

AMATERSKI PACKET-RADIO IN INTERNET

=====

Matjaž Vidmar, S53MV

1. Protokoli za prenos podatkov

Ljudje se lahko sporazumevamo med sabo samo, če govorimo isti jezik. Računalniki niso prav nič na boljšem, le da računalniški jezik za medsebojno sporazumevanje ponavadi imenujemo protokol za prenos podatkov. Tako kot obstaja množica človeških jezikov, obstaja še veliko večja množica računalniških jezikov, prilagojenih najrazličnejšim zahtevam in okusom programerjev.

Med radioamaterji je prav gotovo danes najbolj razširjen protokol za packet-radio, imenovan AX.25. O packet-radiu in protokolu AX.25 smo imeli tudi pri nas kar lepo število predavanj in objavljenega gradiva. Tudi za zamudnike in novopečene radioamaterje je poskrbljeno s knjigami, na primer z zbornikom "Digitalni Mostovi" (evropski projekt Phare, RTV klub Murska Sobota).

Gobezdala, ki se na računalnike čisto nič ne spoznajo, zadnje čase veliko govorijo o internetu kot strašni novosti. Internet je v resnici zelo star protokol, bolj točno zbirka protokolov za prenos podatkov med računalniki, ki so se začeli razvijati pred več kot tremi desetletji na ameriških univerzah na pobudo ameriške ustanove za raziskovanje novih obrambnih (vojaških) tehnologij DARPA (Defense Advanced Research Projects Agency).

Na srečo je DARPA pustila hekerjem dokaj proste roke, da so preizkusili množico najrazličnejših rešitev in se sproti učili na lastnih napakah. Internetni protokol (IP) je na primer načrtovan tudi za slučaj napačnega delovanja ali celo sesutja operacijskega sistema enega ali več računalnikov v omrežju. Najpomembnejša lastnost internetnega protokola pa je verjetno ta, da je bil že od vsega začetka načrtovan tako, da je nadgradljiv v obe smeri: v svoji notranjosti lahko prenaša podatke za protokol na višjem nivoju, sami IP okvirji pa se lahko vgradijo v še večje okvirje protokola na nižjem nivoju, ki je lahko tudi sam zase internetni protokol.

Sestava IP okvirja inačice 4, ki se danes v glavnem uporablja, je prikazana na sliki 1. IP okvir sestavljata glava in podatki. Glava ima najmanj 20 bajtov, sicer pa mora biti dolžina glave celoštevilski mnogokratnik 32 bitov. Takoj na začetku glave je oznaka inačice (4) in števila 32-bitnih besed glave (5 ali več v slučaju opcij), tako da je prvi bajt IP okvirja največkrat heksadecimalno število \$45.

Celotna dolžina IP okvirja, glava in podatki, se meri v bajtih in je zapisana s 16-bitnim številom. Največja dolžina IP okvirja je zato lahko 65535 bajtov, v praksi pa se uporabljajo precej krajši okvirji. Opis internetnega protokola zahteva, da zna vsak udeleženec prenašati in obdelovati

okvirje dolžine vsaj 576 bajtov.

Ob vzpostavitvi zveze se udeleženca seveda lahko dogovorita za medsebojno sporazumevanje z daljšimi okvirji, če to oba zmoreta in želita. Vsak udeleženec sporoči svoj MTU (Maximum Transfer Unit), to je največjo dopustno dolžino okvirja. Zveza potem teče naprej z manjšo od obeh vrednosti MTU. Najpogosteje se uporablja MTU 1500 bajtov, kar dopušča Ethernet vmesnik.

Če IP okvirji naletijo na svoji poti na ozko grlo, ki ne dopušča tako velikega MTU, se veliki okvirji lahko drobijo (fragmentirajo) v več manjših okvirjev. Vsak manjši okvir se potem opremi z lastno IP glavo, ki vsebuje podatke o drobljenju okvirja in točnem položaju drobca v celoti. Vsi ti podatki omogočajo, da se na drugi strani MTU ožine drobci spet sestavijo v prvotni IP okvir.

Pogosta napaka računalniških omrežij je zankanje, ko si zaradi napačnih podatkov o usmerjanju računalniki podajajo v krogu isti okvir. Zankanje rešuje števec življenjske dobe okvirja. IP okvir začne svojo pot z 32 ali 64 "življenji" in pri vsaki obdelavi okvirja izgubi eno "življenje". Ko izgubi vsa "življenja", se okvir zavrže.

Glavni koristni tovor IP glave so naslovi pošiljatelja in naslovnika okvirja. V inačici 4 so naslovi 32-bitna števila, ki jih za človeške uporabnike zapišemo v obliki štirih decimalnih števil med 0 in 255, od katerih vsako predstavlja enega od štirih bajtov 32-bitnega naslova. Decimalna števila ločimo s pikami, 44.150.1.1 je na primer IP naslov strežnika S50LEA.

Pri izbiri števil za naslove veljajo nekatere omejitve. Prvo število na primer ne sme biti večje od 223 in po prvotni delitvi IP naslovov določa tudi velikost omrežja, to je deli 32-bitni IP naslov v naslov omrežja in naslov računalnika. Kot je v računalniških omrežjih v navadi, naslov računalnika s samimi enicami predstavlja krožni poziv (to se je svoj čas uporabljalo celo za SSID 15 v AX.25V1), naslov računalnika s samimi ničlami pa skupni naslov omrežja.

IP glava vsebuje še kontrolno vsoto, ki ščiti vsebino glave okvirja, in opis vrste podatkov, bolj točno višjega protokola, ki ga vsebuje podatkovno polje. Če internet primerjamo s packet-radiom, potem IP glava ustreza klicnim znakom v AX.25 okvirju. IP glavi so lahko dodane še opcije, ki morajo biti v vsakem slučaju zaključene na mejo 32-bitne besede s polnilom.

Ker IP glava vsebuje le naslove, moramo vzpostavljanje in podiranje zveze ter prenos in potrjevanje podatkov zaupati višjemu protokolu. V protokolu AX.25 za packet-radio za to skrbi kontrolni bajt, ki sledi klicnim znakom. V internetu pa lahko podatki, ki sledijo IP glavi, vsebujejo najrazličnejše protokole na višjem nivoju.

V zvezah po internetnem omrežju se najpogosteje uporablja TCP (Transmission Control Protocol). TCP ima lastno glavo dolžine 20 bajtov in podatkovno polje. TCP poskrbi za rušenje in vzpostavljanje zveze, ko sporoči tudi MSS (Maximum Segment Size), to je največjo dolžino podatkovnega polja. MSS je ponavadi nastavljen na 1460 bajtov, kar da skupaj z 20 bajti TCP glave in še 20 bajti IP glave MTU 1500 bajtov.

TCP poskrbi tudi za pravilni vrstni red oddajanja podatkov in potrjevanja sprejema, saj lahko različni IP okvirji iste zveze potujejo po različnih poteh. TCP glava zato vsebuje kazalec na položaj podatkov, ki sledijo glavi, ter kazalec za potrditev sprejema podatkov v obratni smeri. TCP glava hkrati ureja hitrost prenosa s številom RWIN (Receive WINDOW), to je velikostjo sprejemnega okna, preden se pošilja potrditev sprejema.

Končno, TCP glava vsebuje dodatek k naslovu pošiljatelja in naslovnika, dve 16-bitni številki portov. Številki portov omogočata več vzporednih zvez med istima IP naslovoma, podobno kot SSIDji omogočajo več vzporednih zvez med istima klicnima znakoma v AX.25. Številke portov pišemo z decimalnimi števili od 0 do 65535.

Številke portov navedemo le v slučaju, če se razlikujejo od običajnih števil za določeno vrsto usluge. HTTP dostop na primer običajno uporablja port 80, na LJUBBS pa izjemoma 8080. Naslov 44.150.1.200:8080 torej pomeni HTTP dostop do LJUBBS.

TCP glava in podatki, ki ji sledijo, so zaščiteni s 16-bitno kontrolno vsoto v TCP glavi. Enostaven izračun kontrolne vsote tako v IP glavi kot v TCP glavi kaže na zastarelost protokola TCP/IP: CRC polinomske delilnik, ki ga uporabljamo v AX.25 packet-radiu, je veliko bolj zanesljiv detektor napak pri enakem številu kontrolnih bitov.

Razen s TCP se amaterji srečamo še s protokoloma UDP (User Datagram Protocol) in ICMP (Internet Control Message Protocol), ki se prav tako vgrajujeta v IP okvirje. Če naredimo primerjavo s packet-radiom, potem TCP ustreza običajni packet-radio zvezi z vzpostavljanjem zveze, potrjevanjem sprejema in podiranjem zveze, UDP pa ustreza oddaji UI (beacon) okvirjev, ki ne zahtevajo vzpostavljanja zveze niti potrditve sprejema. Končno, ICMP se uporablja za sporočanje izjemnih dogodkov in napak pri prenosu IP okvirjev.

Uporaba TCP/IP je bila dolga leta omejena na povezave med (redkimi) velikimi računalniki, s katerimi so upravljali izključno svečeniki v belih haljah. Radioamaterji smo se sicer zgodaj zagrebeli za celo skupino internetnih naslovov 44.x.x.x, ko IP številke niso zanimale še nikogar. S počasnimi modemi in še počasnejšimi ceneni računalniki pa je dolga leta TCP/IP predstavljal radioamaterjem le kompliciran in neučinkovit način dela v packet-radiu.

Proizvajalci osebnih računalnikov, omrežnih kartic in programske opreme so najprej silili vsak s svojim protokolom za računalniška omrežja, ki je bil seveda nezdržljiv z ostalim svetom in ponavadi tudi poln napak. Zastareli in napol pozabljeni TCP/IP se je naenkrat izkazal kot zanesljiva rešitev, ki jo povrh govoriijo prav vsi računalniki in programska oprema različnih proizvajalcev.

Ceneni računalniki so po drugi strani privedli v strmoglavo povečanje števila uporabnikov TCP/IP protokola. Ker je namen računalniških omrežij prav ta, da omogočajo zvezo med vsemi računalniki na svetu, je začelo primanjkovati prostih IP števil. Proizvajalci programske opreme so si zato izmislili nekaj trikov, kako zmanjšati porabo IP števil, na primer z dinamičnim dodeljevanjem IP števil aktivnim udeležencem v

prometu, ki se po podiranju zveze lahko takoj dodelijo novim udeležencem. Pomanjkanje IP številčk naj bi razrešila nova inačica 6 internetnega protokola, če se bo seveda kdaj uveljavila.

Radioamaterji imamo pri vsem skupaj veliko srečo, da se lahko izvijemo iz vrtinca mračnega dogajanja okoli trženja interneta. Zasnova interneta je bila od vsega začetka pred tridesetimi leti čisto radioamaterska: jaz poskrbim za mojo opremo in ti poskrbiš za tvojo, poveževa pa se zato, ker je to v obojestranskem interesu. Na drugi strani lahko iz razvoja interneta potegnemo radioamaterji zase marsikatero korist, predvsem v obliki uporabe izdelanih protokolov in programske opreme tudi v naših radioamaterskih zvezah.

2. Prenos IP okvirjev po amaterski packet-radio zvezi

Za razliko od profesionalcev smo radioamaterji zelo hitro pogruntali, da je paketni prenos prava stvar za naše radijske zveze. Ker je bil TCP/IP prekompliciran in tudi nekoliko zastarel, smo se pri packet-radiu navsezadnje odločili za enostavnejši protokol AX.25. Izbira določenega protokola na srečo v računalniških omrežjih ni omejujoča: v izbrani protokol lahko v podatkovno polje še vedno vgradimo druge protokole na višjem nivoju.

Vgradnja TCP/IP okvirja v AX.25 okvir je prikazana na sliki 2. Najenostavnejša rešitev je ta, da celoten IP okvir vgradimo v podatkovno polje AX.25 okvirja. Takšen AX.25 okvir označimo s PID bajtom \$CC za razliko od običajnih packet-radio zvez, kjer je PID postavljen na \$F0. Že tu se vidi podobnost med AX.25 in IP: povsem enako kot IP tudi AX.25 dopušča v svoji notranjosti protokol na višjem nivoju, vrsto prenašanih podatkov pa opisuje PID polje.

Ker je TCP/IP sam sposoben vzpostavljati zveze, potrjevati sprejem ter zagotavljati brezhiben prenos podatkov, lahko za prenos IP okvirjev uporabimo kar UI (beacon) okvirje protokola AX.25 s kontrolnim bajtom \$03. Protokol AX.25 služi v tem slučaju le za naslavljanje v packet-radio omrežju, vse ostalo pa počne TCP/IP. Takšnemu načinu vgradnje TCP/IP okvirjev v AX.25 pravimo tudi datagramski (DG) način.

Če je AX.25 zveza slaba in se veliko okvirjev izgubi, postane datagramski način neučinkovit, saj so TCP ponavljanja počasna. TCP ponavljanja so načrtovana za zanesljive žične zveze, kjer so izgubljeni okvirji izjemen dogodek (dupleksne zveze) ali pa za ponavljanja poskrbijo že gonilniki na zelo nizkem (hardverskem) nivoju (Ethernet).

V slučaju slabe AX.25 zveze vgradimo TCP/IP okvirje v Info okvirje AX.25 zveze, kar imenujemo tudi Virtual Circuit (VC) način dela. V slučaju Virtual Circuit moramo seveda poskrbeti za vzpostavljajanje in rušenje AX.25 zveze z običajnimi okvirji SABM, DISC, UA in DM ter potrjevanje sprejema z okvirji RR, RNR in REJ. Če Virtual Circuit deluje pravilno, potem za vsa ponavljanja poskrbi AX.25 in so TCP ponavljanja zelo redka.

Do tu izgleda vse enostavno, če se le AX.25 in IP ne bi stepela pri osnovnem dogovoru o dolžini okvirja. Izvirni AX.25 je nastal v času, ko so imeli amaterjem dostopni računalniki

zelo majhen pomnilnik, kar je bil glavni vzrok za omejitve dolžine podatkovnega polja na 256 bajtov. Na drugi strani IP zahteva možnost prenosa okvirjev brez drobljenja do vsaj 576 bajtov dolžine, običajna vrednost za MTU pa je 1500 bajtov.

V datagramskem načinu dela premagujemo gornjo omejitev na dva načina. Prva, boljša rešitev je ta, da v AX.25 dopustimo tudi daljše okvirje do dolžine vsaj 1500 bajtov, kar za sodobne računalnike ne predstavlja nobene težave. Le kakšen star, nemarno napisan program za packet-radio (TFPCX) se lahko sesuje, ko slučajno sliši predolg okvir, čeprav ni bi njemu naslovljen. V izvirnem opisu AX.25 je namreč predvidena možnost soobstoja drugih protokolov in daljših okvirjev na istem radijskem kanalu. Na predolge okvirje mora AX.25 postaja primerno odgovoriti (okvir FRMR), ko so naslovljeni njej, oziroma jih mora preprosto preslišati, ko so naslovljeni drugam.

Druga, slabša rešitev je ta, da prevelike IP okvirje zdrobimo v manjše (IP fragmentacija) ter vsakemu drobci dodamo IP glavo s podatki o drobljenju. V drugem slučaju seveda prenašamo kup nepotrebne navlake s podatki o drobljenju okvirjev. Prenos in usmerjanje velikega števila majhnih okvirjev predstavlja veliko in povsem nepotrebno obremenitev vseh računalnikov v omrežju. Hkrati ni rečeno, da vsa IP programska oprema podpira drobljenje na manj kot 576 bajtov.

V načinu dela Virtual Circuit imamo še tretjo možnost: drobljenje AX.25 podatkovnega polja (AX.25 fragmentacija), saj tu protokol AX.25 zagotavlja pravilen vrstni red prenosa drobcev ter njihovo sestavljanje v celoto na sprejemni strani. Drobci, od katerih vsak sam zase še ne predstavlja veljavnega IP okvirja, saj nimajo IP glave, se označijo s PID poljem \$08. V načinu dela Virtual Circuit se običajno ne uporabljajo okvirji, ki bi bili daljši od izvirne AX.25 omejitve.

Končno, pri izbiri načina dela (datagram ali VC) ter načina drobljenja (IP ali AX.25) se moramo najprej pozanimati, kaj zmore naša oprema, kaj zmore oprema sogovornika in kaj zmore packet-radio omrežje med nami. Virtual Circuit je vsekakor potratnejši način dela kot pa datagram, kar pomeni večji QRM drugim postajam. Če nam dela VC bistveno boljše kot datagram, je to zanesljiv znak, da z našo radijsko zvezo nekaj ni v redu.

3. Prenos IP okvirjev po omrežju SuperVozljev

Ena sama packet-radio zveza, ki lahko prenaša tudi IP okvirje, danes pomeni zelo malo. Med ostalim dobimo na tržišču cenene "brezžične modeme", to je radijske postaje za ISM frekvenčni pas 2.4GHz, ki po zraku pokrpajo žična Ethernet omrežja nekaj deset km narazen. Ker imamo isti frekvenčni pas dodeljen tudi radioamaterji, lahko opisane "brezžične modeme" kar prekrstimo v amaterske radijske postaje, kar je vsekakor ceneje od razvoja in gradnje posebne opreme za packet-radio.

Posebnost amaterskega packet-radia je omrežje vozljev, ki omogočajo medsebojno zvezo med oddaljenimi postajami preko večjega števila vozljev. Hkrati vozliji omogočajo medsebojne zveze med postajami, ki uporabljajo različne frekvence, modulacije in bitne hitrosti. Omrežje vozljev je tehnično in

organizacijsko preveč zahteven podvig, da bi si ga lahko privoščili med sabo na smrt skregani hekerji, ki so tehnična znanja že zdavnaj zanemarili.

Packet-radio omrežje lahko uporabimo za prenos IP okvirjev na različne načine. Najenostavnejša rešitev je uporaba AX.25 digipitiranja, kjer je domet praktično omejen na dva digija. Pri večjem številu digijev postane verjetnost izgube okvirja tako velika, da zveza ni več uporabna. Način vgradnje IP okvirjev v AX.25 zvezo, datagram ali VC, pri digipitiranju ne igra posebne vloge.

Packet-radio omrežje lahko seveda bistveno boljše izkoristimo, če poznamo njegov način delovanja. RMNC/Flexnet vozlišča se na primer obnašajo kot navidezni digiji, v resnici pa potrjujejo AX.25 okvirje na vsakem odseku posebej. Takšno omrežje je pisano na kožo Virtual Circuit, medtem ko se za UI okvirje (datagram) obnaša povsem enako neučinkovito kot navadni digiji.

Omrežje SuperVozljev se obnaša kot en sam navidezni digi, pri tem pa si posamezni vozlji podajajo okvirje s pomočjo posebnih službenih zvez. Službene zveze so čisto navadne AX.25 zveze, ki se same vzpostavljajo in rušijo glede na zahteve prometa, pri tem pa poskrbijo za zanesljiv prenos okvirjev med vozlji. Omrežje SuperVozljev je zato primerno za datagramski način dela, saj službene zveze poskrbijo, da se okvirji ne morejo izgubljati v omrežju razen na vstopnem in izstopnem odseku.

Virtual Circuit pomeni nepotrebno dvojno potrjevanje v omrežju SuperVozljev. Kar je še slabše, VC potrjevanje teče preko celega omrežja vozljev. Neizbežne zakasnitve pogosto privedejo do rušenja in ponovnega vzpostavljanja AX.25 zveze za Virtual Circuit, kar pomeni še dodaten QRM in nepotrebno zaviranje prenosa podatkov.

Učinkovitost IP zveze preko omrežja SuperVozljev zavisi v veliki meri od pravilne izbire načina prenosa IP okvirjev ter nastavitve nekaj parametrov. Vse te nastavitve lahko opravimo le, če razumemo delovanje navideznega digija oziroma datagramske zveze skozi verigo SuperVozljev ter pripadajoče mešetarjenje z AX.25 klicnimi znaki in SSID-ji, kar je prikazano na sliki 3.

SuperVozelj se obnaša kot digi, če pošljemo okvir via njegov klicni znak (naprimer S59DAY), nikakor pa via IDENT (RAFUT). Kaj bo točno napravil vozelj, zavisi od SSID-ja pri klicnem znaku vozlja. Če je SSID od nič različen, se vozelj obnaša kot čisto navaden digi: prispeli okvir pošlje na kanal, ki ga določa SSID, hkrati pa SSID v klicnem znaku digija zamenja s številko vstopnega kanala, da je omogočena povratna zveza.

Če je SSID pri klicnem znaku vozlja enak nič, bo vozelj pregledal lastno "G" tabelo, kaj naj s takšnim okvirjem stori. Če naslovnika ni v "G" tabeli, se okvir zavrže. Če je naslovnik kdaj prišel neposredno na vozelj, se vozelj obnaša kot čisto navaden digi. Končno, če je v "G" tabeli zapisano, da naslovnik prihaja iz sosednjega SuperVozlja, bo vozelj vzpostavil službeno zvezo s sosedom, mu posredoval digi okvir ter zahteval potrditev sprejema. Z digi okvirjem se potem ukvarja naprej sosed, ki spet pogleda v svojo "G" tabelo, kam

z okvirjem...

S ciljem čimboljšega izkoriščanja radijskega kanala gradijo Supervozlji službene zveze za vsakega pošiljatelja digi okvirjev posebej. V ta namen izkoristijo tudi pošiljateljev klicni znak tako, da mu spremenijo SSID (obrnejo vse štiri bite). Primer na sliki 3 kaže, da digi okvir, ki ga je poslal S53MV-11, gradi za svojo pot službene zveze s klicnim znakom S53MV-4, saj je $15-11=4$. V obratni smeri pošilja digi okvir S50LEA, ki gradi službene zveze s klicnim znakom S50LEA-15, saj je $15-0=15$.

SSID naslovnika službene zveze se izbira enak kot SSID končnega naslovnika. Tako S53MV-4 kliče S55YNG-0, S55YIB-0 in S55YLJ-0, saj je končni naslovnik S50LEA-0. V obratni smeri S50LEA-15 kliče S55YIB-11, S55YNG-11 in S59DAY-11, saj je končni naslovnik S53MV-11. Vse to mešetarjenje s klicnimi znaki in SSID-ji je potrebno, da ne pride do vzpostavljanja in potem štrikanja dvojnih zvez z enakimi klicnimi znaki.

Pri vzpostavljanju IP zvez preko omrežja Supervozljev moramo posvetiti posebno pozornost pravilni izbiri lastnega SSID-ja in SSID-jev vozljev, ki jih hkrati kličemo za navadne AX.25 zveze. Na S50LEA-0 je nesmiselno riniti z lastnim SSID-jem -0, kot to počne na primer S57BLU-0. V tem primeru se pogosto poruši povratna zveza od S50LEA proti S57BLU. Razlog je v tem, da S50LEA tudi sam kliče vozlje s SSID-jem -0 iz drugih razlogov, kar v Supervozljih sproži zaščito proti dvojnimi zvezam.

Pri HTTP dostopu do škatel, ki vzpostavljajo veliko navadnih AX.25 zvez, moramo biti še posebej previdni pri izbiri lastnega SSID-ja. Pri HTTP dostopu do S50BOX-6 se moramo na primer izogibati lastnim SSID-jem -12, -13, -14, in -15, saj s temi SSID-ji S50BOX kliče vozlje za posredovanje sporočil drugim BBSjem, kar pomeni, da je povratna službena zveza za IP okvirje s temi SSID-ji onemogočena.

Pri izbiri SSID-ja je dobro pogledati, kaj počnejo drugi radioamaterji v bližini. Tu v Novi Gorici na primer hkrati rinemo na internet Dolfe kot S52DS-7, Sandi kot S53SM-12 in jaz kot S53MV-11. Različni SSID-ji nam pri tem omogočajo, da se najmanj motimo med sabo, saj ima tako vsak od nas tudi v povratni smeri svojo lastno službeno zvezo iz S50LEA ali S50YFE. Če bi vsi rinili gor z istim SSID-jem, na primer -6, kot to ponekod počnejo po nalogu politkomisarja, bi se po nepotrebnem motili med sabo ter bi se stekli za bornih 30 blokov vmesnega pomnilnika, ki jih Supervozelj dodeli eni sami zvezi.

Če smo hkrati na internetu in na običajnem packetu, potem moramo paziti, da si z običajno packet-radio ne zapremo poti za službene zveze. Če z navadno PR zvezo pokličem S55YNG-0, zaprem pot službeni zvezi proti S50LEA-0 ali S50YFE-0, ki bi prav tako klicala S55YNG-0. Kljub drugačnemu SSID-ju pri klicnem znaku S53MV bi S55YNG to zavrnil kot nevarno dvojno zvezo.

Supervozlji uporabljajo za usmerjanje datagramskih okvirjev "G" tabelo, to je dnevnik zvez na Supervozlju. "G" tabela se torej gradi sama, ko vzpostavljamo ali bolj točno, ko porušimo kakršnokoli AX.25 zvezo, saj šele rušenje zveze sproži vpis zveze v dnevnik. Tudi pri IP delu se "G" tabela sproti

obnavlja, saj se službene zveze prav tako vpišejo v dnevnik, ko se samodejno porušijo.

Dobra in hkrati slaba lastnost poti v "G" tabeli je ta, da jih lahko nastavi vsakdo in ne samo vzdrževalci. To pomeni, da si lahko vsak napelje poljubno neumno pot do svojega IP sogovornika. Na primer, iz Nove Gorice je smiselna pot do S50LEA preko GORICA-BRKINI-LJU, seveda pa deluje tudi GORICA-KUK-CERKNO-CPRST-GOLO-MRZ-IDRIJA-KUM-LJU.

Če si neumen uporabnik sam sebi zafenclja pot do S50LEA, ni to še nič hudega, saj si je svoje nesreče sam kriv. Hujša posledica je to, da se tudi drugim uporabnikom tedaj zveze prestavijo na neumno pot. Najslabši slučaj sta dve poti, ki sta skoraj enakovredni, datagramska zveza lahko začne preskakovati med njima in SuperVozlji v strahu pred dvojnimi zvezami blokirajo obe dobri poti!

Pred uporabo datogramske zveze skozi verigo SuperVozljev je zato pametno preveriti z navadnim programom za packet-radio, kaj počnejo drugi uporabniki in kako so v "G" tabelah vpisane poti do zanimivih IP postaj, na primer S50LEA ali S50BOX-6. Svojo lastno pot do IP sogovornika napeljemo tako, da z navadnim programom za packet pokličemo končni SuperVozelj po najbolj smiselni poti in nato zvezo porušimo.

Vse to nič ne pomaga, če si je naš sosed radioamater za isti cilj izbral drugačno pot: IP zveza bo stalno preskakovala med obema potema in ne bo delala nikomur. Še bolj nujno kot o SSID-jih se moramo v tem slučaju dogovoriti s sosedom o tem, katero pot bomo uporabljali.

4. Amaterski postaja za packet-radio in internet

V prejšnjih odstavkih sem opisal le, kaj se nam ponuja oziroma ka bi radi imeli. Radioamaterji smo seveda praktični ljudje in bi radi ustrezno napravo čimprej tudi sestavili in usposobili za resnično delovanje. Slika 4 prikazuje osnovne gradnike amaterske postaje za packet-radio in internet.

Načrt na sliki 4 je razdeljen na dva dela, radio in računalnik. Na radijski strani vsekakor potrebujemo anteno, primerno radijsko postajo in modem. Za razliko od telefonskih modemov je modem za packet-radio le redko izveden softversko na signalnem procesorju ali z zvočno kartico. Za višje hitrosti prenosa je modem ponavadi kar sestavni del radijske postaje, na primer v megabitnih PSK radijskih postajah, ki jih uporabljamo v našem packet-radio omrežju.

Na računalniški strani je na sliki 4 cela množica škatlic, ki lahko predstavljajo ločene računalnike ali pa samo programe, ki tečejo v večopravilnem operacijskem sistemu na istem računalniku. Seveda smo pred mnogimi leti začeli packet-radio z mnogimi škatlicami: s protokolom AX.25 se je ukvarjal mali računalnik z imenom TNC, terminal pa je bila kar odslužena "PAKA" ali podobna naprava z RS-232 priključkom.

Škatlice so se druga za drugo preselile v notranjost osebnih računalnikov, ki so postali ceneni in dostopni vsem. Osebni računalnik je najprej prevzel vlogo tekstovnega terminala za packet-radio, nato vlogo TNCja (TFPCX, FlexNet) in s prihodom zvočnih kartic celo vlogo modema za nižje bitne hitrosti. Kot zanimivost je programska oprema ostala ločena

in tako še danes dobimo ločen program za terminal, potem program protokol AX.25 in na koncu celo program za modem.

Za delo na internetu radioamaterji ne razvijamo posebnih naprav, saj je namen uporabe interneta prav ta, da dobimo že izdelano programsko opremo za osebne računalnike. Če se namreč gremo sami pisanja programov, potem interneta in njegove zbirke protokolov res ne potrebujemo, saj se dajo povsem iste stvari narediti tudi z našim protokolom AX.25 in to včasih dosti boljše kot z zastarelim TCP/IP. Namen radioamaterske uporabe interneta je samo v tem, da izkoristimo množico že napisanih programov in hkrati poskrbimo za združljivost z ostalim svetom, ki govori le IP in o našem AX.25 nič ne ve.

Združljivost je zelo lepa beseda, ki pa ni enostavno izvedljiva. Pri programih za packet-radio združljivost ni težavna, saj je treba zagotoviti le, da se se program za terminal in program za TNC razumeta med sabo. Ostalo dogajanje na našem računalniku ni pomembno pod pogojem, da se programi za packet-radio ne stepejo z drugimi programi za isti kos pomnilnika ali isti vmesnik.

Pri delu na internetu preko packet-radio omrežja moramo zagotoviti vmesnik, ki bo programe za packet-radio povezal s programi za internet. Ko so proizvajalci programske opreme ugotovili, da njihovi izdelki morajo obvladati IP, so družino internetnih protokolov vgradili v samo jedro operacijskega sistema. Na ta način so po eni strani zagotovili isti vmesnik za vse programe uporabnike interneta in programe ponudnike zvez, po drugi strani pa so si zagotovili monopol, da smo uporabniki zdaj vezani na določene usluge jedra določenega operacijskega sistema.

Med programom za protokol AX.25 in jedrom operacijskega sistema torej potrebujemo vmesnik, ki bo packet-radio zvezo predstavil operacijskemu sistemu kot Ethernet kartico ali telefonski modem. Vmesnik mora tudi poskrbeti za pretvorbo naslovov med AX.25 radioamaterskimi klicnimi znaki na eni strani in IP številkami na drugi strani. Preslikava naslovov ni enostavna, saj lahko po isti packet-radio zvezi dostopamo do različnih IP naslovov.

Pri nalaganju programov moramo naš računalnik najprej usposobiti za packet-radio, to se pravi poskrbeti za povezavo do radijske postaje (TNC ali kartica z ustreznimi kablji) in naložiti program za protokol AX.25 (na primer FlexNet z vsemi krmilniki) in program za terminal (GP, WPP ali kar vam je bolj pri srcu). Delovanje programov za packet-radio natančno preverimo v vseh možnih načinih delovanja: kot opazovalec dogajanja na radijskem kanalu, v AX.25 zvezi, pri AUTOBIN prenosu daljših datotek in ne nazadnje za oddajo čisto navadnih UI (beacon) okvirjev.

Operacijski sistem windows v tem trenutku še nič ne ve o obstoju vmesnika FlexNet za packet-radio omrežje, saj je program za packet le eden od množice programov, ki nekaj meljejo na računalniku. FlexNet moramo zdaj predstaviti operacijskemu sistemu natančno tako, kot če bi vgradili mrežno (Ethernet) kartico. V nadzorni plošči (Control Panel) poklikamo omrežje (Network) in izberemo vgradnjo adapterja. Ker FlexNet ni vgrajen v zbirko gonilnikov v windows, moramo pokazati pot na datoteko IPAX.INF.

V naslednjem koraku moramo v omrežje windows naložiti protokol TCP/IP. To na srečo windows zna sam, izbrati moramo Microsoft, TCP/IP in takoj pobrisati dodatno navlako, ki jo je zmešani windows mogoče naložil v okno omrežja. V oknu omrežja moramo imeti le dva vpisa (če nimamo drugih vmesnikov): en adapter in en protokol. Butasti windows bo po vseh teh zanj strašnih nastavitvah zahteval restart, no pred restartom se splača nastaviti v lastnostih (properties) TCP/IP še IP številko in masko.

Ko smo na računalnik vgradili vse potrebne programe, moramo nastaviti še naslove in teh je kar nekaj na obeh straneh: na IP strani in na AX.25 strani. Na IP strani moramo najprej nastaviti lastno IP številko in masko. Ker je pomen IP številke nekje podoben klicnemu znaku, se moramo odločiti, kako bomo prišli do naše IP številke. Tu obstajajo vsaj tri možne poti:

Prva, najenostavnejša pot je, da si kakršnokoli IP številko preprosto izmišljimo. To vsekakor ni kriminalno dejanje, saj je IP številka le pripomoček, s katero računalnik vzpostavlja zvezo, naša postaja pa se navzven še vedno javlja z radioamaterskim klicnim znakom. Z izmišljenimi IP številkami lahko seveda gradimo le lokalno omrežje, ki ni neposredno povezano z večjimi omrežji, kjer ne moremo vsem ostalim udeležencem preprosto vsiliti naše izmišljene IP številke.

Druga, verjetno najbolj smiselna pot, je pridobitev IP številke iz skupine 44.x.x.x, ki je v celoti dodeljena radioamaterjem. Pri tem je Sloveniji dodeljena podskupina 44.150.x.x in za dodeljevanje številke iz te skupine skrbi Aleš S56SAC. Z dodeljeno številko iz skupine 44.x.x.x lahko brez skrbi vzpostavimo IP zvezo s katerokoli amatersko radijsko postajo.

Tretja možnost je ta, da naš IP vmesnik nastavimo tako, da ob vzpostavljanju zveze zaprosi naš strežnik za dinamično dodelitev IP številke, ki bo veljala le za čas trajanja zveze. Takšen način dodeljevanja IP številke je primeren za ogromno množico hekerjev, ki ne vidijo dlje od lastnega telefonskega modema in vedno visijo na enem samem in istem strežniku. Za nas radioamaterje ni najbolj primeren, saj smo radioamaterji že iz packet-radia razvajeni, da smo hkrati povezani z več različnimi sogovorniki.

Vsak IP vmesnik nas bo poleg naše lastne IP številke vprašal še za masko, ki je na prvi pogled zapisana povsem enako kot IP številka. Maska določa, kateri del IP številke se primerja pri usmerjanju IP okvirjev na računalniku, ki ima mogoče več neodvisnih IP vmesnikov. Če vzpostavimo IP zveze z drugimi amaterskimi postajami s številkami iz skupine 44.x.x.x preko ene same radijske postaje, potem je smiselna izbira za masko 255.0.0.0.

Po nastavitvi lastne IP številke in maske moramo nastaviti naslove (klicne znake) še na packet-radio strani. Če uporabljamo kakšen bolj pameten operacijski sistem, je vrstni red opravič nepomemben. Pri uporabi windows in FlexNet pa najprej nastavimo lastno IP številko in masko, potem restartamo računalnik, da osrečimo windows, in šele nazadnje sprogramiramo klicne znake v FlexNet. Razlog za takšno početje je v tem, da Flexnet sprva hrani te nastavitve v RAM-u in jih

šele čez čas zapiše na disk, kar pomeni, da bi restart računalnika uničil ves naš trud.

Vse nastavitve klicnih znakov opravimo v FlexNet centrali tako, da poklikamo "IP routes". Tu moramo najprej izbrati lastni klicni znak in SSID, ki ga bomo uporabljali za TCP/IP delo. Ta znak in SSID potem jasno nista več na razpolago običajnim programom za packet. Nato moramo nastaviti še poti do naših sogovornikov: za vsakega sogovornika moramo vpisati njegovo IP številko in njegov AX.25 klicni znak. V večini slučajev moramo navesti tudi enega ali več digijev. FlexNet privzame način dela Virtual Circuit, če sami ne navedemo drugače.

Vsem nastavitvam sledi preizkus zveze. Žal je pri preizkusu zveze IP zelo neroden protokol. Večina nerodnosti izvira prav iz dejstva, da je IP namenjen profesionalnim omrežjem, kjer ne smemo izvedeti, kaj počne naš sosed. Edino orodje, ki ga imamo na razpolago v vseh operacijskih sistemih in izvedbah IP protokola, je program PING.

Ukaz PING <naslov> pošlje ICMP okvir za preizkus zveze (Echo Request) na izbrano IP številko. Drugi računalnik bi se moral javiti na pinganje s predpisanim ICMP odgovorom (Echo Reply). Program PING meri čas potovanja okvirja do sogovornika in nazaj v milisekundah, kar je ponavadi dobro merilo za kvaliteto zveze.

Za prvo pinganje si seveda izberemo zanesljivega sogovornika, še najboljše kar soseda radioamaterja, ki ima z internetnim protokolom več izkušenj in ima zanesljivo prav nastavljeno lastno radijsko postajo in računalnik. Med pinganjem opazujemo, kaj se dogaja z našo lastno postajo. Nam AX.25 monitorski program javlja, da se kaj dogaja? Gre naša radijska postaja na oddajo? Sprejema naša radijska postaja okvirje za nas?

Prvotna strojna in programska oprema za povezavo IP in packet-radia je bila zelo nerodna. Za vsako IP številko smo morali ročno vpisati klicne znake za AX.25 pot ter način dela (datagram ali VC) in to na obeh straneh zveze! Če je za navadnega uporabnika tipkanje gore številke in znakov silno zoprno opravilo, je avtomatsko delovanje takšnega omrežja skoraj nemogoče.

Nekaj težav so rešili programi za proxy. Program za proxy teče na primer na S50LEA na TCP portu 3128. Proxy omogoča, da naš računalnik vzpostavlja eno samo zvezo s S50LEA z IP številko 44.150.1.1 na portu 3128, za povezovanje naprej pa poskrbi računalnik S50LEA. Če želimo uporabljati proxy strežnik, moramo to posebej navesti v našem internetnem brskalniku (Netscape, IE ali Opera) tam, kjer izbiramo med direct connection in proxy ter seveda posebej nastaviti IP številko in port našega proxy-ja.

Proxy omogoča, da uporabnik nastavi le pot do enega samega IP naslova. Na drugi strani, na proxy strežniku, lahko vgradimo avtomatiko, ki posluša promet in si zapomni AX.25 klicne znake ter način dela (datagram ali VC) za vsako IP številko posebej. Takšno avtomatiko ima vgrajeno tudi FlexNet, vendar v vseh izvedbah FlexNet-a žal ne dela pravilno.

Pri izvedbi avtomatike si lahko seveda pomagamo s tistim, kar so za IP že naredili profesionalci. Na primer, tudi

Ethernet kartice imajo svoje okvirje in svoje 48-bitne naslove ter so IP okvirji le zapakirani v podatkovno polje na podoben način kot v AX.25 okvirjih. S strani internetnega protokola, ki je vgrajen v operacijski sistem, lahko zato pričakujemo nekaj pomoči pri usmerjanju okvirjev.

V večini slučajev uporabe IP tako na radioamaterskih frekvencah kot na Ethernet žicah imamo neposredno zvezo z nekaj postajami. Pot do vseh ostalih sogovornikov pa poteka preko prehoda (gateway). V tem slučaju se nam v nastavitvah TCP/IP lastnosti spleta nastavi privzeti prehod (default gateway). Vsi okvirji z IP naslovi, ki ne morejo skozi nobeno masko, bodo v tem slučaju usmerjeni po enaki poti, kamor kaže pot za privzeti prehod.

Sam privzeti prehod še ne zadošča, saj AX.25 vmesnik ne ve, kaj naj stori z okvirji s čudnimi naslovi, ki jih je prejel od IP vmesnika v jedru operacijskega sistema. V ta namen moramo nastaviti še privzeto pot (default route) v samem AX.25 vmesniku. V FlexNet-u je ta nastavev dobro skrita v FlexNet centrali, IP routes, Edit, Default route.

AX.25 vmesnik bo potem poslal vse okvirje za neznan IP naslove po privzeti poti nekemu, ki zna te okvirje posredovati naprej. Privzeta AX.25 pot naj pelje na strežnik, ki lahko naše IP okvirje posreduje naprej po lastnih tabelah usmerjanja, na primer S50YFE. Privzete (default) poti nikoli ne nastavljam na strežnike, ki tega ne obvladajo (na primer S50LEA, ki dela kot proxy) ali drugim postajam, ki nimajo povezave naprej.

Na koncu se nam spleta izkoristiti še eno ugodnost, ki nam jo ponujajo izdelani programi za internet, imenovano imenski strežnik ali DNS (Domain Name Server). V IP omrežju načeloma pokličemo drug računalnik tako, da navedemo njegovo IP številko. Dokler je naslovov malo in se s tem ukvarjajo radioamaterji, ki so itak domači s svojimi klicnimi znaki, to nekako gre. Ko IP doseže širne množice in se število naslovov poveča, si ljudje dosti lažje zapomnimo besedo, ki je mogoče v zvezi s ponujeno uslugo ali imenom lastnika računalnika, kot pa golo IP številko.

Če uporabljamo proxy kot je S50LEA, se s pretvorbo imen v IP številke ukvarja proxy in nam na našem računalniku ni treba nastavljan prav ničesar razen IP številke proxy-ja in pripadajoče številke porta, oboje kar v internetnem brskalniku. Če želimo videti sliko z ATV repetitorja na Pohorju, v internetni brskalnik preprosto vtipkamo ime graph-atv.uni-mb.si in za vse ostalo bo poskrbel proxy S50LEA.

Če pa imamo možnost neposrednega dostopa do omrežja, ki pelje do računalnika s slikolovom v Mariboru, na primer preko S50YFE ali S50DXX, moramo v internetni brskalnik vtipkati ustrezno IP številko 164.8.19.222, saj golo ime ni veljaven IP naslov. Tu nam lahko pomaga imenski strežnik, ki bo za vtipkano ime graph-atv.uni-mb.si našemu računalniku samodejno sporočil pripadajočo IP številko 164.8.19.222.

V našem računalniku moramo zato vključiti DNS, nastaviti IP številko imenskega strežnika (zgodba o kuri in jajcu!) in postaviti AX.25 pot do imenskega strežnika ali ustreznega prehoda. Če na primer uporabljamo S50YFE, potem je prvi imenski strežnik kar sam S50YFE ali 44.150.8.1, drugi imenski

strežnik pa 193.2.72.1 za vsak slučaj, če prvi imena ne najde. AX.25 pot do obeh gre preko S50YFE, torej AX.25 poti ni potrebno posebej vpisovati. Ker je v slučaju uporabe S50YFE to tudi privzeta (default) AX.25 pot, ni treba posebej vpisati niti poti do 44.150.8.1.

5. Uporaba SLIP protokola do TNCja

Skoraj vsak dan slišimo izgovor, da nečesa ta trenutek ni možno dobiti, ker računalnik ne dela. Bolj točno povedano, računalnik je kriv dosti manj kot pa gnilo napisana programska oprema. Med programsko gnilobo sodi tudi večina radioamaterskih programov vključno s programi za packet-radio. Kar je za nas še huje, večina programov za packet-radio dela navidez brezhibno s počasnim modemom za 1200bps, a se žal dobesedno sesede, ko priključimo megabitno postajo.

Čeprav leti večina obtožb in hudovanja na windows, niso programi za Linux kaj bistveno boljši. Radioamaterji smo pač na repu lestvice uporabnikov računalnikov, zato je proizvajalcem operacijskih sistemov kaj malo mar, če nam reči delajo ali ne oziroma bolj točno, če bojo naši programi sploh še delovali na naslednji inačici njihovega operacijskega sistema. Tudi proizvajalcem strojne opreme ni prav nič mar, če se v naslednjo izvedbo računalnika ne bo več dalo vtakniti naše kartice-vmesnika za packet-radio.

Po drugi strani se morajo proizvajalci programske opreme pošteno potruditi, da podpora internetnemu protokolu ne zataji v nobenem slučaju. Lokalna računalniška omrežja (Ethernet) in telefonske modeme danes uporablja večina ljudi, ki ima doma računalnik. Konkurenca na tržišču programske opreme komaj čaka, da bi tu šlo kaj narobe. Tudi proizvajalci strojne opreme se morajo pošteno potruditi, da v novih računalnikih delajo prav vse omrežne kartice in prav vsi telefonski modemi.

Radioamaterji nimamo niti volje niti časa, da bi si lahko privoščili pisanje gonilnikov za tisoče nezdružljivih inačic računalnikov in operacijskih sistemov. Če z internetom želimo izkoristiti predvsem veliko množico obstoječih programov, se moramo z našimi napravami prilagoditi obstoječim vmesnikom na osebnih računalnikih in to od električnih lastnosti signalov do zadnjega bitka protokola!

Od obstoječih vmesnikov so nam prav gotovo na razpolago 10Mbps in 100Mbps Ethernet priključki, RS-232 priključek z gonilnikom za telefonski modem in nazadnje še USB. Ethernet priključek je prav gotovo najboljša rešitev, saj je v osnovi namenjen računalniškim zvezam in se razvija naprej: profesionalci že govorijo o 1Gbps in celo 10Gbps Ethernet karticah. Edina slaba lastnost Ethernet priključka je mogoče v tem, da ustrezni čipi niso radioamaterjem enostavno dostopni.

Podobna ugotovitev velja tudi za USB, ki ga Američani imenujejo kar Useless Serial Bus, le da je še manj zmogljiv kot Ethernet, predvsem pa slabše podprt z gonilniki. Radioamaterjem nam vsaj kot prva rešitev ostane dobri stari RS-232, ki navsezadnje sploh ni tako grozno počasen, kot to izgleda na prvi pogled. S pravo kartico z možnostjo povečanja takta in FIFO vmesniki je RS-232 še vedno spoštljiv priključek, ki je za povrh v windows še presenetljivo dobro

programsko podprt.

Ne glede na vmesnik bi iz računalnika najrajši dobili gole IP okvirje, da jih potem z zunanjim TNCjem "oblečemo" v AX.25 protokol. Računalnik in programsko opremo lahko potem poljubno pogosto menjamo. TNC in radijska postaja imata dovolj dolgo življenjsko dobo, da jih radioamaterji sploh utegnemo izdelati in razviti do uporabne mere.

Najenostavnejši IP vmesnik je SLIP (Serial Line IP) na RS-232, ki ga pozna večina operacijskih sistemov. SLIP je sorodnik (predhodnik) KISS protokola za prenos AX.25 okvirjev po RS-232. Povsem enako kot KISS tudi SLIP uporablja posebni znak FEND (\$C0) za oznako začetka in konca okvirjev. Znak FEND v notranjosti IP okvirja se seveda kodira kot FESC TFEND (\$DB,\$DC) in temu ustrezno FESC kot FESC TFESC (\$DB,\$DD). SLIP okvirji naj bi ne bili daljši od 1006 bajtov, vendar večina programov dela brezhibno z okvirji dolžine 1500 bajtov.

V operacijskem sistemu windows moramo za SLIP najprej inštalirati telefonski modem. Seveda ne dopustimo, da bi windows iskal modem, ki ga ni, pač pa sami na seznamu izberemo standardni modem za 28800bps. Za modem moramo seveda izbrati enega od prostih COM portov (po možnosti takšnega s FIFO čipom 16550), nastaviti hitrost in izključiti RTS/CTS kontrolo pretoka.

Nato moramo postaviti telefonsko zvezo in pri tem pustiti windows veselje, da vrtil telefonsko številko v prazno! Lastnosti zveze nastavimo tako, kot je to prikazano na sliki 5. Windows nas bo sicer skušal posiliti z neuporabnim Microsoftovim PPP protokolom, vendar lahko v ustreznem okenčku izberemo tudi SLIP: Unix Connection. Izbira pravilno deluje v večini izvedb win95, win98 in win2000, z izjemo nekaterih zelo starih izvedb windows, na primer v slovenskem win95, ki se ga je sploh pametno izogibati.

Vgradnja telefonske zveze s TCP/IP protokolom je enostavnejša tudi zato, ker windows ne sitnari z restarti računalnika, čeprav se adapter in protokol samodejno naložita v okenček Network. Pri tem nastavimo IP številko in imenski strežnik kar v telefonski zvezi, kot je to prikazano na sliki 5. Enake nastavitve v Network pustimo prazne, kjer nastavimo le privzeti prehod (default gateway). Telefon sicer sam od sebe privzame masko 255.0.0.0, kar nam povsem ustreza, da se lahko povežemo tudi z Italijani (44.134.x.x) ali Hrvati (44.170.x.x).

Nekaj nastavitvev TCP/IP si v windows lahko ogledamo s programom WINIPCFG. Dostop do vseh nastavitvev seveda omogoča program REGEDIT, ki pa je za uporabo silno neroden in kompliciran. V win95 enostavneje nastavimo TCP/IP parametre s programom MTUSPEED, sicer bo win95 sam od sebe privzel dolžino okvirjev MTU 1500 bajtov. V win98 MTUSPEED v celoti ne dela več, pač pa lahko dolžino IP okvirjev nastavimo kar v Control Panel, Network, Dial-Up Adapter, Properties, Advanced, IP Packet Size. Billy Gates je tu izumil novo mersko enoto za MTU, kjer Automatic in Small pomenita 576 bajtov, Medium pomeni 1000 bajtov in Large pomeni 1500 bajtov.

Pripravljeni SLIP vmesnik poženemo tako, kot da bi hoteli vzpostaviti zvezo s telefonskim modemom. Trmastemu windows, ki hoče na vsak način vrteti telefonsko številko, je treba na

"AT" ukaze za modem lepo odgovoriti z "OK" in "CONNECT". Še največ truda nam pri tem vzame zapiranje raznih okenčkov, saj so nekatere kljukice zanje dobro skrite. Nenazadnje se spleča pripraviti tudi nadomestno "script" datoteko, ki vsebuje le dva ukaza in sicer "proc main" in "endproc", da nam windows ne odpira še enega nepotrebnega okna.

Če je šlo vse po sreči, bo windows lepo javil, da je vzpostavil zvezo in se povrhu še zlagal, da je preveril tudi ime in geslo uporabnika! V desnem spodnjem vogalu pri uri dobimo dva televizorčka, ki se prižigata, ko sprejemamo ali oddajamo SLIP okvirje: skoraj kot ledike DCD in PTT na TNCju! Edina stvar, ki jo windows v resnici preveri, je stanje kontrolnih linij na RS-232 priključku (na DB-25 moramo dodatno povezati skupaj nožice 6, 8 in 20).

Na drugo stran RS-232 kabla moramo seveda priključiti TNC, ki v smeri proti windows hinavi, da je telefonski modem. Ta naloga je dosti lažje izvedljiva, kot to mogoče izgleda na prvi pogled. SLIP in KISS okvirji imajo namreč popolnoma enako obliko, razlikujejo se le po vsebini. SLIP in KISS okvirje lahko ločimo med sabo že po prvem bajtu vsebine, ki je v KISS protokolu \$00 (navaden KISS), \$80 (SMACK) ali \$20 (FlexNet-CRC), v SLIP protokolu pa je prvi bajt IP okvirja skoraj vedno \$45 (IP inačica 4 in dolžina IP glave 5). Končno, TNC mora biti dovolj zvit, da pravilno odgovori na vse "AT" ukaze za telefonski modem.

Naveličan napak FlexNet-ovega internetnega vmesnika sem se odločil, da poskusim vgraditi SLIP v program mojega megabitnega KISS TNC-ja (glej CQ ZRS 4/2000). SLIP je seveda samo dodatek in v ničemer ne moti delovanja TNC-ja v KISS načinu, saj je SLIP in KISS okvirje res enostavno razlikovati med sabo. SLIP TNC naj bi premostil omejitve Flexnet-ovih gonilnikov, ki ne dopuščajo okvirjev daljših od 256 bajtov, ki ne delajo najboljšo v datagramskem načinu (Virtual Circuit je za naše omrežje SuperVozljev neprimeren) in ki se radi zakantajo tako s S57MMK DMA-SCC kartico (sesujejo windows) kot z megabitnim KISS TNC-jem (navidez se ne zgodi nič hudega, vendar se IP zveza ustavi).

Tale odstavek sem se seveda odločil napisati v glavnem zato, ker uspeh s SLIP TNC-jem ni izostal. S FlexNet-om in Virtual Circuit zvezo preko štirih vozljev do S50LEA sem dosegel prenos kvečjemu 700byte/s. Z isto radijsko postajo in istim TNC-jem, le z uporabo SLIP vmesnika do windows, okvirjev dolžine 1500 bajtov in datagramsko zvezo hitrost prenosa poskoči na desetkratno vrednost 7kbyte/s. Kar je še bolj pomembno, SLIP TNC in ustrezni SLIP gonilnik v windows se ne zakantajo kot FlexNet in čez noč lahko mirno zložimo z interneta na desetine megabajtov brez bojazni, da bi se prenos ustavil.

S stališča packet-radio omrežja je še bolj važen podatek, da se kljub 10-kratnemu povečanju hitrosti prenosa s SLIP TNC-jem zmanjša QRM: zmanjšajo se odstotki DCD in PTT na SuperVozljih. FlexNet namreč iz neznanih razlogov veliko ponavlja in naredi na vozljih tudi do 10kbyte/s QRM, čeprav hitrost čistega prenosa ne doseže 1kbyte/s. Končno, Pavle S57RA je primerjal SLIP in KISS na Linux-u in tudi tam je SLIP učinkovitejši, čeprav razlika med SLIP in KISS še zdaleč

ni tako velika kot v windows. Radioamaterski programi so izgleda res na psu in to na vseh operacijskih sistemih...

SLIP TNC opravlja seveda precej več nalog kot KISS TNC. Na oddaji mora opremiti SLIP okvirje z ustrezno glavo z amaterskimi klicnimi znaki za protokol AX.25. Na sprejemu mora glavo s klicnimi znaki spet odrezati, da ostane IP vsebina za SLIP okvir. V pomnilnik TNCja moramo zato sprogramirati precejšnje število parametrov, kar najlažje storimo s programom za packet-radio s KISS vmesnikom in UI okvirji, naslovljenimi na klicni znak "TNC".

Primer nastavitve parametrov SLIP TNC-ja je prikazan na sliki 6. Najprej moramo TNC sprogramirati kot običajni KISS TNC: nastavimo klicni znak za sito, SSID, trajanje glave in repa okvirjev, tečnobo prehoda na oddajo (persistence), vrsto KISS protokola (navadni KISS, SMACK ali FLEX) ter način delovanja avtomatike za vpis AX.25 poti IP okvirjev.

Sprejem deluje v SLIP načinu samo za izbrani klicni znak in SSID, vsi ostali okvirji, naslovljeni na drugačen SSID, pa se pretvorijo v KISS okvirje ne glede na vsebino. Na oddaji je naloga bolj zahtevna, saj je treba IP okvirje opremiti z različnimi klicnimi znaki za različne naslovnike. Z ukazom "!" nastavimo privzeto (default) AX.25 pot SLIP okvirjev. Klicne znake pri tem vedno opremimo z SSID-ji in napišemo v vrstnem redu naslovnik, pošiljatelj, prvi digi, drugi digi itd. Program za SLIP TNC pri tem omogoča, da je AX.25 pot oddanih okvirjev drugačna od poti sprejetih okvirjev.

Avtomatika za vpis poti omogoča, da SLIP okvirje pošiljamo po različnih AX.25 poteh. Če nastavimo način delovanja avtomatike na 0:IZKLOP, je avtomatika izključena in vsi SLIP okvirji grejo ven po privzeti AX.25 poti. Če nastavimo avtomatiko na 1:BRANJE, lahko v tabelo poti "ROUTES" ročno vpišemo do 16 različnih AX.25 poti za določene IP številke z ukazom "#".

Na sliki 6 je prikazan primer vpisa AX.25 poti za IP številko 44.150.1.161 in klicni znak S52DS-7. Neželjeno pot v tabeli poti pobrišemo enostavno tako, da v ukazu "#" navedemo samo IP številko, ki jo želimo brisati, brez klicnih znakov. Če IP naslovnika ni v tabeli poti, se SLIP okvir seveda opremi z znaki privzete AX.25 poti.

V načinih 2:VPIS-DEF in 3:VPIS je omogočen tudi samodejni vpis v tabelo poti. Program v TNCju razbere AX.25 pot in IP številko v sprejetem okvirju ter doda oziroma obnovi vpis v tabeli poti, ki se potem uporablja za oddajo okvirjev po isti poti nazaj. Tabela poti se pri tem obnavlja tako, da gre najbolj sveži vpis na vrh tabele, na dnu tabele pa izpade najstarejši vpis, če v tabeli ni več prostora.

V načinu 3:VPIS se vpišejo v tabelo vsi sprejeti AX.25 okvirji, ki v svoji notranjosti vsebujejo IP okvir. V načinu 2:VPIS-DEF, ki se najpogosteje uporablja, se v tabelo poti prav tako vpisujejo AX.25 poti razen za okvirje, ki prihajajo iz izvirnega klicnega znaka, ki ustreza privzeti (default) poti. V primeru na sliki 6 se nobeden od okvirjev, ki prihajajo iz izvora S50YFE (SSID se ne upošteva) ne vpisuje v tabelo poti, saj ti tabele ne potrebujejo in bi po nepotrebem iz tabele izrinili bolj koristne vpise.

6. Uporaba internetnega protokola v radioamaterstvu

Pri uporabi internetnega protokola se moramo zavedati, da zveza še vedno poteka na radioamaterskih frekvencah in mora biti zato vsebina prenašanih sporočil temu primerna. Sam internetni protokol ni pri tem prav nič kriv, saj se da prenašati sporočila z zelo neprimerno (komercialno) vsebino tudi po SSB zvezi, z uporabo Morzejeve telegrafije ali preko SSTV ali ATV slikic. Radioamaterski internet prav gotovo ne more biti samo zastojkarski internet, ki razen frekvenčnih področij z radioamaterstvom nima nobene zveze.

Internetni protokol nam v resnici samo dodaja nove vmesnike: zvočniku, mikrofону, TV kameri, telegrafski tipki, slušalkam in teleprinterju so se pridružili Netscape, razni FTPji, Internet Explorer, Microsoft Netmeeting in kopica drugih programov, ki so navsezadnje samo vmesniki med radijsko zvezo na eni strani in človekom na drugi strani. Povsem pravilno je, da se te nove vmesnike naučimo uporabljati na enak način, kot je bilo nekoč treba obvladati telegrafsko tipko ali mikrofón.

Internetni protokol uporabljamo izključno iz razloga združljivosti z obstoječo programsko opremo na osebnih računalnikih. V primerjavi z radioamaterskim protokolom AX.25 ni kaj bistveno boljši ali slabši. Radioamaterji si zaradi lastne majhnosti nekaterih zahtevnih nalog preprosto ne moremo privoščiti. Tako kot v domači delavnici ne moremo izdelati tranzistorjev in integriranih vezij, se tudi ne moremo lotiti zahtevne programske opreme, še posebno, če moramo računati na združljivost z nepregledno množico različnih računalnikov in operacijskih sistemov.

Združljivost z ostalim svetom preko internetnega protokola lahko pomeni tudi prednost za radioamaterstvo. Predvsem radioamaterji ne moremo več biti vase zaprta skupnost, ki se ukvarja izključno sama s sabo in se za razvoj ostalega sveta sploh ne zmeni. Uporabo omejene naravne dobrine, kot je radiofrekvenčni spekter, moramo pred širšo družbo upravičiti.

Pri tem nam uporaba internetnega protokola lahko le pomaga vrniti del izgubljenega ugleda v družbi. V lastnih vrstah smo zanemarili dejavnosti, ki jih družba od nas pričakuje: eksperimentiranje in izobraževanje. Nesmiselno trmarjenje s preživeli tehnologijami nas je pripeljalo v vase zaprte in za širšo družbo nepomembne dejavnosti. O tem naj razmislijo predvsem tisti, ki trdijo, da internet z radioamaterstvom nima nobene zveze.

Če se že odločimo za internet, moramo najprej pregledati lastno opremo za packet-radio in zmožnosti packet-radio omrežja v naši okolici. Internetni protokol zahteva res dobre radijske zveze z malo ponavljanji. Če nam virtual circuit dela boljše kot datagram, je to zanesljiv znak, da imamo slabo radijsko zvezo z mnogo ponavljanji.

Dostop do interneta zahteva tudi primerno opremo. Z ozkopasovno FM postajo in modemom za 1200bps ali 2400bps verjetno nima smisla riniti na internet. Takšna radijska zveza je preprosto prepočasna za karkoli, kar ni golo besedilo v čisti ASCII kodi. Modeme in postaje za nizke

bitne hitrosti zato rajši prihranimo za takšno uporabo, kjer hitrost ni bistvena, na primer APRS ali pa visenje na DXCLUS.

Pri nas na srečo ne uporabljamo FSK modemov za 9600bps skozi ozkopasovne FM postaje. Sami 9600bps modemi sicer niso nič krivi, krivda je v prehudih zahtevah za radijske postaje, ki jih cenene in enostavne FM radijske postaje ne morejo izpolniti. 9600bps FSK modemi so v "razviti Evropi" dobesedno pokopali packet-radio, saj so v očeh večine radioamaterjev nekaj, kar ne dela.

Celo avtor FlexNet-a, Gunter DK7WJ mi je brez ovinkov izjavil, da svojega programskega paketa ne namerava prirediti za daljše okvirje, saj so za neučinkovite 9600bps modeme že 256-bajtni okvirji predolgi in bi daljši okvirji vodili samo v neskončna ponavljanja. FlexNet zato ostane vezan na Virtual Circuit, ki glede na številna ponavljanja neučinkovitih modemov edini sploh omogoča delovanje internetne zveze.

WBFM radijska postaja z Manchester modemom omogoča povsem spodoben dostop do interneta, primerljiv z dobrim (analognim) telefonskim modemom, če le radijski kanal ni preveč zadrenjan. Najboljše rezultate dosežemo s hitrostjo 38.4kbps, datagramskim prenosom s podaljšanimi okvirji na 1500 bajtov in predelanim Manchester modemom, ki ima v PTT kužapaziju dodano diodo za dolge okvirje.

Ker je IP prenos še posebno v datagramskem načinu zelo občutljiv na ponavljanja, se uporaba 76.8kbps največkrat ne izplača. 76.8kbps skozi Manchester modem in WBFM postajo prizadenejo podobne omejitve kot neučinkoviti 9600bps modemi skozi ozkopasovne FM postaje. Prenos 38.4kbps po kvalitetni zvezi brez ponavljanj se pogosto izkaže celo boljši od megabitne PSK radijske zveze, ki ni povsem brezhibna in ponavlja.

Če se odločimo za gradnjo megabitne PSK radijske postaje, moramo zagotoviti tudi kvalitetno radijsko zvezo. Internetni protokol in daljši 1500-bajtni okvirji zahtevajo 5dB ali 10dB več signala od običajne packet-radio zveze. V dobrih razmerah nam megabitna PSK radijska postaja omogoča internetni dostop, ki je primerljiv z digitalnim ISDN modemom. Glavni krivec za omejitve zmogljivosti ni več končna zveza do našega računalnika, pač pa zmogljivost packet-radio omrežja, kjer se megabitne PSK zveze delijo med številne uporabnike.

Sama megabitna PSK zveza točka-točka sicer omogoča s hitrimi vmesniki (S57MMK DMA-SCC kartica in Linux) precej več in je povsem primerljiva z zadnjim krikom profesionalne tehnike, to je ADSL modemi. Za izkoriščanje polne zmogljivosti uporabniške megabitne PSK zveze bo treba zgraditi novo packet-radio omrežje s hitrejšimi vozli in službenimi zvezami med vozli z bitno hitrostjo nekje med 10Mbps in 40Mbps.

V sedanjem packet-radio omrežju pri internetnem dostopu komaj opazimo razliko med SLIP TNC-jem, ki je povezan na nepredelan računalnik po 115.2kbps RS-232, in SLIP TNC-jem, ki je povezan po 460.8kbps do hitre RS-232 kartice. Razlika med TNC-jema je seveda precej večja v običajnih packet-radio zvezah v KISS načinu in s kratkimi 256-bajtnimi okvirji. Največjo razliko seveda opazimo v internetni zvezi točka-točka

med dvema SLIP TNC-jema, kjer je hitrost prenosa natančno določena s hitrostjo na RS-232.

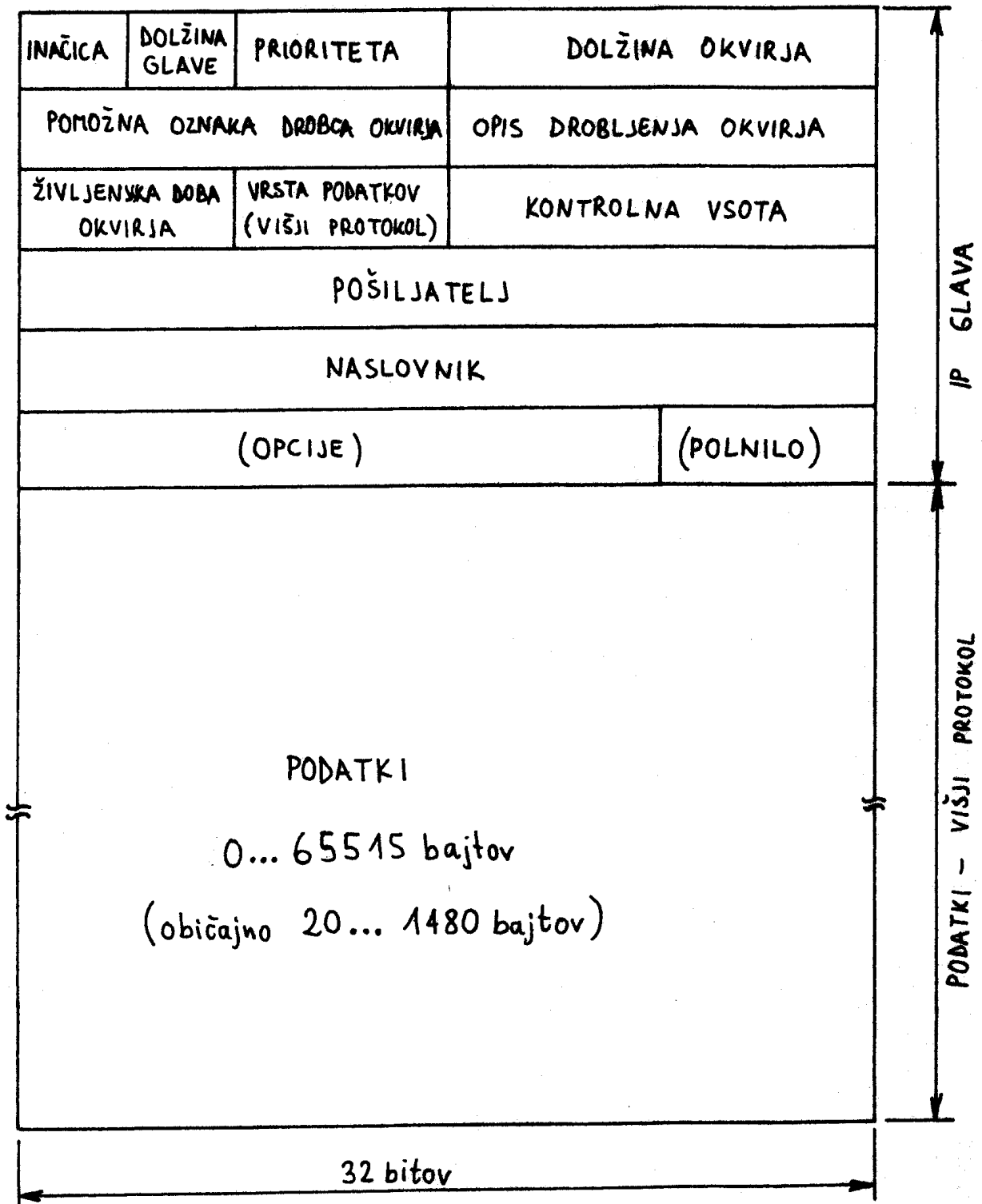
Pri primerjavi hitrosti seveda ne smemo pozabiti na omejitve žičnega internetnega omrežja, kjer se promet pogosto zabaše in hitrost prenosa pade na tisto, kar zmore telefonski modem ali celo manj. Tu je seveda priložnost, da se radioamaterji izkažemo in pokažemo, da naše radijsko omrežje nudi več od strašansko dragih profesionalnih kablov.

SLIP TNC verjetno predstavlja danes najenostavnejšo pot do radioamaterskega interneta, ki zagotovo ne zahteva doktorata iz računalništva. Kako izvleči SLIP iz windows, je na našem koncu prvi pogruntal Massimo IV3ZXF, ki je to izkoristil pri krmiljenju svojih "Itanet" vozlišč. Moj sosed Sandi S53SM je vse še enkrat preveril in preizkusil, preden sem se odločil vgraditi SLIP podporo v megabitni TNC.

Sam SLIP TNC seveda ni zadosti, tudi packet-radio omrežje je zahtevalo predelave. Stari SuperVozlji z MC68450 DMA vezji zahtevajo kar nekaj predelav, da se število izgubljenih okvirjev zmanjša na razumno mejo. Vse vozlje vključno z novimi z vezji MC68360 je bilo treba predelati na daljše okvirje, ki pomenijo bistveno boljše izkoriščanje megabitnih postaj. Končno, poskuši z internetom zahtevajo tudi delujoče prehode na žično omrežje in tu gre zahvala Andreju S51BW, ki skrbi za prehoda S50LEA in S50YFE, na katerih je omogočen dostop do žičnega interneta vsem radioamaterjem, ki si to želijo.

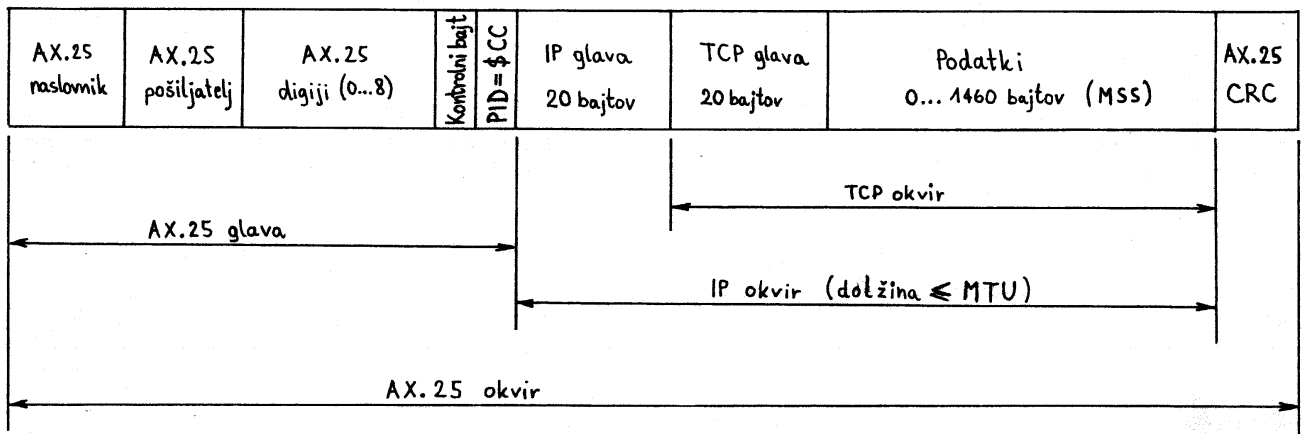
Seznam slik:

- Slika 1 - Sestava IP okvirja (inačica 4).
- Slika 2 - Vgradnja TCP/IP okvirja v AX.25 okvir.
- Slika 3 - Datagranska zveza skozi verigo SuperVozljev.
- Slika 4 - Amaterska postaja za packet-radio in internet.
- Slika 5 - Izbira SLIP vmesnika v operacijskem sistemu windows.
- Slika 6 - Nastavitev parametrov SLIP TNCja.

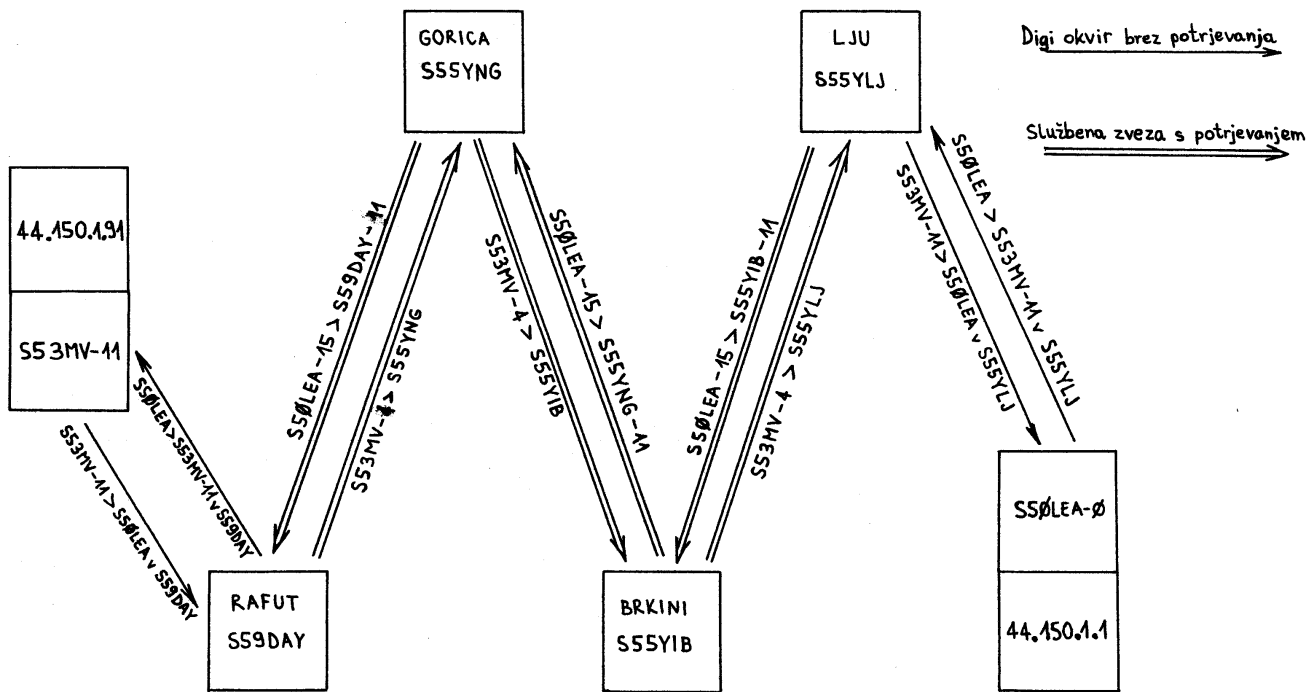


$(576 \text{ bajtov}) \leq \text{DOLŽINA OKVIRJA} \leq 65535 \text{ bajtov}$
(MAX. DOPUSTNA)

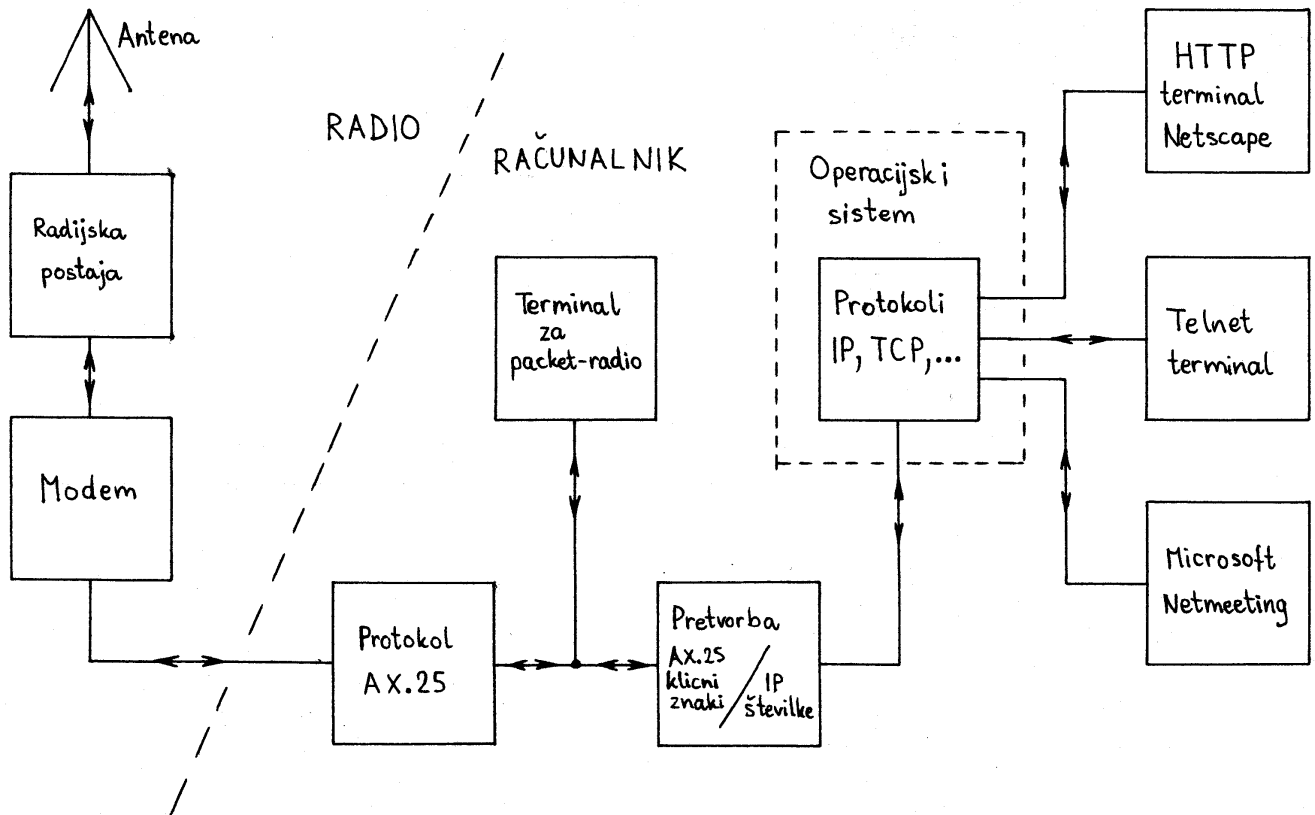
Slika 1 - Sestava IP okvirja (inačica 4).



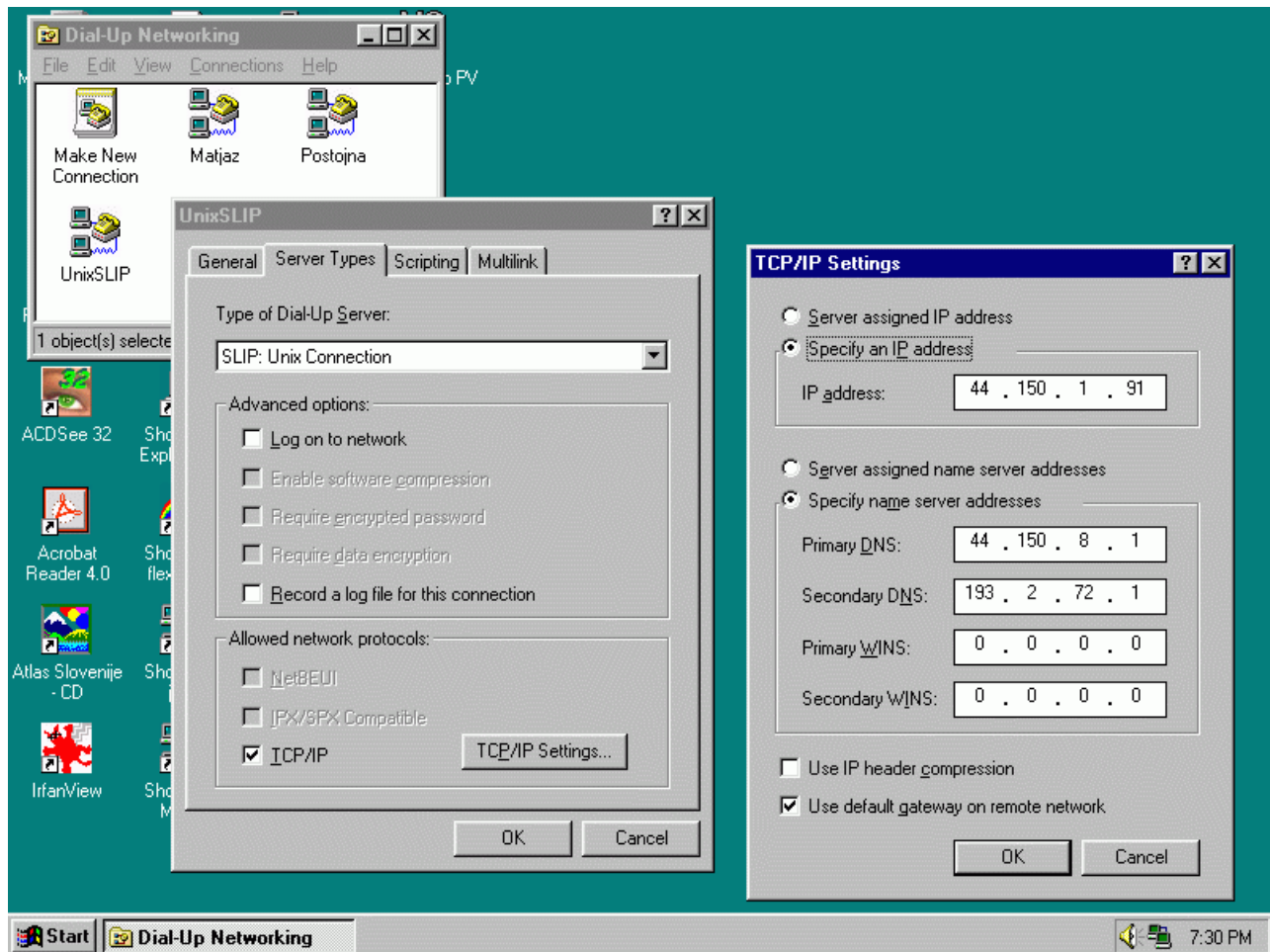
Slika 2 - Vgradnja TCP/IP okvirja v AX.25 okvir.



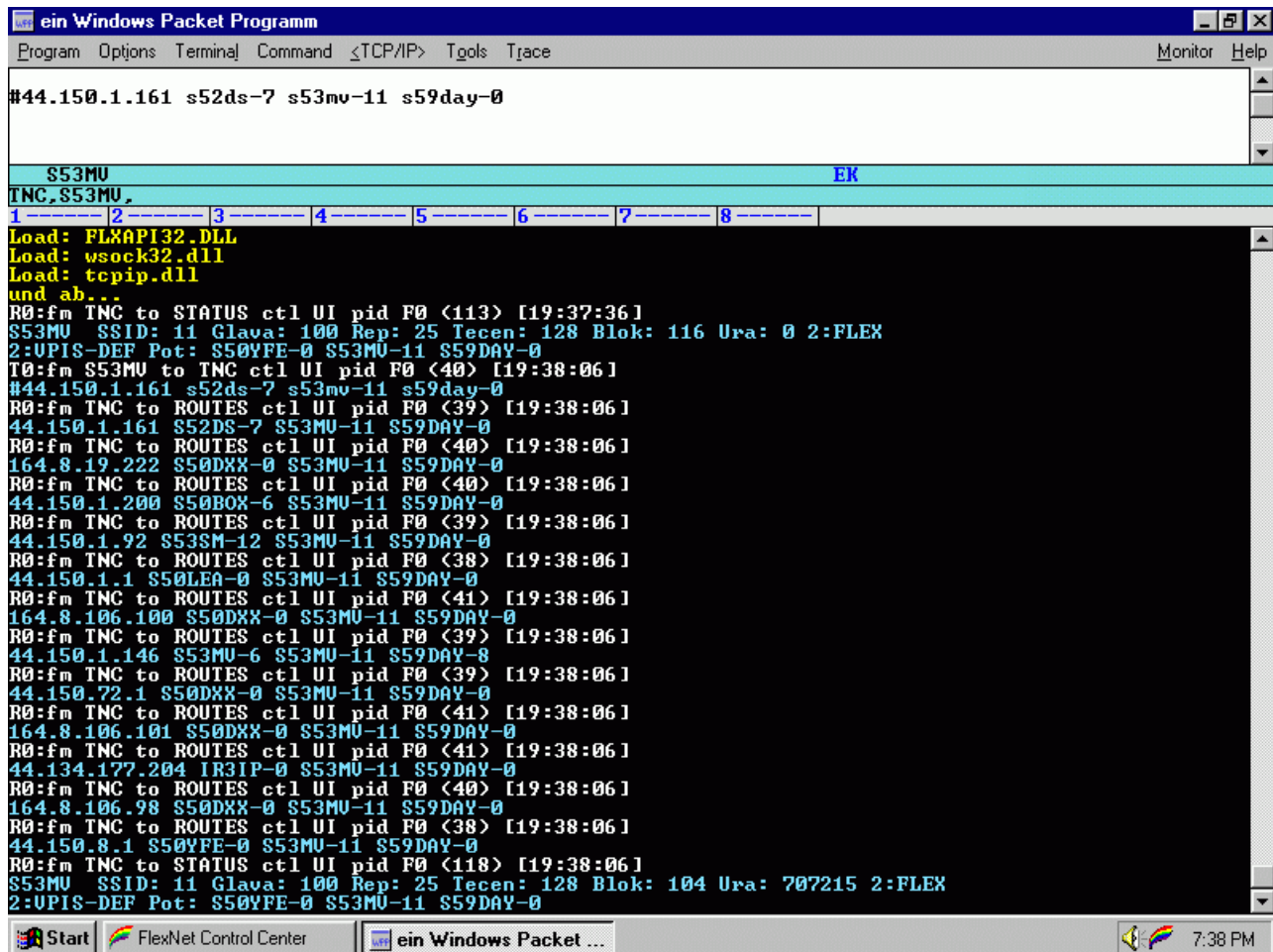
Slika 3 - Datagramska zveza skozi verigo Super Vozljev.



Slika 4 - Amaterska postaja za packet-radio in internet.



slika 5 - Izbira SLIP vmesnika v operacijskem sistemu windows.



slika 6 - Nastavitev parametrov SLIP TNCja.