

Višemedijske komunikacije (ak. god. 2009./2010.)

Pitanja za samostalnu provjeru znanja i pripremu usmenog ispita (1. blok predavanja)

Osnove iz TINFacha

1. Osnovno načelo- Izvorno kodiranje je širok skup raznovrsnih metoda kodiranja koje najčešće vrše kompresiju s gubitcima. Za razliku od metoda entropijskog kodiranja, ove metode ne koriste samo **statistička svojstva izvora**, nego promatraju i **semantiku izvora**, tj. posebna svojstva pojedinih medijskih sadržaja koji se kodiraju (zvuk, slika, tekst, video). Metode izvornog kodiranja koriste i poznavanje karakteristika ljudske percepcije pojedinih medija za ostvarivanje bolje kompresije uz minimalno zapazanje pogreske.
2. Izvorno kodiranje je u velikoj mjeri ovisno o mediju, jer omjer kompresije ovisi o sadržaju. Može li se uspješna metoda izvornog kodiranja razvijena za kodiranje govora jednako uspješno primijeniti na audio...? Nisam baš siguran, al mislim da ne baš, jer je za audio potrebna bolja kvaliteta, pa time je potrebno i bolje (kompleksnije) kodiranje (npr PCM- samo nelinearna kvantizacija, a MP3- potpojasno kodiranje, kvantizacija, maskiranje kojekakvo i tako jel...). O čemu to ovisi? ne kuzim pitanje... pa valjda o nama- jel ocemo slusat mp3 prek telefona il prek playera...
3. Metode izvornog kodiranja koriste se u sklopu hibridnih metoda za kodiranje pojedinih medija. Pritom se obično prvo vrši izvorno kodiranje, često kombinacijom dviju ili više osnovnih metoda, a rezultat tog kodiranja se zatim kodira entropijski. Ocito zato da dobijemo još veću kompresiju... dakle kodiramo izvornim kodiranjem (s gubitcima) i onda još entropijski (bez gubitaka)
4. Kompresija podataka metodama izvornog kodiranja zasniva se na uklanjanju **zalihosti (redundancije) i irelevantnosti** iz podataka. **3 vrste redundancije**: vremenska (sastoji se od korelacije uzastopnih simbola u vremenskom slijedu- naci slični uzastopni uzorci), prostorna (sličnost uzastopnih elemenata u prostoru- boja uzastopnih tocaka na slici npr), spektralna (korelacija spektralnih komponenti medija- sličnost među komponentama boje na slici (RGB)). **Irelevantnost**- pojava dijela podataka koji nisu važni pa ih se može odbaciti a da se pritom zadrži željena razina kvalitete podataka. Dakle nepotrebno je kodirati one dijelove informacije koje ljudska osjetila ne mogu primjetiti (možda si misliš... ima nas i gluhih- diskriminacija!). Ako ovo nije odgovor na pitanje onda je to ono sa kvantizacijom i periodičnim uzimanjem uzoraka... a to nebudem pisao fakju...
5. Koder s gubicima - velik stupanj kompresije; koder bez gubitaka - manje kašnjenje, mogu se u potpunosti rekonstruirati izvorni podatci
6. Obrnuto od prednosti :D
7. Prvi korak je **uzrokovanje**, prilikom kojeg se signal, koji je već odgovarajućim uređajima i sklopovljem pretvoren u električni signal, pretvara u niz digitalnih uzoraka. Uzorci se uzimaju periodički. Sto je više uzoraka uzeto, dakle sto je veća frekvencija uzrokovanja to će uzrokovani signal vjernije prikazivati originalni. **Frekvencija uzrokovanja** mora biti barem dvostruko veća od max frekvencije signala (inace se gubio osnovni signal i preklapaju se spektralni pojasi). Zatim slijedi **kvantizacija**, odnosno svrstavanje ili zaokruživanje uzoraka na jednu od predviđenih diskretnih razina. Sto je manje razina kvantizacije to će opseg podataka nakon kvantizacije biti manji, ali će odstupanje, koje nastaje kvantizacijom biti veće.
8. Izlazna brzina koda = frekvencija uzrokovanja * broj kvantizacijskih bitova Npr.- kod PCM: $f_u = 8\text{kHz}$, broj bitova za kvantizaciju = 8 (256 razina)

9. $5 \cdot 8 < 8 \cdot 8$:D
10. Ako znamo da neko područje vrijednosti signala ima veću vjerojatnost pojavljivanja, onda ima smisla u tom području gušće (masti) rasporediti intervale kvantizacije. Na taj se način smanjuje pogreška kvantizacije, jer se bolje pokriju one vrijednosti koje se česće pojavljuju. To sve skupa dosta ovisi o izvoru jel... ako na izvoru imamo neki kontinuirani signal- bolje linearna... a ako je nesto ludo- ima pa nema pa ima ima pa nema- bolje nelinearna
11. Samo ovo sam našo „The μ -law algorithm provides a slightly larger dynamic range than the A-law at the cost of worse proportional distortion for small signals. By convention, A-law is used for an international connection if at least one country uses it.“ ...pa si vi sami protumacite koji je brzi ili su isti...
12. Ima i slika na 23. slajdu 2. predavanja pa ako to neko kuži znat će i odgovor
13. Ljudi su zapazili da je efikasnije kodirati blokove simbola nego pojedinačne simbole i tako nastala vektorska kvantizacija. **Postupak:** simboli poruke se prvo grupiraju u blokove- vektore- od po n simbola. Ti vektori mogu npr. biti nizovi od po n uzastopnih uzoraka zvucnog signala. Ovi n -dimenzionalni vektori predstavljaju ulaz u koder koji vrši kvantizaciju. U koderu i u dekoderu postoji jednaka **kodna tablica**. Kodna tablica se sastoji od liste n - dimenzionalnih vektora koji se zovu **kodni vektori**, te liste njima pripadajućih **indeksa**- naci svaki kodni vektor ima svoj index. U postupku kodiranja se svaki ulazni vektor uspoređuje s kodnim vektorima u kodnoj tablici i pronalazi se kodni vektor najslabiji ulaznom vektoru(po euklidskoj distanci- zvuci pametno). Indeks pronađenog kodnog vektora je izlaz koda. Dekoder naravno vidi indeks - nadje- dekodira.
14. **Povećavanjem** dimenzije vektora kvantizacije dobivaju se sve bolji i bolji rezultati- dakle **sve manja kvantizacijska pogreška**. **MEDJUTIM**, postoji granica jer za veće grupe točaka pada vjerojatnost da budu slične, tj da se dobro grupiraju za kvantizaciju (naci nie dobro pretjerat- osim s alkoholom). U praksi za sliku- 4 dimenzije, za zvuk- 40-50 dimenzija. E sad na primjeru nepomicne slike- knjiga TINF- 290.str- ko voli nek čita
15. Poduzrokovanje je postupak smanjivanja frekvencije uzrokovanja, tj broja uzoraka. Kod zvuka je to broj uzoraka zvuka u sekundi. Osnovna frekvencija uzrokovanja je određena sklopom za A/D pretvorbu kod signala koji su izvorno u analognom obliku. Smanjivanjem frekvencije uzrokovanja direktno se smanjuje broj uzoraka signala i time postiže kompresija. Poduzrokovanje se često vrši i s manjom freq uzrokovanja nego što dopušta Nyquistov teorem($f_{eq} > 2 \cdot \max \text{freq izvornog sig}$). Npr- u telefoniji zvuk se može poduzrokovati freq 8000Hz, što je sasvim dovoljno za razumjevanje govora. Ali isto to poduzrokovanje za glazbu neku bi jako smanjilo kvalitetu zvuka i nije ok...
16. Prilikom transformacijskog kodiranja poruka se pretvara u neki drugi oblik koji je pogodniji za kompresiju. Npr- prebacivanje iz vremenske u frekvencijsku domenu. Pritom sama transformacija ne vrši kompresiju i najčešće je reverzibilna (bez gubitaka). **Kompresija** nastupa **NAKON** transformacije i to najčešće **kvantizacijom ili odbacivanjem dijela podataka**. Transformacija pretvara podatke u takav oblik da se znatan dio podatka može ili odbaciti ili kvantizirati uz velik stupanj kompresije. Princip transformacijskog kodiranja može se pokazati transformacijom slike u frekvencijsku domenu uporabom diskretne Fourierove transformacije. Izvorna slika se dvodimenzionalnom diskretnom FT pretvara u frekvencijske komponente, na taj način da se slika interpretira kao funkcija dviju varijabli $I=f(x,y)$, gdje su x i y koordinate točke u slici, a I vrijednost svjetline u toj točki. Na funkciju f primjenjuje se DFT.

Odbace se više frekvencijske komponente. Vratiti se natrag u prostornu domenu korištenjem inverzne DFT i dobije se kompresirana slika.

17. U frekvencijskoj domeni se odbace više frekvencije (dakle područja naglih promjena u svjetlini), to se očituje tako da se na kompresiranoj slici manje vidi ostrina slike (rubovi isto) i detalji.
18. Diferencijalno kodiranje koristi pretpostavku da se signal relativno malo mijenja od uzorka do uzorka, tj. da postoji korelacija među susjednim uzorcima. Ona može postojati u vremenu (zvuk) ili prostoru (slika). Korelacija među uzorcima se koristi da bi se vrijednost svakog uzorka predvidjela iz jednog ili više prethodnih uzoraka. Kodira se razlika između stvarnog i predviđenog uzorka. Niz takvih razlika naziva se signal razlike. Signal razlike ima manji raspon vrijednosti i manje se mijenja od izvornog signala, pa ga se može bolje kodirati.
19. Skica. Ako ne bi bilo dekodera unutar postupka kodiranja- pogreske kvantizacije se zbrajaju, i što smo dalje u nizu koji se kodira, to je veća ukupna pogreska. Razlog tome je što koder stvara signal razlike oduzimajući stvarnu vrijednost prethodnog uzorka, a dekođer koristi rekonstruiranu, pogresnu vrijednost prethodnog uzorka. Dakle dekođer u koderu služi za izbjegavanje akumulacije kvantizacijske pogreske.
20. Pretpostavimo da se koristi kvantizator sa 7 razina kvantizacije: -6, -4, -2, 0, 2, 4, 6. Ulazni niz simbola je: 6.2, 9.7, 13.2, 5.9, 8, 7.4, 4.2, 1.8 ; Signal razlike je: 6.2, 3.5, 3.5, -7.3, 2.1, -0.6, -3.2, -2.4.
Nakon kvantizacije i inverzne kvantizacije dobijemo: 6 4 4 -6 2 0 -4 -2
Rekonstruirani signal se dobiva zbrajanjem rekonstruiranog signala razlike i prethodnog uzorka rekonstruiranog signala: 6 10 14 8 10 10 6 4
Pogreska nastala: 0.2 -0.3 -0.8 -2.1 -2 -2.6 -1.8 -2.2
Pogreska na početku mala, a kasnije sve veća....
21. Potpojasno kodiranje se zasniva na razdvajanju signala na frekvencijske komponente. Razdvajanje na freq pojaseve vrši se uporabom skupine filtera, čime se umjesto jednog signala dobiva niz signala, od kojih svaki odgovara jednom frekvencijskom pojasu. Svaki signal je drukcije važnosti s obzirom na ljudsku percepciju. Ti signali se kodiraju svaki zasebno, te se nakon dekodiranja ponovo skupinom filtera slazu natrag u jedinstveni signal. Svaki pojas se kodira s manje ili više bitova (ovisno o važnosti). Pritom se za pojedine frekvencijske pojaseve mogu koristiti i različite metode kodiranja. Primjer potpojasnog kodiranja- MP3 koder
22. Ako možemo postaviti dobar model izvora podataka, i taj model opisati malim brojem parametara, tada je dovoljno kodirati takve parametre, te pomoću njih i modela izvora u dekođeru sintetizirati podatke vrlo slične izvornim podacima. Pritom se uopće ne prenose sami uzorci podataka ni u kakvom obliku, i tu je bitna razlika između koderâ zasnovanih na modelu i ostalih.

Kodiranje zvuka

1. Po čemu se razlikuju, na primjer, frekvencijski spektar zvučnog signala govora, glazbe, zvukova iz prirode?

Za neperiodične zvukove (buka, šum, zvukovi iz prirode, bezvučni glasovi) karakterističan je kontinuirani frekvencijski spektar, odnosno ne postoje jako istaknute pojedine frekvencije kao kod harmoničnog spektra (harmonici- višekratnici osnovne frekvencije). U ljudskom govoru, govorni trakt proizvodi određene frekvencije koje karakteriziraju pojedine glasove. Te se rezonantne frekvencije govornog trakta nazivaju formanti.

2. Ukratko opišite svojstva i ograničenja ljudske percepcije zvuka koja su važna za kodiranje zvuka.

Ljudsko osjetilo sluha primjećuje visinu i glasnoću zvuka. Kada se radi o frekvenciji, jednake rastuće korake frekvencije primjećujemo kao sve manje i manje korake rasta visine zvuka. Ljudsko čujno područje je od 20Hz do 20 kHz. Glasnoća zvuka ovisi o amplitudi zvučnog signala i mjeri se u dB. Jednake rastuće korake *amplitude* (intenziteta) primjećujemo kao sve manje i manje korake rasta *jačine* odnosno, *glasnoće zvuka*. Dinamički raspon ljudskog sluha je 120dB, od granice čujnosti na 0dB do granice bola na 120dB.

3. Ukratko opišite svojstva ljudske "proizvodnje" zvuka koja su važna za kodiranje govora. Što je karakteristično za govor i koji su parametri važni za kodiranje?

Govor nastaje prolaskom zraka kroz govorne organe koji formiraju zvuk. Kod zvučnih glasova, glasnice titraju i stavljaju određene frekvencije zvuka. Kod bezvučnih nema titranja glasnice, samo šum zraka kroz organe. Ljudi mogu proizvesti glasove od 60Hz do 8kHz, s dinamičkim rasponom od 40dB. Najvažnije frekvencije za razumljivost govora su u intervalu od 2 do 5 kHz.

Proces nastajanja govora prilično je dobro poznat i može se simulirati na računalu modelom ljudskih govornih organa. Tada se više ne kodiraju uzorci signala, nego se uzorci signala analiziraju kako bi se iz njih izvukli parametri govornog modela. Ti se parametri kodiraju i šalju dekoderu, gdje se govor rekonstruira pomoću parametara na temelju modela. Pošto su parametri modela puno kompaktniji od samih uzoraka zvuka, ovim se postupkom postiže vrlo visoki stupanj kompresije.

4. Koje su karakteristike (frekvencija, amplituda) zvučnih, odnosno bezvučnih glasova u ljudskom govoru? Kako se mogu iskoristiti za kodiranje?

Amplituda signala zvučnog glasa nekoliko je puta veća od amplitude bezvučnog glasa. Kod bezvučnog glasa ne možemo ustanoviti nikakvu pravilnost signala, dok kod zvučnog uočavamo jasnu periodičnost nastalu zbog titranja glasnice.

Frekvencijski spektar bezvučnog glasa nema pravilnosti ili karakterističnih frekvencija govornog signala, dok se kod zvučnog glasa jasno uočavaju karakteristične frekvencije, odnosno formanti koji karakteriziraju pojedine zvučne glasove.

U govornom signalu češće se pojavljuju uzorci manjih amplituda. Takvo svojstvo govornog signala omogućava korištenje kvantizatora s nelinearnom karakteristikom, tako da se više razina kvantizacije dodjeljuje području manjih amplituda signala.

U zvučnom signalu postoji izrazita vremenska redundancija, odnosno visoka korelacija između uzastopnih uzoraka i uzastopnih okvira zvuka. Ovo svojstvo može se dobro iskoristiti diferencijalnim kodiranjem.

5. Koje su vremenske karakteristike govornog signala i kako se mogu iskoristiti za kodiranje?

Govor ima svojstvo da sadrži relativno puno tišine. Prosječno intervali govora traju od 0,8 do 1,2 sekunde, a između njih su intervali tišine koji prosječno traju 1-1,6s. Ako se na izlaz koda ne šalje ništa u intervalima tišine, može se uštedjeti do 40%.

Govorni signal je nestacionaran, tj. svojstva mu se bitno mijenjaju u vremenu. Međutim, ako se promatra vrlo mali vremenski interval govornog signala, unutar tog se intervala svojstva mijenjaju dovoljno malo da signal u tom intervalu možemo za potrebe kodiranja tretirati kao stacionaran. Stoga se često zvučni signal dijeli u intervale, tzv. okvire, obično od 20-30ms, te se svaki takav okvir zvuka kodira zasebno. Zbog toga se javlja algoritamsko kašnjenje kod prijenosa govora jer se paket može poslati tek kada je obrađen čitav okvir.

6. Objasnite fenomen maskiranja zvuka.

Maskiranje zvuka znači da zvuk na nekoj frekvenciji prikriva (maskira) slabije zvukove na toj i okolnim frekvencijama. U okolini maskirajućeg signala povećava se prag čujnosti- to je simultano maskiranje. Efekt maskiranja javlja se i malo prije i malo poslije trajanja maskirajućeg zvuka (oko 50ms): to je vremensko maskiranje.

Efekt maskiranja koristi se u kodiranju zvuka korištenjem potpojasnog kodiranja tako da se signal dijeli u frekvencijske pojaseve, te se svaki od njih kodira uz veću ili manju kvantizaciju.

7. Objasnite mjerila objektivnog i subjektivnog ocjenjivanja kvalitete zvuka.

Objektivna mjerila kvalitete zvuka podrazumijevaju izobličenje u odnosu na originalni signal i odnos signal/šum, no obzirom da ljudski sluh ne osjeća zvuk na isti način kao mjerni instrumenti, često su važnija subjektivna mjerila. Subjektivni se testovi rade na većem broju ispitanika te se rezultati statistički obrađuju. Za tu svrhu postoje standardizirani obrasci za ocjenu kvalitete zvuka. MOS traži od ispitanika ocjenu u apsolutnom smislu (1.loše, 2. slabo, 3.prihvatljivo, 4.dobro, 5.izvrsno), a DMOS relativno u odnosu na referentni zvuk (1.izraženo pogoršanje, jako smeta 2. podnošljivo, ali smeta 3. primjetno pogoršanje, malo smeta 4.čujno pogoršanje, ali ne smeta 5. nečujno pogoršanje). Za govor se koristi i procjena napora razumijevanja, a za razumljivost se mogu koristiti i testovi s parovima riječi koje slično zvuče. 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)

- metoda procjene subjektivne kvalitete govornih kodeka

- algorithm predviđa subjektivnu ocjenu kvalitete degradiranog uzorka govora

- izlaz iz algoritma je procijenjena vrijednost MOS

- Perceptual Evaluation of Audio Quality (PEAQ)
– algoritam za procjenu kvalitete audio

8. Objasnite kako zadani kriterij(i) brzine kodera/kvalitete/kašnjenja/otpornost na gubitke/kodiranje govora ili bio kakvog zvuka... utječe(u) na odabir za zadanu primjenu. Odaberite primjer mreže s poznatim karakteristikama (npr. fiksna mreža, javna pokretna mreža, Internet) i na njemu objasnite kako biste odabrali kodek za govornu uslugu.

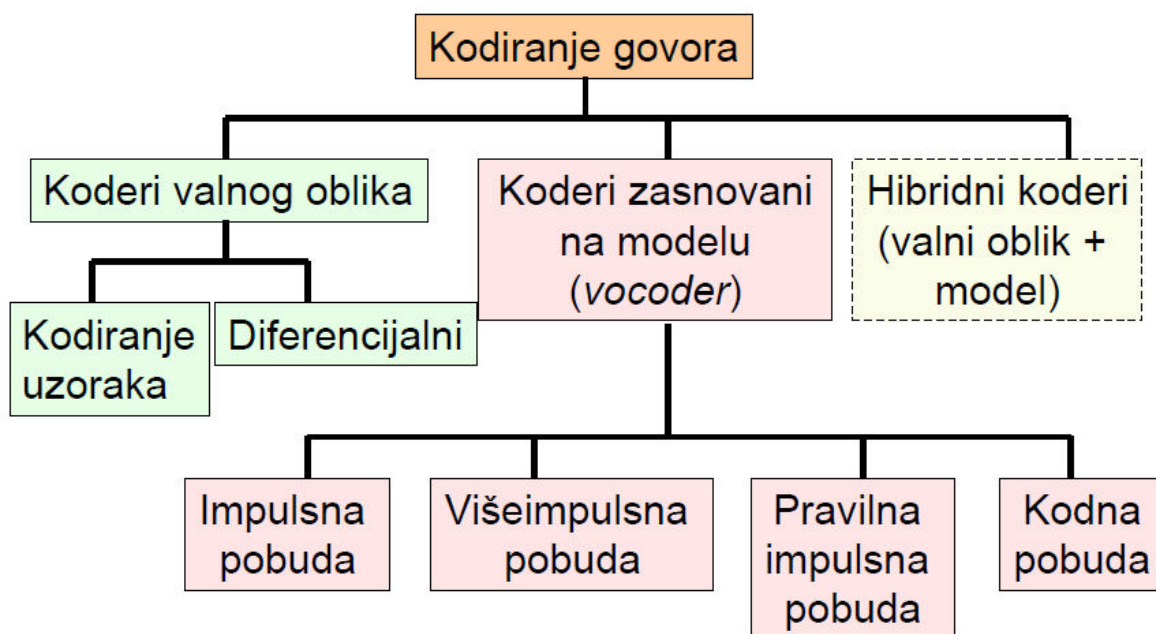
Brzina kodera (engl. **bit rate**) se mjeri u bit/s i izražava broj bitova potreban za kodiranje jedne sekunde zvučnog signala.

Za kvalitetu postoje objektivna mjerenja, kao što su izobličenje u odnosu na originalni signal, te odnos signal/šum. No, kao što smo već vidjeli, ljudski sluh ne osjeća zvuk na isti način kao mjerni instrumenti, te su često važnija subjektivna mjerenja.

Kašnjenje se javlja na koderu i na dekoderu, jer i jedan i drugi postupak traje neko vrijeme. U višemedijskoj aplikaciji svi mediji moraju biti sinkronizirani, a kašnjenje to otežava, odnosno svi se moraju ravnati prema najsporijem. Veće kašnjenje je isto tako neugodno pri dvosmjernoj telefonskoj komunikaciji, jer se teško postiže sinkronizacija među sugovornicima.

Obzirom na strukturu struje bitova, ona može biti manje ili više otporna na gubitke. Na primjer, diferencijalni koder, koji bi u svakom koraku slao samo razliku signala od prethodnog, akumulirao bi grešku nastalu gubitkom svakog osnovnog bloka, te se nikada ne bi oporavio. Otpornost na gubitke je u suprotnosti sa zahtjevom za male brzine: Sto je brzina manja, veći je utjecaj izgubljenog niza paketa, jer su informacije sabijenije i time osjetljivije. Bitan kriterij za kodere zvuka je jesu li primjenjivi samo na govor ili na općeniti zvuk. Napokon, složenost i cijena izvedbe su vrlo važni kriteriji u konkretnoj primjeni.

Kod primjerice fiksne telefonske mreže potrebno je odabrati jednostavan koder, s malim kašnjenjem i visokom kvalitetom.



9. Objasnite način rada PCM kodera. Navedite jedan primjer njegove primjene.

PCM ili impulsno-kodna modulacija je vrlo stari standard, već desetljećima u

Postupak je vrlo jednostavan i uključuje samo kvantizaciju po određenom nelinearnom zakonu. Kvantizacijom se postiže optimalan zapis signala, s obzirom na to da je za manje vrijednosti potrebno više razina kvantizacije. Tako kvantizirane vrijednosti se izravno zapisuju, tj. nema neke dodatne kompresije. Zvuk se uzorkuje s 8 kHz, a uzorci se kvantiziraju s 8 bitova (256 razina).

Prednosti PCM-a su: jednostavnost, visoka kvaliteta i malo kašnjenje.
Nedostaci PCM kodera su: 64 kbit/s je relativno velika brzina i nema mehanizma za kontrolu i ispravljanje pogrešaka (nije dobar kandidat za internetsku telefoniju).

KODER

DEKODER

x - ulazni signal d - signal razlike d' - kvantizirani signal razlike x' - rekonstruirani signal p - predviđeni signal	Q - fiksna ili adaptivna kvantizacija Q^{-1} - inverzna kvantizacija P - linearna predikcija (fiksni ili adaptivni koeficijenti) A - postupci adaptacije
--	---

U koderu je sadržan dekođer, te se računa razlika između signala kojeg bi dekođer predvidio i stvarnog signala; ova razlika se kvantizira i šalje dekođeru.

ADPCM nalazi primjenu u DECT (od engl. Digital European Cordless Telephony) normi za kućne bežične telefone, ISDN aplikacijama i telekonferencijskim aplikacijama.

12. Navedite najvažnije prednosti i nedostatke ADPCM kodera.

Prednosti:

1. nema algoritamskog kašnjenja obzirom da se signal ne dijeli na okvire
2. modemske i faks signali mogu se prenositi bez izobličenja (što nije slučaj s koderima zasnovanim na modelu).

Nedostaci:

1. Relativno velika brzina
2. Diferencijalni koderi imaju i problem osjetljivosti na pogreške. Pošto se šalju razlike signala, greška koja jednom nastane ne ispravlja se sljedećim signalom, makar je on ispravan, nego se akumulira. Stoga je u praktičnoj primjeni s vremena na vrijeme potrebno poslati referentni signal, tj. apsolutnu vrijednost, inače bi se akumulacijom greške u potpunosti izgubio signal.

13. Usporedite PCM koder i ADPCM koder s motrišta brzine i kvalitete (MOS).

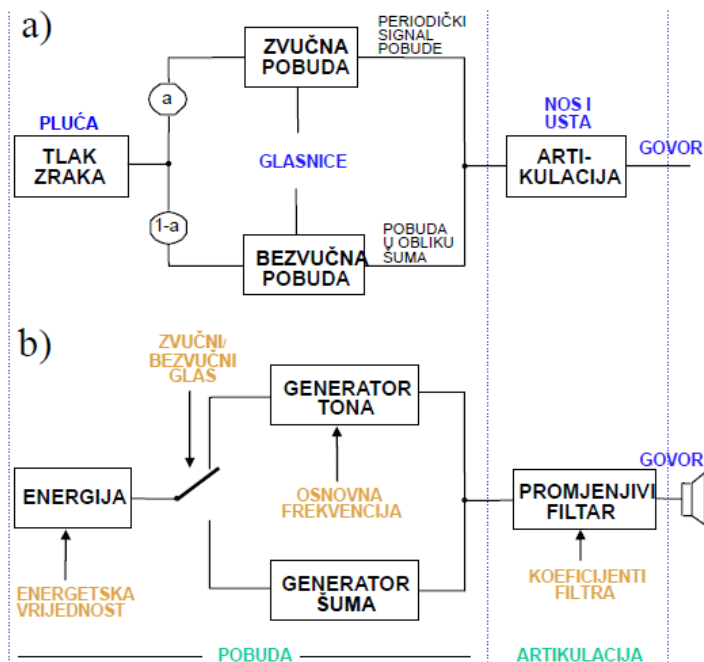
*Karakterizira ih dobra kvaliteta, no i relativno velike brzine prijenosa koje su potrebne.

14. Skicirajte i objasnite način rada osnovnog kodera govora zasnovanog na modelu.

a) blok dijagram
ljudskih
govornih organa

b) blok dijagram
dekodera
zasnovanog
na modelu

(LPC - Linear
Predictive Coder)

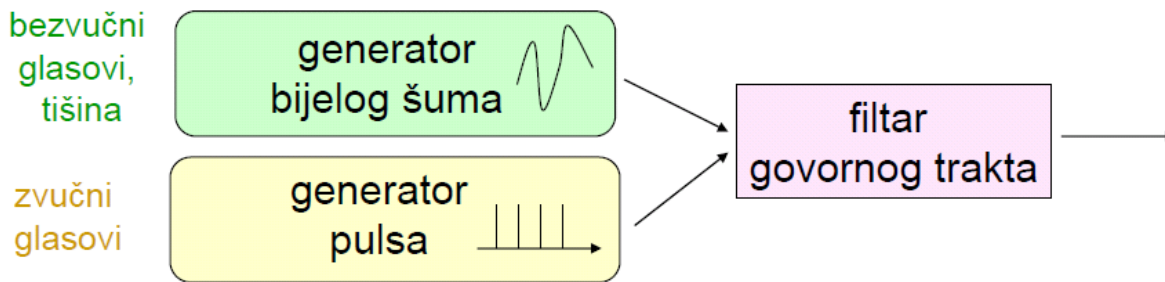


Koderi zasnovani na modelu koriste princip analize i sinteze. Umjesto da prenosimo same uzorke signala, koder i dekodeer imaju isti model za proizvodnju signala iz parametara modela, te se prenose samo ti parametri. Ukoliko je model dobro razrađen, te dobro simulira prirodnu proizvodnju signala, parametri modela bit će po obimu daleko (i do više redova veličine) kompaktniji od izvornog signala. Razlog popularnosti ove vrste kodera upravo za govor leži u tome što je model proizvodnje govora, tj. ljudskog govornog trakta, dobro poznat i razrađen. Signal govora se dijeli na okvire. Svaki okvir se analizira te se procjenjuju parametri modela za pojedini okvir. Ti se parametri prenose dekodeeru, koji ih ubacuje u model govornog trakta i simulacijom generira signal govora, koji je dakle potpuno sintetiziran. Sami uzorci signala se uopće ne prenose.

15. Što čini izlaznu struju bitova kodera valnog oblika zvuka, a što kodera zasnovanog na modelu? Što je potrebno za rekonstrukciju zvuka u jednom, a što u drugom slučaju?

Izlaznu struju kod valnog oblika cine uzorci signala, a kod model-kodera parametri modela. Pa kod valnog oblika za rekonstrukciju- dekodiranje signala (dakle u obrnutom smjeru od kodiranja se dekodira po istim principima-entropijsko, kvantizacija⁻¹, neznam), a kod modela- se dekodiraju parametri (koji su jednaki na obje strane), samo se kod dekodiranja simulira izlaz.

16. Kako se kodiranje govora zasnovano na modelu primjenjuje u osnovnom LPC-koderu?



U svakom vremenskom okviru od 22.5 ms određuju se sljedeći parametri:

frekvencija pobude (6 bita), jačina pobude (5 bita), zvučni i bezvučni glas (1 bit), koeficijenti filtra (42 bita za 10 koeficijenata).

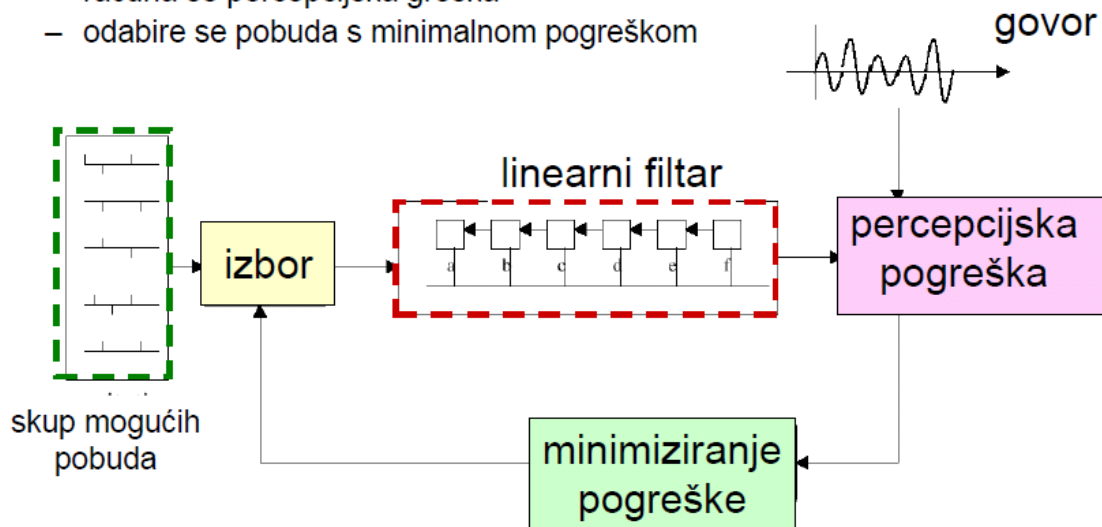
Ukupan broj bitova za jedan vremenski okvir je dakle 54. Pošto jedna sekunda sadrži $1000/22.5 = 44.444$ okvira, brzina je 2.4 kbit/s.

LPC dekodeer koristi generator šuma za bezvučne, te generator periodičnog pulsnog signala za zvučne glasove. Jedna od ovih pobuda, u ovisnosti o tome **da** li je trenutni glas zvučni ili bezvučni, dolazi na filtar govornog trakta te se time sintetizira govor. Centralni problem LPC kodera, kao i drugih kodera zasnovanih na modelu, je procjena parametara modela. Ukoliko je procjena loša, rezultat ne može biti dobar.

Najteže je odrediti frekvenciju pobude. To mora biti karakteristična frekvencija signala unutar okvira, no signal nije čist periodičan signal, a okvir je vrlo kratak, pa je analiza otežana. Stoga se koristi tzv. funkcija AMDF.

17. Kako se kodiranje govora zasnovano na modelu primjenjuje u osnovnom CELP-koderu? Navedite neku normu zasnovanu na načelima CELP koder.

- 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



Koder CELP (od engl. **Code Excited Linear Prediction**) je naprednija generacijakodera zasnovanih na modelu. Osnovni princip je isti, no uvedena su bitna poboljšanja te se postiže puno bolja kvaliteta zvuka. Osnovna je razlika u tome što se umjesto pulsnog signala za pobudu koriste raznovrsni signali pobude iz unaprijed predviđenog skupa mogućih pobuda. Time se postiže bolji model govora, jer ni pobuda koja dolazi iz glasnica nije čisti signal jedne frekvencije.

Dekoder ostaje vrlo sličan, osim što se za pobudu koristi jedna od mogućih pobuda iz skupa. Da bi to bilo moguće, koder mora dekoderu u svakom okviru uz sve ostale parametre poslati i kod za signal pobude koji u datom okviru treba koristiti. Ovaj kod je jednostavno indeks u tablici pobuda.

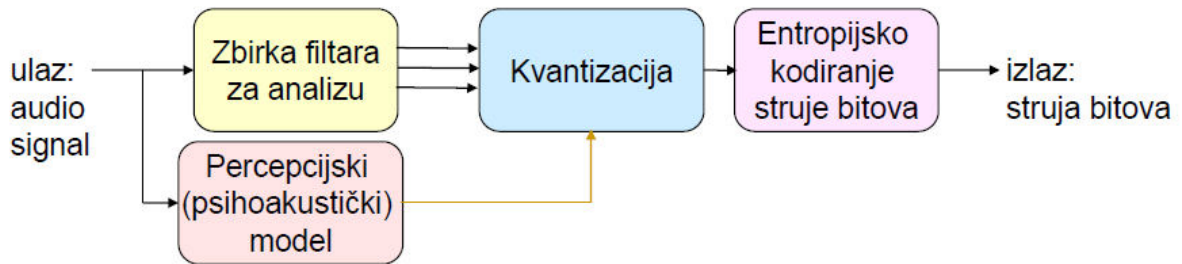
Sam koder mora, uz ostale parametre, odabrati i jednu od mogućih pobuda. Za to se koristi princip analize sintezom.

Primjer: preporuka ITU-T G.728Low Delay CELP (LD-CELP)

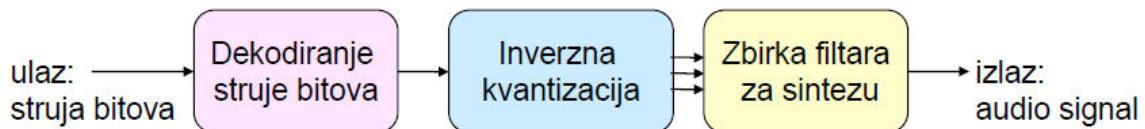
- 16 kbit/s, MOS 4, algoritamsko kašnjenje samo 0,625 ms

18. Skicirajte i objasnite način rada osnovnog percepcijskog koda zvuka. Navedite neku normu u kojoj se on primjenjuje.

Schema koda



Schema dekoda



Zbirkom filtera za analizu signal se dijeli u N potpojasnih signala. Svaki se potpojas posebno kvantizira. Percepcijski model određuje koliko se bitova dodjeljuje svakom potpojasu. Ušteda se ostvaruje dodjeljivanjem manjeg broja bitova onim potpojasevima za koje se vidi da će biti maskirani, tj. da ih ionako nećemo dobro čuti, pa na njih nema smisla trošiti puno bitova. Ova operacija ponavlja se za svaki okvir (osnovni segment) audio signala.

Prilikom dekodiranja se koristi zbirka filtera za sintezu kojima se potpojasni signali ponovo slože u jedinstveni signal.

Primjer: MPEG (Moving Picture Expert Group): MPEG-1, MPEG-2, MPEG-4

19. Zašto PCM koder koji se primjenjuje u PSTN mreži nije dobar izbor za mrežu GSM?

Mozda jer ima preveliku brzinu kodiranja...

20. Zašto CELP koder koji se primjenjuje u mreži GSM ne bi bio dobar izbor za podršku fax-usluge?

Jer ne bi prenio podatke kak spada... zbog moguće pogreske u izboru pobude ili percipiranja pogreske?? (sta sam sad napisao majke ti)... wtf kak da ovo znamo... nek neki lumen odgovori...

Kodiranje nepomične slike

1. Ukratko opišite svojstva i ograničenja ljudske percepcije slike koja su važna za kodiranje nepomične slike.

Vidljiva svjetlost pokazuje dvojna svojstva: svojstva elektromagnetskog vala i svojstva čestica, koje se nazivaju fotoni. Za učinkovito kodiranje važna su valna svojstva, odnosno ljudska percepcija tih svojstava. Ljudsko zapažanje svjetlosti može se opisati trima komponentama direktno vezanim za valna svojstva svjetlosti: Obojenost, zasićenost i svjetlina (HSB model).

2. Ukratko opišite svojstva modela boja RGB i CMY.

RGB (crvena, zelena, plava) - zbrojive boje, CMY (svijetlomodra, ružičasta i žuta) – oduzimljive boje. Ove se tri komponente kombiniraju uz pomoć težinskih faktora koji se obično izražavaju brojevima u rasponu 0 – 255 ili realnim brojevima 0 – 1. Tako se bilo koja boja može prikazati kao trodimenzionalni vektor ovih težina. RGB model služi za prikaz svjetline i zbrajanjem se dobiva sve veća svjetlina, a CMY model služi za dobivanje pigmenta (boje) oduzimajući dio spektra iz bijele svjetlosti.

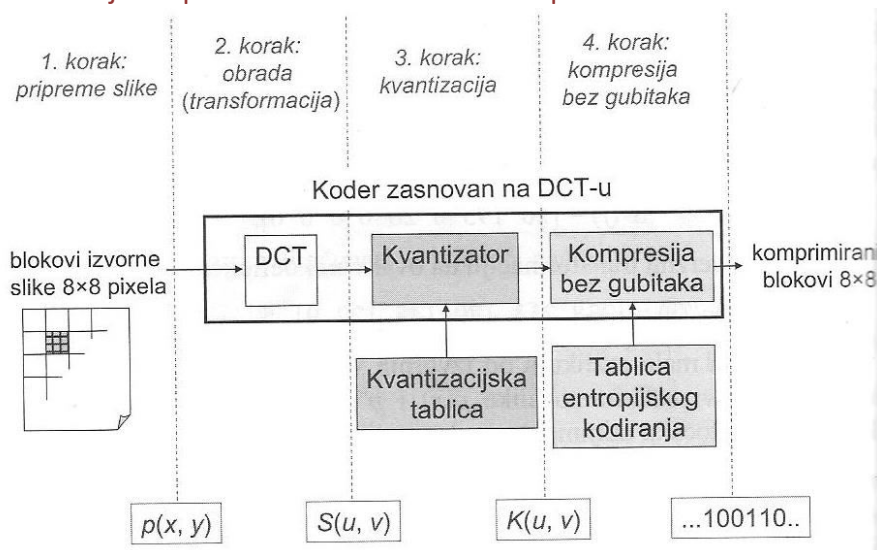
3. Za prikaz slike u boji u računalu koristi se model boje RGB i dubina slike 24 bita. Ako je rezolucija slike $x \times y$ pixela, kolika je veličina zapisa nekomprimirane slike?

Veličina je: $x \times y \times 24$

4. Kako se načela kodiranja s gubicima mogu primijeniti na kodiranje nepomične slike? Koji su kriteriji prihvatljivih "gubitaka" u kodiranoj slici?

Kodiranje s gubicima može se koristiti karakteristikama ljudskog vida i postići puno veći omjer kompresije od kodiranja bez gubitaka, a da potom pogreška ostane neprimjetna ili vrlo teško primjetna za ljudsko oko što je prihvatljivo.

5. Skicirajte i opišite model JPEG koda nepomične slike.



1. Slika se dijeli na blokove, te se nadopunjavaju blokovi tako da dobivamo niz blokova velicine 8×8 tocaka

2. Svaki blok slike transformira se u blok od 64 DCT (diskretna kosinusna transf.) koeficijenta. Zasto DCT? - jer se bilo koja slika može izraziti kao težinski zbroj osnovnih DCT funkcija. Prvi koeficijent u DCT matrici opisuje srednju vrijednost slike i to je DC komponenta, a svi ostali su AC komponente. Više frekvencije se nalaze dolje desno u toj matrici, a niže gore i lijevo.

3. Kvantizacija se vrši tako da se svaki koeficijent dijeli s kvantizacijskim faktorom te zaokružuje na cijeli broj. Dobiva se kvantizacijska matrica.

4. Slijedi kompresija bez gubitaka. Diferencijalno kodiranje se koristi za DC blokove (jer su susjedni DC blokovi slični), a AC komponente se slože linearnim cik-cak redoslijedom i grupiraju se nule (slijednim kodiranjem), a potom slijedi entropijsko kodiranje - najčešće Huffmanovo. Može i aritmetičko, ali je presloženo pa nije za praksu...

6. Koji je tipični omjer kompresije koji se postiže pomoću JPEG koda nepomične slike uz neznatan gubitak kvalitete?

JPEG postiže odlično kvalitetu slike do 0.25 bit/pikselu. Dakle 1:4 je omjer :D

7. Kako podjela na blokove u prvom koraku JPEG koda utječe na kasnije korake kodiranja?

Nakon podjele izvorne slike na blokove slijedi transformacija svakog bloka u blok od 64 DCT koeficijenta S u drugom koraku koji se potom u trećem i četvrtom koraku kvantiziraju i komprimiraju.

8. Što se događa u koraku transformacije u JPEG koderu? Što se transformira u što? Je li DCT transformacija reverzibilna?

U drugom se koraku svaki blok slike transformira u blok od 64 DCT koeficijenata S . DCT transformacija vrši preslikavanje niza vrijednosti piksela u niz koeficijenata težine osnovnih DCT blokova. DCT je reverzibilna u potpunosti.

9. Koji je smisao transformiranja blokova slike pomoću DCT u JPEG koderu? Koja svojstva slike se mogu iskoristiti u frekvencijskoj domeni?

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

Frekvencijske komponente na nižim frekvencijama su izraženije od onih na višim frekvencijama. Visoke frekvencije izražavaju zaista sitne detalje u slici. Doprinos članova na višim frekvencijama je mali kada su razlike susjednih piksela relativno male, dakle kada slika ne sadrži puno detalja.

10. Što se događa u bloku kvantizacije u JPEG koderu? Koji je očekivani rezultat?

DCT koeficijenti se kvantiziraju zaokruživanjem na cjelobrojnu vrijednost omjera matrice koeficijenata S i kvantizacijske matrice Q ; nastaje kvantizirana DCT matrica K . Q mora biti tako izabrana da kvantizacija rezultira visokom kompresijom, ali bez primjetnog gubitka kvalitete.

11. Kako se u bloku kompresije bez gubitaka unutar JPEG koda kodira "DC komponenta" (koeficijent $K(0, 0)$) matrice kvantiziranih koeficijenata K , a kako ostale, "AC komponente" (koeficijenti)? Što je specifično za redoslijed očitavanja AC komponenta?

Za kodiranje „DC komponente“ koristi se diferencijalno kodiranje. „AC komponente“ se kodira tako da se preostale vrijednosti u matrici K slažu po linearnom cik-cak redoslijedu i na taj način se grupiraju nule. To se može iskoristiti varijantom slijednog kodiranja, na taj način da se umjesti čitavog niza nula na kraj doda poseban kod za kraj bloka (EOB=end of block). Nakon ovakvog zapisa svih blokova u slici slijedi slijedi entropijsko kodiranje.

12. U kojem koraku JPEG koda se obavlja entropijsko kodiranje? Koje od metoda entropijskog kodiranja se najčešće primjenjuju u praksi?

Entropijsko kodiranje se obavlja u koraku 4: kompresija bez gubitaka. Najčešće se upotrebljava Huffmanovo kodiranje. Norma JPEG predviđa i mogućnost aritmetičkog kodiranja, no zbog veće se složenosti u praksi to rijetko koristi.

13. Znamo da je JPEG koder s gubicima. Gdje dolazi do gubitka informacije u postupku kodiranja i što se gubi?

Do gubitaka dolazi u koracima kvantizacije i kompresije bez gubitaka. Gube se informacije (nule) koje nisu bitne za prikaz slike.

14. Kako se u JPEG koderu može regulirati omjer kompresije? Koji je tipični omjer (ili red veličine) koji se može postići?

Odnos kvalitete i kompresije se može regulirati pomoću kvantizacijske matrice. Odlična kvaliteta slike do 0.25 bit/piksel.

15. Koje su posljedice po subjektivnu kvalitetu slike kodirane JPEG koderom ako se prilikom kodiranja odabere previsok omjer kompresije? Kako se može očitovati gubitak kvalitete slike? Pri većim faktorima kompresije blokovi u slici postaju jasno vidljivi.

16. O čemu ovisi "najviši mogući" omjer kompresije za zadanu sliku koji se može postići pomoću JPEG koda bez značajnijeg gubitka subjektivne kvalitete slike?

Ovisi o sadržaju slike... Znači ako ima jako malo visokih frekvencija (naglih prijelaza na slici) onda će i kompresija biti veća, a subjektivna kvaliteta se neće pogoršati tako jako kao što bi uz isti omjer kompresije bilo za neku sliku sa više visokih freq.

17. Kako se očituje degradacija kvalitete slike kodirane JPEG koderom ako se slika više puta (npr. 10 puta) uzastopce kodira uz isti zadani omjer kompresije (npr. 1:20)?

Degradacija kvalitete slike kodirane JPEG koderom očituje se pojavom blok efekta. Blokovi na koje je slika podijeljena pod utjecajem prevelike kompresije razmiču i postaju vidljive granice između njih.

18. Web dizajner stranica o turističkoj ponudi Zadra odlučio je na stranicu staviti panoramsku fotografiju Zadra. Pri odabiru fotografije, dizajner je skenirao odabranu fotografiju veličine 25 x 10 cm u boji u rezoluciji 300dpi (uzmite 1 inch = 2.5cm). Program za skeniranje automatski sprema sliku u nekomprimiranom formatu. Kolika je veličina nekomprimirane slike?

$(25/2.5) * (10/2.5) * 300$

19. Za primjenu fotografije iz prethodnog primjera na web stranici, koji bi format bio s motrišta veličine datoteke bio bolji za pohranu slike, gif ili jpeg? Objasnite.

S motrišta veličine, GIF je primjereniji (zbog 8-bitne boje, naprema 24-bitne kod JPEGa).

20. Radeći backup web-sjedišta, dizajner je pokušao smanjiti veličinu prostora za pohranu komprimirajući originalnu sliku i sliku prilagođenu za web u format ZIP. Rezultirajuća datoteka, međutim, ispada veća od originalne. Zašto?

....brain drain... 3:18 je nemogu razmisljat... jel mozda zato sto se vise puta entropijski kodirala?

Kodiranje videa

1. Ukratko opišite svojstva i ograničenja ljudske percepcije slike koja su važna za kodiranje videa.

Video se sastoji od niza nepomicnih slika(okvira), prikazanih dovoljno brzo jedna za drugom da se dobije dojam neprekinutog gibanja. Frekvencija promjene slike (frame rate) izražena je brojem okvira u sekundi (fps). Opazamo gibanje kao neprekinuto ako je freq promjene slike barem 15 fps. Za video se najcesce koristi 25 ili 30 fps. Za 3D 60 ili 75 fps.

2. Objasnite kako zadani kriterij(i) kašnjenja i složenosti utječe(u) na odabir kodera videa s motrišta simetrije, odnosno asimetrije, zadane primjene (primjeri: DVD film, videokonferencija).

Video telefonija/konferencija su simetricne primjene, tj svaka strana ima koder i dekodek. Stoga su koder i dekodek podjednake složenosti. Za ovu primjenu izuzetno je vazno smanjiti kašnjenje. Digitalna TV i pohrana filma su asimetricne primjene. Kodira se jednom a dekodira puno puta. Isplati se dakle razvoj složenog i skupog kodera, dok dekodek treba biti sto jednostavniji i jeftiniji.

3. Ukratko opišite modele boja YUV i YIQ. Zašto se odvajaju komponente boje i svjetline?

Komponente boje i svjetline se odvajaju jer su prvo bili TV crno-bijeli, a onda u boji, al se morala zadržat kompatibilnost, jer nisu svi imali pare za onaj u boji...

YUV model dolazi iz PAL norme za analogni video, a YIQ iz NTSC norme. U oba modela Y komponenta izražava svjetlinu (luminance...a di je epsilon tu?) i može se iz RGB modela izracunati teziškim zbrajanjem komponentata. Ostale 2 komponente izražavaju boju i takodjer se racunaju iz RGB.

4. Objasnite primjenu metode poduzorkovanja na video (4:2:2, 4:2:0).

Posto je ljudsko oko osjetljivije na svjetlinu, boju možemo uzrokovati manjom rezolucijom. Rezolucija se obicno smanjuje 2 puta(horizontalno). ili 4 puta(horizontalno i vertikalno). Izvorno se svaki piksel sastoji od 3 komponente: svjetline Y i 2 komponente boje U i V. Model 4:4:4 je model sa punim uzorkovanjem. Kod modela 4:2:2 se uzimaju svjetline za svaki pixel, ali se poduzorkovanjem odbacuju vrijednosti boje za svaki drugi pixel u svakom redu. Kod modela 4:2:0 od svakog bloka 2x2 pixela se uzima samo po jedna vrijednost komponentata boje U i V

5. Za prikaz videa u TV kvaliteti (npr. PAL 720x576) koristi se model boje YUV i poduzorkovanje boje 4:2:0, uz brzinu 30 fps. Kolika je veličina zapisa nekomprimiranog videa? ma racunaj si sam....

$$30 * (720 * 576 + 180 * 144 + 180 * 144) * 8 \text{ bit} = 13996,8 \text{ kb/s}$$

6. Kako se načela kodiranja s gubicima, u smislu uklanjanja redundancije, mogu primijeniti na kodiranje videa?

Postoji prostorna i vremenska redundancija. Prostorna se javlja unutar jednog okvira slike, te se uklanja kao i kod nepomicne slike- najcesce se koristi transformacijsko kodiranje. Vremenska se pojavljuje izmedju susjednih okvira u videu, a uklanja se diferencijalnim kodiranjem. No, pomaci u slici od jednog do drugog okvira cine diferencijalno kodiranje neefikasnim. To se rješava kompenzacijom gibanja.

7. Zašto se diferencijalnim kodiranjem između uzastopnih okvira (slika) u videu uglavnom ne postiže dobra kompresija? Objasnite na proizvoljnom primjeru.

Ljudski vid uocava ocite slicnosti, a razliku primjecujemo samo malo pazljivijim pogledom (ona slika di se trokut prebaci s desne na lijevu stranu smajlica). Dok ljudsko oko uocava ovu ocitu slicnost, diferencijalni koder jednostavno oduzima vrijednost boje pixel po pixel. Ovakvom naivnom obradom dobili bismo signal razlike u kojemu ima vise informacije nego u originalnoj slici. Naci- neisplati se

8. Objasnite načelo kompenzacije gibanja za primjenu kodiranja videa.

Za svaki blok u slici trazi se najsljedniji blok u prethodnoj slici. Razlika položaja (u pixelima) između ova dva blok naziva se vektor pomaka. Vektori pomaka računaju se za sve blokove u slici. Dobiveni vektori pomaka koriste se za pomicanje blokova prethodne slike. Tako se dobiva kompenzirana slika i na kompenziranoj slici se vrši diferencijalno kodiranje.

9. Što je prednost koda videa s kompenzacijom gibanja u odnosu na koder koji bi kodirao sliku po sliku (npr. kao Motion JPEG)?

Bolja kvaliteta slike...? .wtf...?

10. Skicirajte i opišite način rada hibridnog koda videa (s kompenzacijom gibanja).

11. Koji dio hibridnog koda videa odgovara koderu nepomične slike?

Transformacijsko kodiranje (DTC, kvantizacija, entropijsko)

12. Što se događa u bloku transformacije i kvantizacije u hibridnom koderu videa?

A šta bi se događalo... isto sta i kod JPEG-a... nakon kvantizacije gubi se kvaliteta slike...valjda.

13. Što se događa u bloku predikcije u hibridnom koderu videa?

Vrši se predviđanje slike na bazi dekodirane slike iz prethodnog vremenskog poraka(prediktor), računa se razlika između izvorne i predviđene slike, te se ova razlika kvantizira i entropijski kodira.

14. Čemu služi veza između kompenzacije gibanja i predikcije u shemi hibridnog koda videa?

Vektori pomaka se računaju procjenom gibanja te se prosljeđuju prediktoru, koji ih koristi kako bi izvršio odgovarajuću kompenzaciju gibanja, tj pomake na prethodnoj slici, te na osnovu tako dobivene kompenzirane slike konstruirao predviđenu sliku.

15. U kojem koraku hibridnog koda videa se obavlja entropijsko kodiranje?

Zadnjem, prije nego se puste u struju bitova... i guess...

16. Zašto hibridni koder videa u sebi sadrži i dekode? Što bi bilo da toga nema?

Zato da ne bi doslo do akumuliranja pogreske kod diferencijalnog kodiranja. Bila bi losa slika.

17. Objasnite razliku između I, P i B-okvira dobivene hibridnim kodiranjem videa s obzirom na sadržaj i veličinu zapisa pojedinog okvira.

I- kodirani kao samostalna slika, P- kodirani korištenjem predikcije i kompenzacije gibanja u odnosu na prethodni okvir, B- kodirani korištenjem predikcije u odnosu na prethodni i sljedeći okvir. $I > P > B$ (po broju bitova). Tipični omjeri: $I = 3P = 5B$

18. Zašto se kodirani okviri moraju dekodeu slati drugačije od prirodnog redoslijeda?

Prvo se uvijek salju I ili P okvir prije B okvira, jer se B okvir nemože dekodirati prije P okvira, a P nemože prije I okvira.

19. U nizu okvira za prikaz, pojavljuju se I, P i B okviri u sljedećem redoslijedu: I1 B2 B3 P1 B4 B5 P2 B6 B7 I2 B8 B9 P3 ... itd. Ispišite redosljed kojim okvire treba slati na dekode. Ako je za slanje i dekodiranje svakog okvira potrebno vrijeme t , koliko je algoritamsko kašnjenje u ovom primjeru?

IPBBPBBIBBPBB :D

$t \cdot 13$?

20. Kako veličina i preciznost vektora pomaka utječu na kvalitetu i složenost kompenzacije gibanja?

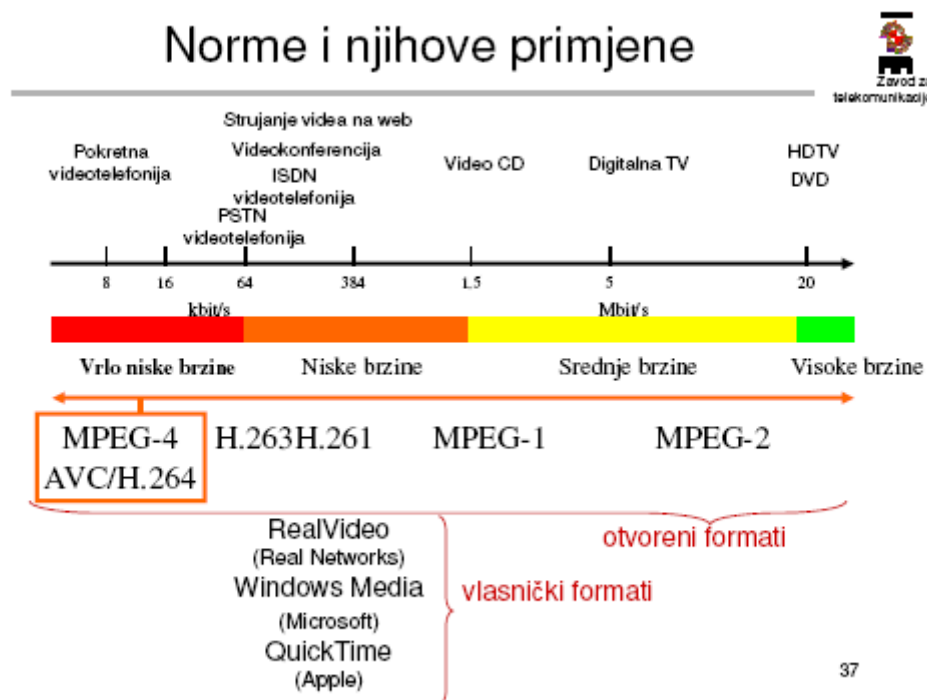
Velicina vektora pomaka određuje površinu pretrage za sličnim blokom pri kompenzaciji gibanja. Veći dozvoljeni vektor pomaka, tj veća površina pretraživanja daje veću vjerojatnost pronalazenja sličnog bloka te u principu rezultira boljom kompenzacijom gibanja. Ako je dozvoljena velicina vektora pomaka premala, moguće je da se uslijed brzog gibanja blok pomakne u odnosu na prethodnu sliku više nego što se može prikazati vektorom pomaka, te se takav blok ne može koristiti u kompenzaciji gibanja. Većim se vektorom pomaka ovaj

problem ibjegava. Međutim, veći dozvoljeni vektor pomaka znači i veću površinu pretrage slike. Pretraga slike je računalno intenzivna operacija, te se povećavanjem površine pretrage postavlja problem zahtjevnosti izvođenja. Ujedno postoje i granice iznad kojih se ne isplati povećavati vektor pomaka i površinu pretrage. U praksi se koristi vektor pomaka 16 - 32 piksela

21. Zašto se uvodi hijerarhija i dodatna zaglavlja prilikom pakiranja videa u struju bitova, na primjer kod normi H.261 i H.264?

Struja bitova formatirana je hijerarhijski prema podjeli slike na jedinice. Prije svake jedinice podataka dolazi zaglavlje s podacima koji vrijede za cijelu jedinicu. Npr. u zaglavlju slike se naznacza radi li se od I ili P okvira. U zaglavlju skupine blokova daje se kvantizacijski faktor za cijelu skupinu blokova, no on se još može podesavati i u zaglavlju makroblokova. Ovakvom strukturom postiže se efikasno spremanje slike u slijed bitova

22. Navedite primjer norme za kodiranje vide niske/srednje/visoke brzine i neku od njenih primjena.



Sinkronizacija medija

1. Što je sinkronizacija unutar medijskog objekta? Objasnite na primjeru.

Vremensko usklađivanje unutar struje vremenski ovisnog (kontinuiranog) medija. Primjer: okviri unutar videa.

2. Što se sinkronizacija između medijskih objekata? Objasnite na primjeru.

Sinkronizacija jedinica između različitih medijskih objekata (engl. inter-stream synchronization). Primjer: vremensko usklađivanje unutar višemedijske prezentacije

3. Koja je razlika između sinkronizacije uživo i umjetne sinkronizacije s obzirom na složenost specifikacije sinkronizacije?

Sinkronizacija uživo- rekonstrukcija vremenskih odnosa uspostavljenih na izvoru (prilikom snimanja medija) npr. televizijski prijenos. Umjetna- vremenski odnosi između objekata ne postoje "sami po sebi", već se uvode eksplicitno, putem specifikacije, npr. 3D animirani lik i govor, prezentacija uz slideove

4. Što uključuje specifikacija sinkronizacije?

Definiranje vremenskih odnosa u modelu sinkronizacije.

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

5. Kakva je to osna specifikacija sinkronizacije? Koje su prednosti/nedostaci?

Na vremenskoj osi se definiraju točke pokretanja i 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

6. Kakva je to intervalna specifikacija sinkronizacije? Koje su prednosti/nedostaci?

Definira se trajanje i usklađenost(međusobni odnos) vremenskih intervala prikaza medija.

- 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

7. Dizajner višemedijske prezentacije treba napisati specifikaciju sinkronizacije za sljedeće medijske objekte: audio1, video1, animacija1, audio2, pri čemu je zadana duljina trajanja svih objekata i sljedeći scenarij: paralelni prikaz audio1 i video1, te nakon toga paralelni prikaz animacija1 i audio2. Koja vrsta specifikacije sinkronizacije, osna ili intervalna, je primjenjiva na zadani slučaj?

Osna.

8. Dizajner višemedijske prezentacije treba napisati specifikaciju sinkronizacije za sljedeće medijske objekte: audio1, video1, korisnikov_ulaz1, audio2, pri čemu je zadana duljina trajanja svih audio i video objekata, dok trajanje objekta korisnikov_ulaz1 nije unaprijed poznato. Scenarij čini paralelni prikaz audio1 i video1, nakon čega se čeka unos od korisnika, nakon čega slijedi animacija. Koja vrsta specifikacije sinkronizacije, osna ili intervalna, je primjenjiva na zadani slučaj?

Intervalna.

9. Koja su mjerila kvalitete sinkronizacije 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.

10. Koja su mjerila kvalitete sinkronizacije za više medijskih objekata međusobno povezanih (u prikazu), npr. audio i video prikaz spikera na TV?

–kvaliteta usluge ovisi o uspješnosti usklađivanja međusobnog odnosa medija

–npr. razilaženje +/-80 ms

11. Koje bi se mjerilo moglo primijeniti za kvalitetu sinkronizacije usana i videa? Koje su tipične vrijednosti?

Sinkronizacija između medijskih objekata.

12. Kako percepcija kvalitete sinkronizacije usana ovisi o udaljenosti i prikazu lica govornika?

Percepcija sinkronizacije ovisi i o pogledu: nije svejedno promatramo li samo glavu odn. lice, glavu i ramena, ili cijelu osobu. Ako je blizi kadar jasnije ćemo vidjeti ako je sinkronizacija lošija.

13. Pretpostavimo da se strujanjem u mreži audio i video javljanja uživo TV-dopisnika prenose u dva odvojena toka koji se sinkroniziraju na prijamniku. Aplikacija na prijamniku ima međuspremnik za audio, odnosno video, u koje može pohraniti po 80 ms svakog medija. Ako znamo da su tipične vrijednosti razilaženja za audio i video kod sinkronizacije usana +/-80 ms, koliko smije biti maksimalno kolebanje kašnjenja u mreži kako bi reprodukcija tekla glatko? (Zanemarite sva ostala kašnjenja u sustavu.)

160ms

14. Pretpostavimo da se strujanjem u mreži audio i video javljanja uživo TV-dopisnika prenose u dva odvojena toka koji se sinkroniziraju na prijamniku. Kašnjenje u mreži je 200 ms, a kolebanje kašnjenja 10%. Aplikacija na prijamniku ima međuspremnik za audio, odnosno video. Ako znamo da su tipične vrijednosti razilaženja za audio i video kod sinkronizacije usana +/-80 ms, koliki minimalno treba biti međuspremnik kako bi bi reprodukcija tekla glatko? (Zanemarite sva ostala kašnjenja u sustavu.)

$200 + 0,01 \cdot 200 = 202 \text{ ms}$? ovo nema smisla... kolega veli 160ms isto... :/

15. (Pitanja vezana za specifikaciju sinkronizacije jezikom SMIL se uglavnom zadaju uz pomodni primjer.)

HAhahahahahaHAAHaa... -,-'

Raspodijeljeni sustavi i modeli komunikacije u raspodijeljenom okruženju

1. Definirajte raspodijeljeni sustav te navedite primjere takvih sustava.

Raspodijeljeni sustav je skup neovisnih računala povezanih komunikacijskom mrežom koji djeluje kao jedinstveni sustav, te krajnjem korisniku pruža definiranu (visemedijsku) uslugu. Sa stajališta korisnika- riječ je o jedinstvenom sustavu. Npr neki zahtjevniji web poslužitelji...

2. Objasnite arhitekturu klijent-poslužitelj te objasnite razlike između poslužiteljskih i klijentskih procesa.

Arhitektura klijent-poslužitelj je prevladavajuća arhitektura raspodijeljenih sustava i raširena arhitektura za izgradnju višemedijskih usluga. Često se naziva i modelom komunikacije na načelu "zahtjev-odgovor" (engl. request-reply) s obzirom da klijent šalje zahtjev (zahtjeva uslugu), a poslužitelj nakon obrade zahtjeva šalje odgovor klijentu. U ovoj arhitekturi postoji jedan poslužitelj i grupa klijenata koji koriste uslugu poslužitelja.

Na strani poslužitelja postoji proces (proces) koji nudi određenu uslugu, prima i obrađuje dolazne zahtjeve klijentskih procesa, te po potrebi šalje odgovor klijentima. Poslužiteljski

proces je povezan s adresom koja ga jednoznačno identificira, a mora biti poznata klijentskim procesima. Klijentski proces zahtijeva neku uslugu od strane poslužitelja, šalje mu zahtjev i po potrebi očekuje odgovor.

3. Navedite primjere višemedijskih usluga koje se temelje na arhitekturi klijent-poslužitelj i identificirajte potencijalne probleme takve arhitekture.

Kojekakvi webovski serveri... Preopterećenje servera sa previse klijenata/zahtjeva bi moglo biti problem..

4. Objasnite pojam međuprocenjske komunikacije te navedite njena obilježja.

U raspodijeljenom sustavu međusobno komuniciraju procesi što se naziva međuprocenjskom komunikacijom koja se temelji na razmjeni podataka među autonomnim raspodijeljenim procesima. Stoga je komunikacija među udaljenim procesima, tj. razmjena podataka među procesima koji se izvode na različitim računalima osnova svakog raspodijeljenog sustava. Podaci se "pakiraju" u poruku i prenose mrežom.

Međuprocenjska komunikacija u svome elementarnom obliku izgleda ovako: Kada proces a koji se izvodi na računalu A želi komunicirati s procesom b koji se izvodi na računalu B, tada proces a kreira poruku u svome adresnom prostoru i poziva funkciju (engl. system call) kojom operacijski sustav računala A preko mreže šalje poruku procesu b na računalu B. Proces a je pošiljalac, a proces b primatelj poruke. Kako bi procesi a i b mogli komunicirati, moraju istovjetno interpretirati niz bitova koji čine prenesenu poruku.

5. Navedite i objasnite obilježja raspodijeljenih sustava.

Paralelne aktivnosti- autonomne komponente sustava istodobno izvode aktivnosti

Komunikacija razmjenom poruka- bez zajedničke memorije

Dijeljenje sredstava- zajedničkim sredstvima pristupa više komponentata

Nema globalnog stanja- niti jedan proces ne zna globalno stanje sustava

Nema globalnog vremenskog takta- ograničena mogućnost vremenskog uskladjivanja

6. Objasnite svojstvo otvorenosti raspodijeljenog sustava. Navedite primjer takvog sustava.

Otvoreni (raspodijeljeni) sustav pruža usluge sukladno normiranim pravilima te definiranoj sintaksi i semantici. Definicija usluge pomoću sučelja. Primjer jezika za opis sučelja: Interface Definition Language (IDL)- COBRA; Web Services Definition Language (WSDL)

7. Objasnite svojstvo skalabilnosti raspodijeljenog sustava. Navedite primjer takvog sustava.

Sposobnost sustava za prilagodbu uslijed porasta broja korisnika/korisničkih zahtjeva ili rasta samog sustava. Problemi neskalabilnih rješenja: centralizirana usluga, podatci, algoritmi.

Primjer skalabilnih sustava: WWW, DNS, P2P

8. Objasnite svojstvo transparentnosti raspodijeljenog sustava. Navedite primjere.

Prikrivanje odabranih značajki raspodijeljenog sustava (npr. prikrivanje činjenice da su procesi i resursi fizički raspodijeljeni na više računala). Utječe na složenost, performanse i troškove sustava. Primjer: transparentnost lokacije: prikrivanje stvarne lokacije resursa (npr. naziv mrežne usluge je neovisan o lokaciji resursa koji realizira uslugu); transparentnost migracije: promjena lokacije resursa se ne odražava na krajnjeg korisnika; transparentnost replikacije: skrivanje činjenice da postoji više kopija određenog resursa

9. Što je programski posrednički sloj i koja je uloga takvog sloja u raspodijeljenim aplikacijama?

Tzv. „medjuoprema“ (middleware)- pruža generičke usluge za jednostavniji razvoj raspodijeljenih usluga. Programski posrednički sloj koristi usluge transportnog sloja radi razvoja generičkih usluga koje omogućuju pojednostavljenu implementaciju raznih raspodijeljenih usluga. Generičke usluge programskog posredničkog sloja se razlikuju po funkcionalnosti, kompleksnosti i svojstvima.

U internetskom modelu- na aplikacijskom sloju- između transportnog i aplikacijskog sloja

10. Što je programski posrednički sloj za komunikaciju raspodijeljenih procesa? Navedite primjere.

Programski posrednički sloj za komunikaciju raspodijeljenih procesa je vrsta posredničkog sloja nužna za razvoj raspodijeljenih usluga. Implementira modele za komunikaciju udaljenih procesa. Skriva kompleksnost i heterogenost transportnog sloja s ciljem pojednostavljene razmjene podataka među raspodijeljenim procesima. Osnova svakog raspodijeljenog sustava omogućuje jednostavniji razvoj raspodijeljenih usluga... ovo je gore od trgovačkog.

11. Objasnite razliku između perzistentne i tranzijentne komunikacije. Navedite primjere.

Komunikacija je perzistentna ako je poslana poruka pohranjena u sustavu do trenutka isporuke primatelju. Perzistentna komunikacija garantira isporuku poruke premda primatelj nije aktivan u trenutku nastanka i slanja poruke. Pretpostavlja postojanje posrednika koji pohranjuje poruku do njene isporuke.

Tranzijentna komunikacija garantira isporuku poruke samo ako su i pošiljalac i primatelj istovremeno aktivni u trenutku slanja poruke. Npr. transportni sloj (TCP, UDP) nudi tranzijentnu komunikaciju.

12. Objasnite razliku između sinkrone i asinkrone komunikacije. Navedite primjere.

Sinkrona komunikacija- Pošiljalac može biti blokiran do primitka potvrde o tome da je:

- a)poruka pohranjena u dolazni spremnik primatelja,
- b)primatelj primio poruku,
- c)primatelj obradio poruku (zahtjev) te šalje odgovor.

Asinkrona komunikacija- Poruka se pohranjuje u izlazni spremnik pošiljalca i omogućuje mu nastavak procesiranja odmah nakon slanja poruke

13. Objasnite razliku između komunikacije na načelu push i pull. Navedite primjere.

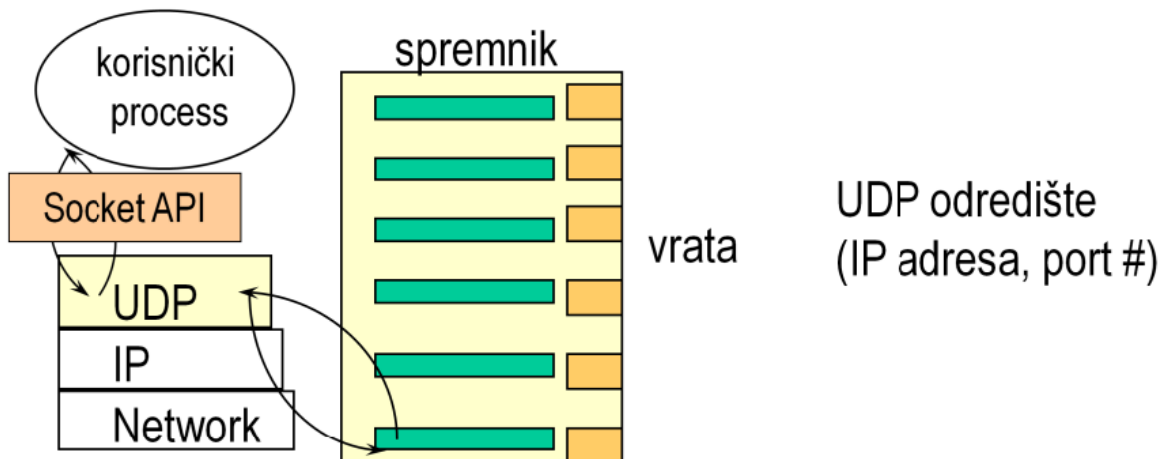
Komunikacija na načelu pull: klijent eksplicitno šalje zahtjev poslužitelju nakon čega slijedi odgovor poslužitelja.

Komunikacija na načelu push: klijent šalje zahtjev i nastavlja dalje s obradom, a registrira poseban proces (engl. listener) koji "osluškuje" i čeka odgovor poslužitelja. Poslužitelj šalje odgovor u nekom budućem trenutku kada završi obradu zahtjeva ili kada mu npr. traženi podaci postanu dostupni. Listener je zapravo poseban poslužiteljski proces koji mora biti aktivan i može primiti poruke.

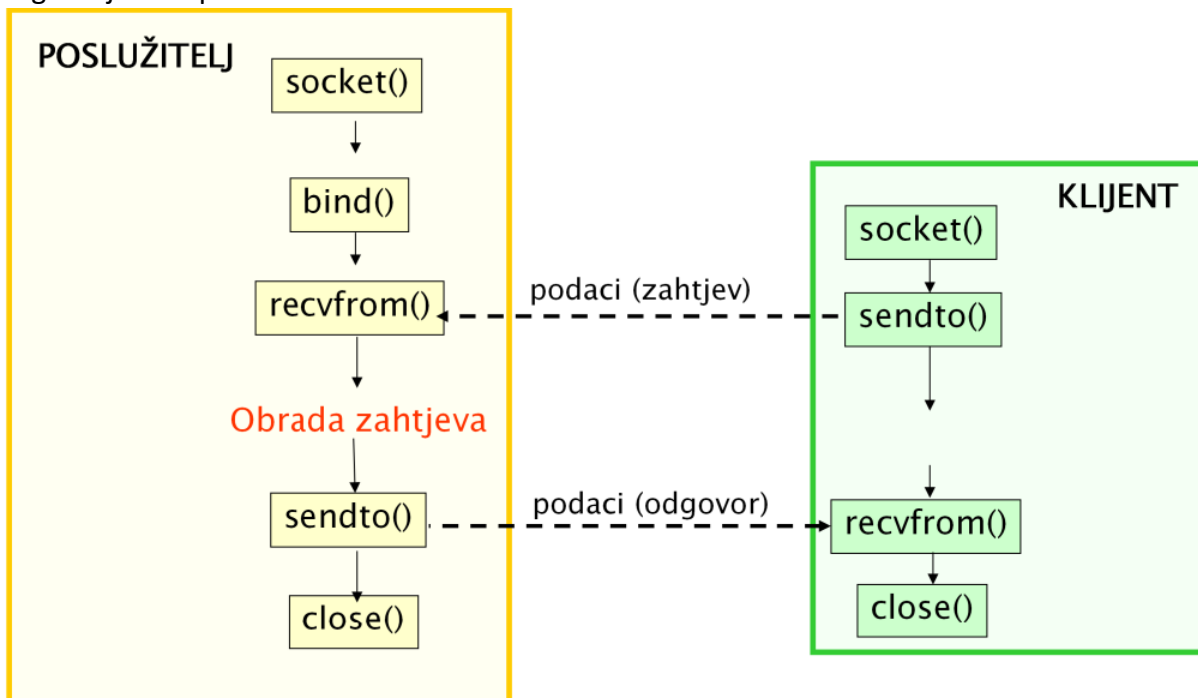
14. Što je priključnica?

Komunikacijska točka prek koje proces šalje podatke u mrežu i iz koje čita primljene podatke. Visi nivo apstrakcije nad komunikacijskom točkom koju operacijski sustav koristi za pristup transportnom sloju. Veze se uz broj vrata(port) koja jednodužno određuju proces kojemu su poruke namijenjene.

15. Skicirajte i objasnite dijagram toka komunikacije korištenjem priključnice UDP. Koja je metoda blokirajuća?



Kada proces čita poruku iz spremnika može ostati blokiran ako je spremnik prazan dok ne stigne sljedeća poruka.



socket: kreira novu komunikacijsku točku (socket)

bind: povezuje transportnu adresu (IP adresa, port) i socket

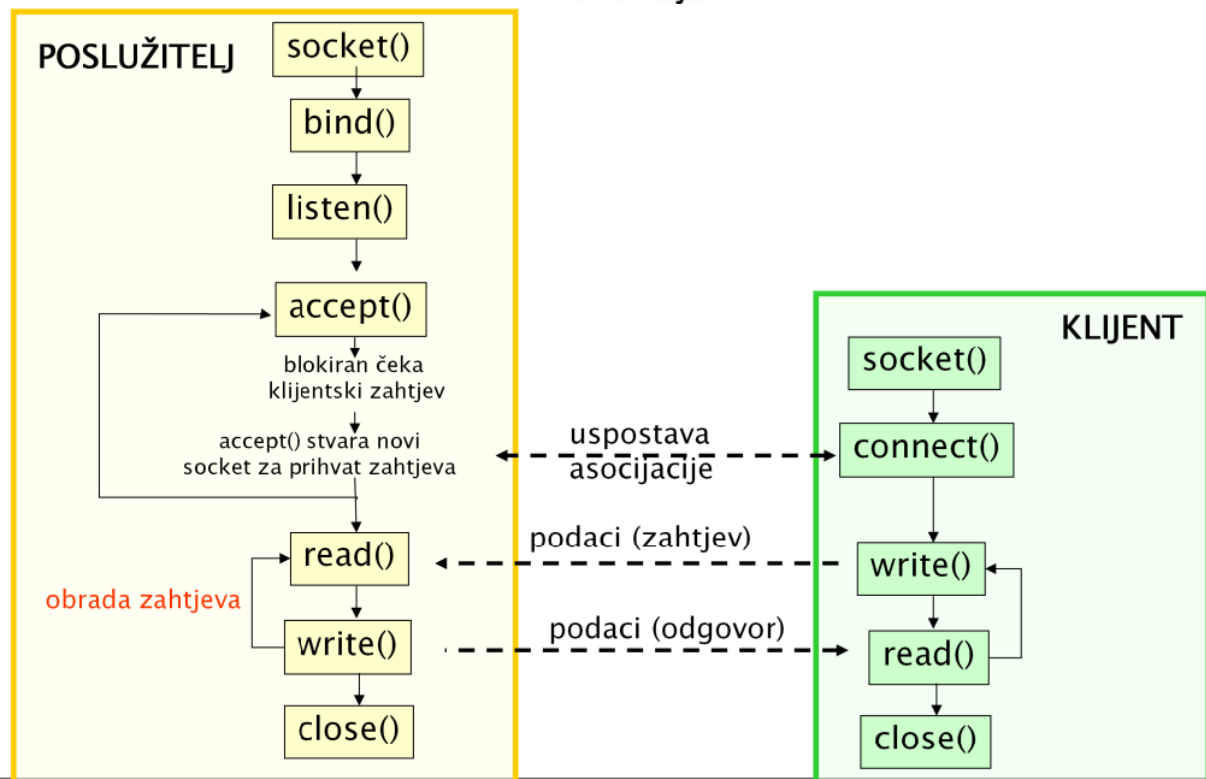
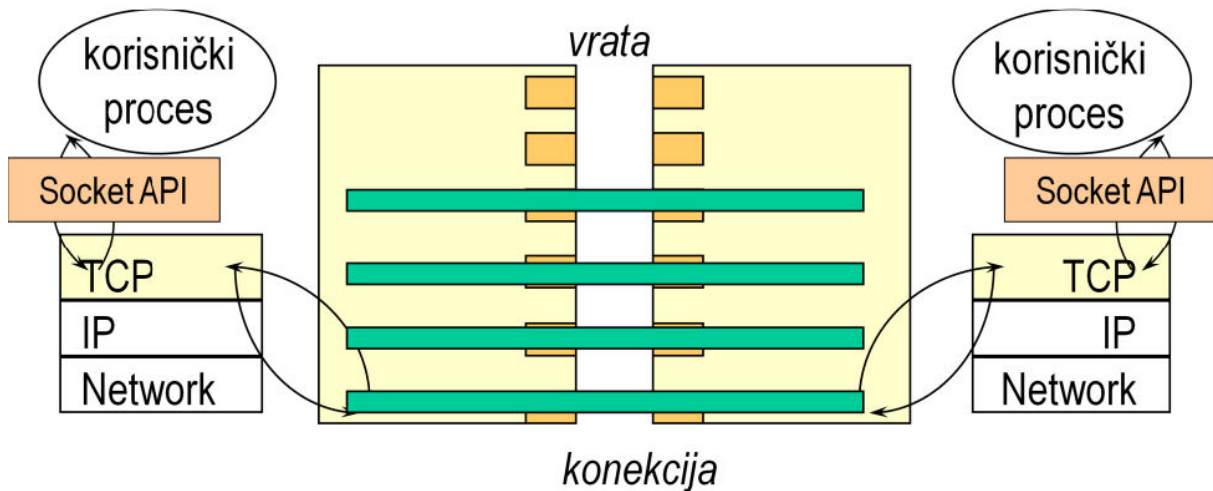
recvfrom: primanje datagrama (zahtjeva ili odgovora)

sendto: slanje datagrama (zahtjev ili odgovor)

close: zatvara konekciju

16. Skicirajte i objasnite dijagram toka komunikacije korištenjem priključnice TCP. Koja je metoda blokirajuća?

Klijent zatraži uspostavu konekcije i ostaje blokiran dok poslužitelj prihvati konekciju.



POSUŽITELJ

socket–kreira komunikacijsku točku, operacijski sustav rezervira resurse koji će omogućiti slanje i primanje podataka koristeći odabrani transportni protokol

bind–povezuje adresu sa socketom. Poslužitelj povezuje IP adresu računala i broj porta sa socketom.

listen–omogućuje operacijskom sustavu rezerviranje resursa (spremnika) za specificirani maksimalni broj konekcija.

accept–poslužitelj prima zahtjev za iniciranje konekcije od strane klijenta (connect).

Poslužitelj stvara novi identičan socket koji se koristi za komunikaciju s klijentom. Originalni socket se koristi za “osluškivanje” novih zahtjeva.

readi write–slanje i primanje podataka

KLIJENT

socket-kreira komunikacijsku točku, bind nije potreban jer OS dinamički alocira port socketu pri kreiranju konekcije.

connect–klijent šalje zahtjev za kreiranje konekcije. Klijent mora definirati transportnu adresu na koju se šalje zahtjev za kreiranje konekcije. Klijent je blokiran do uspostave konekcije.

readi write–slanje i primanje podataka

close–zatvaranje konekcije

17. Usporedite komunikaciju udaljenih procesa korištenjem priključnice UDP i TCP.

TCP:

arhitektura klijent-poslužitelj

vremenska ovisnost- moraju bit obojca dostupni

klijent mora znati id poslužitelja

tranzijentna komunikacija

sinkrona komunikacija- klijent šalje zahtjev za kreiranje konekcije i blokiran je do uspostave konekcije

pokretanje komunikacije na nacelu pull

UDP:

arhitektura klijent-poslužitelj

vremenska ovisnost- poslužitelj mora biti aktivan za primanje datagrama

klijent mora znati id poslužitelja

tranzijentna komunikacija

asinkrona komunikacija- klijent šalje datagram i nastavlja procesiranje

može se koristiti za implementaciju komunikacije na nacelu pull i push

18. Objasnite komunikaciju udaljenih objekata korištenjem poziva udaljene metode te navedite svojstva.

Proces na računalu A poziva proceduru koja se izvodi na računalu B.

Pozivajući proces na računalu A šalje parametre za izvođenje procedure na računalu B i blokiran je čekajući rezultate izvođenja procedure.

Računalu B izvodi proceduru koristeći primljene parametre i šalje odgovor računalu A.

Transparentnost: za proces na računalu A poziv udaljene procedure čini se jednak pozivu lokalne procedure.

RMI:

arhitektura klijent-poslužitelj

transparentan pristup udaljenim metodama

vremenska ovisnost- moraju bit obojca dostupni

klijentski objekt mora znati id udaljenog poslužiteljsko objekta

tranzijentna komunikacija

sinkrona komunikacija- klijent blokiran dok ne primi odgovor od strane poslužitelja

pokretanje komunikacije na nacelu pull

19. Objasnite komunikaciju udaljenih procesa razmjenom poruka te navedite svojstva.

Osnovna ideja komunikacije porukama je razmjena poruka među procesima preko posrednika, tj. postoji poseban rep pridijeljen primatelju poruke koji se održava na posebnom poslužitelju. Pošiljatelju se u načelu garantira isporuka poruke u primateljev rep, ali ne i isporuka poruke primatelju. Primatelj može pročitati poruku iz repa u bilo kojem budućem trenutku. Stoga su pošiljatelj i primatelj poruke vremenski neovisni. Vazna je adresa odrednog repa.

Komunikacija porukama:

vremenska neovisnost- nemoraju bit istovremeno aktivni, sprema se u rep

posiljalatelj mora znati id odredista tj repa

persistentna komunikacija

asinkrona komunikacija- posiljalatelj salje poruku i nastavlja procesiranje

pokretanje komunikacije na nacelu pull- primatelj provjerava postoji li poruka u repu

20. Objasnite komunikaciju udaljenih procesa na nacelu objavi-pretplati te navedite svojstva.

Usluga objavi-pretplati je posrednik između strana u komunikaciji i može se zamisliti kao informacijska sabirница koja povezuje grupu pretplatnika s grupom objavljiivača poruka.

Komunikacijski model objavi-pretplati obuhvaća objavljiivače (objekte koji su pošiljalitelji podataka), pretplatnike (objekte koji su primatelji podataka) i uslugu objavi-pretplati.

Objavljiivači definiraju obavijesti, dok pretplatnici pretplatama pretplata izražavaju interes za primanje određenog skupa obavijesti (pretplata opisuje svojstva obavijesti). U slučaju kada obavijest zadovoljava svojstva definirana pretplatom, usluga objavi-pretplati isporučuje obavijest njenom pretplatniku. Jedna obavijest se isporučuje grupi pretplatnika ovisno o definiranim pretplatama.

Komunikacija porukama:

vremenska neovisnost- nemoraju bit istovremeno aktivni, obavjesti se pohranjuju do isporuke

objavljiivac ne mora znati id pretplatnika, o tome se brine posrednik

persistentna komunikacija

asinkrona komunikacija- objavljiivac salje poruku i nastavlja procesiranje

pokretanje komunikacije na nacelu push- objavljiivac salje poruku posredniku koji je prosljedjuje pretplatnicima bez predhodnog explicitnog zahtjeva

21. Koji biste transportni protokol koristili za implementaciju sustava objavi-pretplati i zašto.

UDP... ma mora negdje i TCP... ovisi na kojem dijelu... pitanje nije na mjestu

22. Navedite karakteristično obilježje(a) komunikacije udaljenih procesa na nacelu objavi-pretplati u odnosu na ostale modele za komunikaciju u raspodijeljenoj okolini.

Evo jos jednom :D

Komunikacija porukama:

vremenska neovisnost- nemoraju bit istovremeno aktivni, obavjesti se pohranjuju do isporuke

objavljiivac ne mora znati id pretplatnika, o tome se brine posrednik

persistentna komunikacija

asinkrona komunikacija- objavljiivac salje poruku i nastavlja procesiranje

pokretanje komunikacije na nacelu push- objavljiivac salje poruku posredniku koji je prosljedjuje pretplatnicima bez predhodnog explicitnog zahtjeva

23. Objasnite komunikaciju udaljenih procesa strujanjem podataka te navedite svojstva.

Kontinuirani mediji zahtijevaju pravovremenu isporuku i vremensko usklađivanje prenesenih podataka (paketa) koje nazivamo strujom podataka, npr. audio i video. Paketi iz struje podataka su međusobno vremenski ovisni jer je kvaliteta reprodukcije vrlo osjetljiva na kašnjenje paketa i varijaciju kašnjenja paketa (*packet jitter*). Često je potrebno vremenski vremenski uskladiti (sinkronizirati) više struja podataka (npr. video i audio) te stoga kažemo da je za ovu vrstu višemedijskih usluga nužna kontrola kvalitete prijenosa s kraja na kraj.

Na strani višemedijskog poslužitelja mora postojati raspoređivač i repozitorij sadržaja, dok je na strani klijenta medijski preglednik i spremnik. Spremnik se puni podacima i repozitorija i prosljeđuje do preglednika, a služi za kontrolu varijacije kašnjenja paketa. Medijski preglednik i raspoređivač razmjenjuju kontrolne podatke za kontrolu prijenosa.

Strujanje podataka:

arhitektura klijent-poslužitelj

vremenska ovisnost klijenta i poslužitelja

perzistentna komunikacija

izokroni prijenos paketa- minimalno i max kasnjenje se definira s kraja na kraj

vremenska uskladenost podataka unutar jedne struje podataka

sinkronizacija više struja podataka

24. Usporedite komunikaciju udaljenih procesa strujanjem podataka s komunikacijom udaljenih procesa razmjenom poruka.

Kod komunikacije porukama postoji vremenska neovisnost, a kod struje je vremenska ovisnost (live stream)