

**ŠIROKOPOJASNE TELEKOMUNIKACIONE MREŽE**  
**– Poglavlje 2 –**

## 2 SDH

SDH (*Synchronous Digital Hierarchy*) mreže predstavljaju transportne mreže namenjene za transport raznovrsnih signala poput IP saobraćaja, ATM ćelija, PDH (*Plesiochronous Digital Hierarchy*) struktura, itd. Za SDH mreže se može reći da predstavljaju transportne mreže druge generacije, razvijene da obezbede veće protoke i fleksibilnije upravljanje i nadgledanje mrežom u odnosu na transportne mreže prve generacije, odnosno PDH mreže. Danas su aktuelne i transportne mreže treće generacije, OTN (*Optical Transport Networks*) mreže koje podižu podržane protoke na veći nivo upotrebom DWDM (*Dense Wavelength Division Multiplex*) tehnike i koje ćemo ukratko obraditi u sledećem poglavlju. Međutim, s obzirom na trenutnu pokrivenost SDH mrežama, SDH mreže će još dugo biti prisutne u telekomunikacionoj infrastrukturi svake zemlje.

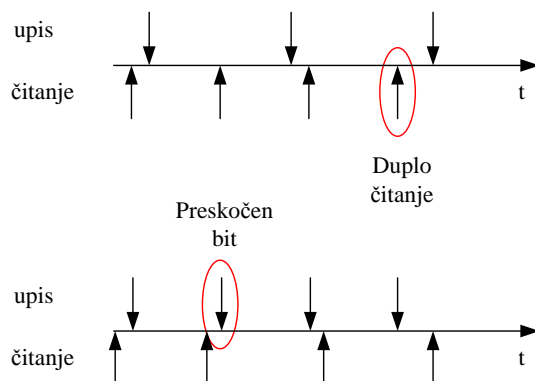
U okviru ovog poglavlja će prvo biti ukratko objašnjene PDH mreže da bi se lakše uočili razlozi za uvođenjem SDH mreža. Potom će biti objašnjeni osnovni elementi SDH mreže, a potom i SDH hijerarhija, kao i strukture okvira koje se koriste u okviru SDH mreža. Na kraju će biti objašnjene pojedine tehnike značajne za kvalitetan rad SDH mreža poput zaštitnih mehanizama, konkatanacije i dr. Napomenimo da SDH predstavlja standard koji se koristi u Evropi, dok se u SAD i Japanu koristi SONET standard.

### 2.1. PDH

PDH tehnologija se zasniva na plesiohronom multipleksiranju kao što i samo ime kaže. Termin plesiohron označava da svi uređaji rade na 'istim' frekvencijama, ali da nisu međusobno sinhronizovani. Pod 'istim' frekvencijama se podrazumeva da se nominalna frekvencija, na kojoj bi uređaj trebao da radi, kreće u određenim granicama oko idealne vrednosti nominalne frekvencije, pri čemu maksimalno moguće odstupanje zavisi od preciznosti oscilatora koji se koristi za generisanje nominalne frekvencije. Što je preciznost oscilatora bolja, odstupanje će biti manje. Na primer, ako je preciznost oscilatora  $A$ , to znači da se generisana frekvencija kreće u granicama od  $f_{nom}-A\cdot f_{nom}$  do  $f_{nom}+A\cdot f_{nom}$ , gde je  $f_{nom}$  idealna vrednost nominalne frekvencije. Na primer, ako je idealna vrednost nominalne frekvencije 100MHz, i preciznost oscilatora  $10^{-10}$ , tada se generisana frekvencija kreće u granicama 99.99999999MHz i 100.00000001MHz. Međutim, ovaj naizgled sitni opseg u kojima se kreće generisana (radna) frekvencija može da izazove velike probleme, tzv. preskoke koji se manifestuju ili u duplom čitanju nekog bita ili u potpunom preskoku nekog bita (bit se uopšte ne pročita). Duplo čitanje se javlja kada je radna frekvencija uređaja koji prima brža od radne frekvencije uređaja koji šalje, a preskok bita se javlja u suprotnom slučaju. Ova dva slučaja su prikazana na slici 2.1.1.

PDH tehnologija se sastoji u kreiranju hijerarhija. E1 nivo predstavlja najniži nivo hijerarhije. Na ovom nivou se šalju ramovi neprestano jedan za drugim. E1 ram se sastoji iz 32 kanala, gde svaki kanal ima 64kb/s protok. Jedan kanal se rezerviše za potrebe sinhronizacije rama. Naime, prijemna strana mora da se sinhroniše na strukturu rama da bi mogla da zna gde se koji kanal nalazi, pa se otuda jedan kanal rezerviše za potrebe sinhronizacije rama (prvi kanal u E1 ramu). Protok kanala 64kb/s je posledica činjenice da su E1 strukture prvenstveno bile namenjene za prenos digitalizovanih govornih signala telefonskih razgovora u doba svog nastajanja. Preskoci su izazivali klizanje strukture rama za jednu poziciju (smer klizanja je zavisio od tipa preskoka), što kao posledicu ima gubitak sinhronizacije rama i proces hvatanja

sinhronizacije je morao ponovo da se pokreće. Pošto preciznost oscilatora ima svoja tehnološka ograničenja, rešenje za proređivanje preskoka je nađeno upotrebom elastične memorije čiji je princip rada objašnjen u predmetu Komutacioni sistemi.



Slika 2.1.1. Tipovi preskoka

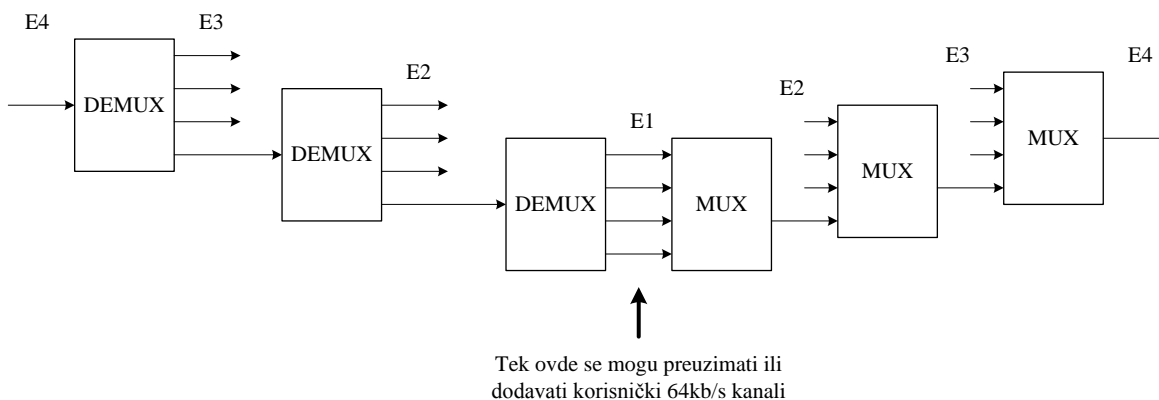
Tabela 2.1.1 - PDH hijerarhija

Nivo hijerarhije	Ukupan protok [Mb/s]	Korisnički protok [Mb/s]	Broj korisničkih 64kb/s kanala	Udeo korisnog dela	ITU-T preporuka
E1	2.048	1.92	30	93.75%	G.704
E2	8.448	7.68	120	90.91%	G.742
E3	34.368	30.72	480	89.39%	G.751
E4	139.264	122.88	1920	88.24%	G.751
E5	564.992	491.52	7680	87.00%	-

Grupisanjem četiri E1 strukture se kreira sledeći nivo PDH hijerarhije - E2 nivo. Grupisanjem četiri E2 strukture se kreira E3 nivo hijerarhije. Na sličan način se kreiraju E4 i E5 nivoi hijerarhije, kao što se može videti iz tabele 2.1.1. E5 nivo nije standardizovan kao što se i vidi iz tabele. Napomenimo da se prikazani nivoi hijerarhije odnose na evropski standard, dok su SAD i Japan imali tzv. T strukture, tj. T nivoe hijerarhije (pri čemu postoje i manje razlike između SAD i Japana u višim nivoima T hijerarhije). Multipleksiranje struktura niže hijerarhije u strukturu više hijerarhije se radi na nivou bita tj. vrši se učešljavanje bita struktura nižeg nivoa. Treba uočiti da protok  $E_{i+1}$  strukture nije jednak četiri puta uvećanom protoku  $E_i$  strukture. Razlog je dodavanje dodatnih bita neophodnih za sinhronizaciju strukture na višem nivou hijerarhije ( $E_{i+1}$ ), ali i bita za pozitivno poravnanje multipleksiranih struktura nižeg nivoa hijerarhije ( $E_i$ ). Usled ovih dodatnih bita, udeo korisnog dela opada sa porastom nivoa hijerarhije kao što se može videti iz tabele 2.1.1. Biti za poravnanje su neophodni zbog potencijalnih razlika u frekvencijama na kojima su generisane  $E_i$  strukture, a koje su posledica plesiohronog rada, kao i konačne tačnosti oscilatora kojima je generisana radna frekvencija uređaja koji su generisali dotičnu strukturu nižeg nivoa. Pri kreiranju  $E_{i+1}$  strukture pretpostavlja se da su  $E_i$  strukture generisane na maksimalnoj mogućoj frekvenciji ( $f_{nom} + A \cdot f_{nom}$ ) pa se vrši tzv. pozitivno poravnanje. Ako su  $E_i$  strukture, ipak, generisane na nižoj frekvenciji, tada će povremeno nedostajati bita za popunu svih mesta u  $E_{i+1}$  strukturi namenjenih dotičnoj  $E_i$  strukturi. Otuda se u tim situacijama ne koriste biti poravnanja, što se signalizira preko kontrolnih bita, tako da prijemna strana zna koji biti su deo  $E_i$  strukture tj. u kojim situacijama treba da uzme u obzir bite poravnanja, a u kojim ih ne uzima u obzir. Očigledno, uz bite poravnanja, moraju da se koriste i

kontrolni biti kao signalizacija kada se biti poravnanja koriste za smeštanje bita  $E_i$  strukture, a kada ne.

Veliki problem PDH mreža, odnosno struktura je bila njihova nefleksibilnost. Da bi se pristupilo samo jednom korisničkom kanalu od 64kb/s u višem nivou hijerarhije, bilo je neophodno izvršiti demultipleksiranje do najnižeg nivoa hijerarhije, što je nepraktičan i neekonomičan proces. Razlog za ovaj problem u pristupu individualnom korisničkom kanalu potiče od činjenice da je teško detektovati koji biti pripadaju dotičnom kanalu uzimajući u obzir multipleksiranje na nivou bita, ali i upotrebu bita za poravnanje. Slično važi i za pristupanje  $E_i$  strukturi u  $E_{i+2}$  ili višoj strukturi. Otuda PDH mreže nisu omogućavale kreiranje fleksibilnih mrežnih infrastruktura koje bi omogućavale lako dodavanje i skidanje korisničkih kanala (ili  $E_i$  struktura u  $E_{i+2}$  ili višim strukturama) u mrežnim čvorovima, već su se za te operacije morali koristiti kompletni multiplekseri i demultiplekseri (po kompletnim podrazumevamo multipleksiranje/demultipleksiranje od najnižeg do najvišeg nivoa hijerarhije i obrnuto). Ovaj problem je ilustrovan na slici 2.1.2. Kao što se vidi mora se izvršiti kompletno demultipleksiranje da bi se preuzeo korisnički kanal ili da bi se dodao korisnički kanal i potom ponovo izvršiti kompletno multipleksiranje da bi se ponovo formirala struktura višeg nivoa hijerarhije, u datom primeru  $E_4$  struktura. Kao što vidimo, mrežni čvorovi ne mogu jednostavno da preuzimaju grupe korisničkih kanala ili da ubacuju grupe korisničkih kanala u strukture viših hijerarhijskih nivoa pošto se uvek mora vršiti neekonomično kompletno demultipleksiranje i potom kompletno multipleksiranje.



**Slika 2.1.2. Problem pristupa individualnom korisničkom kanalu**

Dodatni problem je bila nemogućnost praćenja kvaliteta prenosa, odnosno menadžmenta mreže, koja je posledica upravo navedene nefleksibilnosti PDH mreža.

## 2.2. Struktura SDH mreža

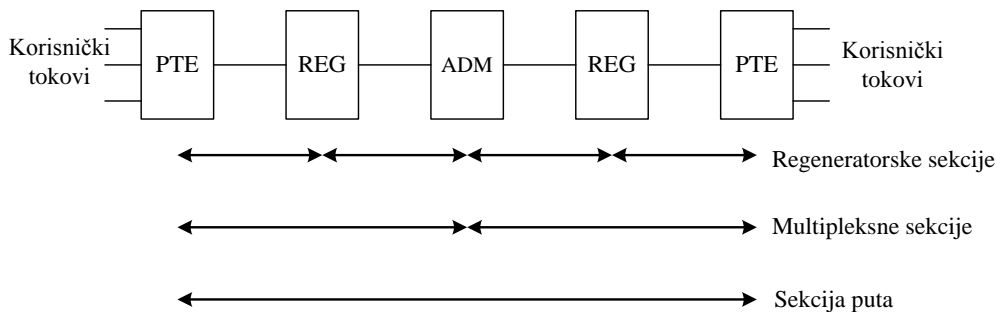
Kao što smo videli osnovni problem PDH mreža je bila njihova nefleksibilnost i kao posledica neekonomičnost pri izradi mrežnih PDH čvorova zbog potrebe za kompletnim multipleksiranjem i demultipleksiranjem. Pri tome, PDH mreže nisu davale mogućnost kvalitetnog upravljanja i nadgledanja mreže, što je otežavalo operaterima održavanje takvih mreža. Takođe, PDH mreže su bile zasnovane na upotrebi bakarnih provodnika što je bilo logično ako se uzme u obzir vreme njihovog nastanka. S početkom upotrebe optičkih linkova pojavila se mogućnost instaliranja još većih protoka u transportnim mrežama. Takođe, razvoj tehnologije je omogućio upotrebu sofisticiranijih tehnika i algoritama u transportnim mrežama,

pa je došlo do potrebe za kreiranjem i standardizacijom nove (druge) generacije transportnih mreža koje bi prevazišle ograničenja PDH mreža, a istovremeno bile interoperabilne sa PDH mrežama s obzirom na njihovu tadašnju rasprostranjenost. Takođe, ideja je bila i smanjenje razlika u standardima između evropske i američke strane koja je bila prisutna u PDH sistemima. Kao rezultat ovih težnji su nastale standardizovane SDH i SONET tehnologije koje su veoma slične, pri čemu je SDH standard koji se koristi u Evropi, a SONET u SAD i Japanu.

**Tabela 2.2.1 - SDH hijerarhija**

Nivo hijerarhije	Ukupan protok [Mb/s]
STM-1	155.52
STM-4	622.08
STM-16	2488.32
STM-64	9953.28
STM-256	39813.12

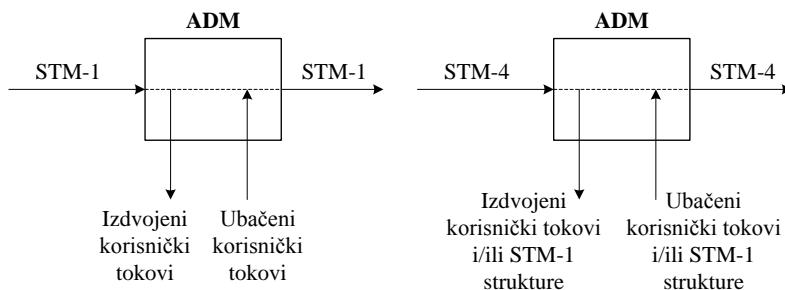
SDH koristi isti princip hijerarhija kao i PDH tehnologija. Tabela 2.2.1 prikazuje definicije SDH nivoa hijerarhije. Očigledno se i ovde grupisanjem četiri strukture sa nivoa STM-*i*, može dobiti sledeći nivo hijerarhije STM-4-*i*, pri čemu ukupni protok raste tačno četiri puta (nema dodatnih bita kao kod PDH struktura, čime je izbegnut problem smanjenja iskorišćenja porastom nivoa hijerarhije). STM (*Synchronous Transport Module*) je skraćenica za sinhroni transportni modul. Za razliku od PDH mreža, SDH je sinhronizovana mreža u kojoj su svi čvorovi sinhronizovani na isti referentni takt, čime se značajno smanjuje učestanost preskoka.



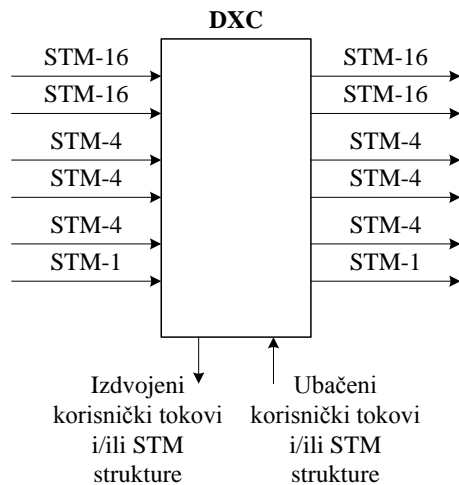
**Slika 2.2.1. Definicije sekcija u SDH mreži**

Kao i većina drugih telekomunikacionih mreža, i SDH je bazirana na slojevitoj strukturi. pri čemu se razlikuju regeneratorska sekcija, multipleksna sekcija i sekcija puta (navođenje je dato od najnižeg sloja ka najvišem sloju). Na slici 2.2.1 je prikazana jedna tipična deonica u SDH mreži i na njoj su prikazani domeni navedenih sekcija. PTE (*Path Termination*) predstavlja uređaj koji omogućava priključenje korisnika tj. njihovih tokova (na primer, ATM tokovi, IP saobraćaj, PDH strukture) i multipleksiranje tih tokova u SDH strukturu koja se prenosi preko SDH mreže (pod SDH strukturom podrazumevamo STM-*i* okvir, pri čemu se ti okviri šalju kontinualno jedan za drugim). Takođe, na prijemu se rade obrnute operacije u kojima se izvlače korisnički podaci iz SDH struktura i šalju korisnicima u vidu korisničkih tokova odgovarajuće strukture kakva je bila na predajnoj strani tj. na ulazu u SDH mrežu. REG predstavlja regeneratore koji vrše osvežavanje signala pod čime se podrazumeva rekonstrukcija vrednosti bita (ne vrši se pojačavanje, već rekonstrukcija da bi se izbegla akumulacija negativnog dejstva šuma) i takta sa kojim ti rekonstruisani biti treba da se šalju na izlazu regeneratora. Regeneratori su neophodni na dugačkim deonicama, pošto te deonice mogu biti duže od maksimalnog

rastojanja koje podržavaju optički linkovi (a koja su velika - nekoliko desetina kilometara, pa i više). ADM (*Add/Drop Multiplexer*) predstavlja tzv. 'add/drop' multipleksere kojima se omogućava izvlačenje pojedinih korisničkih tokova iz SDH strukture, ili ubacivanje korisničkih tokova u SDH strukturu što je funkcija koja je bila veoma kompleksna u PDH mrežama i predstavljala jedan od najvećih problema PDH mreža. Za razliku od PTE, ovde se ne kreira/terminira kompletna SDH struktura već se dodaju/oduzimaju korisnički tokovi po trenutnoj potrebi, odnosno konfiguraciji ADM uređaja. Na primer, u ADM uređaju se tako može izvaditi jedan korisnički tok, a potom ubaciti drugi korisnički tok na njegovo mesto (naravno, pod uslovom da taj drugi korisnički tok ima iste karakteristike tj. da može da stane u oslobođeni deo SDH strukture), kao što je prikazano na slici 2.2.2. Osim korisničkih tokova, mogu se izdvajati i dodavati i SDH strukture nižeg nivoa hijerarhije (na primer, iz STM-4 strukture se može izdvojiti i/ili ubaciti STM-1 struktura kao što je prikazano na slici 2.2.2). Pored ovih uređaja prikazanih na slici 2.2.1, u SDH mrežama je bitan i DXC (*Digital Cross-Connect*) koji predstavlja proširenje funkcionalnosti ADM radi omogućavanja interkonekcije dve ili više SDH mreža. DXC omogućava takođe izdvajanje i ubacivanje korisničkih tokova i SDH struktura nižeg nivoa hijerarhije (ADM funkcionalnost), ali isto tako i komutiranje korisničkih tokova i SDH struktura nižeg nivoa hijerarhije sa SDH struktura na ulaznim portovima na odgovarajuće SDH strukture na izlaznim portovima. U slučaju kada ADM direktno prima korisničke tokove (tj. omogućava povezivanje korisnika na SDH mrežu), tada ADM ima i ulogu PTE uređaja. Isto važi i za DXC. Napomenimo da je SDH mreža zasnovana na komutaciji kola, pa se prilikom uspostave neke veze, vrši konfiguracija mrežnih uređaja (PTE, ADM, DXC) tako da se korisnički tok komutira kroz SDH mrežu od izvorišta do odredišta.

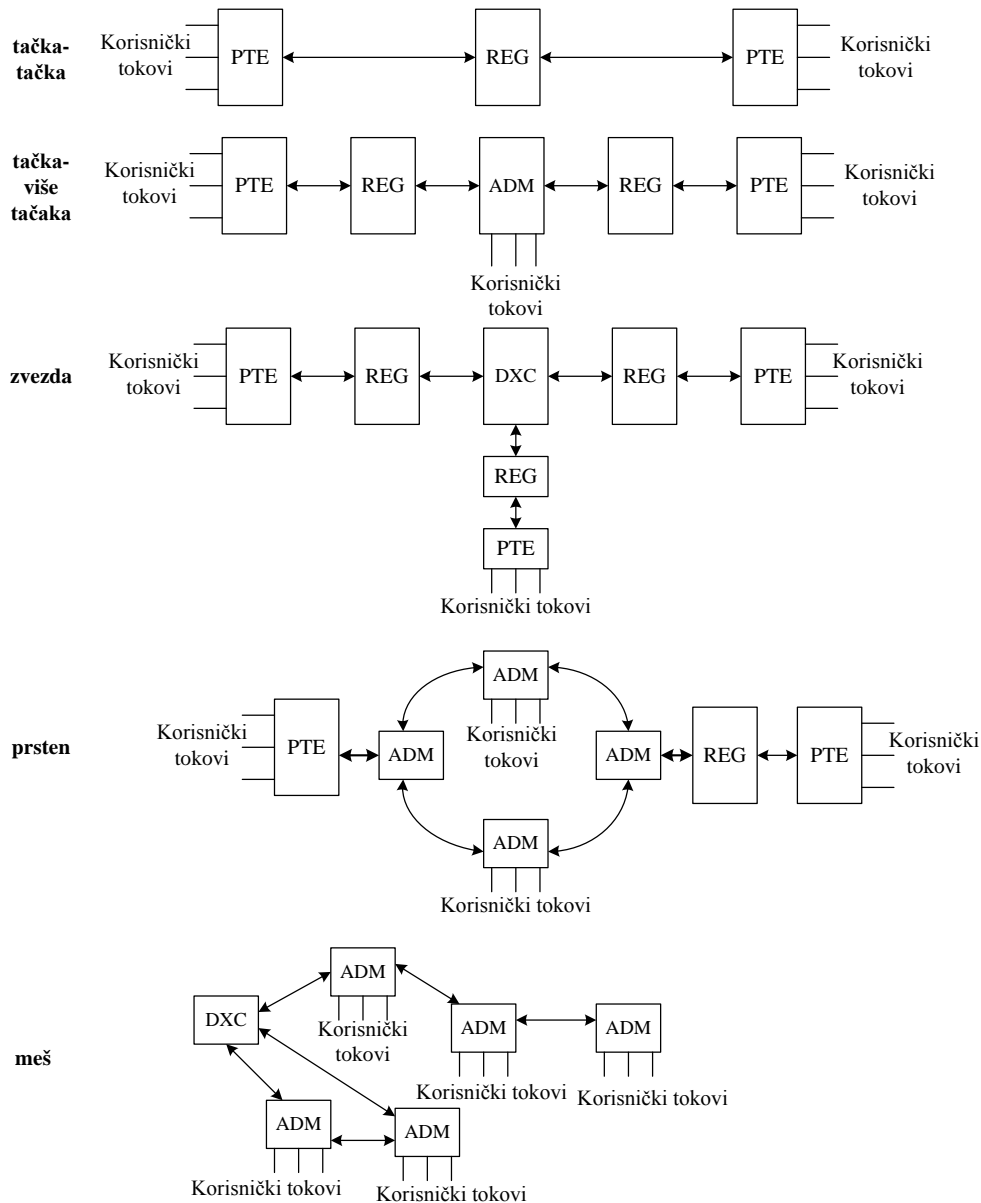


Slika 2.2.2. ADM princip rada



Slika 2.2.3. DXC

Na slici 2.2.3 je prikazan primer jednog DXC komutatora, koji na ulazima/izlazima ima 2 STM-16 toka, 3 STM-4 toka i 1 STM-1 tok. DXC omogućava komutiranje između ovih tokova. Na primer, neka od STM-4 struktura sadržana u gornjem STM-16 toku se može komutirati u donji STM-16 tok na izlazu. Isto važi i za korisničke tokove. Na primer, neki korisnički tok sa STM-16 gornjeg ulaza se može komutirati u STM-1 tok na izlazu. Pri tome, mogu da se obavljaju i ADM funkcionalnosti, tj. mogu se izdvajati i ubacivati korisnički tokovi i SDH strukture nižeg nivoa hijerarhije u SDH tokove koji su prisutni na ulazima/izlazima DXC.



**Slika 2.2.4. Topologije SDH mreže**

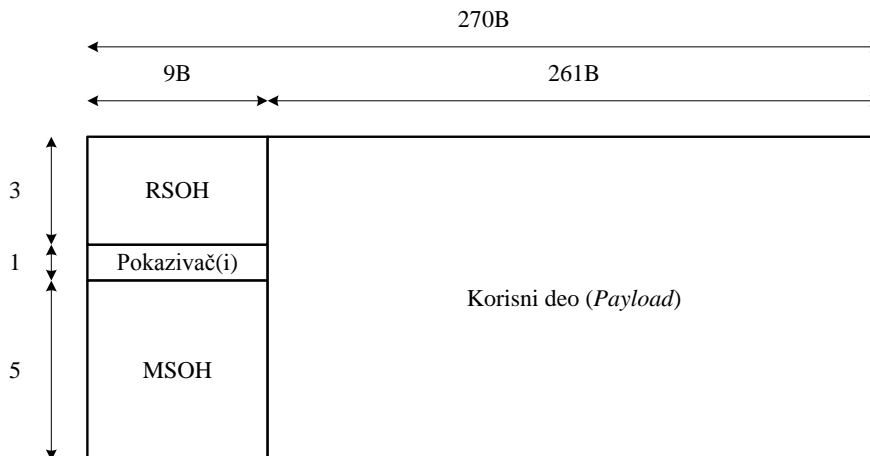
Nakon što smo predstavili funkcije uređaja koji se javljaju u SDH mreži, jasnija je podela sekcija i njihovi domeni važenja u primeru sa slike 2.2.1. Kao što vidimo regeneratorska sekcija je najpribližnija tumačenju fizičkog nivoa i regeneratorska sekcija se definiše na nivou optičkog linka, odnosno između svaka dva međusobno povezana uređaja. Multipleksna sekcija se definiše između inteligentnijih uređaja koji mogu da manipulišu korisnim sadržajem SDH strukture u

smislu njegovog ubacivanja i izdvajanja. U PTE se kreira originalni multipleks korisničkih tokova u SDH strukture, odnosno vrši terminacija SDH strukture i konačno izvlačenje korisničkih tokova iz SDH mreže. ADM takođe omogućava ubacivanje i izdvajanje korisničkih tokova, kao i SDH struktura nižeg nivoa. DXC bi takođe predstavljao granicu za multipleksnu sekciju. Sekcija puta se definiše između krajnjih uređaja (PTE) na koje su povezani korisnički tokovi. Može se reći da sekcija puta predstavlja granicu s kraja na kraj - između ulaza u SDH mrežu i izlaza iz SDH mreže.

Postoji više mogućih topologija SDH mreže. Na primer, tačka-tačka topologija gde su dva PTE elementa direktno povezana ili preko jednog ili više regeneratora. Isto tako može se kreirati i tačka-više tačaka topologija gde se dodatno između PTE elemenata ubacuje i ADM. Takođe, moguće je kreirati i meš topologiju. Može se kreirati i zvezda topologija ako se koriste DXC uređaji. Međutim, najpopularnija topologija koja se dominantno koristi u praksi je prsten topologija zbog omogućavanja implementacija efikasnih mehanizama zaštite od kvarova u mreži kao što ćemo i videti nešto kasnije u poglavlju. Primeri svih navedenih topologija su prikazani na slici 2.2.4. Dvosmerni linkovi prikazani na slici se, ustvari, sastoje od dva jednosmerna optička linka. Videćemo kasnije da se u dominantno korišćenoj prsten topologiji pored prstena od dva optička linka, mogu formirati i prsteni od četiri optička linka radi pružanja još kvalitetnije zaštite.

### 2.3. Struktura STM-1 okvira

STM-1 predstavlja najniži nivo hijerarhije. U okviru ovog potpoglavlja ćemo predstaviti STM-1 okvir koji predstavlja STM-1 periodičnu strukturu koja se kontinualno šalje (STM-1 okviri se šalju neprestano jedan za drugim). STM-1 okvir je prikazan na slici 2.3.1.



Slika 2.3.1. STM-1 okvir

STM-1 okvir se sastoji iz 2430 bajtova ( $9 \times 270B = 2430B$ ), pri čemu jedan STM-1 okvir traje  $125\mu s$ . Okvir se tipično prikazuje kao pravougaonik od devet redova, gde svaki red ima 270 bajtova. Bajtovi okvira se šalju sledećim redosledom - šalju se prvo bajtovi prvog reda, pa drugog reda i tako redom do poslednjeg reda, pri čemu je redosled slanja bajtova u redu sleva nadesno (prvi se šalje krajnji levi bajt reda). Kao što vidimo okvir sadrži zaglavlje koje se sastoji od tri dela. RSOH (*Regenerator Section Overhead*) predstavlja zaglavlje koje je relevantno na nivou regeneratorske sekcije tj. razmenjuje se između dva uređaja na krajevima regeneratorske sekcije. MSOH (*Multiplex Section Overhead*) predstavlja zaglavlje koje je relevantno na nivou multipleksne sekcije tj. razmenjuje se između dva uređaja na krajevima multipleksne sekcije.



MSOH deo transparentno prolazi kroz regeneratore. Pokazivač (*POH - Path Overhead*) deo je relevantan na nivou sekcije puta tj. razmenjuje se između dva uređaja na krajevima sekcije puta. Pri tome, u ovom delu se može nalaziti jedan ili više pokazivača u zavisnosti od strukture korisnog dela kao što ćemo nešto kasnije videti. Pokazivač pokazuje na početak korisničke strukture koja je spakovana u korisnom delu i neophodan je jer korisnička struktura može da pluta u korisnom delu tj. njen početak ne mora da se uvek nalazi na istoj poziciji. Korisni deo sadrži korisničke strukture, tj. korisničke tokove koji se prenose kroz SDH mrežu unutar STM-1 okvira.

Na slici 2.3.2 je prikazana struktura RSOH dela na nivou bajtova. Sa  $\Delta$  označena bajtovi čije tumačenje zavisi od tipa medijuma za prenos tj. koriste se u slučaju aplikacija koje su zavisne od tipa medijuma za prenos. Sa NR su označeni bajtovi koji su rezervisani za nacionalnu upotrebu, dok su neobeleženi bajtovi rezervisani za eventualnu buduću upotrebu. A1 i A2 bajtovi se koriste za sinhronizaciju na STM-1 okvir. Na osnovu ovih bajtova prijemna strana hvata sinhronizaciju na STM-1 okvir, a potom (kada se uđe u stanje sinhronizacije) i prati stanje sinhronizacije što je neophodno za ispravno procesiranje STM-1 okvira, na primer, izvlačenje korisničkog toka iz korisnog dela okvira. Ako se ne bi znalo gde počinje STM-1 okvir, celokupni primljeni sadržaj bi bio potpuno beskoristan. Bajt A1 ima vrednost 11110110, a bajt A2 vrednost 00101000. Napomenimo da se sadržaj STM okvira (i korisni deo i zaglavlje) skrembluje da bi se izbegao dug niz 0 ili dug niz 1 koji bi mogao da onemogući prijemnu stranu da rekonstruiše takt predajne strane. Samo se prvi red RSOH dela ne skrembluje da bi bilo moguće odrediti početak STM-1 okvira.

A1	A1	A1	A2	A2	A2	J0	NR	NR
B1	$\Delta$	$\Delta$	E1	$\Delta$		F1	NR	NR
D1	$\Delta$	$\Delta$	D2	$\Delta$		D3		

**Slika 2.3.2. RSOH deo STM-1 okvira**

J0 bajt se koristi za slanje identifikacije predajne strane. Na osnovu prijema očekivanog identifikatora predajne strane, prijemna strana zna da je konekcija sa suprotnom stranom i dalje operativna tj. aktivna. Identifikator se može slati ili u vidu jednog bajta identifikacije, ili u vidu strukture od 16 bajtova koja sadrži identifikaciju po formatu definisanom u G.831 preporuci. U slučaju međunarodne veze, obavezna je upotreba strukture od 16 bajtova. B1 predstavlja bajt provere parnosti koji se računa za kompletan sadržaj prethodnog STM-1 okvira (pri čemu se misli na sadržaj koji je skremblovan). Proračunata vrednost B1 se postavlja u tekući okvir pre procesa skremblovanja sadržaja tekućeg okvira. B1 bajt predstavlja proveru ispravnosti na nivou regeneratorske sekcije. Vrednost bita u ovom bajtu se proračunava na sledeći način. Sadržaj koji se štiti se deli na bajtove. Bit na poziciji  $i$  B1 bajta predstavlja proveru na parnost svih bita na poziciji  $i$  bajtova korisnog sadržaja tako da ukupan broj 1 na tim pozicijama (računajući  $i$  poziciju  $i$  u B1 bajtu) bude paran. E1 bajt se može koristiti kao govorni kanal između uređaja na krajevima regeneratorske sekcije. Na primer, on se može iskoristiti za govornu komunikaciju između tehničara koji se nalaze na lokacijama dotičnih uređaja. F1 bajt se može iskoristiti kao korisnički kanal između uređaja na krajevima regeneratorske sekcije. Ovaj kanal se može iskoristiti za prenos podataka ili takođe za prenos govorne komunikacije. D1-D3 predstavljaju

kanal za prenos podataka protoka 192kb/s koji se može iskoristiti za nadgledanje ili upravljanje uređaja na krajevima regeneratorske sekcije.

B2	B2	B2	K1			K2		
D4			D5			D6		
D7			D8			D9		
D10			D11			D12		
S1					M1	E2	NR	NR

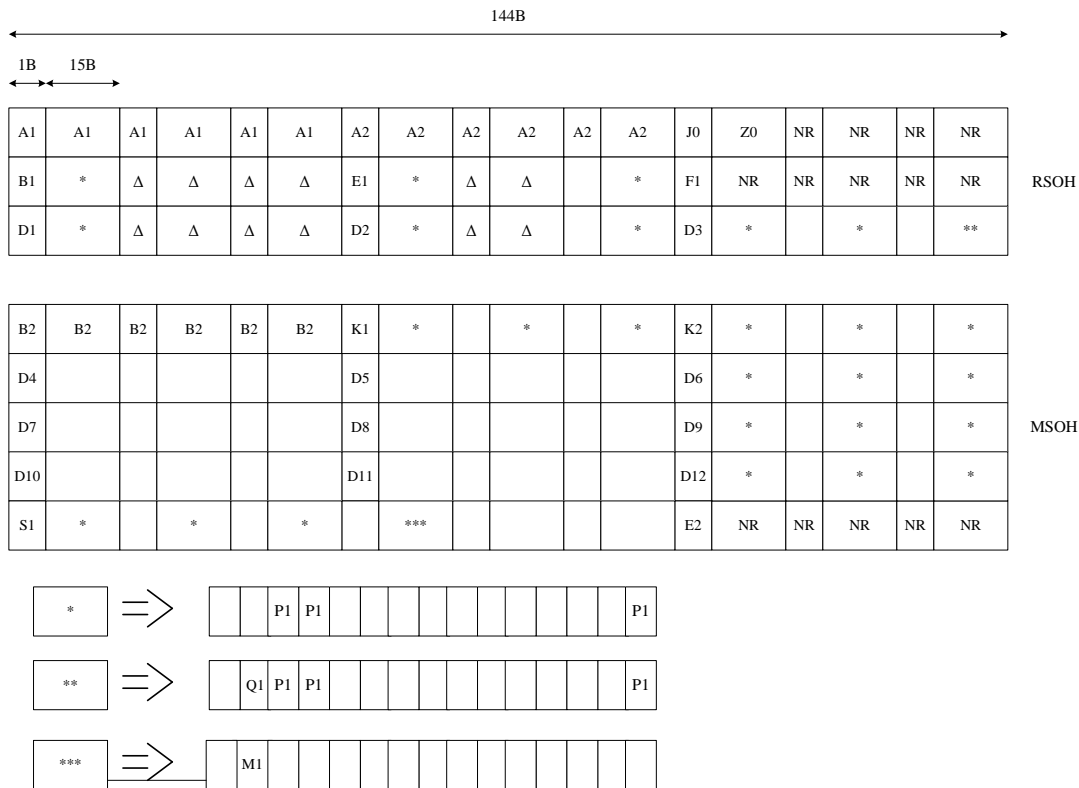
**Slika 2.3.3. MSOH deo STM-1 okvira**

Na slici 2.3.3 je prikazana struktura MSOH dela na nivou bajtova. Sa NR su označeni bajtovi koji su rezervisani za nacionalnu upotrebu, dok su neobeleženi bajtovi rezervisani za eventualnu buduću upotrebu. B2 bajtovi predstavljaju proveru na parnost na nivou multipleksne sekcije. B2 predstavlja bajtove provere parnosti koji se računaju za kompletan sadržaj prethodnog STM-1 okvira, sem MSOH dela. MSOH deo se ne uzima u obzir jer se on menja na nivou regeneratorske sekcije, a B2 treba da radi na nivou multipleksne sekcije. Princip proračuna vrednosti bita u B2 bajtovima je isti kao kod B1 bajta, samo što se sada sadržaj koji se štiti deli na delove dužine 24 bita jer ima ukupno 24 bita u B2 bajtovima (vrednosti pozicije bita  $i$  se kreću u granicama od 0 do 23). K1 i K2 bajtovi se koriste za ostvarivanje zaštite na nivou multipleksne sekcije, pri čemu se pod zaštitom ovde misli na automatsku detekciju kvara radnog puta (multipleksne sekcije) i aktivaciju rezervnog (zaštitnog) puta. D4-D12 predstavljaju bajtove za prenos podataka na nivou multipleksne sekcije (između uređaja na krajevima multipleksne sekcije). D4-D12 bajtovi omogućavaju 576 kb/s kanal za prenos podataka koji se tipično koristi za operacije nadgledanja i upravljanja. Bajt S1 se koristi za oglašavanje statusa sinhronizacije. Tabela 2.3.1 daje vrednosti gornja četiri bita S1 bajta i njihova tumačenja sa stanovišta nivoa kvaliteta sinhronizacije predajne strane multipleksne sekcije (koja je i postavila dotični S1 bajt). M1 bajt se koristi za indikaciju grešaka u prijemu na udaljenoj strani tj. na strani multipleksne sekcije koja je i generisala vrednost M1 bajta. Pošto se bajt tumači na suprotnoj strani multipleksne sekcije, otuda i termin udaljen u 'greška na udaljenoj strani'. M1 bajt (tačnije gornjih sedam bita M1 bajta) predstavlja broj pogrešno primljenih blokova tj. broj blokova koji su neuspešno provereni pomoću B2 bajtova. Pri tome, može se označiti do 24 pogrešna bloka (binarne vrednosti veće od 24 se tumače kao 0 pogrešnih blokova). E2 bajt ima istu ulogu kao E1 bajt iz RSOH dela, samo se koristi na nivou multipleksne sekcije.

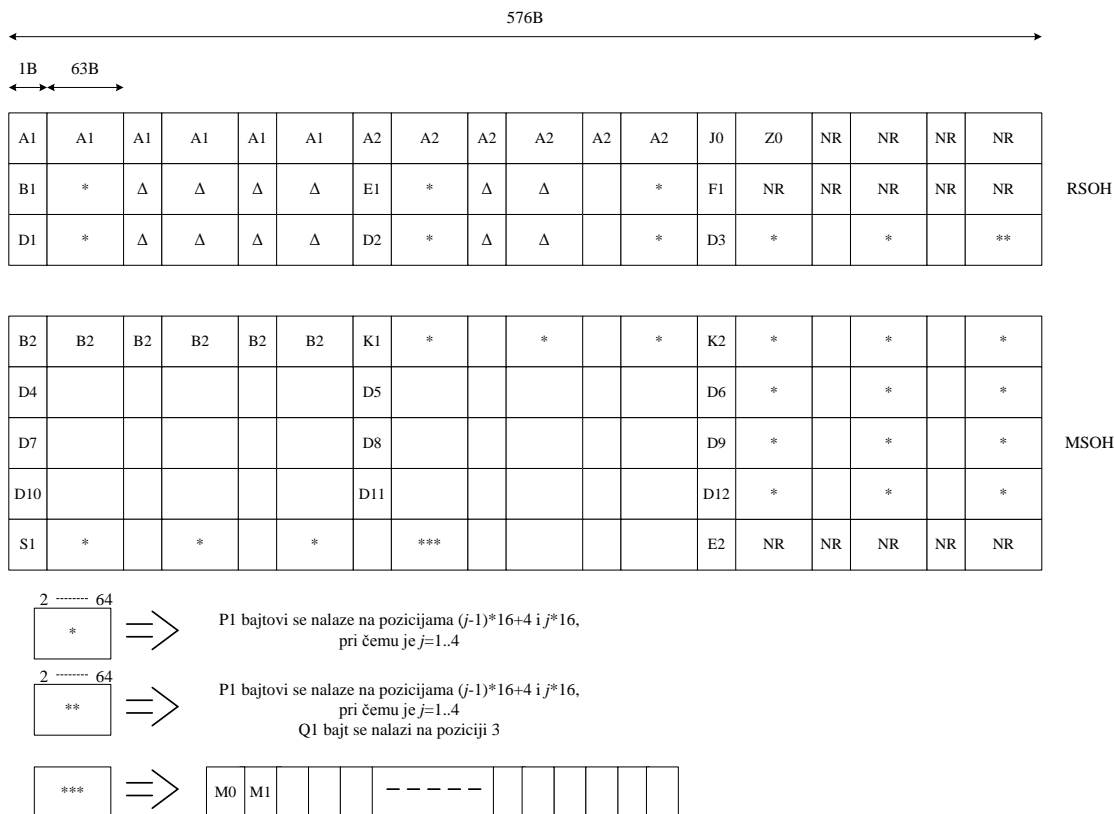
**Tabela 2.3.1 - Nivoi kvaliteta sinhronizacije**

S1 (5..8)	Nivo kvaliteta sinhronizacije
0000	Nepoznat kvalitet
0010	G.811
0100	SSU-A
1000	SSU-B
1011	G.813 (SEC)
1111	Ne koristiti za sinhronizaciju
Ostale vrednosti	Rezervisano za buduću upotrebu





**Slika 2.4.3. Struktura RSOH i MSOH dela STM-16 okvira**



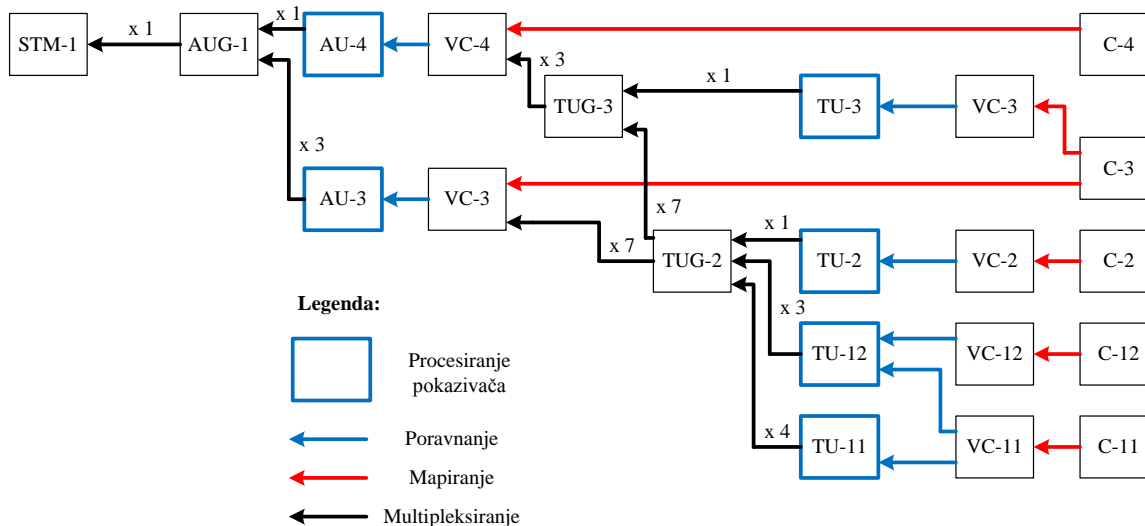
**Slika 2.4.4. Struktura RSOH i MSOH dela STM-64 okvira**

Kao što vidimo sa slike 2.4.2 struktura RSOH i MSOH dela STM-4 okvira je veoma slična strukturi RSOH i MSOH dela STM-1 okvira. Prvi red RSOH dela se ni ovde ne skrembluje iz identičnog razloga kao u slučaju STM-1 okvira. Bajtovi Z0 su rezervisani za buduću upotrebu na međunarodnom nivou. Sa M1 bajtom se može prijaviti do 96 pogrešnih blokova.

Struktura RSOH i MSOH dela STM-16 okvira je takođe veoma slična strukturi iz STM-1 okvira. Međutim, sada se javljaju i P1 i Q1 bajtovi koji se mogu koristiti u FEC metodi za korekciju grešaka, pri čemu je upotreba FEC metode opciona. Sa M1 bajtom se može prijaviti do 255 pogrešnih blokova.

Struktura RSOH i MSOH dela STM-64 okvira je takođe veoma slična strukturi iz STM-1 okvira. I ovde, kao u STM-16 strukturi, postoje P1 i Q1 bajtovi koji se mogu koristiti u FEC metodi za korekciju grešaka, pri čemu je upotreba FEC metode i ovde opciona. Sada se uz M1 bajt koristi i M0 bajt, čime je omogućeno prijavljivanje do 1636 pogrešnih blokova.

## 2.5. Ubacivanje korisničkih tokova u STM-1 okvir



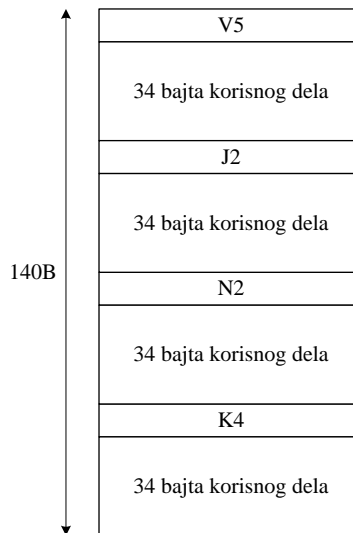
Slika 2.5.1. Princip ubacivanja korisničkih tokova u STM-1 okvir

Da bi se korisnički podaci mogli prenositi preko SDH mreže, neophodno je podatke prilagoditi za slanje preko STM-N okvira. U tu svrhu se definišu tzv. kontejneri u koje se stavljaju korisnički podaci. Međutim, potrebno je kontejnere prilagoditi za smeštanje u STM-N strukturu pa se vrše procesi mapiranja, poravnanja i multipleksiranja. Mapiranje podrazumeva smeštanje korisničkih informacija (kontejnera) u tzv. virtuelne kontejnere pri čemu se korisničkoj informaciji dodaje zaglavlje (*POH - Path Overhead*) i biti za poravnanje koji se koriste za postizanje kontrolisanih preskoka. Poravnanje podrazumeva dodavanje pokazivača virtuelnom kontejneru da bi se označio početak virtuelnog kontejnera pošto pozicija virtuelnog kontejnera može da varira (pluta). Multipleksiranje podrazumeva multipleksiranje sadržaja više korisnika za prenos preko istog STM-N okvira (nekoliko sporijih sadržaja se multipleksira u zajedničku bržu strukturu) ili prilagođenje sadržaja korisnika za prenos preko STM-N okvira (jedan korisnički sadržaj čiji je tok velike brzine se smešta u okvir). Na slici 2.5.1 je prikazano mapiranje korisničkih tokova u STM-1 okvir.

**Tabela 2.5.1 - Protoci virtuelnih kontejnera**

Tip virtuelnog kontejnera VC	Ukupni protok [Mb/s]	Predviđeni korisnički protok [Mb/s]	Protok korisnog dela [Mb/s]
VC-11	1.664	1.544	1.600
VC-12	2.240	2.048	2.176
VC-2	6.848	6.312	6.784
VC-3	48.960	34.368	48.384
VC-4	150.336	139.264	149.760

Razlikuju se tokovi nižeg protoka i tokovi višeg protoka. Kontejneri C-11, C-12 i C-2 su namenjeni korisnicima nižeg protoka, a kontejneri C-3 i C-4 korisnicima višeg protoka. Ovi kontejneri se mapiraju u odgovarajuće virtuelne kontejnere kao što se može videti sa slike 2.5.1. Prilikom mapiranja u virtuelni kontejner, kontejneru se dodaje POH zaglavlje. Ukupni protoci virtuelnih kontejnera i predviđeni protoci korisnog sadržaja virtuelnih kontejnera su prikazani u tabeli 2.5.1. Pod predviđenim korisničkim protokom je navedena vrednost korisničkog protoka za koji je dotični virtuelni kontejner namenjen sa stanovišta odgovarajućeg nivoa PDH hijerarhije. Pod protokom korisnog dela se podrazumeva protok virtuelnog kontejnera ne računajući protok POH zaglavlja. Ako se pogleda tabela 2.1.1, PDH nivo E1 bi koristio VC-12, E3 nivo VC-3, a E4 nivo bi koristio VC-4 virtuelne kontejnere (odnosno C-12, C-3 i C-4 kontejnere, respektivno). VC-11 kontejner je namenjen T1 nivou koji odgovara PDH standardu koji koriste SAD i Japan. Isto tako, VC-2 odgovara T2 nivou. VC-3 kontejner takođe odgovara T3 nivou (T3 nivo ima različite vrednosti u Japanu i SAD, za razliku od T1 i T2 nivoa koji su isti u SAD i Japanu). U tabeli 2.5.1 je za VC-3 kontejner navedena samo evropska vrednost (E3 nivo) za predviđeni korisnički protok. Naravno, pored PDH struktura i drugi tipovi podataka se mogu prenositi dotičnim kontejnerima (i koji mogu bolje iskoristiti korisni deo virtuelnog kontejnera - na primer, u slučaju VC-3 kontejnera, E3 struktura nema dobro iskorišćenje korisnog dela). U nastavku ćemo razmotriti strukture VC-12, VC-3 i VC-4 kontejnera pošto su oni prilagođeni evropskim PDH strukturama.



**Slika 2.5.2. VC-12 struktura**

Na slici 2.5.2 je prikazana struktura VC-12. Prikazana struktura VC-12 sadrži ukupno 140 bajtova, pri čemu traje 0.5ms, tj. proteže se na četiri periode STM-N okvira. Bajtovi V5, J2,

N2 i K4 čine POH zaglavlje. V5 bajt se koristi za proveru bitskih grešaka, signaliziranje grešaka na nivou sekcije puta i za označavanje prenošenog signala. Struktura V5 bajta je prikazana na slici 2.5.3.

2b	1b	1b	3b	1b
BIP-2	REI	RFI	Labela	RDI

**Slika 2.5.3. Struktura V5 bajta**

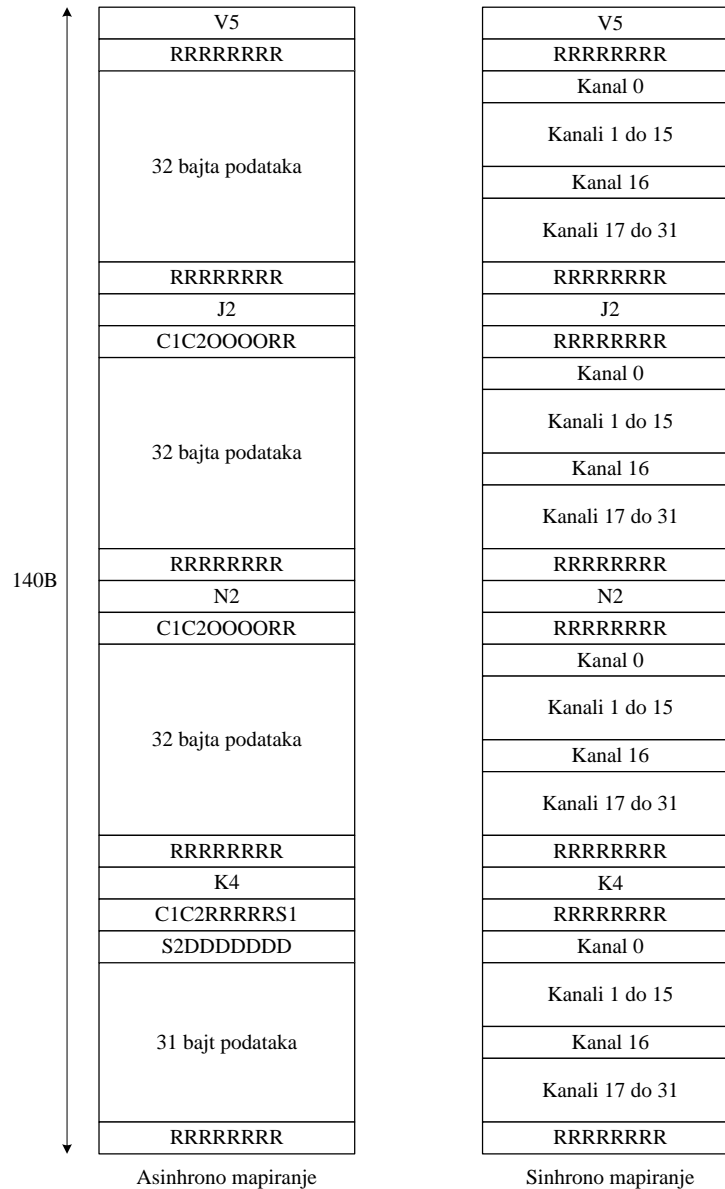
BIP-2 se koristi kao provera na parnost koja funkcioniše po istom principu kao u slučaju B1 i B2 bajtova iz RSOH, odnosno MSOH dela. U suštini jedan bit je provera na parnost svih bita na neparnim pozicijama prethodnog VC-12, a drugi je provera na parnost svih bita na parnim pozicijama prethodnog VC-12. REI bit predstavlja indikaciju da je na suprotnoj strani detektovana greška upotrebom BIP-2 bita (vrednost 0 signalizira da nije bilo grešaka, a vrednost 1 da je bar jedan VC-12 primljen na suprotnoj strani te veze sa bitskim greškama). RFI bit se ne koristi u slučaju VC-12 kontejnera. Labela ukazuje na tip mapiranja koji je korišćen prilikom ubacivanja korisničkog sadržaja u virtuelni kontejner (na primer, asinhron, sinhron na nivou bita, sinhron na nivou bajta). RDI bit se koristi za signalizaciju uočenog kvara na udaljenoj strani (kvarovi koji utiču na povezanost korisnika na nivou sekcije puta). J2 bajt sadrži identifikaciju predajne strane čime prijemna strana može neprestano da proverava da je veza sa predajnom stranom u toku, odnosno aktivna. Kao što vidimo, uloga je slična onoj koju ima J0 bajt iz RSOH dela, a razlika je u tome što se J2 odnosi na sekciju puta. Identifikator se šalje u vidu strukture od 16 bajtova koja sadrži identifikaciju po formatu definisanom u G.831 preporuci. N2 bajt se koristi za tzv TCM (*Tandem Connection Monitoring*) nadzor. U stvari, nadziranjem N2 bajta u okviru VC-12 veze se mogu dobiti informacije o statusu sekcije puta između krajnjih tačaka u SDH mreži, odnosno informacije o određenoj trasi u SDH mreži pošto sekcija puta u suštini odgovara jednoj trasi u SDH mreži. Posmatranjem više veza na sekciji puta i posmatranjem više sekcija puta na ovaj način, administrator može da prikupi važne informacije o funkcionisanju pojedinih delova mreže i brže uoči eventualne kvarove u mreži. U slučaju da tri bita koji odgovaraju labeli iz V5 bajta signaliziraju proširenu labelu, tada se proširena labela smešta K4 bajt. Proširena labela se smešta na bitsku poziciju 1 K4 bajta, tako što se preko tog bita šalje struktura dužine 32 bita prikazana na slici 2.5.4. MFAS (*Multi Frame Alignment Signal*) služi za detekciju početka strukture i vrednost ovog polja je 0111111110, a R predstavlja bite rezervisane za buduću upotrebu. Bit na poziciji 2 u K4 bajtu se koristi za proces virtuelne konkatanacije koja će biti objašnjena nešto kasnije. Biti na pozicijama 3 i 4 se mogu koristiti za signalizaciju za aktivaciju zaštite, slično kao što se koriste K1 i K2 bajtovi iz MSOH dela. Bit na poziciji 8 se može koristiti kao kanal za razmenu podataka na sekciji puta.

11b	8b	1b	12b
MFAS	Proširena Labela	0	R

**Slika 2.5.4. Proširena labela**

Na slici 2.5.5 je prikazana struktura VC-12 kontejnera u slučaju asinhronog i sinhronog mapiranja korisničkog toka protoka 2.048Mb/s. Sinhrono mapiranje se koristi za slučaj 2.048Mb/s toka strukturiranog po G.704 preporuci (E1 linkovi u telefonskim mrežama), a asinhrono za tokove protoka 2.048Mb/s koji su u skladu sa G.702 preporukom. R biti se mogu smatrati popunom da bi virtuelni kontejner imao potreban broj bita. D biti pripadaju korisničkim podacima u slučaju asinhronog mapiranja. U slučaju asinhronog mapiranja se koriste 2 bita za poravnanje (S1 i S2 biti), a C1 i C2 biti se koriste za signalizaciju da li se dotični biti (S1 i S2) koriste kao korisnički podaci ili ne. Ako C1 biti imaju vrednost 000 tada je S1 korisnički bit (bit

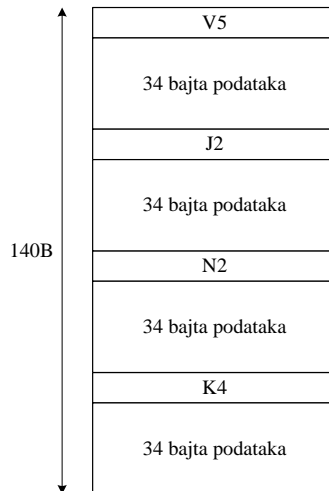
korisničkog podatka), a ako je vrednost C1 bita 111 tada se S1 bit koristi za poravnanje. Ista korelacija važi za C2 bite i S2 bit. Pri tome, ako se vrednost C1 bita razlikuje od 000, odnosno 111 koristi se majoritetno odlučivanje, tj. ako su bar dva C1 bita jednaka 1, smatra se vrednost 111, odnosno ako su bar dva C1 bita jednaka 0, uzima se vrednost 000. Isto važi i za C2 bite. O bitu su rezervisani za buduću upotrebu. U slučaju sinhronog mapiranja vidimo da je očuvana na nivou bajta struktura po kanalima tako da se ona ne mora rekonstruisati na strani korisnika koji se nalazi na prijemnoj strani.



**Slika 2.5.5. Struktura VC-12 kontejnera**

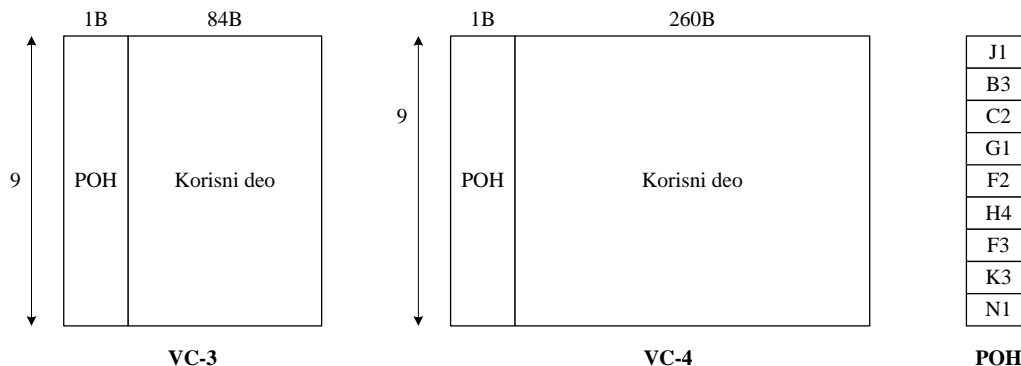
Na slici 2.5.6 je prikazana struktura VC-12 za slučaj kada se on koristi za prenos ATM ćelija. Kao što vidimo, kompletan korisni deo VC-12 se koristi za smeštanje ATM ćelija. Pri tome se vrši poravnanje na nivou bajta, u smislu da ne može bajt jedne ATM ćelije da se preseče tako da deo bajta bude u jednom VC-12, a drugi deo u narednom VC-12 virtuelnom kontejneru.





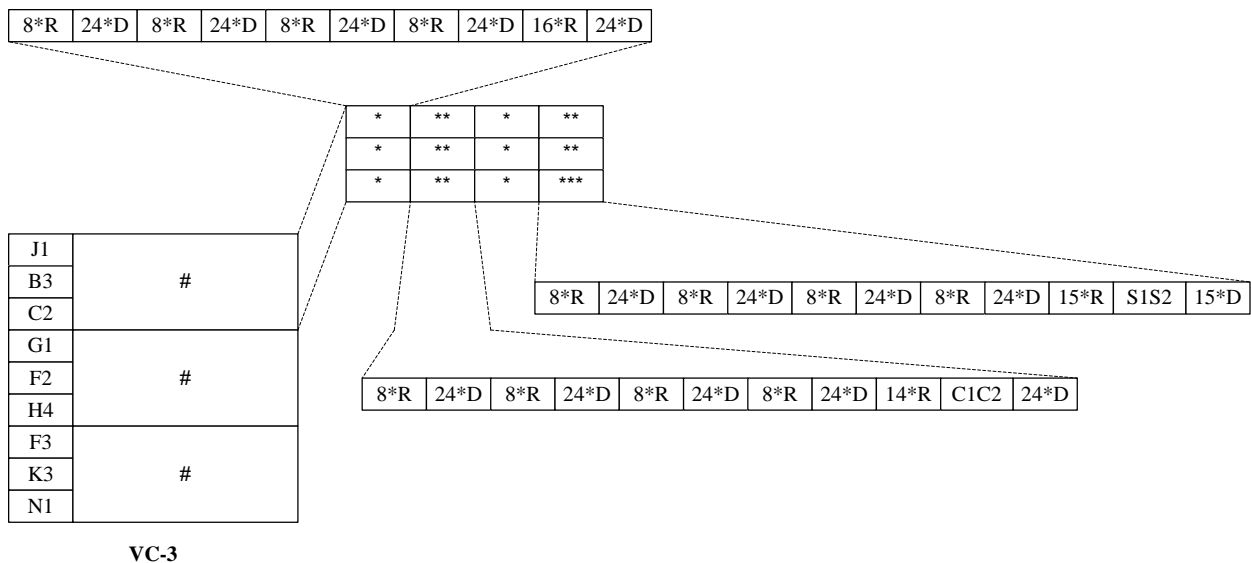
Slika 2.5.6. VC-12 struktura kada se prenose ATM ćelije

Na slici 2.5.7 su prikazane strukture VC-3 i VC-4 virtuelnih kontejnera. Trajanje jedne VC-3, odnosno jedne VC-4 strukture je 125 $\mu$ s.



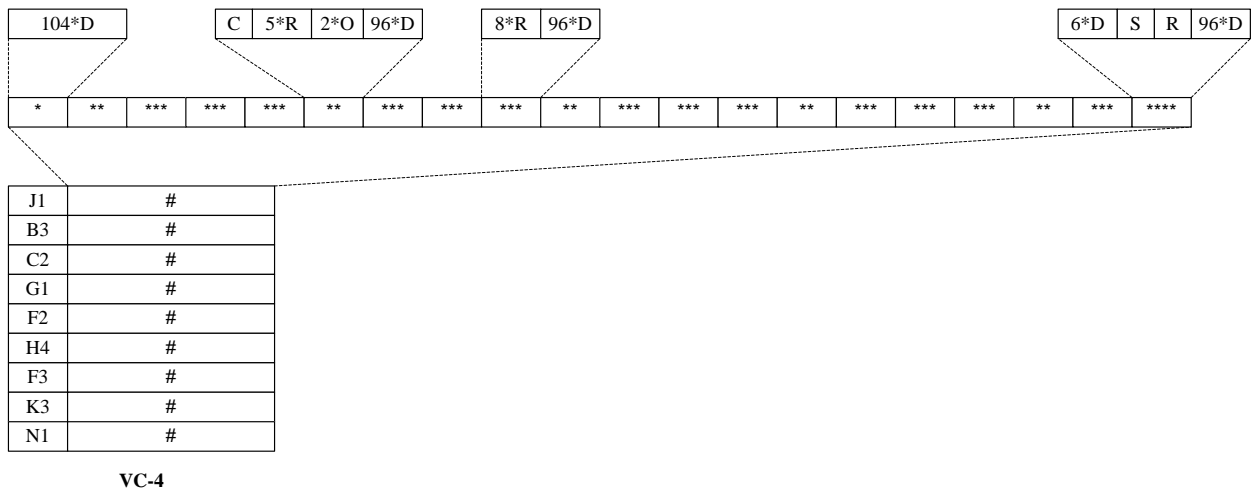
Slika 2.5.7. VC-3 i VC-4 strukture

U obe strukture, POH deo se sastoji od devet bajtova, tj. od J1, B3, C2, G1, F2, H4, F3, K3 i N1 bajtova, kao što je prikazano na slici 2.5.7. J1 ima identičnu ulogu i format kao J2 bajt iz POH dela VC-12 virtuelnog kontejnera. B3 bajt ima identičnu ulogu kao BIP-2 biti iz POH dela VC-12 virtuelnog kontejnera, odnosno B1 i B2 bajtova iz RSOH i MSOH dela, pri čemu se provera parnosti vrši na osam pozicija bajtova iz prethodnog VC-3, odnosno VC-4 virtuelnog kontejnera. C2 ima identičnu ulogu kao polje labela iz V5 bajta POH dela VC-12 virtuelnog kontejnera. G1 bajt sadrži REI i RDI indikacije koje imaju identičnu ulogu onima iz V5 bajta POH dela VC-12 virtuelnog kontejnera, pri čemu sada REI omogućava signaliziranje broja grešaka u opsegu 0 do 8. F2 i F3 bajtovi omogućavaju korisničku komunikaciju između uređaja na krajevima sekcije puta, pri čemu komunikacija zavisi od tipa sadržaja koji se prenosi u VC-3, odnosno VC-4 virtuelnom kontejneru. H4 bajt se koristi u slučaju upotrebe virtuelne konkatanacije ili kao indikator pri čemu tumačenje indikacije zavisi od tipa korisnog sadržaja koji se prenosi. Donja polovina K3 bajta se koristi za signalizaciju za aktivaciju zaštite, slično kao što se koriste K1 i K2 bajtovi iz MSOH dela. Najviša dva bita K3 bajta se mogu koristiti kao kanal za razmenu podataka na sekciji puta. N1 bajt se koristi za tzv TCM (*Tandem Connection Monitoring*) nadzor, odnosno ima istu ulogu kao N2 bajt iz POH dela VC-12 virtuelnog kontejnera.



**Slika 2.5.8. Asinhrono mapiranje 34.368Mb/s toka po G.702 preporuci u VC-3**

Na slici 2.5.8 je prikazano asinhrono mapiranje toka protoka 38.368Mb/s koji je u skladu sa G.702 preporukom u VC-3 virtualni kontejner. Kao što vidimo korisni deo je podeljen na tri identična dela. R biti se koriste kao popuna, a uloga C1, C2, S1 i S2 bita je ista kao kod asinhronog mapiranja G.702 toka od 2.048Mb/s u VC-12 strukturu. Razlika je da se sada za kontrolu Si bita koristi 5 Ci bita, pri čemu se i dalje koristi majoritetno odlučivanje u slučaju da usled bitskih grešaka vrednost Ci bita bude različita od 00000 ili 11111. D biti predstavljaju korisničke bite.



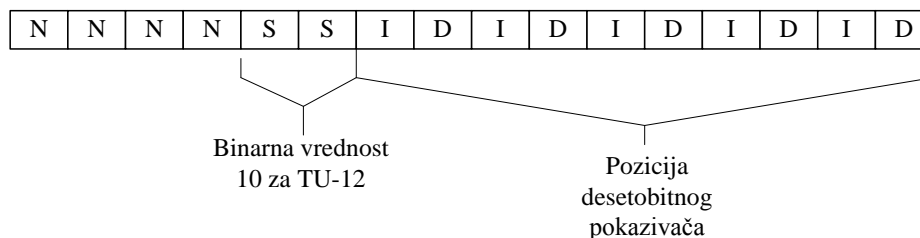
**Slika 2.5.9. Asinhrono mapiranje 139.264Mb/s toka po G.702 preporuci u VC-4**

Na slici 2.5.9 je prikazano asinhrono mapiranje toka protoka 139.264Mb/s koji je u skladu sa G.702 preporukom u VC-4 virtualni kontejner. Kao što vidimo, korisni deo je podeljen na 9 identičnih delova. R biti se koriste kao popuna, a uloga C i S bita je ista kao kod asinhronog mapiranja G.702 toka od 2.048Mb/s u VC-12 strukturu. Razlika je da se sada za kontrolu S bita koristi 5 C bita, pri čemu se i dalje koristi majoritetno odlučivanje u slučaju da usled bitskih grešaka vrednost C bita bude različita od 00000 ili 11111. D biti predstavljaju korisničke bite. O biti su rezervisani za buduću upotrebu.

Mapiranje ATM toka u VC-3, odnosno VC-4 je slično mapiranju ATM toka u VC-12 virtualni kontejner, pošto se ćelije mapiraju u kompletan korisni deo VC-3, odnosno VC-4, pri čemu se takođe vrši poravnanje na nivou bajta.

Virtuelni kontejneri VC-12 i VC-3 nisu dovoljno veliki da popune kompletan korisni deo STM-1 strukture, pa se stoga oni pakuju u odgovarajuće TU (*Tributary Unit*) strukture (TU-12 i TU-3, respektivno), pa se potom više TU struktura multipleksira u TUG (*Tributary Unit Group*) grupu (tri TU-12 u jednu TUG-2, odnosno jedna TU-3 u jednu TUG-3). TU jedinice i grupisanje u TUG grupe omogućava fleksibilno deljenje korisnog dela STM-1 okvira korisničkim tokovima, tako da korisni deo STM-1 okvira bude što više popunjen korisničkim tokovima, a samim tim da se postigne što bolje iskorišćenje linkova u SDH mreži koji prenose odgovarajuće STM-1 okvire (tj. STM-N okvire u opštem slučaju).

Kao što se vidi sa slike 2.5.7, korisni deo VC-3 iznosi  $84 \cdot 9 = 756$  bajtova. Takođe, VC-12 struktura ima periodu od 4 STM-1 okvira, a samim tim i od 4 VC-3 virtualna kontejnera. Ukupna količina bajtova u korisnom delu četiri VC-3 virtualna kontejnera iznosi  $756 \cdot 4 = 3024$  bajtova. Sa slike 2.5.1 vidimo da se 21 VC-12 virtualni kontejner pakuje u korisni deo VC-3 kontejnera. Ukupna količina bajtova svih VC-12 kontejnera je stoga  $140 \cdot 21 = 2940$  bajtova. Kao što vidimo ostaje ukupno 84 bajta na raspolaganju, koji se koriste kao pokazivači na početak VC-12 strukture tj. na V5 bajt. Svakom korisničkom toku, tj. VC-12 virtualnom kontejneru je dodeljeno četiri bajta (V1, V2, V3 i V4) koji sačinjavaju TU-12 pokazivač dotičnog virtualnog kontejnera VC-12, tako da sa ta četiri bajta, VC-12 postaje TU-12 jedinica. V1 i V2 bajt pokazuju na lokaciju V5 bajta odgovarajućeg VC-12 virtualnog kontejnera. Združena struktura V1 i V2 bajtova je prikazana na slici 2.5.10.



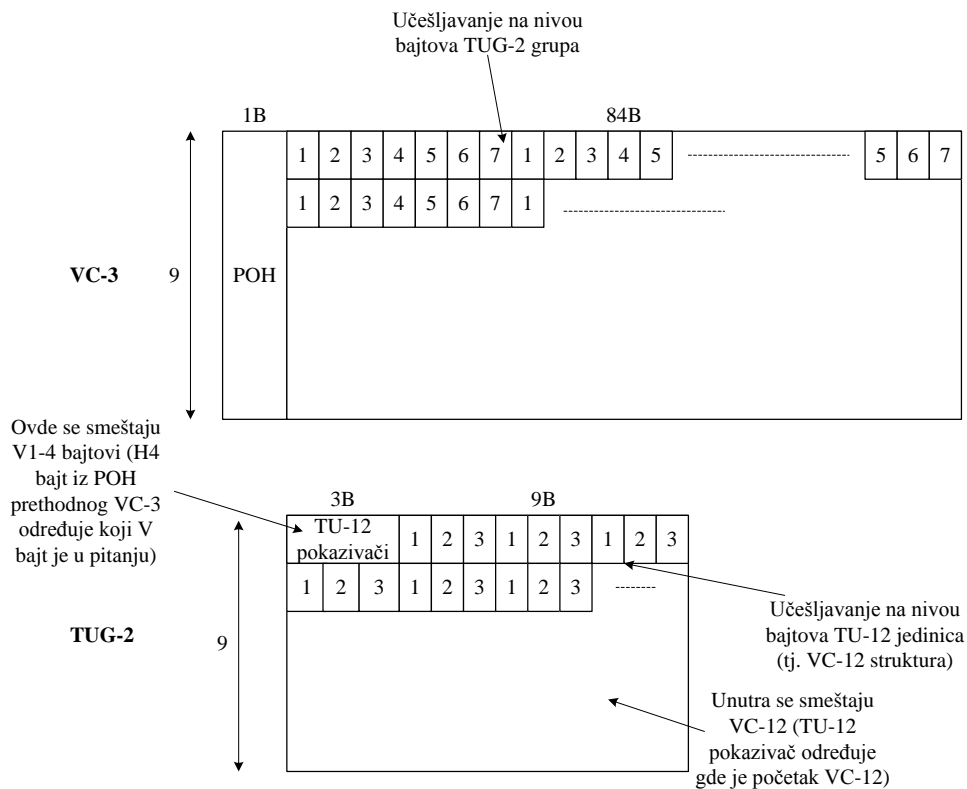
**Slika 2.5.10. Združena struktura V1 i V2 bajtova**

V1 i V2 bajtovi sadrže desetobitni pokazivač čija validna vrednost može da se kreće u granicama od 0 do 139. Vrednost pokazivača ukazuje na ofset početka VC-12 u odnosu na poziciju V2 bajta, pri čemu se ne računavaju bajtovi TU-12 pokazivača, već samo bajtovi koji pripadaju dotičnom VC-12 virtualnom kontejneru. Pošto se koriste četiri VC-3 strukture, neophodno je nekako raspoznati raspored V1-4 bajtova po strukturama, tj. koji bajt se nalazi u kojoj VC-3 strukturi. U tu svrhu se koristi H4 bajt iz POH zaglavlja. Biti na pozicijama 7 i 8 daju indicaciju koji od V1-4 bajtova se javlja u narednoj VC-3 strukturi. Ako biti na pozicijama 7 i 8, respektivno imaju vrednost 00 tada se u narednoj VC-3 strukturi nalazi V1 bajt, vrednost 01 signalizira da se u narednoj VC-3 strukturi nalazi V2 bajt, vrednost 10 signalizira da se u narednoj VC-3 strukturi nalazi V3 bajt i vrednost 11 signalizira da se u narednoj VC-3 strukturi nalazi V4 bajt.

Pokazivač mora da zadrži svoju vrednost u tri ciklusa tumačenja da bi se izbeglo eventualno pogrešno tumačenje pokazivača. Pošto korisnički tokovi mogu da pristižu i nešto brže od očekivane vrednosti i nešto sporije od očekivane vrednosti neophodno je periodično vršiti poravnavanja. Pod očekivanom vrednosti pristizanja se misli na protok VC-12 virtualnog

kontejnera. Invertovanjem I bita (tzv. inkrement biti) vrši se pozitivno poravnanje i vrednost desetobitnog pokazivača će biti inkrementirana za jedan (ova vrednost će biti korišćena u narednim iteracijama na poziciji desetobitnog pokazivača). Pozitivno poravnanje podrazumeva ubacivanje 'praznog' bajta na mesto prvog bajta dotične VC-12 strukture koji nailazi iza V3 bajta čime se izaziva klizanje VC-12 kontejnera za jedno mesto unapred. Pozitivno poravnanje se koristi kada je korisnički tok nešto sporiji od očekivane vrednosti. Invertovanjem D bita (tzv. dekrement biti) vrši se negativno poravnanje i vrednost desetobitnog pokazivača će biti dekrementirana za jedan (ova vrednost će biti korišćena u narednim iteracijama na poziciji desetobitnog pokazivača). Negativno poravnanje podrazumeva da se u V3 bajt stavi korisnički bajt čime se izaziva klizanje VC-12 kontejnera za jedno mesto unazad. Negativno poravnanje se koristi kada je korisnički tok nešto brži od očekivane vrednosti. Nakon promene vrednosti, pokazivač mora da zadrži tu vrednost u minimalno naredna tri ciklusa (promena invertovanjem I ili D bita se šalje samo jednom i nakon toga se šalje promenjena vrednost pokazivača - uvećana ili umanjena stara vrednost pokazivača).

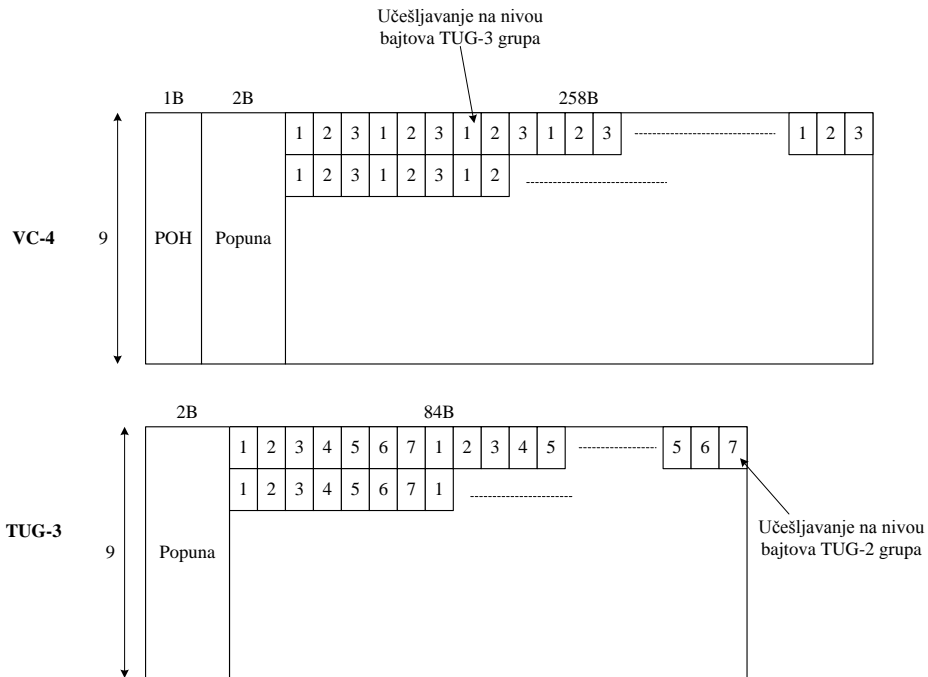
N biti se koriste za signaliziranje novog korisničkog toka tj. sada dotični VC-12 koristi novi tok, čime se omogućava proizvoljno podešavanje vrednosti pokazivača, pošto tek sad počinju da pristižu korisnički podaci, odnosno tek sad VC-12 struktura ponovo počinje da nosi korisničke podatke.



Slika 2.5.11. Multipleksiranje TUG-2 grupa u VC-3

Tri TU-12 jedinice se multipleksiraju u TUG-2 grupu na nivou bajtova. 7 TUG-2 grupa se multipleksira takođe na nivou bajtova u VC-3 virtuelni kontejner. Ova dva procesa su prikazana na slici 2.5.11. Kao što se vidi sa slike, po 12 kolona u VC-3 virtuelnom kontejneru, odgovara svakoj TUG-2 grupi. Isto tako, po četiri kolone u TUG-2 grupi odgovara svakoj u TU-12 jedinica, a može se reći i VC-12 struktura koje zauzimaju u najveći deo strukture TU-12

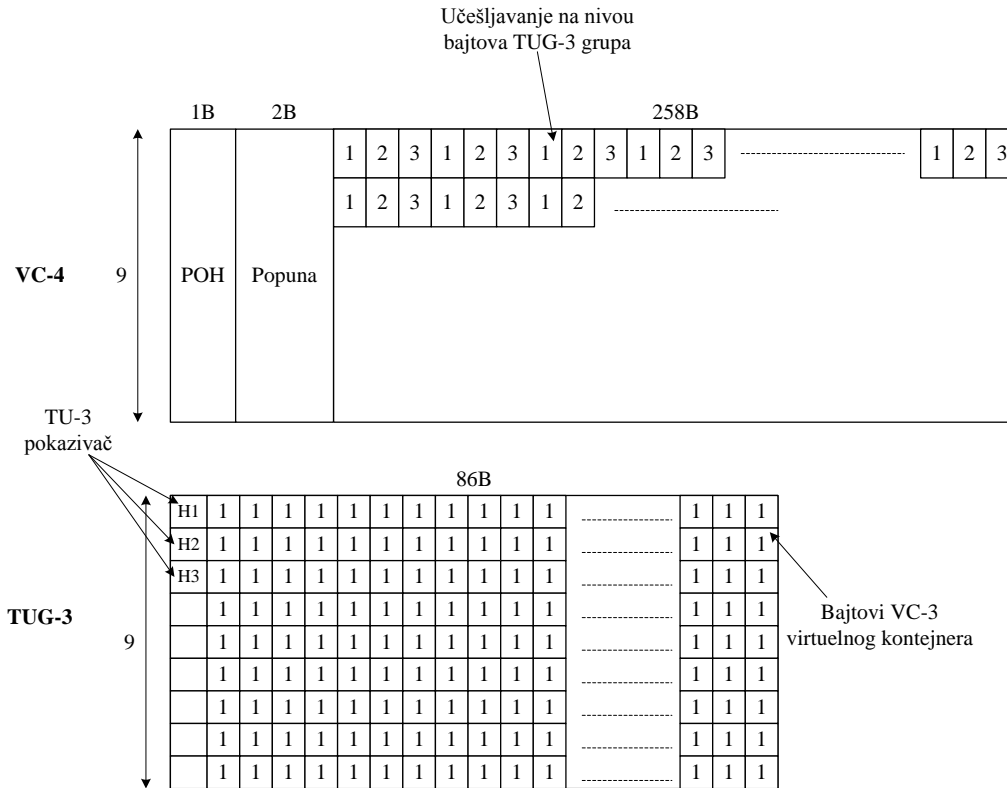
jedinica. Ovakvo strukturiranje gde jedna čitava kolona praktično pripada jednom korisničkom toku omogućava lako izvlačenje i ubacivanje tokova jer je lako odrediti bajtove koji pripadaju istom toku (praktično treba samo identifikovati kolone od interesa).



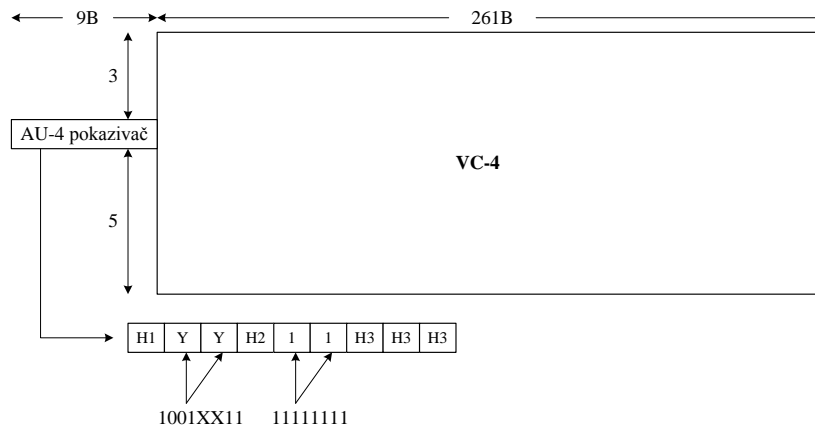
**Slika 2.5.12. Multipleksiranje TUG-2 grupa u TUG-3 grupe i potom TUG-3 grupa u VC-4**

7 TUG-2 grupa se može multipleksirati i u TUG-3 grupu što je prikazano na slici 2.5.12. 3 TUG-3 grupe se multipleksiraju u VC-4 virtuelni kontejner što je takođe prikazano na slici 2.5.12. Oba multipleksiranja se rade na nivou bajtova, pa se i dalje postiže da jedna kolona pripada jednom korisničkom toku. Popuna predstavlja bajtove koji se ne koriste, ali je neophodno njihovo prisustvo za postizanje STM-1 strukture. H4 bajt se i dalje koristi za identifikaciju koji od V1-4 bajtova se pojavljuje u TU-12 jedinicama u narednom VC-4 virtuelnom kontejneru.

Virtuelni kontejner VC-3 se može upakovati u TU-3 jedinicu koja se potom multipleksira u TUG-3 grupu (samo jedna TU-3 jedinica se multipleksira u TUG-3 grupu). Tri TUG-3 grupe se multipleksiraju u VC-4 kontejner, kao i u slučaju kada je TUG-3 grupa sadržavala TUG-2 grupe. TU-3 pokazivači se koriste kao pokazivači na početak VC-3 strukture. TU-3 pokazivač na jednu VC-3 strukturu se sastoji od tri bajta H1, H2 i H3. H1 i H2 bajtovi predstavljaju pokazivač na početak VC-3 strukture i posmatraju se združeno. Format združene strukture je identičan onome prikazanom na slici 2.5.10, pri čemu je vrednost dva S bita sada 10. Upotreba desetobitnog pokazivača, I i D bita, kao i N bita je ista kao i kod TU-12 jedinice. Razlika je u tome što se sada vrednost pokazivača može kretati u granicama 0 do 764. Očigledno, H1 i H2 bajt imaju ulogu V1 i V2 bajta iz TU-12 jedinice. H3 bajt ima ulogu V3 bajta iz TU-12 strukture. Slika 2.5.13 ilustruje proces kreiranja VC-4 virtuelnog kontejnera od tri VC-3 virtuelna kontejnera.



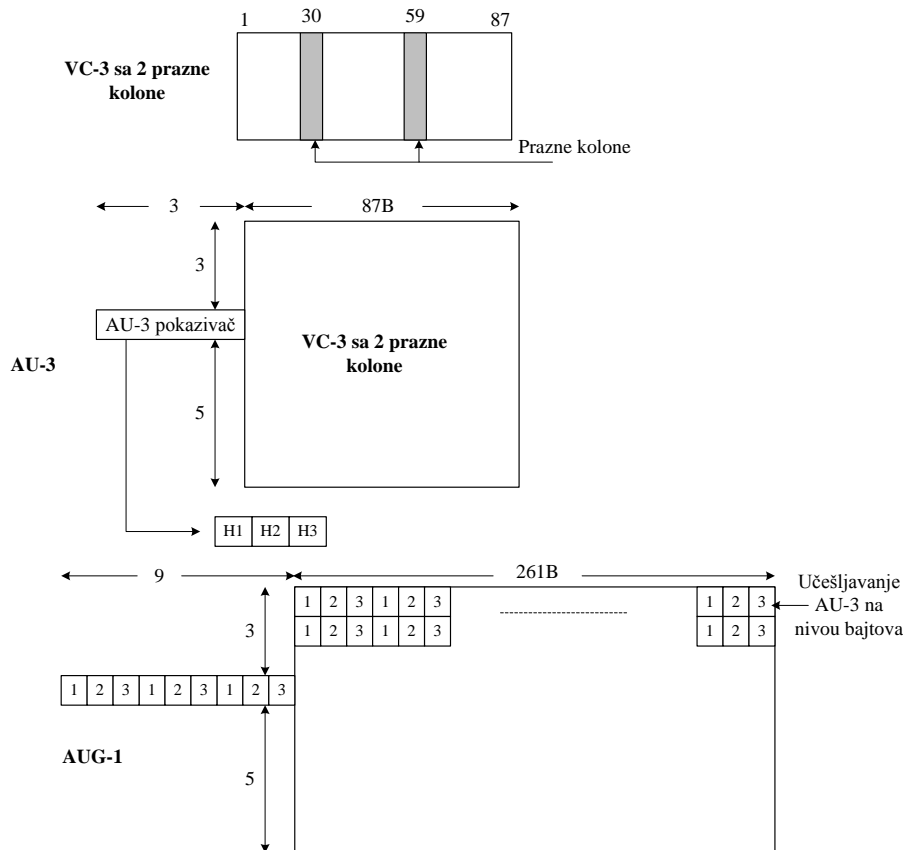
**Slika 2.5.13. Multipleksiranje TU-3 jedinice u TUG-3 grupu i potom TUG-3 grupa u VC-4**



**Slika 2.5.14. Kreiranje AUG-1 grupe od AU-4 (VC-4)**

VC-3 virtuelni kontejneri se mogu pakovati u AU-3 (*AU - Administrative Unit*) jedinicu, a VC-4 u AU-4 jedinicu. Tada se dodaju AU pokazivači. Tri AU-3 jedinice se multipleksiraju u jednu AUG-1 grupu (*AUG - Administrative Unit Group*), odnosno jedna AU-4 jedinica se multipleksira u AUG-1 grupu. AUG-1 grupa se potom multipleksira u STM-1 strukturu. AU pokazivači se smeštaju u polje pokazivač(i) STM-1 okvira. Ukoliko je u pitanju AU-4 jedinica, tada postoji samo jedan AU pokazivač koji pokazuje na početak VC-4 strukture, odnosno ako je u pitanju AU-3 jedinica tada postoje tri AU pokazivača, gde svaki pokazuje na po jednu VC-3 strukturu. Na slici 2.5.14 je prikazano kreiranje AUG-1 grupe polazeći od VC-4 virtuelnog kontejnera. Na slici 2.5.15 je prikazano kreiranje AUG-1 grupe polazeći od VC-3 virtuelnog kontejnera. Kada se VC-3 pakuju u AU-3 jedinicu, pre toga im se dodaju dve prazne kolone,

neophodne da bi se STM-1 struktura mogla popuniti do kraja. Tri AU-3 jedinice se multipleksiraju na nivou bajtova u AUG-1 grupu.



Slika 2.5.15. Kreiranje AUG-1 grupe od AU-3 (VC-3)

H1 i H2 bajtovi imaju ulogu pokazivača. Oni imaju identičnu združenu strukturu kao što je prikazana na slici 2.5.10, pri čemu je vrednost dva S bita sada 10. Upotreba desetobitnog pokazivača, I i D bita, kao i N bita je ista kao i kod TU-12 jedinice. Razlika je u tome što se sada vrednost pokazivača može kretati u granicama 0 do 782. Očigledno, H1 i H2 bajt imaju ulogu V1 i V2 bajta iz TU-12 jedinice. H3 bajt ima ulogu V3 bajta iz TU-12 strukture. Pri tome, u slučaju AU-4 jedinice, tj. kada se AU-4 jedinica multipleksira u AUG-1 grupu, sva tri H3 bajta se koriste za poravnanje, tj. poravnanje se vrši na nivou trojke bajtova, a ne na nivou jednog bajta. Princip poravnanja je isti kao u slučaju V3 bajta. U slučaju AU-3 jedinice poravnanje se vrši na nivou bajta, a svaki H3 bajt odgovara jednoj AU-3 jedinici.

Dodavanjem RSOH i MSOH polja kreira se STM-1 okvir iz AUG-1 grupe.

## 2.6. Mehanizmi automatskog oporavka

SDH je transportna mreža namenjena transportu korisničkih signala između telekomunikacionih mreža, kao i korisnika koji koriste usluge SDH mreže za povezivanje svojih lokacija, odnosno uređaja. Stoga je veoma važno ostvariti visoku pouzdanost SDH mreže, pa je veoma važno uvesti mehanizme koji nakon što se detektuje ispad nekog dela SDH mreže mogu automatski da preusmere saobraćaj i nastave sa njegovim regularnim transportom. SDH uređaji se prave da budu visoko pouzdani i raspoloživi, pa je najčešći ispad u SDH mreži ispad linka

usled njegovog prekida, i eventualno ispadi predajnika i prijemnika na fizičkom sloju na krajevima linkova. U SDH mreže je uvedena APS (*Automatic Protection Switching*) zaštita koja omogućava aktivaciju zaštitnog puta u slučaju detekcije kvara nekog linka ili uređaja u SDH mreži. Važno je naglasiti da se preusmeravanje na zaštitni put mora izvršiti u roku od 50ms. K1 i K2 bajtovi se koriste za aktivaciju APS zaštite, odnosno za iniciranje preusmeravanja saobraćaja na zaštitni put.

Razlikuju se linearna zaštita koja se izvodi u tačka-tačka vezi i prsten zaštita koja se sprovođi u mrežama baziranim na prsten topologiji.

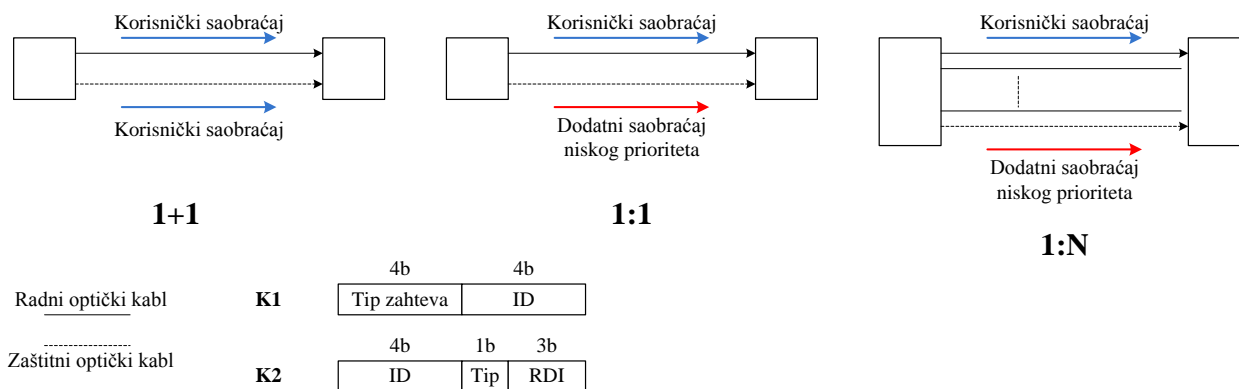
Linearna zaštita obuhvata sledeće varijante:

- 1+1 zaštita
- 1:1 zaštita
- 1:N zaštita

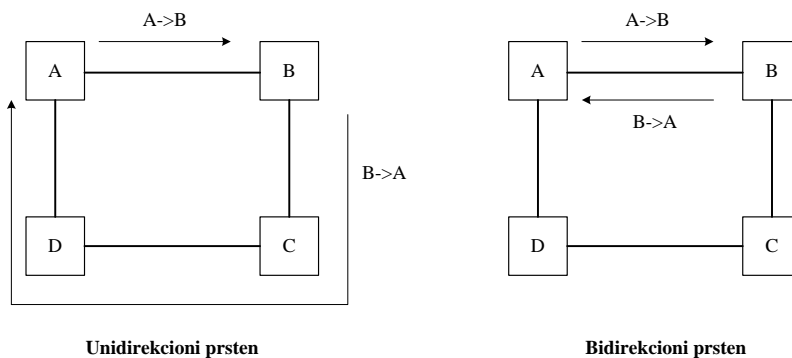
1+1 zaštita podrazumeva da se korisnički saobraćaj istovremeno šalje radnim i zaštitnim putem. Pod radnim putem se podrazumeva originalni put korisničkog saobraćaja, a pod zaštitnim putem se podrazumeva rezervni put za slučaj prekida originalnog puta. Kao što je navedeno, u 1+1 zaštiti se korisnički saobraćaj istovremeno šalje i radnim i zaštitnim putem, a na prijemu se selektuje onaj put koji ostvaruje bolji kvalitet signala na prijemnoj strani. Prednost ove metode je što nije potrebna signalizacija za aktivaciju preusmeravanja saobraćaja na zaštitni put, a takođe reakcija na otkaz radnog puta je veoma brza. Mana je što su resursi mreže predimenzionirani tj. resursi mreže se koriste svega 50% u slučaju da je sav korisnički saobraćaj zaštićen 1+1 zaštitom. Takođe, mana je što se u krajnjim tačkama mora meriti signal sa oba puta da bi se izvršila selekcija koji signal će se koristiti sa radnog ili zaštitnog puta i to se mora raditi za sve signale. U slučaju 1:1 zaštite, korisnički saobraćaj se šalje samo radnim putem, dok se zaštitni put može koristiti za transport saobraćaja niskog prioriteta. U slučaju otkaza radnog puta prijemna strana pomoću K1 bajta obaveštava predajnu stranu da pređe na zaštitni put. U tom momentu se više ne prenosi saobraćaj niskog prioriteta zaštitnim putem, već korisnički saobraćaj preusmeren sa radnog puta. Prednost 1:1 metode je bolje iskorišćenje mrežnih resursa, a mana je potreba za upotrebom signalizacije i sporija reakcija u odnosu na 1+1 metodu. Takođe, prednost je i to što nema potrebe za neprestanim praćenjem kvaliteta signala u krajnjim tačkama radi selekcije kao kod 1+1 varijante. 1:N metoda dodatno poboljšava iskorišćenje mrežnih resursa u odnosu na 1:1 metodu. Naime, sada se zaštitni put koristi kao zaštita N radnih puteva. Ako jedan od tih radnih puteva otkáže, tada se preko K1 bajta signalizira prekid, a unutar dotičnog bajta se navodi i koji od radnih puteva je u pitanju (maksimalno 14 radnih puteva može biti obuhvaćeno 1:N zaštitom). Tada predajna strana preusmerava saobraćaj dotičnog radnog puta na zaštitni put. I kod 1:N zaštite se zaštitni put može koristiti za prenos niskoprioritetnog saobraćaja sve dok ne dođe do aktivacije zaštitnog puta za prenos korisničkog saobraćaja sa nekog od radnih puteva. Mana 1:N zaštite je što u slučaju otkaza više od jednog radnog puta, ne može da se zaštiti kompletan korisnički saobraćaj pomoću zaštitnog puta. 1:1 zaštita je u suštini specijalan slučaj 1:N zaštite. K2 bajtom predajna strana signalizira da li koristi 1+1 ili 1:N zaštitu (1:1 zaštita se po standardu posmatra kao 1:N zaštita). Takođe, K2 bajtom signalizira šta se šalje preko zaštitnog puta (korisnički saobraćaj nekog od radnih puteva, pri čemu se navodi identifikacija tog radnog puta ili dodatni (niskoprioritetni) saobraćaj ili *null* saobraćaj kada se ništa ne prenosi preko zaštitnog puta). Struktura K1 i K2 bajtova, kao i opisani principi varijanata linearnih zaštita su prikazani na slici 2.6.1. Tipično se linearna zaštita vrši na nivou optičkog kabla tj.



fizičkog sloja multipleksne sekcije (jer regeneratori nemaju mogućnost preusmeravanja saobraćaja) kao što je prikazano na slici 2.6.1, ali se može vršiti i na nivou virtuelnih kontejnera ili STM-N okvira u okviru jedne multipleksne sekcije ili sekcije puta. Kao što se vidi sa slike, K1 bajt se sastoji od dva dela. Tip zahteva označava signalizaciju stanja prijemne strane i takođe omogućava prijemnoj strani da tim putem zahteva od predajne strane da aktivira zaštitni put. U drugom delu se nalazi identifikacija toka na koji se odnosi zahtev iz prvog dela K1 bajta. Vrednost 1-14 označavaju redne brojeve tokova, pri čemu se u slučaju 1+1 zaštite može koristiti samo vrednost 1. Vrednost 15 označava dodatni saobraćaj koji se prenosi preko zaštitnog puta (kada se koristi 1:N zaštita - u koju spada i 1:1). Vrednost 0 označava da se tip zahteva ne odnosi na konkretan tok. ID polje u K2 bajtu predstavlja identifikaciju toka koji se prenosi zaštitnim putem (vrednosti ID su iste kao i kod ID polja iz K1 bajta). I ovde u slučaju 1+1 zaštite se koristi samo vrednost 1. Polje tip predstavlja signalizaciju koju zaštitu koristi predajna strana (1 - 1:N zaštita, 0 - 1+1 zaštita). RDI polje predstavlja signalizaciju kvara detektovanog na prijemnoj strani i nije vezano za linearnu zaštitu.



Slika 2.6.1. Linearna zaštita



Slika 2.6.2. Princip rada unidirekcionog i bidirekcionog prstena

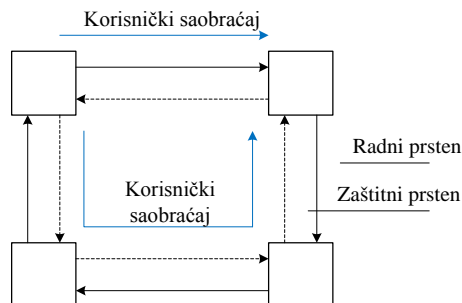
Prsten strukture nude efikasnu zaštitu protiv prekida u mreži jer sve dok se prsten ne prekine u dve tačke mreža ostaje potpuno dostupna u smislu da svi korisnici i dalje mogu međusobno da komuniciraju. Razlikuju se unidirekcionni i bidirekcionni prstenovi. Kod unidirekcionnih prstenova, smer saobraćaja po prstenu je uvek u istom smeru, a kod bidirekcionnih prstenova, smer saobraćaja se bira tako da ide najkraćom putanjom. Princip rada unidirekcionnih i bidirekcionnih prstenova je prikazan na slici 2.6.2. Unidirekcionni prstenovi su jednostavniji za rutiranje saobraćaja, ali je mana što je iskorišćenje slabije, a takođe i kašnjenje je za neke signale nepotrebno veliko usled dužeg puta (naročito problem za dvosmernu komunikaciju između susednih čvorova gde jedan korisnički tok ide najkraćom putanjom, a drugi korisnički tok ide

najdužom putanjom), dok su bidirekcionni prstenovi efikasniji sa stanovišta iskorišćenja i kapaciteta, ali nešto složeniji zbog upotrebe principa rutiranja da bi korisnički saobraćaj išao najkraćom putanjom.

U praksi se mogu koristiti prstenovi od dva ili četiri optička kabla. Imajući u vidu da prstenovi mogu biti unidirekcionni i bidirekcionni, postoje teoretski ukupno četiri moguće kombinacije:

- Unidirekcionni prstenovi sa dva optička kabla
- Bidirekcionni prstenovi sa dva optička kabla
- Unidirekcionni prstenovi sa četiri optička kabla
- Bidirekcionni prstenovi sa četiri optička kabla

U praksi se koriste unidirekcionni prstenovi sa dva optička kabla, i bidirekcionni prstenovi sa dva ili četiri optička kabla. Pri tome, se prstenovi sa dva optička kabla tipično koriste za manje prstenove, a prstenovi sa četiri optička kabla se tipično koriste za veće prstenove koji pokrivaju velika prostranstva.

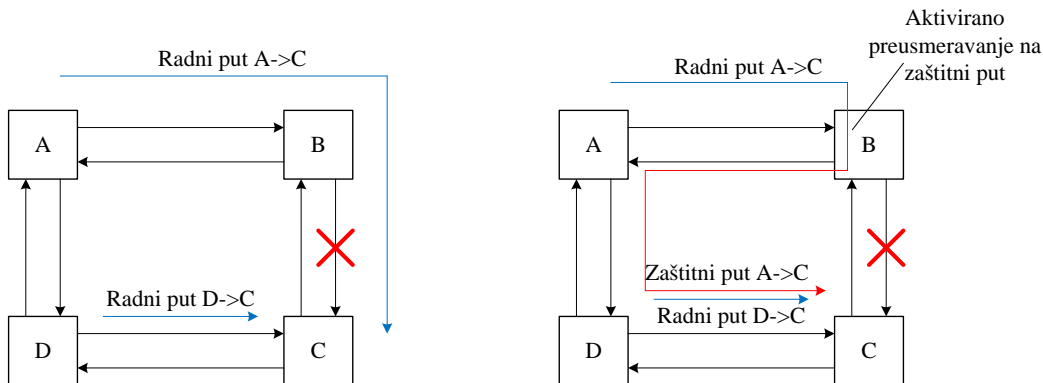


**Slika 2.6.3. Unidirekcionni prsten sa dva optička kabla**

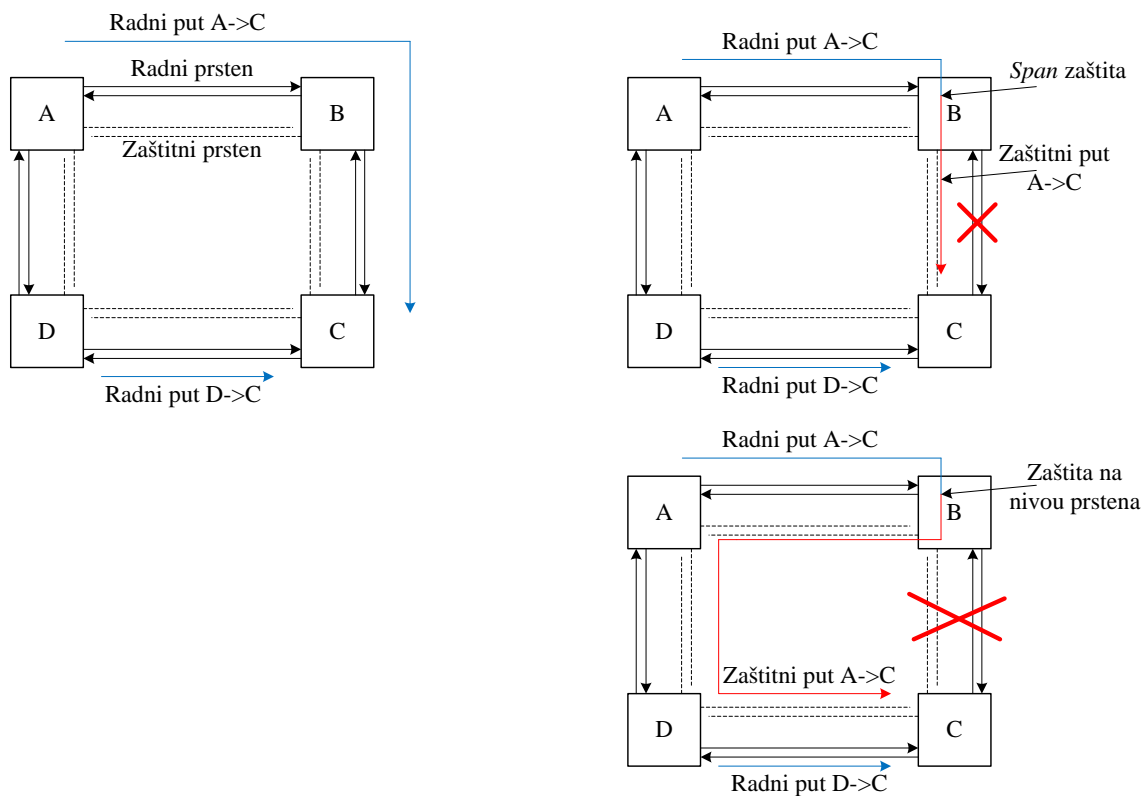
Unidirekcionni prsten sa dva optička kabla u suštini se sastoji od dva prstena, radnog i zaštitnog, pri čemu oni imaju suprotnu orijentaciju. Razlog suprotne orijentacije je što se optički kablovi zaštitnog i radnog prstena, ipak, nalaze jedan blizu drugog i često se dešava da se oba kabla preseku istovremeno (ako naravno dođe do presecanja). Otuda je bitno da postoji suprotan smer po kome bi se moglo zaobići mesto kvara i time i dalje ostvariti potpunu dostupnost u prstenu. Saobraćaj se istovremeno šalje i radnim i zaštitnim putem, a na prijemu se selektuje signal većeg kvaliteta. Očigledno, koristi se 1+1 princip samo na nivou prstena. Sve prednosti i mane navedene za 1+1 princip važe i u ovom slučaju. U slučaju prekida na jednom mestu jednog prstena ili ako se na istom mestu oba prstena prekinu, i dalje će prsten normalno funkcionisati sa stanovišta da će sav korisnički saobraćaj i dalje moći da se transportuje. Isto važi i za ispad jednog čvora, sem što će tada biti pogođeni korisnički tokovi koji počinju ili završavaju u dotičnom čvoru. Princip ove zaštite je prikazan na slici 2.6.3.

Bidirekcionni prsten sa dva optička kabla se takođe sastoji od dva prstena suprotnih orijentacija kao i kod unidirekcionnog prstena. Razlika je u tome što se sada radni saobraćaj šalje i jednim i drugim prstenom u zavisnosti od toga na kom prstenu se ostvaruje najkraći put. Otuda je polovina kapaciteta svakog prstena namenjena radnom saobraćaju, a polovina zaštitnom saobraćaju. Pri tome, resursi namenjeni zaštitnom saobraćaju se mogu koristiti za prenos saobraćaja niskog prioriteta sve dok se ne aktivira zaštita, odnosno dok ti resursi ne počnu da prenose saobraćaj sa radnog puta. Očigledno, koristi se 1:1 princip samo na nivou prstena, pa sve

mane i prednosti navedene za 1:1 princip važe i u ovom slučaju. Na slici 2.6.4 je prikazan princip ove zaštite. Kao što vidimo na slici su prikazana dva radna puta (A ka C i D ka C) koja idu različitim prstenovima zbog bidirekcionog pristupa. Usled kvara linka između B i C, tok između A i C se preusmerava na zaštitni put unutar mrežnog čvora B. K1 i K2 bajtovi se i ovde koriste za aktivaciju preusmeravanja, kao što su se koristili u 1:1, odnosno 1:N varijantama linearne zaštite. Pri tome tumačenja ID polja se sada odnose na ID čvorova, a tumačenje tipa u K2 bajtu se odnosi na tip zaštite u slučaju prstena sa četiri optička kabla (*span* zaštita i zaštita na nivou prstena). Isto tako, s obzirom da je u pitanju zaštita na nivou prstena, tipovi zahteva u K1 bajtu se razlikuju u odnosu na tumačenja iz linearne zaštite.



Slika 2.6.4. Bidirekcionni prsten sa dva optička kabla



Slika 2.6.5. Bidirekcionni prsten sa četiri optička kabla

Bidirekcionni prsten sa četiri optička kabla se takođe sastoji od para dva prstena suprotnih orijentacija. Jedan par prstenova suprotnih orijentacija, predstavlja radni prsten, a drugi par

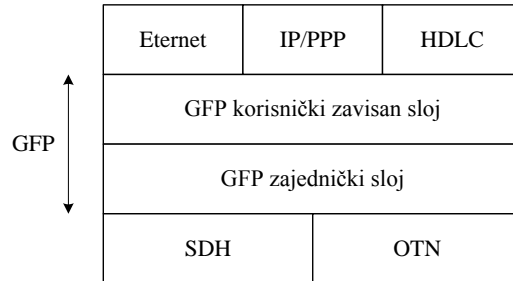
predstavlja zaštitni prsten. U slučaju prekida puta u radnom prstenu, radni saobraćaj koji je pogođen prekidom se preusmerava na zaštitni prsten istog smera (tzv. *span* zaštita). Zaštitni prsten se može koristiti za prenos saobraćaja niskog prioriteta sve dok ne dođe do potrebe da se ti resursi aktiviraju za prenos saobraćaja sa radnog prstena. U slučaju da između dva susedna čvora dođe do prekida sva četiri optička kabla, može se koristiti zaštitni prsten u suprotnom smeru (zaštita na nivou prstena). I ovde se koristi 1:1 princip na nivou prstena, pa sve mane i prednosti navedene za 1:1 princip važe i u ovom slučaju. Pri tome, poredeći sa varijantom koja koristi dva optička kabla, ova varijanta obezbeđuje bolju i efikasniju redundantnost, ali po cenu dodatnog para optičkih kablova. Na slici 2.6.5 je prikazan princip ove zaštite, pri čemu su prikazane i *span* zaštita i zaštita na nivou prstena. Zaštitu na nivou prstena treba koristiti samo ako nije moguće izvršiti *span* zaštitu.

## 2.7. Prenos paketa preko SDH

Već smo videli u potpoglavlju 2.5 da se ATM ćelije lako ubacuju u kontejnere i prenose preko SDH mreže. Međutim, IP paketi ne mogu direktno da se ubacuju jer nemaju funkciju razgraničenja IP paketa koja je u suštini funkcionalnost drugog sloja OSI modela, a isto tako IP paketi nemaju kontinualno slanje već mogu da postoje pauze između IP paketa (koje IP sloj ostavlja sloju dva OSI modela da popuni na fizičkom linku ako je to potrebno). U prvom poglavlju o ATM tehnologiji smo videli na koji način se postiže sinhronizacija na nivou ATM ćelija koja je moguća usled fiksne dužine ćelija i njihovog neprestanog kontinualnog slanja bez pauza (ako nema korisničkih ili kontrolnih ćelija, šalju se prazne ćelije). Samim tim je lako u nizu ATM ćelija izvršiti sinhronizaciju na nivou ćelija i potom razgraničavanje između ćelija. Kada bi se slali IP paketi kontinualno bez pauze, mogla bi eventualno da se vrši sinhronizacija na osnovu polja za proveru u IP zaglavlju, ali bi ona bila komplikovanija i manje pouzdana usled različitih dužina IP paketa. Dodatno, IPv6 zaglavlje čak ni ne sadrži polje za proveru što čini proces sinhronizacije na IPv6 pakete nemogućim, a takođe IP zaglavlja mogu biti promenjive dužine što unosi dodatne probleme. Otuda se IP paketi pakuju u PPP okvire, a PPP okviri se formatiraju u HDLC format, odnosno ubacuju se u HDLC okvire (kao što je opisano u RFC 1662 - *PPP in HDLC-like Framing*). Pošto HDLC okviri imaju funkciju razgraničenja upotrebom međa vrednosti 01111110 omogućena je laka detekcija HDLC okvira, a samim tim i PPP okvira, odnosno IP paketa. Ova tehnika se naziva POS (*Packet Over SDH*) kojom se omogućava prenos IP paketa preko SDH mreže (definisana u RFC 2615). Kada nema IP paketa za slanje, prostor između susednih okvira se popunjava međama. POS tehnikom je omogućeno povezivanje delova IP mreže preko SDH mreže, tj. na ovaj način se može izvršiti direktno povezivanje dva rutera Internet mreže. Napomenimo, da kao što je ATM ćelija mogla da se nađe u dva susedna kontejnera, tako i HDLC okvir može da se nađe u dva susedna kontejnera (prvi deo okvira u jednom kontejneru, a preostali u narednom kontejneru), pri čemu je tada presek napravljen na nivou bajta (ne može deo bajta da završi u jednom kontejneru, a drugi deo bajta u narednom kontejneru), jer su HDLC okviri poravnati na nivou bajta sa kontejnerom (isto kao i u slučaju ATM ćelija).

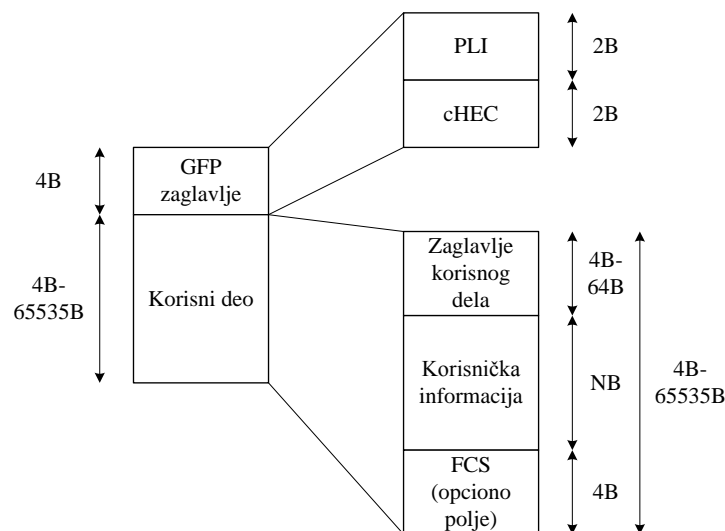
POS tehnika predstavlja dobro rešenje za prenos IP paketa, pri čemu se propusni opseg koristi na efikasan način. Međutim, postoje i određene mane. POS tehnika radi direktno sa IP paketima, što znači da se informacije drugog sloja skidaju, a nekada je zgodno prenositi i te informacije. Na primer, za povezivanje udaljenih ethernet mreža u jednu ethernet celinu, pa je tada poželjnije prenositi ethernet okvire umesto IP paketa. Propusni opseg se mora unapred definisati,

a IP saobraćaj je *bursty* prirode, pa je takav pristup neefikasan. Takođe, u zavisnosti od potrebnog propusnog opsega precizna alokacija zahtevanog propusnog opsega nije uvek moguća, pa može doći do zauzimanja većih resursa u SDH mreži nego što je realno potrebno. Otuda je razvijena i definisana GFP (*Generic Framing Procedure*) procedura za rešavanje navedenih problema i koja je definisana u ITU-T G.7041 standardu.



Slika 2.7.1. GFP slojeva arhitektura

GFP ima sličnu logiku kao i ATM mreža, gde su postojali AAL i ATM sloj, gde je AAL sloj omogućavao adaptaciju različitih tipova saobraćaja na zajednički ATM sloj koji je bio aplikacijski nezavisan. Identično tom principu, GFP sadrži dva sloja - GFP korisnički zavisan sloj (*GFP – Client-specific aspects*) i GFP zajednički sloj (*GFP – Common aspects*), kao što je prikazano na slici 2.7.1. Za fizički transport se koristi usluga SDH mreže ili OTN (*Optical Transport Network*) ili neke druge mreže koja obezbeđuje transport podataka sa sinhronizacijom na nivou bajtova. Inače, OTN mreže predstavljaju transportne mreže treće generacije koje obezbeđuju još veće protokole nego SDH mreže. Na aplikacionom sloju tj. klijent sloju se nalaze tipovi korisničkih podataka koji mogu da koriste usluge GFP procedure, poput ethernet saobraćaja, IP saobraćaja i dr. GFP kreira tzv. GFP okvire koji se transportuju preko transportne mreže, poput SDH mreže. Razlikuju se korisnički data okviri i korisnički kontrolni okviri. Korisnički data okviri nose korisničke informacije tj. podatke, a korisnički kontrolni okviri omogućavaju razmenu kontrolnih poruka između korisnika radi ostvarivanja bolje kontrole nad razmenom korisničkih podataka. Pri tome, GFP obezbeđuje kontinualno slanje GFP okvira, tako da se u momentima kada nema korisničkih okvira za slanje šalju prazni GFP okviri.



Slika 2.7.2. Struktura GFP okvira

GFP zajednički sloj obezbeđuje funkcije zajedničke svim aplikacijama tj. tipovima korisničkog saobraćaja koji koriste GFP, a to su razgraničavanje GFP okvira, sinhronizacija na nivou GFP okvira, skremblovanje, multipleksiranje različitih korisničkih tokova. Kao što vidimo funkcije veoma slične ATM sloju iz ATM mreža. GFP korisnički zavisni sloj pre svega vrši mapiranje korisničkih podataka u GFP okvire. Struktura GFP okvira je prikazana na slici 2.7.2. GFP okvir se sastoji iz GFP zaglavlja (*Core Header*) i korisnog dela (*Payload Area*). GFP zaglavlje sadrži:

- *PLI (Payload Length Indicator)* - Šesnaestobitno polje koje definiše dužinu korisnog dela u bajtovima. Minimalna dužina korisnog dela je 4 bajta, a maksimalna vrednost je 65535 bajtova. Vrednosti 0-3 *PLI* polja su rezervisane za kontrolne GFP okvire (ne kontrolne korisničke okvire, već okvire koji se razmenjuju u okviru samog GFP za upravljanje GFP vezom - za sada je definisan samo prazan GFP okvir).
- *cHEC (Core HEC (Header Error Control))* - Šesnaestobitno polje za detekciju grešaka u GFP zaglavlju. Koristi se CRC zaštita sa generišućim polinomom  $x^{16}+x^{12}+x^5+1$ . Ako sa  $M(x)$  predstavimo *PLI* polje, vrednost *cHEC* polja se računa kao ostatak deljenja  $M(x) \cdot x^{16}$  sa generišućim polinomom. Na prijemu se vrši identičan postupak, pri čemu u  $M(x)$  ulazi i *cHEC* polje. Ako se kao ostatak dobije 0 tada je prenos bio bez grešaka. Pošto se štiti veoma mali broj bita, ova CRC zaštita može da koriguje jednobitske greške i da detektuje višestruke greške.

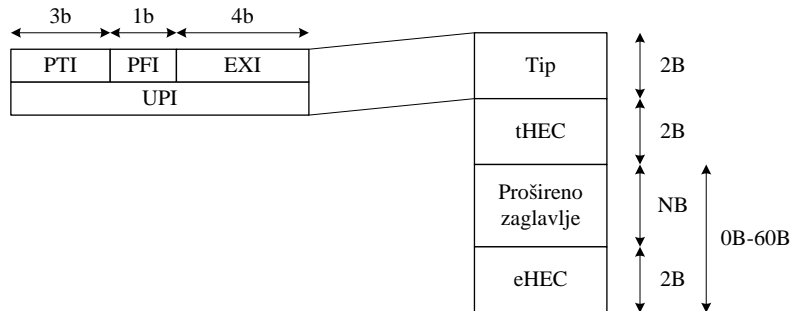
GFP zaglavlje se skrembluje tako što se vrši XOR operacija zaglavlja sa heksadecimalnom vrednošću B6AB31E0. Skremblovanje omogućava dovoljan broj tranzicija u slučaju slanja praznih GFP okvira koji se sastoje od 4 nula bajta, a takođe omogućava pouzdaniju detekciju početka GFP okvira. Detekcija početka GFP okvira tj. funkcija razgraničavanja se vrši po veoma sličnom principu kao kod ATM mreža prikazanom na slici 1.1.2.2. Razlika je u tome što se u HUNT modu vrši provera na nivou bajta, a ne na nivou bita. Takođe, kako GFP okviri nemaju fiksnu dužinu, na osnovu *PLI* polja se utvrđuje početak sledećeg GFP okvira u *PRESYNC* i *SYNC* stanjima.

Korisni deo GFP okvira se sastoji od:

- Zaglavlje korisnog dela (*Payload Header*) - Minimalna dužina ovog polja je 4 bajta, a maksimalna 64 bajta. Uloga ovog polja je da omogući ubacivanje informacija neophodnih za eventualnu obradu korisničkih podataka za prenos preko GFP okvira. Na primer, ako se vrši prenos više korisničkih tokova preko iste GFP veze, zaglavlje može da sadrži identifikaciju korisničkog toka. Takođe, ovo polje identifikuje tip korisničkog sadržaja jer struktura zaglavlja zavisi od toga koji tip saobraćaja se prenosi, pošto različiti tipovi saobraćaja imaju različite zahteve. Takođe, u zaglavlju se nalazi indikacija da li se koristi CRC zaštita korisničke informacije ili ne.
- Korisnička informacija (*Payload Information Field*) - U ovo polje se stavlja korisnička informacija koja se prenosi preko GFP okvira.
- *FCS (Payload FCS (Frame Check Sequence))* - Ovo 32-bitno polje je opciono i predstavlja CRC zaštitu korisničke informacije. Omogućava detekciju bitskih grešaka u polju korisničke informacije. Koristi se generišući polinom

$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ . Kao što vidimo koristi se isti generišući polinom kao kod AAL5 tipa u ATM mrežama. Princip generisanja vrednosti FCS polja je identičan onome opisanom u sekciji 1.3.3 za AAL5 tip. Isto važi i za princip provere ispravnosti na prijemu.

Napomenimo još jednom da je minimalna dužina korisnog dela GFP okvira 4 bajta, a maksimalna 65535 bajtova.



Slika 2.7.3. Zaglavlje korisnog dela

Struktura zaglavlja korisnog dela je prikazana na slici 2.7.3. Zaglavlje korisnog dela se sastoji iz:

- Tip (*Type*) - Ovo je obavezno polje dužine dva bajta. Ukazuje na sadržaj i format polja korisnička informacija. Sastoji se iz:
  - PTI (*Payload Type Identifier*) - Definiše tip GFP okvira. Vrednost 000 ukazuje na korisnički data okvir, 100 na korisnički kontrolni okvir, a 101 kontrolni okvir kontrolnog komunikacionog kanala (ovi okviri omogućavaju formiranje komunikacionog kanala između GFP korisnika preko koga oni mogu da razmenjuju kontrolne informacije). Ostale vrednosti su rezervisane za buduću upotrebu.
  - PFI (*Payload FCS Indicator*) - Ovaj bit predstavlja indikaciju upotrebe FCS polja. Ako je PFI=1, FCS polje se koristi i prisutno je u korisnom delu GFP okvira, a ako je PFI=0, FCS polje se ne koristi i nije prisutno u korisnom delu GFP okvira.
  - EXI (*Extension Header Identifier*) - Predstavlja identifikaciju tipa proširenog zaglavlja. Vrednost 0000 ukazuje da se ne koristi prošireno zaglavlje, 0001 da je u pitanju prošireno zaglavlje koje se koristi za slučaj korisničke informacije iz sistema zasnovanog na tačka-tačka komunikaciji, a 0010 da je u pitanju prošireno zaglavlje koje se koristi za slučaj korisničke informacije iz sistema zasnovanog na komunikaciji u prsten topologiji. Ostale vrednosti su rezervisane za buduću upotrebu. Na osnovu vrednosti EXI polja se zna struktura i format proširenog zaglavlja.
  - UPI (*User Payload Identifier*) - Definiše tip korisničke informacije smeštene u polje korisničke informacije, pri čemu interpretacija ovog polja zavisi od PTI vrednosti.

- tHEC (*Type HEC*) - Predstavlja šesnaestobitnu CRC zaštitu polja tip (ista se zaštita koristi kao i kod cHEC polja). Računa se po istom principu kao cHEC polje. Ovo polje je takođe obavezno.
- Prošireno zaglavlje (*Extension Headers*) - Ovo polje nije obavezno i ne pojavljuje se u slučaju kada je EXI=0000 (situacija kada je korisnička informacija iz sistema zasnovanog na tačka-tačka komunikaciji, pri čemu se GFP koristi samo za jedan korisnički tok). U slučaju EXI=0001 vrednosti, prošireno zaglavlje se sastoji od dva polja dužine jedan bajt - CID (*Channel ID*) i rezervnog polja (*Spare*). Vrednost EXI=0001 se koristi kada se više korisničkih tokova multipleksira na istu GFP vezu. Tada CID omogućava identifikaciju toka na prijemnoj strani. Rezervno polje sadrži bite rezervisane za eventualnu buduću upotrebu. Struktura proširenog zaglavlja za slučaj EXI=0010 nije definisana u G.7041, već je ostavljeno da se kasnije definiše struktura za taj slučaj.
- eHEC (*Extension HEC*) - Predstavlja šesnaestobitnu CRC zaštitu polja prošireno zaglavlje (ista se zaštita koristi kao i kod cHEC polja). Računa se po istom principu kao cHEC polje. Polje eHEC postoji samo ako postoji i polje prošireno zaglavlje.

Kao što smo naveli UPI polje definiše tip korisničke informacije, pri čemu tumačenje polja zavisi od PTI vrednosti. U tabeli 2.7.1 su prikazane neke od standardizovanih vrednosti za slučaj kada je PTI=000. Kao što vidimo na osnovu tabele 2.7.1, u slučaju korisničkih data okvira, UPI polje definiše kakvu vrstu korisničkog saobraćaja enkapsulira GFP okvir - na primer, ethernet okvire, PPP okvire, IPv4 pakete i dr.

**Tabela 2.7.1 - Tumačenja UPI vrednosti za slučaj kada je PTI=000**

UPI	Tip korisničkog saobraćaja
00000001	Ethernet okviri
00000010	PPP okviri
00000011	Transparentan optički kanal
00000110	Transparentan gigabitski ethernet
00001101	MPLS okviri
00010000	IPv4 paketi
00010001	IPv6 paketi

**Tabela 2.7.2 - Tumačenja UPI vrednosti za slučaj kada je PTI=100**

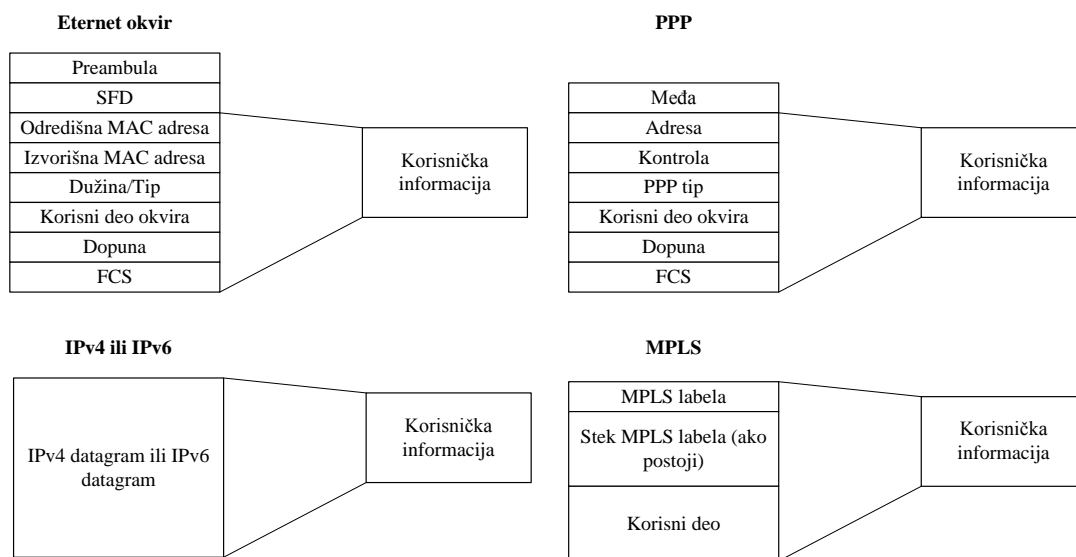
UPI	Tip korisničkog kontrolnog okvira
00000001	Gubitak signala
00000010	Gubitak poravnanja
00000011	Indikacija otklanjanja kvara
00000100	Indikacija kvara unapred
00000101	Indikacija kvara unazad
11100000-11111110	Rezervisano za privatnu upotrebu
Ostale	Rezervisano

U tabeli 2.7.2 su prikazane standardizovane vrednosti UPI polja za slučaj kada je PTI=100. Kao što vidimo, cilj korisničkih kontrolnih okvira je da omoguće signalizaciju grešaka i kvarova u GFP prenosu. Indikacija kvara unazad, omogućava prijemnoj strani da pošalje obaveštenje da je došlo do kvara u smeru od predajne ka prijemnoj strani (smer unazad sa



stanovišta prijemne strane koja šalje ovakvu poruku). Indikacija kvara unapred, omogućava predajnoj strani da pošalje obavještenje da je došlo do kvara u smeru od predajne ka prijemnoj strani (smer unapred sa stanovišta predajne strane koja šalje ovakvu poruku).

GFP omogućava dva tipa prenosa korisničkog saobraćaja preko GFP okvira - transparentni mod (GFP-T - *Transparent GFP*) i mod okvira (GFP-F - *Frame Mapped GFP*). Mod okvira podrazumeva da se korisnički saobraćaj prima u vidu okvira koji se potom pakuju u GFP okvire (jedan korisnički okvir u jedan GFP okvir). Transparentni mod podrazumeva da se korisnički saobraćaj smešta u GFP okvire, ali tako da se očuva informacija vezana za fizički prenos korisničke informacije u mreži iz koje ona potiče. Na primer, ako se koristi linijski kod 8B/10B u izvornoj mreži, korisnik može da želi da očuva tu informaciju prilikom prenosa korisničke informacije pomoću GFP okvira, tako da korisnik na prijemnoj strani dobije i tu informaciju koja mu može biti važna za izvlačenje informacije o taktu, tumačenju kontrolnih karaktera iz 8B/10B koda i sl.



**Slika 2.7.4. Mod okvira**

Na slici 2.7.4 je prikazano nekoliko primera rada moda okvira. Korisnička informacija u vidu okvira se pakuje u polje korisnička informacija GFP okvira. Pri tome, neke od informacija originalnog okvira mogu biti odbačene. Na primer, preambula i SFD polje ethernet okvira, kao i međa PPP okvira se ne prenose (napomenimo da se unutar PPP okvira može nalaziti enkapsuliran IP paket). Razlog je što su navedena polja bitna u samoj izvornoj mreži za funkcije razgraničenja, ali ih nema smisla prenositi GFP okvirima, jer na prijemu te informacije svejedno moraju da se rekonstruišu. Takođe, u slučaju ethernet okvira, između dva susedna okvira mora da postoji razmak tzv. IFG (*InterFrame Gap*), pa se na prijemnoj strani i IFG mora ubaciti između susednih primljenih ethernet okvira. Prilikom pakovanja u polje korisnička informacija mora se izvršiti poravnanje na nivou bajta, tako da se korisnički bajtovi nalaze u granicama bajtova GFP okvira (da se jedan korisnički bajt ne preseče na dva GFP bajta).

Pošto je u nekim situacijama bitno preneti sve informacije, tada se koristi transparentan mod koji omogućava transparentan prenos korisničkih informacija bez ikakvih odbacivanja. Na primer, ako se na fizičkom sloju koristi 8B/10B kod, i koristi se transparentan mod, tada je neophodno omogućiti prenos svih 8B/10B karaktera koda preko GFP veze, pre svega zbog kontrolnih karaktera koji imaju svoje funkcionalnosti u mrežama koje koriste 8B/10B linijski

kod, pa stoga nije dovoljno preneti samo korisničke bajtove. 8B/10B kod koristi gigabitski ethernet, optički kanali, FICON i ESCON. G.7041 preporuka definiše način za transparentan prenos korisničke informacije kodirane 8B/10B kodom. Princip rada na predaji je sledeći:

1. Prvo se izvrši dekodovanje korisničkog 8B/10B toka u bajtove. Pri tome, kontrolni karakteri iz 8B/10B koda se dekoduju u četvorobitne vrednosti, jer postoji manje od 16 kontrolnih karaktera. Napomena, B u 8B/10B predstavlja bit, a ne bajt.
2. Vršiti se formiranje 64B/65B koda, pri čemu ovaj kod omogućava identifikaciju kontrolnih karaktera (ako se oni nađu u toku). Napomena, B u 64B/65B predstavlja bit, a ne bajt. Najviši (65.) bit predstavlja indiciju da li se u bloku od 64 bita nalazi kontrolni karakter ili ne. Princip formiranja 64B/65B kodne reči je prikazan na slici 2.7.5. Kontrolni karakteri se nalaze na početku kodne reči, pri čemu se dodaje četiri bita svakoj četvorobitnoj predstavi kontrolnog karaktera. Tri dodata bita predstavljaju originalnu poziciju kontrolnog karaktera u originalnom 8B/10B toku (preciznije, u delu toka koji je obuhvaćen dotičnom 64B/65B kodnom reči). Četvrti dodati bit određuje da li je u pitanju poslednji kontrolni karakter (0) ili ne (1) u dotičnoj 64B/65B kodnoj reči. Pošto 65. bit samo daje indiciju da li se u kodnoj reči nalaze kontrolni karakteri ili ne, ali ne daje informaciju o broju kontrolnih karaktera i njihovoj poziciji, moraju se kontrolni karakteri ređati na početku kodne reči, pri čemu je bit indicije da li je u pitanju poslednji kontrolni karakter ili ne, neophodan za utvrđivanje početka korisničkih bajtova. Ovaj princip je korišćen da bi se efikasno iskoristila četiri bita koja su na raspolaganju uz svaku četvorobitnu predstavu kontrolnog karaktera u 64B/65B kodnoj reči.
3. Formira se superblok od 8 64B/65B kodnih reči po principu prikazanom na slici 2.7.6. 65. biti kodnih reči se dodaju na kraju iza bajtova u kojima su smešteni kontrolni karakteri i korisnički bajtovi.
4. Na svaki superblok se dodaje šesnaestobitno CRC polje koje vrši CRC zaštitu superbloka po istom principu kao u slučaju cHEC polja. Razlika je u generišućem polinomu koji je sada  $x^{16}+x^{15}+x^{12}+x^{10}+x^4+x^3+x^2+x+1$ .
5.  $N$  ovako kreiranih superblokova se stavlja u polje korisnička informacija GFP okvira.

Na prijemu se od 64B/65B kodnih reči rekonstruiše originalni 8B/10B tok. Kao što vidimo, transparentan mod omogućava prenos očuvanog 8B/10B koda. Takođe, pošto se transparentno prenose bajtovi, a ne okviri, kašnjenje koje unosi GFP je manje jer se ne čeka prijem kompletnog korisničkog okvira da bi se on poslao GFP okvirom, već se šalju bajtovi kada ih se dovoljno nakupi. Pošto se preko GFP okvira ne prenose desetobitne predstave korisničkih bajtova iz 8B/10B koda, već originalni korisnički bajtovi, postiže se i veća iskorišćenost GFP okvira.

Ulaz	65. bit	64 bita							
8 korisničkih bajtova	0	D	D	D	D	D	D	D	D
7 korisničkih bajtova 1 kontrolni karakter	1	0xxx C	D	D	D	D	D	D	D
6 korisničkih bajtova 2 kontrolna karaktera	1	1xxx C	0xxx C	D	D	D	D	D	D
5 korisničkih bajtova 3 kontrolna karaktera	1	1xxx C	1xxx C	0xxx C	D	D	D	D	D
4 korisnička bajta 4 kontrolna karaktera	1	1xxx C	1xxx C	1xxx C	0xxx C	D	D	D	D
3 korisnička bajta 5 kontrolnih karaktera	1	1xxx C	1xxx C	1xxx C	1xxx C	0xxx C	D	D	D
2 korisnička bajta 6 kontrolnih karaktera	1	1xxx C	1xxx C	1xxx C	1xxx C	1xxx C	0xxx C	D	D
1 korisnički bajt 7 kontrolnih karaktera	1	1xxx C	1xxx C	1xxx C	1xxx C	1xxx C	1xxx C	0xxx C	D
8 kontrolnih karaktera	1	1xxx C	1xxx C	1xxx C	1xxx C	1xxx C	1xxx C	1xxx C	0xxx C

D - Korisnički bajt

C - Četvorobitna predstava kontrolnog karaktera

xxx – Pozicija kontrolnog karaktera u originalnom toku

**Slika 2.7.5. Formiranje 64B/65B kodne reči**

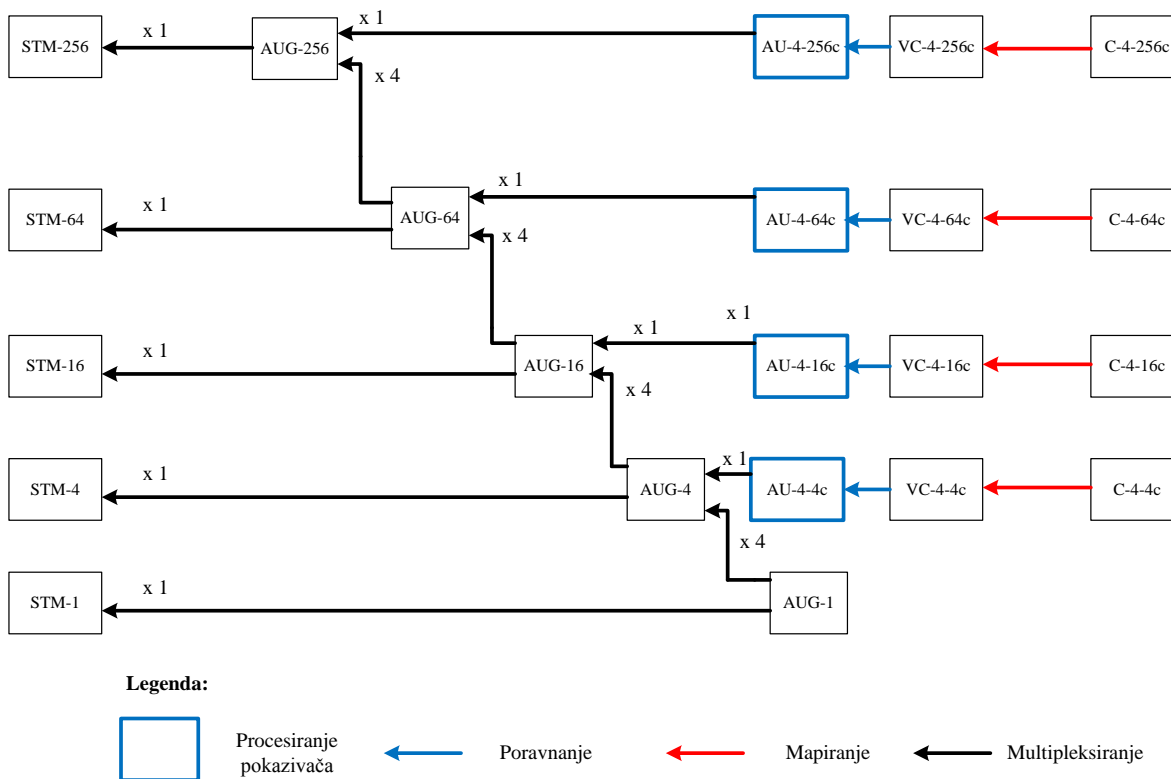
64 bita prve kodne reči
64 bita druge kodne reči
64 bita treće kodne reči
64 bita četvrte kodne reči
64 bita pete kodne reči
64 bita šeste kodne reči
64 bita sedme kodne reči
64 bita osme kodne reči
65. bit kodnih reči superbloka (8 bita)
CRC (16 bita)

**Slika 2.7.6. Formiranje superbloka**

## 2.8. Virtuelna konkatanacija

Na osnovu tabele 2.5.1 se može videti da su strukture kontejnera u koje se pakuju korisnički podaci pre svega prilagođeni standardnim protocima iz PDH mreža. Međutim, korisnici zahtevaju fleksibilniji pristup dodeli protoka, a to je bitno i sa stanovišta operatera radi postizanja bolje iskorišćenosti SDH mreže. Na primer, ako korisnik ima potrebu za protokom od 10Mb/s on bi morao da koristi kontejner C-3, odnosno virtuelni kontejner VC-3, što bi dovelo do veoma slabog iskorišćenja dotičnog kontejnera, što pravi problem korisniku prilikom iznajmljivanja/zakupljivanja resursa od operatera, ali i operateru jer ne bi mogao da pruži dobru ponudu korisniku. Otuda je u SDH mrežama omogućena konkatanacija virtuelnih kontejnera, koja omogućava fleksibilnije opsluživanje korisnika, i preciznije definisanje protoka koje korisnik može da zakupi, odnosno koristi.

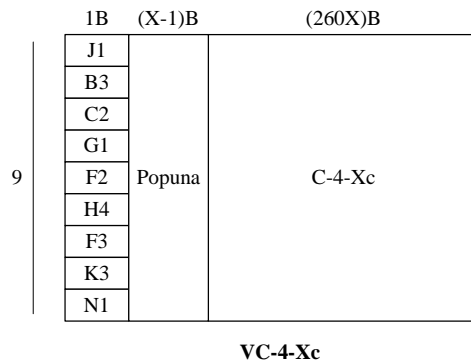
Postoje dva tipa konkatanacije - združena (*contiguous*) konkatanacija i virtuelna (*virtual*) konkatanacija. Združena konkatanacija podrazumeva da se izvrši konkatanacija virtuelnih kontejnera na predaji i da tako združeni virtuelni kontejneri idu zajedničkim putem kroz mrežu do odredišta. Virtuelna konkatanacija podrazumeva da se na predaji korisnički tok deli na više različitih virtuelnih kontejnera, pri čemu se oni prenose zasebno kroz mrežu do odredišta. Ideja je, ustvari, da se korisnički tok na predaji podeli na više zasebnih podtokova, a da se na prijemnoj strani izvrši rekonstrukcija originalnog toka iz primljenih podtokova (svakom podtoku je dodeljen virtuelni kontejner). Prednost združene konkatanacije je što je lakša rekonstrukcija toka na prijemu jer svi podaci idu istim putem kroz mrežu, a mana je manja fleksibilnost u vrednostima protoka koji se se mogu nuditi ovim putem, a takođe, teže je kroz mrežu zauzeti put za kompletnu združenu strukturu tj. veća je verovatnoća neuspeha pri uspostavi veze. Virtuelna konkatanacija nudi fleksibilnije definisanje protoka koji se mogu ponuditi, a takođe je manja verovatnoća neuspeha prilikom uspostave takve veze, jer postoji više alternativa u pogledu puteva koji se mogu izabrati jer se svaki podtok gleda zasebno. Mana je komplikovanija rekonstrukcija korisničkog toka, pošto svaki podtok ide zasebnim putem, pa može doći do razlika u kašnjenjima između podtokova i otuda je neophodna implementacija bafera na prijemnoj strani za apsorbovanje tih razlika u kašnjenjima.



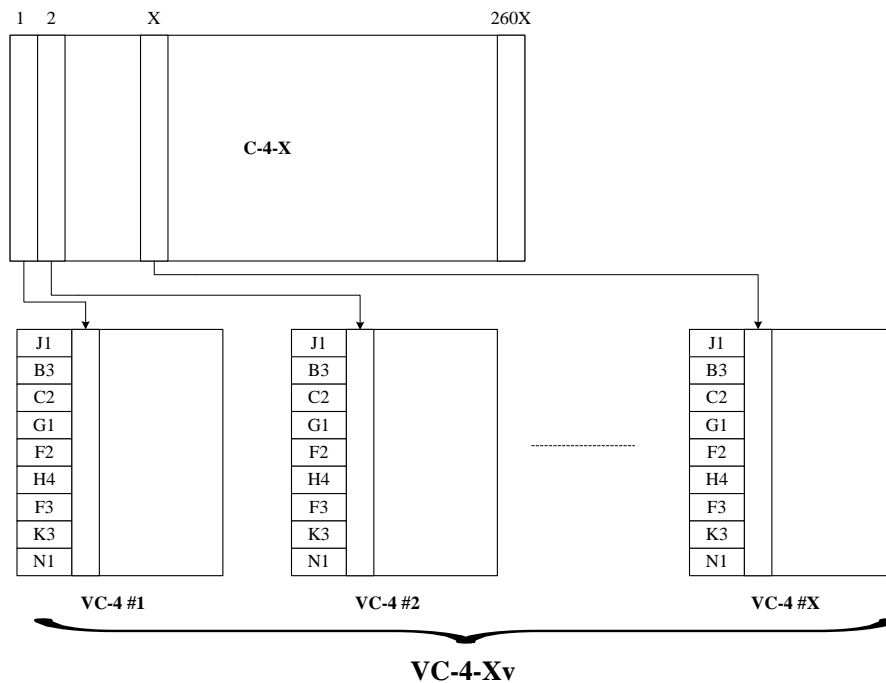
Slika 2.8.1. Formiranje STM-N struktura

Na slici 2.8.1 je prikazano kreiranje STM-N struktura od korisničkih tokova. Slika 2.8.1, ustvari, predstavlja dopunu slike 2.5.1. Možemo videti da se za kreiranje hijerarhijskih viših STM struktura mogu koristiti C-4-Xc kontejneri koji, ustvari, predstavljaju konkatanaciju C-4 kontejnera (naravno, hijerarhijski više strukture mogu da se kreiraju i multipleksiranjem četiri STM-N strukture susednog nižeg hijerarhijskog nivoa). X može imati vrednosti 4, 16, 64 ili 256 i predstavlja broj C-4 kontejnera koji su spojeni u C-4-Xc celinu. C-4-Xc kontejner se mapira u

VC-4-Xc virtuelni kontejner. AU pokazivač pokazuje na poziciju J1 bajta u POH zaglavlju. Postoji samo jedno POH zaglavlje, što je i logično jer se VC-4-Xc virtuelni kontejner posmatra kao jedna celina. Na slici 2.8.2 je prikazana struktura VC-4-Xc virtuelnog kontejnera. VC-4Xc strukture predstavljaju združenu konkatanaciju jer se čitava struktura posmatra kao jedna celina, a tako se i prosleđuje kroz mrežu. Slovo c na kraju naziva VC-4-Xc, odnosno C-4-Xc ukazuje na upotrebu združene konkatanacije.



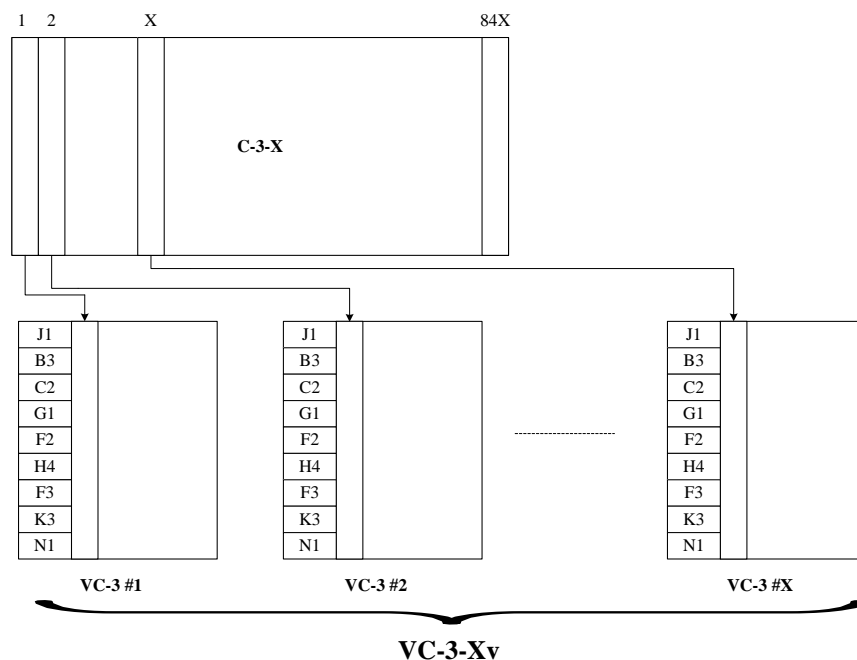
Slika 2.8.2. Formiranje VC-4-Xc virtuelnog kontejnera



Slika 2.8.3. Formiranje VC-4-Xv strukture

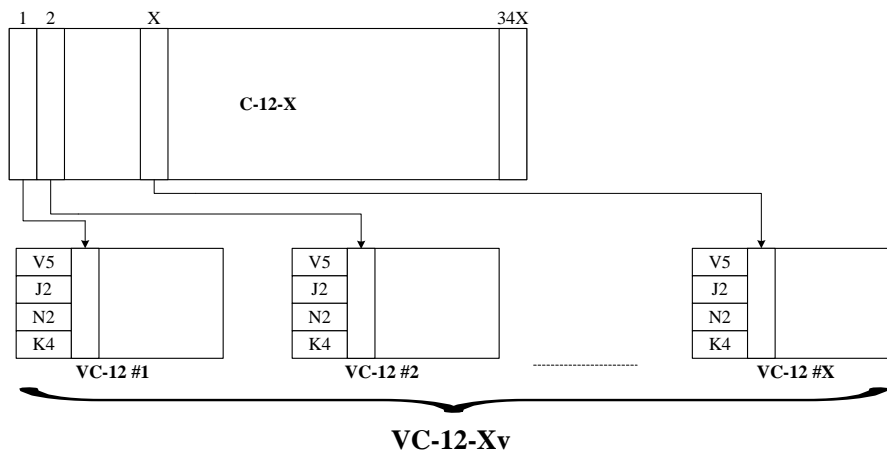
Međutim, združena konkatanacija ne pruža fino podešavanje vrednosti protoka sa stanovišta zakupa, što je veoma bitno sa stanovišta korisnika, jer on ne želi da preplati uslugu koju dobija od provajdera za transport njegovih signala zato što je zauzeo veći protok nego što mu realno treba. Dodatno, združena konkatanacija podrazumeva da se na čitavom putu kroz mrežu moraju zauzeti resursi za kompletan korisnički protok, što može biti problem u slučaju većih vrednosti protoka, pa bi sa stanovišta fleksibilnosti mreže pri zauzimanju resursa bilo poželjnije da se takav tok razbije na nekoliko celina koje bi išle različitim putevima, čime bi se povećala verovatnoća uspeha uspostavljanja takve veze, ali i sprečilo zagušenje pojedinih delova

mreže sa stanovišta zauzetosti resursa (pošto SDH radi na principu komutacije kola nema zagušenja saobraćaja). Stoga je uvedena virtuelna konkatanacija koja donosi dodatnu fleksibilnost u podešavanju protoka koji će korisnik moći da zakupi, pri čemu je i mreži olakšano da ispuni korisnikov zahtev, odnosno da uspešno uspostavi takvu vezu.



Slika 2.8.4. Formiranje VC-3-Xv strukture

Virtuelna konkatanacija podrazumeva da se korisnički saobraćaj razbije na podtokove, gde je svakom podtoku dodeljen virtuelni kontejner u koji će se pakovati korisnički podaci. Na slikama 2.8.3-2.8.5 je prikazan princip virtuelne konkatanacije za kreiranje VC-4-Xv, VC-3-Xv i VC-12-Xv struktura, pri čemu slovo v na krajevima naziva ovih struktura ukazuje da se koristi virtuelna konkatanacija.



Slika 2.8.5. Formiranje VC-12-Xv strukture

Kao što vidimo sa slika 2.8.3-2.8.5, korisnička informacija se postavlja u tzv. C-4-X (C-3-X ili C-12-X) kontejnere, koji se potom razbijaju na X virtuelnih kontejnera VC-4 (VC-3 ili VC-12), odnosno na X podtokova. Radi se raščešljavanje sadržaja kontejnera na nivou kolone,

prva kolona kontejnera u prvu kolonu korisnog dela prvog virtuelnog kontejnera, druga kolona kontejnera u prvu kolonu korisnog dela drugog virtuelnog kontejnera, ..., X-ta kolona kontejnera u prvu kolonu korisnog dela X-tog virtuelnog kontejnera, X+1-va kolona kontejnera u drugu kolonu korisnog dela prvog virtuelnog kontejnera... Svaki virtuelni kontejner ima svoj POH jer se oni prenose zasebno kroz mrežu, odnosno svaki od njih sačinjava zaseban podtok. Usled ovog principa, SDH mreža nije svesna da je urađena konkatanacija, već samo krajnji čvorovi (na krajevima sekcije puta znaju tu informaciju), što je omogućilo laku implementaciju principa virtuelne konkatanacije u postojeću infrastrukturu SDH mreža. Virtuelna konkatanacija omogućava fino podešavanje protoka. Na primer, VC-12-Xv struktura omogućava protoke u rasponu 2176kb/s do 139264kb/s sa korakom 2176kb/s (dozvoljene vrednosti za X se kreću u granicama 1 do 64), što omogućava korisniku da preciznije zakupi protok koji SDH mreža treba da mu obezbedi za prenos svojih podataka. Što se tiče VC-3-Xv i VC-4-Xv struktura, dozvoljene vrednosti za X se kreću u granicama 1 do 256. VC-3-Xv struktura obezbeđuje protoke X·48384kb/s, VC-4-Xv struktura protoke X·149760kb/s.

**Tabela 2.8.1 - Struktura multirama na nivou jedne periode prvog stepena**

Biti 0,1,2,3 H4 bajta	Biti 4,5,6,7 H4 bajta
niža četiri bita osmobitnog brojača drugog stepena multirama	0000
viša četiri bita osmobitnog brojača drugog stepena multirama	0001
0000	0010
0000	0011
0000	0100
0000	0101
0000	0110
0000	0111
0000	1000
0000	1001
0000	1010
0000	1011
0000	1100
0000	1101
niža četiri bita rednog broja podtoka	1110
viša četiri bita rednog broja podtoka	1111

Na prijemnoj strani je neophodno rekonstruisati originalni tok iz primljenih podtokova. Pri tome, neophodno je implementirati bafer na prijemu, pošto podtokovi u opštem slučaju idu različitim putevima kroz mrežu i samim tim imaju različita kašnjenja. Bafer treba da omogućí apsorpciju razlika u kašnjenjima do 256ms. Dodatno, potrebno je obezbediti sinhronizaciju na nivou virtuelnih kontejnera da bi se mogla rekonstruisati C-Xv struktura sa predajne strane, odnosno originalni korisnički tok. Otuda je neophodno u kontejnere ubaciti informacije poput rednih brojeva koje će omogućiti pravilnu rekonstrukciju na prijemu. Ove informacije se u slučaju VC-3-Xv i VC-4-Xv struktura prenose unutar H4 bajta POH zaglavlja virtuelnih kontejnera podtokova. Pomoću H4 bajta se formira dvostepeni multiram koji sadrži ukupno 4096 virtuelnih kontejnera tj. ramova. Viša četiri bita H4 bajta broje ramove u opsegu 0-15 (prvi stepen multirama), a unutar niža četiri bita se nalazi osmobitni brojač drugog stepena multirama koji broji u opsegu 0-255 (ovaj osmobitni brojač se nalazi u ramovima gde viša četiri bita imaju vrednosti 0000 i 0001, kao što je prikazano u tabeli 2.8.1). Na svakih 16 ramova se inkrementira

osmobitni brojač. Multiram omogućava poravnanje podtokova na prijemu (poravnanje virtuelnih kontejnera generisanih u istom trenutku na predajnoj strani). Pošto se korisnički tok razbija na X virtuelnih kontejnera, odnosno podtokova, neophodno je izvršiti i numeraciju podtokova da bi se mogla izvršiti rekonstrukcija, odnosno da bi se kolone iz virtuelnih kontejnera mogle mapirati ispravno u odgovarajući C-4-Xv (ili C-3-Xv) kontejner. Numeracija tj. redni broj podtoka se nalazi u ramovima gde viša četiri bita imaju vrednosti 1110 i 1111, kao što je prikazano u tabeli 2.8.1. Pošto se koristi osam bita za numeraciju podtokova, otuda se vrednost za X može kretati u granicama 1-256. Numeracija ukazuje na redni broj podtoka čime se može izvršiti pravilna rekonstrukcija C-4-Xv (ili C-3-Xv) kontejnera, odnosno izvršiti obrnuti proces od onog prikazanog na slikama 2.8.3 i 2.8.4 gde su kolone iz C-4-Xv (ili C-3-Xv) kontejnera mapirane u virtuelne kontejnere podtokova. Tabela 2.8.1 prikazuje strukturu 16 sukcesivnih ramova (perioda prvog stepena multirama) na nivou H4 bajta.

5b	6b	21b
Redni broj rama	Redni broj podtoka	Rezervisano

**Slika 2.8.6. Struktura rama koji se prenosi u bitu 2 K4 bajta**

Za slučaj VC-12-Xv strukture se za prenos rednog broja podtoka i rednog broja rama koristi bit 2 u K4 bajtu. Periodična struktura koja se prenosi ovim bitom je prikazana na slici 2.8.6. Redni broj rama omogućava poravnanje VC-12 kontejnera na prijemu, redni broj podtoka omogućava pravilnu rekonstrukciju sadržaja C-12-Xv kontejnera, a biti rezervisani za eventualnu buduću upotrebu su postavljeni na vrednost 0. Kao što vidimo, ovde nema signala koji bi omogućio određivanje početka rama u bitu 2 K4 bajta, što onemogućava pravilno tumačenje rama. Razlog za odsustvo signala za sinhronizaciju na nivou rama je što se ram u bitu 2 K4 bajta sinhroniše na multiram strukturu u bitu 1 K4 bajta i koja je prikazana na slici 2.5.4.

Virtuelna konkatanacija je omogućila korisnicima da lakše i preciznije zadovolje svoje potrebe u pogledu protoka koji žele da ostvare preko SDH mreže, a isto tako i operaterima je omogućila lakše dodavanje novih tokova u SDH mrežu i efikasniju ponudu samim korisnicima. Međutim, korisnici često imaju potrebu da povremeno povećaju svoj protok kroz SDH mrežu na izvestan period (na primer, ako se pri kraju radne nedelje vrši transfer podataka iz filijala u centralu tada je saobraćaj intezivniji od proseka pa je potrebno tražiti dodatni protok u SDH mreži). U tu svrhu je razvijen LCAS (*Link Capacity Adjustment Scheme*) metod koji omogućava korisnicima da dinamički menjaju protok kroz SDH mrežu, tako što će zahtevati dodatne resurse od SDH mreže, a pri tome će se korisnički podaci bez prekida i dalje transportovati kroz već zauzete resurse mreže. Ako se zahtev za povećanjem odobri, dodatni kontejneri će biti ubačeni u virtuelnu konkatanaciju, pri čemu naglašavamo još jednom ne dolazi do prekida korisničkog saobraćaja koji neprestano ide kroz već zauzete resurse, odnosno virtuelne kontejnere koji su već deo ostvarene virtuelne konkatanacije. Analogno procesu povećanja, korisnik može i da smanji svoj protok, odnosno da smanji broj podtokova i da oslobodi deo resursa u mreži bez prekida u prenosu saobraćaja preko resursa koji ostaju zauzeti.

U slučaju VC-3-Xv i VC-4-Xv struktura, dodaju se dodatna tumačenja u H4 bajt u okviru multirama kao što je prikazano u tabeli 2.8.2. CTRL omogućava predajnoj strani slanje komandi prijemnoj strani. Definisane su sledeće komande: FIXED, ADD, NORM, EOS, IDLE, DNU. FIXED ima vrednost 0000 i signalizira prijemnoj strani da se ne koristi LCAS (otuda i vrednost 0000 jer će se ta vrednost uvek naći na dotičnoj poziciji i kad se ne koristi virtuelna konkatanacija, a i kad se koristi virtuelna konkatanacija bez LCAS). ADD se koristi za



dodavanje novog člana u virtuelnu konkatanaciju (dotični virtuelni kontejner će se pridružiti tzv. VCG (*Virtual Concatanation Group*) grupi tj. dotičnoj VC-3-Xv ili VC-4-Xv strukturi). NORM označava regularan prenos unutar VCG grupe tj. dotični virtuelni kontejner je i dalje u VC-3-Xv ili VC-4-Xv strukturi i prenosi deo korisničkih podataka originalnog toka. EOS predstavlja indikaciju poslednjeg člana grupe (podtok sa najvećim rednim brojem), pri čemu i taj član grupe vrši regularan prenos (kao u slučaju NORM komande). IDLE označava virtuelni kontejner koji još nije postao član VCG grupe (IDLE se šalje na početku inicijalizacije (stvaranja) VCG grupe sve dok se ne pošalje komanda ADD umesto IDLE čime se dotični virtuelni kontejneri dodaju u grupu) ili koji treba da se ukloni iz VCG grupe pa se stoga više ne koristi. DNU označava da se sadržaj dotičnog virtuelnog kontejnera ne sme koristiti u rekonstrukciji (ova komanda se generiše kada prijemna strana prijavi grešku preko MST polja u vidu FAIL statusa dotičnog člana VCG grupe).

**Tabela 2.8.2 - Struktura multirama na nivou jedne periode prvog stepena u slučaju LCAS podrške**

Biti 0,1,2,3 H4 bajta	Biti 4,5,6,7 H4 bajta
niža četiri bita osmobitnog brojača drugog stepena multirama	0000
viša četiri bita osmobitnog brojača drugog stepena multirama	0001
CTRL	0010
GID (000g)	0011
0000	0100
0000	0101
CRC-8	0110
CRC-8	0111
MST	1000
MST	1001
RSack (000r)	1010
0000	1011
0000	1100
0000	1101
niža četiri bita rednog broja podtoka	1110
viša četiri bita rednog broja podtoka	1111

**Tabela 2.8.3 - MST multiram**

Vrednost osmobitnog (drugi stepen multirama) brojača	Redni brojevi podtokova pokrivenih MST bitima
0, 32, 64, 96, 128, 160, 192, 224	0,1,2,3
	4,5,6,7
1, 33, 65, 97, 129, 161, 193, 225	8,9,10,11
	12,13,14,15
....	.....
31, 63, 95, 127, 159, 191, 223, 255	248, 249, 250, 251
	252, 253, 254, 255

GID (*Group Identification Bit*) bit se koristi za verifikaciju na prijemu da svi virtuelni kontejneri koji pripadaju VCG grupi, odnosno svi podtokovi potiču sa istog izvorišta. Naime, izvorište generiše pseudoslučajnu sekvencu i nju šalje preko GID bita, pri čemu svi virtuelni kontejneri istog rama (ista vrednost brojača prvog i drugog stepena multirama) imaju istu vrednost GID bita tj. pseudoslučajna sekvencu je poravnata između podtokova. RSack (*Re-*

*Sequence Acknowledgment*) bit se koristi za slanje potvrde sa prijemne strane da je prihvaćena renumeracija podtokova koja je tipično posledica dodavanja novog člana u VCG grupu ili uklanjanja nekog člana iz VCG grupe. Slanje potvrde se šalje u vidu invertovanja prethodne vrednosti RSack bita (sa 0 na 1 ili sa 1 na 0, u zavisnosti šta je bila prethodna vrednost). Preko MST (*Member Status*) prijemna strana šalje indikaciju o stanju podtokova sa stanovišta prijemne strane. Indikacija je OK (0) ili FAIL (1). MST kreira svoj multiram po principu prikazanom u tabeli 2.8.3 gde svaki bit odgovara nekom podtoku. CRC-8 zaštita vrši zaštitu vrednosti H4 bajtova počev od kraja prethodne CRC-8 vrednosti. Koristi se generišući polinom  $x^8 + x^2 + x + 1$ .

Proces dodavanja je jednostavan. Predajna strana šalje komandu ADD i ako prijemna strana da potvrdu u vidu MST=OK statusa za dotični virtuelni kontejner, predajna strana zna da može da počne da koristi dotični virtuelni kontejner kao novog člana VCG grupe. Novom članu je dodeljen sledeći redni broj (ako je prethodno bilo X članova grupe, tada će novi član dobiti redni broj X jer se tokovi broje od 0) i umesto ADD se počinje slati EOS komanda kao indikacija poslednjeg člana grupe. Član grupe koji prethodno bio poslednji, a sada je pretposlednji će početi da šalje komandu NORM umesto EOS. Korisnički podaci se počinju slati u prvom virtuelnom kontejneru tj. ramu koji ide nakon CRC bita koji su vršili zaštitu skupa od 16 ramova u kome se našla prva EOS komanda koja je poslata nakon pozitivne potvrde sa prijemne strane o prihvatanju novog člana VCG grupe.

Proces uklanjanja je takođe veoma jednostavan. Vršiti se slanje komande IDLE u podtoku najvećeg rednog broja. Sledeći ram (virtuelni kontejner) nakon komande IDLE više neće sadržati korisničke podatke. Podtok koji je bio pretposlednji, sada postaje poslednji, pa se u njemu umesto NORM komande počinje slati EOS komanda. Moguće je ukloniti i podtok koji nije najvećeg rednog broja, i tada je potrebno renumerisati podtoke većeg rednog broja od uklonjenog tako da se njihovi redni brojevi dekrementiraju za 1. Ova renumeracija se odvija u istom trenutku kad i slanje komande IDLE.

5b	6b	4b	1b	4b	1b	8b	3b
Redni broj rama	Redni broj podtoka	CTRL	GID	Rezerv.	RS-ack	MST	CRC-3

Slika 2.8.7. Struktura rama koji se prenosi u bitu 2 K4 bajta za slučaj LCAS podrške

Tabela 2.8.4 - MST multiram za VC-12-Xv

Redni broj rama	Redni brojevi podtokova pokrivenih MST bitima
0, 8, 16, 24	0,1,2,3,4,5,6,7
1, 9, 17, 25	8,9,10,11, 12,13,14,15
....	.....
7, 15, 23, 31	56, 57, 58, 59, 60, 61, 62, 63

LCAS za VC-12-Xv radi po identičnom principu. Razlika je u tome što se koristi CRC-3 kod sa generišućim polinomom  $x^3 + x + 1$ . Takođe, MST multiram je kraći jer je maksimalan broj članova grupe manji (64 u odnosu na 256 iz VC-4-Xv i VC-3-Xv struktura). Struktura multirama u slučaju podrške LCAS je prikazana na slici 2.8.7. Kao što se vidi, rezervni biti sa slike 2.8.6 su u velikoj meri iskorišćeni za LCAS podršku, pa se može videti da je, ipak, važno ostavljati rezervna mesta u prvobitnim definicijama struktura okvira i zaglavlja paketa da bi se mogle dodavati eventualne dodatne funkcionalnosti u budućnosti. Struktura MST multirama je data tabelom 2.8.4.