KOMUTACIONI SISTEMI
– Poglavlje 8 –

8 Paketizacija govornog signala

U prethodnim poglavljima je obrađena telefonska mreža kao tipičan predstavnik mreže bazirane na komutaciji kola. Takođe, u više navrata je navedeno da su danas dominantne mreže bazirane na paketskoj komutaciji. U ovom i narednim poglavljima će biti objašnjen prenos govornih signala u mrežama baziranim na paketskoj komutaciji kao i osnovni principi mešovitog rada mreža baziranih na komutaciji kola i komutaciji paketa.

Da bi se govorni signal preneo preko paketske mreže neophodno je da se on smesti u pakete, pa je logično da se pre smeštanja u pakete govor mora digitalizovati. Kao što smo videli u slučaju telefonskih mreža, u digitalnim telefonskim centralama se vrši digitalizacija govornog signala po A zakonu kompresije. U slučaju ISDN mreže digitalizacija se vršila već u samom telefonskom aparatu (ISDN telefon), odnosno na korisničkoj strani. Za govorni signal koji je digitalizovan upotrebom A zakona kompresije se kaže da je on kodiran G.711 koderom, jer (ITU-T) preporuka G.711 definiše A i μ zakone kompresije. Pored G.711 kodera, mogu se koristiti i drugi koderi poput G.726, G.729 i dr. Kada je govorni signal digitalizovan, tada se binarna digitalna predstava govornog signala može ubaciti u paket (paketizacija govornog signala) i to u korisni deo paketa (*payload*) - ovaj proces se vrši na aplikacionom sloju. Da bi se paketizovani govorni signal preneo ispravno na odredište i tamo dekodovao, neophodno je koristiti usluge nižih slojeva (transportni, mrežni, sloj link podataka, fizički sloj) pa je potrebno definisati i adekvatne protokole, kao i koristiti već postojeće protokole koji će omogućiti ispravan prenos paketizovanog govora kroz mrežu baziranu na komutaciji paketa.

Ovo poglavlje će da pokrije nekoliko osnovnih oblasti vezanih za prenos govornog signala preko mreže bazirane na komutaciji paketa. Na početku poglavlja će biti objašnjeni osnovni principi digitalizacije govornog signala i biće dat pregled najpoznatijih kodera govornog signala. A zakon kompresije, odnosno G.711 koder će biti detaljno objašnjen. Potom će biti dat kratak pregled najpoznatijih protokola koji se koriste prilikom prenosa paketizovanih govornih signala. Postoji više različitih tehnologija baziranih na komutaciji paketa, ali u okviru ovih skripti fokus će biti na prenosu govora preko Internet mreže, kao i eternet LAN mreže koje su najpoznatije paketske tehnologije. Na kraju će biti predstavljen E model za ocenu kvaliteta prenosa govornog signala. Mnogo faktora utiče na kvalitet govorne komunikacije u paketskim mrežama, pa je razvijen E model za kvantifikaciju uticaja svih faktora. Na osnovu E modela može se planirati i projektovati govorna komunikacija preko paketske mreže.

8.1. Koderi govornog signala

Govorni signal tipično zauzima frekvencijski opseg do 10kHz, ali se najveći deo snage govornog signala nalazi u opsegu 300-3400Hz. Otuda se govorni signal tipično propušta kroz NF filtar čija je gornja granica 3400Hz. Druga osobina govornog signala je da vokali (samoglasnici) nose snagu govornog signala, a suglasnici razumljivost govornog signala, pri čemu su suglasnici manje snage. Pošto digitalizacija inherentno unosi šum kvantizacije, poželjno je da ovaj šum bude mali kada se digitalizuju suglasnici da bi se dobila veoma dobra razumljivost govornog signala, dok kod vokala šum može biti i veći jer oni nose snagu govornog signala, a imaju manji uticaj na razumljivost signala. Postoje i druge osobenosti govornog signala. Sve osobenosti govornog signala se u većoj ili manjoj meri uzimaju prilikom digitalnog kodiranja govornog signala. Kao rezultat koderi mogu da proizvedu kompaktniju digitalnu predstavu govornog

signala koja ima manji protok, ali i manji kvalitet reprodukcije, i obrnuto. Koji koder će se koristiti zavisi od definisanog nivoa kvaliteta govornog signala, odnosno komunikacije, broja istovremenih komunikacija koji se želi ostvariti, dozvoljenog protoka za jednu govornu komunikaciju (razgovor) i sl. Koderi se često nazivaju i kompresorima jer tipično kompresuju govorni signal da bi ostvarili niži protok, ali nauštrb kvaliteta govornog signala. G.711 koder daje najbolji kvalitet govornog signala i jedino se za njega govori da daje nekomprimovani digitalizovani govorni signal. Kodere (kompresore) možemo podeliti u dve grupe:

- Talasni koderi
- Parametarski koderi

Talasni koderi se zasnivaju na ideji da se govorni signal kodira tako da se na prijemu može rekonstruisati originalni talasni oblik govornog signala. Pošto se rekonstruiše originalni talasni oblik signala, ovi koderi se u principu mogu primeniti i na druge tipove signala, a ne samo na govorne signale. Ovi koderi tipično omogućavaju veći kvalitet, ali zahtevaju i veće protoke. Rekonstrukcija talasnog oblika može biti rekonstrukcija talasnog oblika vremenskog domena ili frekvencijskog domena. U slučaju vremenskog domena rekonstruiše se originalan vremenski oblik signala u skladu sa principima zakona o odabiranju, pri čemu su najpoznatiji takvi koderi G.711 i G.726 koderi. S druge strane, ideja rekonstrukcije talasnog oblika u frekvencijskom domenu se zasniva na ideji da se frekvencijski spektar signala podeli na podopsege koji se zasebno kodiraju, pri čemu se važniji podopsezi kodiraju preciznije sa više bita, a manje važni podopsezi se kodiraju sa manje bita tj. manje precizno. Drugi metod za talasno kodiranje u frekvencijskom domenu je primena brzih transformacija poput diskretne kosinusne transformacije za predstavljanje odsečka govornog signala u vidu velikog broja frekvencijskih opsega, pri čemu se vrši adaptivno kodiranje koeficijenata koji opisuju spektralne karakteristike odsečka govornog signala. Koeficijenti se kodiraju adaptivno tako da se preciznije (sa više bita) kodiraju važniji koeficijenti, a manje precizno (sa manje bita) manje važni koeficijenti. Napomenimo još da su talasni koderi jednostavni za implementaciju i brzo vrše kodiranje (procesiranje govornog signala nije kompleksno kod talasnih kodera), pri čemu se za kodiranje govornog signala češće koriste talasni koderi koji rekonstruišu talasni oblik vremenskog domena.

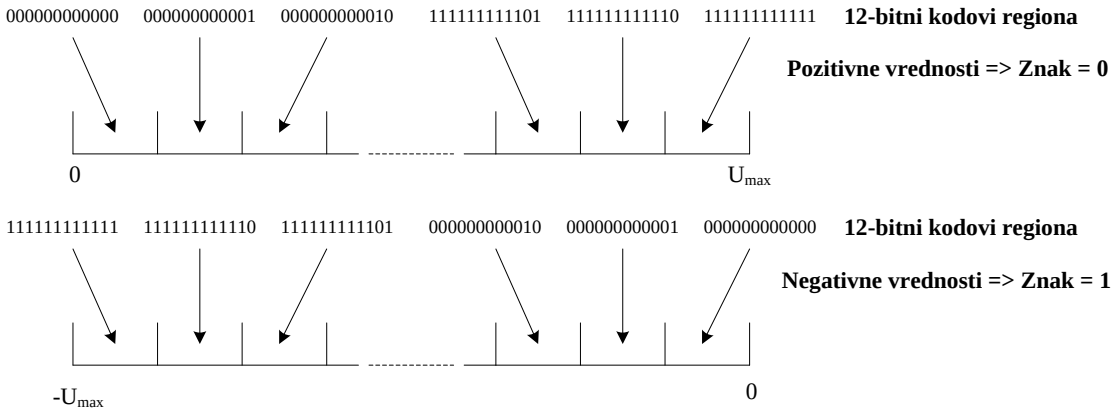
Parametarski koderi se zasnivaju na ideji da se kodira govorni signal, odnosno za razliku od talasnih kodera uzimaju se u obzir sve osobenosti govornog signala, pa parametarski koderi nisu upotrebljivi za ostale tipove signala. Stoga, parametarski koderi vrše modelovanje govornog trakta korisnika i prenose vrednosti karakterističnih parametara govornog signala koji se dobijaju analizom govornog signala. Na prijemu se na osnovu tih prenetih parametara vrši sinteza govornog signala. Međutim, cilj sinteze nije rekonstruisanje originalnog talasnog oblika govornog signala, već dobijanje razumljivog govornog signala koji slušalac može da razume. Otuda se ovi sistemi još nazivaju sistemi analize i sinteze, jer se na predaji vrši analiza govornog signala radi određivanja karakterističnih parametara govornog signala koji se potom prenose, a onda se na prijemu na osnovu tih parametara sintetiše govorni signal. Ovi koderi omogućavaju bolju kompresiju (zahtevaju manji protok), ali imaju i niži kvalitet. Niži kvalitet potiče iz činjenice da se govorni signal na prijemu sintetiše pa slušaocu takav govorni signal deluje veštački tj. neprirodan i bez emocija čime slušalac ima subjektivan osećaj nižeg kvaliteta govornog signala i pored činjenice da je govorni signal u potpunosti razumljiv. Govorni signal je potpuno razumljiv jer je glavni cilj parametarskih kodera da precizno prenesu parametre koji se odnose na razumljivost govora. Takođe, parametarski koderi su procesorski zahtevni pa je

implementacija ovih kodera kompleksnija nego u slučaju talasnih kodera. Parametarski koderi se još nazivaju i vokoderi (*vocoder*). U najvećem broju parametarskih kodera se vrši simulacija dekodera u okviru kodera radi što bolje procene parametara govornog signala jer se odmah u predajniku zna kako će izgledati sintetizovan govor čime je omogućeno preciznije određivanje parametara govornog signala. Napomenimo da postoji i tzv. grupa hibridnih kodera koji kombinuju pristupe talasnih i parametarskih kodera tako da ostvare niži protok i viši kvalitet govornog signala (protok je nešto veći od protoka parametarskih kodera, a kvalitet je nešto niži od talasnih kodera).

U literaturi su definisane i druge podele kodera govornog signala. Cilj ovih skripti je da pruži samo osnovni uvid u oblast kodera govornog signala bez zalaženja u detalje. Čitaoci koje interesuje više detalja iz ove oblasti mogu da potraže odgovarajuću literaturu, pre svega na Internetu gde postoji obilje dostupne i besplatne literature iz ove oblasti.

8.1.1. G.711 koder

G.711 koder je definisan u ITU-T preporuci G.711. Za ovaj koder se kaže da vrši PCM (*Pulse Code Modulation*) modulaciju govornog signala. Unutar G.711 preporuke su definisani A i μ zakoni kompresije, a u okviru ove sekcije ćemo obraditi A zakon kompresije koji se koristi u Evropi. Pre obrade u G.711 koderu, govorni signal se filtrira (NF filtar) i odabire frekvencijom od 8kHz po Nikvistovom kriterijumu jer je za prenos govornog signala predviđen frekvencijski opseg od 4kHz. Naime, govornom signalu iz opsega 300-3400Hz se pridodaju zaštitni opsezi 0-300Hz i 3400-4000Hz. Odmerci govornog signala (praktično impulsi govornog signala) se kodiraju sa 8 bita, pa otuda i naziv PCM - impulsna kodna modulacija. A zakon kompresije podrazumeva kodiranje odmeraka sa po 8 bita i kao rezultat se dobija digitalizovani govorni signal protoka 64kb/s (8 bita na svakih 125μs). G.711 spada u grupu talasnih kodera.



Slika 8.1.1.1. Princip uniformnog PCM kodiranja

Pored kompresije, G.711 koder koristi i pojam uniformne PCM modulacije koja podrazumeva korišćenje 13 bita (ili 14 bita, u Evropi je u pitanju 13 bita pa ćemo u nastavku teksta podrazumevati ovu vrednost). Naime, definiše se maksimalan apsolutni nivo govornog signala (nivo može imati i pozitivnu i negativnu vrednost) i potom se u slučaju uniformne PCM modulacije čitav opseg nivoa govornog signala izdeli na regione iste veličine kao na slici 8.1.1.1. Svaki region se kodira sa 12 bita, dok 13. bit definiše znak regiona (pozitivan ili negativan). Govorni signal se kodira tako što se odredi u koji region upada i potom se uzima dvanestobitni kod regiona ispred kojega se stavlja znak regiona (koji u stvari odgovara znaku nivoa govornog signala). Uniforman pristup isto tretira sve regione, odnosno šum kvantizacije je u proseku uvek

isti nezavisno od nivoa signala, ali ovaj pristup nije optimalan jer, kao što je već navedeno, niske nivoe govornog signala treba finije kvantizovati (tj. praviti manji šum kvantizacije), nego visoke nivoe govornog signala. Iz tog razloga se koristi A zakon kompresije koji uviđa logaritamsku prirodu govornog signala.

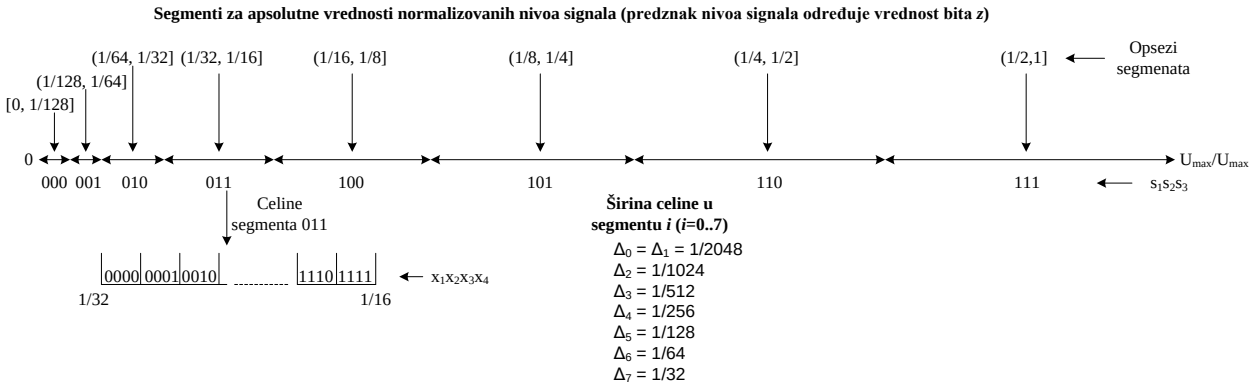
A zakon kompresije $F(x)$ govornog odmerka se definiše sledećim izrazom:

$$F(x)=\begin{cases}\operatorname{sgn}(x)\cdot\dfrac{A|x|}{1+\ln A}, & |x|<\dfrac{1}{A}\\[2ex] \operatorname{sgn}(x)\cdot\dfrac{1+\ln(A|x|)}{1+\ln A}, & \dfrac{1}{A}\le|x|\le 1\end{cases} \tag{8.1.1.1}$$

gde je x normalizovana vrednost odmerka govornog signala, A parametar kompresije koji u Evropi iznosi 87.7 (moguće je koristiti i vrednost 87.6). Na prijemu se mora izvršiti dekompresija koja je definisana sa:

$$F^{-1}(y)=\begin{cases}\operatorname{sgn}(y)\cdot\dfrac{|y|(1+\ln A)}{A}, & |y|<\dfrac{1}{1+\ln A}\\[2ex] \operatorname{sgn}(y)\cdot\dfrac{e^{|y|(1+\ln A)-1}}{A}, & \dfrac{1}{1+\ln A}\le|y|\le 1\end{cases} \tag{8.1.1.2}$$

Naravno, implementirati rad sa ovim logaritamskim krivama nije efikasno rešenje, pa se navedena kriva kompresije predstavlja u vidu osam linearnih segmenata za oba znaka kao što je prikazano na slici 8.1.1.2.



Slika 8.1.1.2. A zakon kompresije

Svaki segment se deli na 16 celina iste veličine. Kodirani govorni odmerak ima tri dela informacije. Znak z definiše predznak govornog odmerka - pozitivan (0) ili negativan (1). Tri bita $s_1s_2s_3$ definišu u koji segment upada govorni odmerak, dok četiri bita $x_1x_2x_3x_4$ definišu u koju celinu dotičnog segmenta upada govorni odmerak. Kao što vidimo sa slike 8.1.1.2 širine segmenata rastu sa povećanjem nivoa signala, a samim tim i širine celina koje odgovaraju segmentima. To znači da će šum kvantizacije biti manji u regionima koji odgovaraju nižim nivoima govornog odmerka, a veći u regionima koji odgovaraju višim nivoima govornog odmerka što je u skladu sa karakteristikama govornog signala koje smo ranije naveli.

Kada se na prijemu dobije kodirani odmerak, dekodovanje (dekompresija) se vrši tako što dekodovani odmerak dobije nivo koji odgovara sredini celine kojoj pripada kodirani odmerak jer se time dobija minimalni šum kvantizacije (minimizuje se srednjekvadratna greška).

Napomenimo da se za govorni signal kodiran G.711 koderom kaže da je nekomprimovan govorni signal iako se koristi A (μ) zakon kompresije, a razlog je što G.711 koder daje najverniju reprodukciju govornog signala, ali pri tome zahteva i najveći protok od 64kb/s za govorni signal.

Tabela 8.1.1.1. Konverzija odmerka kodiranog uniformnim PCM kodiranjem u odmerak kodiran A zakonom kompresije

Uniformno PCM kodiranje	A zakon kompresije
$zx_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12}$	$z\ s_1s_2s_3\ x_1x_2x_3x_4$
z 1 **abcd** *xxxxxxx*	z 111 **abcd**
z 01 **abcd** *xxxxxx*	z 110 **abcd**
z 001 **abcd** *xxxxx*	z 101 **abcd**
z 0001 **abcd** *xxxx*	z 100 **abcd**
z 00001 **abcd** *xxx*	z 011 **abcd**
z 000001 **abcd** *xx*	z 010 **abcd**
z 0000001 **abcd** *x*	z 001 **abcd**
z 0000000 **abcd** *x*	z 000 **abcd**

Tabela 8.1.1.2. Konverzija odmerka kodiranog A zakonom kompresije u odmerak kodiran uniformnim PCM kodiranjem

A zakon kompresije	Uniformno PCM kodiranje
$z\ s_1s_2s_3\ x_1x_2x_3x_4$	$zx_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12}$
z 111 **abcd**	z 1 **abcd** 1000000
z 110 **abcd**	z 01 **abcd** 100000
z 101 **abcd**	z 001 **abcd** 10000
z 100 **abcd**	z 0001 **abcd** 1000
z 011 **abcd**	z 00001 **abcd** 100
z 010 **abcd**	z 000001 **abcd** 10
z 001 **abcd**	z 0000001 **abcd** 1
z 000 **abcd**	z 0000000 **abcd** 1

Veza između A zakona kompresije i uniformnog PCM kodiranja je data tabelama 8.1.1.1 i 8.1.1.2 koje prikazuju konverziju u oba smera (iz formata A zakona kompresije u uniformni PCM format i obrnuto). Sa *x* su označeni biti čija vrednost nije bitna prilikom kompresije i oni se javljaju iz prostog razloga što je ukupan broj regiona (celina) veći u slučaju uniformnog PCM kodiranja. Iz istog razloga se prilikom konverzije iz formata A zakona kompresije u uniformni PCM format dekodovanje vrši na sredinu opsega koji u uniformnom PCM formatu odgovara dotičnoj celini A zakona kompresije (cilj je minimizacija šuma kvantizacije). Ove jednostavne konverzije omogućavaju da primenimo uniformno kodiranje analognog odmerka ukoliko nam takva implementacija odgovara, a potom da jednostavno izvršimo konverziju u format A zakona kompresije po principima prikazanim u tabeli 8.1.1.1. Na prijemu se takođe može izvršiti konverzija iz A zakona kompresije u format uniformnog PCM kodiranja po principima prikazanim u tabeli 8.1.1.2, a potom možemo da izvršimo D/A konverziju u analogni dekodovani govorni odmerak ako nam odgovara da D/A konverziju vršimo iz uniformnog PCM formata. Takođe, neki koderi govornog signala (na primer, G.726) zahtevaju na ulazu uniformni PCM

format, pa je u slučaju konverzije iz G.711 kodera u takve kodere potrebno koristiti i konverzije definisane u tabelama 8.1.1.1 i 8.1.1.2.

Primer 1. Predstavimo konkretnim primerom rad G.711 kodera. Pretpostavimo da je maksimalan nivo govornog odmerka 100mV. Potrebno je kodirati govorni odmerak čiji je nivo 25mV, kao i odmerak čiji je nivo -1mV.

Prvi korak je normalizovanje datih vrednosti odmeraka tako što ih delimo sa maksimalnim nivoom (100mV). Normalizovane vrednosti su 0.25 i -0.01. Na osnovu slike 8.1.1.2 možemo uočiti da 0.25 pada u segment 5 ($s_1s_2s_3$ = 101), a vrednost 0.01 pada u segment 1 ($s_1s_2s_3$ = 001). Ostaje još jedino da odredimo vrednosti $x_1x_2x_3x_4$ za oba normalizovana odmerka. Utvrđivanje vrednosti $x_1x_2x_3x_4$ se vrši primenom sledeće formule:

$$x_{1..4} = \lfloor (n - dg)/\Delta \rfloor \tag{8.1.1.3}$$

gde je *n* normalizovana vrednost odmerka, *dg* je donja granica segmenta u koji je upao govorni odmerak, a Δ ješirina jedne celine u segmentu kojem pripada govorni odmerak. U specijalnom slučaju kada je normalizovani govorni odmerak jednak gornjoj granici segmenta, odmerak se kodira sa $x_1x_2x_3x_4$ = 1111. U slučaju odmerka 0.25 imamo baš takvu situaciju pa se on kodira sa $x_1x_2x_3x_4$ = 1111. U slučaju odmerka 0.01 imamo:

$$x_{1..4} = \lfloor (0.01 - 1/128)/(1/2048) \rfloor = \lfloor 4.48 \rfloor = 4 \tag{8.1.1.4}$$

pa odmerak 0.01 kodiramo sa $x_1x_2x_3x_4$ = 0100.

Kod za odmerke dobijen A zakonom kompresije je 01011111 za nivo 25mV, odnosno 10010100 za -1mV, podrazumevajući G.711 format $zs_1s_2s_3x_1x_2x_3x_4$.

Izvršimo dekompresiju ova dva kodirana odmerka da bi videli odstupanja od originalnih nivoa odmeraka. Dekompresija se vrši koristeći sledeći princip za određivanje normalizovanog nivoa signala:

$$n_d = dg + x_{1..4} \cdot \Delta + \Delta/2 \tag{8.1.1.5}$$

gde je n_d normalizovana vrednost dekodovanog odmerka, *dg* je donja granica segmenta kojem pripada govorni odmerak, Δ je širina jedne celine u segmentu kojem pripada govorni odmerak, a $x_{1..4}$ je vrednost $x_1x_2x_3x_4$. Izraz (8.1.1.5) u stvari definiše da se dekompresovanom govornom odmerku dodeljuje nivo koji odgovara sredini celine kojoj pripada govorni odmerak. Razlog za ovu odluku je pretpostavka uniformne raspodele govornog signala sa stanovišta celine, pa je odabirom sredine celine za nivo dekompresovanog odmerka postignuta minimalna srednjekvadratna greška, odnosno minimalan šum kvantizacije na nivou celine. Primenom (8.1.1.5) dobijamo da se odmerak 01011111 dekoduje u 0.24609375, a odmerak 10010100 u -0.010009765625. Kada se izvrši denormalizacija dobijaju se realne vrednosti dekodovanih odmeraka: 24.609375mV i -1.0009765625mV.

Primer 2. Ako se koristi A zakon kompresije, pri čemu se kodiranjem odmerka 25mV dobija kodna reč 01111111, a dekodovanjem vrednosti 01111111 se isto dobija vrednost 25mV, odrediti kodnu reč za 20mV.

Prvo je potrebno odrediti maksimalni nivo odmerka da bi se mogla izvršiti normalizacija nivoa signala. Kodna reč 01111111 označava maksimalnu vrednost i segmenta i celine u segmentu. Pošto se dekodovana vrednost nalazi na polovini celine, očigledno, normalizovana

vrednost 25mV je udaljena za polovinu celine segmenta 7 od normalizovane maksimalne vrednosti nivoa signala tj. vrednosti 1. Sa slike 8.1.1.2 vidimo širinu jedne celine u segmentu 7. Na osnovu navedenog možemo napisati sledeće jednačine:

$$\frac{25mV}{U_{\max}} = 1 - \Delta_7 / 2$$
$$\Delta_7 = 1/32 \qquad (8.1.1.6)$$

Rešavanjem jednačina (8.1.1.6) dobija se da je maksimalan nivo približno 25.3968mV. Normalizovana vrednost za 20mV je 0.7875. Ova vrednost takođe upada u segment 7. Na osnovu (8.1.1.3) utvrđujemo da normalizovana vrednost 0.7875 pripada celini $x_1x_2x_3x_4$ = 1001. Kodirani odmerak ima vrednost 01111001, podrazumevajući G.711 format $zs_1s_2s_3x_1x_2x_3x_4$. Dekodirana vrednost bi na osnovu (8.1.1.5) imala normalizovanu vrednost 0.796875, odnosno vrednost 20.238075mV.

8.1.2. G.726 koder

G.726 koder je definisan u ITU-T preporuci G.726. Za ovaj koder se kaže da vrši ADPCM (*Adaptive Differential PCM*) modulaciju govornog signala. G.726 koder ima više varijanata koje se razlikuju u protocima. Podržani su protoci 40kb/s, 32kb/s, 24kb/s i 16kb/s. G.726 koder spada u grupu talasnih kodera.

Ideja ADPCM modulacije je da smanji protok u odnosu na PCM modulaciju tako što će se kodirati razlike između nivoa odmerka i nivoa procene odmerka umesto samih odmeraka (tj. njihovih nivoa). Pošto su susedni odmerci govornog signala međusobno visoko korelisani, ima smisla kodirati razliku, a ne same odmerke jer se time bolje iskorištava postojanje redudantnosti u govornom signalu, odnosno njegovim odmercima (zato se i uzima razlika odmerka i njegove procene, a ne razlika susednih odmeraka jer ona ne bi iskoristila korelisanost). Za procenu vrednosti odmerka se koriste prediktivni filtri. U samom predajniku se odmah vrši i rekonstrukcija odmeraka signala (simulacija dekodera u prijemniku) da bi se dobila povratna sprega za bolju procenu odmeraka govornog signala. U G.726 standardu se u prediktivnom filtru koristi šest prethodnih razlika (razlika odmeraka i njihovih procena), kao i dva prethodna rekonstruisana odmerka da bi se odredila procena za tekući odmerak govornog signala i time mogla izračunati razlika tekućeg odmerka i njegove procene. Pošto je za kodiranje razlike potrebno manje bita nego za kodiranje samih odmeraka (opseg vrednosti razlike je manji od opsega vrednosti odmeraka), onda se dobijaju niži protoci u odnosu na PCM modulaciju (G.711 koder), ali je i kvalitet govornog signala nešto niži (ne mnogo). Kodiranje razlike se može vršiti sa 2 (16kb/s), 3 (24kb/s), 4 (32kb/s) ili 5 (40kb/s) bita. Generisanje bita razlike se vrši na svakih 125μs, pa je otuda lako izračunati protoke navedene u prethodnoj rečenici. ADPCM modulacija ima dobru otpornost na greške u prenosu. Pošto je procena vrednosti odmerka izvršena na osnovu prethodnih odmeraka, ukoliko dođe do greške u prenosu, greška će biti smanjena zahvaljujući prethodnim dobro dekodovanim odmercima, a takođe će njen uticaj biti manji i na buduće odmerke. Na osnovu svega prethodno navedenog, jasno nam je otkud pojam diferencijalan u nazivu ADPCM kodera. Pojam adaptivan potiče od mehanizma adaptacije koji je takođe ugrađen u ovaj koder. Naime, korak kvantizacije (kvantizuje se razlika odmerka i njegove procene) je veći ili manji u zavisnosti od brzine promena signala. Ako signal ima velike fluktuacije vrednosti razlika (govor) onda je korak veći da bi se mogle kvalitetnije ispratiti ove nagle promene signala, odnosno ako su fluktuacije signala manje (na primer, sinusoidalni tonovi

koji se koriste za tonsku signalizaciju) onda je korak manji. G.726 koder jeste složeniji od G.711 kodera, ali je i dalje relativno jednostavan za implementaciju. Detaljnije informacije o ADPCM modulaciji se mogu naći u već pomenutoj G.726 preporuci. Postoje i druge moguće varijante i implementacije ADPCM modulacije koje nisu definisane u G.726 preporuci, odnosno ADPCM modulacija je širi pojam od G.726 preporuke koja definiše samo određene implementacije ADPCM modulacije.

Spomenimo i da je svojevremeno u mnogim državama postojao problem dvojnika u telefonskim mrežama koji su bili posledica uštede u broju parica. Jednu paricu su delila dva korisnika, pri čemu je na strani korisnika postavljena dvojnička kutija od koje su se vodile dve parice ka korisnicima (po jedna parica za svakog korisnika) i jedna (zajednička) parica ka centrali. U dvojničkoj kutiji se nalazio relej koji je zatvarao spoj od zajedničke parice ka parici korisnika koji je aktivan čime je drugi korisnik bivao potpuno odsečen od centrale tj. nije bio moguć istovremeni razgovor oba korisnika. Kada se pojavila ISDN tehnologija sa svojim U interfejsom koji je sadržao dva govorna kanala, pojavila se ideja da se dvojničke kutije zamene jednom složenijom kutijom koja bi predstavljala jednostavan mrežni završetak (NT) i koja bi svakom korisniku dodelila po jednu polovinu govornog kanala. Ovaj NT završetak je vršio ADPCM kompresiju govornih signala na protok od 32kb/s čime je omogućeno da dva korisnika dele govorni kanal - svaki bi dobio po pola govornog kanala. Ove kutije (praktično i NT uređaji) su se nazivali ADPCM-4 uređaji jer su omogućavali da četiri telefonska korisnika budu spojena na centralu preko jedne parice, čime bi se brzo mogao rešiti problem dvojnika. Međutim, ubrzo se pojavila potreba da telefonski korisnici ostvaruju modemske veze sa svojim računarima preko telefonske parice (*dial-up* konekcija) i ADPCM kompresija je pravila problem takvim vezama koje su usled upotrebe ADPCM kompresije često pucale i radile na veoma niskim protocima. Razlog je bio prost, ADPCM kompresija je predviđena za rad sa govornim signalima i degradacija kvaliteta koja je bila mala sa stanovišta govornog signala je bila neprihvatljiva sa stanovišta signala podataka koje generiše modem. Otuda, ADPCM-4 uređaji nisu zaživeli, a i oni koji su bili postavljeni su bili zamenjeni ili tako što su ipak sprovedene dodatne parice do korisnika čime je ostvareno pravo ukidanje dvojnika ili tako što je svakom korisniku dodeljen po jedan govorni kanal bez upotrebe ADPCM kompresije pa su tako dva korisnika delila i dalje jednu paricu prema centrali, ali su mogli simultano da ostvaruju telefonske razgovore.

8.1.3. G.728 koder

G.728 koder je definisan u ITU-T preporuci G.728. Ovaj koder je baziran na LD-CELP (*Low Delay - Code Excited Linear Prediction*) algoritmu. G.728 koder u osnovnoj varijanti podržava 16kb/s, dok je u aneksima standarda definisana i podrška za protoke 9.6kb/s. 12.8kb/s i 40kb/s. G.728 koder spada u grupu parametarskih kodera.

CELP (*Code Excited Linear Prediction*) algoritam i njegove varijacije (poput LD-CELP) se nalaze u osnovi mnogih parametarskih kodera. Ideja CELP algoritma je da simulira govorni sistem korisnika. Sam izvor govornog signala (glasne žice) se modeluje tzv. rečnikom (*codebook*) mogućih pobuda, dok se vokalni trakt modeluje linearnim prediktivnim (LP) filtrom i kroz ovaj filtar se propušta pobuda. Signal na izlazu iz LP filtra u koderu se tipično dodatno obrađuje tako da se šum različito potisne u različitim frekvencijskim regionima (tamo gde šum ima veći subjektivan negativan uticaj se potiskuje više, a tamo gde ima manji subjektivan negativan uticaj se potiskuje manje). Razlog za ovakvo dodatno procesiranje je struktura slušnog sistema čoveka koji je osetljiviji na šum u nekim frekvencijskim regionima i obrnuto. Filtar koji vrši ovu dodatnu obradu se označava sa *perceptual* (ili *noise*) *weighting* filtar. Ka slušaocu se

šalju parametri neophodni za sintezu govornog signala na prijemu. Pre svega, šalje se indeks pobude kojom se određuje koja pobuda iz rečnika se koristi na prijemu, ali mogu da se šalju i drugi parametri poput koeficijenata LP filtra, pojačanja pobude (tipično se pobuda iz rečnika propušta kroz svojevrsni pojačavač, tzv. *gain* blok) i dr. Napomenimo da se za prenosnu karakteristiku LP filtra tipično uzima karakteristika koja ima samo polove jer je ona jednostavna za realizaciju, a pri tome predstavlja sasvim dobar model govornog trakta. Što se tiče rečnika pobuda, u opštem slučaju se koriste adaptivni rečnik pobuda i fiksni rečnik pobuda. Adaptivni rečnik, ustvari, predstavlja zakašnjelu pobudu i koristi se za efikasno kodiranje periodičnih signala. Govorni signal ima svoju osnovnu frekvenciju koja se može empirijski izmeriti, pri čemu osnovna frekvencija tokom govora menja svoju vrednost tj. ona nije stacionarna veličina. S druge strane, slušni sistem je nelinearan sistem pa se definiše i veličina *pitch* koja predstavlja subjektivnu percepciju osnovne frekvencije, odnosno to bi po definiciji bila frekvencija sinusoide za koju slušalac (subjektivno) oseća da je najpribližnija osnovnoj frekvenciji govora, pa se otuda pravo merenje *pitch* veličine može izvršiti samo uz pomoć slušalaca. Takođe, važnu veličinu predstavljaju i formanti govornog signala koji definišu važnije delove frekvencijskog spektra u govoru i karakterišu se obično frekvencijom formanta koja tipično predstavlja sredinu formanta tj. značajnog dela u frekvencijskom spektru. Pod značajnim delom se podrazumeva oblast u frekvencijskom spektru u okviru kojeg je koncentrisana veća snaga signala. U zavisnosti od izgovorenog slova/sloga/reči menjaju se formanti pa su oni bitni sa stanovišta razumljivosti govora. Sama anvelopa govornog signala nosi informaciju o intonaciji što je bitno ako se želi rekonstruisati prirodnost (emocije) govora - ovo očigledno postižu talasni koderi jer oni rekonstruišu talasni oblik signala. Očigledno, za sintetisanje govora na prijemu veoma je bitno dinamički generisati korektno *pitch* parametar da bi se u slušnom sistemu izazvao osećaj korektne osnovne frekvencije govora, pa je otuda važan adaptivni rečnik koji kasni pobudu za procenjenu periodu *pitch*-a. S druge strane, fiksan rečnik se koristi da pokrije one informacije koje nisu pokrivene LP filtrom i adaptivnim rečnikom (tj. predikcijom *pitch* parametra) i on učestvuje sa najviše bita (ili bolje rečeno utiče na najviše bita) u kodnoj predstavi govornog signala. Sadržaj fiksnog rečnika (fiksne pobude) se direktno ili indirektno implementira u koder. Direktno podrazumeva svojevrsnu memoriju koja sadrži sve fiksne pobude, dok indirektno podrazumeva da se fiksne pobude mogu rekonstruisati, na primer, pomoću algebarskih izraza. Nakon što smo objasnili pojedine komponente objasnimo preciznije rad kodera i dekodera.

Ideja kodera je da primeni princip analize i sinteze. Vrši se sinteza svih mogućih pobuda iz rečnika, pri čemu se na ulaz sintetišućeg filtra (LP filtar) dovodi zbir izlaza (koji su eventualno pojačani) adaptivnog i fiksnog rečnika. Pod pojmom sve moguće pobude se misli na sve moguće pobude iz fiksnog rečnika jer je adaptivni rečnik, ustvari, zakašnjela pobuda pa se njegov izlaz određuje na osnovu prethodnih pobuda. Izlaz sintetišućeg (LP) filtra se propušta kroz *perceptual weighting* filtar radi boljeg uobličavanja šuma. Cilj je naći pobudu koja najbolje odgovara govornom odsečku koji se analizira, da bi se u dekoderu mogao sintetisati govorni signal koji će najviše podsećati na originalan govorni signal (kao što se vidi, ne radi se rekonstrukcija govornog signala kao kod talasnih kodera). Obično se pod najboljom pobudom smatra ona koja daje najmanju srednjekvadratnu grešku na izlazu iz *perceptual weighting* filtra (određivanje srednjekvadratne greške je deo procesa analize). Kao rezultat, koder daje indeks pobude u fiksnom rečniku, a može da daje i ostale parametre poput koeficijenata u LP filtru, procene *pitch* periode i dr.

Dekoder na osnovu primljenog indeksa vrši generisanje pobude iz fiksnog rečnika. Zbir pobuda (koje su eventualno pojačane) iz fiksnog i adaptivnog rečnika se vode na sintetišući (LP)

filtar koji produkuje govorni odsečak na osnovu kojeg se može reprodukovati analogni govorni signal. Često se iza LP filtra dodaje još jedan filtar za dodatno procesiranje koji služi da poveća prirodnost govornog signala. Vidimo na osnovu strukture dekodera, da koder simulira strukturu dekodera u cilju određivanja najbolje pobude koja opisuje govorni odsečak (kao što smo već naveli ovo je tehnika koju koristi većina kodera govornog signala).

G.728 koder koristi LD-CELP varijaciju CELP algoritma. G.728 kao ulaz prima uniformni PCM signal, pri čemu procesira blokove od po 5 odmeraka. Fiksni rečnik ima 1024 moguće pobude, odnosno za indeksiranje fiksnog rečnika je potrebno 10 bita. Sve pobude se ispituju da bi se odredila ona koja najbolje odgovara procesiranom bloku od 5 odmeraka. Kao rezultat se šalje indeks pobude, tako da se u dekoderu može aktivirati odgovarajuća pobuda koja će rekonstruisati odgovarajući odsečak koji bi trebalo da najviše podseća na obrađeni blok od 5 odmeraka. Pobuda se propušta kroz pojačavač i LP filtar, a razlika između sintetisanog odsečka i procesiranog odsečka se propušta kroz *perceptual weighting* filtar čiji izlaz se ispituje da bi se odredila srednjekvadratna greška radi izbora najbolje pobude. Koeficijenti LP filtra i pojačavača se povremeno ažuriraju preko sopstvenih povratnih sprega (na osnovu prethodnih sintetisanih pobuda se procenjuje tj. predviđa ponašanje govornog signala pa se u skladu sa procenom podešavaju koeficijenti LP filtra, odnosno pojačavača). Dekoder prima indeks pobude i na osnovu ovog indeksa generiše odgovarajuću pobudu iz fiksnog rečnika koja prolazi kroz pojačavač i LP filtar. Izlaz iz LP filtra se procesira u dodatnom filtru (postfiltar) radi postizanja bolje prirodnosti sintetisanog govornog signala. Izlaz postfiltra se konvertuje u uniformni PCM format koji se može iskoristiti za D/A konverziju u analogni govorni signal koji će se reprodukovati slušaocu. Pošto se 10 bita generiše na svakih 0.625ms (trajanje 5 odmeraka), protok je 16kb/s. Upravo ovo procesiranje malog broja odmeraka tj. kratkih blokova je razlog za LD u nazivu LD-CELP, tj. ovaj algoritam baš zbog procesiranja kratkih blokova unosi veoma malo kašnjenje u prenosu govornog signala usled procesiranja.

Varijante od 9.6kb/s i 12.8kb/s su veoma slične osnovnoj varijanti, ali je rečnik manji pa se manje bita prenosi jer je manje bita potrebno za predstavljanje indeksa. 6 bita se generiše svakih 0.625ms za protok 9.6kb/s, odnosno 8 bita svakih 0.625ms za 12.8kb/s protok. Varijanta od 40kb/s se značajnije razlikuje od osnovne varijante i ona je opisana u aneksu J G.728 preporuke.

8.1.4. G.729 koder

G.729 koder je definisan u ITU-T preporuci G.729. Ovaj koder je baziran na CS-ACELP (*Conjugate Structure Algebraic Code Excited Linear Prediction*) algoritmu. G.729 koder u osnovnoj varijanti podržava 8kb/s, dok je u aneksima standarda definisana i podrška za protoke 6.4kb/s, 11.8kb/s, kao i skalabilna podrška za protoke 8-32kb/s. G.729 koder spada u grupu parametarskih kodera.

ACELP koderi su varijanta CELP kodera, gde je fiksni rečnik pobuda zasnovan na algebarskoj strukturi (indirektno) umesto na memorijskoj strukturi (direktno). Na ovaj način je omogućena podrška i za veoma velik broj pobuda, odnosno može se koristiti velik broj bita (>50b) za predstavljanje pobuda čime je omogućeno kreiranje veoma velikih fiksnih rečnika koji sadrže velik broj pobuda. Na ACELP algoritmu je zasnovan velik broj postojećih kodera.

U G.729 koderu fiksni (algebarski) rečnik odgovara rečniku kodiranom sa 17 bita. Obrađuju se govorni odsečci od 10ms, što odgovara bloku od 80 odmeraka ako se pretpostavi frekvencija odabiranja od 8kHz. Kao rezultat obrade, koder generiše parametre koji se šalju

prijemniku tj. slušaocu. Parametri koji se šalju su koeficijenti LP filtra, indeksi fiksnog i adaptivnog rečnika, kao i koeficijenti pojačavača koji se nalaze iza fiksnog i adaptivnog rečnika u dekoderu. Pod indeksima adaptivnog rečnika se podrazumeva vrednost kašnjenja pobude koju unosi adaptivni rečnik. Svakih 10ms generiše se 80 bita koji predstavljaju kodirane parametre, pa je generisani protok kodiranog govornog signala (preciznije njegovih parametara) 8kb/s. Na prijemu se na osnovu primljenih parametara sintetiše govorni signal. Prvo se generišu pobude iz fiksnog i adaptivnog rečnika (pobude iz svakog od rečnika se puštaju kroz pojačavač), pri čemu primljeni indeksi fiksnog i adaptivnog rečnika određuju pobude koje se generišu, a primljeni koeficijenti pojačavača određuju nivo pojačanja pobuda na izlazu iz rečnika. Generisane pobude se potom sabiraju i vode na LP filtar čiji su koeficijenti podešeni na osnovu primljenih koeficijenata, a potom se signal sa izlaza iz LP filtra pušta kroz postfiltar radi dobijanja prirodnijeg govornog signala. Sami detalji oko strukture kodera i dekodera mogu se naći u preporuci G.729, ali generalno slede CELP arhitekturu opisanu u prethodnoj sekciji.

8.1.5. G.723.1 koder

G.723.1 koder je definisan u ITU-T preporuci G.723.1. Ovaj koder je baziran na principu analize i sinteze opisanom u sekciji 8.1.3 prilikom opisa CELP algoritma. G.723.1 koder u osnovnoj varijanti podržava 5.3kb/s i 6.3kb/s protoke. G.723.1 koder spada u grupu parametarskih kodera. Obrađuju se govorni odsečci od 30ms, što odgovara bloku od 240 odmeraka ako se pretpostavi frekvencija odabiranja od 8kHz. Pri tome se prilikom obrade bloka koristi još i 7.5ms narednog govornog odsečka (tzv. *look-ahead* kašnjenje G.723.1 kodera je 7.5ms). Može se reći da je ovaj koder zasnovan na CELP arhitekturi, ali postoji mala razlika u formiranju fiksnog rečnika. Ako se koristi niži protok (5.3kb/s) koristi se algebarska struktura za fiksni rečnik, pa se može reći da koder radi po ACELP principu. Ako se koristi viši protok (6.3kb/s) koristi se MP-MLQ (*MultiPulse Maximum Likelihood Quantization*) metoda za generisanje pobude. Ostatak kodera je zajednički za obe varijante. Na liniju se šalju parametri koji se odnose na koeficijente LP filtra, i na indekse fiksnog i adaptivnog rečnika. Generišu se pobude koje se potom saberu. Sabrana pobuda se propušta prvo kroz *pitch* postfiltar koji služi za podizanje kvaliteta govornog signala koji se sintetiše i to tako što povećava odnos signal/šum na mestima koja predstavljaju umnožak periode *pitch*-a. Zatim se signal propušta kroz LP filtar. Izlaz se potom vodi na pojačavač i na formant postfiltar. Formant postfiltar kontroliše pojačanje pojačavača da bi se izbeglo sintetisanje ravnog, monotonog govora. Izlaz pojačavača predstavlja konačno sintetisani govorni signal.

U aneksu A G.723.1 preporuke je definisan i rad sa detektorom aktivnosti govora (VAD - *Voice Activity Detector*) i generatorom veštačkog šuma (CNG - *Comfort Noise Generator*) pa ćemo iskoristiti ovu sekciju da pobliže objasnimo generalne principe njihovog rada. Govornik prilikom generisanja govornog signala generiše i mnoge pauze u govoru, koje se nazivaju intervali tišine. Na primer, govornik ćuti dok sluša govorni signal suprotne strane, a takođe dok govori pravi pauze između rečenica, reči, pa čak i slogova. S obzirom da se govorni signal predstavlja u vidu parametara koji ga opisuju, bilo bi poželjno kada se ne bi kodirali govorni signali tokom intervala tišine, ili kad bi se bar kodirali u manjoj meri. S druge strane, ako se ne bi sintetisalo ništa na strani slušaoca za vreme tih intervala tišine govornika, to bi izazvalo veoma loš subjektivan osećaj kod slušaoca. Otuda je potrebno proizvesti svojevrstan šum na strani slušaoca za vreme intervala tišine, ali ne bilo kakav šum, već onaj koji odgovara zvukovima okoline govornika (tzv. ambijentalni šum). Ukoliko bi se generisao drugačiji šum, to takođe ne bi bilo prijatno slušaocu jer bi primećivao takve razlike i one bi ga iritirale. VAD se koristi na strani

govornika za detekciju aktivnosti govornika, tj. intervala tišine. Kada VAD detektuje interval tišine on šalje kodiranu informaciju zvuka okoline tj. ambijentalnog šuma. U narednom trajanju intervala tišine VAD će periodično slati parametre ambijentalnog šuma iz prostog razloga da suprotna strana zna da je konekcija u redu. Takođe, u slučaju značajne promene u parametrima ambijentalnog šuma poslaće se automatski novi parametri ambijentalnog šuma. Na ovaj način je u intervalima tišine smanjen protok kodiranog govornog signala tj. njegovih parametara (u intervalu tišine je, ustvari, parametrizovan samo ambijentalni šum, a ne govorni signal korisnika koji nije ni prisutan). Ono što je bitno u radu VAD detektora je da VAD detektor mora veoma pouzdano detektovati intervale tišine, čak i u uslovima kada govorni signal nije mnogo jači od ambijentalnog šuma. Kada prijemnik (dekoder) primi kodirane parametre ambijentalnog šuma, dekoder zna da je trenutno u toku interval tišine i da treba sintetisati ambijentalni šum slušaocu. Tokom sintetisanja ambijentalnog šuma na LP filtar se povezuje izlaz CNG generatora koji na osnovu primljenih parametara generiše pobudu za LP filtar radi sintetisanja ambijentalnog šuma slušaocu. Postfiltriranje izlaza LP filtra se tipično preskače za vreme rada CNG generatora jer se sintetiše ambijentalni šum, a ne govorni signal kome je u suštini i namenjeno postfiltriranje. Parametri ambijentalnog šuma podrazumevaju parametre energije pobude (snage ambijentalnog šuma) koji su neophodni za rad CNG generatora, kao i koeficijente LP filtra, odnosno parametri ambijentalnog šuma treba da što bolje opišu spektralne karakteristike ambijentalnog šuma. Detalji vezani za VAD i CNG u slučaju G.723.1 kodera se mogu naći u aneksu A preporuke G.723.1.

8.1.6. GSM 06.10 koder

GSM 06.10 koder je definisan u ETSI preporukama. Ovaj koder je baziran na RPE-LTP (*Regular Pulse Excitation Long Term Prediction*) koderu koji takođe koristi princip analize i sinteze u samom koderu što je kao što smo mogli videti uobičajen princip za parametarske kodere. GSM 06.10 koder generiše 13kb/s protok i spada u grupu parametarskih kodera. Obrađuju se govorni odsečci od 20ms, što odgovara bloku od 160 odmeraka ako se pretpostavi frekvencija odabiranja od 8kHz. Kao rezultat procesiranja formira se 260 bita koji predstavljaju parametre obrađenog bloka (odsečka) na svakih 20ms, pa otuda protok 13kb/s. GSM 06.10 koder se još označava i terminom FR (*Full-Rate*) da se naglasi da se troši celokupan protok saobraćajnog (govornog) kanala u bežičnom interfejsu između mobilnog telefona i bazne stanice. Ovaj koder je bio veoma jednostavan što je bila značajna i poželjna osobina na počecima mobilne telefonije, ali je kvalitet govornog signala bio slab. Danas postoje bolji koderi koji obezbeđuju isti ili čak i niži protok sa boljim kvalitetom govornog signala, a s obzirom da su procesori danas veoma moćni (ovo važi i za one u mobilnim telefonima) onda se mogu primeniti ti efikasniji i kvalitetniji koderi.

8.1.7. GSM 06.20 koder

GSM 06.20 koder je definisan u ETSI preporukama. Ovaj koder je baziran na VSELP (*Vector Sum Excited Linear Prediction*) koderu koji predstavlja varijaciju CELP kodera. Fiksni rečnik se sastoji iz dva fiksna rečnika pri čemu je pretraga rečnika ubrzana određenim svojstvima kodnih predstava pobuda koje sadrže rečnici. Na primer, upotreba Grejevog koda omogućava da se susedne pobude razlikuju samo u jednom bitu pa je proračun dejstva pobude pojednostavljen i time brži. Sam adaptivni rečnik se označava terminom *self-excitation sequence*. GSM 06.20 koder generiše 5.6kb/s protok i spada u grupu parametarskih kodera. Obrađuju se govorni odsečci od 20ms, što odgovara bloku od 160 odmeraka ako se pretpostavi frekvencija odabiranja od 8kHz. Kao rezultat procesiranja formira se 112 bita koji predstavljaju parametre obrađenog

bloka (odsečka) na svakih 20ms, pa otuda protok 5.6kb/s. Ovaj koder ima malu kompleksnost izračunavanja pa se preporučivao za upotrebu u slučaju kad je baterija na izmaku jer se njegovom upotrebom smanjivala potrošnja baterije mobilnog telefona (uređaja). GSM 06.20 koder se još označava i terminom HR (*Half-Rate*) da se naglasi da se troši samo polovina protoka saobraćajnog (govornog) kanala u bežičnom interfejsu između mobilnog telefona i bazne stanice čime je dobijena mogućnost udvostručavanja govornog kapaciteta mreže mobilne telefonije.

8.1.8. GSM 06.60 koder

GSM 06.60 koder je definisan u ETSI preporukama. Ovaj koder je baziran na CELP koderu čije smo osnovne principe već objasnili (koristi se ACELP varijanta). GSM 06.60 koder generiše 12.2kb/s protok i spada u grupu parametarskih kodera. Obrađuju se govorni odsečci od 20ms, što odgovara bloku od 160 odmeraka ako se pretpostavi frekvencija odabiranja od 8kHz. Kao rezultat procesiranja formira se 244 bita koji predstavljaju parametre obrađenog bloka (odsečka) na svakih 20ms, pa otuda protok 12.2kb/s. GSM 06.60 koder se još označava i terminom EFR (*Enhanced Full-Rate*) da se naglasi da se troši celokupan protok saobraćajnog (govornog) kanala u bežičnom interfejsu između mobilnog telefona i bazne stanice, a termin unapređen (*enhanced*) označava da se postiže bolji kvalitet od GSM 06.10 tj. FR kodera. Naravno, s obzirom da je GSM 06.60 (EFR) ipak kompleksniji od GSM 06.10 (FR) upotreba GSM 06.60 kodera izaziva i nešto veću potrošnju energije baterije mobilnog telefona (uređaja).

8.2. Protokoli u IP mrežama

Kada se formira kodirani sadržaj (kodirani govorni signal) primenom kodera govornih signala, poput nekih opisanih u prethodnom potpoglavlju, potrebno je dotični kodirani sadržaj preneti preko paketske mreže. Očigledno, potrebno je kreirati pakete koji će prenositi govorne signale u odgovarajućem kodiranom formatu upotrebljenog kodera, a na prijemu će se iz paketa izvući kodirani sadržaj na osnovu kojeg će dekoder govornog signala formirati govorni signal za slušaoca (rekonstruisan ili sintetizovan, u zavisnosti od tipa primenjenog kodera govornog signala). Proces kreiranja paketa koji sadrže govorni signal se naziva paketizacija govornog signala.

S obzirom da se za prenos preko paketske mreže koriste usluge nižih slojeva, pri čemu svaki sloj dodaje svoje zaglavlje, potrebno je odrediti frekvenciju formiranja paketa koji sadrže kodirani govorni sadržaj. Ako se paketi formiraju isuviše često, udeo korisnog sadržaja (kodirani govorni signal) u paketu će biti isuviše mali. Na primer, ako bi u slučaju G.711 kodera formirali paket za svaki odmerak ponaosob, kreirali bi pakete čiji bi korisni sadržaj bio svega jedan bajt dužine, što bi značilo da bi se najveći deo kapaciteta paketske mreže trošio na prenos zaglavlja, a ne korisnog sadržaja tj. govornog signala čime bi kapacitet paketske mreže bio slabo iskorišćen. S druge strane, ne sme se ni previše čekati na formiranje paketa, tj. frekvencija formiranja paketa ne sme biti isuviše mala. Naime, prilikom razgovora dva korisnika ukupno kašnjenje jednog smera ne bi smelo preći 200ms. U slučaju narušavanja ove granice dolazi do smanjenja subjektivnog osećaja kvaliteta veze kod korisnika. Korisnici tada imaju osećaj da suprotna strana kasni sa svojim odgovorima jer nisu dobro čuli ili razumeli šta im je korisnik rekao pa ponavljaju ono što su rekli ili proveravaju vezu sa pitanjima poput ’Da li me čuješ?’ i sl. Ustvari, suprotna strana sve čuje korektno, ali ne može da odgovori pre nego što primi govorni signal. Pošto korisnici očekuju iste uslove razgovora kao da pričaju direktno licem u lice, sva prevelika kašnjenja narušavaju takav osećaj pa korisnike takva veza iritira i treba im vremena da se naviknu na takve ’neočekivane’ uslove razgovora u toku veze. Otuda, paketizacija treba da se

vrši u okvirima navedene granice i to znatno ispod navedene granice jer postoje i drugi izvori kašnjenja (na primer, kašnjenje usled propagacije). Tipično se za frekvenciju formiranja paketa uzimaju vrednosti između 10ms i 30ms (tipične vrednosti su 10ms, 20ms i 30ms). Kod parametarskih kodera se najčešće uzima vrednost koja odgovara dužini odsečka koji se procesira, a ona se tipično kreće u navedenim granicama.

U nastavku ovog potpoglavlja će u kratkim crtama biti opisani najpoznatiji protokoli koji se sreću u IP mrežama na transportnom sloju, mrežnom sloju i sloju linka podataka.

8.2.1. Transportni sloj

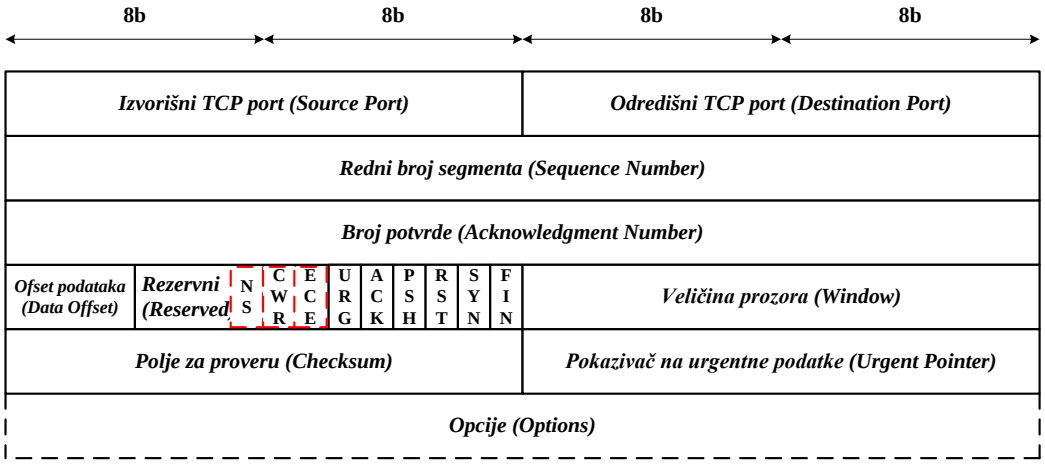
Prvenstvena uloga transportnog sloja je da obezbedi prenos paketa s kraja na kraj. Najpoznatiji protokoli transportnog sloja su TCP (*Transmission Control Protocol*) i UDP (*User Datagram Protocol*). Pored njih, kao samostalan transportni protokol je definisan i SCTP (*Stream Control Transmission Protocol*) protokol.

TCP protokol je prvenstveno namenjen za ostvarivanje pouzdane komunikacije tako što se formira virtuelna veza i potom se preko nje pouzdano prenose podaci (paketi). Pod pouzdanim prenosom se podrazumeva prenos bez grešaka u bitima paketa, sa očuvanim originalnim redosledom paketa, kao i bez dupliranja ili izostavljanja paketa. Pošto je prvenstvena briga TCP protokola pouzdan prenos, a ne samo vreme kompletiranja prenosa, TCP protokol nije najpogodnije transportno rešenje za govorne pakete koji imaju striming (*streaming*) prirodu. Međutim, pogodan je za razmenu signalizacije jer kod nje vreme kompletiranja nije u potpunosti kritično, tj. TCP zadovoljava kriterijume, a veoma je bitna pouzdanost razmene signalizacionih poruka. S druge strane, UDP protokol je nepouzdan protokol koji ne garantuje isporuku paketa, ali je znatno jednostavniji od TCP protokola. Prvenstvena namena UDP protokola je za aplikacije koje ostvaruju kratku komunikaciju, tipično u vidu slanja samo jednog paketa (na primer, DNS, DHCP) ili za aplikacije kojima pouzdanost nije od suštinskog značaja poput striming aplikacija (striming audio ili video signala). U slučaju striming aplikacija TCP pouzdano slanje nema preteranog smisla jer ne vredi retransmitovati izgubljeni/pogrešni paket s obzirom da je na prijemu prošao trenutak njegovog emitovanja, pa je stoga zgodnije koristiti UDP. SCTP protokol je alternativa TCP protokolu za razmenu signalizacije za striming aplikacije poput govorne (telefonske) veze u paketskim IP mrežama. SCTP protokol je prilagođeniji za upotrebu u razmeni telefonske signalizacije (i uopšte signalizacije za striming aplikacije) od TCP protokola, jer je to bio i osnovni razlog kreiranja SCTP protokola.

Prilikom razmene striming (audio ili video) signala, prethodna tri navedena transportna protokola, ipak, nisu u potpunosti zadovoljavala potrebe striming aplikacija, poput sinhronizacije striming tokova, praćenja kvaliteta servisa, eventualnog slanja korektivnih paketa i sl. Otuda su razvijeni RTP (*Real-time Transport Protocol*) i RTCP (*RTP Control Protocol*) protokoli koji su takođe transportni protokoli, ali ne mogu samostalno raditi na transportnom sloju kao TCP, UDP i SCTP, već zahtevaju podršku od TCP ili UDP protokola. S obzirom da je UDP bolji izbor za prenos striming signala, u praksi RTP i RTCP češće koriste UDP protokol kao podršku. RTP je odgovoran za sam prenos striming signala (audio ili video), a RTCP je odgovoran za nadgledanje stanja striming komunikacije koja se odvija putem RTP protokola. U nastavku ćemo ukratko objasniti sve navedene transportne protokole.

8.2.1.1. TCP protokol

TCP protokol je konekcioni protokol (CO - *Connection Oriented*) čime se podrazumeva da jedna TCP veza ima fazu uspostave veze, fazu korišćenja veze (razmena podataka) i fazu raskida veze. TCP veza se još naziva i virtuelna veza preko koje korisnici razmenjuju podatke. Uspostava veze se vrši principom trostrukog rukovanja u okviru koga korisnici razmenjuju inicijalne podatke poput inicijalne vrednosti brojača bajtova (koristi se za praćenje redosleda bajtova koji se šalju, kao i za njihovo potvrđivanje), podržanih opcija, veličine prijemnog bafera, vrednosti TCP portova i dr. Paketi koji se razmenjuju TCP protokolom se označavaju terminom segmenti. U preporuci RFC 793 (RFC preporuke donosi telo IETF - *Internet Engineering Task Force*) su definisani osnovni principi TCP protokola.



Slika 8.2.1.1.1. TCP zaglavlje

TCP protokol obavlja sledeće osnovne funkcije:

- Pouzdani prenos - TCP obezbeđuje pouzdani prenos s kraja na kraj preko uspostavljene virtuelne konekcije. Pod pouzdanim prenosom se podrazumeva prenos podataka bez bitskih grešaka, bez dupliranja ili izostavljanja dela podataka, i sa očuvanim originalnim redosledom podataka. Polje za proveru u okviru TCP zaglavlja se koristi za detekciju eventualnih grešaka u prenosu.
- Kontrola toka - Prijemna strana uvek obaveštava predajnu stranu o broju bajtova koji trenutno može da primi u svom baferu čime se izbegava zagušenje prijemne strane, a time i nepotrebno odbacivanje segmenata na prijemnoj strani.
- Kontrola zagušenja - TCP protokol ima mehanizam kojim indirektno prati stanje u mreži. Ukoliko ovaj mehanizam zaključi da je došlo do zagušenja, tada se obara prozor na minimalnu veličinu čime se smanjuje intezitet slanja. Smanjenjem inteziteta slanja koje će se desiti kod svih (ili bar većine) TCP korisnika, mreža će se postepeno rasteretiti, i korisnici će opet moći da šalju sa većim intezitetom svoje podatke tj. TCP segmente. U preporuci RFC 5681 su definisani osnovni mehanizmi kontrole zagušenja.
- Multipleksiranje - TCP protokol omogućava multipleksiranje više istovremenih konekcija. Svaka konekcija je jedinstveno identifikovana sa četiri informacije - izvorišna IP adresa, odredišna IP adresa, izvorišni TCP port, odredišni TCP port.

Na ovaj način je omogućeno i više istovremenih veza između ista dva korisnika (tipično je jedan korisnik klijent, a drugi server). TCP port predstavlja identifikaciju tačke povezivanja između odgovarajuće aplikacije i TCP protokola.

Struktura TCP zaglavlja je prikazana na slici 8.2.1.1.1. TCP zaglavlje se sastoji iz sledećih delova:

- Izvorišni TCP port (*Source Port*) - Šesnaestobitna identifikacija TCP porta dodeljenog TCP vezi na predajnoj strani (koja je formirala TCP zaglavlje).
- Odredišni TCP port (*Destination Port*) - Šesnaestobitna identifikacija TCP porta dodeljenog TCP vezi na prijemnoj strani (koja prima dotično TCP zaglavlje).
- Redni broj segmenta (*Sequence Number*) - Redni broj prvog bajta podataka u dotičnom TCP segmentu. U slučaju da je SYN bit aktivan (što se dešava pri uspostavi veze), tada se ovo polje tretira kao inicijalna vrednost koja se definiše na početku uspostave veze, i u tom slučaju redni broj prvog bajta podataka ima za jedan veću vrednost od vrednosti ovog polja (tj. inicijalne vrednosti). Dužina ovog polja je 32 bita i ono se koristi za rekonstrukciju originalnog redosleda podataka, kao i za formiranje potvrde za ispravno primljene TCP segmente.
- Broj potvrde (*Acknowledgment Number*) - U slučaju da je ACK bit aktivan, ovo polje sadrži pozitivnu potvrdu u vidu rednog broja prvog sledećeg bajta koji se očekuje (time su svi prethodni bajtovi potvrđeni da su uspešno primljeni - kumulativna potvrda). Onog momenta kada je veza uspostavljena, ACK bit će uvek biti aktivan, odnosno, ovo polje će uvek sadržati pozitivnu potvrdu. Ovo polje je širine 32 bita.
- Ofset podataka (*Data Offset*) - Ovo polje, dužine 4 bita, određuje početak korisnih podataka u TCP segmentu. Ofset podataka, ustvari, nosi informaciju o broju 32-bitnih reči u TCP zaglavlju tako da ovo polje određuje i dužinu TCP zaglavlja. TCP zaglavlje uvek ima dužinu koja predstavlja celobrojni umnožak 32-bitnih reči i iza TCP zaglavlja slede korisnički podaci.
- Rezervni biti (*Reserved*) - Ovo polje od 6 bita je rezervisano za eventualnu buduću upotrebu. Biti koji se ne koriste se postavljaju na vrednost 0. U slučaju podrške za eksplicitno obaveštavanje o zagušenju u mreži poslednja tri bita ovog polja se koriste kao kontrolni biti za ostvarivanje ove podrške. Naime, TCP inherentno prati stanje mreže na indirektan način. Ako dođe do retransmisije usled isticanja tajmera, TCP će zaključiti da je došlo do zagušenja u mreži. Postoje i drugi potencijalni uslovi kada TCP zaključuje da je došlo do zagušenja, ali to već zavisi od same implementacije. Na primer, tri uzastopne potvrde koje se odnose na isti očekivani segment se tumače kao negativna potvrda. Na primer, TCP Tahoe varijanta će tada da zaključi da je došlo do zagušenja u mreži pa će maksimalno oboriti brzinu slanja, dok će TCP Reno varijanta da zaključi da najverovatnije nije došlo do zagušenja u mreži pa će preventivno ipak da uspori slanje, ali ne drastično kao TCP Tahoe varijanta. Ekslipicitno obaveštavanje o zagušenju omogućava mreži da bez gubitaka paketa izvrši obaveštavanje o zagušenju u mreži. Naime, kada uređaj u mreži detektuje da dolazi do zagušenja (baferi u ruteru su se napunili iznad određene granice), on može preko IP

protokola (odgovarajući biti u IP zaglavlju paketa) da signalizira da dolazi do zagušenja. Prijemnik će ispitivanjem primljenog IP paketa da detektuje da dolazi do zagušenja u mreži i obavestiti o tome TCP sloj, koji će obavestiti eksplicitno suprotnu stranu da dolazi do zagušenja i da treba da smanji brzinu slanja TCP segmenata. Biti koji se koriste u ovom procesu su NS (*Nonce Sum*), CWR (*Congestion Window Reduced*) i ECE (*ECN - Echo*), gde je ECN skraćenica za *Explicit Congestion Notification.* Precizna upotreba ovih bita je definisana u RFC 3160 i 3540 preporukama. ECE bit se aktivira kada se primi preko IP sloja obaveštenje da je neki uređaj u mreži oglasio zagušenje. Predajnik kada primi ECE bit obaveštava da je smanjio svoju brzinu slanja aktivacijom CWR bita. ECE bit (aktivna vrednost) na početku uspostave veze (SYN bit aktivan) se koristi za oglašavanje da postoji mogućnost upotrebe ova tri navedena bita, tj. da je podržano eksplicitno obaveštavanje o zagušenju. Samo ako obe strane podržavaju ovu mogućnost će se ona i koristiti. Naravno, da bi ova mogućnost uopšte imala smisla, neophodno je da mrežni uređaji imaju mogućnost eksplicitnog obaveštavanja o zagušenju. NS bit se koristi za zaštitu od (zlonamernog ili nenamernog) sakrivanja obeleženih paketa (paketi koji nose signalizaciju preko ECE bita predajniku da treba da smanji brzinu slanja). Preko NS bita, predajnik može da verifikuje da prijemna strana radi korektno i da ona, kao ni bilo koji mrežni uređaj nije namerno ili slučajno deaktivirao aktivnu vrednost ECE bita. Princip proračuna vrednosti NS bita je definisan u RFC 3540 preporuci.

- URG (*Urgent Pointer Field Significant*) - Kontrolni bit kojim se signalizira da u TCP korisnom delu postoje tzv. urgentni podaci koji se moraju odmah proslediti korisniku.

- ACK (*Acknowledgment Field Significant*) - Kontrolni bit kojim se signalizira da li se u polju potvrde nalazi validna potvrda ili ne. U toku trajanja TCP veze, ovaj bit je uvek postavljen na aktivnu vrednost. Očigledno, TCP koristi *piggybacking* princip potvrđivanja, u smislu da se potvrde šalju zajedno sa korisničkim podacima (TCP segment istovremeno sadrži i potvrdu i korisničke podatke).

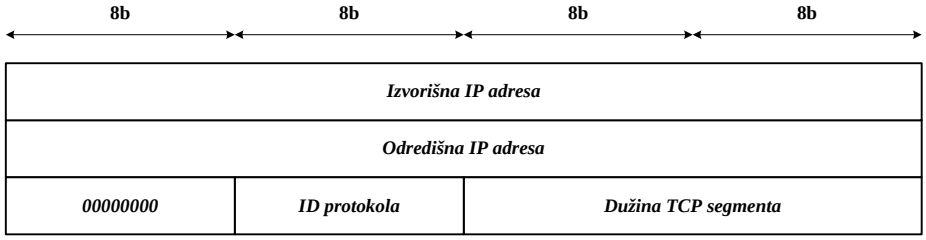
- PSH (*Push Function*) - Kontrolni bit kojim se signalizira prijemniku da sve podatke koje trenutno ima kod sebe u baferu bezuslovno prosledi korisniku (neće se proslediti oni podaci ispred kojih postoji rupa, u smislu da postoje podaci ispred njih koji još uvek nisu primljeni). Ova funkcija je uvedena pošto je u okviru implementacije TCP prijemnika moguće da prijemnik čeka da se nakupi dovoljno podataka pre nego što ih prosledi korisniku, a za neke aplikacije je bitno da im se podaci uvek što pre proslede (na primer, Telnet aplikaciji je bitno da se što pre prosleđuju karakteri koje korisnik kuca da bi se izbegle osetne pauze prilikom pritiska korisnika na taster i ispisa na ekranu koje se dešava tek kad Telnet server odgovori na poslati karakter/karaktere). Napomenimo da je u većini TCP implementacija omogućeno slanje instrukcije *push* sa aplikacionog sloja koje ima sličan efekat. Naime, originalno je zamišljeno da TCP implementacija čeka 200ms da se TCP segment napuni, pa kad istekne tih 200ms tek onda se šalje nepotpuni segment (kad se segment napuni on se odmah šalje) što je problem za neke aplikacije među kojima je i već pomenuta Telnet aplikacija. Izdavanjem *push* komande aplikacioni sloj naređuje TCP sloju da odmah pošalje nepotpuni

segment (time se kod Telnet aplikacije izbegava osetno kašnjenje između kucanja i prikaza na ekranu jer će server ranije primiti podatke i ranije će odgovoriti na njih). Naravno, ako TCP implementacija nema podršku za prijem *push* komande sa aplikacionog sloja ona neće imati nikakav efekat.

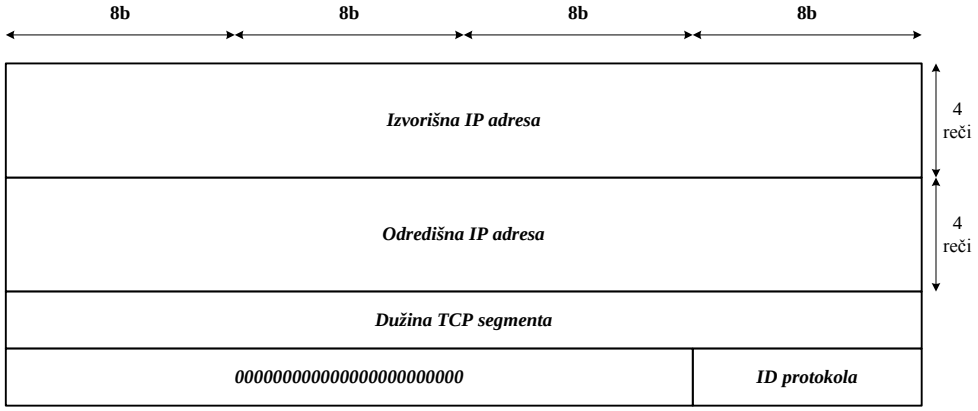
- RST (*Reset the Connection*) - Kontrolni bit kojim se signalizira neregularni raskid veze (reset veze). Naime, postoje neregularne situacije koje mogu da se jave. Pošto je TCP zadužen za pouzdan prenos, on ne sme da dozvoli neregularne situacije i time greške u prenosu, pa je uveden RST kontrolni bit kojim je omogućeno nasilno raskidanje veze, tj. reset veze. Strana, koja je uočila neregularnost, šalje TCP segment u čijem zaglavlju je aktiviran RST bit i raskida vezu i oslobađa sve resurse koji su bili dodeljeni dotičnoj vezi. Strana, koja primi TCP segment u čijem zaglavlju je aktivan RST bit, zna da je došlo do neregularne situacije pa i ona raskida vezu u smislu da oslobađa sve resurse koje je dodelila dotičnoj vezi. Tipična neregularna situacija se javlja prilikom uspostave veze kada jedna strana uđe u režim uspostavljene veze (faza razmene podataka), ali druga strana ostane u nekom prelaznom režimu ili se vrati u zatvoreno stanje (na primer, došlo je do reseta uređaja iz nekog razloga poput pada napona). Tada strana koja smatra da je veza uspostavljena šalje svoje podatke, ali neće primati potvrde (kao ni podatke) od suprotne strane jer ih ona neće ni slati. Posle izvesnog vremena će ta strana koja je ušla u fazu razmene podataka zaključiti da je došlo do neregularne situacije i poslaće TCP segment sa aktivnim RST bitom i raskinuće vezu na svojoj strani (oslobodiće resurse koje je ta veza zauzimala).
- SYN (*Synchronize Sequence Numbers*) - Kontrolni bit kojim se signalizira postavljanje inicijalne vrednosti rednog broja segmenta od koje će se početi brojati bajtovi podataka (prvi bajt će imati redni broj za jedan veći od ove inicijalne vrednosti). Ovaj kontrolni bit se ujedno koristi za uspostavu veze. Uspostava veze se vrši principom trostrukog rukovanja. Strana, koja inicira uspostavu, šalje TCP segment sa aktivnim SYN bitom. Suprotna strana šalje TCP segment sa takođe aktivnim SYN bitom čime ukazuje da ona takođe otvara vezu u suprotnom smeru (TCP virtuelna veza se može posmatrati kao par jednosmernih veza), ali i aktivnim ACK bitom kojim se potvrđuje prijem prvog TCP segmenta kojim je i inicirana uspostava veze. Strana koja je inicirala vezu, po prijemu TCP segmenta od suprotne strane, šalje potvrdu (ACK bit je postavljen na aktivnu vrednost) čime se završava uspostava veze jer su se obe strane dogovorile oko svojih inicijalnih vrednosti brojača bajtova (*sequence number*) - u ovom TCP segmentu SYN bit je neaktivan tj. SYN bit se postavlja na aktivnu vrednost samo u prvom TCP segmentu koji šalje svaka od strana. Termin trostruko rukovanje potiče od činjenice da se razmenjuju tri TCP segmenta prilikom uspostave veze.
- FIN (*No More Data From Sender*) - Kontrolni bit kojim se signalizira da je korisnik poslao sve podatke i da nema više podataka za slanje čime se ukazuje na početak raskida veze. Obe strane zasebno raskidaju vezu, pa se raskid veze može posmatrati kao par dvostrukih rukovanja. Raskid jedne strane se izvršava tako što se aktivira FIN bit, a potom suprotna strana potvrđuje prijem tog TCP segmenta sa postavljenim ACK bitom čime potvrđuje prijem tih poslednjih podataka. Suprotna strana na identičan način raskida vezu (postavljanjem FIN bita i

prijemom postavljenog ACK bita od suprotne strane). Pri tome je čak moguće da jedna strana raskine vezu, a da druga nastavi da šalje podatke pa tek onda da raskine vezu, ali to je redak slučaj. Kao što smo videli osnovnih kontrolnih bitova ima šest. Aktivna vrednost svakog od njih je vrednost 1, a neaktivna vrednost je vrednost 0.

- Veličina prozora (*Window*) - Ovim osmobitnim poljem se signalizira suprotnoj strani ukupan broj bajtova koje može da primi (počevši od prvog bajta koji se očekuje i koji je signaliziran u polju broj potvrde). Ovo polje se koristi za kontrolu toka čime se izbegava zagušenje bafera u prijemniku usled prebrzog slanja predajne strane.



IPv4 slučaj



IPv6 slučaj

Slika 8.2.1.1.2. Deo iz IP zaglavlja koji ulazi u TCP pseudozaglavlje

- Polje za proveru (*Checksum*) - Ovo šesnaestobitno polje se koristi za detekciju bitskih grešaka u prenosu. Ovo polje se formira tako što se izvrši sabiranje u komplementu jedinice svih šesnaestobitnih reči iz TCP pseudozaglavlja, i korisničkih podataka koji se nalaze u TCP korisnom delu (*data* deo). Rezultat sabiranja se predstavlja u komplementu jedinice i stavlja u polje za proveru. Ovaj princip se koristi i za formiranje sume za proveru kod IPv4 zaglavlja, pa će konkretan primer računanja biti prikazan kod IPv4 protokola. Ukoliko je ukupan broj bajtova navedenih polja neparan, tada se poslednji bajt dopunjava nulama da bi se dobila šesnaestobitna reč (preciznije, dodaje se dodatni bajt koji je popunjen nulama). Prilikom računanja vrednosti ovog polja, uzima se da je polje za proveru TCP zaglavlja popunjeno nulama. TCP pseudozaglavlje se sastoji iz TCP zaglavlja i dela polja iz IP zaglavlja. Deo pseudozaglavlja koji se formira na

osnovu IP zaglavlja je prikazan na slici 8.2.1.1.2. U slučaju IPv4 mrežnog protokola (tj. IPv4 protokol enkapsulira TCP segment) taj deo pseudozaglavlja je dužine tri 32-bitne reči i sadrži izvorišnu i odredišnu IP adresu, identifikaciju enkapsuliranog protokola (vrednost ovog polja je 6 za TCP protokol) koja se koristi na mrežnom sloju i dužinu TCP segmenta u bajtovima (u dužinu TCP segmenta ulaze i zaglavlje i korisni deo). U slučaju IPv6 mrežnog protokola, taj deo pseudozaglavlja je dužine deset 32-bitnih reči i sadrži izvorišnu i odredišnu IP adresu (koje su u IPv6 slučaju četiri puta duže), dužinu TCP segmenta (u dužinu TCP segmenta ulaze i zaglavlje i korisni deo) i identifikaciju protokola transportnog sloja (vrednost ovog polja je 6 za TCP protokol, tj. ista kao i u IPv4 slučaju, jer je usvojeno da se identifikacije enkapsuliranih protokola ne razlikuju u IPv4 i IPv6 zaglavljima) koja se koristi na mrežnom sloju. Kao što vidimo, u oba slučaja pseudozaglavlje se formira od odgovarajućih polja zaglavlja mrežnog protokola (IPv4 ili IPv6), jedino se dužina TCP segmenta mora izračunati na osnovu vrednosti odgovarajućih polja zaglavlja mrežnog protokola. Na primer, u slučaju IPv4 protokola, dužina TCP segmenta se računa kao razlika vrednosti polja ukupne dužine IP paketa i polja dužine IP zaglavlja (pri tome se mora izvršiti odgovarajuća konverzija jer ukupna dužina je izražena u broju bajtova, a dužina IP zaglavlja u broju 32-bitnih reči).

- Pokazivač na urgentne podatke (*Urgent Pointer*) - Ovo polje se koristi samo kad je kontrolni bit URG postavljen na aktivnu vrednost. Vrednost ovog pokazivača ukazuje na redni broj poslednjeg bajta koji pripada urgentnim podacima. Urgentni podaci se uvek stavljaju na početak korisnog (*data*) dela TCP segmenta, a za njima slede regularni podaci ako ih ima. Urgentni podaci se odmah bezuslovno prosleđuju korisniku tj. aplikacionom sloju. Ova opcija se ne koristi često u praksi. Napomenimo da je u RFC 793 pogrešno navedeno da pokazivač pokazuje na redni broj prvog bajta koji se nalazi iza urgentnih podataka u korisnom delu TCP segmenta i ova greška je kasnije ispravljena u RFC 961, ali u praksi je većina TCP implementacija sledila uputstvo iz RFC 793 preporuke, tako da se u praksi češće sreće pogrešna definicija iz preporuke RFC 793. Ovo može biti problem ako obe strane imaju različito tumačenje pokazivača, ali, srećom, ova opcija se veoma retko koristi pa se uglavnom problemi nekompatibilnosti TCP implementacija sa stanovišta tumačenja pokazivača na urgentne podatke ne sreću često.

- Opcije (*Options*) - Ovo polje nije obavezno da se pojavi u zaglavlju. Polje opcije je promenjive dužine i sadrži opciona polja u TCP zaglavlju (maksimalna dužina ovog polja je 40 bajtova). Ukoliko ukupna dužina ovog polja nije celobrojni umnožak 32-bitnih reči, onda se vrši dopunjavanje ovog polja nulama da bi se postiglo da dužina ovog polja bude celobrojan umnožak 32-bitnih reči. Opciona polja koja se mogu koristiti su MSS (*Maximum Segment Size*) vrednost, dozvola selektivnih potvrda (*Selective Acknowledgment Permitted*), NOP (*No Operation*) opcija, vremenska oznaka i eho vremenske oznake (*Timestamp and Echo of Previous Timestamp*) i dr. Svako opciono polje se sastoji potencijalno iz tri dela - tip opcije (*Option-Kind*), dužina opcije (*Option-Length*) i podaci opcije (*Option-Data*). Samo se tip opcije javlja u svim opcionim poljima, dok druga dva polja

mogu, ali i ne moraju da se pojave što zavisi od samog tipa opcije. Dužina opcije definiše dužinu čitavog opcionog polja u bajtovima. MSS opciono polje se koristi za javljanje suprotnoj strani koja je maksimalna dužina segmenta koju prijemnik (na strani koja je postavila ovu opciju) može da primi. Ovo opciono polje se šalje samo pri uspostavi veze u TCP segmentu kod koga je aktivna vrednost SYN kontrolnog bita. Vrednost tipa opcije MSS opcionog polja je 2, vrednost dužine opcije je 4, a sami podaci opcije su dužine dva bajta i oni sadrže oglašenu MSS vrednost. Dozvola selektivnih potvrda signalizira da TCP implementacija podržava selektivne potvrde (tip opcije je 4, a dužina opcije je 2, odnosno podaci opcije ne postoje) i razmenjuje se prilikom uspostave veze. Originalno, TCP protokol je sadržao samo kumulativne pozitivne potvrde, pa segmenti koji su stigli van redosleda nisu potvrđivani dok se ne popune rupe u redosledu, što je potencijalno izazivalo problem nepotrebnih retransmisija, ali i usporavanja slanja na predajnoj strani. Da bi se prevazišao taj problem u preporuci RFC 2018 uvedena je podrška za selektivne potvrde u TCP protokol. U polju broj potvrde se i dalje šalje kumulativna pozitivna potvrda, ali sada postoji mogućnost da se dodaju i selektivne potvrde. Pri tome, selektivne potvrde se formiraju u vidu broja blokova van očekivanog redosleda koji se potvrđuju, pri čemu se za svaki blok navodi redni broj prvog i poslednjeg bajta u potvrđenom bloku (redni brojevi su dužine 2 bajta, tj. 16 bita). Selektivne potvrde se šalju u opcionom polju (TCP zaglavlja) selektivna potvrda čiji je tip opcije 5, a dužina opcije je 10, 18, 26 ili 34 bajta u zavisnosti koliko blokova selektivne potvrde se šalje (1, 2, 3 ili 4, respektivno). U RFC 2883 preporuci je proširena opcija selektivnih potvrda mogućnošću da se selektivno potvrđuju primljeni duplikati čime se dobija mogućnost signaliziranja predajniku da je došlo do prijema duplikata nekog TCP segmenta. Duplikat se isto navodi u granicama u vidu rednog broja prvog i poslednjeg bajta bloka koji je duplikat. Pošto je prilikom retransmisije TCP segmenta, predajnik usporio slanje smatrajući da je došlo do zagušenja, po prijemu selektivne potvrde duplikata, predajnik će se vratiti na originalnu (bržu) brzinu slanja. Selektivno potvrđivanje se može koristiti samo ako obe strane podržavaju selektivno potvrđivanje, što je u praksi najčešći slučaj, pošto je ova opcija zaživela i najveći broj današnjih TCP implementacija je podržava. NOP opcija je opcija koja ne izaziva nikakvu akciju, niti zahteva procesiranje. Sadrži samo polje tip opcije koje je postavljeno na vrednost 1. Tipično se NOP operacija koristi za poravnanje sledeće opcije na početak 32-bitne reči radi lakšeg procesiranja opcije, ali TCP prijemnik mora biti spreman da procesira opcije i ako one nisu poravnate na početak 32-bitne reči. TCP vremenske oznake (*timestamps*) su definisane u preporuci RFC 1323 i predajnik u njima označava trenutak slanja dotičnog TCP segmenta. U eho se stavlja vrednost vremenske oznake uspešno primljenog TCP segmenta (kojeg primljenog TCP segmenta zavisi od same situacije i pravila su precizno definisana u okviru preporuke RFC 1323). Vremenske oznake se koriste za kvalitetniji proračun RTT (*Round Trip Time*) vremena, kao i za detektovanje eventualnih starih duplikata koji mogu da izazovu problem pogrešnog brojanja bajtova ako se on protumači kao novi paket. Ako je moduo brojanja isuviše mali može da se desi da neki stari zakasneli paket stigne nakon dužeg vremena, a u međuvremenu se brojač usled uspešnog prijema

velikog broja paketa pomerio na vrednosti u koje spada i vrednost iz tog starog zakasnelog segmenta. Ova situacija je postala problematična tek sa pojavom veoma brzih optičkih linkova koji omogućavaju brzo slanje ogromnih količina podataka jer se pri velikim brzinama slanja brže prođe čitav krug kroz moduo brojanja. Napomenimo da se u slučaju upotrebe vremenskih oznaka rešava problem neodređenosti da li je primljeni TCP segment originalan ili retransmitovan. Takođe, vremenske oznake se mogu iskoristiti i za detektovanje na strani predajnika preko eho vrednosti da li se nepotrebno usporilo slanje jer se ušlo u proces retransmisije segmenata za koje je protumačeno da nisu stigli na odredište što je definisano u preporuci RFC 3522. Ako se detektuje da se nepotrebno ušlo u proces retransmitovanja, predajnik će izaći iz tog procesa i vratiti se na originalnu brzinu slanja. Vremenska oznaka je dužine 4 bajta. Tip opcije vremenska oznaka i eho vremenske oznake je 8, a dužina te opcije je 10 bajtova jer se uvek stavljaju i trenutak slanja dotičnog segmenta, ali i eho izabranog primljenog segmenta. Opcija skaliranje prozora omogućava skaliranje vrednosti veličine prozora iz TCP zaglavlja tako da su podržane i veće veličine prozora. Potreba za ovom opcijom se javila instalacijom linkova velikih brzina kada je omogućeno brzo slanje velikih količina podataka pa je došlo do potrebe da prijemnik dobije mogućnost oglašavanja i veće količine bajtova koju može da primi. Tip ove opcije je 3, a dužina opcije je 3 bajta, što znači da su podaci opcije dužine jedan bajt. Podaci se tumače kao broj pozicija (bita) za koji se mora pomeriti vrednost iz polja veličina prozora ulevo da bi se dobila originalna vrednost na prijemu. Na predaji se vrši pomeranje originalne veličine prozora za isti broj mesta udesno i upisuje u polje veličina prozora koje se nalazi u TCP zaglavlju. Ukoliko je tip opcije jednak 00000000, tada je u pitanju signalizacija da je lista opcija završena - opciono polje kraj liste opcija (nije obavezno koristiti ovo opciono polje što je slučaj kada se opcije završavaju na 32-bitnoj granici). Tabela 8.2.1.1.1 daje sumirani prikaz opcija koje su razmatrane u ovoj sekciji.

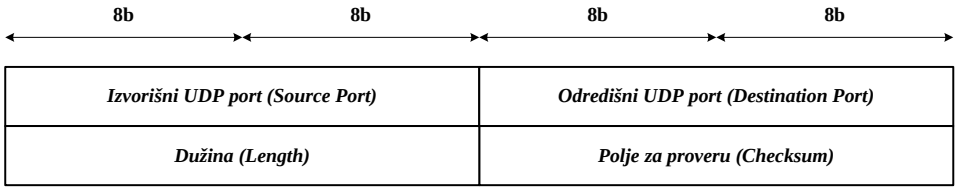
Tabela 8.2.1.1.1. Postojeće opcije u TCP zaglavlju

Opcija	Kod opcije	Dužina opcije u bajtovima
Kraj liste opcija	0	-
NOP	1	-
MSS	2	4
Skaliranje prozora	3	3
Dozvola selektivnih potvrda	4	2
Selektivne potvrde	5	10, 18, 26 ili 34
Vremenske oznake	8	10

8.2.1.2. UDP protokol

UDP protokol je nekonekcioni protokol (CL - *ConnectionLess*) čime se podrazumeva da postoji samo faza razmene podataka (nema uspostave i raskida veze). UDP protokol radi po principu najboljeg pokušaja (*best effort*) i ne garantuje isporuku podataka, pa je na aplikacionom sloju da implementira podršku za pouzdan prenos ukoliko je to neophodno. UDP protokol je razvijen za aplikacije koje ne zahtevaju pouzdan prenos i pre svega su transakcijski orijentisane u smislu da se svaki UDP paket tumači kao jedna transakcija. Upravo zbog odsustva mehanizma

retransmisija za postizanje pouzdanosti, ali i mehanizama kontrole toka i zagušenja, UDP protokol je postao popularan i za audio/video striming aplikacije.



Slika 8.2.1.2.1. UDP zaglavlje

Zaglavlje UDP paketa je prikazano na slici 8.2.1.2.1 i kao što se može videti UDP zaglavlje je veoma jednostavne strukture. UDP zaglavlje se sastoji iz sledećih delova:

- Izvorišni UDP port (*Source Port*) - Šesnaestobitna identifikacija UDP porta predajne strane (koja je formirala UDP zaglavlje, tj. UDP paket).
- Odredišni UDP port (*Destination Port*) - Šesnaestobitna identifikacija UDP porta prijemne strane (koja prima dotično UDP zaglavlje tj. UDP paket).
- Dužina (*Length*) - Dužina UDP paketa u bajtovima. Pod UDP paketom se podrazumevaju UDP zaglavlje i korisnički podaci koji se stavljaju u UDP korisni (*data*) deo.
- Polje za proveru (*Checksum*) - Ovo šesnaestobitno polje se koristi za detekciju bitskih grešaka u prenosu. Ovo polje se formira tako što se izvrši sabiranje u komplementu jedinice svih šesnaestobitnih reči iz UDP pseudozaglavlja, i korisničkih podataka koji se nalaze u UDP korisnom delu (polje za proveru je u ovom procesu popunjeno nulama). Rezultat sabiranja se predstavlja u komplementu jedinice i stavlja u polje za proveru. UDP pseudozaglavlje se sastoji iz UDP zaglavlja i dela polja iz IP zaglavlja. Deo pseudozaglavlja koje se formira od polja IP zaglavlja ima identičnu strukturu kao u slučaju TCP protokola, samo se razlikuje identifikacija protokola jer je sada u pitanju UDP protokol (identifikacija UDP protokola je 17 i za IPv4 i za IPv6 slučaj). Naravno, u pseudozaglavlju se pod dužinom misli na dužinu UDP paketa i ova vrednost se indirektno izračunava na osnovu vrednosti odgovarajućih polja zaglavlja mrežnog protokola, kao i u TCP slučaju. I ovde se, u slučaju da je ukupan broj bajtova neparan, dodaje još jedan bajt da bi se dobio celobrojan umnožak šesnaestobitnih reči neophodnih za sabiranje prilikom formiranja vrednosti polja za proveru. Ako se polje za proveru ne koristi tada se popunjava sa svim nulama čime se signalizira prijemnoj strani da se polje za proveru ne koristi. Iz tog razloga se u slučaju da je rezultat sabiranja jednak svim nulama, ovo polje popunjava svim jedinicama što je u skladu sa formatom komplementa jedinice.

8.2.1.3. SCTP protokol

TCP protokol obezbeđuje pouzdan prenos i poseduje brojne druge korisne funkcionalnosti poput kontrole toka, kontrole zagušenja i dr. Međutim, način rada TCP protokola nije bio adekvatan za sve aplikacije, među kojima se isticala i telefonska signalizacija. Naime, razvojem paketskih mreža i njihovom sve većom penetracijom u mrežnoj infrastrukturi, pojavila se potreba da se pojedini delovi klasične telefonske mreže povezuju preko paketske mreže, pa je

bilo potrebno osmisliti način za prenos signalizacije i govornih signala preko paketske mreže. Telefonska signalizacija zahteva visoku pouzdanost, pa je očigledno potrebno izabrati transportni protokol koji obezbeđuje pouzdanost kao što je, na primer, TCP protokol. Ali, TCP je imao i određene mane sa stanovišta prenosa telefonske signalizacije poput:

- TCP obezbeđuje pouzdan prenos koji između ostalog podrazumeva i isporuku podataka na prijemu u identičnom redosledu kao na predaji. Međutim, određene aplikacije nemaju potrebu za ovako restriktivnim pristupom, već im odgovara i delimično rekonstruisan redosled podataka sa predaje, što umanjuje uticaj tzv. HOL (*Head-Of-Line*) blokade. HOL blokada podrazumeva da ispravna isporuka nekog dela podatka blokira slanje svih sukcesivnih delova podataka. Naime, ako se neki segment izgubi, prozor na predaji, koji određuje koji segmenti imaju pravo da se pošalju, će se zamrznuti na istoj poziciji i neće moći da klizne dalje sve dok se ne dobije potvrda za nepotvrđeni segment sa početka prozora. Stoga, svi sukcesivni segmenti koji nisu ušli u prozor čekaju da klizne prozor da bi mogli da se pošalju, što dovodi do nepotrebnih velikih kašnjenja kod nekih aplikacija koje ne zahtevaju striktnu rekonstrukciju originalnog redosleda podataka na prijemnoj strani.
- TCP tok je bajtovski orijentisan, odnosno TCP posmatra podatke kao niz bajtova. Otuda TCP na prijemnoj strani prikuplja podatke tj. bajtove podataka i s vremena na vreme (kad ih se dovoljno nakupi) ih prosleđuje aplikaciji. Podaci za neke aplikacije mogu da se podele u logičke celine (poruke), ali TCP na prijemu ne prosleđuje poruke već nizove bajtova koji mogu da pripadaju različitim porukama, što može da predstavlja problem pojedinim aplikacijama jer ovakav pristup može da izazove neželjena kašnjenja nekih poruka (neka hitna poruka može nepotrebno da čeka u baferu na prijemnoj strani da se nakupi dovoljno bajtova koji, u stvari, pripadaju narednim porukama da bi se izvršilo prosleđivanje aplikacionom sloju). Takođe, ako se želi postići da TCP na prijemu prosleđuje poruke aplikacionom sloju, aplikacije moraju da vode računa da šalju *push* komandu posle svake poruke, da bi se preko PSH kontrolnog bita izazvalo prosleđivanje podataka aplikaciji na prijemnoj strani, što komplikuje i aplikacije i TCP implementacije koje moraju da podrže prijem *push* komande od aplikacionog sloja.
- Neki uređaji su iz razloga veće pouzdanosti priključeni na više mreža odjednom i imaju više IP adresa (*multi-homing*). Da bi se povećala pouzdanost komunikacije može se u slanju podataka ka takvom uređaju iskoristiti činjenica da se uređaju može pristupiti sa više strana, pa bi bilo zgodno da transportni protokol eksploatiše ovu mogućnost. Naime, tada bi transportni protokol, na primer mogao slati podatke na jednu od adresa uređaja, a ukoliko taj pristup postane problematičan (na primer, usled kvara linka ili rutera dođe do prekida puta), može se iskoristiti neka od drugih adresa uređaja radi nastavka komunikacije bez prekidanja transportne virtuelne veze. TCP ovu mogućnost nema.
- TCP je ranjiv na DoS (*Denial of Service*) napade preko SYN bita. Naime, ako se uređaj bombarduje zahtevima za uspostavom TCP veze (SYN bit u TCP

zaglavlju je aktivan), uređaj će pokušati da uspostavi vezu, pri čemu će zauzeti resurse za tu vezu. Ako se primi veliki broj takvih paketa (sa aktivnim SYN bitom u TCP zaglavlju) u relativno kratkom periodu, može doći do pada sistema uređaja usled preopterećenosti resursa.

Upravo usled navedenih ograničenja TCP protokola je razvijen SCTP protokol, koji je primarno bio namenjen za razmenu telefonske signalizacije preko paketskih IP mreža, ali i za druge aplikacije kojima nisu odgovarala navedena ograničenja TCP protokola. SCTP sadrži mnoge osobine i funkcionalnosti TCP protokola, a razlike se uglavnom odnose na prevazilaženje navedenih TCP ograničenja. SCTP je definisan u preporuci RFC 4960, dok su u preporuci RFC 3286 opisno navedene funkcionalnosti SCTP protokola bez zalaženja u same detalje kao u RFC 4960 preporuci.

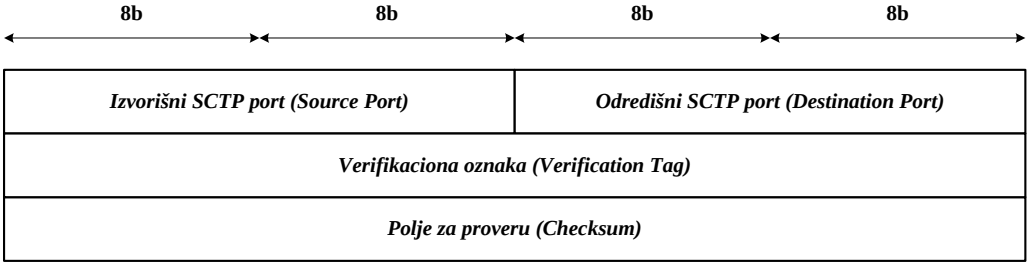
SCTP protokol je konekcioni protokol (CO) čime se podrazumeva da jedna SCTP veza ima fazu uspostave veze, fazu korišćenja veze (razmena podataka) i fazu raskida veze. SCTP je orijentisan na prenos poruka (*message-oriented*), pri čemu je moguće u okviru jedne SCTP veze imati više paralelnih tokova poruka čime se umanjuje efekat HOL blokade. Obezbeđuje pouzdan prenos koji podrazumeva da nema bitskih grešaka, izgubljenih podataka, dupliranih podataka, kao i originalan redosled podataka, ali na nivou toka poruka (svaki tok poruka će na prijemu da se prosledi aplikacionom sloju u originalnom redosledu). SCTP takođe vrši kontrolu toka i zagušenja na veoma sličan način kao i TCP protokol. Kontrola toka podrazumeva, kao i kod TCP protokola, da prijemna strana obaveštava predajnu stranu, koliko ima slobodnog mesta u prijemnom baferu. Previđeno je da SCTP i TCP tokovi postoje u istoj mreži i da se bore za iste resurse, pa da se ne bi favorizovao određeni transportni protokol, usvojeno je da SCTP implementira iste mehanizme kontrole zagušenja kao TCP (definisani u RFC 5681). Pri tome, mehanizam kontrole zagušenja se sprovodi za svaku putanju ponaosob, pri čemu se putanje definišu u vidu IP adresa suprotne strane (kojih može biti više). SCTP podržava i koristi selektivne potvrde. Selektivne potvrde su bile uvedene prvo u SCTP, i to je bila jedna od razlika u odnosu na TCP, ali je posle i u TCP uvedena mogućnost upotrebe selektivnih potvrda pa se SCTP i TCP više ne razlikuju u tom pogledu. Retransmisija se izvršava u slučaju isteka RTO tajmera, ili u slučaju da su primljene četiri uzastopne selektivne potvrde koje javljaju isti nedostajući podatak (ovo pravilo od četiri uzastopne selektivne potvrde je izabrano da bi se smanjila verovatnoća nepotrebnih retransmisija). Funkcionalnosti koje SCTP podržava i implementira, a koje TCP ne podržava su:

- SCTP je orijentisan na prenos poruka (*message-oriented*). SCTP prenosi poruke za razliku od TCP koji prenosi bajtove. SCTP stoga ima očuvanu strukturu poruka na transportnom sloju, pa je poboljšana isporuka poruka na prijemnoj strani. Poruka se prosleđuje aplikacionom sloju kada se kompletira njen prijem. U slučaju TCP protokola to nije moguće bez asistencije aplikacionog sloja i *push* komande na predajnoj strani, pošto TCP protokol nema očuvane granice poruka.
- SCTP omogućava da u okviru jedne veze postoji više paralelnih tokova. Pošto se paralelni tokovi nalaze u okviru iste veze, na sve njih deluje zajednička kontrola toka i zagušenja što smanjuje ukupan broj bita zaglavlja i zauzetost resursa na predajnoj strani, od slučaja kada bi svaki tok bio u okviru zasebne SCTP veze, pa samim tim imao zasebnu kontrolu toka i zagušenja. Greška u jednom toku, utiče samo na podatke tog toka (vrši se retransmisija i javlja se efekat HOL blokade samo na tom toku), dok na druge ne utiče. Tako može da se desi da se poruka

jednog toka ranije isporuči aplikacionom sloju, od neke poruke drugog toka, iako je njihov redosled slanja bio obrnut na predaji. Ova osobina je veoma zgodna za razmenu telefonske signalizacije. Poruke koje se odnose na isti poziv ili isti govorni kanal moraju da se prenose u originalnom redosledu, ali poruke koje se ne odnose na isti poziv ili govorni kanal nisu u korelaciji pa njihov redosled ne mora da odgovara originalnom redosledu na predaji. Otuda je upotreba paralelnih tokova veoma praktična za iskorištavanje ovog svojstva nekorelisanosti jer bi se poruke za jedan poziv prenosile preko jednog toka, a za drugi preko drugog toka, čime eventualni gubici na jednom toku ne bi imali efekta na drugi tok (ako se zanemari eventualna detekcija zagušenja u mreži). Drugi primer gde je ova osobina zgodna je skidanje stranica sa multimedijalnim sadržajem, gde bi se multimedijalni sadržaj i osnovni sadržaj stranice izdelio na zasebne tokove, čime eventualni gubici na jednom toku ne bi uticali na ostale tokove, pa bi se najveći deo stranice skinuo sa manjim kašnjenjem (smanjen efekat HOL blokade koji bi delovao samo na jedan tok, tj. samo na jedan deo stranice), a takođe paralelno skidanje više delova stranice odjednom bi imalo i pozitivan subjektivni efekat kod korisnika koji bi imao dojam da se stranica brže skida.

- Podrška za hostove sa više adresa (*multi-homing*). Tokom uspostave veze SCTP uređaji razmenjuju listu svojih IP adresa. Svaki uređaj izabere jednu od adresa oglašenih od suprotne strane kao primarnu adresu i šalje sve podatke ka toj adresi. Ukoliko dođe do potrebe za retransmisijom, retransmitovani podatak se šalje preko neke od alternativnih adresa suprotne strane da bi se povećala verovatnoća uspeha retransmisije. Ukoliko broj retransmisija postane prevelik, prelazi se na neku od alternativnih adresa, tj. neka od alternativnih adresa se bira kao primarna adresa. Očigledno, cilj je da se poveća pouzdanost SCTP virtuelne veze. Prilikom trajanja SCTP veze, periodično se šalju kratki paketi (tzv. *Heartbeat* paketi) na alternativne adrese da bi se pratila njihova raspoloživost. Ukoliko se detektuje neraspoloživost neke odredišne adrese, ona se stavlja u listu nedostupnih adresa i ne koristi se za slanje. Ispitivanje svake od (alternativnih) adresa se vrši za čitavo vreme trajanja SCTP veze, čak i ako su one proglašene nedostupnim, jer je ideja da se uoči i eventualna ponovna dostupnost adrese koja je prethodno proglašena nedostupnom. Ukoliko sve adrese postanu nedostupne i to potraje dovoljno drugo, suprotna strana se proglašava nedostupnom i zatvara se SCTP veza.
- U slučaju da na predajnoj strani neka vremenski osetljiva poruka nije još poslata, a istekao je njen tajmer koji je definisala aplikacija koja je i kreirala poruku, tada ta poruka može da se odbaci na predajnoj strani i da se ne pošalje suprotnoj strani. Naravno, poruke koje su poslate moraju i da se isporuče, pa će u slučaju njihovih grešaka biti retransmitovane nezavisno od toga da li su vremenski osetljive ili ne.
- Otpornost na DoS napade. Uspostava veze se vrši u vidu četvorostrukog rukovanja, za razliku od trostrukog rukovanja kod TCP protokola. Pri tome, koristi se COOKIE mehanizam za zaštitu od DoS napada. Ovaj mehanizam i sama uspostava veze će biti objašnjeni kasnije u okviru ove sekcije.
- Nepostojanje poluotvorenog režima rada. TCP ima mogućnost da samo jedna strana zatvori TCP vezu, a da suprotna strana nastavi da šalje i tek onda zatvori

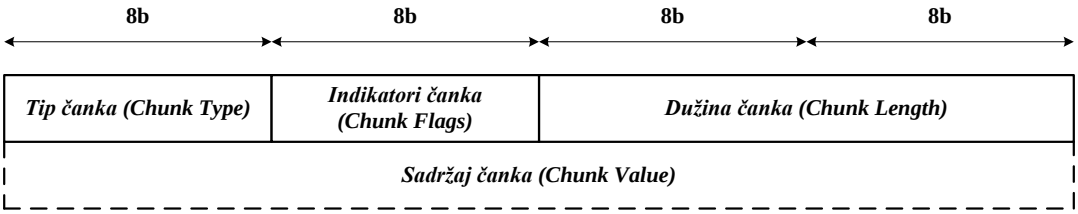
vezu (tzv. poluotvoreni režim rada). SCTP nema ovu mogućnost, već se veza zatvara sa obe strane u vidu trostrukog rukovanja. Napomenimo da, kao i TCP, i SCTP ima mogućnost nasilnog raskidanja veze u slučaju detekcije neregularnih situacija.



Slika 8.2.1.3.1. SCTP zaglavlje

Na slici 8.2.1.3.1 je prikazana struktura SCTP zaglavlja. SCTP zaglavlje se sastoji iz sledećih delova:

- Izvorišni SCTP port (*Source Port*) - Šesnaestobitna identifikacija SCTP porta predajne strane (koja je formirala SCTP zaglavlje, tj. SCTP paket).
- Odredišni SCTP port (*Destination Port*) - Šesnaestobitna identifikacija SCTP porta prijemne strane (koja prima dotično SCTP zaglavlje tj. SCTP paket).
- Verifikaciona oznaka (*Verification Tag*) - Ovo polje se koristi za proveru da je SCTP paket zaista došao od suprotne strane u vezi. Naime, tokom procesa uspostave veze svaka strana prilikom razmene parametara veze poput veličine bafera na prijemu, takođe šalje i svoju verifikacionu oznaku (inicijalna oznaka - *Initiate Tag*) za tu vezu koja se uspostavlja. U nastavku veze, pri formiranju SCTP paketa se u ovo polje stavlja oznaka primljena (oglašena) od suprotne strane da bi suprotna strana tako bila sigurna da je izvorište SCTP paketa validno (odnosno da je u pitanju SCTP host sa kojim je i uspostavljena dotična veza). Postoje i izuzeci za vrednost ovog polja koji su navedeni u preporuci RFC 4960. Na primer, prvi paket kojim se započinje uspostava veze će imati ovo polje postavljeno na sve nule jer još nisu razmenjene verifikacione oznake.
- Polje za proveru (*Checksum*) - Ovo 32-bitno polje se koristi za detekciju bitskih grešaka u prenosu. Očigledno, pošto je polje za proveru duže nego u TCP slučaju manja je verovatnoća da se jave bitske greške koje se ne mogu primetiti. Koristi se CRC32c algoritam zasnovan na generišućem polinomu $x^{32}+x^{28}+x^{27}+x^{26}+x^{25}+x^{23}+x^{22}+x^{20}+x^{19}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^{9}+x^{8}+x^{6}+1$. Štiti se celokupan SCTP paket (i zaglavlje i korisni deo), pri čemu se polje za proveru popunjava nulama prilikom računanja vrednosti koja će se upisati u to polje. Na prijemu se vrši identičan postupak i proverava se da li se primljeno polje za proveru poklapa sa proračunatim. Ako se poklapaju, smatra se da je SCTP paket ispravan (bez bitskih grešaka), u suprotnom SCTP paket je neispravan i stoga se odbacuje. Proračun se vrši na sledeći način. SCTP paket se predstavlja u vidu polinoma $M_{init}(x)$ koji se štiti. Komplementira se najviših 32 bita $M_{init}(x)$ i dobija se polinom $M(x)$. Vrši se deljenje $M(x)\cdot x^{32}/G(x)$, gde je $G(x)$ generišući polinom. Ostatak deljenja se komplementira i postavlja u polje za proveru.



Slika 8.2.1.3.2. Struktura jednog čanka

Tabela 8.2.1.3.1. Tipovi čankova

Tip čanka	Opis
0	Korisnički podaci (DATA)
1	Inicijalizacioni čank (INIT)
2	Potvrda INIT čanka (INIT ACK)
3	Selektivna potvrda (SACK)
4	*Heartbeat* zahtev (HEARTBEAT)
5	Potvrda *heartbeat* zahteva (HEARTBEAT ACK)
6	Nasilni raskid veze (ABORT)
7	Regularni raskid veze (SHUTDOWN)
8	Potvrda regularnog raskida (SHUTDOWN ACK)
9	Greška (ERROR)
10	Eho *cookie* vrednosti (COOKIE ECHO)
11	Potvrda *cookie* vrednosti (COOKIE ACK)
12	Rezervisano za eksplicitno obaveštavanje o zagušenju (ECE)
13	Rezervisano za obaveštenje o smanjenju brzine slanja (CWR)
14	Kraj procesa raskida veze (SHUTDOWN COMPLETE)
63,127,191,255	Rezervisano za buduću upotrebu
15-62, 64-126, 128-190, 192-254	Dozvoljene, ali nedefinisane vrednosti

Korisni deo se sastoji od tzv. čankova (*chunks*) koji mogu biti kontrolni čankovi ili čankovi podataka. Korisni deo može da sadrži jedan ili više čankova. Na taj način je obezbeđeno da se paralelno mogu slati poruke koje pripadaju različitim tokovima. Struktura jednog čanka je prikazana na slici 8.2.1.3.2. Čank se sastoji iz sledećih delova:

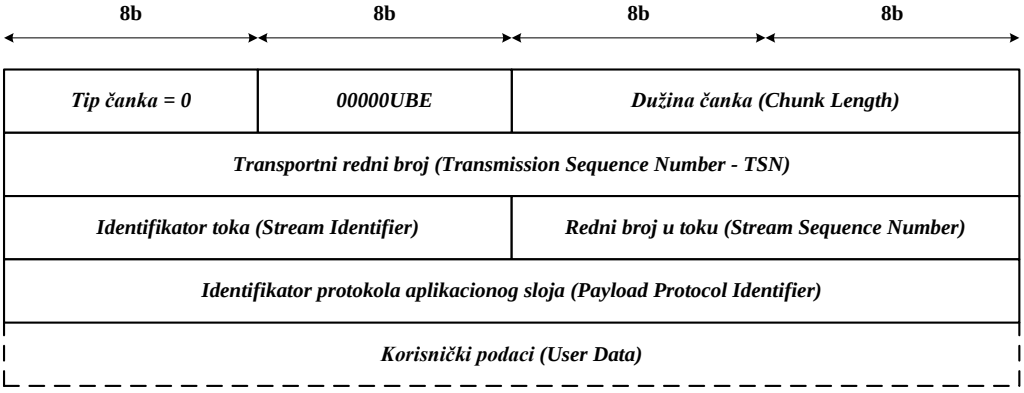
- Tip čanka (*Chunk Type*) - Ovo polje je dužine 8 bita i određuje tip informacije koju sadrži čank. Pregled tipova čankova je dat u tabeli 8.2.1.3.1. Kao što se vidi najveći broj čankova su kontrolni čankovi, a čank podataka (DATA čank) koji nosi korisničke informacije ima vrednost ovog polja 0. Takođe, može se videti da su izvesne vrednosti dozvoljene, ali nisu definisane pa ih korisnik može koristiti za sopstvene definicije tipova ako ima potrebu za tim. Rezervisane vrednosti su zauzete za eventualnu buduću upotrebu, tj. eventualno proširenje koje će definisati IETF telo. Kao što vidimo predviđena je i podrška za eksplicitno obaveštavanje o zagušenju kao i kod TCP protokola. Najviša dva bita iz polja tip čanka daju uputstvo kako da se procesira čank ako njegova vrednost nije prepoznata na prijemnoj strani. Vrednost 00 označava da se zaustavi procesiranje SCTP paketa i da se SCTP paket odbaci. Vrednost 01 označava da se zaustavi procesiranje SCTP paketa i da se SCTP paket odbaci, ali i da se u suprotnom smeru pošalje ERROR čank u kome će se navesti ova neprepoznata vrednost. Vrednost 10 označava da se preskoči neprepoznati čank i nastavi procesiranje

narednih čankova u SCTP paketu. Vrednost 11 označava da se preskoči neprepoznati čank i nastavi procesiranje narednih čankova u SCTP paketu, ali i da se u suprotnom smeru pošalje ERROR čank u kome će se navesti ova neprepoznata vrednost.

- Indikatori čanka (*Chunk Flags*) - Upotreba indikatora zavisi od tipa čanka. Indikatori koji se ne koriste u odgovarajućem tipu čanka se postavljaju na vrednost 0 i ne tumače se na prijemu. Dužina ovog polja je 8 bita.
- Dužina čanka (*Chunk Length*) - Ovo šesnaestobitno polje definiše ukupnu dužinu čanka u bajtovima (računaju se svi delovi čanka - tip, indikatori, dužina i sadržaj čanka). Pri tome, ako ukupna dužina čanka nije celobrojan umnožak 32 bita, vrši se dopuna čanka nulama tako da prostor koji čank zauzima u SCTP paketu bude celobrojan umnožak 32 bita (eventualna dopuna ne ulazi u vrednost dužine čanka).
- Sadržaj čanka (*Chunk Value*) - Ovo polje je promenjive dužine i ono nosi sam korisni sadržaj čanka. Šta je korisni sadržaj zavisi od samog tipa čanka. Strukture čankova za svaki tip čanka ponaosob su detaljno definisane u preporuci RFC 4960. Na slici 8.2.1.3.3 je prikazana struktura DATA čanka koji se koristi za prenos korisničkih podataka sa aplikacionog sloja.TSN broj predstavlja redni broj DATA čanka na nivou svih tokova, tj. na nivou SCTP veze. Razlog za uvođenje ovog globalnog brojača DATA čankova je lakša detekcija duplikata. Identifikator toka omogućava identifikaciju toka kome pripada dotični DATA čank. Redni broj u toku omogućava rekonstrukciju originalnog redosleda podataka u samom toku na prijemu. Identifikator protokola aplikacionog sloja omogućava identifikaciju kom protokolu aplikacionog sloja je namenjen dotični DATA čank (odnosno koji protokol aplikacionog sloja je kreirao korisničke podatke iz dotičnog DATA čanka). Sami podaci se nalaze u polju korisnički podaci koje je promenjive dužine. Takođe, sa slike 8.2.1.3.3 možemo videti da se koriste tri indikatora, U, B i E biti. U (*Unordered*) bit ukazuje da je u pitanju DATA čank koji ne treba da ide u redosledu podataka toka (ide van redosleda) i tada se ignoriše polje redni broj u toku. B (*Beginning*) i E (*Ending*) biti se koriste u slučaju fragmentacije poruke na više DATA čankova, da označe prvi i poslednji fragment (aktivna vrednost bita je 1). Ako je BE=00 tada je u pitanju fragment iz sredine, a BE=11 označava nefragmentisanu poruku.

Kao što je već rečeno, uspostava veze se obavlja četvorostrukim rukovanjem. Označimo SCTP korisnika koji inicira uspostavu veze kao korisnika A, a SCTP korisnika sa kojim se uspostavlja veza kao korisnika B. Korisnik A šalje INIT čank ka korisniku B, pri čemu taj SCTP paket sme da sadrži samo INIT čank. U suprotnom bi korisnik B odbacio paket i SCTP veza se ne bi uspostavila. U INIT čanku se nalazi i vrednost verifikacione oznake korisnika A za tu vezu koja se uspostavlja. Takođe, u INIT čanku, korisnik A oglašava i druge parametre veze, poput broja odlaznih tokova poruka koje želi da uspostavi sa korisnikom B, maksimalnog broja dolaznih tokova koje korisnik B može da uspostavi ka korisniku A, veličinu prijemnog bafera, inicijalni TSN broj i dr. Korisnik B odgovara sa INIT ACK čankom u kome je postavljeno polje verifikaciona oznaka na vrednost koja je primljena od korisnika A. INIT ACK čank sadrži verifikacionu oznaku koju korisnik B dodeljuje toj vezi, a takođe korisnik B javlja parametre veze korisniku A. INIT ACK se šalje na IP adresu sa koje je stigao INIT čank da bi bili sigurni

da poruku primi onaj korisnik koji je zaista i započeo uspostavu veze. U INIT ACK odgovoru se nalazi i *cookie* vrednost koja se štiti sa tajnim ključem koga zna samo korisnik B. Važno je napomenuti da korisnik B iako je poslao parametre veze poput veličine prijemnog bafera, u ovom momentu nije zauzeo nikakve resurse za ovu SCTP vezu koja se uspostavlja, inače bi bio podložan DoS napadima. Korisnik A odgovara sa COOKIE ECHO čankom u kojoj stavlja *cookie* vrednost koju je primio (u polje verifikaciona oznaka se stavlja vrednost dobijena od korisnika B). Eventualni DATA čankovi smeju da se stave iza COOKIE ECHO čanka u okviru istog SCTP paketa, ali naredni SCTP paketi ne smeju da se šalju dok se ne završi uspostava veze. Korisnik B prima COOKIE ECHO čank i proverava njegovu validnost. Na primer, ako je neki napadač generisao naslepo ovaj odgovor onda će on biti odbačen i veza neće biti uspostavljena. Ako je validan zauzimaju se resursi za tu SCTP vezu (rekonstruišu se parametri veze) i šalje se kao odgovor COOKIE ACK čank kojim se okončava uspostava veze (odnosno kada korisnik A primi ovaj čank veza je uspostavljena). Iza COOKIE ACK čanka u istom SCTP paketu mogu da idu i eventualni DATA čankovi. Napomenimo da korisnik po ulasku u uspostavljeno stanje u slučaju da suprotna strana ima više oglašenih IP adresa, bira jednu od njih kao primarnu prema kojoj će slati SCTP pakete, a ostale adrese će se koristiti za retransmitovane SCTP pakete. U slučaju da primarna adresa postane nedostupna, neka od alternativnih adresa se bira kao primarna.



Slika 8.2.1.3.3. DATA čank

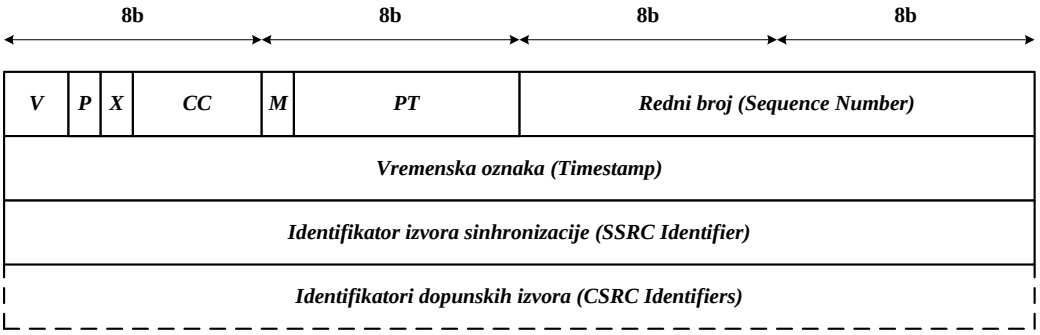
Regularan SCTP raskid veze se vrši u vidu trostrukog rukovanja. Strana koja raskida vezu šalje SHUTDOWN čank (on se šalje tek kada su pozitivno potvrđene sve njene poslate poruke). Uz SHUTDOWN čank se eventualno šalje i SACK čank da bi se oglasile eventualne rupe u pojedinim tokovima tj. nedostajući podaci u prijemnom baferu. Po prijemu SHUTDOWN čanka, SCTP veza više neće primati poruke sa aplikacionog sloja (nema mogućnosti poluotvorene veze kao kod TCP protokola). Kada suprotna strana utvrdi da su uspešno poslate sve njene poruke (sve su pozitivno potvrđene), šalje se SHUTDOWN ACK čank kao potvrda da je i suprotna strana uspešno završila sa slanjem poruka. Po prijemu ove poruke, strana koja je inicirala raskid veze će poslati SHUTDOWN COMPLETE čank i osloboditi sve resurse koje je zauzela za dotičnu vezu. Po prijemu SHUTDOWN COMPLETE čanka suprotna strana takođe oslobađa resurse koje je zauzela za dotičnu vezu čime se završava proces raskidanja veze na obe strane.

8.2.1.4. RTP i RTCP protokol

Za prenos audio i video informacija u realnom vremenu (IPTV, VoIP, Internet radio, video na zahtev i dr.) pouzdan prenos, koji podrazumeva da nema gubitaka u prenosu, nije od

esencijalnog značaja, ali jeste kašnjenje i varijacija kašnjenja. Eventualni gubici u prenosu mogu privremeno degradirati reprodukciju sadržaja, ali ako su oni retki onda kvalitet reprodukcije neće biti preterano ugrožen. Retransmisija izgubljenog paketa nije od preterane koristi jer je trenutak reprodukovanja sadržaja u retransmitovanom paketu već prošao pa je samim tim on neupotrebljiv na prijemu. Očigledno, TCP i SCTP zbog svojih mehanizama za ostvarivanje pouzdanosti nisu najbolje rešenje za prenos audio/video informacija u realnom vremenu. S druge strane, jednostavnost i princip najboljeg pokušaja, kandiduju UDP kao mogući izbor, ali i UDP protokolu nedostaju određene funkcionalnosti koje su bitne za prenos audio/video informacija u realnom vremenu. Na primer, tipično je zgodno prenositi audio i video sadržaj (za slučaj kada se prenosi video sa zvukom, što je gotovo uvek slučaj kada se prenosi video informacija) odvojeno, tako da se ima opcija prilagođenja brzine slanja i kvaliteta servisa u mreži za video i audio ponaosob, jer video zahteva veći protok od audio signala. Ako se audio i video prenose odvojeno, onda je neophodno imati i funkciju sinhronizacije audio i video sadržaja na prijemu. Takođe, potrebno je imati mogućnost nadgledanja ostvarenog kvaliteta same veze da bi se moglo u slučaju narušavanja kvaliteta adekvatno reagovati. Na primer, u slučaju da dođe do degradacije pri prenosu govornog sadržaja usled zagušenja u mreži, može se pokušati preći na koder manjeg protoka kod kojeg će procenat izgubljenih paketa možda biti manji pa će smanjenje kvaliteta govornog signala usled veće kompresije ipak biti manje od smanjenja kvaliteta govornog signala originalnog kodera usled većih gubitaka paketa. Sve ovo može se rešiti i na aplikacionom sloju, ali tada nastaje problem velikog broja implementacija (aplikacija) koje koriste najvećim delom iste ili veoma slične funkcionalnosti što nije ekonomično rešenje jer bi autori aplikacija neprestano morali da implementiraju ista rešenja u svojim aplikacijama, pa je bolje objediniti zajedničke funkcionalnosti pod jedan zajednički protokol koje bi aplikacije koristile (na uređaju bi tada više ovakvih aplikacija koristilo zajednički deo definisan odgovarajućim protokolom što bi bilo značajno ekonomičnije rešenje). Otuda je razvijen protokolski okvir koji pokriva prenos audio/video sadržaja u realnom vremenu i njega čine RTP i RTCP protokoli koji rade u međusobnoj saradnji (definisani su u RFC 3550 preporuci). RTP protokol je odgovoran za prenos samih korisničkih informacija, tj. audio i video sadržaja. Pri tome, svaki izvor audio/video sadržaja u vezi se prenosi u zasebnom RTP toku radi lakšeg održavanja i manipulisanja vezom, odnosno sadržajem koji se prenosi. Na primer, u video telefonskom razgovoru se zasebno prenosi video signal, a zasebno govorni (audio) signal pošto se oni zasebno kodiraju na predajnoj strani, odnosno zasebno dekodiraju na prijemnoj strani. Time je omogućeno i da se, u slučaju kada dođe do zagušenja u mreži, prekine slanje video signala, ali zadrži slanje audio signala da bi komunikacija ipak mogla da se nastavi i u otežanim uslovima. RTCP protokol omogućava nadgledanje ostvarenog kvaliteta RTP tokova koji zavisi od stanja u mreži, i omogućava da korisnici u vezi razmenjuju izveštaje o ostvarenom kvalitetu. Na primer, ako se detektuje pad kvaliteta govorne veze usled nedostataka resursa u mreži može se preći na koder manjeg protoka, ili ako nema dovoljno resursa u mreži za prenos video signala, može se izostaviti video signal i prenositi samo audio signal i sl. Pored toga, RTCP tok omogućava identifikaciju korisnika u vezi, kao i sinhronizaciju sadržaja iz različitih RTP tokova, na primer sinhronizaciju video i audio signala. Pošto je tipično za ostvarivanje veze između korisnika potrebno razmeniti i signalizaciju, RTP/RTCP protokoli se kombinuju sa signalizacionim protokolima poput SIP (*Session Initiation Protocol*) ili RTSP (*Real-Time Streaming Protocol*) protokola u jedinstvenu celinu koja predstavlja skup protokola neophodan za ostvarivanje određenog servisa, na primer, VoIP komunikacije. Napomenimo, da veze u kojima se koristi RTP/RTCP protokoli mogu biti interaktivne poput VoIP razgovora, ali i neinteraktivne poput

videa na zahtev, pri čemu se pod interaktivnim ovde podrazumeva aktivno učešće obe strane - u servisu video na zahtev je interaktivna samo jedna strana (korisnik), dok suprotna strana (server) samo izvršava komande primljene od aktivnog korisnika i šalje video sadržaj u skladu sa komandama korisnika. RTP i RTCP ne mogu samostalno funkcionisati na transportnom sloju, već koriste usluge UDP protokola. Pri tome, vrednost UDP porta za RTCP je za jedan veća od vrednosti koja je dodeljena za RTP kojeg kontroliše RTCP parnjak. Mogu se koristiti usluge i drugih transportnih protokola, ali u praksi se to gotovo nikad ne radi.



Slika 8.2.1.4.1. Obavezno RTP zaglavlje

RTP paket se sastoji iz obaveznog RTP zaglavlja, opcionog proširenja zaglavlja, opcionog zaglavlja korisnog dela i samog korisnog dela (u kome se nalazi audio ili video sadržaj). Obavezno RTP zaglavlje je prikazano na slici 8.2.1.4.1 i sastoji se iz sledećih delova:

- Verzija (*V - Version*) - Polje širine dva bita koje definiše verziju RTP protokola. Trenutno je u upotrebi verzija 2.
- Indikator dopune (*P - Padding*) - Bit koji ukazuje da li se koristi dopuna u RTP paketu ili ne pošto ukupna dužina RTP paketa mora biti celobrojan umnožak 32-bitnih reči. Ako je *P*=1 tada se koristi dopuna. U slučaju da koristi dopuna, poslednji bajt RTP paketa ukazuje koliki je broj bajtova dopune (i ovaj bajt se uračunava u taj broj).
- Indikator proširenja zaglavlja (*X - Extension*) - Bit koji ukazuje da li u RTP paketu postoji proširenje zaglavlja ili ne. Ako *X*=1 tada postoji proširenje zaglavlja. Proširenje zaglavlja (ako postoji) uvek sledi obavezno RTP zaglavlje.
- Broj CSRC identifikatora (*CC - CSRC Count*) - Ovo polje dužine četiri bita predstavlja broj CSRC identifikatora koji se stavljaju na kraj obaveznog RTP zaglavlja ako postoje.
- Marker (*M - Marker*) - Koristi se za označavanje značajnih događaja u RTP toku. Precizna definicija zavisi od toga šta se prenosi, tj. koji RTP profil i tip sadržaja se koriste. Na primer, u prenosu govornog signala, ovaj bit može da označava kraj perioda tišine i ponovni početak korisnog govornog signala. Međutim, pošto paketi sa setovanim markerom mogu biti izgubljeni, aplikacije se kreiraju tako da im marker služi samo kao dodatno obaveštenje, ali ne smeju u potpunosti zavisiti od markera. Na primer, aplikacija za razgovor mora biti sposobna da i bez markera detektuje kraj perioda tišine (možda ta detekcija neće biti kvalitetna kao sa upotrebom markera, ali mora da postoji jer u suprotnom može doći do

značajnog narušavanja kvaliteta razgovora ako bi se oslanjali samo na RTP pakete sa setovanim markerom).

- Tip sadržaja (*PT - Payload Type*) - Ovo sedmobitno polje definiše koji sadržaj se prenosi u RTP paketu da bi se moglo izvršiti korektno njegovo dekodiranje i reprodukcija. Na primer, vrednost 8 označava da je u pitanju G.711 kodiran sadržaj po A zakonu kompresije po RFC 3551 preporuci ako se koriste difolt vrednosti definisane u odgovarajućim RFC preporukama (jedna od njih je upravo pomenuta RFC 3551 preporuka). Međutim, signalizacioni protokoli (poput SIP) mogu da razmene i drugačija tumačenja vrednosti ovog polja, pa će se tako razmenjeno mapiranje koristiti za tumačenje ovog polja. Dodajmo da bajt koji sadrži marker (M) i tip sadržaja (PT), može i drugačije da se tumači u smislu granice između polja M i PT. Na primer, u zavisnosti od profila koji se koristi, M polje može da se proširi nauštrb PT polja.
- Redni broj (*Sequence Number*) - 16-bitni redni broj paketa koji se šalje. Na uspostavi veze se definiše slučajna početna vrednost od koje se broje paketi. Ovaj broj se inkrementira za 1 za svaki poslati RTP paket dotičnog toka. Na prijemu, se ovo polje koristi za identifikaciju redosleda paketa, i što je još važnije njihovih gubitaka što omogućava prijemniku da primeni neku od tehnika za ublažavanje gubitaka. Na primer, u slučaju video toka se može ponoviti prethodni frejm umesto izgubljenog frejma.
- Vremenska oznaka (*Timestamp*) - Predstavlja 32-bitni trenutak generisanja prvog bajta odsečka koji se nalazi u RTP paketu i koristi se za određivanje trenutka reprodukcije sadržaja na prijemu. Početna vrednost od koje se računa vreme se bira na slučajan način, pošto su na prijemu bitne samo apsolutne razlike u odnosu na početak računanja (jer je to bitno za određivanje tajminga reprodukcije odsečaka). Vremenske oznake se određuju na osnovu internog takta koji se koristi za semplovanje sadržaja ili za reprodukciju sadržaja (ako je u pitanju snimljeni sadržaj koji se pušta u realnom vremenu - na primer, video na zahtev) i one su monotono rastuće što može biti problem za neke aplikacije. Na primer, ako želimo da u aplikaciji video na zahtev imamo informaciju na kom trenutku videa se nalazimo, ne možemo se vezati za vremenske oznake jer osim normalnog puštanja videa, postoje i opcije premotavanja unapred i unazad koje bi poremetile odnos izmerenog vremena pomoću vremenskih oznaka i realnog vremena tačke u videu do koje smo došli premotavanjem. Zato se u aplikaciji video na zahtev, upotrebom RTCP protokola šalju nova mapiranja vremenskih oznaka u odnosu na trenutak u videu gde se nalazimo da bismo i dalje imali korektnu informaciju o tome na kojoj poziciji (vremenskom trenutku) se trenutno nalazimo u puštenom videu. Brzina internog takta zavisi od same aplikacije, tj. tipa sadržaja koji se prenosi. Na primer, za G.711 sadržaj bi brzina internog takta bila 8kHz. Napomenimo da neki koderi mešaju redosled odsečaka u odnosu na trenutak generisanja. Na primer, MPEG koder na predaji šalje odsečke izmešane, tako da redni broj (*sequence number*) odražava redosled slanja, ali ne i redosled generisanja odsečaka, dok vremenska oznaka ne odražava u ovoj situaciji redosled slanja, ali odražava redosled generisanja tih odsečaka.

- Identifikator izvora sinhronizacije (*SSRC - Syncronization Source Identifier*) - Ovaj 32-bitni identifikator identifikuje izvor RTP toka, tj. učesnika u komunikaciji. Prijemnici grupišu pakete po SSRC identifikatoru jer on jedinstveno definiše RTP tok. Ovom identifikatoru je pridružen takođe jedinstven naziv CNAME koji se razmenjuje preko RTCP protokola. SSRC se bira na slučajan način, pri čemu se koristi poseban algoritam za minimizaciju verovatnoće kolizije, a to je slučaj kada dva RTP toka u vezi izaberu isti SSRC identifikator. Ako se kolizija ipak desi, ponovo će se generisati novi SSRC identifikatori, pri čemu će pre toga da se pošalju RTCP BYE paketi kojim se obaveštavaju svi korisnici koji primaju dotične RTP tokove da se napuštaju dotični SSRC identifikatori. Svaki RTP tok ima jedinstveni SSRC identifikator. Na primer, ako je u pitanju prenos audio/video signala, tada će se definisati poseban RTP tok za video, a poseban za audio signal i oba će imati sopstveni SSRC identifikator (iako je u pitanju isti korisnik koji ih generiše i koji će imati isti CNAME u oba slučaja, tj. za oba toka) da bi se na prijemu moglo razlikovati kom toku pripadaju primljeni RTP paketi.
- Identifikator dopunskog izvora (*CSRC - Contributing Source Identifier*) - Najčešći slučaj je da postoji samo jedan izvor sadržaja toka i on je označen preko SSRC identifikatora. Međutim, postoje situacije kada mogu da postoje i dodatni izvori koji doprinose sadržaju (tj. generišu deo sadržaja) RTP toka i oni se navode preko 32-bitnih CSRC identifikatora. Broj ovih dopunskih izvora se navodi u CC polju i svaki od njih se navodi u vidu CSRC identifikatora tako da je ovo polje promenjive dužine jer zavisi od vrednosti CC polja. Ako je vrednost CC polja jednaka 0, tada ne postoje dopunski izvori i ovo polje se izostavlja (tj. ovo polje je opciono i ne mora uvek da se javi u obaveznom RTP zaglavlju). Tipično se dopunski izvori javljaju ako se koriste tzv. mikseri koji primaju na svom ulazu više RTP tokova, a potom na izlazu generišu jedan RTP tok u kome su izmešani (miksovani) sadržaji RTP tokova sa ulaza (na primer, u konferencijskoj vezi se mogu koristiti mikseri).

Ako se pogleda preporuka RFC 3550 koja definiše RTP i RTCP, može se videti da oni nisu isuviše kruto definisani. Naime, cilj je bio da RTP/RTCP podrže aplikacije koje zahtevaju prenos audio/video signala u realnom vremenu, a broj takvih je aplikacija velik, pri čemu mnoge od njih imaju različite zahteve. Ideja je bila da RTP/RTCP budu fleksibilno definisani tako da mogu da podrže sve aplikacije. Na primer, video na zahtev pušta već formirani sadržaj i ima opcije za skakanje na različite tačke u video sadržaju (premotavanje), dok audio (telefonski) razgovor formira sadržaj u hodu dok veza traje, pa je očigledno da se ove dve aplikacije ipak razlikuju u pojedinim zahtevima za njihovo kvalitetno opsluživanje. Otuda se definišu tzv. RTP profili koji definišu preciznije RTP/RTCP ponašanje i zajednički su za aplikacije koje imaju slične zahteve. U okviru njih se definišu koji tipovi sadržaja su podržani, a takođe i format tog korisnog (audio ili video) sadržaja koji se stavlja u RTP paket (tj. format korisnog dela RTP paketa) za svaki tip ponaosob. Na primer, za parametarske kodere će biti definisano kojim redosledom se stavljaju kodirani parametri u korisni deo RTP paketa. Profili između ostalog mogu definisati šta su značajni događaji i kako se oni detektuju, kako se koristi marker u RTP zaglavlju, koja je dužina identifikacije tipa sadržaja u bitima (tj. da li je ono smanjeno nauštrb proširenja polja marker), da li se koriste zaštitni enkripcijski mehanizmi, koji se niži protokoli

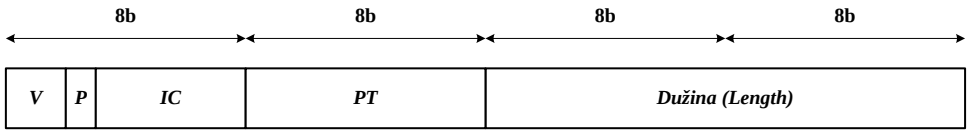
koriste, proširenja osnovne definicije RTCP protokola da podrži i dodatne poruke koje su bitne za profil (tj. aplikacije pokrivene profilom), a nisu definisane u osnovnoj verziji, da li se koristi opciono proširenje zaglavlja (ako obavezno zaglavlje ne ispunjava sve potrebe aplikacija pokrivenih profilom), itd. Profil može da obuhvati više različitih tipova sadržaja što je i logično jer iste aplikacije mogu da koriste različite kodere, na primer, za VoIP aplikaciju se mogu koristiti G.711, G.726, G.729 i mnogi drugi audio koderi. Za svaki tip sadržaja se definiše njegova struktura korisnog dela RTP paketa (na primer, za parametarske kodere se definiše kojim redosledom se stavljaju kodirani parametri u korisni deo RTP paketa). Takođe, ukoliko tip sadržaja zahteva i dodatne kontrolne informacije, definiše se i zaglavlje korisnog dela RTP paketa u koji se stavljaju kontrolne informacije specifične samo za taj tip sadržaja. Kao što vidimo profili i tipovi sadržaja omogućavaju veoma fleksibilnu upotrebu RTP i RTCP protokola. U RFC 3551 je definisan profil za audio i video konferencije sa minimalnom kontrolom koji obuhvata najpoznatije audio i video kodere. Ovaj profil se može koristiti i za aplikacije koje podrazumevaju komunikaciju samo dva korisnika jer je to u suštini specijalan slučaj konferencijske veze. Tabela 8.2.1.4.1 daje listu audio kodera statički pokrivenih ovim profilom i njihove identifikacije tipa sadržaja koje se stavljaju u PT polje obaveznog RTP zaglavlja. Termin statički podrazumeva da je dodeljena PT vrednost fiksna, a postoje i dinamički pokriveni koderi kod kojih PT vrednost nije fiksno dodeljena. G.711 koder po A zakonu kompresije ima samo odbirke govornog signala u korisnom delu RTP paketa, i ne koristi zaglavlje korisnog dela.

Tabela 8.2.1.4.1. Tipovi audio sadržaja profila definisanog u RFC 3551

PT vrednost	Naziv kodera
0	G.711 (μ zakon)
3	GSM
4	G.723
5	DVI4 (8kHz)
6	DVI4 (16kHz)
7	LPC
8	G.711 (A zakon)
9	G.722
10	L16 (2 audio kanala)
11	L16 (1 audio kanal)
12	QCELP
13	CN
14	MPA
15	G.728
16	DVI4 (11kHz)
17	DVI4 (22kHz)
18	G.729

RTP podržava upotrebu transkodera. Transkoder vrši prijem RTP toka, dekoduje sadržaj odgovarajućim dekoderom, pa potom kodira dekodovani sadržaj drugim koderom i tako kodirani sadržaj ubacuje u izlazni RTP tok. Transkoder se može koristiti kada korisnici nemaju mogućnost upotrebe istog kodera ili kada se vrši prelaz između mreža različitih karakteristika, na primer, mreže koja dozvoljava visoke protoke i mreže koja podržava samo niske protoke pa se vrši prevođenje iz kodera visokog protoka u koder niskog protoka i obrnuto u zavisnosti od smera RTP paketa.

RTCP protokol radi u saradnji sa RTP protokolom. Pravilo je da broj UDP porta ka RTP toku bude paran broj, a ka RTCP toku (koji je odgovoran za dotični RTP tok) neparan broj koji je za jedan veći od broja dodeljenog RTP toku (na primer, ako se UDP port 5004 dodeli RTP toku, tada će odgovarajućem RTCP toku da se dodeli UDP port 5005). Ovo pravilo je u početku bilo striktno, međutim, danas više ne mora da se poštuje, ali u praksi se uglavnom i dalje koristi ovo navedeno pravilo ukoliko ga je moguće primeniti. RTCP je nadležan za periodično obaveštavanje o trenutnom kvalitetu prijema RTP toka, o identifikaciji trenutnih učesnika u vezi, kao i informacija neophodnih za sinhronizaciju više RTP tokova (prvenstveno se misli na sinhronizaciju audio i video signala na prijemu, a oni se prenose različitim RTP tokovima).



Slika 8.2.1.4.2. RTCP zaglavlje

RTCP paket se sastoji iz zaglavlja i korisnog dela. RTCP zaglavlje je prikazano na slici 8.2.1.4.2 i sastoji se iz sledećih delova:

- Verzija (*V - Version*) - Polje širine dva bita koje definiše verziju RTCP protokola. Trenutno je u upotrebi verzija 2.
- Indikator dopune (*P - Padding*) - Bit koji ukazuje da li se koristi dopuna u RTCP paketu ili ne, pošto ukupna dužina RTCP paketa mora biti celobrojan umnožak 32-bitnih reči. Ako je *P*=1 tada se koristi dopuna. U slučaju da koristi dopuna, poslednji bajt RTCP paketa ukazuje koliki je broj bajtova dopune (i ovaj bajt se uračunava u taj broj).
- Broj dodatnih delova (*IC - Item Count*) - Ovo polje dužine pet bita predstavlja broj dodatnih delova u RTCP paketu. Pojedini RTCP paketi pored osnovnog dela mogu sadržati i dodatne delove, ili mogu sadržati samo dodatne delove (ako ne postoji obavezni deo u dotičnom tipu RTCP paketa), što zavisi od samog tipa RTCP paketa. Od tipa RTCP paketa zavisi i šta se podrazumeva pod dodatnim delom. Ukoliko je *IC*=0 tada nema dodatnih delova u RTCP paketu.
- Tip sadržaja (*PT - Payload Type*) - Ovo osmobitno polje definiše koji sadržaj se prenosi u RTCP paketu da bi se moglo izvršiti njegovo korektno tumačenje na prijemu. Standardni tipovi koji se koriste su izveštaji prijemnika (*RR - Receiver Reports*), izveštaji predajnika (*SR - Sender Reports*), opis izvora (*SDES - Source Description*), kontrola učešća (BYE), aplikaciono definisani RTCP paketi (*APP - Appication Defined RTCP Packets*). Izveštaji prijemnika imaju *PT*=201 i koriste se za obaveštavanje o kvalitetu prijema (kumulativan broj izgubljenih paketa, procena vrednosti džitera, trenutni procenat gubitaka paketa i dr.). Izveštaji predajnika imaju *PT*=200 i koriste se za obaveštavanje o tipovima sadržaja koji se šalju (audio, video), ali i drugim relevantnim informacijama koje pomažu prijemnoj strani da korektno izvrši sinhronizaciju tokova, pre svega sinhronizaciju video i audio signala. Opis izvora ima *PT*=202 i koristi se za identifikaciju učesnika u vezi i pružanje dodatnih informacija o učesnicima (mejl adresa, lokacija, telefonski broj i sl.). U okviru opisa izvora se koristi CNAME parametar kao jedinstveni identifikator učesnika u vezi. CNAME u RTCP je svojevrsni

ekvivalent SSRC identifikatoru iz RTP, pošto oba jedinstveno identifikuju učesnika u vezi. Učesnik ima jedinstven i SSRC i CNAME, samo što u slučaju kada korisnik generiše više RTP tokova, poput audio i video toka, tada korisnik ima i više SSRC identifikatora koji su i dalje jedinstveni. CNAME je stoga pouzdaniji identifikator jer je korisniku dodeljena samo jedna CNAME vrednost nezavisno od broja RTP tokova koje generiše, a i jer se SSRC vrednost može i promeniti tokom veze, ako, na primer, dođe do kolizije u SSRC vrednostima učesnika. CNAME je veoma bitan parametar jer se preko njega na RTCP nivou mogu identifikovati tokovi koji potiču od istog korisnika (a tokovi će imati različit SSRC identifikator iako potiču od istog korisnika) čime je omogućeno da RTCP može da pomogne u sinhronizaciji različitih RTP tokova istog korisnika poput audio i video toka. RTCP paketi kontrole učešća (*PT*=203) se koriste da se signalizira drugim korisnicima napuštanje veze korisnika koji je poslao takav paket ili da je dotični korisnik doživeo koliziju SSRC vrednosti i da je menja na novu vrednost. Aplikaciono definisani RTCP paketi (*PT*=204) pružaju mogućnost korisnicima da u okviru ovih paketa prenose kontrolne informacije specifične za samu aplikaciju, a koje nisu pokrivene prethodno opisanim tipovima RTCP paketa.

- Dužina (Length) - Ovo šesnaestobitno polje definiše dužinu korisnog dela RTCP paketa u 32-bitnim rečima. Dužina može biti i 0, čime se označava da nema korisnog dela - paket se sastoji samo iz RTCP zaglavlja.

Slanje RTCP paketa ne sme da bude prečesto da se ne bi nepotrebno preopteretili mrežni resursi, ali i da se ima dovoljno vremena da se prikupi dovoljno uzoraka za kvalitetne izveštaje. S druge strane, slanje RTCP paketa ne sme da bude ni preterano retko jer bi tada mogli isuviše kasno da se detektuju potencijalni problemi, ali i zato što bi moglo doći do desinhronizacije audio i video signala ako se vrši njihova sinhronizacija na prijemu. Otuda se prosečna frekvencija slanja proračunava na osnovu parametara poput propusnog opsega (protoka) dodeljenog RTCP toku, prosečne veličine primljenih i poslatih RTCP paketa, ukupnog broja učesnika u vezi i procenta onih koji generišu RTP saobraćaj. Tipično se uzima da ukupan protok RTCP paketa svih korisnika bude oko 5% protoka sesije (veze), odnosno ukupnog protoka svih RTP tokova koji čine sesiju. Ovo predstavlja problem za veze koje sadrže velik broj učesnika jer čak i ako je u pitanju multikast veza gde samo jedan izvor šalje multimedijalni sadržaj, svejedno svi učesnici moraju da generišu RTCP pakete (prijemnici sadržaja uvek moraju da generišu RR tip RTCP paketa). Stoga, da bi RTCP protok ostao u predviđenom procentu protoka sesije, RTCP paketi moraju da se šalju veoma retko (za ogroman broj učesnika, perioda slanja može da bude reda veličine sata), što dovodi do problema kvaliteta nadgledanja veze tj. sesije preko RTCP protokola. Otuda je velika pažnja posvećena metodama koje pokušavaju da razreše ovaj problem.

Da bi se uštedelo na mrežnim resursima i poboljšalo procesiranje RTCP paketa, svi generisani tipovi RTCP paketa se ne šalju zasebno već u okviru jednog grupnog RTCP paketa, pri čemu se pojedinačni RTCP paketi lepe jedan za drugim tako da pojedinačni RTCP paketi i dalje zadržavaju svoju strukturu, samo će zaglavlja nižih slojeva biti zajednička jer se šalje samo jedan paket. Na prijemu, kada se od UDP protokola dobije korisni deo UDP paketa, redom će se procesirati jedan po jedan RTCP paket. Na osnovu polja dužina u zaglavlju pojedinačnog RTCP paketa se može proračunati gde se završava tekući RTCP paket, a počinje sledeći. Eventualna

dopuna se radi samo na poslednjem RTCP paketu u grupi RTCP paketa, a dodatno pravilo je i da prvi (pojedinačni) RTCP paket u nizu mora biti SR ili RR tip RTCP paketa.

8.2.2. Mrežni sloj

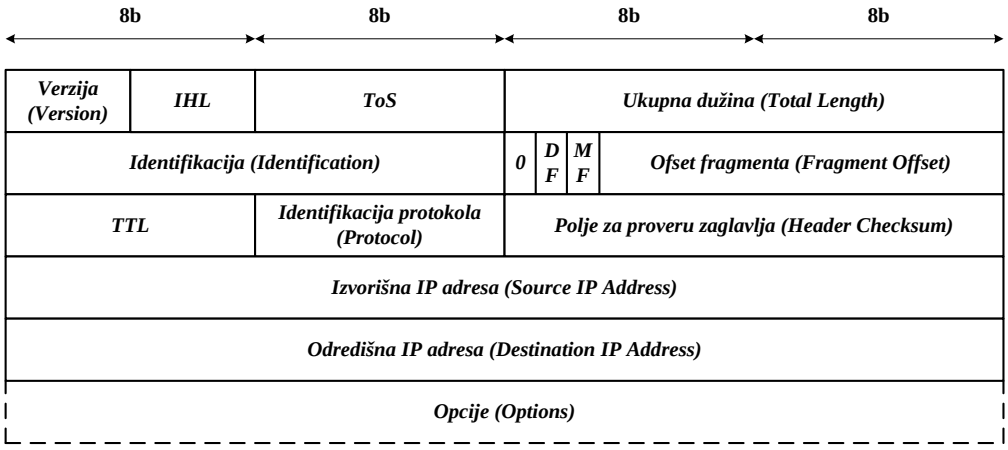
Prvenstvena uloga mrežnog sloja je da obezbedi prenos paketa na krajnje odredište (funkcija usmeravanja) i da obezbedi jedinstvenu adresu svakom korisniku (funkcija adresiranja) da bi uopšte mogla da se izvrši funkcija usmeravanja. Mrežni sloj i slojevi ispod njega su sastavni delovi mrežnih uređaja (rutera), sem u mrežama u kojima se funkcija usmeravanja vrši na drugom sloju poput eternet LAN mreže (u njima svičevi izvršavaju funkciju usmeravanja na drugom sloju tj. sloju linka podataka). Viši slojevi (slojevi iznad mrežnog sloja) su sastavni delovi hostova i odgovorni su za komunikaciju s kraja na kraj, pa se otuda tipično ne implementiraju u mrežnim uređajima. Gejtvej mrežni uređaji mogu da implementiraju ove više slojeve radi ostvarivanja mogućnosti komunikacije između različitih tehnologija.

Internet mreža je zasnovana na IP mrežnom protokolu, pa se često protokolska arhitektura po slojevima prikazuje u vidu peščanog sata, u čijem je najužem delu IP protokol. Naime, svi ostali slojevi sadrže više različitih protokola, kao što smo mogli i videti za transportni sloj prethodno u tekstu. Samo mrežni sloj implementira jedan protokol, a to je IP protokol. Međutim, u opticaju su dve verzije IP protokola, verzije 4 i 6 (IPv4 i IPv6). IPv4 je još uvek dominantniji u upotrebi, ali velika mana IPv4 verzije je suviše mali adresni prostor koji je već potrošen, pa je prelaz na IPv6 protokol, koji nudi veći adresni prostor, neminovan. Iako su predviđanja očekivala bržu tranziciju na IPv6 verziju, tranzicija je, ipak, još uvek u toku. Napomenimo da IP nije jedini mrežni protokol koji postoji, već da postoje i drugi mrežni protokoli koji se koriste u drugim mrežnim tehnologijama poput DECnet, IPX (*Internetwork Packet Exchange*) i dr. Međutim, pošto se u Internet mreži koristi samo IP, i pošto Internet tehnologija dominira tržištem, uglavnom se u praksi sreće IP mrežni protokol.

8.2.2.1. IPv4 protokol

IPv4 protokol je protokol mrežnog sloja koji predstavlja okosnicu Internet mreže. Između ostalog, zahvaljujući jednostavnosti i robusnosti IPv4 mrežnog protokola, Internet mreža je doživela globalnu popularnost. IP protokol je originalno razvijen u vojne svrhe. Američkoj vojsci je bio neophodan mrežni protokol koji bi omogućio da se paketi između proizvoljna dva korisnika mogu razmenjivati sve dok postoji bar jedan fizički put između njih kroz mrežu. Očigledno, ideja je bila da komunikacija bude moguća i u slučajevima značajnih oštećenja u mrežnoj infrastrukturi što je veoma verovatan događaj u ratnim uslovima. Naravno, da bi IP protokol mogao da izvršava svoju funkciju usmeravanja, moraju se definisati tzv. protokoli rutiranja (RIP, OSPF, BGP i dr.) koji su zaduženi da ruteri razmenjuju između sebe informacije o njima poznatim odredištima i tako podese tabele usmeravanja na osnovu kojih se izvršava funkcija usmeravanja. Ruteri su mrežni uređaji koji vrše funkciju usmervanja na osnovu odredišne adrese korisnika (odredišne IP adrese u paketu). Kada paket stigne u ruter, izvršava se pretraga tabele usmeravanja na bazi odredišne adrese u paketu i potom se na osnovu rezultata pretrage, paket usmerava na izlazni port koji vodi do traženog odredišta, pri čemu izlazni port, ustvari, predstavlja rezultat pretrage tabele usmeravanja. IPv4 protokol je definisan u RFC 791 preporuci.

IPv4 paketi se još označavaju i terminom datagrami. IPv4 paket se sastoji iz IPv4 zaglavlja i IPv4 korisnog dela. Struktura IPv4 zaglavlja je prikazana na slici 8.2.2.1.1 i sastoji se iz sledećih delova:



Slika 8.2.2.1.1. IPv4 zaglavlje

- Verzija (*Version*) - Četvorobitno polje verzije IP protokola. Za IPv4 verziju, vrednost ovog polja je 4.
- Dužina IP zaglavlja (*IHL - Internet Header Length*) - Ovo četvorobitno polje predstavlja dužinu IP zaglavlja u 32-bitnim rečima. Minimalna vrednost ovog polja je 5 (samo obavezni deo IP zaglavlja).
- Tip servisa (*ToS - Type of Service*) - Ovo osmobitno polje se originalno definisalo kao tip servisa kojim se označava klasa servisa korisničkih podataka (sadržaja) koji se nalaze u korisnom delu IP paketa. Namena ovog polja je bila ostvarivanje kvaliteta servisa u IP mrežama, na primer, tako što bi se svakoj klasi dodelio nivo prioriteta za opsluživanje paketa, čime bi prioritetniji paketi bili brže opsluživani i time brže stizali do svojih odredišta. U RFC 2474 najviših 6 bita ovog polja je definisano kao DSCP (*Differentiated Services CodePoint*) polje. DSCP polje se koristi za podršku tzv. diferencijalnim servisima (DiffServ) za ostvarivanje kvaliteta servisa (*QoS - Quality of Service*) u IP mrežama. Diferencijalni servisi podrazumevaju klasifikaciju servisa u mrežama, tako da ruteri svaku klasu servisa opslužuju prema njihovim potrebama. Na primer, prenos striming audio i video sadržaja zahteva mala kašnjenja i varijacije kašnjenja pa će takav saobraćaj da ima visok prioritet opsluživanja, dok na primer, pristup veb stranicama i skidanje fajlova nije vremenski kritično pa će takav saobraćaj da se opslužuje sa manjim prioritetom opsluživanja. Ideja diferencijalnih servisa je da obezbede jednostavne mehanizme u mreži za ostvarivanje željenog kvaliteta servisa, ali problem je što nema garancija kvaliteta servisa (ostvaruju se samo statističke garancije kvaliteta servisa). Pored pristupa diferencijalnih servisa, u praksi postoji i pristup integrisanih servisa (IntServ) koji obezbeđuju garancije kvaliteta servisa, tako što se u ruterima zauzimaju odgovarajući resursi za svaki tok za koji se želi garancija kvaliteta servisa, čime se postiže garancija kvaliteta servisa (na primer, RSVP (*Resource ReserVation Protocol*) protokol se može koristiti u tu svrhu zauzimanja resursa). Ukoliko se ne može garantovati kvalitet servisa (nema dovoljno slobodnih resursa u ruteru), korisnik će biti obavešten, pa korisnik može odlučiti da li će da odustane od veze ili će da nastavi vezu sa smanjenim zahtevima u pogledu kvaliteta servisa koji mogu da se garantuju ili će da uspostavi vezu pri

čemu neće biti garancija kvaliteta servisa za nju. Naravno, integrisani servisi stavljaju veće opterećenje na rutere, pri čemu ruteri moraju da vode računa o svakom toku za koji su rezervisani resursi, a broj takvih tokova može biti velik. Takođe, moraju da se obezbede mehanizmi koji će omogućiti komunikaciju od rutera do rutera radi zauzimanja resursa za vezu koja se uspostavlja, a koja zahteva striktne garancije kvaliteta servisa (očigledno je, pošto se resursi zauzimaju u određenom nizu rutera, da se ovde primenjuje i svojevrstan princip komutacije kola). U RFC 3168 najniža 2 bita ToS polja su predviđena za upotrebu u mehanizmu eksplicitnog obaveštavanja o zagušenju u mreži (*ECN – Explicit Congestion Notification*), čiji su osnovni principi i namena objašnjeni u sekciji 8.2.1.1. Preko ova dva bita ruter ima mogućnost da eksplicitno obavesti korisnika (kome je namenjen dotični IP paket) da je došlo do zagušenja u mreži.

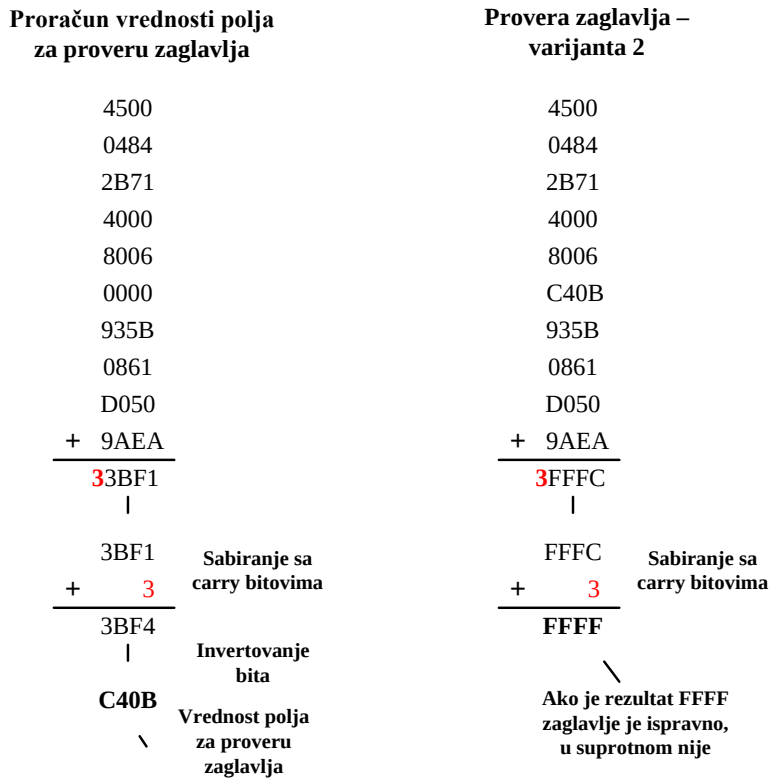
- Ukupna dužina (*Total Length*) - Ovo 16-bitno polje predstavlja ukupnu dužinu IP paketa u bajtovima, pri čemu se u obzir uzimaju i IP zaglavlje i korisni deo IP paketa.
- Identifikacija (*Identification*) - Ovu vrednost postavlja pošiljalac kao identifikaciju paketa koja se koristi u eventualnom procesu fragmentacije, preciznije u procesu rekonstruisanja originalnog paketa u slučaju da je izvršena fragmentacija. Naime, protokoli drugog sloja tipično imaju ograničenja u pogledu maksimalne dužine korisnog dela u koji se stavlja IP paket, pa u slučaju da je IP paket predugačak, ili se vrši njegova fragmentacija na više delova ili se takav paket odbacuje i pošiljalac obaveštava odgovarajućom ICMP porukom kojom se signalizira da je IP paket prevelik. U slučaju da se izvršila fragmentacija, ovo polje omogućava laku identifikaciju kom paketu pripada koji fragment, pa je samim tim omogućena rekonstrukcija originalnog paketa (nema opasnosti da se greškom izmešaju fragmenti različitih paketa). Tipično se rekonstrukcija vrši na odredištu, ali postoje i situacije kada mrežni uređaji vrše rekonstrukciju paketa. Naravno, pored ovog polja se za proces identifikacije kom paketu pripadaju fragmenti koriste i drugi delovi IP zaglavlja, izvorišna i odredišna IP adresa, kao i identifikacija protokola. Otuda je bitno, da pošiljalac IP paketa uvek bira jedinstvenu vrednost polja identifikacija za istu kombinaciju vrednosti tri polja IP zaglavlja navedena u prethodnoj rečenici.
- Indikatori (*Flags*) - Tri indikator bita. Najviši bit je rezervisan za buduću upotrebu i postavlja se na 0. Sledeći bit je DF (*Don't Fragment*) indikator kojim se u slučaju njegove aktivne vrednosti (vrednost 1) ruteru signalizira da ne vrši fragmentaciju u slučaju da je IP paket prevelik, već da pošalje ICMP (*Internet Control Message Protocol*) obaveštenje pošiljaocu kojim će ga obavestiti da je njegov paket odbačen jer je prevelik. Najniži bit je MF bit (*More Fragments*) kojim se signalizira da li je u pitanju poslednji fragment paketa ili ne. Ukoliko je vrednost ovog bita 1, tada ima još fragmenata, u slučaju da je vrednost 0, tada je u pitanju poslednji fragment.
- Ofset fragmenta (*Fragment Offset*) - 13-bitno polje kojim se daje informacija o poziciji fragmenta u originalnom paketu. Ofset prvog fragmenta je 0. Ofset se

navodi u 64-bitnoj veličini tj. vrednost ofseta predstavlja ukupan broj 64-bitnih reči korisnog dela u prethodnim fragmentima.

- Vreme života paketa (*TTL - Time To Live*) - Ovo osmobitno polje predstavlja vreme života paketa. Prilikom svakog prolaska kroz ruter, vrednost ovog polja se dekrementira za 1. Kada vrednost polja dosegne 0, paket se odbacuje čime se sprečava njegovo beskonačno kruženje u mreži, što je, ustvari, primarni cilj ovog polja.
- Identifikacija protokola (*Protocol*) - Ovo osmobitno polje predstavlja identifikaciju protokola čiji je paket enkapsuliran u IP korisni deo. Neke od poznatih vrednosti su: 1 – ICMP, 2 – IGMP, 6 – TCP, 17 – UDP, 46 – RSVP, 132 – SCTP.
- Polje za proveru zaglavlja (*Header Checksum*) - Ovo 16-bitno polje se koristi za proveru ispravnosti IP zaglavlja tj. detekciju bitskih grešaka u njemu. IP ne štiti korisni deo IP paketa. Vrednost ovog polja se proračunava kao suma u komplementu jedinice svih 16-bitnih reči zaglavlja, pri čemu se rezultujuća suma takođe predstavlja u komplementu jedinice i stavlja u polje za proveru zaglavlja. Pri proračunu se polje za proveru zaglavlja stavlja na vrednost 0. Ovo polje se proverava u svakom ruteru kroz koji prolazi paket, kao i na samom odredištu. Ukoliko se greška detektuje, paket se odbacuje. Provera ispravnosti se vrši ponavljanjem proračuna sa predaje i ako se dobije ista vrednost kao ona stavljena u polje za proveru paket je ispravan. Drugi način je da se provera ispravnosti izvrši ponavljanjem proračuna sa predaje, ali uzimajući postavljenu vrednost polja za proveru (a ne vrednost 0), i ako se proračunom dobije vrednost sve jedinice, onda je zaglavlje ispravno, u suprotnom nije.
- Izvorišna IP adresa (*Source IP Address*) - 32-bitna IP adresa korisnika (hosta) koji je formirao dotični IP paket.
- Odredišna IP adresa (*Destination IP Address*) - 32-bitna IP adresa korisnika (hosta) kome je namenjen dotični IP paket.
- Opcije (*Options*) - Ovo polje je promenjive dužine i nije obavezno (njegova upotreba je opciona). Ukoliko dužina ovog polja nije celobrojan umnožak 32-bitnih reči tada se ono dopunjava nulama tako da dopunjeno polje opcije bude celobrojan umnožak 32-bitnih reči. U praksi se ovo polje veoma retko koristi da bi se izbeglo nepotrebno opterećivanje rutera. Danas ruteri rade na gigabitskim brzinama sa tendencijom prelaska i na terabitske brzine pa procesiranje IP paketa mora biti veoma brzo. Procesiranje IP paketa podrazumeva pre svega proveru ispravnosti zaglavlja IP paketa, njegovu modifikaciju (pošto se TTL polje dekrementira, to znači da se polje za proveru zaglavlja mora ponovo proračunati), kao i određivanje na koji izlazni port rutera paket mora biti prosleđen (tzv. IP lukap (*lookup*) funkcija). Otuda je poželjno ne dodavati nepotrebno opcije koje bi povećale dužinu procesiranja paketa, a time mogle dovesti i do zagušenja rutera usled njegove nemogućnosti da dovoljno brzo obradi IP pakete.

Pokažimo na konkretnom primeru proračun polja za proveru zaglavlja. Uzmimo da je heksadecimalni sadržaj zaglavlja 4500 0484 2B71 4000 8006 **0000** 935B 0861 D050 9AEA

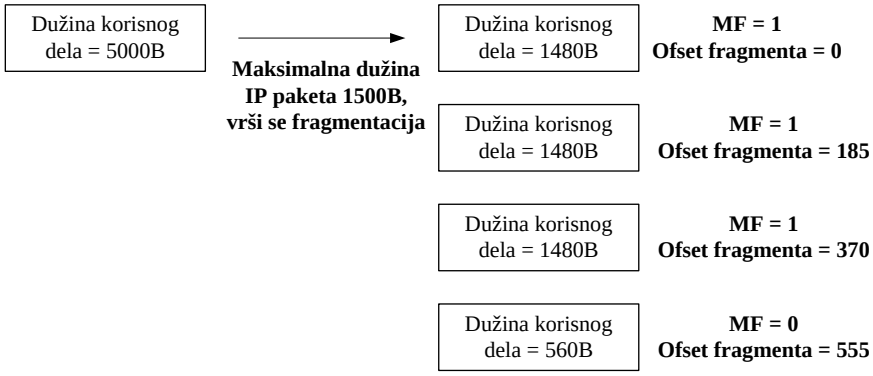
(boldovanim fontom je označeno polje za proveru zaglavlja). Polje za proveru je popunjeno nulama i želimo da proračunamo njegovu vrednost. Princip računanja je prikazan na slici 8.2.2.1.2. Kao što se vidi, prvi korak je sabiranje svih šesnaestobitnih reči zaglavlja kao da su u pitanju nenegativni brojevi, pri čemu su dozvoljeni preneseni (*carry*) biti, odnosno rezultat može biti duži od 16 bita. U drugom koraku, vrednost prenesenih bita (*carry* biti), sabiramo sa nižih 16 bita rezultata jer je dužina polja za proveru zaglavlja 16 bita. Na kraju, vrši se invertovanje bita rezultata drugog koraka, što je ekvivalentno komplementiranju u komplementu jedinice. Ta vrednost se stavlja u polje za proveru ispravnosti zaglavlja. U datom primeru tako proračunata vrednost iznosi C40B u heksadecimalnom formatu. Na prijemu se prima zaglavlje čija je vrednost 4500 0484 2B71 4000 8006 C40B 935B 0861 D050 9AEA. Ako izvršimo isti proračun kao na predaji dobićemo istu vrednost polja za proveru zaglavlja kao i na predaji, ako nije bilo grešaka u prenosu. Drugi način za proveru ispravnosti zaglavlja uzima u zbir primljenu vrednost polja za proveru. Tada je rezultat sabiranja jednak FFFF čime je potvrđena ispravnost IP zaglavlja. Ova varijanta je prikazana na slici 8.2.2.1.2. Naravno, postoji teoretska mogućnost da se dogode greške koje ne mogu da se detektuju na ovaj način, što je uostalom slučaj sa svim mehanizmima zaštite. Podsetimo se da ovaj princip zaštite koriste i TCP i UDP paketi.



Slika 8.2.2.1.2. Proračun polja za proveru zaglavlja

Kod fragmentacije treba imati na umu da će svaki fragment imati svoje IP zaglavlje, što znači da ruter koji je kreirao fragmente, mora da kreira i IP zaglavlja za njih na osnovu IP zaglavlja originalnog paketa. U odnosu na originalno zaglavlje će biti promenjeni polje ukupna dužina (fragmenti su kraći od originalnog paketa), polje fragment ofset (samo prvi fragment će zadržati originalnu vrednost ovog polja - vrednost 0) i vrednost MF indikatora (samo poslednji fragment će zadržati originalnu vrednost MF indikatora - vrednost 0), i, naravno, polje za proveru zaglavlja koje se takođe menja usled promena u ostalim navedenim poljima IP zaglavlja.

Podaci će biti raspoređeni po fragmentima tako da ukupna dužina IP paketa bude u skladu sa ograničenjem dužine zbog koje je i vršena fragmentacija. Na slici 8.2.2.1.3 je prikazan princip fragmentacije i računanje vrednosti polja ofset fragmenta, pri čemu se podrazumeva da originalni paket ima samo osnovno IP zaglavlje (nema opcija), pa samim tim i paketi dobijeni fragmentacijom imaju samo osnovno zaglavlje (opcije, ako postoje, se stavljaju u prvi fragment). Originalni IP paket iz datog primera ima ukupnu dužinu 5020B, odnosno dužina korisnog dela je 5000B. Dotični paket treba da se prosledi na link gde je maksimalna dozvoljena dužina celog IP paketa 1500B. Ako DF bit nije aktivan, ruter može izvršiti fragmentaciju paketa. Pošto je dužina osnovnog zaglavlja 20B, to znači da je maksimalan broj bajtova za korisni deo 1480. Traži se najveći ceo broj L koji ispunjava uslove $L \leq L_{max}$ i $L=8n$, gde je L_{max} maksimalan broj bajtova za koristan deo u fragmentu (u prikazanom primeru je L_{max} = 1480), a n ceo pozitivan broj. Drugi uslov je posledica formata polja ofset fragmenta, jer broj naveden u tom polju odgovara broju 64-bitnih reči. U datom primeru tražena vrednost za L je 1480, pa se korisni deo IP paketa deli na tri dela od po 1480B, a poslednji (četvrti) deo će sadržati preostalih 560B. Ovi delovi (fragmenti) korisnog dela originalnog IP paketa se dele na četiri fragmentisana IP paketa. U prva tri fragmenta će MF indikator biti postavljen na 1, a u poslednjem fragmentu će MF indikator biti postavljen na 0 čime će se signalizirati da je u pitanju poslednji fragment. Ofset fragmenta se računa kao ukupan broj 64-bitnih reči korisnog dela originalnog paketa koji je sadržan u prethodnim fragmentima. U prvom fragmentu je stoga vrednost ovog polja 0. U drugom fragmentu je vrednost polja ofset fragmenta 185 (decimalni format), jer je prvi fragment sadržao ukupno 185 64-bitnih reči (tj. 1480B) korisnog dela originalnog IP paketa. U trećem fragmentu je vrednost polja ofset fragmenta 370, jer su prvi i drugi fragment zajedno sadržali ukupno 370 64-bitnih reči (tj. 2960B) korisnog dela originalnog IP paketa. Ofset fragmenta u četvrtom fragmentu ima vrednost 555, jer je toliko 64-bitnih reči originalnog IP paketa bilo sadržano u prva tri fragmenta. Napomenimo da je cilj u mrežama da se fragmentacija što ređe koristi jer ona opterećuje rutere - ruteri moraju da formiraju nove IP pakete, na primer, za svaki od novih paketa mora da proračuna vrednost polja za proveru zaglavlja i podese vrednosti polja ofset fragmenta. Dodatno, problem može nastati ako se jedan fragment izgubi - pošto IP nema mehanizme retransmisije to je ekvivalentno gubitku čitavog paketa, pri čemu se postavlja i pitanje koje je optimalno vreme čuvanja fragmenata nekompletiranog paketa u prijemniku da ne dođe do njihovog preranog odbacivanja jer zakašnjeli fragment može stići kasnije od ostalih zbog privremenog zagušenja u mreži. Otuda, nije redak slučaj da ruteri vrate ICMP poruku da je paket prevelik, iako DF indikator nije bio aktivan. U IPv6 je uvedeno ispitivanje maksimalne dužine IP paketa koja se može poslati, upravo da bi se izbegla fragmentacija paketa.



Slika 8.2.2.1.3. Primer fragmentacije

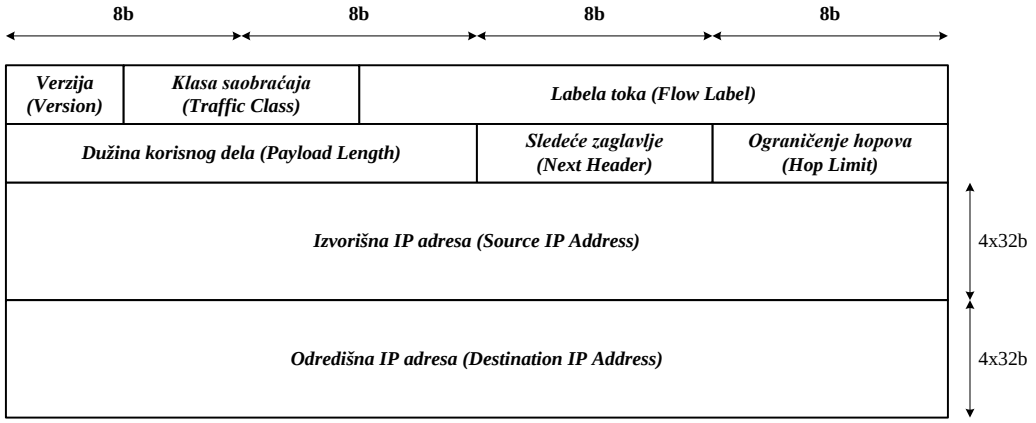
Kao što smo već rekli, tabela usmeravanja sadrži informacije o svim mrežama koje su poznate ruteru i preko kojih portova rutera se može doći do njih. IP adresa koristi hijerarhijski princip adresiranja, gde IP adresa može da se podeli na mrežni i host deo. Mrežni deo identifikuje mrežu, a host deo identifikuje hosta unutar dotične mreže (svi hostovi iste mreže imaju istu vrednost i dužinu mrežnog dela adrese). U tabelama usmeravanja se stoga čuvaju mrežne adrese. U početku se koristilo klasno adresiranje gde se na osnovu nekoliko prvih bita određivalo koja klasa adrese je u pitanju, a time i dužina mrežnog i host dela dotične adrese. Na primer, klasa A je imala 8 bita mrežni deo (prvi bit adrese ima vrednost 0), klasa B 16 bita mrežni deo (prva dva bita adrese imaju vrednost 10), klasa C 24 bita mrežni deo (prva tri bita adrese imaju vrednost 110). Naravno, preostali biti pripadaju host delu. Klasno adresiranje je omogućavalo da se tumačenjem vrednosti početnih bita odredi dužina mrežnog dela adrese i da se potom za tu vrednost pretraži tabela usmeravanja i odredi izlazni port rutera na koji treba poslati dotični paket. Klasno adresiranje je bilo adekvatno dok je Internet imao mali broj korisnika i pored neracionalne potrošnje adresnog prostora. Međutim, naglim porastom popularnosti Interneta, a time i broja korisnika došlo je do potrebe da se racionalizuje upotreba adresnog prostora, pa je uvedeno besklasno adresiranje. Besklasno adresiranja podrazumeva da je granica između mrežnog i host dela adrese proizvoljna čime je omogućena racionalnija upotreba adresnog prostora. Takođe, time je omogućena u ruterima i agregacija nekoliko mrežnih adresa (bliskih vrednosti) u jedan zajednički zapis u tabeli usmeravanja rutera ukoliko su one dostižne preko istog porta čime se smanjivao broj zapisa u tabelama usmeravanja. Međutim, javio se problem da se za jednu adresu može naći više rešenja u tabeli usmeravanja (ovo se tipično javlja used postojanja izuzetaka iz agregiranog zapisa, gde je izuzetak mrežna adresa koja je dostižna preko nekog drugog porta, od onog koji odgovara agregiranom zapisu koji pokriva i taj izuzetak tj. tu mrežnu adresu), pa je definisano LPM (*Longest Matching Prefix*) pravilo kojim se za konačno rešenje uzima onaj zapis koji se najduže poklapa sa odredišnom IP adresom paketa. Ovime je pretraga značajno komplikovana jer nije dovoljno naći rešenje, već je bitno i da to rešenje bude najbolje (pretraga je postala dvodimenzionalna, osim vrednosti, sada je bitna i dužina mrežnog dela adrese). Porastom brzine linkova, vreme dozvoljeno za nalaženje rešenja je značajno smanjeno (svega nekoliko ns za 100Gb/s linkove), a tabele usmeravanja mogu biti ogromne i sadržati i preko 400000 zapisa. Otuda je pretraga tabele usmeravanja (tzv. IP lukap funkcija) veoma zahtevna i komplikovana i kod rutera velikih kapaciteta zahteva značajne resurse za njeno efikasno obavljanje. Predloženo je mnogo rešenja za efikasno obavljanje ove funkcije, ali i dalje se radi na nalaženju još boljih i efikasnijih rešenja. Ovaj problem će biti još izraženiji kod dužih IPv6 adresa (prostor pretrage tj. adresni prostor je veći) kada tabele usmeravanja IPv6 adresa postanu dovoljno velike (za sada su one još uvek male zbog relativno slabe trenutne penetracije IPv6 protokola na Internetu).

8.2.2.2. IPv6 protokol

IPv4 protokol je radio sa 32-bitnim IP adresama, što znači da je maksimalan broj adresa bio 2^{32} (realan maksimalan broj adresa je bio manji zbog diskretne granularnosti dodele IP adresa mrežama u praksi, privatnih IP adresa, i dr.). Taj broj je zbog ogromne popularnosti Interneta, velikog broja korisnika i brojnosti uređaja povezanih na Internet (trend je da se i kućni aparati povezuju na Internet) isuviše mali. Štaviše, IPv4 prostor je u potpunosti potrošen i više nema slobodnih blokova adresa. Nedostatak adresnog prostora predstavlja najvažniju motivaciju za definisanje i uvođenje IPv6 protokola. Pošto se već morao razviti novi protokol usled nedostatka mrežnih adresa, IPv6 implementira i neke funkcionalnosti koje nisu bile adekvatno (tj. na najbolji način) pokrivene IPv4 protokolom poput multikasta, bezbednosti, mobilnosti, i drugih

funkcionalnosti što je i logično jer navedene funkcionalnosti nisu bile od (velikog) značaja u vreme nastajanja IPv4 protokola. IPv6 adrese su sada 128 bita dužine, što znači da je maksimalan adresni prostor 2^{128} (naravno, isto kao i u IPv4 slučaju adresni prostor je nešto manji iz istih razloga kao i kod IPv4 slučaja) i taj adresni prostor je dovoljan za sve buduće potrebe i neće se moći lako istrošiti. IPv6 protokol je definisan u RFC 2460 preporuci.

Napomenimo da se očekivala brža tranzicija na IPv6 protokol, ali zbog nekompatibilnosti IPv6 i IPv4 zaglavlja nije bila moguća jednostavna interoperabilnost između ove dve verzije protokola što predstavlja velik problem sa stanovišta tranzicije jer na Internetu egzistira ogroman broj mrežnih uređaja koji rade samo na IPv4 protokolu i nije ih nimalo jednostavno sve odjednom zameniti. Isto tako, velik značaj u brzini tranzicije ima i angažovanost i strategija vlasti u državama, pa su tako pojedine zemlje (na primer, SAD i Japan) ozbiljnije i ranije krenule u proces tranzicije. U svakom slučaju, trenutno IPv6 jeste neminovna budućnost ako se ne desi nešto nepredviđeno u razvoju telekomunikacija, odnosno Interneta.

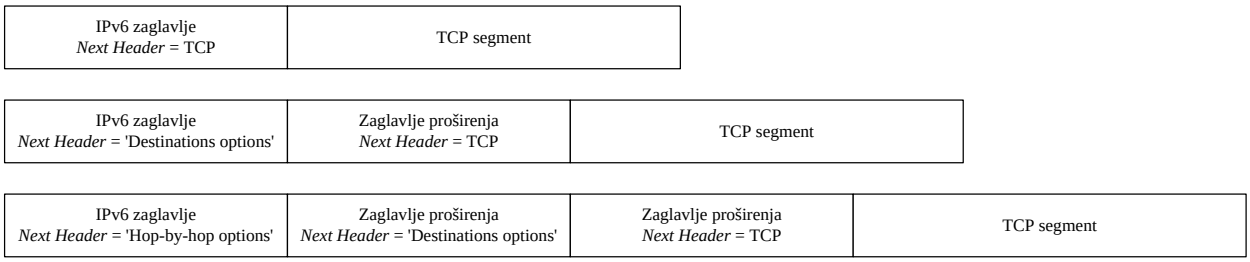


Slika 8.2.2.2.1. IPv6 zaglavlje

IPv6 paket se sastoji iz IPv6 zaglavlja i IPv6 korisnog dela. Struktura IPv6 zaglavlja je prikazana na slici 8.2.2.2.1 i sastoji se iz sledećih delova:

- Verzija (*Version*) - Četvorobitno polje verzije IP protokola. Za IPv6 verziju, vrednost ovog polja je 6.
- Klasa saobraćaja (*Traffic Class*) - Ovo osmobitno polje ima sličnu ulogu kao ToS polje u IPv4 zaglavlju, a to je da obezbedi podršku za kvalitet servisa u vidu DiffServ pristupa.
- Labela toka (*Flow Label*) - 20-bitna labela toka predstavlja identifikaciju toka kojem dotični paket pripada u kombinaciji sa izvorišnom i odredišnom IP adresom. Ideja je da se upotrebom ovog polja na jednostavan način obeleže paketi koji zahtevaju određenu obradu/opsluživanje u ruterima. Na primer, u nizu paketa koji šalje host, neki od paketa se mogu obeležiti labelom koja signalizira da dotični paketi zahtevaju drugačiji kvalitet servisa od ostalih paketa iz te komunikacije (tako obeleženi paketi bi činili jedan tok paketa). Ovo polje je još otvoreno za dalja istraživanja i praktičnu upotrebu. RFC 6437 definiše generalne okvire za ovo polje. Ono što predstavlja problem za efikasnu upotrebu ovog polja je opterećenje rutera u slučaju pojave velikog broja tokova koje mora da dodatno opsluži što bi moglo da dovede do preopterećenja samog rutera.

- Dužina korisnog dela (*Payload Length*) - Ovo 16-bitno polje predstavlja dužinu korisnog dela IP paketa u bajtovima. Korisni deo se nalazi iza obaveznog IPv6 zaglavlja (preciznije, iza polja odredišna IP adresa) i u njega se uračunavaju i eventualna zaglavlja proširenja (*extension headers*).
- Sledeće zaglavlje (*Next Header*) - Ovo osmobitno polje identifikuje zaglavlje (može se reći i protokol) koje sledi obavezno zaglavlje. Vrednosti ovog polja odgovaraju vrednostima polja identifikacija protokola iz IPv4 zaglavlja.
- Ograničenje hopova (*Hop Limit*) - Ovo osmobitno polje predstavlja ekvivalent TTL polja iz IPv4 zaglavlja i koristi se na identičan način. Napomenimo da je TTL polje originalno trebalo da predstavlja vreme u sekundama, ali se u praksi, ipak, TTL polje koristi kao brojač tzv. hopova, odnosno prolaza kroz rutere.
- Izvorišna IP adresa (*Source IP Address*) - 128-bitna IP adresa korisnika (hosta) koji je formirao dotični IP paket.
- Odredišna IP adresa (*Destination IP Address*) - 128-bitna IP adresa korisnika (hosta) kome je namenjen dotični IP paket.



Slika 8.2.2.2.2. Zaglavlja proširenja

Eventualna zaglavlja proširenja predstavljaju ekvivalent opcija iz IPv4 zaglavlja. Na njih ukazuje polje sledeće zaglavlje po principu prikazanom na slici 8.2.2.2.2. Tri primera su data na slici 8.2.2.2.2, prvi primer nema zaglavlje proširenja, drugi primer sadrži samo jedno zaglavlje proširenja i treći primer prikazuje slučaj kada IP paket sadrži dva zaglavlja proširenja. Zaglavlja proširenja moraju da se nalaze pre enkapsuliranog sadržaja drugog protokola, na primer, TCP ili UDP protokola. Zaglavlja proširenja imaju takođe polje sledeće zaglavlje za identifikaciju sledećeg zaglavlja (ili enkapsuliranog protokola) u nizu. Ono što je bitno napomenuti je da se zaglavlja proširenja tumače samo na odredištu (host čija adresa je odredišna IP adresa iz paketa), čime se izbegava problem opcija i njihovog procesiranja u ruterima kao u slučaju IPv4 protokola. Jedini izuzetak od ovog pravila je *hop-by-hop options* proširenje zaglavlja koje se mora procesirati u ruterima kao što i samo ime sugeriše. Ovo proširenje zaglavlja mora da se nalazi neposredno iza obaveznog zaglavlja sa slike 8.2.2.2.1. Proširenja zaglavlja se procesiraju u originalnom redosledu, pri čemu ako se naiđe na nepoznatu ili zabranjenu vrednost polja sledeće zaglavlje, paket se odbacuje i preko odgovarajuće ICMP poruke se obaveštava pošiljalac dotičnog paketa. Tipovi zaglavlja proširenja i njihove strukure su definisane u RFC 2460.

Iz strukture IPv6 zaglavlja se može videti da je cilj bio što prostije zaglavlje da bi se smanjilo procesiranje zaglavlja u ruterima. Za razliku od IPv4 zaglavlja, nema polja za proveru čime se izbegava njegova provera i novi proračun u svakom ruteru kroz koji paket prolazi. Zaglavlja proširenja omogućuju proširenje funkcionalnosti zaglavlja pri čemu je akcenat na tome

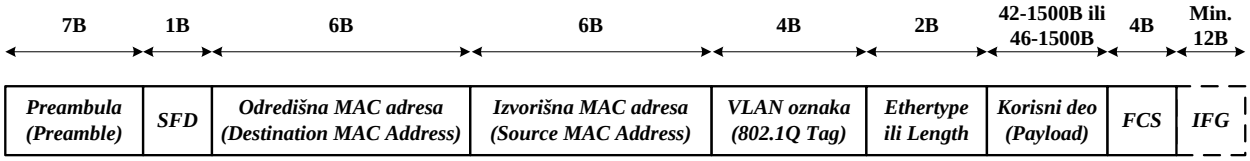
se oni procesiraju samo na odredištu, ali je ostavljena i mogućnost da se vrši i procesiranje određenih funkcionalnosti u ruterima kroz koje paket prolazi ukoliko je to potrebno.

8.2.3. Sloj linka podataka

Sloj linka podataka ima višestruke funkcionalnosti, poput formiranja okvira, razgraničavanja okvira, detekcije i korekcije grešaka i dr., pri čemu se pod okvirima podrazumevaju jedinice podataka koje se prenose na sloju linka podataka. Sloj linka podataka obavlja svoje funkcije po principu link-po-link (*link-by-link*), kao i mrežni sloj. Ovaj sloj u kombinaciji sa fizičkim slojem definiše mrežnu tehnologiju i može se reći da ta dva sloja definišu mrežni hardver, pošto se tipično najveći deo funkcionalnosti obavlja hardverski. Mrežni sloj predstavlja mešavinu hardverskog i softverskog pristupa, jer se deo funkcionalnosti obavlja u hardveru, a deo u softveru. Međutim, u slučaju nižih kapaciteta neretko se sve funkcionalnosti obavljaju softverski. Transportni i viši slojevi se tipično izvršavaju softverski. U ovoj sekciji će biti ukratko obrađeni eternet standard i PPP (*Point-to-Point Protocol*) protokol koji se veoma često sreću u praksi.

8.2.3.1. Eternet

Eternet je dominantna tehnologija u ostvarivanju LAN komunikacije. U ovoj sekciji se nećemo baviti detaljima rada eternet mreže, već ćemo samo ukratko opisati strukturu eternet okvira. Struktura eternet okvira je prikazana na slici 8.2.3.1.1 i okvir se sastoji iz sledećih delova:



Slika 8.2.3.1.1. Eternet okvir

- Preambula (*Preamble*) i SFD (*Start Frame Delimiter*) - Preambula se sastoji od 7 bajtova vrednosti 10101010, a SFD od jednog bajta vrednosti 10101011 (pri tome se prvo prenose niži biti u bajtu tj. koristi se *little endian* princip). Preambula i SFD se koriste za detekciju početka okvira.
- Odredišna MAC adresa (*Destination MAC Address*) - Odredišna MAC adresa hosta kojem je okvir namenjen. MAC adrese su dužine 48 bita, tj. 6 bajtova. Važno je napomenuti da su MAC adrese jedinstvene na globalnom nivou i da se koristi princip ravnog (*flat*) adresiranja u eternet LAN mrežama. Na osnovu odredišne MAC adrese se vrši prosleđivanje u eternet svičevima po stablu LAN mreže kreiranim STA (*Spanning Tree Algorithm*) algoritmom.
- Izvorišna MAC adresa (*Source MAC Address*) - Izvorišna MAC adresa hosta koji je kreirao okvir.
- VLAN oznaka (*802.1Q Tag*) - Ovo polje je dužine 4 bajta i definiše VLAN mrežu kojoj dotični okvir pripada. Ovo polje je opciono, tj. koristi se samo u slučaju da je konfigurisana upotreba VLAN mreže (tipično se u slučaju prenosa VoIP saobraćaja koristi VLAN mreža za prenos VoIP saobraćaja radi mogućnosti bolje kontrole kvaliteta servisa). Viših 16 bita ima heksadecimalnu vrednost 8100 čime se identifikuje da okvir sadrži VLAN oznaku. Nižih 16 bita se sastoji od 3-bitnog polja PCP (*Priority Code Point*) koji definiše prioritet okvira u opsluživanju –

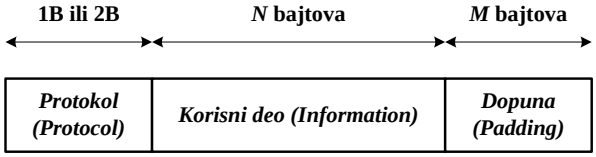
veća vrednost označava viši prioritet; od 1-bitnog indikatora DEI (*Drop Eligible Indicator*) koji se koristi kao indikator da li je dozvoljeno odbaciti okvir u slučaju zagušenja, pri čemu se indikator tipično koristi u kombinaciji sa PCP poljem; od 12-bitne identifikacije VLAN mreže kojoj pripada okvir (*VID - VLAN Identifier*). VID vrednosti sve 0 i sve 1 su zabranjene tj. ne smeju se dodeliti VLAN mrežama kao njihovi identifikatori. VID vrednost sve 0 označava da okvir ne pripada nijednoj VLAN mreži i tada se koriste samo PCP i DEI polja za određivanje prioriteta u opsluživanju i ponašanje pri zagušenju u sviču.

- Tip korisnog sadržaja (*Ethertype*) ili dužina korisnog dela (*Length*) - Tumačenje ovog polja dužine dva bajta zavisi od tipa eternet okvira. U slučaju da se koristi Ethernet II format tada se ovo polje tumači kao tip enkapsuliranog sadržaja u korisnom delu okvira (*Ethertype*). Poznate vrednosti su (u heksadecimalnom formatu): 0800 – IP, 0806 – ARP, 22F3 – TRILL, 86DD – IPv6, 8847 – MPLS unikast, 8848 – MPLS multikast i dr. Važno je napomenuti da je ARP (*Address Resolution Protocol*) protokol neophodan za određivanje MAC adrese korisnika sa kojim želimo komunicirati na osnovu poznate IP adrese tog traženog korisnika, što je tipična situacija na eternet mreži. U slučaju da se koristi 802.3 format okvira tada se ovo polje tumači kao dužina korisnog dela u bajtovima (*Length*). Tumačenje ovog polja se može odrediti na osnovu vrednosti samog polja. Ukoliko je vrednost polja (u heksadecimalnom formatu) manja ili jednaka 05DC, tada je u pitanju 802.3 format, a ukoliko je vrednost polja veća ili jednaka od 0600 tada je u pitanju Ethernet II format. Napomenimo da se praksi češće sreće format Ethernet II.
- Korisni sadržaj (*Payload*) – Ovo polje sadrži enkapsulirani sadržaj iz protokola koji koristi eternet okvire za prenos svog sadržaja. Dužina ovog polja se kreće u granicama od 46-1500 bajtova ako se ne koristi VLAN oznaka, odnosno 42-1500 ako se koristi VLAN oznaka. Eternet svičevi se mogu konfigurisati i za rad sa tzv. *jumbo* okvirima, koji omogućavaju prenos korisnog sadržaja većeg od 1500 bajtova, koji predstavlja limit u slučaju normalnih eternet okvira.
- Polje za proveru (*FSC - Frame Check Sequence*) – 32-bitno polje za proveru ispravnosti. Koristi se CRC-32 provera čiji je generišući polinom $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$. Pri tome, štiti se celokupan eternet okvir bez preambule i SFD bajta (IFG nije deo okvira). Proračun se vrši na sledeći način. Eternet okvir (od odredišne MAC adrese do korisnog sadržaja, uključujući i odredišnu MAC adresu i korisni sadržaj) se predstavlja u vidu polinoma $M_{init}(x)$ koji se štiti. Komplementira se najviših 32 bita $M_{init}(x)$ i dobija se polinom $M(x)$. Vrši se deljenje $M(x)\cdot x^{32}/G(x)$, gde je $G(x)$ generišući polinom. Ostatak deljenja se komplementira i postavlja u polje za proveru.
- IFG (*InterFrame Gap*) – Ovo polje, ustvari, nije deo eternet okvira, već predstavlja obavezni razmak između susednih okvira. Minimalna dužina ovog razmaka je 12 bajtova.

8.2.3.2. PPP protokol

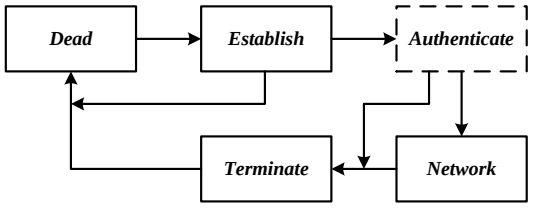
PPP protokol se koristi za ostvarivanje tačka-tačka veze na linku između susednih čvorova i definisan je u RFC 1661. PPP protokol omogućava da se pomoću njega ostvari

simultani prenos paketa za više mrežnih protokola (IP, IPX, ...) preko istog linka tj. da se paketi različitih mrežnih protokola simultano šalju preko PPP okvira. PPP podrazumeva upotrebu LCP (*Link Control Protocol*) i NCP (*Network Control Protocol*) protokola. LCP protokol se koristi za uspostavu, konfigurisanje i testiranje PPP veze na sloju linka podataka. NCP protokol, ustvari, predstavlja skup protokola koji se definišu za mrežne protokole i koji se koriste za uspostavu i konfigurisanje mrežnih protokola za rad sa PPP protokolom (odnosno za korišćenje usluga PPP protokola za prenos paketa dotičnih mrežnih protokola). Na primer, za IPv4 protokol se koristi IPCP (*IP Control Protocol*) protokol kao NCP protokol za IPv4 protokol. PPP protokol je čest u praksi i često ga koriste Internet provajderi, na primer, za povezivanje svojih korisnika preko xDSL pristupa ili *dial-up* pristupa. Takođe, PPP protokol se često koristi i za povezivanje mrežnih uređaja preko linka koji ih spaja. PPP omogućava i upotrebu autentifikacije, enkripcije i kompresije, što je u značajnoj meri doprinelo njegovoj popularnosti.



Slika 8.2.3.2.1. PPP enkapsulacija

PPP protokol enkapsulira koristan sadržaj na veoma jednostavan način prikazan na slici 8.2.3.2.1. Ispred korisnog sadržaja (polje korisni deo) se stavlja informacija o protokolu (polje protokol) koji je formirao koristan sadržaj. Na kraju se može dodati i dopuna (polje dopuna), ukoliko je ona neophodna. Polje protokol je dužine 1 ili 2 bajta. Na primer, heksadecimalna vrednost C021 odgovara LCP protokolu, a 8021 odgovara IPCP protokolu. Po difoltu je protokol uvek dužine 2 bajta, sem ako se prilikom konfigurisanja PPP veze (upotrebom LCP protokola) ne dozvoli da ovo polje može da bude dužine 1 bajt. Koristan deo (računajući i dopunu) ima maksimalnu dužinu koja se definiše kao MRU (*Maximum Receive Unit*) parametar PPP veze. Difolt vrednost MRU je 1500 bajtova, ali može da se promeni tokom konfigurisanja PPP veze koje vrši LCP protokol. Prikazani PPP enkapsuliran sadržaj se dalje stavlja u okvir koji se koristi na dotičnom linku. Na primer, kod serijskih linkova se po difoltu koriste okviri veoma slični HDLC (*High-Level Data Link Control*) okvirima za prenos PPP enkapsuliranog sadržaja sa slike 8.2.3.2.1, i ova varijanta je definisana u RFC 1662. Napomenimo da su LAPD okviri opisani u prethodnom poglavlju takođe bazirani na HDLC okvirima. Veoma popularan je i PPPoE (*PPP over Ethernet*), gde se PPP enkapsuliran sadržaj prenosi preko eternet okvira na način definisan u RFC 2516.

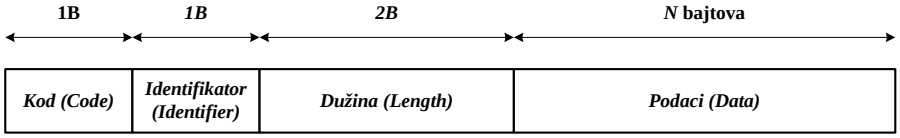


Slika 8.2.3.2.2. Faze PPP konekcije

PPP konekcija prolazi kroz više faza kao što je prikazano na slici 8.2.3.2.2. Mrtva faza (*Dead*) predstavlja fazu u kojoj PPP veza nije otpočela da se uspostavlja jer, na primer, nije detektovan nosilac ili signal (na primer, prekinut link). Kada se detektuje nosilac ili sam administrator zada komandu da se uspostavi PPP konekcija prelazi se u fazu uspostave

(*Establish*). U okviru faze uspostave se koristi LCP protokol za konfigurisanje parametara PPP veze. Ukoliko LCP konfiguracija bude neuspešna, vrši se povratak u mrtvu fazu. Opcije koje se konfigurišu LCP protokolom su:

- MRU (*Maximum Receive Unit*) - Parametar koji definiše maksimalnu dužinu u bajtovima korisnog dela i dopune zajedno. Difolt vrednost je 1500 bajtova.
- Protokol za autentifikaciju (*Authentication Protocol*) - Konfiguriše koji protokol za autentifikaciju će se koristiti. U RFC 1660 predviđeni su protokoli PAP (*Password Authentication Protocol*) i CHAP (*Challenge Handshake Authentication Protocol*). Ako se želi raditi autentifikacija, preporuka je da se koristi pouzdaniji CHAP protokol definisan u RFC 1994. PAP protokol se uglavnom sreće kod starijih implementacija, a u novijim se ne koristi zbog toga što pruža manju sigurnost u odnosu na druge protokole autentifikacije koji se mogu koristiti (poput CHAP). Po difoltu se autentifikacija ne koristi. U RFC 3748 je predviđen i EAP (*Extensible Authentication Protocol*) protokol koji se može koristiti za autentifikaciju. Preciznije, kada se uđe u fazu autentifikacije (*Authenticate*) sa slike 8.2.3.2.2, EAP protokol će omogućiti da se obe strane PPP veze dogovore oko mehanizna (algoritma) autentifikacije koji će se koristiti čime je postignuta velika fleksibilnost, kako u dodavanju novih mehanizama autentifikacije, tako i u nivou sigurnosti koji se želi postići.
- Protokol za nadgledanje kvaliteta (*Quality Protocol*) - Konfiguriše koji protokol za nadgledanje kvaliteta će se koristiti. Protokol za nadgledanje kvaliteta prati pre svega učestanost odbacivanja paketa, pa tako ako učestanost odbacivanja bude prevelika može se odlučiti da se PPP veza zatvori tj. raskine. U RFC 1660 predviđen je LQR (*Link Quality Report*) protokol. Po difoltu se ne koristi protokol za nadgledanja kvaliteta.
- Magični broj (*Magic Number*) - Ova opcija omogućava konfiguraciju metoda za otkrivanje anomalija na linku poput zatvorene petlje (*looped-back* linkovi). Po difoltu se magični brojevi ne koriste. Opis upotrebe magičnih brojeva je dat u RFC 1660.
- Kompresija polja protokol (*PFC - Protocol Field Compression*) - Po difoltu polje protokol u PPP enkapsuliranom formatu je dužine dva bajta. Upotrebom ove opcije se može konfigurisati da neki protokoli koriste vrednost dužine jedan bajt, tj. da za njih polje protokol bude dužine jedan bajt.
- Kompresija adrese i kontrolnog polja (*ACFC - Address and Control Field Compression*) - Ova opcija omogućava da izvrši kompresija adresnog i kontrolnog polja u okviru u koji se stavlja PPP enkapsulirani format. Očigledno, misli se pre svega na okvire koji su slični HDLC formatu okvira. Naravno, po difoltu se ne vrši kompresija.



Slika 8.2.3.2.3. Format LCP paketa

Format LCP paketa koji se stavlja u polje korisni deo PPP enkapsuliranog formata je prikazan na slici 8.2.3.2.3. Polje kod (*Code*) definiše koji LCP paket je u pitanju. Polje identifikator (*Identifier*) se koristi za uparivanje LCP odgovora sa odgovarajućim LCP zahtevima. Polje dužina (*Length*) definiše ukupnu dužinu LCP paketa u bajtovima, pri čemu se računa celokupan LCP paket sa slike 8.2.3.2.3 (sva polja LCP paketa). Koristan sadržaj LCP paketa se stavlja u polje podaci (*Data*). Tipovi LCP paketa definisani u RFC 1660 su dati u tabeli 8.2.3.2.1. U RFC 1660 je dato i njihovo tumačenje i formati. Može se iz naziva samih LCP paketa zaključiti namena dotičnih paketa.

Tabela 8.2.3.2.1. Tipovi LCP paketa

Vrednost koda	LCP paket
1	*Configure-Request*
2	*Configure-Ack*
3	*Configure-Nak*
4	*Configure-Reject*
5	*Terminate-Request*
6	*Terminate-Ack*
7	*Code-Reject*
8	*Protocol-Reject*
9	*Echo-Request*
10	*Echo-Reply*
11	*Discard-Request*

Nakon završetka konfiguracije veze, prelazi se u fazu autentifikacije (*Authenticate*) ako je u fazi uspostave konfigurisana upotreba nekog od protokola autentifikacije. Ukoliko je autentifikacija neuspešna, vrši se povratak u mrtvu fazu preko faze okončanja PPP veze (*Terminate*), u suprotnom se prelazi u fazu mrežne konfiguracije (*Network*). Ako se ne vrši autentifikacija, tada se iz faze uspostave direktno prelazi u fazu mrežne konfiguracije. U fazi mrežne konfiguracije se vrši konfiguracija mrežnih protokola čiji paketi će se prenositi preko PPP linka. Koriste se odgovarajući NCP protokoli (na primer, za IPv4 se koristi IPCP protokol), Kada se izvrši konfiguracija mrežnog protokola, mogu se prenositi paketi za taj mrežni protokol tako što se izvrši njihova PPP enkapsulacija. Mrežni protokol se može konfigurisati u bilo kom momentu ove faze i čim se izvrši dotična konfiguracija moguć je PPP prenos paketa konfigurisanog mrežnog protokola. Takođe, u bilo kom momentu je moguće i okončati vezu za neki mrežni protokol primenom odgovarajućeg NCP protokola. Ako neka strana želi da zatvori PPP konekciju preći će se u fazu okončavanja PPP veze (*Terminate*). Takođe, ako se detektuje neregularna situacija poput prekida linka ili prevelikog broja grešaka na linku doći će takođe do zatvaranja PPP veze, tj. preći će se u fazu okončavanja PPP veze. Tada se šalje odgovarajući LCP paket (*LCP terminate-request*) za zatvaranje PPP veze, na koji suprotna strana treba da odgovori sa *LCP terminate-ack* paketom (strana koja je primila *LCP terminate-request* paket će sačekati dovoljno dugo da bi bila sigurna da je strana koja je inicirala raskid prešla u mrtvu fazu pa će i sama preći u mrtvu fazu). Kada se primi *LCP terminate-ack* ili kad istekne tajmer, prelazi se u mrtvu fazu. Tranzicije između faza PPP veze, kao i same faze su detaljnije opisane u RFC 1660.

Već smo napomenuli da se IPCP protokol koristi kao NCP protokol za konfigurisanje rada IPv4 protokola preko PPP protokola. IPCP protokol je definisan u RFC 1332 preporuci. IPCP se koristi za konfigurisanje, aktiviranje i deaktiviranje rada IPv4 protokola preko PPP

protokola. IPCP koristi identičan format paketa (slika 8.2.3.2.3) i princip razmene paketa kao LCP protokol. Kodovi koje koristi IPCP protokol su kodovi 1-7 sa istim značenjem kao kod LCP paketa iz tabele 8.2.3.2.1. Naravno, opcije koje se mogu konfigurisati se razlikuju u odnosu na LCP protokol i odnose se na konfigurisanje vezano za IPv4 protokol čiji paketi će se prenositi preko PPP protokola. Opcije koje se mogu konfigurisati po preporuci RFC 1332 su protokol za kompresiju IPv4 paketa i IPv4 adresa. Po difoltu se ne koristi kompresija, ali ako se koristi, IPCP paket će sadržati kod protokola za kompresiju koji će se koristiti, poput *Van Jacobson Compressed TCP/IP* protokola koji vrši kompresiju TCP/IP zaglavlja i koji je naveden u RFC 1332 preporuci, a detaljnije opisan u RFC 1144. Po difoltu IP adrese nisu definisane na krajnjim tačkama PPP linka, ali se mogu konfigurisati njihove vrednosti. Kada se završi proces konfiguracije mogu se početi razmenjivati IPv4 paketi preko PPP linka tj. protokola na način prikazan na slici 8.2.3.2.1, gde se IPv4 paket smešta u polje korisni deo. Polje protokol se podešava na heksadecimalnu vrednost 0021 čime se signalizira da je enkapsuliran IPv4 paket. U slučaju da se, na primer, koristi *Van Jacobson Compressed TCP/IP* protokol za kompresiju zaglavlja, tada postoje tri moguće vrednosti polja protokol kada se prenosi IPv4 paket, da bi se mogle razlikovati sve situacije koje mogu da nastanu jer IPv4 paket ne mora da enkapsulira TCP protokol, već može da enkapsulira i druge protokole poput UDP protokola. Heksadecimalna vrednost 0021 signalizira IPv4 paket koji enkapsulira protokol koji nije TCP, 002D signalizira IPv4 paket sa komprimovanim TCP/IP zaglavljem, a 002F signalizira TCP/IP paket kod koga nije izvršena kompresija zaglavlja.

Za IPv6 protokol se koristi kao NCP protokol IPv6CP protokol opisan u RFC 5072. IPv6CP koristi identičan format paketa (slika 8.2.3.2.3) i princip razmene paketa kao LCP protokol. Kodovi koje koristi IPv6CP protokol su kodovi 1-7 sa istim značenjem kao kod LCP paketa iz tabele 8.2.3.2.1. Naravno, opcije koje se mogu konfigurisati se razlikuju u odnosu na LCP protokol. U RFC 5072 je predviđeno da se može konfigurisati 64-bitna identifikacija interfejsa na svakoj od strana PPP linka. Detaljniji opis konfigurisanja identifikacije interfejsa i namena identifikacije interfejsa su dati u RFC 5072 preporuci. Kada se završi proces konfiguracije mogu se početi razmenjivati IPv6 paketi preko PPP linka tj. protokola na način prikazan na slici 8.2.3.2.1, gde se IPv6 paket smešta u polje korisni deo. Polje protokol se podešava na heksadecimalnu vrednost 0057 čime se signalizira da je enkapsuliran IPv6 paket. Napomenimo da ako je heksadecimalna vrednost polja protokol 8057, da je tada enakpsuliran IPv6CP paket.

8.2.4. Kompresija zaglavlja

Kao što je već naglašeno ranije, generisanje paketa koji sadrže signal govora u telefonskom razgovoru, ne sme da se generiše sa isuviše malom frekvencijom jer bi se tako unosilo preveliko kašnjenje i smanjio kvalitet prenosa govornog signala, a sa druge strane paketi ne smeju da se generišu ni sa isuviše visokom frekvencijom jer bi tada korisna informacija (govor) bila isuviše mala i zauzimala mali procenat paketa. Tipično se paketi generišu na svakih 10-30ms, pri čemu je kod parametarskih kodera ova veličina tipično jednaka dužini odsečka koji se obrađuje i koja uglavnom upada u pomenuti interval.

Pretpostavimo da se G.711 kodiran signal prenosi paketima koristeći RTP, UDP i IP protokol, pri čemu sva zaglavlja imaju minimalnu dužinu (koriste samo obavezni deo). Tabela 8.2.4.1 prikazuje broj bajtova zaglavlja i broj bajtova korisničke informacije tj. signala govora za slučaj kada se paketizacija vrši svakih 10/20/30 ms. Ista tabela prikazuje i slučaj kada se prenos vrši preko eterneta tj. poznat je format okvira drugog sloja. Tipično se za prenos VoIP

telefonskog razgovora koriste navedeni transportni i mrežni protokoli, dok drugi sloj zavisi od konkretne mrežne tehnologije, i ovde se kao primer uzima eternet tehnologija. G.711 koder generiše 8 bita na svakih 125μs, a pošto se ti generisani biti direktno stavljaju u RTP korisni deo (korisni deo samo sadrži te osmobitne odmerke), lako je izračunati broj bajtova koji se stavlja u korisni deo paketa. RTP obavezni deo zagavlja je dužine 12 bajtova, UDP zaglavlje je dužine 8 bajtova, IPv4 obavezni deo zaglavlja je dužine 20 bajtova, a IPv6 obavezni deo zaglavlja je dužine 40 bajtova. Eternet zaglavlje (i FCS polje računamo kao deo zaglavlja iako se nalazi na kraju okvira) je dužine 26 bajtova (ako se ne koristi VLAN oznaka), odnosno ako uračunamo i minimalni IFG razmak onda je ukupna dužina 38 bajtova. Procenti navedeni uz korisni deo označavaju udeo korisnog dela paketa u procentima.

Tabela 8.2.4.1. Odnos dužine zaglavlja i korisnog dela paketa za G.711 koder

Protokoli	10ms		20ms		30ms	
	Zaglavlje	Korisni deo	Zaglavlje	Korisni deo	Zaglavlje	Korisni deo
RTP, UDP, IPv4	40B	80B (67%)	40B	160B (80%)	40B	240B (86%)
RTP, UDP, IPv6	60B	80B (57%)	60B	160B (73%)	60B	240B (80%)
RTP, UDP, IPv4, Eternet bez IFG	66B	80B (55%)	66B	160B (71%)	66B	240B (78%)
RTP, UDP, IPv6, Eternet bez IFG	86B	80B (48%)	86B	160B (65%)	86B	240B (74%)
RTP, UDP, IPv4, Eternet sa IFG	78B	80B (51%)	78B	160B (67%)	78B	240B (75%)
RTP, UDP, IPv6, Eternet sa IFG	98B	80B (45%)	98B	160B (62%)	98B	240B (71%)

Kao što se može videti, udeo korisne informacije u paketu je relativno mali pogotovo kada se uzme u obzir i zaglavlje eternet okvira, pa je procenat iskorišćenja mrežnih resursa slab u slučaju prenosa takvih paketa jer se veliki procenat resursa troši na prenos zaglavlja različitih protokola. Naravno, može se primetiti i da je uvek bolje prenositi informacije koje odgovaraju dužem odsečku jer je tada udeo korisne informacije u paketu veći. Ovo slabo iskorišćenje paketa je ozbiljan problem posebno u WAN mrežama gde je veoma važno što bolje iskorišćenje kapaciteta linkova tj. cilj je što više korisnih informacija preneti preko linkova jer se time omogućava opsluživanje većeg broja korisnika, a time i veći profit za operatera. Otuda je uvedena tehnika kompresije zaglavlja.

Kompresija zaglavlja koristi činjenicu da se veliki broj polja zaglavlja ne menja tokom veze ili se menja po unapred poznatom šablonu. Na primer, tokom veze se neće menjati IP adrese u IP zaglavlju, UDP portovi u UDP zaglavlju, i SSRC identifikator u RTP zaglavlju. Takođe, polje verzije se ni u jednom zaglavlju neće menjati ako postoji, itd. Cilj je da se kompresijom izgube ova statička polja, i da ostanu samo ona polja koja se slučajno menjaju (na primer, polje za proveru u UDP zaglavlju jer ono zavisi i od korisnog dela UDP paketa u kome se nalaze govorni odmerci), a i da se ona polja koja se menjaju po unapred poznatom šablonu zamene sa skraćenim verzijama tih polja koja će omogućiti njihovu rekonstrukciju. U procesu kompresije zaglavlja je bitno napomenuti da je kompresija bez gubitaka (u procesu dekompresije se mora bez greške rekonstruisati originalno zaglavlje) i da se kompresija zaglavlja vrši na link-po-link nivou jer su, na primer, IP adrese neophodne za usmeravanje paketa kroz mrežu. Takođe, kompresija zaglavlja vrši kompresiju zaglavlja protokola transportnog i mrežnog sloja, jer je drugi sloj odgovoran za prenos preko linka i stoga se ne može vršiti njegova kompresija u većini slučajeva (kod PPP protokola smo videli da se može konfigurisati kompresija adresnog i

kontrolnog polja). Koja zaglavlja će se komprimovati zavisi od konfiguracije kompresije i tipa upotrebljene kompresije. Na primer, može se komprimovati samo RTP zaglavlje, samo IP zaglavlje, samo UDP+IP zaglavlje, celokupno RTP+UDP+IP zaglavlje, itd. Pre izvršavanja same kompresije, krajnje tačke linka se moraju dogovoriti o tipu kompresije, a takođe moraju razmeniti i bar jedan nekomprimovani paket da bi mogli da dobiju statičke informacije iz zaglavlja, neophodne u procesu dekompresije naknadnih komprimovanih zaglavlja. Takođe, neophodno je razlikovati pakete sa komprimovanim zaglavljem po tokovima da bi se mogla ispravno izvršiti dekompresija zaglavlja. Tipično se definiše identifikator toka koji se stavlja u komprimovano zaglavlje tako da se na prijemu jednostavno može utvrditi kom toku pripada dotični paket i koje su njegove statičke informacije zaglavlja. Kompresijom zaglavlja se zaglavlja mrežnog i transportnog sloja mogu zajednički predstaviti komprimovanim zaglavljem dužine svega nekoliko bajtova čime bi se udeo korisnog dela paketa značajno povećao.

Već smo napomenuli jedan algoritam za kompresiju zaglavlja definisan u RFC 1144 (*Van Jacobson Compressed TCP/IP* protokol za kompresiju zaglavlja), ali postoje i drugi protokoli za kompresiju zaglavlja. *Van Jacobson Compressed TCP/IP* protokol za kompresiju zaglavlja se označava i kao CTCP (*Compressed TCP*), jer vrši kompresiju TCP+IPv4 zaglavlja dužine 40 bajtova na 4 bajta (napomena obavezni deo TCP zaglavlja je 20 bajtova, a obavezni deo IPv4 zaglavlja je takođe 20 bajtova). U RFC 2507 je definisan IPHC (*IP Header Compression*) metod kompresije koji može da komprimuje UDP+IP i TCP+IP zaglavlja, pri čemu su podržana i IPv4 i IPv6 zaglavlja. U RFC 2508 je definisan CRTP (*Compressed RTP*) metod kompresije koji se koristi za kompresiju RTP+UDP+IPv4 zaglavlja na minimalno 2 ili 4 bajta (minimalno 2 bajta ako se UDP polje za proveru ne koristi, u suprotnom minimalno 4 bajta). RFC 3095 daje generički opis robusne kompresije zaglavlja (*ROHC - Robust Header Compression*) koja je veoma pogodna za primenu u sredinama sa gubicima paketa usled grešaka, pri čemu su u istoj RFC preporuci definisani i profili upotrebe ROHC. Profili definisani u RFC 3095 su RTP+UDP+IP profil, UDP+IP profil i ESP+IP profil, gde se ESP (*Encapsulating Security Payload*) koristi za tajnost prenosa korisničkog saobraćaja. U RFC 3241 je opisana upotreba ROHC preko PPP protokola, a u RFC 3843 je dat opis IP profila za ROHC (samo se IP zaglavlje komprimuje). ROHC može da komprimuje RTP+UDP+IP zaglavlje na minimalno 1-3 bajta, što bi drastično povećalo udeo korisnog dela u takvom paketu.

8.3. Ocena kvaliteta govornog signala u paketskim mrežama

Za razliku od komutacije kola gde su u mreži jednom telefonskom razgovoru dodeljeni resursi koje koristi samo dotični razgovor, u paketskim mrežama resurse dele svi korisnici, odnosno razgovori međusobno dele iste resurse zajedno sa drugim vrstama konekcija. Očigledno, kvalitet servisa je lakše garantovati u mrežama sa komutacijom kola, dok u paketskim mrežama se mogu pružiti statističke, ali i striktne garancije u zavisnosti od primenjene strategije. Statističke garancije obezbeđuje DiffServ (diferencijalni servisi) pristup, dok striktne garancije obezbeđuje IntServ (integrisani servisi) pristup. DiffServ pristup je jednostavniji, ali manje pouzdan što se tiče striktnosti garancija, dok je IntServ pristup bolji sa stanovišta garancija, ali je i komplikovaniji za implementaciju sa dodatnim problemom podrške velikom broju istovremenih konekcija koje zahtevaju garancije kvaliteta servisa. Istraživanja ove problematike su veoma aktuelna i često se predlažu kombinovana rešenja koja kombinuju DiffServ i IntServ pristupe da bi se dobilo najbolje od oba pristupa.

Pošto na kvalitet govornog signala, tj. razgovora utiče mnogo parametara (kašnjenje, varijacija kašnjenja, tip kodera, itd.) bilo bi poželjno razviti analitički model koji bi uključio sve relevantne parametre i omogućio dobru procenu kvaliteta govorne veze. Razlikujemo objektivno ocenjivanje i subjektivno ocenjivanje kvaliteta govorne veze. U objektivnom ocenjivanju se vrši merenje fizičkih veličina (odnos signal/šum, kašnjenje, varijacija kašnjenja, snaga eha i dr.) i na osnovu izmerenih vrednosti se donosi objektivna ocena govorne veze. U subjektivnom ocenjivanju se vrši testiranje govorne veze na velikom uzorku korisnika koji daju svoje subjektivne ocene. Na taj način se može proceniti i subjektivan uticaj pojedinih parametara na kvalitet veze, tj. uticaj parametara na subjektivnu ocenu kvaliteta govorne veze. Subjektivne ocene su važnije jer one odražavaju subjektivni doživljaj korisnika koji je relevantan jer će isti ti korisnici koristiti usluge mreže za ostvarivanje razgovora i samim tim je cilj da korisnici budu zadovoljni, tj. da njihov subjektivni doživljaj kvaliteta razgovora bude dobar. Međutim, za planiranje i projektovanje mreža koje će se koristiti i za servis telefonskog razgovora je bitno da se kreira dobar model za procenu kvaliteta govornih veza i da se taj model poveže sa subjektivnim modelom ocenjivanja čime bi se omogućilo kvalitetno planiranje i projektovanje telekomunikacionih mreža koje bi se koristile i za telefonske razgovore. Model je očigledno potrebno zasnovati na uticaju pojedinih parametara govorne veze na subjektivni doživljaj korisnika, pri čemu fizička merenja vrednosti parametara u okviru testiranja daju njihove prave (objektivne) vrednosti koje će se koristiti u analitičkom modelu. Otuda je očigledno da dobar model mora da uzme u obzir i objektivne i subjektivne ocene da bi napravio dobru predikciju kvaliteta govorne veze za određenu kombinaciju vrednosti relevantnih parametara govorne veze. Napomenimo da se pod najkvalitetnijom vezom smatra lokalna ISDN veza. Naime, ova veza koristi G.711 koder koji unosi najmanju degradaciju kvaliteta usled kompresije (samo šum kvantizacije), kašnjenje i varijacija kašnjenja govornog signala su zanemarljivi jer se prolazi kroz samo jednu centralu i pri tome se digitalizacija govornog signala vrši na strani korisnika (za razliku od analognog telefonskog aparata gde se digitalizacija vrši u centrali, pa se digitalizacija radila na već, ipak, malo degradiranom govornom signalu usled šumova i drugih smetnji na pretplatničkoj liniji).

Najpoznatiji model subjektivnog ocenjivanja jeste MOS (*Mean Opinion Score*) model koji predstavlja srednju ocenu korisnika za govornu vezu u određenim uslovima (svi korisnici su ocenjivali razgovor pod istim uslovima tj. parametri su bili isti za vreme testiranja). MOS ocena se kreće u granicama od 1 do 5, pri čemu ocena 5 odražava najviši kvalitet, a ocena 1 najniži, tj. neprihvatljiv kvalitet. Tabela 8.3.1 daje pregled relacije MOS ocene i subjektivne percepcije korisnika.

Tabela 8.3.1. Relacija između MOS ocene i subjektivne percepcije korisnika

MOS ocena	Subjektivna percepcija korisnika
1	Gotovo svi (ili svi korisnici) su nezadovoljni
2	Velik deo korisnika je nezadovoljan
3	Neki korisnici su nezadovoljni
4	Korisnici su zadovoljni
5	Korisnici su veoma zadovoljni

ITU-T preporuka G.107 definiše E model koji se može koristiti za predikciju kvaliteta govornog signala ako su poznate vrednosti parametara govorne veze i samim tim se E model može koristiti u planiranju i projektovanju telekomunikacionih mreža namenjenih, između ostalog, i prenosu telefonskih razgovora. E model vrši proračun ocene *R* kvaliteta govorne veze,

pri čemu E model uzima u obzir sve relevantne parametre koji utiču na kvalitet govorne veze. Ocena *R* kvaliteta veze se izražava na sledeći način:

$$R = R_0 - I_s - I_d - I_e + A \tag{8.3.1}$$

R_0 predstavlja osnovni odnos signal/šum, pri čemu se u izvore šuma ubrajaju šum linije kod analognog prenosa i šum okoline. I_s predstavlja negativan uticaj činilaca koji se javljaju uporedo sa korisnim signalom govora poput šuma kvantizacije kod A/D konverzije, neoptimalnosti glasnosti signala govora, negativnog uticaja lokalnog eha (*sideton*) i sl. Pod neoptimalnošću glasnosti signala govora se podrazumeva odstupanje od idealne glasnosti govora (pretih ili preglasan govorni signal). Lokalni eho predstavlja signal iz mikrofona govornika koji se (oslabljen) namerno vraća u slušalicu govornika da bi se postigao pozitivan efekat da govornik u slušalici čuje sebe dok priča, međutim, ako se ovaj lokalni eho vrati sa prevelikom glasnošću onda dolazi do njegovog negativnog efekta na subjektivnu percepciju korisnika u pogledu kvaliteta govorne veze. I_d predstavlja negativan uticaj kašnjenja i eha. I_e predstavlja negativan uticaj kompresora govornog signala (kodera), kao i gubitaka paketa u mreži. *A* predstavlja subjektivna očekivanja korisnika i jedini je faktor koji može pozitivno uticati na ocenu veze. Naime, ako korisnik u startu ima niska očekivanja u pogledu kvaliteta, pozitivnije će oceniti čak i vezu slabijeg kvaliteta koju bi generalno niže ocenio u uslovima normalnih očekivanja. Na primer, korisnik ima velika očekivanja u pogledu kvaliteta za klasičnu telefonsku vezu u fiksnoj telefoniji, ali ima niska očekivanja za mobilnu telefonsku vezu u pokretnom vozilu. Važno je uočiti da je E model aditivan model, odnosno svi uticaji su aditivnog karaktera na konačnu ocenu što ide u prilog jednostavnosti E modela. Takođe, na sajtu ITU-T organizacije se može naći onlajn kalkulator E modela, gde se unosom parametara govorne veze može proračunati vrednost ocene *R*. Takođe, zajedno sa preporukom G.107 je moguće skinuti programski kod ovog kalkulatora sa istog sajta. Ocena *R* se kreće u granicama 0-100, pri čemu nije preporučljivo projektovati govorne veze čija je *R* ocena niža od 50. Pomenuta lokalna ISDN veza za koju je rečeno da ima najbolji kvalitet ima ocenu $R = 94$, u slučaju difolt vrednosti parametara govorne veze navedenih u G.107.

Parametar R_0 se proračunava iz:

$$R_0 = 15 - 1.5\left(SLR + N_o\right) \tag{8.3.2}$$

gde je SLR (*Send Loudness Rating*) gubitak glasnosti govornog signala na predajnoj strani, a N_o predstavlja zbirni uticaj šumova na linijama u slučaju analognog prenosa, šuma okoline na predaji, šuma okoline i svih drugih izvora šumova na prijemu. Precizan proračun parametra N_o je dat u ITU-T G.107 preporuci. Napomenimo da su pored SLR parametra u ovom proračunu bitni i parametri RLR (*Receive Loudness Rating*) i OLR (*Overall Loudness Rating*). SLR predstavlja gubitak glasnosti govornog signala od korisnika (njegovih usta) do električnog interfejsa, pri čemu se pod električnim interfejsom tipično podrazumeva momenat gde je govor digitalizovan. Na primer, u klasičnoj telefoniji bi to bio SLAC u kome je izvršena digitalizacija govornog signala po A zakonu kompresije. RLR predstavlja gubitak glasnosti govornog signala na prijemnoj strani i to od električnog interfejsa (gde je govorni signal dekodovan) do korisnika (tj. njegovog uva). OLR predstavlja gubitak glasnosti od govornika do slušaoca na celokupnoj putanji (od usta govornika do uva slušaoca) i jednak je zbiru SLR i RLR.

Parametar I_s se proračunava iz:

$$I_s = I_{olr} + I_{st} + I_q \tag{8.3.3}$$

gde I_{olr} predstavlja uticaj neoptimalnosti glasnosti signala govora, I_{st} predstavlja uticaj neoptimalnosti lokalnog eha (*sideton*), I_q predstavlja uticaj šuma kvantizacije A/D konverzije, pri čemu ako je ovaj uticaj već uračunat kroz sam pad kvaliteta koji unosi koder onda se on izostavlja što je i tipičan slučaj. Napomenimo da ako se pri prenosu govornog signala kroz mrežu vrše dodatni parovi A/D i D/A konverzije onda se oni moraju uzeti u proračun. U G.108 preporuci je dat dijagram uticaja broja ovih parova na pad ocene *R*, i može se uzeti na osnovu dotičnog dijagrama da svaki dodatni par A/D i D/A konverzije snižava ocenu *R* za 2. Precizan proračun navedena tri parametra (I_{olr}, I_{st}, I_q) je dat u ITU-T G.107 preporuci.

Parametar I_d se proračunava iz:

$$I_d = I_{dte} + I_{dle} + I_{dd} \tag{8.3.4}$$

gde I_{dte} predstavlja uticaj eha govornika, I_{dle} predstavlja uticaj eha slušaoca, I_{dd} predstavlja uticaj ukupnog kašnjenja u jednom smeru. Precizan proračun navedena tri parametra je dat u ITU-T G.107 preporuci. Napomenimo da uticaj eha zavisi naravno od same snage eha, kao i od samog kašnjenja eha. Snaga eha govornika se posmatra u vidu parametra TELR (*Talker Echo Loudness Rating*) koji predstavlja gubitak glasnosti signala eha govornika (od usta govornika do uva govornika) na putanji samog eha. Ako je kašnjenje eha veće negativan uticaj eha će biti veći, a isto tako ako je snaga eha veća negativan uticaj eha će biti veći. Na primer, kada eho govornika dolazi sa kašnjenjem većim od 50ms korisnik (govornik) to počinje da oseća. Isto tako ako je kašnjenje u jednom smeru veće od 200ms, to takođe počinje da smeta korisnicima jer imaju osećaj da ih suprotna strana nije čula pa se zato ne odaziva u razgovoru, što dovodi do toga da sagovornici upadaju jedan drugome u reč. Napomenimo da postoje tehnike (poništavanje eha) kojima se mogu umanjiti negativni uticaji eha. Kašnjenje u jednom smeru u slučaju paketskih mreža zavisi od mnogo činilaca poput obrade govora na predajnoj strani i vremena paketizacije, vremena čekanja za slanje paketa ako se i drugi paketi šalju iz izvorišnog čvora, vremena propagacije kroz linkove, vremena obrade u mrežnim čvorovima, vremena čekanja u izlaznim baferima mrežnih čvorova (pošto se deli link sa paketima drugih tokova), vremena za otklanjanje džitera, vremena nadoknade izgubljenih paketa i dr. U proračun se uzima tipično maksimalno kašnjenje, kao najgori slučaj. Napomenimo da koderi takođe unose kašnjenje. Naime, većina kodera uzima odsečak govornog signala na obradu, pa je otuda kašnjenje koje koder unosi $2T_o+T_{la}$, gde je T_o vreme trajanja jednog odsečka koje je ujedno jednako i maksimalnom vremenu obrade jednog odsečka, a T_{la} je vreme *look-ahead*. Naime, prvo je potrebno vreme da se formira jedan odsečak (vreme T_o), pa je potom potrebno obraditi taj odsečak (vreme T_o), pri čemu je za neke kodere neophodno da imaju uvid i u deo narednog govornog odsečka (T_{la}). U proračun se nekada uzima i vreme paketizacije (pre svega u slučaju sporijih linkova) za koje se uzima da traje T_o. Napomenimo da se ukupno kašnjenje koje unosi koder naziva i akumulirano kašnjenje. Algoritamsko kašnjenje predstavlja vreme obrade odsečka uvećano za vreme dela narednog odsečka koje je neophodno za proces obrade (T_o+T_{la}). Tabela 8.3.2 daje pregled akumuliranog kašnjenja za više kodera. Napomenimo da u slučaju da se u paket stavlja više govornih odsečaka akumulirano kašnjenje uvećava za trajanje dodatnih odsečaka (jedan odsečak traje T_o).

Takođe, kašnjenje usled čekanja na svoj red na slanje u mrežnim čvorovima unosi značajnu varijaciju u ukupno vreme kašnjenja, tj. dolazi do izraženije pojave džitera. Da bi se kompenzovao džiter, koriste se tzv. *playout* baferi koji unose dodatno kašnjenje na prijemu da bi

kompenzovali varijacije kašnjenja (džiter) i tako izbegli da se usled preteranog kašnjenja kroz mrežu odbaci paket na prijemu jer je prošlo vreme za reprodukciju govornog sadržaja koji je on nosio.

Tabela 8.3.2. Akumulirano kašnjenje kodera

Koder	Protok [kb/s]	T_o [ms]	T_{la} [ms]	Akumulirano kašnjenje $2T_o+T_{la}$ - $3T_o+T_{la}$ [ms]
G.711	64	0.125	0	0.25 - 0.375
G.726	40	0.125	0	0.25 - 0.375
G.726	32	0.125	0	0.25 - 0.375
G.726	24	0.125	0	0.25 - 0.375
G.726	16	0.125	0	0.25 - 0.375
G.728	16	0.625	0	1.25 - 1.875
G.728	12.8	0.625	0	1.25 - 1.875
G.729	8	10	5	25 - 35
G.723.1	6.3	30	7.5	67.5 - 97.5
G.723.1	5.3	30	7.5	67.5 - 97.5
GSM 06.10	13	20	0	40 - 60
GSM 06.20	5.6	20	0	40 - 60
GSM 06.30	12.2	20	0	40 - 60

Tabela 8.3.3. Degradacija kvaliteta kodera

Koder	Protok [kb/s]	I_e
G.711	64	0
G.726	40	2
G.726	32	7
G.726	24	25
G.726	16	50
G.728	16	7
G.728	12.8	20
G.729	8	10
G.723.1	6.3	15
G.723.1	5.3	19
GSM 06.10	13	20
GSM 06.20	5.6	23
GSM 06.30	12.2	5

Parametar I_e se odnosi na degradaciju kvaliteta koji unosi sam koder u uslovima kada nema gubitaka paketa. Tabela 8.3.3 daje pregled degradacije kvaliteta I_e za više kodera. Međutim, u slučaju postojanja gubitaka paketa dolazi do dodatne degradacije kvaliteta. U preporuci G.107 se za slučaj postojanja gubitaka paketa definiše vrednost $I_{e\text{-}eff}$ koja se koristi potom u izračunavanju ocene R u (8.3.1) umesto I_e. $I_{e\text{-}eff}$ se proračunava na sledeći način:

$$I_{e-eff} = I_e + (95 - I_e)\frac{P_{pl}}{B_{pl} + P_{pl}/Burst_R} \qquad (8.3.5)$$

gde je P_{pl} verovatnoća gubitaka paketa, B_{pl} faktor otpornosti kodera na gubitke paketa, a $Burst_R$ predstavlja koeficijent sporadičnosti (vrednost 1 odgovara slučaju da su gubici potpuno slučajni, a vrednosti veće od 1 označavaju da se gubici dešavaju u grupama, pri čemu što je veći

koeficijent gubici se dešavaju u većim grupama tj. dužim burstovima). U G.113 preporuci su date vrednosti za B_{pl} za pojedine kodere.

Važno je naglasiti da ne reaguju svi koderi isto na gubitke paketa, tj. neki su otporniji na gubitke od drugih, što se izražava preko parametra B_{pl}. Na primer, G.711 koder ne koristi nikakvu korelaciju između susednih odmeraka pa je stoga veoma slabo otporan na gubitke, dok su parametarski koderi zbog toga što dominantno koriste visoku međusobnu korelisanost susednih odmeraka i odsečaka otporniji na gubitke jer im je lakše sintetisati izgubljeni govorni odsečak (tačnije vrednosti njegovih parametara) na osnovu govornih odsečaka koje su primili. Veoma je bitna i razlika u tipu gubitka paketa. Manju degradaciju unose slučajni gubici paketa, nego gubici paketa u burstovima (grupama). Za izbegavanje prevelike degradacije kvaliteta usled gubitaka paketa, koriste se postupci maskiranja gubitaka paketa (*PLC - Packet Loss Concealment*). Na primer, može se ponavljati poslednji odsečak govornog signala u slučaju gubitka paketa, ali očigledno ovo nije adekvatno rešenje u slučaju kada se paketi gube u burstovima. Takođe, mogu se dodati dodatni zaštitni biti, tj. izvršiti korekcija grešaka unapred (*FEC - Forward Error Correction*). Može se izvršiti i preventivna retransmisija bitnih paketa koji nose bitnije informacije na osnovu kojih se može izvršiti bolja procena budućih paketa tj. govornog sadržaja u njima. Sami parametarski koderi mogu obradom odsečaka koji su primljeni da procene vrednost izgubljenih odsečaka tj. paketa, ali naravno po cenu dodatne obrade i samim tim i kašnjenja na prijemu (najbolja procena se dobija kada se u obradu uzmu i odsečci koji su prethodili izgubljenim odsečcima, kao i oni koji su sledili za njima).

Parametar *A* jedini može pozitivno uticati na konačnu ocenu *R*. Naime, uočeno je da ispitanici (korisnici) daju bolje ocene kada su im očekivanja u pogledu kvaliteta govorne veze niža. Na primer, za govornu vezu koja se ostvaruje iz pokretnog vozila, korisnici u startu imaju niska očekivanja u pogledu kvaliteta, tako da će tada prosečan kvalitet veze oceniti kao dobar kvalitet veze jer su subjektivno doživeli vezu bolje od svojih očekivanja. Tabela 8.3.4 daje pregled faktora koji utiču na parametar *A*.

Tabela 8.3.4. Proračun vrednosti parametra *A*

A	**Situacija**
20	Veze do teško dostupnih područja, na primer, veze koje se ostvaruju preko satelitskih linkova
10	Mobilna veza iz vozila u pokretu
5	Mobilna veza u zgradi
0	Ostale situacije

U ITU-T preporukama G.107 i G.108 su date difolt vrednosti za parametre koji utiču na kvalitet govorne veze (ali i dozvoljeni opsezi vrednosti parametara), što olakšava proračun (predikciju) kvaliteta veze tako što menjamo samo pojedine parametre koji u planiranoj mreži odstupaju od difolt vrednosti.

Tabela 8.3.5. Veza između E modela i MOS ocene

E model (R ocena)	**MOS**	**Subjektivna percepcija korisnika**
90-100	4.34-4.5	Korisnici su veoma zadovoljni
80-90	4.03-4.34	Korisnici su zadovoljni
70-80	3.60-4.03	Neki korisnici su nezadovoljni
60-70	3.10-3.60	Velik deo korisnika je nezadovoljan
50-60	2.58-3.10	Gotovo svi korisnici su nezadovoljni

Veza između E modela i MOS ocene definisana u G.107 preporuci je data izrazom:

$$MOS = 1 + 0.035R + R(R-60)(100-R)\cdot 7\cdot 10^{-6},\ 0 < R < 100 \qquad (8.3.6)$$

Pri tome je u samoj preporuci G.107 definisan i odnos za vrednosti ocene *R* manje od 0 (MOS ocena je tada 1) i veće od 100 (MOS ocena je tada 4.5). Tabela 8.3.5 daje odnose između pojedinih opsega MOS ocene i E modela. Napomenimo još jednom da nije preporučljivo projektovati sisteme kod kojih će ocena *R* biti ispod 50.

Za slučaj kada je *R* u opsegu [6.5, 100], tada je obrnuta relacija:

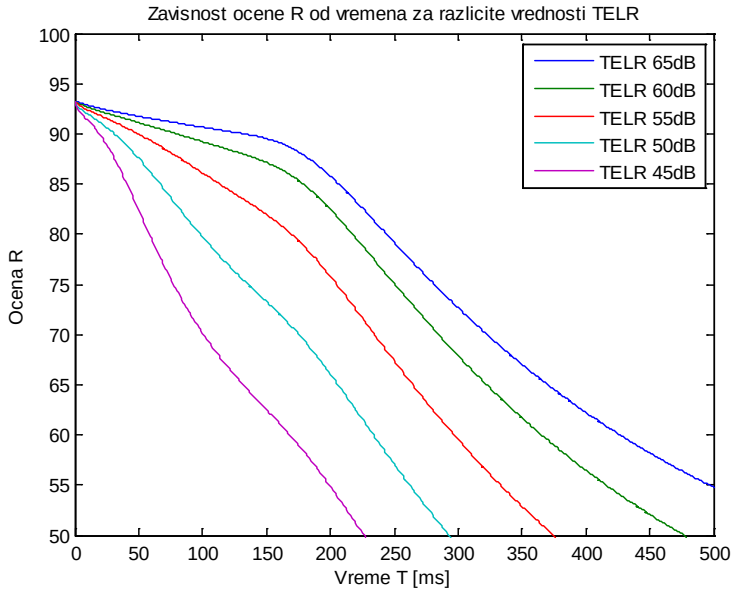
$$R = \frac{20}{3}\left(8 - \sqrt{226}\cos\left(h + \frac{\pi}{3}\right)\right)$$

$$h = \frac{1}{3}\arctan 2\left(18566 - 6750MOS, 15\sqrt{-903522 + 1113960MOS - 202500MOS^2}\right) \qquad (8.3.7)$$

$$\arctan 2(x,y) = \begin{cases} \arctan(y/x), & x \geq 0 \\ \pi - \arctan(y/(-x)), & x < 0 \end{cases}$$

8.3.1. Primeri

Na slici 8.3.1.1 je prikazan uticaj vremena kašnjenja u jednom smeru *T* za G.711 koder i različite vrednosti parametra *TELR*. Pošto se koristi G.711 koder, praktično mrežno kašnjenje ima najveći udeo. Svi ostali parametri su postavljeni na difolt vrednosti kao u preporuci G.107 (difolt vrednost za *TELR* je 65dB, a za parametar *T* je 0ms).

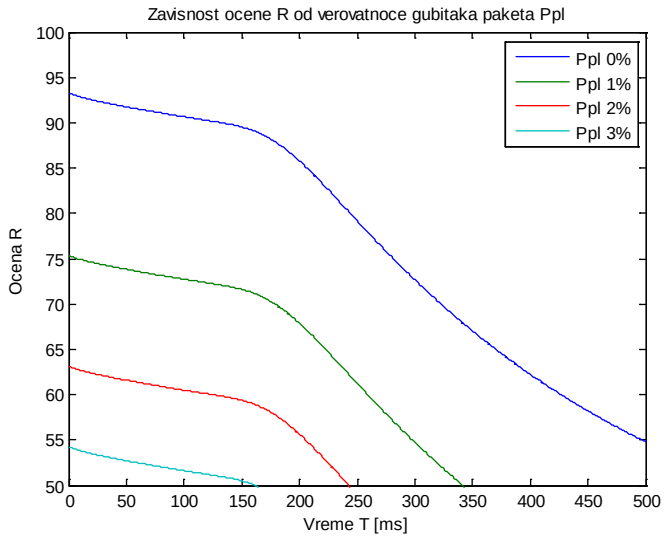


Slika 8.3.1.1. Uticaj parametra *TELR* i *T*

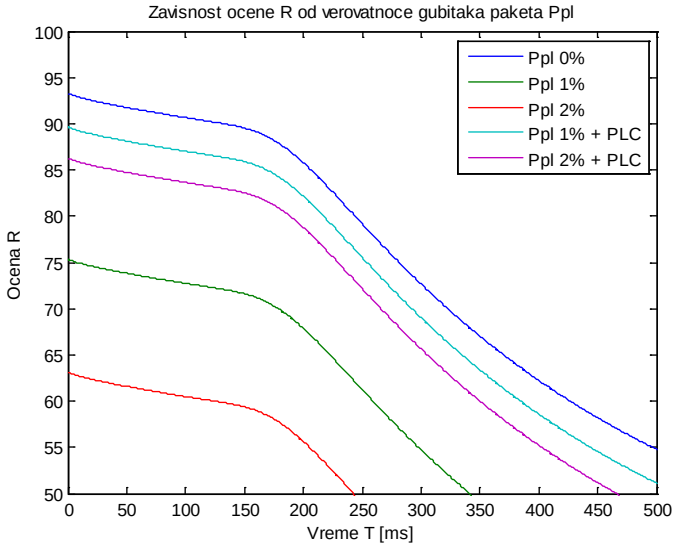
Može se videti da za najpovoljnija dva slučaja sa stanovišta parametra *TELR* (65dB i 60dB) postoje dva uočljiva nagiba. U prvom delu krive je nagib blaži, a u drugom delu krive je nagib strmiji. Naime, za ove dve vrednosti *TELR* parametra, eho još uvek nije dovoljno glasan, tako da je bitnije ukupno kašnjenje, pa kada ono dođe u region oko 200ms, dolazi do strmijeg

pada u kvalitetu veze jer je kašnjenje postalo preveliko za udoban razgovor (problem da korisnik nije siguran da ga je suprotna strana čula zbog presporog odziva koje nastaje zbog velikog vremena propagacije govornog signala od govornika do slušaoca). Međutim, u ostalim slučajevima je eho dovoljno glasan da počinje da smeta pa već u startu dolazi do strmijeg pada u kvalitetu govorne veze tj. oceni *R*.

Na slici 8.3.1.2 je prikazan uticaj vremena kašnjenja u jednom smeru *T* za G.711 koder i različite vrednosti verovatnoće gubitaka paketa P_{pl}, pri čemu se podrazumevaju slučajni gubici ($Burst_R$ je jednak 1), i takođe nema nadoknade izgubljenih paketa. Pošto se koristi G.711 koder, B_{pl} je postavljen na vrednost 4.3 u skladu sa preporukom G.113. Kao što se može videti sa slike, gubici paketa drastično obaraju kvalitet razgovora. Već za 1% gubitaka paketa, dolazi u startu do obaranja ocene na 75 čak i kada nema kašnjenja (*T*=0ms). Gubici veći od 3%, praktično nisu dopustivi jer u startu obaraju ocenu *R* ispod 50, što nije preporučljivo na osnovu preporuke G.107.



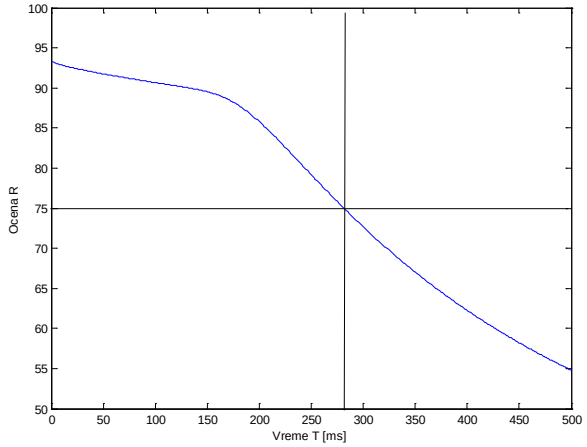
Slika 8.3.1.2. Uticaj parametra P_{pl} i *T*



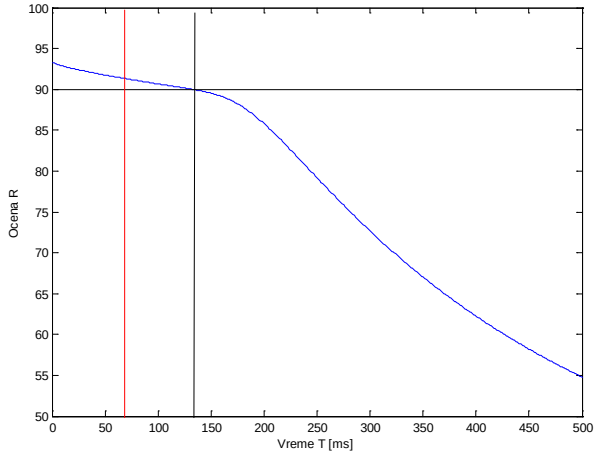
Slika 8.3.1.3. Uticaj PLC na ocenu *R*

Ako se vrši nadoknada izgubljenih paketa (PLC) tada se situacija može značajno popraviti. Ukoliko se koristi PLC definisan u dodatku 1 G.711 preporuke, tada je vrednost parametra B_{pl} jednaka 25.1. Na slici 8.3.1.3 je prikazana zavisnost ocene *R* od gubitaka paketa za slučajeve kada se koristi i ne koristi nadoknada izgubljenih paketa (PLC). Kao što se vidi, krive za slučajeve kada se koristi PLC su znatno bliže slučaju kada nema izgubljenih paketa, što ide u prilog upotrebi PLC, pa je upotreba PLC zaštite praktično obavezna u mrežama sa značajnim gubicima.

Grafici mogu da pomognu u vizuelnom utvrđivanju budžeta za kašnjenje u mreži. Kao primer uzmimo slučaj kada koristimo G.711 koder i svi ostali parametri su postavljeni na difolt vrednosti kao u G.107 preporuci (nema gubitaka u mreži, *TELR*=65dB, ...). Pretpostavimo da je minimalna dozvoljena ocena *R*=75 za govorne veze u projektovanoj mreži. Tada povlačenjem horizontalne linije na krivu zavisnosti ocene *R* u nivou ocene 75 dobijamo tačku na krivoj koja odgovara oceni 75. Vertikalna linija koja prolazi kroz tu tačku će dati presek na x osi, tačnije dobićemo vrednost maksimalnog kašnjenja mreže (pošto G.711 koder unosi zanemarljivo kašnjenje kao što se vidi iz tabele 8.3.2). Kao što se vidi sa slike, maksimalno kašnjenje je oko 280ms da bismo ostali u definisanim granicama za ocenu *R*.



Slika 8.3.1.4. Određivanje budžeta mrežnog kašnjenja za G.711 koder



Slika 8.3.1.5. Određivanje budžeta mrežnog kašnjenja za G.723.1 koder

Pretpostavimo da umesto G.711 kodera koristimo G.723.1 koder sa protokom 6.3kb/s. Tada treba da uračunamo pad kvaliteta I_e kodera G.723.1, kao i akumulirano kašnjenje kodera. Na osnovu tabela 8.3.2 i 8.3.3 imamo da je I_e=15, a akumulirano kašnjenje 67.5ms ako pretpostavimo da je vreme paketizacije zanemarljivo, tj. da su svi linkovi u mreži dovoljno brzi (megabitski linkovi ili brži). Pošto nema grešaka u prenosu možemo koristiti krivu dobijenu i za G.711 jer su svi ostali parametri sem I_e isti (parametar T se varira na x-osi, a u njegov sastavni deo ulazi akumulirano vreme koje je takođe razlika u odnosu na G.711 slučaj). Sada I_e u startu obara ocenu R za 15 jedinica, pa je minimalna dozvoljena ocena 75+15=90. Takođe u startu dodajemo fiksnu količinu kašnjenja od 67.5ms (crvena vertikalna linija). Ponavljamo postupak iz prethodnog primera i dobijamo da je presek linije (crne) sada na oko 133ms. Pošto je deo budžeta kašnjenja potrošen na akumulirano kašnjenje kodera, budžet kašnjenja u mreži je jednak razlici pozicije na x-osi crne i crvene linije tj. 133ms-67.5ms=65.5ms (ako se dobije negativna vrednost onda koder ne može da se upotrebi jer unosi preveliku degradaciju kvaliteta, tj. čak i za kašnjenje u mreži od 0ms donja granica za R je probijena). Vidimo da je u ovom slučaju budžet za kašnjenje u mreži mnogo manji što je posledica upotrebe drugog kodera koji je uneo i degradaciju usled kompresije i akumulirano kašnjenje. Međutim, protok govornog sadržaja je znatno niži, što znači da može da se ostvari veći broj istovremenih govornih veza.

Na sličan način se određuje budžet mrežnog kašnjenja kada ima gubitaka u mreži samo se tada za G.711 koder koriste krive sa slika 8.3.1.2 i 8.3.1.3 u zavisnosti koji procenat gubitaka je u pitanju i da li se koristi PLC. Ukoliko je drugi koder u pitanju tada se i dalje može koristiti kriva dobijena za G.711, ali je potrebno proračunati konkretnu vrednost $I_{e\text{-}eff}$ za oba kodera da bi se dobilo konstantno odstupanje drugog kodera u odnosu na G.711 koder u pogledu ovog parametra. Ovo odstupanje se koristi na identičan način kao I_e parametar u prikazanom primeru za G.723.1 koder kada nema gubitaka paketa (kada nema gubitaka paketa I_e=0 za G.711 koder, pa je odstupanje jednako I_e parametru drugog kodera).

Takođe, prikazani grafici se mogu koristiti i za planiranje izbora kodera u zavisnosti od željenog nivoa kvaliteta (definisan kroz minimum dozvoljene vrednosti ocene R). Tada se postupak iz prethodnih primera ponavlja za svaki od kodera i određuje se maksimalan budžet kašnjenja u mreži. Oni koderi koji probiju dozvoljeni budžet kašnjenja u mreži se odbacuju (ako je definisano maksimalno dozvoljeno kašnjenje u mreži), kao i oni koji u startu probiju minimalnu vrednost ocene R usled pada kvaliteta zbog kompresije (I_e ili $I_{e\text{-}eff}$ u zavisnosti da li mreža ima gubitke paketa ili ne) i akumuliranog kašnjenja. Oni koderi koji ispune uslove (ne budu odbijeni zbog ograničenja iz prethodne rečenice) ulaze u izbor. Izbor može biti takav da se izabere koder koji postiže maksimalnu ocenu R, ili koji postiže maksimalan budžet za kašnjenje u mreži (ako nije definisano maksimalno kašnjenje u mreži), ili koji ima najmanji protok, ili da se definiše kombinacija prethodno navedenih kriterijuma koja bi se koristila u izboru.