



**VISOKA POSLOVNA ŠKOLA
STRUKOVNIH STUDIJA
ČAČAK**

DIPLOMSKI RAD

Predmet: Multimedijalne tehnologije

Multimedijalne komunikacije

Mentor: _____
Profesor: _____

Student: _____
Br.Indeksa: _____

SADRŽAJ

Uvod.....	3
1. Digitalizacija zvuka i slike	4
1.1 Digitalizacija zvuka.....	4
1.2 Digitalizacija slike	4
2. Kompresija zvuka i slike.....	4
2.1 Kompresija zvuka	4
2.1.1 Prediktivno kodiranje	4
2.1.2 Perceptualno kodiranje (MP3).....	4
2.2 Video kompresija	5
2.2.1 JPEG	5
2.2.2 MPEG	8
3. Streaming snimljenog audio/video materijala.....	10
3.1 Prvi pristup: Web Server	10
3.2 Drugi pristup: Web Server sa <i>Metafile</i> -om	10
3.3 Treći pristup: Media Server.....	11
3.4 Četvrti pristup: Media Server i RTSP	12
4. Streaming živog audio/videoa.	13
5. Interaktivna audio/video komunikacija u realnom vremenu.	13
5.1 Karakteristike	13
5.1.1 Odnos u vremenu.....	13
5.1.2 <i>Timestamp</i> (vremenski kod)	14
5.1.3 Playback Buffer (Bafer reprodukcije).....	15
5.1.4 Sortiranje.....	15
5.1.5 Multikasting	16
5.1.6 Prevodioci (<i>translators</i>)	16
5.1.7 Mešanje (<i>mixing</i>)	16
5.1.8 Podrška transportnog sloja (OSI model)	16
6. RTP (Real-time Transfer Protocol).....	17
6.1 Format RTP datagrama	18
7. RTCP (Real-time Transport Control Protocol).....	19
7.1.1 Sender report (Izveštaj pošiljaoca)	20
7.1.2 Receiver report (Izveštaj prijemnika)	20
7.1.3 Source Description Message (Opis izvora).....	20
7.1.4 Bye Message (Pozdravna poruka)	20

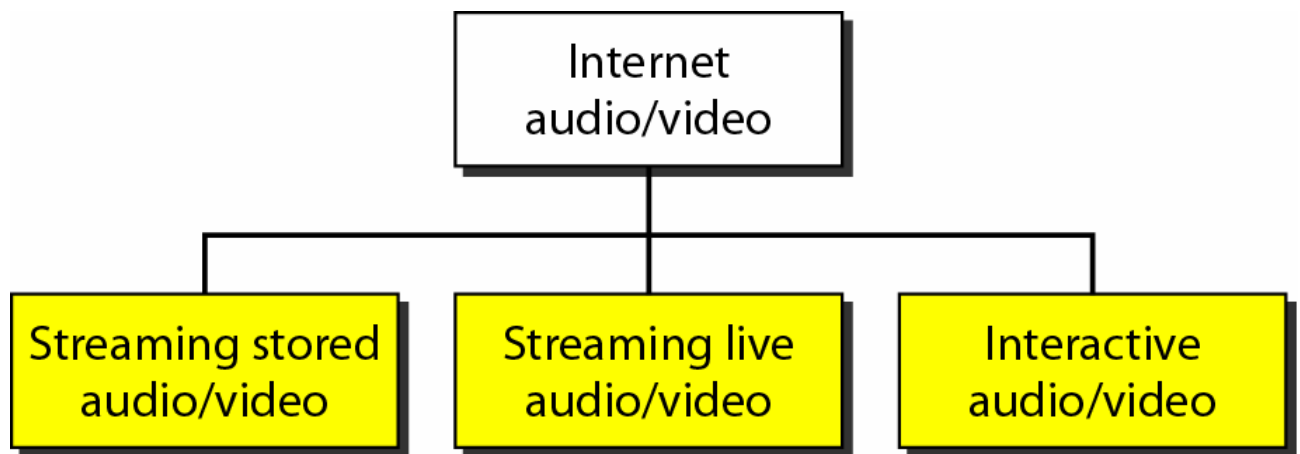
7.2 UDP Port.....	21
8.Voice Over IP (VOIP, Glas preko IP-a).....	21
8.1 SIP	21
8.2 Poruke	21
8.3 Adrese.....	22
8.4 Jednostavna sesija.....	22
8.5 Praćenje sagovornika	23
8.6 H.323	23

Uvod

Pojam multimedija odnosi se na takav sadržaj koji od nas zahteva upotrebu više čula da bi smo ga shvatili u potpunosti. Recimo, televizija je tipičan primer multimedijalnog sadržaja, zato što nam na raspolaganje stavlja i sliku i zvuk, aktivira nam i čulo vida, i čulo sluha, dok nam na primer čitanje ili gledanje fotografija aktivira samo čulo vida.

Napredak tehnologije promenio je način na koji koristimo zvuk i sliku. Ranije smo slušali radio i gledali program preko televizije, koristili smo telefon za interaktivnu komunikaciju, ali vremena su se promenila. Ljudi žele da koriste Internet ne samo za prenos teksta i statičnih slika, već i za audio i video komunikaciju.

Audio i video Internet servise delimo u tri kategorije. *Streaming* snimljenog audio/video zapisa, *streaming* žive slike i zvuka i interaktivni audio/video, Kao na slici (1). Termin *streaming* znači da korisnik može da sluša ili gleda fajl nakon što je *download* počeo.



SLIKA 1. Podela audio/video Internet servisa.

Pri *streaming*-u snimljenog audio/video fajlovi su komprimovani i čuvaju se na serveru. Klijent *download*-uje fajlove preko Interneta. Ovo se ponekada naziva video-audio na zahtev. Primeri snimljenog audio zapisa su pesme, simfonije, knjige, predavanja. Primeri snimljenog video zapisa su filmovi TV emisije i muzički spotovi.

Za *streaming* audio/video materjala potrebna je kompresija tog materjala.

Pri *streaming*-u živog materjala korisnik prati emisiju putem Interneta. Dobar primer ovoga je Internet radio. Neke radio stanice svoj program emituju samo preko Interneta; dok mnoge emituju i preko Interneta i u etar. Internet TV je sve popularniji, ali mnogi ljudi veruju da će klasična televizija još dugo opstati.

Termin *streaming* živog audio/video odnosi se na emitovanje radio i TV programa putem Interneta.

Što se tiče interaktivnog audio/video on se sada koristi za komunikaciju na daljinu. Dobar primer za to Internet telefonija i telekonferencije.

1. Digitalizacija zvuka i slike

Pre nego što se zvučni i video signali mogu poslati putem Interneta moraju se digitalizovati.

1.1 Digitalizacija zvuka

Kada mikrofoni detektuju zvučni talas na svom izlazu daje analogni signal koji predstavlja amplitudu tog zvuka. Ovaj signal se zove analogni audio signal. Prema Nyquist-ovoj teoremi ako je najveća frekvencija signala f moramo ga uzorkovati $2f$ puta u sekundi. Postoje i druge metode za digitalizaciju zvuka ali je princip isti.

Ljudski glas se uzorkuje 8000 puta u sekundi sa 8 bita po uzorku. Rezultujući signal je 64 Kbps. Muzika se uzorkuje sa 44100 uzorka u sekundi sa 16 bita po uzorku. To daje signal 705,6 Kbps za mono ili 1,411 Mbps za stereo.

1.2 Digitalizacija slike

Video snimak se sastoji od sekvence *frame*-ova. Ako se ti *frame*-ovi pokazuju dovoljno brzo imamo utisak pokretne slike. Razlog za to je tromost oka. Postoje više standarda za broj *frame*-ova u sekundi (fps) : 25 fps PAL, 30 fps NTSC ... Da bi se izbeglo treperenje slike svaki *frame* se prikazuje dva puta.

Svaki *frame* je podeljen na male delove koji se zovu pikseli. Za crno-belu sliku to je 8 bita (256 nijansi sivog), a za kolor 24 bita (po 8 za svaku od osnovnih boja).

Možemo da izračunamo broj bitova u jednoj sekundi za određenu rezoluciju. Na primer za rezoluciju 1024 x 768 potrebno je $2 \times 25 \times 1024 \times 768 \times 24 = 944$ Mbps.

2. Kompresija zvuka i slike

Da bi se zvuk i slika poslali putem Interneta moraju se komprimovati.

2.1 Kompresija zvuka

Potrebni su različiti stepeni kompresije za prenos govora i muzike. Za govor je to 64 KHz digitalni signal, a za muziku 1,411 MHz signal. Postoje dve tehnike za komprimovanje zvuka: prediktivna i perceptualna.

2.1.1 Prediktivno kodiranje

Kod ovakvog tipa kodiranja beleže se razlike između uzoraka umesto vrednosti svih uzoraka. Ovakva vrsta kompresije se obično koristi za govor. Postoje nekoliko definisanih standarda kao što su GSM (13 Kbps), G.729 (8 Kbps), G.723.3(6,4 Kbps ili 5,3 Kbps).

2.1.2 Perceptualno kodiranje (MP3)

Najčešće korišćena tehnika kompresije koja daje CD kvalitet zvuka je zasnovana na perceptualnom kodiranju. MP3 je deo mpeg standarda koji koristi ovu tehniku.

Perceptualno kodiranje je zasnovano na psiho akustici koja proučava ljudsku percepciju zvuka. Ova ideja je zasnovana na nesavršenosti našeg slušnog aparata. Neki zvuci mogu da prikriju neke druge zvuke. Ovo "maskiranje" se može desiti i po frekvenciji i po vremenu. Ko frekventnog maskiranja glasan zvuk jedne frekvencije može potpuno zamaskirati tiši zvuk druge. Na primer ne možemo da čujemo glas našeg prijatelja ako se u pozadini čuje glasna muzika. Pri vremenskom maskiranju glasan zvuk može da nam odzvanja u slušnom aparatu i nakon što je prestao.

MP3 koristi oba fenomena pri kompresiji zvuka. Spektar signala se deli u nekoliko grupa. Potpuno maskirani delovi spektra kodiraju se nolama. Mali broj bitova se koristi za delimično maskirane frekvencije, a najveći broj bitova za one koji nisu maskirani.

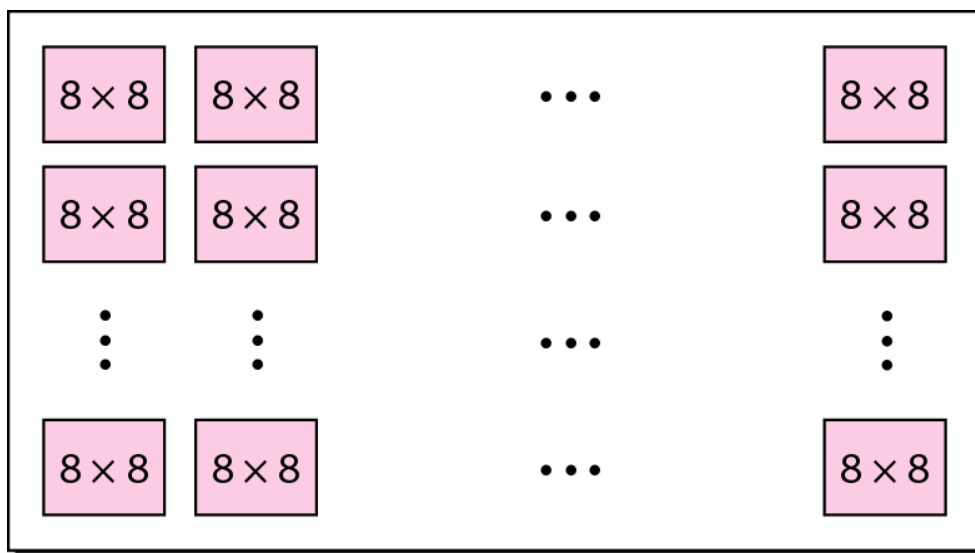
MP3 podržava razne brzine protoka, na primer 96 Kbps, 128 Kbps, 160 Kbps ...

2.2 Video kompresija

Svaki *frame* u video zapisu je posebna slika. Da bi smo komprimovali video zapis prvo moramo komprimovati svaku od tih slika. Na tržištu prevladavaju 2 standarda JPEG i MPEG. Prvi se koristi za statične a drugi za pokretne slike.

2.2.1 JPEG

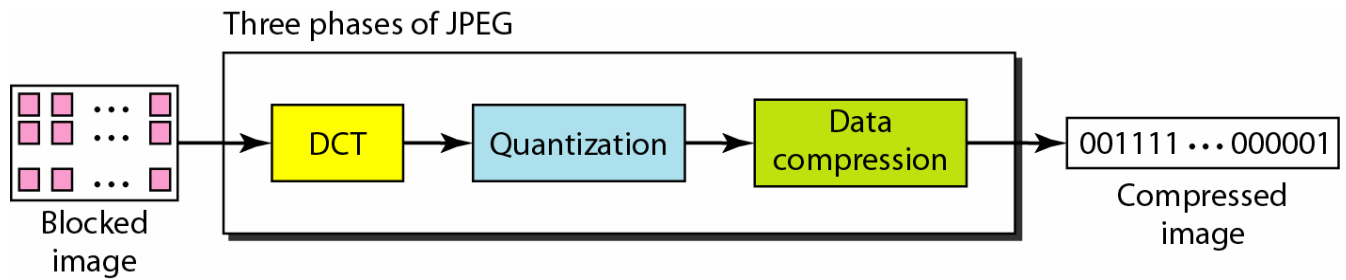
Kao što smo ranije utvrdili za crno-belu sliku potrebno je 8 bita (256 nijansi sivog), a za kolor 24 bita (po 8 za svaku od osnovnih boja). Objasnićemo na primeru crno bele slike.



SLIKA 2. Podela slike na blokove kod JPEG kompresije.

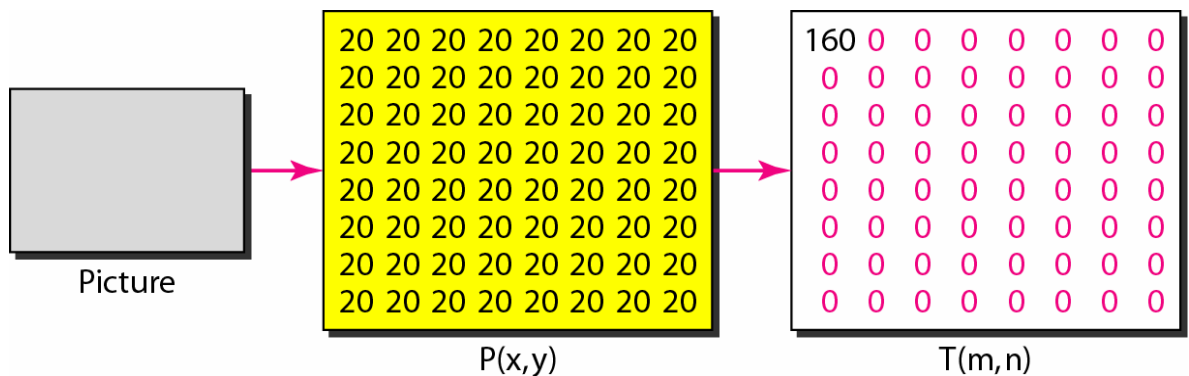
Slika se prvo podeli na blokove 8 x 8 piksela. Svrha ove podele je smanjivanje broja potrebnih računskih operacija. Ideja JPEG-a je u tome da se slika pretvori u vektor kako vi se otkrilo ponavljanje bitova. Ovo ponavljanje se može ukloniti nekom od tehnika kompresije teksta.

Diskretna kosinusna transformacija (DCT): u ovom koraku svaki blok od 64 piksela prolazi kroz DCT algoritam. Ova transformacija menja ovih 64 vrednosti tako da se očuva relativan odnos između piksela, ali se redundantnost uklanja. Moguća su tri slučaja.



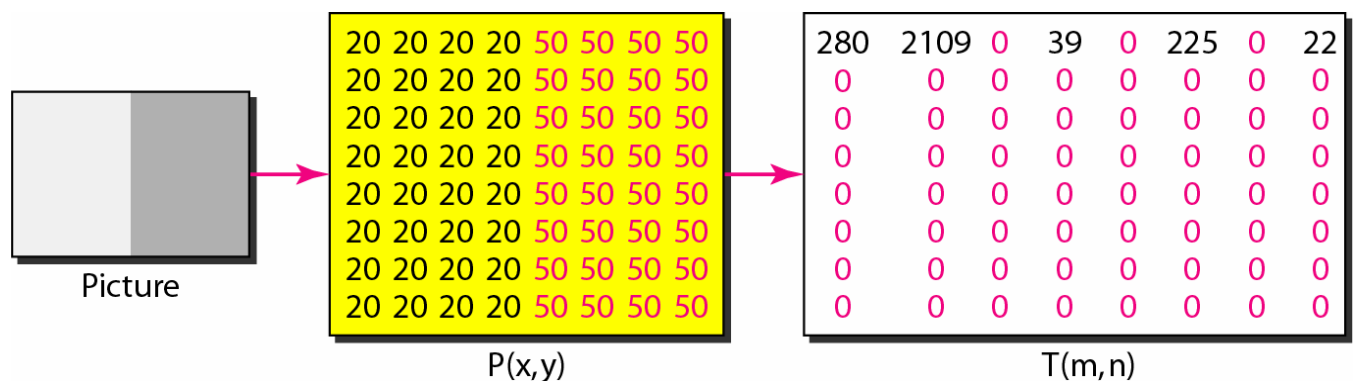
SLIKA 3. Tri faze JPEG kompresije: DCT, kvantizacija, kompresija podataka.

Prvi slučaj. U ovom slučaju imamo ravnomerno sivi blok. Vrednost svakog piksela je 20. Nakon DCT transformacije dobijamo vrednost samo za prvi element ostali pikseli se postvljaju na nulu. Vrednost prvog elementa je prosečna vrednost blokakoja se često naziva jednosmerna kompnenta. Ostatak vrednosti, naizmenične komponente predstavljaju promene vrednosti piksela. Pošto nema promena ostale vrednosti su nule.



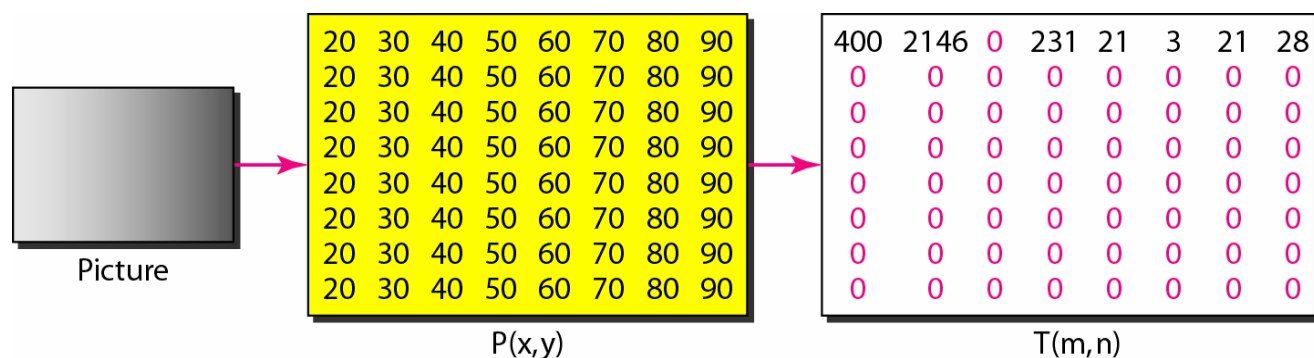
SLIKA 4. Primer JPEG kompresije, jednobojni blok.

Drugi slučaj. Sada imamo blok podeljen na dva dela. Jedna polovina je jedna nijansa sive a druga polovina je druge nijanse. Postoji oštra razlika između vrednosti piksela. Nakon DCT-a dobijamo jednosmernu komponentu kao i naizmeničnu. Međutim postoje samo nekoliko vrednosti različitih od nule.



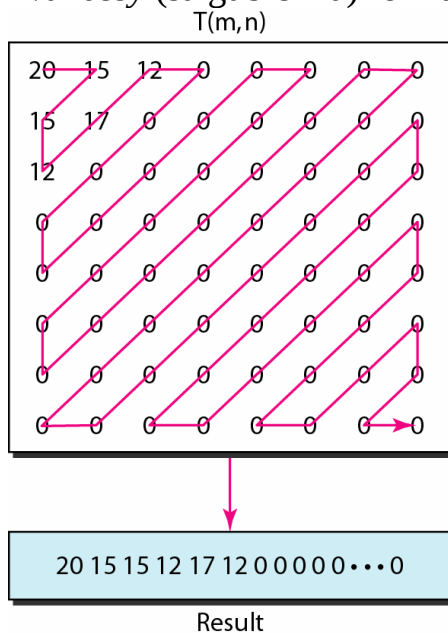
SLIKA 5. Primer JPEG kompresije, oštar prelaz.

Treći slučaj. Sada imamo blok čije se vrednosti postepeno menjaju. Nema nagle promene između vrednosti susednih piksela. Nakon DCT transformacije dobijamo jednosmernu komponentu, ali su i dalje mnoge naizmenične komponente jednake nuli.



SLIKA 6. Primer JPEG kompresije, mekan prelaz.

Kvantizacija. Nakon što je tabela T kreirana upotrebom DCT algoritma, vrednosti se kvantizuju da bi se smanjio broj bitova potrebnih za kodiranje. U prethodnim kvantizacijama jednostavno smo odbacili decimalni deo razlomka, ostavljajući samo ceo broj. Sada razlomak prvo delimo konstantom, pa zatim odbacujemo decimalni deo. Ova operacija još više umanjuje broj bitova potrebnih za kodiranje. U većini slučajeva, kvantizacija se definiše tabelom kvantizacije, u kojoj je opisano kako se kvantizuje svaka od mogućih vrednosti. Delitelj zavisi od pozicije vrednosti u tabeli. Ovo je potrebno da bi se optimizovao broj izbačenih vrednosti za posebnu aplikaciju algoritma. Zapazimo da je da je jedina faza procesa kompresije koja nije reverzibilna kvantizacija. Kvantizacijom se nepovratno gubi deo informacija. To je i jedini razlog zašto se JPEG naziva *lossy* (sa gubicima) formatom.



SLIKA 7. Čitanje kvantizirane tablice kod JPEG kompresije.

Kompresija. Nakon kvant-izacije, vrednosti se čitaju iz tabele, i redundantne nule se uklanjaju. Međutim, da bi se postojeće nule grupisale, tabela se čita dijagonalno (cik-cak). Razlog za takav način čitanja umesto čitanja red po red ili kolonu po kolonu je u tome što ako je prelaz između vrednosti gladak, većina nula će se naći na kraju ovako pročitane niza.

2.2.2 MPEG

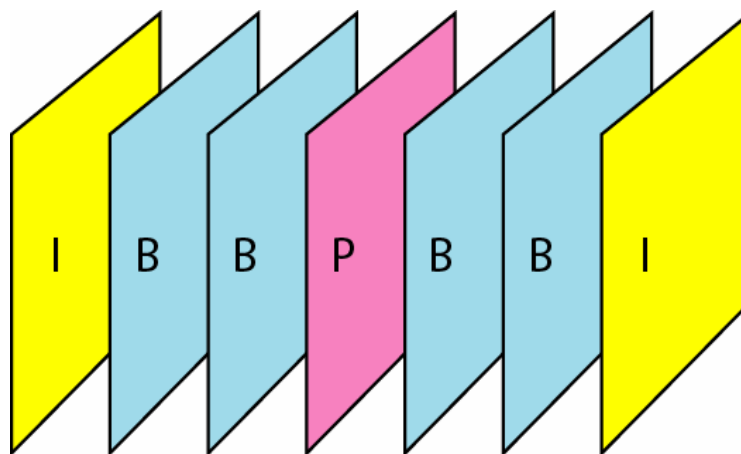
Ova metoda se koristi pri kompresiji video zapisa (pokretnih slika). U principu, video je vremenska kombinacija statičnih slika koje se šalju jedna za drugom. Komprimovati video zapis dakle znači komprimovati slike prostorno, a sekvencu slika vremenski.

Prostorna kompresija. Prostorna kompresija vrši se već pomenutim JPEG algoritmom. Algoritam se primenjuje na svaki *frame*.

Vremenska kompresija. Kod vremenske kompresije, redundantni *frame*-ovi se izbacuju. Kada gledamo PAL televiziju, mi ustvari pratimo 50 interlaced *frame*-ova u sekundi (50 poluslika). Međutim, većina susednih *frame*-ova su veoma slični. Naprimera, kada neka osoba govori, veliki deo slike je statičan, osim dela oko usana koji se menja iz *frame*a u *frame*.

Da bi vremenski komprimovao video zapis, MPEG algoritam deli *frame*-ove u tri kategorije: *I-frame*, *P-frame* i *B-frame*.

- **I-frame.** *Intracoded frame* je nezavistan *frame* koji nije povezan ni sa jednim od susednih. Oni se postavljaju na određenom intervalu (na primer, svaki deveti *frame*). *I-frame* se mora ubaciti kako bi se vernije prikazali nagli pokreti. Takođe, kada se video emituje, gledalac se može uključiti u bilo kom trenutku. Ako bi postajao samo jedan *I-frame* na početku emisije, gledalac koji se uključio kasnije neće moći da dobije kompletnu sliku.



SLIKA 8. MPEG sekvenca sa tri vrste *frame*-ova.

- **P-frame.** Predicted (predviđeni, u smislu da je moguće predvideti njegov

- **B-frame.** Bidirekcionni *frame* je vezan i za prethodni i za sledeći *frame*, koji mogu biti I ili P tipa. Drugim rečima, B-frame se odnosi i na prošlost i na budućnost. Zapažimo da se ne može vezati za drugi B-frame.

The diagram illustrates a feedforward neural network with 7 input nodes and 7 output nodes. The input nodes are labeled 1 through 7. The output nodes are labeled I, B, B, P, B, B, I. The network is fully connected, with solid lines representing connections from input nodes to output nodes and dashed lines representing connections from output nodes to input nodes.

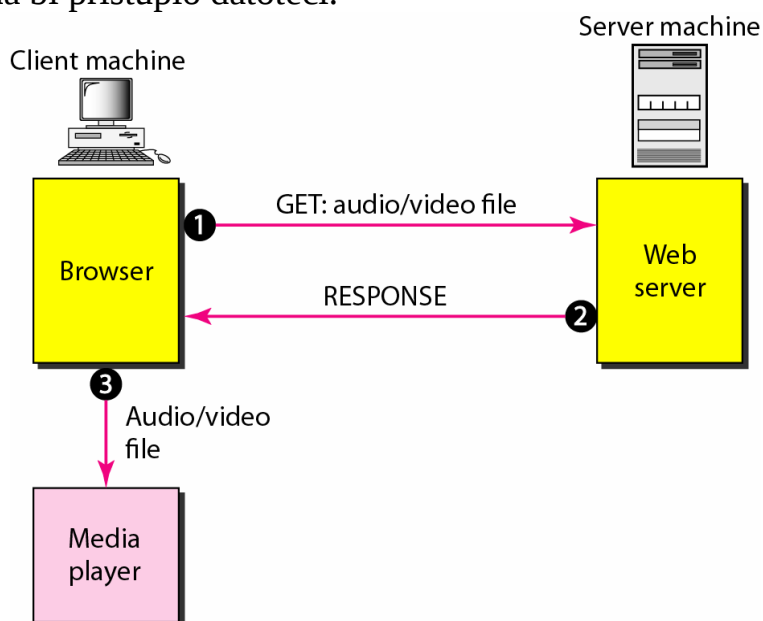
MPEG je prošao kroz dve verzije. Prva, MPEG1 je projektovana za CD-ROM, sa protokom od 1.5 Mbps. MPEG2 je projektovan za DVD snimak visokog kvaliteta sa protokom od 3 do 6 Mbps.

3. Streaming snimljenog audio/video materijala

Download-ovanje ("skidanje") ovakvog sadržaja sa Interneta razlikuje se od *download*-ovanja drugih tipova fajlova. Da bi smo razumeli koncept, razmotrićemo četiri različita pristupa problemu, svaki različite složenosti.

3.1 Prvi pristup: Web Server

Komprimovani audio/video zapis može se *download*-ovati kao tekstualni fajl. Klijent (*browser*) koristi servise HTTP-a i šalje GET zahtev serveru. Server onda može poslati traženi fajl. *Browser* na klijentovom računaru onda koristi pomoćnu aplikaciju, "*media player*", da bi pristupio datoteci.



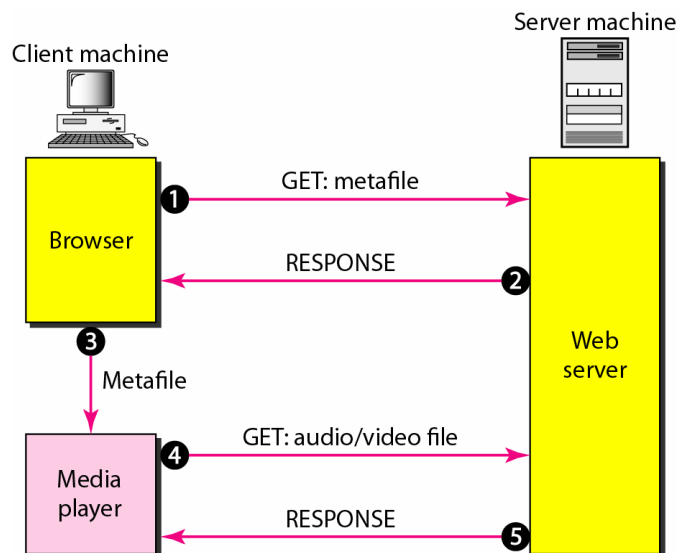
SLIKA 10. Pristup multimedijalnom sadržaju putem Web servera.

Ovaj proces je veoma jednostavan i ne zahteva *streaming*. Međutim, on ima svoje loše strane. Multimedijalni fajlovi su obično veliki, čak i posle kompresije. Audio zapis može imati nekoliko desetina megabajta, dok video čak i više stotina. Pri upotrebi Web Servera, kompletan fajl mora biti prenešen na računar klijenta kako bi mu se pristupilo. To zahteva od korisnika da čeka nekoliko minuta, ili čak sati kako bi pregledao (preslušao) zapis.

3.2 Drugi pristup: Web Server sa *Metafile*-om

Ovim postupkom se *media player* direktno vezuje za web server. Server skladišti dva fajla: audio/video zapis, i *metafile* koji sadrži informacije o zapisu. Pri ovakvoj vrsti pristupa prolazi se kroz sledeće korake:

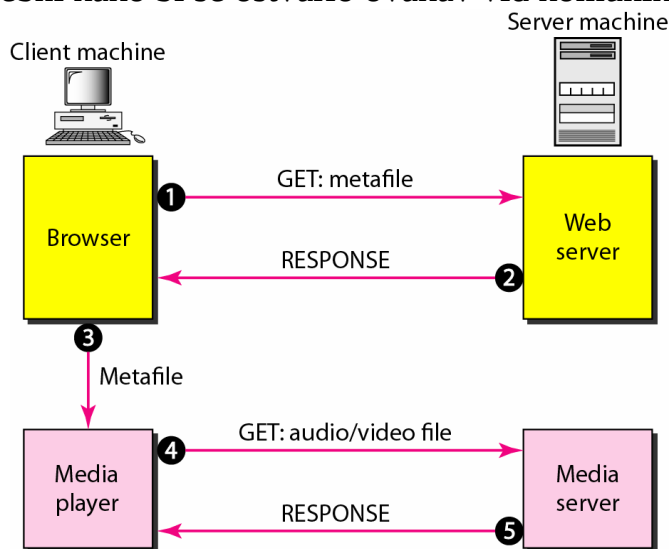
1. HTTP klijent pristupa Web serveru naredbom GET.
2. Kao odgovor klijent dobija *metafile*.
3. Metafajl se predaje *media player*-u.
4. Media plazer koristi URL iz *metafile*-a kako bi pristupio audio/video zapisu.
5. Web server odgovara.



SLIKA 11. Pristup multimedijalnom sadržaju putem Web servera koristeći *metafile*.

3.3 Treći pristup: Media Server

Problem sa drugim pristupom je u tome što *browser* i *media player* oboje koriste usluge HTTP-a. HTTP je projektovan da komunicira preko TCP protokola. On je pogodan za prijem *metafile*-a, ali ne i za audio/video zapis. TCP protokol prati tok podataka i ponovo šalje izgubljene pakete, čime krši samu filozofiju *streaming*-a. Potrebno je zaobići TCP i njegovu kontrolu grešaka. Koristimo UDP. Kako su Web Serveri okrenuti isključivo TCP-u, potreban nam je još jedan sever: Media Server. Sledeći koraci su potrebni kako bi se ostvario ovakav vid komunikacije.

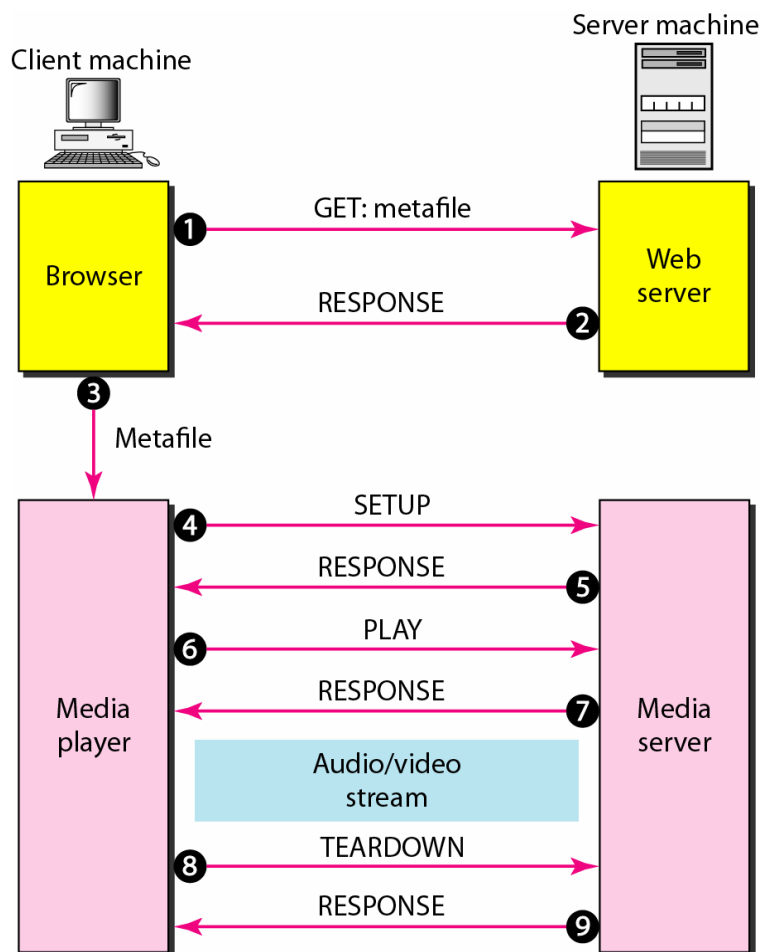


SLIKA 12. Pristup multimedijalnom sadržaju putem Media servera.

1. HTTP klijent pristupa Web serveru naredbom GET.
2. Kao odgovor Web server šalje *metafile*.
3. Metafajl je prosleđen *media player*-u.
4. Media plazer koristi URL iz *metafilea* da bi pristupio Media serveru, koristeći UDP protokol.
5. Media server odgovara traženim *stream*-om.

3.4 Četvrti pristup: Media Server i RTSP

Real Time *Streaming* Protocol (RTSP) je upravljački protokol projektovan da uveća funkcionalnost *streaming* procesa. Koristeći RTSP imamo punu kontrolu nad reprodukcijom audio/video zapisa. RTSP je *out-of-band* upravljački protokol koji je sličan sekundarnoj konekciji kod FTP-a. Sledeći koraci su potrebni kako bi se ostvario ovakav vid komunikacije.



SLIKA 13. Pristup multimedijalnom sadržaju putem Media servera i RTSP-a.

1. HTTP klijent pristupa Web serveru naredbom GET.
2. Kao odgovor Web server šalje *metafile*.
3. Metafajl je prosleđen *media playeru*.
4. *Media player* šalje komandu SETUP da bi uspostavio vezu sa Media serverom.
5. Media server odgovara.
6. *Media player* šalje komandu PLAY, kako bi započeo reprodukciju (*streaming*).
7. Audio/video zapis se prenosi mrežom koristeći protokol zasnovan na UDP-u.
8. Konekcija se prekida TEARDOWN komandom.
9. Media server odgovara.

Postoje i druge komande kojima klijent i server komuniciraju, kao što je na primer PAUSE, koja privremeno obustavlja *streaming* do prijema PLAY komande.

4. Streaming živog audio/video.

Streaming živog audio/video je sličan emitovanju TV-a. Umesto predajnika koji emituje radiotalase u etar, stanice emituju program putem Interneta. Postoje nekoliko sličnosti između *streaming*-a snimljenog i živog programa. Obadva su osetljiva na kašnjenje i ne dozvoljavaju retransmisiju izgubljenih ili oštećenih paketa. Međutim, postoji razlika. Kod prvog je komunikacija unicast i na-zahtev tipa. Kod živog *streaming*-a (*live streaming*) komunikacija je multicast tipa i odvija se u realnom vremenu. Međutim, iako je mnogo pogodnije koristiti multicast mogućnosti IP protokola i UDP-a ili RTP-a, trenutno se uglavnom koristi TCP i višestruki *unicast*, a ne pravi *multicasting*.

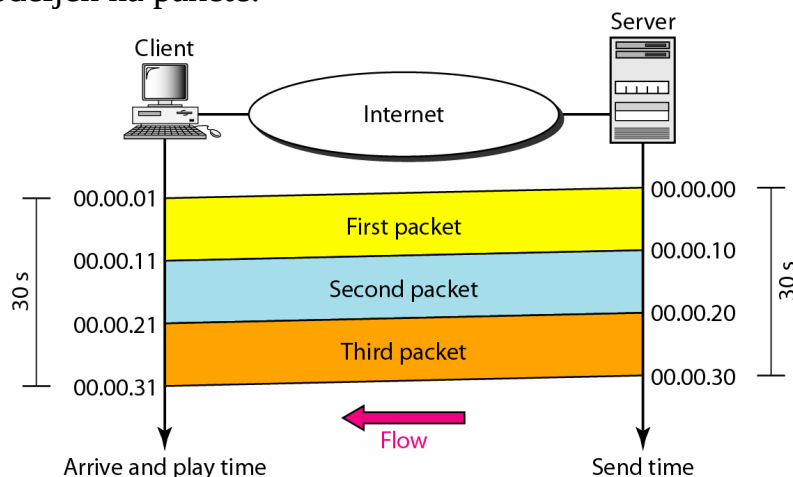
5. Interaktivna audio/video komunikacija u realnom vremenu.

U takozvanim real-time aplikacijama, ljudi komuniciraju međusobno na daljinu u realnom vremenu. Internet telefonija ili voice over IP (VOIP) je primer ovakve aplikacije. Video konferencije su još jedan primer vizuelne i oralne komunikacije.

5.1 Karakteristike

5.1.1 Odnos u vremenu

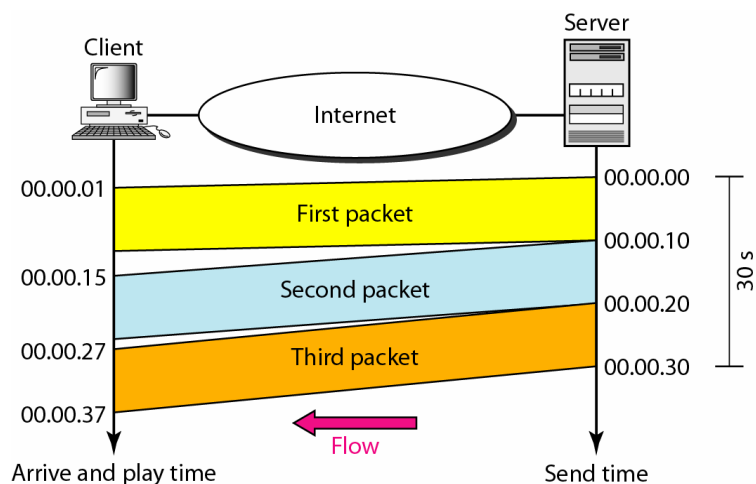
Podaci prenošeni u realnom vremenu preko mreže sa komutacijom paketa (*packet switching*) zahtevaju održavanje vremenskog odnosa između paketa jedne sesije. Napreimer, pretpostavimo da *real-time* server šalje stream video zapis. Signal je digitalizovan i podeljen na pakete.



SLIKA 14. Vremenski odnos.

Načinjena su tri paketa, svaki sa po deset sekundi video zapisa. Pretpostavimo i da je svakom paketu potrebna jedna sekunda da stigne do odredišta. Prijemnik će moći da pusti svaki od paketa po redosledu, jer je vreme potrebno da oni stignu kratko. Tako dobijamo iluziju prenosa u realnom vremenu. Vremenski odnos među paketima je očuvan.

Ali šta će se dogoditi ako paketi imaju različita kašnjenja? Na primer, prvi stiže u vremenu 00:01 (1 s kašnjenje), drugi u 00:15 (5 s kašnjenje), a treći u 00:27 (7 s kašnjenje). Ako prijemnik počne reprodukciju, drugi paket će mu biti potreban pre nego što on stigne, četiri sekunde kasnije. Pojaviće se praznina od četiri sekunde između segmenata. Ova pojava se u *streaming*-u naziva **jitter**.

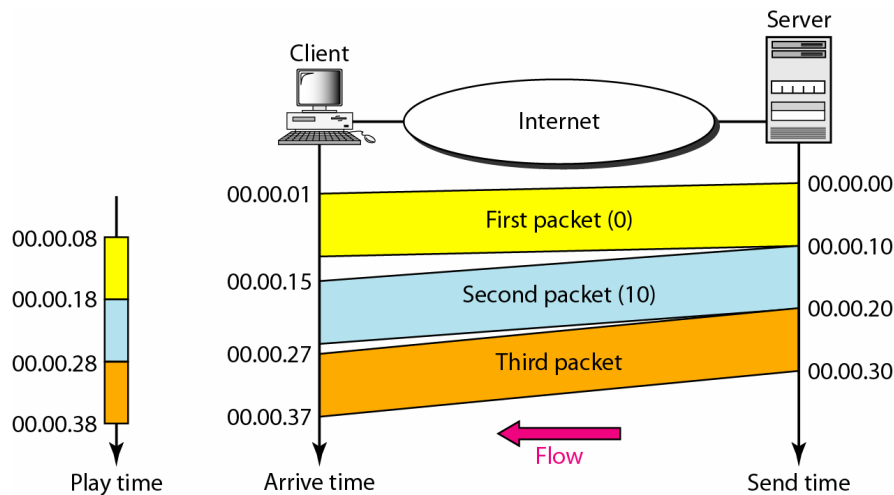


SLIKA 15. Pojava *jitter*-a.

5.1.2 *Timestamp* (vremenski kod)

Jedno rešenje je korišćenje *timestamp*-a. Ako svaki paket sadrži informaciju o vremenu slanja u odnosu na prvi (ili prethodni) paket, onda prijemnik može da doda ovo vreme vremenu od koga počinje reprodukciju. Drugim rečima, prijemnik zna kada treba da reprodukuje svaki pojedinačni paket. Zamislamo da je prvi paket imao *timestamp* 0, drugi 10, treći 20. Ako je reprodukcija počela od 00:08, sledeći paket će biti reprodukovani u 00:18, a poslednji u 00:28. Nema praznine između paketa.

Kako bi izbegli *jitter*, dodajemo vremenski kod (*timestamp*) paketima kako bi razdvojili vreme prijema od vremena reprodukcije.

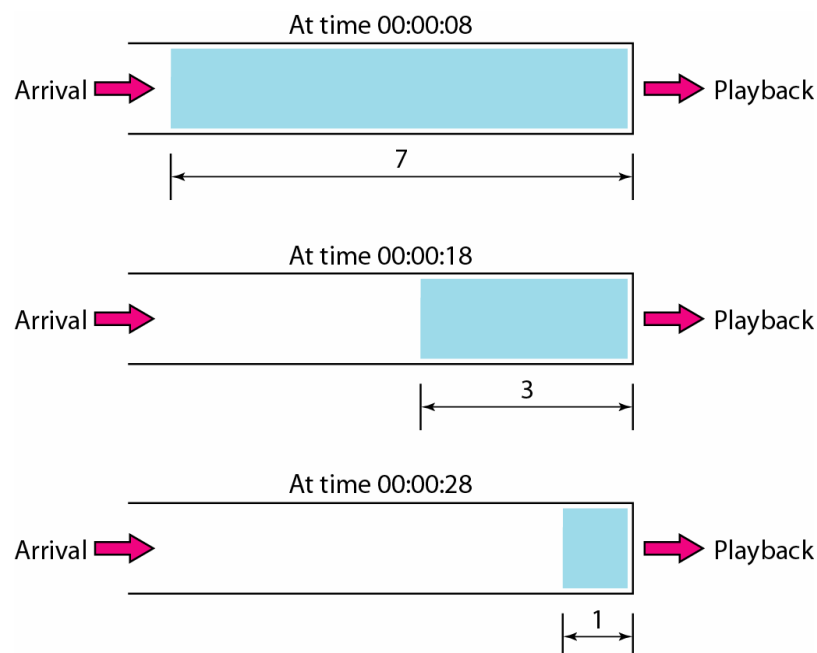


SLIKA 16. Upotreba *timestamp*-a.

5.1.3 Playback Buffer (Bafer reprodukcije)

Kako bi odvojili vreme prijema od vremena reprodukcije, potreban nam je bafer u kome će mo da skladištimo podatke dok nam ne budu potrebni. Kada sesija počne (prvi bit podataka stigne), prijemnik odloži reprodukciju dok ne dostigne određeni prag popunjenosti bafera. U predhodnom primeru, prvi bit dolazi u 00:01, prag je 7s, a reprodukcija počinje u 00:08. Prag se meri jedinicama vremena. Reprodukcija ne počinje dok se bafer ne napuni (dostigne prag).

Podaci pristižu u bafer različitim brzinama, ali se uvek čitaju istom brzinom. Količina podataka u baferu nije konstantna, ali dokle god *download* ide većom brzinom od reprodukcije, ne dolazi do *jitter*-a.



SLIKA 17. Bafer reprodukcije.

5.1.4 Sortiranje

Pored vremenskog odnosa paketa koji je definisan *timestamp*-om, za prenos u realnom vremenu potreban je još jedan važan parametar. Potreban nam je broj sekvence za svaki paket. Prijemnik ne može utvrditi da li je neki paket izgubljen na osnovu

timestampa. Naprimjer, neka su vrednosti *timestampa* 0, 10 i 20. Ako je drugi paket izgubljen, prijemnik ne može da zna za to, on će samo da reprodukuje treći paket 20 sekundi nakon početka, jer pretpostavlja da je to u stvari drugi paket. Dodavanjem broja sekvence ovaj problem se premošćava.

Bafer i broj sekvence su neophodni za reprodukciju materijala u realnom vremenu.

5.1.5 Multikasting

Multimedija ima glavnu ulogu u audio/video konferencijama. Saobraćaj na mreži može biti gust, a podaci se prenose *multicast* metodama. Konferencije zahtevaju dvosmernu komunikaciju između korisnika.

5.1.6 Prevodioci (*translators*)

Ponekada je emisiji u realnom vremenu potreban i prevodioc. Prevodioci su programi koji menjaju format signala, smanjuju njegov kvalitet kako bi oni bili poslati kroz mrežu male propusnosti. Na primer, prevodioc je neophodan pri prenosu stream-a od 5 Mbps kroz mrežu koja podržava samo 1 Mbps. Kako bi se održao stalan, real-time protok podataka, kvalitet signala se mora degradirati, tj. Mora se upotrebiti veći stepen kompresije.

5.1.7 Mešanje (*mixing*)

Ako postoje više izvora koji mogu slati podatke u isto vreme (kao u telekonferencijama), kroz mrežu se prenose više tokova podataka. Kako bi se više tokova (*stream-ova*) sklopilo u jedan, radi lakšeg prenosa, oni se spajaju (mešaju) u jedan. Mixer je uređaj koji matematički sabira dva signala iz različitih izvora kako bi generisao jedan signal.

5.1.8 Podrška transportnog sloja (OSI model)

Predhodno objašnjeni postupci implementiraju se u sloju aplikacije. Međutim, oni se tako često koriste da je njihova ugradnja u transportni sloj veoma poželjna.

TCP nije pogodan za interaktivni saobraćaj (*streaming*). Nema mogućnost *timestamp-a*, i ne podržava *multicasting*. Ali zato podržava numerisanje paketa (broj sekvence). Međutim glavni razlog zbog kojeg je TCP nepogodan za interaktivne aplikacije je njegova korekcija grešaka. Kod aplikacija u realnom vremenu ne možemo gubiti protok na retransmisiju izgubljenih ili oštećenih paketa. Ako dođe do gubitka ili oštećenja, paket mora biti ignorisan. I pored kompresije, postoji veliki stepen redundantnosti kod audio/video signala, da se takvi gubici ne registruju.

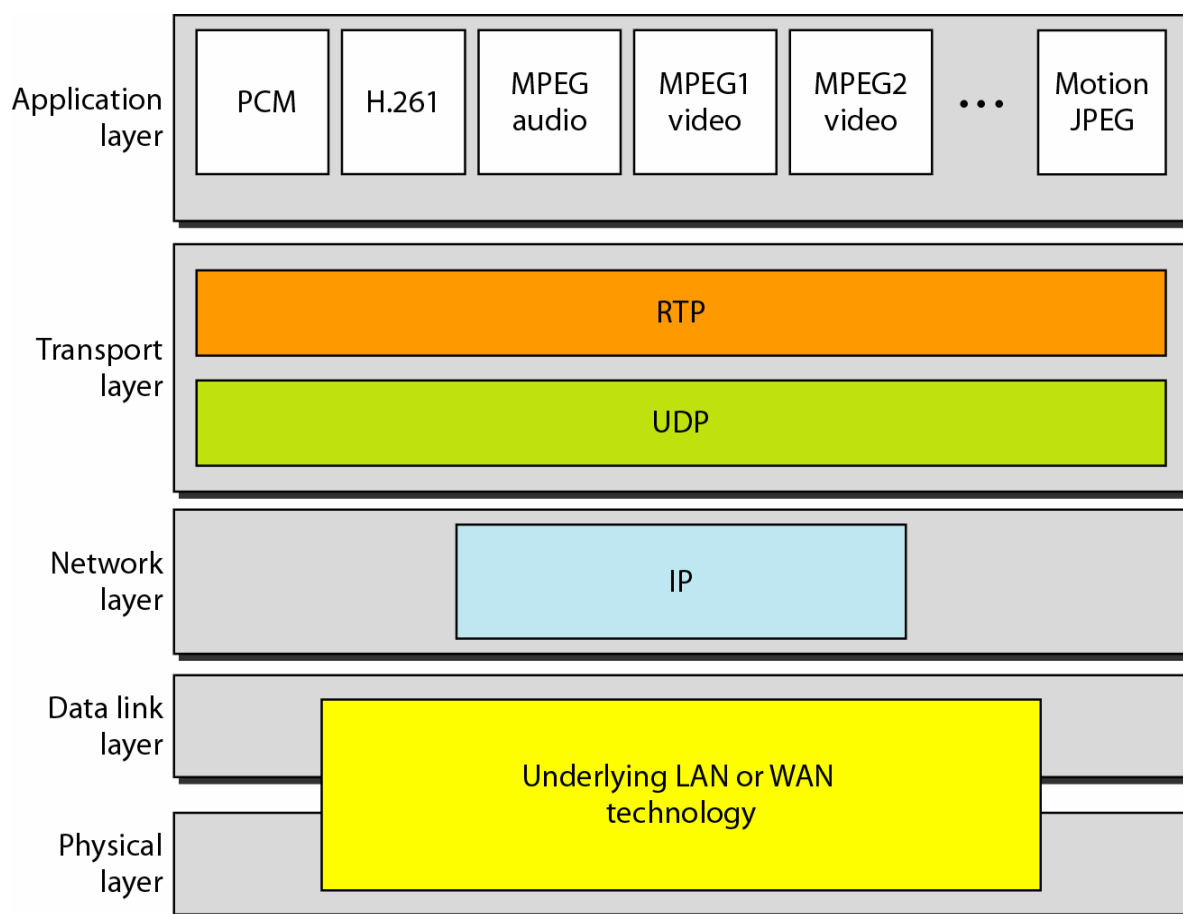
TCP, iako veoma napredan, nije pogodan za real-time aplikacije, jer je retransmisija nedozvoljena.

UDP je daleko bolje rešenje. On podržava *multicasting*, i nema retransmisije paketa. Ali UDP takođe ne podržava *timestamp*, numerisanje sekvenci, ili mešanje *stream*-ova. Novi protokol, *Real-time Transfer Protocol* (RTP) pruža usluge koje nedostaju UDP-u.

UDP je pogodniji za real-time aplikacije od TCP-a. Njegove nedostatke nadomeštavamo upotrebom RTP-a.

6. RTP (Real-time Transfer Protocol)

RTP ne poseduje mehanizme dostave paketa (*multicasting*, brojeve portova...), neophodan mu je UDP da bi funkcionisao. RTP se postavlja kao posrednik između UDP-a i aplikacije. Glavne osobine RTP-a su *timestamp*, numerisanje sekvenci i mešanje *stream*-ova.



SLIKA 18. Slojevi RTP-a.

6.1 Format RTP datagrama

Format RTP datagrama je veoma jednostavan, i dovoljno upošten da pokrije sve *real-time* aplikacije. Aplikacija kojoj su potrebne dodatne informacije dodaje ih na početak *payload*-a (korisnih podataka u paketu).

Ver	P	X	Contr. count	M	Payload type	Sequence number
Time stamp						
Synchronization source identifier						
Contributor identifier						
⋮						
Contributor identifier						

SLIKA 19. Oblik zaglavlja RTP datagrama.

- **VER** – 2 bita koja definišu verziju RTP-a.
- **P** – 1 bit, definiše da li je upotrebljen *padding* (dopunjavanje) paketa.
- **X** – 1 bit, ukazuje na dodatno zaglavlje između standardnog zaglavlja i data polja. Ako je vrednost 0, dodatnog zaglavlja nema.
- **Contributor count** – 4 bit-no polje, definiše broj učesnika (izvora). Pošto je u pitanju četvorobitno polje, maksimalan broj učesnika je 15.
- **M** – 1 bit, marker bit, koriste ga neke aplikacije za, na primer, označavanje kraja stream-a.
- **Payload type** – 7 bit-na informacija koje definiše tip podataka. Nekoliko tipova podataka su već predefinisana, ostali su stavljeni za buduću upotrebu.
- **Seq. Number** – 16 bit-a. Ovo polje se koristi za numeraciju RTP paketa, pošto UDP ne numeriše svoje pakete. Broj prvog paketa je izabran slučajno, svaki sledeći se uvećava za 1. Prijemnik ovo polje koristi za identifikaciju izgubljenih paketa.

Type	Application	Type	Application	Type	Application
0	PCMμ Audio	7	LPC audio	15	G728 audio
1	1016	8	PCMA audio	26	Motion JPEG
2	G721 audio	9	G722 audio	31	H.261
3	GSM audio	10–11	L16 audio	32	MPEG1 video
5–6	DV14 audio	14	MPEG audio	33	MPEG2 video

SLIKA 20. Tipovi podataka kod RTP-a.

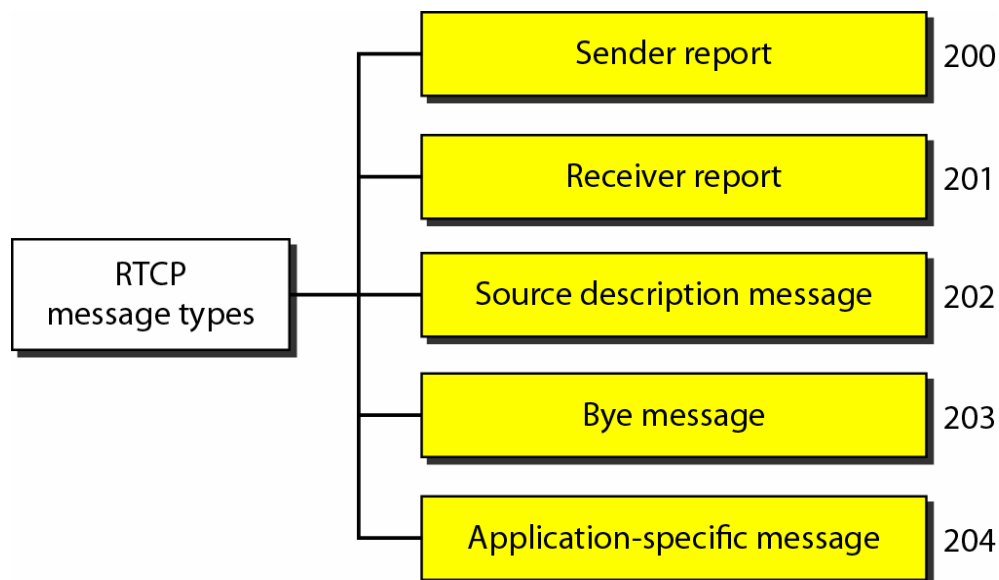
- **Timestamp** – 32 bit-no polje koje određuje vremenski odnos između paketa. Prvi paket je označen slučajno izabranim brojem. Za svaki sledeći paket vrednost je zbir predhodnih *timestamp*-ova i vremena slanja prvog sempla (uzorka). Vrednost vremenske jedinice zavisi od aplikacije. Na primer, audio aplikacije obično generišu segmente od 160 bajtova. Svaki sledeći *timestamp* je veći za 160.
- **Synchronization source identifier** – 32 bit-a. Ako postoji samo jedan izvor, ovaj broj ga definiše. Međutim, ako ima više izvora, mešač je izvor sinhronizacije (*sync. Source*), a ostali izvori su učesnici (kontributori). Ovo polje je slučajan broj generisan od strane izvora. Protokol definiše strategiju koja se primenjuje ukoliko se pojave konflikti.
- **Contributor identifier** – 32 bit-a. Svaki od ovih identifikatora definišu jednog učesnika (maksimalno 15). Kada ima više izvora podataka, mešač je izvor sinhronizacije, a ostali izvori su učesnici.

6.2 UDP port

Iako RTP funkcioniše na transportnom nivou, RTP paket nije direktno enkapsuliran u IP datagram. Umesto toga, RTP se tretira kao deo aplikacionog sloja, i enkapsulira se u UDP datagram. Međutim, za razliku od drugih aplikacija, nijedan dobro-poznati port (*well-known port*) nije dodeljen RTP-u. Port se dodeljuje po potrebi uz samo jedno ograničenje. Broj porta kod RTP-a mora biti paran. Sledeći port, neparan, se koristi za RTCP, *Real-time Transport Control Protocol*.

7. RTCP (Real-time Transport Control Protocol)

RTP omogućava slanje samo jedne vrste poruka, one koja šalje podatke od izvora do odredišta. U mnogim slučajevima je neophodno slanje i drugih poruka. Ove poruke upravljaju tokom podataka i kvalitetom zapisa, i omogućavaju povratnu vezu između prijemnika i predajnika (izvora). RTCP je projektovan baš za tu potrebu. RTCP ima pet tipa poruka.



SLIKA 21. Tipovi RTCP poruka.

7.1.1 Sender report (Izveštaj pošiljaoca)

Ova poruka se periodično šalje od strane aktivnih izvora u konferenciji i daje izveštaj o statistici svih poslatih i primljenih RTP paketa. Izveštaj sadrži apsolutni vremenski kod (*timestamp*), koji predstavlja broj sekundi koji je protekao od ponoći 1. januara 1970. godine. Ovaj apsolutni vremenski kod omogućava prijemniku da sinhronizuje različite RTP poruke. Narčito je važan ukoliko se šalju i audio i video signal (svaki koristi poseban relativan vremenski kod).

7.1.2 Receiver report (Izveštaj prijemnika)

Izveštaj prijemnika je za pasivne učesnike, one koji samo primaju, a ne šalju RTP datagrame. Obaveštava izvor o i druge prijemnike o kvalitetu servisa.

7.1.3 Source Description Message (Opis izvora)

Izvor periodično šalje svoj opis da bi dao dodatne informacije o sebi. Ovo uključuje ime, e-mail adresu...

7.1.4 Bye Message (Pozdravna poruka)

Izvor šalje ovu poruku kada zatvara tok podataka (*stream*). Omogućava izvoru da najavi svoje isključivanje iz konferencije. Iako drugi izvori mogu da otkriju odsustvo signala, ova poruka je važna za mešač *stream*-ova.

7.1.5 App. Specific Message (Poruka karakteristična za aplikaciju)

Ovo je paket podataka koji definiše upotrebu nove aplikacije (koja još nije definisana standardom). Omogućuje definiciju novog tipa poruke.

7.2 UDP Port

RTCP, kao i RTP, ne poseduje dobro-poznati port. Koristi privremeni port. UDP port za RTCP mora biti izabran odmah nakon izbora porta za RTP. Ovaj port nosi neparan broj.

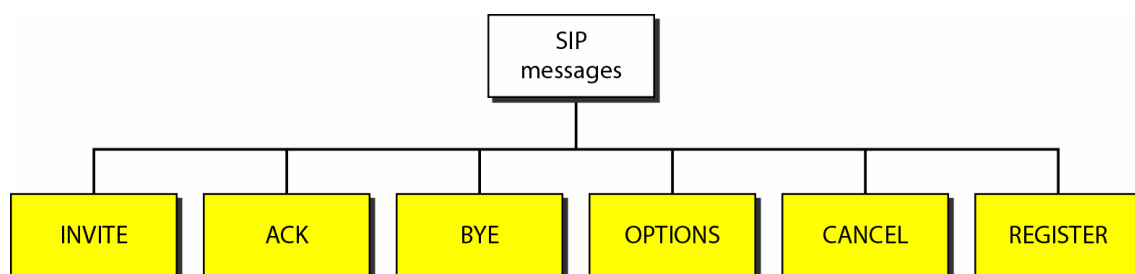
RTCP koristi port neparanog broja, koji sledi za brojem porta izabranog za RTP.

8.Voice Over IP (VOIP, Glas preko IP-a)

Ova aplikacija se često naziva i Internet telefonija. Ideja je da se Internet mreža koristi kao telefonska mreža sa još nekim dodatnim mogućnostima. Umesto komunikacije preko mreže sa direktnom vezom (*circuit switching*), ova komunikacija se odvija preko mreže sa komutacijom paketa. Postoje dva protokola koji su zaduženi za ovu svrhu: SIP i H.323.

8.1 SIP

Session Initiation Protocol (Protokol započinjanja sesije) je projektovan od strane IETF-a. Radi na sloju aplikacije, i on ostvaruje, kontroliše i prekida multimedijalnu sesiju (poziv). Može se koristiti za razgovor dvoje ili više učesnika, u bilo kojoj kombinaciji. SIP je projektovan tako da može da funkcioniše sa sledećim protokolima transportnog sloja: UDP, TCP i SCTP.



SLIKA 22. Tipovi poruka kod SIP-a

8.2 Poruke

SIP je u osnovi tekstualni protokol, kao i HTTP. Kao i HTTP, SIP komunicira tekstualnim porukama.

Svaka poruka ima zaglavlje i telo. Zaglavlje sadrži nekoliko linija koje opisuju strukturu poruke, mogućnosti pozivaoca, tip media i tako dalje.

Pozivaoc započinje sesiju INVITE porukom. Nakon što pozvani odgovori na poziv, pozivaoc šalje nazad ACK poruku, kojom potvrđuje uspešno ostvarenu vezu. BYE poruka prekida sesiju. OPTIONS poruka šalje upit mašini o njenim mogućnostima. CANCEL poruka prekida započeti proces inicijalizacije. REGISTER poruka pravi konekciju kada pozvani nije dostupan.

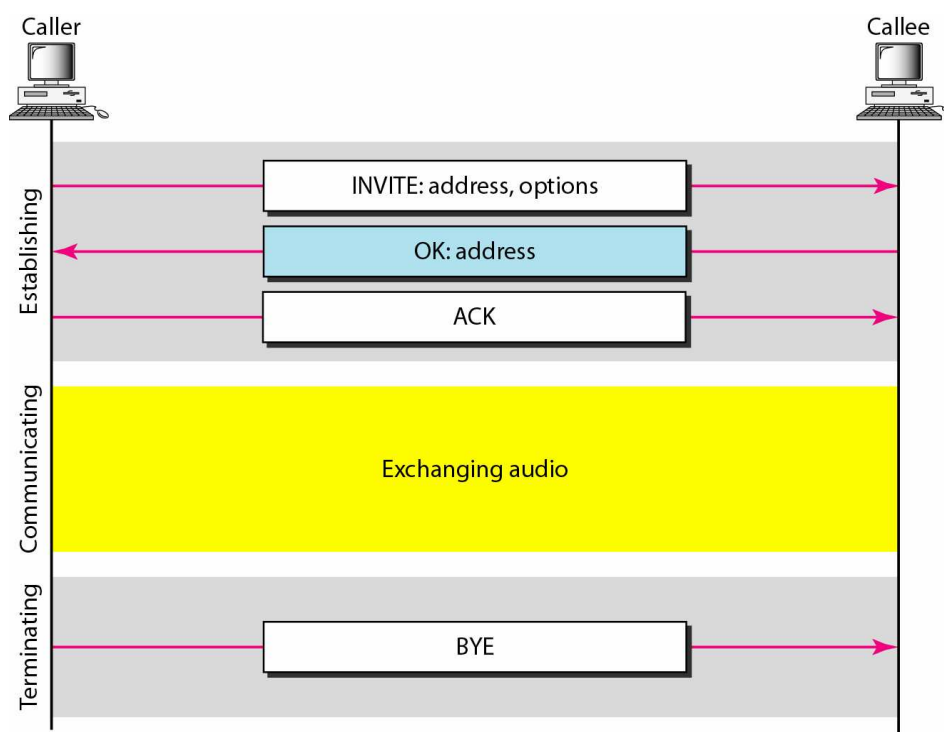
8.3 Adrese

Kod standardne telefonske veze, broj telefona identifikuje pošiljaoca, a drugi broj telefona primaoca. SIP je veoma fleksibilan. Kod SIP-a se za svrhu identifikacije može koristiti IP adresa, e-mail adresa, broj telefona ili nešto drugo. Međutim, adresa mora biti u SIP formatu, koji se naziva SIP šema.

8.4 Jednostavna sesija

Jednostavna SIP sesija se sastoji iz tri dela: povezivanje, komunikacija i zatvaranje sesije.

Povezivanje. Uspostavljanje veze SIP protokolom podrazumeva trostruki *handshake*. Onaj koji poziva šalje INVITE poruku, putem UDP-a, TCP-a ili SCTP-a kako bo započeo komunikaciju. Ako je pozivani voljan da prihvati poziv, šalje odgovor. Kako bi potvrdio prijem odgovora, pozivaoc vraća pozvanom ACK poruku.



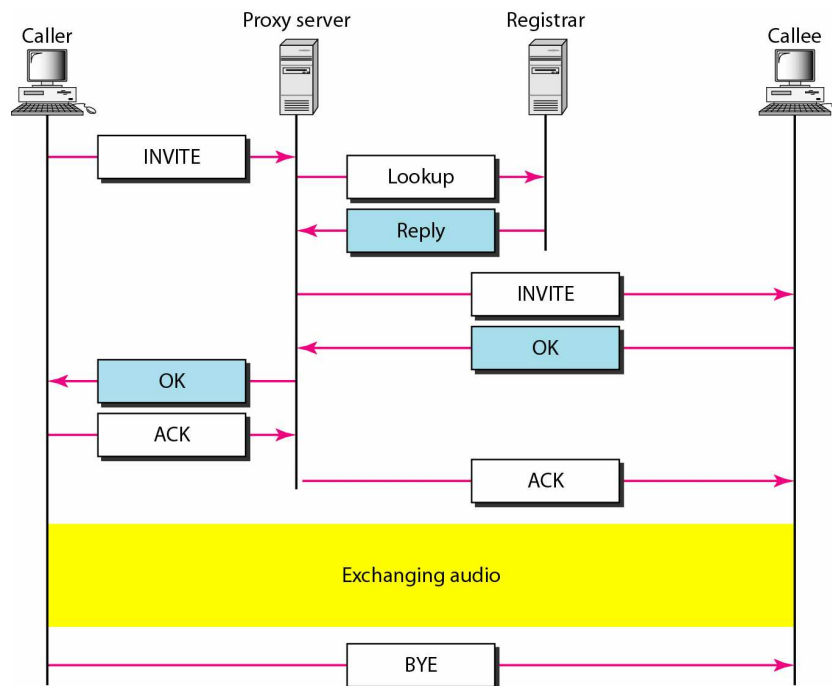
SLIKA 23. Primer jednostavne SIP sesije.

Komunikacija. Nakon što je sesija otvorena, komunikacija se obavlja putem dva privremena porta.

Zatvaranje sesije. Sesiju može prekinuti bilo koja strana, slanjem BYE poruke.

8.5 Praćenje sagovornika

Šta se dešava ako onaj koga pozivamo nije trenutno za svojim terminalom? Možda se nalazi na drugom terminalu, ili možda nema fiksnu IP adresu, ako koristi DHCP. SIP poseduje mehanizam (sličan DNS-u) koji pronalazi IP adresu terminala za kojim se trenutno nalazi pozivana strana. Za tu svrhu, SIP koristi bazu podataka (registar). SIP definiše server koji će se koristiti kao registar. Server uvek zna IP adrese korisnika u registru.



SLIKA 24. Praćenje sagovornika.

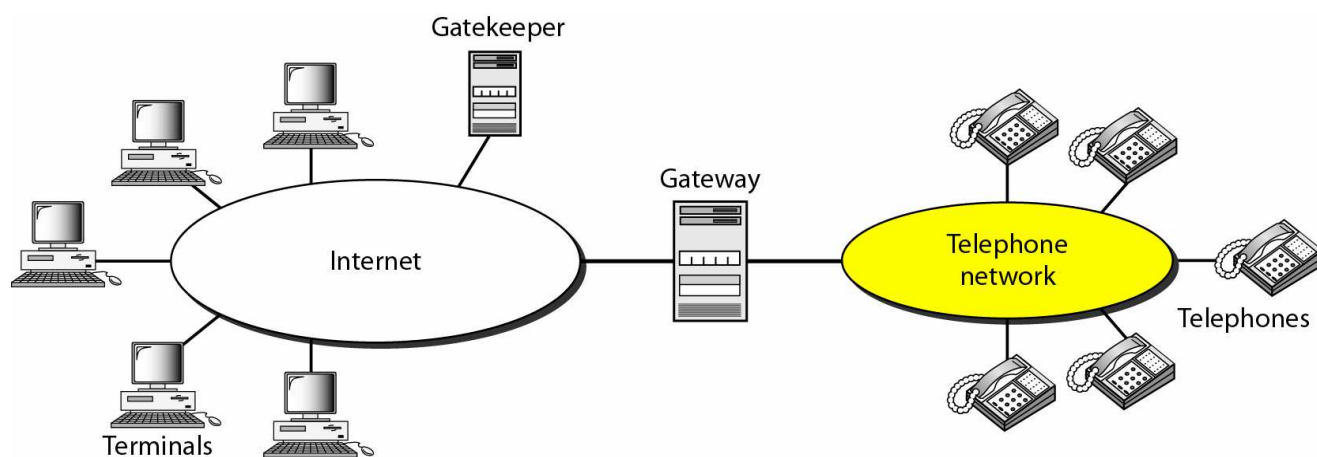
Kada želimo da pozovemo nekog, možemo upotrebiti e-mail adresu umesto IP adrese za slanje INVITE poruke. Poruka stiže do *proxy* servera. On dalje šalje *lookup* (upitnu) poruku, koja nije deo SIP-a, nekom od registar servera, koji ima adresu pozivanog. Kada *proxy* server dobije odgovor od registra, on ubacuje novootkrivenu IP adresu u INVITE poruku, koju prosleđuje dalje SIP-u.

8.6 H.323

H.323 je standard koji je projektovao ITU da bi omogućio telefonima iz javne telefonske mreže pristup računarima povezanim na Internet (po ovom standardu nazivaju se terminali).

Gateway, ili kapija povezuje Internet sa telefonskom mrežom. U principu, *gateway* je uređaj sa pet ISO-OSI slojeva, i može da prevodi poruke iz jednog protokol steka u drugi. On pretvara telefonski signal u Internet poruku. *Gatekeeper* (vratar) je server na lokalnoj mreži koji igra ulogu registra.

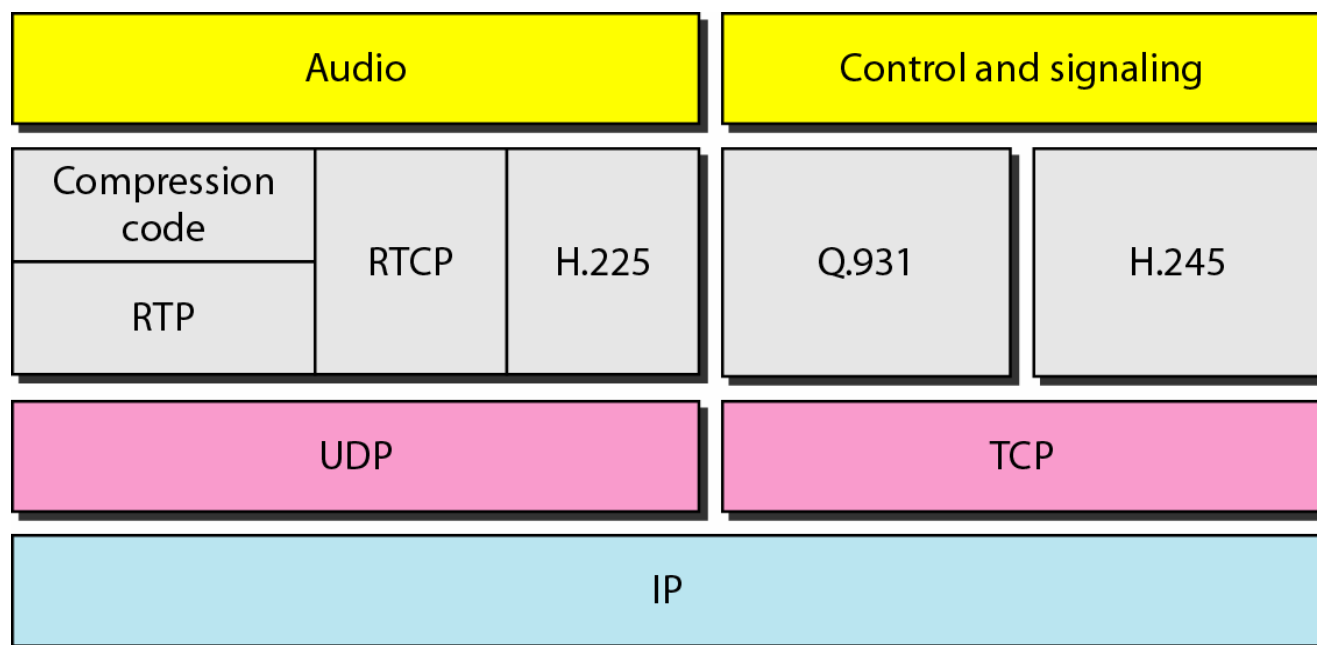
H.323 koristi više različitih protokola za uspostavljanje i održavanje glasovne (ili vizuelne) komunikacije.



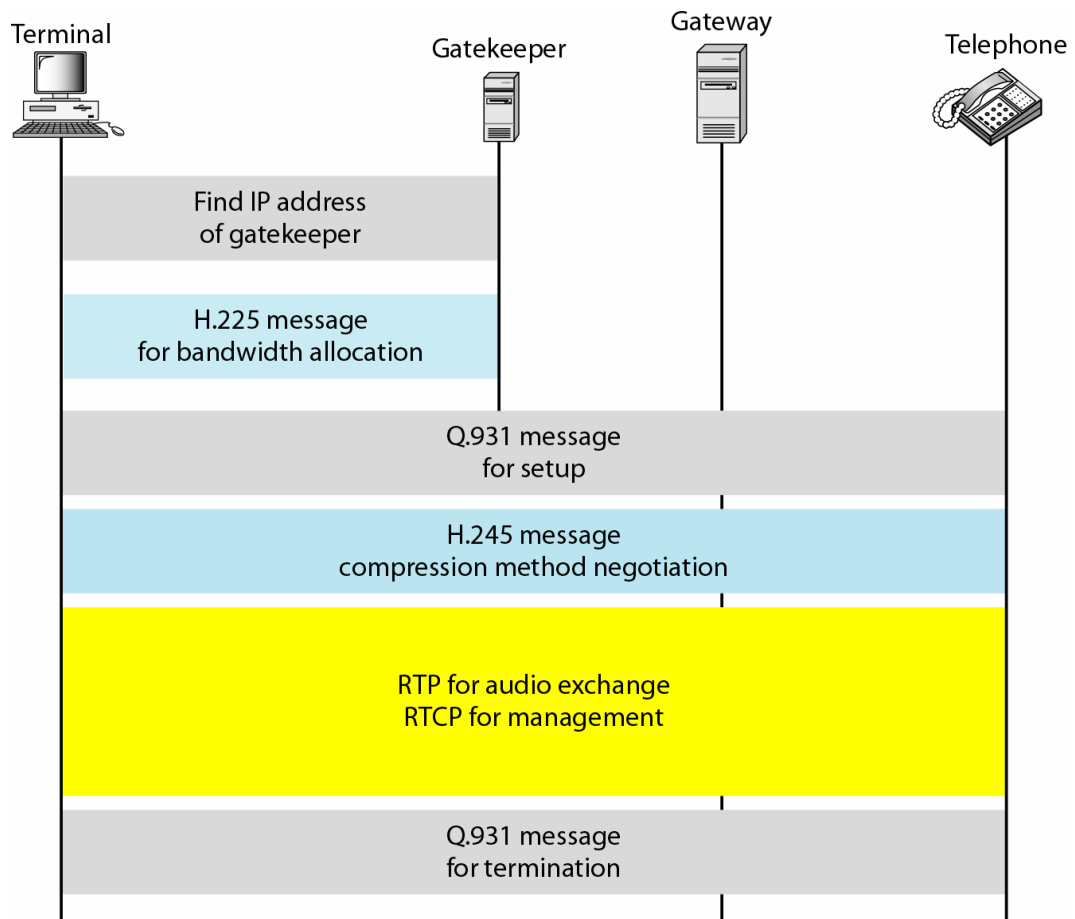
SLIKA 25. Arhitektura H.323.

H.323 koristi G.71 ili G.723.1 algoritme za kompresiju. Koristi protokol po imenu H.245, koji omogućava strankama da pregovaraju o metodi kompresije. Protokol q.931 se koristi za uspostavljanje i prekidanje sesija. Protokol H.255, ili RAS (*Registration/Admistration/Status*) se koristi za ažuriranje registra na *gatekeeper* računar.

Na sledećoj strani dat je jednostavan primer H.323 komunikacije



SLIKA 26. Protokoli koje koristi H.323.



SLIKA 27. Primer H.323 komunikacije

1. Terminal šalje *broadcast* poruku *gatekeeper* serveru. *Gatekeeper* odgovara svojom IP adresom.
2. Terminal i *gatekeeper* komuniciraju pomoću H.225 protokola, kako bi utvrdili brzinu mreže, i uskladili protok podataka.
3. Terminal, *gatekeeper*, *gateway* i telefon komuniciraju koristeći protokol Q.931 da bi otvorili sesiju.
4. Terminal, *gatekeeper*, *gateway* i telefon komuniciraju koristeći protokol H.245 da bi se "dogovorili" oko algoritma za kompresiju.
5. Terminal, *gateway* i telefon vrše *streaming* zvuka koristeći RTP pod nadzorom RTCP-a.
6. Terminal, *gatekeeper*, *gateway* i telefon koriste Q.931 protokol kako bi prekinuli vezu.

Literatura:

1. *Data Communications and Networking, fourth edition*
(Copyright © The McGraw-Hill Companies, Inc.)
2. *Wikipedia* (www.wikipedia.com)

Gotovi seminarski, maturski, maturalni i diplomski radovi iz raznih oblasti, lektire , puškice, tutorijali, referati - specijalizovan tim za usluge visokokvalitetnog pisanja, istraživanja i obradu teksta za kompletan region Balkana.

Posetite nas na sajtovima ispod:

WWW.MATURSKIRADOVI.NET

WWW.SEMINARSKIRAD.ORG

WWW.MATURSKI.NET

WWW.MATURSKI.ORG

WWW.SEMINARSKIRAD.INFO

Dostupni smo Vam 24h 365 dana u godini.

Za gotove verzije rada obratiti se na mail:

maturskiradovi.net@gmail.com

061/ 11-00-105

Seminarski, diplomski, maturski radovi, prevodi na engleski i eseji...