

RTV KLUB MURSKA SOBOTA

DIGITALNI MOSTOVI

"HITRI PACKET RADIO"

Murska Sobota, september 2000

Blišč in beda protokola AX.25

Matjaž Vidmar, S53MV

1. Beg iz mračnih časov RTTY

Packet radio vsakodnevno uporablja dobršen del aktivnih slovenskih radioamaterjev. Ker je od zadnjega seminarja ZRS o packet-radiu minilo že skoraj desetletje, se mi zdi potrebno, da v tem članku ponovim osnove protokola AX.25 (postopka vzpostavljanja zveze pri packet radiu), saj je med bralci CQ ZRS in uporabniki paketa marsikdo, ki teh osnov ne pozna.

Razen tega se je v desetih letih nabralo marsikaj novega. Marsikatero ne delujočo teorijo, slabo napisano programsko opremo in ne kakovostno strojno opremo smo povsem zavrgli. Ker je večina razpoložljive literature bodisi zastarela in pomanjkljiva, na primer (1), (2) in (3), ali pa je sploh ni, se mi zdi nujen članek o osnovah delovanja packet-radia.

Opis packet radia moram nujno začeti s kratko zgodovino. Amaterski

packet radio se je pojavil pred skoraj dvajsetimi leti. Takrat so se pojavili prvi amaterjem dostopni mikroročunalniki. Ena od prvih amaterskih uporab računalnikov je bilo nadomestilo za nepredvidljivo ropotajočo elektromehansko pošast z imenom teleprinter.

Računalniki so seveda omogočali še kaj več kot le novo obliko strojne telegrafije. Računalnik naj bi predvsem nadomestil operaterja pri najbolj zamudnih in duhamornih opravilih na radijski postaji, kot je preklon sprejem-oddaja, preverjanje kvalitete sprejema in potrjevanje pravilno sprejetih sporočil oziroma zahteva ponavljanja manjkajočih delov sporočila. Računalnik je pri enostavnem RTTY skoraj neizkoriščen, od njega bi radioamaterji želeli bistveno več.

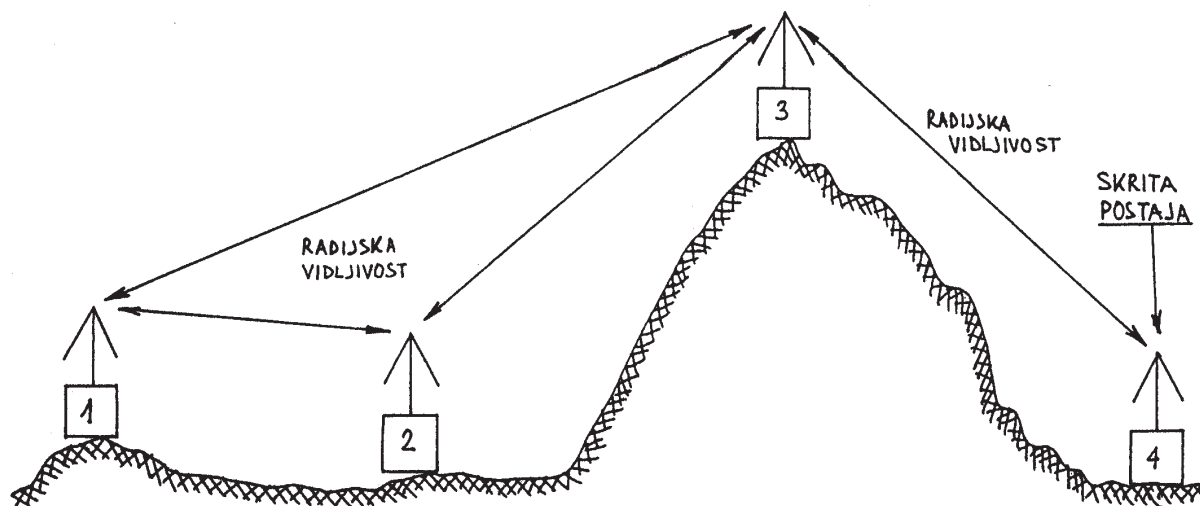
Ker je različnih možnosti skoraj nešteto, se je najprej nujno dogovoriti o pravilih igre, z drugimi besedami o protokolu komuniciranja med radioamaterskimi računalniki. Pred dvema

desetletji so se pojavili številni predlogi.

AMTOR je na primer le prirejeni profesionalni protokol TOR (Teletype Over Radio), ki omogoča le RTTY zvrst zveze z omejeno možnostjo nadzora napak (7-bitni znaki z razmerjem enic in ničel 3:4). Ker je TOR nastal v dobi, ko so teleprinterski inženirji prisegali na mehanske releje in z nezaupljivostjo gledali na vakuumске elektronke, je v dobi računalnikov zaslužen pristal med staro šaro v muzeju.

Sami računalniki sicer razpolagajo z RS-232 asinhronim vmesnikom, ki je najpreprostejša različica računalniškega RTTY, vendar se stvar v amaterskih zvezah ni obnesla. Tudi vmesniki za kasetofon (pri Mavrici, C64 ipd) so se obnesli le na kratke razdalje in med računalniki iste vrste.

Zgledujoč se po profesionalnih satelitih so pri AMSAT-u razvili učinkovit protokol 400bps Manchester/PSK za telemetrijo in telekoman-



- Vsi udeleženci imajo simpleksne radijske postaje, vsi sprejemajo ali oddajajo na istem radiofrekvenčnem kanalu.
- CSMA dostop (Carrier-Sense Multiple Access) - udeleženci oddajajo samo takrat, ko je kanal prost in ko imajo novo sporočilo za oddajo.

Slika 1 - Radijski sistem zvez s paketnim prenosom.

do satelita AMSAT-P3A (predhodnik AO-10). Žal je AMSAT-P3A meseca maja 1980 končal v atlantskem oceanu zaradi okvare prve stopnje nosilne rakete Ariane. Učinkoviti 400bps protokol je pri tem utonil v pozabo in se danes uporablja edino za upravljanje amaterskih satelitov.

Konec sedemdesetih let so kanadski radioamaterji iz mesta Vancouver pod vodstvom VE7APU poskusili z uporabo profesionalnega protokola HDLC ali X.25. V začetku osemdesetih let so jim sledili amaterji iz mesta Tucson (ZDA), ki so en sam bajt naslovnega polja HDLC protokola zamenjali z radioamaterskimi klicnimi znaki in nastali protokol poimenovali AX.25. Za razliko od kanadskih kolegov se je njihovo združenje TAPR (Tucson Area Packet Radio) izkazalo tudi bolj poslovno: kaj kmalu so pripravili prodajo kit-sestavljank za napravo z imenom TNC-1.

Čeprav ni TNC-1 nikoli dočakal brezhibne programske opreme, se je napravica izkazala za uspešnico in kmalu so jim sledili drugi proizvajalci radioamaterske opreme. Ker sam protokol ni bil še povsem točno definiran in se naprave različnih proizvajalcev niso vedno "razumele" med sabo, se je ameriška zveza radioamaterjev ARRL odločila za izdajo opisa izboljšane protokola AX.25 V2 v obliki knjige (1) leta 1984. Slovenski prevod tega opisa smo dobili že štiri leta kasneje, se pravi 1988 v (3).

Razvoj se je seveda nadaljeval v številnih smereh, ki jih v tem članku niti omeniti ne morem. Žal je večina dosežkov slabo dokumentiranih. Delovanje nekaterih sodobnih sistemov,

kot sta CLOVER oziroma PACTOR-2, je tako komplicirano, da brez univerzitetnega znanja matematike sploh ne gre.

Tudi osnovni protokol AX.25 je še vedno poln napak in nedefiniranih stanj, tako da je pravi čudež, da nam danes sploh deluje amatersko packet radio omrežje širom sveta. Na srečo je delovanje protokola AX.25 razmeroma enostavno in razumljivo povprečnemu radioamaterju s srednješolsko izobrazbo. Namen tega članka je zato opis delovanja protokola AX.25, vključno z njegovimi dobrimi in slabimi stranmi.

2. Radijski sistem zvez s paketnim prenosom

Osnovna zamisel radijskega sistema zvez s paketnim prenosom je prikazana na sliki 1. Vsi udeleženci v prometu uporabljajo simpleksne radijske postaje na enem samem radiofrekvenčnem kanalu. Udeleženci so stalno na sprejemu in oddajajo le takrat, ko imajo novo sporočilo za oddajo.

Dostop do radijskega kanala je naključen. Vsak udeleženec preprosto poskusi oddati svoje sporočilo in pričakuje potrditev sprejema od naslovnika. V takšnem sistem se kaj lahko zgodi, da dva ali več udeležencev nevede oddaja hkrati in v tem primeru so vsa oddana sporočila izgubljena. V slučaju neuspešne oddaje udeleženci sporočilo čez čas ponovijo, vse do sprejema potrditve.

Da bi omejili medsebojne motnje, vsak udeleženec pred svojo oddajo prisluškuje vsemu prometu na kanalu. Udeleženec preklopi svojo

radijsko postajo na oddajo samo v slučaju, ko najde prost kanal (Carrier Sense Multiple Access ali CSMA). Ta postopek seveda ne izloči vseh motenj, saj lahko čaka s s sporočili za oddajo več udeležencev hkrati.

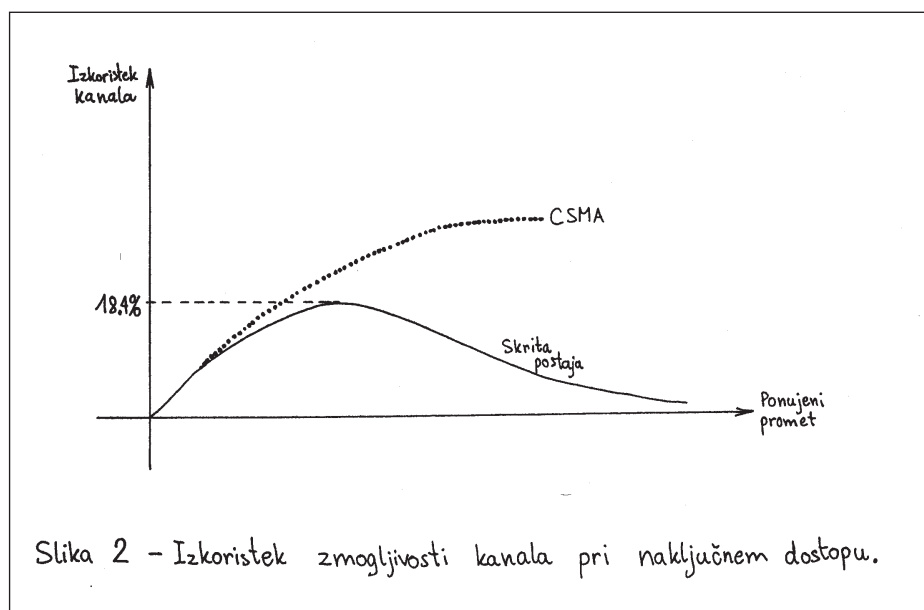
Pri naključnem dostopu je izkoristek zmogljivosti kanala razmeroma majhen, kot to prikazuje slika 2. Izkoristek zmogljivosti je še posebno neugoden v slučaju skrite postaje, ki nima radijske vidljivosti do drugih postaj. Skrita postaja onemogoča urejen CSMA dostop do kanala, zato začne zmogljivost kanal celo upadati, ko postane ponudba prometa prevelika.

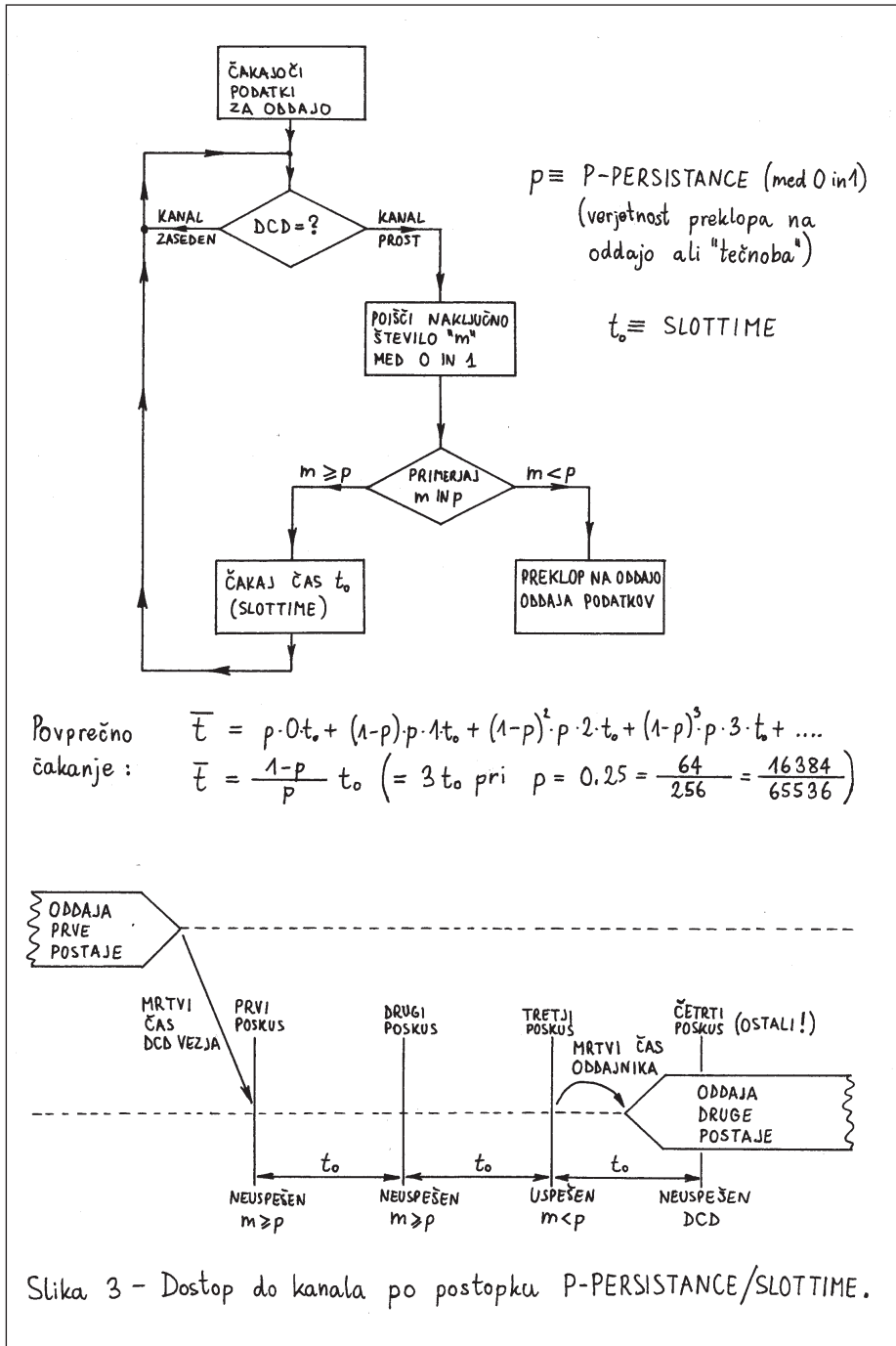
Problem dostopa do radijskega kanala v osnovnem dokumentu protokola AX.25 v (1) (oziroma v prevodu v (3)) sploh ni omenjen. Večina radioamaterske strojne in programske opreme podpira postopek P-PERSISTENCE/SLOTTIME, ki je prikazan na sliki 3. Prikazani postopek je neposredno privzet iz žičnih računalniških omrežij, zato na radijskih napravah ni najbolj učinkovit.

Delovanje postopka P-PERSISTENCE/SLOTTIME je naslednje. Ko udeleženec želi na oddajo z novimi podatki, najprej preveri zasedenost kanala. Ustrezní signal imenujemo DCD (Digital Carrier Detect), lahko bi mu rekli tudi digitalni skvelč. V slučaju zasedenega kanala udeleženec jasno čaka, da se kanal sprostí.

Ko udeleženec najde prost kanal (neaktiven DCD), poišče naključno število "m" med 0 in 1 in ga primerja z verjetnostjo preklopa na oddajo "p" ali P-PERSISTENCE. Če je število "m" preveliko, udeleženec ponovno čaka. Najprej počaka čas "to" ali SLOTTIME, potem pa spet preverja DCD in ponovno čaka na prost kanal. Smisel opisanega postopka je v tem, da se zmanjša verjetnost, da bi dva udeleženca hkrati prešla na oddajo.

Postopek P-PERSISTENCE/SLOTTIME jasno dodaja neko povprečno čakanje, ki je neposredno odvisno od velikosti SLOTTIME. SLOTTIME je lahko zelo kratek v žičnih omrežjih, kjer je preklon na oddajo trenuten in delovanje DCDja ni odvisno od sinhronizacije sprejemnika (na primer Ethernet). V radijskih omrežjih je preklon na oddajo počasen (vnihanje oscilatorjev v oddajniku). Počasno je tudi delovanje DCD vezja v vseh slučajih: skvelč radijske postaje pri enostavnem Baycom modemu ali vnihanje DPLL zanke v bolj kompliciranih modemih.





Slika 3 - Dostop do kanala po postopku P-PERSISTANCE/SLOTTIME.

meta zanje.

DAMA ima več slabih strani. Ciklično klicanje vseh ostalih udeležencev po vrsti povzroča velik in nepotreben QRM. Neposredne zveze med podrejenimi udeleženci brez posredovanja vozlišča so skoraj nemožne. Precej programske opreme ne podpira postopka DAMA. Končno, pri nas nihče ne želi imeti policajja na vrhu hriba, ki bi vsem drugim samo ukazoval. Na osojni strani Alp je mnenje izgleda drugačno.

Problem skrite postaje lahko rešimo tudi drugače, s primernejšim krmljenjem drugih časovnih konstant, kar sploh ne moti združljivosti z običajnimi AX.25 postajami. Žal večina programske opreme podpira le opisani AX.25 in popolnoma zanemarljiv problem skrite postaje.

Nobeden od opisanih postopkov sicer ne rešuje nekaterih pogostih pojavov, na primer QRM oddaljene postaje, ki blokira DCD. Večina programske opreme v tem slučaju ne gre na oddajo in čaka v neskončnost. Zveza jasno pade. Težavo bi rešila dodatna "kužapazi" časovna konstanta, da gre udeleženec čez določen, dovolj dolg čas, v vsakem slučaju na oddajo.

3. AX.25 paketi in okvirji

En preklon udeleženca na oddajo in potem nazaj na sprejem imenujemo en AX.25 paket. Vsak AX.25 paket nujno vsebuje sinhronizacijsko glavo in enega ali več okvirjev s podatki, kot je to prikazano na sliki 4. Pri tem so lahko različni okvirji v istem paketu namenjeni različnim naslovnikom v različnih AX.25 zvezah.

AX.25 paket vsebuje celo vrsto različnih mrtvih časov in vmesnih polj zaradi omejitev strojne in programske opreme. Od vseh zakasnitev pobere največ časa vklop oddajnika in sinhronizacijska glava, kar skupno nastavljammo s parametrom z neustreznim imenom TXDELAY. Neustreznim zato, ker TXDELAY ne vsebuje le mrtvega časa oddajnika, pač pa tudi mrtvi čas skvelča ali DCD vezja na drugem koncu zveze.

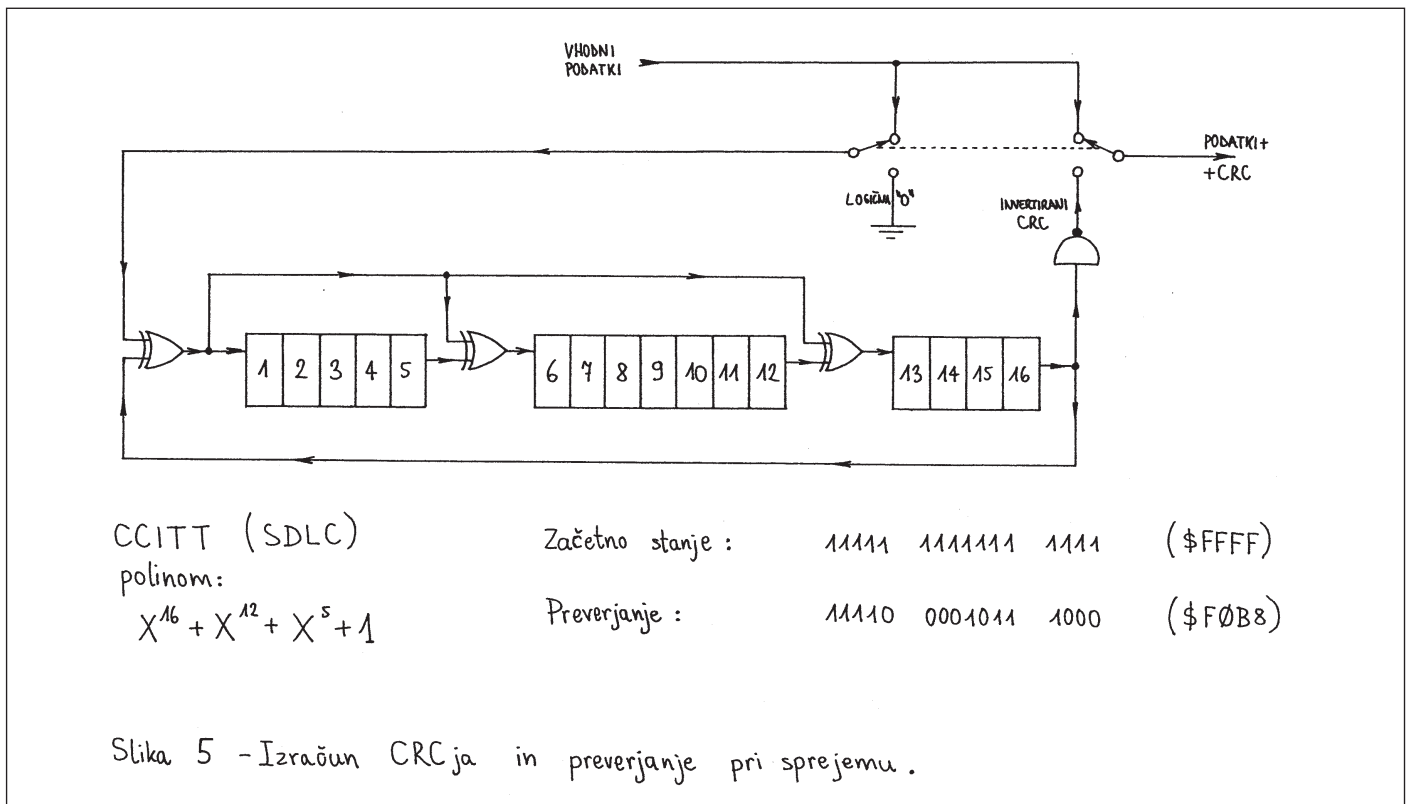
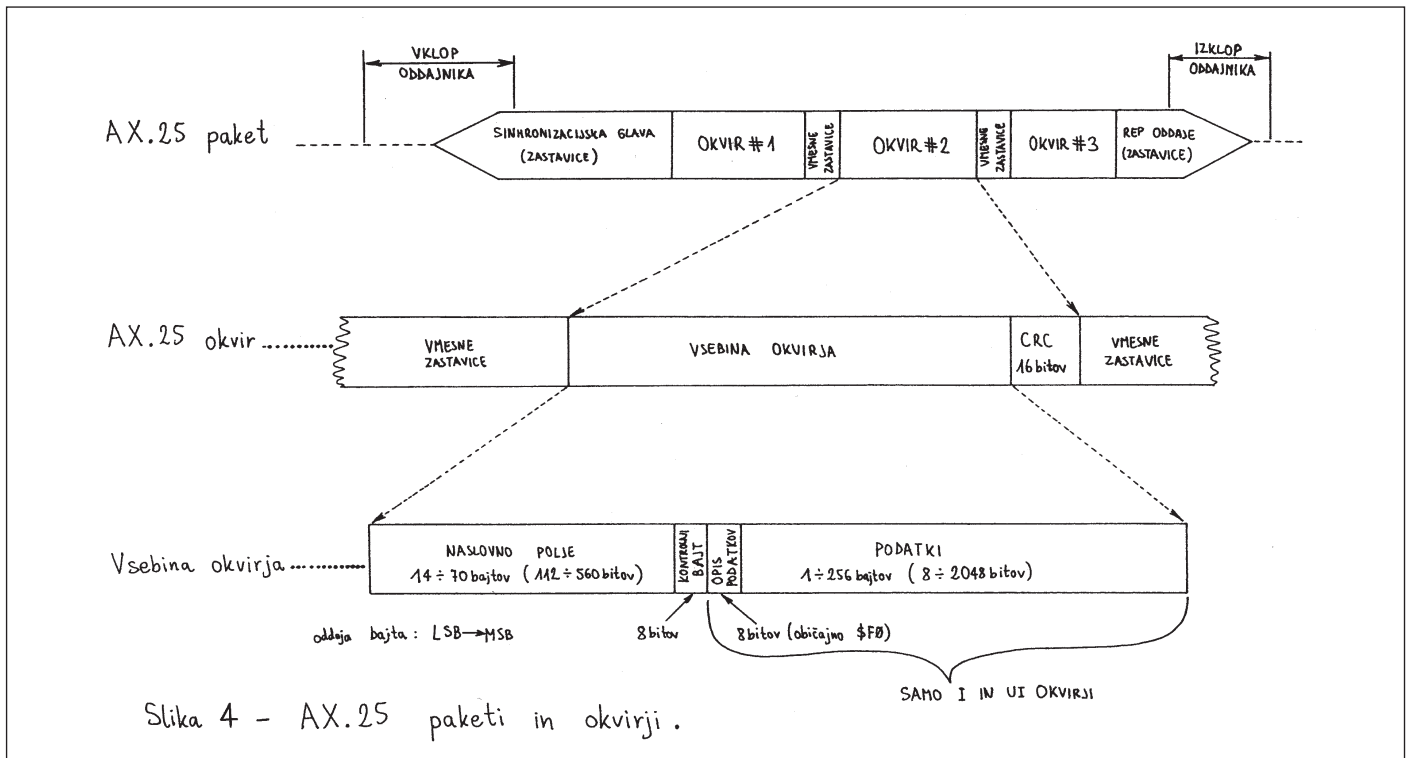
Prav mrtvi čas sprejemnika zahteva, da parametra TXDELAY ne moremo nastavljati kar po svoje, pa če imamo še tako hitro radijsko postajo. Parameter TXDELAY mora biti usklajen z vsemi ostalimi udeleženci na istem radijskem kanalu, da CSMA dostop do kanala sploh lahko deluje. Prekratek TXDELAY povzroča

V radijskem sistemu mora biti SLOTTIME vedno večji od vsote mrtvega časa oddajnika in mrtvega časa DCDja sprejemnika. SLOTTIME mora biti vedno večji ali enak parametru TXDELAY, da ostale postaje zanesljivo detektirajo DCD in ostanejo na sprejemu ob naslednjem izteku svojega SLOTTIME. Žal ima večina radioamaterjev ta parameter povsem napačno nastavljen in nekateri (sicer v dobri veri) takšne napačne nastavitve celo razglašajo naprej (4).

Kljub napačno nastavljenim parametrom večine udeležencev to še ni katastrofa, saj je sam postopek P-PERSISTANCE/SLOTTIME povsem neučinkovit v slučaju skrite postaje.

V našem slovenskem omrežju je to tudi najbolj pogost primer, saj ima večina uporabnikov le radijsko vidljivost do vozlišča na hribu in drugih uporabnikov sploh ne more neposredno poklicati.

Da bi se izognili težavam skritih postaj, so nemški radioamaterji razvili programsko opremo, ki podpira postopek DAMA (Demand-Assigned Multiple Access). Postopek DAMA potrebuje za vsak radijski kanal eno upravno postajo (to je običajno vozlišče na hribu). Vsi ostali udeleženci so podrejeni upravni postaji in smejo na oddajo samo takrat, ko jim to dovoli upravna postaja. Upravna postaja zato ciklično kliče vse udeležence, tudi če nima radijskega pro-



obilico medsebojnih motenj, ki bistveno upočasnijo koristni pretok podatkov na radijskem kanalu.

Učinkovitost prenosa podatkov na radijskem kanalu zavisi od tega, koliko smo se udeleženci sposobni med sabo dogovoriti za uporabo smiselnih parametrov. Največjo učinkovitost jasno dosežemo takrat, ko vsi udeleženci na istem kanalu uporabljajo isti TXDELAY, ki mora biti prilagojen NAJPOČASNEJŠI postaji. SLOTTI-

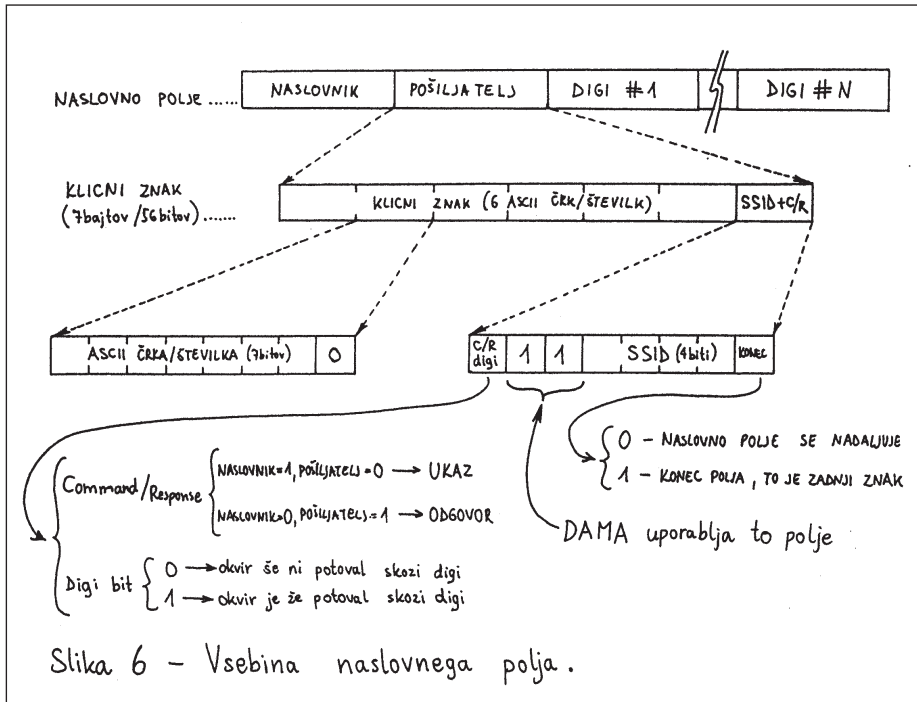
ME nastavimo potem kar enak TXDELAY.

V radioamaterski literaturi pogosto prečitamo ravno obratno trditev: za vsako ceno nastaviti čim krajši TXDELAY! Nekateri so šli celo tako daleč, da so svoja Flexnet vozlišča sprogramirali tako, da rušijo zveze uporabnikom z dolгим TXDELAY.

In koliko naj bo TXDELAY (oziroma SLOTTIME)? S hitro postajo in pravim DCDjem stvari delujejo na

1200bps brezhibno celo s 100ms. Z Baycom modemom in staro radijsko postajo, ki ima počasen PLL in še počasnejši skvelč, moramo TXDELAY včasih povečati tudi preko 500ms. Večina naših vozlišč ima na 1200bps nastavljen TXDELAY na 400ms, kar naj bi zadoščalo za večino uporabnikov, tudi tistih z Baycom modemom.

Na višjih hitrostih je TXDELAY lahko ustrezno manjši, če so radijske postaje temu prirejene. Za naše



pomikalne registre za izračun in preverjanje CRCja. CRC zato ne štiti sporočila pred napakami, ki se pojavijo med zaporednim vmesnikom in računalnikom (izgubljene prekinitev) oziroma med TNCjem in računalnikom na RS-232 prenosni poti. Večina napak v protokolu AX.25 lahko zato pripišemo slabi strojni in programski opremljeni, ki izgublja prekinitev in zameša znake v sporočilu. Sam protokol AX.25 in predvsem CCITT polinom sta pri tem povsem nedolžna!

4. Vsebina AX.25 okvirjev

Sestava koristne vsebine okvirja (brez CRCja) je prikazana na spodnji vrstici na sliki 4. Vsak AX.25 okvir vsebuje naslovno polje in kontrolni bajt, ki opiše, za kakšen okvir gre. Nekateri okvirji (predvsem I in UI) vsebujejo tudi podatkovno polje. Vsi AX.25 okvirji vsebujejo vedno celo število bajtov (ki jih (1) imenuje "okteti") za razliko od profesionalnih protokolov, kjer okvirji vsebujejo poljubno število bitov, kar običajno ni deljivo z 8. Vrtni red oddaje bajta je najprej najnižji bit (LSB) vse do najvišjega bita (MSB) skladno z večino zaporednih računalniških vmesnikov.

Prvi bajt podatkovnega polja je namenjen opisu podatkov. V običajnih AX.25 zvezah je ta bajt vedno postavljen na \$F0. V (1) je prvi bajt zelo neustrezno imenovan PID ali Protocol Identifier. Povsem jasno nima ta bajt nobene zveze s protokolom AX.25, pač pa le opisuje podatke, ki lahko pripadajo kakšnemu višjemu protokolu, ki je vgrajen v AX.25 podatkovne okvirje (na primer TCP/IP). PID torej spada med neobdajtraba navlako, ki se je moramo držati zaradi kompatibilnosti z obstoječo programsko opremo.

PID bajtu sledijo pravi podatkovni bajti. Podatkovni bajt bi moral biti vsaj eden, v opisanem protokolu AX.25 pa naj bi jih bilo največ 256. Ustrezni parameter se v (1) imenuje "N1" ali "maximum number of octets in an I field", v vsakodnevem radioamaterskem žargonu pa se je ustalilo zelo neustrezno ime PACLEN. Povsem jasno tu ne gre za dolžino paketa, pač pa za dolžino okvirja in še to le podatkovnega dela okvirja. Ustreznejši izraz bi zato bil "FRAMLEN".

Omejitev "N1" na največ 256 je razumljiva, če upoštevamo količino

WBFM radijske postaje običajno zadošča TXDELAY okoli 20ms in to skoraj ne glede na hitrost 19k2, 38k4 ali 76k8. Megabitne PSK radijske postaje zahtevajo minimalni TXDELAY okoli 2ms, zaradi varnosti pa trenutno uporabljamo večjo vrednost 5ms.

AX.25 paket sicer vsebuje mrtve čase oziroma polnilna polja tudi med posameznimi okvirji in na koncu samega paketa (TXTAIL). Sam opis protokola sicer zahteva le eno samo zastavico (flag) med dvema okvirjema, kar pa pri višjih hitrostih prenosa ne zadošča za programsko opremo, ki mora obdelati sprejeti okvir še pred prihodom naslednjega. Še bolj neroden je rep oddaje (TXTAIL). Ta mrtvi čas je potreben zato, ker programska oprema običajno ne ve, kdaj je strojna oprema zaključila s koristno oddajo in izpraznila vse vmesne pomnilnike.

Celotna koristna vsebina vsakega AX.25 okvirja je vedno zaščitena s CRCjem (Cyclic Redundancy Check), ki se odda na koncu vsakega okvirja. CRC uporabljamo kot detektor napak pri prenosu. Sprejemnik vedno preverja CRC vseh sprejetih okvirjev in vedno zavrne vse napačno sprejete okvirje. V protokolu AX.25 uporabljamo zaenkrat 16-bitni CRC, profesionalni protokoli (Ethernet) pa so že prešli na 32-bitni CRC, kar omogoča večjo zanesljivost ugotavljanja napak pri prenosu.

CRC se izračuna kot ostanek pri verižnem deljenju polinomov z binarnimi koeficienti. Pri tem predstavljajo deljenec kar koristni podatki okvirja,

delitelj pa je skrbno izbran polinom, da zazna čimveč vzorcev napak. Napako torej spregledamo le v slučaju, ko je deljiva s polinomom delilcem.

Polinom delitelj je zato standardiziran. V protokolu AX.25 uporabljamo CCITT (SDLC) polinom, kot je to prikazano na sliki 5. Polinomsko deljenje uporabljamo zato, ker lahko sicer komplicirano matematično operacijo izvedemo s preprostimi pomikalnimi registri in EXOR vrati. Polinom delitelj 16. reda zahteva 16-bitni pomikalni register. Navidez komplicirana matematika se torej pretvori v enostavno električno vezje.

Pred začetkom oddaje oziroma sprejema postavimo pomikalni register v stanje samih enic. Nato oddane oziroma sprejete podatke preprosto vstavljamo v pomikalni register. Za ustrezne matematične operacije poskrbi povratna vezava z EXOR vrati, ki opravljajo nalogo binarnega seštevanja.

Na koncu oddaje koristne vsebine okvirja preklapimo vhod pomikalnega registra na logično ničlo, izhod pa vodimo preko inverterja v oddajnik za nadaljnjih 16 taktov. Na sprejemni strani obdelamo CRC povsem enako kot koristno vsebino okvirja. V primeru brezhibnega sprejema se invertirani CRC natančno odšteje od izračunane vsebine v pomikalnem registru in na koncu dobimo v pomikalnem registru fiksni vzorec \$F0B8.

CRC se sicer da izračunati programsko, vendar to nalogo v večini slučajev opravlja HDLC (X.25) zaporedni vmesnik, ki vsebuje ustrezne

pomnilnika, s katero so razpolagali radioamaterski mikroračunalniki pred dvemi desetletji. Danes predstavlja omejitve "N1" na 256 največjo oviro pri gradnji hitrejših omrežij, saj mrtvih časov, ki se seštevajo v TX-DELAY, ne moremo poljubno manjšati.

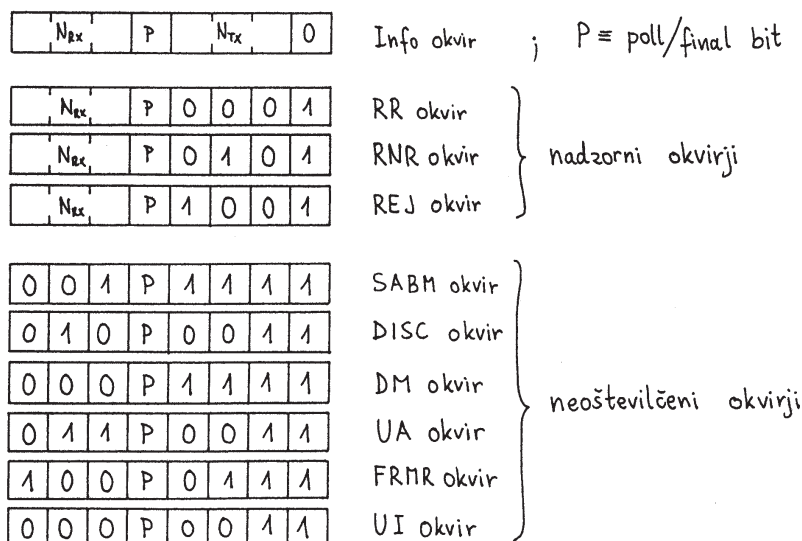
Vsebina naslovnega polja je prikazana na sliki 6. Naslovno polje vsebuje najmanj dva in največ deset radioamaterskih klicnih znakov. Dolžina naslovnega polja je označena na zelo potraten način z najnižjim bitom, ki je postavljen na "1" v zadnjem bajtu naslovnega polja, v vseh ostalih bajtih naslovnega polja pa mora biti postavljen na "0".

Vsak klicni znak je zapisan v sedmih bajtih. Prvih šest bajtov vsebuje običajen radioamaterski klicni znak dolžine do šest črk ali številke. Črke in številke so zapisane v ASCII kodi v gornjih sedem bitov, ker se najnižji bit uporablja za označevanje konca naslovnega polja. Pri klicnih znakih z manj kot 6 črkami/številkami se neuporabljeni bajti zapolnijo z ASCII znakom za presledek.

Zadnji, sedmi bajt klicnega znaka vsebuje več različnih informacij. Štirje biti so namenjeni SSIDju (Secondary Station Identification). SSID je v bistvu dodatek k klicnemu znaku, da se isti klicni znak lahko pojavi na več različnih radijskih postajah istočasno z različnimi SSIDji brez medsebojnih motenj. Štirje biti sicer omogočajo 16 različnih SSIDjev od -0 do -15. SSID -15 je bil v začetku mišljen kot "univerzalni" SSID, na katerega se klicana postaja vedno odzove ne glede na svoj trenutni SSID. Pozneje so to opustili, kljub temu pa stara programska oprema pogosto povzroča težave s SSID -15.

SSID je bil v začetku namenjen možnosti postavljanja lastnega repertitorja, BBSja ipd. Danes se SSID v glavnem uporablja pri posredovanju radijske zveze preko vozlišč. Vsa znana vozlišča (TheNet, KaNode, Flexnet, SuperVozelj, Itanet) uporabljajo SSID v ta namen, zato moramo za pravilno uporabo vozlišč vedno prebrati navodila, na kakšen način določeno vozlišče uporablja oziroma spreminja naš SSID. Končno lahko s smotro uporabo SSIDjev vzpostavimo več vzporednih radijskih zvez, kar se pred dvemi desetletji v začetku packet radia nihče niti pomisliti ni drznil.

Sedmi bajt vsebuje dva neizkoriščena bita, ki naj bi bila (1) vedno postavljena na logično enico. Res-



Slika 7 - Vsebina kontrolnega bajta in vrste okvirjev.

ničnost je drugačna in programerji pogosto pozabijo na ta dva bita, ki imata potem naključno vrednost. Tudi v obratni smeri delovanje programa na sprejemu ni zagotovljeno, če sta ta dva bita naključno postavljena. Iskanje takšne napake je težavno, ker večina programske opreme sploh ne prikaže vsebine teh dveh bitov v monitorskem načinu. Končno, nekatere izvedbe DAMA protokola uporabljajo ta dva bita, da določijo upravno (master) in podrejene (slave) postaje.

Naslovno polje vedno vsebuje na prvem mestu klicni znak naslovnika, ki mu sledi klicni znak pošiljalca. Nadalje naslovno polje lahko (ne nujno) vsebuje do osem klicnih znakov digijev, to je postaj, ki jih uporabimo kot enostavne repertitorje okvirjev. Kljub temu, da je prenos preko digijev neučinkovit, ker se sprejem okvirjev ne potrjuje na vsakem odseku poti posebej, so digiji še vedno dobrodošel dodatek protokolu AX.25. Digi način delovanja predvsem pozna prav vsa programska oprema.

Najvišji bit sedmega bajta naslova se pri tem uporablja kot digi bit. Okvir začne svojo pot z vsemi digi biti na "0", posamezni digiji pa svoje bite postavljajo na "1". Na ta način preprečimo, da bi isti okvir večkrat potoval skozi iste digije.

Kljub temu, da klicni znaki digijev predstavljajo del naslovnega polja, jih različna programska oprema različno uporablja. Nekateri programi smatrajo klicne znake digijev za del na-

slova, drugi pa samo kot pripomoček pri vzpostavljanju zveze. Pri prvi skupini programov lahko vzpostavimo dve ločeni radijski zvezi z enakima klicnima znakoma pošiljalca in naslovnika (z enakima SSIDjema) preko različnih digijev, druga skupina programov pa to obravnava kot eno samo radijsko zvezo, ki preskakuje iz ene verige digijev na drugo.

Najvišji bit sedmega bajta naslovnika in pošiljalca se uporablja v povsem drugačne namene. Ta dva bita (eden pri naslovniku in drugi pri pošiljalcu) določata vrsto AX.25 protokola. Če sta oba bita postavljena na "0" ali oba na "1", to določa izvirno različico V1 protokola AX.25. Če pa sta bita različna, je s tem mišljen protokol AX.25 V2. V protokolu V2 ta dva bita tudi določata smer okvirja: ukaz ali odgovor C/R (Command/Response).

Izumitelji različice V2 so s to potezo naredili obilico težav vsem pisem programske opreme, saj informacija o smeri okvirja sodi v kontrolni bajt in nikakor ne med naslove. Navsezadnje, informacija C/R bitov potrebujejo le trije okvirji v različici V2: RR, RNR in REJ. Smer vseh ostalih okvirjev je enoveljavno določena s kontrolnim bajtom, zato C/R bitov ne potrebujejo.

5. Vrste AX.25 okvirjev

Vrsta AX.25 okvirja je določena z vsebino kontrolnega bajta, kot je to

prikazano na sliki 7. AX.25 zvezo vzpostavljamo, podiramo in resetiramo z neoštevilčenimi (unnumbered) okvirji. Zahtevo za vzpostavljanje zveze sporočimo z okvirjem SABM (Set Asynchronous Balanced Mode), ki hkrati resetira vse števec v obeh udeležencih radijske zveze. Odgovor na SABM okvir je lahko UA (Unnumbered Acknowledge) v slučaju uspešne vzpostavitve ali pa DM (Disconnected Mode), ko je druga postaja zasedena (BUSY) in z nami ne želi vzpostaviti zveze.

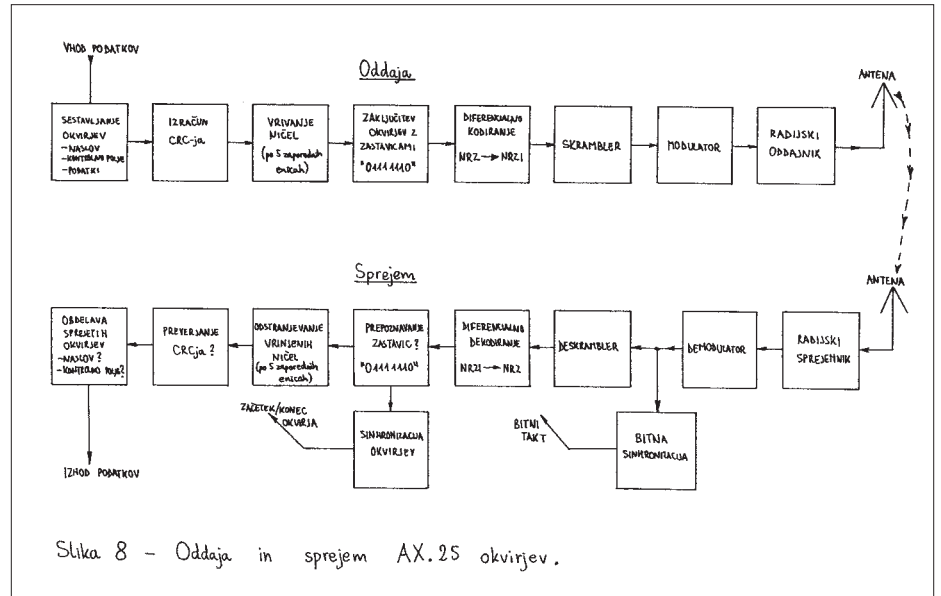
Vzpostavljeno zvezo lahko podre katerikoli od obeh udeležencev z okvirjem DISC (DISConnect). Odgovor na DISC je spet UA okvir oziroma DM, če je druga postaja medtem zvezo že podrla. Neoštevilčena potrditev UA je povsem enaka za SABM kot za DISC okvir, kar je lahko izvor težav pri hitrem zaporednem vzpostavljanju in podiranju zvez.

Podatke, za katere ne zahtevamo potrditve sprejema (beacon) oddajamo v UI (Unnumbered Information) okvirjih. Nekateri protokoli (na primer TCPIP) sicer uporabljajo UI okvirje le kot zunanje ohišje, da lahko izkoriščajo standardno AX.25 omrežje. V notranjosti podatkovnega polja UI okvirja seveda prenašajo povsem svoj protokol z drugačnimi naslovi, kontrolnimi polji in podatki.

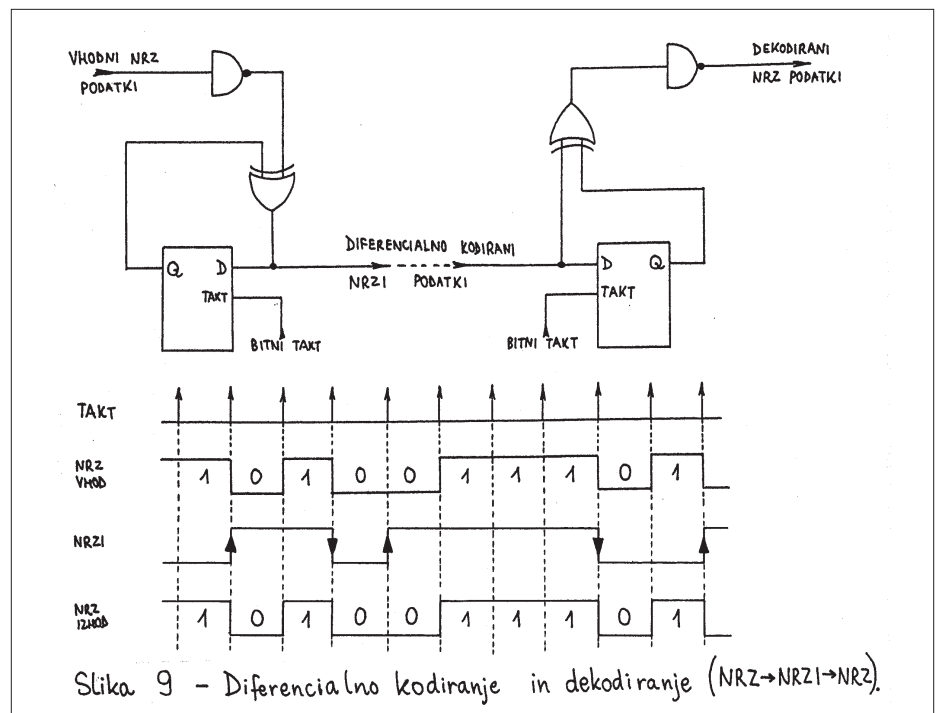
Posebnost je FRMR okvir, ki označuje napako v protokolu. Kontrolnemu bajtu FRMR okvirja sledi kratko podatkovno polje, ki naj bi opisovalo napako. Odgovor na FRMR mora biti v vsakem slučaju SABM, torej ponovno vzpostavljanje zveze in resetiranje števec. FRMR okvir se v praksi skoraj nikoli ne pojavi in nekateri programi ga sploh nikoli ne oddajo. FRMR okvirja zato ne smemo zamešati z REJ okvirjem, ki se običajno pojavlja v vseh AX.25 zvezah.

Ko je AX.25 zveza vzpostavljena, lahko začnemo s prenosom koristne informacije v I (Information-transfer) okvirjih. Kontrolni bajt I okvirja vsebuje sprejemni in oddajni števec. Oddajni števec nosi zaporedno številko I okvirja, sprejemni števec pa potrjuje sprejem v obratni smeri. Vsak števec ima le tri bite, zato lahko šteje od 0 do 7, okvirju številka 7 pa sledi spet okvir z zaporedno številko 0.

Če v obratni smeri ni prometa, lahko sprejem potrdimo tudi z nadzornimi (supervisory) okvirji RR (Receive Ready), RNR (Receive Not Ready) ali REJ (REject). RR okvir potrjuje sprejem in nosi številko naslednjega pričakovanega info okvirja.



Slika 8 - Oddaja in sprejem AX.25 okvirjev.



Slika 9 - Diferencialno kodiranje in dekodiranje (NRZ → NRZI → NRZ).

RNR in REJ okvirja sta bolj nevarna, saj sporočata, da gre nekaj narobe.

RNR odgovor sicer potrjuje pravičen sprejem, vendar je sprejemnik okvir zavrgel, ker je trenutno prezaseden in ga ne more uporabiti oziroma posredovati naprej. REJ odgovor pomeni, da okvirji niso bili sprejeti v vrstnem redu, ki ga določajo zaporedne številke in jih je zato sprejemnik zavrgel. Oba, RNR in REJ, seveda nosita zaporedno številko naslednjega pričakovanega I okvirja enako kot RR.

En sam AX.25 paket lahko vsebuje več zaporednih I okvirjev za istega naslovnika. Največje dopustno število nepotrjenih I okvirjev (maximum number of I frames outstanding) "k" ali MAXFRAME je omejeno na se-

dem oziroma eden manj od zmogljivosti 3-bitnih števec. MAXFRAME pogosto nastavimo tudi manj, ko so pogoji zveze slabi.

Pri sprejemu zaporedja (paketa) okvirjev ni potrebno potrjevati sprejema vsakega I okvirja posebej. Zadošča potrdilo sprejema zadnjega okvirja. Po uspešnem sprejemu I okvirja se zato splača počakati z odgovorom za "T2" (response delay timer) ali RESPTIME, če bi medtem prispel še kakšen okvir. Konstanta "T2" mora biti zato daljša od najdaljšega pričakovanega okvirja. Smiselna RESP-TIME je dve sekundi pri 1200bps in sorazmerno manj na višjih hitrostih.

Če pošiljatelj SABM, DISC ali I okvirja ne dobi nobenega odgovora, bo čez čas "T1" (acknowledgement

timer) ali FRACK poskusil znova. Število zaporednih neuspešnih poskusov je omejeno z "N2" (maximum number of retries) ali RETRY. Dokument (1) predpisuje fiksen, nespremenljiv čas "T1", kar pa se v praksi ni obneslo. Vsa sodobna programska oprema prilagaja FRACK pogojem zveze. Prve ponovitve naj bojo hitre, po velikem številu zaporednih ponavljanj pa je smiselno čas "T1" povečati.

V marsikaterem programu je FRACK nastavljen premajhen, ker marsikdo upa, da bo tako pohitril zvezo. Če je FRACK krajši od najdaljšega možnega paketa, bo takšna postaja proizvajala ogromno motenj sebi in drugim postajam, saj bo začela ponavljati sporočilo, še preden ga je v celoti oddala. Na primer, paket iz sedmih I okvirjev lahko traja 15 sekund pri hitrosti 1200bps, v marsikaterem programu pa najdemo FRACK nastavljen na samo 3 ali 4 sekunde.

Števec "N2" ali RETRY običajno nastavimo čimveč in rajši ročno prekinemo zvezo, ko vidimo, da naša škatla neuspešno ponavlja. Če ima naša programska oprema vgrajen prilagodljiv FRACK, je smiselno število za RETRY nekje med 20 in 30. Žal imajo stari programi fiksen FRACK, RETRY pa je v samem programu omejen na 15, tako da se z njimi celo na 1200bps zveza dokaj hitro poruši.

Opisani okvirji in postopki v grobem ustrezajo izvornemu AX.25 protokolu V1. Novi protokol AX.25 V2, uveden leta 1984 z dokumentom (1), prinaša celo vrsto dodatkov, ki se v glavnem niso obnesli in predstavljajo danes resno oviro razvoju hitrejšega paketa. Na kratkih valovih še danes radioamaterji uporabljamo izvorno inačico V1, ker je v slučaju motenj in nizkih hitrosti prenosa veliko bolj učinkovita od V2, če si seveda ne moremo privoščiti CLOVER ali PACTOR-2 modema. Na UKV smo se po drugi strani nekako sprijaznili z neučinkovitostjo V2 in poskusili nadomestiti izgubljeno s hitrejšimi modemi in radijskimi postajami.

Novosti AX.25 V2 so RR-ukaz (RNR, REJ) ter poll/final bit. V protokolu AX.25 V2 imamo torej dva povsem različna RR okvirja: RR-ukaz in RR-odgovor, ki ju ločimo med sabo s pomočjo C/R bitov v naslovnem polju. RR-odgovor se uporablja povsem enako kot v V1, RR-ukaz pa ponavljamo namesto I okvirjev, ko naša postaja ne sprejme potrditve. Na RR-ukaz odgovori druga postaja z

RR-odgovor povsem enako kot na I okvir.

Osnovna predpostavka protokola V2 je v tem, da se pogosto izgubi RR-odgovor in ne I okvirji. V tem slučaju je bolj smiselno povprašati drugo stran za stanje s kratkim RR-ukaz okvirjem, kot pa ponavljati oddajo I okvirjev. Na žičnih zvezah je takšna predpostavka povsem pravilna. Na radijskih zvezah se je predpostavka izkazala povsem napačna zaradi velikih zakasnitev pri preklopu na oddajo in sinhronizaciji sprejemnika na drugem koncu zveze.

Tudi poll/final bit je v protokolu AX.25 V2 slabo definiran. Postavljeni poll bit v ukazu zahteva takojšen odgovor s postavljenim final bitom. Pri tem imajo okvirji SABM, DISC in RR-ukaz vedno poll bit postavljen na "1". I in UI okvirji imajo poll bit ponavadi izključen na "0". Odgovori morajo seveda vsebovati enak final bit skladno z okvirjem, na katerega odgovarjajo. Poll in final biti so v monitorskem izpisu večine programov označeni z znaki plus in minus takoj za vrsto okvirja.

Neučinkovitost AX.25 V2 hitro opazimo v monitorskem izpisu programa. Tam vidimo naš klicni znak in množico RR-poll (RR+) ukazov in RR-final (RR-) odgovorov, le pretoka koristnih podatkov ni od nikjer. Zveza visi in se sploh ne podre, ker RETRY števec ne šteje ponavljanj I okvirjev, pač pa ponavljanja RR-poll okvirjev.

V slučaju RNR odgovora predvideva protokol AX.25 V2 dve možni rešitvi. Pošiljatelj lahko čez čas "T3" (inactive link timer) ali LINKTIME preveri z RR-poll stanje prejemnika. Po drugi strani lahko prejemnik obvesti pošiljatelja s svojim RR-poll, da se mu je pomnilnik sprostil. Če prejemnik ne zna (AX.25 V1 ne pozna RR-poll), ne more (vozlišče ne pozna cele radijske poti po omrežju) ali pozabi (pomankljivo napisana programska oprema) obvestiti pošiljatelja in ima pošiljatelj nastavljen "T3" na 30 minut, kot je to lahkomišlno predlagano v (4), se bo med čakanjem pošiljatelju brada devetkrat ovila okoli mize.

"T3" ali LINKTIME je v večini programov nastavljen na tri minute (180s), kar je povsem smiselna vrednost za 1200bps. Pri delu na višjih hitrostih je treba seveda "T3" sorazmerno zmanjšati. Žal se v večini programov LINKTIME nastavlja s korakom ene minute, kar je še sprejemljivo za 9600bps, toda povsem neuporabno na višjih hitrostih. LINK-

TIME ali "T3" naj bi bil približno desetkratna vrednost FRACK.

V sedanjem slovenskem packet-radio omrežju se kaj lahko zgodi, da je na enem koncu zveze 1200bps Baycom modem in na drugem megabitna postaja. V takšnih slučajih velikih preskokov hitrosti se kaj lahko zgodijo dogodki, ki jih pisci programske opreme niso predvideli. En takšen dogodek je sprejem zaporedja REJ okvirjev, ki sam po sebi sicer ne predstavlja kršitve protokola, vendar nekateri (predvsem nemški) programi pri tem preprosto po-bezljajo.

Glavna pomanjklivost vseh inačic AX.25 sicer ni nikjer opisana: kako s primernejšim protokolom rešiti problem skrite postaje? Rešitev je v upočasnitvi oddaje: po uspešni oddaji I okvirjev naj pošiljatelj počaka naključen čas, preden poskusi z oddajo novih I okvirjev. Edino na ta način ima šibkejša skrita postaja možnost, da sploh pride do vozlišča. Smotna upočasnitev oddaje lahko tudi prepreči prepogoste RNR okvirje in s tem nepotrebno zasedanje pomnilnika na vozliščih.

Žal upočasnitve oddaje ne podpira prav noben uporabniški program za packet radio. Na vseh vozliščih SuperVozelj pa smo takšno namerno upočasnitev vgradili prav na vseh kanalih. S tem smo si sicer nakopali jezo nekaterih 1200bps uporabnikov, češ da so vozlišča zdaj še počasnejša. Po drugi strani pa opisani ukrep omogoča, da 5 do 10 uporabnikov na enem samem vstopnem kanalu vozlišča istočasno "molze" BBS in se jim pri tem zveze ne rušijo, kar si na osojni strani Alp brez DAMA niti zamisliti ne morejo.

6. Oddaja in sprejem AX.25 okvirjev

Oddajo in sprejem AX.25 sem delno opisal že v prejšnjih odstavkih: vsebino okvirja, izračun in preverjanje CRCja ter sestavo paketov. Celoten postopek oddaje in sprejema okvirjev vsebuje še vrsto drugih postopkov, ki so prikazani na sliki 8 vključno z radijskim oddajnikom in sprejemnikom.

Na oddajni strani moramo najprej sestaviti vsebino okvirja: naslovno polje, kontrolni bajt in podatke. Ko je okvir dokončno sestavljen, lahko izračunamo CRC. Celoten okvir s CRCjem moramo zdaj zapakirati v HDLC obliko.

HDLC protokol uporablja okvirje kot osnovno enoto informacije. Okvirji so med sabo ločeni z zastavicami, ki jih predstavlja vzorec 01111110, se pravi šest zaporednih enic, ki so spredaj in zadaj zaključene z ničlo. Povsem jasno moramo zagotoviti, da se takšen vzorec ne pojavi nikjer sredi okvirja ali CRCja, sicer sprejemnik ne bi mogel določiti mej okvirja.

Da preprečimo šest ali več zaporednih enic v okvirju, v celoten okvir s CRCjem vred najprej vrivamo ničle. Ničlo vrinemo po petih zaporednih enicah v izvornem sporočilu ne glede na to, ali je naslednji znak v sporočilu enica ali ničla. Vrinjene ničle zagotovijo, da se vzorec zastavice ne more pojaviti znotraj okvirja ali CRCja.

Okvir s CRCjem in vrinjenimi ničlami zaključimo spredaj in zadaj z vzorcem zastavice. Zastavice običajno uporabimo tudi za sinhronizacijsko glavo (TXDELAY) in za zapolnitev mrtvih časov med okvirji in na koncu oddaje (TXTAIL). Za vse te "polnilne" zastavice poskrbi že sam hardverski zaporedni HDLC vmesnik.

Celotni signal, se pravi vse okvirje s CRCji, vrinjenimi ničlami in zastavicami vred nato diferencialno kodiramo. Ker oznake NRZ (Non Return to Zero) in NRZI (Non Return to Zero Invert) radioamaterjem niso najbolj domače, jih bom poskusil tule na kratko razložiti: v NRZ signalu predstavlja logično enico visoki nivo in logično ničlo nizki nivo, v NRZI signalu pa kodiramo logično ničlo kot preskok nivoja in logično enico kot nespremenjen, visok ali nizek nivo.

Postopek diferencialnega kodiranja in dekodiranja je prikazan na sliki 9. Tako koder kot dekodeer potrebujeta le po en D-flip-flop in po ena EXOR vrata. Če je prenos brez napak, je

vsebina obeh D-flip-flop-ov ista in dvakratna EXOR operacija z vsebino D-flip-flop-ov da na koncu nazaj izvorni signal.

Diferencialno kodiranje je nujno potrebno, da na sprejemni strani zanesljivo regeneriramo bitni takt. Zaradi vrivanja ničel AX.25 oddaja nikoli ne vsebuje več kot šest zaporednih enic (zastavica), pač pa lahko vsebuje dolga zaporedja ničel. V NRZ pomenijo dolga zaporedja ničel konstanten logični nivo brez preskokov, ki so potrebni za sinhronizacijo sprejemnika. V NRZI pa se ničle preslikajo v preskoke nivojev, zaporednih enic pa je največ šest, kar pomeni največ sedem taktov razmaka med dvema preskokoma logičnega nivoja.

Starejši modemi in TNCji za packet radio sicer vsebujejo mostičke, s katerimi lahko preklaplamo NRZ/NRZI. Mostičke moramo seveda vedno pustiti v položaju NRZI. Na obojni strani Alp nekateri o tem nočejo razmišljati in uporabljajo kar NRZ oddajo brez diferencialnega kodiranja. Skozi takšno omrežje se ASCII tekst sicer prebije, binarni podatki z dolgimi zaporedji ničel pa ne pridejo nikoli skozi. Od tod potreba po čudnem "7+" protokolu, ki ne uporablja ASCII kontrolnih znakov, ker vsebujejo veliko ničel...

Diferencialnemu kodiranju lahko sledi skrambliranje podatkov, ki pa ni vedno nujno potrebno. V našem omrežju ga ne uporabljamo pri hitrostih pod 100kbps. Skrambliranje tudi ne smemo zamešati s kriptozoščito, čeprav je končni rezultat podoben: naključno premešane enice in ničle v sporočilu. Algoritem skrambliranja je vedno objavljen in njegova edina naloga je izboljšanje lastnosti modemov in radijskih postaj, ne pa preprečevanje prisluškovanja.

Enostaven algoritem skrambliranja, ki se uporablja v K9NG in G3RUH modemih ter v naših PSK zvezah je prikazan na sliki 10. Skrambliranje je pravzaprav podobno diferencialnemu kodiranju, le da D-flip-flop nadomešča sedemnajst-stopenjski pomikalni register. Primerna povratna vezava z EXOR vrati poskrbi zato, da so izhodni podatki res dobro premešani. Dobro premešane enice in ničle zagotavljajo majhno enosmerno komponento signala, odsotnost ponavljajočih se vzorcev in veliko prehodov logičnega nivoja, kar vse poenostavlja gradnjo in pospešuje sinhronizacijo sprejemnika.

Diferencialno kodirani (in skramblirani) signal vodimo na modulator in potem na oddajnik. Analogni izhod sprejemnika pretvori demodulator v zaporedni niz bitov, ki mu moramo najprej določiti takt. Bitni takt potrebuje deskrambler, diferencialni dekodeer in vse nadaljnje operacije do zaporedno/vzporedne pretvorbe.

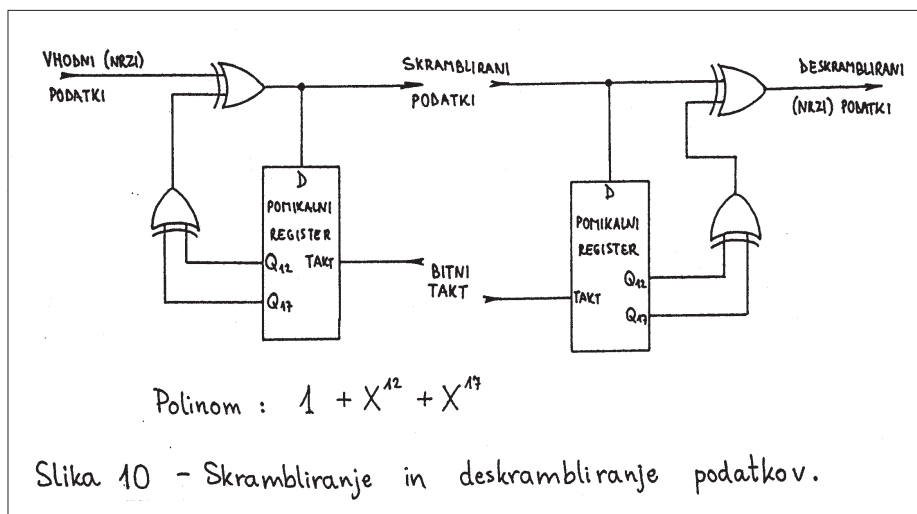
Po diferencialnem dekodiranju moramo v zaporednem nizu bitov najprej poiskati zastavice, ki omejujejo okvirje. Iz vsebine okvirjev nato odstranimo vrinjene ničle in preverimo CRC. Samo pravilno sprejeti okvirji z veljavnim CRCjem grejo v nadaljnjo obdelavo, kjer najprej preverimo naslov. Na koncu iz pravilno naslovljenih okvirjev izluščimo kontrolno polje in uporabimo sprejete podatke.

7. Modemi in radijske postaje

Radioamaterji smo pravzaprav največ novega pričakovali ravno na področju razvoja novih modemov in radijskih postaj. Izumitelji paketa na drugi strani velike luže so že od vsega začetka trdili, da so preprosti AFSK modemi za 1200bps le začasna rešitev. Izgleda so imeli prav, saj italijanski pregovor pravi: "Začasna rešitev traja največ časa!"

Razvijalci packet radia so pozabili, da so postavili za izbiro modulacije štiri povsem nezdržljive zahteve, ki so prikazane na sliki 11. Nobena vrsta modulacije ne more zadostiti niti dvem od navedenih zahtev hkrati: (1) močnostna učinkovitost, (2) spektralna učinkovitost, (3) hitrost preklopa RX/TX in nazaj ter (4) enostavnost modemov in radijskih postaj.

Močnostno učinkovita modulacija pomeni, da z omejeno močjo od-



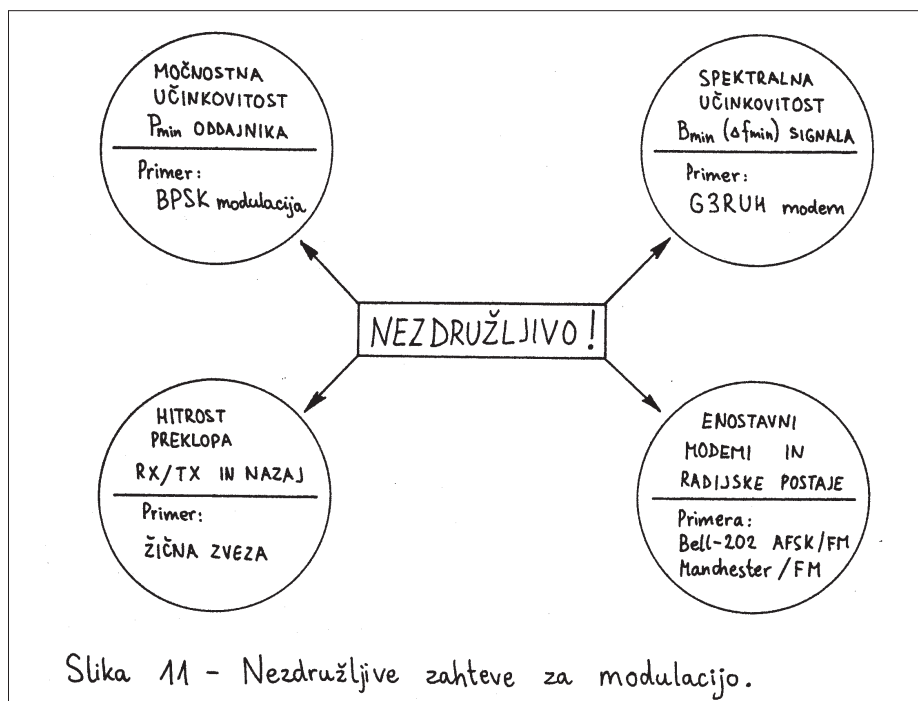
dajnika dosežemo največji domet radijske zveze. Močnostno učinkovite modулacije zahtevajo koherentne detektorje, na primer različne oblike PSK modулacije. Koherentni detektor je komplicirano vezje (ne ustreza zahtevi 4), ki rabi dosti časa za sinhronizacijo (ne ustreza zahtevi 3). Visok izkoristek oddajnika pomeni "kosmat" spekter, kar je v nasprotju z zahtevo 2. PSK (Phase Shift Keying) modулacijo zato uporabljamo na satelitih in v mikrovalovnih področjih, kjer je pasovne širine dovolj, a so moči oddajnikov zelo omejene.

Spektralno učinkovita modулacija pomeni uporabo najmanjše možne pasovne širine. Spet potrebujemo kompliciran modem (ne ustreza zahtevi 4), ki prenaša celoten nizkofrekvenčni pas od enosmerne naprej, kar zahteva dolge preklopne čase (ne ustreza zahtevi 3). Modулacija je občutljiva na motnje in popačenja, kar zahteva močnejši oddajnik (ne ustreza zahtevi 1), kot to dobro poznajo uporabniki G3RUH 9600bps FSK modemov za packet.

Vse do danes ni še nihče uporabil modулacije, ki bi omogočala resnično hiter preklon sprejem/oddaja in hkrati hitro sinhronizacijo sprejemnika na drugem koncu zveze. Problem packet radio omrežja je v tem, da je to omrežje z naključnim dostopom: iz sprejema prejšnjega paketa ne moremo sklepati na sinhronizacijo naslednjega. Vsi profesionalni TDMA sistemi so sinhroni (od TOR na kratkih valovih do GSM telefonov), ki dosežejo sinhronizacijo šele po sprejetju več zaporednih paketov.

Radioamaterji navsezadnje še najraje uporabljamo enostavne modeme, ki jih priključimo na enostavne radijske postaje, kot na primer Bell-202 AFSK modem ali Manchester modem. Takšne modулacije niso niti močnostno učinkovite (ne ustrezajo 1) niti spektralno učinkovite (ne ustrezajo 2), pa tudi čas preklopa sprejem/oddaja in nazaj ni ravno kratek (ne ustrezajo 3).

Najbolj običajne vrste modулacij za amaterski packet radio so prikazane na sliki 12. Radioamaterji pogosto uporabljamo kombinacijo več vrst modулacij. FSK izhod iz modema na primer peljemo na mikrofonski vhod FM oddajnika in dobimo dvojno modулacijo, ki jo imenujemo AFSK/FM, kjer pomeni kratica AFSK Audio Frequency Shift Keying. Tudi Manchester kodiranje pogosto uporabljamo skupaj s FM oddajniki



(Manchester/FM v slovenskem packet radio omrežju) oziroma s PSK oddajniki (Manchester/PSK telemetrija profesionalnih satelitov, pa tudi AO-10 in AO-13).

8. Kako naprej s packet radiom?

Razvoj packet radia se zadnje čase nekam ustavlja. To pa sploh ne pomeni, da je s tem konec razvoja radioamaterskih digitalnih komunikacij. Dokument (1), ki je pred dobrih desetletjem pomenil pospešek razvoja packet radia, saj je poenotil postopek (protokol) AX.25, je izgleda danes glavna ovira nadaljnemu razvoju paketa.

Protokol AX.25, ki je opisan v dokumentu (1), ima nekaj napak, ki so povsem razumljive, če pomislimo na čas, ko je nastajal. Prva napaka je v tem, da so velikosti in dolžine naslovov, okvirjev in paketov premajhne in prirejene računalnikom izpred dveh desetletij. To napako bi zlahka presegli z novimi mejami, če je seveda sploh smiselno postavljati kakšne meje, na primer za dolžino okvirja.

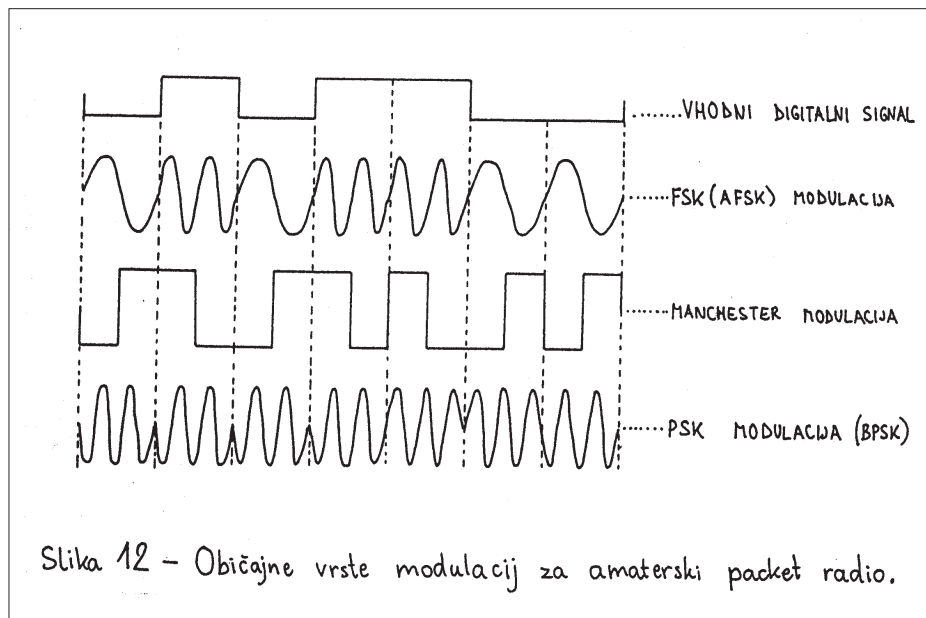
Druga, še dosti hujša napaka pa je v tem, da je sedanji protokol AX.25 preprosto privzet iz žičnih računalniških omrežij. Na žicah gre marsikaj drugače: skritih postaj ni, preklopi sprejem/oddaja so trenutni in oddajo lahko drugi sprejemniki zaznajo takoj brez dolgotrajne sinhronizacije podatkovnega takta. Amaterski protokol za packet radio bi torej morali prilagoditi resničnim

radijskim postajam.

Hiter preklon sprejem/oddaja in nazaj namreč sploh ni nujen pogoj za hiter prenos velikih količin podatkov. To zahtevo smo povsem nepremišljeno privzeli iz žičnih omrežij in se je na radijskih valovih izkazala povsem zgrešena. Sedanje enostavne modeme za packet radio bi bilo pametno zamenjati z močnostno ali spektralno učinkovitimi modemi, nikakor pa postavljati nesmiselne zahteve za modem zaradi slabo načrtovanega protokola, ki ni prirejen radijskemu kanalu.

Neučinkovitost AX.25 V2 smo radioamaterji hitro spoznali. Neskončni ping-pong RR-poll in RR-final brez prenosa podatkov pomeni samo nov izvor QRM na naših prezasedenih frekvenčnih področjih. Na kratkih valovih zato še vedno živi stara inačica AX.25 V1, ki ne pozna RR-poll. Tudi SuperVozlji se med sabo vedno pogovarjajo v AX.25 V1, sicer megabitnih postaj sploh ne bi mogli izkoristiti.

Na megabitnih hitrostih je račun naslednji: TXDELAY in SLOTTIME je nastavljen na 5ms. Pri verjetnosti P-PERSISTANCE 25% to pomeni povprečno čakanje 15ms do začetka oddaje in potem še 5ms sinhronizacijske glave. En sam RR-poll/RR-final ping-pong pomeni skoraj 50ms. V tem času bi pri hitrosti 1.2Mbps lahko oddali več kot 20 I okvirjev največje dovoljene dolžine. Z drugimi besedami ena sama ponovitev okvirja na 1.2Mbps potrebuje zaradi RR-poll/RR-final ping-ponga precej več časa kot oddaja istega okvirja s



Slika 12 - Običajne vrste modulacij za amaterski packet radio.

staro WBFM postajo s hitrostjo 38,4kbps.

Žal na mednarodni ravni vse do danes še ni prišlo do kakršnegakoli dogovora. Za AX.25 so bili predlagani le "kozmetični" popravki, na primer kako vgraditi možnost klicnih znakov, ki imajo več kot 6 črk ali števil. "Žični" teoretiki, ki so skomplicirali AX.25 V2 z RR-poll/RR-final ping-ponganjem, so celo predlagali zamenjavo DCD vezja z novim ping-ponganjem. Na srečo jih ni nihče več poslušal in so te blodne zamisli ostale le na papirju.

Če se na mednarodni ravni nič ne premakne, lahko vsaj sami poskusimo s preprostimi ukrepi povečati učinkovitost protokola. Vsaj dva ukrepa ne bi smela zahtevati večjih predelav programske opreme: povratek k učinkovitejšemu protokolu AX.25 V1 in uporaba daljših okvirjev (večji "N1"). Žal povsod naletimo na težave: ameriški programi za TNCje še nekako prebavijo V1, nemški programi (ki so pri nas zelo razširjeni) pa se z V1 radi zataknejo ali celo sesujejo. Še huje je z daljšimi okvirji: veliko programov za packet se enostavno zatakne ali sesuje, ko sliši daljši okvir, pa čeprav ta okvir sploh

ni bil naslovljen na to postajo. Učinkovitost packet radio omrežja torej predvsem zavisi od naše sposobnosti dogovarjanja na vseh ravneh. Vzdrževalci se morajo med sabo dogovoriti, kako med sabo povezati vozlišča in BBSje, sicer packet radio sploh ne dela. Uporabniki bi se morali dogovoriti z vzdrževalcem svojega lokalnega vozlišča (in ne z nekom na drugem koncu Slovenije), kako urediti dostop do vozlišča. Pri tem je verjetno vzdrževalec tista oseba, ki najbolje pozna vozlišče, dobre in slabe lastnosti njegovih radijskih postaj, usmeritve anten, poznane izvore radijskih motenj in pričakovano pokrivanje posameznih kanalov vozlišča.

Sodobni programi za packet radio vsebujejo celo morje različnih parametrov in v tem morju je pametno poiskati tiste, ki se nanašajo na nastavitve TNCja (ali drugačnega vmesnika) in radijske postaje. Teh parametrov potem ne nastavljamo po enem kopitu, pač pa v skladu z zmogljivostjo svoje postaje, postaj drugih uporabnikov in predvsem v skladu s tistim, kar zahteva radijska postaja v vozlišču na vrhu bližnjega kuclja.

Osnovne časovne parametre packet radio postaje nastavimo v dogovoru z vzdrževalcem vhoda v omrežje, ki ga sami največ uporabljamo: TXDELAY, TXTAIL, SLOTTIME, RESPTIME, RETRY. Nekatere parametre je pametno prilagoditi temu, kar trenutno počnemo na packetu: P-PERSISTENCE, PACLEN, FRACK, MAXFRAME, LINKTIME. Večina teh zadnjih se nanaša samo na našo oddajo in so skoraj nepomembni, če na packetu le ročno tipkamo drugim postajam ali "molzemo" BBS.

Za zaključek poudarjam, da najboljši protokol še ni bil izumljen in verjetno nikoli ne bo izumljen. To pa še ne pomeni, da se pri packet radiu stvari ne da prav nič več izboljšati. Vsaka sprememba protokola je seveda mučna zadeva: zahteva veliko dela, za zahvalo pa na inovatorja takoj zgrmi plaz obtožb, da zdaj nič več ne dela...

9. Reference:

- (1) Terry L. Fox, WB4JFI (urednik ARRL): "AX.25 Amateur Packet-Radio Link-Layer Protocol, Version 2.0", 48 strani, oktober 1984, ARRL, Newington, USA.
- (2) ZRS (več avtorjev): "AMATERSKE DIGITALNE KOMUNIKACIJE", 52 strani, skripta seminarja ZRS "Uporaba računalnikov v radio-amaterski praksi", 21/2/1987, ZRS, Ljubljana.
- (3) ZRS (več avtorjev): "PACKET RADIO", 145 strani, skripta istoimenskega seminarja ZRS, 25/3/1988, ZRS, Ljubljana.
- (4) Andrej Souvent, S51BW: "Parametri packet radio postaje", stran 26/5-97, CQ ZRS, Ljubljana.