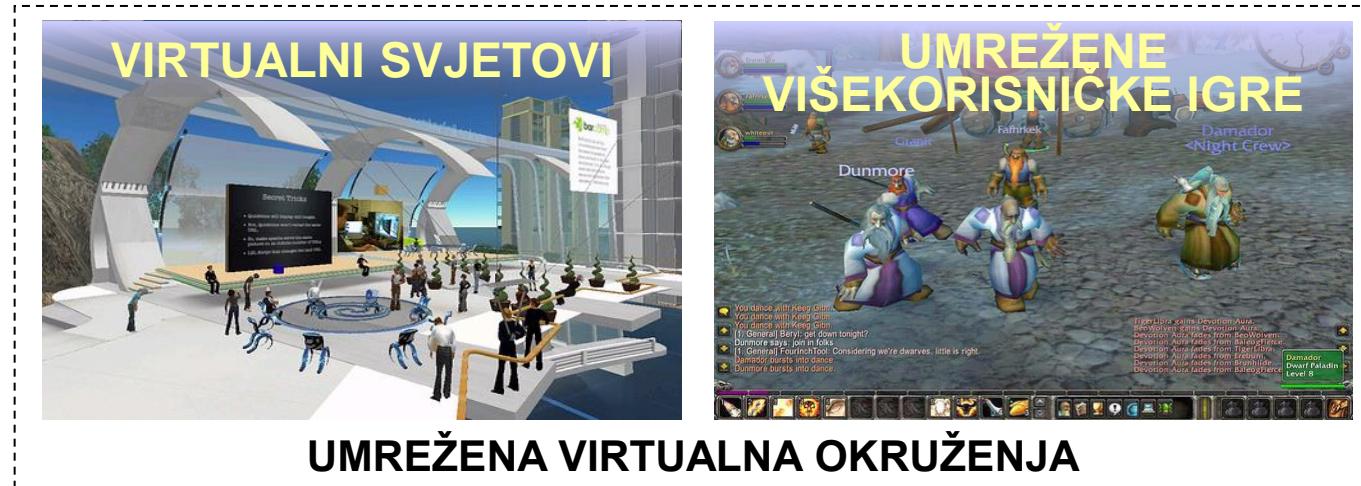
- ◆ Umrežena virtualna okruženja
- ◆ Umrežene višekorisničke igre



Više na predmetima **Osnove virtualnih okruženja**  
**(preddiplomski studij)** i **Virtualna okruženja** (**diplomski studij**)

# Umrežena virtualna okruženja (UVO)

- ◆ engl. *Networked Virtual Environments*
- ◆ Primjene
  - virtualni svjetovi – 3D okolina je dinamična
    - druženje
    - virtualna telekonferencija/zajednički rad
    - marketing, prodaja
    - učenje/obuka na daljinu
  - višekorisničke igre – 3D okolina je predefinirana

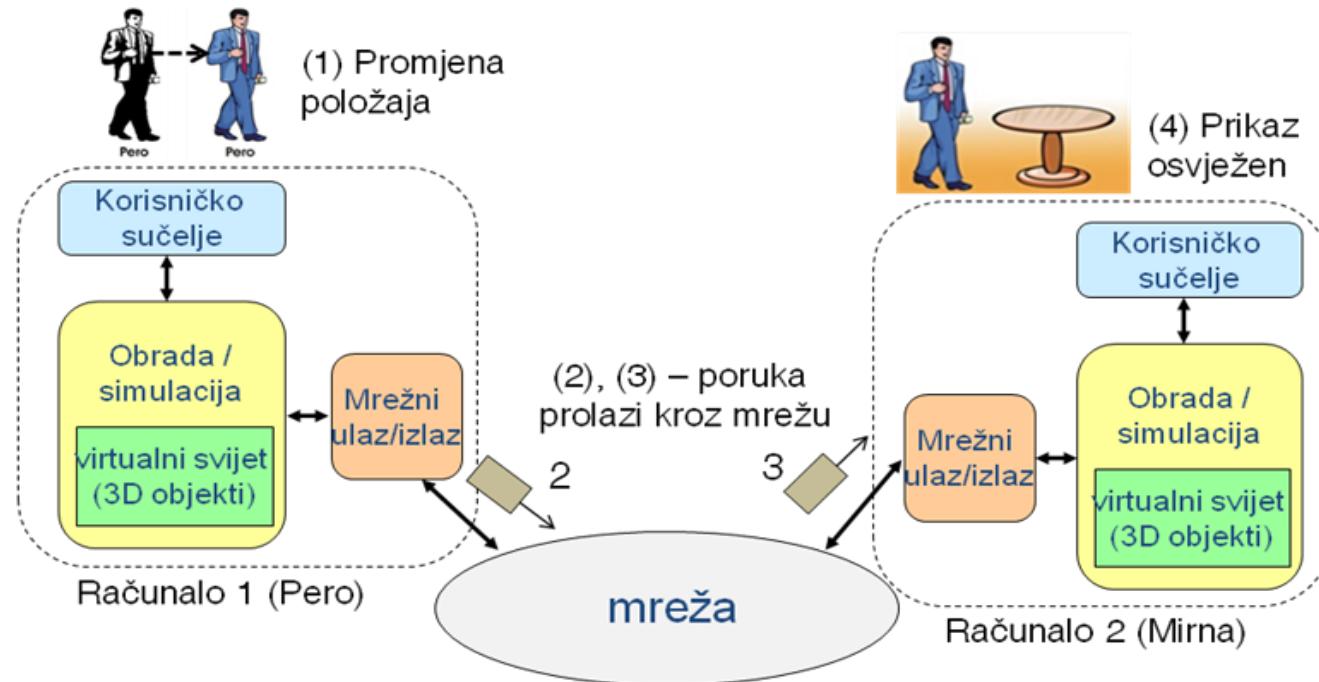


# Razlike između igara i virtualnih svjetova



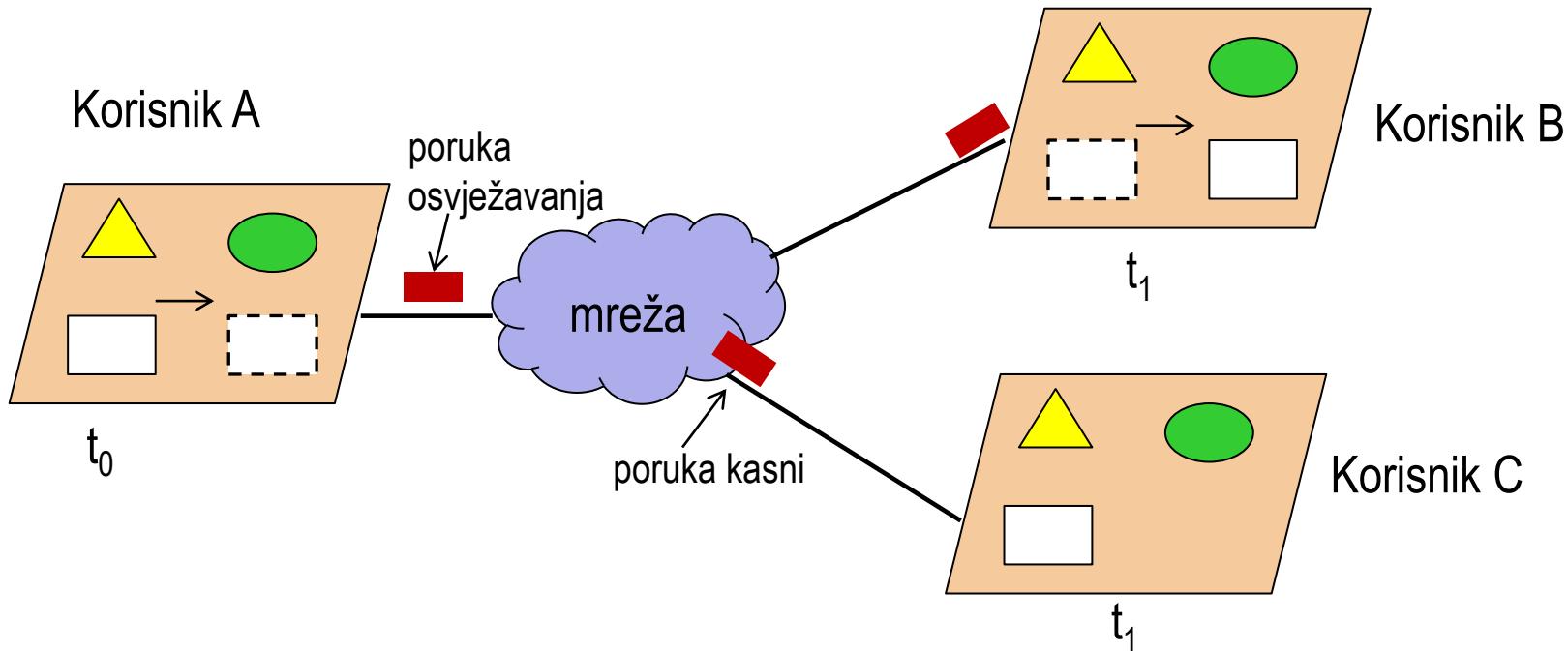
- ◆ Svrha
- ◆ Perzistentnost
  - Virtualni svjetovi su uglavnom perzistentni – postoje i mijenjaju se i dok korisnik nije u njima
  - Kod igara je puno češće iznova stvaranje virtualnog svijeta iz pohranjenih podataka (primjerice svaka bitka u Call of Duty se odvija na istoj mapi koja se iznova kreira)
- ◆ Mijenjanje 3D svijeta
  - Virtualni svjetovi dopuštaju dodavanje novih slika, tekstura, zvukova, videa i 3D objekata (primjerice Second Life)
  - Igre ne dopuštaju dodavanje novih informacija u svoj klijent (iako mogu dopuštati mijenjanje svijeta primjerice Fortnite)
- ◆ Karakteristike mrežnog prometa – jako ovisne o „promjenjivosti“ 3D svijeta jer ako korisnici mogu dodati nove informacije, drugi korisnici moraju te iste informacije preuzeti

# Usklađivanje instanci UVÖ putem mreže



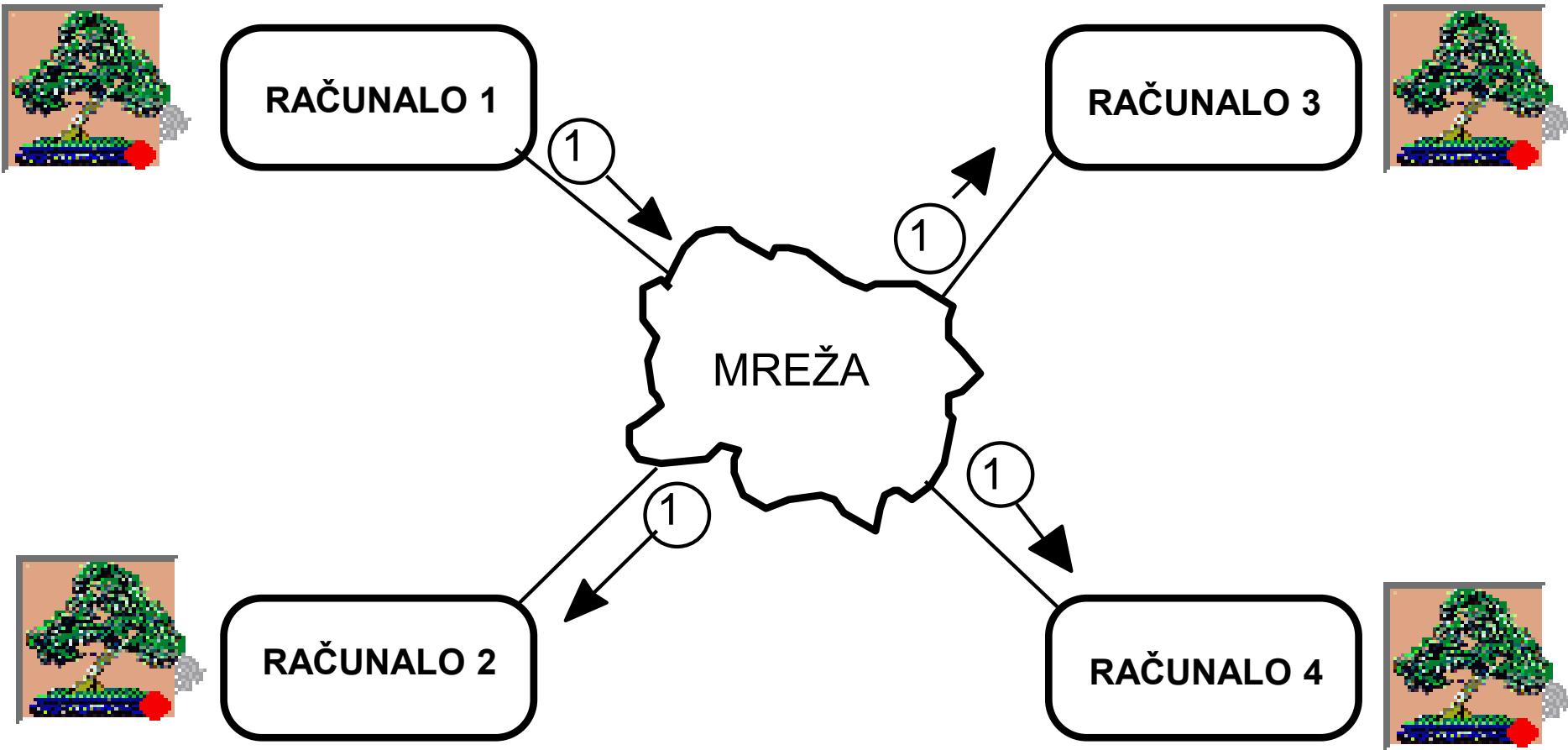
- ◆ Svaki korisnik ima pokrenutu instancu VO
- ◆ Promet raste s brojem korisnika
- ◆ Ne moramo slati elemente koji se ne mijenjaju (npr. teksture, gotovi 3D objekti)
- ◆ Kašnjenje nije isto za sve korisnike

# Problem nekonzistentnosti

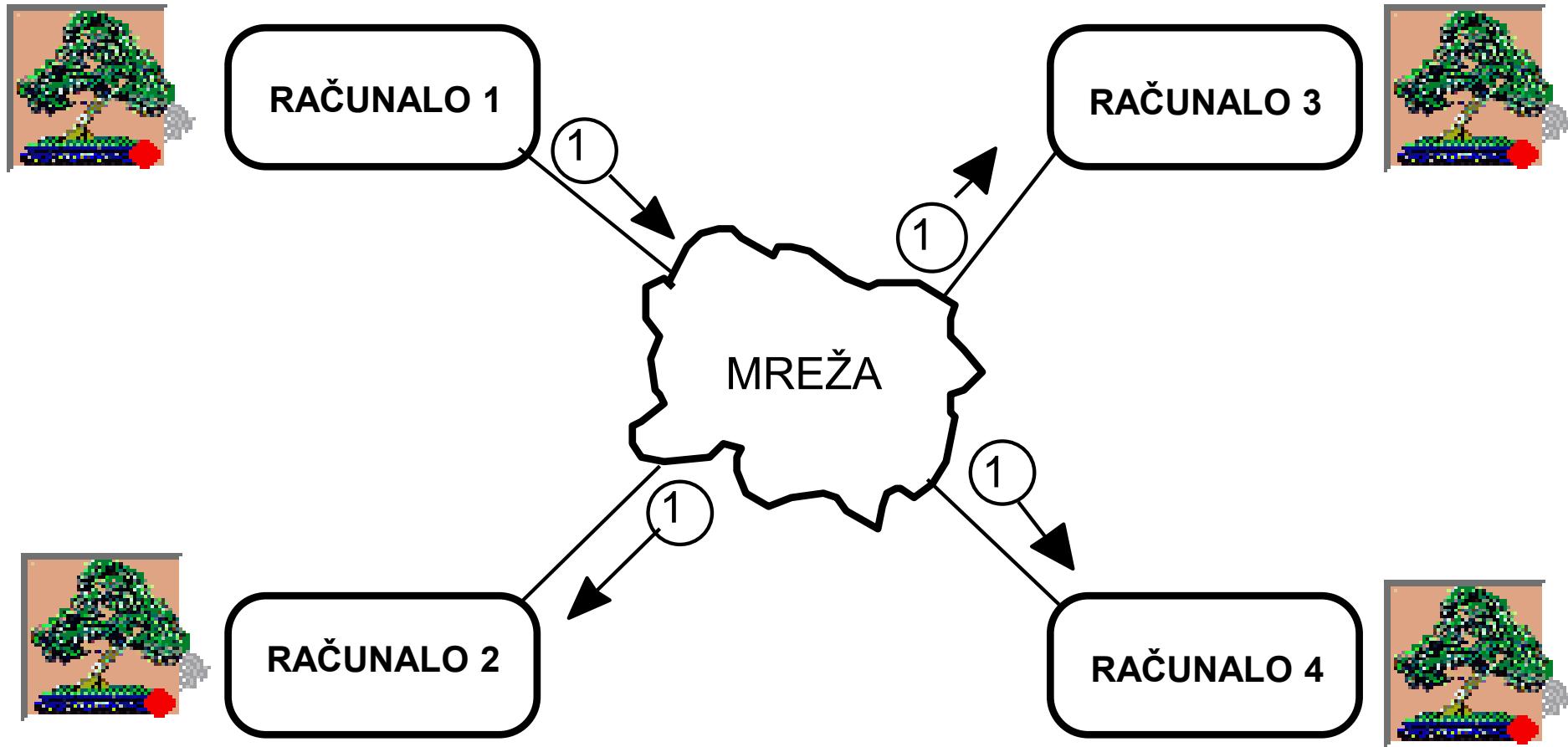


- ◆ Ključni izazov: održavanje raspodijeljenog zajedničkog stanja konzistentnim, ili približno konzistentnim, uz promjene koje nastaju (asinkrono, dinamički i raspodijeljeno).
- ◆ Zahtjeve na konzistentnost treba uskladiti s namjenom aplikacije
- ◆ Češće slanje poruka osvježavanja skraćuje potencijalni vremenski interval u kojem može doći do odstupanja, ali više opterećuje mrežu i proces za slanje i primanje poruka

# Izvršavanje odmah – nekonzistentno, ali brzo



# Odgodeno izvršavanje - konzistentno



# Sadržaj predavanja



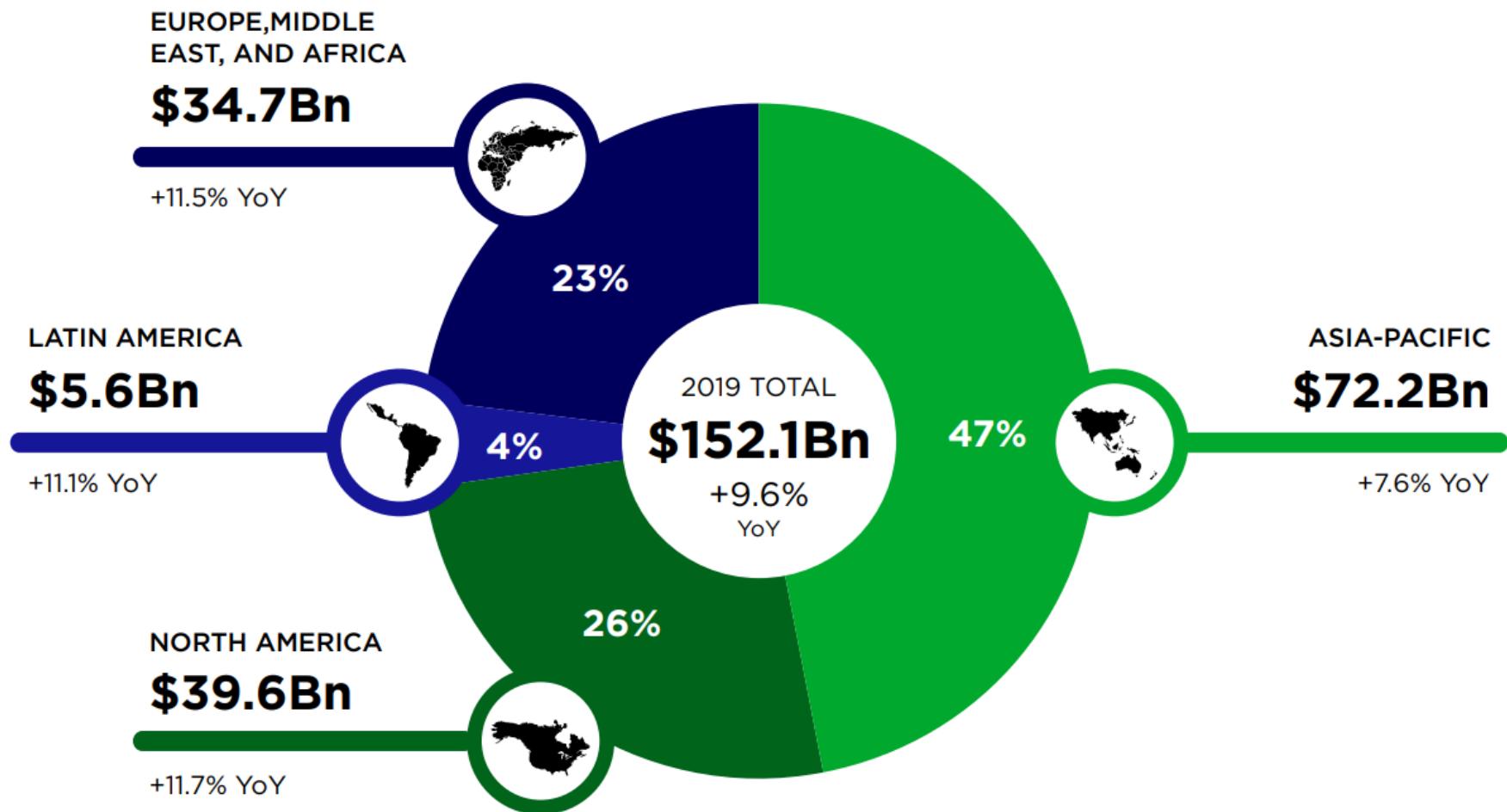
- ◆ Umrežena virtualna okruženja
- ◆ Umrežene višekorisničke igre

- ◆ Digitalne igre - igre koje se igraju pomoću računala ili igraće konzole
  - “Video games” – igre na konzoli
  - “Computer games” – igre na računalu
- ◆ Višekorisničke igre – više od jednog igrača može istovremeno igrati u istoj virtualnoj okolini
- ◆ Umrežene igre – potrebna mrežna povezanost
- ◆ Fokus predavanja – umrežene višekorisničke igre

- ◆ Klasifikacija igara (po mehanizmima igre):
  - Akcije (npr. *Mortal Combat*)
  - Pucačine (engl. *shooters*) (npr. *Quake*, *Call of Duty*)
  - Avanture (npr. *Broken Sword*)
  - Igre preuzimanja uloga (engl. *Role-Playing Game*, RPG) (npr. *WoW*)
  - Sportske igre (npr. *Pro Evolution Soccer*)
  - Trkaće igre (npr. *Need for Speed*)
  - Simulacije (npr. *Sim City*)
  - Slagalice i logičke igre (npr. *Mah Jong*)
  - Strategije (npr. *Starcraft*)
  - Ostale

# Tržište igara – procjene 2020 (prvi dio godine)

FER



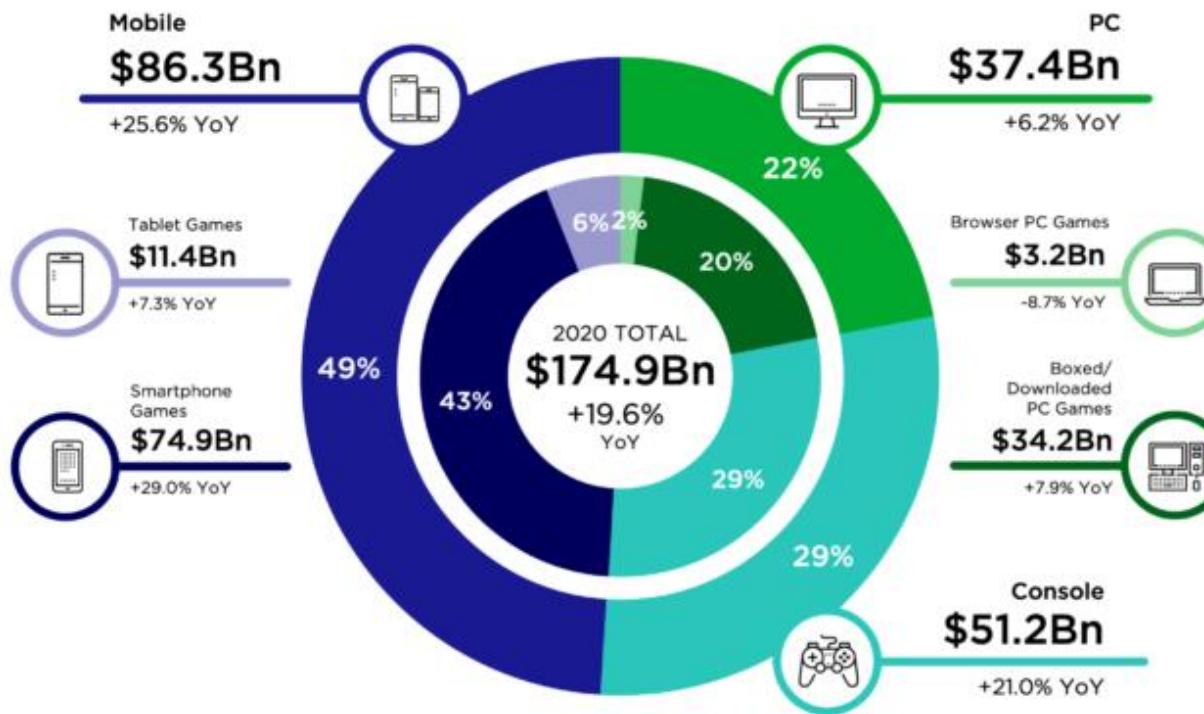
# Tržište igara – procjene 2020 (drugi dio godine)

FER



## 2020 Global Games Market

Per Device & Segment With Year-on-Year Growth Rates



**\$86.3Bn**

Mobile game revenues in 2020 will account for 49% of the global market

Global Games Market Report  
Now available in the Newzoo Platform



[newzoo.com/globalgamesreport](http://newzoo.com/globalgamesreport)

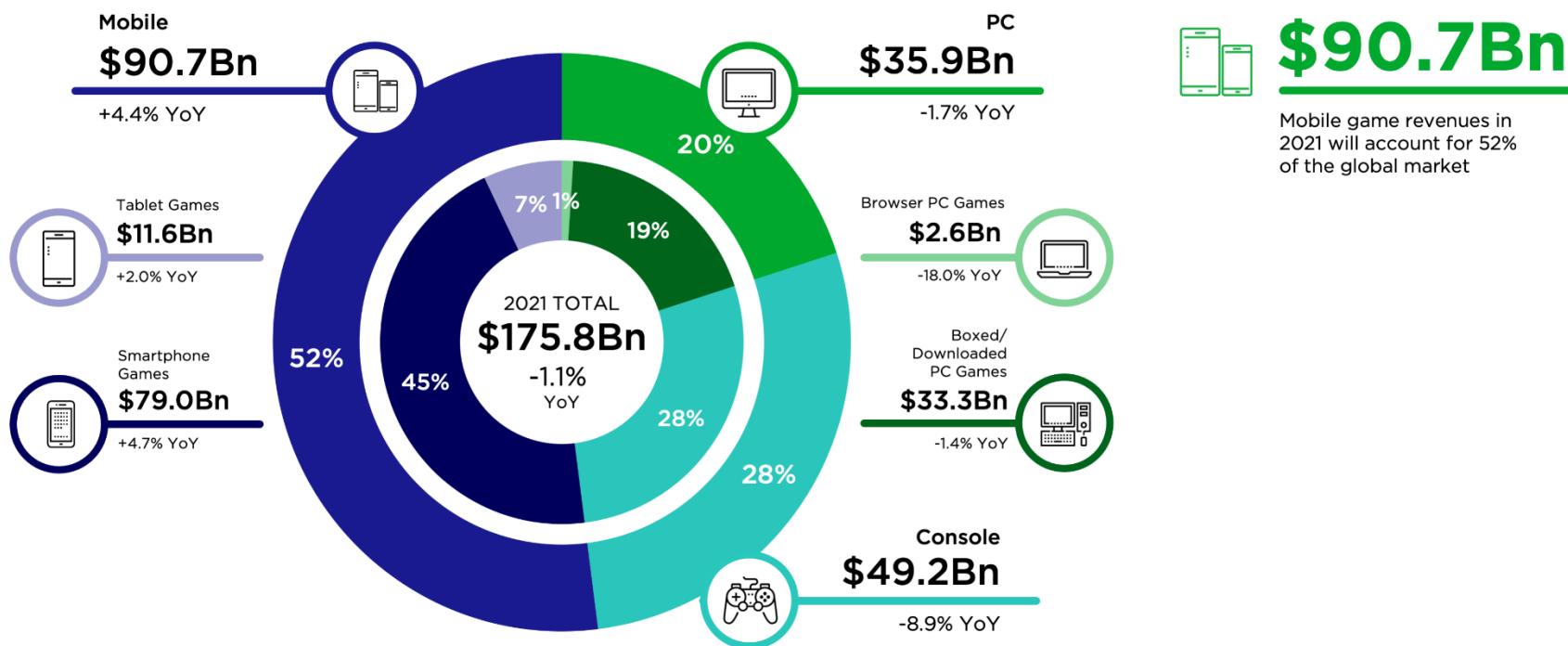
Source: ©Newzoo | 2020 Global Games Market Report | October Update

[newzoo.com/globalgamesreport](http://newzoo.com/globalgamesreport)



## 2021 Global Games Market

Per Device & Segment With Year-on-Year Growth Rates



Source: ©Newzoo | Global Games Market Report | April 2021

[newzoo.com/globalgamesreport](http://newzoo.com/globalgamesreport)

# Porast korištenja igara preko Interneta

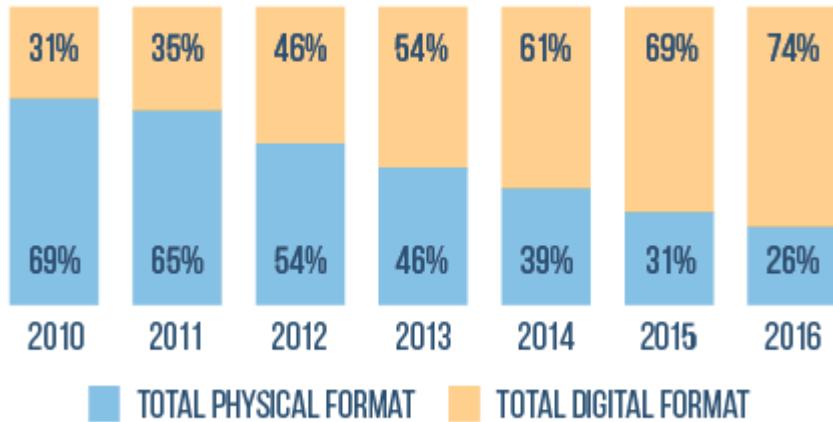


- ◆ Više igrača
- ◆ Zaštita autorskih prava (Digital rights management – DRM)
- ◆ Društvene igre
- ◆ Igre na pokretnim uređajima
- ◆ Distribucija sadržaja
- ◆ Igre na konzolama (XboX one – 300 000 poslužitelja, prethodna generacija XboX live samo 30 000)



## RECENT DIGITAL\* AND PHYSICAL SALES INFORMATION

\*Digital format sales include subscriptions, digital full games, digital add-on content, mobile apps, and social network games.



Source: The NPD Group

**Peak concurrent players**

League of Legends – 7,5 million  
Dota 2 – 1,2 million  
World of Tanks – 1,1 million  
Steam – 9,4 (3 in game) million

# Mobilne igre – veliki rast

FER

## Top Grossing iPhone - Games UNITED STATES

#	FREE	PAID	GAME	PUBLISHER	PRICE	REVENUE	NEW INSTALLS
#	FREE	PAID	GAME	PUBLISHER	PRICE	REVENUE	NEW INSTALLS
1	-	-	Fortnite	Epic Games	Free	\$1,451,543	12,479
2	-	-	Candy Crush Saga	King	Free	\$1,407,996	15,346
3	-	-	Pokémon GO	Niantic, Inc.	Free	\$1,113,188	16,059
4	-	-	Clash of Clans	Supercell	Free	\$761,353	14,802
5	-	-	Brawl Stars	Supercell	Free	\$738,513	16,462
6	-	-	Toon Blast	Peak Games	Free	\$653,109	14,257
7	-	-	Golf Clash	Playdemic	Free	\$344,148	15,485
8	-	-	ROBLOX	Roblox Corporation	Free	\$317,298	16,232
9	-	-	Slotomania™ Vegas Casino Slots	Playtika Ltd	Free	\$277,483	11,843
10	-	-	MARVEL Contest of Champions	Kabam	Free	\$178,757	12,758

1,4 miliona \$ DNEVNO –  
samo na iPhoneu u SAD-u

Mobilne igre – 52 % tržišta danas!



## 2012-2021 GLOBAL GAMES MARKET

REVENUES PER SEGMENT 2012-2021 WITH COMPOUND ANNUAL GROWTH RATES



Source: ©Newzoo | April 2018 Quarterly Update | Global Games Market Report  
newzoo.com/globalgamesreport

newzoo



# Zašto igre na mobitelima idu online?

FER

## Najprodavanije igre

Pogledajte više



Minecraft  
Mojang

★★★★★ 59,00 HRK



Grand Theft Auto: S  
Rockstar Games

★★★★★ 59,99 HRK



Evertale  
ZigZaGame Inc.

★★★★★ 6,60 HRK



RFS - Real Flight Si  
RORTOS

★★★★★ 7,00 HRK



Terraria  
505 Games Srl

★★★★★ 41,00 HRK



Hitman Sniper  
SQUARE ENIX Ltd

★★★★★ 8,00 HRK



Bloons TD 6  
ninja kiwi

★★★★★ 40,00 HRK



Monopoly - Board g  
Marmalade Game Stud

★★★★★ 33,00 HRK



Ultimate Custom N  
Clickteam USA LLC

★★★★★ 26,00 HRK

## Igre s najvećim prihodom

Pogledajte više



Coin Master  
Moon Active

★★★★★



Brawl Stars  
Supercell

★★★★★



Empires & Puzzles  
Small Giant Games

★★★★★



Gardenscapes  
Playrix

★★★★★



PUBG MOBILE - Ma  
Tencent Games

★★★★★



Roblox  
Roblox Corporation

★★★★★



State of Survival: St  
KingsGroup Holdings

★★★★★



Lords Mobile: King  
IGG.COM

★★★★★



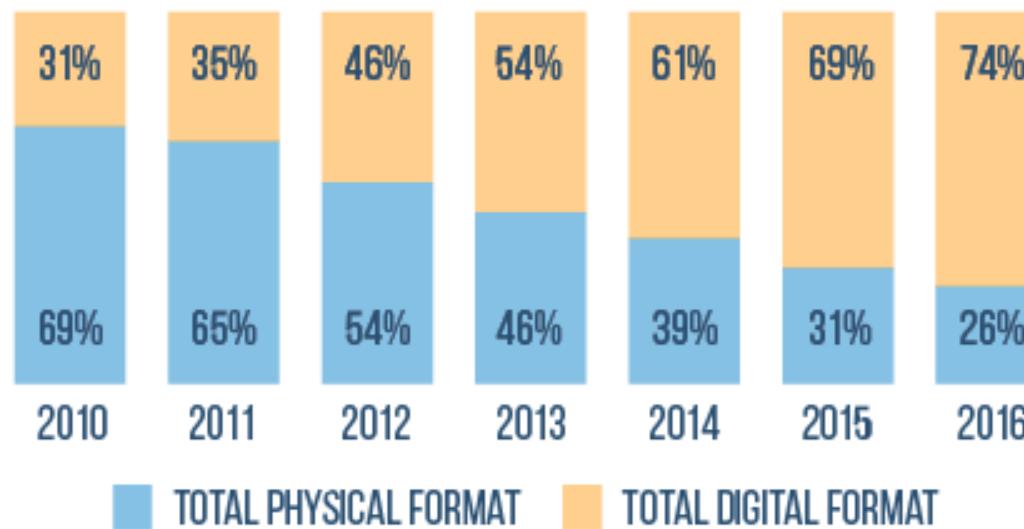
Rise of Empires: Ice  
Age Long Tech Network Lim

★★★★★

- ◆ Digitalna distribucija postaje dominantan način prodaje igara
- ◆ Steam je u travnju 2019 imao milijardu korisnika te oko 90 milijuna mješečno aktivnih

## RECENT DIGITAL\* AND PHYSICAL SALES INFORMATION

\*Digital format sales include subscriptions, digital full games, digital add-on content, mobile apps, and social network games.



Source: The NPD Group

## ◆ Pay-to-Play (P2P)

- Prodaja igre (Guild Wars)
- Pretplata (World of Warcraft)
- Prodaja dodataka na igru
  - Manji – (Downloadable Content - DLC) (Mass Effect 3)
  - Veći – ekspanzije (Age of Conan: Rise of the Godslayer)

## ◆ Free-to-Play (F2P)

- Besplatni dio igre ograničen (Star Wars: The Old Republic)
- Prodaja virtualnih dobara koji utječu na performanse (Travian)
- Prodaja kozmetičkih dodataka (Dota 2)
- “Porez” na trgovinu između igrača (Diablo 3)

## ◆ Hibridni modeli

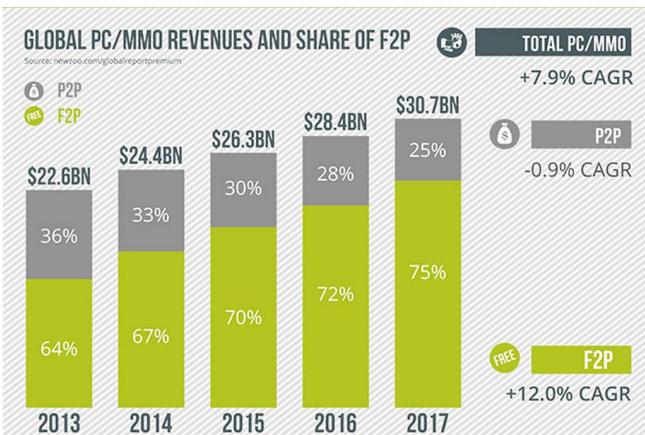
- Igra se plaća, ali postoje i dodatne mogućnosti unutar igre



# Tržište – udio pojedinog modela



- ◆ F2P model preuzima tržište
- ◆ Mnoge „premium“ igre imaju F2P komponente (primjerice World of Warcraft)
  - Prvi plaćeni jahači konj unutar WoW igre je imao cijenu od 25 USD (dvije mjesечne pretplate)
  - Nakon izdavanja se stvorio rep čekanja od 140 000 ljudi koji su htjeli ga kupiti (čekalo se i do 7 sati) – 3,5 milijuna USD



TOP PC GAMES BY REVENUE IN 2017			
FREE-TO-PLAY		PREMIUM	
TITLE	REVENUE	TITLE	REVENUE
LEAGUE OF LEGENDS	\$2.1B	PUBG	\$74M
DUNGEON FIGHTER ONLINE	\$1.6B	OVERWATCH	\$382M
CROSSFIRE	\$1.4B	CS:GO	\$341M
WORLD OF TANKS	\$471M	DESTINY 2	\$218M
DOTA 2	\$406M	GRAND THEFT AUTO V	\$118M
ROBLOX	\$310M	BATTLEFIELD 1	\$113M
MAPLESTORY	\$279M	MINECRAFT	\$92M
HEARTHSTONE	\$217M	GUILD WARS 2	\$87M
BLADE & SOUL	\$178M	DIVINITY: ORIGINAL SIN 2	\$85M
FIFA ONLINE 3	\$163M	RAINBOW SIX: SIEGE	\$67M
<b>TOTAL:</b>	<b>\$7.1B</b>	<b>TOTAL:</b>	<b>\$2.2B</b>



Izvori: SuperData research <https://www.superdataresearch.com/market-data/market-brief-year-in-review/>  
<https://newzoo.com/globalgamesreport>

# F2P model i skalabilnost



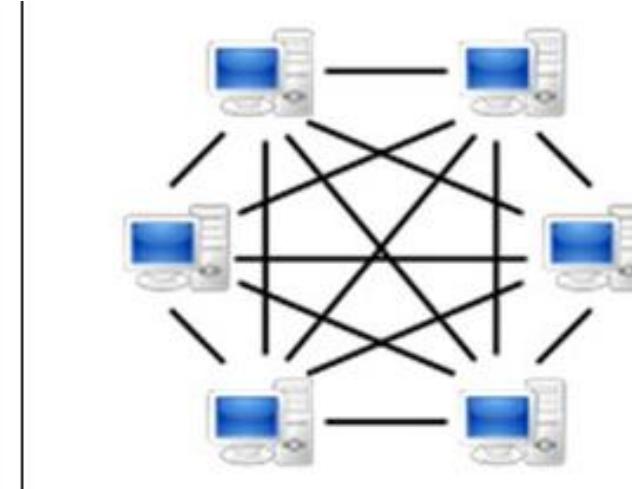
- ♦ F2P miče inicijalnu “branu” korisnicima koji bi igrali igru te omogućuje veći broj korisnika i brži rast
- ♦ Bitka za nasljednika DOTA-e
  - Dvije igre: HoN, 2010 and LoL 2009
  - Slične ocjene: LoL – 78%, HoN – 76%
  - LoL – F2P, HoN – P2P
- ♦ Danas: LoL – 90 milijun korisnika mjesечно, HoN oko 2 miliona
- ♦ LoL – problemi sa skalabilnošću

Date	Registered players	Monthly players	Daily players	Peak concurrent players	Daily hours of play
Jul. 2011	15 million	4 million	1.4 million	0.5 million	3.7 million
Nov. 2011	32.5 million	11.5 million	4.2 million	1.3 million	10.5 million

# Funkcijska arhitektura

- ◆ Peer-to-Peer (P2P)
  - Dobra skalabilnost
  - Nedostaci: loša kontrola varanja, distribucija stanja virtualnog svijeta, problem konzistencije
  - Danas se rijetko koristi “samo P2P” (*Demigod*)
- ◆ Klijent – poslužitelj
  - Klasičan pristup
  - Dobra kontrola
  - Usko grlo – sve ide kroz poslužitelj
    - Više poslužitelja na strani proizvođača (World of Warcraft)
    - Farme poslužitelja na strani proizvođača (EvE online)
    - Klijent postaje poslužitelj (Warcraft III)
    - Namjenski (engl. dedicated) poslužitelji koje mogu održavati igrači (Call of Duty)

- ◆ Veliki broj igara distribuiraju poslužitelje na klijente
  - P2P – potpuna ili ne u potpunosti potpuna povezanost
  - Klijent kao poslužitelj – jedan igrač je poslužitelj a ostali se povezuju na računalo igrača koji služi kao poslužitelj
  - Ukoliko se igrač poslužitelj odspoji iz meča poslužitelj migrira na računalo drugog igrača



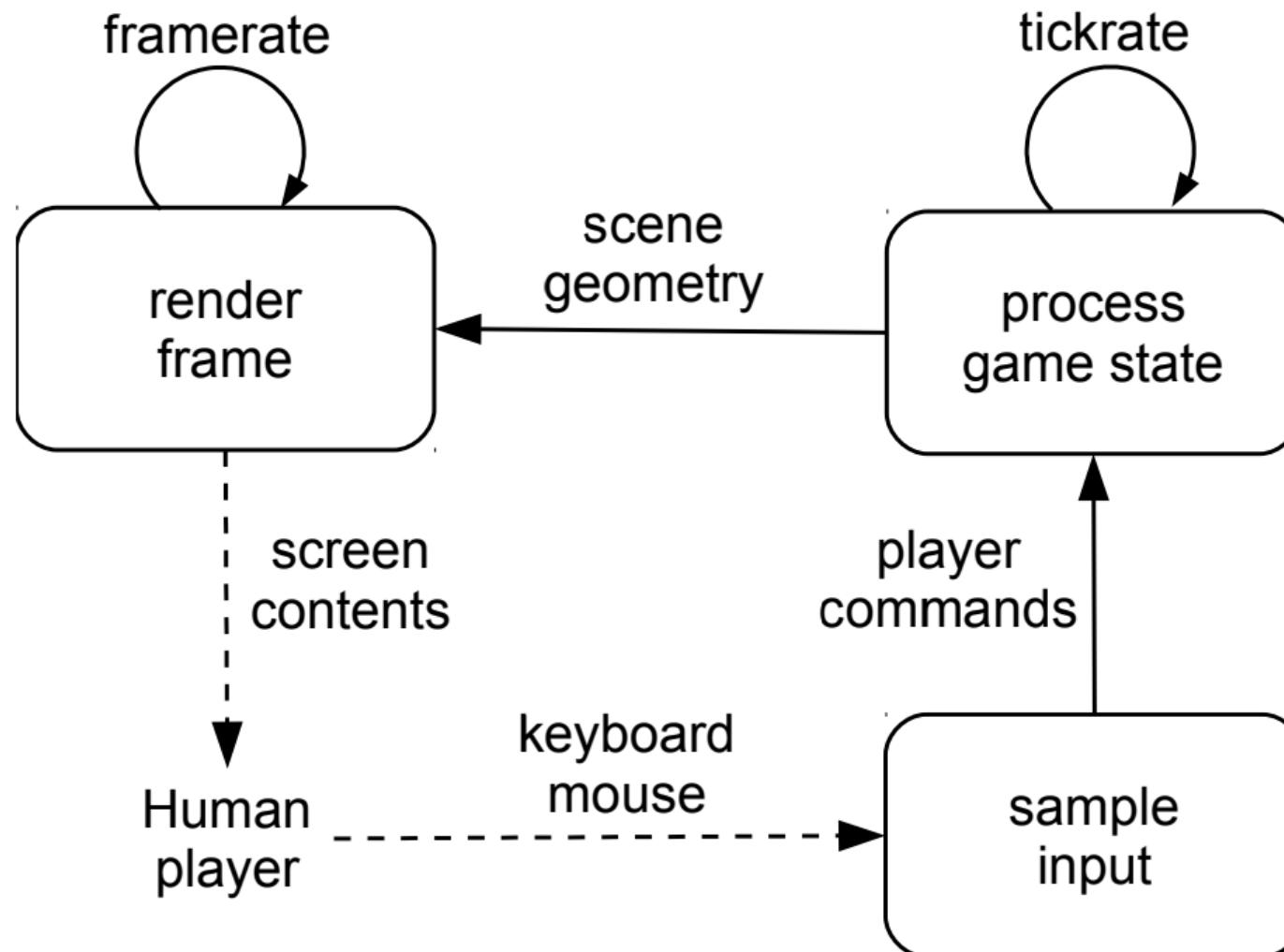
# Demigod – zašto je P2P loš u praksi

- ◆ Prva igra nastala na konceptu DOTA-e
- ◆ Potpuna P2P arhitektura
- ◆ Dosta dobre ocjene kritičara
- ◆ Inicijalno privučeno puno igrača iako je igra naplaćivana
- ◆ Tehnički problemi
  - Problemi s Network Address Translator (NAT) poslužiteljima – poslužitelji za probijanje NAT-a nisu mogli obraditi veliki broj zahtjeva (veliko kašnjenje u obradi zahtjeva)
  - Veliki broj korisnika koji su bili ilegalni (18 000 regularnih i 100 000 ilegalnih)
  - Veliki broj korisnika su bili od spojeni od svojih mečeva
  - Vrlo loša iskustvena kvaliteta – katastrofalno lansiranje igre što je dugoročno upropastilo



- ◆ Osnovne funkcije
  - Unos komandi (uvijek na klijentu)
  - Izračun stanja virtualnog svijeta
  - IsCRTavanje stanja virtualnog svijeta (renderiranje virtualne scene)
- ◆ Sve funkcije na klijentu (nema poslužitelja)
- ◆ IsCRTavanje stanja virtualnog svijeta na klijentu, logika izračuna stanja virtualnog svijeta na poslužitelju (napomena može se lokalna simulacija izvoditi i na klijentu)
  - Tradicionalni način rasporeda funkcija
  - Poslužitelj šalje osvježenja stanja
- ◆ Logika izračuna stanja virtualnog svijeta te isCRTavanje virtualnog svijeta na poslužitelju – igre temeljene na računalnom oblaku (engl. cloud gaming)
  - Poslužitelj šalje video strujanje (dodatna funkcija kodiranja videa na poslužitelju)
  - Visoki zahtjevi na propusnost mreže
- ◆ Hibridni – dio na klijentu, a dio funkcija na poslužitelju

# Osnovna petlja unutar video igre



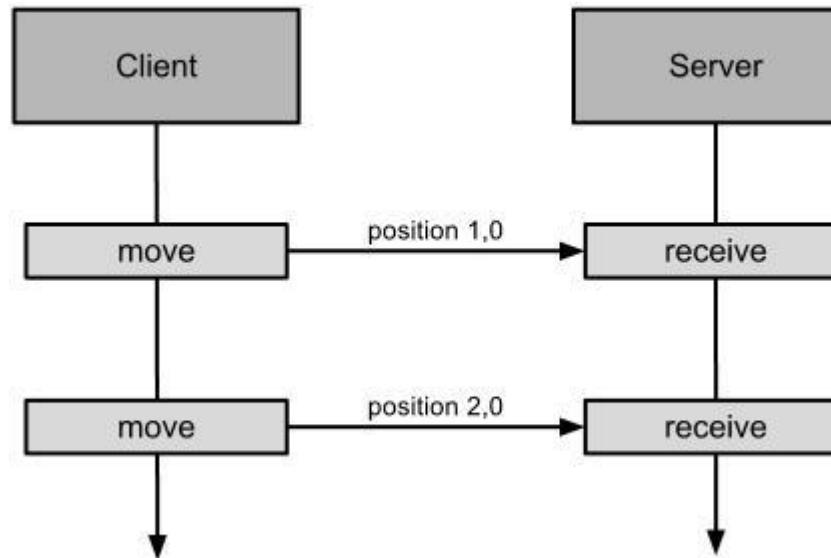
Metzger, F., Rafetseder, A. and Schwartz, C., 2016. A comprehensive end-to-end lag model for online and cloud video gaming. *5th ISCA/DEGA Work. Percept. Qual. Syst.(PQS 2016)*, pp.15-19.

- ◆ Svakih period  $t$  server izračunava novo stanje svijeta na temelju primljenih unosa klijenata
- ◆ Taj period definira broj otkucaja (engl. tickrate) – koliko puta u sekundi se treba izračunati stanje svijeta – mjeri se u Hz
- ◆ Brzina izračuna stanja na poslužitelju ne mora biti ista kao i broj poslanih osvježenja – primjerice ako se stanje nije promijenilo ne mora se slati osvježenje
- ◆ Igre interpoliraju pozicije i stanje virtualnog svijeta između dva osvježenja stanja radi fluidnog prikaza, ali na razini poslužitelja postoje samo stanja izračunata svakog period  $t$  (interpolacija)
- ◆ Brzina prikaza igre na računalu je izražena u broju sličica u sekundi (engl. framerate)

- ◆ Klijent samo izvršava akcije koje mu je server autorizirao na temelju novog stanja – nema lokalne simulacije
- ◆ Klijent radi simulaciju izvođenja na temelju lokalnog unosa komandi – lokalna simulacija
- ◆ Simulirane kretnje kasnije autorizira poslužitelj
- ◆ Prednost lokalne simulacija
  - Smanjuje percepciju mrežnog kašnjenja – ne čeka se odgovor od poslužitelja nego se pretpostavlja da klijentska komanda je ispravna
- ◆ Mane lokalne simulacije
  - Problemi sa varanjem
  - Problemi s „povratkom“ na stvarno autorizirano stanje zbog interakcija s drugim igračima na poslužitelju
  - Problemi s „povratkom“ na stvarno autorizirano stanje zbog gubitka paketa

- ◆ Bez autoriteta poslužitelja

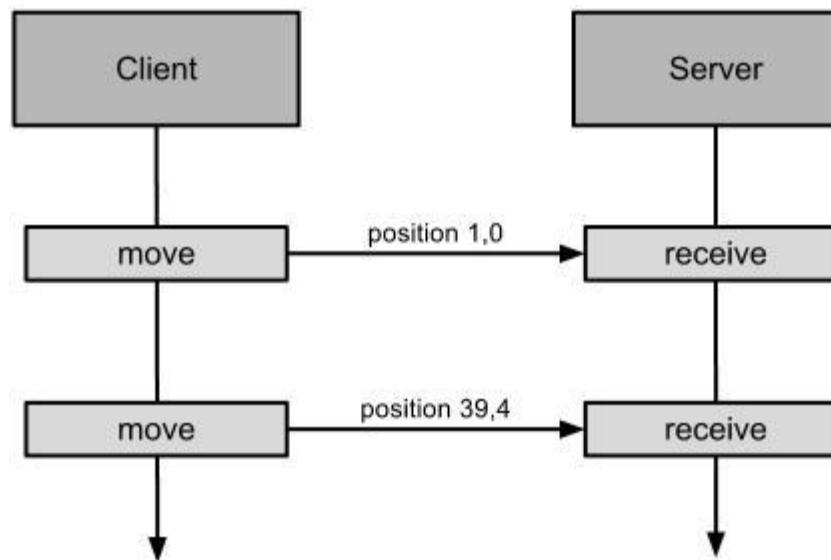
- Klijent pošalje novu poziciju i komande poslužitelju
- Poslužitelj računa prosljeđuje komande drugim igračima
- Nema utjecaja kašnjenja na klijentskoj strani (odmah se izvršava svaka komanda)



Izvor: <https://gamedevelopment.tutsplus.com/articles/gamedev-glossary-what-is-client-side-prediction--gamedev-3849>

# Bez autoritativnog poslužitelja

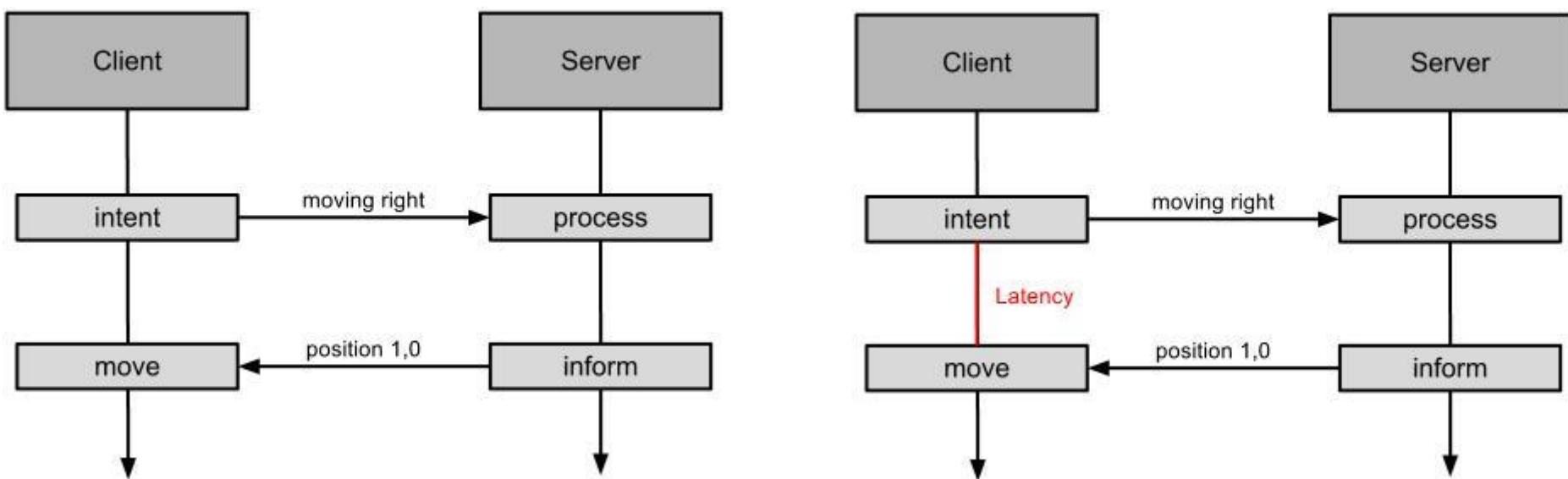
- ◆ Što ako klijent šalje neispravne podatke (varanje)?
- ◆ Primjer slanje promjene velike u poziciji koja inače nije u igri moguća (teleportiranje)
- ◆ Varanje uništava iskustvenu kvalitetu igre za druge igrače



Izvor: <https://gamedevelopment.tutsplus.com/articles/gamedev-glossary-what-is-client-side-prediction--gamedev-3849>

## Autoritativni poslužitelj

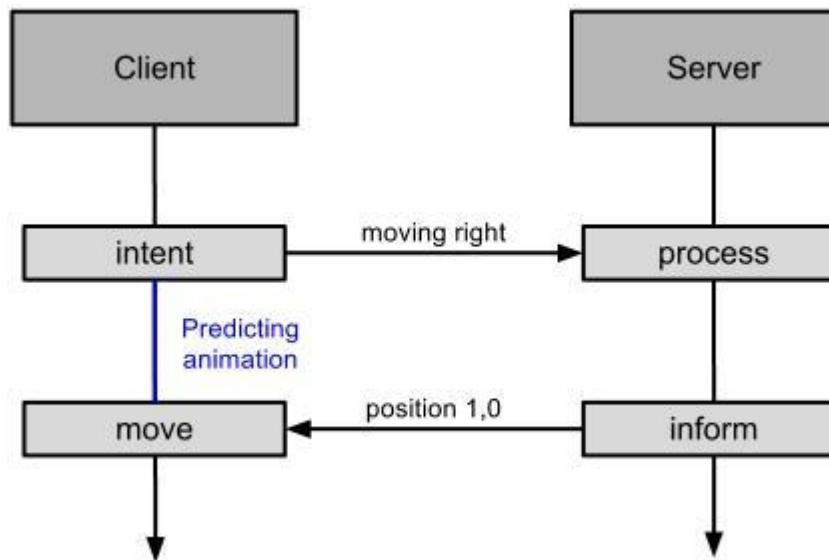
- ◆ Klijent prvo iskazuje namjeru za kretanje, odnosno šalje komandu na autorizaciju poslužitelju
  - ◆ Nakon provjere komande da se provjeri je li ista legalna server je dozvoljava te šalje obavijest o promjeni pozicije klijentu
  - ◆ Unosi se dodatno kašnjenje koje ovisi o procesorskom i mrežnom kašnjenju



Izvor: <https://gamedevelopment.tutsplus.com/articles/gamedev-glossary-what-is-client-side-prediction--gamedev-3849>

# Autoritativni poslužitelj – simulacija na razini klijenta (predikcija kretanja)

- ◆ Klijent odmah nakon slanja zahtjeva za kretanjem i izvršava (animira) dano kretanje na temelju predikcije kako će se isto dogoditi
- ◆ U slučaju da je predikcija dobra (najčešće je) rezultira smanjenim doživljajem kašnjenja
- ◆ U slučaju da je nešto sa strane poslužitelja zaustavilo to kretanje (primjerice šok granata) mora doći do korekcije predviđenog kretanja i pozicije što je novi problem

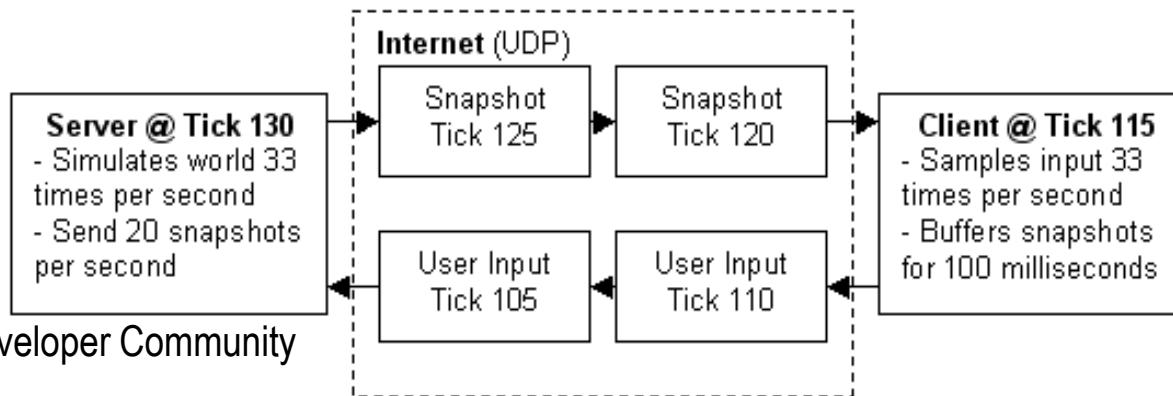


Izvor: <https://gamedevelopment.tutsplus.com/articles/gamedev-glossary-what-is-client-side-prediction--gamedev-3849>

- ◆ Snimak – pojedino stanje virtualnog svijeta na poslužitelju za pojedini “tick”
- ◆ Tick – vremenski period u kojem poslužitelj izračunava stanje virtualnog svijeta
- ◆ Interpolacija – klijentska metoda u kojem se objekti renderiraju između dva dobivena osvježenja njihovih stanja

# Mrežno funkcioniranje Source 1 pokretačkog sustava (tradicionalni koncept)

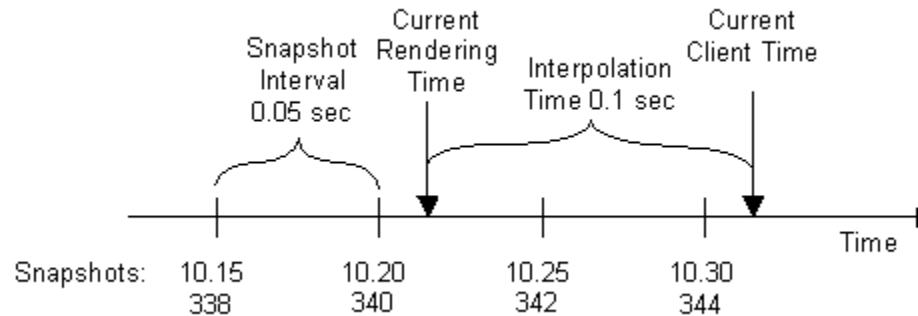
- ◆ Source pokretački sustav (engl. game engine) pokreće igre poput Counter Strike Source (CSS), Team Fortress 2 (TF2), Left 4 Dead (L4D) itd.
- ◆ Moguće uključiti i/ili isključiti lokalnu simulaciju
- ◆ Simulacija se izvršava na poslužitelju.
  - Tickrate 66 za CSS, a 30 za TF2 i L4D – ovisno i interaktivnosti igre
  - Poslužitelj šalje razlike između pojedinih snimaka, a ne cijelo stanje
- ◆ Klijent određuje mrežne parametre
  - Dojavljuje mrežnu propusnost – dizajnirano za propusnost od 40-60 Kb/s
  - Dojavljuje koliki broj osvježenja želi po sekundi (20 je predefinirani broj)
  - Klijent ne šalje zasebno svaku komandu već ih šalje određenom brzinom (30 puta u sekundi je predefiniran broj) te je obično više od jedne naredbe u jednom paketu
  - Klijent može zahtijevati kompletну snimku stanja



Izvor: Valve Developer Community

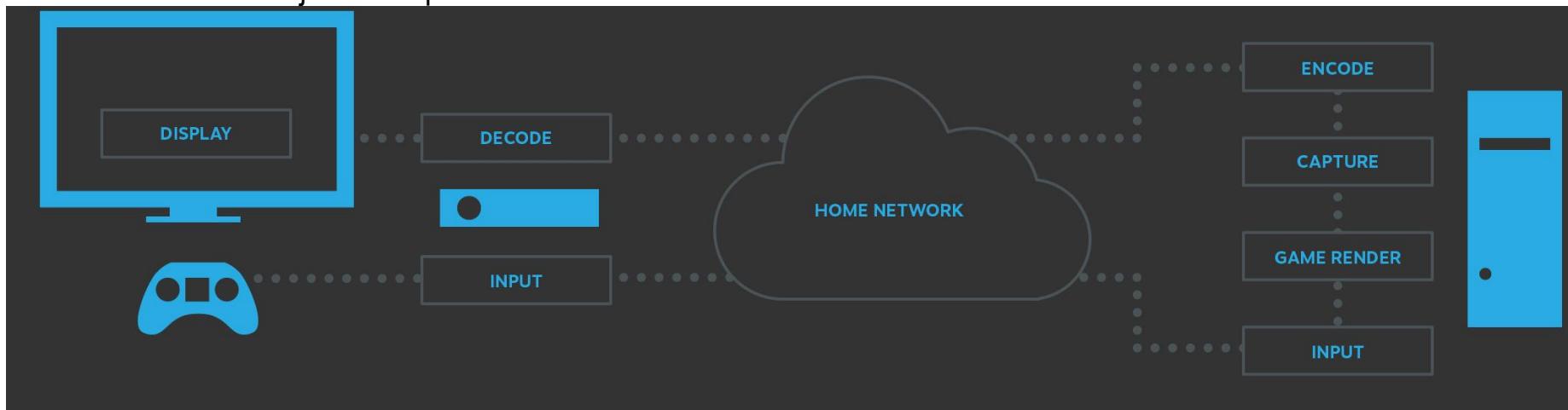
# Mrežno funkcioniranje Source 1 pokretačkog sustava (tradicionalni koncept)

- ◆ Interpolacija na strani klijenta – odgoda izvođenja određene snimke kako bi se mogla interpolirati stanja entiteta između dvije snimke
- ◆ Pretpostavljeno frekvencija dostave osvježenja sa strane poslužitelja u Source engineu je 20 snimki u sekundi
- ◆ Pretpostavljeno vrijeme između dva osvježenja sa strane poslužitelja je 50 ms
- ◆ Ako bi se entiteti na klijentu renderirali samo u trenutcima dolaska novih osvježenja došlo bi do „trzavog“ izvođenja – entiteti se renderiraju većom frekvencijom
- ◆ Ako se vrijeme izvođenja nove snimke na klijentu „odgodi“ za 50 ms onda možemo cijelo vrijeme interpolirati između prošle snimke stanja i novoprdošle
- ◆ Ako se vrijeme izvođenja nove snimke na klijentu „odgodi“ za 100 ms onda možemo ispravno interpolirati čak i kad se izgubi jedna snimka zbog gubitka paketa



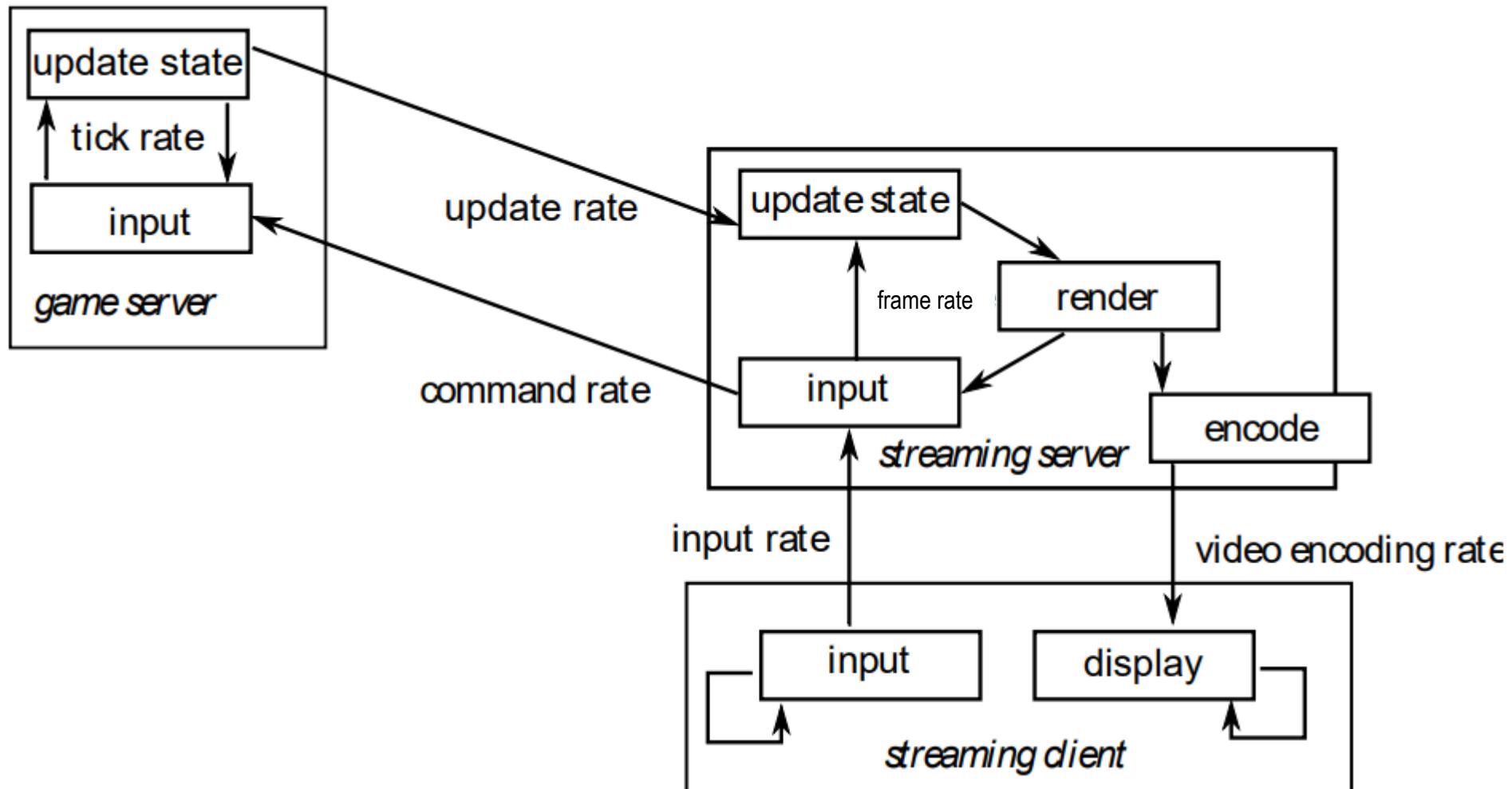
# Koncept igranja u oblaku

- ◆ Kod igranja u oblaku (engl. cloud gaming) igra se izvršava na udaljenom poslužitelju u cijelosti, a klijentu se šalje video visoke razlučivosti dok klijent šalje komande ka poslužitelju
- ◆ Primjer na platformi Steam In-Home Streaming
- ◆ "Oblak" je snažno kućno računalo
- ◆ Osnovne funkcije:
  - Unos komandi
  - Računanje stanja igre
  - Renderiranje igre
  - Snimanje i kodiranje videa
  - Dekodiranje videa i prikaz



Izvor: [https://support.steampowered.com/kb\\_article.php?ref=3629-RIAV-1617](https://support.steampowered.com/kb_article.php?ref=3629-RIAV-1617)

# Osnovna petlja za cloud gaming višekorisničku igru



Izvor: Jarschel, M., Schlosser, D., Scheuring, S. and Hoßfeld, T., 2011, June. An evaluation of QoE in cloud gaming based on subjective tests. In *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (pp. 330-335). IEEE.

# Karakteristike mrežnog prometa

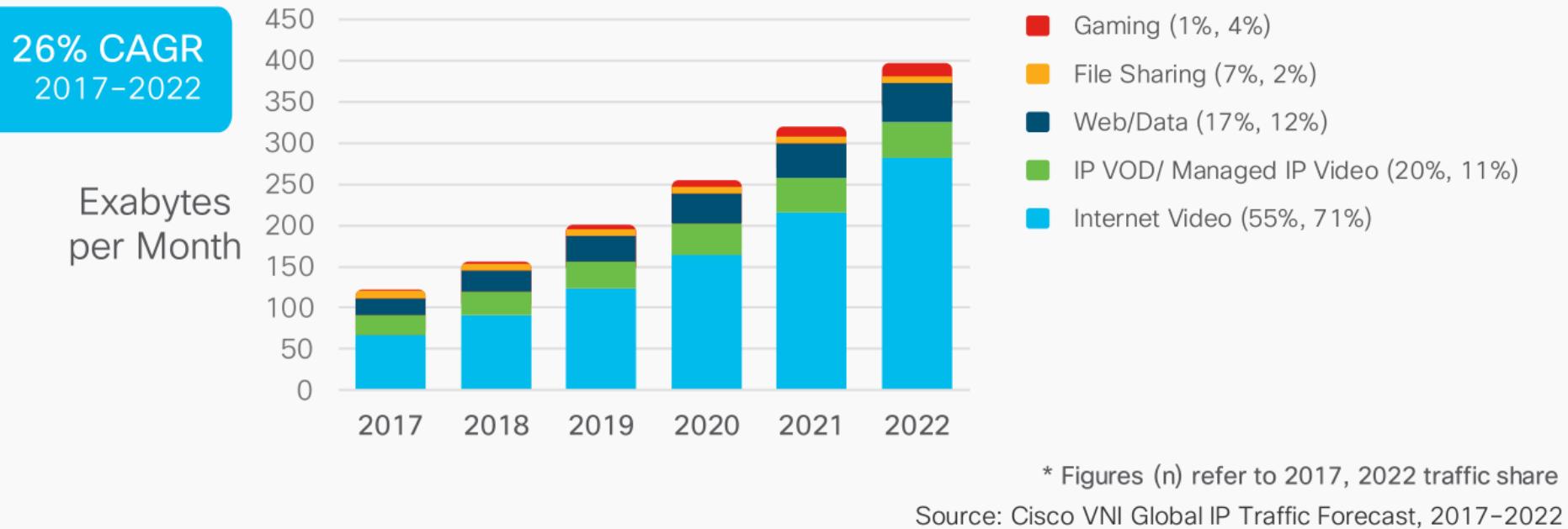
- ◆ Koje informacije prenosi mrežni promet igara?
  - Izlazni promet na klijentu - naredbe igrača
  - Izlazni promet na poslužitelju - osvježenja stanja virtualnog svijeta
  - Tekstualna komunikacija
  - Audio komunikacija među igračima
    - Neke igre imaju ugrađene sustave za VoIP komunikaciju
    - Mnogi igrači koriste i posebne aplikacije (TeamSpeak, Ventrilo, Skype...)
  - Igre u pravilu NE prenose 3D podatke – to je karakteristično za umrežene virtualne svjetove (npr. Second Life), informacije o 3D objektima su pohranjene na klijentima
  - Video – igre u oblaku
    - Strujanje igračih sjednica ([www.twitch.tv](http://www.twitch.tv))

- ◆ Protokoli TCP ili UDP?
  - Ovisno o tipu i zahtjevima igre
- ◆ FPS najčešće koriste UDP
  - Vrlo česte promjene stanja
  - Veliki broj paketa po sekundi
- ◆ MMORPG
  - Oba protokola
  - Češće TCP
  - Nagle algoritam? Odgođena potvrda? (Delayed Acknowledgment)

Protokol	MMORPGs
TCP	World of Warcraft, Lineage I/II, Guild Wars, Ragnarok Online, Anarchy Online, Mabinogi
UDP	EverQuest, City of Heroes, Star Wars Galaxies, Ultima Online, Asherons Call, Final Fantasy XI
TCP/ UDP	Dark Age of Camelot

# Predviđanja rasta mrežnog prometa

- ◆ Promet igara je mali dio sveukupnog prometa (1%), dok mu je stopa rasta 4%
- ◆ Igre generalno imaju niske zahtjeve na mrežnu propusnost, ali visoke zahtjeve na kvalitetu mreže (traže vrlo niska kašnjenja, kolebanja kašnjenja i gubitke paketa )

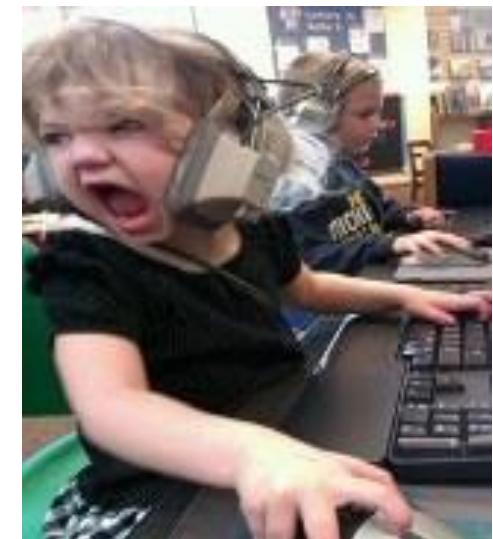


Izvor: Cisco Visual Networking Index

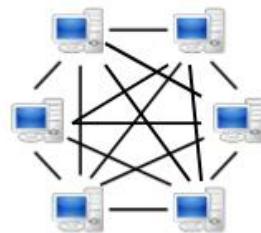
- ◆ Mrežni tokovi igara:
  - Dugo trajanje
  - Visoka brzina paketa
  - Mala veličina paketa
  - Niski zahtjevi na mrežnu propusnost
  - Korištenje i UDP-a i TCP-a
  - **Karakteristike jako ovise o tipu igre**
- ◆ Zahtjevi :
  - Osjetljivi na kvalitetu mreže
  - Veliki signalizacijski overhead
  - Različita zahtijevana pouzdanost
- ◆ Izuzetak su igre u oblaku

*120 hours of World of Warcraft*

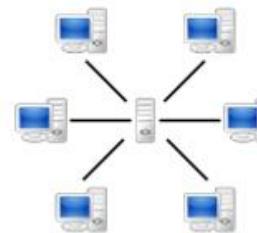
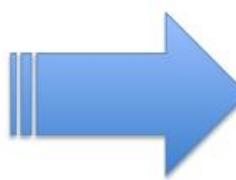
by Elizabeth Harper Jul 24th 2007 at 8:10PM



# Evolucija mrežnog prometa? – Ne baš...



Peer-to-peer  
Architecture



Server-client  
Architecture



## StarCraft I (1998-2010)

1 - 5 kbps  
(2-8 igrača)

## StarCraft II (2010-present)

2 - 3 kbps  
(neovisno o broju  
igrača)

M. Claypool, D. LaPoint, and J. Winslow, "Network Analysis of Counter-strike and Starcraft," in Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference (IPCCC), USA, April 2003.

C-S. Lee, "The Revolution of StarCraft Network Traffic" in Proceedings of the 11th Annual Workshop on Network and Systems Support for Games NetGames 2012

# Revolucija mrežnog prometa igara? Da\*

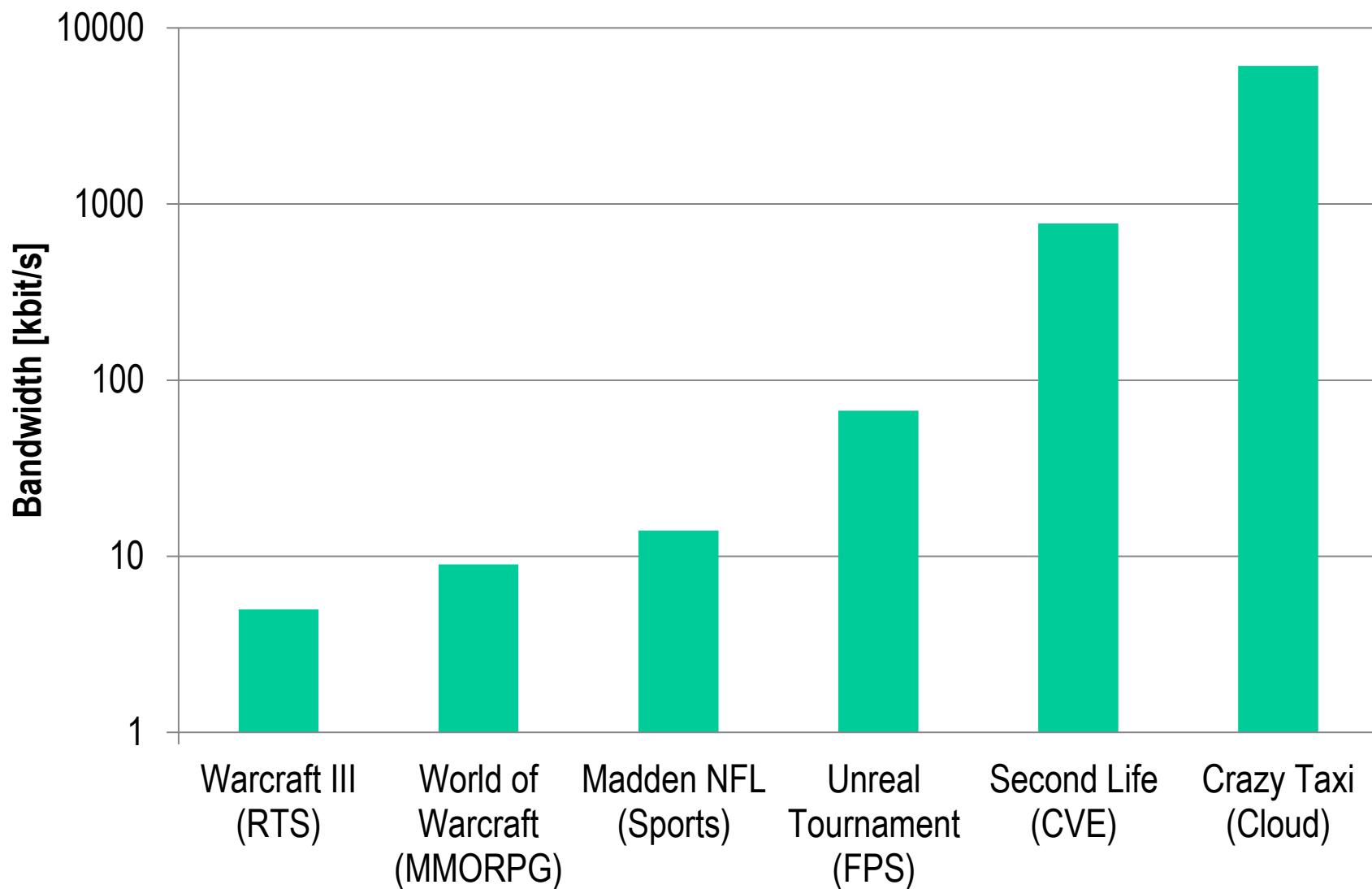
- ◆ Igre u oblaku
  - Izračun stanja i renderiranje virtualne scene se događa na poslužitelju
  - Klijentu se šalje strujanje videa visoke razlučivosti
- ◆ Fundamentalno različite karakteristike prometa
  - Vrlo visoki zahtjevi na propusnost veze
  - Viši zahtjevi na mrežno kašnjenje nego "standardne" igre
- ◆ \*Bez većeg komercijalnog uspjeha
- ◆ Ako igranje u oblaku zaživi promet igara će postati jedna od najznačajnijih kategorija prometa u Interneta



RTP/UDP flows of the OnLive Streaming Protocol

Direction	RTP SSRC	RTP Payload Type	Flow description
Downstream	0x00000000	100	QoS monitoring flow
Downstream	0x00010000	100	OnLive Control
Downstream	0x00030000	100	Audio stream (CBR Codec)
Downstream	0x00040000	100	Cursor position
Downstream	0x00050000	101	Audio stream (VBR Codec)
Downstream	0x00060000	96	Video stream
Downstream	0x00080000	100	Voice Chat (Sound from other players)
Upstream	0x0000XXXX	100	User input (keyboard and mouse buttons)
Upstream	0x0001XXXX	100	Cursor movement
Upstream	0x0004XXXX	100	OnLive Control ACK
Upstream	0x0008XXXX	100	Voice Chat (Microphone from the user)

# Mrežna propusnost – razlika među tipovima igara

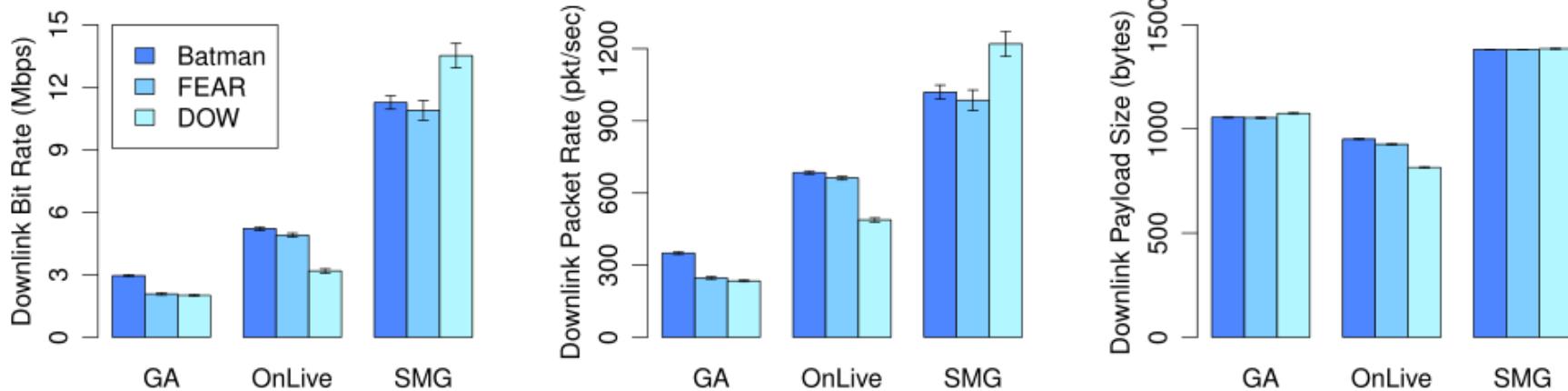


# Razlike u propusnosti kod igra u oblaku

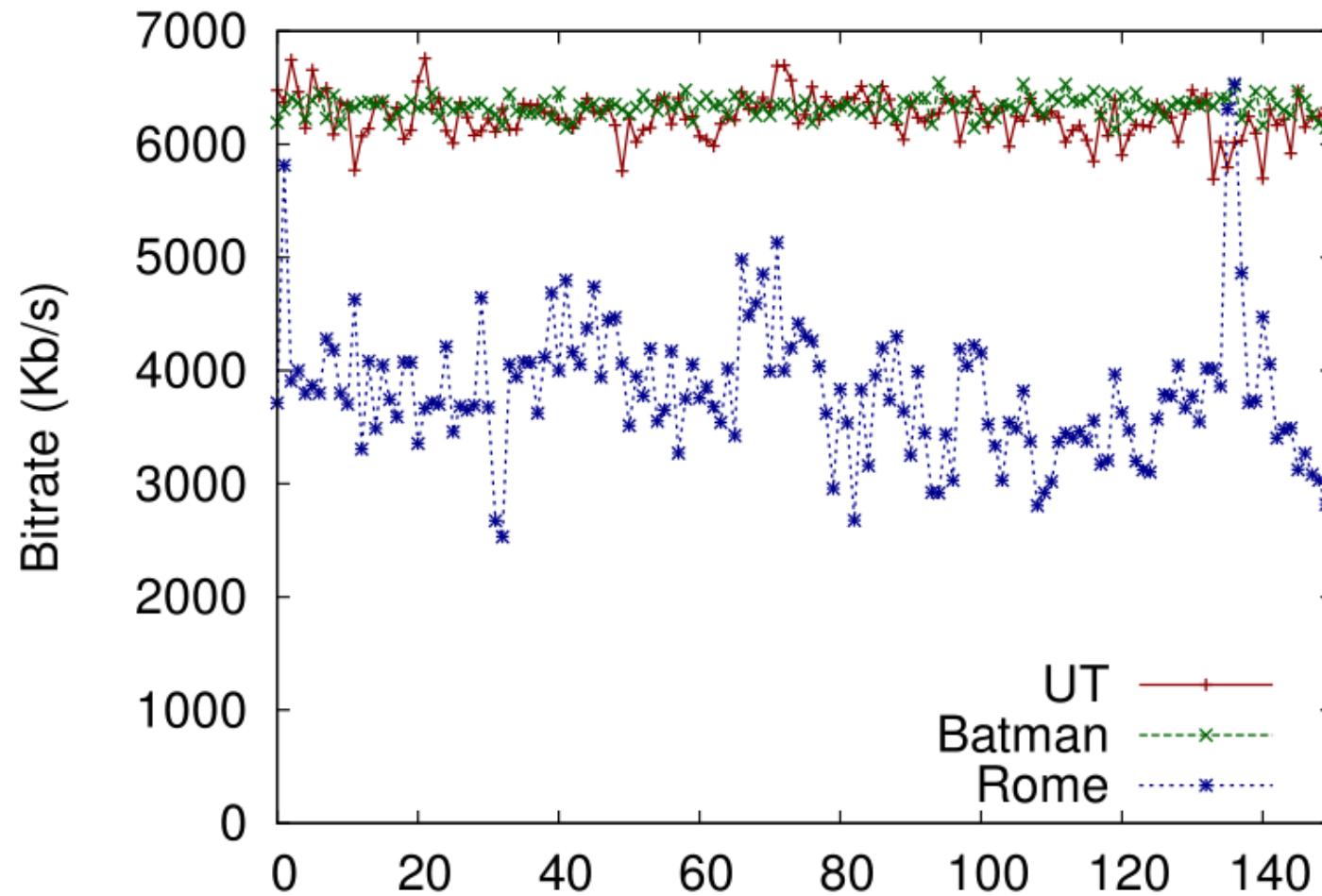
- ◆ Razlika zbog dinamike igre – različita dinamika videa

		<i>pes2012</i>	<i>unreal3</i>	<i>crazytaxi</i>	<i>aircombat</i>	<i>4elements</i>
Downstream	Total time (s)	249.41	261.16	200.56	239.68	236.59
	Number of packets	149004	174265	13867	153798	108619
	Avg. packets / sec	597.41	667.25	691.39	641.66	459.09
	Avg. packet size (B)	915.57	975.05	1014.99	955.65	722.58
	Bit rate (Mbps)	4.37	5.21	5.61	4.91	2.65
Upstream	Total time (s)	249.48	261.31	200.69	239.83	236.72
	Number of packets	8947	13943	6825	14677	14849
	Avg. packets / sec	35.86	53.35	34.01	61.19	62.72
	Avg. packet size (B)	168.49	157.8	170.08	154.81	154.91
	Bit rate (Mbps)	0.0048	0.067	0.046	0.075	0.077

## Razlika u implementaciji platformi



# OnLive downstream traffic



M. Claypool, D. Finkel, A. Grant, and M. Solano: "Thin to win? Network performance analysis of the OnLive thin client game system". *11th Annual Workshop on Network and Systems Support for Games (NetGames)*, 2012 (pp. 1-6). IEEE.

# Utjecaj mrežnih degradacija

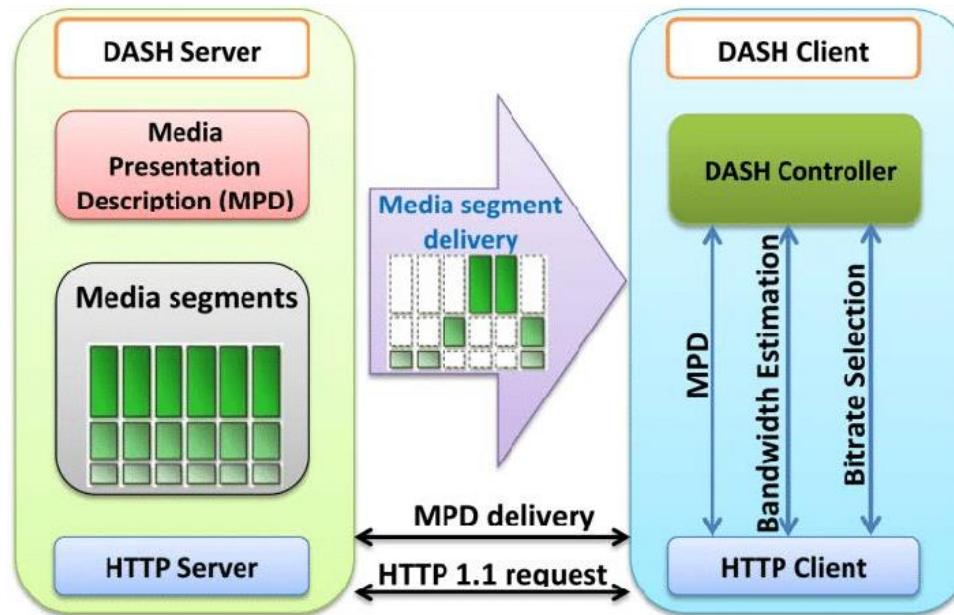
- ◆ Kvaliteta usluge (engl. Quality of Service - QoS) u najširem smislu je stupanj zadovoljstva korisnika usluge (ITU-T E.800)
- ◆ U praksi, QoS se pretežno odnosi na tehničke parametre (npr. parametri sustava, mreže)
- ◆ Iskustvena kvaliteta (engl. Quality of Experience – QoE) je sveukupna prihvatljivost aplikacije ili usluge, subjektivno percipirana od strane krajnjeg korisnika
- ◆ Pojam iskustvene kvalitete obuhvaća sve učinke sustava s kraja na kraj (klijent, terminal, mreža, itd.). Korisnikova očekivanja i kontekst mogu također utjecati na ukupnu prihvatljivost aplikacije ili usluge.
- ◆ Glavni mrežni parametri koji utječu na iskustvenu kvalitetu igara su:
  - Propusnost (primarno za igre temeljene na računalnom oblaku)
  - Kašnjenje – osnovni parametar koji definira interaktivnost igre
  - Gubitak paketa – različito djeluje na igre temeljene na računalnom oblaku (artefakti u video prikazu) i na “tradicionalne” mrežne igre (u mnogome ovisi o načinu implementacije igre)

Karakteristike	QoS	QoE
<b>Područje primjene</b>	Uglavnom telekomunikacijske usluge	Šira primjena (nisu nužno u pitanju samo uslugu u mrežnom okruženju)
<b>Fokus</b>	Tehničke performanse sustava (mehanizmi kao npr. Diffserv)	ICT usluge ili aplikacije
<b>Metode mjerena</b>	Tehnološki orijentirane; empirijska ili simulacijska mjerena	Multi-disciplinaran pristup (razna subjektivna i objektivan mjerena)

Izvor: Qualinet White Paper on definitions of QoE, 2012

# Prilagodljivost propusnosti

- ◆ Usluga ima više verzija koje se mogu dostaviti
- ◆ Svaka od verzija ima određene zahtjeve na mrežu – najčešće u smislu propusnosti
- ◆ Primjer prilagodljivo strujanje na temelju protokola HTTP (engl. HTTP Adaptive Streaming – HAS)



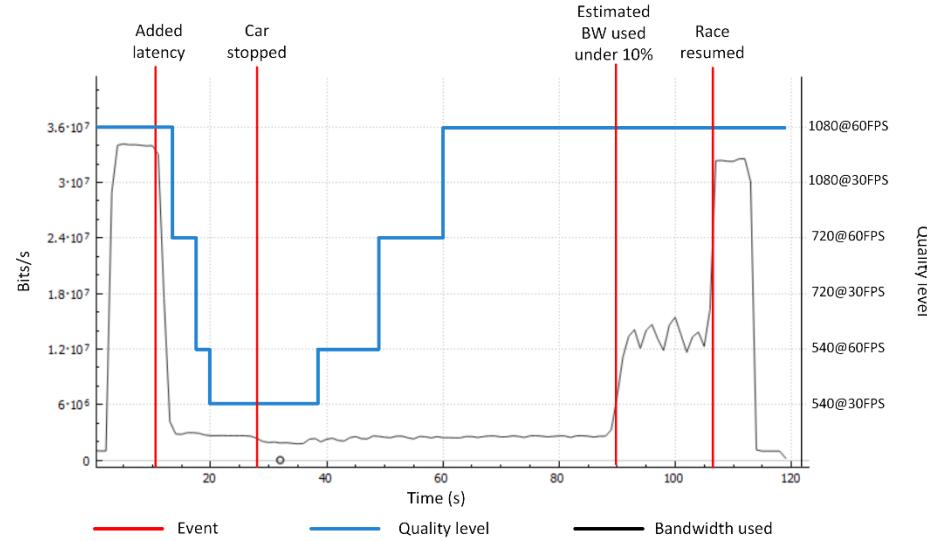
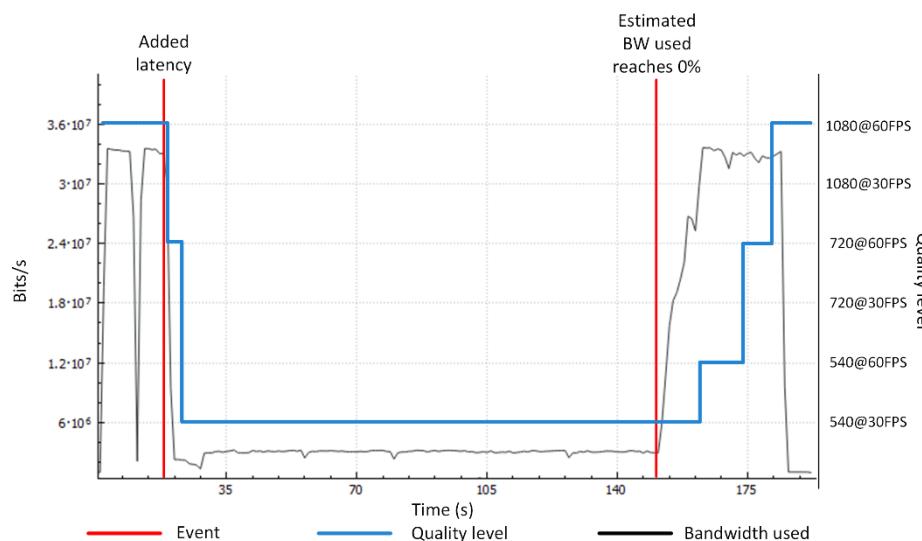
Vu, V.H., Mashal, I. and Chung, T.Y., 2017. A Novel Bandwidth Estimation Method Based on MACD for DASH. *KSII Transactions on Internet & Information Systems*, 11(3).

# Prilagodljivost propusnosti kod igara temeljenih na računalnom oblaku

- ◆ Mrežne karakteristike mogu varirati u vremenu
- ◆ U slučaju da se brzina video strujanja ne prilagodi dolazi do gubitka paketa te vidljivih degradacija video toka
- ◆ Igre temeljene na računalnom oblaku imaju specifične karakteristike
  - Vrlo zahtjevna usluga – čak do 50Mbit/s
  - Sadržaj ne postoji unaprijed – dinamički se generira
  - Kratko vrijeme za kodiranje – uvjetovano stvarnovremenskim zahtjevima na interaktivnost igre
  - Potrebna vrlo brza procjena propusnosti te algoritam prilagodbe

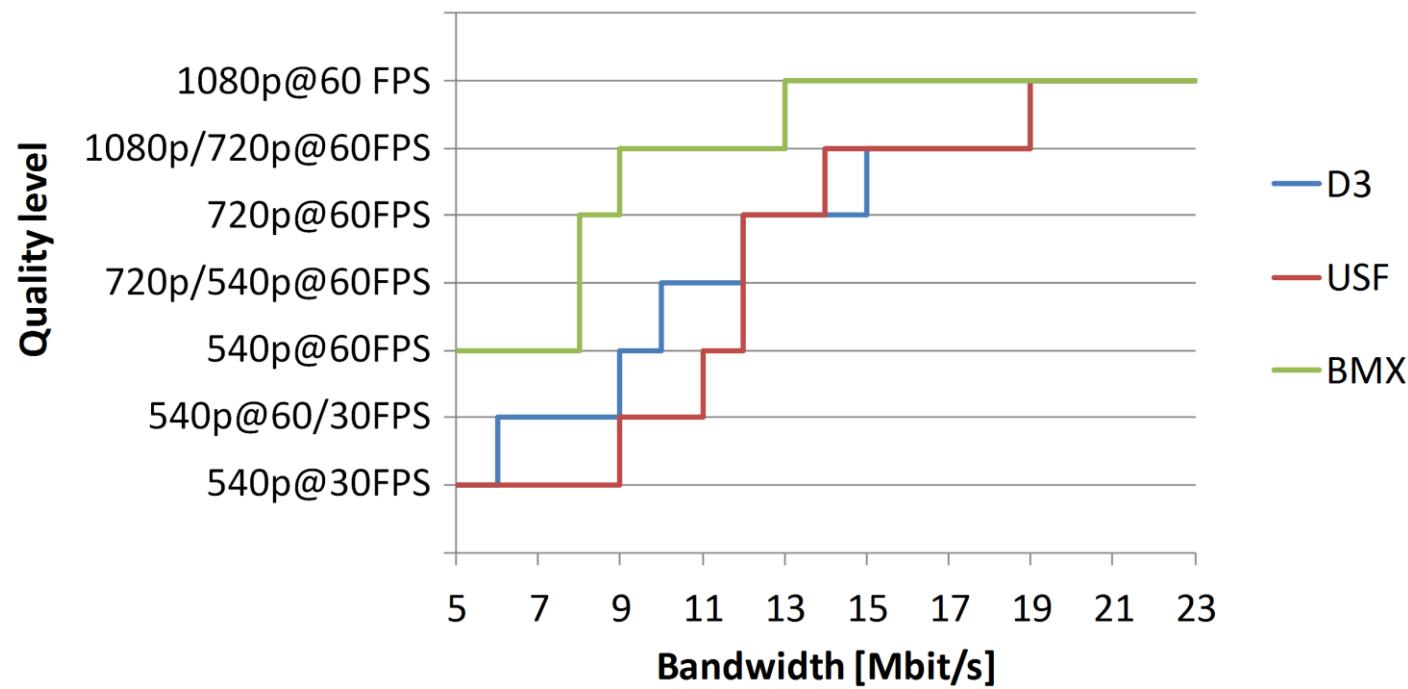
# Algoritam procjene propusnosti – GeForce Now

- ◆ Algoritam za procjenu propusnosti temelji se na mjerenu mrežnog kašnjenja
- ◆ Nakon uvođenja dodatnog kašnjenja sustav za procjenu propusnosti prepostavlja da je mreža preopterećena - smanjuje bitrate kodiranja na vrlo nisku razinu
- ◆ Nakon određenog vremena procjene pojase širine „oporavlja“ i procjenjuje točnu širinu pojasa
- ◆ Vrijeme oporavka razlikuje se ovisno o značajkama videozapisa



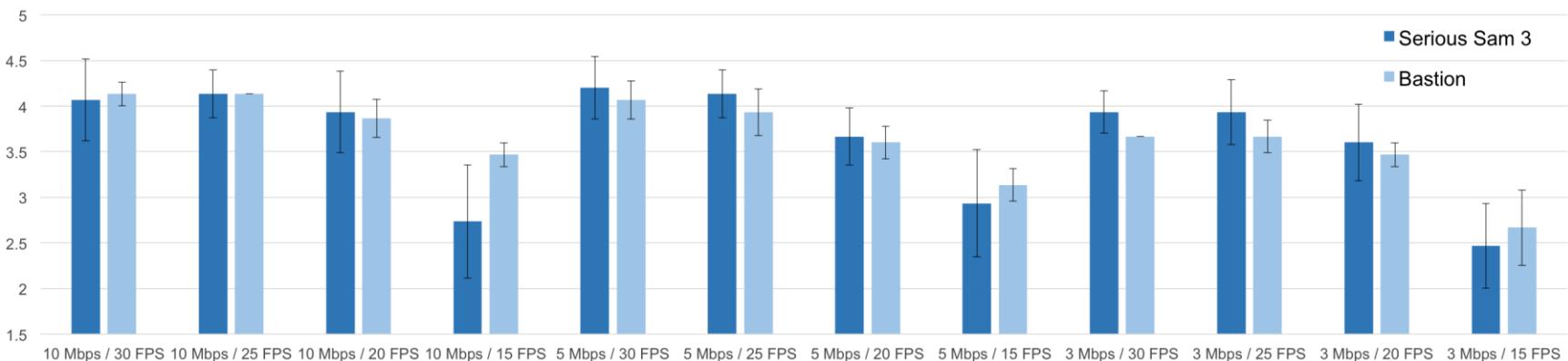
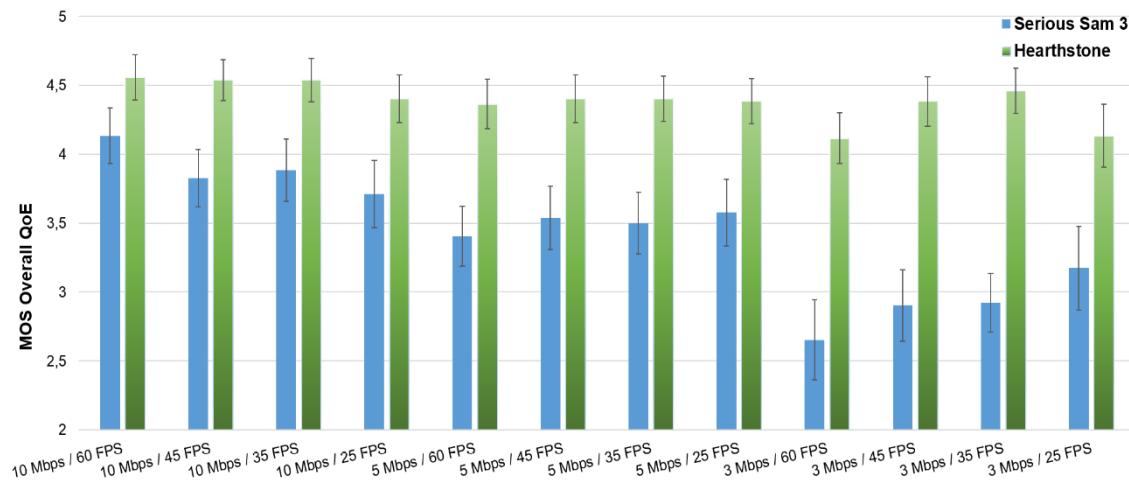
## ◆ Razlike po platformama

- Steam InHome Streaming – ne mijenja se rezolucija ni brzina osvježavanja ekrana s padom propusnosti
- GeForce NOW – traži 30 Mbit/s za punu kvalitetu, ali implementira adaptacijski algoritam koji snižava rezoluciju pa potom brzinu osvježavanja sličica na 30 FPS
- Koji je najbolji način prilagodbe? Temeljeno na QoE modelima

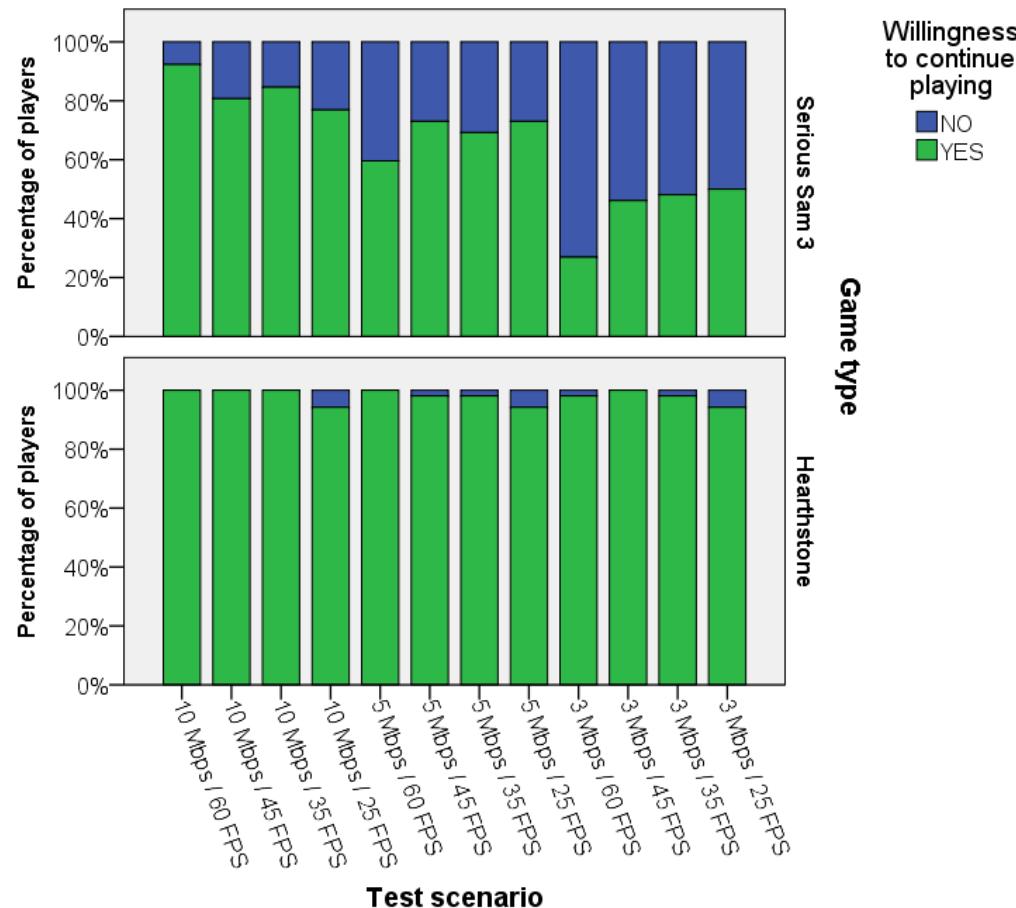


# QoE različitih igara

- ◆ Subjektivne studije na Sveučilištu u Zagrebu
- ◆ Nekoliko testiranih igara: Serious Sam 3 (igra FPS) Hearthstone (kartaška igra) Bastion (platformska igra) Orcs must die (akcijska igra)
- ◆ Neke igre isto, a neke različito utječu na iste postavke kodeka (frame rate / resolution)
- ◆ Zašto?



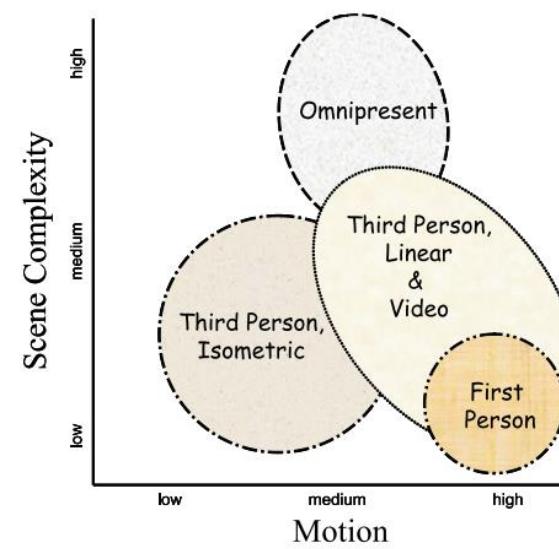
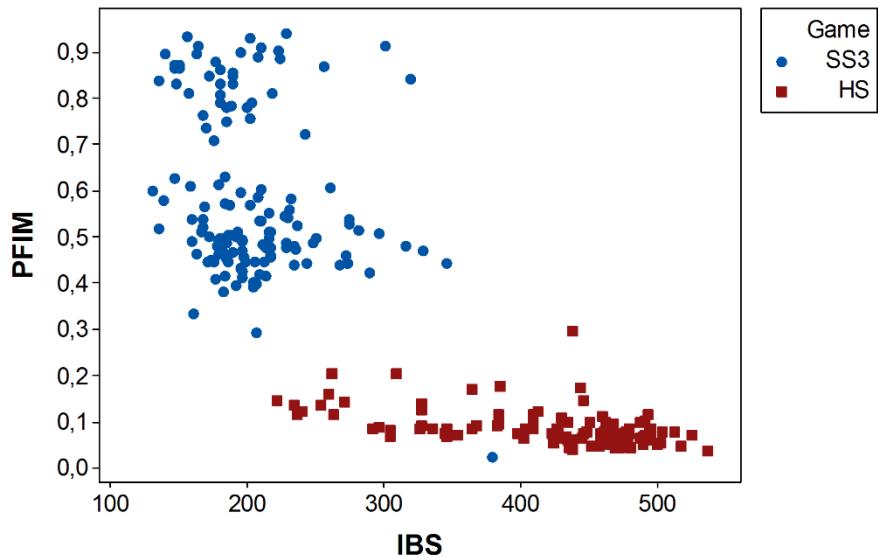
# Voljnost nastavka igranja



Slivar, Ivan, Mirko Suznjevic, and Lea Skorin-Kapov. "The impact of video encoding parameters and game type on QoE for cloud gaming: A case study using the Steam platform." *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on.* IEEE, 2015.

# Karakteristike videa

- ◆ Mjera pokreta - postotak kodiranih makroblokova, tj. Postotak makroblokova naprijed / nazad ili unutar kodiranih (PFIM) u svim okvirima
- ◆ Mjerenje složenosti scene, prosjek Intra-kodiranih veličina bloka (IBS)
- ◆ Značajke videa, a ne vrsta igre - diktira QoE na različitim postavkama video kodeka?

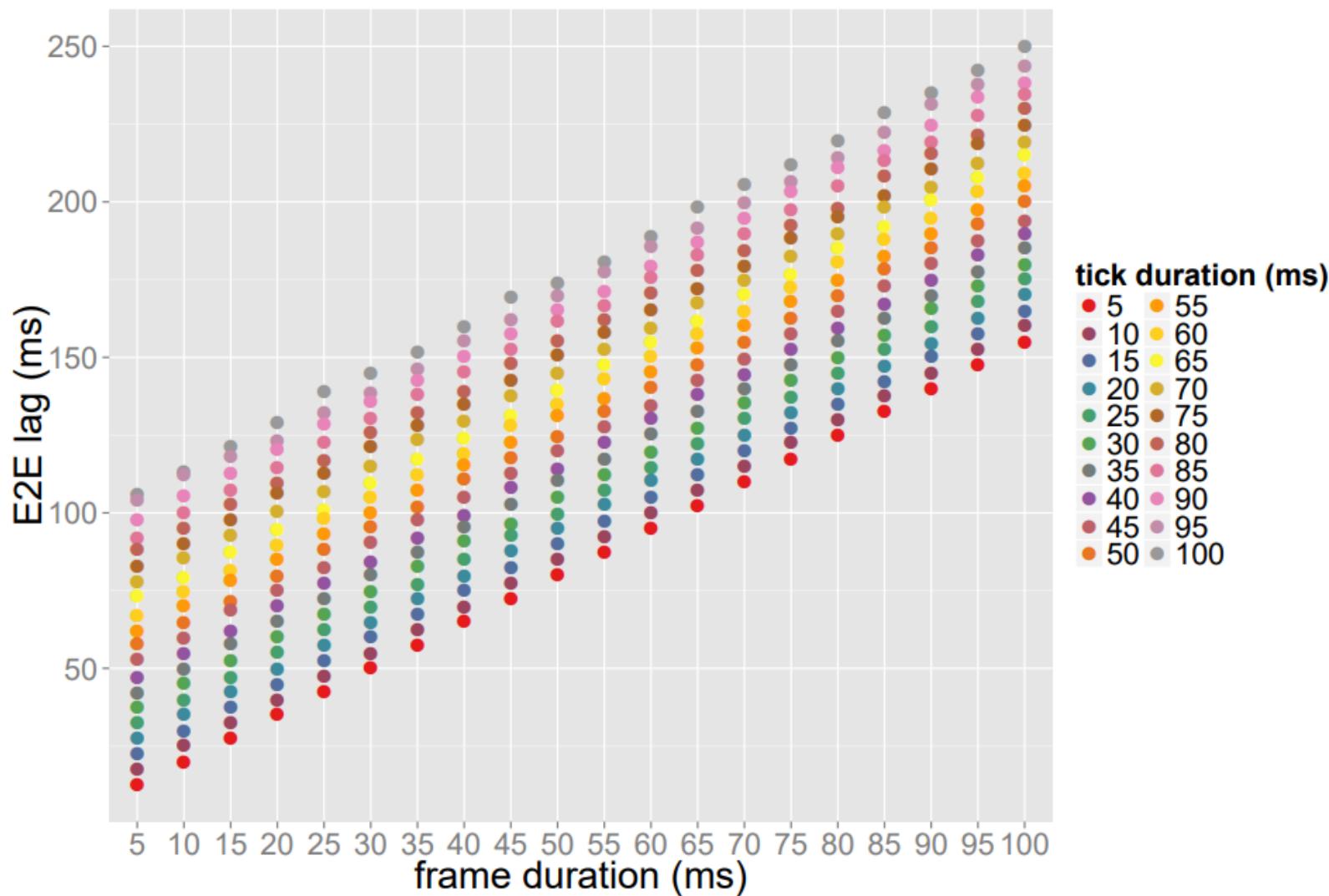


Slivar, Ivan, Lea Skorin-Kapov, and Mirko Suznjevic. "Cloud gaming QoE models for deriving video encoding adaptation strategies." *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016.

Claypool, Mark. "Motion and scene complexity for streaming video games." *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 2009.

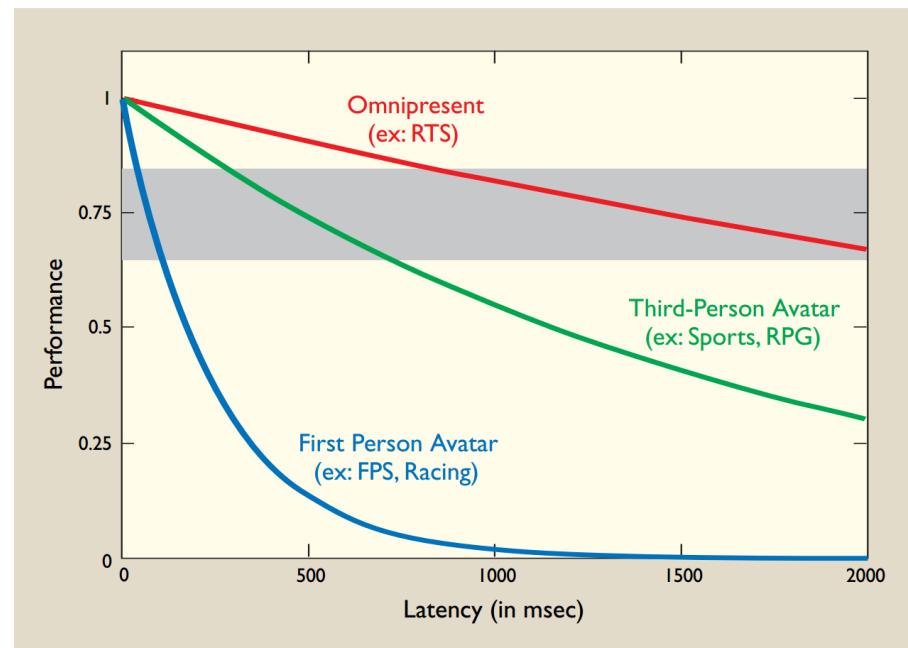
- ◆ Osnovne komponente mrežnog kašnjenja
  - Propagacijsko – ograničenje: brzina svjetlosti
  - Procesorsko – kašnjenje zbog obrade u usmjeriteljima
  - Transmisijsko – kašnjenje uzrokovano slanjem podataka na prijenosni medij
  - Kašnjenje u redovima čekanja – kašnjenje uzrokovano čekanjem u međuspremnicima
- ◆ Kašnjenje je UVIJEK prisutno! (kao i razlika u kašnjenju pojedinih igrača)
- ◆ Mrežno kašnjenje je jedan od najvažnijih parametara kvalitete umreženih igara – potrebno je održati iluziju da virtualni svijet nije distribuiran u mreži

# Ukupno kašnjenje u sustavu



# Utjecaj kašnjenja na ishod igre

- ◆ Utjecaj kašnjenja ovisi o vrsti igre!
  - Omnipresent – igre s pogledom “odozgo” na cijeli virtualni prostor (primjerice Warcraft 3)
  - 3 lice – pogled koji prati avatara iz određene perspektive (sportovi, RPG-ovi)
  - 1 lice – pogled iz perspektive avatara (FPS-ovi)
- ◆ Subjektivne studije:
  - Igre iz prvog lica – do 160 ms kašnjenje u oba smjera (engl. Round Trip Time)
  - Igre iz trećeg lica – do 250 ms RTT
  - Omnipresent igre – do 400 ms RTT, ‘

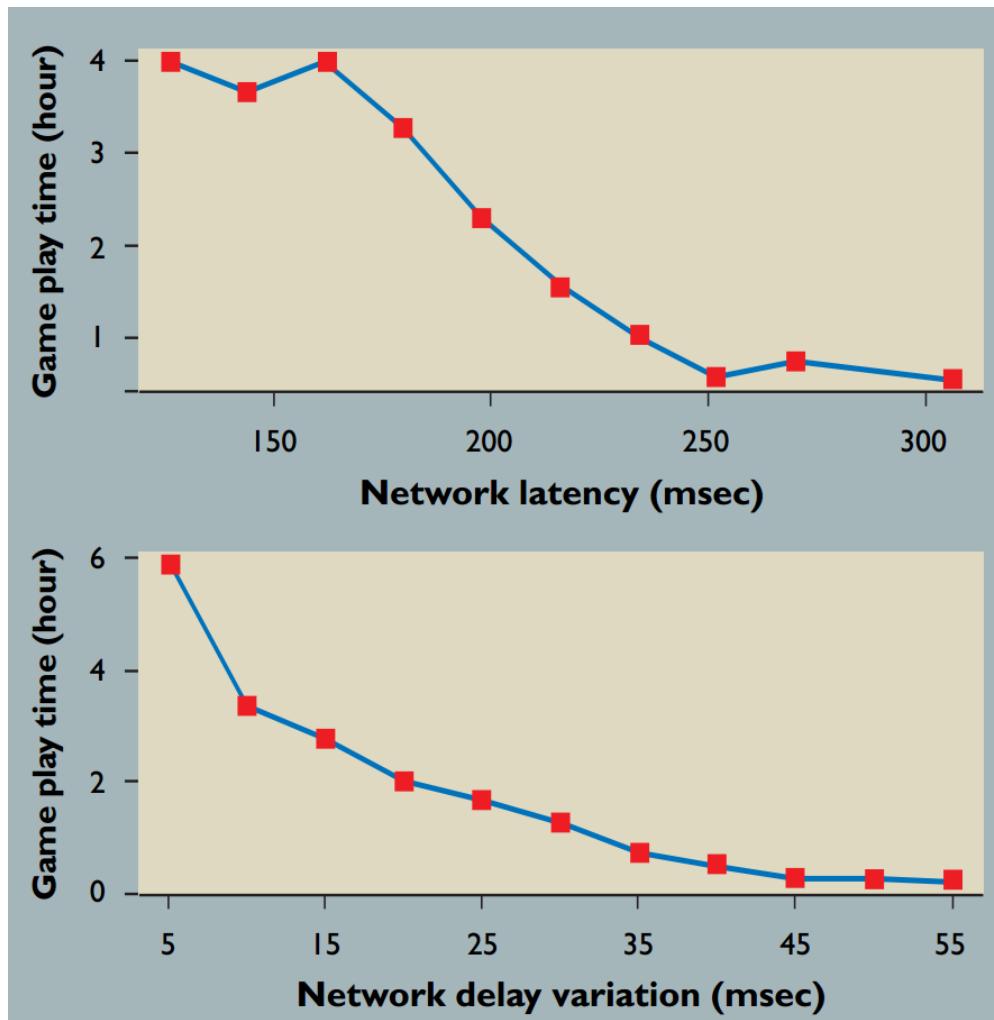


Izvori: M. Claypool and K. Claypool, Latency and player actions in online games. *Commun. ACM* 49, 11 (November 2006), 40-45

Dick, M., Wellnitz, O., and L. Wolf, "Analysis of factors affecting players' performance and perception in multiplayer games", Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games, pp. 1 - 7 , 2005.

Cajada, M., "VFC-RTS: Vector-Field Consistency para Real- Time-Strategy Multiplayer Games", Master of Science Dissertation , 2012.

# Utjecaj kašnjenja na prosječno trajanje sjednice



- ◆ K.-T. Chen, P. Huang, and C.-L. Lei, "How sensitive are online gamers to network quality?" Communications of the ACM, vol. 49, no. 11, pp. 34–38, 2006.

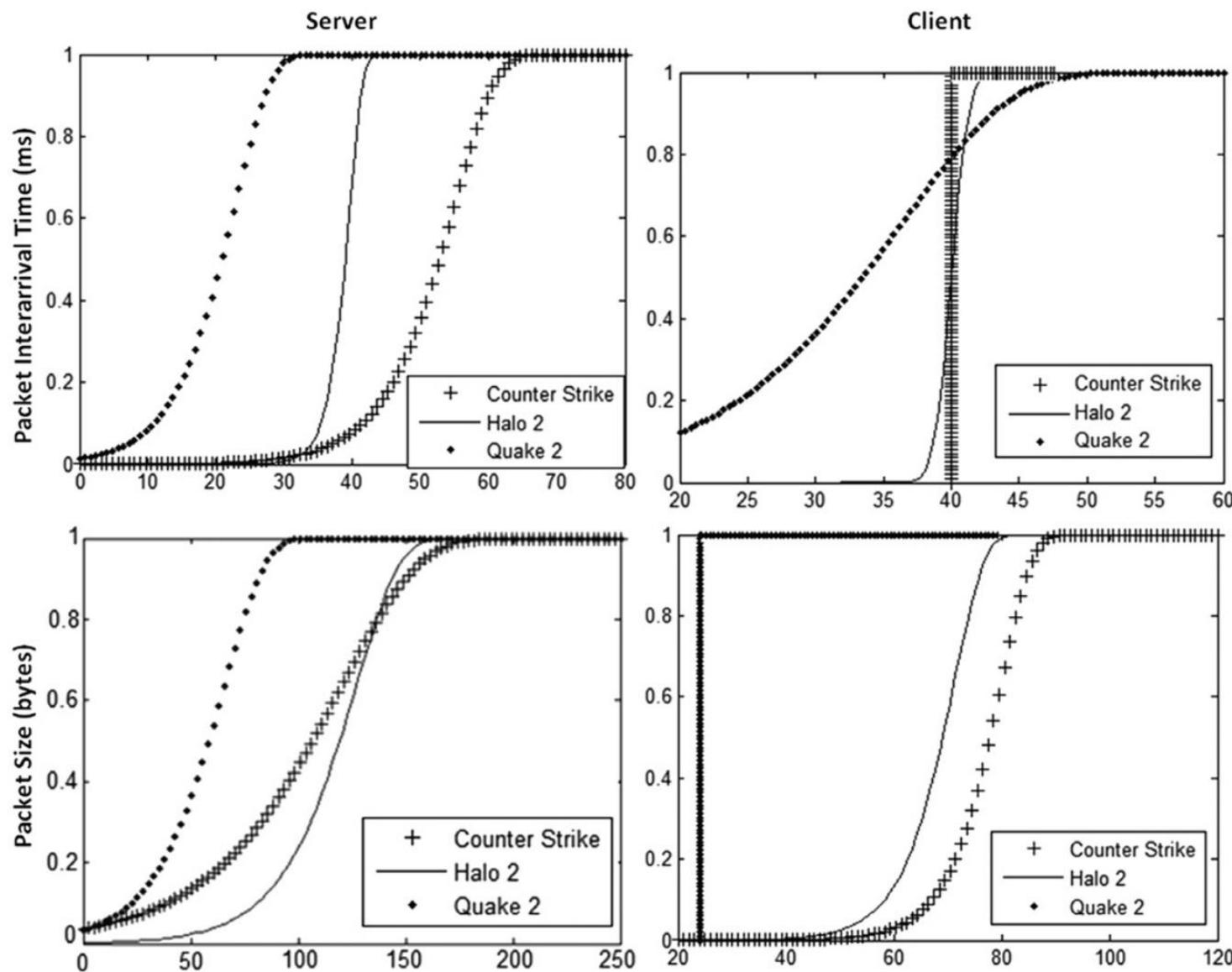
- ◆ Karakteristike žanra:

- Vrlo interaktivne igre koje zahtijevaju brze reakcije
- Vrlo osjetljive na kašnjenje (potrebno ispod 160 ms RTT)
- Nekoliko desetina igrača u istoj inačici virtualnog svijeta

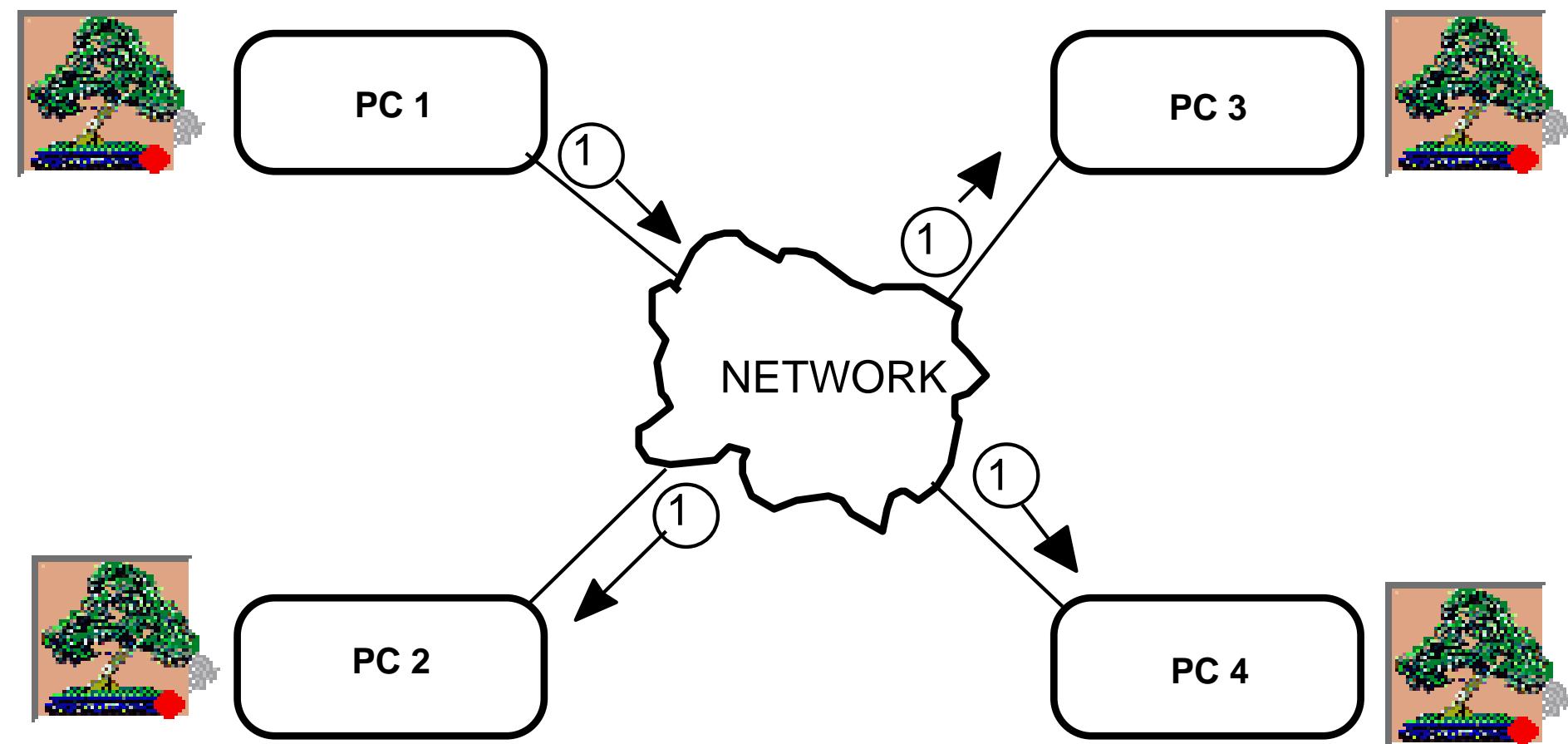
- ◆ Karakteristike prometa

- Koriste UDP
- Visoka tolerancija na gubitke paketa (ovisi o igri)
- Učestala osvježenja virtualnog svijeta
- Najčešće predefinirane veličine paketa
- Najčešće predefinirana međudolazna vremena paketa
- Najzahtjevniji žanr igara što se tiče propusnosti (obično manje od 300kb/s)

# Mrežni promet – karakteristike (FPS)



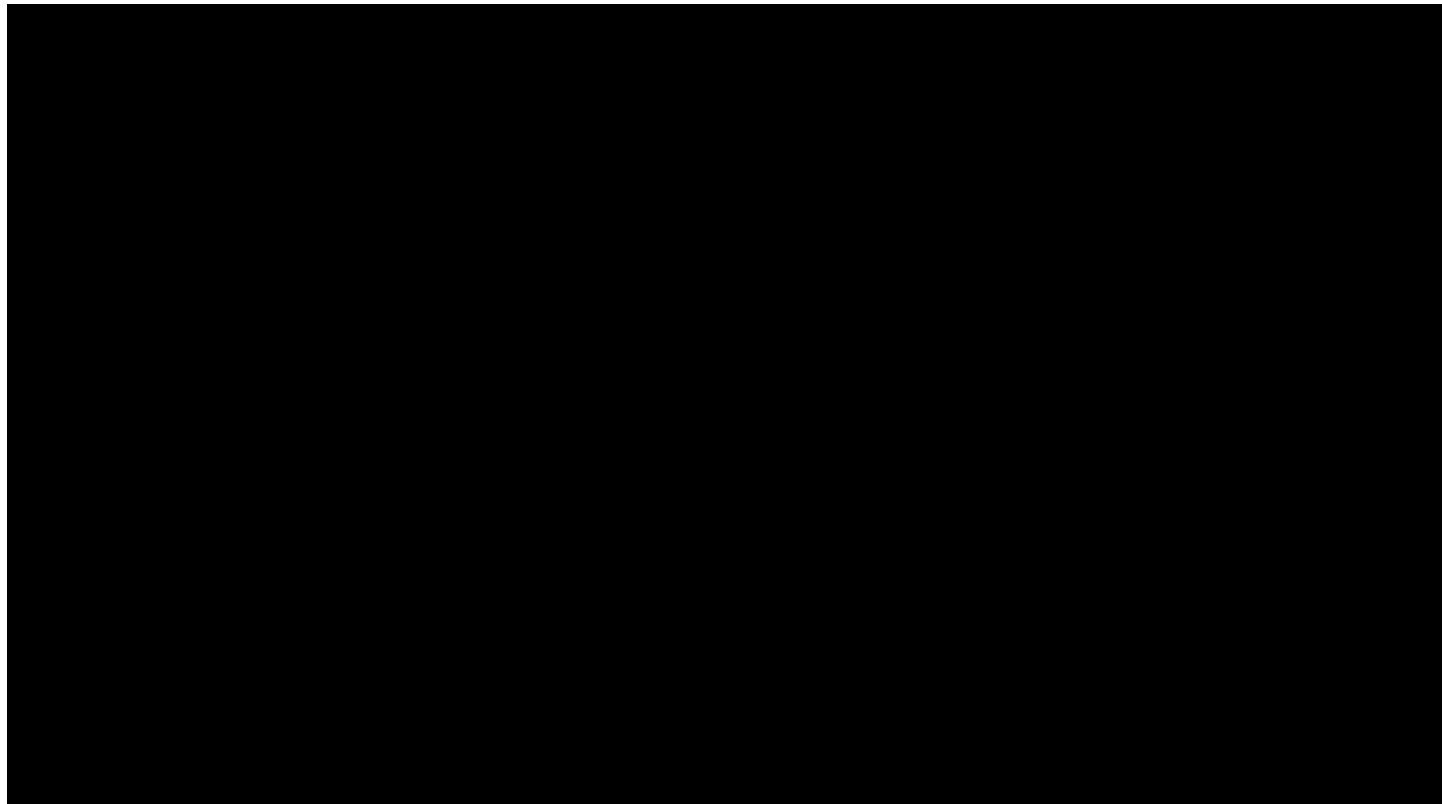
# Kašnjenje i distribucija stanja virtualnog svijeta – asinkroni model



# Paralelni svjetovi pomaknuti u vremenu



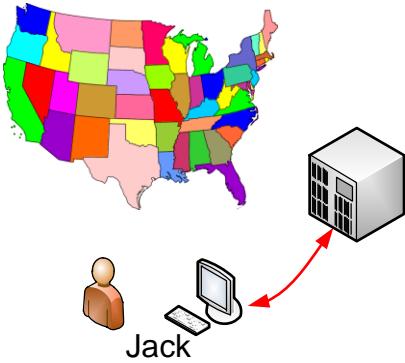
- ◆ Primjer FPS sa 4 igrača (jedan služi kao poslužitelj, a ostali imaju 50, 100, i 150 ms kašnjenja)



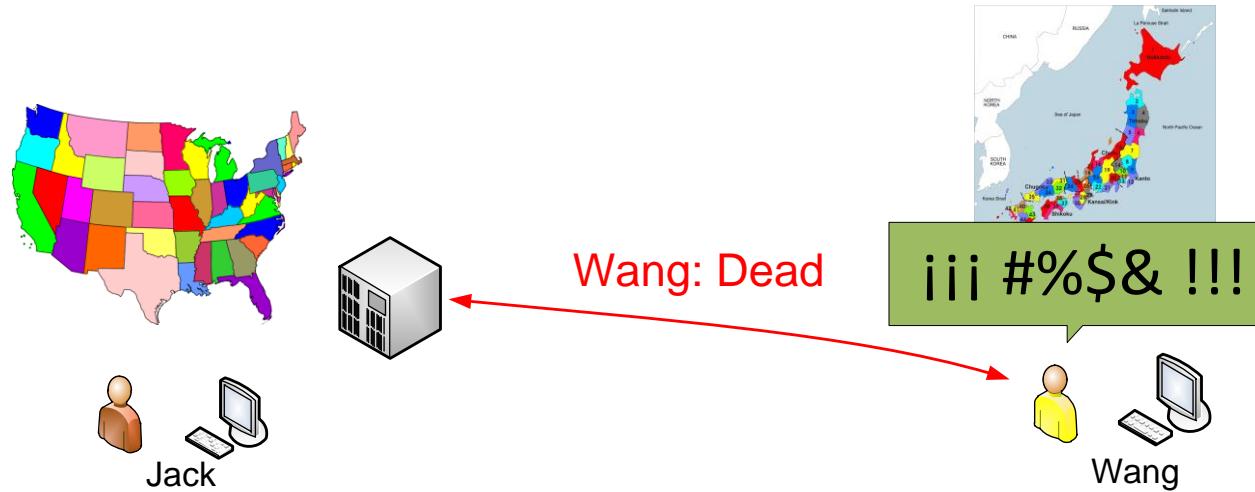
Video URL: <https://www.youtube.com/watch?v=xyCQtUFOJmA#t=728>

# Kašnjenje i konzistentnost stanja - primjer

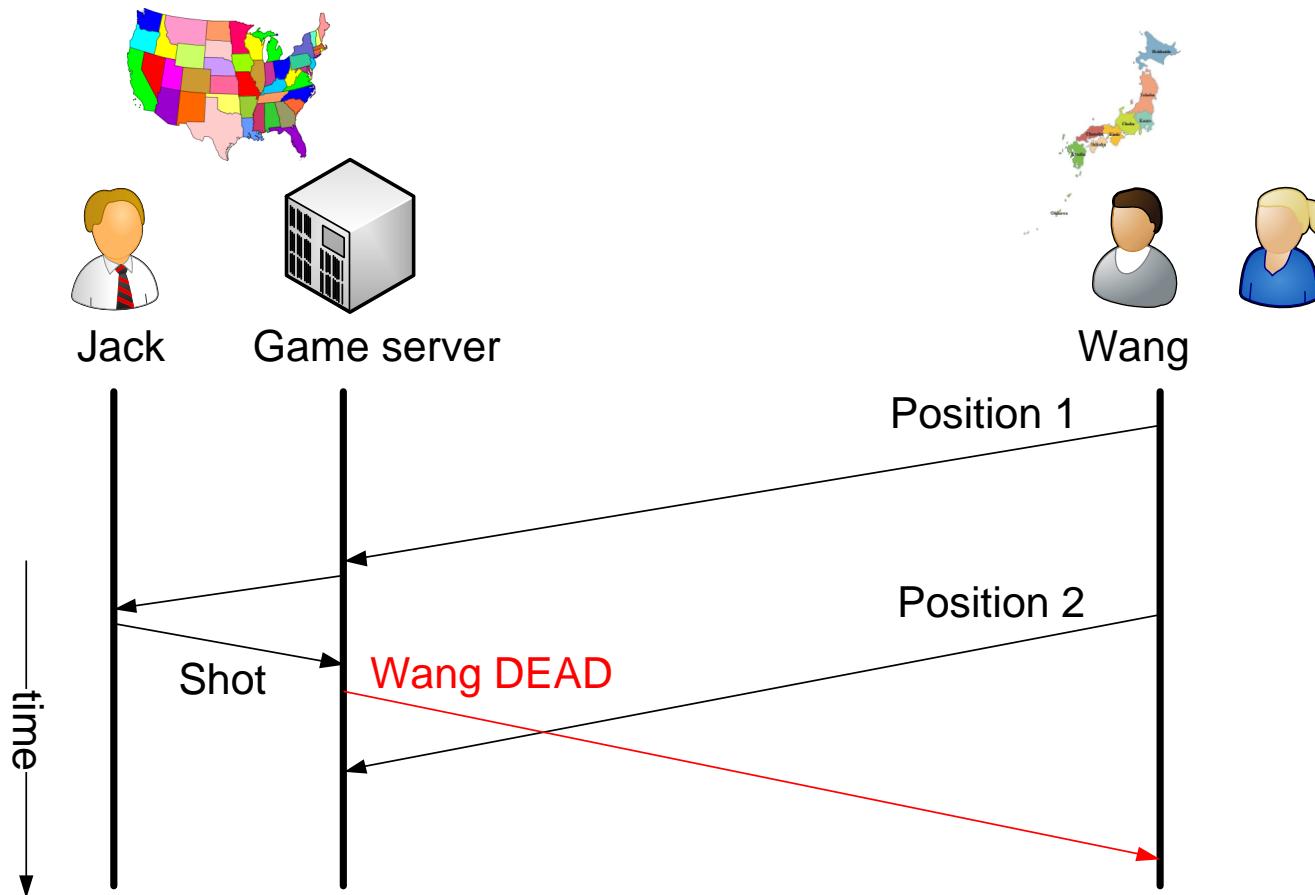
- ◆ Problem smrti iza zida u FPS-u
- ◆ FPS-ovi koriste asinkroni model



# Kašnjenje i konzistentnost stanja – primjer II



# Kašnjenje i konzistentnost stanja – primjer III



***Kako osigurati pravednost?***

# Utjecaj zemljopisne lokacije na kašnjenje



# Zemljopisna lokacija – FPS

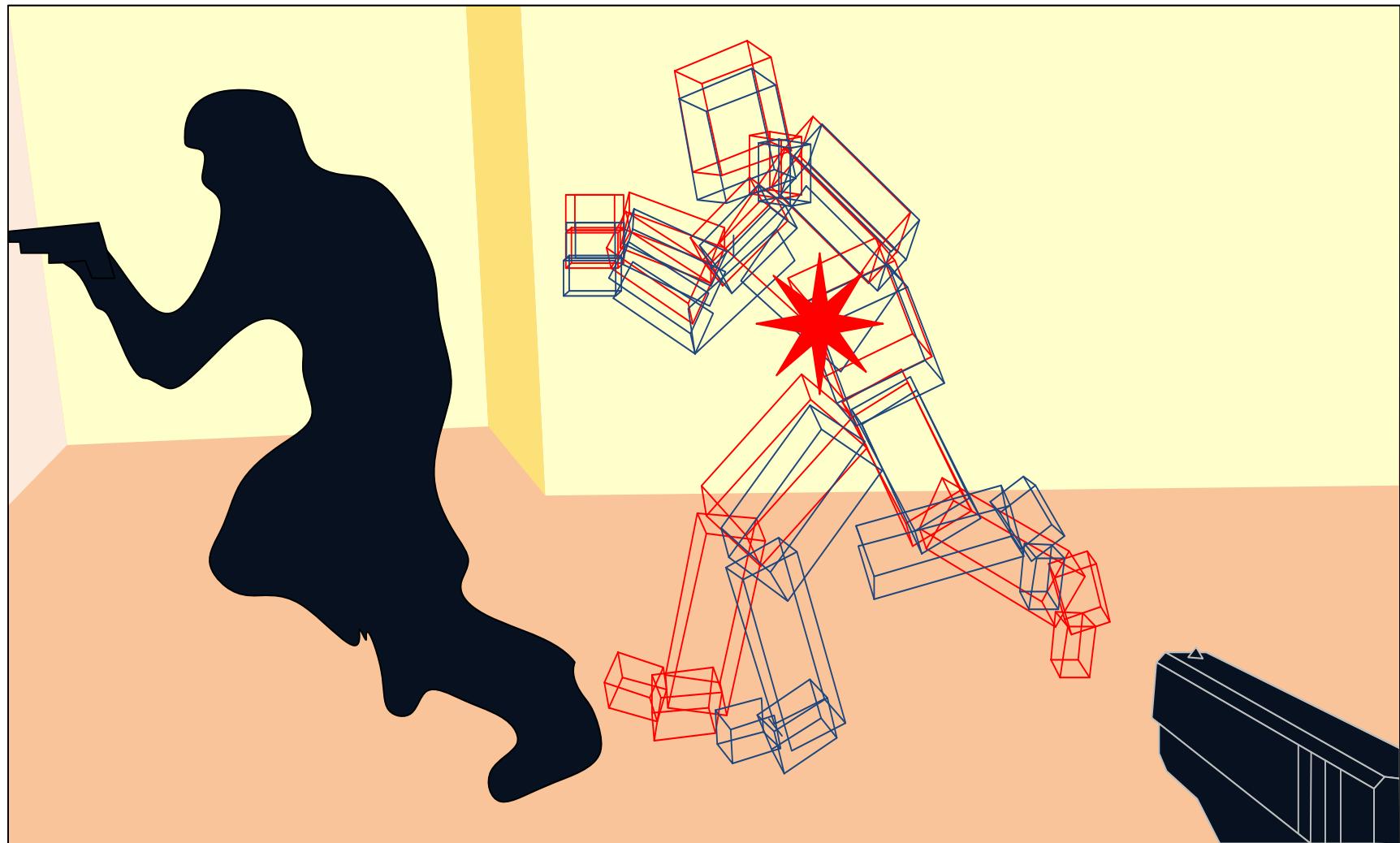
- ◆ Tip igara osjetljiviji na kašnjenje
  - “Time to kill” (vrijeme od prvog pucnja do smrti protivnika) u Call of Duty Black Ops 2 može biti 185 ms!!!
- ◆ Puno veća granulacija poslužitelja
  - Manja složenost poslužitelja
  - Integracijski sustavi koji obuhvaćaju više poslužitelja (statistiku, komunikaciju i sl.)

SEARCH RESULTS      Searching: Call of Duty 2 Servers × in Croatia ×

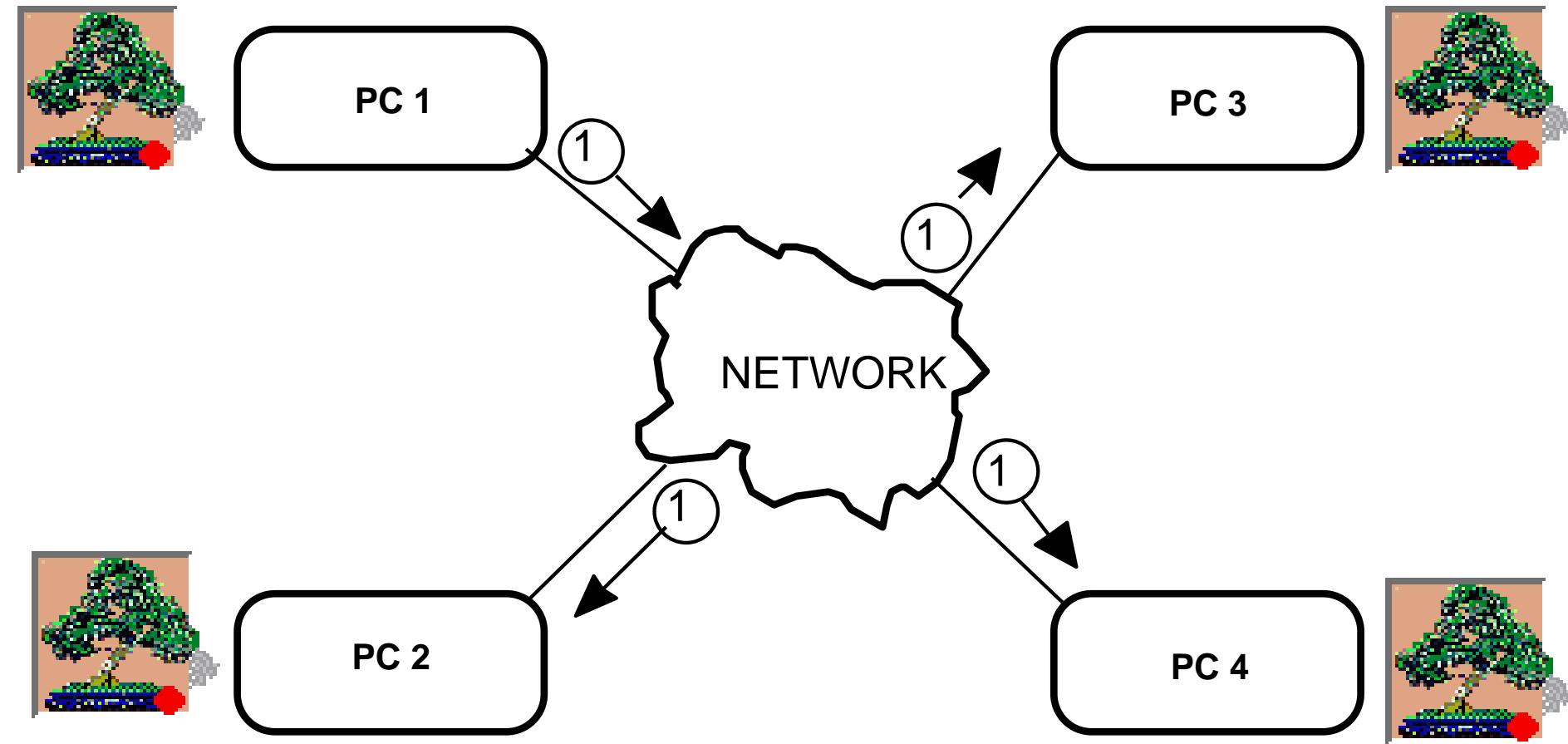
Rank ↓	Gm	Server Name	Players	Loc	IP:Port	Server Map
8.	★ 2	Kameni*Momci Cracked TDM 7/24*hrs	JOIN 24/30	SRB	213.149.60.38:28952	mp_trainstation
11.	★ 2	OMNIGENUS CALLOFDUTYMANIA SD SERVER    omnigenus	JOIN 23/26	SRB	213.149.60.42:29008	mp_matmata
32.	★ 2	Lan-Wars.com Toujane TDM Only TS3:ts.lan-wars.com	JOIN 25/28	SRB	212.92.192.214:28960	mp_toujane
36.	★ 2	Crn@M@mb@CrackedExtrem1.3v	JOIN 16/30	SRB	213.149.60.35:28839	mp_breakout
43.	★ 2	Jigsaw'Networks ~ Rifle/Scope & Toujane ~ 24/7 ~ Li	JOIN 7/28	SRB	213.149.60.34:28952	mp_toujane
49.	★ 2	Legionari Team Cracked v1.3	JOIN 15/28	SRB	213.149.60.35:28928	mp_decoy
59.	★ 2	e-sport.hr DM Server   Powerd by Omnipgenus!	JOIN 16/22	SRB	213.149.60.44:28989	mp_toujane
60.	★ 2	ZiZ  Public Server All Weapons [SD] www.ziz-clan.co	JOIN 2/24	SRB	213.149.60.35:28957	mp_toujane
81.	★ 2	Drunk'Clan   All RiFleS MoD   ToujaNe OnLy   ~HardC	JOIN 20/20	SRB	85.94.70.50:28870	mp_toujane
97.	★ 2	CRO JUDGES     ALL RIFLE WEAPONS   TDM   Powered	JOIN 18/26	SRB	82.193.210.125:28961	mp_toujane
101.	★ 2	Alcoholic-Team  Public server	JOIN 20/28	SRB	213.149.60.37:28993	mp_toujane
107.	★ 2	EAGLESS SD ~ PUBLIC     RIFLES & SNIPERS     1.0v	JOIN 19/32	SRB	46.4.19.83:28942	mp_toujane
113.	★ 2	QUBiC GaminG     SYMBOLIC x Team     Public SD Serv	JOIN 12/16	SRB	213.149.60.36:28945	mp_toujane

- ◆ Predikcija na strani klijenta
  - Predikcija kretanja samog igrača - primjerice izvršavanje komande prije nego što ju je poslužitelj autorizirao
  - Predikcija kretanja drugih entiteta (dead reckoning) - na temelju dosadašnjeg vektora kretanja izračunava se novo stanje iako osvježenje stanja nije došlo na vrijeme
- ◆ Mehanizmi na strani poslužitelja
  - Veća geografska granulacija poslužitelja (što manje propagacijsko kašnjenje)
  - Vremenski odmak u izračunu stanja nakon dolaska komandi kako bi se kompenziralo klijente s većim kašnjenjem
  - “Prenos vremena” – poslužitelj čuva prošla stanja te za pojedinog klijenta prenosi vrijeme kako bi znao koje točno stanje klijent vidi u pojedinom trenutku

# Kompenzacija kašnjenja na strani poslužitelja – premotavanje vremena



# Kašnjenje i distribucija stanja virtualnog svijeta – sinkroni model



# Primjer sinkronog modela – Starcraft 2



- ◆ Starcraft 2 koristi sinkroni model determinističke simulacije
  - Arhitektura je P2P kroz poslužitelj
  - Poslužitelj ne drži repliku stanja igre na klijentima
  - Igra je potpuno deterministička – isti unosi na svim klijentima daju istovjetne rezultate
  - Komanda svakog igrača se stavlja u rep čekanja te izvršava u određenom trenu u budućnosti (200ms).
  - Svaki igrač šalje svoje komande ostalim sudionicima u meču kroz poslužitelj
  - Kada su svi unosi primljeni za prethodni period novo stanje “tick” se izračunava na strani svakog klijenta

- ◆ Prednosti

- Puno je jednostavnija obrada komandi i njihova sinkronizacija e nego da se šalje informacija o pokretu i interakciji svake pojedine jedinice
- Značajno manje korištenje mrežnog prometa

- ◆ Mane

- Postoji kašnjenje u “unisu” komandi – igra ne reagira odmah
- Najsporiji igrač može usporiti igru za druge igrače
- Kako stanje se ne čuva na centralnom poslužitelju može doći do desinkronizacije

# Skalabilnost

- ◆ Skalabilnost – koliko neki sustav može rasti, a da se pri tom ne naruši njegova funkcija
- ◆ Virtualni svjetovi
  - Najčešće kvazi-srednjovjekovna okruženja inspirirana literaturom iz žanra fantastike, uključuju čarobnjake, vitezove, patuljke...
  - Rjeđe futuristički ili iz današnjeg vremena
- ◆ Igrači
  - Preuzimaju uloge jednog virtualnog lika
  - Uloge mogu biti različite lovac, čarobnjak, trgovac
  - Likovi se razvijaju i personaliziraju tijekom igre
  - Igrači često trebaju uskladiti svoje stvarne obveze
- ◆ Aktivnosti u virtualnom svijetu
  - Istraživanje svijeta, suradnju s drugim igračima, trgovinu, borbu...
  - Ciljevi često zahtijevaju suradnju više igrača

# MMORPG - primjeri

FER

Walk here

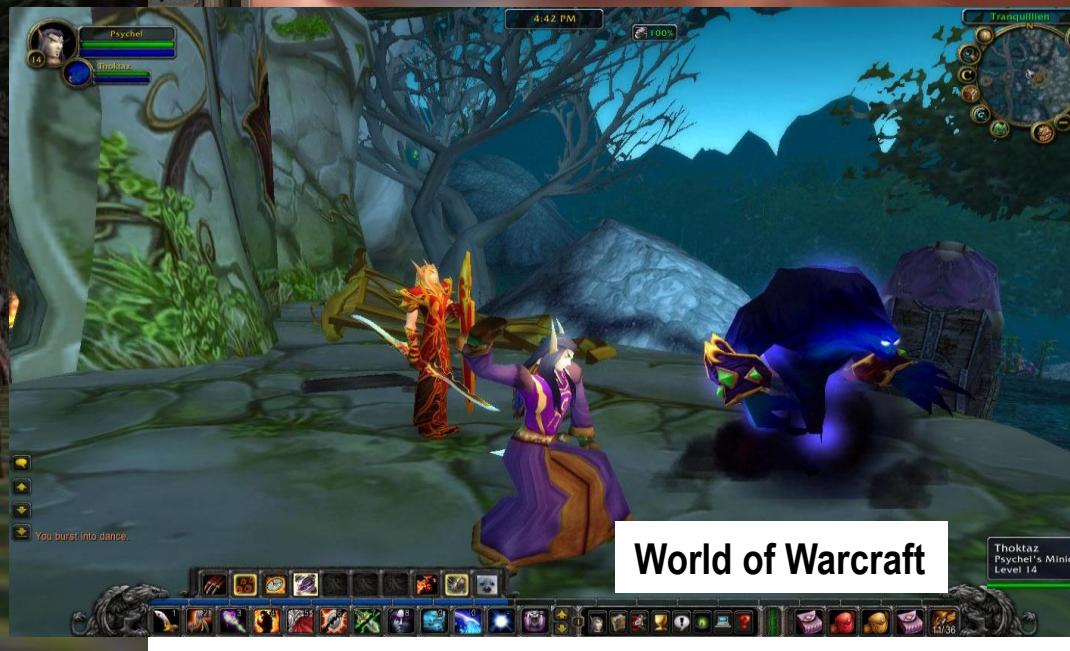
RuneScape



EVE Online



EverQuest

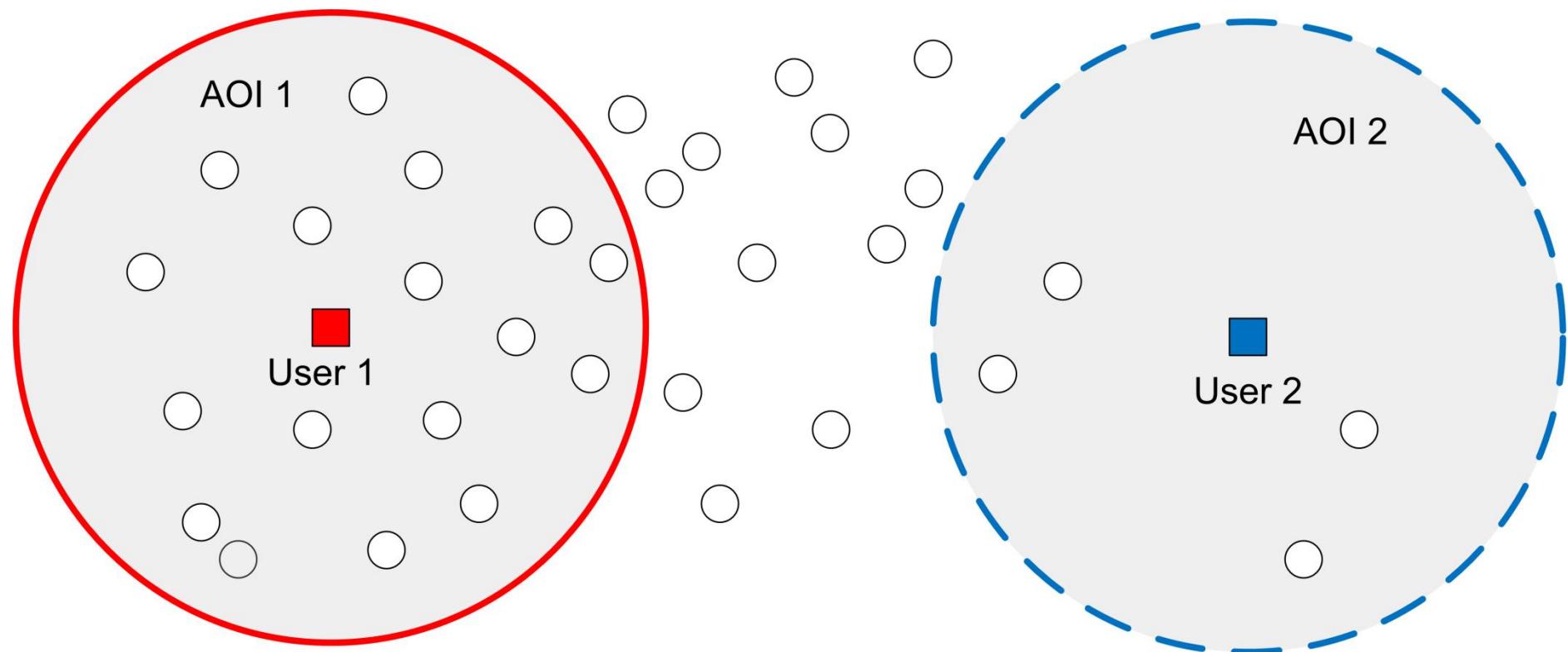


World of Warcraft

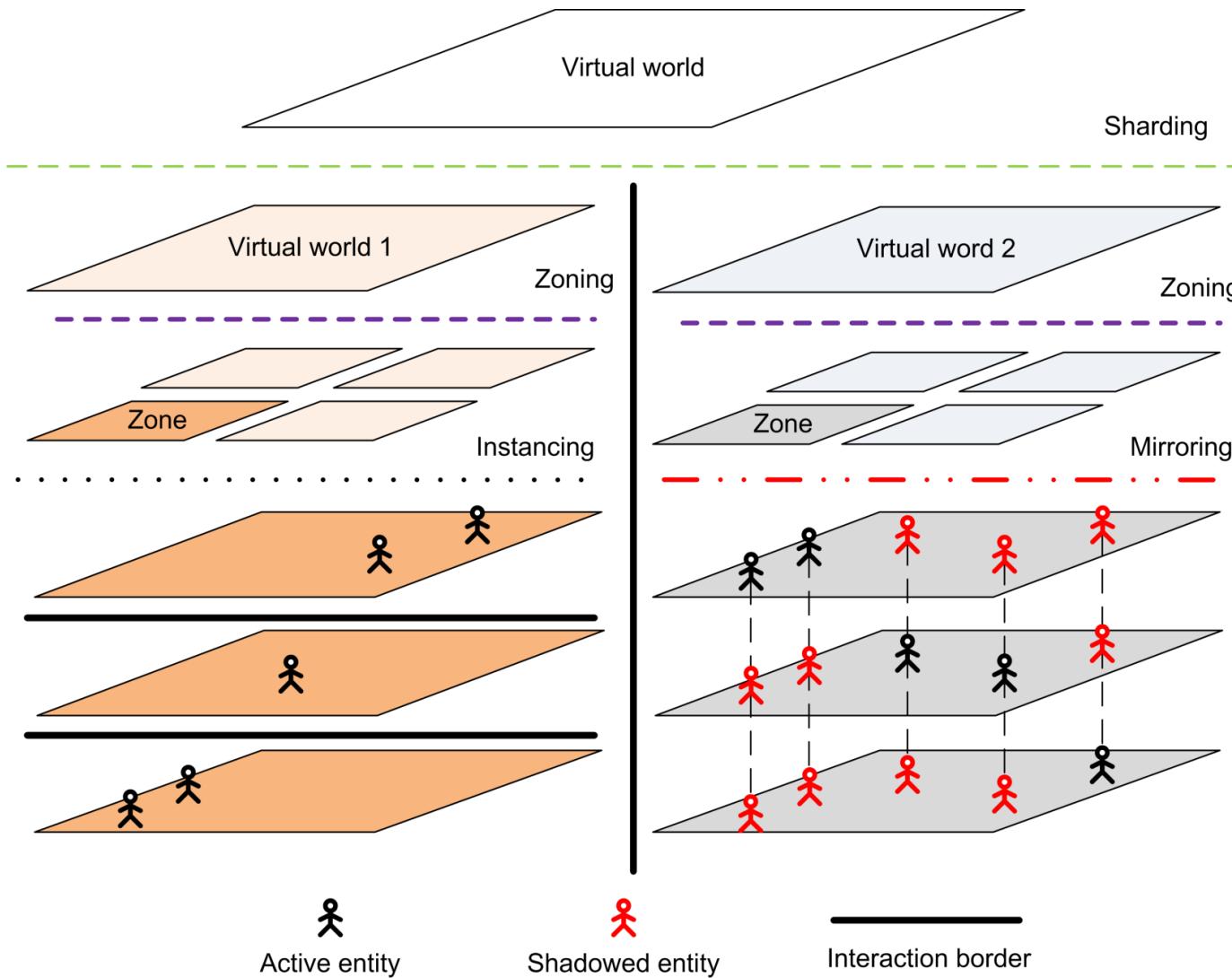
- ◆ **Massively** Multiplayer Online Role-Playing Games
  - Veliki broj korisnika koji dijele jedan virtualni svijet
  - WoW – 12 miliona korisnika (na vrhuncu popularnosti)
- ◆ Glavni problemi:
  - Izračunavanje stanja virtualnog svijeta ( $O(n^2)$ )
  - Održavanje konzistentnosti stanja virtualnog svijeta
  - Kontrola varanja
  - Svi poslužitelji na strani proizvođača

# Tehnike za skalabilnost: područje interesa

- ◆ Područje interesa (Area of Interest – AOI)



# Tehnike za skalabilnost



- ◆ **Zoniranje** – podjela virtualnog svijeta na geografska područja koja će se samostalno obraditi odvojenim strojevima
- ◆ **Zrcaljenje** – način distribucije opterećenja umnožavanjem iste zone igre na nekoliko poslužitelja. Svaki replicirani poslužitelj izračunava stanje za podskup entiteta koji se naziva aktivni entiteti, dok se preostali, koji se nazivaju entitetima sjene (koji su aktivni u drugim poslužiteljima koji sudjeluju), sinkroniziraju preko poslužitelja
- ◆ **Instanciranje** – pojednostavljenje zrcaljenja koje distribuira opterećenje sesije pokretanjem više paralelnih instanci visoko naseljenih zona. Igrači u različitiminstancama ne mogu komunicirati

Prodan, Radu, and Vlad Nae. "Prediction-based real-time resource provisioning for massively multiplayer online games." *Future Generation Computer Systems* 25.7 (2009): 785-793.

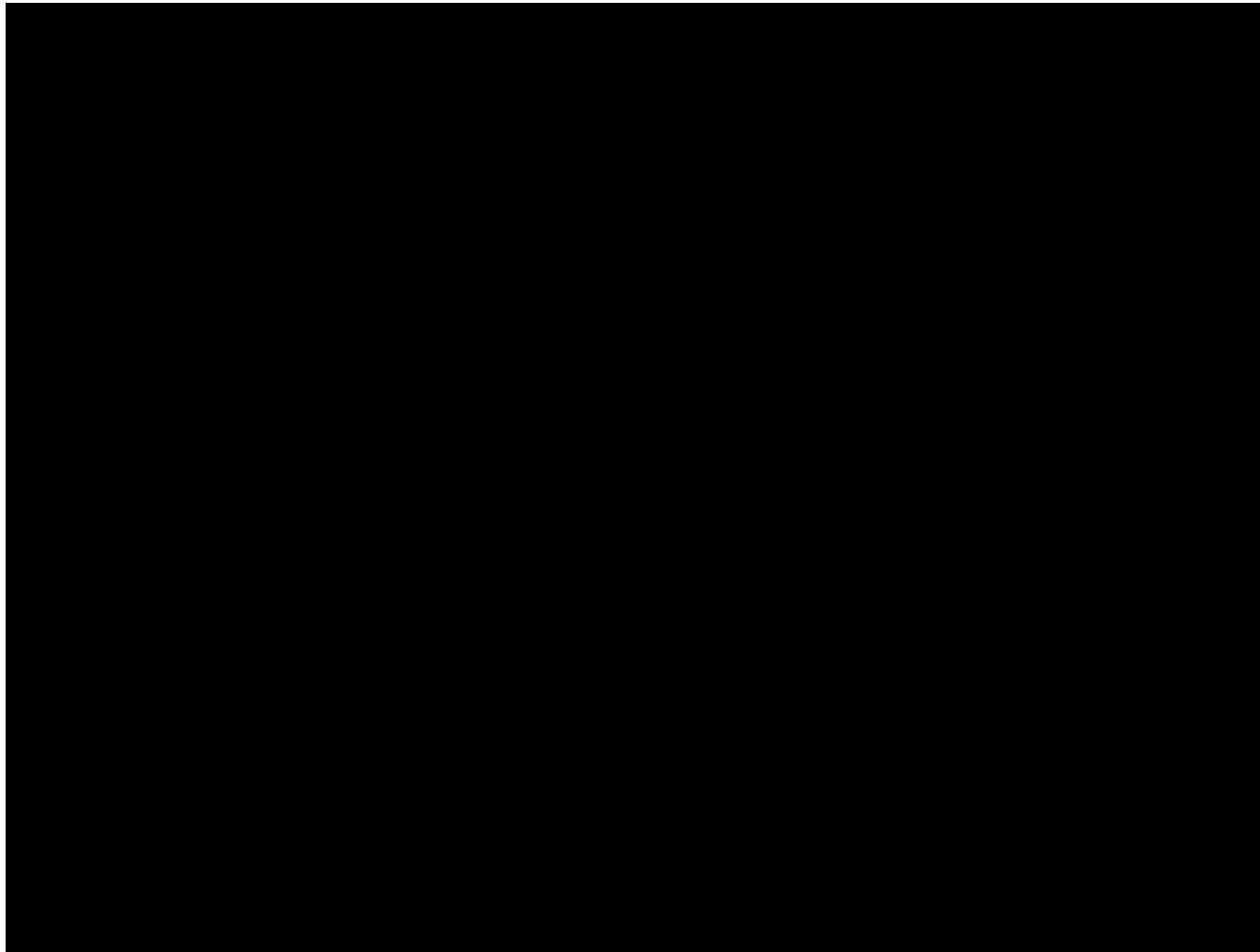
- ◆ Dva arhitekturalna rješenja
  - Velike poslužiteljske farme
  - “Komadanje” virtualnog svijeta (engl. žargon: server “shards”)
- ◆ World of Warcraft-a (2009.):
  - 13 250 poslužitelja
  - 75 000 CPU jezgri
  - 11,5 TB blade RAM
  - 11 podatkovnih centara u svijetu (USA, Europe, China, Australia...)
- ◆ EvE online (2013.):
  - Supercomputer “Tranquility”, u Londonu, UK
  - 3 936 GB RAM
  - 2 574 GHz of CPU snage

# Farma poslužitelja (grozd)

- ◆ Svi korisnici unutar istog virtualnog svijeta (EvE online, World of Tanks)
- ◆ Velike farme poslužitelja
- ◆ Problemi s izračunom stanja virtualnog svijeta (moguć veliki broj korisnika na jednom mjestu)

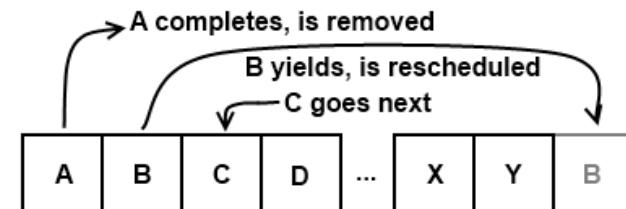


# The Asakai incident



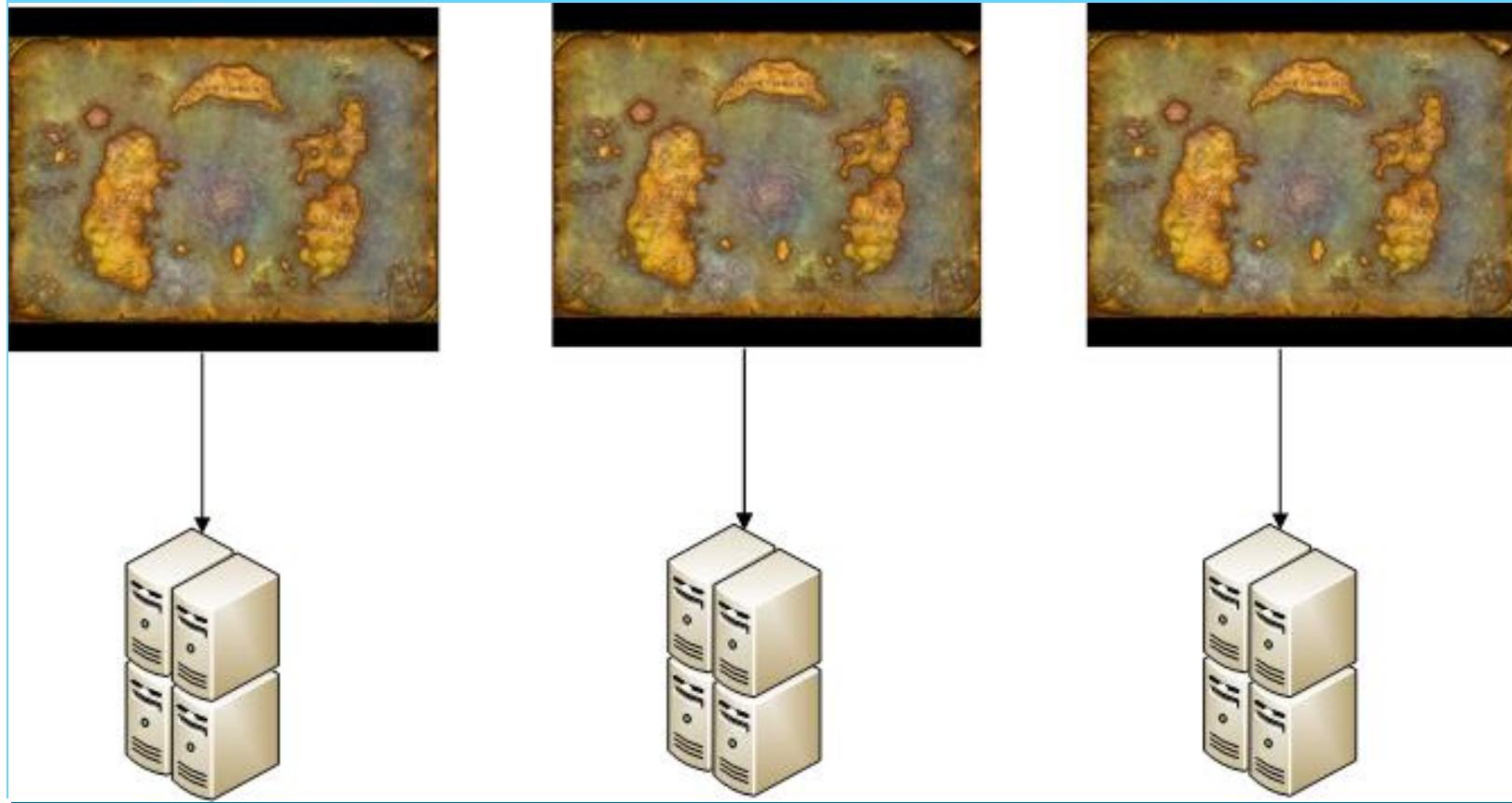
Video URL: [https://www.youtube.com/watch?v=\\_jQw3YcLoQU](https://www.youtube.com/watch?v=_jQw3YcLoQU)

- ◆ Rastezanje vremena (engl. time dilation)
- ◆ EvE online funkcionira na sustavu zadataka (svaki korisnički ulaz, stanje pojedinog modula itd.)
- ◆ Raspoređivač (engl. scheduler) raspoređuje zadatke
- ◆ EvE online koristi se raspoređivanje unaokolo (engl. round robin) – svaki zadatak ima mogućnost ili se izvršiti ili odustaje daje priliku drugim zadacima čime ponovno ide na kraj repa čekanja
- ◆ Odustajanje
  - Čekanje na informacije iz drugih zadataka (primjerice baza podataka)
  - Dugi zadaci koji trebaju više vremena mogu pustiti hitnije zadatke
  - Pod velikim opterećenjem nakon odustajanja može doći do velikog čekanja (primjerice uništenje broda koji koristi jako puno interakcija s bazom podataka)
  - Veliko čekanje – nerealistično ponašanje, gubitak kauzalnosti
- ◆ Kako smanjiti čekanje?
  - Povećati procesorske mogućnosti
    - Optimizacije – korištenje višedretvenih sustava može pomoći do određene razine
    - Kupovina nove opreme - povećava troškove
  - Smanjiti opterećenje
    - Metode dizajna igre – ograničavajuće za igrače
    - Usporavanje vremena unutar igre – većina zadataka unutar igre vezana je za vrijeme (moduli, fizika, putovanja, ulasci brodova u sustave – sve je vezano za vrijeme unutar igre)

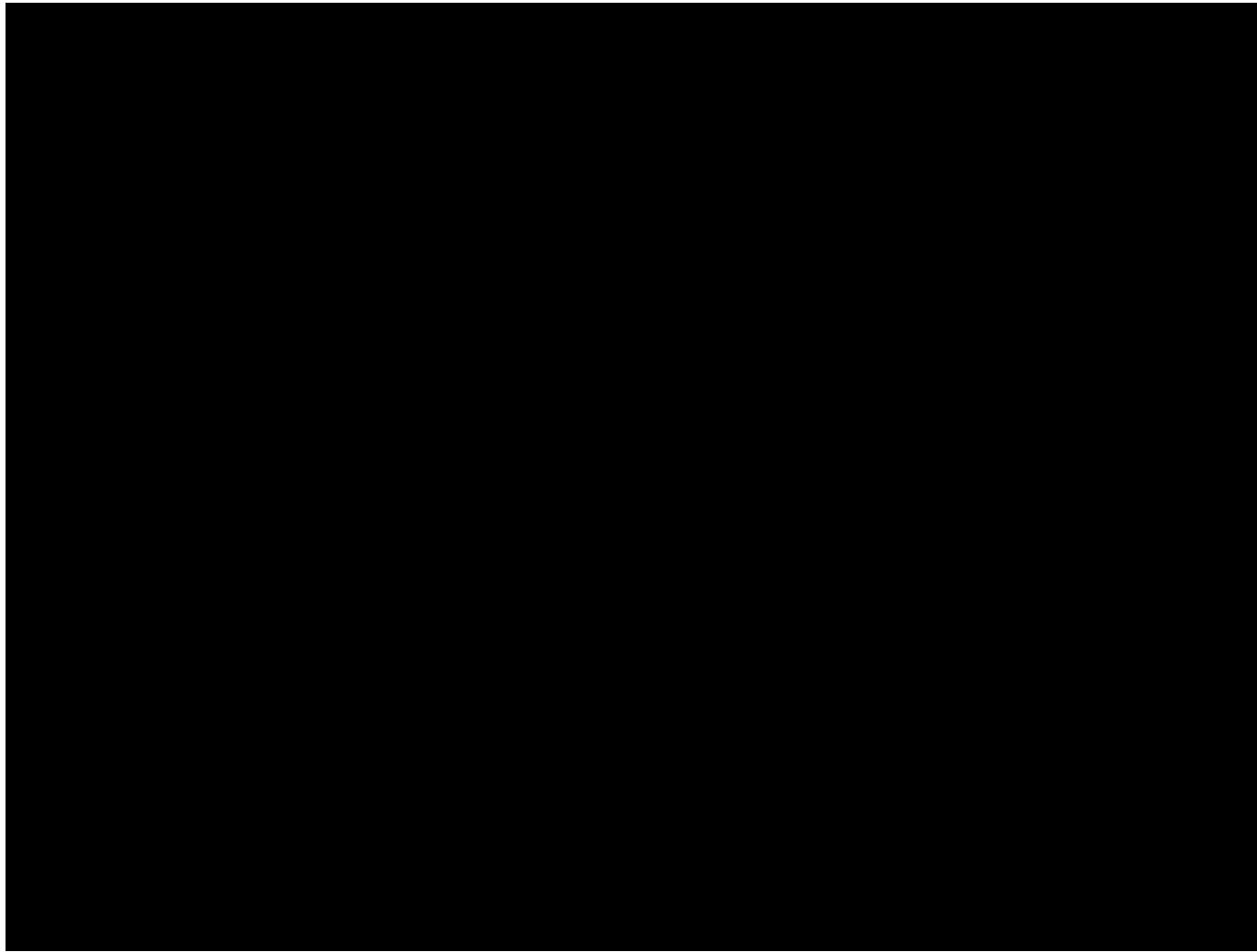


Izvor: <https://www.eveonline.com/article/introducing-time-dilation-tidi>

- ◆ Repliciran cijeli virtualni svijet na svakom “komadu”
- ◆ Razdvojeni korisnici
- ◆ Smanjenje skale (sa 10 miliona na nekoliko tisuća)



# War without the warchief



Video URL: <https://www.youtube.com/watch?v=ZzsliSTnQfI>

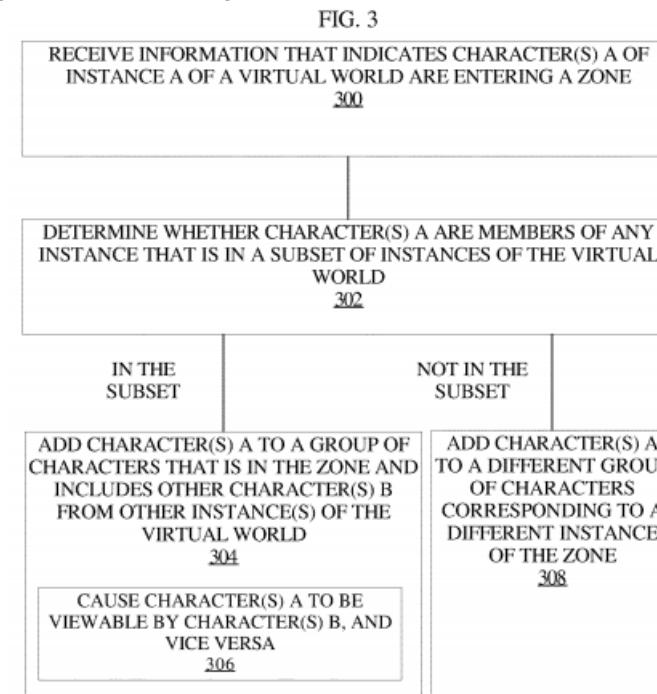
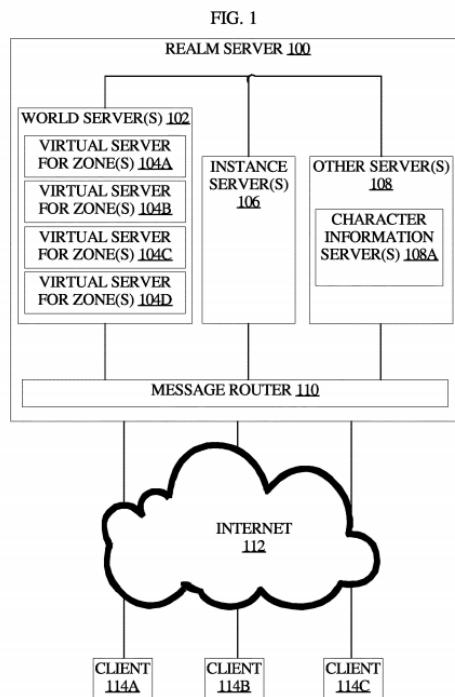
# Izlazak Warlords of Draenor ekspanzije za World of Warcraft



- ◆ 13.11.2014
- ◆ Veliki problemi s poslužiteljima
- ◆ Usluga je bila djelomično ili u potpunosti nedostupna 4 dana
- ◆ Riješen problem kroz ograničavanje broja igrača na pojedinom poslužitelju – uzrokovalo redove čekanja ponekad i po nekoliko sati
- ◆ ***Skalabilnost u velikim virtualnim svjetovima je još uvijek neriješen problem!***

# Dinamičke zone

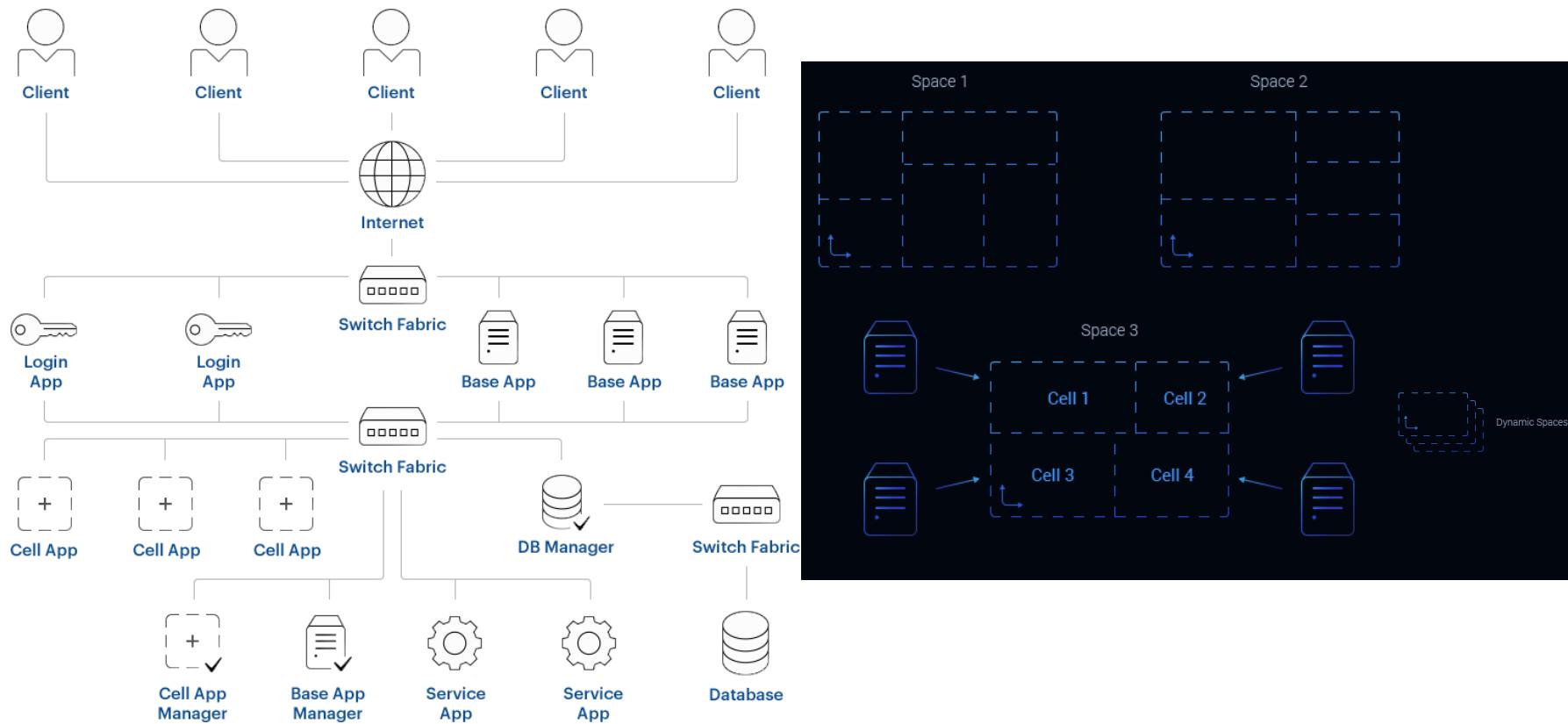
- World of Warcraft sa smanjenjem broja ljudi uvodi "cross-realm" mogućnost – dinamička pripadnost pojedinog korisnika određenoj zoni
  - Korisnik može se u stvarnom vremenu transferirati između zona
  - Ograničenje broja korisnika u pojedinoj zoni
  - Nema više „mrtvih“ zona odnosno zona s jako malim brojem korisnika



Izvor: Patent No: US 10 , 086 , 279 B2

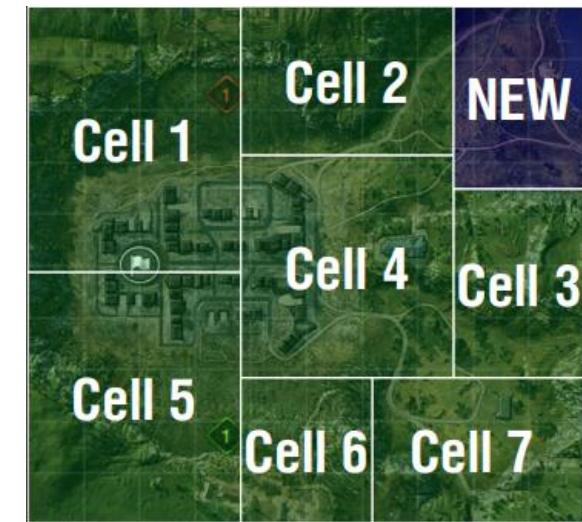
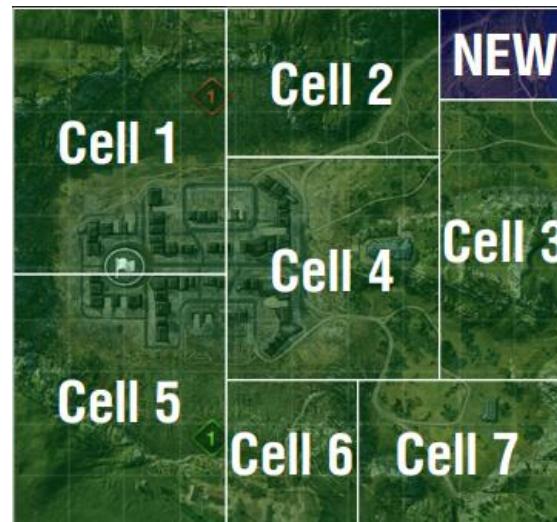
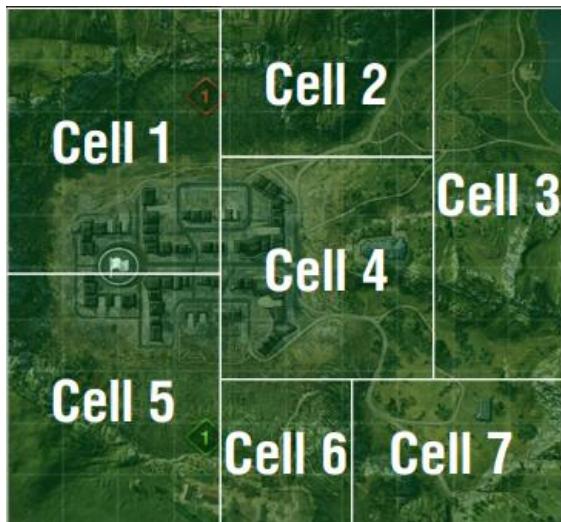
# Riješenja za skalabilnost– BigWorld (Wargaming)

- ◆ Dinamička prilagodba veličine područja u virtualnom svijetu koje obrađuje pojedini poslužitelj u ovisnosti o broju korisnika
- ◆ Ukoliko neko područje ima smanjeni broj korisnika njegovo područje se povećava
- ◆ Uspješno simulirano čak 100 000 korisnika



# Princip funkcioniranja

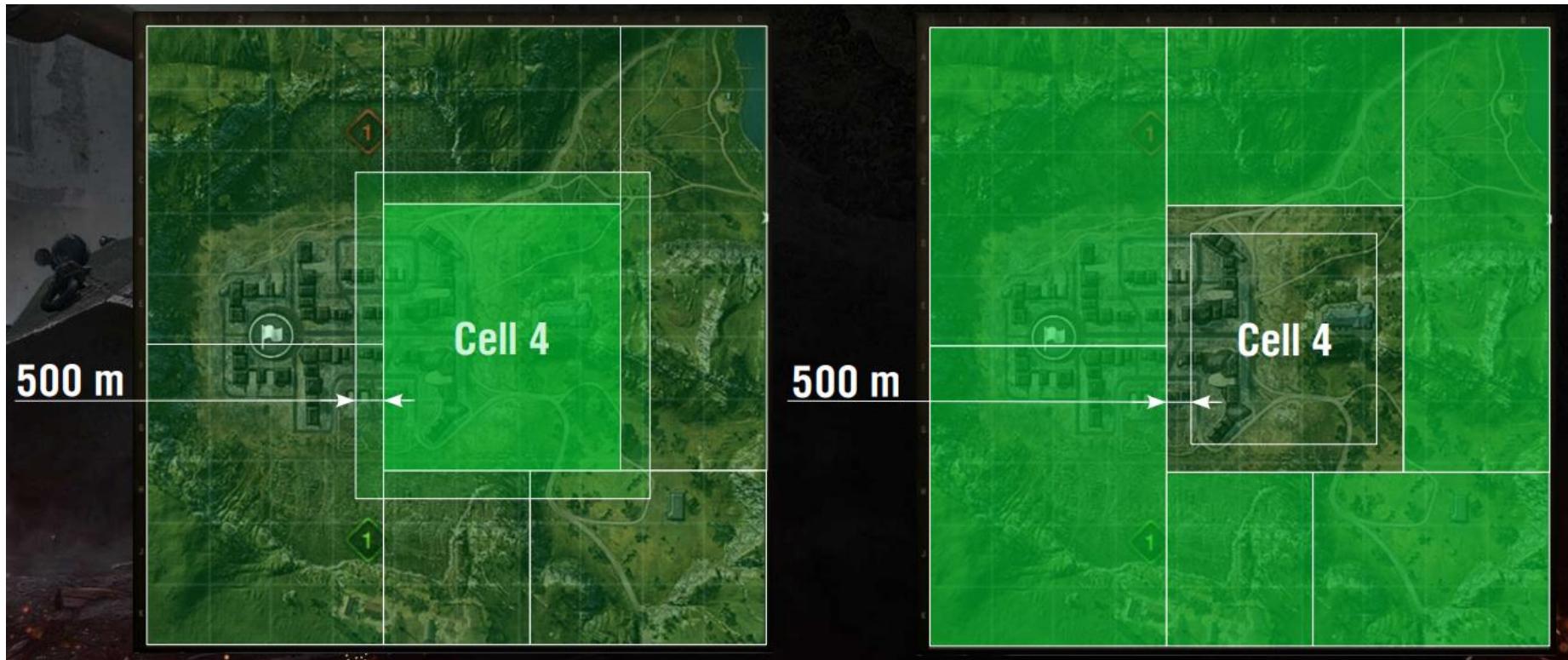
- ◆ Ćelijsko opterećenje – količina vremena koju stanica troši pri izračunu jednog ticka igre podijeljena s duljinom ticka
- ◆ CellAppMgr – mijenja veličine stanica u stvarnom vremenu kako bi zadržite opterećenje svake ćelije ispod postavljenog praga
- ◆ CellApMgr – može dodati i nove ćelije



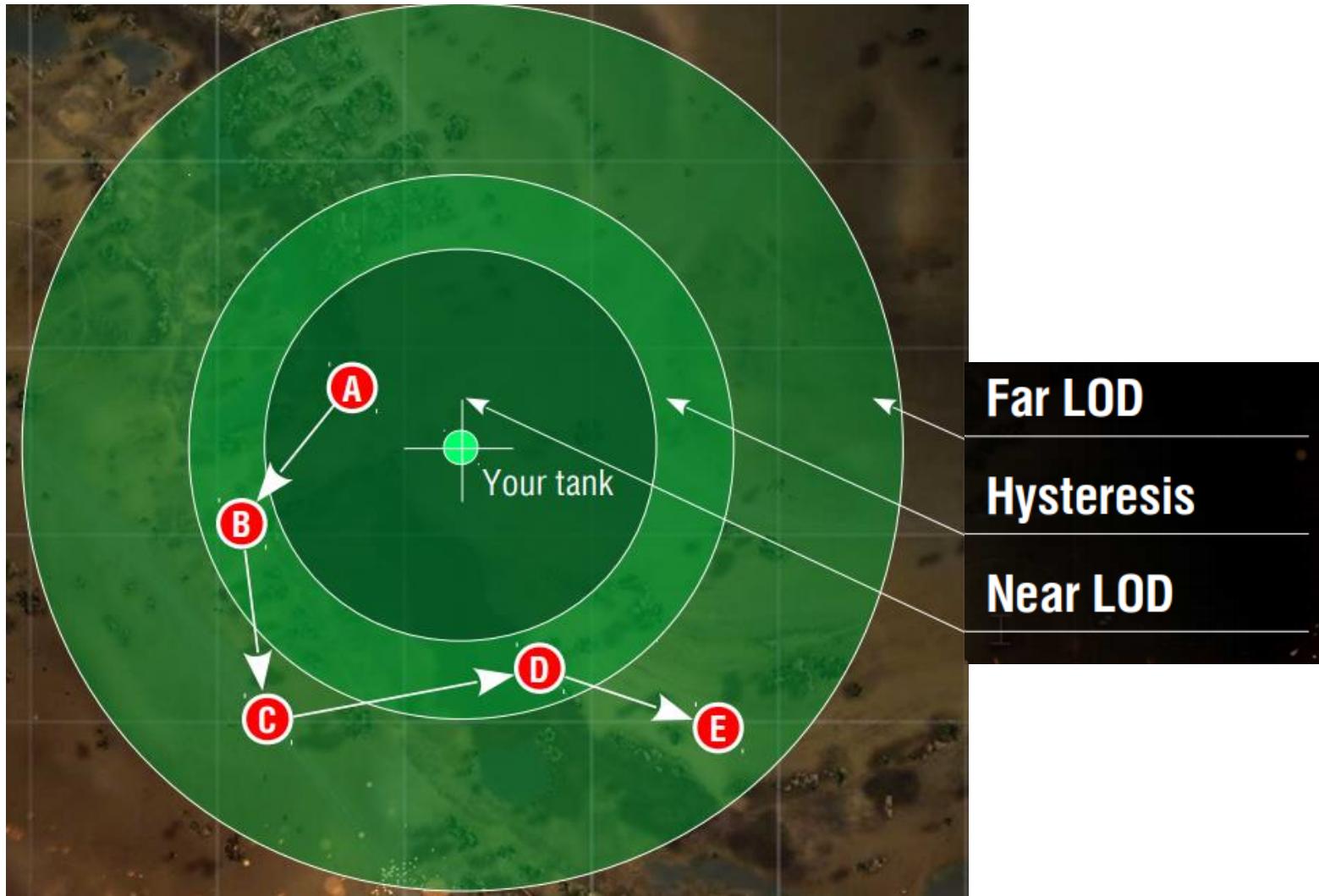
Izvor: <http://netgames2015.fer.hr/presentations/MaksimBaryshnikov.pdf>

# Spriječavanje čestog prebacivanja ćelija

- Postoji udaljenost koja se mora preći nakon granice ćelije da bi došlo do prebačaja u iduću ćeliju – histereza



# Dinamička promjena nivoa detalja





Diplomski studij

Informacijska i  
komunikacijska tehnologija:

Obradba informacija  
Telekomunikacije i informatika

# Višemedijske komunikacije

14.

Prijenos govora protokolom IP (VoIP)  
Protokoli i standardi za signalizaciju  
(prijenos kontrolne informacije): SIP, H.323

# Pojmovi i terminologija

- **sjednica**
  - pojam koji obuhvaća grupu pošiljatelja i primatelja (više)medijskog sadržaja te podatkovne tokove između njih
- VoIP, internetska telefonija, IP telefonija, itd.
- **VoIP (Voice over Internet Protocol)**
  - generički naziv za **svaku govornu komunikaciju** putem protokola IP umjesto tehnologije s **komutacijom kanala** (npr., javna telefonska mreža, PSTN)
- **internetska telefonija**
  - usluga krajnjim korisnicima, tj., posebna vrsta VoIP-a u kojem se poziv ostvaruje kroz javnu infrastrukturu Interneta, uz (djelomično ili potpuno) "zaobilaznje" PSTN-a
  - može se odvijati između računala (računalo - računalo), između računala i (IP) telefona te između (IP) telefona

# Svjetsko VoIP tržište

- Broj VoIP pružatelja usluge: 15300[\*]
- Veliki broj operatera uz PSTN ima i VoIP pozive
- Trend rasta broja VoIP pretplatnika u SAD-u i Evropi (2005 - 2011) [\*\*]; predviđeni broj mobilnih VoIP korisnika po regijama (2010-2012) [\*\*\*]

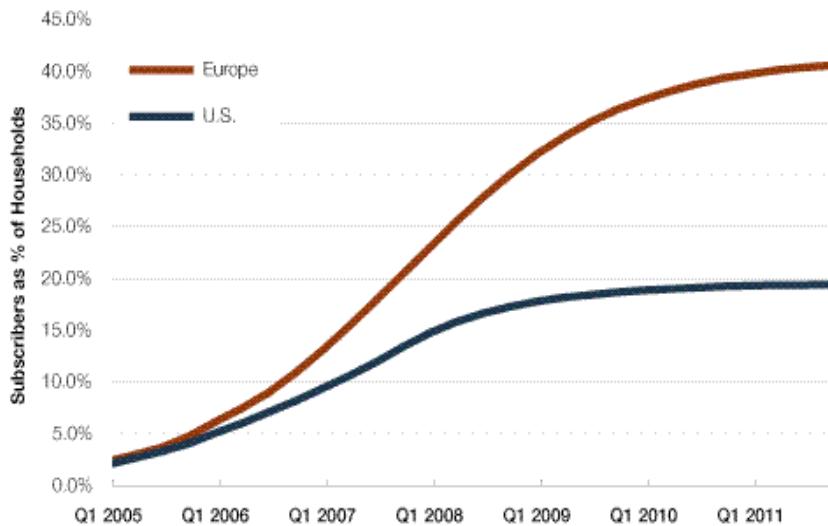
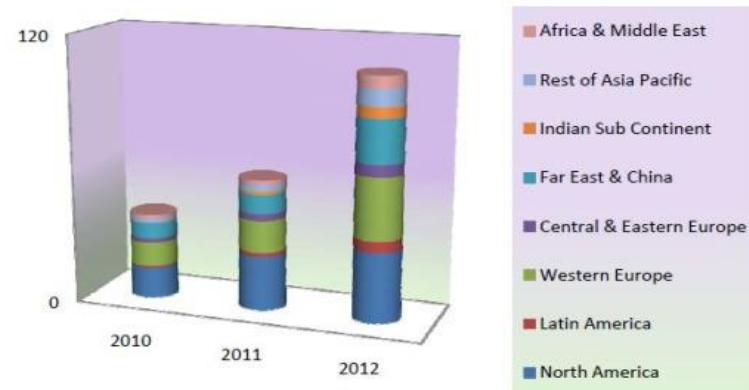


Figure 1: Number of Mobile VoIP users (m) 2010-2012 by 8 Key Regions



Source: Juniper Research

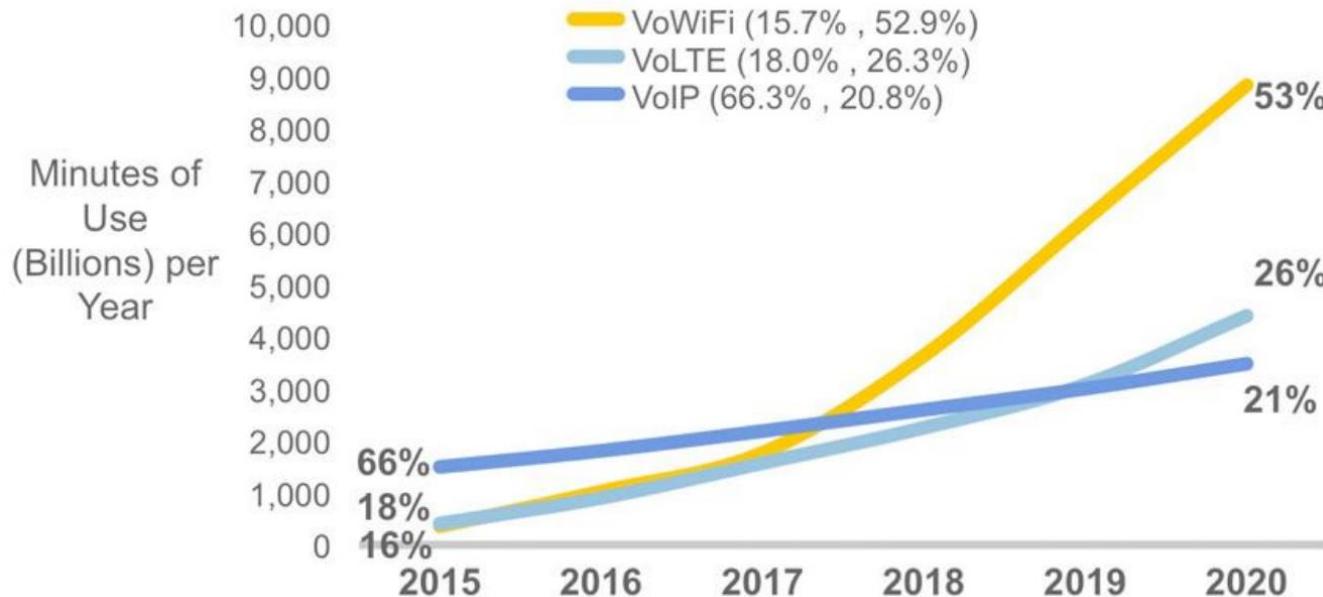
[\*] Izvor: <http://www.voipproviderslist.com/> (stanje 06/2018)

[\*\*] Izvor: <http://ipvoip.blogspot.com/2007/09/us-voip-market-is-growing-fastbut.html> (TeleGeography's US VoIP and Euro-VoIP reports)

[\*\*\*] Izvor: <http://www.marketingcharts.com/direct/107m-mobile-voip-users-expected-by-2012-13061/juniper-mobile-voip-june-2010jpg/> (Juniper Research, 2010)

- Svjetsko VoIP tržište su preuzele mobilne aplikacije!
- Danas sve najpopularnije aplikacije za razmjenu instant poruka podržavaju audio pozive putem VoIP tehnologije
- Broj korisnika:
  - WhatsApp – 1,5 milijardi
  - Facebook Messenger – 1,3 milijarde
  - Viber – 900 milijuna
  - WeChat (Kina) – 900 milijuna
  - Skype – 300 milijuna
- Trend većeg korištenja mobilne u usporedbi s fiksnom telefonijom

- VoIP – tehnologija, ali u ovom slučaju pokazuje komunikaciju Over The Top (OTT) pružatelja usluga
- VoWiFi – VoIP poziv podržan od operatera usluge preko WiFi mreže
- VoLTE – VoIP pozivi podržani od operatera usluge preko LTE



[\*] Izvor CVNI Global Mobile Data Traffic Forecast Update, 2015– 2020  
[https://www.cisco.com/c/dam/m/en\\_innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf](https://www.cisco.com/c/dam/m/en_innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf)

# Prednosti korištenja VoIP-a

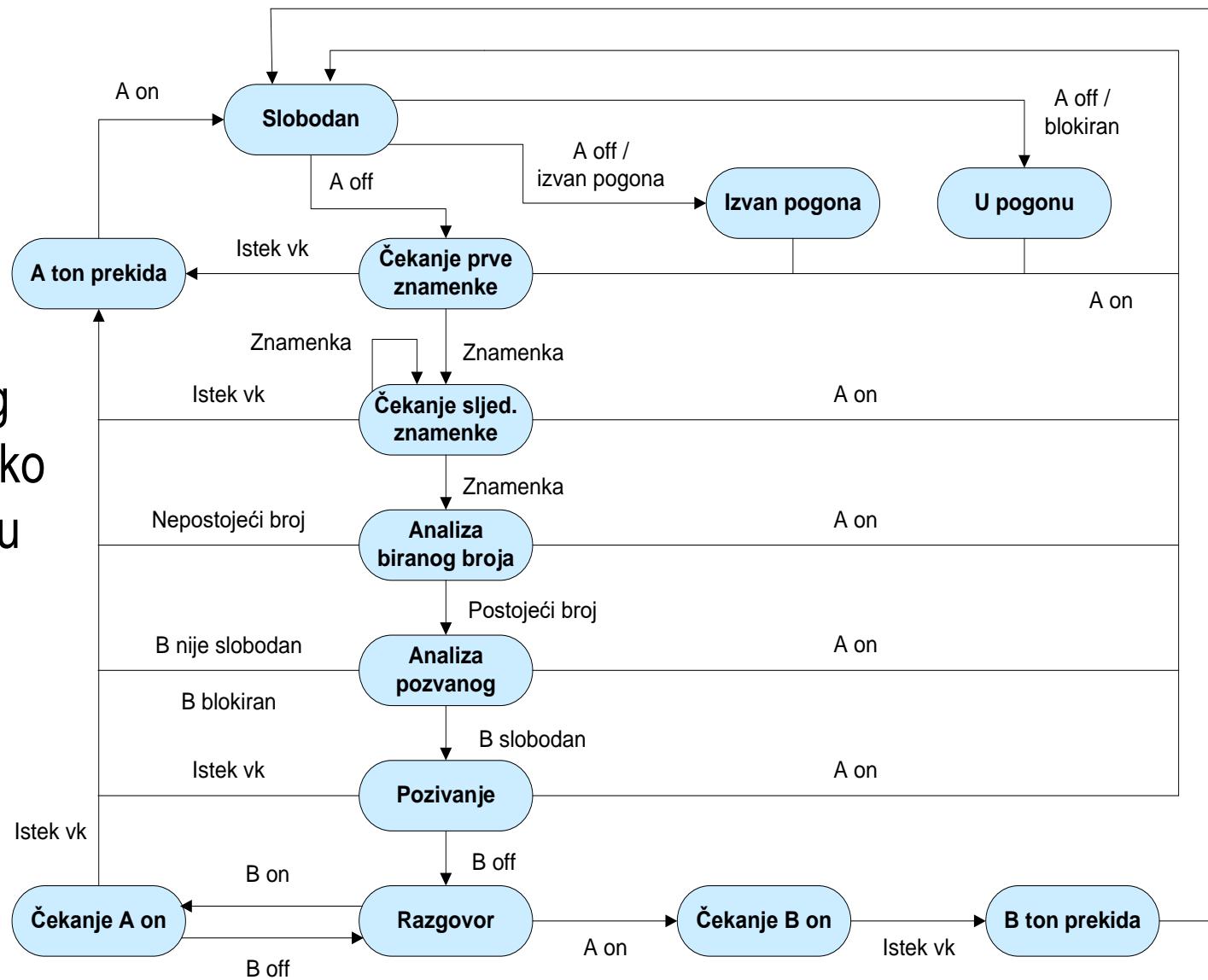
- Iz perspektive krajnjeg korisnika
  - smanjenje troškova
  - dodatne usluge u govornoj komunikaciji dostupne besplatno (npr., skraćeno biranje ili preusmjeravanje poziva)
  - olakšano prenošenje pozivnog broja
  - jednostavnije korištenje drugih vidova komunikacije (npr., video) / drugih aplikacija (konferencijski poziv, dijeljenje podataka)
  - gotovo neograničena dostupnost usluge (uvjetno jednaka širokoj rasprostranjenosti pristupa Internetu)
- Iz perspektive VoIP operatora/pružatelja usluge
  - smanjenje ukupnih troškova (ali, nakon "određenog" razdoblja)
  - jednostavniji instalacija i održavanje opreme/infrastrukture
  - jednostavnije dodavanje/uvođenje novih usluga

# Nedostaci korištenja VoIP-a

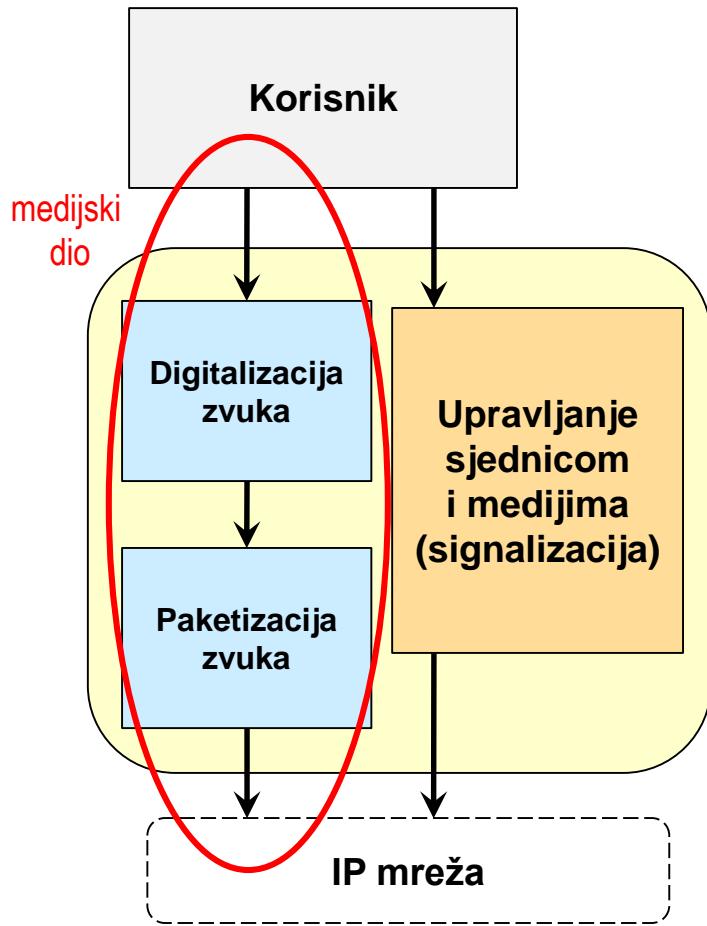
- **Kvaliteta usluge**
  - prijenos IP mrežom može dovesti do kašnjenja/gubitaka paketa koji nose govor
  - dolazi do izražaja u slučajevima velikog broja korisnika i izostanka kontrole pristupa mreži
  - stalnim razvojem tehnologije razlike u odnosu na "klasičnu" telefoniju ipak su sve manje
- **Raspoloživost usluge**
  - ovisi o pouzdanosti mreže (ispadom internetske mreže usluga postaje neraspoloživa)
- **Nekompatibilnost VoIP sustava (uređaja)**
  - nepostojanje jedinstvenog standarda
- **Potreba stalnog napajanja uređaja**
  - za razliku od "klasičnog" telefon. uređaja, u slučaju nestanka napajanja, VoIP neće raditi
- **Sigurnost**
  - prisluškivanje komunikacije

# "Klasična" telefonija - PSTN

- Model telefonskog poziva - kako je izведен u Internetu?



- Pojednostavljeni prikaz – medijski i signalizacijski dio



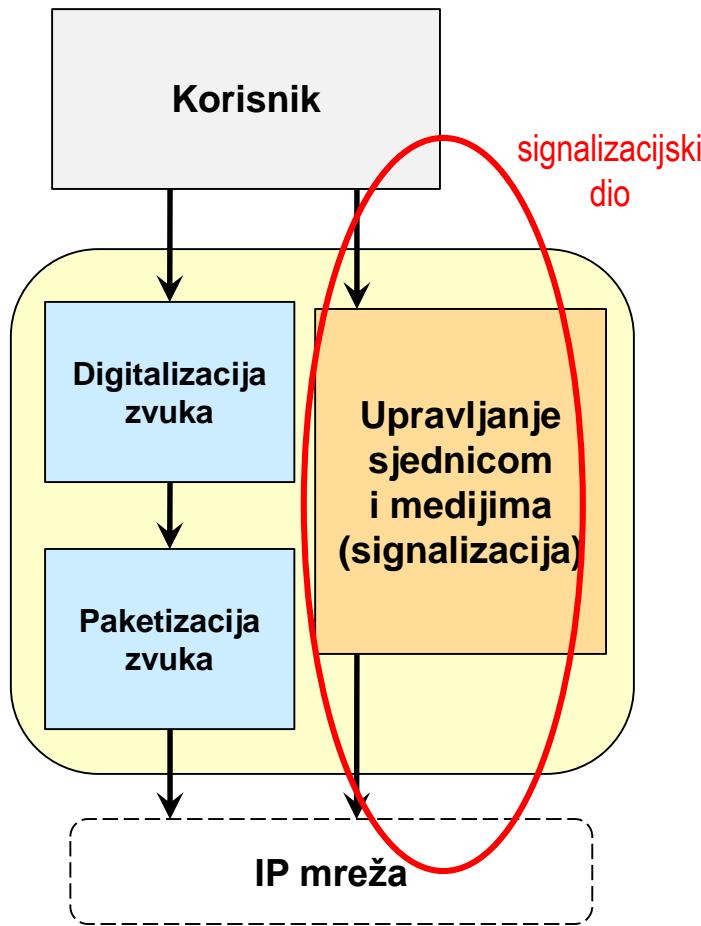
Korisnik priča u mikrofon "klasičnog" ili IP telefona, odnosno mikrofon spojen na osobno računalo

Uredaj digitalizira zvuk korištenjem određenog kodeka (PCM, odnosno ITU-T G.711; ITU-T G.723.1; verzija koder CELP u preporuci ITU-T G.729; Opus; itd. )

Digitalni zvuk se pakira u odgovarajuće protokolne jedinice (pakete) te prenosi infrastrukturom zasnovanom na protokolu IP

IP mreža prenosi datagrame koji u sebi nose govor po istim načelima kao i sav drugi promet ("best-effort" usluga)

- Pojednostavljeni prikaz – medijski i signalizacijski dio



Da bi se ostvarila komunikacija, prvo se moraju se razmijeniti upravljačke informacije o pozivu, odnosno sjednici, što obavlja protokol SIP.

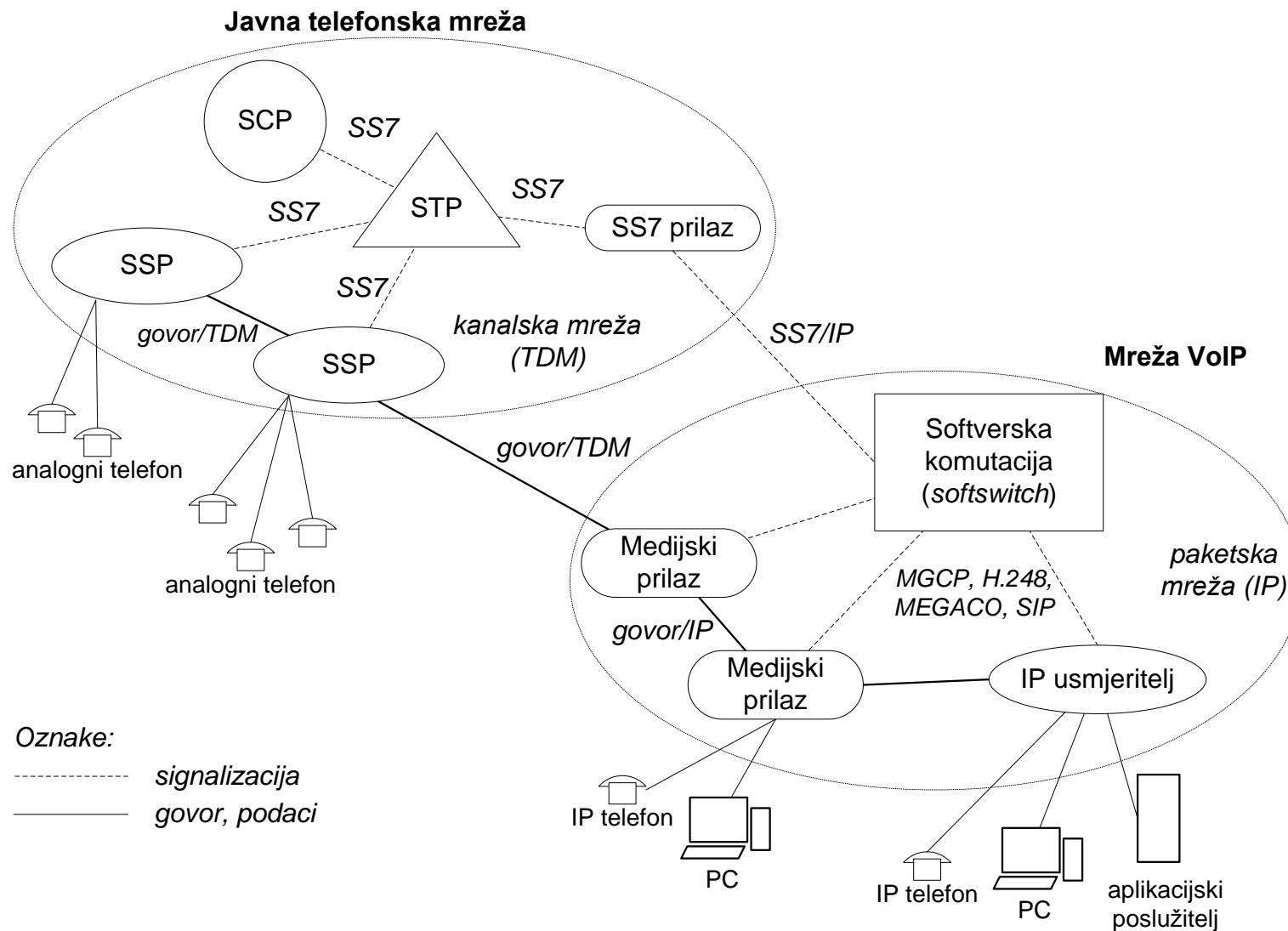
### Session Initiation Protocol (SIP)

- signalizacijski internetski protokol
- uspostavlja i raskida poziv (sjednicu)
- upravlja uslugama vezanima uz poziv (sjednicu)
- omogućuje razmijenu informacija o adresama, medijima, kodecima, i dr.

SIP kao format za opis sjednice koristi **Session Description Protocol (SDP)**.

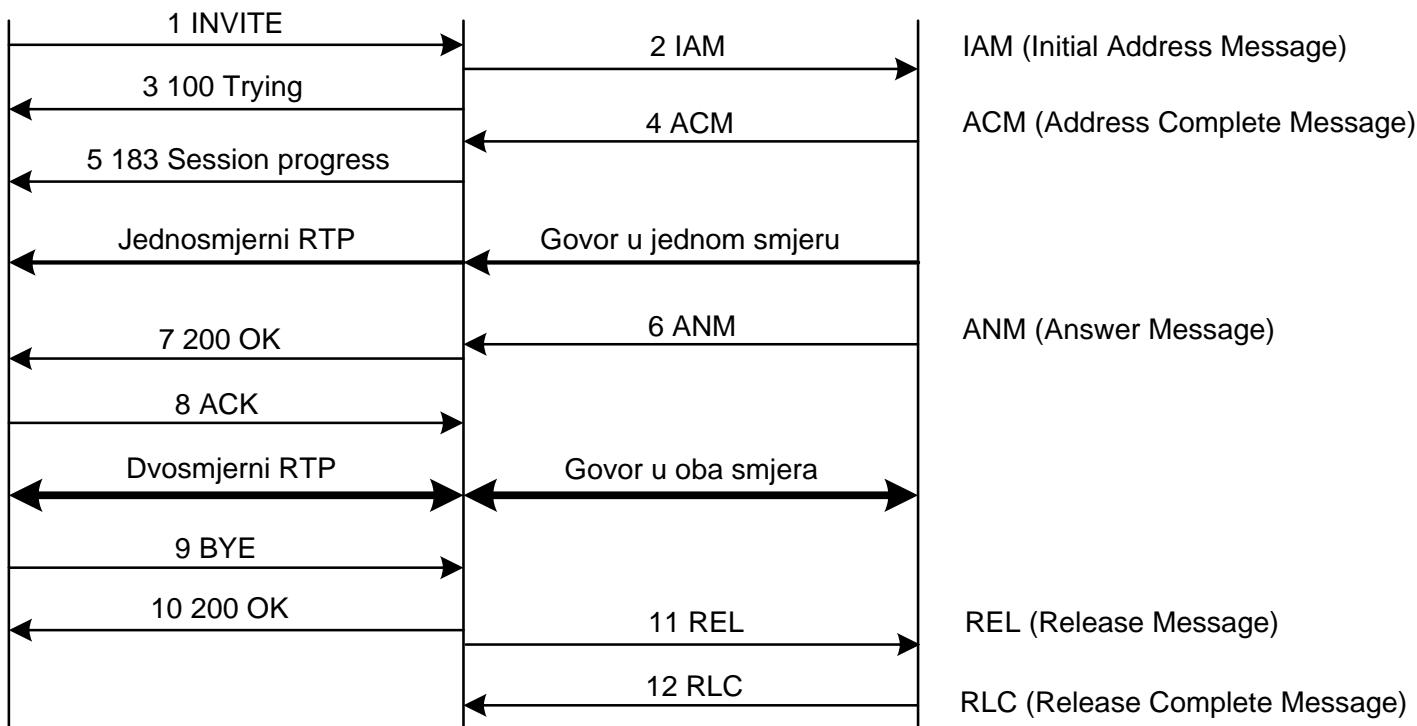
Neke aplikacije se temelje na SIP-u, dok neke koriste i posebne vlastite zaštićene protokole

# Povezivanje javne telefonske mreže i VoIP mreže



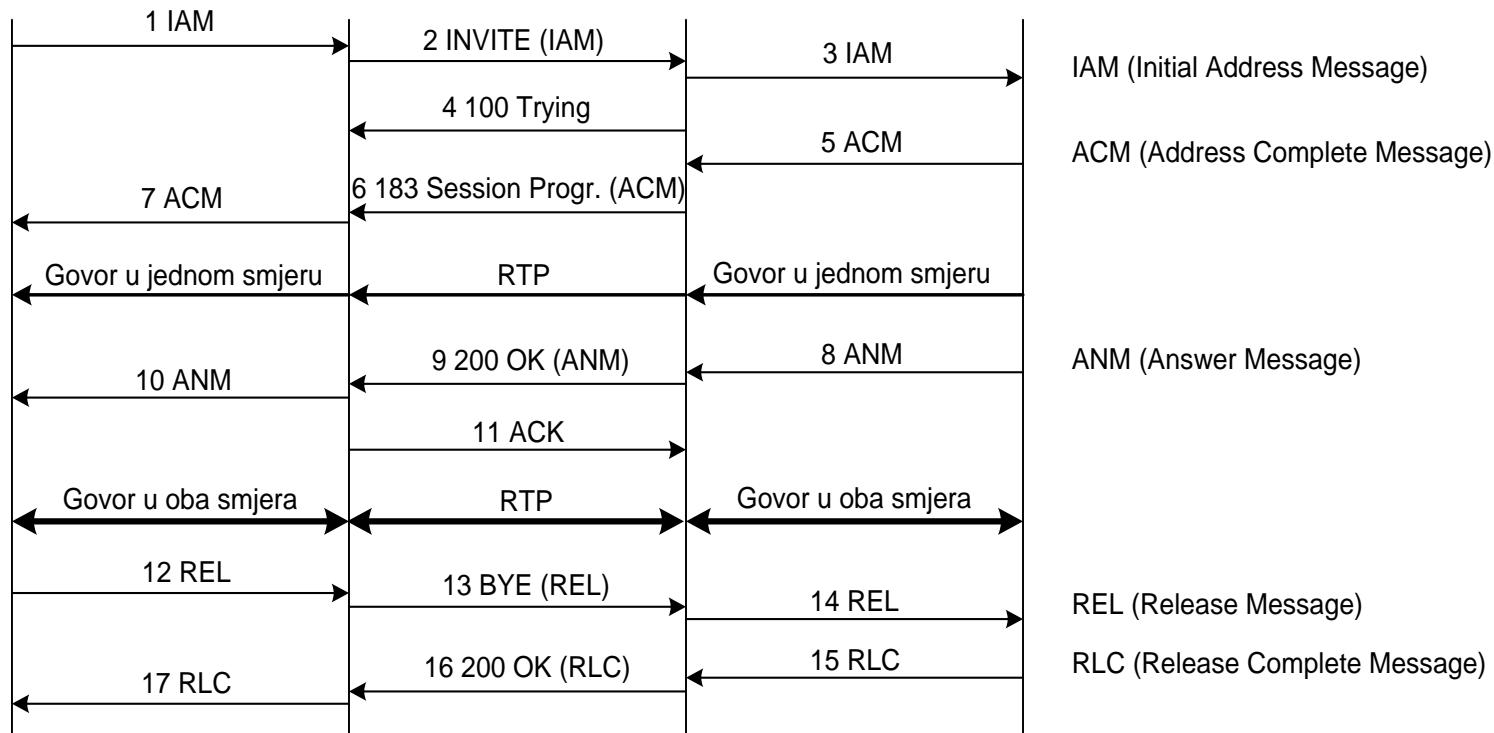
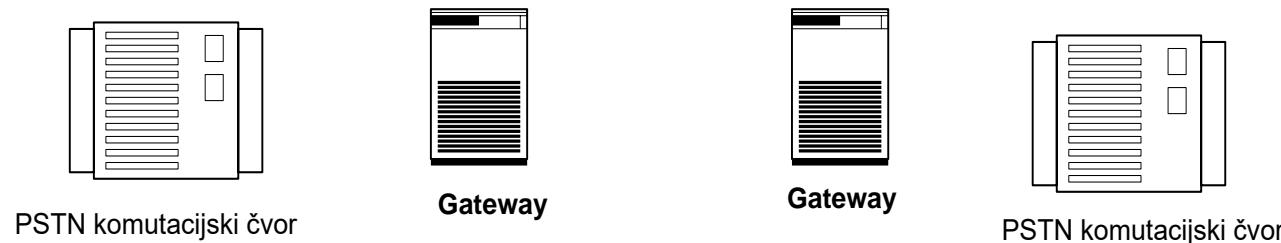
# Primjer međudjelovanja klasične i IP telefonije (1/2)

## ◆ Poziv SIP - PSTN



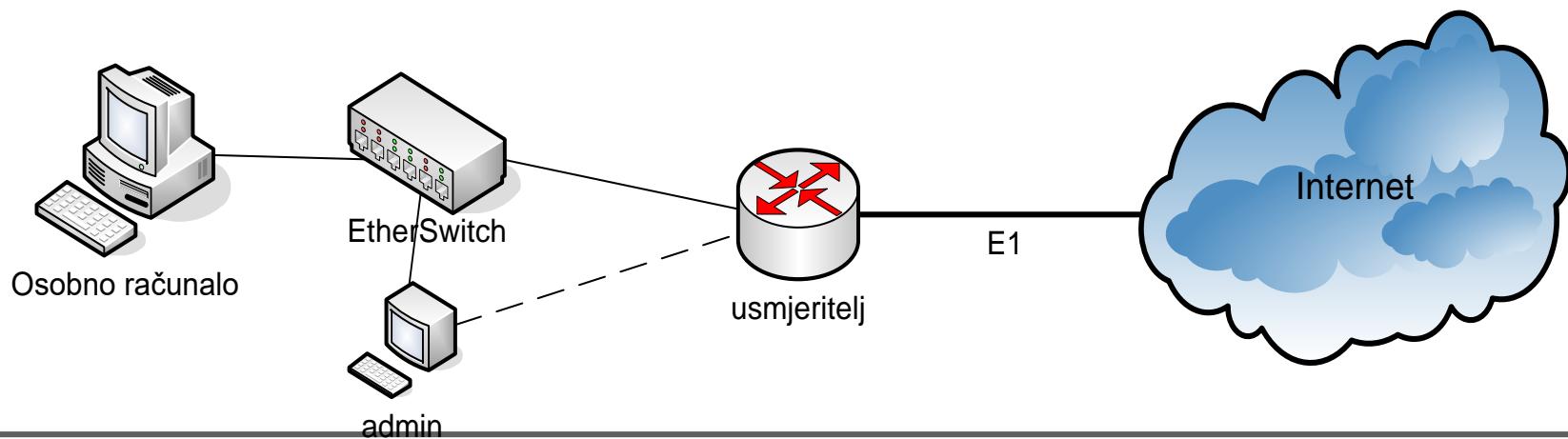
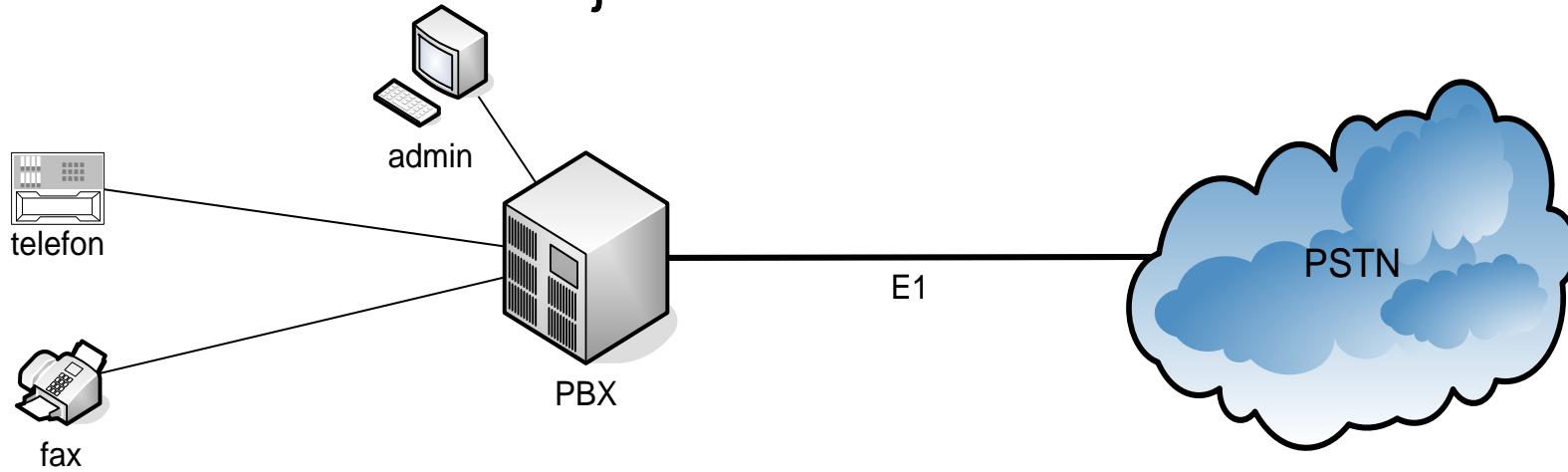
# Primjer međudjelovanja klasične i IP telefonije (2/2)

## ◆ Poziv PSTN – SIP - PSTN



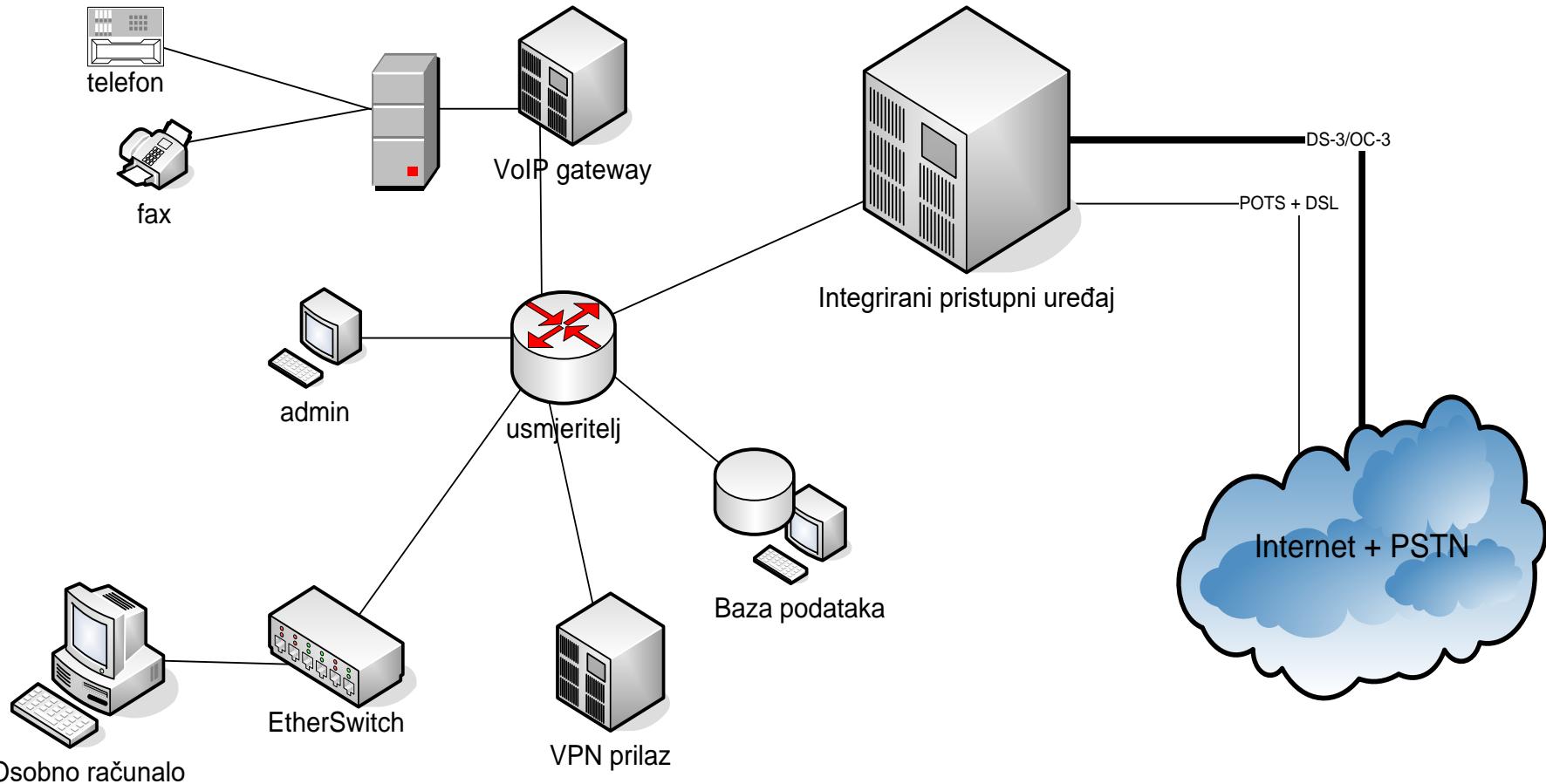
# Integracija govorne i podatkovne komunikacije (1/2)

- Primjer poduzeća s odvojenim mrežama za govornu i podatkovnu komunikaciju



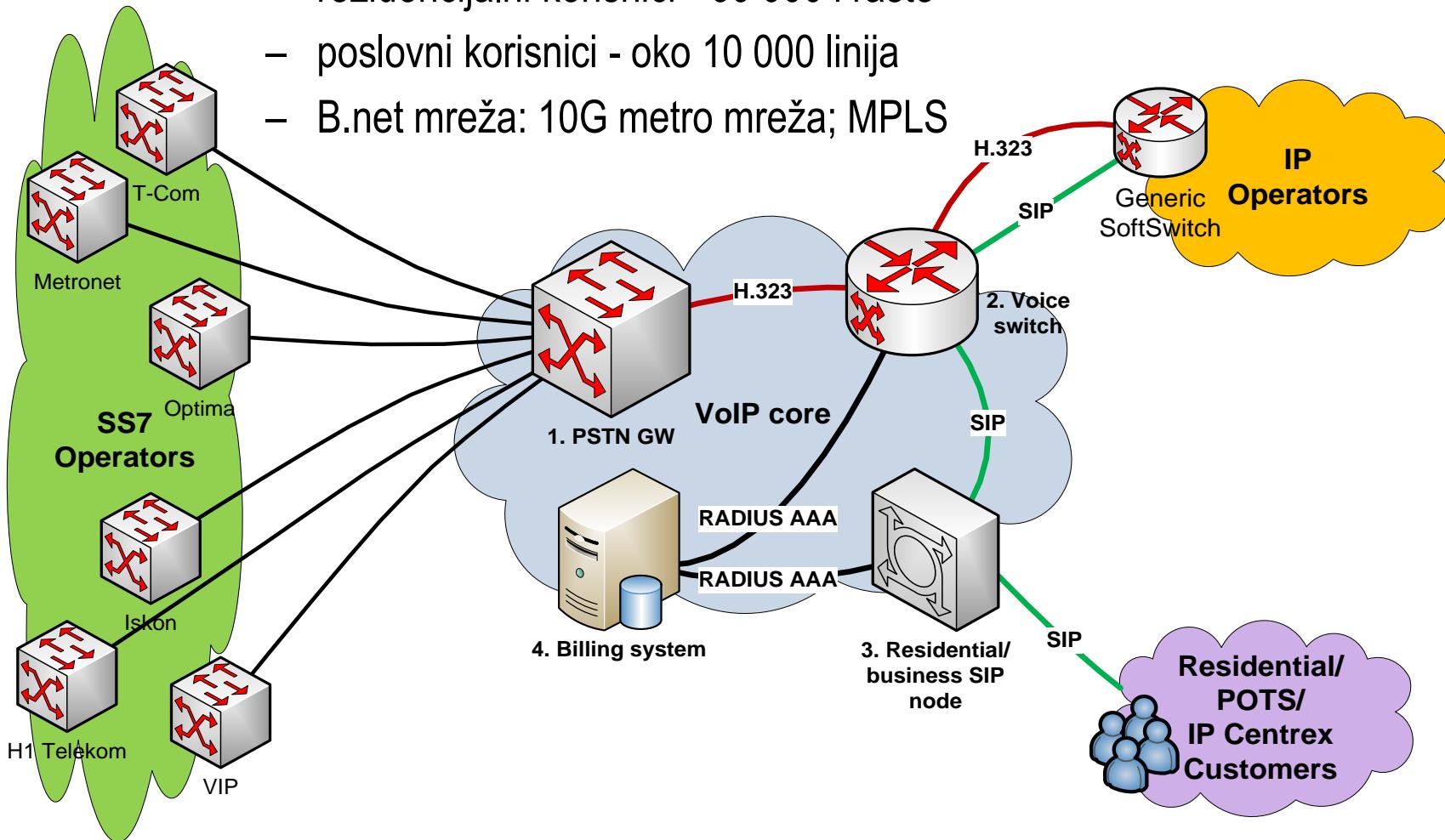
# Integracija govorne i podatkovne komunikacije (2/2)

- Integracija govorne i podatkovne komunikacije u mreži poduzeća

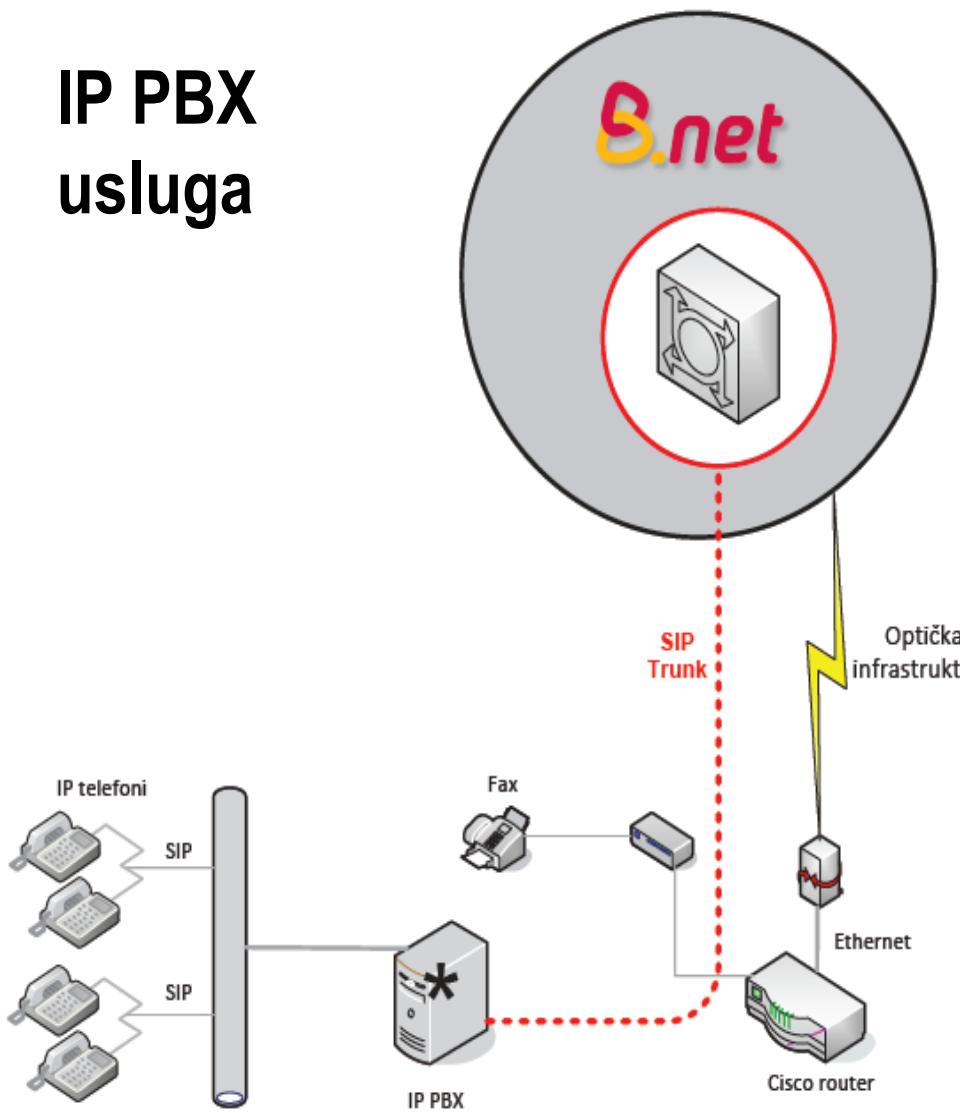


# Studijski slučaj u Hrvatskoj: B.net (1/2)

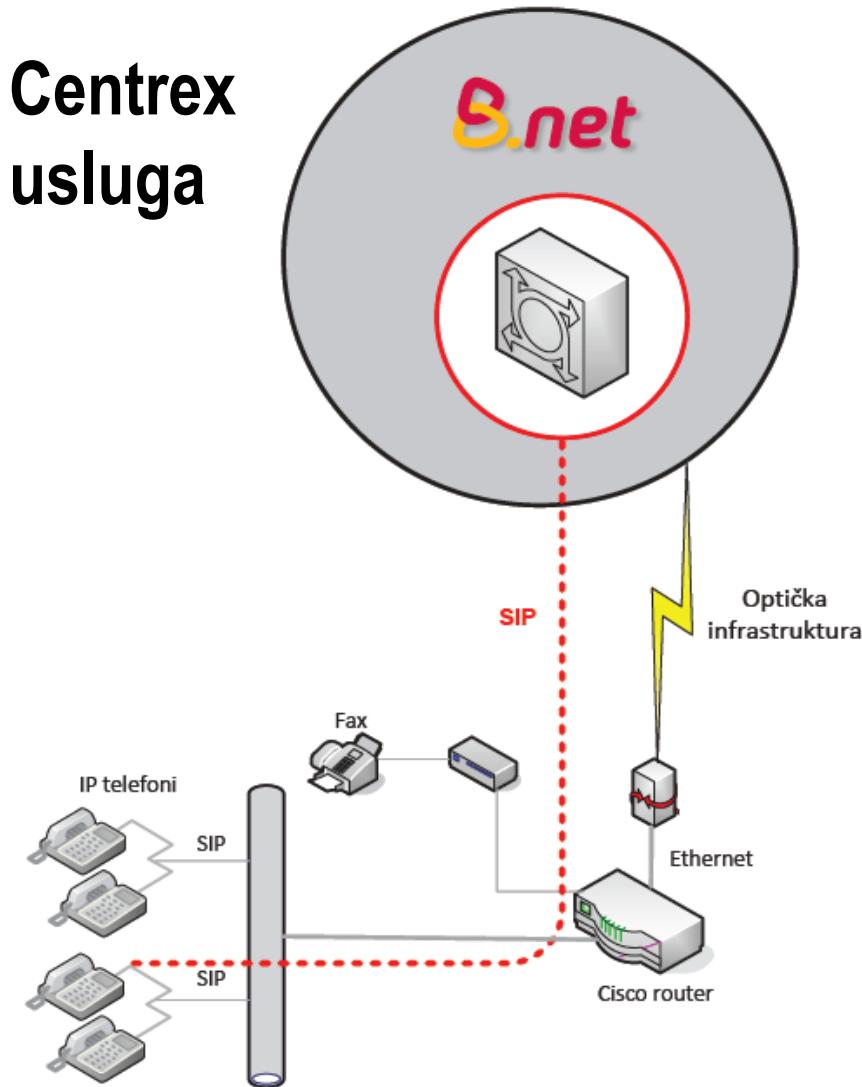
- Brojke (2011):
  - rezidencijalni korisnici - 60 000 i raste
  - poslovni korisnici - oko 10 000 linija
  - B.net mreža: 10G metro mreža; MPLS



## IP PBX usluga



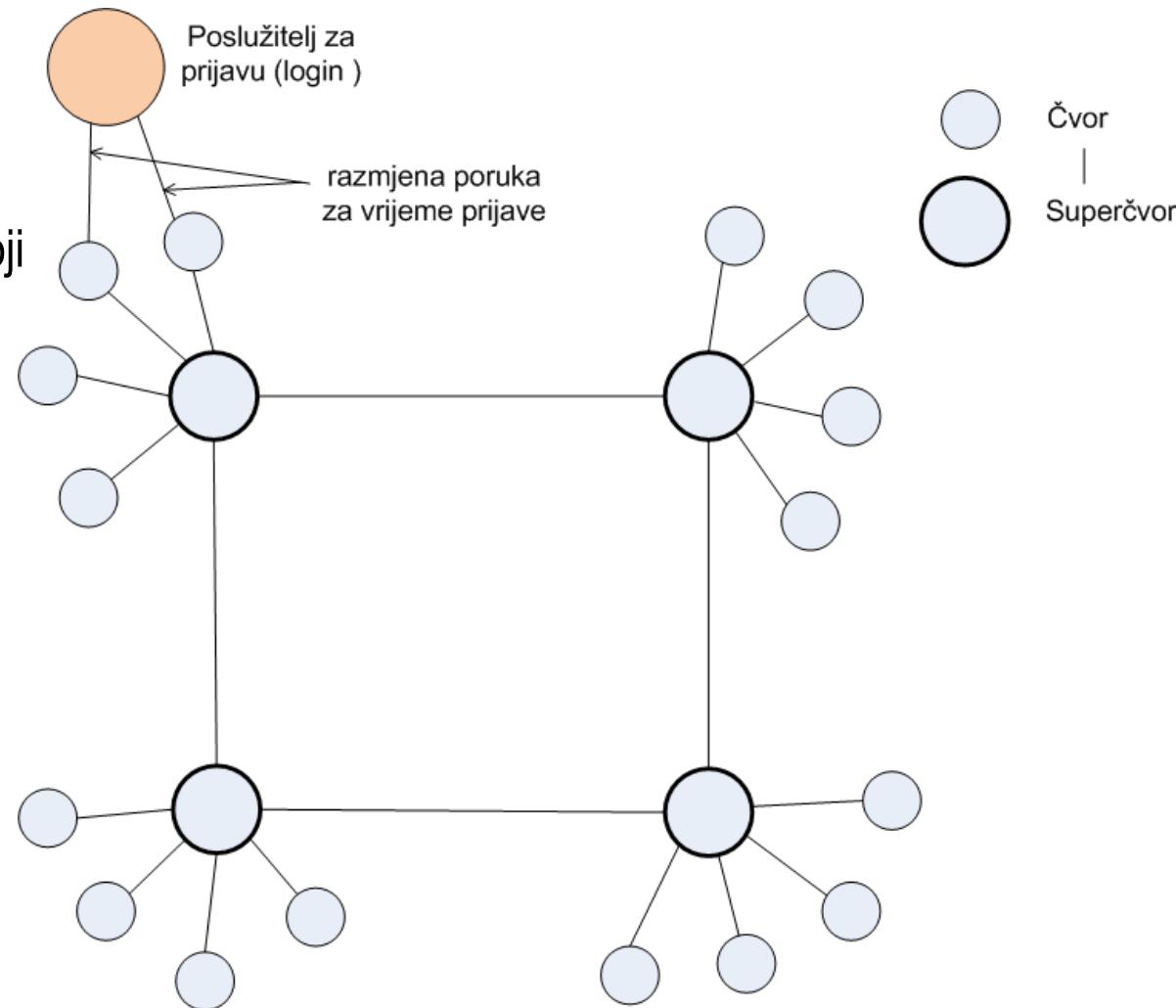
## Centrex usluga



izvor: [http://www.bnet.hr/cro/poslovni\\_korisnici/govorna\\_usluga](http://www.bnet.hr/cro/poslovni_korisnici/govorna_usluga)

# Primjer pružatelja VoIP usluga: Skype

- komunikacija zasnovana na mreži ravnopravnih entiteta (*peer-to-peer*)
- primjenjuje vlasnički protokol koji omogućuje govornu komunikaciju putem IP-a
- uz VoIP, nudi usluge videokonferencije, trenutno poručivanje, slanje datoteka
- pozive moguće ostvariti prema internetskim korisnicima, te na telefonske aparate u fiksnim i mobilnim mrežama



# Skype klijenti na raznim operacijskim sustavima



Zavod za telekomunikacije

Windows



Linux



Windows Mobile



iPhone



Blackberry



Android



- Pojmovi i terminologija
- **Arhitektura za VoIP zasnovana na protokolu SIP**
- Arhitektura za VoIP zasnovana na standardu H.323

# Protokoli za podršku sjednice - podsjetnik



Zavod za telekomunikacije

- Protokol za opis sjednice (engl. ***Session Description Protocol, SDP***)
  - obuhvaća propisani skup parametara koji služi za opis sjednice
  - standardni format za opis medija koji sudjeluju u sjednici, podataka o protokolima i formatima koji će se koristiti u sjednici i sl.
- Protokol za pokretanje sjednice (engl. ***Session Initiation Protocol, SIP***)
  - služi za razmjenu podataka o sjednici
  - služi kao poziv određenom korisniku za sudjelovanje u sjednici

# SIP: Session Initiation Protocol

- Definiran od strane standardizacijskog tijela IETF (dokument: RFC 3261)
- SIP je protokol aplikacijskog sloja koji služi za pokretanje, promjenu i raskid sjednice s jednim ili više sudionika
- Primjeri sjednica:
  - pozivi u internetskoj telefoniji
  - distribucija višemedijskog sadržaja
  - višemedijska konferencija
- osnovna ideja: omogućiti pozivanje osobe u sjednicu putem *jedinstvene adrese* (neovisno o trenutnom položaju)

[sip:]<user>@(<host>|<domain>)



Osobna  
pokretljivost!
- SIP koristi posredničke (*proxy*) poslužitelje za preusmjeravanje poziva prema trenutnom položaju pozivane osobe (o tome više kasnije)

# Osnovne SIP funkcionalnosti

- Određivanje lokacije krajnjeg korisnika
- Određivanje dostupnosti krajnjeg korisnika
- Određivanje parametara medija koji će se koristiti tijekom sjednice
- Uspostava sjednice
- Upravljanje sjednicom (promjene parametara, raskid)

# SIP usluge općenito

- **SIP usluge**
  - Uspostava VoIP poziva
  - Uspostava višemedijskih konferencija
  - Obavijesti o događajima – usluga prisutnosti
  - Tekstualne poruke i trenutno poručivanje
- Usluge koje SIP pruža mogu se izvesti u sljedećim entitetima komunikacijskog puta:
  - poslužitelji - usluga se kreira i pruža korisniku u poslužiteljima na komunikacijskom putu
  - pozvani UA - usluga se kreira i pruža korisniku u pozvanom agentu
  - pozivajući UA - usluga se kreira i pruža korisniku u pozivajućem agentu

- Koristi URL (*Uniform Resource Locators*), te podržava podržava internetske i PSTN adrese
- Primjeri:
  - sip:pero.peric@tel.fer.hr*
  - sip:Pero <pero@tel.fer.hr>*
  - sip:+1-385-1-6129-123@tel.fer.hr;user=phone*
  - sip:pero@136.16.20.100:8001*
  - sip:790-7360@pulver.com;phone-context=VNET*
- Za siguran prijenos se može koristiti SIPS URI (koristi šifrirani transport: TLS preko TCP-a), npr: *sips:pero.peric@tel.fer.hr*

# Format SIP poruke (1/2)

- format SIP poruke je tekstualni format (ISO10646 UTF-8)
- SIP poruke su slične porukama protokola HTTP i RTSP, osnovna podjela na **zahtjeve** (metode) i **odgovore** (statusni kod)
- zahtjevi i odgovori koriste generički oblik poruke:
  - početni redak, sadrži zahtjev ili statusni kod odgovora
  - jedno ili više zaglavljia
  - prazni redak za odvajanje zaglavljia poruke i opcionalnog tijela poruke
  - opcionalni dio poruke – npr. SDP opis sjednice
- za pozivanje korisnika u sjednicu, SIP koristi SDP za opis sjednice (SDP opis umeće se u SIP poruku)
- SIP je neovisan o transportnom protokolu i o vrsti sjednice

# Format SIP poruke (2/2)

## Zahtjev

*method URL SIP/2.0*

## Odziv

SIP/2.0 status reason

**Via:** SIP/2.0/ protocol host:port  
**From:** user <sip:from\_user@source>  
**To:** user <sip:to\_user@destination>  
**Call-ID:** localid@host  
**CSeq:** seq#method  
**Content-Length:** *length of body*  
**Content-Type:** *media type of body*  
**Header:** parameter ;par1=value;par2="value"

prazni redak

**V=0**

**o=** orgin\_user timestamp timestamp **IN IP4 host**

**c=IN IP4 media destination address**

**t=0 0**

**m=** media type port RTP/AVP payload types

zaglavlje poruke

tijelo poruke

# SIP zahtjevi i odgovori

## ZAHTJEVI (METODE)

- INVITE
  - Poziv na sjednicu
- ACK
  - Potvrda, uspješan odgovor
- CANCEL
  - Opoziv zahtjeva
- BYE
  - Završetak poziva ili zahtjeva
- OPTIONS
  - Provjera mogućnosti primatelja
- REGISTER
  - Prijava trenutnog položaja korisnika

## ODGOVORI (STATUSNI KODOVI)

- 1xx: info o statusu poziva
  - npr. 180 "Ringing", 181 "Call is Being Forwarded"
- 2xx: uspješni ishod
  - npr. 200 "OK"
- 3xx: preusmjeravanje
  - npr. 301 "Moved Permanently", 302 "Moved Temporarily"
- 4xx: pogreška klijenta
  - npr 404 "Not Found", 420 "Bad Extension", 486 "Busy Here"
- 5xx: pogreška poslužitelja
  - npr. 500 "Internal Server Error", 504 "Server Time Out"
- 6xx: globalna pogreška
  - npr. 603 "Decline", 604 "Does Not Exist Anywhere"

# Primjer: SIP zahtjev i SIP odgovor

**INVITE** sip:ana@example.se **SIP/2.0**

**Via:** SIP/2.0/UDP science.fiction.com  
;branch=z9hG4bKnashd

**Max-Forwards:** 70

**To:** Ana <sip:ana@example.se>

**From:** Pero <sip:pp@fiction.com>; tag=123455

**Call-ID:** 1234567890@science.fiction.com

**CSeq:** 1 INVITE

**Subject:** Rucak u gradu

**Content-Type:** application/sdp

v=0

o=

s=

c=IN IP4 128.2.3.1

t=

m=audio 5004 RTP/AVP 0 4

a=rtpmap:0 PCMU/8000

a=rtpmap:4 GSM/8000

**SDP**

**SIP/2.0 200 OK**

**Via:** SIP/2.0/UDP sippo.example.se  
;branch=z9hG4bKkljdrf

**From:** SIP/2.0/UDP science.fiction.com  
;branch=z9hG4bKnashd

**To:** Pero <sip:pp@fiction.com> ; tag=123455

**Call-ID:** 1234567890@science.fiction.com

**CSeq:** 1 INVITE

**Subject:** Rucak u gradu

**Content-Type:** application/sdp

v=0

o=

s=

c=IN IP4 16.2.3.1

t=

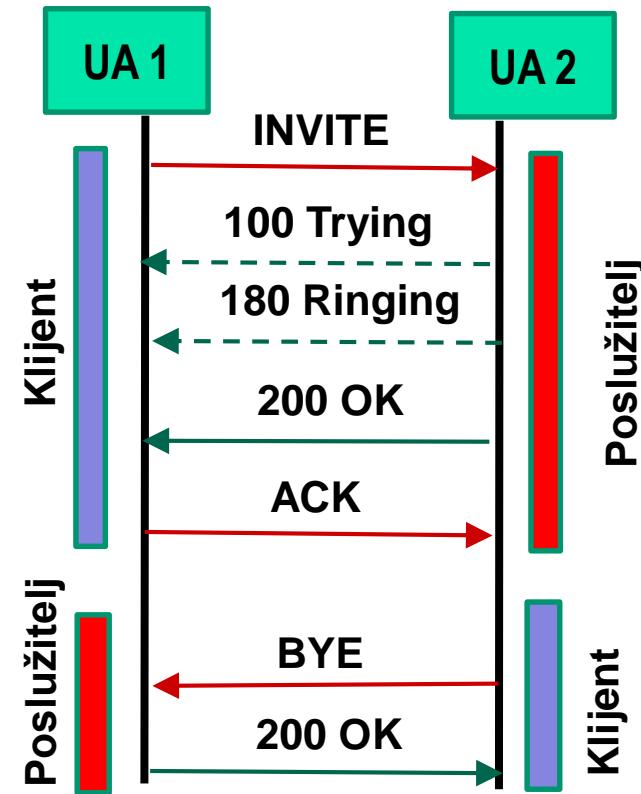
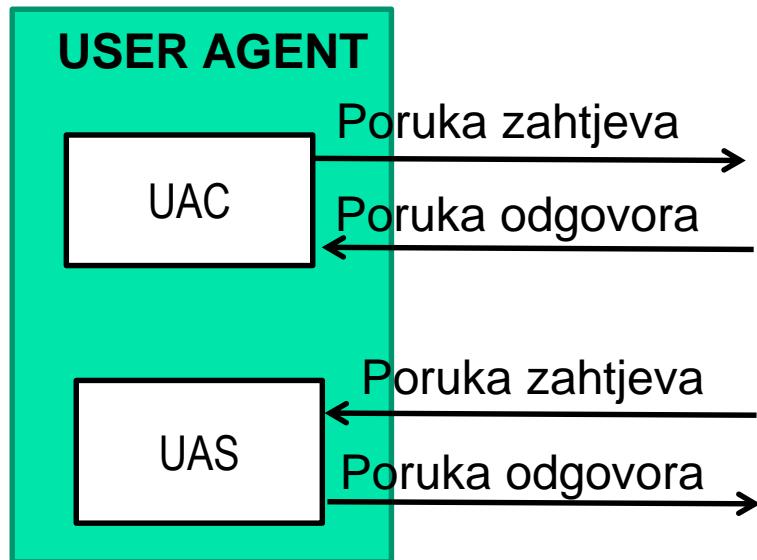
m=audio 6004 RTP/AVP 0

a=rtpmap:0 PCMU/8000

**SDP**

# SIP mrežni entiteti (1/2)

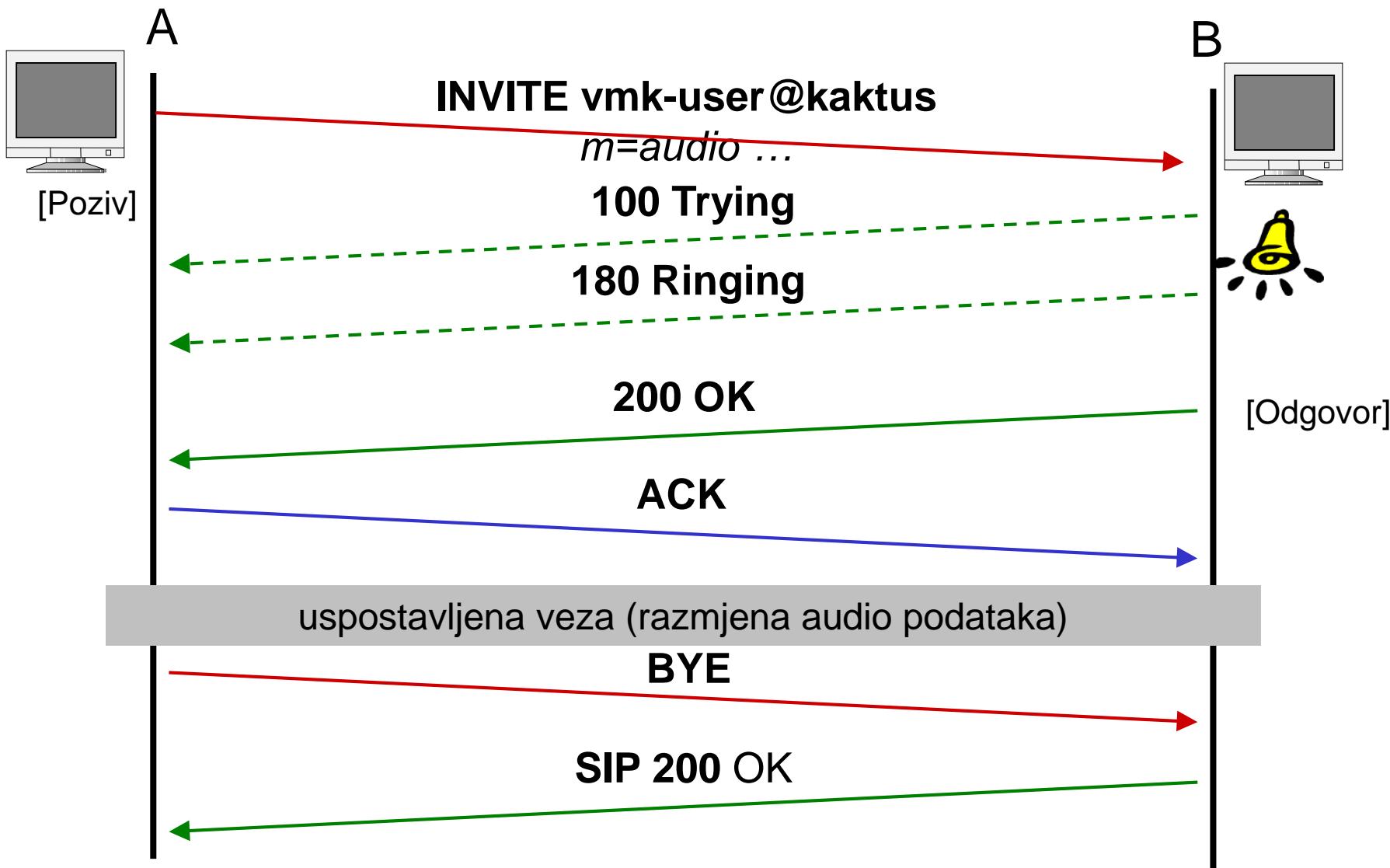
- **SIP klijent**, odnosno **korisnički agent** (engl. *User Agent*, UA)
  - krajnja točka koja koristi SIP za uspostavu i raskid sjednica
  - nalazi se na korisničkim uređajima uglavnom u obliku aplikacija
  - dijeli se na klijentski UA (engl. UA Client, UAC) i poslužiteljski UA (engl. UA Server, UAS)



# SIP mrežni entiteti (2/2)

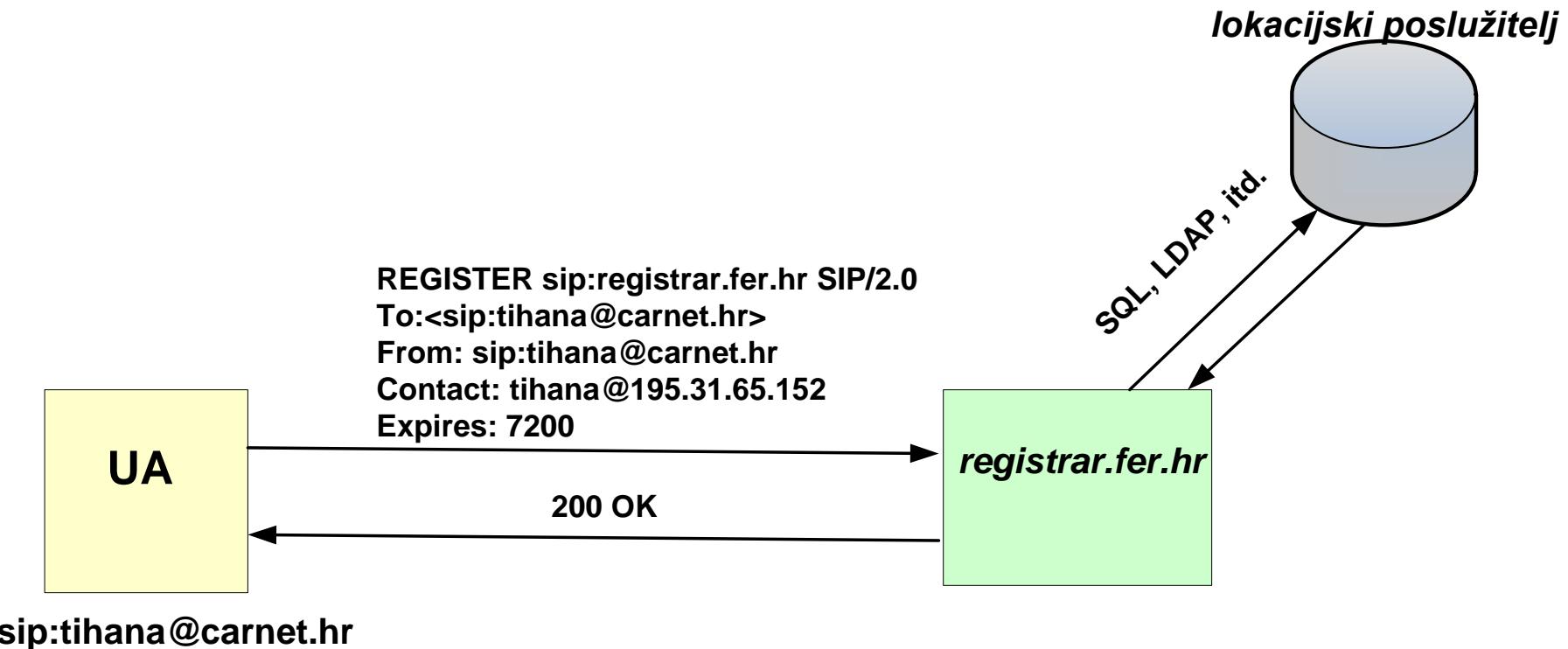
- **SIP poslužitelji:**
  - **Posrednički poslužitelj** (engl. *Proxy server*) - usmjerava zahtjeve (i odgovore) do trenutnog položaja korisnika (korisničkih agenata) koristeći podatke iz Registra
  - **Poslužitelj preusmjeravanja** (engl. *Redirect server*) - prima odgovarajuće zahtjeve; odgovara s popisom svih mogućih adresa korisnika (na temelju podataka iz Registra ili Lokacijskog poslužitelja)
  - **Registar** (engl. *Registrar*) - entitet kojem korisnički agenti prijavljuju trenutni položaj (trenutnu IP adresu) s ciljem ispravnog usmjeravanja zahtjeva
  - **Lokacijski poslužitelj** (engl. *Location Server*): čuva podatke o trenutnoj lokaciji korisničkog agenta

# Primjer izravnog poziva

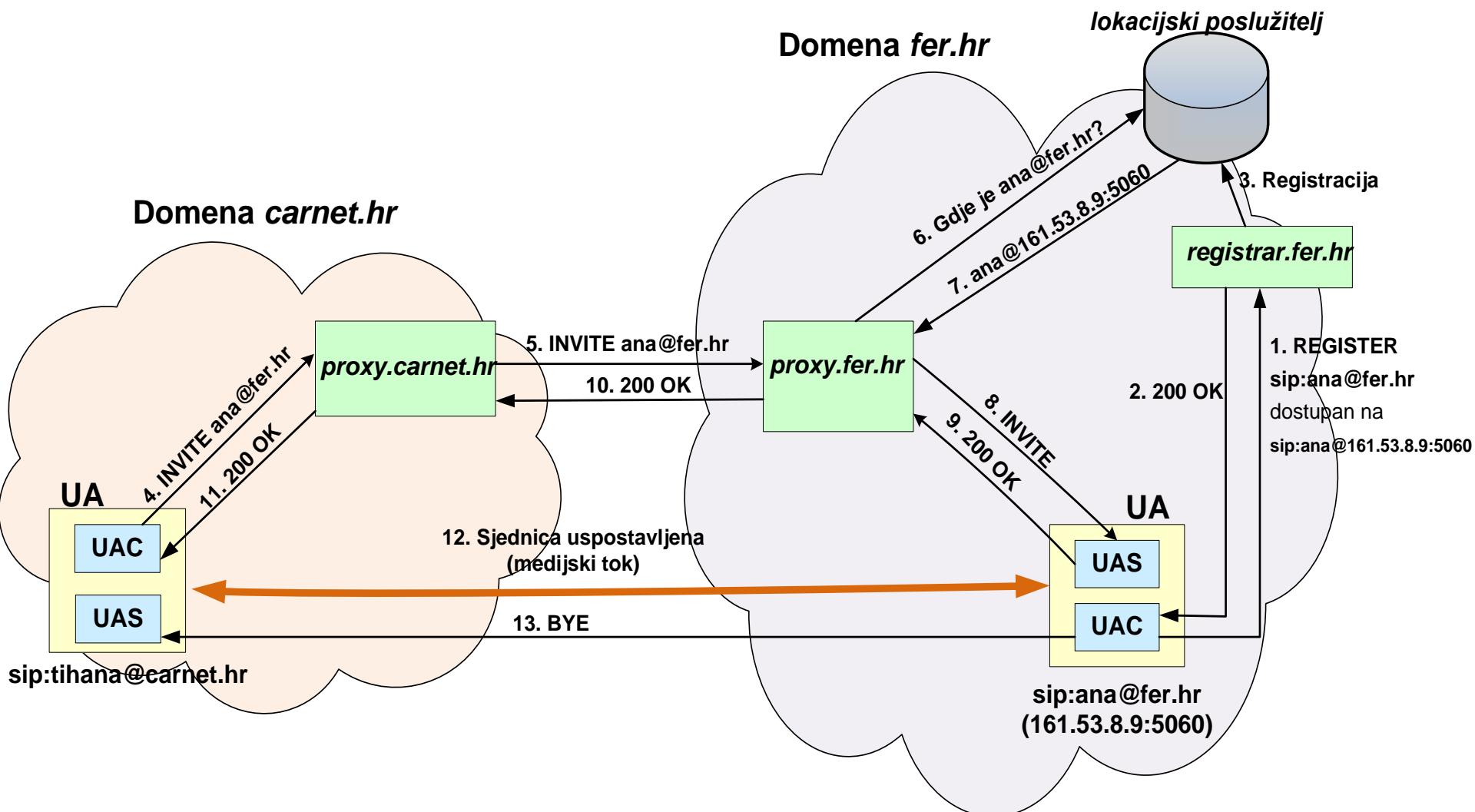


# Registracija

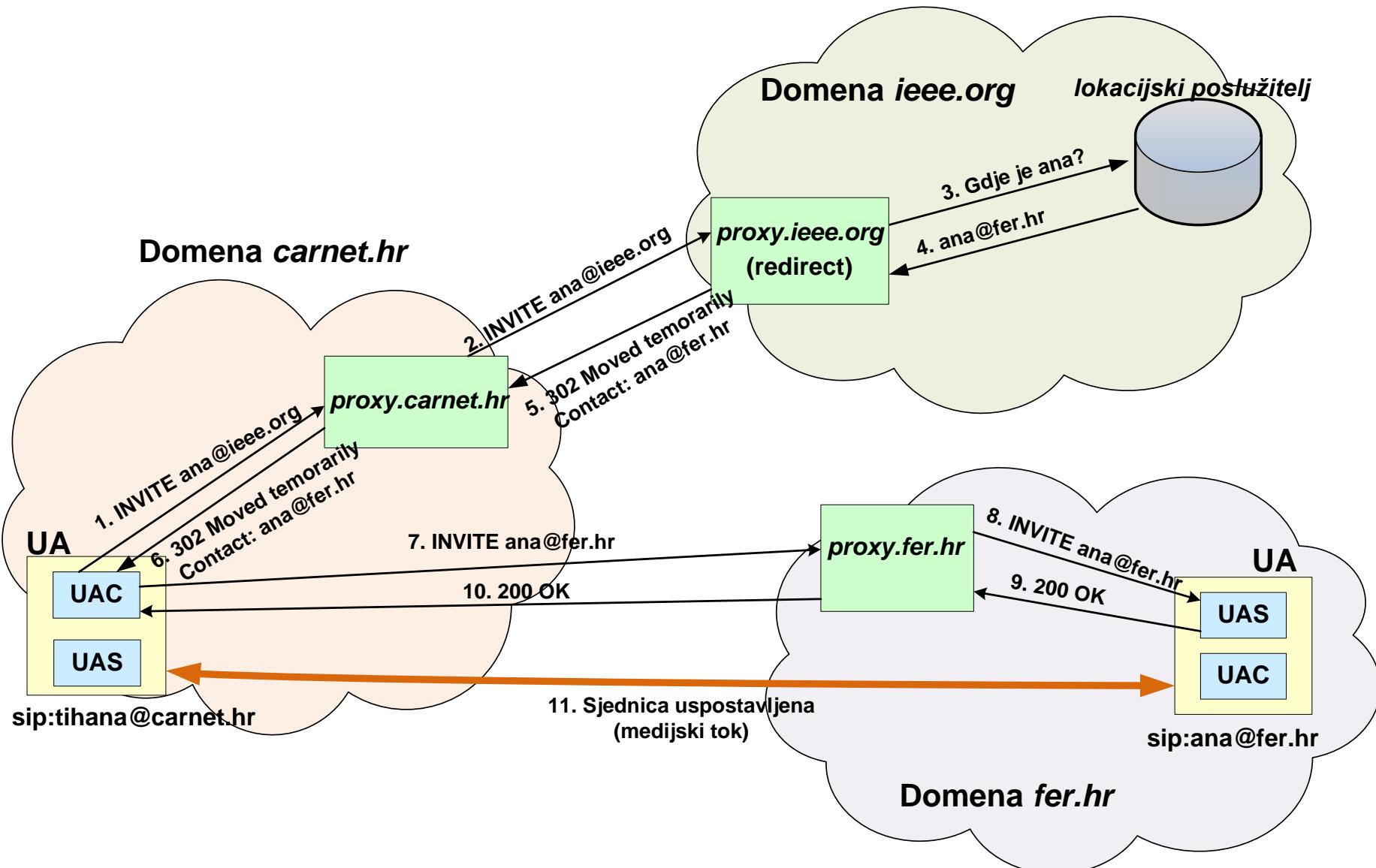
- Korisnički agent obavještava mrežu o svojoj lokaciji
- Zahtjev se šalje i proslijeđuje dok ne dođe do nadležnog poslužitelja za registraciju u domeni



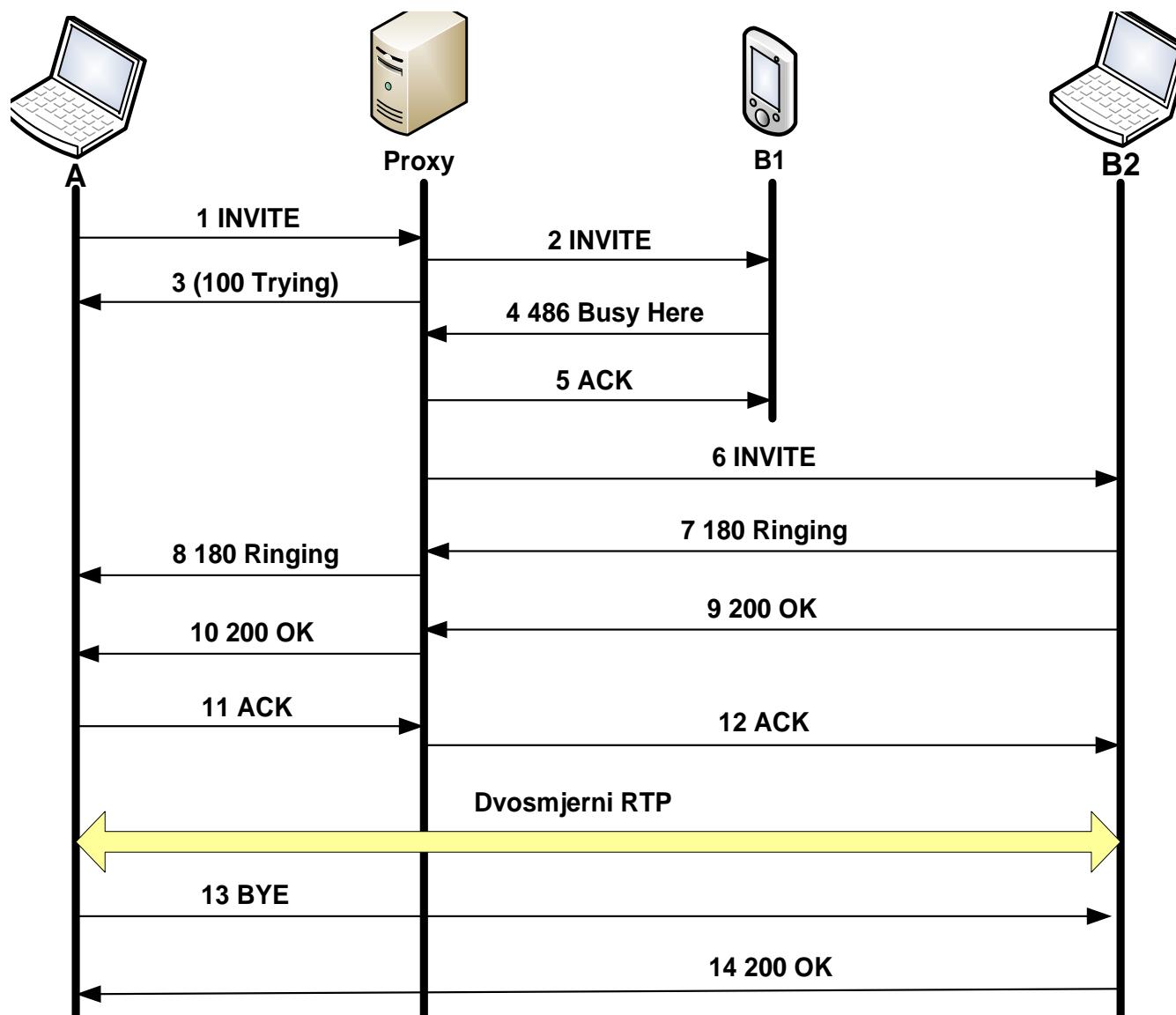
# Primjer poziva preko posredničkih poslužitelja

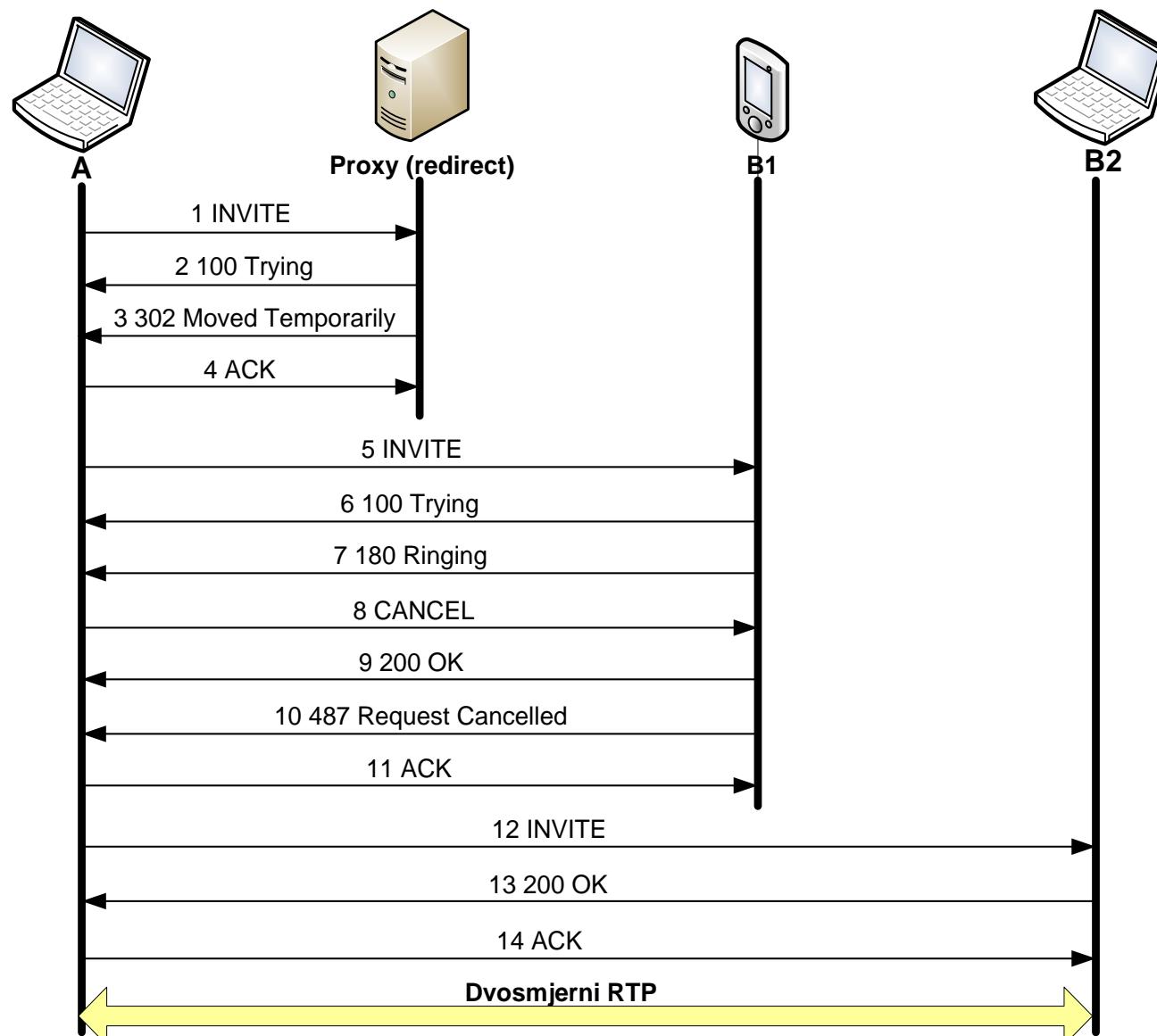


# Primjer preusmjeravanja poziva

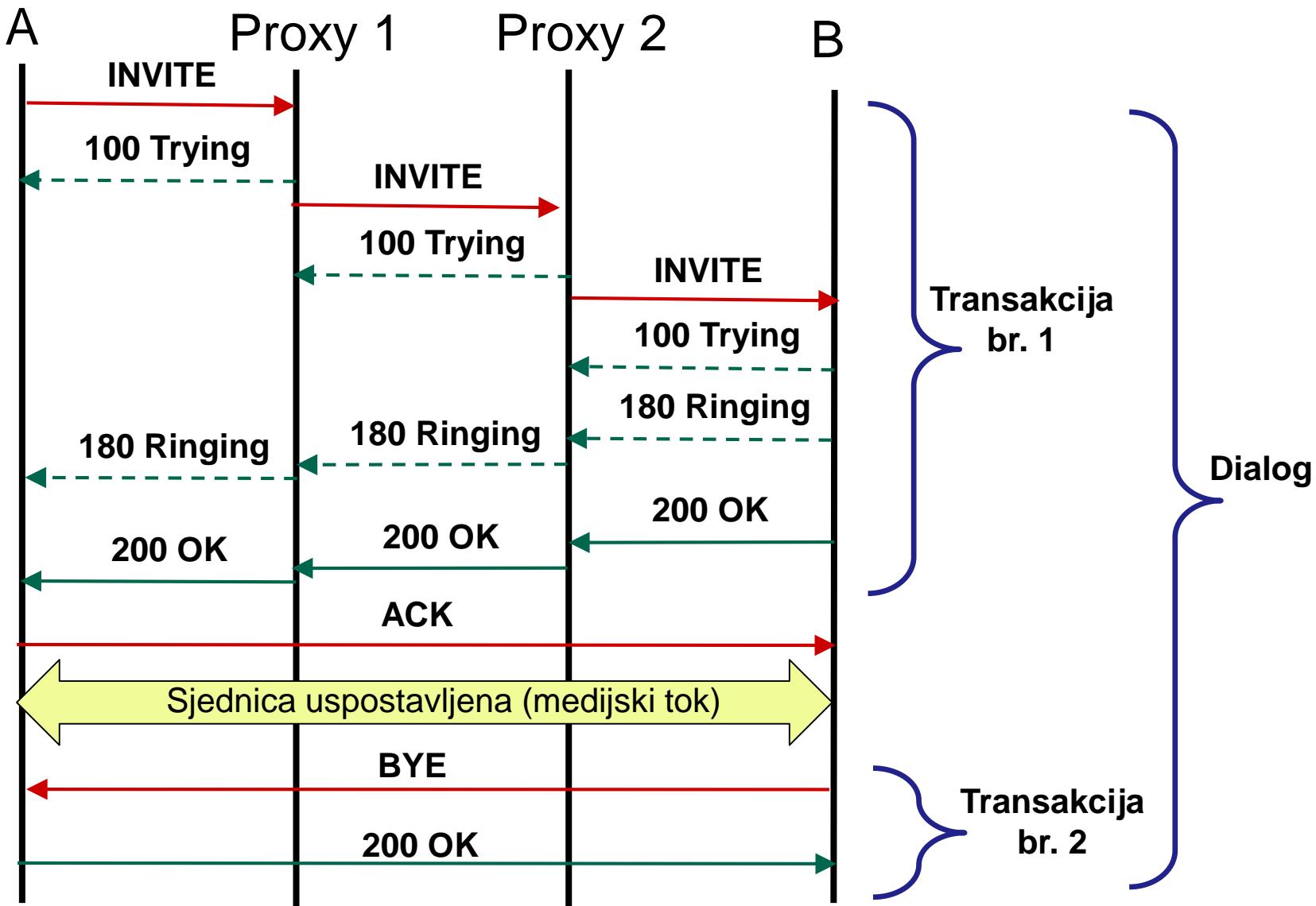


# Preusmjeravanje kod zauzeća





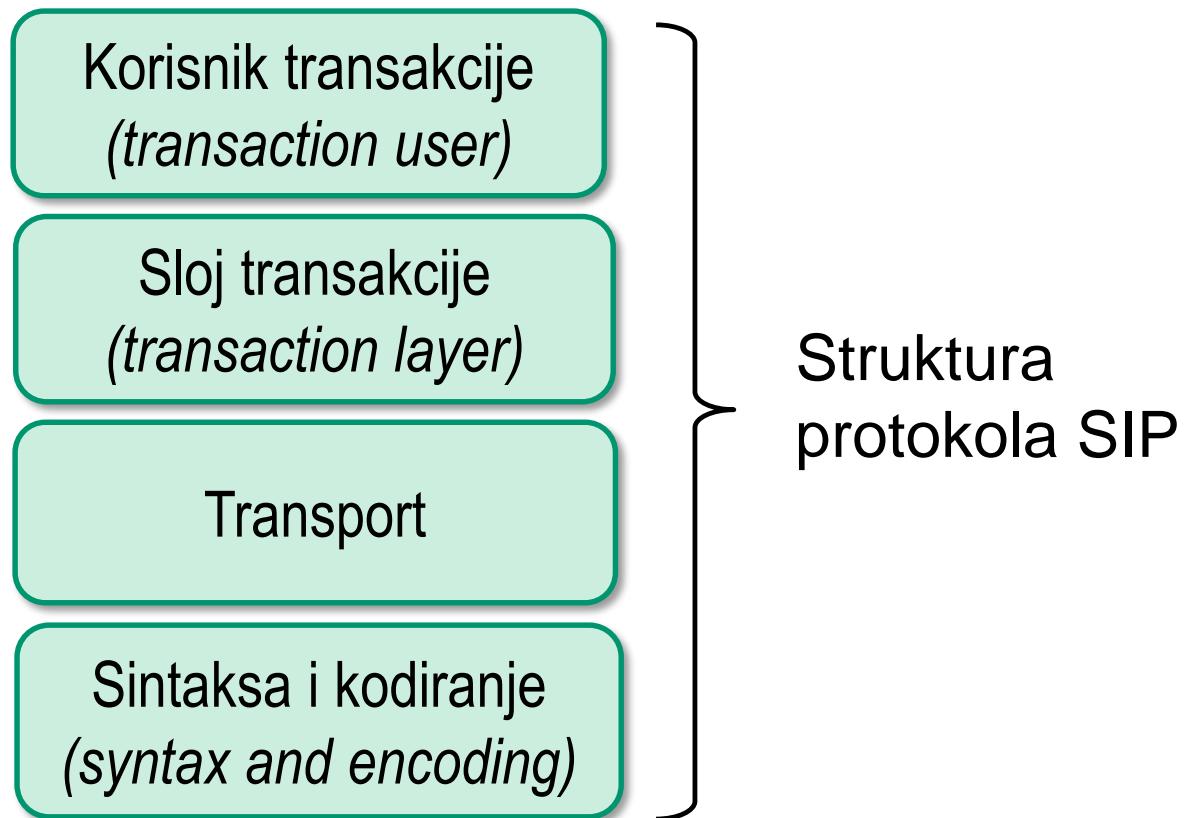
# SIP transakcije i dialozi (1/2)



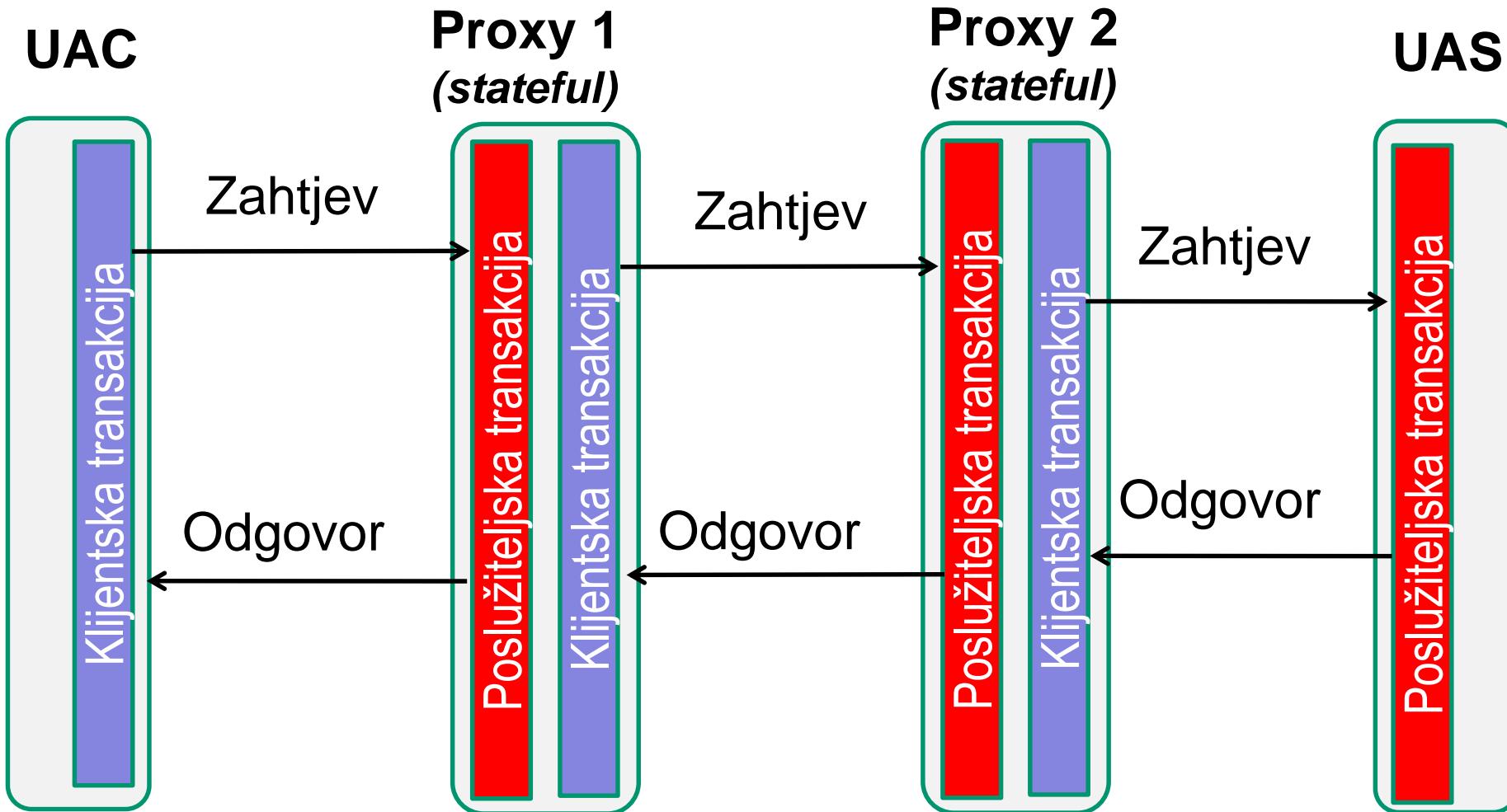
# SIP transakcije i dialozi (2/2)

- Posrednički poslužitelji mogu čuvati stanje (*stateful*) ili biti bez stanja (*stateless*)
  - ***Dialog stateful***: čuva stanje dialoga od inicijalnog zahtjeva (INVITE) do terminirajućeg zahtjeva (BYE)
  - ***Transaction stateful***: za vrijeme trajanja transakcije čuva stanje transakcije
  - ***Stateless***: nema stanja transakcije prilikom prosljeđivanja zahtjeva i odgovora
- U slučaju grananja poziva (*forking*) – stvara se više dialoga

# Struktura protokola SIP

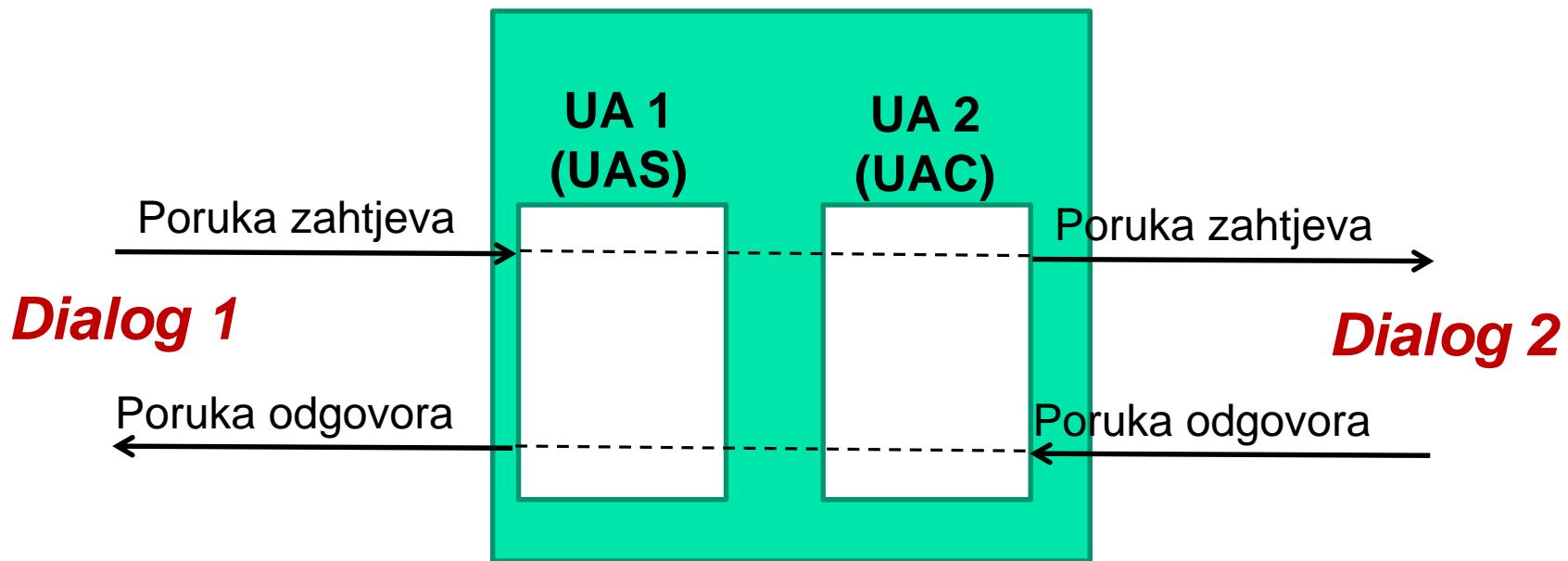


# Klijentske i poslužiteljske transakcije

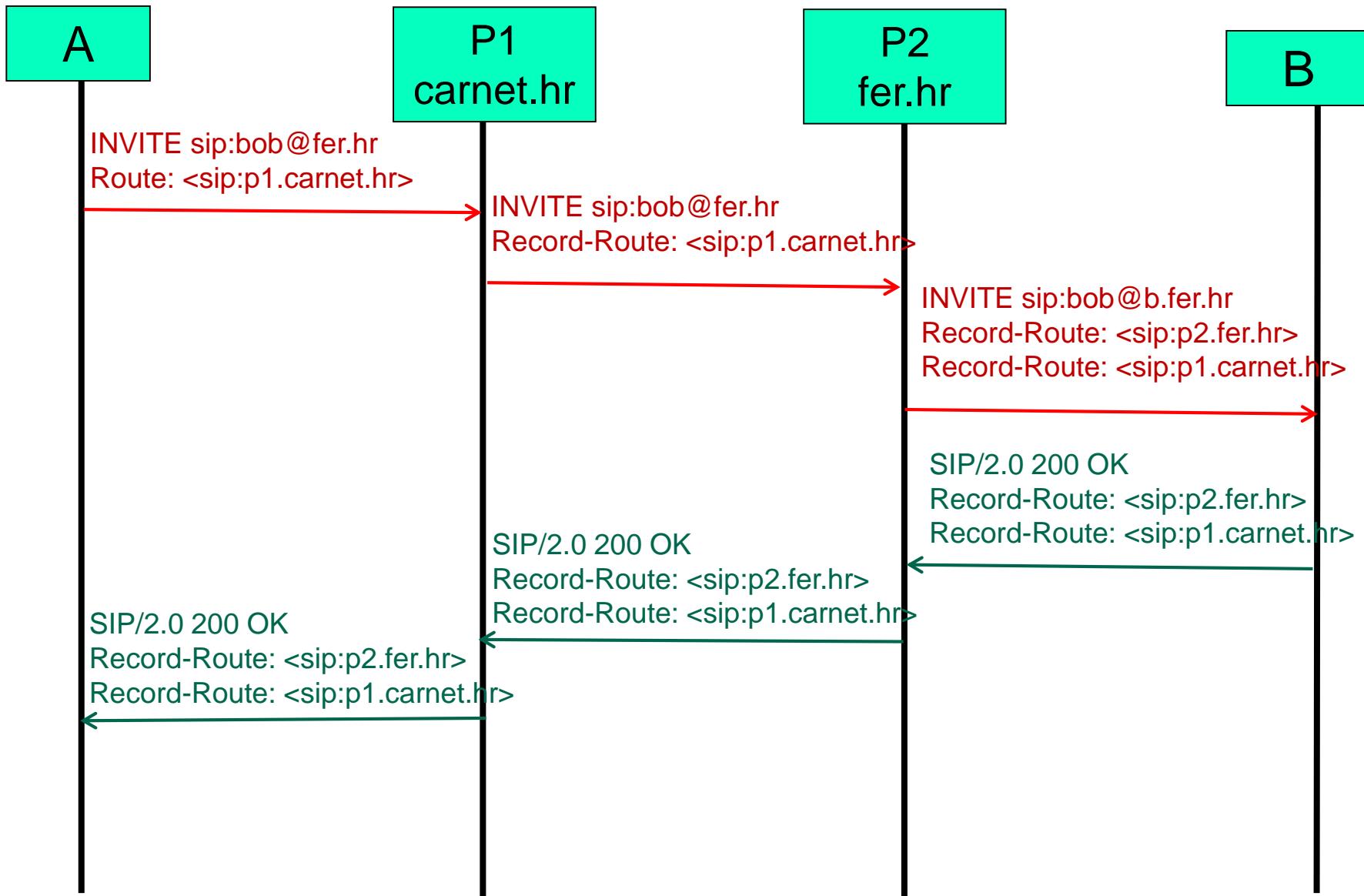


# Back to Back User Agent (B2BUA)

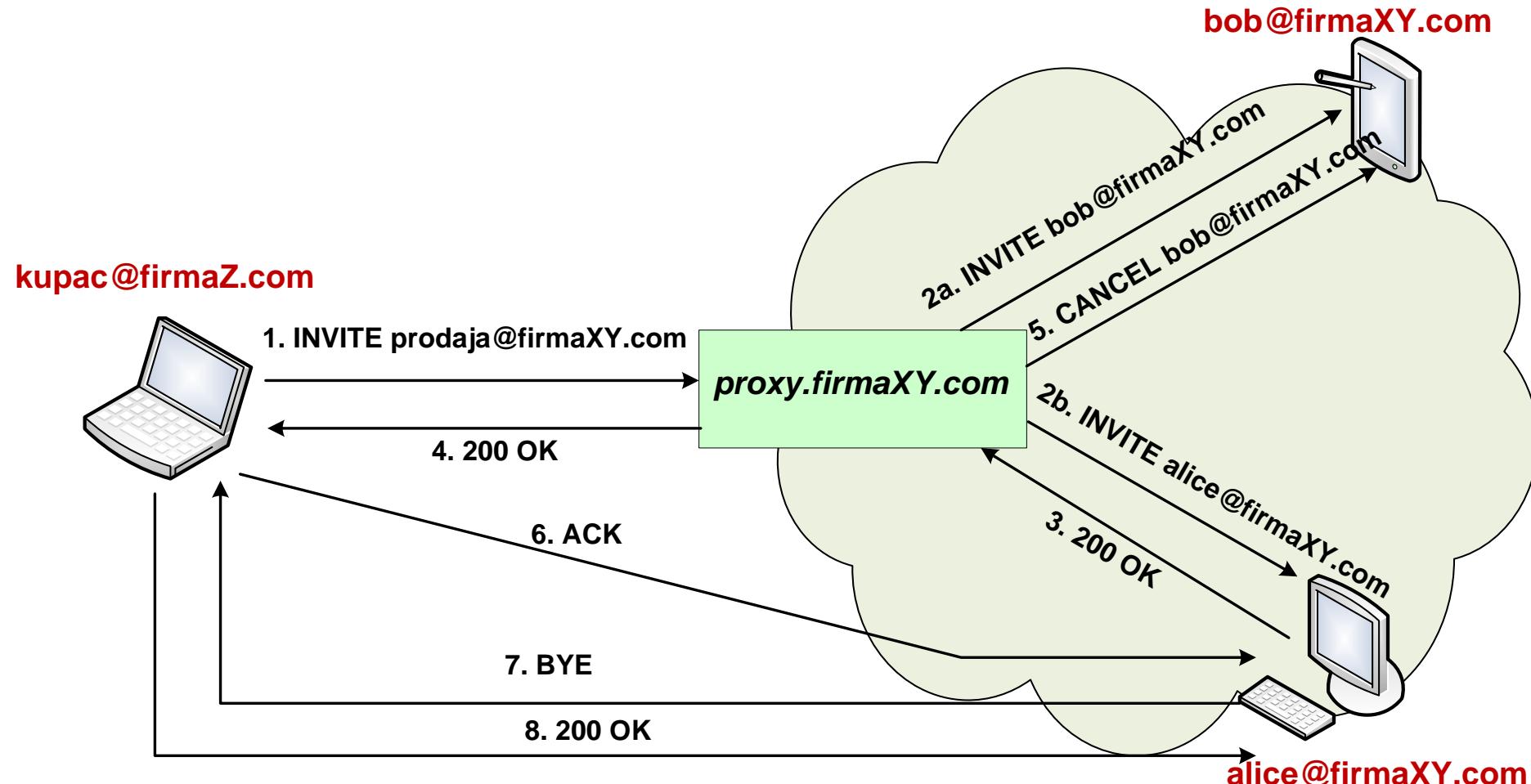
## Posrednički poslužitelj (B2BUA)



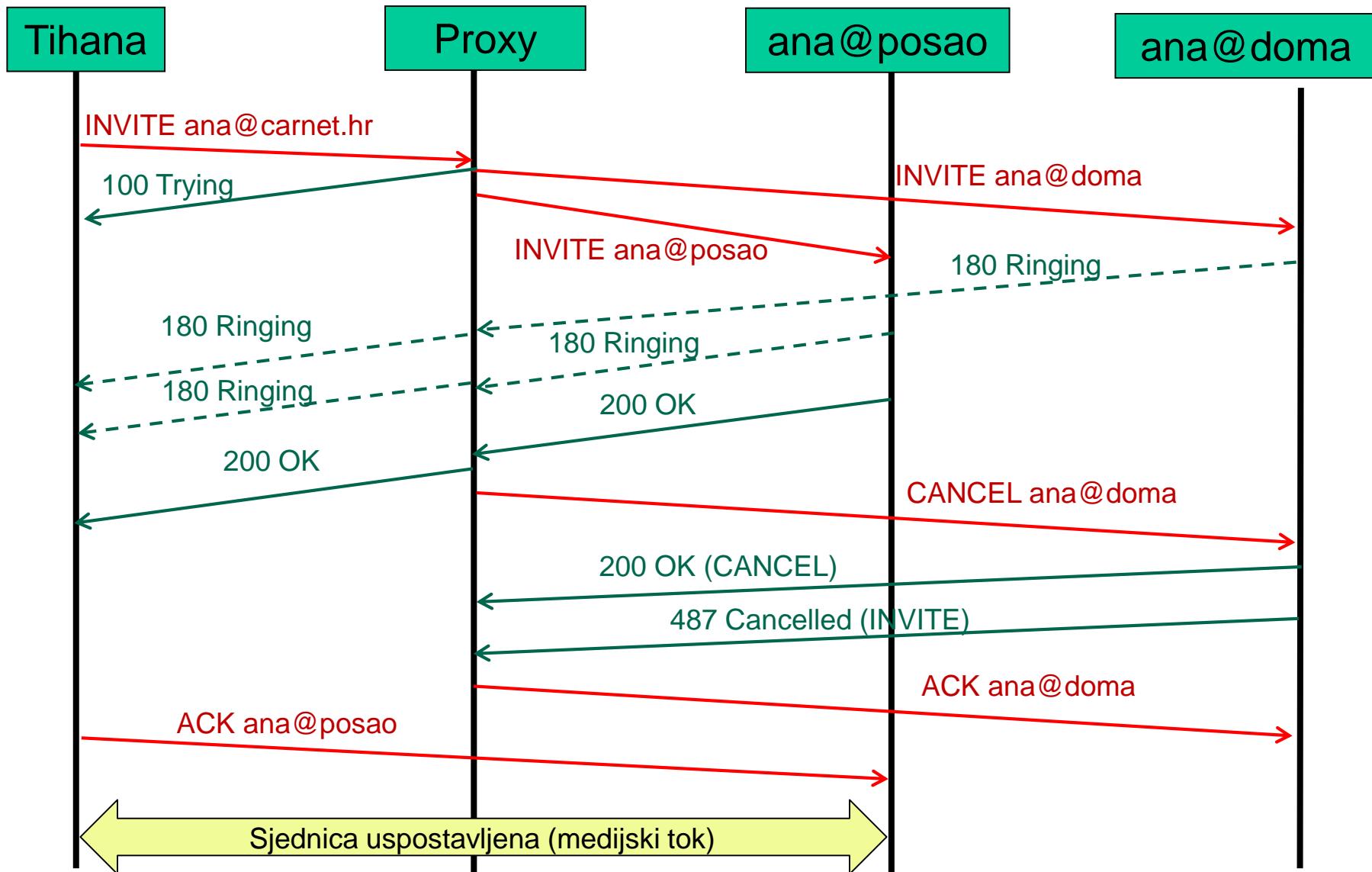
# Bilježenje rute (Record-Route)



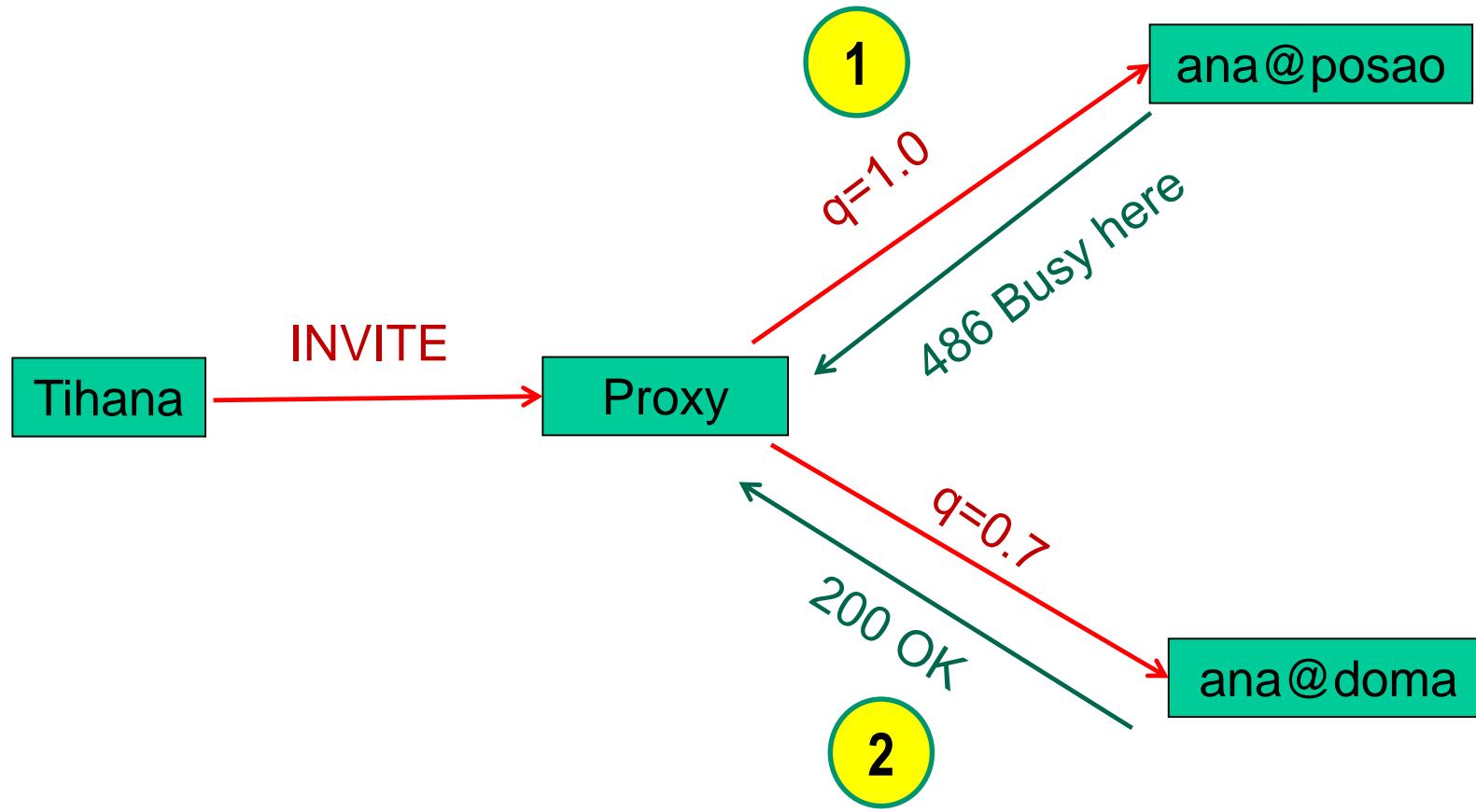
# SIP grananje zahtjeva (engl. *request forking*)



# Paralelno grananje zahtjeva

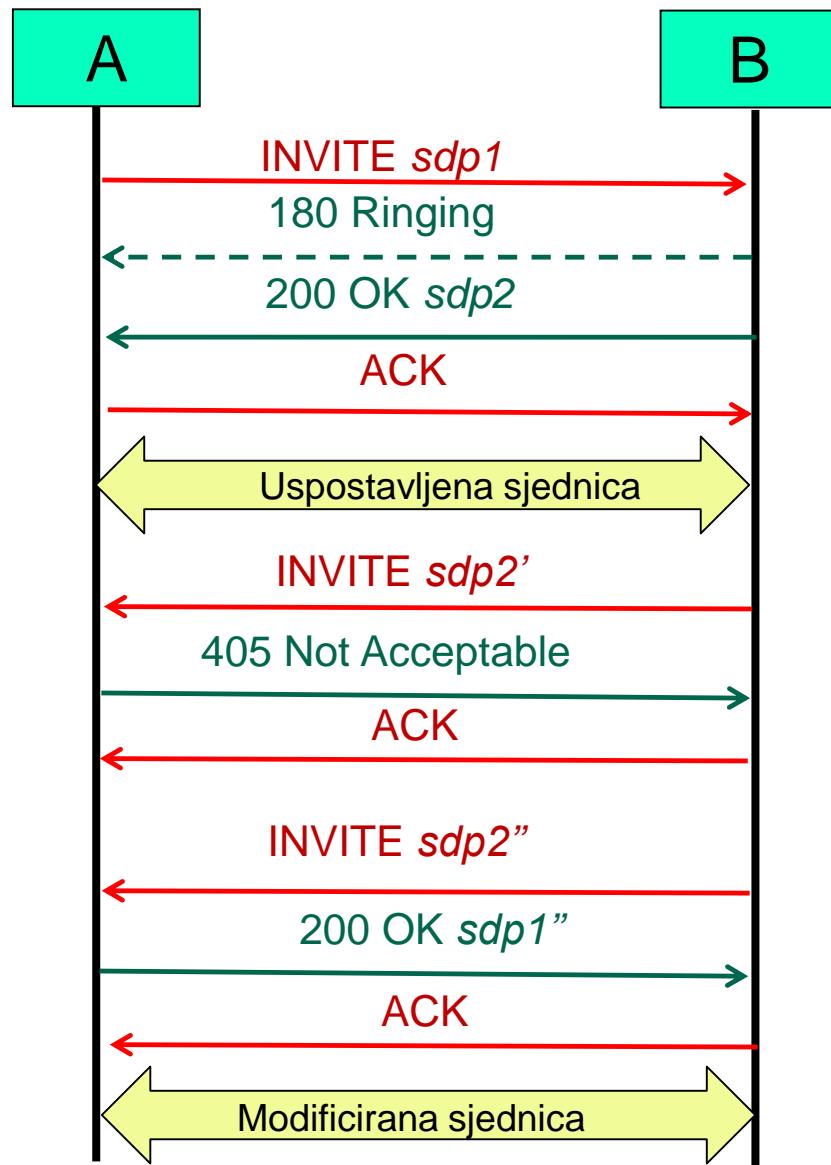


# Sekvencijalno grananje zahtjeva



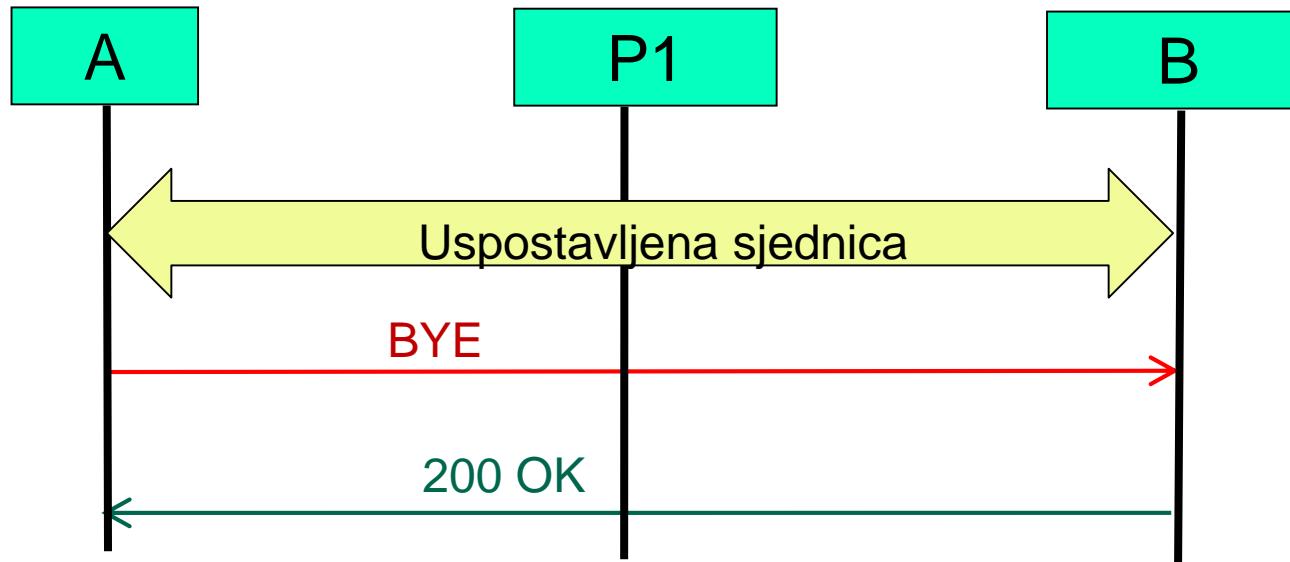
# Modifikacija postojeće sjednice

- šalje se **re-INVITE** s novim opisom sjednice (novi SDP)
- referencira se postojeći dialog
- ukoliko pozvani korisnik ne prihvata promjene, parametri sjednice ostaju nepromjenjeni



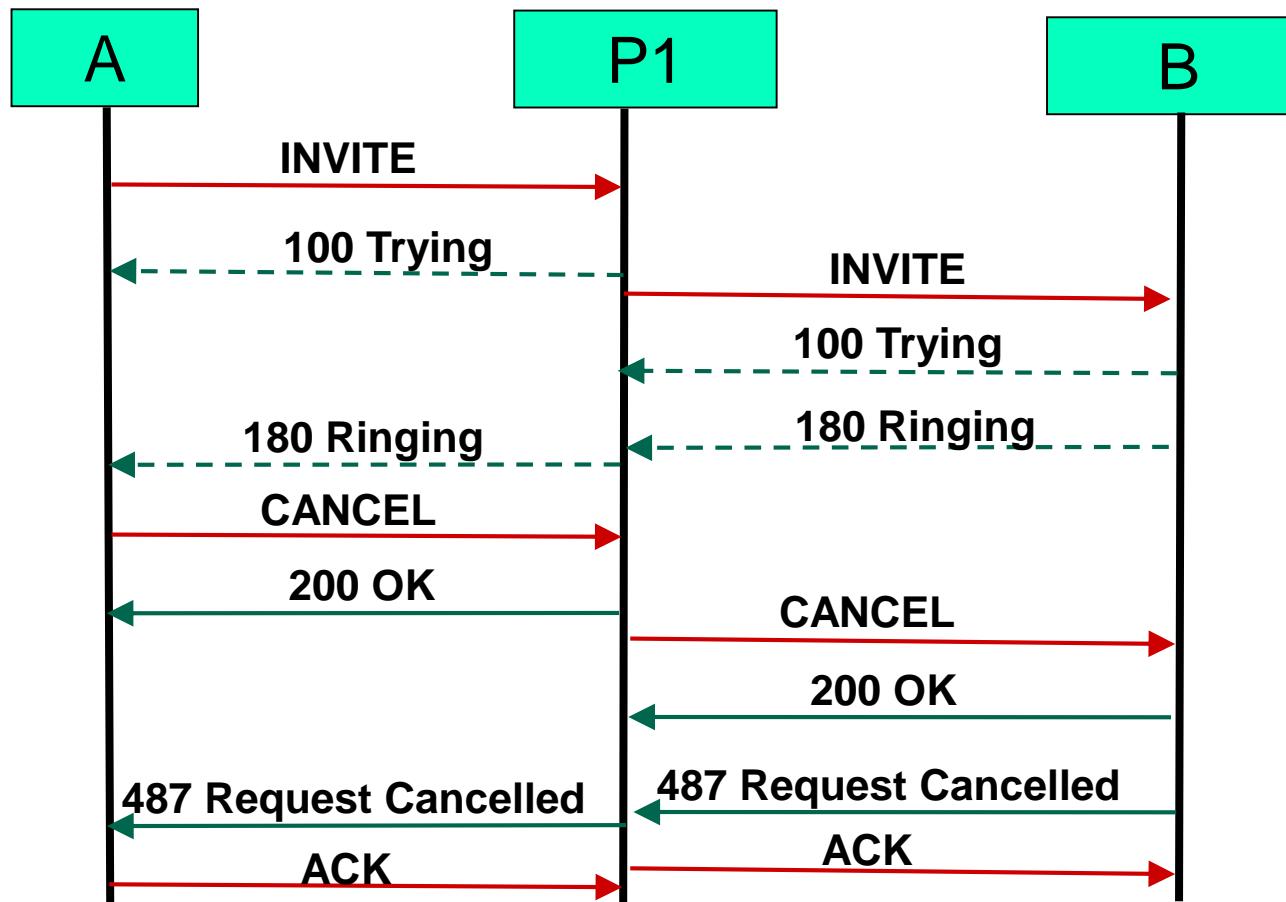
# Raskid sjednice

- Raskid sjednice: jedan od korisničkih agenata šalje zahtjev **BYE** nakon što je sjednica uspostavljena; zahtjev **s kraja na kraj**



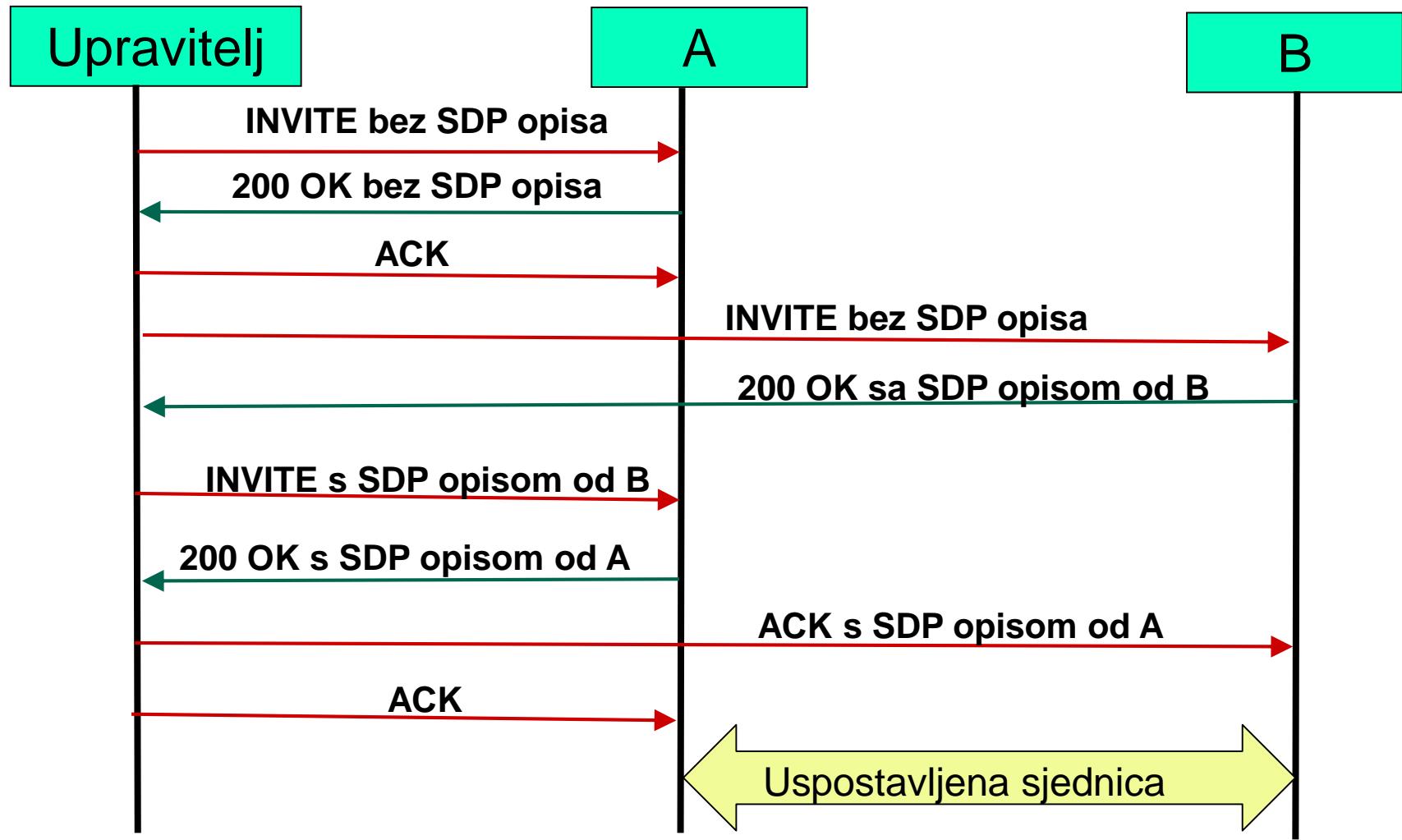
# Prekid sjednice

- Prekid sjednice: iniciran od korisničkih agenata ili posredničkog poslužitelja; šalje se zahtjev **CANCEL** tijekom uspostave sjednice; zahtjev **od točke do točke**



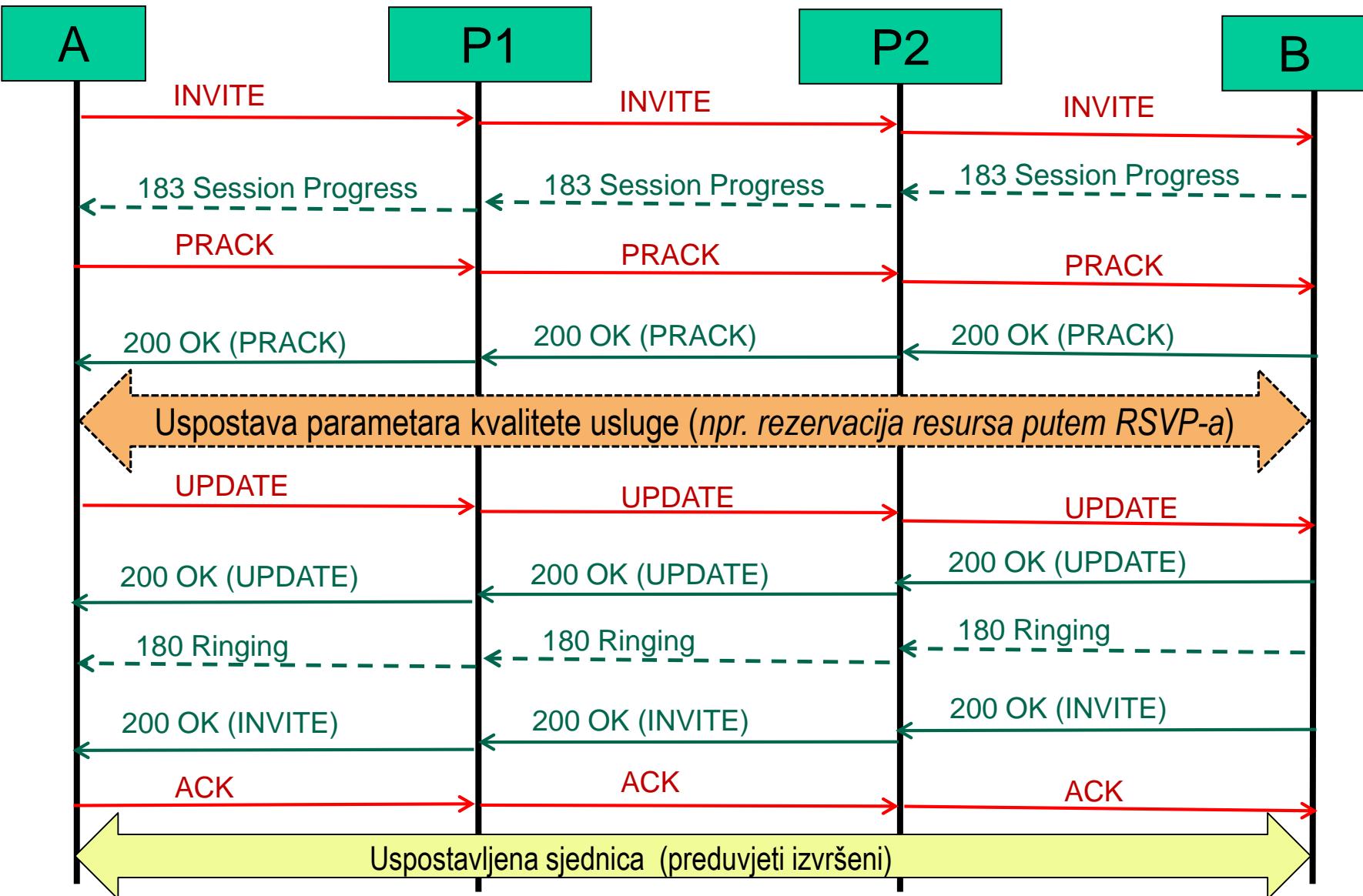
# Upravljanje pozivom od treće strane

- Engl.: *Third Party Call Control (3PCC)*



- RFC 3312: *Integration of Resource Management and SIP*
- Specificiraju se **preuvjeti** za uspostavu poziva (engl. *preconditions*)
- **Preuvjeti** zahtjevaju rezervaciju mrežnih resursa prije uspostave sjednice → osigurava se **kvaliteta usluge**
- Dodatna signalizacija potrebna
- Koristi se metoda **PRACK** (**P**rovisional **R**esponse **A**CK, RFC 3262):
  - osigurava pouzdanost privremenih odgovora
  - isto kao ACK, ali se šalje nakon primitka privremenog odgovora (183 Session Progress)
- Pozvanom korisniku ne “zazvoni” telefon dok preuvjeti (rezervacija mrežnih resursa) nisu uspješno izvršeni

# Integracija SIP-a i mehanizama upravljanja resursima (2/2)

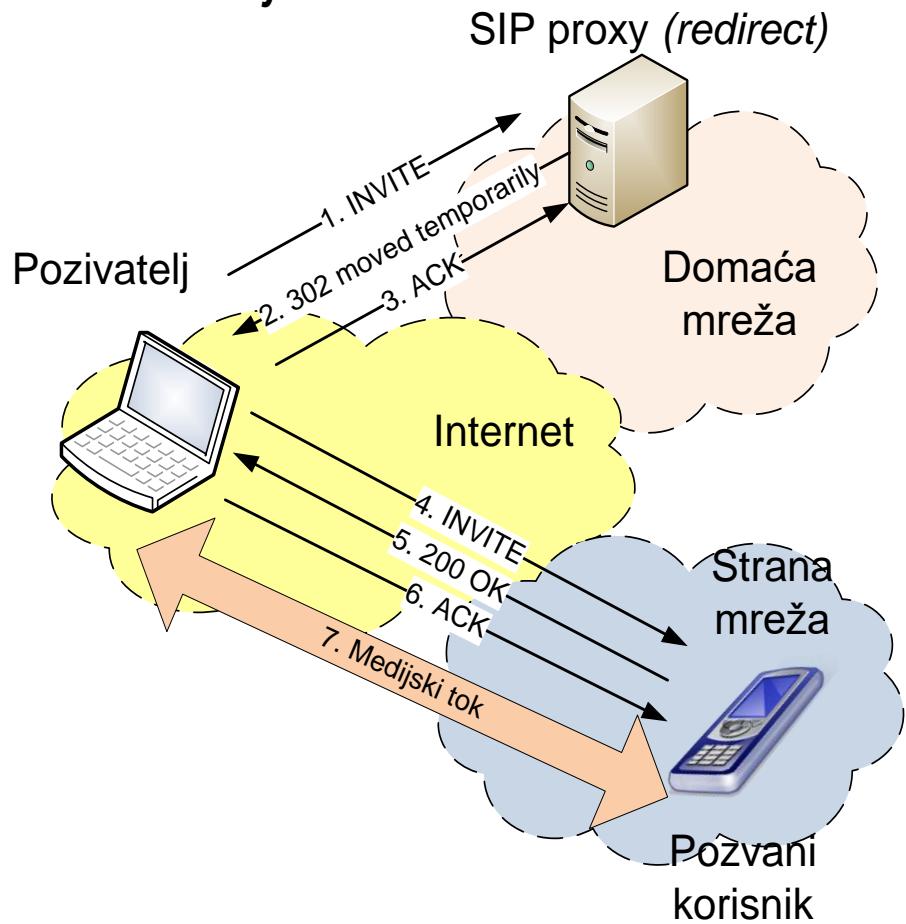


- SIP pruža podršku za pokretljivost na **aplikacijskom** sloju

<b>pokretljivost uređaja/terminala</b> (engl. <i>terminal mobility</i> )	uređaj mijenja položaj i/ili pristupnu točku u mreži
<b>pokretljivost osobe</b> (engl. <i>personal mobility</i> )	osoba koristi različite uređaje za pristup uslugama; adresiranje korisnika pomoću jedinstvene adrese
<b>pokretljivost usluge</b> (engl. <i>service mobility</i> )	korisnik ostvaruje usluge „u pokretu“ (pristup uslugama neovisan o promjeni uređaja i/ili mreže )
<b>pokretljivost sjednice</b> (engl. <i>session mobility</i> )	korisnik mijenja uređaje za vrijeme odvijanja komunikacije

# Pokretljivost uređaja/terminala

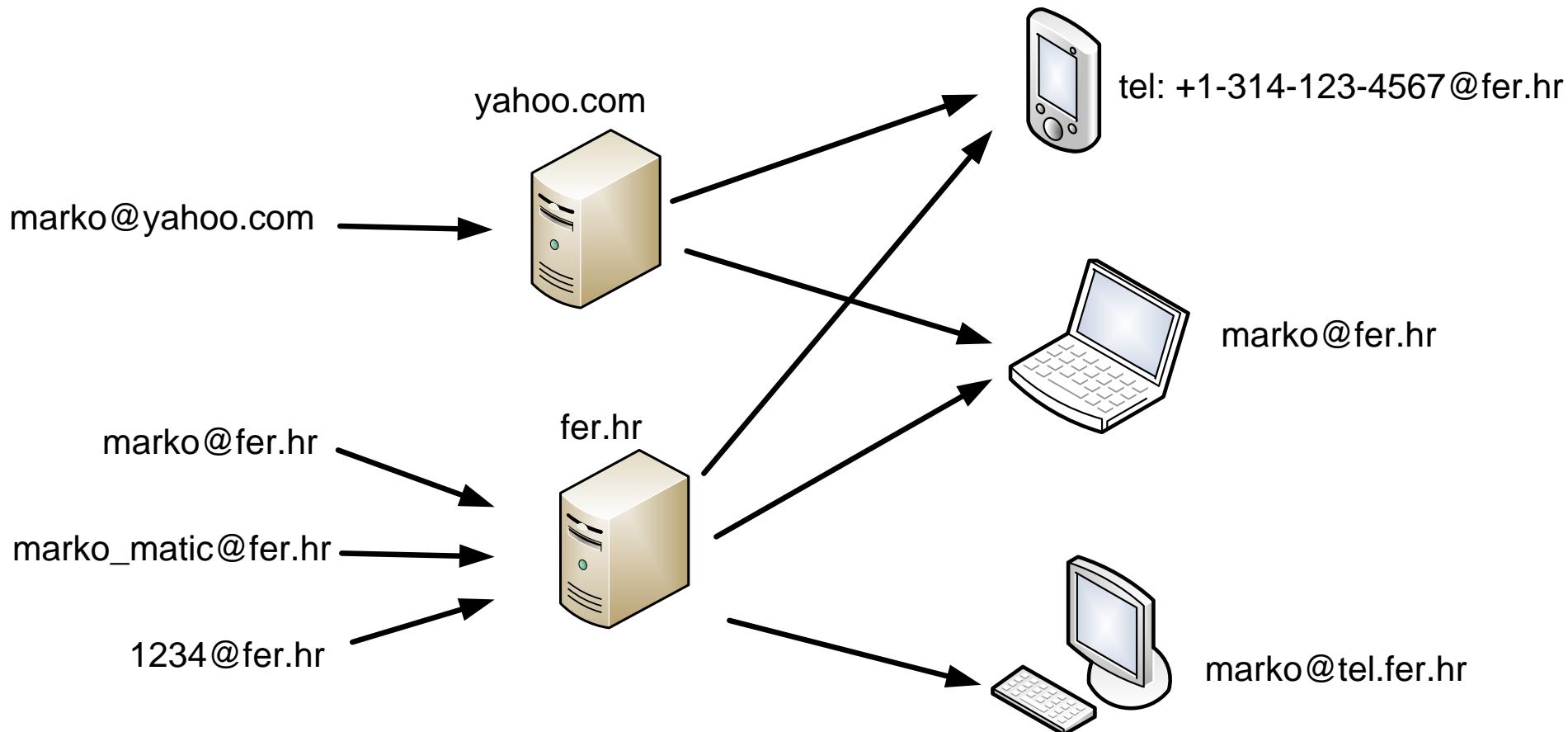
## Pokretljivost prije uspostave sjednice



## Pokretljivost tijekom uspostavljenje sjednice (re-INVITE)



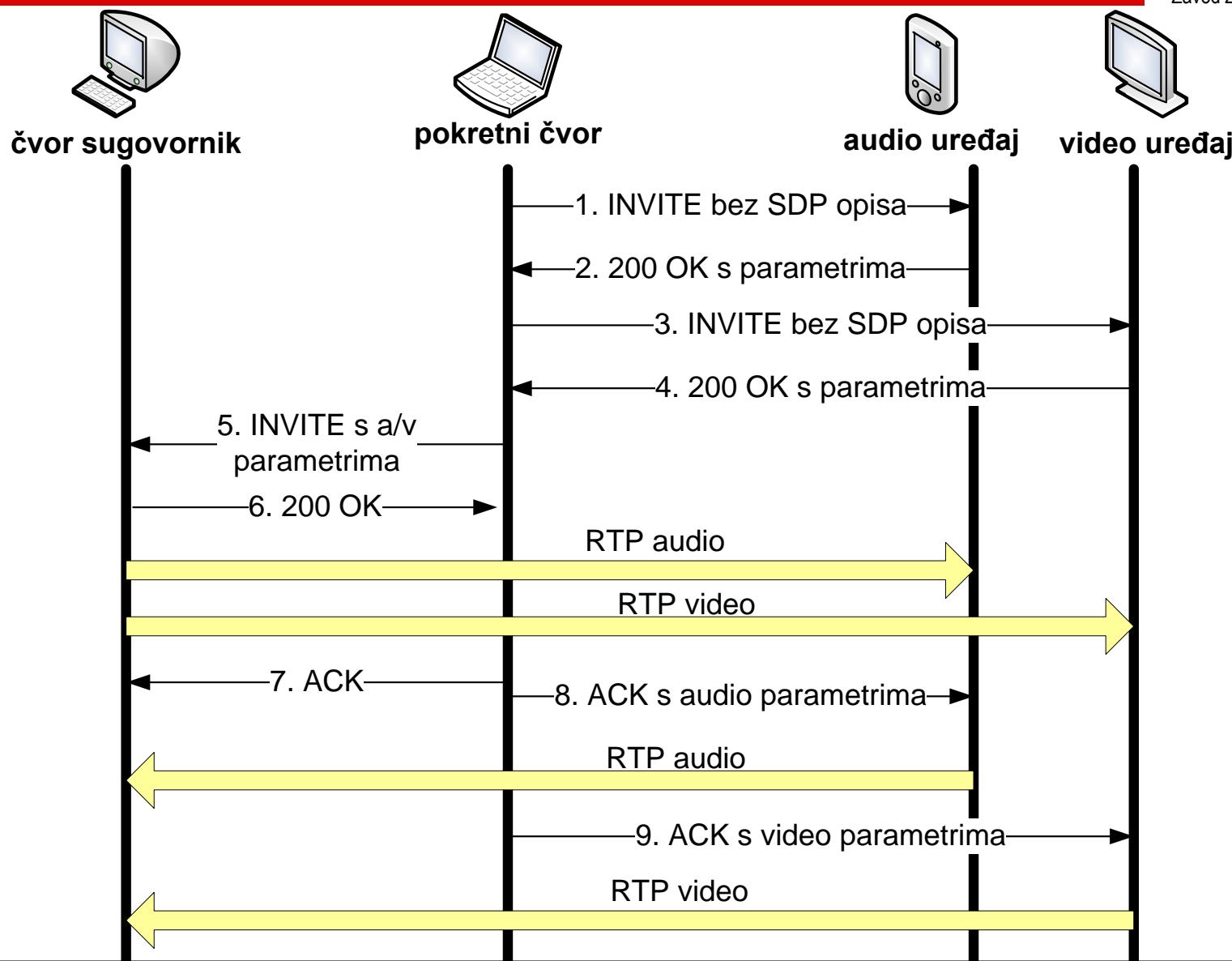
# Pokretljivost osobe



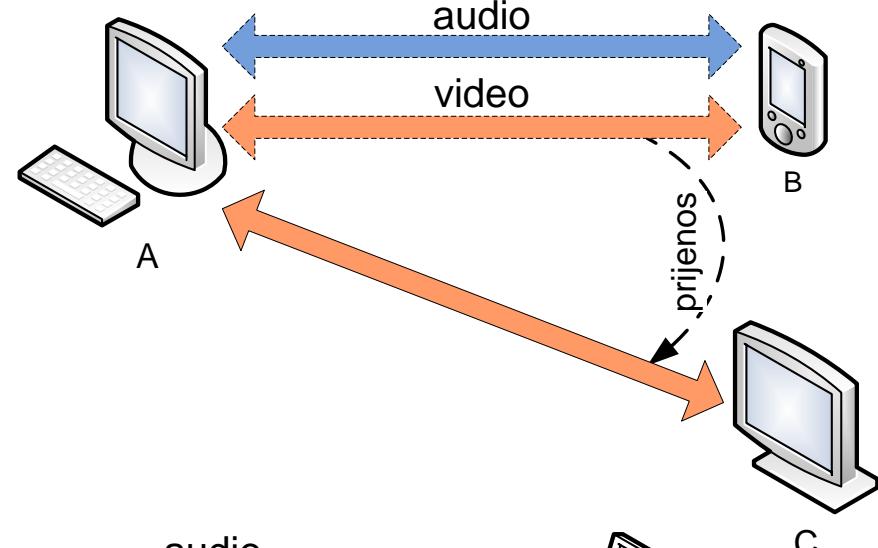
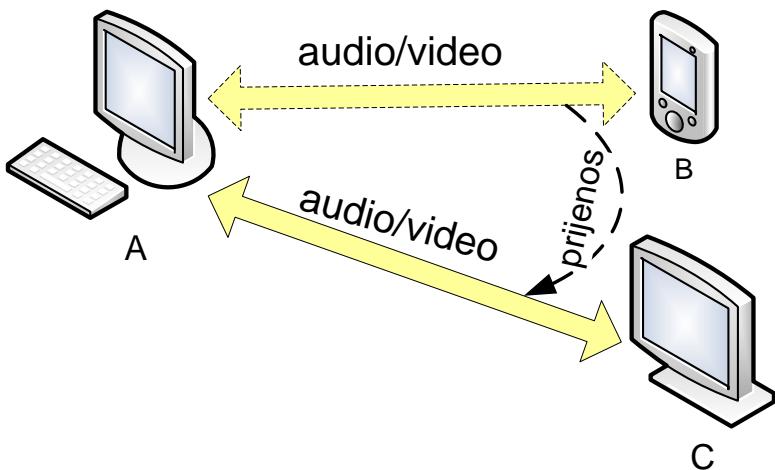
# Pokretljivost sjednice

- RFC 5631 (2009): *SIP Session Mobility*
- Moguće prenijeti sjednicu (engl. *transfer*) na drugi uređaj, te ju ponovnu vratiti na početni uređaj (engl. *retrieval*)
- Dva načina prijenosa sjednice:
  - **Metoda upravljanja pomoću pokretnog čvora** (engl. *Mobile Node Control*, MNC): pokretni čvor koristi mehanizam upravljanja pozivom treće strane (3PCC) te kontrolira signalizaciju
  - **Metoda preuzimanja sjednice** (engl. *Session Handoff*, SH): pokretni čvor prenosi sjednicu slanjem SIP zahtjeva *REFER*; potpuno se prenosi signalizacija i medijski tokovi

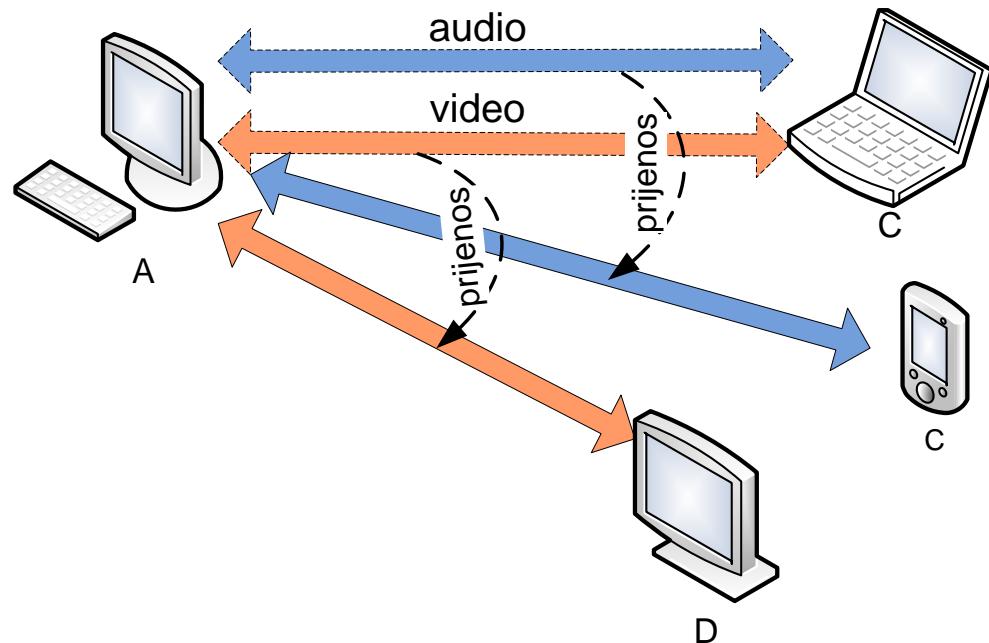
# Metoda upravljanja pomoću pokretnog čvora



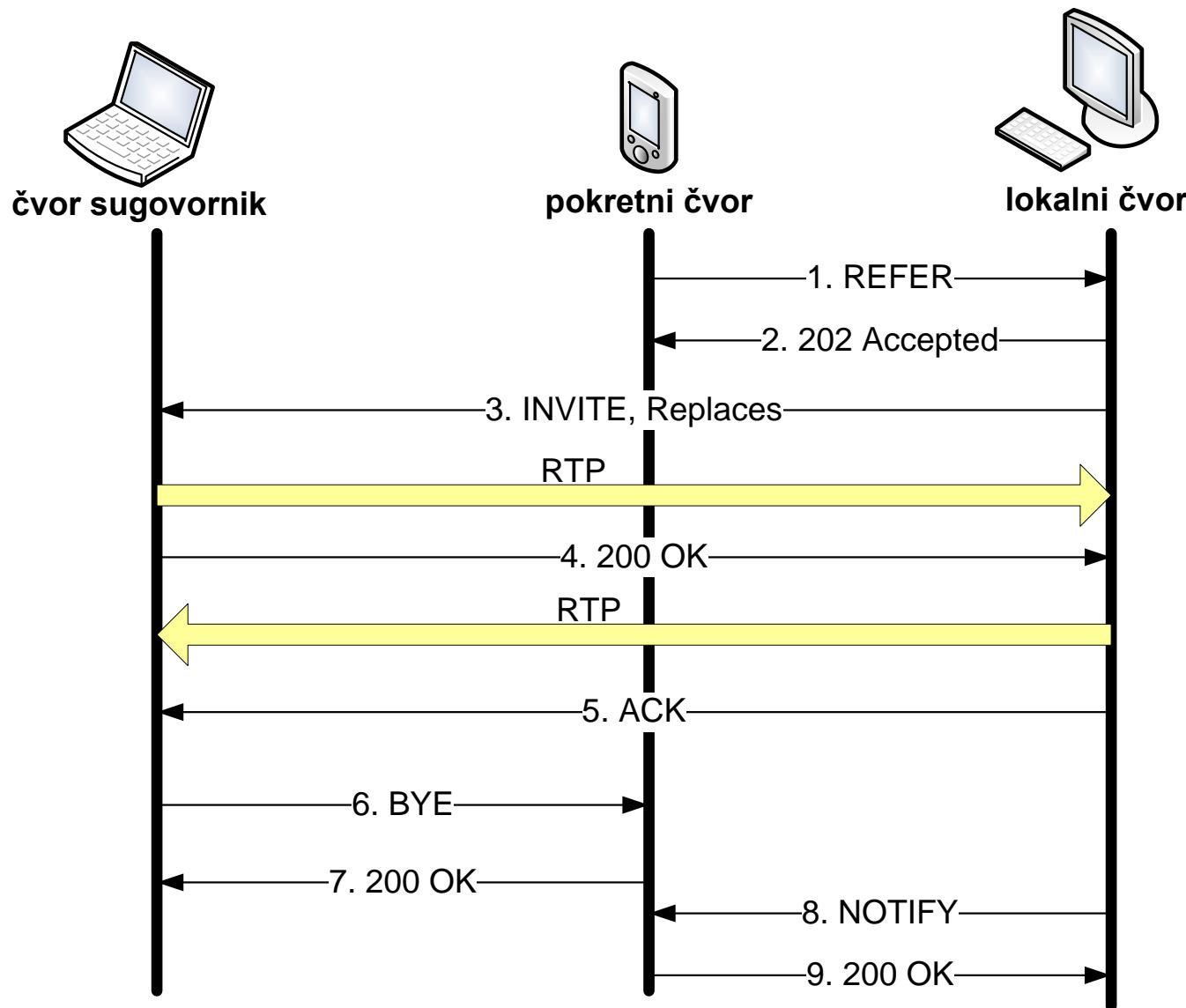
# Metoda preuzimanja sjednice (1/2)



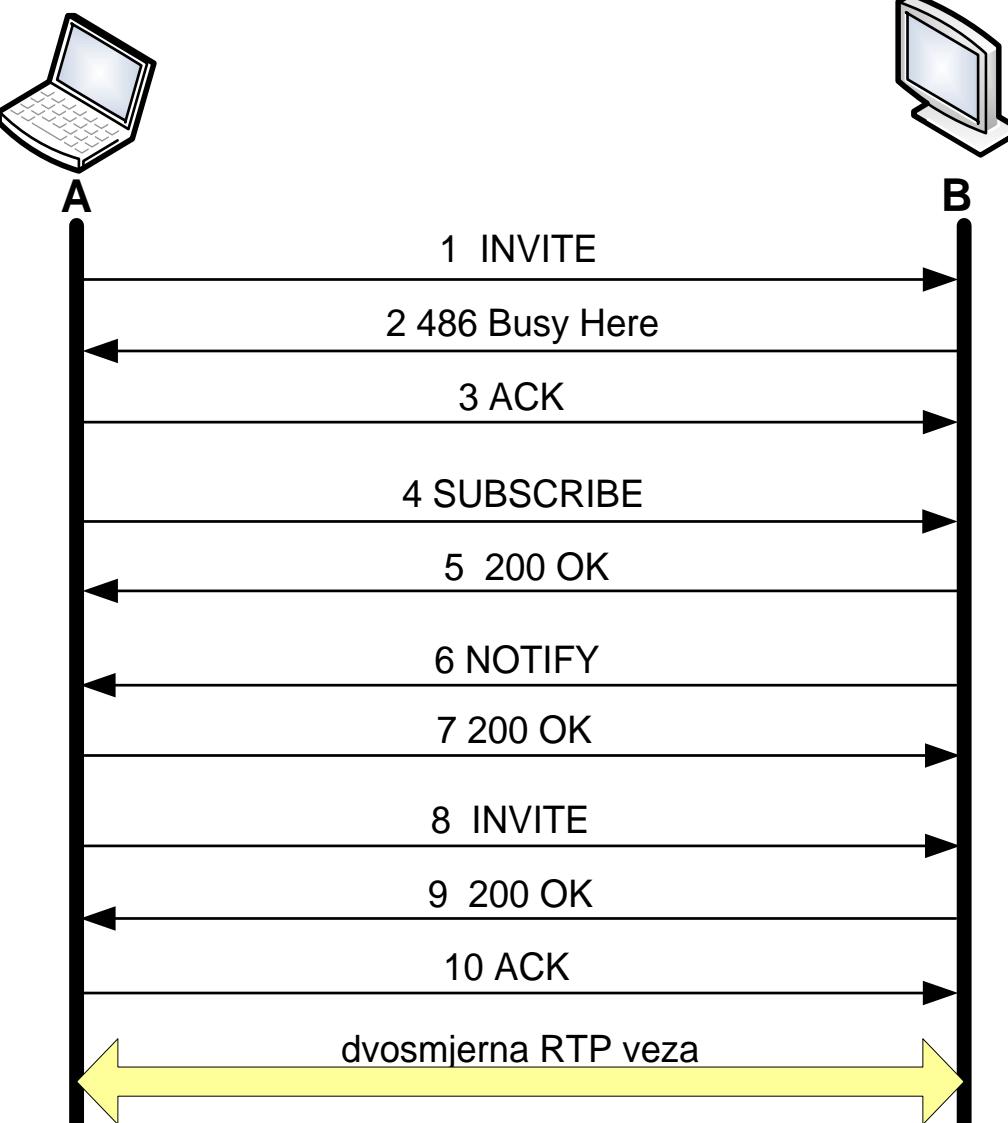
Prijenos cijele  
sjednice ili  
prijenos pojedinih  
komponenata



# Metoda preuzimanja sjednice (2/2)



# SIP PROŠIRENJA



**SUBSCRIBE** sip:UserB@there.com SIP/2.0

Via: SIP/2.0/UDP 4.3.2.1

To: User B <sip:UserB@there.com>

From: User A <sip:UserA@here.com>

Call-ID: a5-32-43-12-77@4.3.2.1.

CSeq: 1 SUBSCRIBE

Event: Available

Content-Length: 0

**NOTIFY** sip:UserA@here.com SIP/2.0

Via: SIP/2.0/UDP 129.5.3.2:5060

To: User A <sip:UserA@here.com>

From: User B <sip:UserB@there.com>

Call-ID: 52525213@129.5.3.2

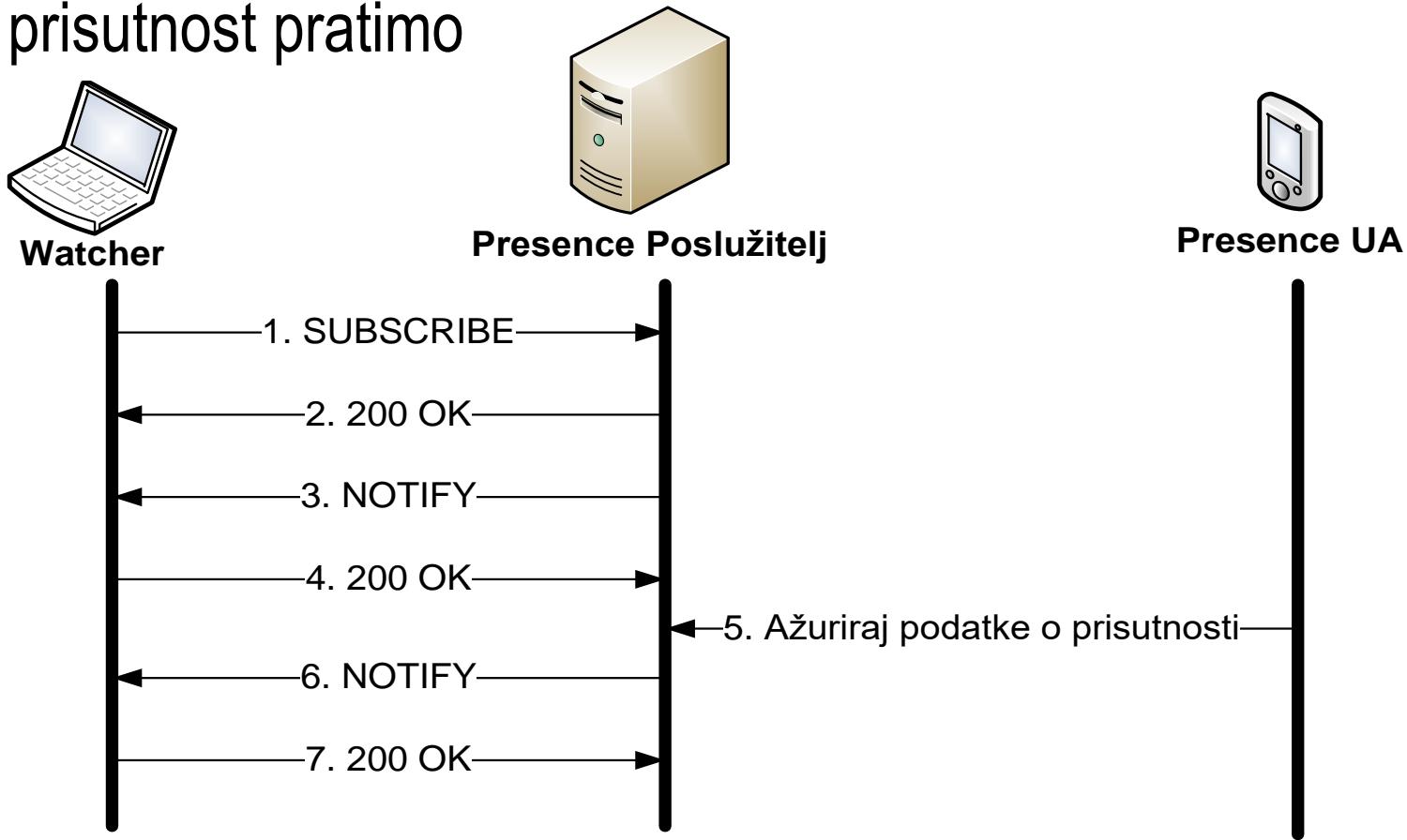
CSeq: 5 NOTIFY

Event: Available

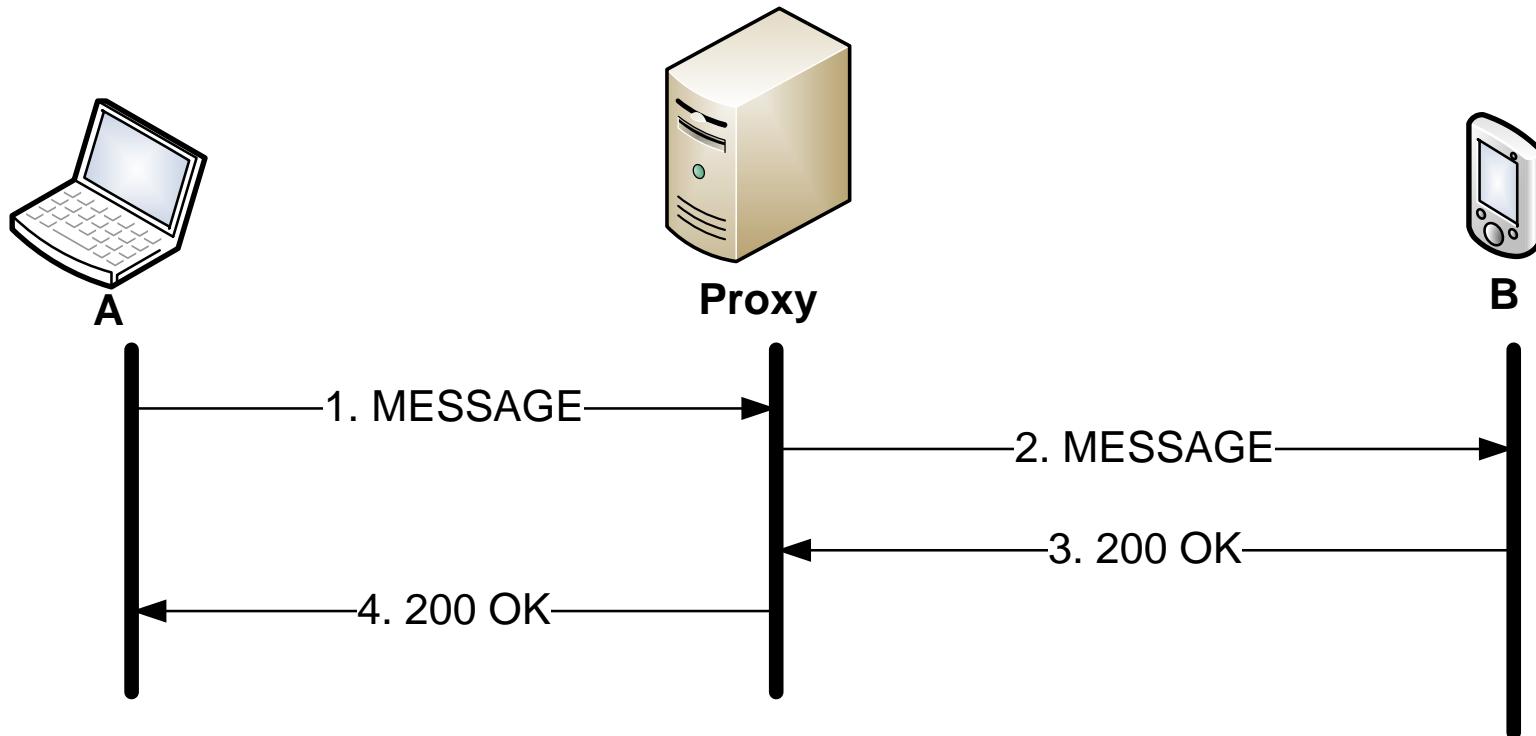
Content-Length: 0

- SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE)
  - Radna skupina IETF-a koja definira proširenja protokola SIP za prisutnost i trenutno poručivanje
- SIP poruke:
  - SUBSCRIBE i NOTIFY za prisutnost [RFC 3856]
  - MESSAGE za trenutno poručivanje [RFC 3428]

- *Watcher*: korisnik koji prati prisutnost
- *Presence UA*: manipulira podacima o prisutnosti za korisnika čiju prisutnost pratimo



- Razmjena kratkih tekstualnih poruka između grupe korisnika u gotovo stvarnom vremenu
- MESSAGE poruke same po sebi ne uspostavljaju SIP dialog



# Kreiranje SIP usluga

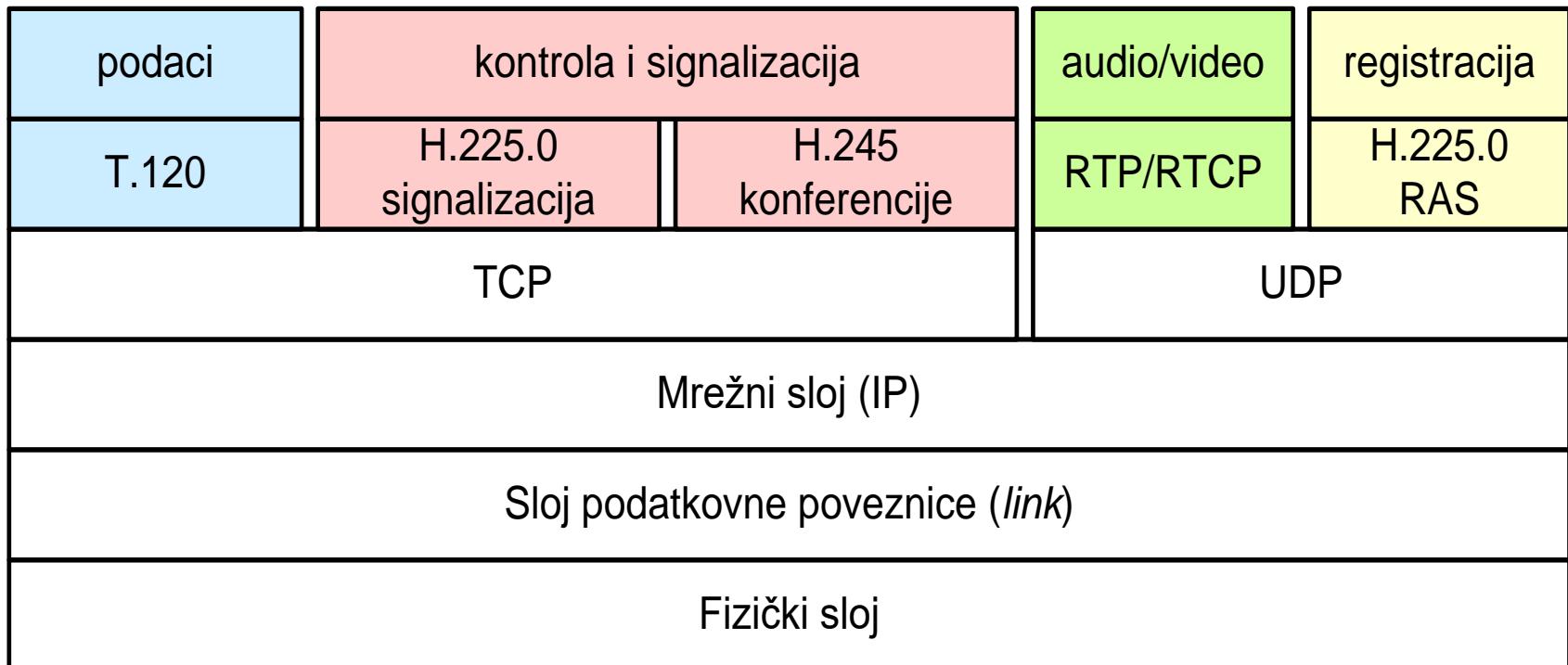
- Za izvedbu usluga koriste se sljedeće tehnologije:
  - CPL (*Call Processing Language*)
  - SIP CGI (*Common Gateway Interface*)
  - SIP API (*Application Programming Interface*):
    - . JAIN (*Java APIs for Integrated Networks*) SIP protokolni složaj
    - . SIP servleti (IETF i JAIN)
  - SIP i VoiceXML

- Pojmovi i terminologija
- Arhitektura za VoIP zasnovana na protokolu SIP
- **Arhitektura za VoIP zasnovana na standardu H.323**

# Standard H.323

- ITU-T preporuka H.323 opisuje terminale i druge entitete te definira protokole za pružanje višemedijskih komunikacijskih usluga u paketskim mrežama bez garantirane kvalitete usluge (primjerice, u Internetu)
- specifikacija (okvirne) arhitekture za višemedijsku komunikaciju - obuhvaća niz drugih specifikacija
- svojstva:
  - standardna kompresija/dekompresija
  - povezivanje različite opreme
  - neovisnost o mreži
  - neovisnost o opremi i aplikacijama
  - podrška za konferencijsku vezu
  - nadzor mreže
  - podrška za komunikaciju s više krajnjih točaka

# H.323 protokolni složaj

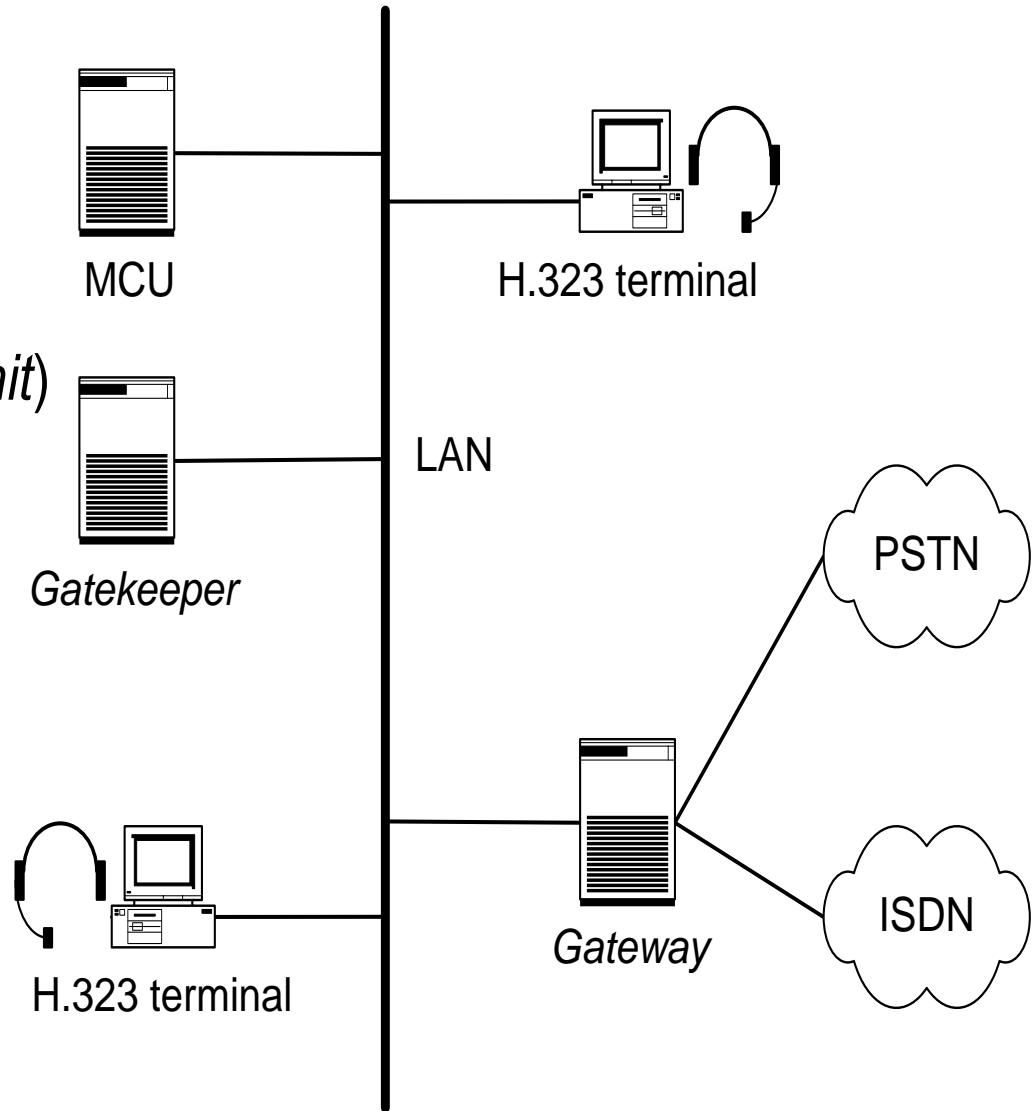


# Protokoli standarda H.323

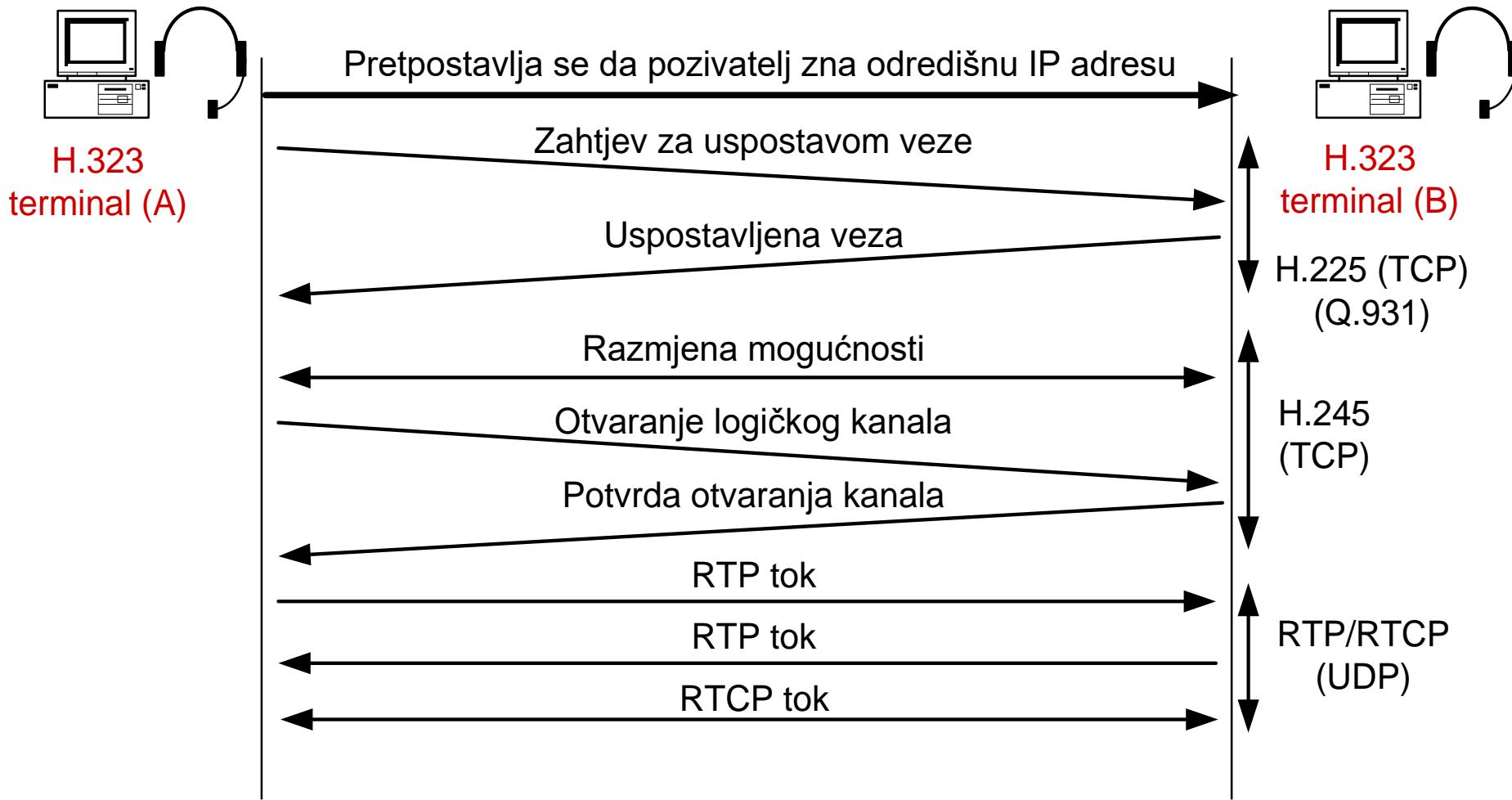
- H.225.0 Registration, Admission, and Status (RAS)
  - upravljanje prijavom krajnje točke, kontrola pristupa krajnjim uređajima, razlučivanje adresa
- H.225.0 Call Signaling
  - signalizacija između krajnjih točaka kod uspostave veze
- H.245 (Control Signaling)
  - kontrola višemedijske veze/komunikacije
- Real-time Transport Protocol (RTP)
- RTP Control Protocol (RTCP)
- T.38, T.120, V.150, itd.

# Komponente standarda H.323

- Terminal
- *Gateway*
- *Gatekeeper*
- MCU (*Multipoint Control Unit*)



# Uspostava veze



# H.323 Gateway

- Omogućava vezu s različitim vrstama ne-H.323 terminala:
  - analogni PSTN terminali, ISDN terminali, B-ISDN terminali
- Obavlja pretvorbu protokola za uspostavu i raskidanje veze/poziva
- Obavlja pretvorbu formata medija između različitih mreža
- Obavlja prijenos podataka između H.323 i ne-H.323 mreža

