



15th

ARRL and TAPR DIGITAL COMMUNICATIONS CONFERENCE

Seattle, Washington
September 20-22, 1996





15th ARRL and TAPR DIGITAL COMMUNICATIONS CONFERENCE

Local Co-Hosts: Puget Sound Amateur Radio TCP/IP Group
Boeing Employees Amateur Radio Society (BEARS)



Conference Coordinators:

Steve Stroh, N8GNJ
Keith Justice, KF7TP
Steve Ford, WB8IMY
Greg Jones, WD5IVD



American Radio Relay League, Inc.
225 Main Street
Newington, CT 06111-1494 USA

tel: 860-594-0200 WWW: <http://www.arrl.org/>



Tucson Amateur Packet Radio
8987-309 E. Tanque Verde Rd #337
Tucson, Arizona 85749-9399 USA

tel: 817-383-000 WWW: <http://www.tapr.org>

Copyright © 1996 by

The American Radio Relay League, Inc.

Copyright secured under the Pan-American Convention

International Copyright secured

This work is Publication Number 244 of the Radio Amateur's Library, published by the League. All rights reserved. No part of this work may be reproduced in any form except by written permission of the publisher. All rights of translation reserved.

Printed in USA

Quedan reservados todos los derechos

ISBN: 0-87259-568-4

ARRL Order Number: 5684

First Edition

On Amateur Digital Communications

This is the first time I've participated in an ARRL Digital Communications Conference. I know that these conferences have served as a sounding board for technical ideas. Some have become standards and accepted by the mainstream. Amateur packet radio is certainly an example of amateurs contributing to the state-of-the-art.

The League is now faced with increasing difficulty justifying our precious spectrum. It doesn't at all reflect poorly on amateurs. The problem is that commercial services are seeking spectrum, on a shared basis if necessary, when they can't get exclusive allocations. If you'd asked me earlier this year if extensive amateur use of a band would protect it against encroachment, I would have said "yes." You will remember the saying, "Use it or lose it." Well, we certainly can lose a band by not using it. But the little LEO experience has taught us that even the most-used band can become the target for someone desperately seeking spectrum.

This means that we must fully develop our bands, not just to occupy them, but to fulfill the purposes of the Amateur Radio Service. Ultimately, this means public service in the broadest sense of the term. We either provide the services that the world and the country think worth the investment of spectrum or we see our bands increasingly targeted by other services. Advancing the radio art and technical training certainly are included in this broad definition of public services. The Amateur Radio digital community and these conferences have upheld this public trust.

One of the League's strategic objectives is to foster closer relationships with Amateur Radio technical organizations. TAPR is one of the organizations with which we already have close ties. We have agreed that the ARRL Digital Communications Conference and the TAPR Annual Meeting can be associated in time and place, and that TAPR would do most of the organizing. There are some other details that make it a win-win solution.

I wish you a productive conference.



Rodney J. Stafford, KB6ZV
President, ARRL
5155 Shadow Estates
San Jose, CA 95135
kb6zv@arrl.org
September 1996

Table of Contents

Learning DSP by Porting Programs to the TAPR/AMSAT DSP-93 Modem John B. Bandy, W0UT.....	1
Linking BPQ Switches via Ethernet Bill Barnes, N3JIX.....	5
javAPRS: Implementation of the APRS Protocols in Java Steve Dimse, KO4HD.....	8
The Radio Amateur Digital System Artificial Intelligence Project Garry W. Joerger, N5USG.....	15
Fast Flow Control in High-Speed Communication Networks C.M. Kwan, R. Xu and L. Haynes.....	19
Nonlinear Channel Equalization Using Fuzzy CMAC Neural Network C.M. Kwan, R. Xu, L. Haynes and J.D. Pryor.....	25
Optimization of Phase-Locked Loops With Guaranteed Stability C.M. Kwan, H. Xu, C. Lin and L. Haynes	36
Baseband Group Delay Equalization of IF Filters for Data Communications T.C. McDermott, N5EG.....	42
Easy to Follow Packet James C. Nobis.....	53
Object-Oriented Modeling of a Satellite Tracking Software M. Normandeau and M. Barbeau, VE2BPM.....	59
XNET: A Graphical Look at Packet Radio Networks Richard Parry, W9IF.....	64
A 9600-Baud Modem for the LPT Port Wolf-Henning Rech, N1EOW/DF9IC and Donald Rotolo, N2IRZ.....	76
Amateur Radio Digital Voice Communications Paul L. Rinaldo, W4RI.....	80

WinAPRS: <i>Windows</i> Automatic Position Reporting System	
Mark Sproul, KB2ICI , and Keith Sproul, WU2Z	130
Automatic Radio Direction Finding Using MacAPRS & WinAPRS	
Keith Sproul, WU2Z	136
Circus of the Stars	
Michelle Toon, KC5CGH	141
13-cm PSK Transceiver for 1.2 Mbit/s Packet Radio	
Matjaz Vidmar, S53MV	145
23-cm PSK Packet-Radio RTX for 1.2 Mbit/s User Access	
Matjaz Vidmar, S53MV	176
Packet and Internet	
James Wagner, PhD, KA7EHK	203
Strategies for Improving Wide-Area Networks	
James Wagner, PhD, KA7EHK	206
The Word Storage Relay (WSR)	
Pat West, PE, W7EA	212
On-Air Measurements of HF Data Throughput Results and Reflections	
Ken Wickwire, KB1JY	221
On-Air Measurements of MIL-STD-188-141A ALE Data Text Message	
Throughput Over Short Links	
Ken Wickwire, KB1JY	233
CLOVER-The Technology Grows and Matures	
Bill Henry, K9GWT	237
Construcing a Worldwide HF Data Network	
Craig McCartney, WA8DRZ	248

Learning DSP by Porting Programs to the TAPRIAMSAT DSP-93 Modem

John B. Bandy, WOUT, ex-WAOUTT, ex-WNOTSK; Wichita, Ks U.S.A.

4th of July,

1996

Abstract.

This paper is about porting some assembler programs in the text authored by Rulph Chassaing and Darrell W. Horning, titled, "Digital Signal Processing With the TMS320C25", published by John Wiley & Sons, New York, ISBN 0-471-51 066-1, 1990, phone 1-800-CALL-WILE, (hereafter the text) to the TAPR/AMSAT DSP-93 ham radio modem kitted by the Tucson Amateur Packet Radio, Corp. (TAPR), Tucson, Az, phone 1-817-383-0000. Porting these senior/1st year graduate electrical/computer engineering student level programs taught the author many things about digital signal processing (DSP) in general and TAPR/AMSAT DSP-93 modem in particular. AMSAT is an acronym for Radio Amateur Satellite Corporation.

Key Words

Assembler, DSP, Programming, Modem, TMS320C25

Introduction.

There are several different dialects of the programming language of Assembler. It is fortunate from a learning prospective that the Speech Technology Inc.'s shareware Table Driven Assembler (TASM) program⁷ supplied with the TAPR/AMSAT DSP-93 modem will not process the Texas Instruments, Inc. (TI) dialects used in the text. These dialects are TI-Tag, COFF (common object file format), and TI-Tag macro.

Porting Programs to the TAPR/AMSAT DSP-93 Modem.

While porting some of the numerous TI-Tag programs the author made some notes that might make the learning path easier for other hams.

	TI-Tag	TASM	
Change	A ORG to	.ORG	
	END to	.END	
	DATA to	.WORD	
	> to	h	hex example: >2E to 02Eh
	* to	;	for comments
	EQU to	.EQU	
	BIOZ to	WAIT4	

1. Use .MSFIRST before the first .WORD or the bytes will store reversed.
2. Move tables to the bottom of the program.
3. Move interrupt routines to the bottom of the program.
4. On the modem BO is in program memory, initially.
5. Prefix a comment in an instruction line with a semi-colon.
6. .WORD converts a minus base 10 number to a 2's compliment number in base 16.

example: -7782 will appear on the assembly listing as E19A

10

16

7. The use of .ORG in the table definition is not required and using it may result in code overlaying other code in memory.

8. Any address in a statement in the inclusive range of **60-6F** on page 0 must be changed to another address as the modem's monitor uses these addresses.
9. All 4 digit hexadecimal numbers must be prefix with a zero if the left digit is not zero.

Example: **7E52** is coded **07E52**

10. Do not use the SXF and BIOZ instructions because they cause halts.
11. Insert the following at the top of the program.

```

AR0      .EQU    00h
AR1      .EQU    01h
AR2      .EQU    02h
AR3      .EQU    03h
AR4      .EQU    04h
AR5      .EQU    05h
AR6      .EQU    06h
AR7      .EQU    07h

```

12. Do not use MODE and RATE instructions because they are Analog Interface Board (AIB) instructions. The parameters in them can be used in the Analog Interface Chip (AIC) initialization routine.
13. **TAPR/AMSAT DSP-93's** AIC is TLC32044 (sin x/x feature).
Text's AIC is TLC32041 (no sine x/x feature).
14. 10 - 22,000 hz is the range of music.

The text has Texas Instrument (TI) assembler programs for both the Analog Interface Board (AIB) and the Analog Interface Chip (AIC), similar to the one that's in the modem. Porting the AIC programs just requires some syntax changes, address changes, and re-ordering the data blocks and routines. The AIB programs also requires the AIB initialization code (mode, rate) to be replaced by AIC registers initialization code and interrupt routines.

The monitor* (operating system) supplied with the modem uses certain registers and memory locations, so any portion of the text program using these memories must be moved by changing register names and memory addresses, and/or memory page numbers. To manage memory it might be helpful to draw and color code registers, memory blocks, and memory pages on paper. Save it for future reference.

The software development kit supplied with the modem has several routines whose calls can be temporary placed in the ported program code to display registers & memory data (CALL **DEBUG**)² and addresses on a computer screen to assist debugging. Most of these programs loop continuously until the reset button is pushed. If the same address displays every so often, it is looping. Placing LED lighting **code**⁸ after certain instructions can also be helpful in determining what routines are executing, etc.. If a program is not looping, then moving a "display my address" call (TRACE) ² around in the code can pin point the instruction causing the halt. If the "display my address" call executes before an instruction and not after it, then that's the instruction causing the halt. After the debugging process is complete remove (or comment) all these temporary display calls because they slow the program down.

The following were at-my-elbow (must haves) books/papers during porting.

1. Above text.

2. Ron Parsons, **W5RKN**, Don Haselwood **K4JPJ**, & Bob Stricklin, **N5BRG**, DSP-93 Programming Guide, published by Tucson Amateur Packet Radio Corporation, Tucson, Arizona, June 1995, paper, phone 817-383-0000.
3. Frank H. Perkins, Jr., **WB5IPM**, "DSP-93 Programming Hints", Proceedings of the 14th ARRL DIGITAL COMMUNICATIONS CONFERENCE, published by The American Radio Relay League, Inc. (ARRL), Newington, Conn, ISBN: O-87259-526-9, ARRL Order Number: 5269, 1995, phone 1-860-594-0200.
4. Ron Parsons, **W5RKN**, "Analog Block" and "Digital Block", published by Tucson Amateur Packet Radio Corporation, Tucson, Arizona, [ftp.tapr.org /tapr/dsp93/schems/](ftp://ftp.tapr.org/tapr/dsp93/schems/) or <http://www.tapr.org/tapr>
5. "Chapter 3, Architecture" and "Chapter 4, Assembly Language Instructions" , **TMS320C2x** User's Guide, published by Texas Instruments, Inc., Dallas, Texas, 1604907-9721 revision C, **SPRU014C**, January 1993, free, phone **214-644-5580**.
6. "Section 4 - Analog Interface Circuits and Codec, **TLC32044C**....., Voice-Band Analog Interface Circuits", Data Acquisition Circuits - Data Conversion and DSP Analog Interface - Data Book, published by Texas Instruments, Inc., Houston, Texas, 1995, **SLAD001**, pages 4-35 thru 4-72, free, phone **214-644-5580**.
7. TASM USER'S MANUAL, published by Speech Technology Incorporated, Issaquah, Washington, Version 3, October 1993.
8. TESTLEDS.ASM source code, [ftp.tapr.org /tapr/dsp93/software/](ftp://ftp.tapr.org/tapr/dsp93/software/) or <http://www.tapr.org/tapr>

The author has not as yet done a live on-the-air test of any of the programs he has ported, nor used an audio signal generator, **oscilloscope**³, or spectrum analyzer to test them. He was satisfied at the time with the modem sending (TRACE)* the same addresses repeatedly to the PC screen during looping, flashing its LED's, and sending expected sounds to the stereo broadcast amplifier-receiver. He learned about DSP and the **TAPR/AMSAT** DSP-93 by porting, but has more to learn before he writes his first useful filter, modem, etc., program, if ever. Some of the math he chose to ignore because he did not understand it.

The porting of the few TI-Tag macro and COFF programs briefly discussed in appendix A and C respectively of the text is outside the scope of this paper.

Porting the Hexadecimal Coefficient Table Generators, etc..

The text's **GWBASIC** programs ported to Microsoft Corp. MS-DOS **QBasic**, Version 1 .1, by changing vector name AS () and variable name TYPE to some other names. And by changing the British pound symbol to the American pound symbol # for double precision.

DSP Articles/Source Code.

The following articles/source code helped the author understand DSP in general, and may be of value to the reader depending on her/his place on the learning curve. Copies of articles are probably available from your nearest college/university library thru inter-library services, or the publisher for a small fee. Most college/university libraries serve the public in addition to the student body and faculty.

A. Jon Bloom, **KE3Z**, ARRL Staff, "Digital Signal Processing", Chapter 8 - Digital Basics, The ARRL Handbook for Radio Amateurs, published by The American Radio Relay League, **Inc.(ARRL)**, Newington, Conn, ISBN: **0-87259-171-9**, 71st Edition, 1994, pp. 8-36 thru 8-38, phone 860-594-0200

B. Mac A. Cody, "The Fast **Wavelet** Transform", Dr. Dobb's Journal, M&T Publishing, Inc., San Mateo, Ca, **#187**, Vol. 17, Issue 4, pp. 16-28, April 1992, phone 415-358-9500.

C. Bill **deCarle**, **VE2IQ**, "Fourier Transforms: Math or Magic?", Chapter 29 - Digital Equipment, The ARRL Handbook for Radio Amateurs, published by The American Radio Relay League, Inc., Newington, Conn, ISBN: **O-87259-1 71-9**, 71st Edition, 1994, p. 29-1 3, phone 860-594-0200

D. Johan Forrer, **KC7WW**, "A Low Cost DSP Modem for HF Digital Experimentation", Proceedings of the 13th ARRL DIGITAL COMMUNICATIONS CONFERENCE, published by The American Radio Relay League, Inc. (ARRL), Newington, Conn, ISBN: 0-87259-483-1, ARRL Order Number: 4831, 1994, phone 1-860-594-0200. Source code and schematics were available from Johan Forrer, **KC7WW**, forrerj@ucs.orst.edu

E. Dave Hershberger, **W9GR**, "DSP - An Intuitive Approach", QST, published by The American Radio Relay League, Inc., Newington, Conn, ISSN: 0033-4812, Vol. 80, No. 2, pp. 39-42, Feb 1996, phone 860-594-0200

F. Thomas E. Janzen, "Recovering Corrupted Waveforms", The C Users Journal, published by R&D Publications, Inc., Lawrence, Ks, Vol. 11, No. 6, pp. 39-48, Jun 1993, phone **913-841-1631**.

G. Roy E. Kimbrell, "Finding Significance in Noisy Data", Dr. Dobb's Journal, M&T Publishing, Inc., San Mateo, Ca, **#189**, Vol. 17, Issue 6, pp. **30-42**, June 1992, phone 415-358-9500.

H. David L. Mills, **W3HCF**, FSK modem/TNC for HF RTTY, source code, object code, and documentation, 1995, <ftp://tapr.org/tapr/dsp93/software/modem> or <http://www.tapr.org/tapr>

No Warranty.

This material is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of merchantability and fitness for a particular purpose.

"The Learning is in the Porting"

Linking BPQ Switches via Ethernet.

By Bill Barnes, N3JIX

Brief technical description: Two or more computers, running GSBPQ node software. There is a need to link all these switches together. There are two options: a 9600 baud RS-232 kiss port, or a Ethernet port. BPQ wrote a driver for ODI that will allow the switch to talk to Ethernet. This is a document on that process.

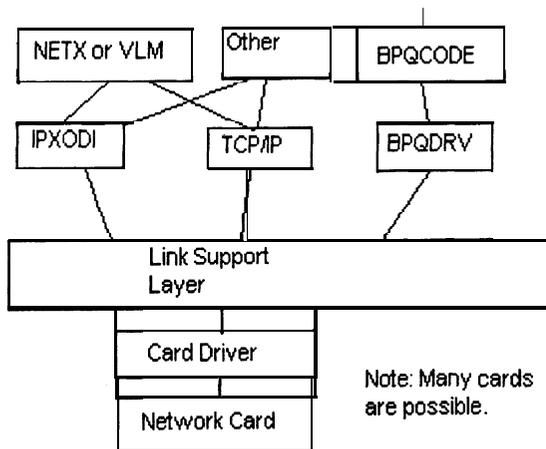
Key Words: G8BPQ, Ethernet, ODI.

What is ODI?

ODI is Novell's newest idea for Clients. Before, when you changed cards, you had to change IPX versions as well. Also, that old IPX wasn't as "flexible" on card settings either. So, Novell decide to make a flexible IPX, and well, it grew way over that, into ODI. What ODI allows is card manufactures to write a driver for their card, and use a generic IPX. So, the only thing that need to be changed is the card driver and edit the section in the NET.CFG file.

ODI depends on a Link Support Layer or LSL. This file always needs to be loaded first, LSL.COM.

Here is a diagram of how ODI works, and how G8BPQ's driver fits in there.



Does that make any sense? No? Well, the idea is that many network cards can talk to LSL, and many protocols can talk to LSL. So LSL is kinda like a traffic cop.

Ok, Here's how to make it work.

My AUTOEXEC.BAT:

```
@echo off
prompt $p$g$g
cd\network
lsl
ne2000
cd\bpq
odidrv 125

bpqcode

cdl
ipxodi
```

, LSL is always needed first.
, This is your card driver.

, This is your External ODI drv for BPQ.
, 12.5 is the *Intlevel* from your Ports section.
, Of course, BPQCODE and anything else
, you may need to load

, IPX is only needed for Novell. Optional
, IPXODI needs to be loaded after ODIDRV
, To Avoid Lock up problems later...

My NET.CFG:

```
Link Driver NE2000
PORT 280
INT 12
FRAME Ethernet_802.2
FRAME ETHERNET II
PROTOCOL IPXO ETHERNET_802.2
PROTOCOL BPQ 8FF ETHERNET_II
BPQPARAMS
ETH_ADDR FF:FF:FF:FF:FF:FF
```

Your driver section.
Your I/O address for card.
Your IRQ address for card.
Frame for Novell. Optional
Frame for BPQ.
Needed for Novell Optional
Needed for BPQ
, BPQ Driver info
; Set to broadcast to sense all
, nodes. If you change that to the other card's Ethernet address, it should be faster, and
, generate less traffic on a LAN. I use the broadcast because the LAN is just for BPQ.

My PORT section of BPQCFG.TXT:

```
PORT
ID=Ethernet Port
TYPE=EXTERNAL
PROTOCOL=KISS
INTLEVEL=125
; SPEED=9600
; CHANNEL=A
QUALITY=203
MAXFRAME=7
TXDELAY=0
SLOTTIME=100
PERSIST=255
FULLDUP=1
FRACK=7000
RESPTIME=100
```

; This is an external driver
; KISS or Netrom, both should work. I tried
, KISS only. (Note: 0x96=125)
; Should Not be needed.
; Should Not be Needed.
, Netrom quality for this port.
; Send as many **frames** as possible because
, of a dedicated high speed link.

, No need to wait for other stations, wire link.

```
RETRIES=1 0
PACLEN=234
USERS=8
ENDPORT
```

Run BPQCFG, try the drivers by hand, and make sure they work before rebooting. Most common problems are wrong setting for the network card in the NET.CFG or something spelled wrong in NET.CFG.

Where to Get it:

G8BPQ 4.08a:

Internet URL: ftp://ftp.tapr.org/tapr/software_lib/switch/bpq408a.zip

Other Sources: Unknown.

ODI Drivers for Use in this project:

Internet URL: ftp://ftp.novell.com/pub/updates/nwos/dsclnt12/vlmkt*.exe

** Where vlmkt*.exe = vlmktl.exe through vlmkt6.exe

** These are the install disks for Novell DOS Client.

Resources Used: (Reference List...)

G8BPQ, "G8BPQ's DRIVERS.DOC" from Version 4.08A of the software.

J. Chellis, R. Easlick, M. Moncur, A. Olsen, J. Tanner, "The CNE-4 Study Guide," SYBEX Books, 1996.

javAPRS

Implementation of the APRS Protocols in Java

Steve Dimse KO4HD

sdimse@bridge.net <http://www.bridge.net/~sdimse>

This paper describes my implementation of the Automatic Position Reporting System (APRS) protocols in the computer language Java. **APRS** is one of the most innovative uses of ham radio in recent years. javAPRS extends the usefulness of APRS to the internet.

Java Basics

Java was designed by Sun Microsystems as a language to promote the use of distributed computing resources over the internet. Based on the C language, with an object programming philosophy drawn from Smalltalk and Lisp, platform independence, and built-in security, it is a language uniquely suited to network programming.

The principle drawback to the use of Java at present is slow execution speed. Unlike traditional programs, which are compiled into machine specific object code before distribution, Java is compiled down to *byte code* which is then interpreted on the local machine, often executing at 10-25% of the speed of native code. There is reason to be hopeful, however. Several companies are working on 'just-in-time' compilers, which convert byte code to native code on the local machine just prior to execution. Also, the horsepower of today's computers makes even the interpreted version run at an acceptable speed in most applications.

The leading use of Java at present is to write programs, called *applets*, which are run within a World Wide Web (WWW) browser, such as Netscape. javAPRS can work in this fashion, or as a stand alone program. Security restrictions limit some of the interesting possibilities for javAPRS as an applet, such as using separate servers for map and APRS data, or connecting to multiple other computers to plot data from multiple LANs.

javAPRS Design

In the basic design of javAPRS, I have tried to create a system which can be used now by people without programming knowledge to add **APRS** data to their web pages. In addition, using the object oriented programming (OOP) features of Java, the system is designed to be easily extended by other Java programmers. This sort of extension does not require access to source code of javAPRS. The interfaces used by other programs will be posted in my web pages as they are finalized. Source code will not be freely available, but I will consider requests for the source code on a specific basis. The remainder of this paper discusses the use of javAPRS in web page creation. Programmers interested in extending javAPRS should contact me directly for more info.

javAPRS Applet Parameters

The basic syntax to call an applet in hypertext markup language (HTML) is:

```
<APPLET codebase = "javAPRS/" CODE="javAPRS.class" WIDTH=400 HEIGHT =300>
```

This executes the applet "javAPRS.class" in the directory "javAPRS" (relative to the HTML file the call is found in). It allocates a space of the size indicated in the browser window. Other HTML commands, such as <CENTER> work just as they would for other graphic elements such as images. The lines after APPLET tag consist of a series of parameters which are passed to the applet to control its behavior. Each parameter has a default value, usually the most commonly chosen option, and if the default parameter is desired, it need not be declared.

Map Parameters

At this time, javAPRS understands two kinds of maps. One or the other map must be used, or the applet will not run. The map files are stored in a subfolder "/maps" relative to the "codebase" named in the applet call.

```
<PARAM name = "dosMap" value = "anymap.map">
<PARAM name = "gifMap" value = "anymap.map">
```

Maps can be automatically or manually scaled.

```
<PARAM name = "autoScale" value = "true"> (Default true)
```

This will cause the map to be scaled to fit the window the applet is presently running in. If autoscaling is not used, then the following three parameters may be used to set the magnification and offset of the map. For now, the only way to figure out the value of these parameters is trial and error.

```
<PARAM name = "scale" value = "2.0"> (Default 1.0)
<PARAM name = "offsetX" value = "100"> (Default 0)
<PARAM name = "offsetY" value = "100"> (Default 0)
```

Two options work only with dosMaps, namely:

```
<PARAM name = "showMapLabels" value = "true"> (default true)
<PARAM name = "showAllMapLabels" value = "true"> (default false)
```

which will show either all map labels or those designated at the present scale.

Data Parameters

Data to be displayed by javAPRS is one of three types, either NMEA (only RMC and GGA are recognized at present), TNC data (raw data from a TNC which is the MacAPRS log file format), and HST files produced by dosAPRS. Any or all of the three types of data may be displayed, but only one file of each type can be used.

```
<PARAM name = "NMEAfile" value = "NMEA.data">
<PARAM name = "TNCfile" value = "ko4hd.data">
<PARAM name = "HSTfile" value = "marathon.data">
```

The way the data is displayed is affected by several parameters:

```
<PARAM name = "displayVectors" value = "true"> (default true)
```

This will draw vectors for course and speed info if present in a position report.

```
<PARAM name = "showCallsigns" value = "true"> (default true)
```

This prints a **callsign** next to each position report.

```
<PARAM name = "stationList" value = "true">
```

javAPRS can keep a station list for the stations that are 'heard' in the data stream. At the end of a file **readin**, the contents are dumped to the java console (select the option "Open Java Console" in Netscape. It will also speed up the redrawing once the data have been read, only the last position of each station will be plotted.

```
<PARAM name = "showNewStations" value = "false">
```

If **stationList** is true, then this option will show the name of each new station as it is heard. If the home station has been specified, the bearing and distance to the station will also be displayed.

There are a number of other, less important parameters that are available to fine tune the display to suit the user. Please refer to my web pages for more details.

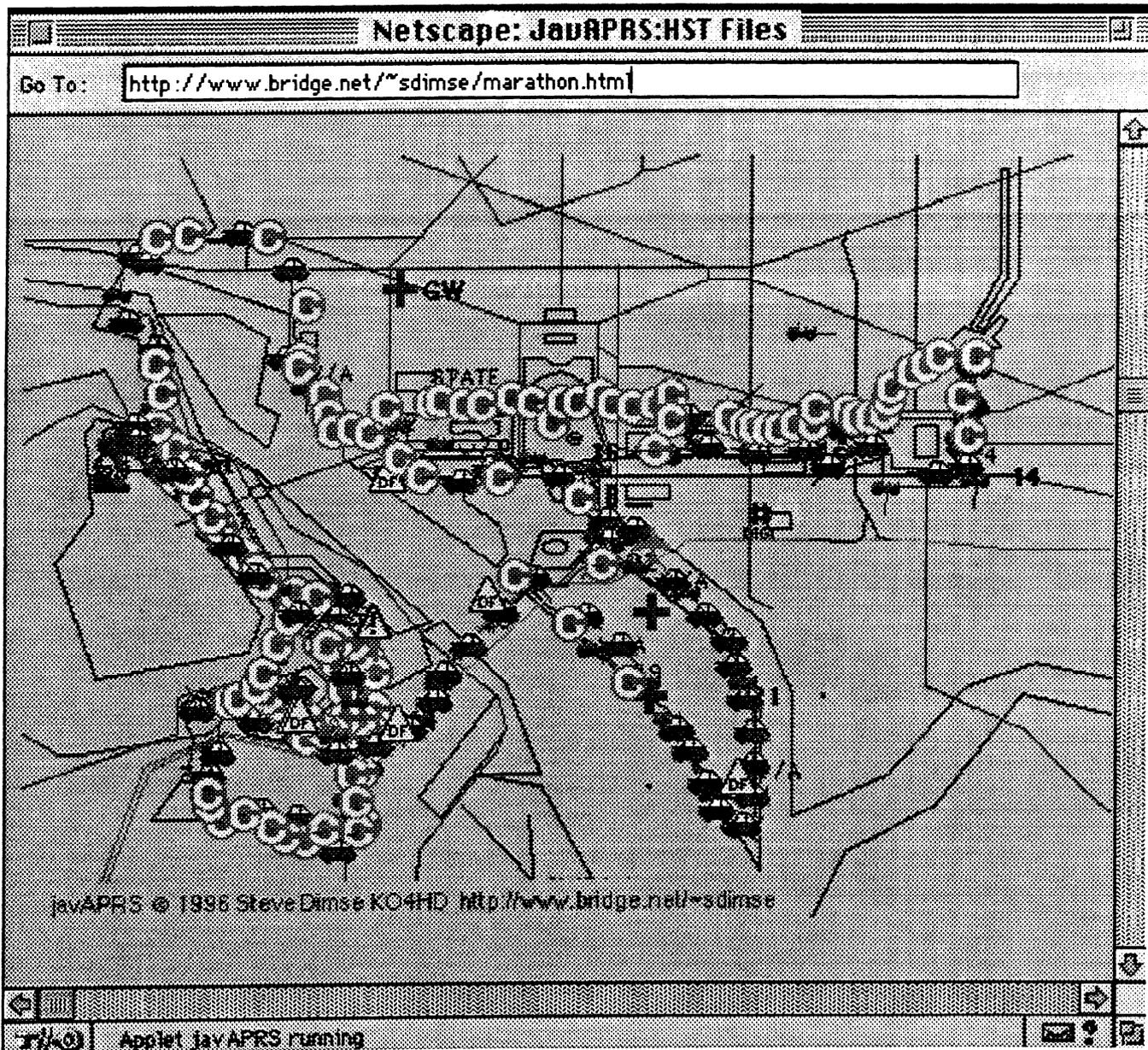
javAPRS Sample Applications

The **javAPRS** classes that exist now allow a person without programming skills to create a World Wide Web page containing an APRS map, and display a file containing positions of various APRS stations, objects, and track plots. More complete instructions may be found on my web site. Here are three examples, with the HTML code used to call the applet, and the URL's to reference them.

HST file replay

(<http://www.bridge.net/~sdimse/marathon.html>)

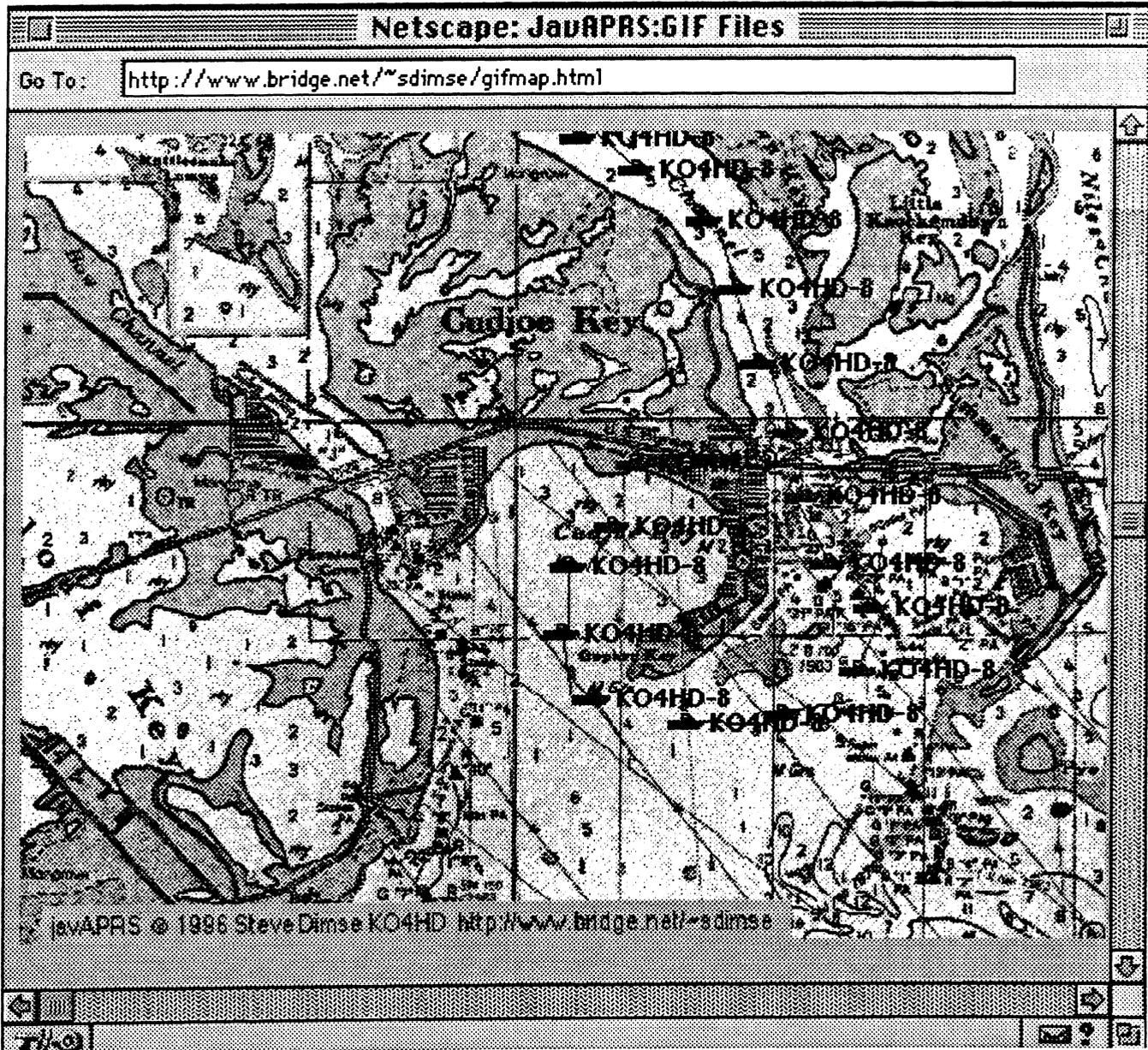
```
<APPLET codebase = "javAPRS/" CODE="javAPRS.class" WIDTH=500 HEIGHT=350>  
<PARAM name = "dosMap" value = "washdc.map">  
<PARAM name = "HSTfile" value = "marathon.hst">  
<PARAM name = "sleep" value = "50">  
<PARAM name = "stationlist" value = "false">  
<PARAM name = "showStationNames" value = "false">  
<PARAM name = "copyrightTop" value = "false">  
<PARAM name = "scale" value = "1.6">  
<PARAM name = "offsetx" value = "620">  
<PARAM name = "offsety" value = "310">  
</APPLET>
```



GIF files for maps

(<http://www.bridge.net/~sdimse/gifmap.html>)

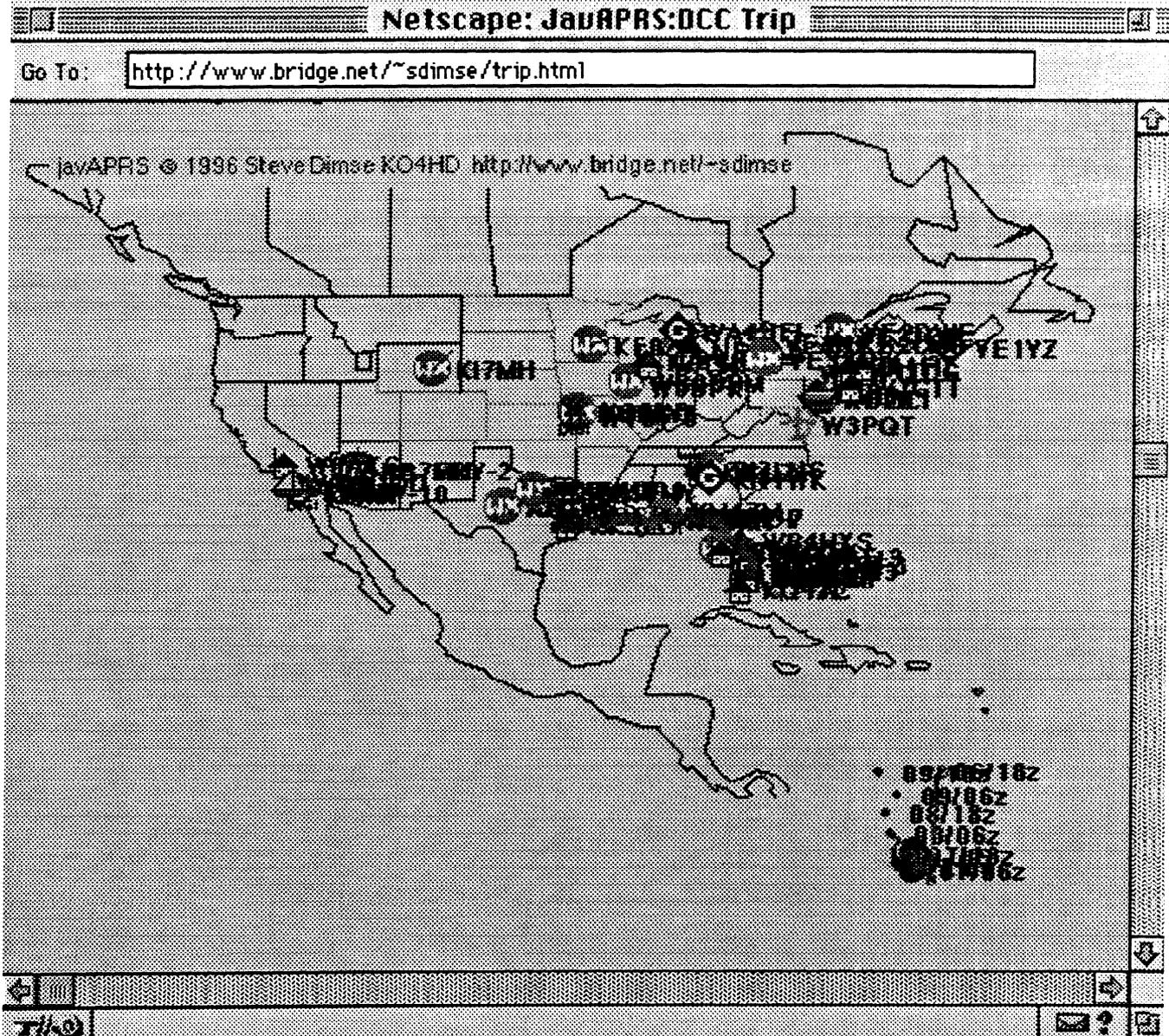
```
<APPLET codebase = "javAPRS/" CODE="javAPRS.class" WIDTH=500 HEIGHT =350>  
<PARAM name = "gifMap" value = "cudjoe.gif">  
<PARAM name = "gifMapLeft" value = "81.558">  
<PARAM name = "gifMapTop" value = "24.7">  
<PARAM name = "gifMapPPDh" value = "3800">  
<PARAM name = "gifMapPPDv" value = "3900">  
<PARAM name = "sleep" value = "700">  
<PARAM name = "LLfile" value = "boattrip.ll">  
<PARAM name = "stationlist" value = "false">  
<PARAM name = "copyrightTop" value = "false">  
</APPLET>
```



Display of TNC data capture

(<http://www.bridge.net/~sdimse/trip.html>)

```
<APPLET codebase = "javAPRS/" CODE=" javAPRS.class" WIDTH=500 HEIGHT =350>  
<PARAM name = "dosMap" value = "usa.map">  
<PARAM name = "TNCfile" value = "tnc.data">  
<PARAM name = "drawvectors" value = "true">  
<PARAM name = "homeID" value = "KO4HD-9">  
</APPLET>
```



Future directions

The samples above are static data, and although there is some degree of animation while the data is plotted, they could be done just as easily with screen captures and GIF files. Soon I hope to have the system able to access real time data over the net. This is where **javAPRS** can fill a unique role. The possibility exists to incorporate internet connectivity directly into the standalone APRS programs currently in use, allowing web users to see the data obtained at **various** APRS stations across the country in real time. Also, it would be possible to write a server program, that could be connected to many different **APRS** sites and share their data, creating a nationwide APRS network, no longer limited to 300 baud and the vagaries of HF propagation. I plan to pursue these plans, and hope to have some results to share by the 1997 DCC, if not sooner.

The Radio Amateur Digital System
Artificial Intelligence Project

Garry W. Joerger N5USG @ WO5H.#DFW.TX.USA.NOAM

Abstract

This paper outlines information on the Radio Amateur Digital System Artificial Intelligence Project from its conception to current capabilities and ongoing development. The platforms used in this project include amateur radio, facilities management and weather. Many more applications could be used. One of the areas could be radio frequency propagation. The applications chosen are intended to support amateur radio, weather service, emergency management and ultimately the public.

Background

Applications in artificial **intelligence** have been explored in a variety of degrees **by** many nations including the Japanese. From experiments in The Fifth Generation Attempt to today's medical and financial applications which are currently in use. This interest is no stranger to NASA and many other institutions. Applications in this area today serve as a vital tool **by** providing solutions to the problems facing **everyday** decisions.

1.0 INTRODUCTION

The purpose of this paper is to assist in understanding some of the issues involved in **successfully** applying expert systems **technology** **otherwise** known as artificial intelligence and neural network systems to the applications in the areas of ham radio, facilities management and weather. It is my intent to only provide a basic definition of expert **systems capabilities**, limitations and benefits.

2.0 EXPERT SYSTEM CAPABILITIES

In order to understand how expert systems can be used advantageously, you must first understand how knowledge processing in expert systems **differs** from conventional data processing. Expert systems are unique in their ability to process [knowledge], not just data. Knowledge processing **differs** from data processing in the type of information **used**, the techniques used to **analyze** the information, and in the form that the results of the knowledge processing are presented to the [user].

Conventional systems limit the developer to **data** representation using only numbers and text (parameters). They process data using complex [algorithms] that complete a discrete number of steps to reach a predetermined conclusion. Expert systems permit knowledge representation -- the encoding of human decision-making processes using **symbolic** terms or [symbols]. Because expert systems process knowledge, they are often referred to as

[knowledge-based systems].

The **ability** to represent knowledge in symbolic terms expands the range of analysis techniques that computers can apply to information thus enabling a system to emulate some aspects of human performance. The expert system uses problem solving procedures such as pattern-matching to reason about the **symbolic** terms. An example of [symbolic reasoning] by an expert system that determines the daily forecast is, “if the sky is cloudy, then the forecast might include rain.” This phrase could be used within an expert system. A conventional system would require that the symbolic terms such as “sky” and “cloudy” be defined concretely as numbers or text. The expert system uses pattern-matching on the phrases “**sky** is cloudy” and “forecast might include rain” to reach a conclusion about the forecast without requiring **definition** of each of the terms within the phrases. A conventional system would require that the developer define a set number of steps to determine the daily forecast. The expert system examines all possible solutions using the problem solving procedure that has been programmed into it.

The combination of problem solving procedures that are built into expert systems, together with the developer’s ability to define problems using **symbolic** terms, gives expert systems the **capability** to store and manipulate more complex relationships between **individual** pieces and groups of information than can be accomplished with the processing supported by conventional systems.

In addition to knowledge representation, expert systems also provide the capability to simplify the user/computer interaction. Expert systems can be programmed to employ more of the conventions that people use when communicating with each other. Expert systems can be designed with the ability to explain the “reasoning” used in reaching a recommendation and to **justify** their approach to a problem, much as people do. The more sophisticated expert systems employ a [natural language] processor to permit users to consult with the system using near-English rather than structured query languages or menus.

3.0 EXPERT SYSTEM BENEFITS

Expert system applications take advantage of the above capabilities in two ways. First, information becomes more accessible so people can make better decisions. Second where useful information is accessible, people can be more productive. We can be more productive in the use of controlling station facilities management, early warning notification and information resources through the application of expert systems. The two major benefits associated with expert systems are better decision making and increased productivity resulting in better facilities security, accurate environment control, increased awareness and faster notification.

3.1 BETTER DECISION MAKING

Expert systems improve the quality of decision making by providing a mechanism for pooling the knowledge of multiple experts and **making** that knowledge available to the control operators. This leveraging of knowledge

results in improved quality of complex work products such as, technical reports., and analyses that recommend actions.

Expert systems establish a basis for defensible decision-making **by** capturing and applying knowledge in verifiable form. For example, in developing work products such as technical reports and environmental analyses, a given set of inputs, no matter how complex, should result in consistent results given by closely similar data, advice, or recommendations. In addition, the process of reaching a conclusion can be **explicitly** demonstrated. This will ensure consistency in many decision-making activities.

3.2 INCREASED PRODUCTIVITY

Expert **systems** offer significant, measurable increases in **productivity** by **effectively** capturing the knowledge of experts and by minimizing the loss of [expertise] and knowledge due to attrition. Expert systems provide a means of extracting, storing, and sharing knowledge in a variety of disciplines. Thus, more people have access to expertise. In turn, the experts are freed from relatively mundane tasks to focus on demanding ones.

4.0 EXPERT SYSTEM LIMITATIONS

Expert systems provide valuable new capabilities, but **they** also have clear limitations. As with all new technology, developers must weigh the limitations associated with the use of expert systems technology against benefits. Because expert systems emulate human performance in decision making, they may be incorrectly thought of as having the capacity to make independent judgments. Expert systems are capable of communicating advice that has been coded into them. They are not capable of producing independent decisions. Their application is limited to strictly defined domains (i.e., areas of expertise where boundaries on what expertise should be included in the system can be defined); their performance degrades **dramatically** when dealing with information that is beyond those boundaries.

Expert systems can manipulate only symbolic information, that is, all “real” information that is collected by observing an event. For instance, temperature and humidity in the case of weather forecasting must be translated into a form acceptable to the expert system. Any errors and biases incorporated in the translation process will be accepted by the expert system without question. An example of a translation bias is if temperature measurements input to the system are in Fahrenheit when the logic or knowledge encoded into the system is based on Celsius measurements, then the conclusions reached will be invalid

5.0 PHASE IMPLEMENTATION STRUCTURE

The information provided below will provide an outline of what will take place with each phase of system installation.

PHASE I	SYSTEM PLANNING, PROCUREMENT AND CONSTRUCTION OF DIGITAL EQUIPMENT AND SOFTWARE BASED ON CONVENTIONAL PROGRAMMING
PHASE II	PRE-INSTALLATION TESTING AND EVALUATION
PHASE III	INSTALLATION OF DIGITAL, VOICE EQUIPMENT, CONVENTIONAL SOFTWARE AND OPERATIONS
PHASE IV	DIGITAL SYSTEM ENHANCEMENTS
PHASE V	EXPERT SYSTEM RESEARCH AND DEVELOPMENT

6.0 Current System Capabilities

The **system** is **currently** providing data information to storm spotters and **ARES/RACES** groups in the Dallas - Ft. Worth metroplex including adjacent counties. The sysop is also provided with facilities management information in the areas of station communications reliability and power (voltage) management. Station security is also handled by the system as well. Weather alerts are disseminated by way of **automatically** paging a beeper, uploading data via packet, phone modem and generates alerts using a voice synthesizer in a voice repeater system. The sysop is first notified of an impending alert and given the opportunity to either allow the alert to take place or to **intervene** in order to alter or cancel the alerts. This is the same concept used in nuclear power facilities. A weather **forecast** knowledge based application also provides input into the system. This allows notification of conditions which are current and also provides information as to conditions as they are developing. Data input for the system is provided in a manner so that storm tracking can also be accomplished. The sysop is provided with an audit trail of all data and provided with information on how the system has reached it's conclusion.

7.0 Conclusion

The project is in a long term development phase with future applications and implementations of enhancements as they are developed and tested. It is important to realize that an expert system is not a replacement for the experts of a given field, but a supplement. The high profile nature of an expert system can be addressed by classifying it as an assistant or advisor. This sets the user's expectations to a more reasonable level. If people believe that they are receiving accurate, "expert" information, then they may act on it without the proper skepticism. The expert system is a tool and the liability for any outcome is held by the decision-maker. With proper use, many advancements can be made by the use of expert systems.

Fast Flow Control in High-Speed Communication Networks

C. M. Kwan, R. Xu, and L. Haynes

Intelligent Automation Inc.
2 Research Place
Suite 202
Rockville, MD 20850
phone: (301) - 590 - 3155
fax: (301) - 590 - 9414
e-mail: ckwan@i-a-i.com

Keywords: flow control, high speed networks, time delay

Abstract

A new approach to flow control in high speed communication networks is proposed where the flow control problem is modeled as a dynamic system with time delay. The main advantage is that it can assure stability of system as well as maintaining certain throughput of the communication channel. Inside the controller, there is a term which predicts the future backlogs in the system. The controller is easy to implement. Simulation results show that the method offers significant less delay than existing methods.

1. Introduction

Tremendous amount of research has been done recently on the distribution of video, including how to accommodate both broadcast and interactive services (video conferencing and video on demand) onto a single system. It is expected that a significant portion of the communication capacity of future networks will be consumed by these signals. The transmission of these signals requires high speed networks such as broadband ISDN through the use optical fibers. To maintain certain maximum delay and minimum transmission rate is crucial in these applications related to voice and video signals; if the delay is beyond certain levels, the received signals will not be tolerable.

In many networks, even the traffic is routed optimally, there are situations where the total traffic into the network is much larger than the network can handle. If no strategies are present to limit the traffic into the network, queue sizes at bottleneck links will grow and packet delays will increase which may violate some maximum delay specifications. Moreover, the buffer space at some nodes may be exhausted. Some of the packets arriving at these nodes will have to be discarded and later retransmitted, thereby wasting communication resources.

There are several ways to flow control such as call blocking (circuit switched networks in telephony systems), packet discarding and packet blocking (packet switched networks), etc. One popular technique for implementing flow control is the window flow method. A virtual circuit between a transmitter A and a receiver B is under window flow control if there is an upper bound on the number of data units that have been transmitted by A and are not yet known by A to have been received by B . The upper bound is called the window size.

The window flow control is not suitable for high-speed wide area networks because the propagation delays are relatively large. An even more important reason is that windows do not regulate

end-to-end packet delays well and do not guarantee a minimum data rate. Voice, video services depend on upper bounds on delay and lower bounds on rate. High-speed wide area networks increasingly carry such traffic, and many lower-speed networks also carry such traffic, making windows inappropriate.

In [5], the flow control within a virtual circuit on a high speed network is modeled as a fluid-flow queue with a fixed propagation delay. Data from other virtual circuits are modeled as disturbances of the available service capacities. The model has the following features:

- The rate of data entering the network is controlled.
- The available service capacity is reduced by an unspecified but observable piecewise constant disturbances which correspond to capacity assigned to traffic from other virtual circuits.
- The information of backlog and disturbance are continuously fed back to the source through a reverse channel.
- There is a fixed delay on the forward and reverse channels.

In Section 2, we will consider the flow control model of a single channel. Section 3 introduces the control strategy. Extensive simulation results will be presented in Section 4. Finally, some comments will be included in Section 5.

2. Flow Control Model

Consider a fluid-flow channel with flow control shown in Fig. 1.

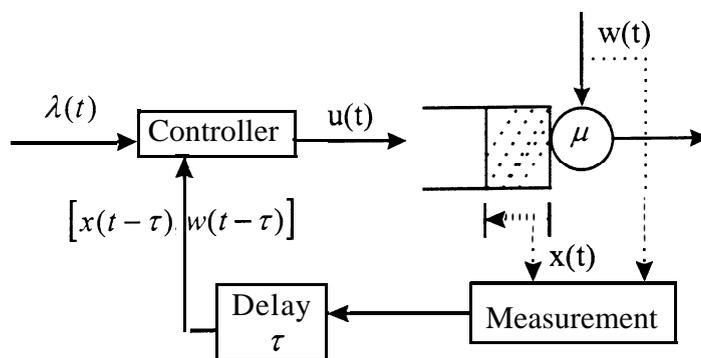


Fig. 1 A fluid flow channel model.

The dynamics of the backlog can be written as

$$\dot{x} = bu - b\mu(1 - w) \quad (2.1)$$

with x the backlog of data in the channel from the virtual circuit, μ the transmission rate of the channel, λ the traffic load on the virtual circuit, u the rate of traffic allowed into the network by the controller ($0 \leq u \leq \lambda$), w the portion of service capacity used by other virtual circuits, τ the lumped propagation delay from the source to the channel transmitter and back, including any additional fixed processing times, b a logical variable indicating whether data arriving at the channel transmitter can be served immediately, i.e.

$$b = \begin{cases} 1, & x > 0 \text{ or } u > \mu(1 - w) \\ 0, & x = 0 \text{ and } u \leq \mu(1 - w) \end{cases} \quad (2.2)$$

The disturbance $w(t)$ is assumed to be piecewise constant and driven by an unspecified train of weighted impulses $\sigma_w(t)$ as

$$\dot{w} = \sigma_w. \quad (2.3)$$

The arrival times and weights of the impulses are constrained only in that w is bounded by 0 and 1, and in that only a finite number of arrivals are permitted in a finite interval.

The control objectives are: (1) to drive the backlog x to zero; (2) to maintain the rate of traffic departing the channel at a rate $\mu\psi$ where ψ is a specified portion of channel capacity. The value of ψ may be chosen by the network designer for the best tradeoff of throughput and delay. Driving x to zero is equivalent to reducing the amount of queuing delay and maintaining certain rate of traffic is necessary in certain applications such as voice and video messages.

From basic system control theory, the presence of time delay in information feedback may cause the system to go unstable. In other words, the backlog x may go to very large values which also means the packet delay may go beyond certain maximum delay tolerance. The key to guarantee the stability of the system is to find some ways to compensate the propagation delay.

In the next section, we will propose a new method to flow control that can guarantee the maximum packet delay and, at the same time, can maintain a desired throughput rate. The controller also uses the fluid-flow model as described in [6]. Inside the controller, a term is present which can generate some predictive information about the future backlogs x in the system. This predictive information is crucial to the success of the control scheme since it guarantees that the time delay of packets will not go unstable. Thus the maximum time delay performance criterion can be assured. In addition, the new flow control method can also maintain certain throughput of the system which is very important in applications such as video distribution. Another advantage is that the controller is easy to implement.

3. Fast Flow Control Strategy

Throughout this paper, we will assume $b = 1$ since this is the case that needs the controller to be activated. We will divide the study into two cases. The first case deals with the ideal situation where the information in backlog $x(t)$ and disturbance $w(t)$ are fed back to the controller with no delay. The purpose of this is to set a benchmark to evaluate the performance of control strategies when delay is present. The second case deals with the more realistic situation where delay in information feedback is present.

3.1 No delay in information feedback

To maintain the system throughput at, for example, $\mu\psi$ with $0 < \psi < 1$. We propose the following controller

$$u = \mu\psi - kx - \mu w(t). \quad (3.1)$$

Substituting (3.1) into the system equation (2.1) yields

$$\dot{x} = -kx + \mu(\psi - 1) \quad (3.2)$$

which implies the queuing backlog will decay since $\psi < 1$. The rate of convergence is controlled by k . A control block diagram is shown in Fig. 2.

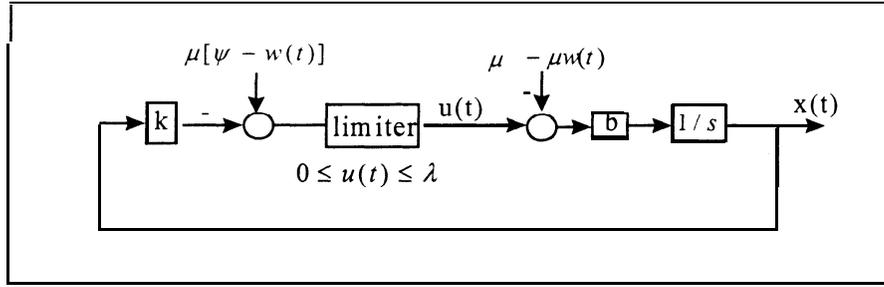


Fig. 2 Linear controller with no information delay.

Note that the steady state value of backlog is nonzero. It can be verified that, the steady state value of backlog equals to

$$x(\infty) = \frac{\mu(\psi - 1)}{k}. \quad (3.3)$$

Hence the value of backlog can be reduced to arbitrarily small values by choosing large enough k .

3.2 Time delay in information feedback

In systems with time delay, it is very hard to control them since all the available information happened τ seconds ago. Without predictive information about the system behavior, it would be very difficult to control the system. We propose the following control structure to deal with the time delay effect. The idea is to make use of the known dynamics of the system to compensate for the time delay effects. This technique is well known in control system community and is known as the Smith predictor. Fig. 3 shows a block diagram of the linear controller with compensation of time delay.

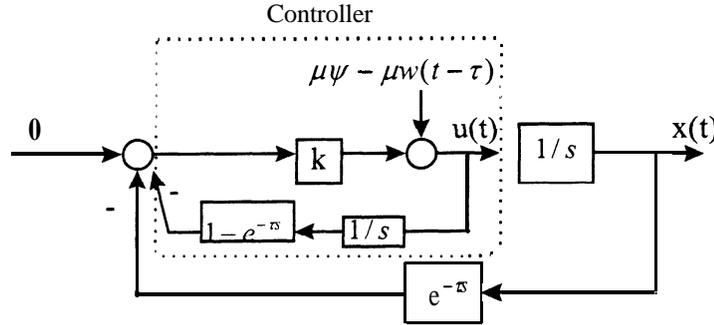


Fig. 3 Linear controller with information feedback delay.

From Fig. 3, it can be seen that the controller is of the form

$$u = \mu\psi - kx(t - \tau) - \mu w(t - \tau) - [x(t) - x(t - \tau)].$$

The term inside the square bracket provides predictive information about the system. The implementation of the above system is not difficult since the time delay is known in communication networks.

4. Simulation Studies

To facilitate our verification of the proposed method, we developed a simulation tool based on the SIMULINK environment. A diagram of the tool is shown in Fig. 4. The dynamics of the system were simulated with a propagation delay of 5 ms. The service capacity of the channel is 100 Mbps and

5. Conclusion

A new framework for flow control in high speed communication networks is proposed, which makes use of the known dynamics between backlog and disturbance. Within the framework, a new controller has been proposed which can guarantee the system stability and certain throughput rate. Extensive simulations have been done to verify the performance of the controller.

Future work includes extending the current method to tandem communication links.

References

- [1] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comput. Networks ISDN Syst.*, vol. 17, no. 1, pp. 1-14, 1989.
- [2] L. Dittmann, S. B. Jacobsen, and K. Moth, "Flow enforcement algorithms for ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, pp. 343-350, 1991.
- [3] A. E. Eckberg, D. T. Luan, and D. M. Lucantoni, "Meeting the challenge: Congestion and flow control strategies for broadband information transport," *Globecom '89*, pp.49.3.1-49.3.5.
- [4] R. Jain, "Congestion control in computer networks: Issues and trends," *IEEE Network Mag.*, pp.24-30, 1990.
- [5] S.-Y. R. Li, "Algorithms for flow control and call set-up in multi-hop broadband ISDN," *IEEE Infocom '90*, pp.889-895.
- [6] D. W. Browning, "Flow Control in High-Speed Communications Networks," *IEEE Trans. On Communications*, vol. 42, pp2480-2489, 1994.
- [7] D. Bersekas and R. Gallager, *Data Networks*, Prentice-Hall, 1992.
- [8] J. E. Slotine, and W. Li, *Applied Nonlinear Control*, Prentice-Hall, 1991.
- [9] F. L. Lewis, *Applied Optimal Control and Estimation*, Prentice-Hall, 1992.

Nonlinear Channel Equalization Using Fuzzy CMAC Neural Network

C. M. Kwan*, R. Xu*, L. Haynes*, J. D. Pryor **

* Intelligent Automation Inc.

2 Research Place

Suite 202

Rockville, MD 20850.

Tel. (301) - 590 - 3155

Fax. (301) - 590 - 9414

e-mail: ckwan@i-a-i.com

** United States Army

Space & Strategic Defense Command

Concepts and Applications Division

P.O. Box 1500

Huntsville, AL 35807-3801.

Keywords: nonlinear channel, equalization, Fuzzy CMAC

Abstract

It is well known that nonlinear distortion over a communication channel is now a significant factor hindering further increase in the attainable data rate in high-speed data transmission. Since the received signal over a nonlinear channel is a nonlinear function of the past values of the transmitted data pulses, it is not surprising that the linear equalizers do not work efficiently. We propose a new nonlinear equalizer that uses a new type of neural network called Fuzzy CMAC which combines the advantages of both fuzzy logic and CMAC (Cerebellar Model Arithmetic Computer) networks. The learning speed is an order of magnitude faster than conventional neural nets. Moreover, human expert knowledge in the form of linguistic rules can be easily incorporated into the scheme.

1. Introduction

In practical digital communications systems designed to transmit at high speed over band-limited channels, a succession of pulses transmitted through the channel at rates comparable to the channel bandwidth are smeared to the point that they are no longer distinguishable as well-defined pulses at the receiving terminal [1]. This is due to several factors. First, *linear amplitude and delay distortion* caused by the **nonideal** channel frequency response characteristic makes the symbols overlap. Also the delay distortion makes the zero crossings no longer periodically spaced. Second, signals transmitted through telephone channels are subject to other impairments such as nonlinear distortion, frequency offset, phase jitter, impulse noise, and thermal noise. *Nonlinear distortion* in telephone channels is due to the nonlinearities in amplifiers and companders used in the telephone system. This type of distortion is very difficult to correct. A small *frequency offset* (< 5 Hz) results from the use of carrier equipment in the telephone channel. Such an offset cannot be tolerated in high-speed digital transmission systems which use synchronous phase-coherent demodulation. The offset can be compensated for by the carrier recovery loop in the demodulator. *Phase jitter* is a low-index frequency modulation of the transmitted signal with the low frequency harmonics of the power line frequency (50 or 60 Hz). Phase jitter poses a serious problem in digital transmission of high rates. However, it can be tracked and compensated for at the demodulator. *Impulse noise* is an additive disturbance. It comes from the switching equipment in the telephone system. *Thermal noise* is also present at levels of 20 to 30 dB below the signal.

In this paper, we will deal with the signal distortion due to the linear and nonlinear channel characteristics and additive noises. Since frequency offset and phase jitter can be remedied by some other means, we will not discuss these impairments.

The popular approach to channel equalization is to model the channel as a linear finite-impulse-response (FIR) filter. Then an adaptive filter is used to cancel the distortion caused the channel. Various weight adjustment schemes such as LMS (Least Mean Square), RLS (recursive least-squares), and lattice algorithms are used to tune the filter parameters [1][2]. However, these schemes can only handle linear distortions. It is well known that nonlinear distortion over a communication channel is now a significant factor hindering further increase in the attainable data rate in high-speed data transmission. Since the received signal over a nonlinear channel is a nonlinear function of the past values of the transmitted data pulses, it is not surprising that the linear equalizers do not work efficiently. Therefore, an efficient and effective equalizer should possess certain capability that can learn the nonlinear behavior of the channel characteristics. In [4] and [6], polynomial adaptive filters were developed for nonlinear channel equalization. In [5], multilayer perceptrons have been used as equalizers. In [3], fuzzy adaptive filters were used to cancel the nonlinear distortion.

Since nonlinear channels cover many kinds of nonlinear distortions, it is hard to single out one method that is dominant. Hence it is necessary to develop new nonlinear equalizers and this is the goal of this paper. We propose a new nonlinear equalizer that uses a new type of neural network called Fuzzy CMAC which combines the advantages of both fuzzy logic and CMAC (Cerebellar Model Arithmetic Computer) networks. The learning speed is an order of magnitude faster than conventional neural nets. Moreover, human expert knowledge in the form of linguistic rules can be easily incorporated into the scheme.

The paper is organized as follows. In section 2, we will introduce some backgrounds on Fuzzy CMAC. Then we will describe the channel equalization problem and our approach in section 3. Simulation results will be included. Finally, conclusions will be drawn in Section 4.

2. Fuzzy CMAC Neural Network

2.1 Comparison of a CMAC Network with a Fuzzy Logic Controller (FLC)

Two decades ago, a unique neural network model called a Cerebellar Model Arithmetic Computer (CMAC) was established by J. Albus [8] based on a model of the human memory and neuromuscular control system. The CMAC is a perceptron-like associative memory that performs nonlinear function mapping over a particular region of a function space. A CMAC network has the capability to learn an unknown nonlinear mapping by examples, and to reproduce multiple outputs in response to multiple inputs. Because of its table look-up mechanism, and its hash-code based mapping structure, CMACs are able to cope with high dimensional input/output applications without severely deteriorating their processing speed and performance.

Fuzzy set theory was initially proposed by L. Zadeh as a tool to model the imprecision that is inherent in human reasoning, especially when dealing with complexity. The fuzzy theory has seen its most widespread application in the area of control. Controllers using control laws specified with fuzzy set theory (or fuzzy logic) are known as fuzzy controllers. Such controllers are easier (relative to non-fuzzy controllers) to design, especially in cases where the laws are non-linear and the systems are complex.

A fuzzy logic controller consists of an input fuzzification module, a set of fuzzy control rules based on which the approximate reasoning technique is used to make a control decision, and an output defuzzification module.

Fuzzy logic controllers differ from classical math-model controllers in that they do not require an explicit mathematical model of how control outputs functionally depend on control inputs. Also fuzzy logic controllers allow designers to incorporate human knowledge into the control decision making process.

The advantages of a CMAC over a FLC are:

- There are very efficient learning laws to update the values of weights based on experience and examples.
- There is a random mapping mechanism to reduce the physical memory requirement for multiple input and high resolution applications.
- There exist efficient input encoding schemes for high dimensional input vectors.

The advantages of a FLC over a CMAC are:

- It is possible to interpret the implication of weight values using linguistic labels.
- The membership functions and the firing strengths contain additional information as to how close the input vector is to each linguistic variable. Therefore the number of input space partitions may be smaller to achieve the same generalization and output smoothness.
- The fuzzy rules can take a variety of forms while only numeric values can be associated with CMAC associative memory locations.
- There are many methods to construct a fuzzy control knowledge base, such as expert's experience and knowledge.

In the next section, we will propose a new neural network called Fuzzy CMAC which combines the advantages of both fuzzy logic and CMAC network.

2.2 Fuzzy CMAC Architecture

In this section, we present the novel concept of a Fuzzy CMAC neural network. This novel network was developed by Intelligent Automation Inc. (IAI). The idea is to combine the advantages of the FLC and the CMAC (Cerebellar Model Arithmetic Computer) and eliminate the respective disadvantages of the two. IAI has successfully applied this network to active vibration control, finger print analysis, and chaotic time-series prediction. Fig. 1 illustrates the architecture of the Fuzzy CMAC. The Fuzzy CMAC inherits the preferred features of arbitrary function approximation, learning, and parallel processing from the original CMAC neural network, and the capability of acquiring and incorporating human knowledge into a system and the capability of processing information based on fuzzy inference rules from fuzzy logic. The combination of a neural network and fuzzy logic yields an advanced intelligent system architecture.

At the input stage, the Fuzzy CMAC uses the fuzzification method of a FLC as its input encoding scheme. Fuzzy rules can be assigned to each associative memory cell. These rules may not necessarily have a crisp consequent part. The output generation uses a **defuzzification** approach which sums weighted outputs of the activated rules based on the firing strengths s_i . The overall mapping function of a Fuzzy CMAC can be formalized as

$$z(\mathbf{u}) = \sum_{p=1}^{A4} s_p w_p \quad (2.1)$$

where $u = [u_1 \ u_2 \ \dots \ u_N]^T$ is the input vector. w_p , $p = 1, 2, \dots, M$, are the weights of the network.

$A_4 = j_1$ if $N = 1$ and $M = \sum_{i=2}^N (j_i - 1) \prod_{l=1}^{i-1} m_{l-1} + j_1$, for $N > 1$. $i = 1, 2, \dots, N$. m_i is the number of knot points on the i^{th} input dimension. The j_i^{th} knot point on the i^{th} input dimension is denoted as u_{i,j_i} , $j_i = 1, 2, \dots, m_i$. s_p , $p = 1, 2, \dots, M$, are the firing strength of the fuzzy rules. They are calculated as

$$s_p = \mu_{1,j_1}(u_1) * \mu_{2,j_2}(u_2) * \dots * \mu_{N,j_N}(u_N) \quad (2.2)$$

where $*$ denotes the T-norm operator. There are many types of T-norms such as the min and product operators. Throughout this paper, we choose the T-norm as the product inference method since it is easy to implement. Hence, (2.2) can be written as

$$s_p = \mu_{1,j_1}(u_1) \mu_{2,j_2}(u_2) \dots \mu_{N,j_N}(u_N) = \prod_{i=1}^N \mu_{i,j_i}(u_i) \quad (2.3)$$

where $\mu_{i,j_i}(u_i)$ is the j_i^{th} membership function of the i^{th} input.

A Fuzzy CMAC neural network combines the advantages listed for CMAC and FLC. One desirable feature the Fuzzy CMAC inherits from the CMAC model is that the receptive field of the sensing element has limited width. This means that there are only a small number of sensing elements to be activated for any sensor reading. In conventional neural nets such as those based on multiple layer feedforward structure and backpropagation learning algorithms, all neurons are required to perform computation in order to compute the forward mapping or to perform learning. In Fuzzy CMACs, operations are localized and only a small subset of all the neurons need to be computed. Our initial comparisons of fuzzy CMACs with conventional neural nets show orders of magnitude increase in the speed of both function mapping and learning for typical problems on conventional computational hardware. Comparing the activating function of CMAC to the linguistic variables in the Fuzzy CMAC, one can view the activating function as the membership function of the Fuzzy CMAC input variables. Fig. 2 shows the comparison.

In terms of the control knowledge rule-base, the proposed Fuzzy CMAC differs from Albus's CMAC [8] in that the weight values can be interpreted as knowledge rules through linguistic variables. This feature permits us to validate a learned Fuzzy CMAC in terms of the reasonableness of the learning results. This unique feature also provides a practical channel for knowledge acquisition. A knowledge base (a set of rules) can be established based on the learning results of a Fuzzy CMAC.

On the other hand, the Fuzzy CMAC distinguishes itself from Zadeh's fuzzy controller in that it is able to build a learning control system starting with an empty knowledge base. The linguistic knowledge of human experts can be incorporated into these rules. Based on the initial knowledge base, a self learning algorithm is employed to modify the existing rules in order to improve the system performance.

2.3 Fuzzy CMAC Mapping Using B-Spline Membership Function

The fuzzy membership function can be chosen with considerable freedom, and this freedom can be used to optimize system performance. Membership functions should be computationally simple, flexible, and continuous in order to optimize the Fuzzy CMAC system design. We have compared many candidates for membership function, such as bell shape Gaussian functions, sinusoidal functions, etc., and we believe that the B-Spline function family possesses certain preferred properties that make them well suitable for the Fuzzy CMAC membership function. Let the j_i^{th} knot point on the i^{th} input dimension

be denoted as u_{i,j_i} . Throughout this paper, we use the third order B-spline function. The firing strength of a fuzzy membership function $\mu(u_i)$ is the value of the function μ at the input u_i . For example, if u_i lies in the range of $[\mu_{i,j_i-1}(u), \mu_{i,j_i}(u)]$, then the firing strength of the j_i^{th} membership function will be

$$S_{i,j_i} = \frac{u - u_{i,j_i-2}}{u_{i,j_i} - u_{i,j_i-2}} \frac{u_{i,j_i} - u}{u_{i,j_i} - u_{i,j_i-1}} + \frac{u_{i,j_i+1} - u}{u_{i,j_i+1} - u_{i,j_i}} \frac{u - u_{i,j_i-1}}{u_{i,j_i} - u_{i,j_i-1}}. \quad (2.4)$$

The third order B-Spline membership functions have the following desirable properties:

- (a) Positivity: $\mu_{i,j_i}(u_i) > 0$ for all $u_i \in [u_{i,j_i-3}, u_{i,j_i}]$.
- (b) Compact support: $\mu_{i,j_i}(u_i) = 0$ for all u_i not belonging to $u_i \in [u_{i,j_i-3}, u_{i,j_i}]$
- (c) Normalization: $\sum \mu_{i,j_i}(u_i) = 1$ for any u_i .
- (d) Derivatives exist and can be recursively calculated.

It should be noted that the positivity property is not that important in Fuzzy CMAC. However, the other three properties require more attention. The existence of derivatives is very important for many real-time control applications where the tracking errors are back-propagated through the Fuzzy CMAC. Compact support property means that, for a given input vector, only a small number of the membership functions will be fired and need to be computed. For our third order B-spline function, only 3 membership functions will be activated in each input dimension. Thus only a small number of weights need to be updated. In other words, our Fuzzy CMAC enjoys the localization property which represents a considerable amount of computational savings when compared to a general feedforward backpropagation neural network which must calculate all the network weights whenever there is a new input vector. The smoothness of the B-spline functions also give the Fuzzy CMAC the generalization property which means similar inputs will give similar outputs. The normalization property will help in simplifying the computations. Since the output of a Fuzzy CMAC can be expressed as

$$z(u) = \frac{\sum_{j_N=1}^{m_N} \sum_{j_{N-1}=1}^{m_{N-1}} \dots \sum_{j_1=1}^{m_1} \left[\prod_{i=1}^N \mu_{i,j_i}(u_i) \right] w(j_1, j_2, \dots, j_N)}{\sum_{j_N=1}^{m_N} \sum_{j_{N-1}=1}^{m_{N-1}} \dots \sum_{j_1=1}^{m_1} \prod_{i=1}^N \mu_{i,j_i}(u_i)}, \quad (2.5)$$

it seems from the above formula that the calculation of the numerator as well as the denominator of the $z(u)$ both involve a summation of $M (= \prod_{i=1}^N m_i)$ terms, each in turn requires N multiplications. By using the normalization property of the B-spline membership function, the calculation of the $z(u)$ becomes much simpler. This fact is proven in the following theorem.

Theorem 1

Because of the normalization property of B-spline function, the denominator in (2.1) always takes the value 1, e.g.

$$\sum_{j_N=1}^{m_N} \sum_{j_{N-1}=1}^{m_{N-1}} \dots \sum_{j_1=1}^{m_1} \prod_{i=1}^N \mu_{i,j_i}(u_i) = 1 \quad (2.6)$$

Proof: See [7].

2.4 Fuzzy CMAC Universal Approximation Theorem

We have proven the following universal approximation theorem for the Fuzzy CMAC neural network. The theorem essentially guarantees that the Fuzzy CMAC neural network can uniformly approximate any nonlinear continuous function over a specific compact region to any degree of accuracy. The universal approximation theorem provides us with justification for applying the Fuzzy CMAC to almost any nonlinear system modeling problem.

Theorem 2

For any given real continuous function $g(u)$ on a compact set $U \subset R^N$, and arbitrary $\varepsilon > 0$, there exists a Fuzzy CMAC neural network $z(u)$ in the form of (2.1) such that

$$\sup|z(u) - g(u)| < \varepsilon \quad \text{for} \quad \forall u \in U \quad (2.7)$$

where $u = [u_1 \ u_2 \ \dots \ u_N]^T$ is the input vector; $w(j_1, j_2, \dots, j_N)$ are the weights of the Fuzzy CMAC; and μ_{i,j_i} are the membership function values.

Proof: See [7].

2.5 Learning Algorithm for the Fuzzy CMAC

The conventional CMAC learning process is designed so that the correction initiated by an output error is evenly distributed among all weights that contribute to the output, regardless of their true proportion of contribution to the output. In the Fuzzy CMAC, a more intelligent method is adopted in which the correction of the output error is distributed to the weights in proportion to their contribution. The weight updating algorithm is described below.

Given a training pair (u, z_d) where $u = [u_1, u_2, \dots, u_N]^T$ is input vector and z_d is the desired output, update weights and knot points associated with fired fuzzy rules, such that the error between z and z_d is reduced. Suppose there is only one output and many inputs. The Fuzzy CMAC mapping can be expressed in terms of the product of firing strength and weights

$$z = \sum_{p=1}^M s_p w_p \quad (2.8)$$

The learning algorithm is targeted to reduce error between the desired output z_d and the actual output of Fuzzy CMAC by adjusting the values of weights

$$E = \frac{1}{2}(z_d - z)^2 = \frac{1}{2}e^2 \quad (2.9)$$

The adjustment of the weights is based on the following rules

$$\Delta w_p = -\beta e s_p, \quad p=1, 2, \dots, M \quad (2.10)$$

where β is the learning rate. These learning rules differ from the original CMAC learning rule proposed by Albus in that the adjustment of a weight is proportional to its contribution (measured by its firing strength) to the output signal based on which the error is generated. The conventional CMAC distributes the output error evenly among all the contributing weights.

3. Nonlinear Channel Equalization Using Fuzzy CMAC

A typical digital communication system is shown in Fig. 3 [3]. The channel includes the effects of the transmitter filter, the transmission medium, the receiver matched filter, and other components. The

transmitted data sequence $s(k)$ is assumed to be an independent sequence taking values from $\{-1,1\}$ with equal probability. The inputs to the equalizer, $x(k), x(k-1), \dots, x(k-n+1)$ are the channel outputs corrupted by an additive noise $\eta(k)$. The task of the equalizer at the sampling instant k is to produce an estimate of the transmitted symbol $s(k-d)$ using the information contained in $x(k), x(k-1), \dots, x(k-n+1)$. The integers n, d are known as the order and lag of the equalizer, respectively.

The equalization problem can be formulated as follows. Similar to [5], we define

$$\begin{aligned} P_{n,d}(1) &= \{\underline{\hat{x}}(k) \in R^n | s(k-d) = 1\}, \\ P_{n,d}(-1) &= \{\underline{\hat{x}}(k) \in R^n | s(k-d) = -1\} \end{aligned} \quad (3.1)$$

where

$$\underline{\hat{x}}(k) = [\hat{x}(k), \hat{x}(k-1), \dots, \hat{x}(k-n+1)]^T.$$

Note that $\underline{\hat{x}}(k)$ is the noise-free output of the channel, and $P_{n,d}(1)$ and $P_{n,d}(-1)$ represent the two sets of possible channel noise-free output vectors $\underline{\hat{x}}(k)$ that can be produced from sequences of the channel inputs containing $s(k-d) = 1$ and $s(k-d) = -1$, respectively. The equalizer can be characterized by the function

$$g_k: R^n \rightarrow \{-1,1\} \quad (3.2)$$

with

$$\hat{s}(k-d) = g_k(\underline{x}(k))$$

where

$$\underline{x}(k) = [x(k), x(k-1), \dots, x(k-n+1)]^T$$

is the observed channel output. Let $p_1[\underline{x}(k) | \underline{\hat{x}}(k) \in P_{n,d}(1)]$ and $p_{-1}[\underline{x}(k) | \underline{\hat{x}}(k) \in P_{n,d}(-1)]$ be the conditional probability density functions of $\underline{x}(k)$ given $\underline{\hat{x}}(k) \in P_{n,d}(1)$ and $\underline{\hat{x}}(k) \in P_{n,d}(-1)$, respectively. It was shown in [5] that the equalizer which is defined by

$$f_{opt}(\underline{x}(k)) = \text{sgn}[p_1(\underline{x}(k) | \underline{\hat{x}}(k) \in P_{n,d}(1)) - p_{-1}(\underline{x}(k) | \underline{\hat{x}}(k) \in P_{n,d}(-1))] \quad (3.3)$$

achieves the minimum bit error rate for the given order n and lag d , where $\text{sgn}(y) = 1$ (-1) if $y \geq 0$ ($y < 0$). If the noise $\eta(k)$ is zero-mean and Gaussian with covariance matrix

$$Q = E[(\eta(k), \dots, \eta(k-n+1))(\eta(k), \dots, \eta(k-n+1))^T], \quad (3.4)$$

then

$$\begin{aligned} & p_1(\underline{x}(k) | \underline{\hat{x}}(k) \in P_{n,d}(1)) - p_{-1}(\underline{x}(k) | \underline{\hat{x}}(k) \in P_{n,d}(-1)) \\ = & \sum_{\underline{\hat{x}}_+ \in P_{n,d}(1)} \exp\left[-\frac{1}{2}(\underline{x}(k) - \underline{\hat{x}}_+)^T Q^{-1}(\underline{x}(k) - \underline{\hat{x}}_+)\right] - \sum_{\underline{\hat{x}}_- \in P_{n,d}(-1)} \exp\left[-\frac{1}{2}(\underline{x}(k) - \underline{\hat{x}}_-)^T Q^{-1}(\underline{x}(k) - \underline{\hat{x}}_-)\right]. \end{aligned}$$

Let us consider a nonlinear channel described by the following discrete model

$$\hat{x}(k) = s(k) + 0.5s(k-1) - 0.9[s(k) + 0.5s(k-1)]^3 \quad (3.5)$$

and the white Gaussian noise $\eta(k)$ with $E(\eta^2(k)) = 0.2$. The optimal decision boundary for this case is shown in Fig. 4. The elements of the sets $P_{2,0}(1)$ and $P_{2,0}(-1)$ are illustrated in Fig. 4 by the “o” and “*” respectively. It can be seen that the boundary is very nonlinear. Linear equalization techniques will not perform well for such a nonlinear boundary.

Here we propose to use Fuzzy CMAC as a nonlinear equalizer, The procedure consists of two steps.

Step 1: Generation of training data

For Fuzzy CMAC to be an efficient nonlinear equalizer, a thorough training of various possible cases is necessary. From Fig. 4, it can be seen there are 8 possible noise free elements. We generate 10,000 data pairs around the 8 noise free elements by adding Gaussian noise to them. The data sets are shown in Fig. 5. The desired outputs of the network are either 1 or -1. The Fuzzy CMAC equalizer has two inputs and one output. The training of the network is done by evaluating the error and the error is used to adjust the weights in the Fuzzy CMAC. We used 10 membership functions for each input of the equalizer.

Step 2: Performance test

To test the performance of the nonlinear equalizer, we generate another 100,000 data pairs with various noise amplitudes. The output of the Fuzzy CMAC is compared with the desired values to check whether the decision is right or wrong. Due to the generalization property of the Fuzzy CMAC, the network can make good decision about untrained cases. The overall performance is shown in Fig. 6, which shows the bit error rate versus the signal-to-noise ratio.

4. Conclusion

A new nonlinear equalizer using Fuzzy CMAC neural network is proposed to perform equalization for nonlinear channels. The training speed of the network is an order of magnitude faster. Moreover, the human expert knowledge can be incorporated into the equalizer design. Simulation results show that the performance is similar to existing approaches.

References

- [1] J. Proakis, *Digital Communications*, McGraw-Hill Book Co., New York, 1989.
- [2] C.F.N. Cowan, "Performance Comparisons of Finite Linear Adaptive Filters," *IEE Proc.*, vol. 134, pt. F, no.3, pp. 211-215, 1987.
- [3] L. Wang, *Adaptive Fuzzy Systems and Control*, Prentice-Hall, New Jersey, 1994
- [4] E. Biglieri, A. Gersho, R.D. Gitlin, and T.L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," *IEEE J. on Selected Areas in Communications*, no.5, pp. 765-777, 1984.
- [5] S. Chen, G.J. Gibson, C.F.N. Cowan, and P.M. Grant, "Adaptive equalization of finite nonlinear channels using multilayer perceptrons," *Signal Processing*, vol. 20, pp. 107- 119, 1990.
- [6] D.D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission systems," *The Bell System Technical Journal*, vol. 57, no. 7, pp. 2589-2611, 1978.
- [7] Z. Geng, H. Xu, and L.S. Haynes, "Fuzzy CMAC Neural Network Application to Flight Vehicle Control," *Phase I report to KS. Army*, November, 1994.
- [8] J. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Trans. ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 97, pp. 220-227, 1975.

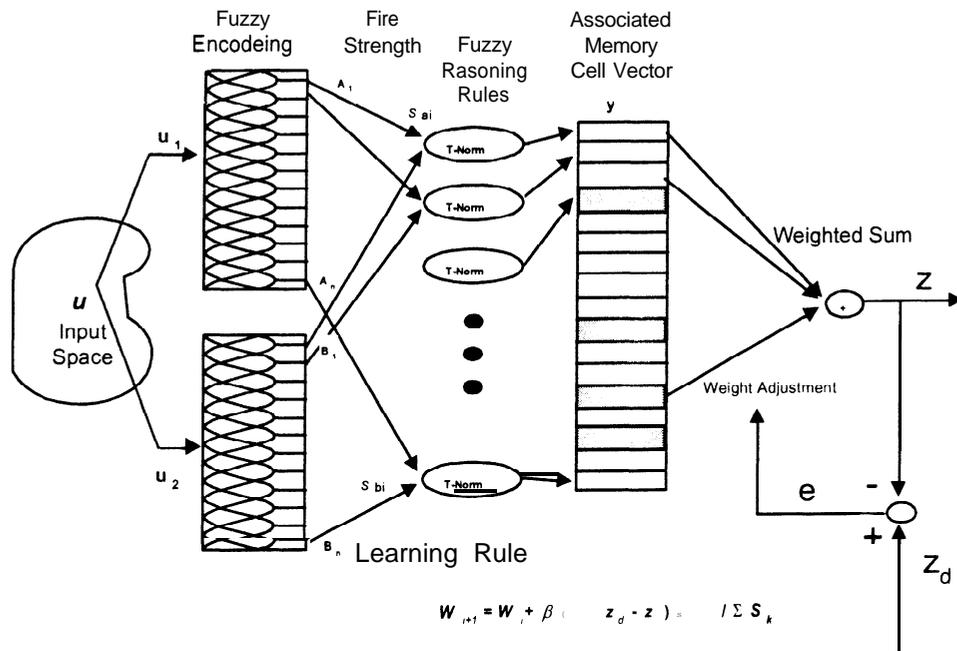


Fig. 1 Architecture of Fuzzy CMAC Neural Network

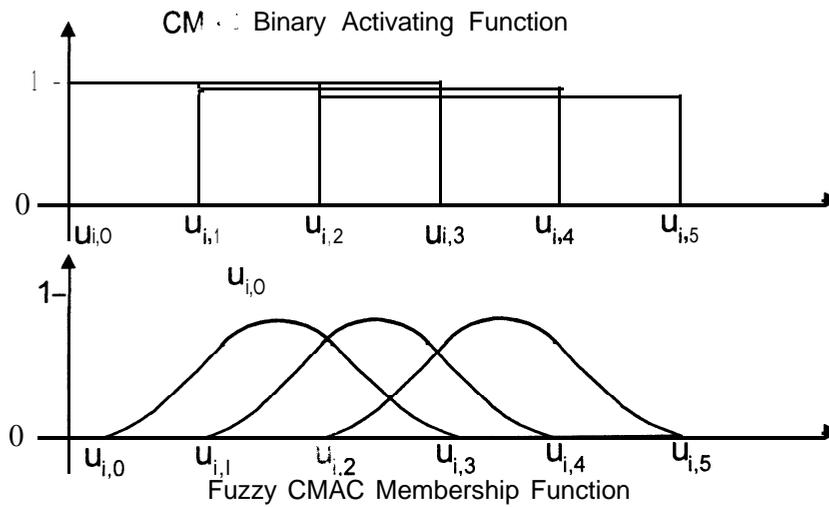


Fig. 2 Comparison of CMAC Binary Activating Function with Fuzzy CMAC Membership Function

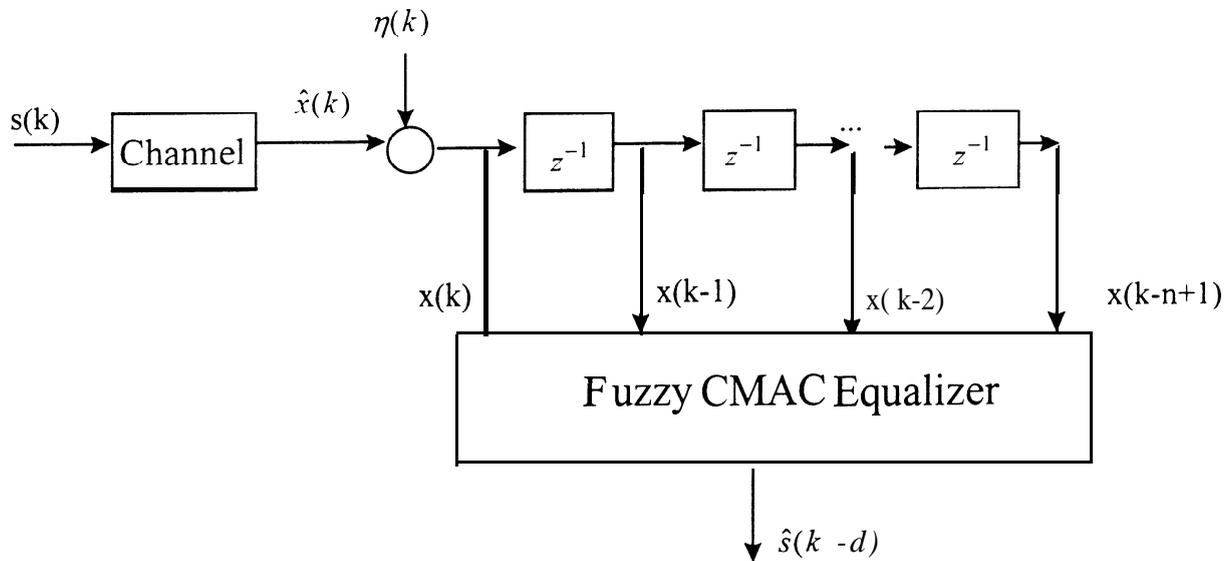


Fig. 3 Data transmission system.

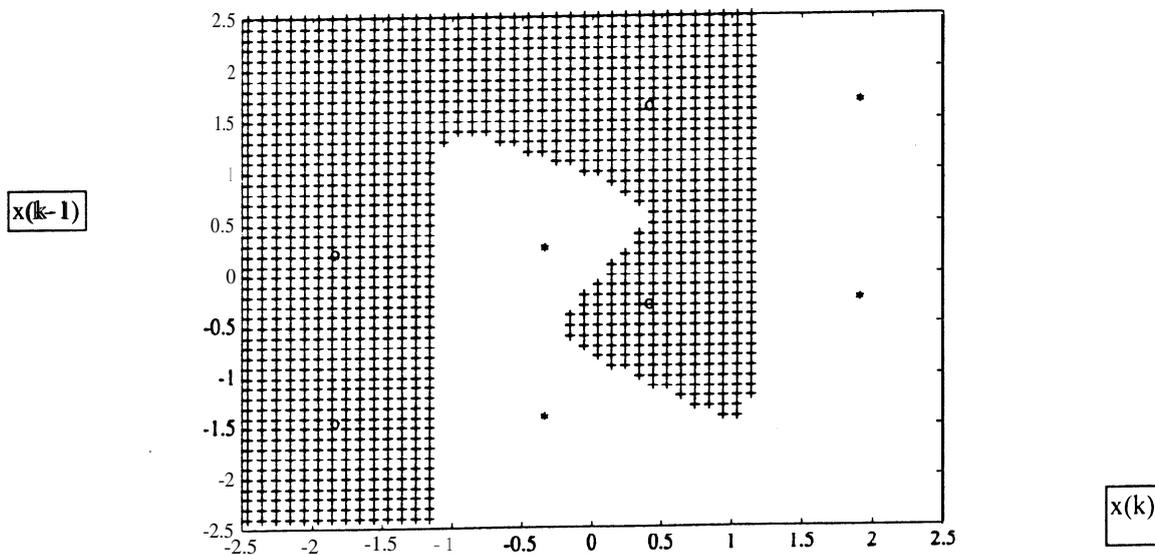


Fig. 4 Optimal decision region for channel (3.5).

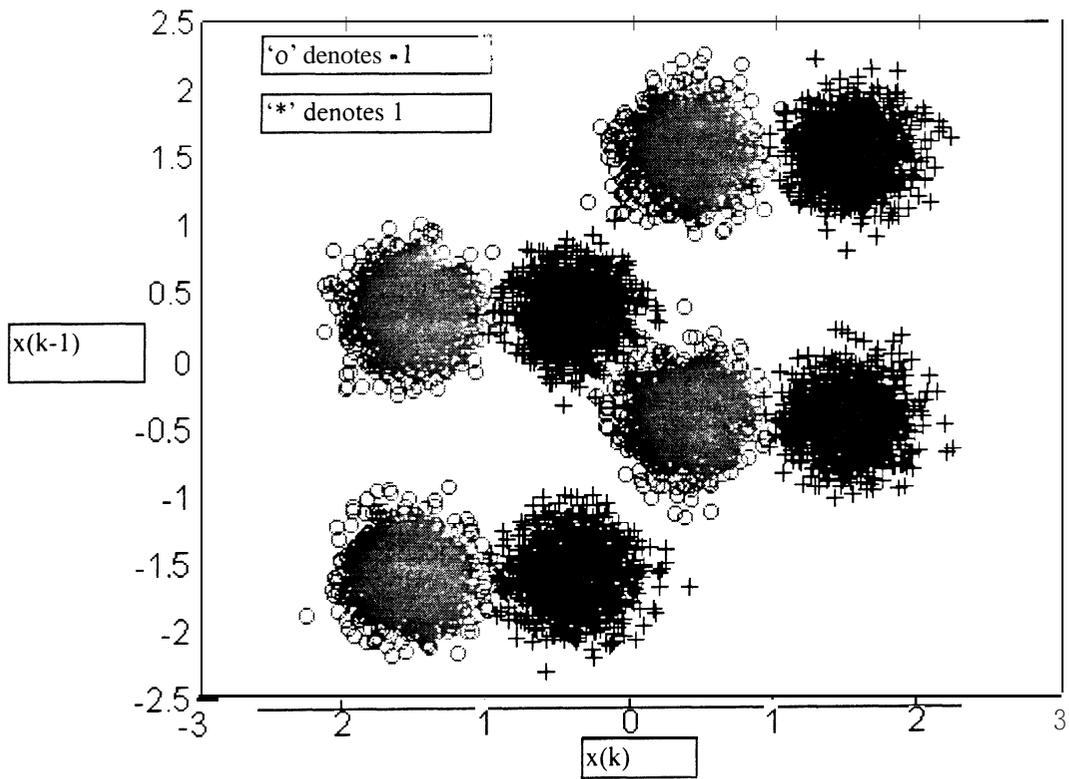


Fig. 5 Training data.

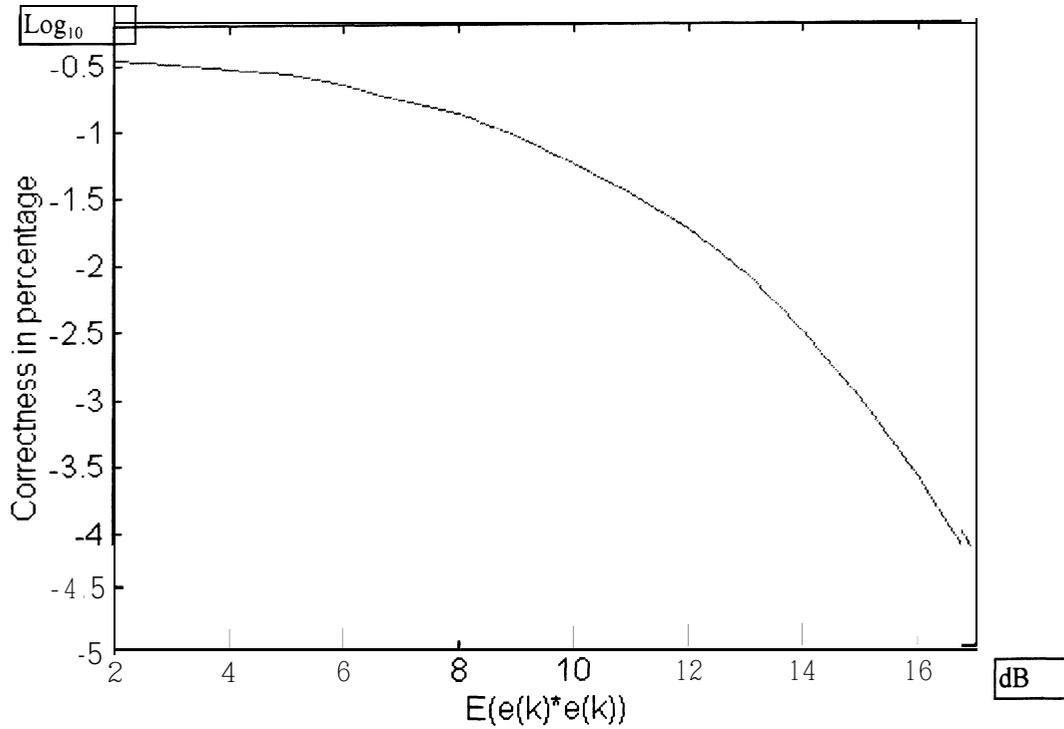


Fig. 6 Bit error rate versus signal to noise ratio.

Optimization of Phase-Locked Loops With Guaranteed Stability

C. M. Kwan, H. Xu, C. Lin, and L. Haynes

Intelligent Automation Inc.
2 Research Place
Suite 202
Rockville, MD 20850
phone: (301) - 590 - 3155, fax: (301) - 590 - 9414, e-mail: ckwan@i-a-i.com

Keywords: stability, phase-locked loop, multi-objective optimization

Abstract

Phase-locked loops (PLL) have found applications in many industrial applications such as communication and control systems. The key requirements are stability and loop performance in terms of signal-to-noise ratio and tracking errors. Here we present a two-step approach to PLL design. First, we present a Lyapunov approach to analyze the loop stability. The parameter range that can guarantee stability can be easily derived in the process. Second, we present a multi-objective optimization method that can search a set of values within the above range of parameters to achieve an optimal trade-off between loop bandwidth, transient and steady-state performance. Simulation results are contained to illustrate the performance of our procedure.

1. Introduction

The principle of phase-locked loops (PLL) is simple; it consists of a phase detector, a loop filter, and a voltage controlled oscillator [1] [3-4]. There are two key characteristics that make PLL widely used in communications systems and control applications. First, the bandwidth of PLL can be very small which means the signal-to-noise ratio can be improved in several orders of magnitude. Second, the PLL can automatically track signal frequency.

Due to the nonlinear nature of the phase detector, the PLL is basically a nonlinear system. For PLL to work in a wide range of conditions, the stability of PLL is very important. Conventional approaches to stability analysis included linear analysis, phase plane plots, rule of thumb, and simulation. All these methods have limited applications in low-order PLLs. If the order of PLL goes beyond three, it becomes almost impossible to analyze the stability of the system.

Lyapunov stability theory has been widely used in control systems, especially in nonlinear systems such as robotics, motors and spacecrafts. A Lyapunov function is essentially an energy function. The essential idea of Lyapunov approach is to find a Lyapunov function for the system and if it can be shown that the derivative of the Lyapunov function is negative, then the overall system will be stable. In this paper, we present an approach to stability analysis of second order PLLs since they have practical applications in many areas.

Although stability is most important in the successful applications of PLL, it only provides a range of parameter values which guarantee the stability. Trial-and-error and/or extensive simulations are still needed to fine tune the system parameters to achieve optimal performance. There are many performance objectives: bandwidth of PLL, signal-to-noise ratio, steady-state tracking error, rise time, overshoots, etc. Some of these objectives are conflicting. For example, in order to have a high signal-to-

noise ratio, we need a lower bandwidth. However, a lower bandwidth implies slow rise time and hence poor tracking performance. To satisfy all these objectives manually requires a lot of simulations and experience. We herein propose a general multi-objective optimization method which can satisfy all the performance objectives simultaneously even in the presence of nonlinearity such as phase detector in the system.

The paper is organized as follows. Section 2 will describe the Lyapunov approach to stability analysis of a second-order PLL. Section 3 will introduce the multi-objective optimization paradigm. A simulation example will be detailed in Section 4 to illustrate the performance of the proposed approach. Finally, some comments on future research directions will be included in Section 5.

2. Stability Analysis Using Lyapunov Function

2.1 Model of PLL

A general block diagram of an analog PLL is shown below.

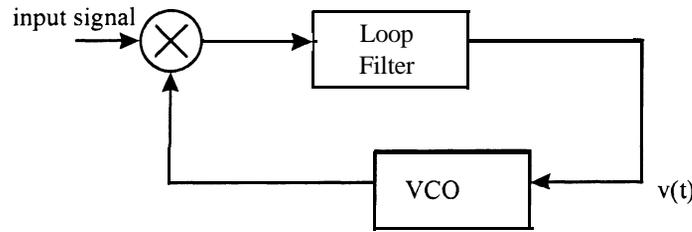


Fig. 1 An analog PLL.

Assume the input to the PLL is a sinusoid $\cos(\omega_i t + \phi)$ with ω_i the input frequency and ϕ the input phase and output of the VCO is $\sin(\omega_i t + \psi)$ with ψ the phase signal. A phase detector PD is just a multiplier which produces an output

$$\begin{aligned} e(t) &= \cos(\omega_i t + \phi) \sin(\omega_i t + \psi) \\ &= \frac{1}{2} \sin(\phi - \psi) + \frac{1}{2} \sin(2\omega_i t + \phi + \psi). \end{aligned} \quad (1)$$

The high-frequency component should be eliminated by using a low-pass filter. The cutoff frequency of the filter should be selected around ω_i . The reason is that we do not want to interfere with the function of the loop filter. The loop filter acts as controller to the PLL. Its function is to make sure the transient and steady-state response as good as possible. This filter is usually selected to have the form

$$G(s) = \frac{s + b}{s + a} \quad (2)$$

where a, b are design parameters which control the bandwidth of the loop. The output of the loop filter provides the control voltage $v(t)$ for the VCO. The VCO is a sinusoidal signal generator with an instantaneous phase given by

$$\omega_i t + \psi = \omega_i t + K \int_{-\infty}^t v(\tau) d\tau \quad (3)$$

with K a constant.

With a proper design of the low-pass filter to filter out the high-frequency component of the PD, we can therefore neglect the high-frequency component of the phase detector. Hence we can cast the PLL into a simpler and equivalent model shown in Fig. 2.

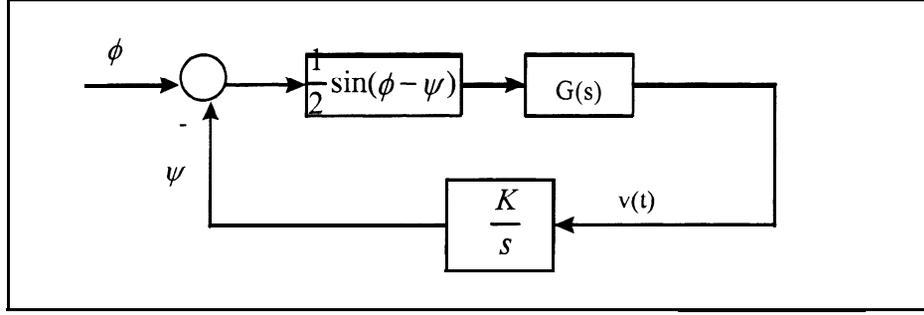


Fig. 2 Equivalent model of PLL.

2.2 Stability Analysis of PLL Using Lyapunov function

It should be emphasized that there is a nonlinearity in the system, i.e. the term $\sin(\phi - \psi)$. To analyze the stability, the conventional way is to linearize the nonlinear term by assuming $(\phi - \psi)$ is small. However, in many situation the phase error is not small due to either frequency error or phase error. To use Lyapunov approach, we first need to convert the system into a state-space model. Defining $\varepsilon = \phi - \psi$, we can write a model for the system in Fig. 2

$$\begin{aligned}\dot{x} &= -ax + 0.5(b-a)\sin \varepsilon, \\ \dot{\varepsilon} &= -Kx - 0.5K \sin \varepsilon.\end{aligned}\quad (4)$$

The state variable x denotes an internal state in the loop filter $G(s)$. Lyapunov approach is a method which can analyze the internal stability of the system, i.e. the stability of the states. If the states are stable, then the system will also be stable from the input-output point of view. The very first step of the analysis begins with the search of a Lyapunov function. This process depends heavily on one's experience. In [2], there are quite a number of examples on how to choose a proper Lyapunov function for a given system. For the PLL system, we choose the following Lyapunov function candidate,

$$V = (1 - \cos \varepsilon) + \frac{1}{2} z^T P z \quad (5)$$

where

$$z = \begin{bmatrix} x & \varepsilon \end{bmatrix}^T, \\ P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$$

is a positive definite matrix. V is a locally positive definite function (l.p.d.f.) since

- (i) $V(z)=0$ if $z = 0$, and
- (ii) $V(z) > 0$ if $z \neq 0$ for $z \in B$ where B is a ball around the equilibrium point $z = 0$.

There exists a well known theorem from nonlinear system analysis, which states the condition for system stability.

Theorem: The equilibrium point $z = 0$ is stable if there exists a continuously differentiable l.p.d.f. V such that

$$\dot{V}(z) \leq 0, \quad \forall z \in B. \quad (6)$$

Proof: See [2].

Thus our objective is to show that the derivative of V is negative for the PLL system. Differentiating Yin (5) gives

$$\begin{aligned}\dot{V} &= \sin \varepsilon + \frac{1}{2} \dot{z}^T P z + \frac{1}{2} z^T P \dot{z} \\ &= [0.5(b-a)(p_1 - 0.5Kp_2 - K)x \sin \varepsilon + [0.5(b-a)(p_1 - 0.5Kp_2)] \varepsilon \sin \varepsilon - \\ &\quad - (ap_1 + Kp_2)x^2 - (ap_2 + Kp_3)x \varepsilon\end{aligned}\quad (7)$$

Under the following conditions,

$$(1) K > 0, b > a \quad (8a)$$

$$(2) 0 > p_2 > -a/b \quad (8b)$$

$$(3) p_1 = \frac{K}{(b-a)}(p_2 + 1), p_3 = -ap_2 / K \quad (8c)$$

it can be seen that

$$\begin{aligned}\dot{V} &= -0.5K \sin^2 \varepsilon - (ap_1 + Kp_2)x^2 + [0.5(b-a)p_2 - 0.5Kp_3] \varepsilon \sin \varepsilon \\ &< 0\end{aligned}\quad (9)$$

since

$$(ap_1 + Kp_2) > 0 \quad (10)$$

and

$$[0.5(b-a)p_2 - 0.5Kp_3] < 0. \quad (11)$$

Hence the PLL system is locally asymptotically stable. To extend the case for other types of second-order systems is not difficult and can be achieved following the same route as before.

3. Multi-Objective Optimization Technique for Optimal Loop Performance

The basic idea of the approach is to formulate the PLL design as a constrained optimization problem. Although there exist some approaches to optimization of PLL using Wiener filters, they only tradeoff the bandwidth with the total transient error. The approach that we propose to use is called multi-objective optimization method. It is more powerful than other methods because it can consider all specifications, including time domain and frequency domain specifications. It can also take into account nonlinear effects of the system.

In PLL, the performance of tracking is judged based on many factors:

- (a) steady-state error,
- (b) transient response due to a step of phase,
- (c) transient response due to a step of frequency,
- (d) transient response due to a step of acceleration,
- (e) loop behavior in the presence of an angle modulated input signal.

Besides these, another important criterion to judge the performance is the signal-to-noise ratio. These performance criteria are conflicting to one another. For example, if we want to have a fast transient response, we need a high loop gain. However, a high loop gain implies a high noise bandwidth which will reduce the signal-to-noise ratio. Our method can handle time-domain graphical constraints as well as nonlinear inequality constraints. One example of graphical constraint is shown in Fig. 3 below.

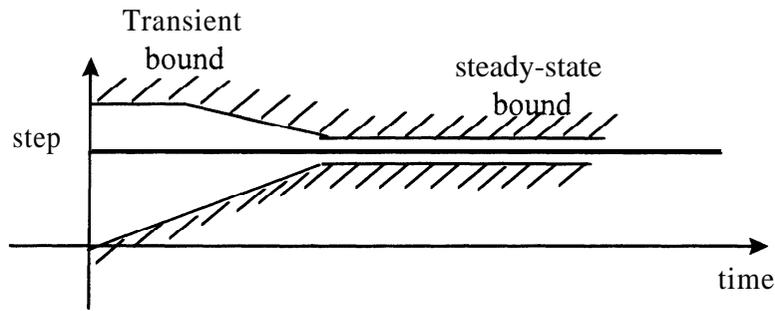


Fig. 3 Graphical performance constraints.

The transient performance is constrained by the transient bound shown in Fig. 3. In certain applications, percentage overshoot cannot be too large. The transient bound provides an upper limit for the overshoot. The steady-state bound is self-explanatory. Its purpose is to limit the steady-state tracking error.

The constraint can also be in the form of nonlinear inequality constraint. For example, if we do not want the noise bandwidth to exceed certain maximum values. We can write a constraint of the form [1]

$$B_L = \frac{1}{8} K \frac{0.5K + b}{0.5K + a} \leq 50 \text{ rad./s} \quad \text{for a second-order lag-lead loop filter} \quad (12)$$

Now all the stability and performance constraints can be translated into a set of performance objectives $\{J_i(p), i = 1, 2, \dots, m\}$, which are chosen such that if $J_i(p_1) < J_i(p_2)$ then the design parameter vector p_1 is better than the design parameter p_2 , as far as the objective of J_i is concerned. This set of objectives can be considered to be a vector of objective functions, or a multi-objective function:

$$J(p) = \begin{bmatrix} J_1(p) \\ \vdots \\ J_m(p) \end{bmatrix}$$

The performance specification is satisfied if $J(p) < c$ for some positive vector c of “thresholds”, i.e. if $J_i(p) < c_i$ for each i , and $c_i > 0$. To solve the above multiobjective function, there exists some procedures in the literature [5].

A typical design session proceeds as follows. First of all, we need to specify the constraints, the loop filter structure (active or lag-lead), the order of the loop filter, bandwidth of the PLL, etc. Then the simulation tool will simulate the performance through a few iterations. The software will compute values of the design parameters that improve the satisfaction of the constraints at each iteration. One can monitor the progress of the optimization by various graphical indicators and response plots.

4. Simulation Studies

We consider a second-order loop with a lag-lead compensator. The loop bandwidth should be less than 15 rad./s. The stability constraints are listed in Equations (8), (10), and (11). The settling time should be 0.1 second with respect to a step change of frequency of 50 rad./s. The initial parameters are

$$a = 1, \quad b = 50, \quad K = 50$$

After the optimization process, the optimal parameter values are:

$$a = 59.19, \quad b = 126.55, \quad K = 50.$$

In the optimization, we did not change the value of K since K is directly related to the bandwidth of the PLL.

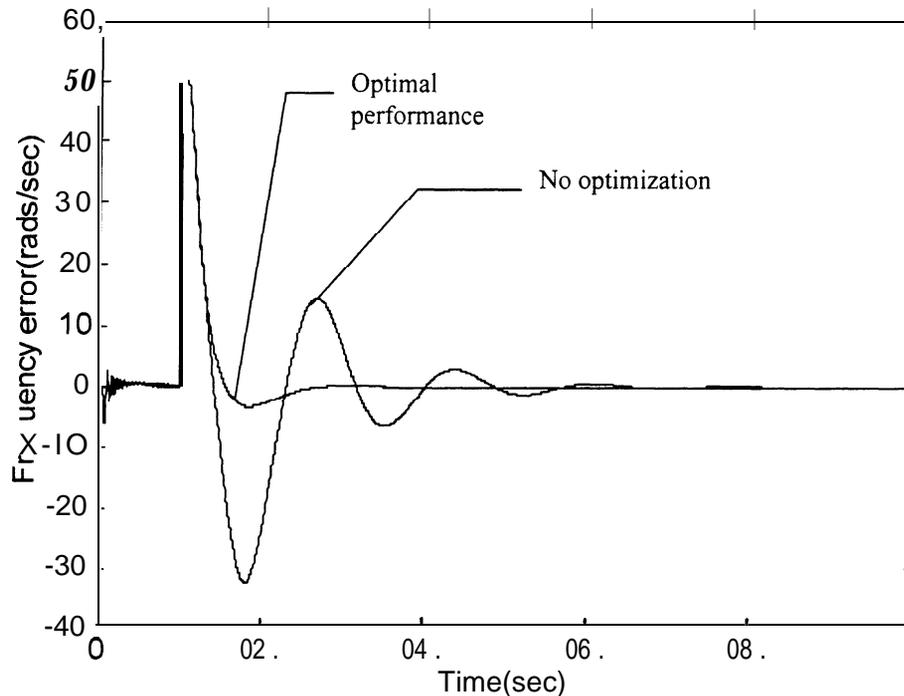


Fig. 4 Performance comparison of optimized and un-optimized PLL.

The performance comparison is shown in Fig. 4 which shows the un-optimized performance with the optimized performance. Even with the adjustment of two parameters, the performance can make quite a big difference.

5. Conclusions

A two-step procedure to the optimization of PLL with guaranteed stability is presented in this paper. The first step is to guarantee the loop stability by using the Lyapunov approach. In the process, parameter ranges (within the range the loop stability is guaranteed) can be generated. The second step consists of a multi-objective optimization method that can choose a set of parameters within the previously mentioned parameter ranges to trade-off the loop bandwidth, tracking performance, and steady-state errors in an optimal fashion. Simulation results show that the proposed method indeed improves the performance of loop quite significantly.

References

- [1] F. M. Gardner, *Phaselock Techniques*, John Wiley and Sons, New York, 1979.
- [2] M. Vidyasagar, *Nonlinear Systems Analysis*, Prentice-Hall, 1979.
- [3] U. L. Rohde, *Digital PLL Frequency Synthesizers: Theory and Design*, Prentice-Hall, 1983.
- [4] A. Blanchard, *Phase-Locked Loops*, John Wiley and Sons, New York, 1976.
- [5] J.M. Maciejowski, *Multivariable Feedback Design*, Addison Wesley, 1989.

Baseband Group Delay Equalization of IF Filters for Data Communications

T.C. McDermott, N5EG
Texas Packet Radio Society
265 Daniel Dr.
Piano, Texas 75094
n5eg@tapr.org

Abstract

One source of phase (and thus group delay) variation is the IF filter in the receiver. This is either a mechanical filter or more commonly, a crystal filter. These IF filters typically have been optimized for best amplitude rejection of out-of-band signals vs. achievable passband flatness for a given number of filter poles. This typically leads to a Chebychev filter implementation which can have significant group delay variation. If the group delay variation vs. frequency is significant compared to the symbol rate (in baud) then the bit-error-rate of the received signal may be substantially degraded. Techniques to equalize the filter at IF are possible, but may prove difficult to implement and adjust, however they work well independent of the modulation type. On the other hand equalization of the filter delay at baseband is possible, but is only exact for linear modulation types (PSK, QAM). FSK modulation (which is non-linear) cannot be exactly compensated at baseband but if the deviation index is low, then a reasonably good job of delay equalization at baseband can be accomplished. This article examines some amplitude, phase, and delay properties of first-order, second-order and all-pass filters, and illustrates examples of Chebychev and Butterworth IF filters. Finally, a graphical representation of delay equalization at baseband is shown.

Introduction

It can be shown that there exists several optimal channel amplitude responses for data communications channels, one of the more popular being the raised-cosine channel response. It can also be shown that the optimum bit-error-rate (BER) vs. received power level occurs when the channel phase is linear (group delay is flat)¹. The optimum amplitude response vs. frequency has been discussed in a previous paper², this paper will examine the flat group delay requirement, and equalization of IF filter group delay variation.

Filter Transfer functions

All linear filtering functions can be described as a transfer function relating the filter's output amplitude and phase with respect to the filter's input amplitude and phase. For circuits using reactive components (inductors, capacitors), the transfer function depends on frequency, and includes terms involving complex numbers. The transfer function is usually abbreviated $H(s)$,

where H is a function of the variable, s. In performing analysis of the filter to a sine wave input, we substitute for s:

$$s = j\omega = j2\pi f \tag{1}$$

where j is equal to the square-root of negative 1. That is, j is the imaginary operator, f is the frequency in hertz, and omega is the frequency in radians/second. In order to describe the properties of a filter, it is necessary to derive the transfer function and then analyze the transfer vs. frequency. To describe the transfer function of a network, it is necessary to describe the complex impedance of each of the components, and then describe the output to input transfer in those terms. The complex impedance of an inductor is given by:

$$Z(s) = j2\pi fL = sL \tag{2}$$

This shows that the impedance of an inductor increases with frequency, and that there is a +90 degree phase shift associated with the inductor (the +j term). Similarly, the impedance of a capacitor is given by:

$$Z(s) = \frac{1}{j2\pi fC} = \frac{1}{sC} \tag{3}$$

which shows that the impedance of a capacitor decreases with frequency, and there is a -90 degree phase shift associated with the capacitor (the 1/j term, since 1/j = -j). The impedance of a resistor is just R of course, independent of frequency and with no associated phase shift.

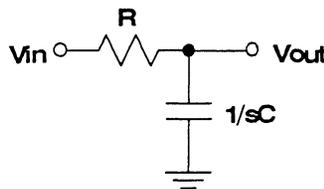


Figure 1 - First-order low-pass filter

To derive the transfer function of a first-order low-pass filter (figure 1), we can write the voltage divider equation:

$$H(s) = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} = \frac{1}{RsC + 1} \tag{4}$$

If we assume that R= 1 ohm, and C=1 farad, then the time constant is one second, and RC = 1. Then we can *normalize* the response to 1 second, which is a filter -3 dB. cutoff of 1 radian/second. This can be expressed as:

$$H(s) = \frac{1}{s+1} \quad (5)$$

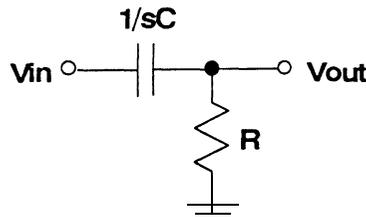


Figure 2 - First-order high-pass filter

Similarly, we can derive the transfer function for a first-order high-pass filter response (figure 2):

$$H(s) = \frac{R}{R + \frac{1}{sC}} = \frac{RsC}{RsC + 1} \quad (6)$$

When normalized to $R=1$ ohm, and $C=1$ farad, this is simplified to:

$$H(s) = \frac{s}{s+1} \quad (7)$$

An interesting active circuit is the all-pass filter (figure 3). We can derive the transfer function for this circuit in a manner similar to above, and when normalized to $R=1$ ohm and $C=1$ farad, the transfer becomes:

$$H(s) = \frac{s-1}{s+1} \quad (8)$$

It can be seen that the amplitude does not change with frequency, s , but the phase response does. This circuit can thus be used to compensate phase delay variation (and thus group delay variation) without affecting the amplitude response.

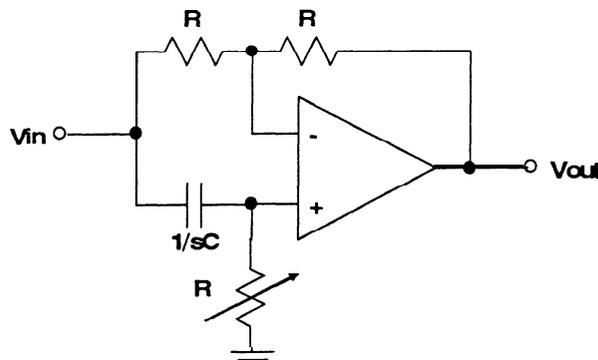


Figure 3 - First-order all-pass filter

The group delay of a transfer function is related to the phase slope, steeper phase change means greater group delay. Assuming ϕ is the phase in radians and ω is the frequency in radians/second, then group delay, in seconds, is:

$$\text{group delay} = - \frac{d\phi}{d\omega} \tag{9}$$

If the units are degrees and hertz, then the group delay in seconds is (degrees per hertz)/360. If the phase is linear (changes at the same rate regardless of frequency) then the group delay is constant vs. frequency. To analyze and plot these three transfer functions, an Excel™ spreadsheet was set up to evaluate the amplitude, phase, and group delay of each. Since Excel can directly manipulate complex numbers, this is relatively easy to do.

Figure 4 illustrates the amplitude and phase response of the low-pass filter, figure 5 illustrates the amplitude and phase of the high-pass filter. Figure 6 illustrates the phase and group delay of the low-pass and high pass filters. The two filters have the same phase shape, even though there is a difference in the absolute phase. This absolute phase difference has no effect on the group delay, which is identical for both filters.

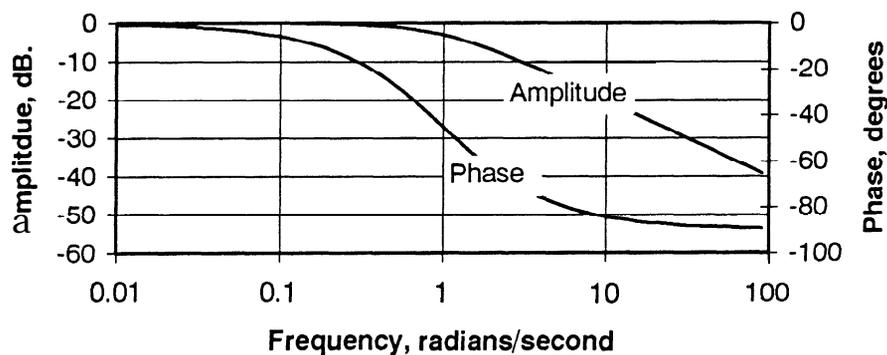


Figure 4 - Amplitude and phase of a low-pass filter

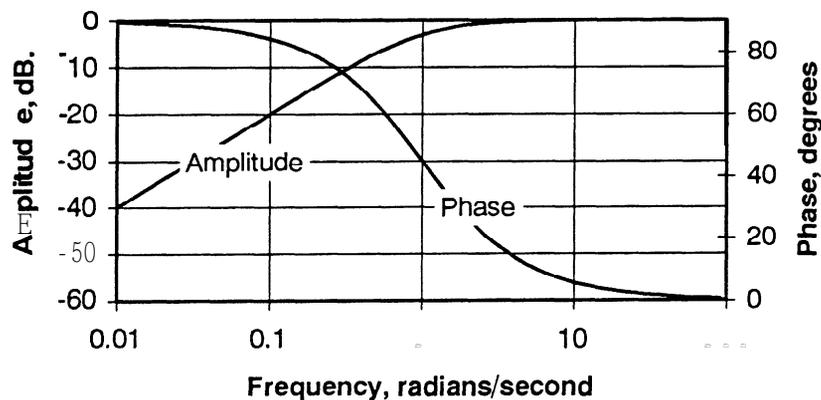


Figure 5 - Amplitude and phase of a high-pass filter

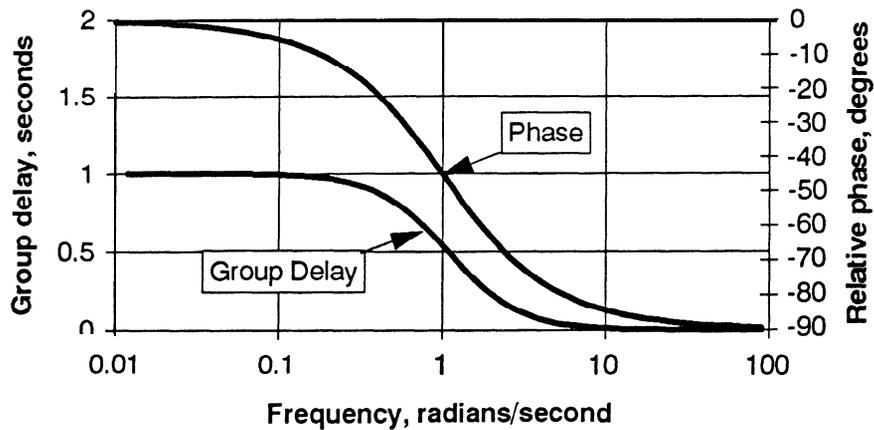


Figure 6 - Phase and Group delay of low-pass and high-pass filters (they have exactly the same group delay)

The all-pass filter has flat amplitude response, which is not shown. The phase response and group delay are shown in figure 7.

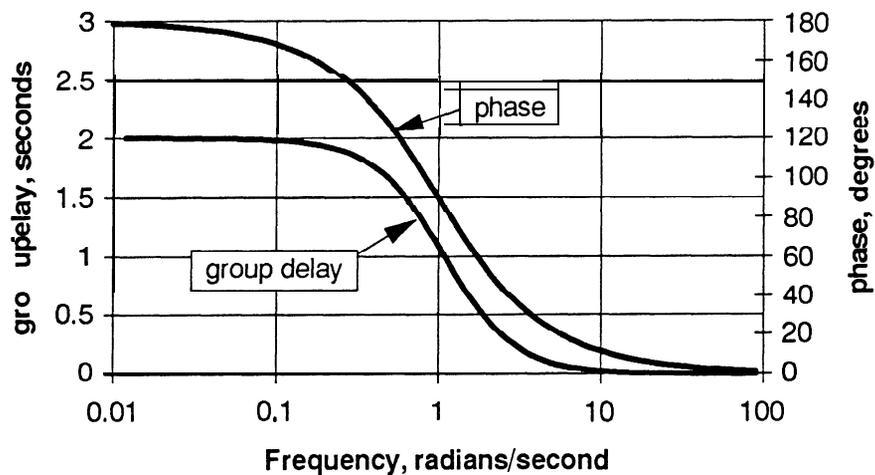


Figure 7 - Phase response and group delay of the all-pass filter, with a break frequency of 1 radian/second.

As the frequency response of the all-pass is adjusted, the phase and delay responses change. Figure 8 shows the phase and delay response of a the all-pass filter when the variable resistor of figure 3 is adjusted for a break frequency of 3 radians/second. Note that the absolute delay at low frequency is only one-third as large as in figure 7.

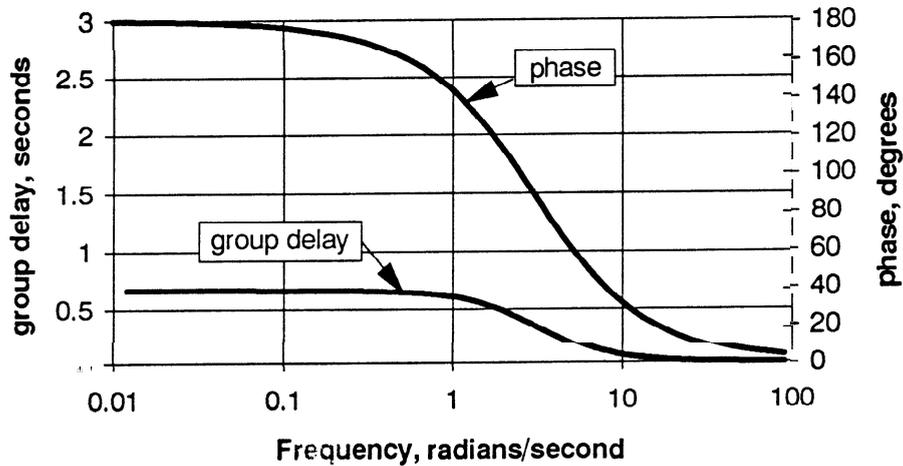


Figure 8 - Phase and Group Delay response of all-pass filter, with a break frequency of 3 radians/second

Higher Order Filters

Higher order filters, such as band-pass filters can be analyzed just like the low-, high-, and all-pass filters. A simple bandpass filter composed of a series LC and a shunt R yields the amplitude, phase, and group delay response as shown in figure 9. This filter has a resonant frequency of 1 radian/second, and a damping of 0.707 (a Q-factor of 1.414). Q is the inverse of damping.

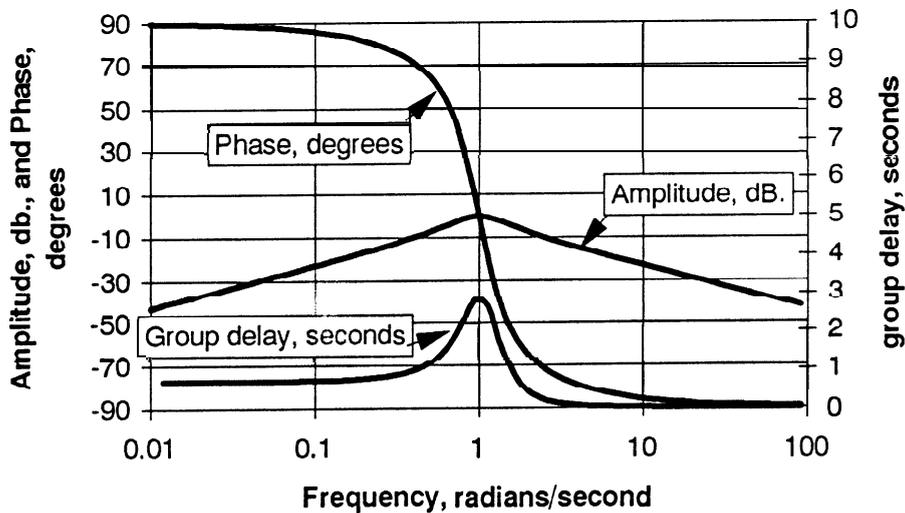


Figure 9 - Amplitude, Phase, and Group Delay response of a bandpass filter

Here it can be seen that the group delay has a peak at the center frequency of the filter. As the damping of the filter is changed, the group delay also changes, with low damping (high-Q) filters having more group delay at the peak. When constructing more complex bandpass

filters, the delay will peak at the corners of the filter. The sharper the corners, the greater will be the group delay, generally. To illustrate with two examples, the frequency and group delay response of a 6-pole Butterworth, a 6-pole Chebychev, and an 8-pole Chebychev filter will be analyzed. These filters have been moved up to around 12 radians/second center frequency.

Analysis of Butterworth and Chebychev Bandpass Filters

The transfer function, $H(s)$ for these filters can be constructed and analyzed just as has been done previously. The transfer function of the 6-pole filters is described by equation 10:

$$H(s) = \frac{-(s - z_1)(s - z_2)(s - z_3)}{(s - p_1)(s - p_2)(s - p_3)} \quad (10)$$

Where p_1, p_2, p_3 are the *poles* of the filter response, and z_1, z_2, z_3 are the *zeros* of the filter response. The zeros determine where the filter response goes to zero (no output), and the poles determine where the filter response peaks (maximum output). For the Butterworth and Chebychev filters, all three zeros are at zero hertz, so equation 10 can be simplified to equation 11:

$$H(s) = \frac{-s^3}{(s - p_1)(s - p_2)(s - p_3)} \quad (11)$$

The Butterworth filter has a passband response that is maximally-flat, and the Chebychev Type-I filter has an equiripple passband response (the Chebychev Type-II has an equiripple stopband response). The difference between these two filter types is the placement of the poles. In a Butterworth filter, the poles are placed in a circle on the s-plane, and in a Chebychev filter, they are placed on an ellipse. The two axis of the s-plane correspond to the real and the imaginary parts of the pole. Since a pole position may in fact be complex, then the filter pole is actually a complex pole-pair (one pole at positive frequency, the other pole at negative frequency). The filter described by equation 11 is a 3-order filter, which has 3 complex pole-pairs, and is referred to as a 6-pole filter. Thus the 6-pole Butterworth bandpass filter consists on two sets of 3 pole pairs on two circles, one with it's center at the filter center frequency, the other with it's center at the negative of the filter center frequency. Each of the 3 poles lies on the half of the circle that lies in the left-hand plane. Figure 10 is a pole-zero diagram of the Butterworth filter. The 3 zeros are piled up on top of each other at the origin.

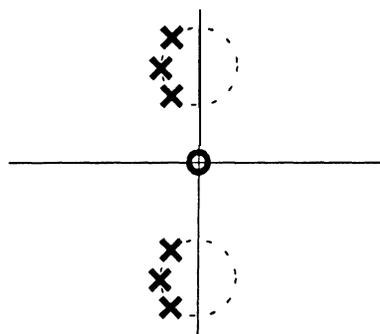


Figure 10 - pole / zero diagram of the 6-pole Butterworth filter.

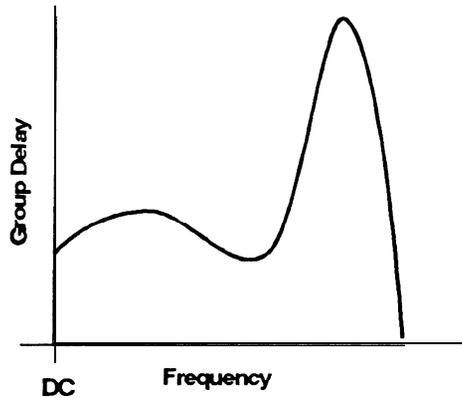


Figure 14 - Baseband delay of Chebychev filter

Figure 15a shows the delay of the all-pass filter chain and the Chebychev filter on the same graph, and figure 15b shows the total delay of the all-pass and Chebychev in series.

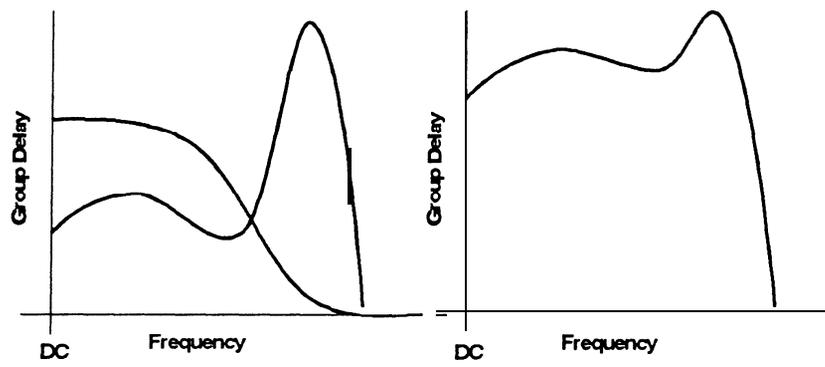


Figure 15 - A delay of all-pass filter and Chebychev filter on the same graph
B total (sum) of the delay for the equalizer and filter.

It can be seen from figure 15b that the total delay variation *in the region of interest* is less than in figure 14. The region of interest includes the passband of the filter, but not the stopband of the filter. Since the received signal that is outside the passband is rejected by the filter, it's group delay is not important.

Required Filter Bandwidth

The required IF filter width, when properly group delay equalized needs to be wide enough to pass the received data signal. The baseband width of a data signal can range from one-half the baud rate up to the baud rate depending on the type of filtering performed. A data signal limited to one-half the baud rate (Le.: 4800 Hz. for a 9600 baud signal) requires a great deal of precision in the filters, and much control of all the filters, and is not usually practical. On the other hand when the bandwidth is equal to the baud rate (i.e.: 9600 Hz. for a 9600 baud signal) many properties of the signal are easy to control. The IF filter width needs to be twice the baseband width, so the required width would be 9600 Hz to 19200 Hz for a 9600 baud signal (for linear modulation, QAM and PSK), with 19200 Hz resulting in easier to realize

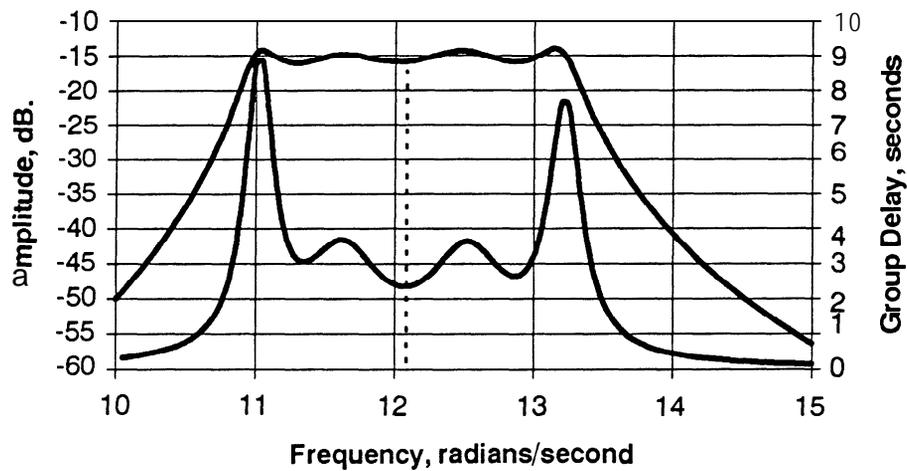


Figure 13 - 8-pole Chebychev amplitude and group delay response.

Delay equalization

It can be seen from the three filters that the group delay is largest at the 'corner' of the filter. The vertical dashed line represents the center frequency of the filter. Since each of these filters is at the Intermediate Frequency, or IF of the radio, the center frequency of the filter will be demodulated down to zero hertz at baseband by the last mixer (or by the discriminator in an FM detector). Frequencies to the left of the center frequency are negative frequencies at baseband, those to the right are positive frequencies. At baseband, it is impossible to distinguish the positive from the negative frequencies, and they all look like positive frequencies. Thus the IF response, *when translated to baseband* appears as that part of the response that lies at the filter center frequency and towards the right of that point.

Thus the task of group delay equalization is to impart significant delay at low frequencies, and less delay at higher frequencies, hopefully the inverse of the filter's baseband delay characteristic. Referring back to figures 7 and 8, it can be seen that the all-pass filter has exactly this needed delay characteristic. Thus using one of these all-pass filters at baseband in the receiver should allow us to cancel some of the filter delay. It may require several cascaded all-pass filters to achieve the 'necessary delay, and some of the all-pass filters may have to be adjusted to different break frequencies in order to properly compensate for the IF filter's baseband delay characteristic.

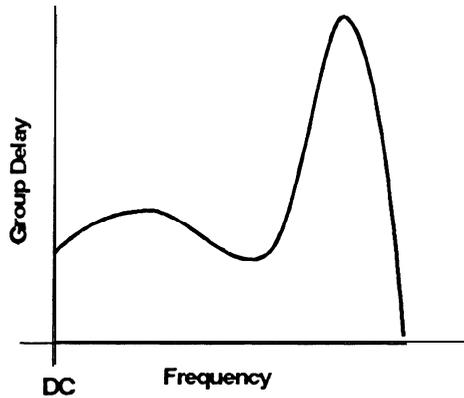


Figure 14 - Baseband delay of Chebychev filter

Figure 15a shows the delay of the all-pass filter chain and the Chebychev filter on the same graph, and figure 15b shows the total delay of the all-pass and Chebychev in series.

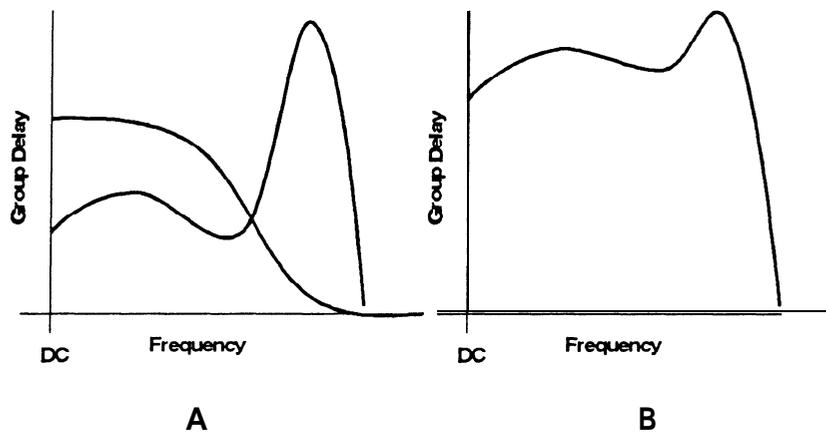


Figure 15 - A) delay of all-pass filter and Chebychev filter on the same graph
 B) total (sum) of the delay for the equalizer and filter.

It can be seen from figure 15b that the total delay variation *in the region of interest* is less than in figure 14. The region of interest includes the passband of the filter, but not the stopband of the filter. Since the received signal that is outside the passband is rejected by the filter, it's group delay is not important.

Required Filter Bandwidth

The required IF filter width, when properly group delay equalized needs to be wide enough to pass the received data signal. The baseband width of a data signal can range from one-half the baud rate up to the baud rate depending on the type of filtering performed. A data signal limited to one-half the baud rate (i.e.: 4800 Hz. for a 9600 baud signal) requires a great deal of precision in the filters, and much control of all the filters, and is not usually practical. On the other hand when the bandwidth is equal to the baud rate (i.e.: 9600 Hz. for a 9600 baud signal) many properties of the signal are easy to control. The IF filter width needs to be twice the baseband width, so the required width would be 9600 Hz to 19200 Hz for a 9600 baud signal (for linear modulation, QAM and PSK), with 19200 Hz resulting in easier to realize

modems. If FSK (non-linear modulation) of sufficiently narrow deviation is utilized, the filter width needs to also include the deviation of the FSK in addition to the data spectrum.

Required Group Delay Flatness

The group delay should be flat, within about 20% of a bit duration over the frequency range of interest. Thus, for a 9600 baud signal, where the bit interval is **104** microseconds, the group delay should be flat to within 20 microseconds for best performance, from DC up to **9600** Hz. in the baseband.

Excel is a trademark of Microsoft, Inc.

References

¹ Proakis, John G., "Digital Communications", 3rd edition, McGraw-Hill, 1995

² McDermott, T.C., "Modem Channel Response", Proceedings of the TAPR 1995 annual meeting.

Easy to Follow Packet

James C. Nobis
2407 Nettle Ln., Euless, TX 76039
(8 17) 540-1277 jnobis@cs.gte.net

Abstract

This paper is a general introduction to packet radio. It explains how to setup your own packet station. Then, it explains connecting to stations and how to use Rose Switch, Texnet, and Nodes. Key Words: Packet, Rose, Texnet, and Nodes.

Introduction

When you hear someone on the radio talking about a topic unfamiliar to yourself, you immediately want to learn all about it. Have you ever heard someone talking about packet, rose, texnet, or nodes? What equipment do you need? What is packet, rose, texnet, and a node?

Setup

First, you have to get the proper equipment in order to communicate using packet. A typical packet station consists of a PC (dummy terminal), a two meter radio, and a tnc, a packet modem. If you don't already have a personal computer, an older model such as a 286 or 386 computer will run packet just fine. A 286 computer costs about \$20-\$100 (used) and a 386 computer costs about \$200-\$500 (used). You won't find these types of computers brand new. A brand new computer (a pentium) costs \$1300-\$5000. Because of this high price, you can see why it's better to get an older computer. If you want speed and you have extra money, a 486 would be even better than a 286 or 386 (\$1000 for a 486, 8 MB of Ram, 4x CD-ROM, speakers, 500 Mb hard drive, etc.) Basically any two meter radio will work. Even a handy talky works all right. You will also need a packet modem. A good beginner packet modem is the Kantronics KPC-3. Tuckers will sell one to you for \$117 plus tax (if you live in Texas or buy it at the store) plus shipping. Investigate Kantronics web page if you have access to the web (<http://www.kantronics.com>.) After you have the major components, there are: connectors (computer, radio, power, etc.), soldering irons, wire, etc. After you have the equipment connected, it's time to test it.

Connecting to Stations

Make sure everything is plugged in. You can't transmit well using a radio's antenna jack **Hi Hi**. Go ahead and start the computer program for the tnc; turn on the power, etc. If you haven't already, it's probably a good time to read the beginning of the TNC'S manual (setting up.) While you do this, check for a part that explains how to tell the tnc what your **callsign** is. Once you have your **callsign** set into memory, your call will be sent with each packet. Now that you have a general knowledge of the tnc you can precede into the fun of packet radio! Now you can connect to.....hmm if you don't have someone there awaiting your connect packet; then you can't connect...but wait there are **PBBS's** (Packet

Bulletin Boards.) Usually your first connect will be to a PBBS station. Some good stations to connect to are:

KF5MG (144.69) & (144.67)
K9MK (144.91) DX Cluster
W5AH (144.97)
N5TTU (145.01)
WG5E (145.01)
'GOKA (145.01) Node
WR5C-4 (145.05) Texnet
N5VGC-9 (145.05) Texnet
W05H (145.07)
CLEKA (145.07) Node
DFWKA (145.07) Node
N5AUX (145.09)
KC5COF-1 (145.09)

These are just a few of the packet stations in the area. Most of these stations on 145.01 can be reached using either Dallas Rose Switch numbers or using 8 1749 1. (They were chosen because they are easy to use and are great stations.) However these are in the Dallas area. When you connect to a few of these stations you could actually get the control operator. With my Baycom packet modem, I connect by typing "Tab" c "callsign." With the KPC-3 you edit a connect to list and then you can click on the station. Most of these stations can be used as your home BBS. Once you designate one station as your home BBS all your personal messages will be sent to this station. Make sure the station is dependable and doesn't take a rocket scientist to operate. The first time you connect to a station they may ask you for your home BBS so type in the callsign of what you want for your home BBS. However, in order to use any station as your home BBS you must be able to connect to that station to receive your messages. Every time a packet is distorted in route it will ask the other station for a retry (a retransmission of the previously sent data.) Now that you are connected to a station you have many options. To get a list of these commands and what they do you must issue a help command (Typically you type "help" or "?" and push the enter key to send the command.) Here are some basic commands:

L list all messages
LL list unlisted messages
R # read message #
KM erase all your personal messages
S send a message
B disconnect from the station

As with the frequencies of packet stations the commands are also numerous in many BBS and commands other than the basic commands (such as the ones I listed) usually vary depending on the system. When you are finished playing.. .umm.. I mean learning you can issue the command b or bye to disconnect. That is basically all there is to it.

'Special thanks to Barry/N5TTU for helping me when I didn't understand something and for the listing of Rose Switches.

Rose

Rose Switch (shortened to Rose) is a network of packet stations acting similar to repeaters. There are a few differences: they transmit and receive on the same frequency; there isn't a PL tone, and they are packet stations not typical repeaters transmitting data. There are Rose Switches in eight states: Texas, Oklahoma, Louisiana, Arkansas, Florida, Minnesota, Illinois, and New Mexico. Rose is a good way to extend the range of your station. A neat feature of this is that in Texas every Rose Switch is on 145.01. You can use this to talk to your friend in Austin (or anywhere in the eight states) on packet **from** here with just a handy talky and a rubber duck. Rose is a little more complicated than a typical connect message. Example: **C KC5NAS V WB5CQU-9, 8 17491** This says: connect to **kc5nas** through the switch of 8 17491 **from** the station close to me of **WB5CQU-9**. Broken down a little more it is similar to a regular connect until the v (via), the **callsign** is the **callsign** used by the rose switch near you (**WB5CQU-9** is just an example) and then the rose switch number near your buddy. The computer then looks up where the signal should go (by using the 6 digit number) and it then uses the appropriate antenna, radio, etc. After you get a connect, it will say call being setup and then it will acknowledge that you got a connect or it will give you a message saying why it couldn't connect. Although going through another station should typically slow down your messages, it does not. It can speed up your packets because there are less rejects than when your message goes directly instead of using a relay. Rose uses either a telephone line or a 9600 baud packet station on 70 cm. Usually, it uses 70 cm but occasionally it will use a telephone line. Here are the Rose Switches in the US:

(Texas)

Carthage	903693	KA5HSA-2	
Cleveland	713592	KB5NX-2	
College Station	409845	W5AC-2	
Dallas North	214234	N5BCA-2	
Dallas Texnet	214050	N5BCA-6	connect to WR5C-4 for texnet
Dallas/Ft. Worth	214223	N5BCA-4	
Double Mtn.	915735	AB5GE-5	
Double Oak	817491	WB5CQU-9	
Double Oak	817105	WB5CQU-15	
Dublin	817445	WB5MAT-2	
Elmo	903560	WA5YTD-12	
Gainesville	817668	WB5CQU-4	
Kilgore	903984	W5KPZ-1	
Littlefield	806299	KA5AJA-3	
Lubbock	806745	WA5TBB-3	
Olney	817564	N5OLP-5	
Seymor	817888	N5ENS-5	
Sherman	903429	K5KQG-4	
Temple	817778	W5JEH-5	
Tyler	903561	N5YPB-2	
Waco	817863	N5BCA-1	
Westcamp	806272	K5RKL-3	
Wichita Falls	817691	W5VAC-5	

(Oklahoma)		
Ardmore	405226	WA5YOM-5
El Reno	405262	AC5C-5
El Reno 145.07	405263	AC5C-6
Lawton	405248	WJ5Y-5
Midwest City	405732	K5JB-5
Noble	405872	WD5IAT-5
Pauls Valley	405238	WB5CQU-5
Velma	405444	KB5OJR

(Louisiana)		
Bastrop	318281	KE5L-3
Monroe	318323	N5PXW-3
Ruston	318257	W5HGT-2
Shreveport	318996	N5JH-2

(Arkansas)		
Camden	501836	

(Florida)		
Brooksville	904799	WA3YMV-8
Clear-water	813442	KC5FF-7
Iverness	904233	K4GBB-3
Sarasota	813371	WA3LKW-7
St. Leo	904567	AA4RV-7
Tampa	813040	KOZYF-4
Tampa	813878	KOZYF-9

(Minnesota)		
Minneapolis	612341	WAOCQG-1

(Illinois)		
Chicago	3 12245	KA9ODA-3

(New Mexico)		
Elida	505274	KA5BAT-3
San Jon	505357	W5CXP-3

Now just because Texas uses 145.0 1, doesn't mean all these other states will. Most of Louisiana, however, uses the same frequency. This should not cause a problem unless you want to use rose in another state. If you don't know the frequency of your area, just ask someone on a repeater and if they don't know the answer they can probably refer you to someone who knows the answer. Again, you must know what station to connect. If the PBBS is on 145.01 or on the appropriate frequency in another state, you can use Rose to act like a better antenna and radio. Now you are **curious...why** use a 6 digit number and why those numbers. That is a good question. You will, however, notice that the first three numbers are typically the area code (for a telephone) of where the switch is. Such as 2 14

being in the 214 telephone area code. A good way to find a station is to connect to heard. For example c heard v local switch call, local switch number. Also, IPCQU is a good place for heard stations and for information about rose. c IPCQU v local call, 8 17 105.

Texnet

Texnet is something similar to Rose Switch. However to get into texnet you use a typical connect message. In the Dallas area turn your radio to 145.05 and connect to **WR5C-4**. Once you are in, the texnet part starts. There are different names for places such as NDALLAS for north Dallas. To connect to **kf5mg** you can type c ipgate v iplink @ Denton “enter” and when **wr5c-4** receives that message, it will attempt to connect. If you want to connect to a node or PBBS somewhere else just type c “callsign” v callsign @ and the location near that place. Only use v and a callsign if you want to use a relay station. After you are connected to texnet, you can go to any of the locations that exists unless that particular link is broken. You can only connect to the station if they are on the proper texnet frequency such as for Rose. Texnet is usually easier to use, but you can be connected link after link sometimes. An example of this is typing c **gvl** @ greenvl and then typing c 7 KSDDBS. This will take you to a station around Greenville, but because you have 2 digipeaters the process is slowed down. You transmit to **WR5C-4**; he transmits to **gvl** and **gvl** transmits to **KSDDBS**. Also **after** you connect to **gvl**, you can type c 2 KS I G to take you to a DX Cluster (a place where you can receive information on DX stations that are currently on the air, the frequency, call, and time, and also get sunspot information.) As soon as you type bye or b, you will be disconnected from all digipeaters and returned to **WR5C-4**. The nice thing is that you get back to the **WR5C-4** so that you don’t have to reconnect to him (**Everytime** you use rose you have to first connect to local switch. Then, it can connect elsewhere. Once you are in Texnet you are there until you say good- bye at the texnet screen. As you can see, Texnet is a little easier to use than rose switch.

Nodes

A node is almost identical to Texnet. There are few exceptions. When you type c “callsign” you don’t have to type a location (NDALLAS.) This is because Nodes are stand alone; they aren’t really built to link between each other. You can however, connect to your local node and link to another node and so on. Of course, this means that you can only connect to another station in the nodes particular coverage area. Also when you disconnect from the linked station, you are completely disconnected from everything. You will also be disconnected from the main node unless issue a special connect command when you connect to the other station. Nodes are also handy because there is not a specific frequency. They are on many frequencies. Although you can’t link around different frequencies, there is a node or two on many frequencies. Many nodes are on 145.01 and 145.07. When you connect into different packet stations, you can find lists of frequencies for stations. Nodes like Rose and Texnet are designed to help extend your stations coverage area. Unlike Rose and Texnet which are multiple stations setup as links, Nodes are stand-alone stations.

Conclusion

As you can see there is much more to packet than just telling your station to connect directly to one station. In fact if you were limited to connecting directly, you would not have too many stations to connect to unless you had great antennas and massive power. With all these “extras” you are able to

have a small station with little power and a small antenna. This allows you to connect to many stations throughout greater distances. Like repeaters are good for asking questions and getting help packet is great for getting help with your computer. The next time someone says I can connect to you via Rose you can say, "I'll connect to you via Rose if you tell me your local switches number."

References

There are no references; most of this knowledge was gained from experimenting.

Object-Oriented Modeling of a Satellite Tracking Software*

M. Normandeau and M. Barbeau (ve2bpm)
Department of mathematics and computer science
University of Sherbrooke
Sherbrooke, Québec CANADA J1K 2R1
{normand,barbeau}@dmi.usherb.ca

Abstract

This paper presents a case study of an object-oriented development of a satellite tracking software. It is designed following the Real-Time Object-Oriented Modeling (ROOM) methodology. The design resulting from the application of ROOM is implemented in C++ on the QNX platform. Concurrent actors are naturally mapped to parallel processes. ROOM yields a modular architecture which is clear, reusable, and maintainable. Use of QNX leads to a highly performant and reliable system. This work is important because it shows application of advanced software engineering principles in a field where most of the development is still based on structured (and non-structured) techniques.

Key words : Object-Oriented Modeling, Real-Time Systems, Satellite Tracking.

1 Introduction

The problem addressed in this paper is the development of a software for ground tracking in real-time of non geostationary satellites in circular or elliptical orbits [2]. Tracking is necessary for communicating through the satellites. Tracking of a satellite means determining its position in space, when it is accessible (in *range*), and where the antennas should be pointed. Prediction of access time is required because, for a given ground station, satellites are in range only during certain periods (*satellite passes*) during a day. Antennas are directional and determination of pointing direction is necessary and expressed in terms of two parameters: azimuth and elevation. Software inputs are the orbital element sets of satellites being tracked, Greenwich Mean Sidereal Time for January 00.002 of each year, and latitude and longitude of a ground station.

This paper presents a case study consisting of the application of an object-oriented methodology to the development of a satellite tracking software. This experiment is not limited to a simulation but is a complete development from analysis to implementation in an actual tracking station. Development of the software follows the Real-time Object-Oriented Methodology (ROOM) [4]. ROOM is based on an operational actor model. Actors are active entities of a model. Each

*The research described in this paper was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR).

actor consist of data, behavior, and structure. Its execution is concurrent with other actors and it communicates with them by means of messages. The actor model supports the basic elements of the object-oriented paradigm such as abstraction, **encapsulation**, inheritance, aggregation, typing, and concurrency [1]. Employment of such a **methodology** leads to high quality designs which are very well documented, understandable, and easy to communicate as well as a evolve.

This paper is structured as follows. Section 2 presents the object-oriented model. The implementation of this model is discussed in Section 3. We conclude with Section 4.

2 Object-Oriented Model

This section details the object-oriented ROOM model devised for the satellite tracking problem. As the ROOM methodology dictates, our satellite tracking software model is made of actors. These actors represent the main active and concurrent components of the system. For each of them, a structure and a behavior are specified. They communicate among themselves and with the outside world through ports which are references to specific communication protocols. These protocols are sets of messages that actors intend to exchange with one another. The structure presents the components which make an actor and how they are related to one another. Components can be either actors, ports or bindings which are links between pairs of ports. Diagrams may be used to illustrate the structure and they represent actors as white boxes, ports as squares on the boundaries of actors, and bindings as lines between pairs of ports (see Fig. 1). On the figure, only the names of the actors and ports are displayed.

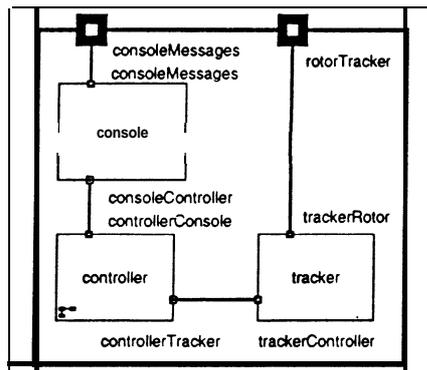


Figure 1: Tracking system structure

A behavior specification is made of states and events. An actor moves from state to state when events are triggered. Diagrams may illustrate the behavior of actors and they represent states as rounded boxes and events as arrows (see Fig. 2). The black circle at the top left corner of the diagram leads to the first state when the system starts.

Our model is made of three main actors : a *console*, a *controller*, and a *tracker* (see Fig. 1). The *console* is the interface between the user and system. It is responsible for offering possible actions to the user, getting his inputs, and sending them to the controller. The structure of *console* is fairly simple as it contains only two ports. The first one, *console User*, is used to communicate with the user. It refers to the *ConsoleMessages* protocol which specifies the possible messages the

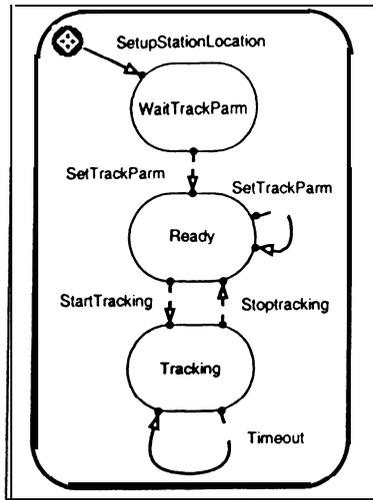


Figure 2: Controller behavior

user can send to *console* and vice versa. The actions available to the user are: setup a *tracking run*, start a *tracking run*, stop a *tracking run*, and *quit*. While setting up a tracking run, the user is also asked for a choice of a satellite. The set of actions offered to the user depends on the state of *console*. For example, if no tracking run is executing, it is impossible to stop one. The second port, *consoleController*, allows communication with the *controller* through the *ControllerMessages* protocol. It mainly relays user's decisions to the *controller*.

The second actor, *controller*, executes the user's decisions received from *console*. Once the system is tracking a satellite, it is also responsible for providing to *tracker* the direction in which the antenna should point. The *controller* contains component actors which are : *groundTracker* and *azElProvider* (see Fig. 3). Their own description will be provided further on. The *controller* receives messages from *console* that drives its behavior (see Fig. 2). Its main operations are to set the station's parameters, to initialize a tracking run, and to compute the azimuth and elevation of the antenna at specified intervals. The *tracker* has one port, *ControllerConsole*, referencing the *ControllerMessages* protocol allowing communication with *console*. It also has another port, *controllerTracker*, referencing the *TrackerMessages* protocol through which messages are sent to *tracker*.

The last actor at this level is *tracker* which is a driver to the rotor's interface card. It receives the azimuth and elevation from *controller* through its *trackerController* port. It then sends it to the rotor's interface in a way it can comprehend it using its *trackerRotor* port. These ports are the only components of the *tracker* actor. The Rotor protocol referenced by the *trackerRotor* port represents the driver specification of the rotor interface.

Explosion of *controller* uncovers two sub-actors (see Fig. 3): *groundTracker* and *azElProvider*. These embedded actors communicate through ports *controllerGround Tracker* and *controllerAzElProvider*. The *groundTracker*, given orbital elements and current GMT, is responsible for providing the Sub-Satellite Point (SSP) and altitude of a satellite. It encapsulates the satellite-orbit model described in Ref. [2] This communication with *controller* is done via the *controllerGround Tracker* port. The *azElProvider*, given the SSP, altitude, and tracking station's coordinates (latitude and

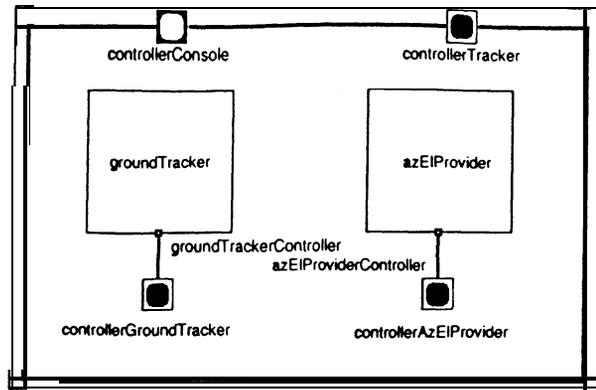


Figure 3: Tracking controller actor

longitude), is responsible for providing the azimuth and elevation of the antenna. The exchange of messages with the *controller* is done through the *azElProviderController* port. The *controller*, as its name implies, coordinates its components and communicates with *tracker* and *console*. It gets inputs from *console* and sends outputs to *tracker*. In order to promote modularity and reuse of actors, the components of *controller* communicate only with their container and not between themselves. This way, module coupling is greatly reduced and reusability is enhanced.

The controller behaves as pictured in Fig. (see Fig. 2). The transition between the initial state (black circle) to state *Wait TrackParm* reads the station's latitude and longitude and transmits this information to *azElProvider*. The transition (either from *WaitTrackParm* or *Ready* to *Ready*) labeled *Set TrackParm*, is triggered by the *SetElements* message from *console* that conveys orbital elements for a selected satellite. The transition labeled *Start Tracking* from *Ready* to *Tracking* is triggered by the *StartTracking* message from *console* and begins a tracking run. The transition labeled *Timeout* is triggered on a periodic basis and initiates the process of updating the direction of the antenna. A tracking run is halted when the transition labeled *StopTracking* is triggered.

3 Implementation

In this section, we discuss the implementation of our software. Issues are the hardware and Operating System (OS) on which our software must execute, programming language in which our application is coded, and mapping of concepts of our design model to OS and programming language concepts. Let us first consider the hardware. We have the following pieces of equipment: a Pentium class micro computer, a V/UHF all mode transceiver, a multi-mode digital signal processor, an azimuth-elevation rotor along with its computer interface, a VHF antenna, and a UHF antenna.

We have chosen the QNX operating system which has been developed specifically for real-time applications. It supports multitasking and fast context switching. QNX has a micro kernel architecture which means that its kernel is light enough to reside in the processor cache. Consequently, it is very performant. Several processes can run concurrently and QNX provides message based interprocess communication primitives (send, receive, and reply). The programming language selected for the implementation is C++ because, conceptually, it is the closest to the ROOM model

among those available on QNX.

Actors of the ROOM model can be mapped either to QNX concurrent processes or C++ sequential objects. Note that in the ROOM model, there are actors that can potentially run in parallel (such as *console*, *controller*, and *tracker*) whereas others run purely in sequence with respect to each other. (such as *controller*, *groundTracker*, and *azElProvider*). On the one hand, actors that have the potential to run in parallel are mapped to processes communicating with the QNX message passing primitives. On the other hand, every group of actors that run in sequence is mapped to a single process. Each actor becomes an object and communicates with the other actors in the group with C++ method calls. This avoids the overhead that results from the calls to OS primitives thus increasing efficiency. Hence, the group of actors *controller*, *groundTracker*, and *azElProvider* is mapped to a single process whereas the actors *console* and *tracker* are both individually mapped to a process. The resulting implementation therefore results in three processes. These implementation strategies result in a simple and efficient organization.

4 Conclusion

This paper has presented the object-oriented development of a real-time satellite tracking system. It illustrates application of object-oriented design principles in the field of satellite telecommunications. Use of a methodology such as ROOM leads to a very well structured software that makes it easy to understand and maintain. This project is not only a simulation but a full development including a working implementation based on the QNX operating system running applications with real-time performance. The demonstration performed in this project is important because it shows the relevance and suitability of state of the art software development techniques and tools in a field where classical structured (and non structured) software development techniques are still largely employed.

Future work in our project includes development of a graphical user interface, remote control through TCP/IP of the station, automatic control of the transceiver, and improvement of the satellite-orbit mathematical model.

Acknowledgments The authors would like to thank Francis Bordeleau from Carleton University for fruitful discussions about the ROOM model of our satellite tracking software.

References

- [1] G. Booch. *Object-Oriented Analysis and Design with Applications*. The Benjamin/Cummings Publishing Company, Inc. Second Edition, 1994.
- [2] M. Davidoff. *The Satellite Experimenter's Handbook*. The American Radio Relay League. Second Edition, 1994.
- [3] QNX Software Systems Ltd. *QNX System Architecture*. 1993.
- [4] B. Selic, G. Guzikson, and P. T. Ward. *Real-time Object-Oriented Modeling*. John Wiley and Sons, Inc. 1994.

XNET

A GRAPHICAL LOOK AT PACKET RADIO NETWORKS

Richard Parry, W9IF
rparry@qualcomm.com
<http://www.qualcomm.com/~rparry/xnet>

ABSTRACT

This paper describes AX.25 packet networks from a graphical standpoint using XNET, a software program specifically designed for network analysis. Networks are complex entities most easily explained visually. Through the graphical displays, one can more easily gain an appreciation and understanding of a network. It also allows one to see problems and the general behavior of the network. *XNET* runs on UNIX/LINUX systems supporting the Tcl/Tk language.

KEYWORDS

Packet Radio, Tcl/Tk, Linux, UNIX, AX.25

INTRODUCTION

Amateur packet radio networks can be an enigma. The average packet user sits in the ham shack enjoying error-free communication while sporadically hearing the Terminal Node Controller (TNC) toggle the transmitter on and off for no apparent rhyme or reason. Somehow messages are sent and received, files are uploaded and **downloaded**, countless messages fly through the ether without error, all despite the whims of both atmospheric and man-made interference. What is most miraculous, is that the users operating in this manner share the same frequency, an impossibility on virtually all other standard analog modes of communication such as SSB, AM, or FM; spread spectrum being one of the few exceptions.

Understanding and studying a network is a complex matter. Fortunately, network analyzers are available to aid the packet user in understanding how a network operates. Two such programs are *XNET* for UNIX/LINUX systems, and *Packet Tracker* for Apple Macintosh computers. In this paper, XNET was used for all graphic network displays. Even for those who have no intention of using a network analyzer program, the diagrams in this paper and the XNET Home Page on the World Wide Web should prove to be of interest as they clearly show how a packet network functions thereby providing greater insight into networks.

XNET

XNET is an X-Windows based network analyzer designed specifically to monitor AX.25 packet radio networks. It will collect and display network data allowing the user to understand network traffic and channel utilization. The program is written in Tcl/Tk (Tool Command Language / Toolkit) and can operate on any computer system supporting Tcl/Tk and the UNIX operating system. The only additional hardware necessary to use XNET is a radio receiver and a Terminal Node Controller.

XNET provides many features that are useful to both the casual packet user as well as a packet radio BBS or DX packet cluster system operator wishing to better understand a packet network. Functions performed by XNET include:

- Packet counting
- Displaying network nodes and statistics
- Graphical representation of network utilization
- Displaying raw network traffic
- Playback of network traffic
- Visual display of network connections
- Extensive use of color and GUI to display information

It is reasonable to mention what XNET does not do, since there are some limitations. The limitations are not necessarily in the program, but in the system. For example, data received by XNET from the TNC is filtered, resulting in lost information that cannot be used for network analysis. Specifically, all packets sent to XNET from the TNC have a valid FCS (frame checksum), since the TNC will not pass packets to XNET with an invalid FCS. Therefore XNET has no way of keeping statistics regarding collisions, communication errors, and other similar parameters that the TNC filters out.

These limitations can be cured by placing the TNC in the KISS mode which causes the TNC to pass raw bit level data. XNET would then be responsible for parsing the packets, error checking, and numerous other functions. The resulting program would provide additional useful information, however, it would be a significantly different program.

MAIN CONSOLE

When XNET first begins, a single window called the *console* is displayed as shown in Figure 1. This console provides the primary user interface for the operation of XNET. The main console's primary purpose is to allow the user to select desired features such as the MAP, TERM, NODES, and GRAPH windows, as well as setting preferences using the SIMUL, PORT, and PREFS windows to configure the program. The START and STOP buttons perform the function of beginning and ending the program, and the ABOUT button provides a short description of the program, author, and version.

The Start Time as shown in the console, is the actual time and date when XNET was started by the user. This time is used for computing *Total Packets* which represents the total number of packets received from the starting time and date.

Elapsed Time represents the duration in hours, minutes, and seconds since XNET was started. In the example, XNET had been running nearly two hours.

Total Packets provides the grand total number of all packets received by XNET from the initial start of the program. This value is always equal to or larger than the number of *active packets*.

Active Packets are packets from stations that are currently on the network. The difference between *active* and *total packets* arises from the fact that stations leave the network and thus have timed-out, resulting in packet traffic being removed from the active packet count. In this example, during the nearly

two hour duration that the program has been running and collecting statistics, a total of 1,442 packets have been received, and 363 packets have been received from nodes that are currently on the network .

Total Nodes represents the total nodes that have been monitored from the initial *Start Time*. This value is similar to the *Total Packets* since it represents all nodes that are presently on the network as well as those that have disconnected from the network. Therefore, *Total Nodes* is always larger than or equal to the *Active Nodes*.

Active Nodes, as its name indicates, represents all nodes that are currently active on the network. As we shall see when we discuss the NODES window, details of these active nodes are displayed there. The difference between the *Active* and *Total Nodes* arises from the fact that some stations are removed from the active node list when they have timed-out. This results in the nodes being removed from the active node count, while still being accounted for in *Total Nodes*.

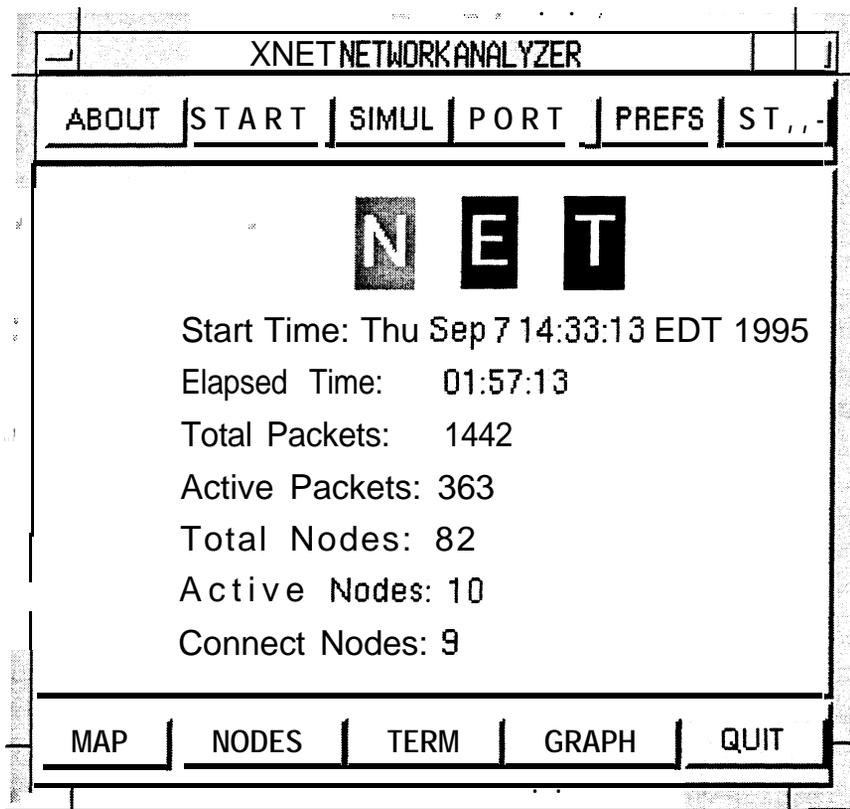


Figure 1. Main Console

Connect Nodes represents the number of nodes displayed in the MAP window. All the nodes listed on the MAP and therefore counted, are nodes which have transmitted packets or have been sent packets. Therefore, in the example, during the nearly two hour running period, 82 nodes were active on the network, only 10 nodes are currently active, and 9 of the nodes have sent and received packets.

MAP WINDOW DEFINITIONS

The *map window* gives the “big picture” of the network. It is intended to be a graphical representation of a heard node on the network sending packets to a destination node. Before examining

an actual map window, let's take a moment to understand how to read the maps and define a few key terms.

Heard *Nodes* are stations that have actively sent a packet; they are displayed on the map in blue. Nodes that have been sent a packet are referred to as *Destination Nodes*; they are displayed in brown. It is important to note that a destination node may or may not be on the network. Just because a node sends another node a packet, we have no way of knowing if that destination node is on the network until that destination node sends a packet. Then and only then is it promoted to heard station status. Other nodes which are on the network, but not heard directly, are referred to as *unheard stations* and are not displayed on the map.

Referring to Figure 2, first note that each node is drawn twice, once as a source (left side) and once as a destination (right side). In addition, although no arrowheads are attached to the line by XNET, the line always represents packet traffic from source to destination (i.e., left to right). If NODE2 sends a packet to NODE1, a line is drawn from NODE2 on the left (source) to the NODE1 on the right (destination). Similarly, if NODE 1 sends a packet to NODE4, a line is drawn from NODE1 on the left (source) to NODE4 on the right (destination). Lines are drawn indicating that a packet was sent, however, there is no *connection*. We will shortly discuss exactly what we mean by a *connection*.

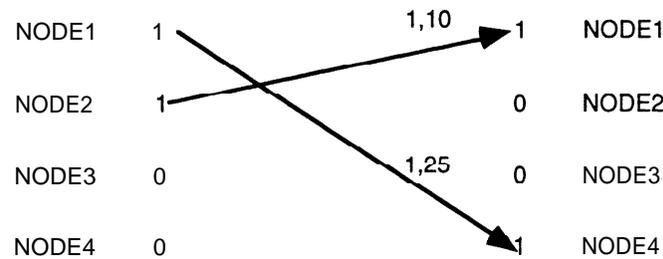


Figure 2. Unconnected Nodes

Let's look at the example shown in Figure 3. If NODE1 sends a packet to NODE2, a line is drawn from NODE1 on the left (source) to NODE2 on the right (destination). If NODE2 returns a packet to NODE1, a line is drawn from NODE2 on the left (source) to NODE1 on the right (destination). The result is a connection. XNET displays a connection using blue lines rather than brown.

A further explanation of connected nodes is in order. AX.25 is a *connection oriented protocol*. To a telecommunication's engineer this has a very specific meaning. If a node sends a packet to another node, a line is drawn from the source to the destination node. If the destination node returns a packet to the source node, XNET assumes that a connection between the nodes is in place. Admittedly, XNET loosely interprets a connection. One could argue that each node is sending packets to each other but neither is listening, thus a connection is not in place. Although possible, this scenario is assumed to be unlikely.

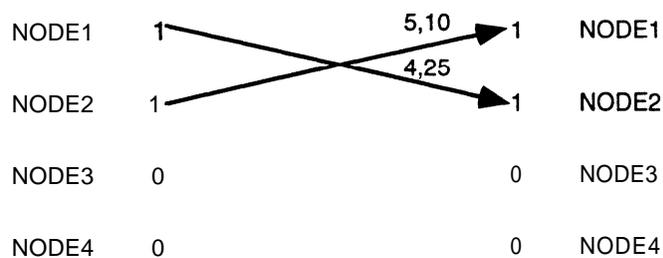


Figure 3. Mutually Connected Nodes

The XNET maps display additional information. Near the destination node column on the right of the map, there is a pair of numbers separated by a comma. The first number represents the number of packets sent to the destination node by the source node. The second value is the time in seconds from when the source node sent a packet to the destination node. Returning to Figure 3, note NODE 1 has sent 4 packets to NODE2 and 25 seconds has passed since that packet was sent. Similarly, NODE2 has sent 5 packets, and 10 seconds has elapsed since that last packet was sent.

MAP WINDOW EXAMPLES

We are now ready to examine a few real network maps. These maps were developed from monitoring actual on the air packet networks in the Dallas, Texas area.

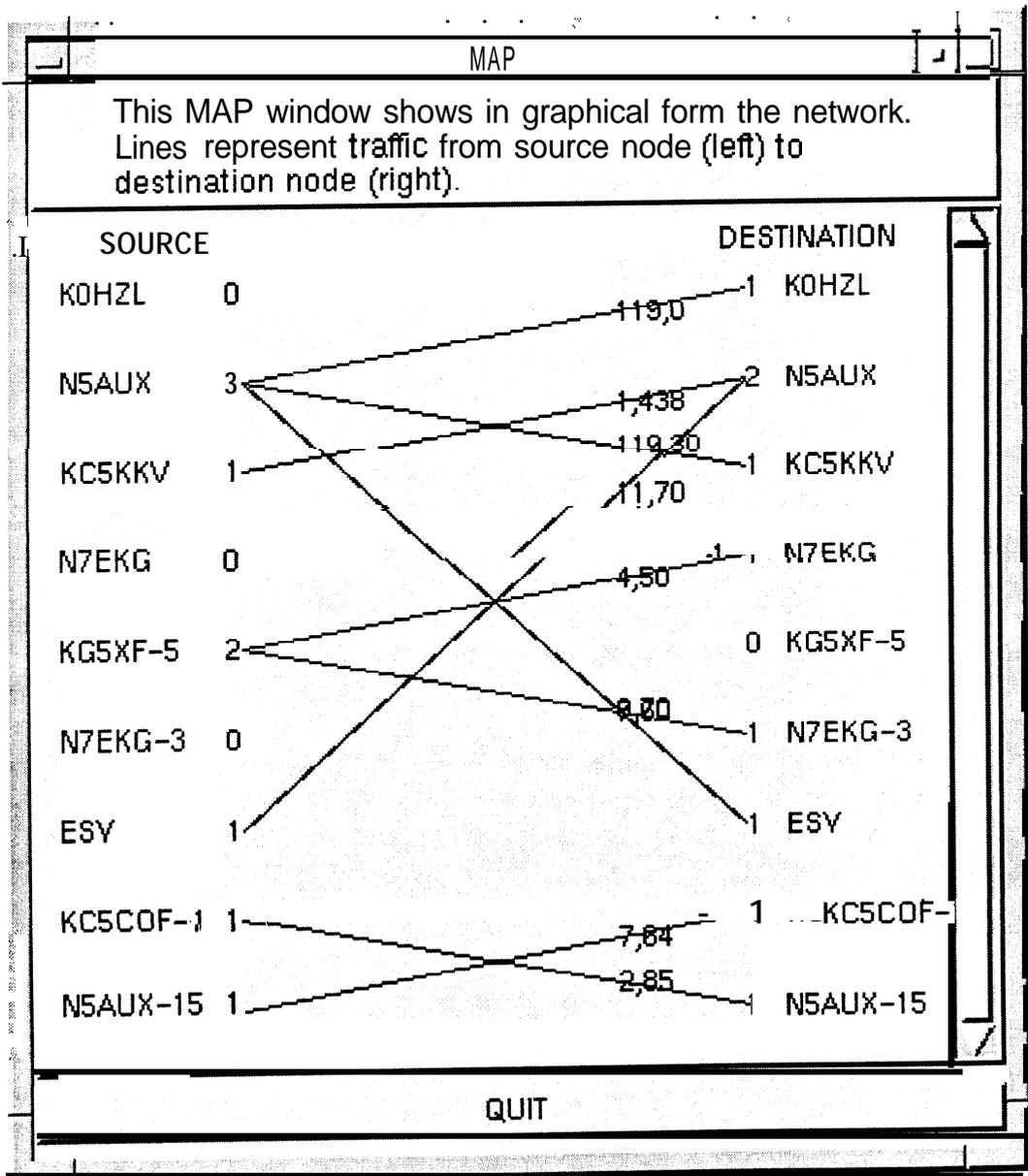


Figure 4. Simple Network

The figures shown below represent packet networks from two different types of networks. Specifically, Figures 4 and 5 depict a normal simple channel network and a APRS packet network respectively. It is important to emphasize that not all nodes are displayed; the map displays only heard stations and stations that have been sent a packet.

In Figure 4, N5AUX sends a packet to node ESY. ESY returned a packet to N5AUX resulting in a connection as previously defined. A pair of blue lines are drawn by XNET to indicate the connection. N5AUX is also connected KC5KKV, shown again by the pair of blue lines. KC5COF and N5AUX-15 are yet a third pair of connected nodes. The remaining nodes are not connected, meaning that packets have been sent, but the destination nodes did not return a packet to the sender.

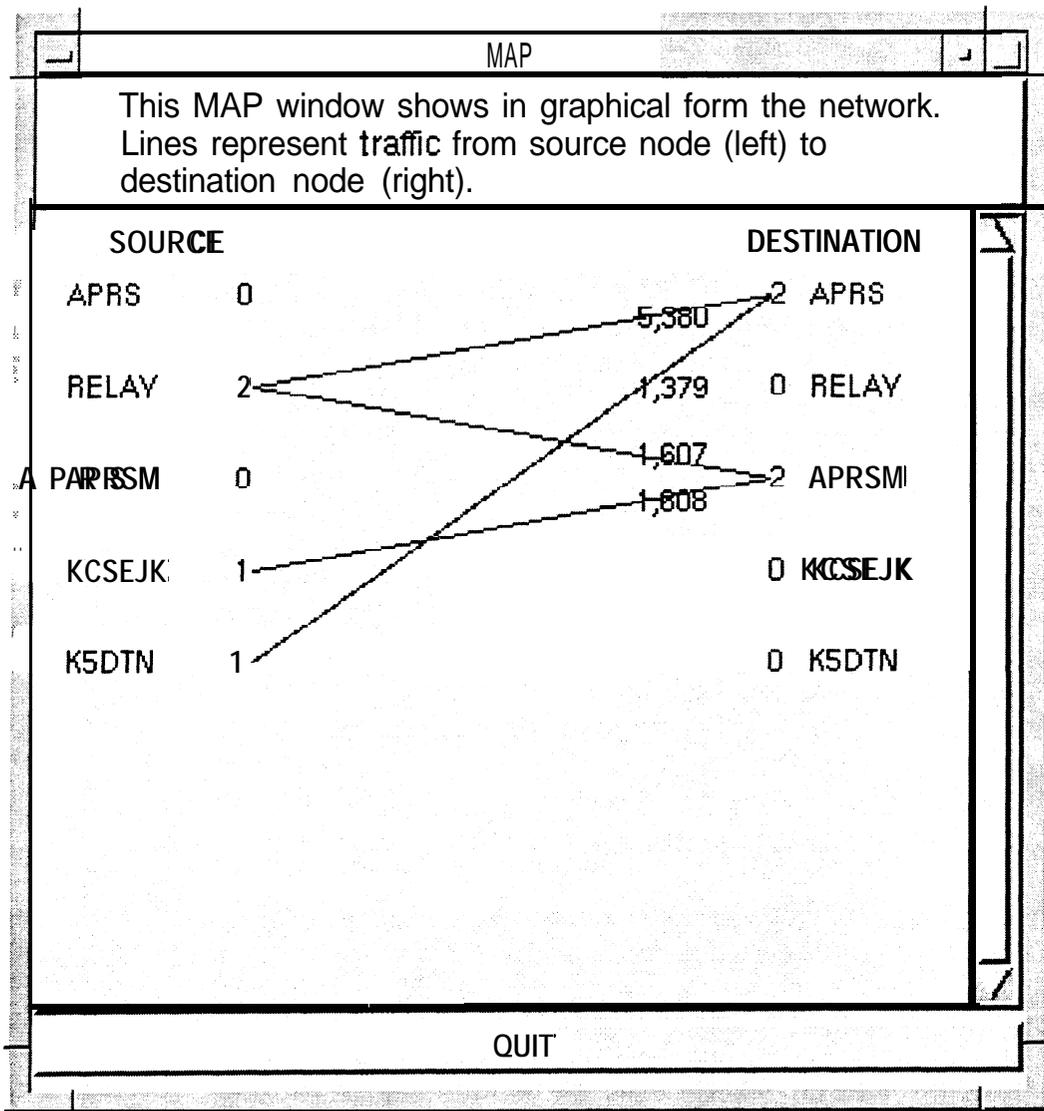


Figure 5. APRS Packet Network

Figure 5 shows an APRS (Automatic Packet Reporting System) Packet Network. As one might expect, there are no AX.25 connections. XNET displays this by showing only brown lines. This is not surprising since all stations in an APRS merely transmit UI (Unnumbered Information) frames at

intervals. The source node does not listen, it only transmits packets which include the nodes present latitude and longitude. They do not request or expect a response. Typically intervals for transmission is 1 minute if the station is mobile, and 60 minutes for fixed station transmitters. These UI frames contain the latitude and longitude of the station.

Those familiar with APRS networks will note the existence of the RELAY and APRS nodes which are used to “repeat” the packet to provide wider coverage for both fixed and mobile nodes on the network.

The reader is encouraged to browse the XNET web page for further examples and for colored map displays which greatly enhances the ease of understanding the network.

NODES WINDOW

The NODES window is intended to provide additional information that is not conducive to displaying in the graphical MAP window. It also displays all nodes, both heard and unheard, while the MAP window is limited to displaying heard and destination nodes only.

Figure 6 shows an example of a NODES window, which shows all currently active nodes. Other network statistics include the number of packets transmitted by that station, network utilization in percent, and the elapsed time since a packet was last sent by that station.

Remember that XNET does not monitor disconnect packets to detect a node disconnect. It determines if a node is active from the age of the last packet transmitted by that node. Detecting disconnect packets could be useful, however, many nodes leave the network, without sending a proper disconnect frame. Therefore another method needs to be used by XNET to ascertain connect status. That method is to *age* the packets and remove nodes that have exceeded the time-out period. The PREFS menu, which we will discuss subsequently, is provided to allow the user to specify the *time-out* period. If the time out period is exceeded, the node is assumed to be disconnected. For example, if the time-out is set to 1 minute and XNET has not heard from that node for 1 minute, the node is assumed to have disconnected from the network and is removed from the window.

STATION This column is reserved for displaying the name of the active stations. Active stations are those heard and unheard stations that have not timed out (e.g., they have not exceeded the time-out period specified by the preferences).

PACKETS (PKTS) The total number of packets transmitted by the specified station is displayed here. Note that the unheard station packet count is always zero since only heard stations transmit packets.

PERCENT (%) This metric represents the percentage of total packets transmitted by the currently active node with respect to other currently active nodes. This value is very useful as it quickly shows which of the nodes is responsible for most of the traffic. This value should not be confused with network utilization. That information is best gleaned from the graph window. In the example, W5XJ has sent 172 packets which represents 97% of the total traffic.

LAST COMMAND (FRAME TYPE) The AX.25 data link layer protocol specifies a type for all frames. Frames fall into three main categories, Information, Unnumbered, and Supervisory. For W5XJ, the last command sent was RR which is a Receiver Ready command.

AGE The age indicates the time in hours, minutes, and seconds, since the last packet was sent. For example, W5XJ has not sent a packet for 3 minutes 10 seconds.

MESSAGE (LAST INFO) Information packets contain the useful information that is to be transmitted to the receiving station. The term useful is used here to indicate that this **frame** contains the information that is of primary interest to the user. Most of the other **frames** are overhead frames

required by the protocol. Only the first few characters of the information string are displayed in this window when it is initially opened. The window may be re-sized to show the entire field if the user wishes.

COLORS Heard stations are shown in blue and unheard stations are shown in brown.

NODES LIST						
This NODES window shows in tabular form all nodes. The originating "heard" station is shown in blue, "unheard nodes" are brown.						
	Station	Pks	%	Lst Cmd	Age	Info
00	W5XJ	172	97	(RR.P:1)	0:03:10	
01	W5IWE	0	0		0:03:30	
02	NA5S	4	2	(RR.F:4)	0:03:20	
03	N4YGP	0	0		0:03:10	
04	KJ5NB	0	0		0:03:40	
05	DXC	0	0		0:03:16	
06	W5KVO	0	0		0:03:16	
07.						
08.						
09.						
10.						
11.						
12.						
13.						
14.						
15.						
16.						
17.						
18.						
19.						
QUIT						

Figure 6. Nodes Window

GRAPHWINDOW

The GRAPH window shown in Figure 7, provides channel utilization information. This window can display channel utilization for 1, 5, and 25 hour periods. The user may freely move from graph to graph to obtain more or less resolution as desired.

The graph should be used as a *relative* rather than *absolute* value of network utilization. Specifically, XNET computes channel utilization based on the number of characters received during a one minute sampling period. The magnitude (height) of the vertical line is also based on the RADIO BAUD rate selected in the PREFS menu. Note that this value is used for computational purposes only and represents the baud rate of transmission over the radio link (not the data rate between the computer and the *TNC*).

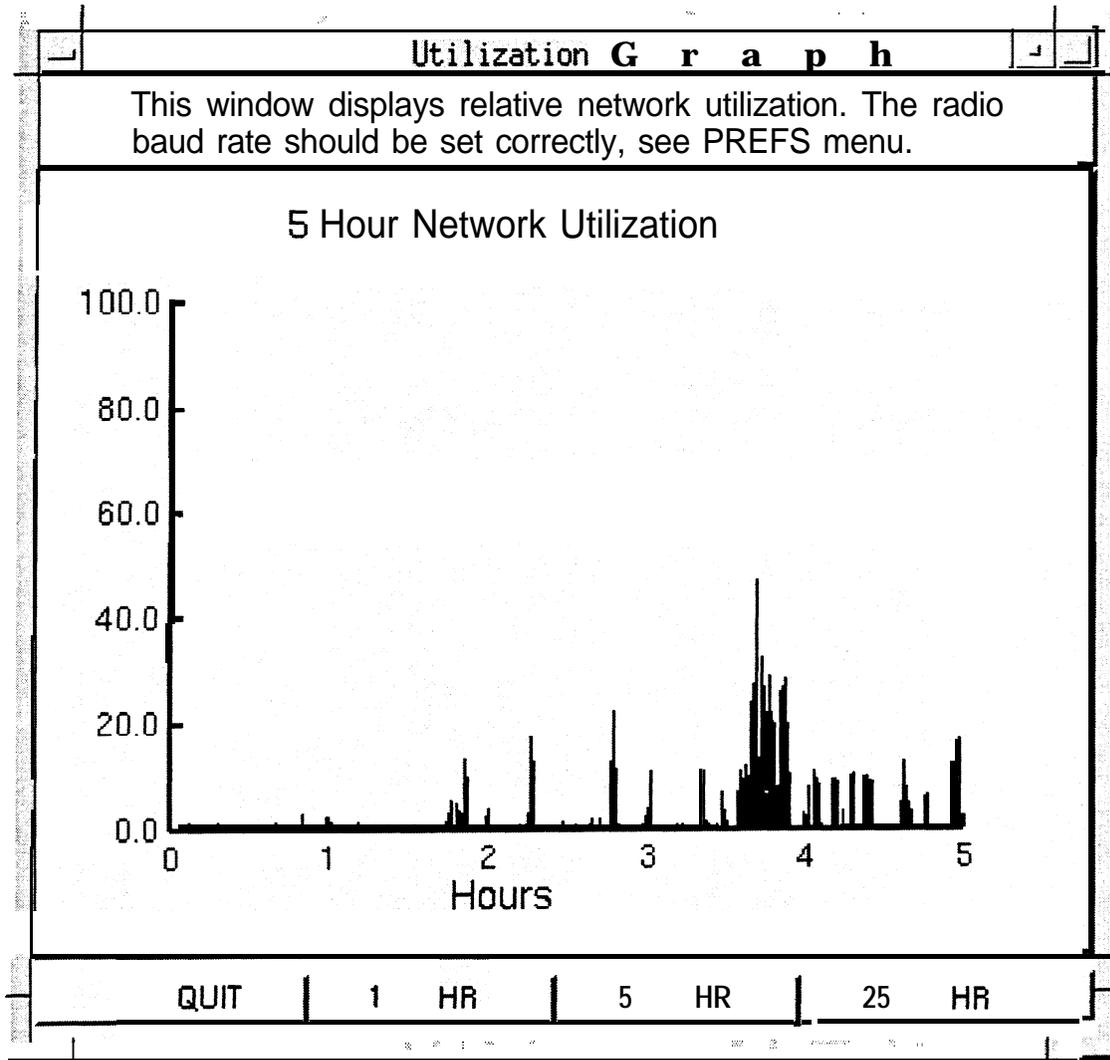


Figure 7. Graph Window (5 Hour Logging)

PREFERENCES

The PREFS window shown in Figure 8, is provided to allow the user to specify miscellaneous XNET preferences. These include the baud rate of the radio link, and various time out periods. The Radio Baud Rate represents the bit rate of the radio link and is used exclusively for the computation of network utilization as displayed in the GRAPH window.

configured for the desired port parameters such as the baud rate, stop bits, parity, etc. Note that if a simulation is selected, the port parameters have no affect (i.e., they are not needed).

Selecting a simulation is ideal for testing and educational purposes. The simulations are actual prerecorded off the air network traffic. However, the traffic is not played back in real time. The traffic has been designed to send information at a pre-defined interval (1 second in most cases).

TERM WINDOW

The TERM window (not shown) displays raw network traffic. However, the packet is truncated if it exceeds the width of the window. This window is intended to allow the user to see packets as they are received by XNET. Truncating the packets is performed to allow the user to easily see the header of each packet, since that is of most interest.

SERIAL PORT PREFERENCES

The serial PORT configuration window (not shown) is provided to allow the user to specify port parameters. Here the user specifies the port, stop bits, frame size, and parity. Since all information that XNET uses comes from the TNC which provides the information over an RS-232 link, it is necessary that the two devices use the same transfer protocol.

CONCLUSION

XNET is intended to provide information about AX.25 packet networks that is difficult and perhaps impossible to ascertain by other means. The program displays network statistics in both tabular form and an easily understood graphical form. Through the use of graphical displays, color, and an easy to use interface, the program provides an excellent tool for the analysis of amateur radio AX.25 networks.

ACKNOWLEDGMENT

I would like to thank Mark Sproul, KB2ICI, for *Packet Tracker*, a program similar to XNET for the Apple Macintosh. Packet Tracker provided the inspiration for XNET and many of its features have been included.

DISTRIBUTION

XNET is a freeware program available under the GNU General Public License, Version 2, June 1991, Free Software Foundation, Inc. 625 Massachusetts Avenue, Cambridge, MA 02139. It may be downloaded from any of the following sources. Further information regarding the necessary files to download and the system requirements are included there. The present version as of this writing is 1.1. The files are tarred under than name `xnet-1.1.tar` and need to be extracted. The method is described in more detail in the `readme` file, `xnet-1.1.README`.

The author's home page is the primary download site and source for information.

<http://www.qualcomm.com/~rparry/xnet>

The following is another possible download site for XNET and other amateur radio software programs.

<ftp.ucsd.edu/hamradio/packet>

REFERENCES

1. Fox, Terry L, "*AX.25 Amateur Packet-Radio Link-Layer Protocol*," Version 2.0 October 1984, American Radio Relay league, Inc. Newington, Connecticut, October 1984.
2. Ousterhout, John K, "*Tcl and the Tk Toolkit*," Addison-Wesley, Reading, Massachusetts, 1994
3. Sproul, Mark, "*Packet Tracker: A Graphical Packet Tracking Program*," 12th ARRL Digital Communications Conference, Tampa Florida, September 1993, pp 76-82.
4. Welch, Brent B, "*Practical Programming in Tcl and Tk*," Prentice Hall, Upper Saddle River, NJ 1995
5. Key Tcl/Tk Web page URLs
<http://www.sunlabs.com:80/research/tcl/>
<http://www.sunlabs.com:80/~rjohnson/tcl/>
<http://www.tios.cs.utwente.nl/~binkhors/tcl.html>
6. Key Linux Web page URLs.
<http://sunsite.unc.edu/mdw/linux.html/>
<http://www.cris.com/~hallow/linux/2.0.html/>
<http://www.ddj.com/ddj/1995/1995.05/welsh.htm/>
<http://www.geog.ubc.ca/sparclinux.html/>
<http://www.redhat.com/linux-info/>
7. Amateur Radio Specific Linux Web page URL.
<http://www.rahul.net/perens/LinuxForHams/>

A 9600 Baud modem for the LPT port

Wolf-HenningRech,N1EOW,DF9IC@DB0GV.DEU.EU,df9ic@amsat.org,ReichenbergerStr.7,D-71229Leonberg,
Germany
Donald Rotolo, N2IRZ, 545 Baylor Avenue, River Vale, NJ07675

Introduction

The 9600 Baud FSK modulation according to **G3RUH** uses our limited bandwidth **ressources** much more efficiently than the popular 1200 Baud AFSK. But one reason for the packet user not to switch to FSK is the higher price of the modem hardware - usually a TNC2 clone with a large FSK modem board. In contrast, 1200 Baud AFSK can be decoded with a very simple and cheap adapter to a PC's COM port containing not much more than a **TCM3105**, and originally designed by Johannes **DG3RBU (BayCom)**. Packet decoding is done by software in the PC, and there is a wide variety of drivers. Even the power supply is derived from the COM port.

We present here a simple modem for 9600 Baud FSK which can be connected to a LPT' port (provided its **IRQ** is installed). It is also powered from the port and does not need any alignment. Several drivers for DOS [1] and **Linux** are available because of its compatibility to the **BayCom** PAR96 modem (and its **PacComm** clones). With DOS and the **BayCom** program it operates with any computer higher than a 10 MHz 286. It has been originally published in [2].

Hardware description

Fig. 1 shows the circuit diagram. The key to the high functionality with few parts is the use of a microcontroller from Arizona Microchip, the **PIC16C84** [3]. This IC contains a complete microprocessor system which can operate from a **2V/1mA** power supply. A dual CMOS shift register is used to decouple the time-critical data transfer to the host PC, and a quad opamp provides all necessary analogue filtering including the threshold comparator function. The Analog Devices **OP491** is a micropower opamp with medium speed and rail-to-rail capability down to 2.7 V single supply. The analog circuit is equivalent to other FSK modems, thus preserving the high signal purity.

The complete circuit is supplied via 3 data lines from the interface. Any typical LPT port output is similar to at least a LS-TTL output and can drive few **mA** against 3.3 V. Desktop PCs usually have a lot more power in the LPT to drive long shielded printer lines with high speed. The modem has been tested with a variety of notebooks and desktops, and the internal modem supply never fell below 3.0 V.

Microcontroller internals

The processor is always running in one of two software loops with a length of exactly 96 command cycles, one for receive and one for transmit operation. Because one command cycle uses 4 clock cycles this is just one bit period of a 9600 Baud signal. Both loops are written such that the execution time is independent of all branches and subroutine calls. No interrupt or timer is used in order to maximize execution speed.

During transmit most time is used for the calculation of the output samples which is done 4 times per bit length. The 8 stage FIR filter response on a bit stream is stored in two tables, and combined by the PIC in real time in order to save ROM space. One single table would require $256 \times 4 = 1\text{k}$ byte (8 bit length, 4 times oversampling) which exceeds the limits of the 16C84. By using two tables, where the response is separated into two 4-step parts, only $2 \times 16 \times 4 = 128$ bytes are required. The calculated sample is then output on a 7 bit wide port and D/A converted by a metal film resistor array.

During receive the threshold comparator output is sampled four times per bit. These binary samples are the only input to the decoder. First the clock is regenerated by a smart DPLL with a fast lock time but a smooth track as long as DCD is active. The DPLL is realized by incrementing or decrementing the code loop by one NOP - the whole software runs behind the input signal. Then the correct sample per period is used for demodulation. DCD is generated from the 4 samples which characterize the position of the input zero crossings. The derived raw DCD information is averaged over 50... 100 periods for a reliable DCD operation. Scrambling and bit stuffing are done in the PC driver, according to the original BayCom PAR96, to maintain compatibility.

The PC interface

Every 16th bit of the HDLC data stream the PICPAR modem generates an interrupt via pin 10 of the LPT. The PC responds to this interrupt and inputs or outputs 16 bit of data via pin 2/12 with the clock coming from the PC at pin 4. This burst data transfer minimizes the interrupt load and allows operation even with moderate speed PCs. The data is stored in a 16 stage shift register and then transferred serially again to (or from) the PIC - this time the clock comes from the PIC to match with its close real time HDLC port operation. DCD and PTT are connected statically via extra lines.

Construction

The prototypes have been built on a two layer PCB of about 2.5 x 3.7 sq. inch. This version is shown in the Figures 2 and 3. The component density is low, and there are only few tracks on the component side of the board. The board layout is public domain for noncommercial use in amateur radio and can be obtained from one of the authors (df9ic) as HP/GL or PCL. The processor firmware will not be disclosed but programmed processors are available from [4]; also complete modems based on this design, as kits and finished modules. Especially attractive is a very small SMD version which fits into a D shell adapter case.

References

- [1] PC/FlexNet, a modular packet radio software package from Gunter Jost DK7WJ.
- [2] Rech, W.-H., DF9IC: Einfache Modems mit PIC-Prozessoren (simple modems with PIC processors). Proceedings of the 12th International Packet Radio Conference Darmstadt 1996, 7-17. (in German)
- [3] PIC 16/17 Microcontroller Data Book. Microchip Technology Inc, 1995.
- [4] BayCom GmbH, Bert-Brecht-Weg 28, D-30890 Barsinghausen, Germany. e-mail: kneip@mst.uni-hannover.de.

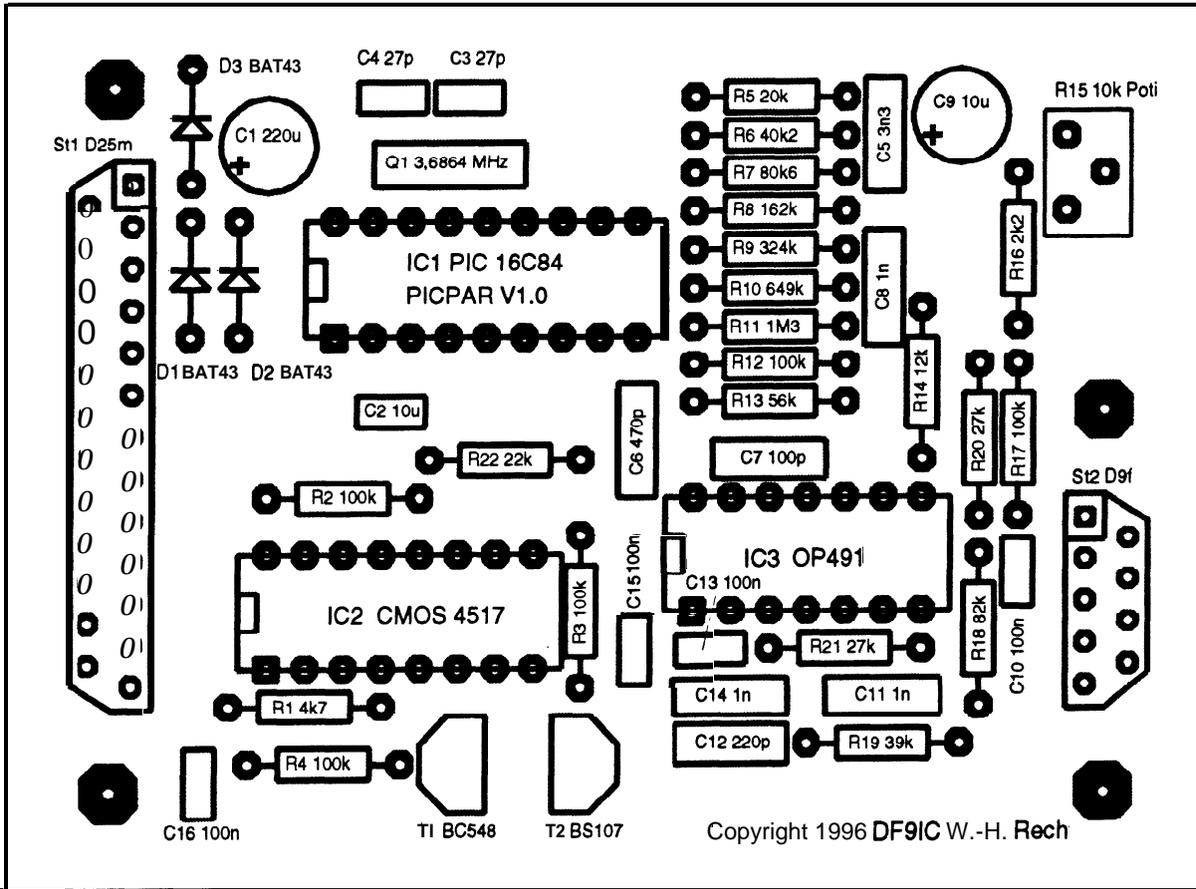


Fig. 2 PICPAR modem: component assembly

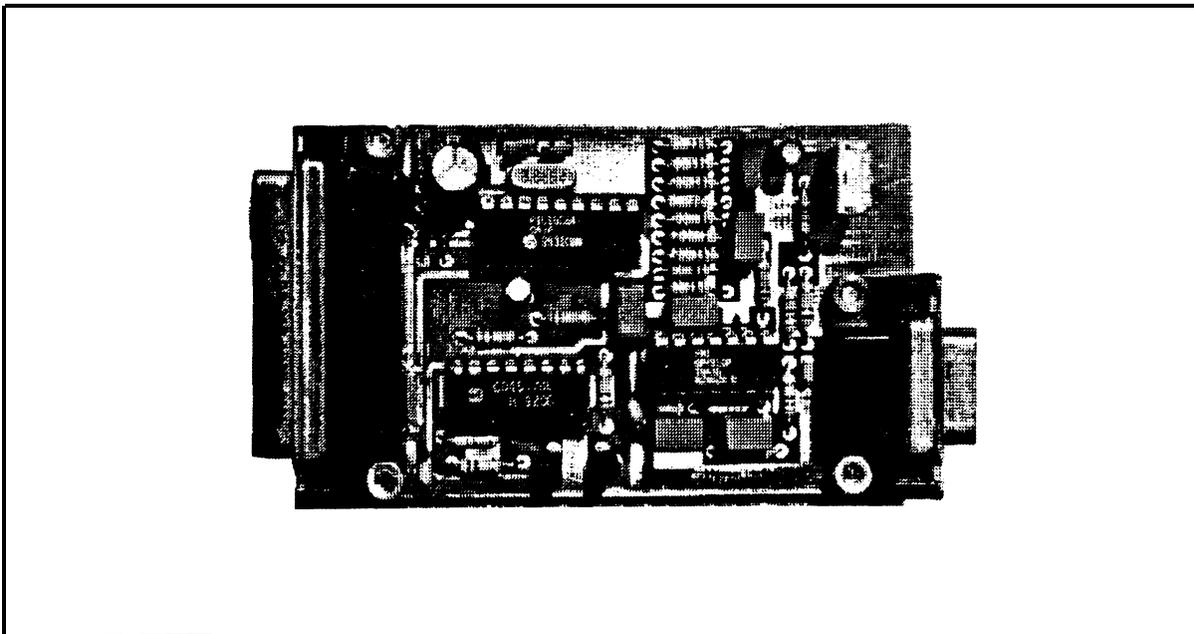


Fig. 3 PICPAR modem (original size)

Amateur Radio Digital Voice Communications

Time 's up.

Paul L. Rinaldo, W4RI
4559 Quality Street
Fairfax, VA 22030
prinaldo@arrl.org

The telecommunication industry has been busy developing digital voice transmission technologies. While amateurs have made progress in other areas, digital voice has not kept pace. Maybe its because of lack of suitable standards, the unavailability of hardware, satisfaction with the existing SSB and FM voice systems or the lack of urgency.

Well, *time 's up*. We need to apply the same energies and talents we did previously in developing single sideband, amateur television, small satellites and packet radio.

A design engineer once told me, the longer you wait, the easier the job. Obviously, he meant that new devices are constantly becoming available, maybe even **the specific application you have in mind**. This is true in the case of digital voice coder-encoders (CODECs). Instead of developing coding algorithms and designing application-specific integrated circuits (ASICs), someone else may have handled these chores for you.

CD-quality Digital Voice

Digital audio broadcasting (DAB) standards are now shaking. The original hope was that one standard could be used for both satellite and terrestrial delivery to cars, **portables and the home**. **Eureka 147 is a system that has gained some favor in Europe and Canada**. For a while, it looked like the United States was going to accept it as a standard but domestic broadcasters felt it important to keep broadcasting local and hang onto their existing market areas. This made the so-called **in-band on-channel (IBOC)** scheme look more attractive. In fact, it was found that stereo digital sound could be transmitted some **dB** down from an existing FM program in a VHF channel.

While interesting, these technologies haven't been directly applicable to amateur communications. DAB strives for high fidelity stereo, comparable to that of a **compact disk**. **So far, at least, amateurs haven't needed CD quality**. **That's a dangerous statement to make**, of course, as it's possible that another speaker will be making a presentation along those lines.

Digital Cellular Telephones

Here's something a little closer to our needs. Audio quality requirements are similar to the public switched telephone network. While not being CD quality, it's good enough to accommodate speech of all languages and unfamiliar topics with clarity and low noise. Amateurs can tolerate lower voice quality and lower signal-to-noise (S/N) ratios, or have conditioned themselves to it so far. Lurking there is operator pride in being able to pull a signal out of the muck. Nevertheless, just listen to some of the lousy SSB voice signals we endure on HF or even on amateur satellites. VHF/UHF FM sounds lots better. Even that isn't up to the quality of the digital cellular telephone systems.

Chips Abound

If we start with the premise that we don't need digital cellular telephone quality or don't want to devote the bandwidth and power (or S/N) they do, we still can benefit from the abundant integrated circuits spawned by this industry. A romp through the trade journals, data sheets and appnotes will boggle the mind. One way to get a crash course is to sign up for itinerant seminars put on by semiconductor manufacturers. **They'll also tell you about their products for cordless phones and other wireless applications.**

Communications-Quality Digital Voice

FM two-way radios are generally compatible with each other, except perhaps in channel width. Channels ratcheted down from 60 kHz in the Jurassic FM age to 12.5 kHz these days with a strong feeling that half that would **be needed to refarm the two-way commercial spectrum**. Regrettably, the quality degrades to that of AM when an AM bandwidth is reached, according to the FM Improvement Ratio curves. **Both fidelity and frequency reuse are affected**, so you can't expect to get a 2:1 improvement in spectrum efficiency by chopping 12.5-kHz channels into twice as many 6.25 kHz wide.

What industry or the standards bodies have not done is to agree on a single digital voice technology for VHF/UHF two-way radios. While there is no clear winner at this stage, a number of digital voice coding schemes have been developed, notably APCO Project 25 in the United States, TETRA in Europe, IDRA in Japan and DIMRS in North America. A complete technical description of these technologies would be impractical. Fortunately, however, International Telecommunication Union Radiocommunication Working Party 8A has captured their salient features in a scant 47 pages, in a Draft New Recommendation entitled, Spectrum Efficient Digital Land Mobile Systems for Dispatch Traffic, which is reprinted with permission of the ITU as an appendix to this paper.

Of the above technologies, Project 25 looks particularly interesting in that RF channel spacings of 12.5 kHz and 6.25 kHz are shown, while the others use 25-kHz channels. Without considering other factors such as frequency reuse, it is not clear whether Project 25's reduced bandwidth offers commensurate improvement in spectrum efficiency.

There is a new US Federal Standard for a 2.4-kbit/s digital voice technique known as Mixed Excitation Linear Prediction (MELP). A brief description can be obtained as an IEEE reprint, O-7803-3 192-3/96 \$5.000 1996 IEEE. Listener tests have shown MELP at 2.4 kbit/s performs as well as a higher bit rate CELP at 4.8 kbit/s specified in Federal Standard 10 16.

Lead, Follow or Get Out of the Way

In the case of VHF/UHF amateur digital voice communications, it looks like we ought to *follow* what **industry** is doing, or at least a subset thereof. After all, there isn't that much difference between our two-way radios and commercial two-way radios. Before the amateur radio manufacturers will produce digital voice products, they'll need to have confidence that the standard they use will be stable and preferably the same one as they use commercially. That doesn't translate to just sitting and waiting for the commercial shakeout. We need to do some standards shopping.

We may be able to *lead* at HF. But we must hurry. There is work going on in the HF communications and broadcasting industries. The communications applications of digital voice are probably easier to implement than those for broadcasting, for the simple reason that broadcasting requires cheap receivers for the mass audiences. It doesn't mean that digital communications receivers should be expensive, but there is more price flexibility and the quantities are manageable.

Is it feasible to use something like 2.4-kbit/s MELP for amateur HF digital voice? Does it need a clear channel or can it tolerate QRM? What if the QRM is on-channel or mistuned? Will it have advantages over SSB? How do we socialize the introduction of digital voice into what other hams consider SSB voice bands? Remember the onslaught of SSB into what were Ah4 voice bands? Add more questions of your choice.

It's not generally known that the emission designators for digital voice are already written into Part 97 of the FCC's Rules. See §97.3(c)(5) and §97.305(c) Emission Types Authorized column where "phone" is permitted. At least we don't need to go through the process of getting a Special Temporary Authority (STA) to experiment with digital voice. But we do need to give thought to the socialization issues mentioned above.

Spread Spectrum and Digital Voice

We need to be sensitive to voices that say, "Take it easy; don't QRM other users in the bands while introducing spread spectrum technology." Spread spectrum is still a controversial topic among amateurs and telecommunications professionals. Nevertheless, one of its forms, code division multiple access (CDMA) is making gains commercially. SS is neither a panacea or the end of the universe as we know it and everything we hold dear. It has its applications; we just need to choose the right ones and get on with it.

Present FCC Rules permit SS on amateur bands above 420 MHz. A current STA permits SS on all bands above 50 MHz. There are those who feel that SS should be permitted under the Rules above 50 MHz and there are others who think it ought to be allowed on all bands. Maybe this debate will be settled sometime soon.

Meanwhile, however, digital voice and SS are natural companions. We need to consider whether digital voice should use SS access techniques above 420 MHz where it's already authorized. While not wishing to debate the rule-change controversy, we can still treat it like dreams of the celibate. Would digital voice on SS be the solution to the short-range/long-range QRM problem at 50 MHz? Is SS on an overlay basis compatible with existing narrow-band modes in the HF bands?

Maybe the government has HF SS systems that they'd have to kill us if they told us about them. Industry is now working on HF SS technologies but there are still many unanswered questions. But it does offer the possibility of mitigating distortion resulting from multipath fading and it could reduce interference in crowded bands given the right techniques.

What Can a Body Do?

We should be following the industry and standards organizations in their pursuit of VHF/UHF two-way radio standards. This should also include a dialog between amateurs and Amateur Radio industry. There'll be a time to move, and we should recognize it when we see it. There is room for experimentation, **but** this may be more of an engineering problem of scoping what we want to do and then doing it.

Digital voice for the MF and HF bands requires a different strategy. We need to study the work being done by others, come up with some ideas for amateur applications and design some experiments. Because of the vagaries of ionospheric propagation, coding schemes that work fine at VHF or even in the MF broadcast band may flake out at HF. Here is a technical area where amateurs can contribute to the radio art.

Further Reading in the Amateur Literature

J. Bloom, KE3Z, "Help! CELP!," Empirically Speaking, *QEX*, Nov 1992.

J. Kneip, DG3RBU, and F. Radlherr, DL8MBT, "Voice Mailbox at the UHF/VHF Conference in Weinheim," translation by D. Moe, DJ0H/KE6MN, *QEX*, Dec 1991.

P. Hawker, G3VA, "The Bits of Speech - Digital Radiophones," Technical Topics, *RSGB Radio Communication*, Jun 1996.

A. Langi, VE4ARM, and W. Kinsner, VE4WK, "CELP High-Quality Speech Processing for Packet Radio Transmission and Networking," *proc., ARRL/CRRRL Amateur Radio 9th Computer Networking Conference*, London, Ontario, Sep 1990.

A. Langi, VE4ARM, and W. Kinsner, VE4WK, "Design and Implementation of CELP Speech Processing System Using TMS320C30," *proc., ARRL Amateur Radio 10th Computer Networking Conference*, San Jose, CA, Sep 1991.

N. Stone, WG1C, "Cellular Radio and the Modem Amateur," *QST*, Mar 1994.



Source: Document 8A/TEMP/86

APPENDIX

See p. 3 for copyright notice.

Working Party 8A

DRAFT NEW RECOMMENDATION ITU-R M.[8A/XB] SPECTRUM EFFICIENT DIGITAL LAND MOBILE SYSTEMS FOR DISPATCH TRAFFIC

(Question ITU-R 37/8)

Summary

Demand in the land mobile service is on the increase due to annual growth as well as to new data-based service requirements. This has led to the development of more spectrally efficient technologies utilizing digital modulation and in many cases trunking. These technologies are being introduced worldwide to accommodate this demand.

This Recommendation provides the technical and operational characteristics for spectrum efficient digital dispatch systems for **international** and regional use. The Recommendation also provides details of systems being introduced throughout the world.

The ITU Radiocommunication Assembly,

considering

- a) that the rapid development of digital speech transmission now enables both public and private mobile systems to operate in both national and international arenas;
- b) that digital transmission techniques offer flexibility for maintaining good communications quality;
- c) that equipment using digital modulation, speech coding, channel coding and digital signal processing techniques are now comparable in price with those using analogue technology;
- d) that digital transmission techniques can incorporate both speech and data facilities;
- e) that digital transmission systems can accommodate **different** types of data services;
- f) that digital systems can **offer** greater spectrum **efficiency** than existing **analogue** systems;
- g) that automatic sharing of channels using trunking is shown to improve channel efficiency;
- h) that there is continuing demand for privacy facilities;
- j) Question ITU-R 37/8 on systems with improved spectrum efficiency;

- k) that considerable development of multichannel digital systems, worldwide, has produced regional and national standards;
 - l) standardization of systems is required to permit roaming;
 - m) that digital systems may need to be optimized for the frequency bands in which they operate;
- recommends*

that the following technical and operational characteristics be adopted for spectrum efficient digital land mobile systems for international or regional use:

1 General objectives

The general objectives of a spectrum efficient digital land mobile system, for dispatch in either private or public systems, are to provide:

- systems that offer a higher spectrum efficiency, thereby accommodating more users within limited spectrum resources than analogue systems;
- a higher average level of voice quality over the network and enciphered speech for privacy;
- users with a wide range of services and facilities, both voice and non-voice, that are compatible with those offered by the public fixed networks (**PSTN**, PDN, ISDN, etc.);
- users with a variety of applications to satisfy their requirements, ranging **from** handheld stations to vehicle mounted stations, with voice and data interfaces;
- mobile and infrastructure equipment which use state of the art technology to provide savings in weight, power consumption and cost.

2 Service types

The basic **services** offered by a digital dispatch **traffic** system can be divided into three types:

- teleservices;
- bearer services; and
- supplementary services.

2.1 Teleservices

Teleservices provide the user with **full** capability, including terminal equipment functions, to communicate with other users. These services are typified by both lower layer (**OSI Layers 1 through 3**) and higher layer (**OSI Layers 4-7**) functionality.

Typical **teleservices** should include:

- a **trunked** and non-trunked capability to permit direct mobile-to-mobile and group speech call facilities with user options to permit selective and secure calling;
- telephony, facsimile and some extended service offerings, e.g. videotex, telex, etc.

2.2 Bearer services

Bearer **services** give the user the capacity needed to transmit appropriate signals between certain access points. These services are typified by lower layer functionality, typically limited to OSI Layers 1 through 3.

Typical bearer services should include:

- a circuit mode data facility to permit a minimum of 7.2 **kbit/s** for unprotected data and a minimum of 4.8 **kbit/s** for protected data;
- a packet mode connection-oriented data and connectionless data facility.

2.3 Supplementary services

The range of supplementary services varies depending on the system and also the particular implementation.

3 Channel design

The systems may have two types of channel categories:

- **traffic** channels which are used for voice and data transmission; and
- control channels which are used for **signalling** and control purpose, e.g. access control, broadcast messages, synchronization, etc.

4 Channel access techniques

Systems should use either **FDMA**, **TDMA**, **CDMA**, or hybrids of these. Digital cellular technology may be adaptable for dispatch use.

5 Systems being installed or planned

General details of the systems are given in Annex 1.

Annexes 2, 3, 4 and 5 give general descriptions of specific systems.

This Appendix is a **Draft** New Recommendation which has been approved by ITU-R Working Party XA. Prior to its publication, it is subject to approval by ITU-R Study Group 8 and administrations. This text has been reproduced with the prior authorization of the ITU as copyright holder. Complete volumes of ITU material can be obtained from:

International Telecommunication Union
General Secretariat - Sales and Marketing Service
Place des Nations - CH - 1211 GENEVA 20 (Switzerland)
Telephone: +41 22 730 6141 (English) / + 41 22 730 61 42 (French)
Telex: 421 000 uit ch / Fax: +41 22 730 51 49
X.400: S=Sales; P=itu; C=ch
Internet: Sales @itu.ch

ANNEX 1

Systems being installed and planned

1 Introduction

Digital land mobile radio systems for dispatch and fleet management applications are being developed worldwide. Although these systems have been developed to meet the requirements of either general purpose applications or more specific groups of users, they share some of the basic objectives and characteristics outlined in the Recommendation.

A description of the systems is given below and more detailed descriptions can be found in Annexes 2, 3, 4 and 5.

1.1

The development of the Standard? for **TETRA** has been carried out in the European Telecommunications Standards Institute (ETSI), a recognized standardization organization.

The technical requirements specification aims to satisfy the needs of a wide range of professional users ranging **from** emergency services to commercial and industrial organizations.

1.2 Project 25

The development of the standards for Project 25 has been carried out by Project 25, a combined effort of US local (Association of Public-safety Communications Officials International, APCO), state (National Association of State Telecommunications Directors, NASTD) and federal government users; in collaboration with the Telecommunications Industry Association, a recognized standardization organization.

The Project 25 standards aim to satisfy the needs of a wide range of users, primarily in the area of public safety and governmental operations.

1.3 Integrated Dispatch Radio System (IDRA)

The development of the Standards for the IDRA System has been carried out by the Association of Radio Industries and Businesses (**ARIB**) in Japan. **ARIB** is an external MPT (Ministry of Post and Telecommunication) **affiliate**, a recognized standardization organization.

The technical requirements of the specification aim to satisfy the needs of users over a wide range of professions, **from** emergency services to commercial and industrial organizations.

1.4 Digital Integrated Mobile Radio System (DIMRS)

The Digital Integrated Mobile Radio System (**DIMRS**) is one of the methods being used in North America to provide integrated dispatch services and increase spectrum efficiency.

2 Explanation of table

Table 1 presents the core parameters for these systems. In each case, complete specifications are, or will be, available from the relevant authorities as indicated in the annexes.

TABLE 1
Core parameters

Parameter	Project 25	TETRA	IDRA	DIMRS
Designation of emission				
- Traffic channels	8K10F1E, 5K76G1E ⁽¹⁾	25K0D7W/25KWDW ⁽²⁾	20K0D7W/20KWDW ⁽²⁾	20K0D7W/20KWDW ⁽²⁾
- Control channels	8K10F1E, 5K76G1E ⁽¹⁾	25K0D7W/25KWDW ⁽²⁾	20K0D7W/20KWDW ⁽²⁾	20K0D7W/20KWDW ⁽²⁾
Frequency bands (MHz)	Not yet determined but likely to be 130 - 200 (VHF-150) 360 - 512 (UHF-400) 800 - 941 (UHF-800)	Not yet determined but likely to be 380 - 390/390 - 400 or 410 - 420/420 - 430 or 450 - 460/460 - 470 or 870 - 888/915 - 933	Presently used 1 453 - 1 477/1 501 - 1 525 Future use 905- 915/850 - 860	806 - 821/851-866
Duplex separation (MHz)	varies or none at VHF-150 band 3 and 5 at UHF-400 band 39 and 45 at UHF-800 band	5 - 10 (400 MHz band) 10 - 45 (800/900 MHz band) dependent on system design	48 (55 in the 800 MHz band)	45
RF carrier spacing (kHz)	12.5 for 8K10F1E(C4FM) 6.25 for 5K76G1E(CQPSK)	25	25	25

TABLE 1 (CONT'D)

Core parameters

Parameter	Project 25	TETRA	IDRA	DIMRS
Maximum base station ERP (W) - Peak - Average	500 500	25 25	Not specified typically 40-300	Not specified 250
Nominal mobile stations transmit power (W) Peak/Average - Mobile - Handheld	ranges from 10/10 to 110/110 ranges from 1/1 to 5/5	10/2.5 1/0.25	-/2 Not yet specified	10.4/0.5 3.5/0.17
Cell radius(km) - Handheld/Suburban - Mobile/Rural	7.6 - 35 7.6 35	3.8 - 17.5 38 175	Not yet determined Not yet determined 20-40	5 - 40 (Design dependent) 5 40
Area coverage technique	Cellular channel reuse Simulcast Voting receivers	Cellular channel reuse Quasi synchronous (Simulcast) Time sharing transmission Diversity receivers	Cellular channel reuse Diversity receivers (Base)	Cellular channel reuse Diversity receivers
Access method	FDMA	TDMA	TDMA	TDMA
Traffic channels/RF carrier - Initial - Design capability	1 1	4 8	6 6, 3, 12	6 6, 4, 3, 8, 12, etc.
Transmission rate(kbit/s)	9.6	36	64	64

TABLE 1 (CONT'D)
Core parameters

Parameter	Project 25	TETRA	IDRA	DIMRS
Modulation	QPSK - c family includes C4FM and CQPSK	$\pi/4$ DQPSK	M16QAM (M = 4)	M16QAM (M = 4)
Traffic channel structure - basic rate speech codec - bit rate (kbit/s) - error protection - coding algorithm Alternative rate speech codec - bit rate (kbit/s) - error protection - coding algorithm - circuit mode data - protected (kbit/s) - non-protected (kbit/s) - Packet mode data	4.4 28 IMBE N/A 6.1 9.6 IP - Internet protocol	4.567 2.633 ACELP , rate tbd up to 19.2 up to 28.8 Connection-oriented connectionless	Bit rate with error protection is less than 7.466 Not specified N/A up to 4.8/slot 7.466/slot Connection-oriented (option) connectionless	4.2 3.177 VSELP (6: 1) 8.0 67 VSELP (3: 1) 7.2 None Connection-oriented, connectionless Supports IP and other network protocols
Control channel structure* - common control channel - associated control channel - broadcast control channel * number of channel types	-2 -3 -2	-2 -3 -2	-1 -2 1 (option: 5)	- slot information channel: 1 - primary control channel: 3 - temporary control channel: 1 - dedicated control channel: 1 - associated control channel: 1

TABLE 1 (CONT'D)

Core parameters

Parameter	Project 25	TETRA	IDRA	DIMRS
Delay spread equalization capability(microseconds) ⁽³⁾	Class A - 50 μ s Class Q - 50 μ s	Class A - no equalization Class B - 55.5 μ s Class Q - 111.1 μ s	Class A - no equalization Class B - no equalization Class Q - N/A	Class A - 39.8 μ s without equalizer Class B - 65.5 μ s without equalizer Class Q - N/A
Channel coding	BCH code for network ID Trellis codes for data Golay & Hamming codes for voice Reed-Solomon codes for embedded signals	Convolutional codes with interleaving plus error detection	Multirate trellis coding with interleaving plus error detection and bit prioritization/Convolutional codes with interleaving plus error detection	Multirate trellis coding with interleaving plus error detection and bit prioritization
Encipherment - Security levels - Multi-algorithm - Multikey - Encipherment control - Over the air rekeying	Types 1, 2, 3 and 4 Yes Yes Yes Yes	Air interface is exportable plus authentication. Plus end-to-end encryption user definable up to the highest level of security Yes Yes Yes Yes	Not specified	Allowed for
Handover	Yes	Yes	Option	Yes

TABLE 1 (CONT'D)

Core parameters

Parameter	Project 25	TETRA	IDRA	DIMRS
Inter-system roaming capability	Yes	Yes	Yes	Yes
Design capability for multiple operators (systems) in same area	Yes	Yes	Yes	Yes
Direct mode	Mobile-to-mobile Channel scan ⁽⁴⁾ Repeater Trunking node gateway	Mobile-to-mobile Dual watch ⁽⁵⁾ Repeater Trunking mode gateway	Not yet determined	Allowed for

NOTE 1 – Denotes the emission classifications for **C4FM** and CQPSK modulations. Both alternatives utilize a common receiver and are **thus interoperable**.

NOTE 2 – Denotes the emission classification for base stations/mobiles (handportables).

NOTE 3 – Classes A and B refer to single transmitter operation. Class Q refers to quasi-synchronous (simulcast) operation.

NOTE 4 – Scanning channels for the purpose of alternative channel communication.

NOTE 5 – Allows a terminal using Direct Mode service to monitor the trunking control channel for any incoming **signalling**. **It** also allows a terminal in trunking mode to monitor a direct mode channel.

ANNEX2

General description of the TETRA system

1 Introduction

TETRA is a high-performance mobile radio system which has been developed primarily for professional users such as the emergency services and public transport. The TETRA suite of mobile radio specifications provide a comprehensive radio capability encompassing **trunked**, non-trunked and direct mobile-to-mobile communication with a range of facilities including voice, circuit mode data, short data messages and packet mode services. **TETRA** supports an especially wide range of supplementary services, many of which are exclusive to **TETRA**.

TETRA is designed to operate in the bands below 1 **GHz** and the 25 **kHz** channel structure allows it to fit easily into existing PMR frequency bands.

The specifications cover three distinct telecommunication services corresponding to:

- voice plus data;
- packet data optimized; and
- direct mode.

The packet data optimized (**PDO**) standard is based on the same physical radio platform as the TETRA25 voice plus data standard but implementations are not expected to interoperate at the physical layer. Full interoperability is foreseen at **OSI** Layer 3.

Direct mode (**DM**) provides direct mobile-to-mobile communications when outside the coverage of the network or can be used as a secure communications channel within the network coverage area. It will interoperate with **TETRA25** both at **OSI** Layer 1 and **OSI** Layer 3.

2 Services

2.1 Teleservices

Clear speech or enciphered speech in each of the following:

- individual call (point-to-point);
- group call (point-to-multipoint);
- acknowledged group call;
- broadcast call (point-to-multipoint one way).

2.2 Bearer services

Individual call, group call, acknowledged group call, broadcast call for each of the following:

- Circuit mode unprotected data 7.2, 14.4, 21.6, 28.8 **kbits/sec**.
- Circuit mode protected data (low) 4.8, 9.6, 14.4, 19.2 **kbits/sec**.

- Circuit mode protected data (high) 2.4, 4.8, 7.2, 9.6 kbits/sec.
- Packet connection-oriented data.
- Packet connectionless data.

2.3 **Supplementary services supported**

PMR type supplementary services

Access priority, pre-emptive priority, priority call.

Include call, transfer of control, late entry.

Calls authorized by dispatcher, ambience listening, discreet listening.

Area selection.

Short number addressing.

Talking party identification.

Dynamic group number assignment.

Telephone type supplementary services

List search call.

Cdl forwarding - unconditional/busy/no reply/not reachable.

Call barring - incoming/outgoing calls.

Cdl report.

Cdl waiting.

Call hold.

Calling/connected line identity presentation.

Calling/connected line identify restriction.

Call completion to busy subscriber/on no reply.

Advice of charge.

Call retention.

2.4 **Security aspects**

The TETRA system is designed to ensure high levels of security. The security objectives are listed below:

Correct charging - primarily of interest to commercial systems.

Authenticity - proving the true identity of the communicating parties and of the network.

Confidentiality of communication - protection against unauthorized reading of transmitted information.

Integrity of communication	-	protection against unauthorized modification of transmitted information.
Privacy	-	privacy of people using or operating the network, e.g. personal information, identities, location.
Traffic flow confidentiality	-	to prevent disclosure of information which can be inferred from observing traffic patterns.
Monitoring	-	to permit authorized monitoring of communications, uninhibited by the security mechanisms.
Security management	-	to enable administration of a secure network.

3 Overview of the system

The functional architectures for V+D and PDO are shown in Figures 1 and 2, including their respective standardized interfaces.

4 System specifications

Refer to Table 1.

4.1 Logical channels

The following logical channels are defined:

Common Control Channel (**CCH**) comprising:

Main Control Channel (**MCCH**).

Extended Control Channel (**ECCH**).

These channels deal with control information addressed to or received **from MSs** not involved in a circuit mode call.

- Associated Control Channel (**ACCH**) comprising:

Fast Associated Control Channel (**FACCH**).

Stealing Channel (**STCH**).

Slow Associated Control Channel (**SACCH**).

These channels deal with control information intended for or received **from MSs** involved in a circuit mode call.

Broadcast Common Control Channel (**BCCH**) comprising:

Broadcast Synchronization Channel (**BSCH**).

Broadcast Network Channel (**BNCH**).

These channels carry the **downlink** system broadcast information.

Traffic channels (**TCH**) comprising:

Speech traffic channel (TCWS).

Speech or data **traffic** channels (TCW7.2, TCH/4.8, TCH/2.4).

These channels carry the circuit mode voice or data traffic information.

4.2 TDMA frame structure – Voice and data

The TETRA frame structure, shown in Figure 3, has four slots per TDMA frame. This is further organized as 18 TDMA frames per multiframe of which one frame per multiframe is always used for control signalling. This eighteenth frame is called the control frame and provides the basis of the slow associated control channel (SACCH).

The circuit mode voice or data operation traffic from an 18-frame multiframe length of time is compressed and conveyed within 17 TDMA frames, thus allowing the eighteenth frame to be used to control signalling without interrupting the flow of data. Besides the basic TDMA frame structure described above, there is a hyperframe imposed above the multiframe structure. This is for long repeat frame purposes such as encipherment synchronization. Furthermore, it can be seen that each time-slot is of 5 10 modulation bits in duration.

4.3 Burst structure – PDO

The PDO access schemes are Statistical Multiplexing (STM) for the downlink and Statistical Multiple Access (STMA) for the uplink. The carrier separation is 25 kHz.

The basic radio resources are subbursts, transmitting information at a modulating rate of 36 kbit/s. On the uplink there are four types of subbursts. On the downlink, there are two types of subbursts. Figure 4 describes the PDO up and down burst format.

4.4 Traffic channels

4.4.1 Speech traffic channels

The speech codec, and the associated error correction and detection mechanisms have been defined in the TETRA standard. Speech frames of 30 ms, each comprising 137 bits provide a net bit rate of 4.567 kbit/s. The coding method, ACELP, has been designed to achieve robustness to transmission errors, and to offer a high quality in the presence of background acoustic noise while using a limited bit rate.

Error correction (consisting of a 1/3 rate punctured convolutional code) and interleaving schemes, to selectively protect the most important bits within the speech frame, have been specified. Furthermore, an error detection mechanism has been included and bad frame replacement techniques can be used, in order to minimize the impairment of the speech quality resulting from speech frames not correctly received.

4.4.2 Data traffic channels

Data services of up to 19.2 kbit/s are supported with channel coding and interleaving schemes by using up to four time-slots per TDMA frame.

Unprotected digital bearer services with a bit rate up to 28.8 kbit/s are also supported.

5 Operational characteristics

5.1 Location updating and roaming

The mobile station evaluates the received signal and initiates the location updating procedure when necessary.

A location area is the area in which a mobile terminal can move freely without updating the location information maintained in the network. The paging area is the area in which a mobile is paged.

The Switching and Management Infrastructure (SWMI) will page the mobile terminal in every location area where it is registered.

To facilitate mobility management, a mobile terminal may be temporarily registered in a number of location areas so that a mobile terminal may travel freely between the areas without the need to reregister.

Roaming is possible within a **TETRA** network and between **TETRA** networks.

5.2 Communication protocols

The communication protocols are layered according to the **OSI** model and are specified in the **TETRA** standards.

Layers 1 to 3 are subdivided as shown in Figure 5. The C-plane corresponds to all signalling information, both control and data and also packet mode data **traffic**. U-Plane information corresponds to circuit mode voice or circuit mode data.

The MM, CMCE and PD are defined in Figure 5.

The MLE (Mobile/base Link Control Entity) performs management of the **mobile-to-base/base-to-mobile** connection, mobility within a registration area, identity management, quality of service selection, protocol discrimination (i.e., routing to the higher layer applications).

The LLC (Logical Link Control) layer is responsible for scheduling data transmission and **retransmissions**, segmentation/reassembly, logical link handling.

The MAC (Medium Access Control) layer performs frame synchronization, interleaving/de-interleaving channel coding, random access procedures, fragmentation/reassociation and BER and MER measurements for control purposes.

5.3 Call set-up

5.3.1 Broadcast phase

The base station is continuously transmitting the following control and identification information:

- system identify (e.g. country code, operator code, area code etc.);
- system timing information (e.g. slot synchronization, frame synchronization etc.);
- control channel organization and loading information (e.g. announce slot structure especially for random access);
- requests for or denial of system registrations.

Information (such as paging messages addressed to a particular mobile or group of mobiles) is transmitted on a per call basis.

5.3.2 Set-up

Information is exchanged between the infrastructure and mobile. Five elements of the mobile procedure are:

- wake up (if a battery economy mode);

- presence check on control channel (if required);
- transfer to the traffic channel;
- acknowledgement on traffic channel (if required);
- **traffic** information transfer (voice or data).

Further elements need to be taken into account, especially concerning invoking supplementary services during this phase, conveying this information to the infrastructure, checking the subscriber database to ensure these services have been subscribed to. On **successful** conclusion of this stage, the mobile progresses to the call in progress stage.

5.3.3 Call in progress

Terminals are now concerned primarily to communicate with each other rather than signal to the infrastructure. However, even during the **traffic** phase a substantial amount of control information should be supported to allow "**traffic** channel acknowledgement", caller authentication, notification of call waiting, call hold and transfer to waiting, priority pre-empt, include call (**IC**) and speaker identification during a call.

5.3.4 Call clear down

The mobile relinquishes traffic channel and returns to monitoring the control channel. If the call is on "hold" the system will retain details of the mobile and the call reference for subsequent reconnection. The system may optionally retain line resources. When the call is complete all radio and line resources should be cleared of **traffic** and returned to the resource pool.

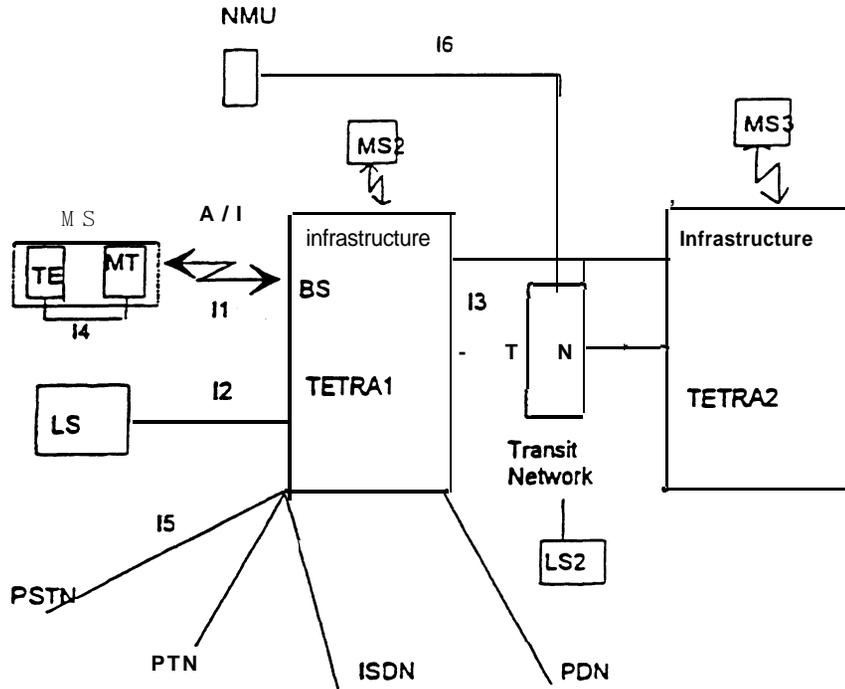
5 4 Connection restoration

A number of network procedures are supported in the **TETRA** specifications to provide continuity of service when a mobile encounters adverse propagation effects, moves between **different** cells or encounters interference. Connection restoration may also be required for **traffic** reasons; to redistribute the load on a particular cell such as during minimum mode operation; to allow the frequency allocations at a particular cell to be reorganized, or for maintenance or equipment fault reasons.

The responsibility for initiating the connection restoration procedures can rest with the mobile station or with the base station, depending on the reason for restoration.

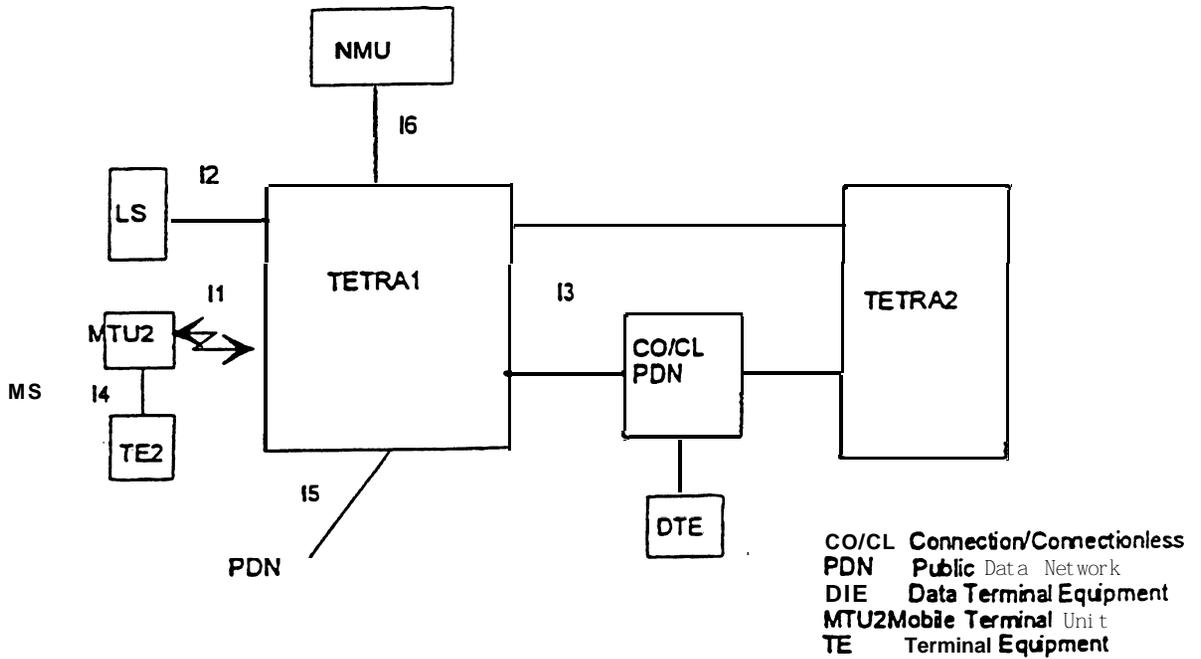
The mobile station is responsible for monitoring the quality of the **downlink** transmissions and may request an alternative channel on the same serving cell if interference is encountered or may request service on another cell if the received signal strength drops below a predefined level. The TETRA air interface protocol provides a range of restoration procedures (of **different** quality) which a network operator may wish to install, and to which users may choose to subscribe. These range from a totally unprepared restoration taking several seconds during which time the connection is broken, to seamless **handover** where the break in **service** is imperceptible to the user.

The base station may choose to move the MS to another channel on the same servicing cell if interference on the **uplink** is encountered. The BS may wish to hand-off the call to an adjacent cell if the loading becomes too high on a particular site (load shedding). This would be performed by altering the acquisition and relinquishing criteria defined in the broadcast (**BCCH**).



MS Mobile Station
 MT Mobile Terminal
 LS Line station
 NMU Network Management Unit

FIGURE 1
 TETRA voice plus data



CO/CL Connection/Connectionless
 PDN Public Data Network
 DTE Data Terminal Equipment
 MTU2 Mobile Terminal Unit
 TE Terminal Equipment

FIGURE 2

TETRA packet data optimized

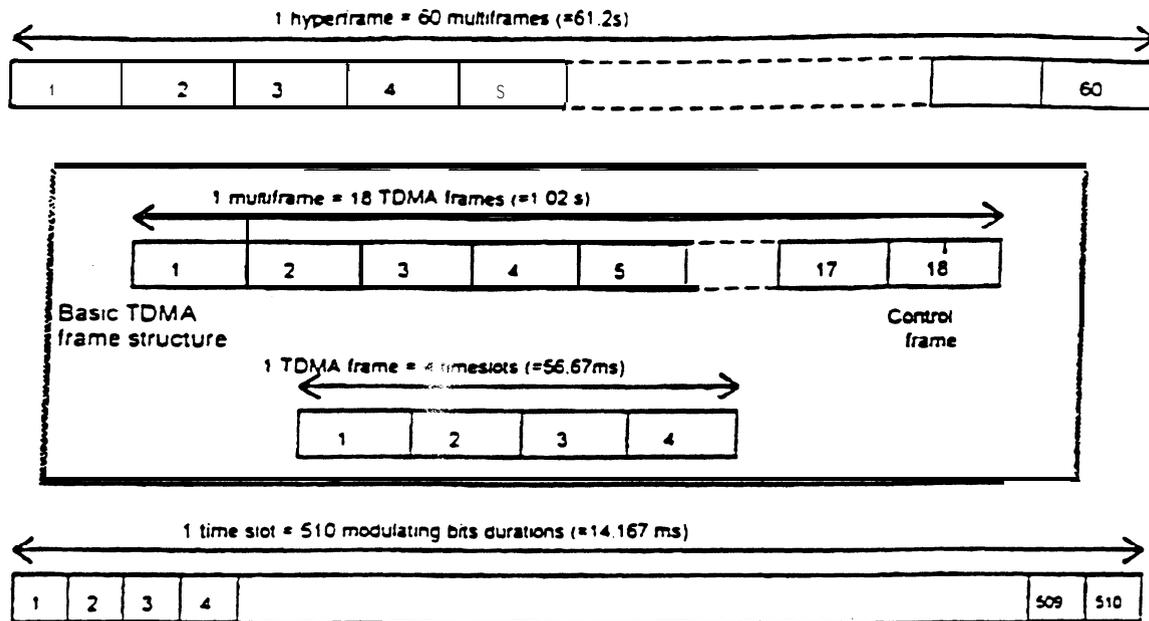


FIGURE 3
TETRA TDMA frame structure

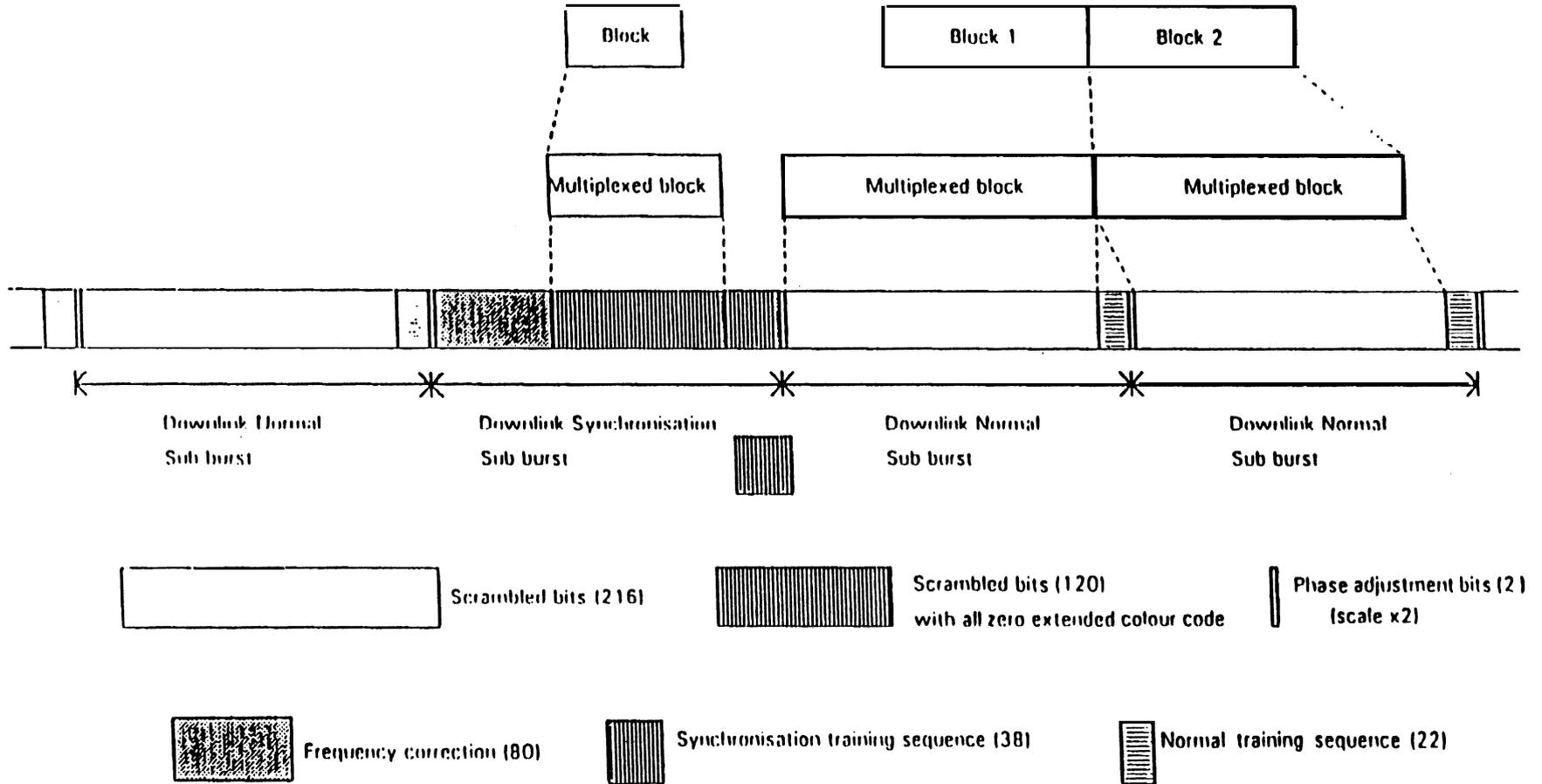


FIGURE 4A

PDO downlink burst structure

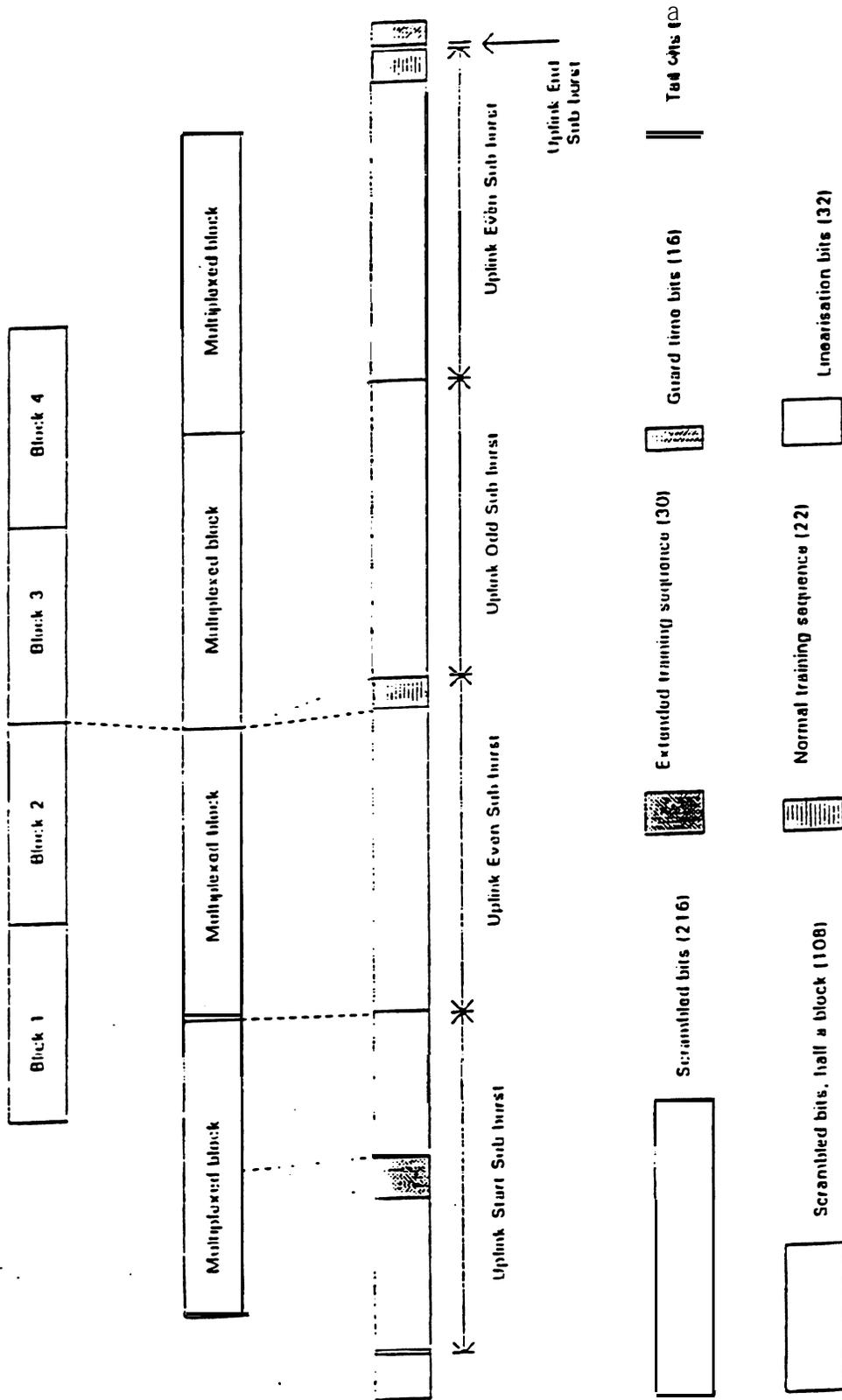


FIGURE 4B
PDO uplink burst structure

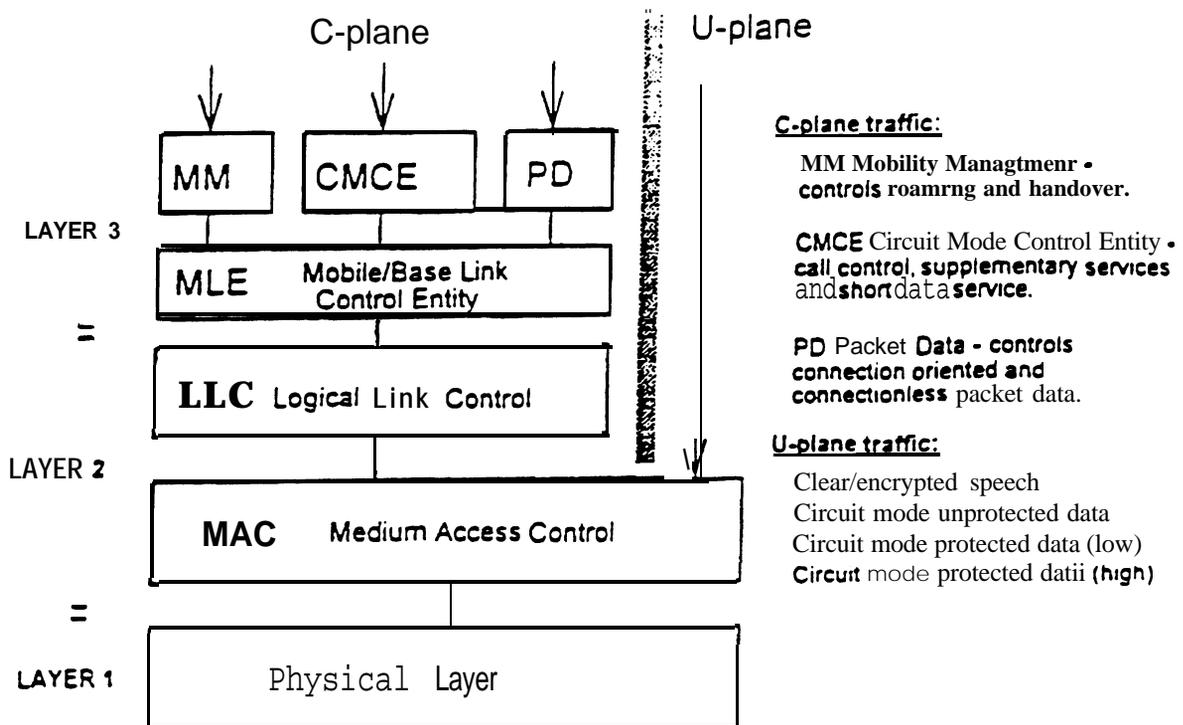


FIGURE 5
Mobile/base station protocol stack

REFERENCES

- ETSI [January, 1994] ETR 086-1 - Trans European **Trunked** Radio (**TETRA**) system - Technical requirements specification, Part 1 - Voice plus Data (V+D) systems.
- ETSI [January, 1994] ETR 086-2 - Trans European **Trunked** Radio (**TETRA**) system - Technical requirements specification, Part 2 - Packet Data Optimized (**PDO**) systems.
- ETSI [January, 1994] ETR 086-3 - Trans European **Trunked** Radio (**TETRA**) system - Technical requirements specification, Part 3 - Security aspects.
- ETSI [November, 1994] prETS 300 392-1 - Trans European **Trunked** Radio (**TETRA**) - Voice plus Data (V+D), Part 1 - General network design.
- ETSI [November, 1994] prETS 300 392-2 - Trans European **Trunked** Radio (**TETRA**) - Voice plus Data (V+D), Part 2 - Air Interface.
- ETSI [November, 1994] prETS 300 392-10 - Trans European **Trunked** Radio (**TETRA**) - Voice plus Data (V+D), Part 10 - Supplementary services stage 1.

ETSI [November, 1994] prETS 300 393-1 – Trans European **Trunked** Radio (TETRA) – Packet Data Optimized (PDO), Part 1 – General network design.

ETSI [November, 1994] prETS 300 393-2 – Trans European **Trunked** Radio (TETRA) – Packet Data Optimized (PDO), Part 2 – Air Interface.

ETSI [November, 1994] prETS 300 394-I – Trans European **Trunked** Radio (TETRA) – Conformance testing specification, Part 1 – Radio.

ANNEX3

General description of the Project 25 system

1 Services supported

Services will be available on Project 25 systems in accordance with system type and other specifications within this annex. Where a service is mandatory for a Project 25 system type, such a system must provide that service. Where a service is a standard option, and a Project 25 system provides that service, it shall be provided in compliance to the standard. Technological limitations may preclude some systems from supporting certain **services**.

1.1 Types of systems

Two types of systems are defined: Non-trunked (conventional) and **trunked**. All Project 25 **trunked** radios shall be capable of operation in both types of systems.

1.1.1 Non-trunked (conventional)

Non-trunked (conventional) systems possess no centralized management of subscriber operation or capability. All aspects of system operation are under control of the system users. Operating modes within non-trunked systems include both direct (i.e., radio-to-radio) and repeated (i.e., through an RF repeater) operation.

1.1.2 Trunked

Trunked systems provide for management of virtually all aspects of radio system operation, including channel access and call routing. Most aspects of system operation are under automatic control, relieving system users of the need to directly control the operation of system elements.

1.2 Availability

The following tables of telecommunications services show service availability by system type. The services are further denoted as either mandatory or as a standard option, by system type.

Telecommunications services		
Bearer services	Non-trunked	Tt-unked
Circuit Switched Unreliable Data	Standard Option	Standard Option
Circuit Switched Reliable Data	Standard Option	Standard Option
Packet Switched Confirmed Delivery Data	Standard Option	Standard Option
Packet Switched Unconfirmed Delivery Data	Standard Option	Standard Option

Teleservices	Non-trunked	Trunked
Broadcast Voice Call	Not Available	Mandatory
Unaddressed Voice Call	Mandatory	Not Available
Group Voice Call	Standard Option	Mandatory
Individual Voice Call	Standard Option	Mandatory
Circuit Switched Data Network Access	Standard Option	Standard Option
Packet Switched Data Network Access	Standard Option	Standard Option
Pre-programmed Data Messaging	Standard Option	Standard Option
Supplementary services	Non-trunked	Trunked
Encipherment	Standard Option	Standard Option
Priority Call	Not Available	Standard Option
Pre-emptive Priority Call	Not Available	Standard Option
Call Interrupt	Standard Option	Standard Option
Voice Telephone Interconnect	Standard Option	Standard Option
Discreet Listening	Standard Option	Standard Option
Radio Unit Monitoring	Standard Option	Standard Option
Talking Party Identification	Standard Option	Standard Option
Call Alerting	Standard Option	Standard Option
Services to the subscriber	Non-trunked	Trunked
Intra-system Roaming	Standard Option	Standard Option
Inter-system Roaming	Standard Option	Standard Option
Call Restriction	Not Available	Standard Option
Affiliation	Not Available	Standard Option
Call Routing	Not Available	Standard Option
Encipherment Update	Standard Option	Standard Option

2 Functional groups

2.1 MES (Mobile End System)

In the mobile end system functional group, the term “Mobile” is used as in Land Mobile Radio, which includes **all** mobile radios, portable radios, and fixed remote radios. The **MES** functions include the voice and/or data user interface built into a radio.

2.2 MDP (Mobile Data Peripheral)

The mobile data peripheral **functional** group includes all mobile, portable, and fixed remote data peripherals. The MDP functions include the data user interface of any data peripheral attached to a radio.

2.3 MRC (Mobile Router and Control)

The mobile router and control **functional** group includes **functions** of voice and/or data routing, as well as control of the Mobile Radio, MR.

2.4 MR (Mobile Radio)

The mobile radio functional group includes functions of transmission and reception of all RF signals.

2.5 BR (Base Radio)

The base radio **functional** group includes only the functions of modulation and demodulation of the Radio Frequency energy. Elements within the BR include the Power Amplifier (PA), RF front-end, IF selectivity, and end-IF detection device.

2.6 BA (Base Audio)

The base radio audio **functional** group includes the **functions** of frequency/level shaping and signal processing associated with transmitted signals and received signals coupled to the BR. The interface to the BR and BC are manufacturer-specific, and may be at any level or **frequency**.

2.7 BC (Base Control)

The base radio control functional group includes the automated control functions of an individual radio.

2.8 RFC (Radio Frequency Control)

The radio **frequency** control functional group includes all logic for translating user command signalling and control into base radio command signalling and control for one or more base radios. The RFC functions further include **all** logic for generating command signalling and control to a **RFS** functional group, if present.

2.9 RFS (Radio Frequency Switch)

The radio **frequency** switch functional group includes all switching for establishing interconnection paths between gateways and base radios, as directed via command and control signalling **from** an **RFC**.

2.10 CON (Console)

The **CON**sole functional group includes all end system functionality for dispatcher(s); including a dispatcher's Man Machine Interface, control and audio functions.

2.11 MSC (Mobile Service Switching Centre)

The mobile service switching centre is a switching centre for services between radio subnetworks. The MSC is the combination of the RFC and RFS functional groups.

2.12 HLR (Home Location Register)

The home location register is a dynamic database service which tracks the mobility of radios associated with a particular radio subnetwork, that roam to other radio subnetworks.

2.13 VLR (Visitor Location Register)

The visitor location register is a dynamic database service which tracks the mobility of roaming radios which enter a radio subnetwork, but that are associated with a different radio subnetwork.

2.14 RFG (Radio Frequency Gateway)

The radio frequency gateway functional group functions include direct interface with any/all end systems with the exception of the CONSOLE (where the end system may be an RFG into another radio subsystem), and any translation of command signalling between the end system/user and the RFC. The RFG functions further include any translation of end system/user payload between the user and the RFS. The RFG also includes interface between VLRs, HLRs, and MSCs between RF subsystems.

3 Signalling description

3.1 Data units

Information is transmitted over the air, using the Common Air Interface (CAI), in data units. There are five types of data units defined for voice channel operation, one type of data unit for data packets, and one type of data unit for control functions.

3.1.1 Voice data units

Voice information is transferred in a sequence of Logical Link Data Units (LDUs), each convey 180 ms of voice information. There are two kinds of LDUs, denoted as LDU1 and LDU2. Each LDU conveys additional embedded information, which includes a link control word, an encipherment synchronization word, and low-speed data. LDU1 conveys the link control word. LDU2 conveys the encipherment synchronization word. Both LDU1 and LDU2 convey low-speed data.

Voice information in the LDUs is conveyed as nine frames of vocoder information, with each frame containing 20 ms of digitized voice information.

The LDUs are paired into superframes of 360 ms. Each superframe has an LDU1 and an LDU2. The last superframe of a voice transmission may terminate after LDU1, if the transmission ends before the LDU2 portion of the superframe has begun. Since LDU2 is present in each superframe (except possibly the last one), it is possible for the transmission recipient to synchronize decipherment in the middle of the transmission, and begin receiving a voice transmission on a superframe boundary.

Voice transmission begins with a header data unit, which conveys the synchronization of the encipherment algorithm. This allows voice information in LDU1 of the first **superframe** to be deciphered. The header data unit takes 82.5 ms to transmit.

Voice transmission terminates with one of two types of terminator data units. A simple terminator is a short word, 15 ms in duration, signifying the end of a transmission. A terminator with link control conveys a link control word for supervisory functions when terminating a transmission. A terminator with link control is 45 ms in duration.

3.1.2 Packet data unit

A packet data unit conveys general purpose data information. A packet **data** unit is split into blocks of information. The first block conveys addressing and service information, and is designated as a header block. Subsequent blocks are designated as data blocks. The length of the data packet is contained in the header block.

Each block is protected with either a rate $1/2$ trellis code, or a rate $3/4$ trellis code. The rate $1/2$ trellis code encodes 12 octets of information into exactly 196 bits. The rate $3/4$ trellis code encodes 18 octets of information into exactly 196 bits. A header block always uses the rate $1/2$ trellis code. Data blocks use a rate $1/2$ trellis code for unconfirmed delivery data packets, and a rate $3/4$ trellis code for confirmed delivery data packets. The type of data packet (confirmed or unconfirmed) is indicated in the header block.

3.1.3 Control data unit

A special short data packet is defined for control functions. It consists of a single block protected with the rate $1/2$ trellis code defined for the packet data unit. It requires 37.5 ms of air time to transmit.

3.2 Media access control

Data units are transmitted over the air preceded by a short burst of frame synchronization and network identity. The **frame** synchronization is exactly 48 bits, 5 ms in duration. The network identity is a 64 bit codeword. These allow the recipient of the transmission to determine the beginning of the message, and to distinguish **traffic** on the proper radio system **from** interference or co-channel **traffic** on nearby systems. The network identifier also contains a data unit identifier which identifies among the seven possible data units.

Channel access is controlled with status symbols which are periodically interleaved throughout transmissions. Each status symbol is two bits, transmitted after every 70 bits within a data unit. This spaces the status symbols exactly 7.5 ms apart. The 7.5 ms interval is designated as a microslot time interval. If a data unit happens to end before a microslot boundary, then additional null bits are inserted to pad the transmission to the next microslot boundary.

An RF subsystem indicates activity on an inbound channel by setting the status symbols on the corresponding outbound channel to a "busy" state. Radios wishing to access the inbound channel are inhibited **from** transmission when the status symbols indicate "**busy**". When status symbols indicate "idle", they **may** transmit. A third state, indicating "unknown" is used for slotting status symbols.

4 Operational characteristics

Operation over the CAI is dependent on mode, i.e., whether the message is voice or data, and whether the system is **trunked** or non-trunked. In general, **trunked** operation requires radios to request service on a control channel using a control data unit. The RF subsystem then assigns the

radio to a working channel for **further** operations. **After** the operations are complete on the working channel, the call is cleared for assignment of the channel to other calls. Operation in a **non-trunked** system does not have the service request phase and the call clearing phase.

4.1 Voice transmit operation

Operation of a transmitter for voice messages has three main cases, with several options and variations of each case. The three main cases consist of Routine Group Calls, Emergency Group Calls, and Individual Calls.

4.1.1 Controls. A transmitter may have several controls which **affect** transmit operations. Controls sufficient for a radio to support all of the call types are defined below. These controls are:

PTT switch – A Push-To-Talk switch is activated when an operator wishes to transmit, and released when a transmission is finished.

Channel selector – The channel selector is a switch or control that allows the operator of a radio to select a radio's operational parameters. The operational parameters that can be selected include the following items:

- 1) Transmit **frequency**.
- 2) Transmit Network Access Code.
- 3) Talk Group.
- 4) Other parameters for setting the vocoder and encipherment **functions**. For example, the enciphering key variable may be selected.

Emergency switch – The **Emergency switch is asserted by a radio operator for emergency calling. Once this switch is asserted, the emergency condition remains asserted until it is cleared by a different means, e.g. turning the radio off.**

Numeric keypad/display – This allows a radio operator to set numeric values. This is most useful for individual calls.

4.1.2 Call types. The different types of calls are defined as follows:

Routine group call – This is a transmission that is intended for a group of users in a radio system. Typically, it is the type of call that is made most **often**. These calls are typically made when the PTT switch is asserted.

Emergency group call – This is a transmission that is intended for a group of users in a radio system, during an emergency condition. The definition of an emergency condition depends on a system's operators, but it typically signifies an exceptional condition with more urgency. These calls are typically made **after** the emergency switch is asserted.

Individual call – This is a transmission which is addressed to a specific individual radio. The individual radio's address to which the call is directed is called the destination address. These calls are typically made after the destination address is entered into the radio.

4.1.3 Procedures. The procedures for each of **these calls** in the transmitter are based on the procedure for the Routine Group Call. Consequently, that type of call is described first, and then the other types of calls are described.

Routine group call procedure

- 1) PTT. The radio operator asserts the PTT switch.
- 2) Pre-transmit. The radio selects the channel parameters as determined by the channel selector switch. The radio may check the status symbols, if present, to determine if the channel is busy or idle. If busy, it may optionally hold off the activation of the transmitter until the channel is idle. If the status symbols are not checked, or if the channel is idle, then the radio simply keys the transmitter on the transmit frequency. The radio also activates the voice encoder. The radio also activates the encipherment function, if present.
- 3) Header Data Unit. The radio transmits the Header Data Unit with the following selected information fields:
 - Network Access Code as determined by the channel selector switch.
 - Manufacturer's ID.
 - Message Indicator, Algorithm ID, and Key ID are determined by the encipherment function.
 - Talk Group/Individual ID is determined by the channel selector switch, as appropriate.
- 4) Format selection. The following recurrent voice message parameters are set:
 - Network Access Code as determined by the channel selector switch.
 - Manufacturer's ID.
 - Emergency bit is set to indicate routine operation.
 - Talk Group/Individual ID is determined by the channel selector switch, as appropriate.
 - Source ID is set to the unit ID of the radio.
 - Message Indicator, Algorithm ID, and Key ID are determined by the encipherment function.
- 5) Transmission. The voice link data units, **LDU1** and **LDU2**, are sent with the message parameters set above in step 4. The information contents of the Link Control word is enciphered if specified by the encipherment function. Link Control shall only be enciphered if the voice frames are also enciphered. Transmission is sustained until the PTT switch is released.
- 6) End of Transmission. Transmission terminates when the PTT switch is released, or some other event forces a dekey, and the transmission has reached the end of an LDU. The radio terminates the voice encoder. Then the radio sends a terminator data unit. A radio always sends the simple terminator, consisting of frame synchronization and the Network ID word. **After** termination, the radio notifies the encipherment function to terminate, as defined in the encipherment protocol.
- 7) Dekey. The radio ceases transmission.

Emergency group call procedure

- 1) Emergency stitch. The radio operator asserts the emergency switch. This sets the emergency condition until it is cleared by some other action, e.g., turning the radio off.
- 2) Group Calls. Activation of the PTT switch now initiates calls that are very much like the Routine Group Call described above. The only difference in procedure is that the Emergency bit is asserted to indicate an emergency condition. Group calls can be made repeatedly, and each group call will indicate the emergency condition.

- 3) Emergency termination. The emergency condition is cleared by turning the radio off. When the radio is turned on, the emergency condition is cleared and Routine Group Calls are made **after** PTT assertion. In addition to this method, other methods of termination may also be available.

Individual call procedure

- 1) Select Called Party. The unit ID of the individual radio to be called can be entered into the radio via a keypad or by some other means. This **becomes the destination ID** of the call.
- 2) Make the call. The procedure for group calls is followed, with the following exceptions:
 - i) The Talk Group ID in the Header Data Unit is cleared to the null talk group (0000).
 - ii) The Link Control field is formatted with the individual call format, containing the source ID and destination ID of the **call**.

4.2 Voice receive operation

The operation of a receiver for voice messages consists of three main cases, with variations that depend on the transmitter's operation. The three main cases are called Squelch conditions in this document. They are: Monitor, Normal Squelch and Selective Squelch.

As in the case of the transmitter, receiver operation will be **affected** by the channel selector switch. This switch can select:

- 1) Receive **frequency**.
- 2) Receiver **Network** Access Code.
- 3) Talk Group.
- 4) Other parameters for setting the vocoder and encipherment functions. The encipherment function is particularly significant to the receiver.

An additional radio control which can affect a receiver is the **monitor switch**. This switch **allows** the operator of a radio to disable any selective squelch of **the** receiver so that an operator can hear any sign of voice **activity**. This **can be useful for avoiding collisions on non-trunked** channels between voice users.

The types of squelch operation described are defined as follows:

Monitor – This **enables** the receiver to unmute on any recognizable voice signal. Selective muting based on the network **access code, talk group ID, or unit address is not performed**. This is analogous to monitor mode in **analogue** receivers. This is normally activated with a monitor switch.

Normal squelch – This enables **the receiver to unmute on any voice signal** which has the correct network access code. Voice messages **from** co-channel users which **are using different** network access codes **will be muted**.

Selective squelch – This mutes all voice **traffic** except that which is **explicitly** addressed to the radio. Messages which contain **the talk group or unit address of the receiver, as well as the network access code, will be received**.

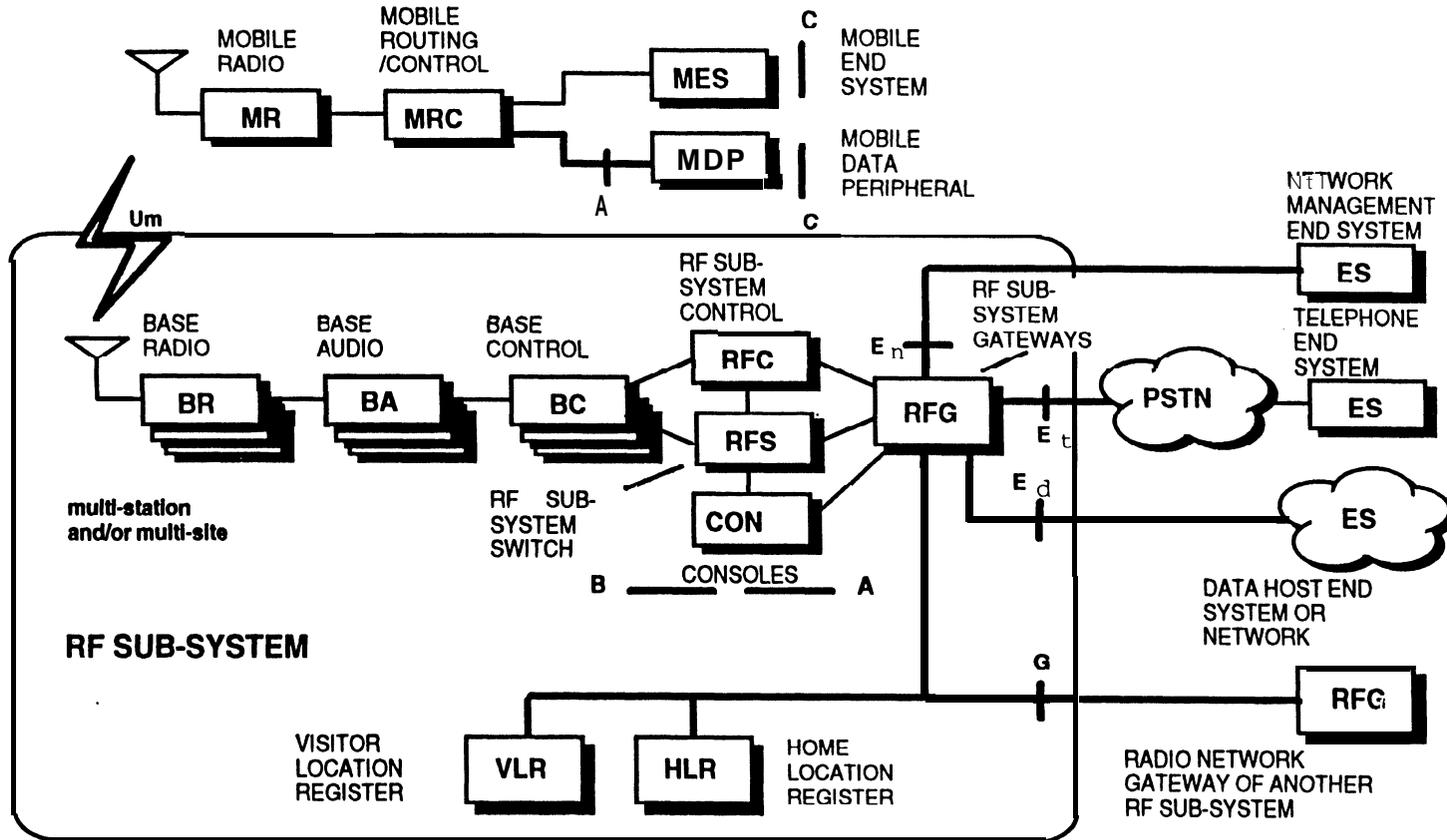


FIGURE 6A

Project 25 Repeater (example) reference configuration

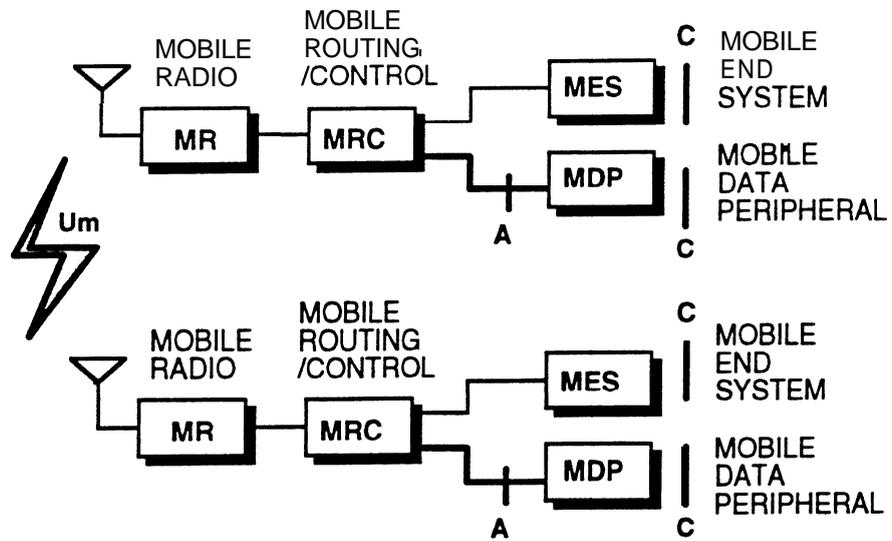


FIGURE 6B
Project 25 Non-repeater reference configuration

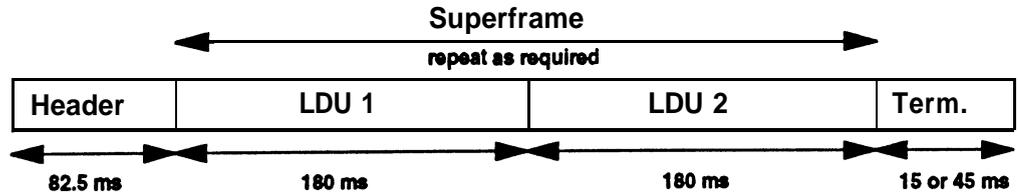


FIGURE 7A
Project 25 voice structure

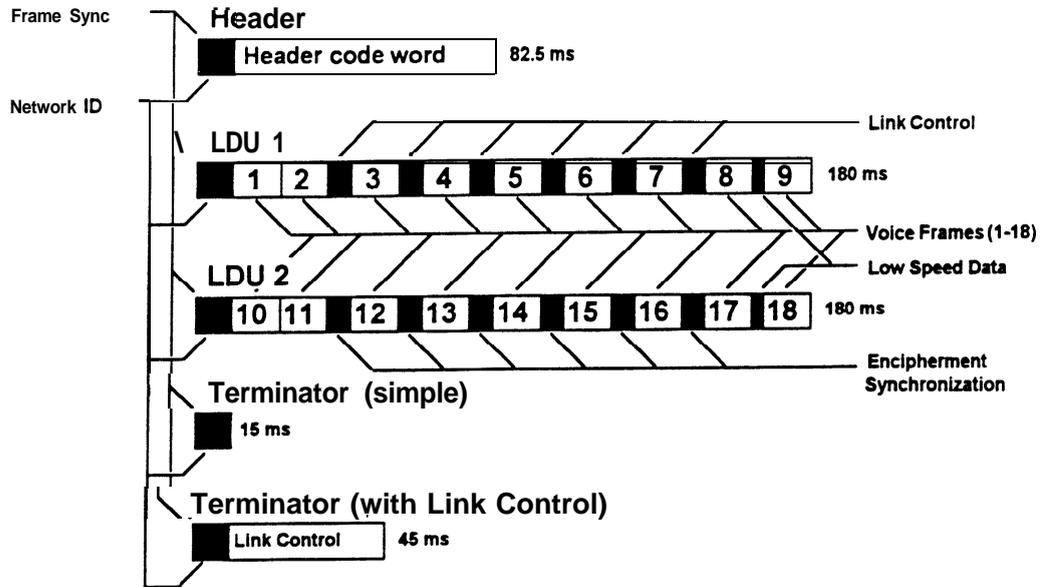


FIGURE 7B
Project 25 voice data unit structure

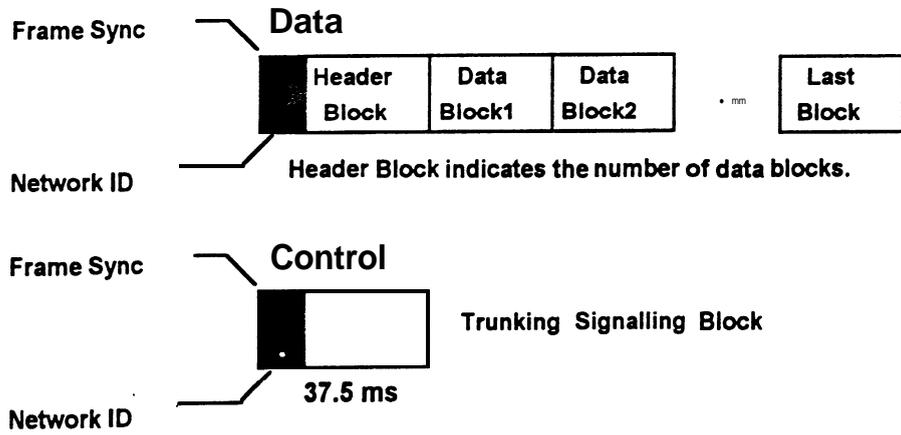


FIGURE 8
Project 25 data and control signal structure

REFERENCES

- [1] TIA/EIA TSB 102 – Project 25 System and Standard Definition.
- [2] TIA/EIA TSB 102.BAAA – Common Air Interface.
- [3] TIA/EIA TSB102.BAAB – CAI Conformance Testing.
- [4] TIA/EIA TSB102.BAAC – CAI Reserved Values.
- [5] TIA/EIA TSB102.BAAD – CAI Operational Description for Conventional (non-trunked) Channels.
- [6] TIA/EIA IS 102.BABA – Vocoder Description.
- [7] TIA/EIA TSB 102. CAAA – Transceiver Measurements and Methods.
- [8] TIA/EIA TSB 102. CAAB – Tranceiver Performance Recommendations.
- [9] TIA/EIA IS 102.AAAA – DES Encryption Protocol*.
- [10] TIA/EIA IS102.BABB – Vocoder Mean Opinion Score Test.
- [11] TIA/EIA IS 102.BABC - Vocoder Reference Test.
- [12] TIA/EIA TSB 102.BABD – Vocoder Selection Process.
- [13] TIA/EIA TSB 102.AABA – Trunking Overview.
- [14] Draft TIA/EIA TSB 102.AABB – Trunking Control Channel Formats.
- [15] Draft TIA/EIA TSB 102.AABC – Trunking Control Channel Messages.
- [16] TIA/EIA TSB 102.BAEA – Data Overview.
- [17] TIA/EIA TSB 102.BAEB – Packet Data Specification.
- [18] TIA/EIA TSB 102.BAEC – Circuit Data Specification.
- [19] TIA/EIA TSB 102.BAFA – Network Management Interface Definition.
- [20] TIA/EIA IS 102.AAAC – DES Encryption Conformance*.
- [21] TIA/EIA TSB 102.AACA – Key Management and OTAR Protocol*.
- [22] TIA/EIA TSB 102.BAEE – Radio Control Protocol Specification.
- [23] TIA/EIA TSB 102.AAAB - Security Services Overview*.
- [24] Draft TIA/EIA IS 102.BADA – Telephone Interconnect Requirements and definitions (voice service).
- [25] Draft TIA/EIA TSB102.AABF – Link Control Words.
- [26] Draft TIA/EIA TSB 102.AABG - Conventional Control Messages.
- [27] Draft TIA/EIA TSB 102.AABD – Trunking Procedures.
- [28] TIA/EIA TSB 102.AACB – OTAR Operational Description*.

• These documents are referenced for completeness only. The selection of encipherment algorithm should remain a national option.

ANNEX 4

General description of the IDRA System

1 Introduction

The **IDRA** System has been developed for use mainly in business-oriented mobile communications applications. Both voice and data communications in the **IDRA** System offer inter-mobile communications in a single cell and inter-mobile communications between cells, as well as communications between a PSTN user and a mobile subscriber to the **IDRA**. The **IDRA** System satisfies the following three fundamental specifications:

- 1) Voice only.
- 2) Voice and data (Circuit mode data, short message mode data, and packet mode data).
- 3) Data only (Circuit mode data, short message mode data, and packet mode data).

2 Services

2.1 Teleservices

Clear speech or enciphered speech in each of the following:

- Individual call (point-to-point).
- Group call (point-to-multipoint).
- Broadcast call (point-to-multipoint, one way).
- Full-duplex interconnect call.
- Full-duplex dispatch call (option).

2.2 Bearer services

Individual call, group call, and broadcast call for each of the following:

- Circuit mode protected data 3.044 and **4.8 kbit/slot**.
- Circuit mode non-protected data 7.466 **kbit/slot**.
- Packet connection-less data.
- Packet connection-oriented (option).

2.3 Supplementary services

Telephone type supplementary services:

- Call completion to busy/no-reply subscriber.
- Call barring incoming/outgoing call.
- Calling line identity presentation.
- Calling line identity restriction.
- Voice operation guide (option).
- List search call (option).

- Call waiting.
- Advice of charge (option).
- Short message service (option).
- Call traffic monitor.
- Call monitor with late entry.
- Priority call.
- Conference call (option).
- Area selection.
- Subgrouping Call.

Network access supplementary services:

- Multiple-zone access.
- PSTN/PSDN** access.

2.4 Security aspects

Special security aspects are not specified, but the system provides a high level of security with authentication and identification.

Authentication

During power up, mobile origination, mobile termination, location updating, supplementary service, and/or short message service.

Identification

By individual identification and/or temporary identification.

3 Overview of the system

The network approach showing the major architectural components of the system is shown in Figure 9.

4 System specifications

Refer to Table 1.

4.1 Logical channels

The following logical channels are defined:

- Broadcast Control Channel (**BCCH**).
- Common Control Channel (**CCCH**).
- Associated Control Channel (**ACCH**).
- Traffic Channel (TCH)**.
- Packet Channel (**PCH**).
- Slot Information Channel (**SICH**).

- Random Access Channel (RACH).
- Temporary Control Channel (TCCH).
- Dedicated Control Channel (DCCH).
- Radio Control Channel (RCCH).

4.2 TDMA frame structure

The basic **frame** is prescribed at six slots. The corresponding outbound and inbound frames make a pair. The frame offset, the outbound frame delay relative to the inbound **frame**, is 70.955 ms.

Conversely, the inbound frame delay, relative to the outbound frame (referred to as transmit-receive offset) can be calculated by the formula, (frame length)-(frame offset). Accordingly, transmit-receive offset is 19.045 ms. Figure 10 shows the general frame structure of the **IDRA** System.

4.3 Traffic channels

4.3.1 Speech traffic channels

The speech codec for voice communication services, including error correction and error detection mechanisms, has not been defined in the **ARIB** standard. However, the **ARIB** defined the **frame** structure of the voice channel to have 90 ms speech frames comprised of a total of 672 bits, including the additional bits for error correction. The system operator is **free** to choose the codec bit rate and error control scheme up to a total of 7.467 **kbit/s**.

4.3.2 Data traffic channels

A circuit data protocol is available for circuit data applications. The circuit-switched data protocol offers a full-duplex packet stream.

Packet data transmission is a planned feature of the **IDRA**. Airtime for packet transmission is dynamically allocated to the user devices according to their instantaneous communication need. **The** packet data protocol is planned to allow an auto-bauding capability so that different net burst transfer rates will be available to the user.

5 Operational characteristics

5.1 Location updating and roaming

5.1.1 Roaming

Roaming, which enables automatic switching of the infrastructure when a Mobile **Station (MS)** moves into a different location area, is possible between **IDRA** Systems.

5.1.2 Location updating (option)

The **IDRA** System tracks an individual **MS's** location to allow the **MS** to move **freely throughout** the system and receive or originate calls. Location areas, which are composed of one or more sites, are used to define geographical areas in the system. The mobile terminal must **report its position each** time it moves between location areas.

5.1.3 Handover (option)

The IDRA supports **handover** between zones and between systems. **Handover** allows for maintaining the link quality for user connections, minimizing interference, and managing traffic distributions.

5.2 Communication protocols

The communication protocols of the IDRA are layered according to the OSI model as shown in Figure 11. However, it does not strictly match the standard model because press-to-talk communication is the basic operation, so a protocol providing a faster response is required.

The layers are subdivided as shown below:

- Layer 1: This layer specifies the physical structure of the channel. (Basic slot format, **subslot** format, etc.).
- Layer 2: This layer specifies communication control between the MS and the **infrastructure** such as random access control, polling control and time alignment control.
- Layer 3: This layer performs as a network layer and is divided into the following three sublayers:
 - Connection Management (CM)
Call set-up, Call management/control, Call clear down, etc.
 - Mobility Management (**MM**, option)
Location Registration, Authentication, etc.
 - Radio Resource Management (**RR**, option)
Cell selection, channel assignment, handover, etc.

5.3 Call set-up

5.3.1 Broadcast phase

The base station is continuously transmitting the following control and identification information:

- Control channel information (e.g. physical **structures** of control channel for system identification and call set-up).
- System information (e.g. types of communication services and protocols which IDRA can provide).
- Restriction information (e.g. types of communication **services** and protocols which IDRA now restricts).
- System structure information (e.g. location area and target cell information; optional).

5.3.2 Set-up

Necessary information is exchanged between the infrastructure and MS. The elements of the mobile procedures are:

- Wake up (if in battery saving mode).
Receive the control channel.
- Exchange the necessary information for call set-up.
- Receive the **traffic** channel.

- Transfer traffic information (voice or data).
- Registration and authentication (option).

5.3.3 Call clear down

The following six procedures are available for call clear down:

- The MS and the infrastructure clear down when the time limit for communication is reached.
- The infrastructure clears down when the time limit for no response is reached.
- The infrastructure clears down when the time limit for no communication is reached.

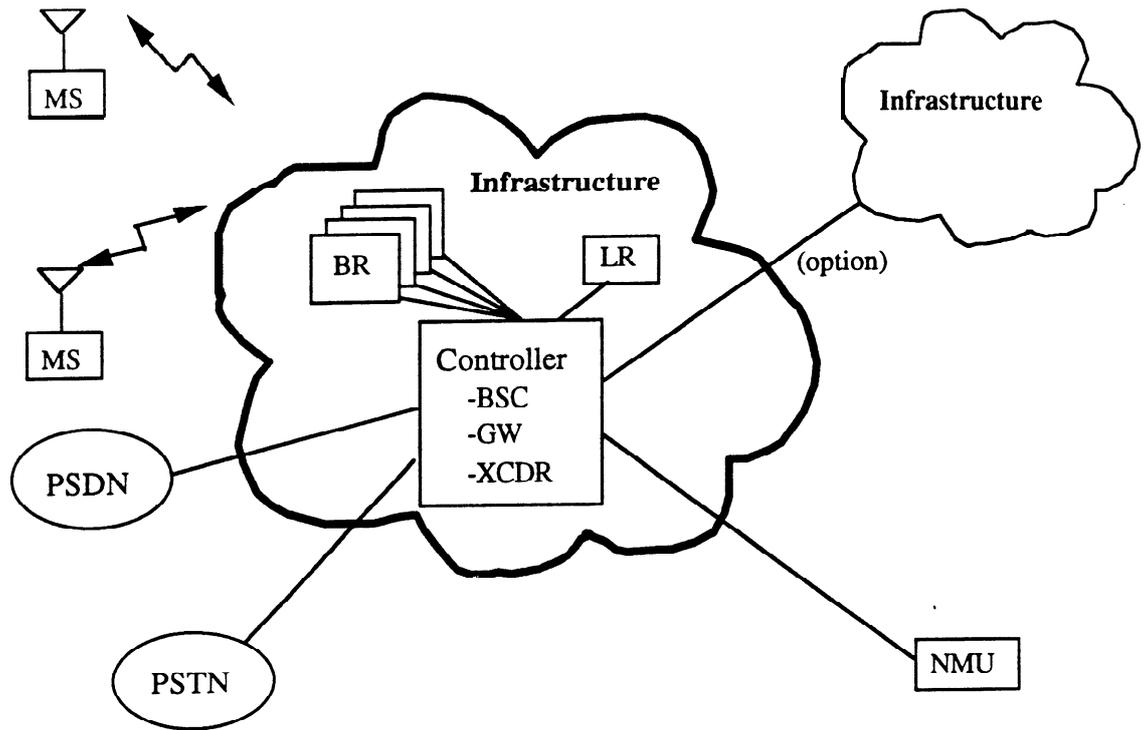
The MS clears down on detection of poor **traffic** conditions.

Clear down occurs on demand of disconnection from a mobile terminal, a fixed terminal, or a telephone on the PSTN.

- Disconnection from the base.

5.4 Connection restoration (option)

- MS knows where to monitor from information on Broadcast Control Channel.
- MS continuously measures parameters during call:
 - $C/(I + N)$.
 - RSSI.
 - Primary serving channel.
- When MS detects trouble on primary server:
 - MS sends in parameter samples.
 - Base evaluates potential servers.
 - Base assigns new server.
 - MS switches to new server.



- BR : Base Radio
- BSC : Base Site Controller
- LR : Location Register
- GW : Gate Way
- XCDR : Speech Transcoder
- NMU : Network Management Unit
- MS : Mobile Station
- PSTN : Public Switched Telephone Network
- PSDN : Public Switched Data Network

FIGURE 9
IDRA Network approach

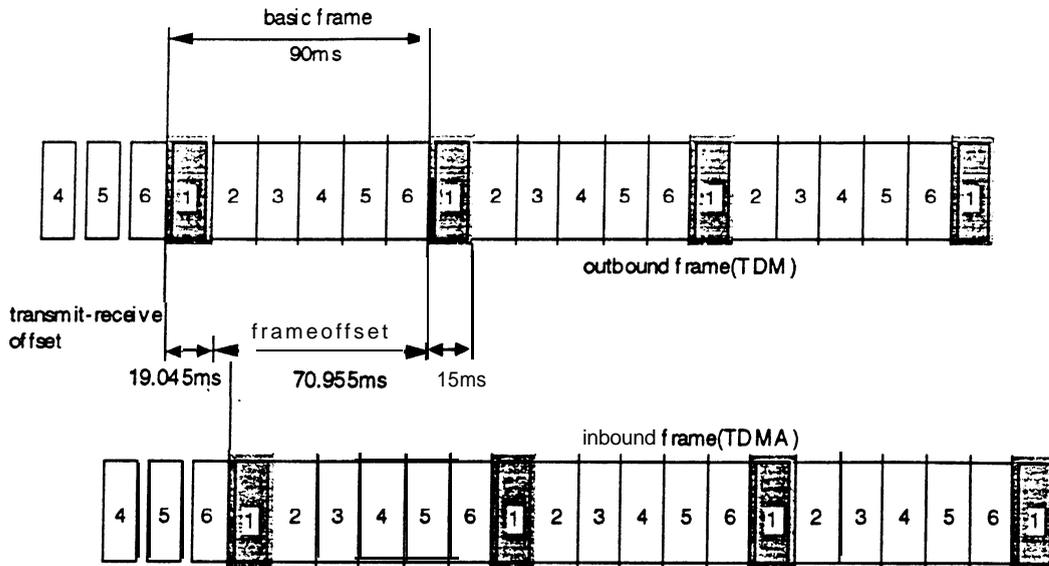


FIGURE 10
IDRA TDMA Frame structure

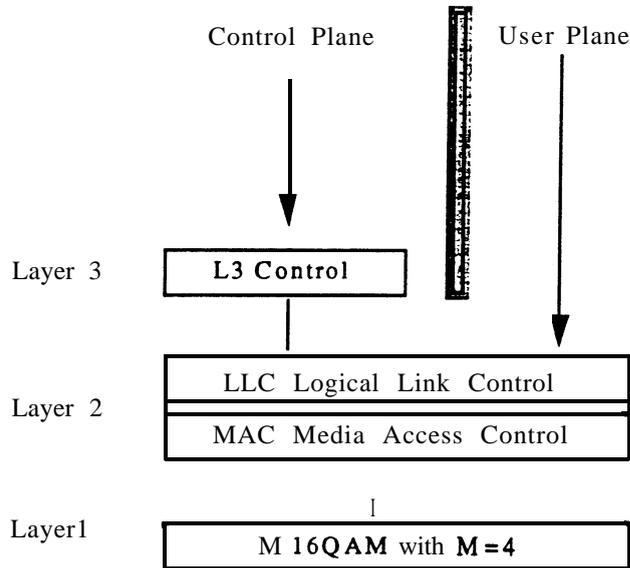


FIGURE 11
Protocol stack

REFERENCE

ARIB (November, 1995) RCR STD-32A – Integrated Dispatch Radio System.

ANNEX5

General description of the DIMRS System

1 Introduction

The Digital Integrated Mobile Radio System (DIMRS), using new digital technology, **fully** integrates multiple services including, radio-telephone, paging and dispatch communications into a single infrastructure. DIMRS caters both to users who require an integrated system with enhanced services as well as users who cannot **justify** the use of a separate pager, cellular phone, dispatch radio and data modem.

2 System services

The services provided are:

Dispatch

- 1) Group call.
- 2) Private call.
- 3) Call alert.
- 4) Push-to-Talk (**PTT**) ID.
- 5) **Landline** to individual private call.
- 6) Selective "area" calling.

Interconnect

- 1) Interconnect with other switched networks.
- 2) Full-duplex operation.
- 3) Handover.
- 4) Custom calling features (call waiting, three party calling, DTMF access to services, call forwarding, busy transfer, no answer transfer, call restrictions, access to information services).

Roaming services

- 1) Intra-system roaming.
- 2) Inter-system roaming.
- 3) System-to-system handover.
- 4) Inter-system calling features.
- 5) Registration/de-registration.

Message paging

- 1) Paging.
- 2) Short message service.

Data communications

- 1) Circuit mode (protected).
- 2) Packet mode:
 - with handshake;
 - without handshake.

3 Authentication mechanism

DIMRS provides system security control with an authentication mechanism which may be invoked prior to any chargeable service initiation.

Authentication is used to **verify** that a mobile station is registered in the system. It may take place during the location updating, mobile origination, mobile termination, supplementary service, and short message service procedures for an interconnect subscriber. For a dispatch only subscriber, authentication will occur during power-up or when a subscriber crosses certain system boundaries such as into another service provider's area.

Each Mobile Station (MS) user is assigned an individual ID, referred to as an International Mobile Subscriber Identity (IMSI), which is understood by both the dispatch and interconnect call processing programmes. The system will validate the user IMSI each time an interconnect call processing procedure is performed.

For interconnect call processing, a temporary ID, referred to as the Temporary Mobile Station Identifier (TMSI), is used to identify the MS to the system. This minimizes broadcasting the **IMSI** over the air.

4 Overview of the system

The network approach showing the major architectural components of the system is shown in Figure 12.

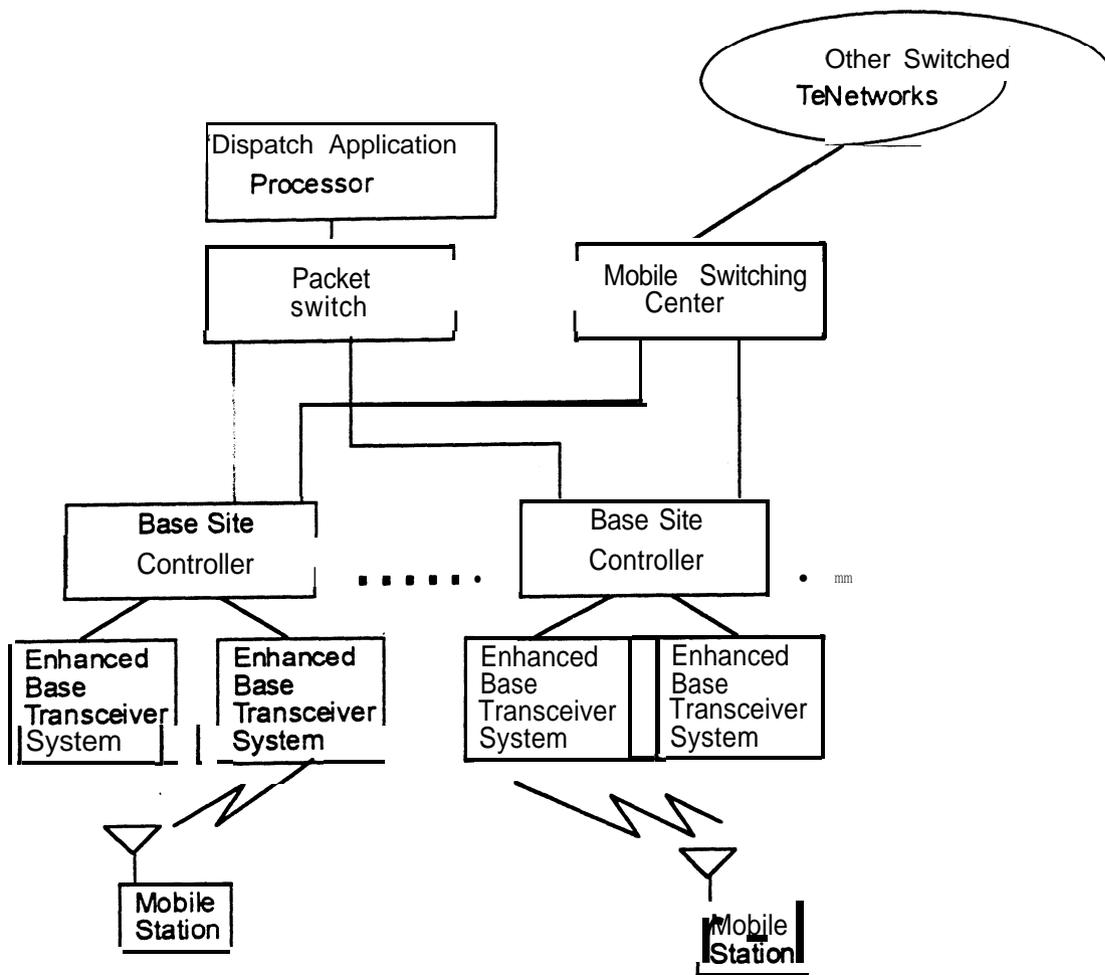


FIGURE 12
DIMRS Network approach

5 System specifications

Refer to Table 1.

5.1 Logical channels

The following logical channels are defined:

Slot Information Channel (SICH)

A broadcast channel used for transmission of slot control information.

Primary Control Channel (PCCH) comprising:

- Broadcast Control Channel (BCCH).
- Common Control Channel (CCCH).
- Random Access Channel (RACH).

The PCCH is a multiple access channel used for Layer 3 control signalling between the fixed network equipment and the mobile stations. Each cell has one PCCH.

Temporary control channel (TCCH)

A temporarily allocated multiple access channel used to provide a means for inbound random access on a channel which is normally reserved access.

Dedicated control channel

Supports more extended Layer 3 control procedures which would be inefficient if conducted on the PCCH.

Associated control channel (ACCH)

The ACCH provides a signalling path on the **traffic** channel. The main application of the ACCH is to support whatever Layer 3 control signalling is required for **traffic** channel supervision. Bandwidth for the ACCH is obtained by dynamically stealing on the TCH.

Traffic channel (TCH)

- Circuit-Switched Channels
These channels are used to transport voice or circuit-switched data **traffic**.
- Packet-Switched Channel (**PCH**)
These channels will support packet-switched user data communications.

5.2 TDMA frame structure

The DIMRS data stream **structure**, shown in Figure 13, has six slots per TDMA cycle. A frame structure is **further** superimposed on this cyclical structure. Inbound and outbound frames consist of 30 240 slots, each 15 ms long. The duration of the frame is 453.6 seconds.

A **hyperframe** structure is also defined, in addition to the frame structure. A **hyperframe** comprises 256 frames, thus, it contains a total of 7 741 440 slots and has a duration of 116 121.6 seconds (32 hours, 15 minutes, 2 1.6 seconds). The large number of slots in the **hyperframe** is useful for implementing encryption.

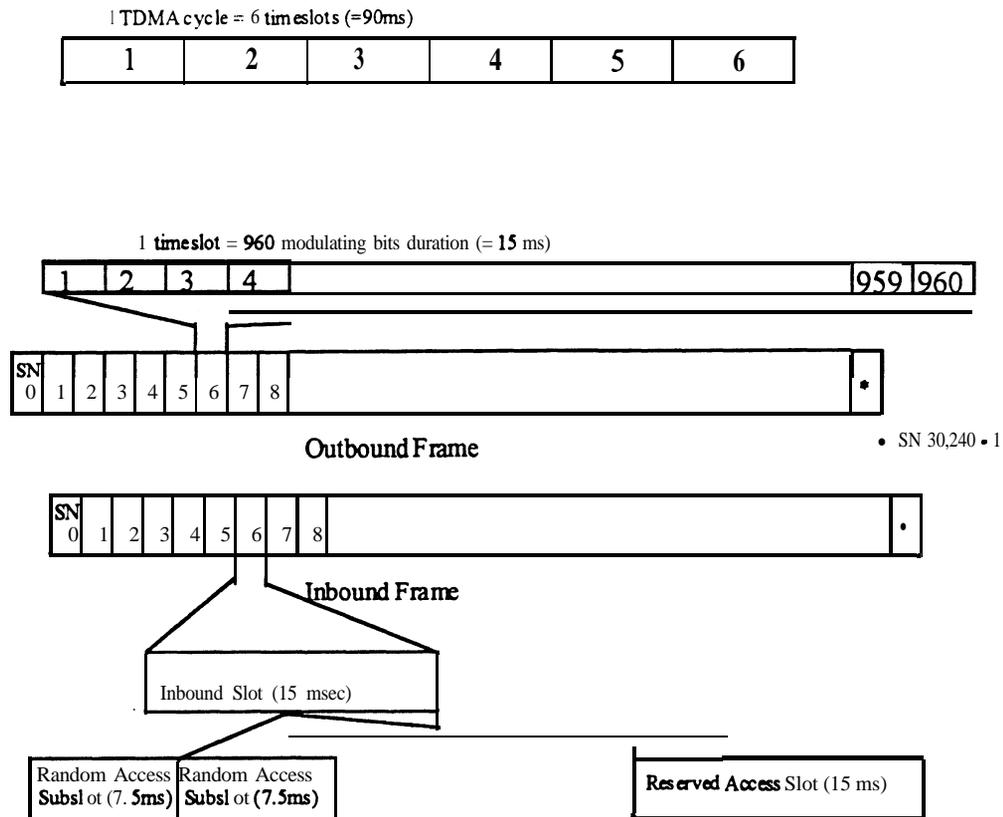


FIGURE 13
DIMRS Frame structure

5.3 Traffic channels

5.3.1 Speech traffic channels

The speech coding technology used is Vector Sum Excited Linear Prediction (VSELP). Acceptable quality is maintained at channel bit error ratios as high as **4-5%** in **Rayleigh** fading, or 10% in static conditions. Error correction is realized through a variable rate strategy whereby the **uncoded** and trellis-coded 16 QAM **modulations** are applied selectively to speech bits in accordance with their perceptual significance.

5.3.2 Data traffic channels

A circuit data protocol is available for circuit data applications such as laptop or palmtop computers, fax and image processing, and file transfer applications. The circuit-switched data protocol offers a full-duplex packet stream with a single rate of **7.2 kbit/s** (six users per RF carrier). This includes forward error correction coding and selective re-transmission of non-correctable blocks.

Allowance has been made for packet data in DIMRS. **Bandwidth** will be dynamically adjusted to accommodate demand.

6 Operational characteristics

6.1 Location updating and roaming

6.1.1 Intra-system roaming

DIMRS tracks a unit's location so that calls can be routed to it. Both the dispatch and interconnect calls require the current location of an MS. The DIMRS system will utilize a Location Area (LA). The unique identity of a location area is conveyed via cyclic broadcast on the primary control channel. The mobile monitors the preferred primary control channel and issues a location update request when it finds its location area is no longer supported. The location update request is sent to the Visitor Location Register (VLR) that holds the current location of MS units operating in that system.

6.1.2 Inter-system roaming

The ability to travel freely throughout the single service area and originate or receive calls without regard to current location can be extended to allow MS's to travel from one service area to another. A single service area can consist of multiple cells covering a large geographical area (e.g. entire metropolitan area). Alternatively, it may be necessary or desirable to subdivide it into multiple service areas, because of RF coverage gaps, management, or regulatory issues.

6.1.3 System-to-system handover

DIMRS supports **handover** between cells, between Location Areas, and between systems. **Handover** allows for maintaining the link quality for user connections, minimizing interference, and managing **traffic** distributions. The inter-system **handover** is facilitated in the MS's switch.

6.1.4 Inter-system calling features

The MS's in the DIMRS can achieve inter-operability between any system configurations.

6.2 Communication protocols

The communication protocols are layered according to the Open Systems Interconnection (OSI) reference model.

6.3 Operation

6.3.1 Dispatch call operation

- 1) A dispatch call is requested via PTT activation.

The call request packet is routed to the Dispatch Application Processor (**DAP**).

The DAP recognizes the MS unit's group affiliation and tracks the group members' current location area.

- 2) The DAP sends location requests to each group member's location area to obtain current sector/cell location.
- 3) The MS units in the group respond with current sector/cell location.
- 4) The DAP instructs the originating EBTS with packet routing information for all group members.

- 5) Call voice packets are received by the Packet Duplicator (PD), replicated, and distributed to the group's end nodes.

6.3.2 Telephone interconnect operation

A Call initiation – Inbound

- 1) Random Access Procedure (RAP) on primary control channel.
- 2) Get dedicated control channel assigned.
- 3) Authentication (optional).
- 4) Call setup transaction.
- 5) Get assigned to a traffic channel.
- 6) Talk.
- 7) Call termination request on associated control channel.
- 8) Channel released.

B Call initiation – Outbound

- 1) Page MS on primary control channel.



WinAPRS™

Windows Automatic Position Reporting System **A Windows™ Version of APRS™**

Mark Sproul, KB2ICI
sproul@ap.org

Keith Sproul, WU2Z
ksproul@noc.rutgers.edu

Abstract

WinAPRS is a Windows version of the popular *APRS*, Automatic Position Reporting System. WinAPRS is fully compatible with *APRS™*, The DOS version, and *MacAPRS™*, the Macintosh version. Due to the larger amounts of memory available in the Windows operating system, WinAPRS, just like MacAPRS has many additional features not available in the DOS version.

WinAPRS

WinAPRS is growing rapidly. Just like APRS and MacAPRS, the users are finding more and more things to do with this technology. We (Bob Bruninga and the Sproul Brothers) are committed to keeping the on-air protocols the same and are working with many different groups to expand and add many different capabilities to the APRS group of programs. One of the recent developments along these lines is a large interest from several National Weather Service groups across the country.

WinAPRS uses the exact same map files as MacAPRS, and will also use the map files from DOS APRS. Most of the source code of WinAPRS is the exact same code as MacAPRS, so it has been around for a few years, and has been thoroughly tested. See the discussion below about the development system used for MacAPRS/WinAPRS.

WinAPRS is a full Windows-95 **32-bit** application that follows the Windows User Interface Guidelines. It runs under Windows-95 and Windows-NT, and will run under Windows 3.1 and 3.1 .1 if you have the Win32 **DLLs** installed that allow Win95 applications to run under the older versions of Windows.

History of APRS

1992

APRS™ was first introduced by Bob Bruninga, WB4APR, in the fall of 1992 at the ARRL Computer Networking Conference in Teaneck, New Jersey. [1]. We, (Mark and Keith) were at this conference and saw Bob's program. Keith commented that he wanted to do some of this, but when we asked how much a GPS (Global Positioning Unit) cost, we got an answer of \$3000!. We decided to wait.

1993

APRS started gaining popularity. There were several articles in different magazines and many new uses for this growing technology. The article that caught a lot of attention was about using APRS to track the football from the Naval Academy to the Army-Navy game in Philadelphia. [2]

1994

In the fall of 1993, just about a year later, Keith started working on MacAPRS. [3] He contacted Bob Bruninga in February of 1994 and went to see him, with a working version of APRS that ran on a Macintosh. This version had many enhancements over the basic APRS features, including Call sign look-up from CD-ROM, and multiple maps open at the same time.

When Bob introduced APRS, all of his maps were made by hand! Keith, having had experience in college doing Cartography programming, refused to do maps by hand and did all of the maps for MacAPRS using USGS (US Geological Survey) map data, available on CD-ROM. Soon after Keith's visit to Bob, he started using the USGS CDs too. This improved the map quality greatly.

1995

By this time, APRS, and MacAPRS were becoming very popular and the uses of this technology had expanded much beyond the original concepts. The APRS programs have been used for Fox Hunting, Balloon Tracking, Weather Networks, DX Cluster monitoring, and many other applications. [4] [5]

At the Dayton Hamvention in April of 1995 Mark and Keith presented more and more of the fancy capabilities of MacAPRS. During 1995, we were invited to **give** talks at other hamfests and clubs in the New York/New Jersey/Connecticut area. During this time, one of the more common questions was "... do you have a WINDOWS version?..."

One of the more 'popular' features was the fact that MacAPRS **did not really have any** limitation as to the number of points that could be in **a map**. The DOS version, which when it first came out, was limited to 1,500 points had **been upgraded so that it could** handle 3,000 points, But the typical MacAPRS maps **STARTED at 10,000 points, with** some maps as large as 300,000 points. Other features that people were interested in that were not in the DOS version were the interface to the many different types of **call-sign** databases on CD-ROM.

At the Dayton Hamfest, we started getting more and more pressure from the Ham **Radio community** to do a Windows version. This **PRESSURE** got really severe at the ARRL DCC in Arlington, Texas.

When Keith **got** back from the ARRL DCC in Texas, we had long talks about **doing** a Windows version. Mark made the comment:

"I have never had so much peer pressure in all of my life..."

At this time, several critical items came together. **CodeWarrior**, the development system that the Sproul brothers used for MacAPRS came out with support for developing Windows programs on the Macintosh. Mark Sproul, who is porting **MacAPRS** to Windows finally succumbed to the pressure from APRS users. When these things happened, we determined that it was realistic to port the already developed Macintosh code to Windows and decided to do a Windows version of APRS. On September 15th, Keith went to down to see Bob Bruninga to discuss doing a Windows version. On September 16th, the following announcement was put up on the Internet:

**MacAPRS™ for Windows
(WinAPRS™)
Automatic Position Reporting System for Windows**

September 16, 1995

NORTH BRUNSWICK, NJ: Mark Sproul (KB2ICI) and Keith Sproul (WU2Z) authors of **MacAPRS™**, the Macintosh version of Bob Bruninga's (WB4APR) popular packet **radio mapping system** announced today that they will be porting their Macintosh **version** to Windows. This will be the official version and has the backing of Mr. **Bruninga**. The current plans are for beta release by Christmas 1995 and for the final release to be at the Dayton Hamvention in May of 1996.

APRS is a multi-faceted system used primarily within Amateur **Radio for tracking many** different types of things. APRS is used for tracking Weather, for tracking moving **cars, boats, weather balloons, and many other things. It can also be used as Graphics Information System** for many different aspects of Amateur Radio.

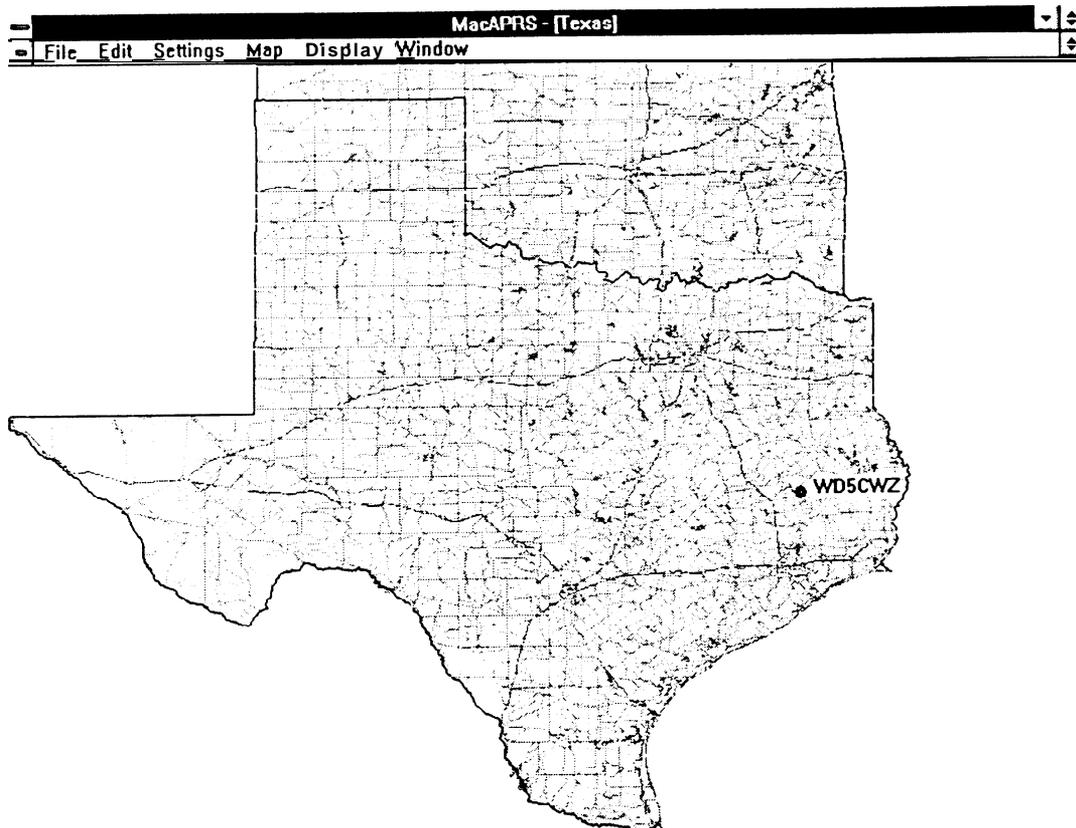
The original version of APRS was developed by Bob Bruninga, **WB4APR, to run under DOS** and was introduced at the 1992 ARRL **Computer Networking Conferences. MacAPRS** was released at the Dayton Hamvention in 1994.

The Macintosh version is written entirely in C and will **port easily to Windows. Keith and Bob** have worked hard at keeping the two versions **compatible and by using all of the C code** already developed for the Macintosh **version, it will ensure complete compatibility** on the Windows version. In addition, the two versions will **use the exact same map file format** so all of the wonderful maps **that the Mac users have will be** immediately usable by the Windows version.

When asked about future plans, Mark said, "When we finish **with the Windows version, we are planning on doing an X-Windows version as well.**"

October 14, 1995

One day less than one month after deciding to do WinAPRS, we had the maps drawing on a Windows computer and put screen-dumps of these maps up on the Web for all to see.



December 22, 1995

As promised in the original announcement, we released WinAPRS before Christmas. This release was to about 20 people.

January 28, 1996

We released a public beta version to the ham radio community. We showed WinAPRS publicly for the first time at the Wharton Hamfest near Chicago, Illinois.

May 1996

Again, as promised in the original announcement, we released WinAPRS version 1 .O.O at Dayton Hamfest 1996! This release had more features in it than we originally expected to have done at this time.

Development System of WinAPRS and MacAPRS

METROWORKS CODE WARRIOR

The Development system that we have been using for MacAPRS is Code Warrior by Metroworks. This development environment is a **full C/C++ development** system for the Macintosh. MacAPRS was written entirely in straight 'C', with no C++ at all.

In September 1995, Metroworks added the capability to **compile code** and create executable files for the Intel processors. You still have to write the code for the operating system that you want, i.e. it will NOT take the Macintosh program and simply re-compile it for Windows. You **MUST** write Windows code for the Windows applications and Macintosh code for the Macintosh applications. However, the routines that are not machine dependent end up being exactly the same.

What we have for done for the MacAPRS/WinAPRS system is to create two different applications that use most of the same code. For example, doing the math for drawing maps from a map file is the same no matter what platform it is on. Similarly, decoding data from a TNC is the same, etc. The source code that is different mostly involves the user interface.

All of the source code is written on the Macintosh. It is then compiled on the Mac. Then the executable file is transferred via **TCP/IP-EtherNet** to the **Windows computer**. The Code runs on the Intel processor, but the source-level debugging is done on the Macintosh via the network.

The source code for the entire MacAPRS/WinAPRS project is written with what is called **CONDITIONAL COMPILE** flags. This means that a specific section of **source** code may or may-not get compiled, depending on what flags **are set**. We have Macintosh Flags, Windows Flags, and several other internal **flags**. The objective of the system is to have as much of the code to be common, i.e. compiled in **ALL cases**, and as little as possible to be specialized code, i.e. compiled **ONLY** for Mac, or **ONLY** for Windows. By doing this, we have a much easier **system to maintain**, and a much more compatible system **across** different platforms

X-APRS, APRS for X-Windows (UNIX)

At the Dayton Hamfest in May, we had a SUN workstation **running a very preliminary** version of X-APRS (X-Windows is the Graphical **User Interface for UNIX computers**). This too is being done with the conditional compiles described above. **Doing the** development this way allows us to use code that has **been around a long time that has** been fully tested, thus speeding up development time. We **hope to have X-APRS out** sometime next year. (1997)

FUTURE

APRS, MacAPRS, WinAPRS and X-APRS are continuing to **evolve**. These programs **have** proven themselves to be useful in many more applications **than originally** imagined. This type of system is a system that takes full **advantage of the technology** available only in portable radio communications and **cannot be replaced with the** Internet.

References

- [1] ***Automatic AX.25 Position and Status Reporting***
Bob Bruninga, WB4APR
American Radio Relay League, 11 th Computer Networking Conference
Teaneck, New Jersey, November 7, 1992

- [2] ***Up Front In QST***, December 1994, p 14

- [3] ***MacA PRS, Automatic Position Reporting System, A Macintosh version of APRS***,
Keith Sproul, WU2Z and Mark Sproul, KB2ICI
American Radio Relay League, 13th Digital Communications Conference,
Bloomington, Minnesota, August 19-21, 1994. pp 133-145

- [4] ***Advances in APRS Technology***
Keith Sproul, WU2Z and Mark Sproul, KB2ICI
Proceedings of the 1995 TAPR Annual Meeting
St. Louis, MO, March 1995, pp 55-59.

- [5] ***Graphical Information Systems and Ham Radio
(The Future of A.P.R.S. Technologies)***
Keith Sproul, WU2Z and Mark Sproul, KB2ICI
American Radio Relay League, 14th Digital Communications Conference,
Arlington, Texas, September 8-10, 1995. pp 108-117

Internet Resources

Web sites with APRS Information

<http://aprs.rutgers.edu/APRS/>
<http://www.tapr.org/tapr/html/sigs.html>



Automatic Radio Direction Finding Using MacAPRS™ & WinAPRS™ Automatic Position Reporting System

Keith Sproul, WU2Z
ksproul@noc.rutgers.edu
<http://aprs.rutgers.edu/APRS/>

Abstract

Radio Direction Finding has been around for almost as long as radio itself. Doppler-based RDF systems have been around for quite awhile too. In the recent past, people have developed computer interfaces to Doppler-based RDF systems. APRS has the ability to display the RDF information on maps, giving the user a graphical way to view the RDF patterns.

Over the last few years, the call sign databases available on CD-ROM from several companies have become more and more sophisticated. There are also databases of commercial frequencies and locations available.

Most of us involved in Amateur Radio have experienced situations where we need to track down the cause of an unwanted radio signal, i.e. stuck microphone, improperly tuned equipment, or even a jammer.

With all of the available technology, we should be able to develop a system that zeros in on a location and automatically shows us the possible transmitters in the area.

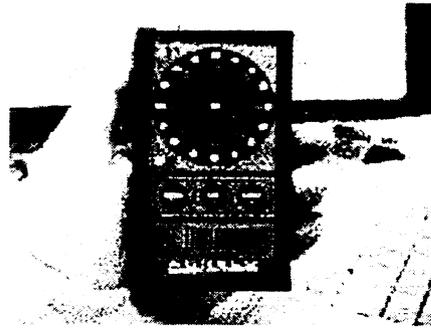
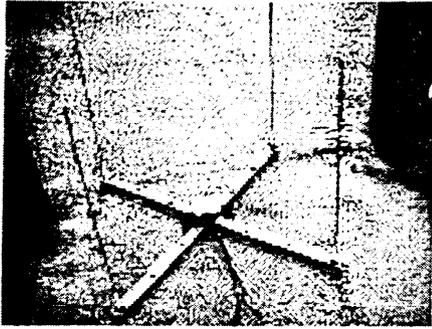
Computerized Radio Direction Finding

Doppler RDF units have been around for many years. Several years ago, people started trying to get the output of these RDF units to feed directly into a computer. One of the early versions of this was simply a method for reading the status of the **LEDs** on the RDF unit via a computer interface. Later on, these interfaces became more sophisticated. The current RDF units have serial ports that report not only the direction, but also signal strength indicators. The direction vectors are also reported in much higher accuracy resolution.

This year at the Dayton Hamfest, Agrelo Engineering introduced the **DFjr**. This unit is a complete computerized RDF unit. During the development of this unit, Agrelo worked with the developers of APRS to ensure smooth operation of their unit **and** the **APRS** software.

The 'normal' mode of operation of the **DFjr** is to have **it in** a car for **doing** RDF work. However, this unit also can be configured to be hooked up **to a TNC so that each** time

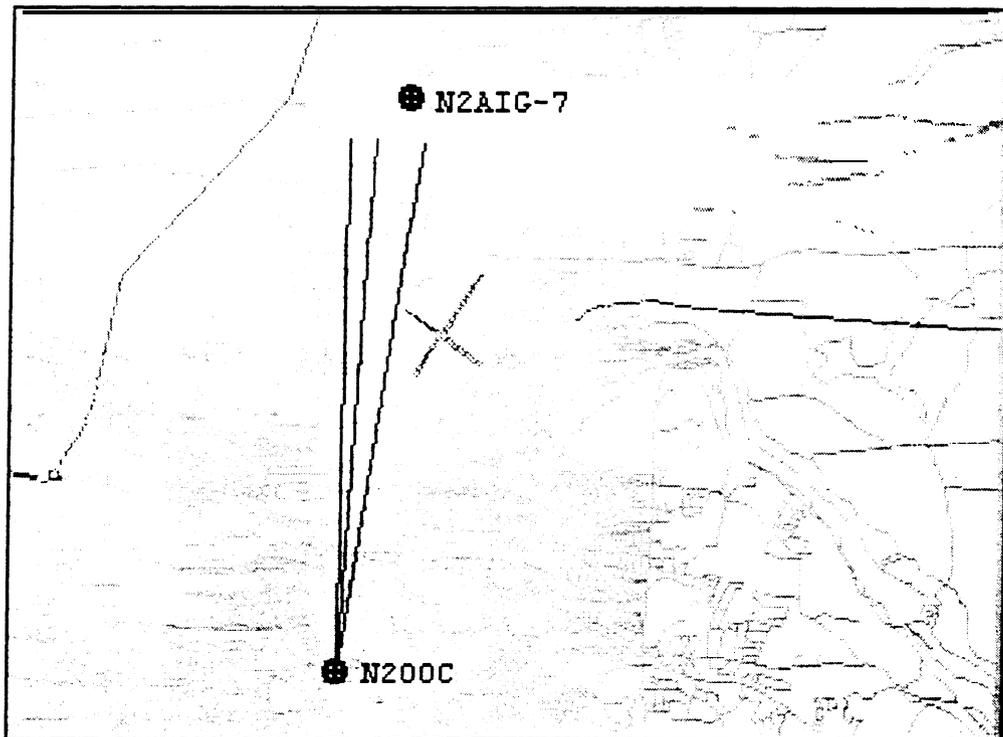
it hears a signal on the frequency it is monitoring, it will transmit the RDF information over Packet, using APRS protocols.

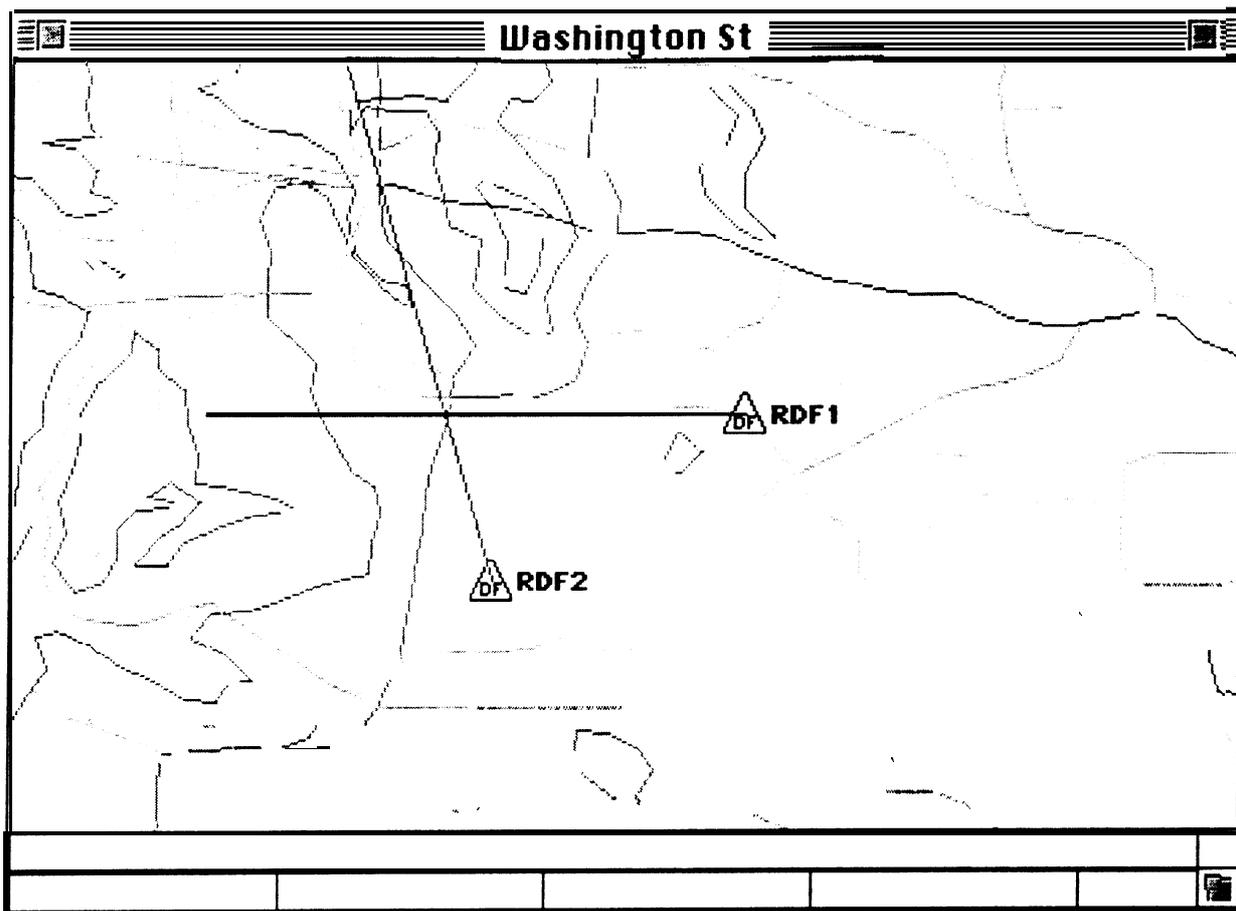


Agrelo DFjr, Computerized Doppler RDF Unit

Computerized RDF and APRS

APRS will take the output of the RDF units and display the information on any of the APRS maps. This gives you a geographical representation of the RDF data. If you have more than one RDF/APRS station participating, then you can get real-time intercept vectors. The first picture below shows WinAPRS and the vectors from a DFjr. The second picture below shows MacAPRS and two stations reporting RDF vectors.





Combining RDF and Call-sign Databases

Once the RDF information lets you know the area of interest, you can find **all of the stations** in the area with the help of the call-sign databases on CD-ROM. **MacAPRS** and **WinAPRS** can search through the database and show you **all of the stations** located in that general area. This is done via a database containing the **latitude / longitude** of all of the post offices in the US. Some of the CD-ROMs **are starting to add the Zip+4 lat/lon** to their databases. The **Buckmaster CD was the first to do this**. (This, alone, makes their CD one of the best available for this type of use).

The user can then search for all of the call signs reported to be in this **area**. **The user can select how big of an area to search**. The initial **search is done on the lat/lon of the zipcode**. This is done for speed. Then, once this group of data **has been selected**, it is further enhanced using the Zip+4 data, if available. **The chart below shows the information obtained from the Buckmaster Hamcall CD**.

The table below shows one page of approximately 110 people found within a 1 mile radius of the intersection point shown above. Realize that this is the **FIRST pass based on the 5-digit zipcode**. The table shows the actual distance from the intersection of the RDF vectors to each station based on the its zip+4 **lat/lon**. **If the CD-ROM database you are using has the Zip+4 location data**, you can double click on each one of the

stations in the list and it will show you exactly where that person lives on the map. (Within the accuracy of the Zip+4 system which is generally about 1/2 block).

Selected Call Sign List														
Call	LC	First Name	Last Name	Street	City	St	Zip	DOB	Lic Issue	Lic Expire	Area	County	Other	Dist
KB7OAT	N	Jeffrey	L. Pulten	4602 S 170th	Seattle	WA	98188-3254	19341017	19910820	20010820	206	King	I	0.2
KB7OHL	P	Eric	tl Emry	5565 S 152nd Apt 34	Tukwila	WA	98188-7815	19690708	19930928	20030928	206	King	I	1.3
KB7OTF	P	James	L. Quinton	3705 S 172nd	Sea Tac	WA	98188-3628	19490721	19920602	20020602	206	King	I	0.4
KB7RJA	T	Steve	P. Olson	3749 S 194th	Seatac	WA	98188-5360	19740715	19921229	20021229	206	King	I	1.5
KB7RYJ	P	Howard	T. Mayhew Jr	15325 Sunwood Blvd	Tukwila	WA	98188-5726	19410929	19940716	20030119	206	King	I	0.3
KB7SAQ	T	Theresa	A. Kennedy	3714 S 152nd 27	Tukwila	WA	98188	19580706	19930119	20030119	206	King	I	0.1
KB7TTH	T	Vaughan	F. Philpot	4011 s 152 St	Tukwila	WA	98188-2231	19240913	19930413	20030413	206	King	I	1.1
KB7UFM	T	Zita	Joan Halstrom	17047 35th Ave S	Seatac	WA	98188-3608	19330820	19930504	20030504	206	King	I	0.4
KB7YUQ	N	Gregory	S. Berglund	3754 s 172nd	Seatac	WA	98188-3627	19640501	19930921	20030921	206	King	I	0.3
KC7AHT	G	Nancy	B. Schimmelman	645 S Center 260	Seattle	WA	98188	1952021s	19940104	20040104	206	King	I	0.1
KC7AUZ	T	Jason	E. Parvu	3738 S 164th	Sea Tac	WA	98188-3040	19780212	19940215	2004021s	206	King	I	0.5
KC7AZX	T	Margaret	K. Thomasson	16432 32nd Rue S	Seatac	WA	98188-3021	19570805	19940222	20040222	206	King	I	0.7
KC7AZY	T	Norma	H. Thomasson	16432 32nd St	Seatac	WA	9818819310214		19940222	20040222	206	King	I	0.1
KC7BNS	T	Danice	tl Fisher	17343 Military Rd	Seatac	WA	98188-3651	19590727	19940329	20040329	206	King	I	0.3
KC7COS	T	Sidney	W. Anderson	3408 S 175th	Seattle	WA	98188-3662	19281125	19940531	20040531	206	King	I	0.5
KC7DBN	P	Douglas	W. Hans	16037 45 Th Rue S	Tukwila	WA	9818819470825		19940705	20040705	206	King	I	0.1
KC7FBP	T	James	E. Mitchell	17230 Military Rd	Seattle	WA	98188-3648	19710318	19950505	20040817	206	King	I	0.2
KC7FPM	T	Todd	J. Rogers	3054 S 150th	Seattle	WA	98188-2107	19830509	19941221	20041221	206	King	I	1.4
KC7HUZ	P	Lloyd	L. Crabtree	18625 39th Rue S	Seatac	WA	98188-5007	19351112	19950906	20041229	206	King	I	1.1
KC7IG0	T	Diosdado	A. Alejo	3511 S 160th St B1	Seattle	WA	98188-2634	19740127	19950117	20050117	206	King	I	0.7
KC7IGR	P	Tina	M. Patton	17341 32 Ave S A10	Seatac	WA	98188-4436	19581029	19950201	20050117	206	King	I	0.6
KC7IUC	T	Michael	S. Ward	3200 S 176th St 408	Seattle	WA	98188-4072	19670319	19950209	20050209	206	King	I	0.7
KC7KLU	T	Binyamin	Y. Levine	16801 33rd Ave	Sea Tac	WA	98188-3132	19461105	19950425	20050425	206	King	I	0.5
KC7LDN	T	Diane	L. DeMeerleer	4024 F S 158th	Seattle	WA	98188	19480202	19950522	20050522	206	King	I	0.1
KC7MFC	T	Quentin	W. Rapp	3806 S 179th St	Seattle	WA	98188-4167	19301015	19950714	20050714	206	King	I	0.7
KC7MUG	T	Jana	E. Ward	3200 S 176th St 408	Seattle	WA	98188-4072	19701208	19950819	20050819	206	King	I	0.7

Conclusion

This kind of Geographical Information System has many potential uses within the ham-radio community. This type of search is not limited to ham-radio databases only. There are databases available that contain similar information about commercial transmitters. These databases not only include latitude and longitude, but also actual frequencies etc. Over a year ago, when I started doing demonstrations of this type of capability, many people wanted to have it immediately. However, at that time, the computerized RDF units were either done as build-it-yourself kits, or for the most part, were just not available. Now, with the DFjr from Agrelo Engineering, this type of automatic RDF Unit is easily available and affordable. This type of technology will allow us to do semi-automatic Radio Direction Finding for such things as tracking down interference problems etc.

References

- [1] ***MacA PRS, Automatic Position Reporting System, A Macintosh version of APRS,***
Keith Sproul, WU2Z and Mark Sproul, KB2ICI
American Radio Relay League, Digital Communications Conference,
Bloomington, Minnesota, August 19-21, 1994. pp 133-145
- [2] ***Graphical Information Systems and Ham Radio
(The Future of A. P. R.S. Technologies)***
Keith Sproul, WU2Z and Mark Sproul, KB2ICI
American Radio Relay League, 14th Digital Communications Conference,
Arlington, Texas, September 8-10, 1995. pp 108-1 17

Internet Resources

Web sites with APRS Information

<http://aprs.rutgers.edu/APRS/>
<http://www.tapr.org/tapr/html/sigs.html>

Aarelo DFjr

<http://home.navisoft.com/agrelo/ae.htm>

CIRCUS OF THE STARS

Michelle Toon, KC5CGH
Doctoral candidate in Information Science
University of North Texas in Denton

Abstract

A unique collaboration between diverse groups is proving that, in Waco, Texas, the sky's the limit. Think of the most unlikely pairings you can and you will most likely find them in this conglomeration of interests. You will find Heart of Texas Amateur Radio Club (HOTARC) members alongside students in grades K-6 attending Baylor University's University for Young People. You will find astrology buffs and members of the Baylor Amateur Radio Club (BARC), KC5OJY. You will find Eagle Scouts and members of the Psychology and Sociology departments. You will find public school teachers and EMS Technicians. The MBA and the PhD. You will find gimme caps and belt buckles alongside mortar boards and sensible shoes. These unique collaborations are behind what is coming to be known as a "Circus of the Stars." This paper describes some of the events leading up to the "circus" coming to town and our plans for the event itself.

Key Words

astronomy, collaboration, community projects, packet radio, radio telecommunications

Introduction

In the spring of 1993, after viewing a packet radio demonstration during a telecommunications class, I went home wondering what this interesting technology was all about. I was thinking about it the next day when I arrived at work. I saw a computer technician who had something on his belt and asked him if he had ever heard of "packet radio." He pulled a laptop out of his bag, hooked up a few things, clicked a few keys, and asked me if this was what I meant. The technician, Kenneth Ransom, N5VHO, became my mentor on that day and we have been working together ever since. First, we created the Baylor Educational Amateur Radio Service (B.E.A.R.S.) which we hope will become a clearinghouse of information on using Amateur Radio in the classroom. We are in the process of forming the Central Texas Packet Educational Network. Six schools in the Central Texas area are interested in becoming involved in the Network. They are: China Spring Elementary School, Hillcrest Professional Development School, Parkdale Elementary School, Connally High School, LaVega High School, and Harker Heights Elementary School. To attract more potential members, we decided to organize a summer event that we have dubbed, "The Circus of the Stars"—"Stars" for Space, Telecommunications, Amateur Radio, and Schools, and "Circus" for the circus we have created and the atmosphere surrounding it.

Project Description

In "The Circus of the Stars" we plan to assemble a diverse group of people from the Central Texas area for fun, learning, and partnership. We would use telescopes, binoculars and our eyes to enter the world of the night sky, guided by amateur astronomers, scientists, teachers, children, parents, and our imaginations. We would use radio communications, both voice and digital, to communicate among three bases of operations around the Waco area—two in the field and one at the campus of Hillcrest Professional Development School, for the less adventurous or less able to negotiate the country footpaths. We would use amateur television to send from site to site our live reactions to what we see and do. We would send digital star maps to the sites to show them what to look for. All the while we would be forging partnerships: The college professor and the 6-year-old; the mommy and the amateur astronomer; the octogenarian CW operator and the 9-year-old's little sister. These partnerships, more than stars, planets, galaxies, or radio, are what this project is all about.

Initial Stages

In the spring of 1996, I spoke to the Heart of Texas Amateur Radio Club (HOTARC) in Waco about my research in using Amateur Radio in education. My goal was to find at least eight potential mentors for the six schools who had shown an interest in helping start a Central Texas Packet Educational Network. The HOTARC members were politely reserved at first as I began to explain some of the projects my colleagues of the North Texas Packet Radio Group had come up with over the past 3 years. I told of a project where a classroom sent out a CQ asking for people to send a square inch of dirt from their neighborhood. The idea of the class being able to claim ownership of a square inch of property from around the world made the HOTARC hams begin to take notice. I could see nods of agreement as I explained the geographical, scientific, language, and mathematical applications of packet radio in the classroom. By the time I had finished with my presentation, I could hear excited chatter. Folks who had been hams for years had discovered a new application for their hobby. At least eight hams volunteered to be mentors or showed an interest in helping in the schools. One of these hams, Steve Smith, KC5NGW, tried to contact me for several days after the HOTARC meeting. When we finally connected, Steve told me that he not only had ham equipment, but was an amateur astronomer with several telescopes. In addition, he worked with Amateur Television, of which I had heard but never seen. He volunteered to join me in some demonstrations of Amateur Radio and I accepted.

I began thinking about how we could use Steve and his equipment to attract more interest in our Central Texas Packet Educational Network. Rick Strot, a lecturer in Curriculum and Instruction at Baylor, approached me during this time. Rick works with Baylor's annual University for Young People (UYP—described below). We had talked earlier in the year about doing a unit on radio for this summer's UYP. Students entering grades 1-3 were to participate in a space program June 24-28 at Hillcrest Professional Development School in Waco. The students would learn how astronauts are trained, how spaceships are constructed, how astronomers study planets and stars. We decided to try for the night of June 25. We could have a "star party" with telescopes and radios. What better place to use Steve's talent and knowledge? I called Steve. He thought it was a great idea, just happened to be free that night, and so the "circus" was born. We would have three sites, connected by voice radio, packet, and Amateur TV. Word spread, both by mouth and airwaves, and soon, we had a real circus on our hands. Each day brought another group or expert who wanted to participate.

Description of Participants

Hillcrest Professional Development School (PDS) is a partnership between Waco Independent School District and Baylor University School of Education. Many of our faculty members hold their classes in this ungraded open environment. The **Baylor Amateur Radio Club (BARC), KC5OJY**, has a student membership of over 20 and almost as many faculty advisors. Its new hamshack, also the home of B.E.A.R.S., houses many modes of communication and is the home of the BEARSGW packet gateway we are in the process of constructing. The **Heart of Texas Amateur Radio Club (HOTARC)** is a large radio club in the Central Texas area that takes its teaching role in the amateur radio community very seriously. One can see the HOTARC trailer at various community events in the Central Texas area promoting Amateur Radio as a fun and interesting hobby. We asked **Professor Adam's Baylor Astronomy Class** to join in the fun, adding to the number of telescopes and to the number of "experts" available to answer our various questions. We figured that the more experts we had, the better the project would be. That's why we have invited the local **amateur astronomers, led by local stargazer, Paul Derrick**. This would give the children an assortment of adults with whom they could consult. We have invited **Baylor Science Education Classes to take part**. By coming out into the field and working in a situation like this first hand, future science educators would have a chance to interact with their curriculum and their charges. This is life in the real world. **Teachers would have an opportunity to interact with other subject matter experts (SMEs)** in a nonthreatening situation and would, no doubt, benefit from the chance to interact with students in an atmosphere of fun and excitement. Baylor has an **Eagle Scout troop** that takes part in events and helps with setup and arrangements. Many of these scouts have radio licenses so they are instrumental in keeping things running smoothly and are always ready to help in case of emergency. Baylor's **University for Young People (UYP)** is now in its thirteenth year. This commuter program, designed for gifted students, consists of two two-week

sessions for students entering grades one through eight. Younger students participate in an interdisciplinary, thematic curriculum that emphasizes problem-solving. Their classes are part of the practicum program for educators seeking advanced training in working with gifted students. It is within this thematic curriculum that our “circus” would fit nicely. In addition to hams, astronomers, parents, teachers, elementary students and college students, and media, we would involve others in the area. For instance, we might invite a van-load of veterans from the local VA hospital as well as those participating in a local day care program for the elderly.

Project Details

By way of preplanning, we would want to make sure that we and the students are **familiar** with the constellations that will be visible in the summer sky. These would include Leo the lion, Ursa Major (The Big Dipper), Bootes the herdsman, the Diamond of Virgo, Scorpius the scorpion, Sagittarius the archer and its teapot asterism, Lyra the lyre, Aquila the eagle, Cygnus the swan and the Milky Way (**summer**) Triangle. Students would study charts of these constellations so that when they were in the field, they would know what they were looking for. Jupiter and Saturn should both be visible. The children would study these planets in the classroom so that they might identify them in the field. Classroom descriptions of the planets’ sizes, locations and characteristics would take on new meaning when the students view the actual planets in the night sky.

Before participating in the field activities, students would discuss the various uses of the telescope. They would understand the concepts of distance and magnification and be prepared for what they might expect to see. They would also discuss astronomical distance and how light travels to the earth through space.

A space shuttle launching was planned for that day. It might be possible to see it pass over and hear the radio broadcasts from the shuttle astronauts. Discussions of space travel would be an integral part of the Space unit, including space suits, space foods, weightlessness, and other issues facing the astronauts. In addition, radio in its various incarnations would be an important part of the unit. Students would learn about some of the most popular formats-voice, packet, amateur-television, code-and would have a chance to participate in several of these modes at the “circus.” They would also learn that many of our astronauts hold amateur radio licenses and often arrange contacts with the folks back on Earth.

An important part of any project involving community partners is making the connections. This has been the easy part. Every time I talk to someone about our event, I get another group to add to the participants’ list. To further publicize our event, we planned to advertise on the local college channel in the form of an electronic bulletin board notice. I would also post a bulletin to our local packet BBS, **WD5KAL**. Word of mouth would take care of the rest.

We planned to set up at two sites in the country, one belonging to Steve, the amateur astronomer without whom this whole thing would not be possible, and one belonging to one of the participants in the UYP unit. The third would be at the PDS. We would need to go to these sites and make sure they are safe, that we can connect between them by voice and packet, and that we would not be compromising the sites by our presence.

It would be important, as this project got bigger and more complex, that I, as the ringleader, try to keep everyone informed as to any changes in plans, locations, or schedules. I would do this by any means possible and these means could include e-mail, snail-mail, TV, packet, phone, voice, or CW. After the event, we would compile a visual record of this event and make it available to those who participated.

Technologies Used

With the aid of telescopes and binoculars, we would view the stars, Jupiter and Saturn, the Moon, and, maybe, the Space Shuttle. The details of this would be left to those who have promised to bring the telescopes. We would use Voice Radio for communication among the three sites to give progress reports and tell kids at other sites where to look and what to look for. We would use Packet Radio communications ahead of time to get the word out on local BBS asking for participants and experts, to

download star maps, and to download the space shuttle passby schedule. During the event itself, we would use Packet to send digital drawings of constellations to other sites to see if they can find them, too. The Amateur Television Net would be taking place as we get our "circus" underway. We would not only get to see what is going on in the net, but we would also participate by sending video from site to site to show what is happening at any given time. Later, we would make a tape from these broadcasts to document the entire event.

Conclusion

It is easy to see why I almost decided to change the name of this "circus" to a "zoo." The more time that passed, the more frenzied it became. But, as quietly as it began, the "Circus of the Stars" ended. As with many plans that include Mother Nature, this project was not to be. At six o'clock on the night of June 25, I got a call from Steve Smith. The weather was cloudy and was expected to remain that way for the rest of the week. By the time it cleared, UYP would be over. Was all this planning and collaboration for nothing? The UYPers did learn a lot about the sky and I learned how to **plan** a big event. The community learned that there are many ways to collaborate. I talked to all the parties involved and we agreed that we would try to hold the event some time in the fall, when the weather cooled off and the viewing conditions were good. I am in touch with all concerned on a regular basis and we plan to make this an event to remember, whenever it takes place. And when it does, it will truly be a night for Space, Telecommunications, Amateur Radio, and Schools-A Circus of the Stars!

13cm PSK transceiver for 1.2Mbit/s packet radio

Matjaz Vidmar, S53MV

1. Introduction

The choice of a transceiver design for high-speed packet radio is not simple. Is it better to use an apparently simpler **FM** transceiver or to go for a more sophisticated **PSK** transceiver? Both choices have their advantages and disadvantages and at this time it is difficult to predict which one will become more practical. However, increasing the transmission speed both the signal bandwidth and the radio range need to be considered.

Increasing the data speed beyond about 100kbit/s, the resulting signal bandwidth is only acceptable at microwave frequencies. The transmitter power available at microwave frequencies is small and expensive. Therefore the radio range becomes a limitation even for line-of-sight terrestrial packet-radio links. A PSK transceiver with a coherent detector offers a radio range that is between 5dB and 15dB larger and a signal bandwidth that is less than half when compared with a FM transceiver.

In packet radio the main problem of a PSK transceiver is the initial RX signal acquisition. The latter is a function of the carrier frequency uncertainty. In a simple biphase PSK (BPSK) system with 0/1 80 degrees modulation, the initial signal acquisition requires a complicated searching loop, if the frequency error exceeds 10% of the bit rate. Quadriphase PSK (QPSK) allows a further halving of the signal bandwidth at the expense of a much more sophisticated demodulator design and an even more critical initial signal acquisition.

Therefore PSK becomes simple at high data rates. On the other hand, the signal acquisition of low-Earth orbit amateur packet-radio satellites transmitting at only 1200bit/s PSK is very difficult. This unfortunate PSK design made radio amateurs believe that PSK is not suitable for packet radio, being just an unnecessary complication at low data rates like 1200bit/s.

In this article a successful 13cm BPSK transceiver design will be described. In the 13cm amateur band, the sum of the frequency uncertainties of both receiver and transmitter is at least 10kHz using top quality temperature-compensated xtal oscillators. A real-world figure is 100kHz frequency uncertainty that requires a MINIMUM bit rate of about 1Mbit/s!

With the above restriction, a convenient choice is to use 1.2288Mbit/s for packet radio. This figure can easily be obtained with standard baud-rate xtals, being the 32nd multiple of 38.4kbit/s or the 1024th multiple of 1200bit/s. Of course the described transceiver can also be used for other digital data transmissions that require megabit rates, like compressed digital television transmission.

2. 13cm PSK transceiver design

Since the abovementioned PSK modulation is relatively unknown to most radio amateurs, the 13cm PSK transceiver block diagram will be discussed first. The same form of PSK modulation, namely 0/1 80 degrees BPSK, allows many different transceiver concepts. For example, a PSK signal may be generated at an IF frequency and then upconverted to the final transmitter frequency. A PSK signal can also be generated directly at the final frequency and even after the transmitter power amplifier. Finally, a PSK signal can also be fed through frequency multiplier stages, but here one should not forget that the PSK modulation phase angles are multiplied by exactly the same factors as the carrier frequency.

A PSK demodulator may be coherent or non-coherent. A coherent PSK demodulator offers a larger radio range, but requires a local carrier regeneration. A PSK signal is demodulated coherently by multiplication

with the regenerated carrier in a balanced mixer. Carrier regeneration requires a nonlinear processing of the PSK signal (in the case of BPSK this may be a frequency doubler) and a narrow bandpass filter (usually in the form of a phase-locked loop).

A PSK signal may be demodulated at a convenient IF frequency or directly at the receiver input frequency. A PSK receiver can be designed as a direct-conversion receiver just like a SSB receiver. Carrier regeneration may be performed by a squaring loop (frequency doubler) or by a Costas loop. Just like SSB, all PSK demodulators are very sensitive to small carrier frequency inaccuracies.

The block diagram of the described 13cm PSK transceiver is shown on Fig. 1. The transmitter includes a crystal oscillator followed by a multiplier chain. The PSK modulator - a balanced mixer operates at the final transmitter frequency and generates the desired signal directly. Modem semiconductor devices provide high gains per stage. The mixer is followed by just two amplifier stages at 2.36GHz to obtain about 0.5W of microwave power.

The receiver includes a double downconversion with the corresponding intermediate frequencies of 75MHz and 10MHz. The 10MHz coherent PSK demodulator is a squaring loop PLL. Although the receiver and the transmitter circuits are almost completely independent, the 13cm PSK transceiver is intended for standard CSMA (carrier-sense multiple access) simplex operation as usual for packet radio. Therefore the transceiver includes a PIN antenna switch and all of the remaining RX/TX switching is completely electronic as well. The RX/TX switching delay is in the range of 2ms and is mainly caused by the turn-on delay of the transmitter crystal oscillator.

3. TX exciter 590MHz / +10dBm

The circuit diagram of the transmitter exciter is shown on Fig.2. The exciter includes a crystal oscillator operating around 18.4MHz, followed by a multiplier chain. The exciter includes multiplier stages up to 590MHz. These are followed by additional multipliers located in the following module, the PSK modulator, mainly because of the different construction technology. A PLL synthesizer is not recommended in the exciter, since it was found difficult to isolate the PSK modulator from pulling the VCO frequency.

The oscillator uses a fundamental resonance crystal, since fundamental resonances have a lower Q than overtone resonances. The turn-on delay of the transmitter crystal oscillator can be reduced in this way. The transmitter crystal oscillator is turned off when receiving, since its fourth harmonic could disturb the first IF at 75MHz. For operation at 2360MHz, a "computer" crystal for 18.432MHz can be tuned to the desired frequency with a series capacitive trimmer. Using different crystals for other frequencies, a series inductivity L1 may be required in place of the capacitive trimmer.

The oscillator transistor is also used as the first multiplier, since the output circuit (L2 and L3) is tuned to the fourth harmonic of the oscillator frequency. Three additional frequency-doubler stages are required to obtain about 10mW at 590MHz. The first doubler stage uses air-wound, self-supporting coils L4 and L5, while the remaining two doubler stages use "printed" inductors L6, L7, L8 and L9. The supply voltage for the oscillator and the first doubler stage is stabilized by a 8V2 zener diode.

The transmitter exciter is built on a single-sided PCB with the dimensions of 40mmX120mm, as shown on Fig.3. The PCB is made of 0.8mm thick glassfiber-epoxy laminate to shorten the wire leads of the components and in this way reduce the parasitic inductivities. The component location of the transmitter exciter is shown on Fig.4.

L2 and L3 have about 150nH each or 4 turns each of 0.25mm thick copper-enamelled wire. They are wound on 36MHz (TV IF) coilformers with a central adjustable ferrite screw, plastic cap and 10mmX10mm square shield. L4 and L5 are self-supporting coils with 4 turns each of 1mm thick copper-enamelled wire, wound on an internal diameter of 4mm. Finally, L6, L7, L8 and L9 are etched on the PCB.

The transmitter exciter is simply tuned for the maximum output power. The individual stages are tuned to obtain the maximum drop of the DC voltage on the base of the next-stage transistor. Of course, the base voltage has to be measured through a RF choke. The base voltage may become negative, but should not exceed -1 V. Finally, the crystal oscillator is tuned to the desired frequency with the corresponding capacitive trimmer (or L1).

4. 2360MHz PSK modulator

The circuit diagram of the 2360MHz PSK modulator is shown on Fig.5. Except for the modulator (balanced mixer) itself, the module includes the last frequency-doubler stage, bandpass filters for 590MHz, 1180MHz and 2360MHz and an output amplifier stage to boost the signal level to about 15mW. All of the filters and other frequency-selective components are made as microstrip resonators on a 1.6mm thick glassfiber-epoxy laminate FR4.

The input resonator (L1) functions as an open circuit for the input frequency (590MHz) and as a short circuit for the output frequency (1180MHz) of the frequency doubler. In this way the operation of the doubler is less sensitive to the exact cable length and output impedance of the exciter. The output bandpass (L3, L4, L5 and L6) should not only suppress the input frequency (590MHz) but also its fourth harmonic (2360MHz) that could disturb the symmetry of the balanced mixer resulting in an unsymmetrical, distorted PSK.

A harmonic mixer with antiparallel diodes is used as the PSK modulator, since this circuit provides a reasonable unwanted carrier suppression (25dB) without any special tuning and without access to expensive test equipment (spectrum analyzer). The harmonic mixer uses a quad schottky diode BAT14-099R, since four diodes provide a higher output signal level than just two antiparallel diodes.

The mixer is followed by a bandpass filter for 2360MHz (L11, L12, L13 and L14) to remove the 1180MHz driving signal and other unwanted mixing products far away from the 13cm frequency band. The generated PSK signal at 2360MHz does not require any filtering itself. Since the 2360MHz signal level is low, about 0.3mW, a GaAs FET amplifier stage (CFY30) is used to raise the signal level to about 15mW.

The PSK modulator is built on a double-sided PCB with the dimensions of 40mmX120mm. Only the upper side is shown on Fig.6, since the lower side functions as the microstrip groundplane and is not etched. The PCB is made of 1.6mm thick glassfiber-epoxy laminate FR4, although this material has substantial RF losses at 2.36GHz. The component location of the PSK modulator is shown on Fig.7 for both sides of the PCB.

Although most of the transmission lines are etched on the PCB, L2, L9 and L15 are air-wound quarter-wavelength chokes. L2 is a quarter-wavelength choke for 1180MHz, L15 is a quarter-wavelength choke for 2360MHz while L9 should be a quarter-wavelength somewhere in the middle (around 1700MHz), since it has to be effective for both frequencies.

The described PSK modulator can simply be tuned for the maximum output signal level. Besides the 590MHz exciter signal, a digital modulating signal is required as well. The latter may be a square wave of the appropriate frequency or better the real digital packet-radio signal. Without any alignment, the PSK modulator will already provide an output of a few milliwatts. After any alignment of the microstrip resonators one has to check the modulation signal level to find the best operating condition of the harmonic mixer.

5. 2360MHz RF front-end

The circuit diagram of the 2360MHz RF front-end is shown on Fig.8. The RF front-end includes the transmitter power amplifier, the receiver low-noise preamplifier and the PIN antenna switch. The RF front-

end is the only module including microstrip circuits, that is built on a low-loss, 0.8mm thick glassfiber-teflon laminate with a dielectric constant of 2.5.

The circuit of the RF front-end is simplified by using modern SMD semiconductor devices, originally developed for cellular telephones. The transmitter power amplifier uses a single GaAs transistor CLY2 that provides both 15dB gain and more than 500mW of output power at the same time. Just a few years ago, an equivalent circuit would require three or four silicon bipolar transistors. The CLY2 is a low-voltage power GaAs FET that operates at a drain voltage of just 4.5V, while generating its own negative gate bias voltage by rectifying the input RF signal.

The antenna switch includes two different PIN diodes: BAR63-03W and BAR80. The semiconductor chips of these two diodes are similar, but there is an important difference in the packages. The BAR63-03W is built in a standard microwave SMD diode package with a low parasitic capacitance and is used as a series switch. On the other hand, the BAR80 diode is built in a low parasitic inductance package and is used as a shunt switch. Both diodes are turned on while transmitting. The quarter-wavelength line L7 transforms the BAR80 short circuit into an open circuit for the transmitter.

The RF front-end also includes a low-noise receiving preamplifier to improve the sensitivity and image rejection of the receiver. The low-noise preamp uses a CFY35 transistor, followed by a bandpass filter. The preamplifier provides a gain of about 11dB including the antenna switch and output filter losses. The bandpass filter is required to attenuate the image response around 2210MHz.

The RF front-end is built on a double-sided teflon PCB with the dimensions of 40mmX80mm. Only the upper side is shown on Fig.9, since the lower side functions as the microstrip groundplane and is not etched. The PCB is made of 0.8mm thick glassfiber-teflon laminate with a dielectric constant of 2.5. The component location of the RF front-end is shown on Fig. 10 for both sides of the PCB. Except the printed microstrip lines, there are three air-wound quarter-wavelength chokes for 2360MHz: L3, L5 and L8.

Assembling the RF front-end, the most critical item is the correct grounding of the microwave semiconductors CLY2, BAR80 and CFY35. The CLY2 and the BAR80 are grounded through drops of solder, deposited in 2mm diameter holes at the marked positions in the teflon laminate. On the groundplane side these holes are covered with small pieces of copper sheet that also act as heat sinks for these semiconductors. The CFY35 is grounded through two leadless ceramic disk capacitors installed in 5.5mm diameter holes at the marked positions. The capacitors are connected to the groundplane with small pieces of copper sheet on the other side. Finally, L6 is grounded with a 2.5mm wide strip of copper foil inserted in a slot in the teflon laminate.

The transmitter power amplifier is simply tuned for the maximum output power by adding capacity (small pieces of copper foil) to L1. Small sheets of copper foil can also be added in other parts of the circuit, but their influence is usually small when compared to L1. If the specified output power can not be obtained, the cable length between the PSK modulator and RF front-end needs to be changed.

The receiving preamplifier is also tuned for the maximum gain, but here it is more important to bring the bandpass filter to the correct frequency. The latter is adjusted with L1, while L10 only affects the CFY35 output impedance matching. Before making any RF adjustments, the DC operating point of the CFY35 has to be set by selecting appropriate source bias resistors for a Vds of 3-4V.

6. RX converter with PLL LO

To avoid several multiplier stages the receiving converter includes a microwave PLL frequency synthesizer. The converter is built as two separate modules to prevent the digital part from disturbing the low-level analog circuits. Of course each module is shielded on its own. The described RX converter is derived from a 2400MHz SSB converter published in [1].

The circuit diagram of the analog section of the RX converter is shown on Fig. 11. The analog section includes the second RF amplifier stage, the subharmonic mixer, the VCO including a buffer stage and the first 75MHz IF amplifier. The analog circuits are built as microstrip circuits on a 1.6mm thick glassfiber-epoxy laminate.

The main function of the second RF amplifier is to cover the noise figure of the harmonic mixer. The second RF amplifier is followed by another bandpass filter (L3, L4, L5 and L6), but unfortunately due to the high substrate losses this filter is unable to provide any significant rejection of the image frequency at 221 0MHz. Its main purpose is to reject far-away interferences like subharmonics or even signals at the IF frequency.

The harmonic mixer uses two antiparallel Schottky diodes and is very similar to the PSK modulator. Such a mixer requires a local oscillator at half of the desired conversion frequency thus simplifying the design of the PLL synthesizer. The resulting IF signal is amplified immediately to avoid any further degradation of the already poor noise figure.

The VCO uses a microstrip bandpass filter (L13, L14 and L 15) in the feedback network to obtain low phase noise. The tuning range of this VCO is thus restricted to a few percent of the central frequency. The VCO is followed by a buffer stage and part of the buffered VCO signal is coupled by L 10, L 11 to feed the digital section of the PLL.

The analog section of the receiving converter is built on a double sided PCB with the dimensions of 40mmX 120mm. Only the upper side is shown on Fig. 12, since the lower side functions as the microstrip groundplane and is not etched. The PCB is made of 1.6mm thick glassfiber-epoxy laminate FR4, although this material has substantial losses at 2.36GHz. The component location of the analog section of the RX converter is shown on Fig. 13 for both sides of the PCB.

Although most of the transmission lines are etched on the PCB, there are two discrete inductors in this module. L2 is a wire loop with a 2mm internal diameter made of 0.6mm thick silver-plated copper wire. L2 may need adjustments during the alignment of the complete transceiver. L8 is a quarter-wavelength choke around 1700MHz to be effective for both the RF and LO frequencies.

Most of the RF active devices (BFR90, BFR91 and BB105) are installed in 6mm diameter holes in the PCB. These holes are afterwards covered on the groundplane side by soldering small pieces of copper foil. The same installation procedure also applies to the two 470pF source bypass capacitors for the CFY30 transistor. The corresponding source bias resistors are adjusted for a Vds of 3-4V.

The alignment of the analog section should start with bringing the VCO to the desired frequency range by adjusting L14. This is done easily if the PLL is already operating. L14 usually needs to be made slightly longer to obtain a 2.5V PLL control voltage in the locked condition. Then L7 is adjusted for the maximum mixer conversion gain and finally L4 and L5 may need some small adjustments. L1 and L2 should be adjusted to match the RF front-end. If the second RF stage (CFY30) is self-oscillating, the L2 wire loop has to be made shorter.

An alternative solution is to replace the CFY30 GaAs FET with the silicon MMIC INA-03 184. The latter has a higher noise figure but offers more gain and does not self oscillate. When using the INA-03 184, L2 has to be replaced with a 6.8pF capacitor, the output bias resistor has to be increased from 4700hm up to 6800hm and the source bypass capacitors and bias resistors are no longer required, since the two INA-03 184 common pins can be grounded in a straightforward way. The circuit diagram of PLL section of the RX converter is shown on Fig. 14. The PLL includes the /64 prescaler (U664), the reference crystal oscillator at about 8.9MHz, two additional dividers (HC393) and the frequency/phase comparator (HC74 and HCOO). The PLL module has its own 5V supply voltage regulator 7805.

The abovementioned PLL is intended to replace a chain of frequency multipliers. Therefore it does not contain variable modulo dividers. The multiplication ratio is fixed to 128 (256 when considering the harmonic mixer) and the crystal frequency has to be selected according to the desired RF channel. In the frequency range around 8.9MHz, a "CB" crystal can usually be used on its fundamental resonance. Due to the wide tolerances of CB crystals either a capacitive trimmer or a series inductor L1 may be required to bring the crystal to the desired frequency. For operation at 2360MHz, the best choice is a crystal for 26.770MHz (CB channel 22 RX).

The frequency/phase comparator drives a charge-pump output network. The correct operation of such comparators is limited to low frequencies. Therefore both the VCO and reference signals have to be divided down to about 2.2MHz when using 74HC logic in the frequency/phase comparator. Fast (schottky) diodes BAT47 are required in the charge-pump network to avoid backlash problems that seriously deteriorate the phase noise of the frequency synthesizer.

The PLL is built on a single-sided PCB with the dimensions of 40mmX80mm, as shown on Fig. 15. The PCB is made of 0.8mm thick glassfiber-epoxy laminate. The corresponding component location is shown on Fig. 16. The only component installed below the PCB is the 1uH choke on the output. The only adjustment of the PLL is to bring the crystal oscillator to the required frequency. The PLL lock test point is not brought out of the shielding enclosure since it is only required during the adjustment of the PLL.

7. RX IF chain 75MHz/10MHz

The circuit diagram of the RX IF chain is shown on Fig. 17. The RX IF chain includes the second amplifier stage at 75MHz (BF98 1), the second mixer to 1 0MHz (another BF98 1) with its own crystal oscillator (BFX89) and the 10MHz limiting IF amplifier (CA3 189).

To receive correctly the 1.2Mbit/s BPSK signal, an IF bandwidth of about 2MHz is required. Most of the receiver selectivity is provided at 75MHz, especially the two tuned circuits with L2 and L3. The contribution of the tuned circuits with L1 at 75MHz and L5 at 10MHz is smaller, since the main function of the latter is the attenuation of far-away spurious responses.

The overall IF gain is even too large, although this does not cause instability problems. The IF gain can be decreased by replacing both BF98 1 MOSFETs with older devices like the BF960. The second conversion oscillator uses a fifth overtone crystal at 65MHz. L4 prevents the crystal from oscillating at its fundamental resonance around 13MHz and/or at its third overtone around 39MHz.

The integrated circuit CA3 189 includes a chain of amplifier stages with a high gain at 1 0MHz. In the described circuit, the CA3 189 functions as a limiter since limiting does not distort PSK signals. Although the gain of the CA3 189 drops quickly with increasing frequency, overloading the CA3 189 input with the remaining 65MHz LO signal has to be prevented with the lowpass filter with L5. The CA3 189 includes a S-meter output with a logarithmic response that may be very useful during receiver alignment.

The receiver IF chain is built on a single-sided PCB with the dimensions of 40mmX120mm, as shown on Fig. 18. The corresponding component location is shown on Fig. 19. L1, L2, L3 and L4 have about 400nH each or 5 turns of 0.15mm thick copper enamelled wire. They are wound on 36MHz (TV IF) coilformers with a central adjustable ferrite screw, ferrite cap and 10mmX1 0mm square shield. L5 has about 15uH or 25 turns of 0.15mm thick copper enamelled wire. L5 is wound on a 10.7MHz IF transformer coilformer with a fixed central ferrite core, adjustable ferrite cap and 1 0mmX1 0mm square shield.

The IF chain alignment should start by checking the operation of the 65MHz crystal oscillator on the desired overtone and adjusting L4 if necessary. All other tuned circuits (L1, L2, L3 and L5) are simply aligned for the maximum gain. Since the same circuits also define the selectivity of the receiver, the alignments have to be performed using a suitable 75MHz signal source: signal generator or grid-dip meter. The receiver thermal noise or other noise sources can not be used for this purpose.

8. 1.2Mbit/s, 10MHz PSK demodulator

Describing a PSK transceiver to radioamateurs, the least conventional circuit is probably the PSK demodulator. There are several different possible technical solutions for a BPSK demodulator. The circuit diagram shown on Fig.20 is probably one of the simplest coherent BPSK demodulators. Its principle of operation is a squaring-loop carrier recovery, followed by a PLL filter and a mixer. EXOR gates are used elsewhere for the squaring and mixing functions.

The input 10MHz IF signal is first boosted to TTL level with an emitter follower (2N2369) followed by one of the gates of a 74HC86 (pins 1, 2 and 3). Next the IF signal is multiplied by its delayed replica (squaring or second-harmonic generation) in another EXOR gate (pins 4, 5 and 6). The delay is obtained with a RC network. On the output of this circuit, pin 6 or test point #1, a double IF carrier frequency is obtained, since the BPSK modulation is removed by the frequency-doubling operation. The latter transforms 180 degrees phase shifts into 360 degrees phase shifts or in other words a 0/180 degrees phase modulation is completely removed.

The signal available at test point #1 includes a strong spectral component at twice the carrier frequency around 20MHz, but also many spurious mixing products and lots of noise. The desired 20MHz spectral component is "cleaned" by a PLL bandpass filter, since the phase shift between the input and output signals in a PLL is well defined. A mixer is used as the phase comparator, in practice another EXOR gate (pins 8, 9 and 10). The VCO operates at 40MHz, so that a perfect square wave can be obtained at 20MHz with a simple divider by two (one half of the 74F74).

The regenerated BPSK carrier is obtained by another frequency division by two (other half of the 74F74). The BPSK demodulation is finally performed by the remaining EXOR gate (pins 11, 12 and 13 of the 74HC86). Because of the division by two, the regenerated carrier phase is ambiguous 0 or 180 degrees. As a consequence, the polarity of the demodulated data is also ambiguous and this ambiguity can not be removed in a 0/180 degrees BPSK system regardless of the type of demodulator used.

Fortunately amateur packet-radio usually uses NRZI (differential) data encoding, where level transitions represent logical zeroes and constant levels represent logical ones. The polarity of the signal is therefore unimportant and the abovementioned drawback of 0/180 BPSK modulation does not represent a limitation in a packet-radio link. However, the polarity ambiguity has to be considered when designing data scramblers and/or randomizers for NRZI signal processing.

The PSK demodulator is followed by a RC lowpass filter to remove the carrier residuals. The lowpass is followed by an amplifier (74HC04) to boost the demodulated signal to TTL level and eventually drive a 75-ohm cable to the bit-sync unit. The PSK receiver therefore only has a digital output, there are no outputs for loudspeakers or headphones.

The PSK demodulator is built on a single-sided PCB with the dimensions of 40mmX 120mm, as shown on Fig.2 1. The corresponding component location is shown on Fig.22. The VCO components have to be selected carefully to avoid frequency drifts. The VCO capacitors must be NPO ceramic or stiroflex types with a low temperature coefficient. The VCO coil L 1 has around 400nH or 6 turns of 0.15 thick copper enamelled wire on a 36MHz (TV IF) coilformer with a central adjustable ferrite screw, plastic cap and 10mmX 10mm square shield.

The alignment of the PSK demodulator should start with the adjustment of the delay of the input signal frequency doubler. A DC voltmeter is connected to test point #1 through a RF choke. The capacitive trimmer on pin 5 of the 74HC86 is adjusted to obtain an average (DC) voltage of 2.5V on test point #1 with some input signal: either receiver noise or a valid PSK signal.

Next a coarse adjustment of L1 is performed to bring the VCO frequency to 40MHz with no input signal. Then a valid PSK signal is applied and the DC voltage on test point #2 is measured through a RF choke. The DC voltage on test point #2 should follow even small movements of the core of L1 when the PLL is locked. The core of L1 is finally adjusted for 2.5V in the locked state or in other words the DC voltage should not change when the input signal is removed and only noise is present.

Finally, the correct phase of the regenerated carrier has to be set. An oscilloscope is connected to test point #3 through a RF choke and a valid PSK signal is applied to the input. The capacitive trimmer on pin 13 of the 74HC86 is adjusted to obtain the maximum amplitude of the demodulated signal. Alternatively, a DC voltmeter can be connected to test point #3 and the PSK demodulator is driven by an unmodulated carrier. The capacitive trimmer on pin 13 is adjusted either for the maximum or minimum DC voltage, depending on the (phase ambiguity!) locking point of the PLL.

9. Supply switch interface

The circuit diagram of the supply switch and some additional interface circuits is shown Fig.23. Most of the receiver circuits receive a continuous supply voltage of +12V. The supply switch only turns on the transmitter circuits (+12V_{TX}) and at the same time removes the supply voltage to the RX RF preamplifier (+12V_{RX}). The supply switching is performed by CMOS inverters (4049UB). The high TX current drain requires an additional PNP transistor BD 138.

The RX/TX switching is driven by the PTT line. Just like with other transceivers, the PTT input is defined as a switch that closes towards ground when transmitting. The antenna PIN switch is driven by the +12V_{TX} line and does not require any additional switching signals. Since most of the receiver circuits remain operational when transmitting, several of the receiver circuits (converter with PLL, PSK demodulator) can be tested with their own transmitter signal due to the inevitable crosstalk between the transmitter and the receiver.

The supply switch interface module also includes the modulator driver. The TTL input includes termination resistors to prevent cable ringing, if a longer coaxial cable is used between the transceiver and the digital equipment. The TTL input signal is first boosted by a 74HC 125, followed by a resistive trimmer for the modulation level and a lowpass filter with the 1uH inductor. The modulation level is simply adjusted to obtain the maximum transmitter output power.

The 74HC 125 receives the supply voltage +5V also while receiving and only its **tri-state** outputs are disabled during reception. The two 1.8kohm resistors keep the 33uF tantalum capacitor charged to 2.5V to speed-up the RX/TX switching. The 33uF tantalum capacitor is the only capacitive signal coupling in the whole transceiver. All other signal couplings allow the transmission of the DC component of the digital signal. If the described PSK transceiver is to be used without a data scrambler or **randomizer**, the described capacitive signal coupling has to be removed by redesigning the modulator driver only, while the other circuits need not be modified.

The supply switch interface is built on a single-sided PCB with the dimensions of 30mmX80mm, as shown on Fig.24. The corresponding component location is shown on Fig.25. The PCB is intended to be installed behind the front panel of the transceiver and is intended to carry the RX and TX LEDs.

10. Assembly of the 13cm PSK transceiver

Building a PSK transceiver certainly represents something new for most radioamateurs, while the microwave frequencies make the job even more difficult. Except for the careful design of the various circuits, the mechanical layout and assembly also have to be considered right from the beginning. To avoid any possible shielding or crosstalk problems, the described transceiver employs a large number of shielded enclosures and feedthrough capacitors.

The PSK transceiver is enclosed in a custom-made aluminum box measuring 320mm(width)X 175mm(depth)X32mm(height). The individual module locations and RF interconnects are shown on Fig.26. The box is made of two "U"-shaped pieces of aluminum sheet. The front, bottom and back are made of 1mm thick aluminum sheet, while the cover and the two sides are made of 0.6mm thick aluminum sheet. The cover and sides are 190mm deep to exceed the size of the bottom by 7.5mm on the front and on the back.

The individual modules of the PSK transceiver are all (except the supply switch interface) installed in shielded enclosures made of 0.5mm thick brass sheet. The PCBs are soldered into a brass frame as shown on Fig.27. A brass cover is then plugged onto the frame to complete the shielding enclosure. The shielded module is then installed on the bottom of the aluminum box with four sheet-metal screws. The height of the aluminum box is selected so that the main aluminum cover keeps all seven small brass covers in position.

To retain the shielding efficiency of the single modules, all of the supply and low-frequency interconnects go through 220pF feedthrough capacitors soldered in the narrow sides of the brass frames. The RF interconnects are made with thin 50-ohm teflon cables (RG- 188 or similar). It is extremely important that the coax shielding braid is soldered in a "watertight" fashion to the brass sheet all around the central conductor using a suitable soldering iron.

The size and shape of the single-module shielded enclosures is selected so that the lowest waveguide mode cutoff frequency is well above the operating frequency of the transceiver in the 13cm band. The described shielded enclosures usually do not require any microwave absorbers or other countermeasures to suppress cavity resonances.

The described PSK transceiver probably represents the first serious construction using SMD parts for many amateur builders. Unfortunately SMD parts can not be avoided: at high frequencies it is essential to keep package parasitics small enough to obtain a good device gain, noise figure and/or output power. The described 13cm PSK transceiver was designed with Siemens SMD semiconductors originally intended for cellular telephones. Since these devices are relatively new, their packages and corresponding pinouts are shown on Fig.28. Please note that due to space restrictions, the package markings are necessarily different from the device names!

11. Experimental results

The design goal of the described transceiver was to develop a packet-radio transceiver capable of transmitting data at 1Mbit/s with a free-space radio range between 500km and 1000km using moderate-size antennas. Such equipment is required for real-world line-of-sight packet-radio links of 30-100km with a single transceiver connected to more than one antenna (to support more than one link) and with a reasonable link margin of 10-15dB to counter propagation effects.

The first two transceivers were finished in April 1995 and some laboratory bit-error rate measurements were made. The acknowledgements go to Knut Brenndoerfer, DF8CA, that supplied the author with up-to-date microwave SMD components. The first packet-radio link was installed in June 1995 between the SuperVoze!j packet-radio node GORICA:S55YNG and the experimental node RAFUT:S59DAY at the author's QTH.

Although the distance is only 5.8km, there is no optical visibility between these two locations. The obstacle (hill) exceeds the 10th Fresnel zone at 13cm and the reception of a commercial UHF TV repeater installed in the same location is not possible due to reflections corrupting the horizontal sync pulses. Nevertheless, two-way packet-radio communication at 1.2288Mbit/s was found possible although affected by fading, using the described 13cm PSK transceivers, 16dBi short-backfire (SBF) antennas and about 5dB of antenna cable loss at each side of the link!

The first operational 1.2288Mbit/s packet-radio link was installed at the end of July 1995 between the SuperVozelj packet-radio nodes GORICA:S55YNG and KUK:S55YKK at a distance of 22.1km. Next this link was extended to the SuperVozelj node IDRJA:S55YID in the beginning of October 1995, at a distance of 36.6km from KUK:S55YKK. The measured YKK-YID link margin is 17dB, although there are two SBF antennas at KUK:S55YKK pointed in different directions, but connected to one single 13cm PSK transceiver. The estimated cable losses are around 3dB at each side of the link.

All of these experiments are using SuperVozelj node computers [2]. The SuperVozelj packet-radio node computer is based on the MC68010 16-bit CPU and offers 6 low-speed interrupt-serviced channels up to 76.8kbit/s for user access (three 28530 SCC chips) and two high-speed DMA-serviced channels for megabit interlinks (28530 SCC + MC68450 DMA). The interface to the described 13cm PSK transceiver includes external bit-sync/clock recovery and a $1+X^{12}+X^{17}$ polynomial data scrambler/randomizer.

Currently seven prototypes of the described 13cm PSK transceiver have been built and four are already installed on mountaintop digipeaters. Together these prototypes accumulated more than one year of continuous operation with no failures. However, the described transceivers have not been checked in winter conditions yet, under wider temperature excursions to lower temperatures.

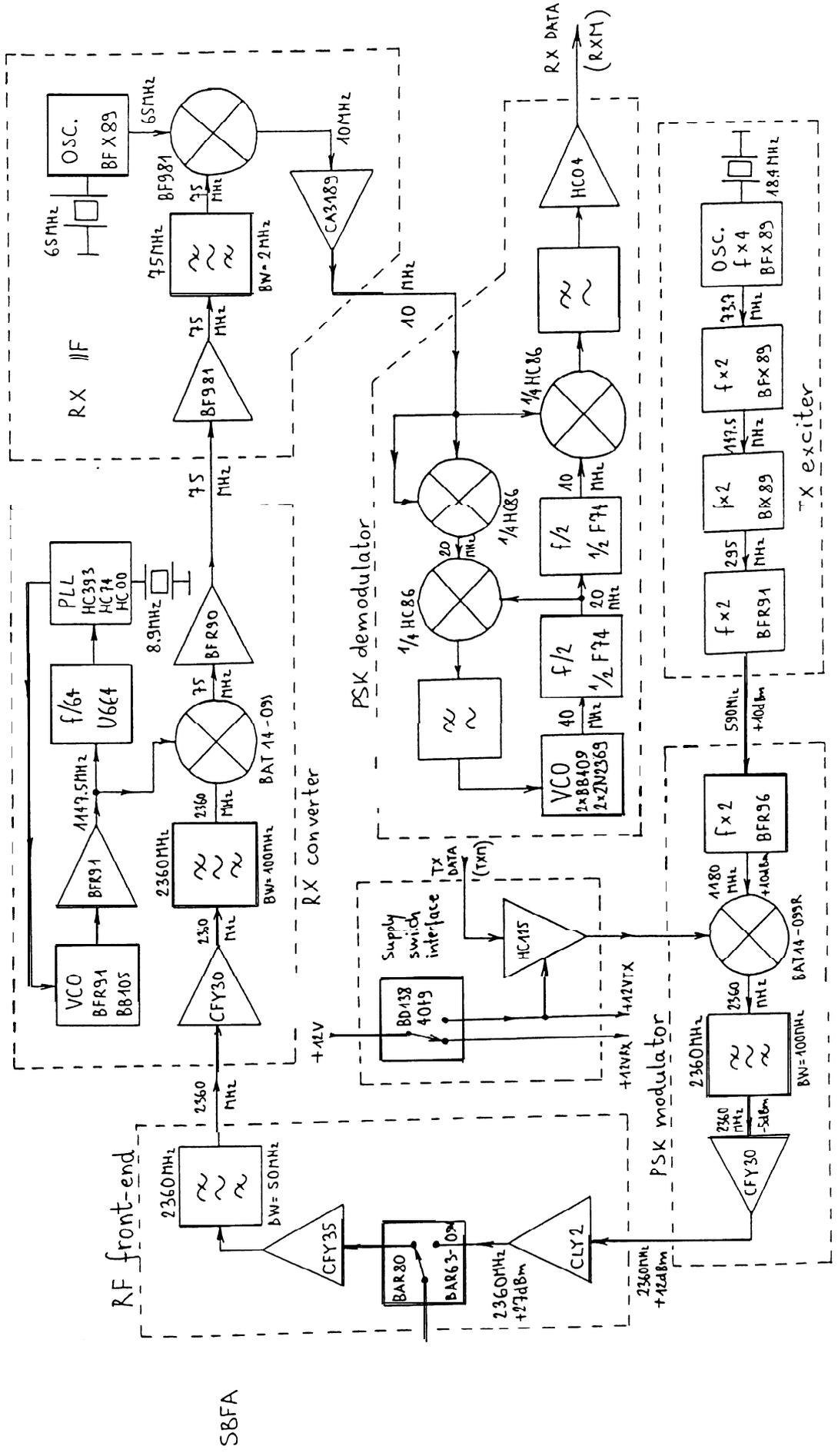
The described 13cm PSK transceivers finally demonstrated that megabit amateur packet-radio is not just possible but it is also a practical alternative. Using more sophisticated PSK transceivers with a larger radio range, a single PSK transceiver can be connected to more than one antenna and thus replace many narrowband FM "interlink" transceivers resulting in a simpler and cheaper packet-radio network. Of course, the next logical step is to develop simpler PSK transceivers for the user community, maybe using direct-conversion PSK demodulation.

12. References

- [1] Matjaz Vidmar, S53MV: "Ein Front-End fuer den Satellitenempfang im 13-cm-Band", AMSAT-DL Journal 2-94, pages 21-33.
- [2] Matjaz Vidmar, S53MV: "1.2Mbit/s SuperVozelj packet-radio node system", 40. Weinheimer UKW-Tagung, 16./17. September 1995, Scriptum der Vortraege, pages 240-252.

List of figures:

- Fig. 1 - 13cm PSK transceiver block diagram (2360MHz/1.2288Mbit/s).
- Fig.2 - TX exciter circuit diagram.
- Fig.3 - TX exciter PCB (single-sided 0.8mm glassfiber-epoxy).
- Fig.4 - TX exciter component location.
- Fig.5 - PSK modulator circuit diagram.
- Fig.6 - PSK modulator PCB (double-sided 1.6mm glassfiber-epoxy, microstrip circuit with a continuous groundplane).
- Fig.7 - PSK modulator component location.
- Fig.8 - RF front-end circuit diagram.
- Fig.9 - RF front-end PCB (double-sided 0.8mm glassfiber-teflon, microstrip circuit with a continuous groundplane).
- Fig. 10 - RF front-end component location.
- Fig. 11 - RX converter, analog section circuit diagram.
- Fig. 12 - RX converter, analog section PCB (double-sided 1.6mm glassfiber-epoxy, microstrip circuit with a continuous groundplane).
- Fig. 13 - RX converter, analog section component location.
- Fig. 14 - RX converter, PLL circuit diagram.
- Fig.15 - RX converter, PLL PCB (single-sided 0.8mm glassfiber-epoxy).
- Fig.16 - RX converter, PLL component location.
- Fig.17 - RX IF circuit diagram.
- Fig. 18 - RX IF PCB (single-sided 1.6mm glassfiber-epoxy).
- Fig. 19 - RX IF component location.
- Fig.20 - PSK demodulator circuit diagram.
- Fig.21 - PSK demodulator PCB (single-sided 1.6mm glassfiber-epoxy).
- Fig.22 - PSK demodulator component location.
- Fig.23 - Supply switch interface circuit diagram.
- Fig.24 - Supply switch interface PCB (single-sided 1.6mm glassfiber-epoxy).
- Fig.25 - Supply switch interface component location.
- Fig.26 - 13cm PSK transceiver module location.
- Fig.27 - 13cm PSK transceiver shielded module enclosure.
- Fig.28 - SMD component packages and **pinouts**.



F. 1 - 13 cm PSK transceiver block diagram (2360 MHz / 1.2288 Mbit/s).

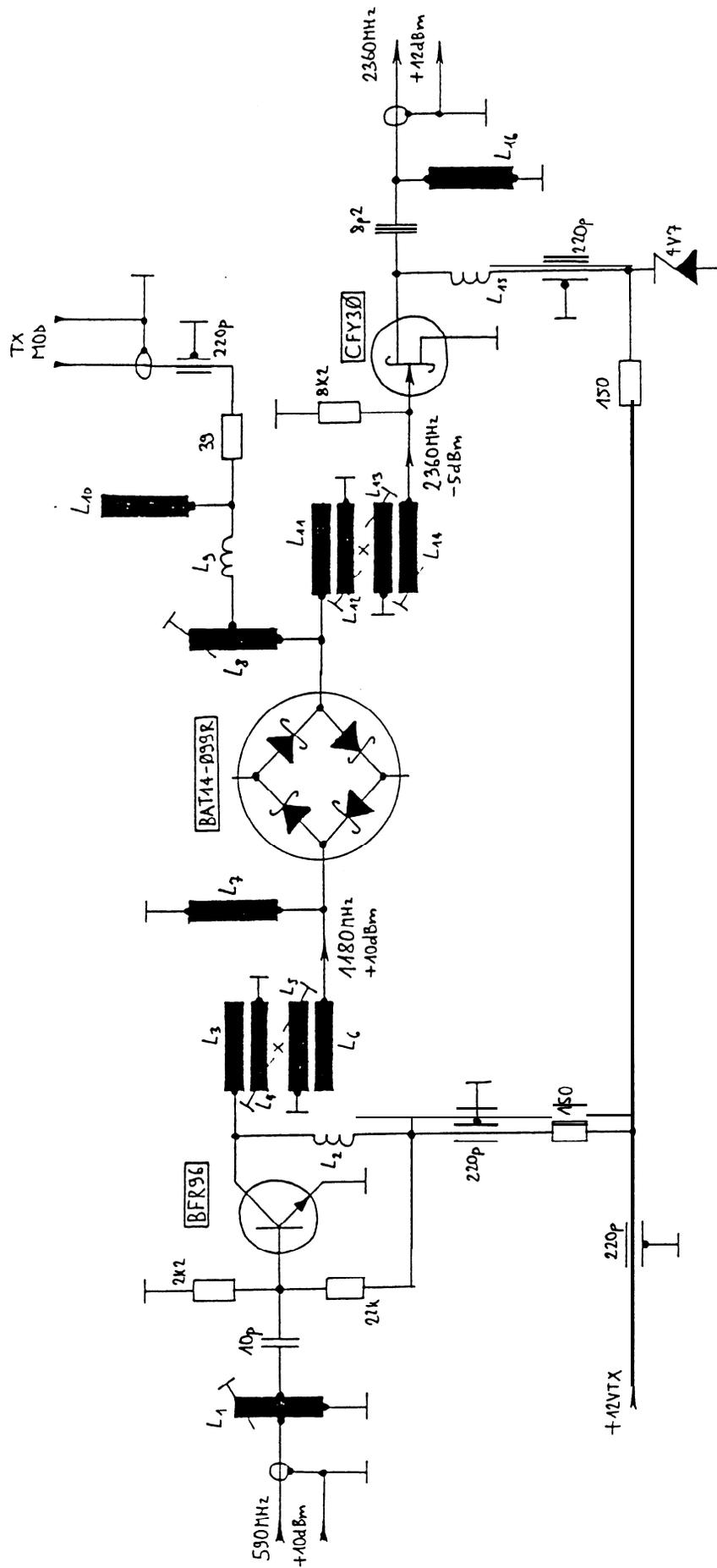


Fig. 5 - PSK modulator circuit diagram

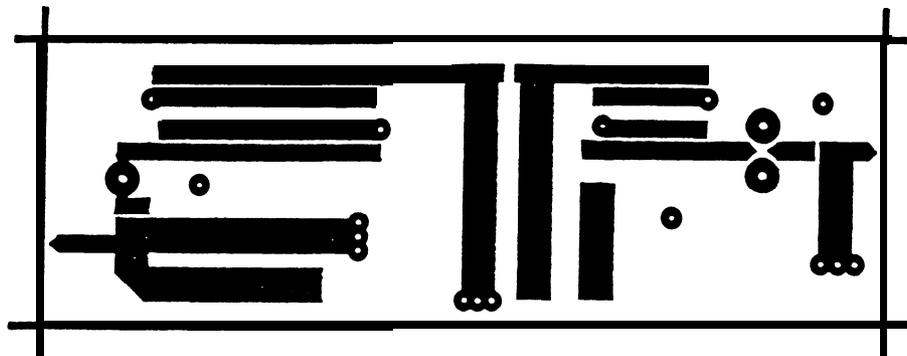


Fig. 6 - PSK modulator PCB (double-sided 1.6 mm glassfiber-epoxy, microstrip circuit with a continuous groundplane).

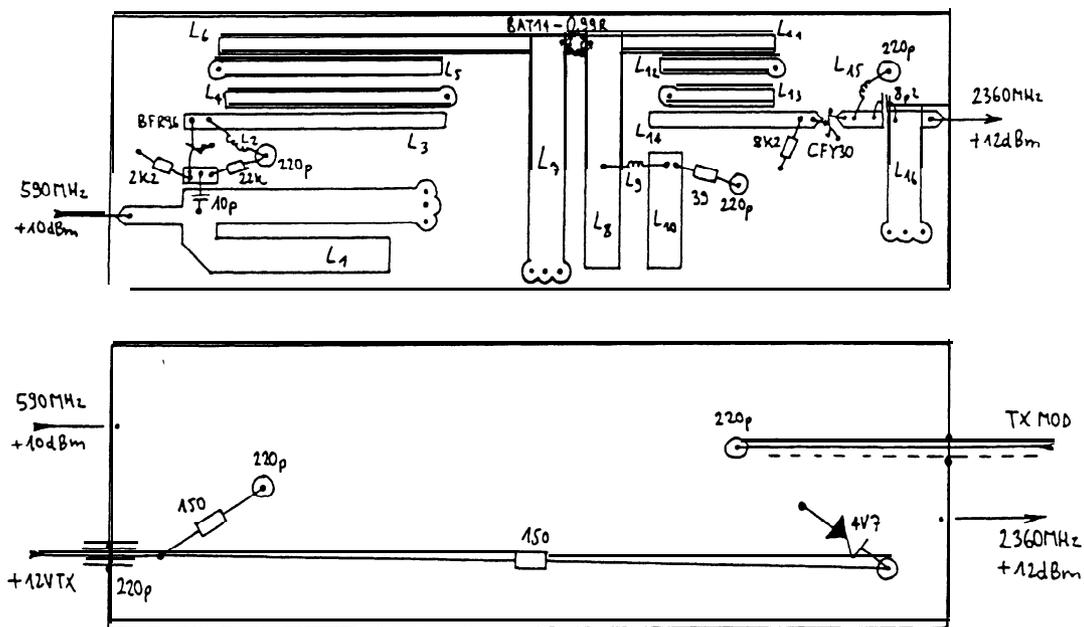


Fig. 7 - PSK modulator component location.

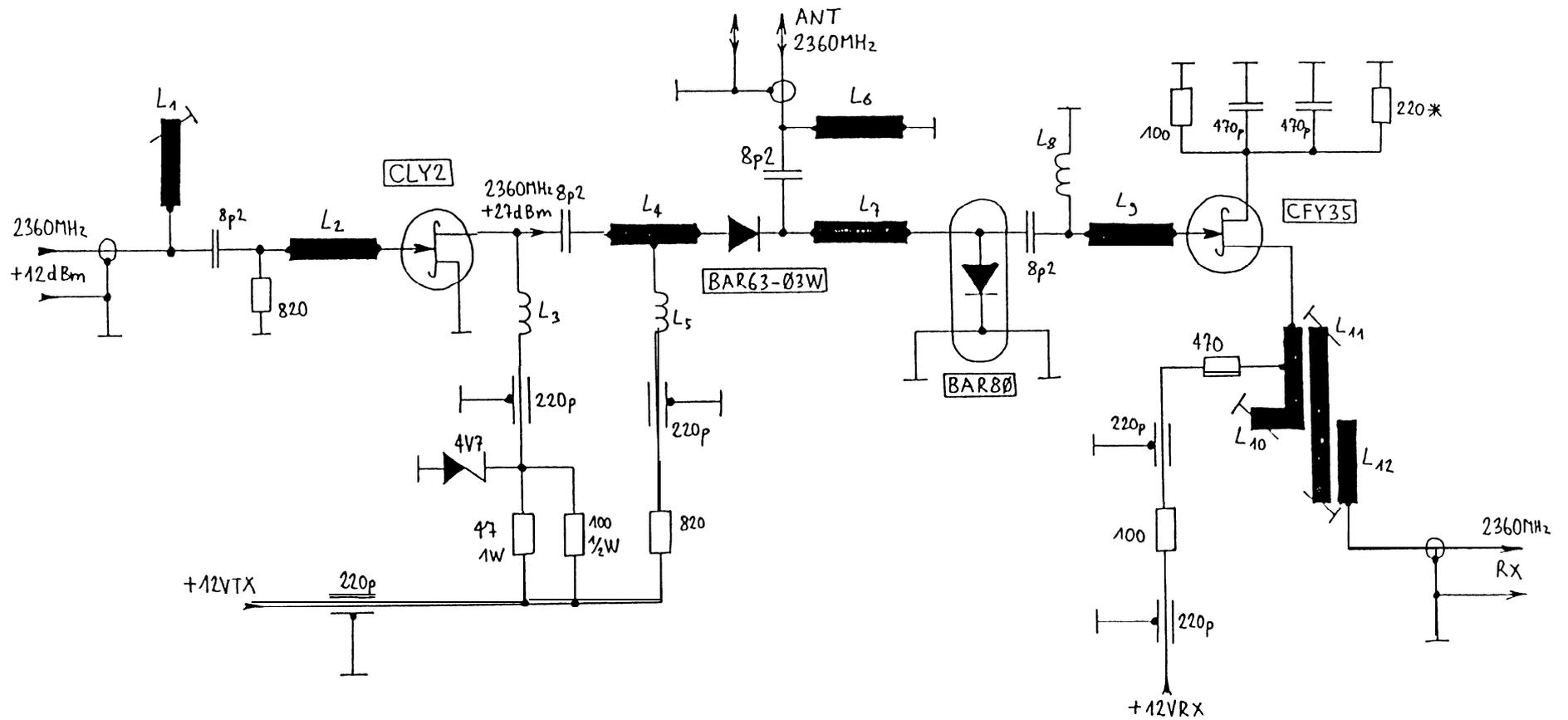


Fig. 8 - RF front-end circuit diagram.

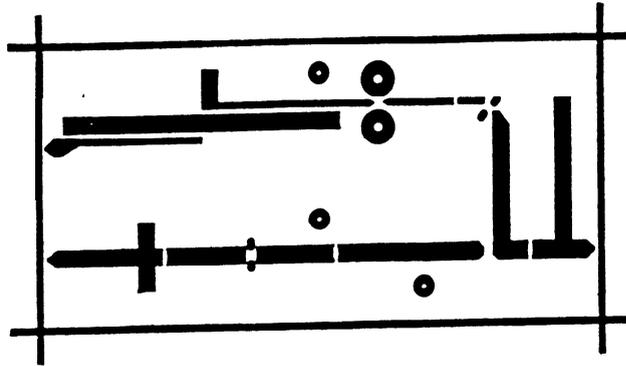


Fig. 9 - RF front-end PCB (double-sided 0.8 mm glassfiber- teflon, microstrip circuit with a continuous ground plane).

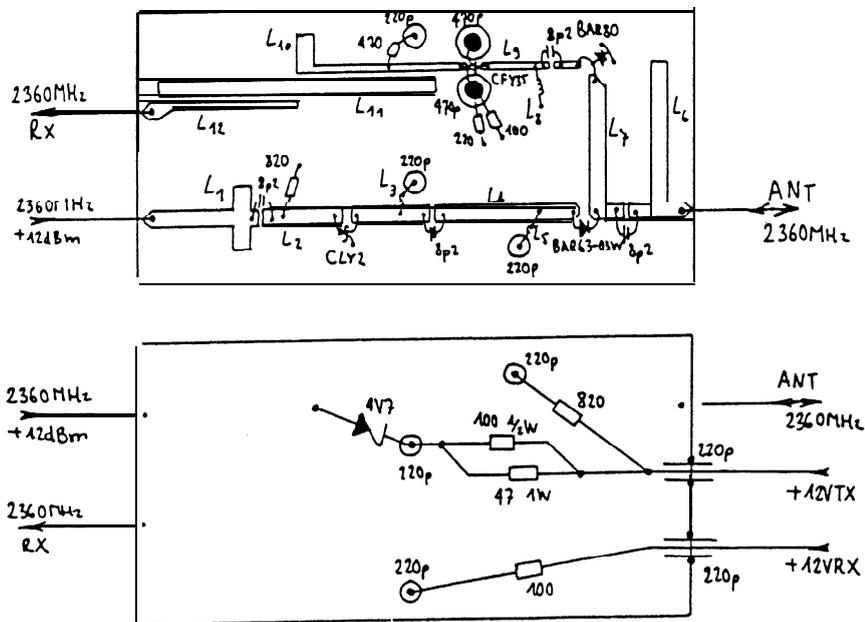
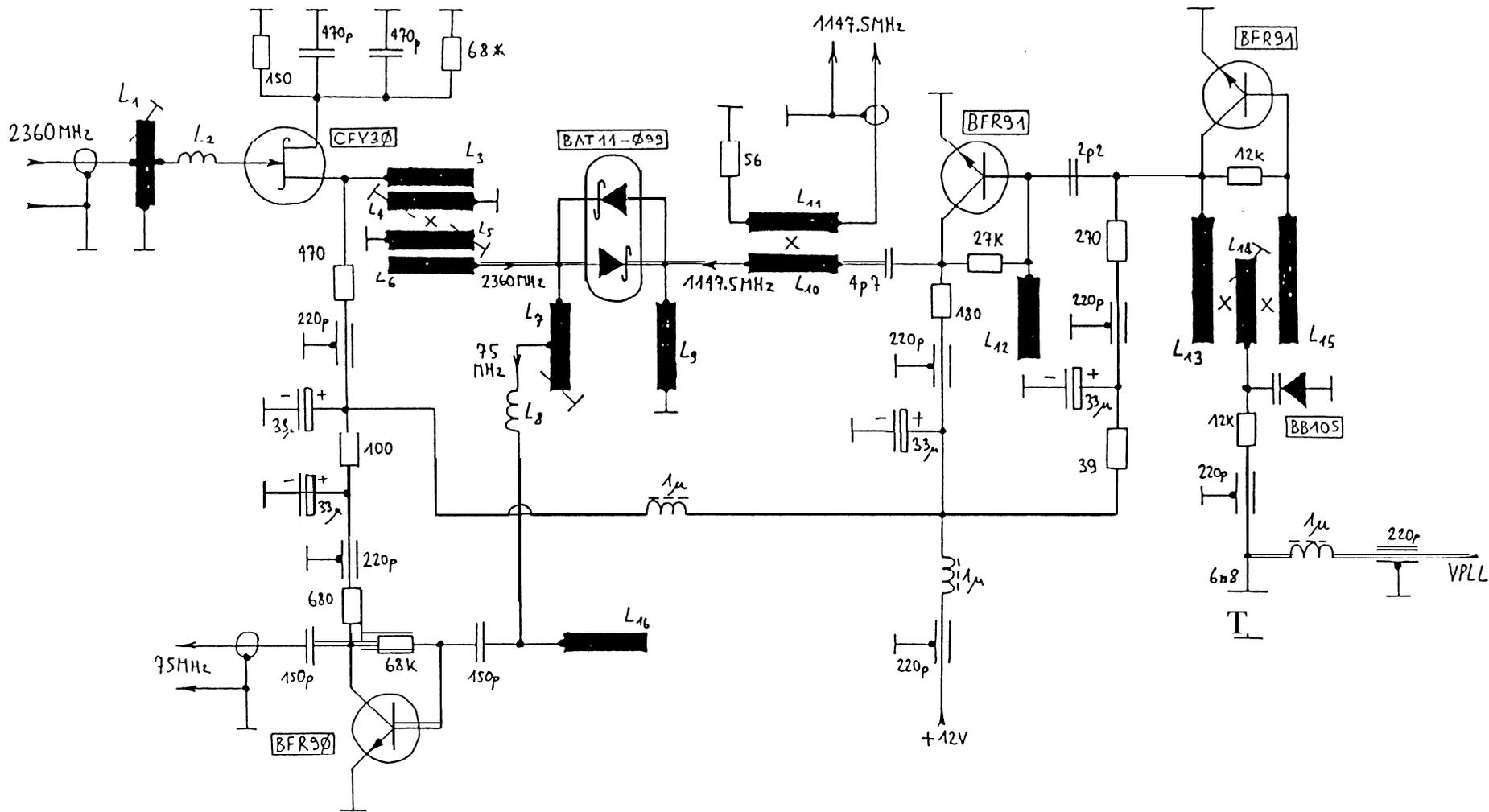


Fig. 10 - RF front-end component location.



163 Fig. 11 - RX converter, analog section circuit diagram.

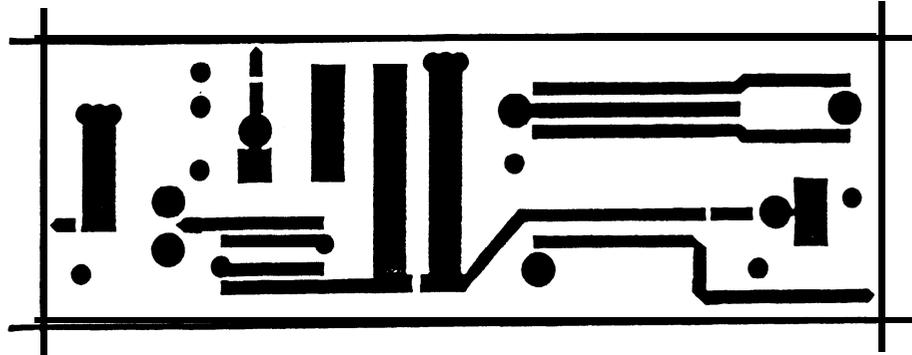


Fig. 12 - RX converter, analog section PCB (double-sided 1.6 mm glassfiber-epoxy, microstrip circuit with a continuous groundplane).

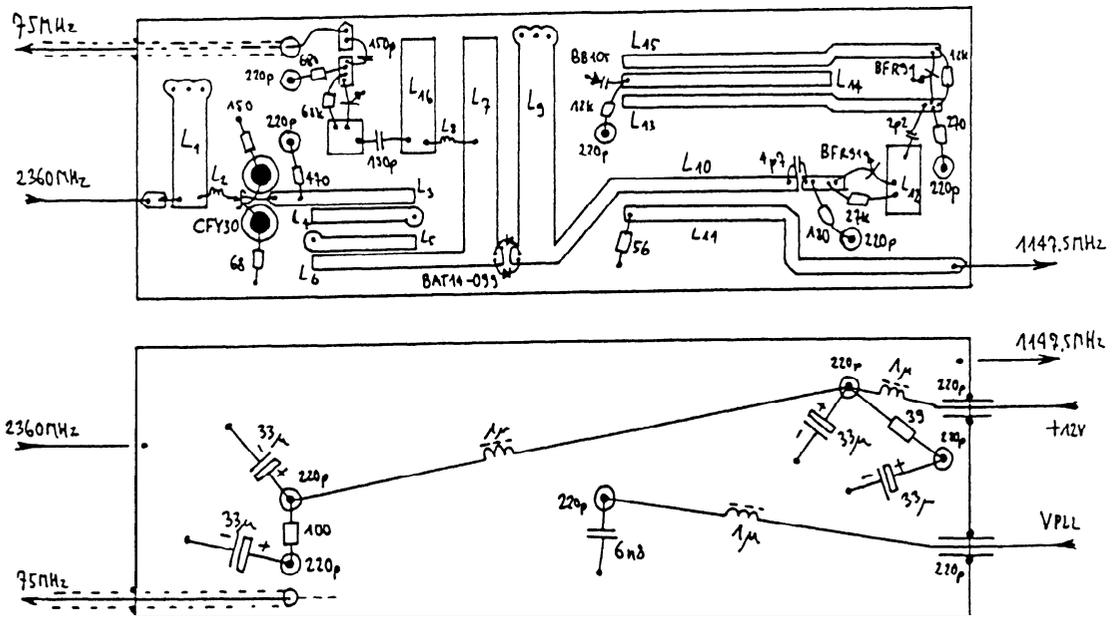


Fig. 13 - RX converter, analog section component location.

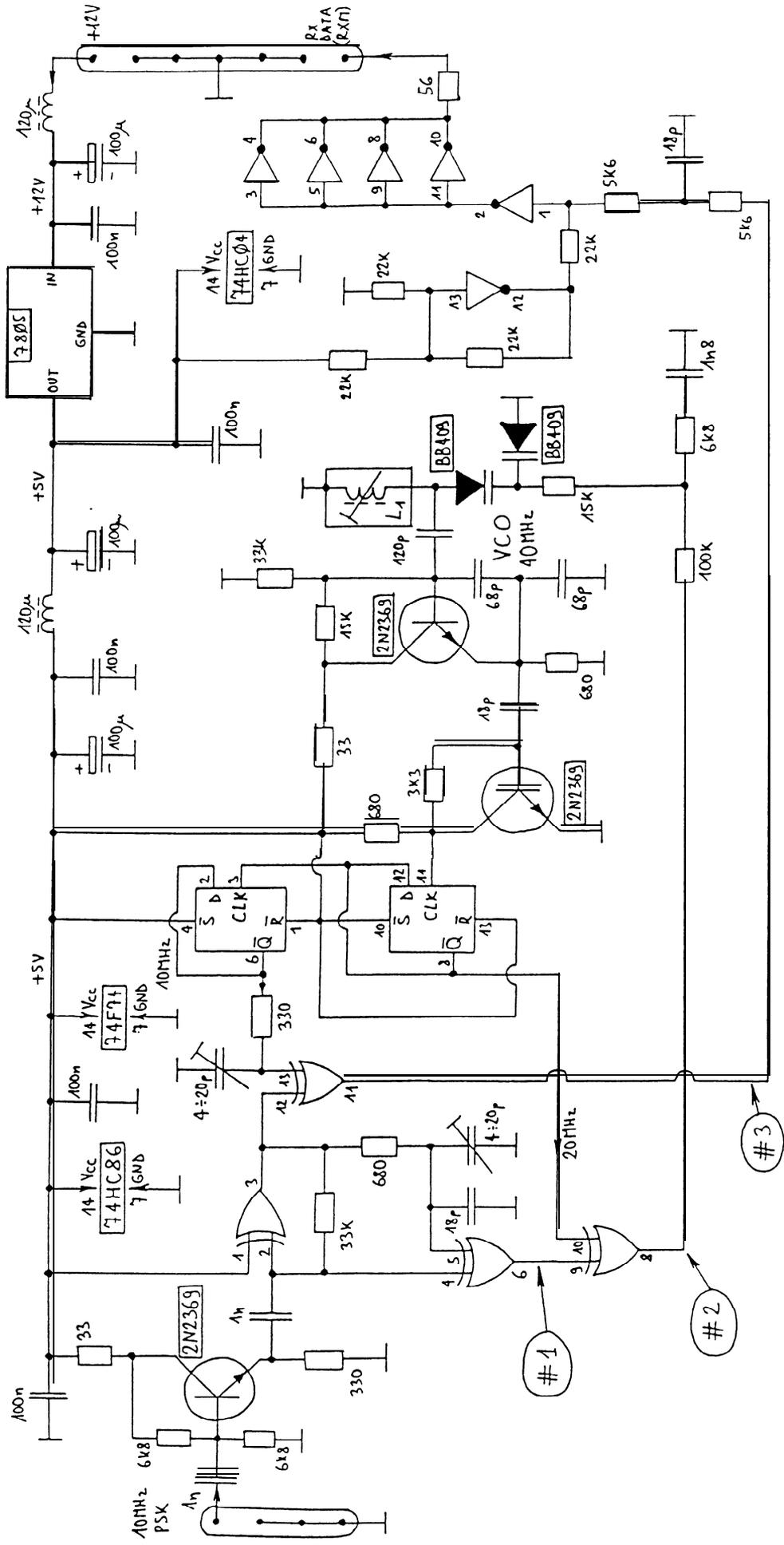


Fig. 20 - PSK demodulator circuit diagram.

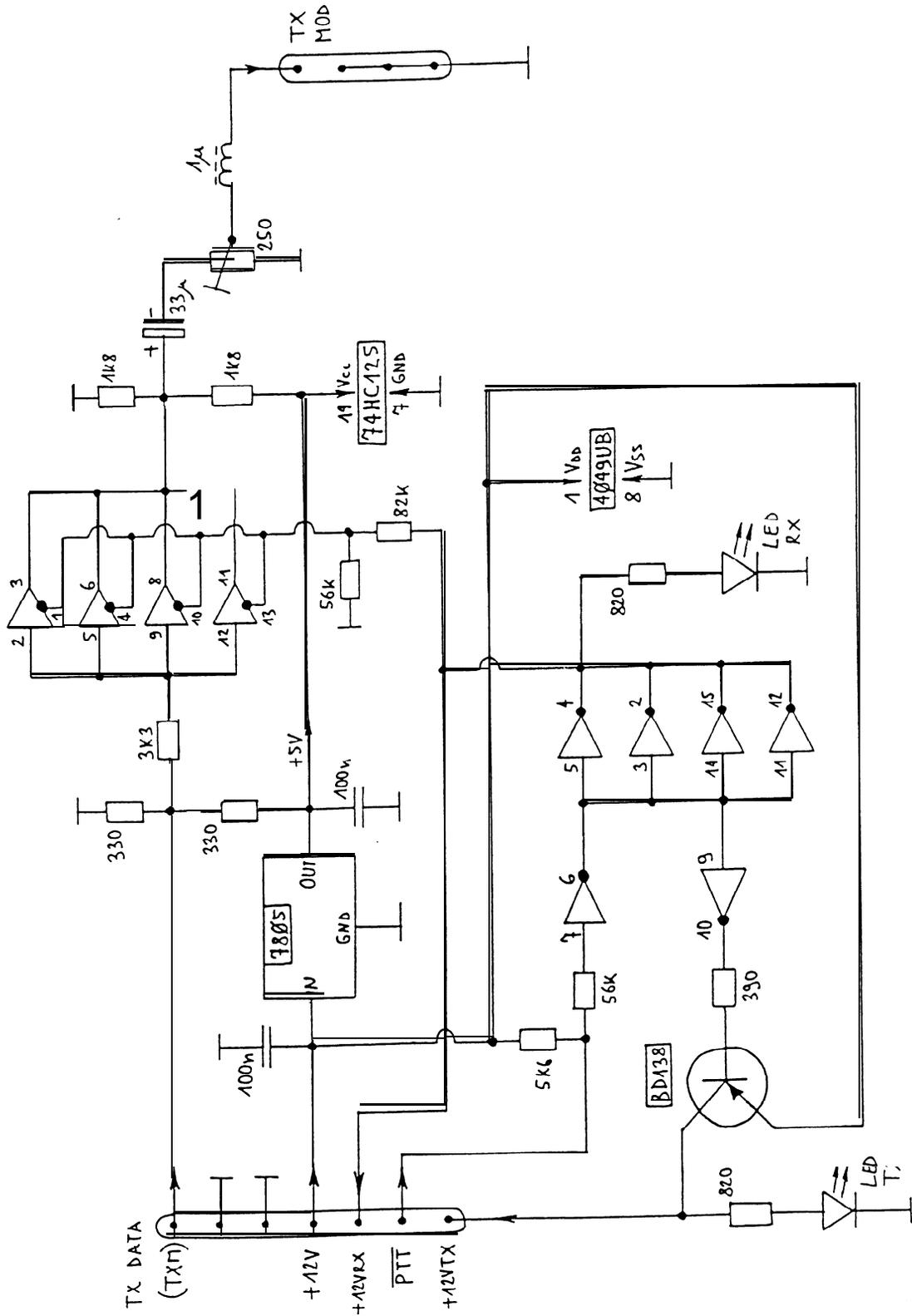


Fig. 23 - Supply switch interface circuit diagram.

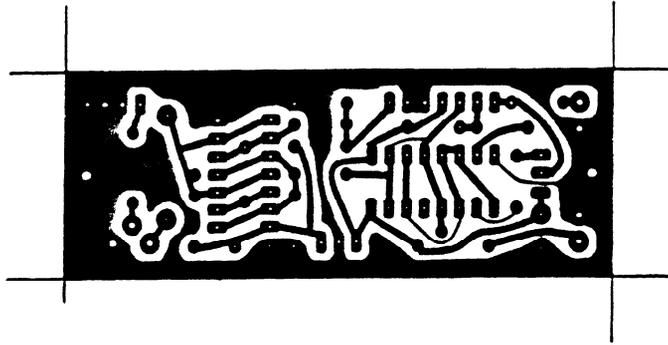


Fig. 24 - Supply switch interface PCB (single-sided 1.6 mm glassfiber - epoxy).

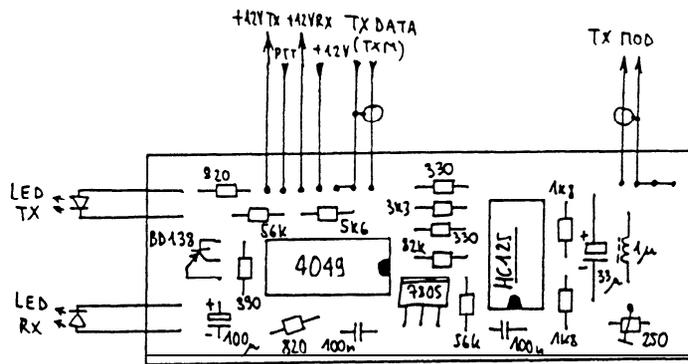


Fig. 25 - Supply switch interface component location.

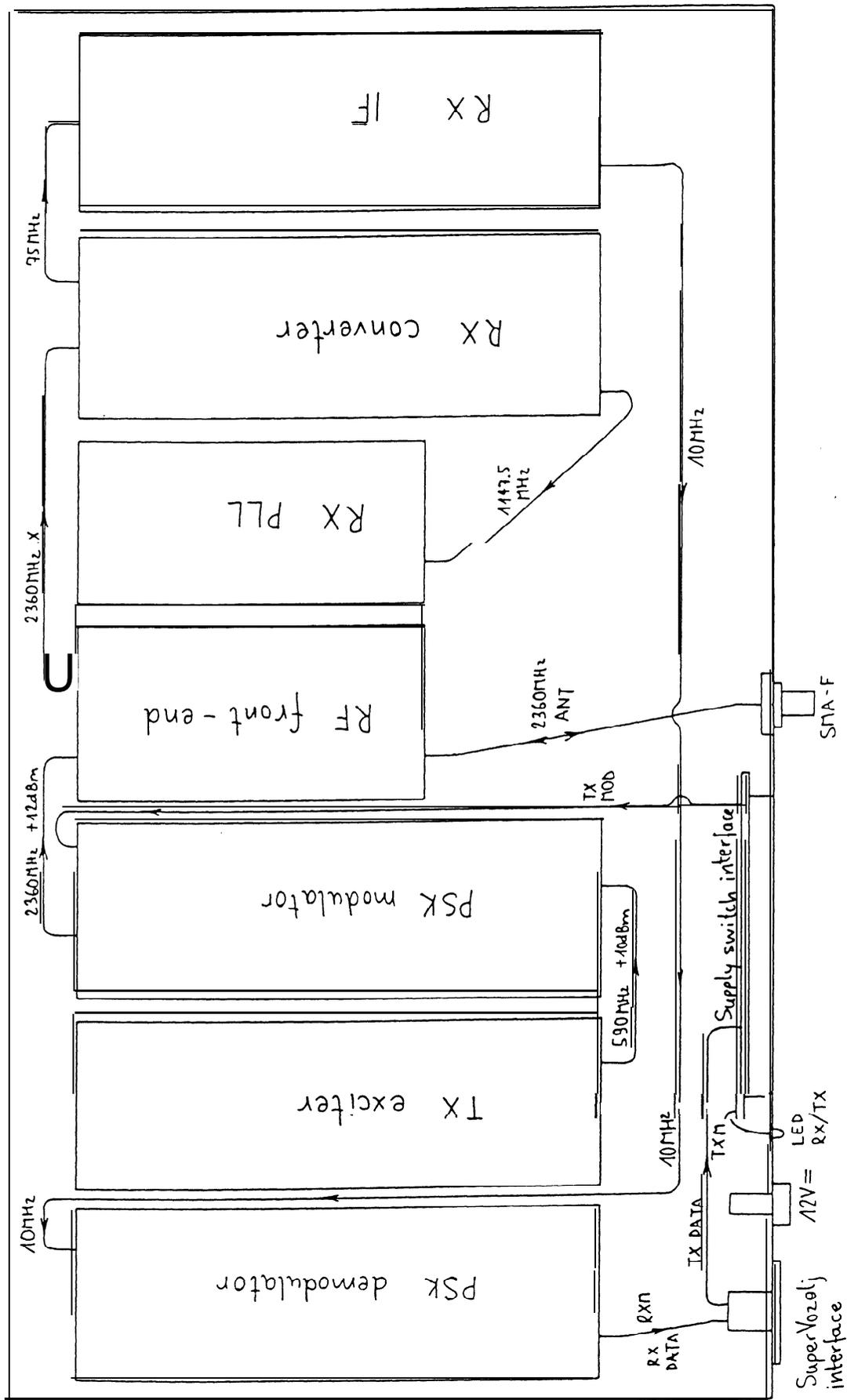


Fig. 26 - 13cm PSK transceiver module location.

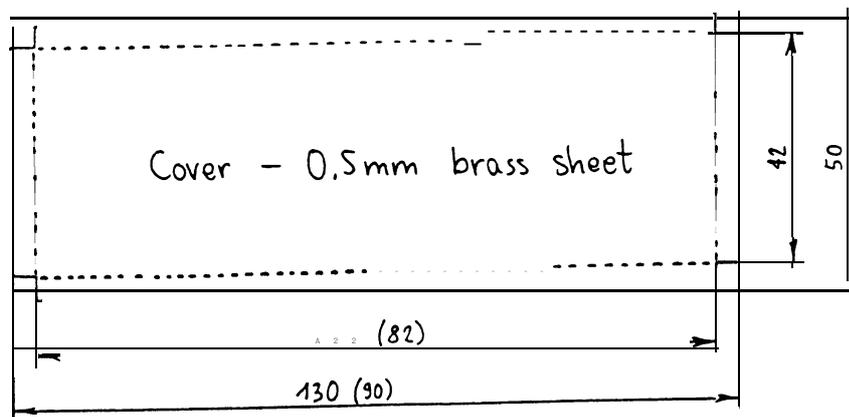
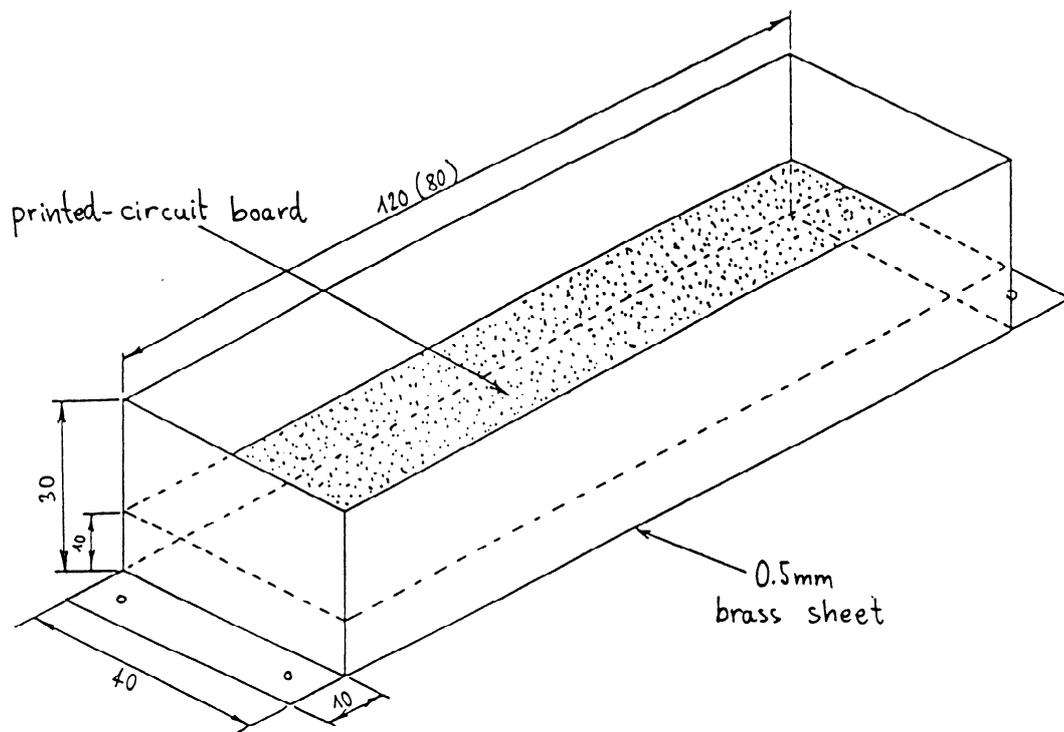


Fig. 27 - 13cm PSK transceiver shielded module enclosure.

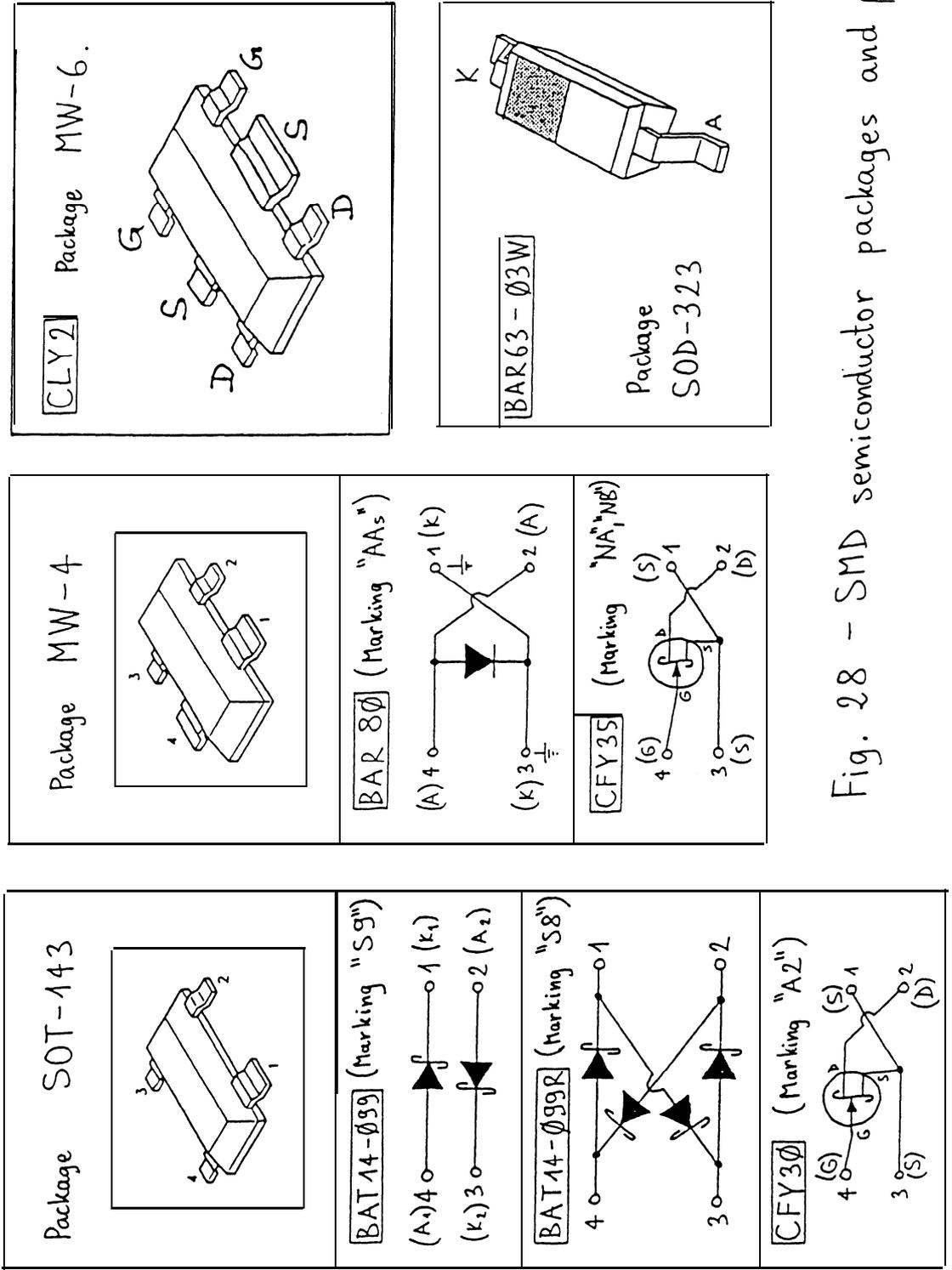


Fig. 28 - SMD semiconductor packages and pinouts.

23cm PSK packet-radio RTX for 1.2Mbit/s user access

Matjaz Vidmar, S53MV

1. Why biphase PSK modulation?

Upgrading the packet-radio network to higher data-rates also requires using more efficient modulation and demodulation techniques both to reduce the signal bandwidth and to increase the radio range of the system. In particular, inefficient modems coupled to standard FM transceivers have to be replaced with custom-designed radios for data transmission. Considering the bandwidth and TX power available to radio-amateurs, it is necessary to switch to coherent demodulation techniques at data-rates around 100kbit/s in terrestrial packet-radio and at even lower data-rates in satellite communications.

One of the simplest forms of digital modulation, that can be demodulated in a coherent way, is biphase PSK. The usual amateur approach to implement biphase PSK is to use already existing equipment like linear transverters or SSB transceivers coupled to custom-designed modems operating at an intermediate frequency. While this approach may be acceptable for satellite work, it is rather complex and inconvenient for conventional terrestrial packet-radio.

On the other hand, professionals developed very simple and efficient digital radios like GSM cellular telephones. Professionals also found out that they can not use the frequency spectrum efficiently with narrowband FM radios: all new cellular phone system use high-speed TDMA techniques or even spread-spectrum modulation. If we radio-amateurs want to improve our digital communication, it is therefore necessary to develop and build new equipment. The only place for obsolete narrowband FM equipment is a museum!

Maybe PSK modulation is not considered very efficient by many amateurs, since it is used on satellites at data rates of only 400bit/s or 1200bit/s. On the other hand, in Slovenia (S5) we installed our first 1.2Mbit/s PSK links in 1995, operating in the 13cm amateur band at 2360MHz. This equipment resulted very reliable and the PSK links never failed, even when the 70cm and 23cm 38.4kbit/s links were out due to heavy snowfall in the 1995/96 winter.

The 13cm PSK 1.2Mbit/s link transceiver used in these links (shown in Weinheim in September 1995) was only the first attempt towards a dedicated PSK radio. The 13cm transmitter was simplified by using direct PSK modulation on the output frequency, but the 13cm receiver is still using a double downconversion followed by a conventional IF squaring-loop PSK demodulator. The construction of this transceiver is not simple: there are several shielded modules and especially the double-conversion receiver requires lots of tuning.

2. Direct-conversion PSK data transceiver

Similarly to a SSB transceiver, a PSK transceiver can also be built as a direct-conversion radio as shown on fig. 1. The Costas-loop demodulator can be extended to include most of the amplification in the receiving chain. Since such a receiver does not require narrow bandpass filters, the construction and alignment can be much simplified. In addition, some receiver stages can also be used in the transmitter (like the local oscillator chain) to further simplify the overall transceiver.

A direct-conversion PSK receiver also has some problems. Limiting is generally not harmful in the signal amplifier, however it increases the noise in the error amplifier chain. In practice the loop bandwidth has to be decreased, if no AGC is used and both amplifiers operate in the limiting regime. It is also very difficult

to have both amplifiers DC coupled as required by the theory. If AC coupled amplifiers are used, randomization (scrambling) has to be applied to the data stream and some additional noise is generated. However, in a well-designed, direct-conversion PSK receiver the signal-to-noise ratio degradation due to AC coupling can be kept sufficiently small.

Building a real-world, direct-conversion PSK receiver one should also consider other unwanted effects. For example, the Costas-loop demodulator includes very high-gain stages. Unwanted effects like AM modulation on the VCO or FM-to-AM conversion in the multiplier stages can lead to unwanted feedback loops. However, the most critical component seems to be the VCO.

In a practical microwave PSK transceiver the VCO is built as a VCXO followed by a multiplier chain. Although the static frequency-pulling range of fundamental-resonance and third-overtone crystals is sufficient for this application, their dynamic response is totally unpredictable above 1kHz. The latter may be enough for full-duplex, continuous-carrier microwave links, but it is insufficient for CSMA packet-radio, where a very fast signal acquisition is required.

3. Zero-IF PSK data transceiver

Most of the problems of a direct-conversion PSK receiver can be overcome in a so called “zero-IF” PSK receiver, as shown on fig.2. Incidentally, a zero-IF PSK transceiver requires very similar hardware to a direct-conversion PSK transceiver. The main difference is in the local oscillator. A zero-IF PSK receiver has a fixed-frequency, free-running local oscillator, while the demodulation is only performed after the main receiver gain stages.

A zero-IF PSK receiver includes a quadrature mixer that provides two output signals I' and Q' with the same bandwidth as in a direct-conversion RX. The signals I' and Q' contain all of the information of the input RF signal, but they do not represent the demodulated signal yet. Since the zero-IF RX contains a free-running LO, its phase is certainly not matched to the transmitter. Further, if there is a difference between the frequencies of the transmitter and of the receiver, the phasor represented by the I' and Q' signals will rotate at a rate corresponding to the difference of the two frequencies.

To demodulate the information, the I' and Q' signals have to be fed to a phase shifter to counterrotate the phasor. The phase shifter is kept synchronized to the correct phase and rate by a Costas-loop feedback. Since the whole Costas-loop demodulator operates at high signal levels and at relatively low frequencies, it can be built with inexpensive 74HCxxx logic circuits that require no tuning at all!

A zero-IF PSK receiver requires linear amplification of the I' and Q' signals. Limiting of the I' and Q' signals is very harmful to the overall signal-to-noise ratio. If the zero-IF amplifiers are AC coupled, data randomization (scrambling) is required. On the other hand, a zero-IF PSK transceiver does not include any critical stages or unstable feedback loops and is therefore easily reproducible.

Searching for a simple PSK transceiver design I attempted to build both a direct-conversion and a zero-IF PSK transceiver for 23cm. The 23cm band offers sufficient bandwidth for 1.2Mbit/s operation. Further, the whole transceiver can be built on conventional, inexpensive glassfiber-epoxy laminate FR4. Finally, the propagation losses without optical visibility are smaller in the 23cm band than at higher microwave frequencies.

A direct-conversion PSK transceiver for 23cm resulted very simple. The signal and error amplifiers used just one LM311 voltage comparator each, operating as a limiting amplifier. The only limitation of this transceiver was the VCXO. Due to the undefined dynamic response of the VCXO, the capturing range of the Costas-loop RX was only about +/-5kHz. Further, even this figure was hardly reproducible, since even two crystals from the same manufacturing batch had a quite different dynamic response in the VCXO.

A zero-IF 23cm PSK transceiver resulted slightly more complex, due to the linear IF amplification with AGC and the additional Costas-loop demodulator. On the other hand, the zero-IF 23cm PSK RTX resulted fully reproducible, since there are no critical parts or unstable circuits built in. Since the additional complexity of the zero-IF RTX is in the IF part, using only cheap components and no tuning points, it does not add much to the overall complexity of the transceiver.

4. Design of the zero-IF 23cm PSK transceiver

In this article I am therefore going to describe the abovementioned successful design of a zero-IF PSK data transceiver. The transceiver is built on seven printed-circuit boards, four of which (the RF part) are installed in metal shielded enclosures. The RF part is built mainly as microstrip circuits on **0.8mm** thick glassfiber-epoxy laminate FR4.

Subharmonic mixers are used both in the transmitter modulator and in the receiver quadrature mixer. Subharmonic mixers with two antiparallel diodes are simple to build. Since the LO signal is at half of the RF frequency, RF signals are easier to decouple and less shielding is required. Finally, it is very easy to build two identical subharmonic mixers for the receiver quadrature mixer.

The whole transceiver therefore requires a single local oscillator operating at half of the RF frequency or at about 635MHz for operation in the 23cm amateur band. The local oscillator including a crystal oscillator and multiplier stages is shown on fig.3. The LO module is built on a single-sided PCB, as shown on fig.4 and fig.5.

To speed-up the TX/RX switching, the receiving mixers are powered on and are receiving the LO signal all of the time. On the other hand, the LO signal feeding the modulator has to be turned off to avoid any interference during reception. Therefore the LO signal is fed to the receiving mixers through a directional coupler located in the 1270MHz PSK modulator module as shown on fig.6.

Only a small fraction of the LO power (-20dB) is fed to a separation amplifier stage (BFP183). The 635MHz BPF ensures a good residual carrier suppression (>30dB) in the PSK modulator. The 1.27GHz BPF is used to suppress the 635MHz LO signal and its unwanted harmonics. Finally, a two-stage MMIC amplifier (INA-1 0386) is used to boost the signal level to +14dBm.

The 1270MHz PSK modulator is a microstrip circuit built on a double-sided PCB as shown on fig.7 and fig.8. The bottom side of the PCB is not etched to serve as a groundplane for the microstrip circuit. The RF signal losses in the FR4 laminate are rather high at 1.27GHz. For example, the 1.27GHz BPF has a **passband** insertion loss of about 5dB. On the other hand, all of the microstrip **bandpass** filters are designed for a bandwidth of more than 10% of the center frequency and therefore require no tuning considering the laminate and etching tolerances.

The RF front-end of the 23cm PSK transceiver, shown on fig.9, includes a TX power amplifier with a CLY5 power GaAsFET to boost the TX output power to about 1 W (+30dBm), a PIN diode antenna switch (BAR63-03 W and BAR80) and a receiving RF amplifier with a BFP18 1. The latter has about 15dB gain, but the following 1.27GHz BPF has about 3dB **passband** loss. The RF front-end is also built as a microstrip circuit on a double-sided PCB as shown on fig. 10 and fig. 11.

The quadrature I/Q mixer for 1270MHz, shown on fig. 12, includes an additional gain stage at 1.27GHz (26dB MMIC INA-03 184), two **bandpass** filters at 1.27GHz (3dB insertion loss each), a quadrature hybrid for the RF signal at 1.27GHz, an in-phase power splitter for the LO signal at 635MHz, two identical subharmonic mixers (two BAT1 4-099R schottky quads) and two identical IF preamplifiers (two BF 199).

Since the termination impedances of the subharmonic mixers depend on the LO signal power, the difference ports of both the quadrature (RF) and in-phase (LO) power splitters have to be terminated to ensure the correct phase and amplitude relationships. Considering the manufacturing tolerances of the

groundplane of the PCBs is soldered along all four sides to the brass frame to ensure a good electrical contact.

Special care should be devoted to the assembly of the microstrip circuits. The microstrip resonators are grounded at the marked positions using 0.6mm thick CuAg wire. The SMD components (shown on fig.26) are grounded through 2.5mm, 3.2mm or 5mm diameter holes at the marked positions. The holes are first covered with a piece of thin copper sheet on the groundplane side, then they are filled with solder and finally the SMD part is soldered in place.

The assembled PSK transceiver requires little tuning. The only module that needs to be tuned in any case is the local oscillator module. Since most of the stages are just frequency doublers, it is very difficult to tune this module to the wrong harmonic. The TX power amplifier may need some tuning to get the maximum output power. As printed on the circuit board, L1 in the RF power amplifier should not require any tuning if the interconnecting 50-ohm teflon cable from the modulator is exactly 12cm long. Tuning L3 and L6 the output power can only be increased by less than 100mW. All of the other microstrip resonators should not be tuned. Finally, the 2500hm trimmer in the supply switch interface is adjusted for the maximum TX output power (usually 2/3 of the full scale).

5. Interfacing the 1.2Mbit/s PSK transceiver

Amateur packet-radio interfaces for data-rates above 100kbit/s are not very popular. One of the most popular serial interfaces, the Zilog 28530 SCC, only includes a DPLL for RX clock recovery that can operate up to about 250kbit/s. Other integrated circuits, like the old Z80SIO, the MC68302 used in the TNC3 or the new MC68360 do not include any clock recovery circuits at all. In addition to the RX clock recovery, data scrambling/descrambling and sometimes even NRZ/NRZI differential encoding/decoding have to be provided by external circuits.

The circuit shown on fig.27 was specially designed to interface the described PSK transceiver to a 28530 SCC, although it will probably work with other serial HDLC controllers as well. The circuit includes an interpolation DPLL that only requires an 8-times higher clock frequency (9.8304MHz), although provides the resolution of a /256 conventional DPLL with a 315MHz clock.

The scrambler/descrambler uses a shift register with a linear feedback with EXOR gates. The scrambling polynomial is the same as the one used in K9NG/G3RUH modems: $1+X^{12}+X^{17}$. Due to the redundancy in the AX.25 data stream (zero insertion and deletion), a simple polynomial scrambler is completely sufficient to overcome the AC coupling limitation of the described PSK transceivers.

The interface circuit also includes 75-ohm line drivers and receivers, if the PSK transceiver is installed at some distance from the interface. However, connections have to be kept short on the side towards the computer serial port. The described interface only provides one clock signal, since it is intended for simplex operation with the described PSK transceiver. Of course the DPLL is disabled during transmission, so that the circuit supplies a stable clock to the transmitter. The polarity of the clock signal can be selected with a jumper. When using the 28530 RTxC or TRxC clock inputs, this jumper should be connected to ground.

The bit-synchronization/scrambler circuit is built on a single-sided PCB as shown on fig.28 and fig.29. It only requires one adjustment, the DCD treshold, and the latter can only be performed when noise is present on the RXM input.

List of figures:

- Fig. 1 - Direct-conversion PSK data transceiver.
- Fig. 2 - Zero-IF PSK data transceiver.
- Fig. 3 - 635MHz local oscillator.
- Fig. 4 - 635MHz LO PCB (0.8mm single-sided FR4).
- Fig. 5 - 635MHz LO component location.
- Fig. 6 - 1270MHz PSK modulator.
- Fig. 7 - 1270MHz PSK modulator PCB (0.8mm double-sided FR4).
- Fig. 8 - 1270MHz PSK modulator component location.
- Fig. 9 - 23cm PSK transceiver, RF front-end.
- Fig. 10 - RF front-end PCB (0.8mm double-sided FR4).
- Fig. 11 - RF front-end component location.
- Fig. 12 - Quadrature I/Q mixer for 1270MHz.
- Fig. 13 - Quadrature mixer PCB (0.8mm double-sided FR4).
- Fig. 14 - Quadrature mixer component location.
- Fig. 15 - I/Q dual amplifier with common AGC stages.
- Fig. 16 - I/Q dual amplifier PCB (1.6mm single-sided FR4).
- Fig. 17 - I/Q dual amplifier component location.
- Fig. 18 - Costas-loop I/Q PSK demodulator.
- Fig. 19 - Costas-loop demodulator PCB (1.6mm double-sided FR4).
- Fig. 20 - Costas-loop demodulator component location.
- Fig. 21 - Supply switch interface circuit diagram.
- Fig. 22 - Supply switch interface PCB (1.6mm single-sided FR4).
- Fig. 23 - Supply switch interface component location.
- Fig. 24 - 23cm PSK transceiver module location.
- Fig. 25 - 23cm PSK transceiver shielded module enclosure.
- Fig. 26 - SMD semiconductor packages and pinouts.
- Fig. 27 - Bit-synchronization/scrambler circuit diagram.
- Fig. 28 - Bit-sync/scrambler PCB (1.6mm single-sided FR4).
- Fig. 29 - Bit-sync/scrambler component location.

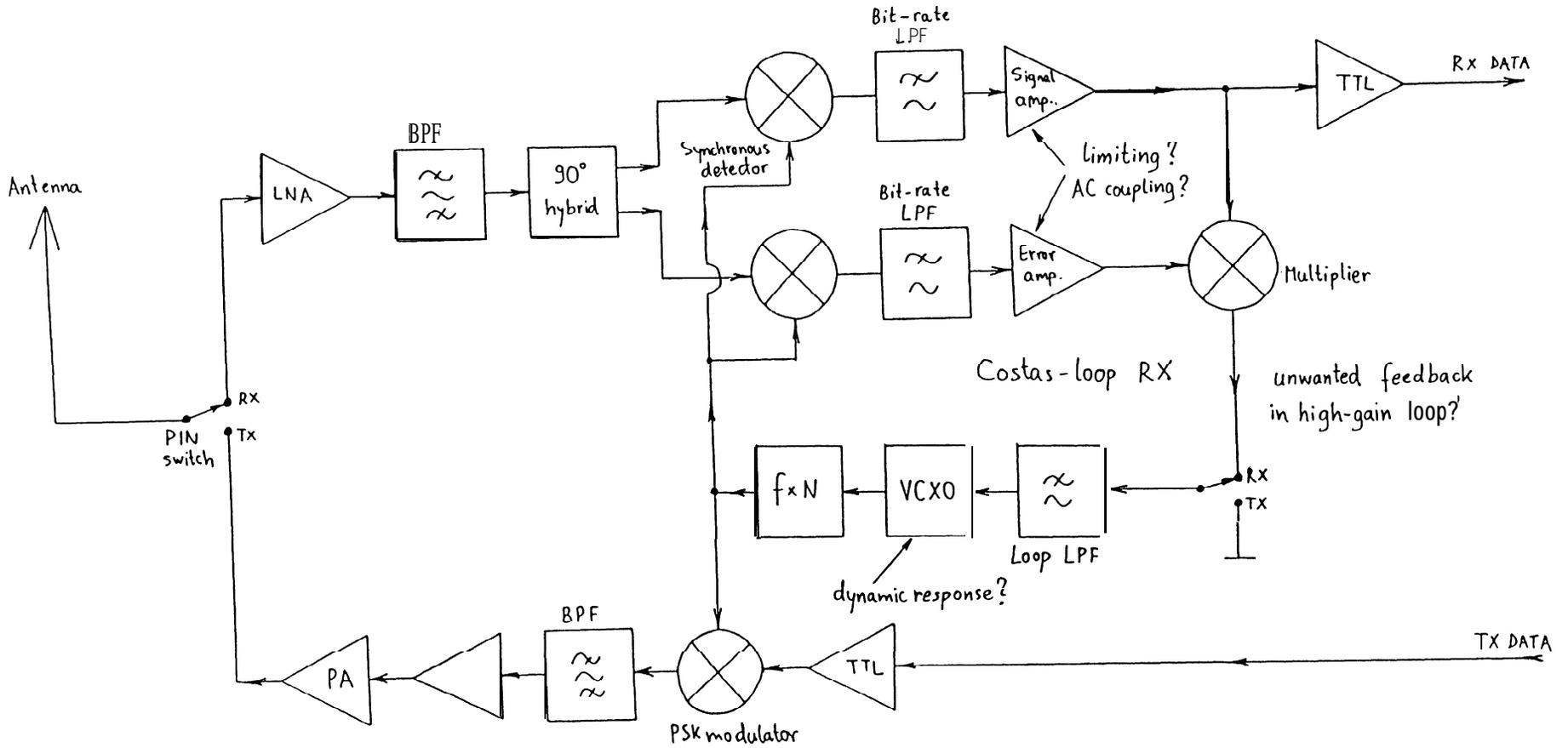


Fig 1 - Direct-conversion PSK data transceiver.

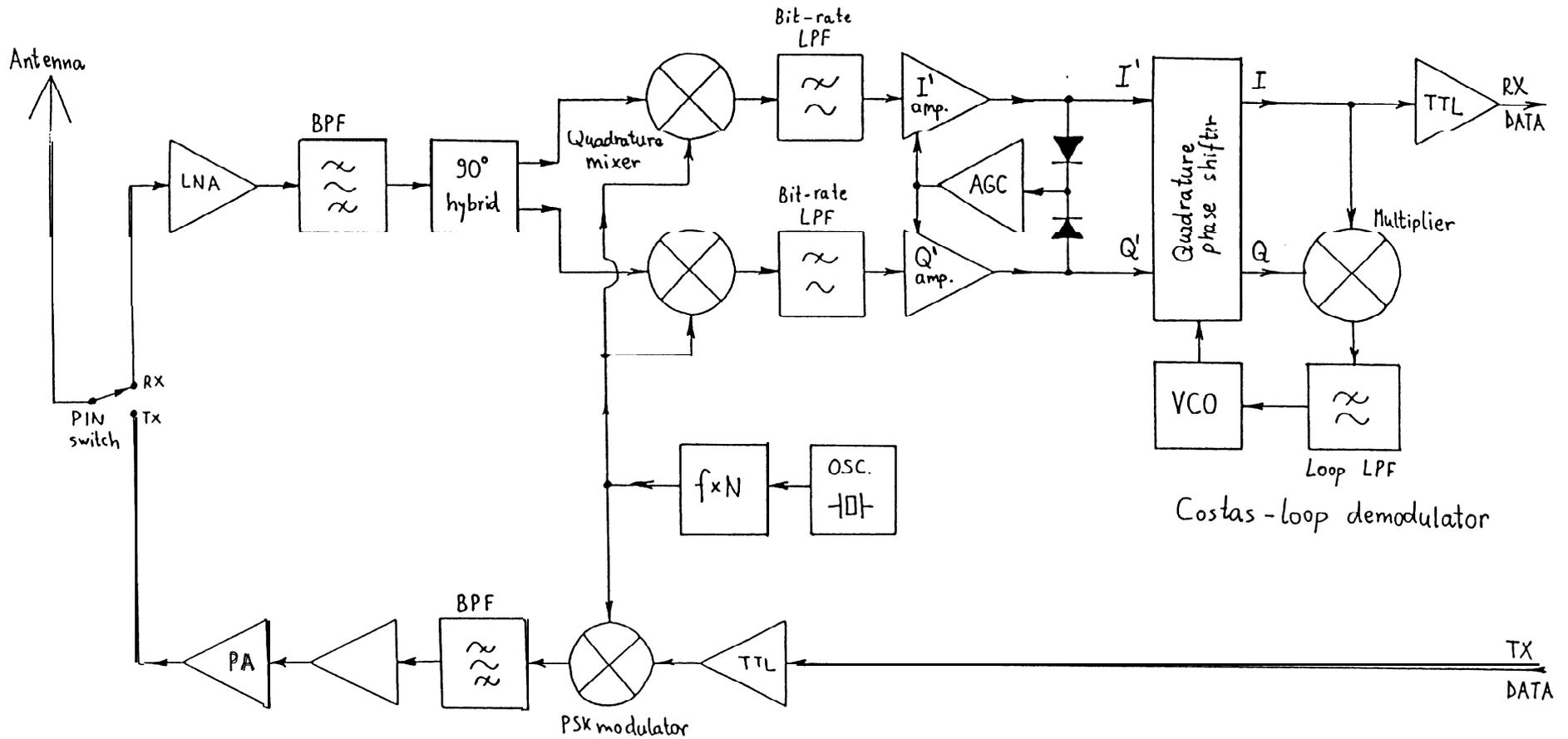


Fig. 2 - Zero-IF PSK data transceiver.

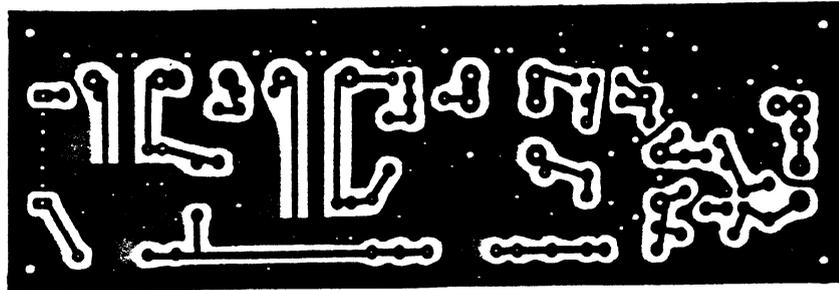


Fig. 4 - 635 MHz LO PCB (0.8mm single-sided FR4).

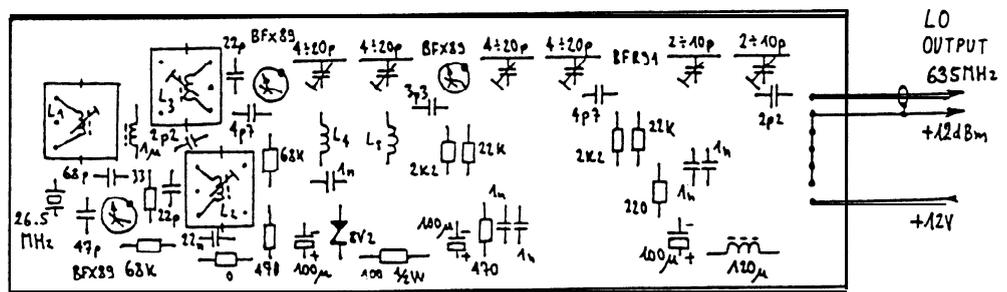


Fig. 5 - 635 MHz LO component location.

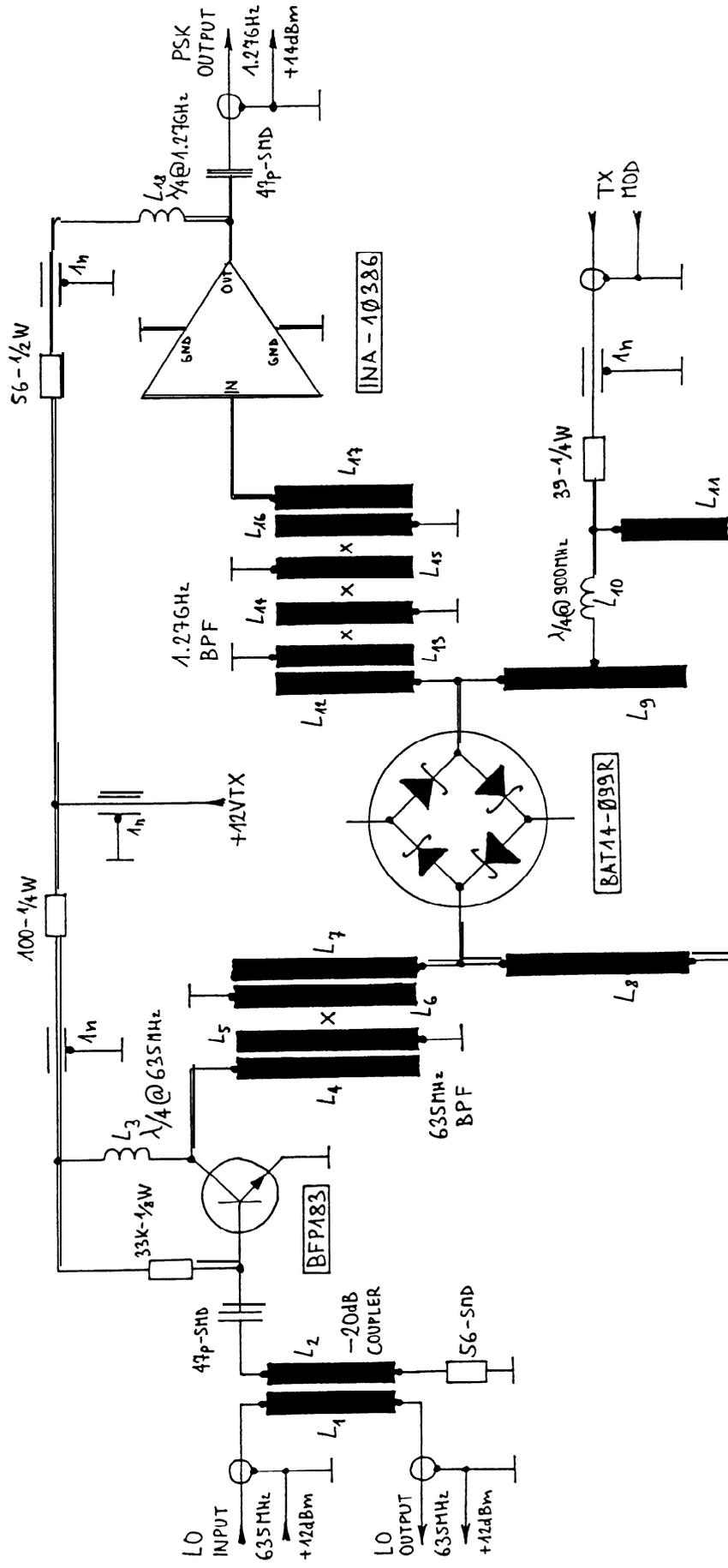


Fig. 6 - 1270MHz PSK modulator.

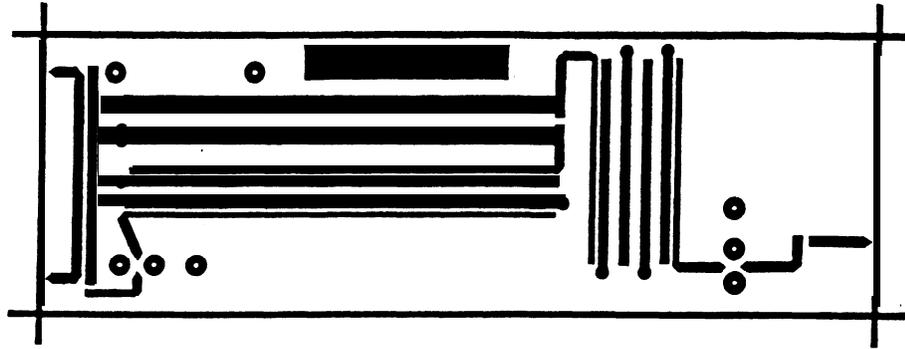


Fig. 7 - 1270 MHz PSK modulator PCB (0.8mm double-sided FR4).

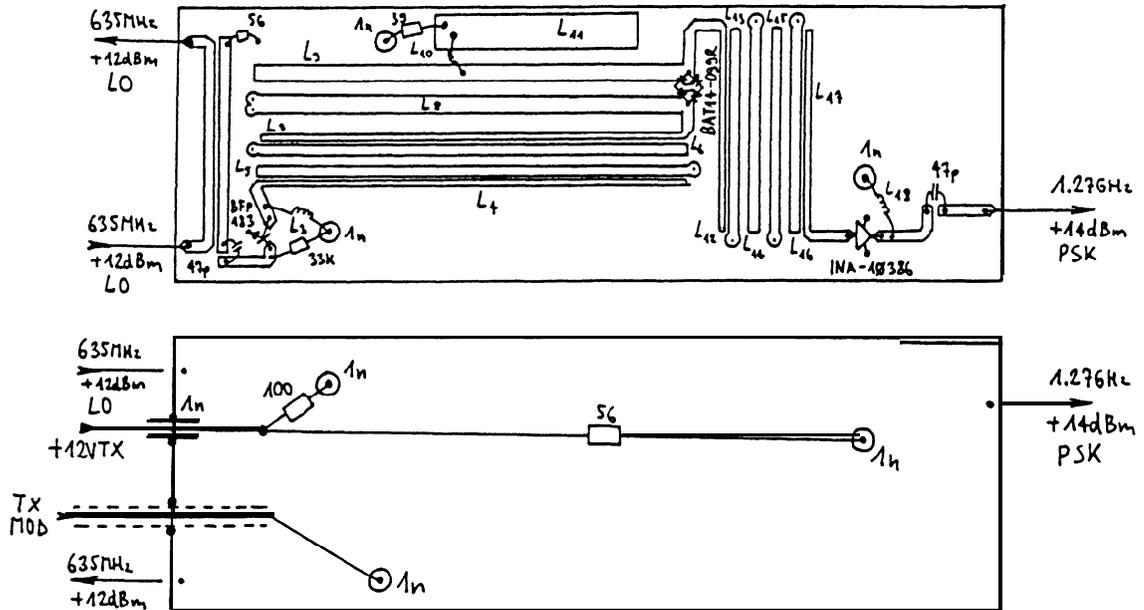


Fig. 8 - 1270 MHz PSK modulator component location.

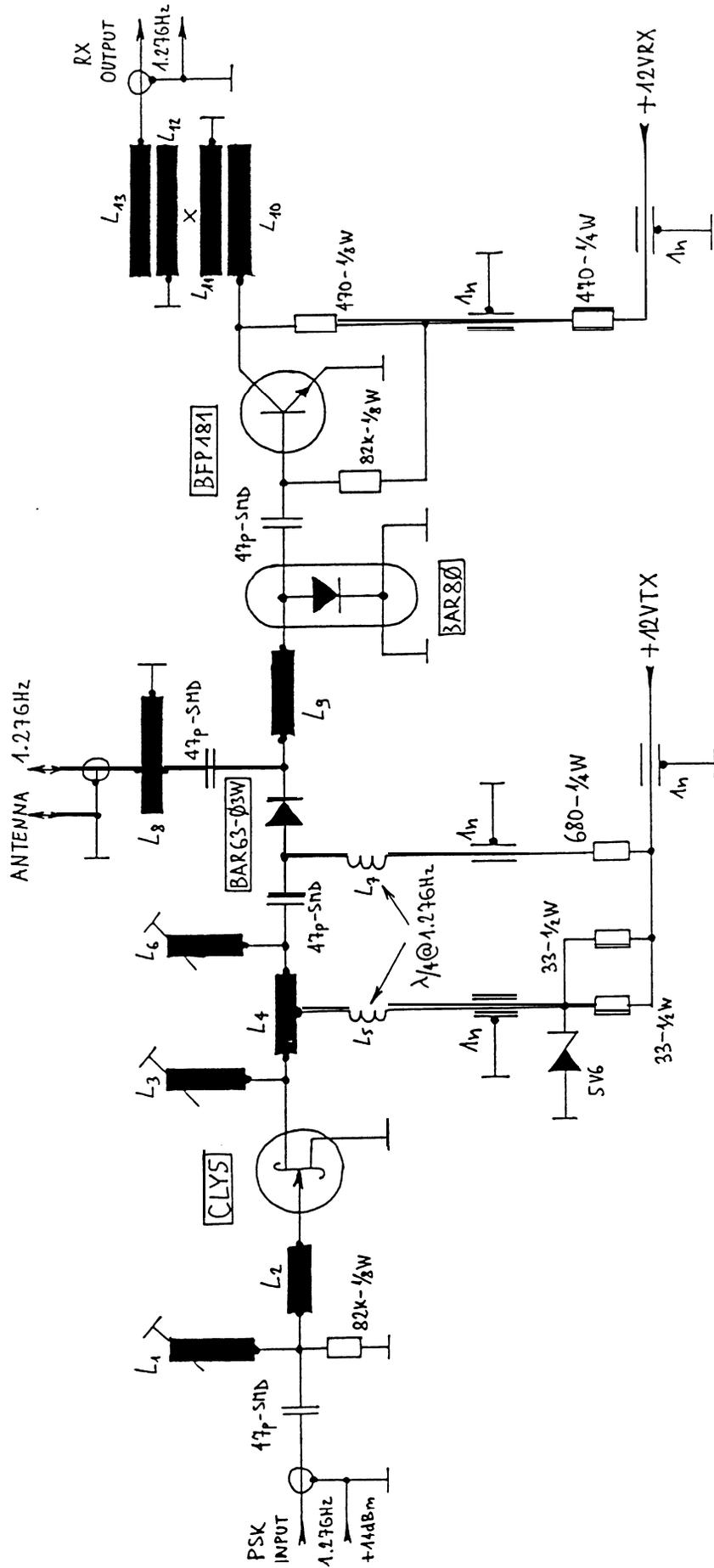


Fig. 9 - 23cm PSK transceiver , RF front-end.

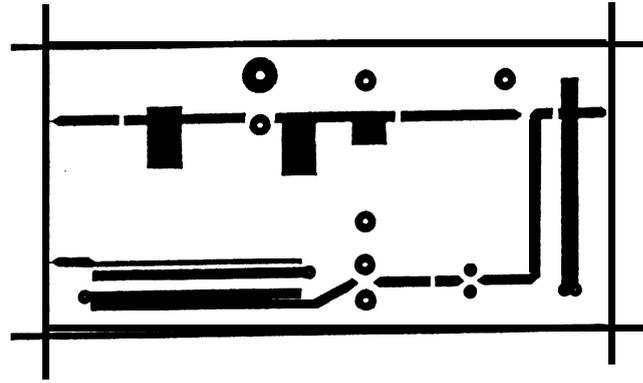


Fig. 10 - RF front-end PCB (0.8mm double-sided FR4).

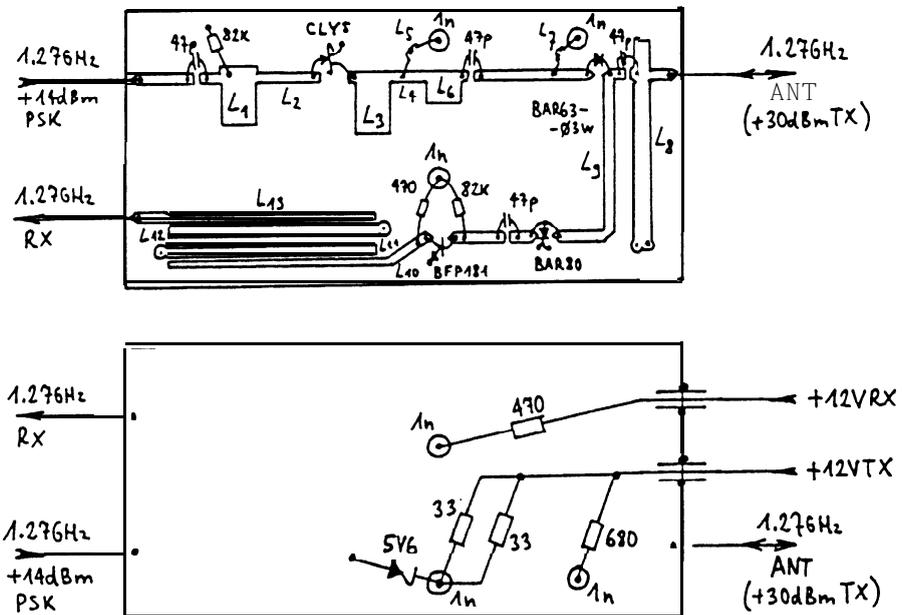


Fig. 11 - RF front-end component location.

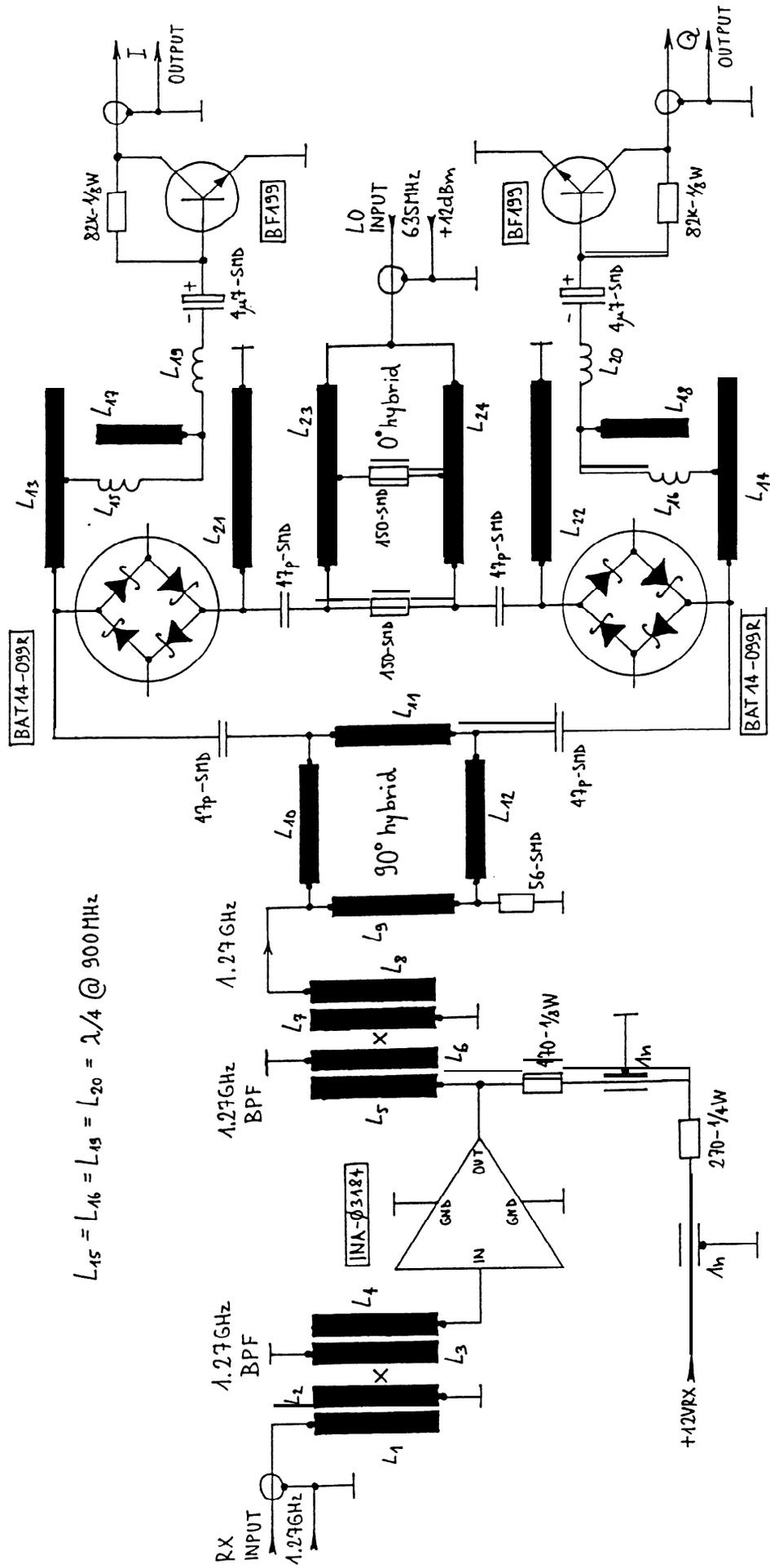


Fig. 12 - Quadrature I/Q mixer for 1270 MHz.

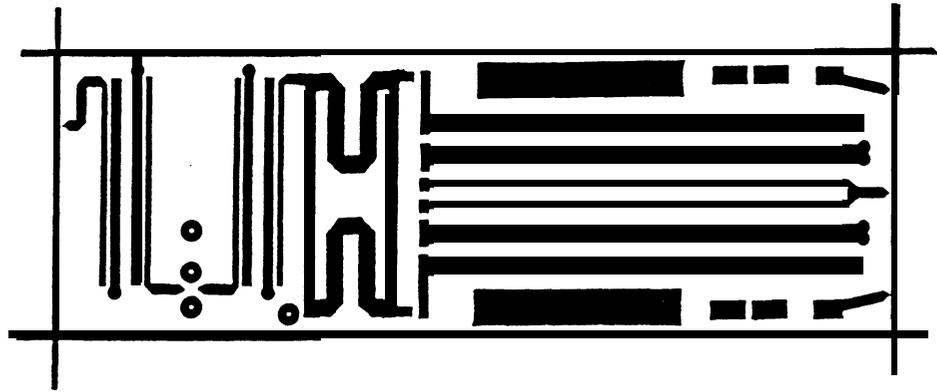


Fig. 13 - Quadrature mixer PCB (0.8mm double-sided FR4).

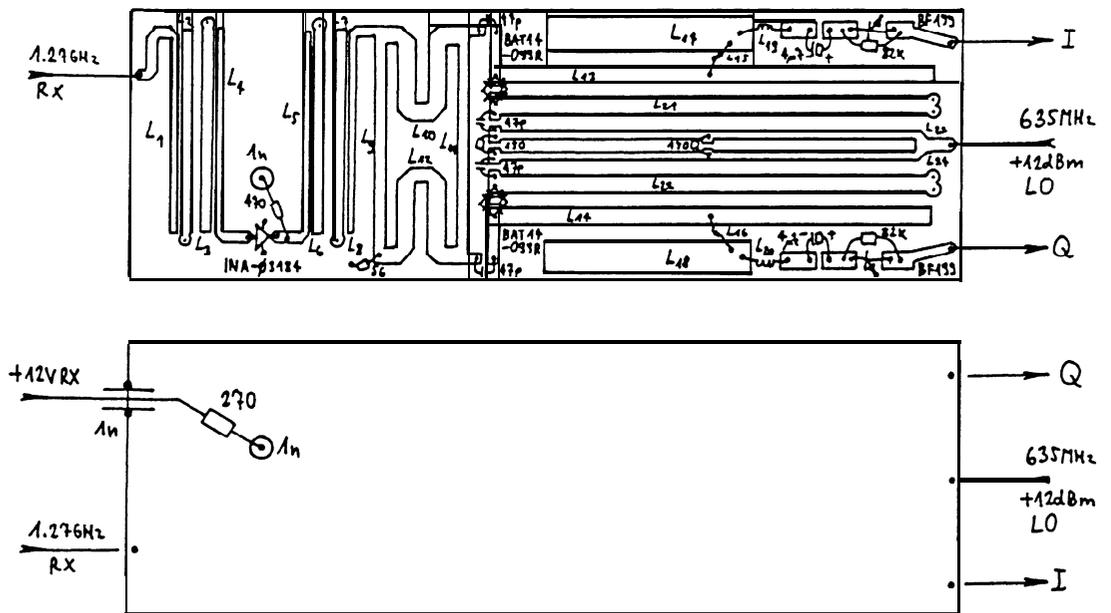


Fig. 14 - Quadrature mixer component location.

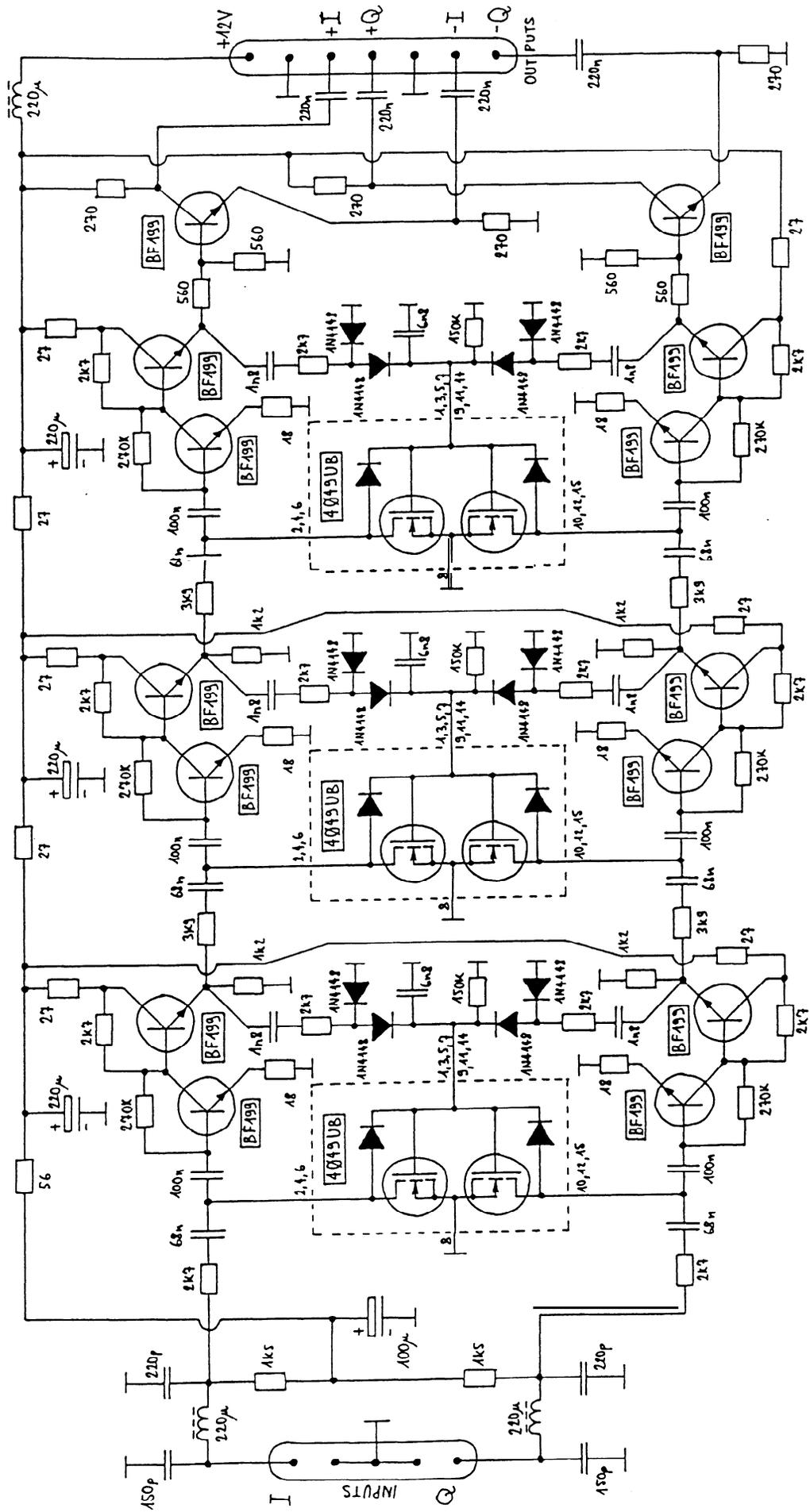


Fig 15 - I/Q dual amplifier with common AGC stages

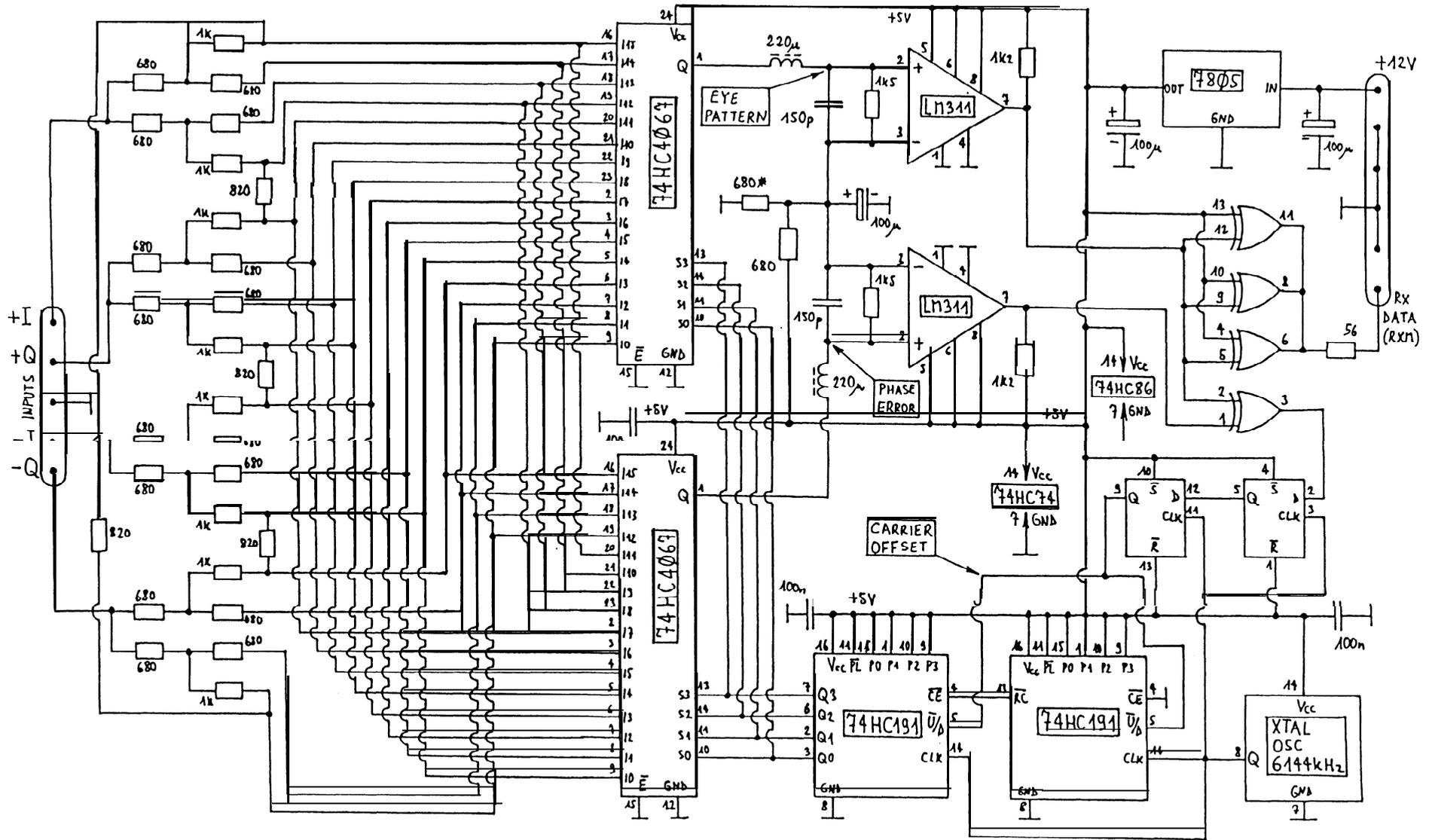


Fig. 18 - Costas-loop I/Q PSK demodulator.

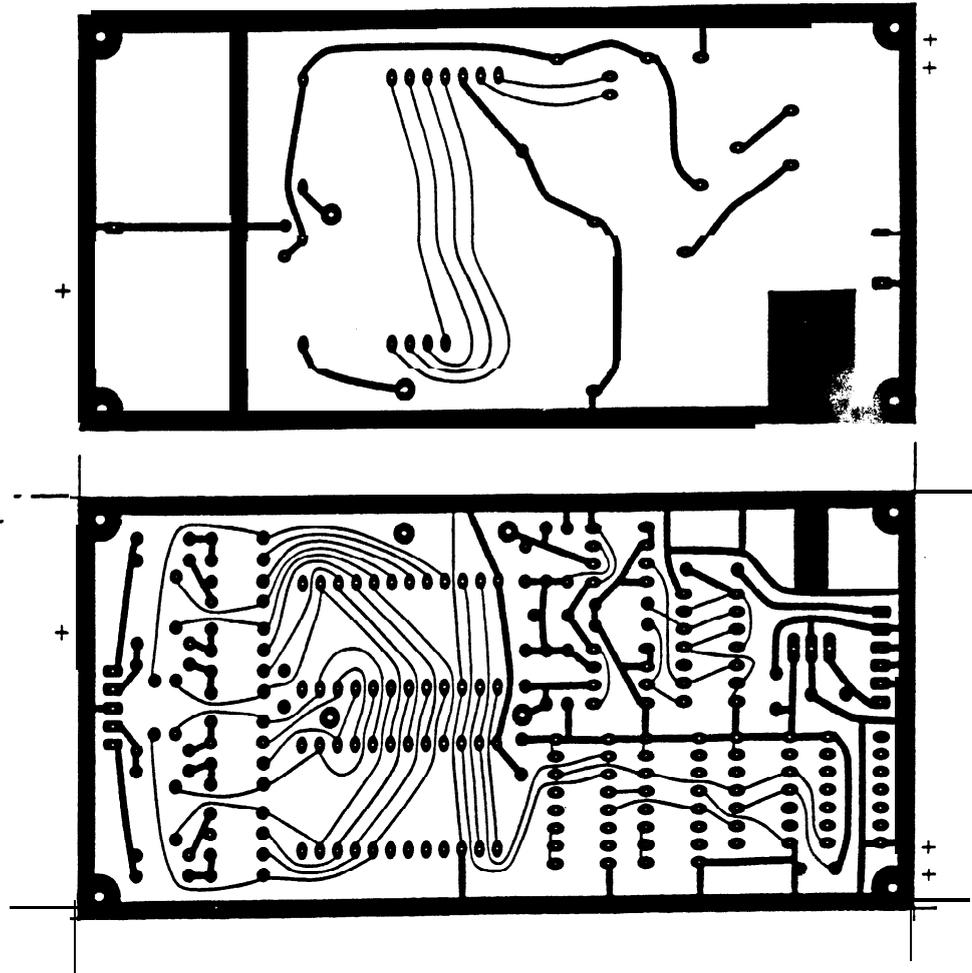


Fig. 19 - Costas-loop demodulator PCB (1.6mm double-sided FR4).

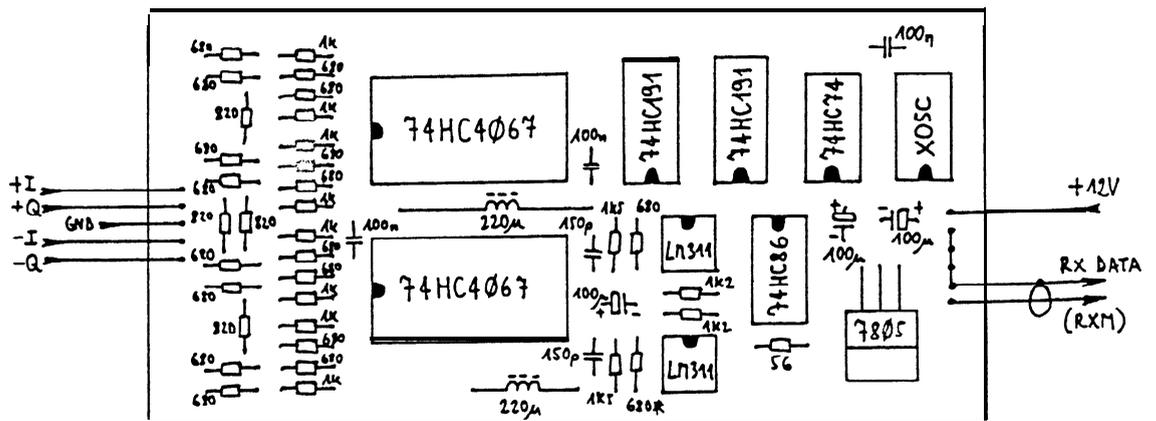


Fig. 20 - Costas-loop demodulator component location.

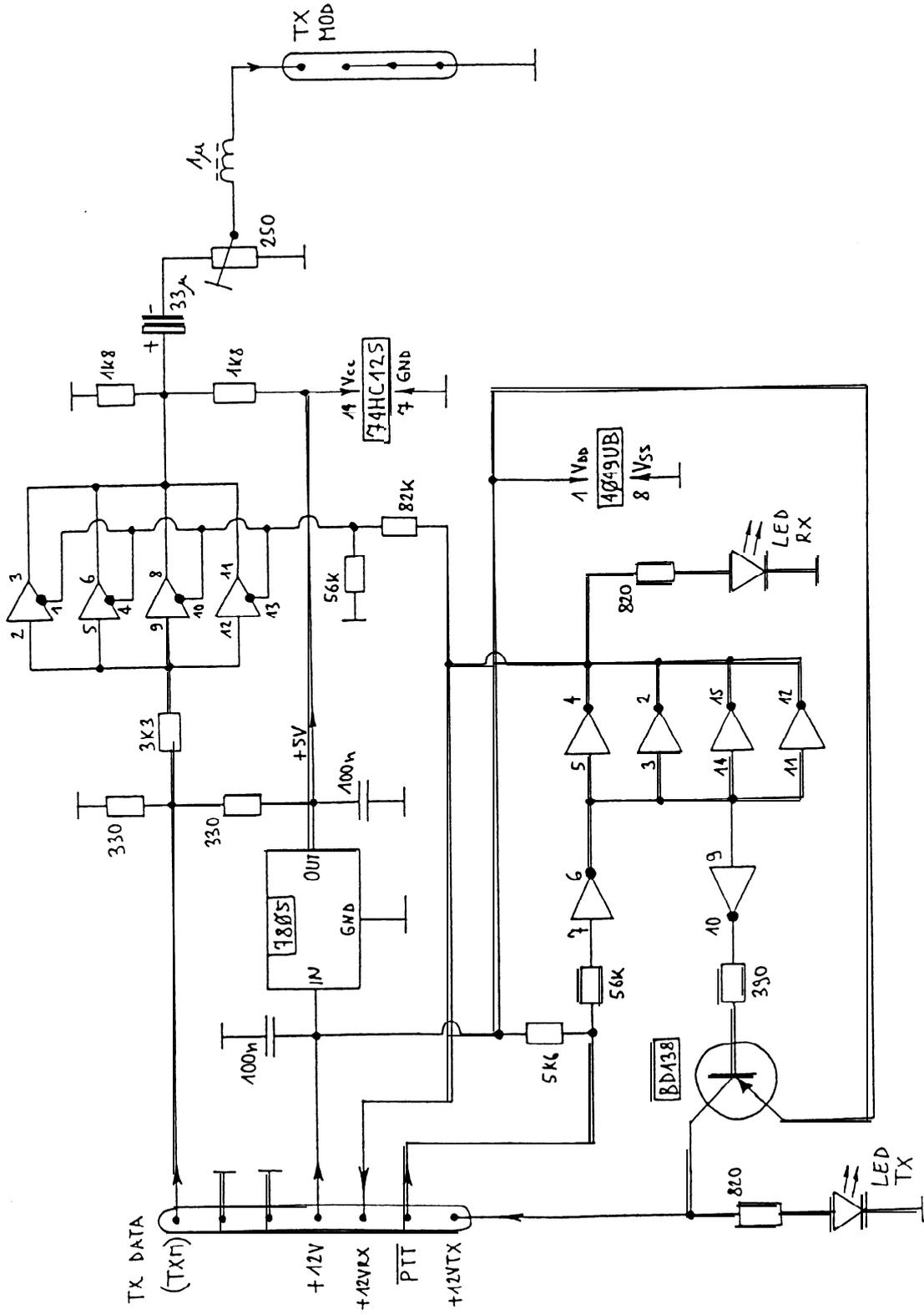


Fig. 21 - Supply switch interfacer circuit diagram.

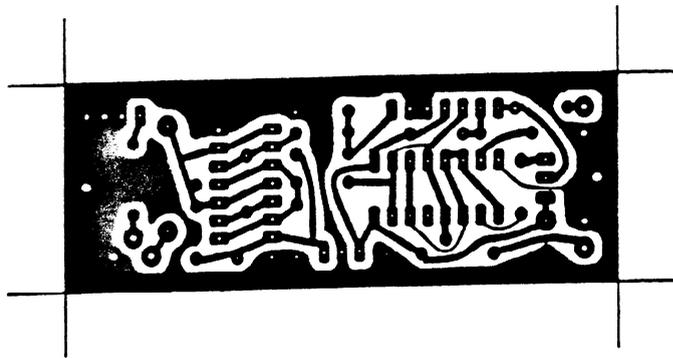


Fig. 22 - Supply switch interface PCB (1.6mm single-sided FR4).

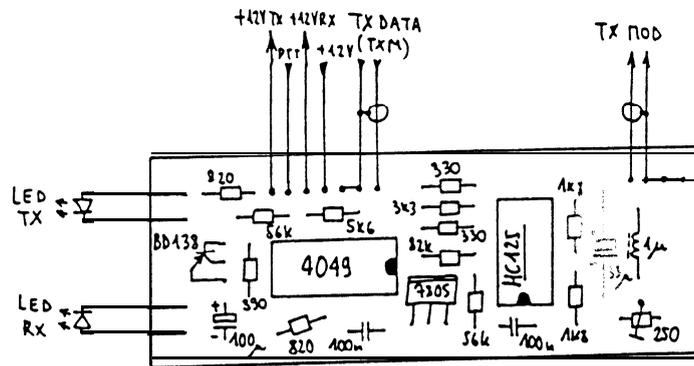


Fig. 23 - Supply switch interface component location.

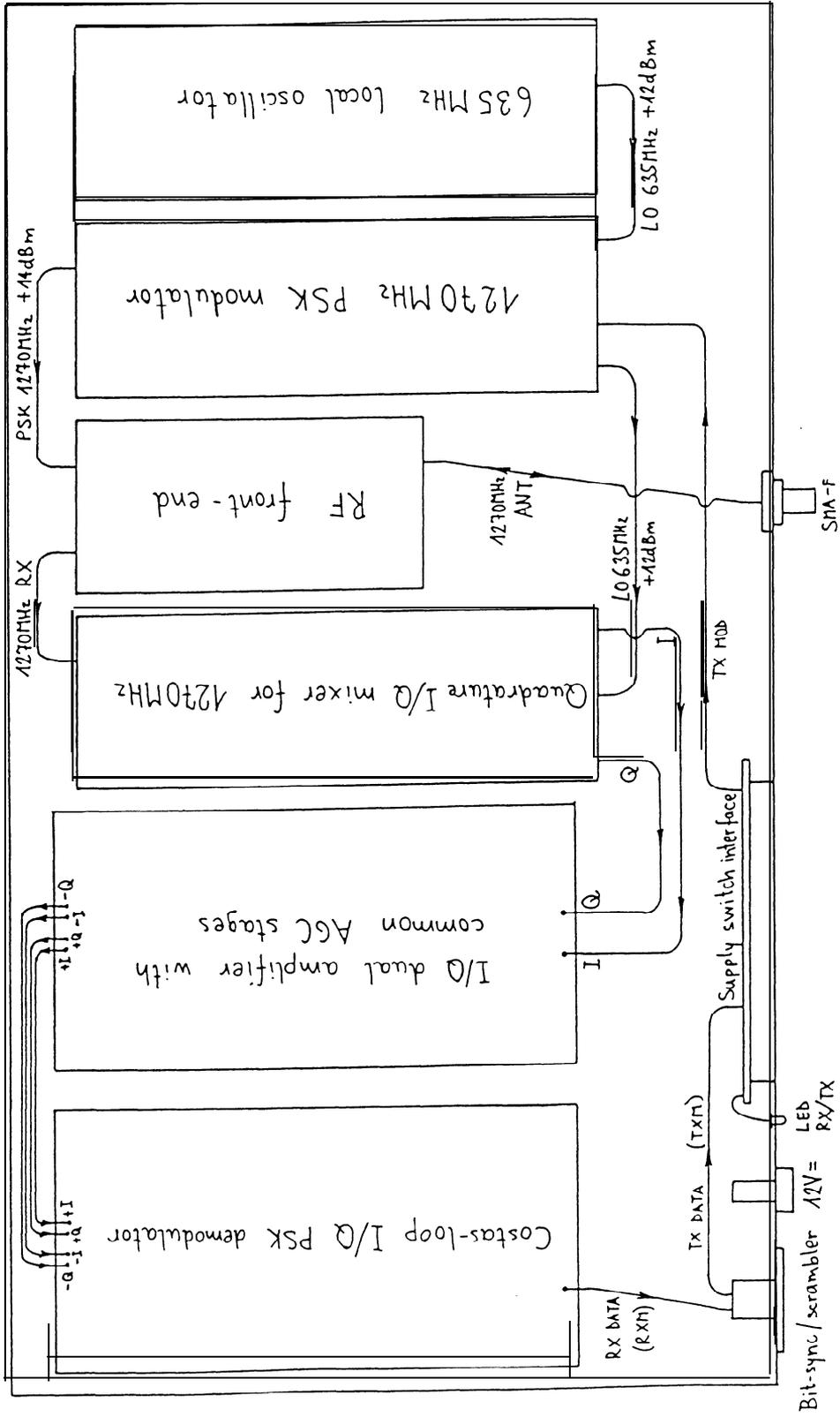


Fig 24 - 23cm PSK transceiver module location.

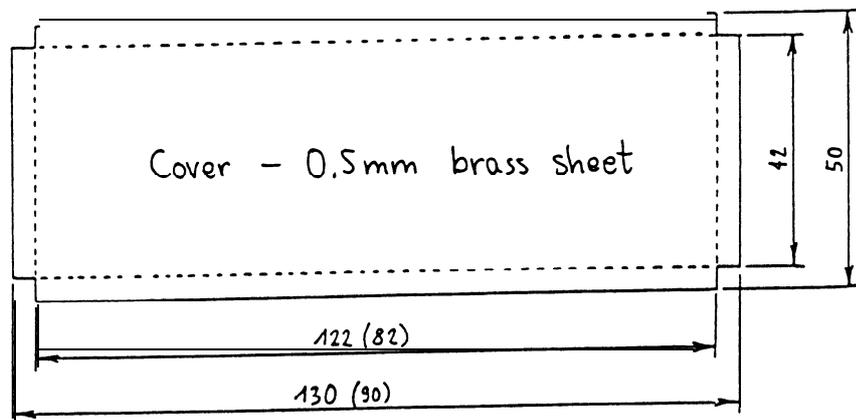
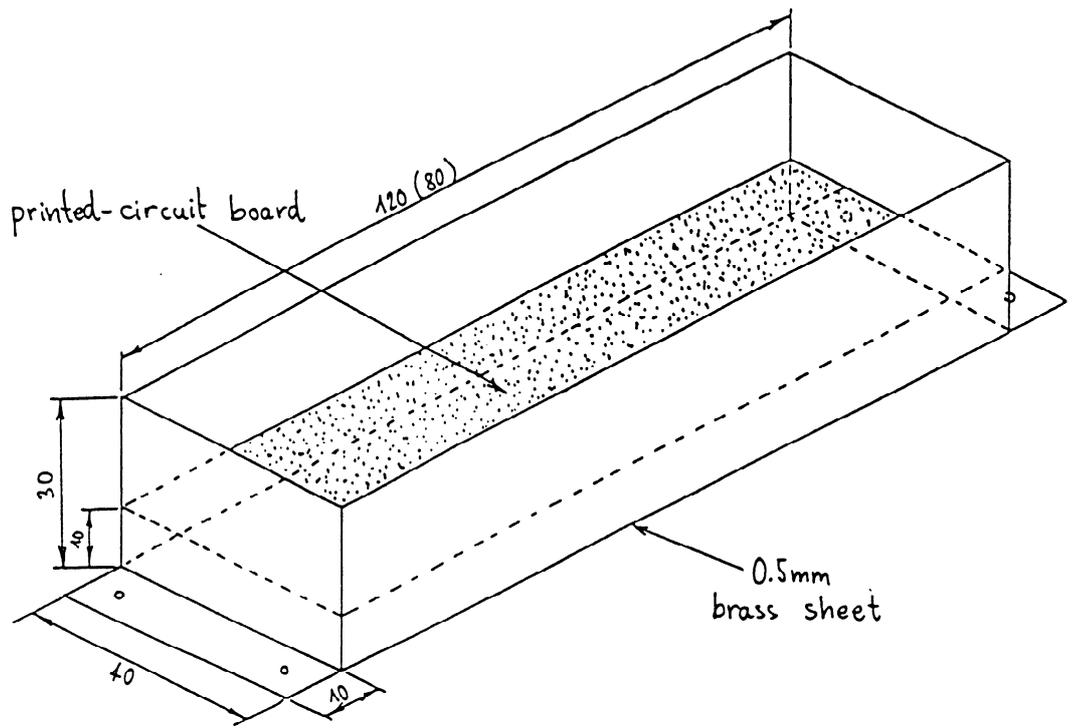


Fig. 25 - 23 cm PSK transceiver shielded module enclosure.

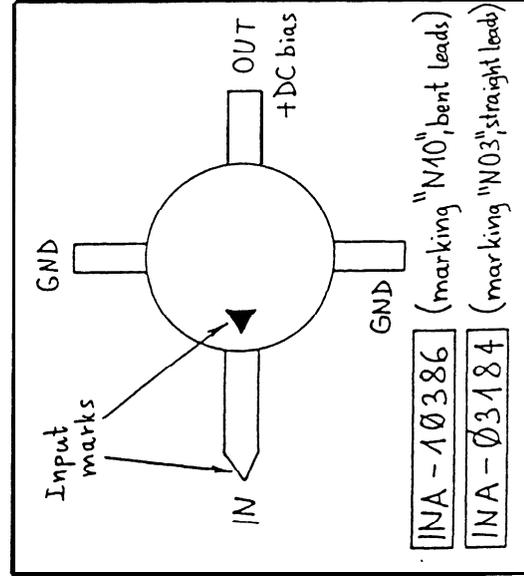
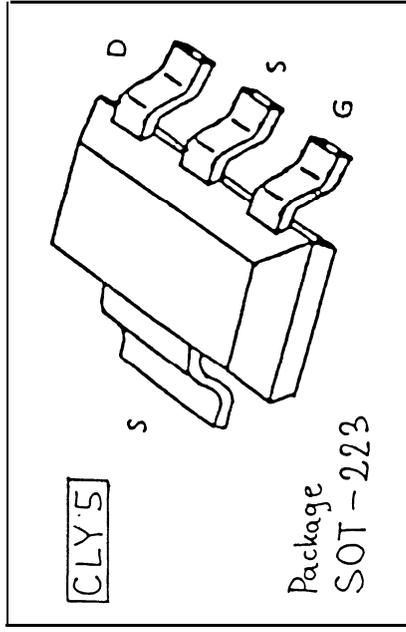
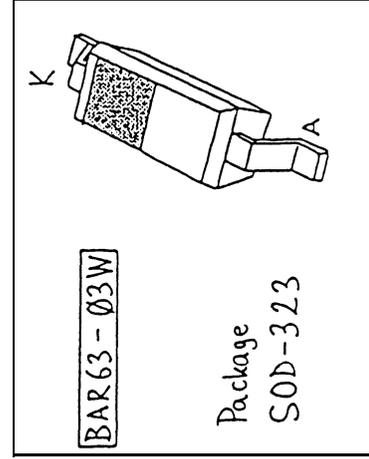
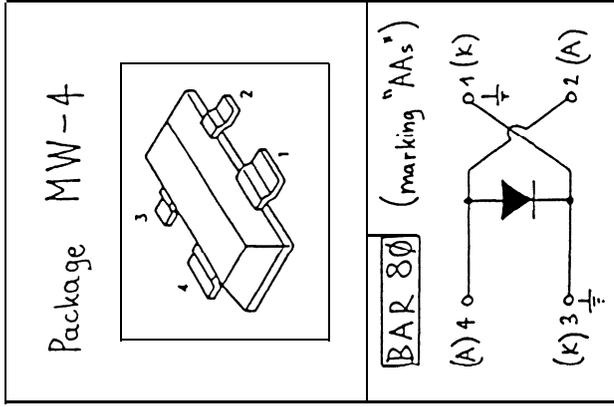
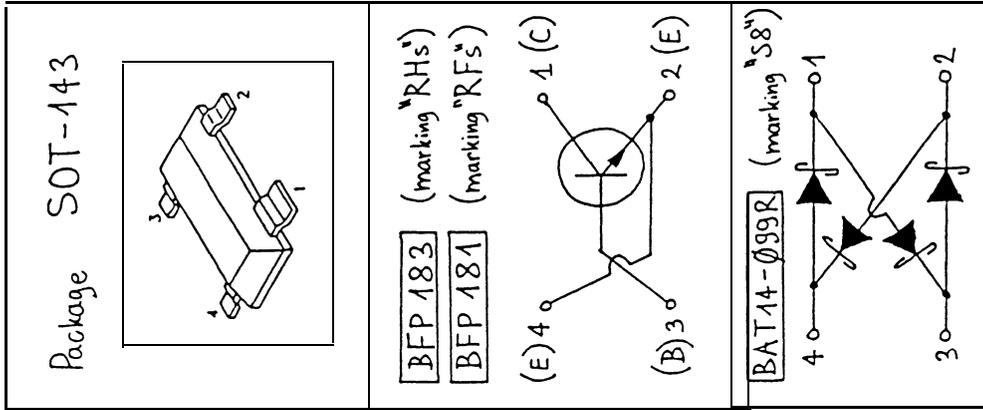


Fig. 26 - SMD semiconductor packages and pinouts.

PACKET AND INTERNET

James Wagner, PhD - KA7EHK
3 1677 N. Lake Creek Drive Tangent, OR 97389
(541) 928-7869 wagnerj@proaxis.com <http://www.proaxis.com/~wagnerj>

ABSTRACT

Debate is one of the interesting aspects of the packet bbs system. One of the recent debate issues is really quite important to all of us. It concerns the question of bbs mail forwarding by methods other than the ham RF network. Whichever side proves to be “right”, (and it is possible that both may be right), the answers to this debate will have an impact on all packet users. KEY WORDS: BBS, NETWORK, FORWARDING, INTERNET

INTRODUCTION

A major debate has raged in the packet bulletin board system over the last year or so. This debate concerns the use of alternate forwarding methods for moving packet messages. In particular, the use of (commercial) internet has been a major issue.

The issues in this debate warrant presenting here because they do represent ideas which are having, and will continue to have, major impact on the future of packet radio.

THE ISSUES, SIMPLIFIED

One viewpoint in this debate argues that the use of alternate forwarding methods (telephone, internet, etc) will result in a deteriorating RF network. The logic is that when alternative methods are used, there is no longer pressure to upgrade existing networks, fix broken ones, or maintain the ones we have. The argument continues with the idea that a network which is allowed to deteriorate will not be there when emergencies arise. And, when emergencies arise, it is also likely that portions of the internet infrastructure will fail. The result will be failure of the ham packet system to perform in emergencies.

The other viewpoint argues that the ultimate responsibility of bulletin board operators is to move “mail”. If the ham RF infrastructure is not capable of moving that mail, this argument continues, then they have the responsibility to find some method which will allow the mail to be moved. If that method happens to involve the telephone system or internet (ie, “wire line”), then so be it.

Network Quality

There are a number of factors which combine to represent the quality of a packet network. Of course, not all users have the same idea of quality. None the less, there are a number of general things:

- a. links need to be reliable
- b. unexpected disconnects should not occur
- c. reasonable throughput must be available most of the time
- d. hardware is physically reliable

When any of these factors gets worse, usually we perceive the quality of the network to be “worse”. A network can remain physically the same, but be perceived as deteriorating because it is not able to handle an increasing message load, for example.

Thus, for a network to remain as a high quality one, just keeping the hardware working is not enough. If the network cannot support increased demands placed on it, then it is not doing its job. Unfortunately, many users consider bbs forwarding to be the culprit, rather than the “miner’s canary” which warns us of impending difficulty. To such users, it’s probably just **fine** that their local bbs forwards over internet because it makes things seem to work better.

It is also an unfortunate human attribute that often, the quiet wheel does not get improved. If bbs sysops use other methods for forwarding messages, then a visible pressure for network improvement goes away.

Mail Movement

For many bbs sysops, mail movement is their entire reason for participating in ham radio. And, for some, at least, ham radio is just another access method for their bbs. When you have bbs sysops of this sort, what technical methods are they going to choose for linking? Certainly, the most familiar ones. If they are more comfortable with wire-line, that is what they will choose.

While “Clover”, for example, may do an excellent job of handling messages via HF, it is probable that more bbs sysops are familiar with wire-line modem technology than they are with Clover. So, is it any wonder that non-ham message passing technology is frequently the method of choice?

It is also likely that arguments that “it is not ham radio” will be quite ineffective. It is probable that many of the bbs sysops in this category do not have a big stake in ham radio and that this argument results in a big “so what?”

On June 30, 1996, WORLI wrote the author: “Yes, there is still a small amount of traffic handled via satellite, HF digital modes, and long haul vhf/uhf links. In fact, all PRESENTED traffic is easily handled. However, very little traffic is presented to the radio network; it is instead moved via commercial networks. When I first started speaking out on this issue, 18 months ago, about 50% of the long haul traffic was still being carried by radio. That percentage has now reached 0.”

The practice of wire-line forwarding is actually having a far bigger impact than it might seem. When strategically located bbs, in widely separated locations, forward almost instantaneously to each other, bulletins arrive in the other area more rapidly and get distributed to those stations which do use RF forwarding. Since the messages are already there when attempted by traditional means, they are rejected. This “capture” phenomenon results in an **artificially** forced reduction of RF forwarding.

CONCLUSIONS

It seems probable that both sides are “correct”. The sad part is that the cases of network deterioration seem to be growing more numerous with the use of forwarding over wire-line. It is also likely, however, that the cause-and-effect is not so clear. Bbs forwarding is moving off the RF network because it is deteriorating in many places and the deterioration is accelerating because there is less reason to keep it up. In other words, it is likely that these two effects go hand-in-hand and neither is the cause of the other.

What does seem fairly clear is that bbs sysops who move their forwarding off the RF network are not doing hams much of a favor. This is, in fact, one area where users can apply pressure, encouragement, and support to sysops. Hams with solid HF experience can help a sysop to set up a reliable forwarding system using Clover, Pactor, Amtor, or packet. Hams with good VHF/UHF network experience can help to make the bbs VHF/UHF packet equipment as good as it can be.

Likewise, bbs sysops AND users can apply pressure and offer assistance to their local ham clubs and packet organizations, and to node operators. Make sure that network capability improvement is planned, that groups involved in packet networking get together and figure out what is needed on a regional basis, and make sure that there is a solid commitment to carrying through on those plans.

Failing all else, bbs sysops in some areas of the country are rerouting messages to avoid forwarding them to bulletin boards which use wire-line forwarding. This is certainly a drastic measure but it is one of the few ways available to avoid the capture effect previously described. As unpleasant as this measure may be, the health of our network(s) may depend on it!

BIOGRAPHY

James Wagner is an electrical engineer & programmer employed by Kalatel Engineering in Corvallis, OR. His work is in the area of embedded controllers and design of system components for the closed-circuit video security industry. He has a BA in Physics from Oregon State Univ, an MS in Electrical Engineering, also from O.S.U, and a PhD in Electrical Engineering from Colorado State University. He has been employed by Tektronix and by the College of Oceanography of Oregon State University. His interest in ham radio began in the 1950's but did not actually get a license until 1979. He has been the advisor to the Oregon State University ARC and the node-op of their packet node. He is the author of “The Amateur Packet Radio Handbook”.

STRATEGIES FOR IMPROVING WIDE-AREA NETWORKS

James Wagner, PhD - KA7EHK

3 1677 N. Lake Creek Drive

Tangent, OR 97389

(54 1) 928-7869

wagnerj @proaxis.com <http://www.proaxis.com/~wagnerj>

ABSTRACT

Wide-area single-frequency networks still cover large areas of this country. A number of strategies have developed for improving such networks, but these strategies are very slow to be adopted. This paper discusses some of the reasons for the continued existence of these networks, some of the strategies and their likelihood of success. **KEY WORDS: WIDE AREA NETWORK, BACKBONE**

INTRODUCTION

Packet networking began as a single-frequency network, often on 145.01 MHz. NET/ROM software for TNCs permitted easy linking among nodes and it made it possible for users many miles apart to converse via packet. These networks are also the lowest cost, since only one transceiver, one antenna, and one TNC is required per node.

While this kind of network provided great advantages in comparison to the previous state of affairs, we now know that they have some major limitations. Yet, they persist. If one observes where they exist, at least in the west, it is possible to guess why. For example, a 145.01 network exists (with no overlaying backbone) over (almost) all of Wyoming, eastern Montana, Eastern Idaho, almost all of Nevada, most of Alberta, some of Utah, and other adjoining areas to the west, east, and south. Single frequency networks also exist in large areas of the mid-west and other parts of this country.

There are a number of reasons why these networks continue. At least some of the reasons are:

1. Population density is very low. Thus, hams are scarce and resources are very limited.
2. 2Meters may be the only VHF/UHF band node-ops are comfortable with.
3. There may be propagation problems on higher bands.
4. Land relief seems to be either very flat with sparse to no high points or very mountainous.
5. Weather conditions (especially winter) can be severe
6. Land ownership is often in federal hands. In many places, this makes access to high points difficult and/or expensive.
7. Power sources may be very limited, though this is changing with the installation of cell sites.
8. The initial step of introducing a backbone is often seen as disruptive.
9. There is often a cultural resistance to providing resources for an activity such as BBS forwarding which is seen as either no benefit or antagonistic to local users.
10. Such networks tend to inhibit the growth of packet & this can be seen as desirable by existing users.

11. Clubs are often not used to cooperating about anything more than a local repeater. Packet networking sometimes seem to be more than they can cope with.

Given this situation, what has been done and what can be done to improve these networks?

BACKBONE MYTHS

There are, of course, some myths which have grown up about backbones and these myths often contribute to the reluctance to adopt this technology.

Myth 1: Backbones need to be on UHF. Of course this is false, but still widely believed. As a simple example, there is an excellent 6M, 2400 baud backbone which extends along the southern half of California-Nevada border to Las Vegas, then east to Phoenix. While this backbone is subject to the wide-area problems as single-frequency networks, it gets users off the linking frequency. 2400 baud also makes it less simple for users to connect directly to it and increases throughput.

Myth 2: Backbones can't be on 2M. Again, false. An interesting scheme is used in Alberta with a 2M backbone on 145.01 and user ports at the high end of the 2M band. This provides better than 2MHz of separation and standard repeater cavities can take care of this very well.

Myth 3: You can't convert to a backbone without disrupting the network. False. Some creative strategies for setting route quality are about all that is needed. Of course, there will be disruption when nodes change from their old frequency to a new one, but experience has shown that the result is seldom catastrophic.

Myth 4: Backbone nodes are difficult to make. This may be somewhat true for a node with more than 2 frequencies, but for 2, only, its a "snap".

Myth 5: You can't teach an old dog new tricks. Sorry, but packeteers have disproven this long ago. A backbone can be perplexing to users who have never used one, but the principles are pretty basic and any club worth its name can solve this without much difficulty. Besides, packeteers are often there because they WANT to be at the cutting edge, or near it, at least.

TWO METER BACKBONES

As Alberta hams have discovered, sparse population can be an advantage. There are not many repeaters and not much simplex use. The result is that there are often repeater pairs at the high end of the band which are not used. It also means that the simplex slot from 147.42MHz to 147.57MHz is probably lightly used. The biggest problem may be that even a very lightly used simplex frequency or one which does not seem to be used at all may be "claimed" by a group of hams. Real care needs to be exercised when using the simplex option, but it can often work.

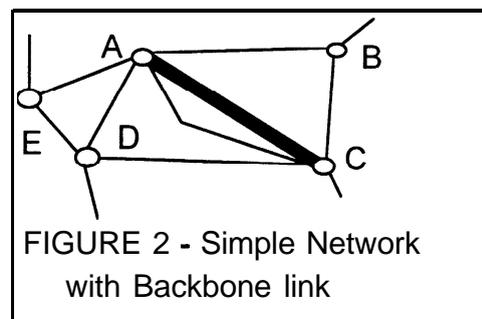
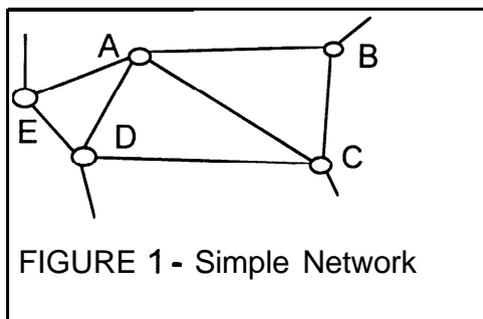
Using this scheme, a second 2M port is added to the networking node. This port is at the high end of the band. Sometimes, a duplexer allows using one antenna for both ports. Then, after a "get acquainted" time, the node parameters are set to disallow user connections on the networking frequency.

This greatly reduces the hidden transmitter problem for both users and the network. Users do not have to compete with network activity and visa versa. Users can also see, quite quickly, the benefits of a backbone and it can provide some real encouragement for technical or financial support.

This strategy can easily be used as a stepping-stone to backbones on other bands or at higher baud rates. Once users are off the backbone, the actual backbone frequency becomes less important.

INSERTING A BACKBONE IN AN EXISTING SINGLE-FREQUENCY NETWORK

While the methods which have sometimes been used to add a backbone to an existing network HAS resulted in disruption, it does not have to be very much. It only requires some thought, some planning, and some careful execution. But its NOT “rocket science”.



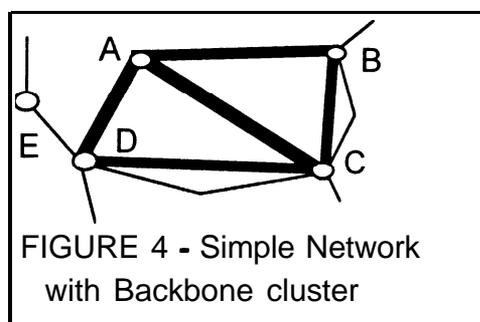
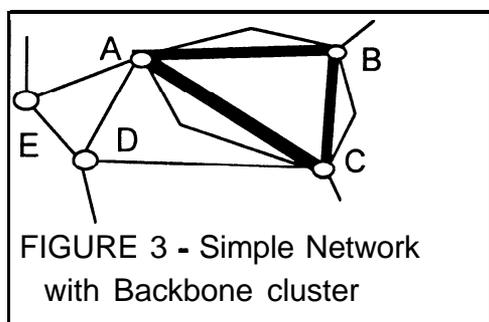
Consider the simple hypothetical network shown in Figure 1. Suppose that node-ops want to convert the link between nodes A and C to a backbone. Technically, its not difficult to do that. A second TNC, a second radio, and (often) a second antenna at each end is about all it takes. Without changing the user frequencies at either end, the result is shown in Figure 2. The heavy line represents the backbone link.

At first, this arrangement would seem to be quite useless. But it is a very important first step. It is critical, however, to choose some carefully thought-out route qualities to get any advantage from it. Lets suppose that 192 is commonly used as the quality for good paths to neighbors. It is only necessary to select a slightly higher path quality for the backbone link, but this must be done at both ends. A value of 205 would work, so lets see what happens. The quality for any two-link path in the old network, say from E to A to B, is 144. Likewise, the route from E to D to C is 144. But E to A to C is 153. The preferred routing from E to C is now via the new backbone link even though a parallel link on the user frequency is still present. This arrangement also provides an alternate route if the backbone does not work as well at first as expected.

What are the consequences of this choice? Clearly, D will carry less of the traffic from E to C but A will carry more. The users at D will have less network competition but those at A will see more. Depending on existing traffic flow, this may be significant or it may not. But, in spite of this, there will be a net reduction in user frequency activity at C because much is now going via backbone. And, while A gets more traffic to and from E, it is transported by backbone to C, and the former traffic directly between A and C is also now hidden. Users at A will probably see little net change, and it may improve. Clearly, how the network traffic flow redistributes itself and what the consequence is depends on how much

traffic and which nodes carry it. Generally, some users will see a clear benefit and others will see anywhere from a slight benefit to a slight degradation.

But, in spite of this addition, the node-ops are left with a dilemma. A and C need to retain their old frequency in order to keep connectivity to the rest of the old network. What to do? The next thing might be to add B or D (or both) to the backbone. Adding both is not really desirable, given the links shown in Figure 1, because there is no link between B and D, and that probably means that one is hidden from the other. Lets suppose that B is the node which is added to the backbone. Figure 3 shows the result.



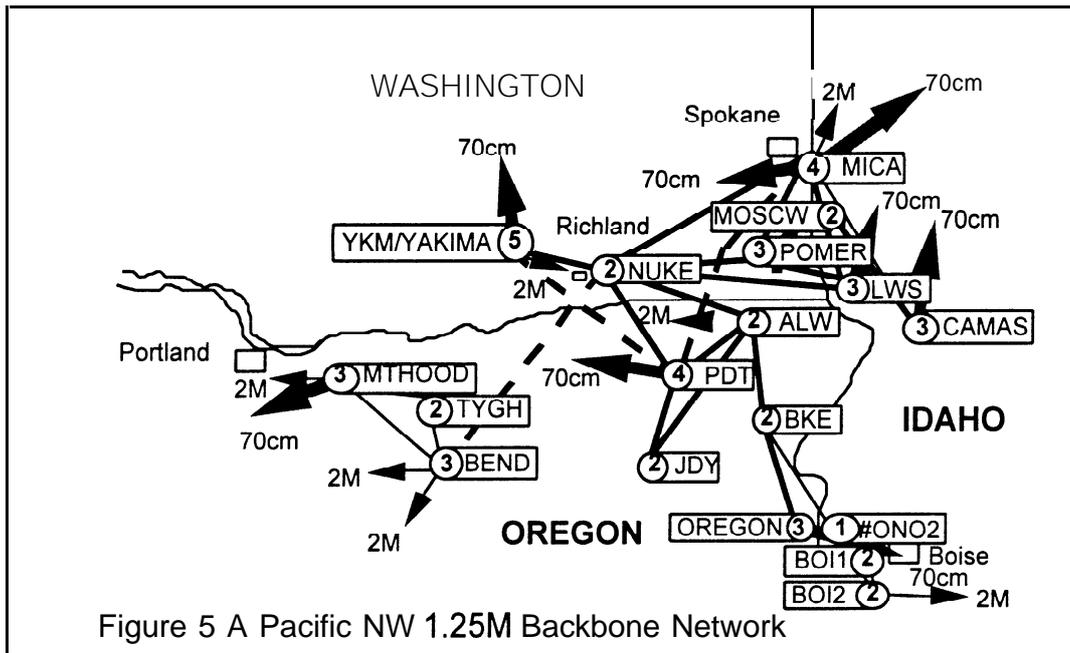
This arrangement is properly called a “cluster” but the node-ops of A, B, and C still have the same dilemma. They have to stay on the old network frequency in order to retain connectivity. Users at both B and C should still see a clear improvement, but it might not be dramatic.

Adding D to the cluster, while not desirable from a hidden-transmitter standpoint, will still result in significant improvement, especially at A. Now, all network-level traffic passes through on a non-user frequency at A, eliminating all competition with users. Because there are no remaining links (without alternatives) to the old frequency for A, it can change frequency. The result of adding D to the cluster and changing the user frequency of A is shown in Figure 4. From this simple move, users at D, C, and B will see a major improvement because their nodes no longer have to compete with A. Drawbacks are: traffic between A and E must now travel via D, and D becomes a single-point-of-failure in the new network.

While no network will behave exactly like this example, it should be apparent that careful planning and execution can result in the addition of a backbone without major disruption. In all of this, the only real disruption was the frequency change at A, but I’d bet that the users would really like the results! And, all of the gain resulted from a link which, at first, appeared to offer no real benefit.

IMPROVING EXISTING WIDE-AREA BACKBONES

There are many areas covered by backboned networks which cover far too large an area. A good example is the 1.25M network in eastern Washington, eastern Oregon, and western Idaho. This network contains at least 15 nodes from Mt. Hood in Oregon to Boise, Idaho to Spokane, Washington. The east-west extent is about 280 miles (450km) and the north-south extent is about 300 miles (480 km). Clearly, hidden transmitters are a problem and a simple check of the network map shows this to be true, even in smaller sections of the network. Figure 5 shows this network and some of the connecting links to other network s.



The network shown in Figure 5 has some clear problems. Perhaps the most obvious is the group of 3 nodes at the west side which is connected only tenuously to the rest of the network. The one link works only part of the year, and then, only for part of the time. Several of the western nodes are heard occasionally by other nodes in the eastern part of the network. For all concerned, operation would be improved in both parts of the network if one section, say the western part changed frequency. The east-west extent would then be reduced by over 100 miles (160 km).

A fairly clear improvement would be to change a portion of the eastern part of the network to another frequency. This requires one or more nodes to become 3-port locations. But, several of the nodes already have 3 or more ports and those should not have to carry added burden. One good candidate would be the node called ALW; the linear path to Boise would then be away from the rest of the network. Much of the remaining at least DCDs each other (most of the time) so makes sense as a connected cluster. Here, we have one node adding one port and 5 others (BKE, #ONO2, OREGON, BOI1, and BOI2) changing frequency. The cost is one radio and TNC and the result would be significant improvement for all.

This example should show that one of the real aids to network improvement is to construct a map of the network, and try to determine which paths work well, some of the time, or only rarely. Such maps often help to show how the network functions and then, how it can be improved.

CULTURAL IMPEDIMENTS

While it is not frequently discussed, culture may play a role equal to, or greater than, technical considerations in the development of networks.

For example, in the Northwest, and I suspect also in other parts of the country, there have been literal “wars” over BBS forwarding and message content. These conflicts have often concerned issues of whether a node should be “clogged” by BBS messages which some local users find worthless (ie, for-sale messages from another part of the country). In some cases, nodes have been purposely left unimproved just so that the flow of BBS traffic would not take place.

Another cultural issue has to do with keeping a packet node “for me and my buddies”. If the node is not improved, newcomers become discouraged and put their TNC back on the shelf. Then, the node is not excessively congested and there is no obvious reason to “improve” it.

The biggest problem is often that this kind of thinking is seldom discussed in the open. People rarely admit that this is really why the local node has stayed single port for 10 years.

When reasons like these surface, there are only a few “solutions”. One is to just leave things as they are and walk away. Another is to attempt to convince the responsible people that they, other packeteers, and ham radio in general would all benefit from an improvement. Another is to make an end run around the existing network, find a few who are interested in really improving things, and go at it yourself.

Another problem which is not well recognized is the need for cooperation. Ham groups are often not used to working with neighboring clubs, especially over the wide areas required for a functioning packet network. Often, there is suspicion or outright animosity, sometimes generated by prior conflict over repeater frequencies. Sometimes, this can be overcome by some respected group inviting others to a regional meeting about packet networking.

CONCLUSIONS

Several methods have been discussed for improving single frequency, wide area networks. Some of these ideas may work in some situations, and some won't. But above all, do not be afraid to think, question, work with other hams, and find even better solutions than the ones suggested here.

BIOGRAPHY

James Wagner is an electrical engineer & programmer employed by Kalatel Engineering in Corvallis, OR. His work is in the area of embedded controllers and design of system components for the closed-circuit video security industry. He has a BA in Physics from Oregon State Univ, an MS in Electrical Engineering, also from O.S.U, and a PhD in Electrical Engineering from Colorado State University. He has been employed by Tektronix and by the College of Oceanography of Oregon State University. His interest in ham radio began in the 1950's but did not actually get a license until 1979. He has been the advisor to the Oregon State University ARC and the node-op of their packet node. He is the author of “The Amateur Packet Radio Handbook”.

THE WORD STORAGE RELAY (WSR) by Pat West, P.E. (W7EA)

One of the **more** complex systems developed during the cold war was the **BOMARC** missile system developed by Boeing and the University of Michigan. **The BO in the acronym stands for Boeing and the MARC stands** for Michigan Aeronautical Research Center. Boeing built two versions of the BOMARC, the first one designated “**IM99A**” and the second “**IM99B**”. The “A” version used a liquid propellant boost system with a range of about 250 miles and the “B” used a solid propellant for boost and had a range of at least **400** miles. The BOMARC **was a pilotless aircraft vehicle, developed and deployed to counter the Soviet** massed bomber threat. An artist’s version of a BOMARC missile is shown in Figure 1.

One of several electronic systems on the BOMARC was the command system. Figure 2 shows the major part of the command system, ready for installation in missile Section 41. The UHF receiver was mounted separately and is not shown in the figure. The **command** system responded to UHF radio commands, to control the launch, boost, cruise and terminal dive time phases of the missile. I believe the **command system** was an element of the first ever ground-to-air digital control system.

A brief **summary** of the role of the command system in a missile is as follows, please refer to Figure 3. A Semi-Automatic Ground Environment (SAGE) system used control centers at strategic locations in the USA. Each control center received radar target reports via a radar network. **If** hostile aircraft such as a bomber raid was identified by a control center, computers at the control center would generate digital control messages and would transmit them via land lines to a supporting BOMARC missile base. The control center would supply data to a selected missile that would include takeoff azimuth and dive timer setting. This information would be updated until such time that a launch command would be given. At the missile base, the flight control data would be transmitted to the missile by a low power UHF transmitter via a coaxial distribution system. The UHF receiver in the missile would detect the data and pass the information on to the Word Storage Relay (**WSR**). Each command would ultimately reach the missile flight control system.

When the liquid propellant booster was activated, the missile would follow a boost phase as controlled by a transducer in the command system. It would take off on the azimuth set by the control center computer. Commands from the control center during cruise phase would control the missile. These commands were transmitted to the missile from a high power UHF transmitter located on the missile base. When the dive timer was activated by the control center computer, the missile would dive on the target, the terminal guidance radar would lock on the target and control the remainder of the flight. At a certain distance from the target the fuse system would be activated to ultimately ignite the payload ordinance.

The digital data link **from** ground to air was a 14 bit system using a 4 bit address, a 9 bit position command for the addressed transducer and a 1 bit parity. Each bit took 10 ms and a complete word was 140 ms. A **460 ms** period was allowed for transducer positioning before the next word was transmitted. Total cycle time was therefore **600ms**. Real slow when compared to today’s digital systems!

Each command system receiver on a missile contained a unique sub-channel module which was the technique used for missile selection. The leading edge of a transmitted word would trigger the WSR and the word would be stored for 460ms until the next cycle.

The brain of the command system was the Perkins-Reynolds Selector Relay, Patent No. 2795,773 as illustrated by Figure 4. This unit as used on the **BOMARC** “A” missile was referred to as Word Storage Relay (**WSR**). **The WSR** was conceived and developed in **1950/1951** by L. C. Perkins and F. D. Reynolds, engineering managers at Boeing. They developed the unit for a hobby, for the radio control of a complex model fire boat. The WSR was used only on the **IM99A** missile.

Figure **5** shows the principle of operation of the WSR. The figure shows 20 reeds as used on their model boat, whereas the BOMARC “A” WSR contained only 14. A non-ferrous metal disk with a slot in

the edge of it is mounted on a ferrous shaft. Surrounding the edge of the disk are a number of spring steel reeds or fingers, whose outer ends are fixed to the frame of the device. The inner ends of the reeds are located so that they may pass through the slot, one at a time, from one side of the disk to the other when the slot is aligned with the reed. At all other times, each reed is trapped on one side or the other of the disk. A trapped reed supplies ground to its associated relay. Every 600ms as synchronized by the command system receiver, the WSR would be rotated a complete revolution in 140ms (Drive motor is not shown on the figure). Reeds would be synchronously activated or released.. The WSR would be idle for 460ms until the next cycle.

A magnet coil surrounds the shaft. An arm which is fixed to the shaft extends close to the slot in the disk. When the coil is energized, this arm concentrates the magnetic flux on the one reed which is at that time aligned with the slot, thereby pulling the end of the reed through the slot. In their free positions, the reeds are on the side of the disk away from the flux concentrator arm and do not touch the disk. When the reeds are attracted through the slot they become trapped on the concentrator-arm side of the disk, as a result of the rotation of the disk and make electrical contact with the disk.

It is therefore evident that once per revolution, we have the choice of closing any reed switch or a combination of them, leaving any or all closed that are already closed, opening any or all or leaving any or all open. At all times when the slot does not line up with a particular reed switch, the switch remains in the position it was left in the last time the slot passed it. Any configuration of open and closed switches representing any binary number may therefore be set up on the device in one revolution and retained as long as desired.

It is therefore obvious that the slotted disk must be rotated in a timed relationship with respect to the information pulses applied to the reed switching coil, in order that the proper reed switches will be closed. The rotation is controlled by a governed motor (not shown on Figure 5) so that the slot in the disk is aligned with each reed in turn at the exact time that the pulse intended for that reed actuates the magnet.

The command system included a box containing relays and a network of resistors. Certain relays would be activated depending on the transmitted word contents. A transducer would be selected and a transducer position would be derived by the relay/resistor bank. The existing transducer position would be represented by a 900 Hz voltage magnitude as picked off the transducer rheostat. This voltage and the 900 Hz voltage from the resistor network would be fed to a servo amplifier. The voltages would be compared and the transducer would be repositioned to the commanded position.

The WSR was ultimately replaced with an all solid state unit as developed by Motorola. The BOMARC bases no longer exist and deployed missiles were used for other purposes including their employment as targets for other missile types.

There is an interesting story dealing with the invention of the Perkins-Reynolds Selector Relay. It appears that they, as Boeing employees, were required to advise Boeing of their patent. Boeing told them that the company had no requirements for a device to control a toy boat and the patent was theirs. Ten years later, the inventors took the model boat to England and won a world championship with it. The model fire boat is illustrated in Figure 6.

In the meantime, back at Boeing, they were having trouble coming up with a good system for radio control of the BOMARC missile. The inventors made a pitch to Boeing management, they bought the idea and directed the Boeing patent staff to reacquire the rights to the model-boat control invention. The inventors signed a contract with Boeing and Mr. Reynolds became the manager in charge of the Boeing version of the device. According to a study, the use of that "toy boat" invention in the guided missile saved 350 vacuum tubes per missile. (This was before the availability of transistors and integrated circuits.)

Author: C. P. West, P.E. (W7EA)
29825 8Th Place South
Federal Way, WA 98003
Telephone: (206)839-4827

The author is a retired Boeing Engineer, who was a Lead Missile Test Engineer on the BOMARC "A" development program and was also involved in system engineering on the BOMARC interface with the SAGE system. He is a Senior Member of The Radio Club of America, a Life Senior Member of the IEEE and has been a licensed radio amateur for 59 years and has an Extra Class rating. In 1981, he led the initial radio amateur delegation to Communist China and **played** a significant role in getting them back into amateur radio,

CREDITS: 1. Photos- The Boeing Company, Seattle, WA

2. Description of WSR operation- F. D. Reynolds. Mr. Reynolds is the author of the book, "Crackpot or Genius", published in 1993, a complete guide to the uncommon art of inventing. Book is available from: Chicago Review Press, Inc., 8 14 North Franklin Street, Chicago, IL 60610.

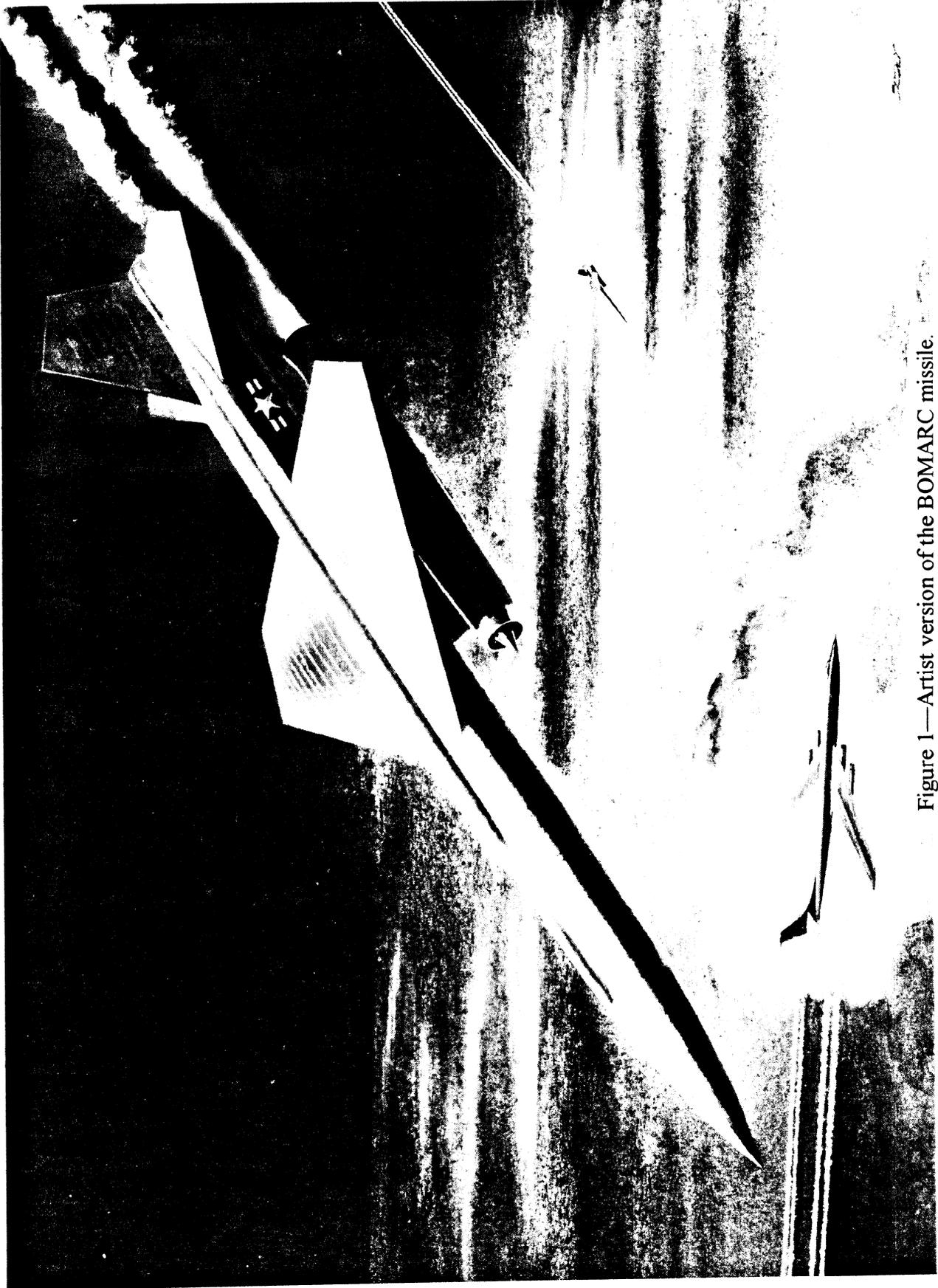


Figure 1—Artist version of the BOMARC missile.

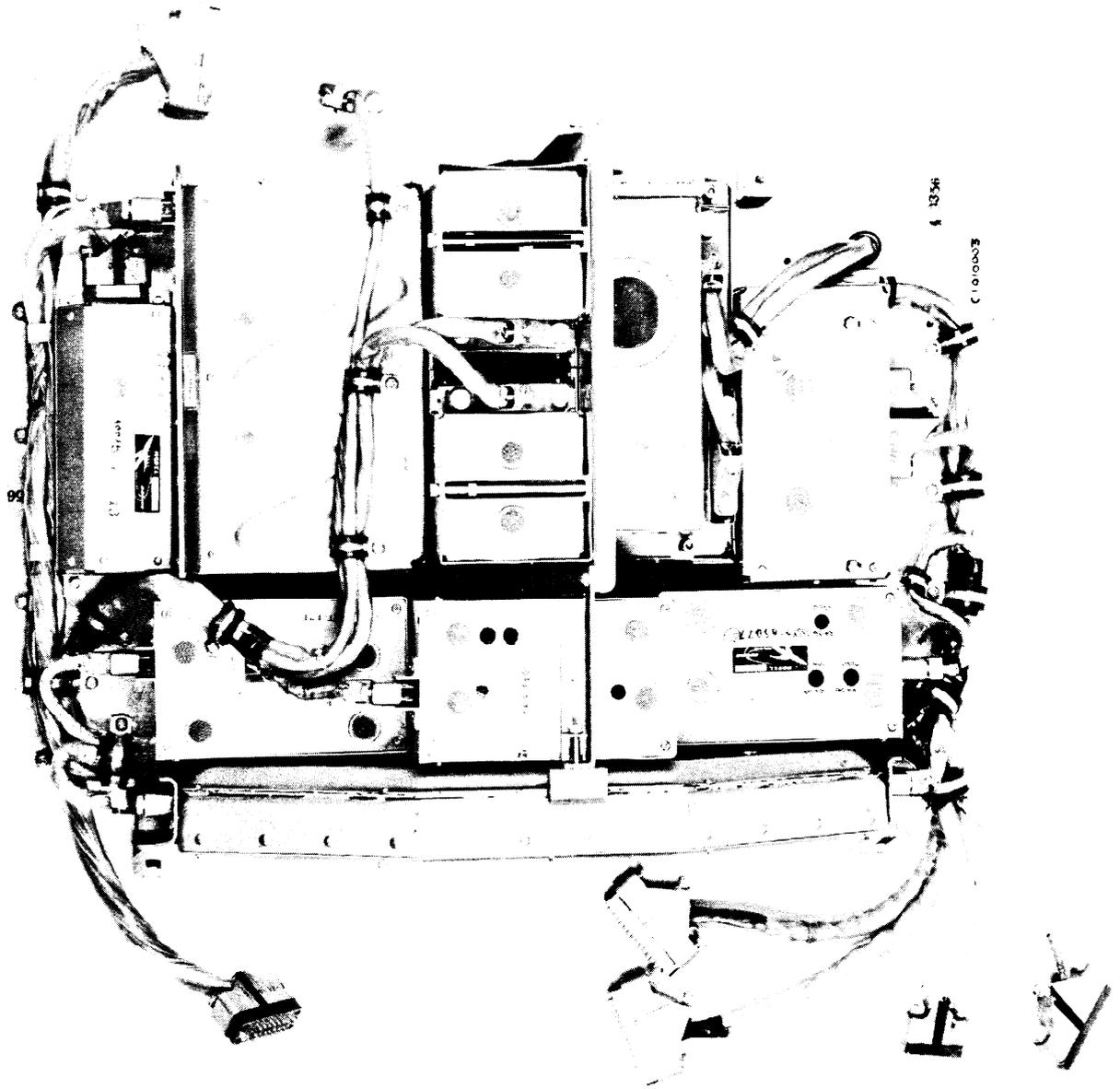


Figure 2—BOMARC Command System hardware.

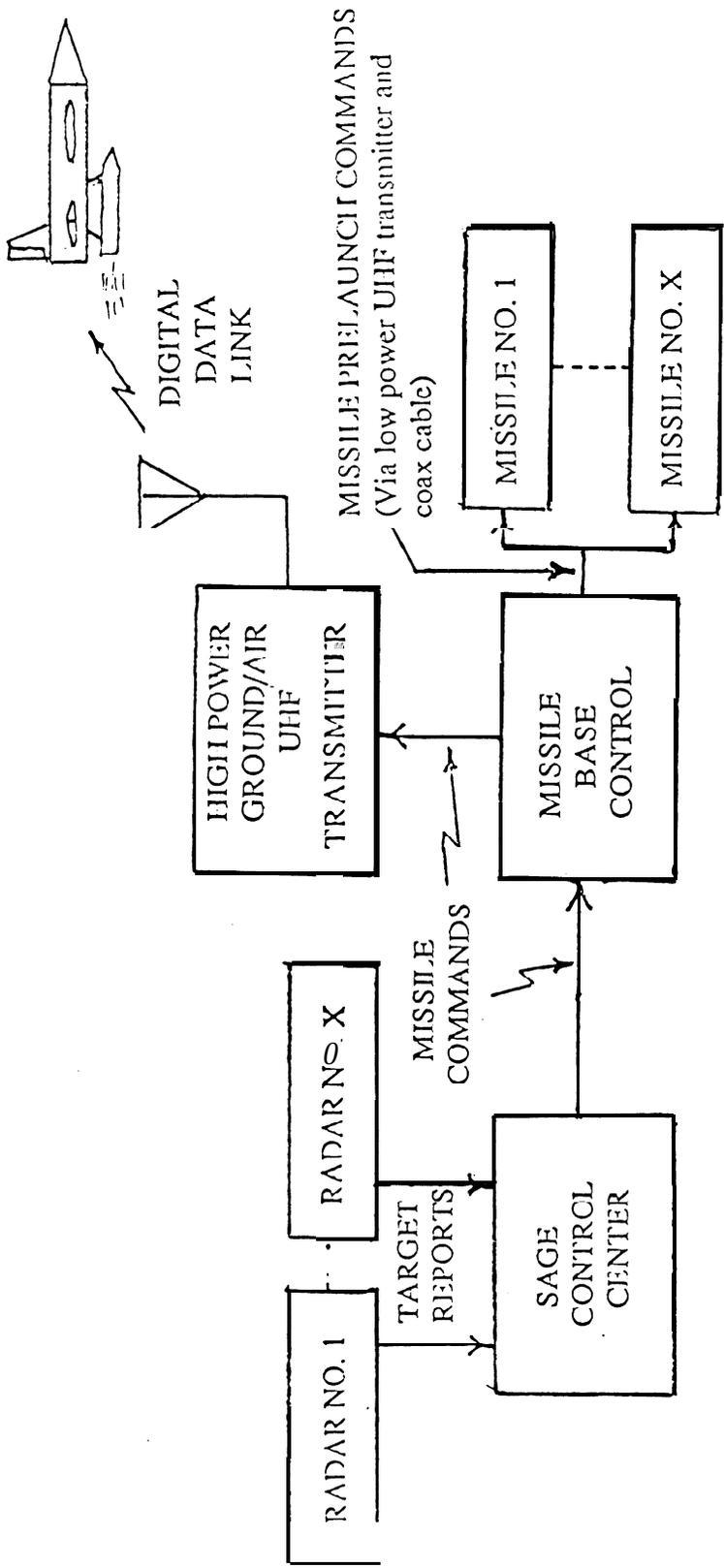


Fig. 3. SAGE control of BOMARC.

Ref to summary
9/23

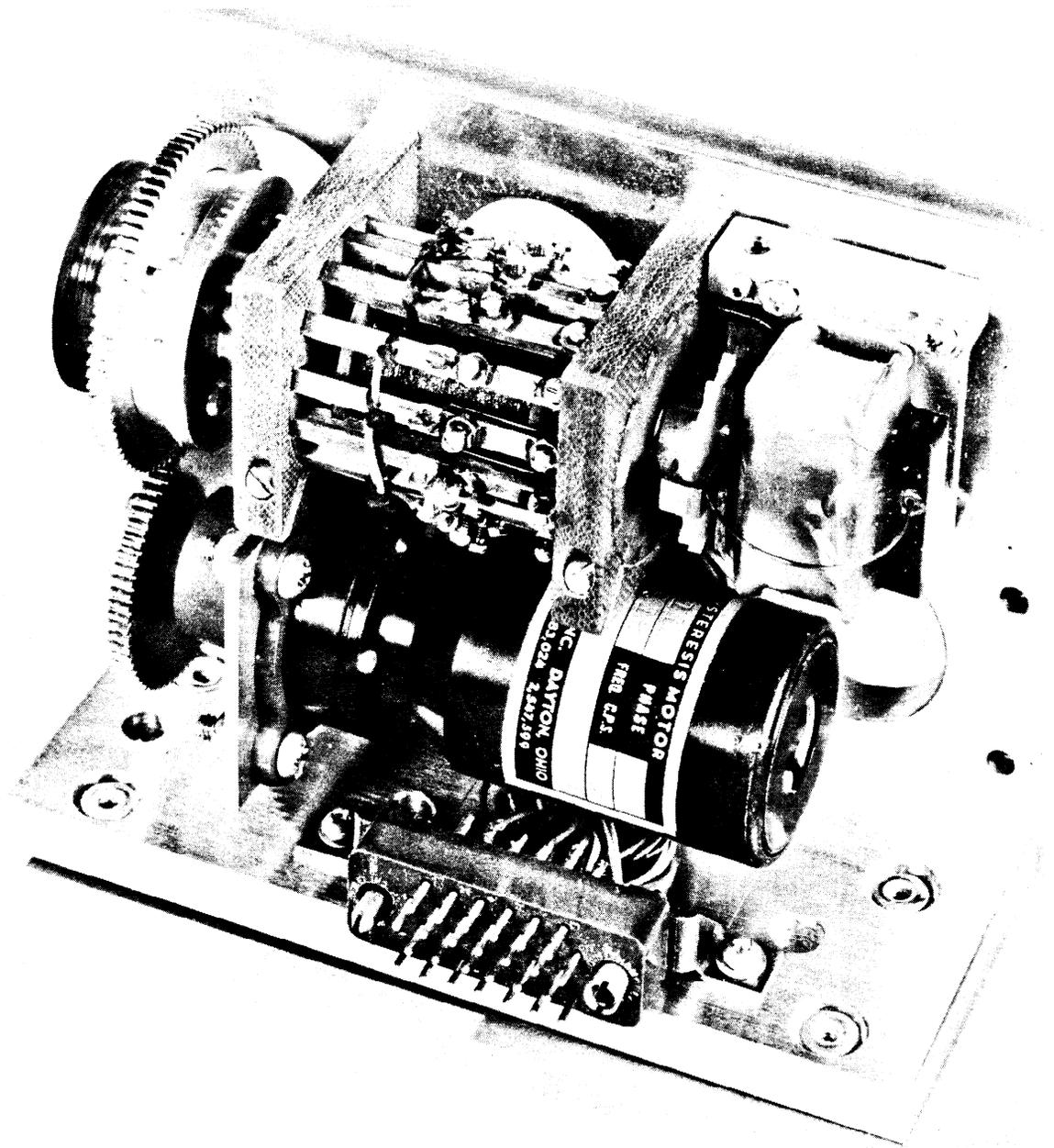


Figure 4—Perkins/Reynolds Selector Relay which was called word Storage Relay on the BOMARC Command System.

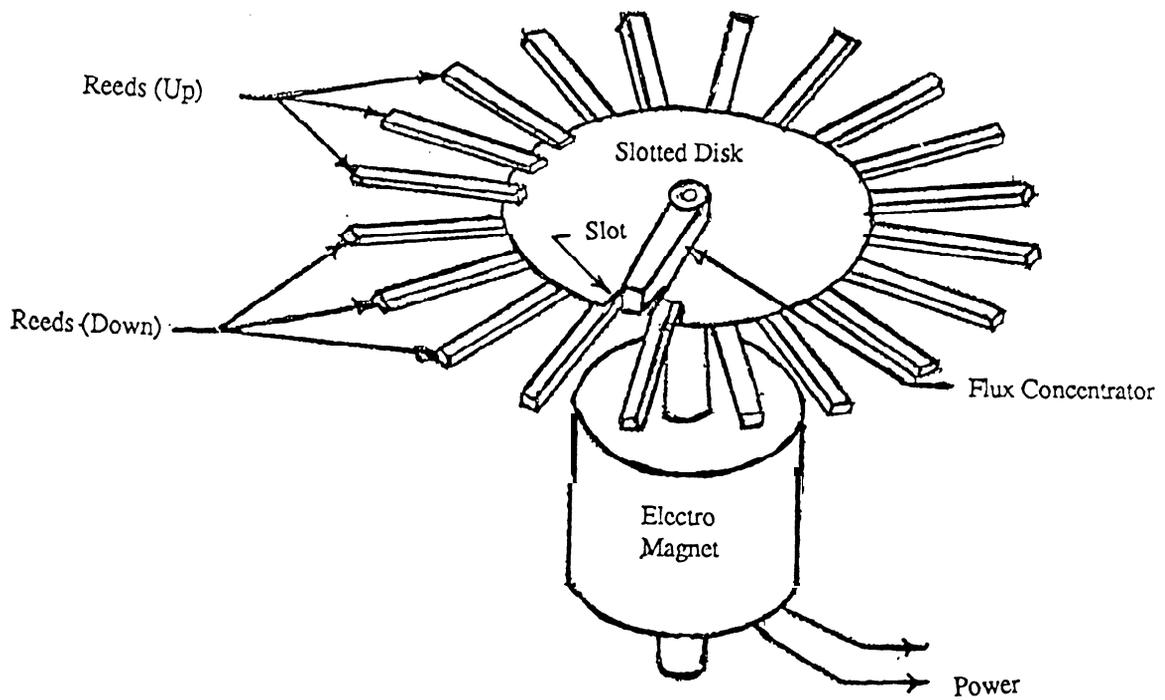


Fig. 5. Illustration for explaining Word Storage Relay operation

ret Au Nov 9/23

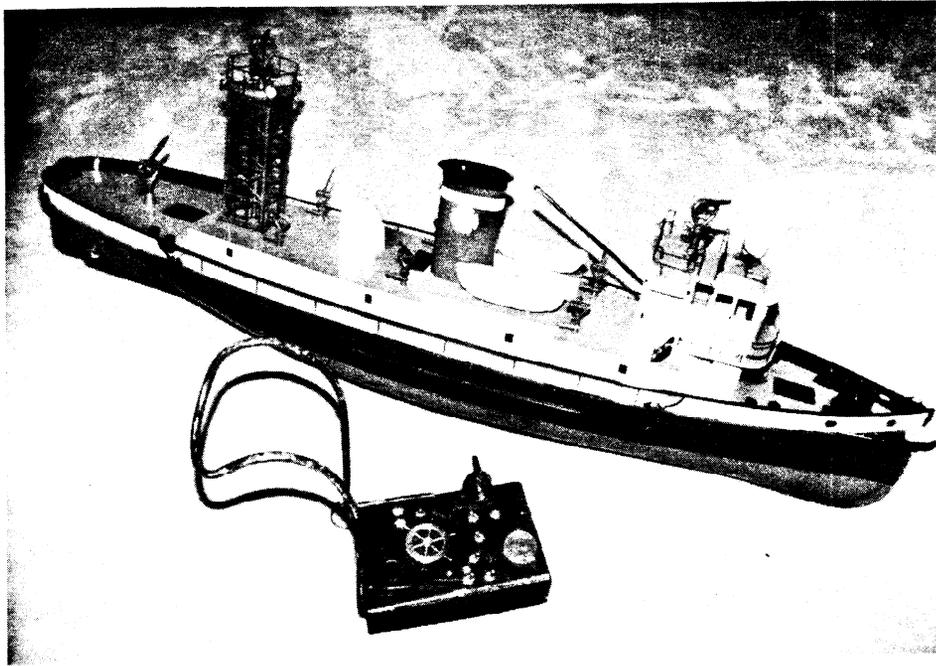


Figure GA—Model fireboat with controller.

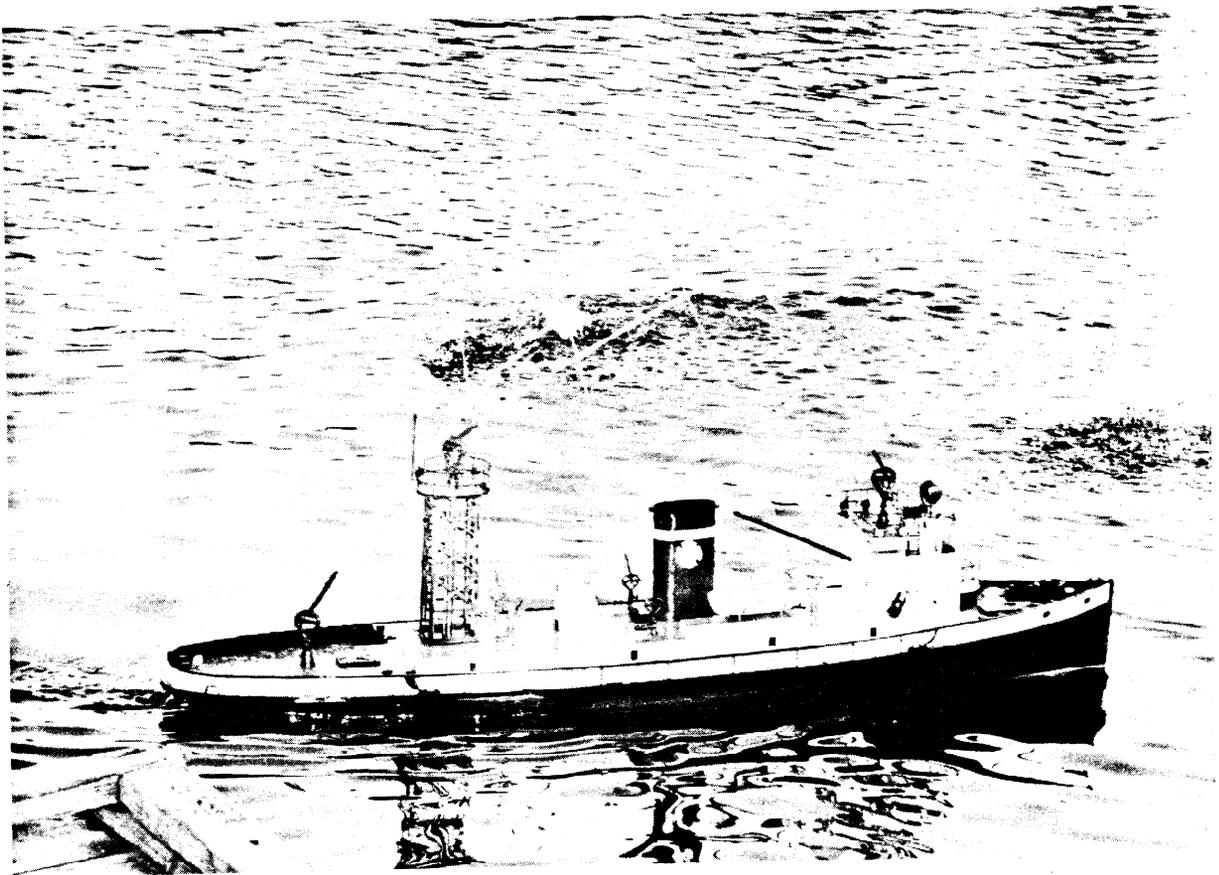


Figure 6B—Model fireboat with all hoses working and controlled by the WS

On-air Measurements of HF Data Throughput Results and Reflections

Ken Wickwire (KB1 JY)
kwick@mitre.org
KB1JY@WA1PHY

1. Introduction

For the past year a team of colleagues and I¹ have been collecting and analyzing data on the throughput and other characteristics of various ARQ protocols available to hams and commercial users for HF work. This activity was motivated by discussions (especially among hams) about the relative merits of the new HF digital modes, such as PacTOR, GTOR, CLOVER II, CLOVER-2000 and PacTOR II. Since the discussions often centered on throughput in various conditions, and we were already running several of the protocols, we decided to see for ourselves. This paper describes our assessment approach and measurement campaign, gives a summary of our main conclusions, and lists some findings worth noting before protocol choices are made and protocol performance is compared. The paper treats the packet and TOR modes in detail. More extensive reports on CLOVER II, NOS TCP/IP and the ALE order-wire will appear elsewhere.

2. Our Approach to Throughput Measurement

The randomly varying HF “channel”; that is, the combination of propagation conditions (fading, dominant ionospheric layer, etc.), and propagated and local noise, is generally agreed to be the worst radio channel. Over the past 20 years, powerful DSP techniques have been developed to tame this wild conduit and put it to work for data transmission, even when it resists being used for voice traffic. These techniques are now embodied in a surprisingly large and growing number of data transmission protocols whose performance is often impressive by HF standards. What people mean when they say (or write in an advertisement) that one of these protocols is better than another is not always clear, however.²

HF data transmission protocols can be divided for general discussion into two categories: those with automatic repeat request (ARQ) and those without. In some cases, the latter are called forward error correction (FEC) protocols, because they use FEC but not ARQ to control errors. ARQ protocols, which are almost always combined with FEC in modern systems, generally deliver error-free data, although there is no guarantee that the data will be delivered quickly. Since many users (especially military and governmental users, and operators of forwarding stations) demand **error-free** transmission, ARQ protocols have come to dominate technical discussions of late. For ARQ protocols, the definition of throughput, for example, is relatively straightforward; for protocols without ARQ, which can deliver erroneous data, the concept of HF throughput is more difficult to define. For these reasons we have decided to concentrate on ARQ protocols in our assessments.

There are three basic ways to assess the throughput (and most other kinds of performance) of an HF data transmission protocol. First, you can try to devise a

¹ See the Acknowledgments and reference list at the end of the paper.

² Although we recognize this shortcoming, we sometimes suffer from it ourselves.

mathematical model of what happens during data transmission with it, convert the model, if necessary, into computer code, and run the code (or your pencil) to assess (i.e., predict) performance. While this is sometimes advocated as the best approach by those too lazy, poor or otherwise constrained to try other approaches, it frequently produces unconvincing or incomprehensible results. Many believe that the modeling approach is best suited to the design stage of a new protocol rather than to the performance assessment stage.

Second, you can connect a pair of working systems through an analog or digital HF channel simulator, which can be set up to accurately produce various levels of some of the main phenomena of the HF channel, like multipath spread, fading and noise (usually Gaussian). Using a channel simulator with real hardware and software produces statistically repeatable results and allows reasonable—if not necessarily **convincing**—comparisons of different systems operating in so-called “standard channels,” namely the ones whose statistics are programmed into the channel simulator. A channel simulator cannot, however, reproduce the statistical variations in transmission quality that occur on a real HF channel; it can’t faithfully reproduce those caused by non-Gaussian (e.g., impulse) noise, intermittent and random interference by man-made signals with various waveforms, day-night transitions, and polar and equatorial propagation anomalies.

The third approach is through on-air measurements. This has the advantage that any one measurement is in a sense completely realistic and convincing, but the disadvantage that the conditions in which the measurement was taken are not generally repeatable. This means that producing *statistically* convincing assessments with this approach requires that a large number of measurements be made (resulting in a large sample-size) and that attention be paid to realistic and representative path lengths, power levels, antennas, diurnal variations and the spreads (variances) of performance statistics. This takes time and a lot of cooperation from several outlying stations.

We believe that a combination of channel simulator and on-air measurements leads to the most convincing assessment of ARQ performance in the HF bands.³ The simulator creates repeatable channel extremes, while properly conducted on-air measurements comprise channel conditions the simulator hasn’t been (or can’t be) set for. This paper discusses a measurement campaign we’ve pursued in that belief for the past year. We should note that although our results allow an informative comparison of the throughputs of the protocols we’ve treated, the past year’s measurements need to be continued to cover all seasons with all protocols and a wider range of sunspot numbers.

For our on-air measurements we try to write software that allows tests to be run automatically, so that the mistakes—that we all tend to make during manual time-recording and data-entry can be avoided. (Sometimes—as with CLOVER and NOS TCP/IP implementations,—protocols come with their own interface software, and we use the existing software capabilities. That leads to some manual data logging.) So far, our software has been written in C for the Macintosh operating system, but it would work (with different I/O calls) on different operating systems.

With our software, we always measure file transfer time from the start of **character-by-character** uploading of a file to the sending modem to the time that a “message saved” (or equivalent) sent by the receiving station arrives via the sending modem to the test

³Of course, this is only true when the two approaches produce results that agree, at least qualitatively. For some simulator results that agree qualitatively with our measurements, see Refs. 1, 2 and 3.

program⁴. Waiting for a “message saved” makes the transfer times a little longer than they would be if a human operator (or program) at the receiving station recorded the error-free arrival of the file. That in turn makes the throughputs slightly lower (i.e., more conservative) than they would be otherwise, but that is a small and reasonable price to pay for measurements that require only one program and no operator intervention during tests.

In the next section we’ll describe the philosophy behind our choice of equipment and file types for most of the measurements made during our campaign.

3. Our Philosophy for Choosing Equipment to Use and Files to Be Sent

In choosing equipment (e.g., the KAM for PacTOR assessment) for our tests, we have been motivated not only by expediency (we have KAMs), but by the view that assessments of most interest to the most (prospective) operators are those of “common” operating setups; that is, ones widely available at competitive prices, and ones that offer a wide choice of operating modes and good technical support. While it is no doubt true that some implementations of PacTOR, for example, may have higher throughput than others because they use A/D quantization of bit energy or more advanced filtering, they are probably not in wide enough use to be part of a “common” operating setup. Nevertheless, if we had the time and money to buy and test all possible pairings of implementations of a particular protocol, we would gladly do it, since performance of the “best” or the “official” implementation is obviously of interest. In the meantime, we unselfishly invite others to fill in the gaps left by our work.

Likewise, we have chosen at this stage ASCII English text files of various sizes to compare the transfer capabilities of protocols. With due respect to the many who probably send text files written in other languages, we believe that sending such files represents a “common” application of the HF ARQ protocols described below. It should be borne in mind that languages other than English and German, and files with a non-standard distribution of characters (e.g., all upper-case characters), may benefit very little from the Huffman text compression used in current PacTOR implementations.

When a protocol like PacTOR, GTOR or CLOVER II comes with defaults for some of its “protocol-tuning” parameters (e.g., GTTRIES and GTUP for GTOR and BIAS for CLOVER II), we have used these defaults. This has been based on the belief that a common setup would not have these parameters changed. (Optimal tuning of such parameters is an area that should be looked at, however, and a few operators have recently started to do so.) For packet, on the other hand, we consider good values of PACLEN and MAXFRAMES to be highly dependent on channel conditions, and we juggled these values frequently to increase throughput in our tests (see below).

Finally, our philosophy says that if a protocol or common implementation offers data compression, then it should be used (if there’s a choice) unless we think it might seriously expand a file (see the section below on data compression). This means that in the case of PacTOR we used Huffman compression and in the case of CLOVER II, we used the “PKLIB” compression (probably a Liv-Zempel-Welch variant) offered by the standard (i.e., “common”) P38 terminal software provided by HAL with the modem.

⁴That is, we don’t include linking and “negotiation” times in our throughput calculations. Others may view these times as legitimate components of transfer times.

In the next section we'll discuss some of the protocols we have assessed in the past year of our campaign along with some more advanced ones we may get to later.

4. ARQ Protocols Developed for HF Use and Their Throughput

Table 1 at the end of the paper lists most of the ARQ protocols that are in common use on the air⁵. With the exception of the last three, all are used in amateur work, and the last three, developed originally for military use, will probably enter the amateur world in some form in the next few years.

The table classifies each protocol according to its modulation scheme, signaling bandwidth, forward error correction capability, ARQ scheme, channel rate, character format, compression capability and measured throughput for "standard" (mainly lower-case) English text files. The throughputs with a O-symbol beside them have been measured as part of our campaign, with enough samples for statistical significance in current sunspot conditions. The other throughputs are based on channel simulator measurements. The measured throughputs for packet and the TOR modes are from an aggregate of short near-vertical-incidence skywave (NVIS) and longer one-hop skywave (OHS) paths. For CLOVER II and the ALE engineering orderwire, the throughputs are from only NVIS paths. (We expect to begin OHS tests with CLOVER II this summer, and to publish ALE, NOS TCP/IP and CLOVER results this year.)

It should be kept in mind that in agreement with our measurement philosophy, for our packet and TOR throughput measurements we have used Kantronics KAMs with firmware version 7.1 or higher. Other PacTOR implementations than the KAM's may yield higher or lower throughputs than ours. Note also that we have used the HAL P38 for all of our CLOVER II measurements; more expensive models, like the PCI-4000, have the computing power to select a 16-symbol signaling set, and may produce higher throughputs.

5. Differences Between NVIS and OHS Throughput for TOR and Packet

NVIS throughput is generally lower than throughput over "standard" one-hop skywave (OHS) paths; that is, fairly long paths on which fading (and resulting inter-symbol interference) is relatively slight, and average signal-to-noise ratios are comparatively high. In fact, one-hop skywave measurements paint a relatively optimistic picture of what operators can expect in day-to-day communications over HF.

However, some protocols appear to improve more than others when you go from NVIS to OHS operations. Tables 2 and 3 below (reprinted from recent papers listed in the References) give NVIS and OHS throughput and other statistics for AMTOR, PacTOR, GTOR and packet. (Recall that for packet, we juggled PACLEN and MAXFRAMES to increase throughput.)

Throughputs in the tables are in characters/sec and times are in seconds. The first column gives the average throughput and its standard deviation, the average throughput per Hertz, the standard deviation of the mean throughput and the maximum observed throughput. The second column gives the number of links and the mean and standard deviation of the

⁵A recent newsgroup FAQ on signalling formats lists a number of ARQ protocols in use in Europe, the CIS and Asia that we never heard of, so we may be misleading our readers with this statement. Most of these protocols may be rather old and inefficient, like AMTOR, but we can't be sure.

“link time.” The third column gives the number of “negotiation times” and the mean and standard deviation of the negotiation time. Link time is the time (in seconds) between sending the link command and receipt by the program of the “LINKED TO” notification. Negotiation time is the time between sending the link command and the start of message-file transfer. In most cases there are fewer negotiation than link times because we started measuring the former part way through the campaign. The fourth and fifth columns give the means and standard deviations of the transfer time and the number of transferred characters.

The standard deviation of the mean (equal to the standard deviation of the throughput divided by the square root of the sample size) is an assessment of the variability of the mean itself (which has its own statistical variability). The *sd_mean*'s in the tables suggest that our sample sizes are big enough to give us pretty high confidence that if we collected many more throughput measurements under roughly the same conditions, we would not get average throughputs that differed from the ones above by more than about a character per second.

To calculate the average throughputs per Hertz [$E(\text{tput}/\text{Hz})$], we divided the average throughput by the average signaling bandwidth. We calculated the latter using the formula for “necessary telegraphy bandwidth” (from the 1992 Dept. of Commerce *RF Management Handbook*) $BW = \text{baud rate} + 1.2 \times \text{shift}$, where *shift* for most of our TOR and packet tests was 200 Hz. For AMTOR, the baud rate is of course 100; for PacTOR, GTOR and packet, we used the rough average of the baud rates chosen automatically in the PacTOR and GTOR modes and manually in packet. Our estimates of these average baud rates were 150 (PacTOR), 200 (GTOR) and 200/300 (NVIS/OHS packet). The resulting average bandwidths were AMTOR: 340 Hz, PacTOR: 390 Hz, GTOR: 440 Hz and NVIS/OHS packet: 440/540 Hz.

The majority of NVIS measurements were at 3.606 MHz LSB, with some at 7.085 MHz LSB and 1.815 MHz LSB. They were made during the winter over all daylight hours and also in the evening, after dark; a few were made in the middle of the night. Interference usually prevented throughput tests from about six to ten in the evening (2300Z-0300Z) on 3.606 and 7.085 MHz.

Most of the OHS measurements were at 10.141 MHz LSB. About 20% were taken at 3.640 MHz, 14.075 MHz, 14.123 MHz or 18.075 MHz, all LSB. These measurements were made during the winter and spring over all daylight hours and also in the evening, after dark. However, interference often prevented throughput tests from about six to ten in the evening (2300Z-0300Z) on 3.640 MHz. The NVIS and OHS tests covered roughly the six-month period from November, 1995, to April, 1996.

All measurements were made using transmitter output of around 100 watts, and all stations generally used sloping longwires or dipoles. These setups can be viewed as embodying average station capabilities. NVIS paths (in New England) were from 30 to 200 miles long and OHS paths (on the east coast and from New England to the midwest) were from 400 to 1200 miles long.

In discussing the TOR and packet results let's start with some observations on NVIS and OHS communications quality in general. First of all, note that we haven't collected data on the fraction of tries in each mode that we were successful in linking, “negotiating” and transferring a file. However, we have found that over OHS paths, the three TOR modes and packet can get files through in the absence of strong interference on most tries during the day. This is in contrast with our NVIS results, which showed that except during the

mid-morning and mid-afternoon “windows,” packet and AMTOR had transfer probabilities well below one.

Under difficult conditions (especially those on NVIS paths leading to marginal SNRs) PacTOR occasionally out-performed GTOR in terms of throughput, although GTOR has higher average throughput. This seems to confirm the rumor that GTOR needs high SNRs for high performance. However, this “role-reversal” happened much less frequently over OHS than over NVIS paths.

In the early evening on both NVIS and OHS paths, there was sometimes increased interference on the frequencies we used. During these periods of interference it was rare to see a file transferred. (An automatic link establishment (ALE) system, such as prescribed in MIL-STD-188-141A, could probably have found a frequency without interference.)

Table 2. Statistical Summary of NVIS Throughput Data

Mode	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No_links E(lnk_tm) sd(l_tm)	No_neg_tms E(neg_tm) sd(neg_tm)	E(xfer_tm) sd(xfer_tm)	E(No_char) sd(No_chr)
AMTOR	5.20 cps 1.13 cps 0.015 cps/Hz 0.08 cps 6.33 cps	226 3.02 s 3.16 s	70 82.4 s 30.1 s	473.5 s 234.0 s	2358.1 974.7
PacTOR	17.83 cps 5.50 cps 0.046 cps/Hz 0.30 cps 25.10 cps	344 5.44 s 8.39 s	95 38.7 s 22.7 s	146.1 s 90.0 s	2452.7 1110.1
GTOR	23.52 cps 10.06 cps 0.053 cps/Hz 0.55 cps 44.12 cps	335 5.54 s 10.30 s	76 58.7 s 30.9 s	120.0 s 95.8 s	2531.7 1580.3
packet	5.68 cps 3.53 cps 0.014 cps/Hz 0.25 cps 17.34 CDS	197 8.73 s 10.48 s	119 102.7 s 66.9 s	556.7 s 367.6 s	2484.9 1043.1

Turning to particulars, you can see that AMTOR and PacTOR average throughputs don’t differ much on OHS and NVIS paths, although there is a slight tendency toward greater statistical variation (as measured by standard deviations) on NVIS paths. This similarity of average throughputs may explain why you don’t hear much about differences between performance on long and short paths in these two modes.

The big story is the differences between GTOR and packet performance on long and short paths. Average GTOR throughput on OHS paths was almost 50% higher on OHS paths

than on NVIS ones (32 char/s vs 23 char/s). This may reflect the presence of consistently higher signal-to-noise ratios (SNRs) on OHS paths, since GTOR is said to thrive on high SNRs and suffer more than the other modes on low ones.

Packet throughput was two-and-a-half times higher on long paths than on NVIS ones (16 char/s vs 6 char/s). Some of this difference may have been caused by the fact that we restricted all our NVIS tests with packet to 200-baud operation. Although we based this restriction on observations of performance, it's possible that a more aggressive choice of baud rate on packet during the mid-morning and mid-afternoon "NVIS windows" could have raised NVIS packet throughput somewhat. However, this does not explain all of the improved performance, whose source must be the better OHS channel (fewer packet bit errors).

Another striking difference appeared in the average packet negotiation times (OHS: 35 s, NVIS: 103 s). (Recall that negotiation time is the difference between the time a connection request is sent and the time file transfer starts.) This difference in average negotiation times apparently reflects the fact that the negotiation process for a packet BBS upload, which involves transmission of frames of various sizes, exposes packets at 200 baud much higher bit-error rates on NVIS paths than 300-baud negotiations over OHS paths.

Table 3. Statistical Summary of OHS Throughput Data

Mode	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No_links E(lnk_tm) sd(l_tm)	No_neg_tms E(neg_tm) sd(neg_tm)	E(xfer_tm) sd(xfer_tm)	E(No_char) sd(No_chr)
AMTOR	5.70 cps 0.80 cps 0.017 cps/Hz 0.08 cps 6.33 cps	104 2.62 s 3.81 s	92 69.7 s 15.2 s	543.2 s 109.8 s	3009.6 98.1
PacTOR	20.19 cps 5.49 cps 0.052 cps/Hz 0.44 cps 25.00 cps	153 4.70 s 6.53 s	139 40.8 s 29.4 s	176.1 s 105.2 s	3058.8 308.5
GTOR	32.30 cps 9.88 cps 0.073 cps/Hz 0.79 cps 44.12 cps	158 4.44 s 7.96 s	144 50.9 s 21.7 s	119.9 s 102.6 s	3126.6 501.4
packet	15.67 cps 4.58 cps 0.029 cps/Hz 0.44 cps 24.59 cps	108 6.46 s 8.50 s	108 34.6 s 17.7 s	221.9 s 141.4 s	2975.0 259.8

Maximum observable TOR throughputs were about the same for NVIS and OHS paths, although, as mentioned above, individual measurements came closer to their maxima

more often on the long than on the short paths. On packet, we achieved maximum OHS throughput of about 25 char/s vs about 17 char/s for NVIS.

6. Discussion of Packet Results

Our packet experiments over both NVIS and OHS paths have led to surprising results in view of what we have read on newsgroup discussions and elsewhere. For example, over OHS links we have consistently achieved average packet throughputs two to three times higher than average AMTOR throughputs, although not quite as high as PacTOR, and only about half of the GTOR average (see Table 3 above). The parameters we have adjusted to do this are PACLEN, MAXFRAMES, FRACK, SLOTTIME, RESPTIME and PERSIST, and we have done all our OHS file transfers at 300 baud. (Since we have tried to choose frequencies and times where there is little interference, we have set PERSIST very high and FRACK, SLOTTIME and RESPTIME low for aggressive use of the channel.)

These high packet throughputs have been achieved, however, *only during the day*, and by means of very frequent, manual, changes of PACLEN and MAXFRAMES. Furthermore, we have managed to find frequencies that were by and large free of significant interference from other signals (this appears to rule out most of the 20m band). For example, we have often been able to transfer files over both NVIS and OHS paths with combinations like PACLEN = 100 and MAXFRAMES = 5 in the absence of contention, which may be a revelation to some hams who have tried HF packet.

As a general rule, as packet begins to work in the morning on our links, values of PACLEN/MAXFRAMES around 40/1 work best. From mid-morning till late afternoon, combinations like 80-100/4-7 often lead to high throughput. As the bands begin to deteriorate, it's back to near 40/1. PACLENS greater than about 120 bytes usually suffer too many bit errors on our NVIS and OHS links to be worth trying.

After about 5 PM local time during the winter, throughput rapidly falls, and for most of the evening, getting files through in any mode was difficult. (We got some NVIS transfers through in the middle of the night during the winter, but we didn't try any OHS transfers in the middle of the night.) On our links, trying a lower ham frequency in the evening usually led to increased interference, against which none of the modes did a great job.

Our experience with HF packet on OHS and NVIS links has convinced us that an adaptive protocol that adjusted HBAUD, PACLEN and MAXFRAMES using feedback on throughput could go a long way toward polishing HF packet's tarnished reputation. However, with much better systems now available at reasonable prices, it is probably no longer worth developing such a protocol.

7. The Effects on Throughput of Data Compression and File Type

Three of the protocols we have assessed over the air provide one or more types of optional or hard-coded data compression: PacTOR has (optional) Huffman compression, GTOR has hard-coded Huffman and run-length compression and CLOVER II with the HAL interface software has one or more hard-coded compression techniques from the so-called "PKLIB" suite. Other and future protocols may also include one or more

compression capabilities⁶. Of course, even when a protocol doesn't include compression, the user is free to compress his files off-line before he sends them, provided that the protocol can handle the compressed format and the receiving station can de-compress the files. (For an introduction to data compression see Ref. 7.) As mentioned above, we almost always choose the **Huffman** option in **PacTOR** transfers of English text files.

How much a file gets reduced by a compression technique is strongly affected by the file's type and the technique's approach, so that the user must have some understanding of the interplay of the two if he wants to use compression for high throughput.

In general, the closer the distribution of a file's ASCII characters to the distribution of characters in "typical" English (or other language to which the **Huffman** code has been tailored) text, the more **Huffman** will compress the file. The more repeated contiguous characters or bytes (e.g., spaces) in the file, the more run-length coding will compress it. The more repetitions of byte-pairs in a file ("an," "th" in "the," etc.), the more so-called Markov coding (multi-level **Huffman**) will compress it. The more repeated byte strings in the file, the more "dictionary-based" methods like Lempel-Ziv-Welch (LZW) compression will squeeze it⁷. Finally, the bigger the monochromatic patches (e.g., big expanses of white background) in a graphics file, the smaller a graphics compression technique (like those used in JPEG and for GIF files) will make it.

These facts means that if you send a text file consisting of a high proportion of upper-case characters with **PacTOR**, you won't get much benefit from **Huffman**, which relies (in most **PacTOR** implementations) on a fixed text-character distribution in which certain lower-case characters (like "e") occur with relatively high frequency. Likewise, if you compress a file off-line (e.g., zip it), you produce a compressed (g-bit, or binary) file that looks a lot like a pseudo-random string of bytes. If you then apply a built-in compressor like one from the **PKLIB** suite used in the P38 **CLOVER** software from HAL, you will find that the "compressed" file is actually a bit larger than the zip-file. (Of course, this is all right if the zip-ing did a good job.)

On the other hand, if you try to send an uncompressed executable (".exe") image as a binary file with **CLOVER II** and the HAL software, you'll find that the already pseudo-random structure of most executable (binary) files is likewise expanded rather than compressed by **PKLIB**. To get efficient transfer by **CLOVER II**, you should compress executables off-line before submitting them to the HAL P38 terminal software.

Graphics files (not yet the main focus of our throughput experiments) are another story. If they're GIF or JPEG files, they're generally already compressed, so **CLOVER** and most other compressors won't make them any smaller*. **PICT** and **BMP** files, on the other hand, are not compressed, and often have big monochromatic chunks, so that the

⁶The TACO2 protocol suite developed by the DOD for transmission of battlefield-situation graphics over HF has data compression as an integral part.

⁷For this reason it is not a good idea to compare throughput for (cooked-up) files that consist of repeated sections (for example, those made by repeatedly pasting a section to the end of the file). LZW will generally compress such a file by much more than 50%, whereas **Huffman** will only compress it by as much as it compresses the first section. The resulting comparison is therefore probably unfair to the **Huffman**, since in many cases one would send just the first section of such a file with the advice that it is to be repeated N times at the receiver for whatever reason.

⁸The popular shareware **EXPRESS** terminal program that also runs the P38 and other HAL **CLOVER** hardware offers built-in compression of files and tailored compression and transmission of graphics images. Since **EXPRESS** doesn't (yet) fall under our definition of a "common" implementation ("comes with the modem"), we don't cover it here.

PKLIB compressor(s) in the HAL software usually make them a lot smaller, with correspondingly higher throughput.

So far in our NVIS experiments with CLOVER II using both compressed and uncompressed files we have found that compression plays a crucial role in the relatively high average throughput (above 40 characters/s) we report in Table 1. (Recall that this average applies to compressed English text files, and that OHS transfers are not included in our CLOVER data.) The average CLOVER II throughput over NVIS paths for *uncompressed* files is only around-25 char./s, which is about the same as the GTOR throughput of text files⁹.

As we pointed out in Section 3, we have not generally sent off-line-compressed files for throughput comparison, so as not to penalize unfairly common implementations of protocols (like AMTOR and standard AX.25 packet) that can't easily handle binary files. The field of throughput comparison using compression techniques that aren't part of "common implementations" is a wide open and important one.

8. Concluding Remarks

One of the conclusions we've reached in our throughput assessments is that hams and others need to separate long from short distance paths when they compare HF throughput performance of ARQ modes, especially the amateur GTOR and packet modes. Some moderation of opinion on HF packet performance may also be called for.

For HF data transfer, data compression plays a crucial role in increasing throughput, and it should always be used when it significantly lowers file or message size and the receiving station is equipped to handle decompression.

We hope that our throughput data will further clarify discussions of the HF digital modes. Our results should put throughput measurements of PactOR II and other HF data-transmission systems in perspective. We have plans to report someday on the performance of some of those newer modes, and encourage those already in a position to do so to publish their measurements.

Acknowledgments.

I'm grateful to Mike Bernock (KB1PZ), Dennis Gabler (KB5HVN), Doug Hall (KF4KL), Richard Harrison (NT2Z) and Bob Levreault (W1IMM) for handling numerous requests to put their stations on the air in various modes on various frequencies, and for regular mailbox cleanings.

Mike Bernock and Bob Levreault made a number of useful comments on the text of the paper.

References

1. Young, T., et al., "A Preview of HF Packet Radio Modem Protocol Performance," 13th ARRL Digital Communications Conference, Bloomington, Minn. 1994.

⁹As already mentioned,GTOR applies Huffman and run-length compression to all transferred files.

2. Reynolds, P., "HF channel Simulator Tests of Clover," *QEX*, Dec. 1994.
3. Riley, T., et al., "A Comparison of HF Digital Protocols," *QST*, July 1996.
4. "HF Throughput Truths," in Packet Perspectives, *QST*, Feb. 1996.
5. Wickwire, K., et al., "On-air Measurements of HF TOR and Packet Throughput, Part I: Near-Vertical-Incidence-Skywave Paths," *Digital Journal*, March 1996.
6. Wickwire, K., "On-air Measurements of HF TOR and Packet Throughput, Part II: One-hop Skywave Paths," *QEX*, June 1996.
7. Nelson, M. and J-L Gailly, *The Data Compression Book*, 2nd Ed., M&T Books 1996.

Table 1. Reliable Protocols for HF Communication

Protocol	Modulation	Bandwidth (Hz)	FECA	ARQ	Chan. Rate (Symbols/s)	Character Format	Compression	Avg. Tput (char./s)
NOS TCP/IP+	BFSK	≈ 500	NO	CRC§	≈ 200#	8-bit ASCII	No	2-6◇
AX.25 Packet	BFSK	≈ 500	NO	CRC*	≈ 200#	7-bit ASCII	No	110
AMTOR	BFSK	≈ 500	NO	parity check§	100#	Baudot	No	50
PacTOR	BFSK	≈ 500	NO	CRC**	100/200#	8-bit ASCII	Huffman	190
GTOR	BFSK	≈ 500	Golay	CRC**	100/200/300#	8-bit ASCII	Huffman/run-len	280
CLOVER I	BPSK-8PSK†	500	Reed-Solomon	CRC***	31.25#	8-bit ASCII	PKLIB (LZW?)	40-500
PacTOR II	BPSK-8PSK	450	Convolutional	CRC**	< 300#	8-bit ASCII	Huff/run-len/Markov	40-50
CLOVER-2000	00 BPSK-16PSK¶	2000	Reed-Solomon	CRC***	31.25#	8-bit ASCII	PKLIB (LZW?)	160-200
ALE EOW⊠	8FSK	≈ 1800	Golay	CRC**§	125	7-bit ASCII	No	70
FED-STD-1052	BPSK-8PSK‡	≈ 2700	Convolutional	CRC***	2400	8-bit ASCII	TBD	150
SHAPE TC	BPSK-8PSK‡	≈ 2700	Convolutional	CRC***	2400	8-bit ASCII	TBD	150

- # Amateur channel rates are limited by an (outdated) FCC restriction to 300-baud operation at HF. (Shown rates are those **settable** by an adaptive protocol or those we set for on-air measurements.)
- + TCP/IP can go much faster with much higher throughput when a more robust modem is used.
- A Protocols with interleaving: GTOR, PacTOR II, ALE, FED-STD-1052, SHAPE Technical Centre.
- § Stop-and-Wait ARQ.
- ◇ On-air measurements using **text** files over actual NVIS or OHS paths.
- * Stop-and-Wait or Go-Back-N ARQ.
- ** Stop-and-Wait with “memory ARQ.”
- **§ Stop-and-Wait with memory ARQ, up to six retries.
- *** Selective Repeat ARQ.
- † For the P38. Other models can go up to 16PSK and 16QAM (quadrature amplitude modulation). (The “PSK” waveforms are actually four-tone hybrids that switch phases during zero-amplitude intervals.)
- ¶ Also uses various QAM modulations.
- ⊠ Engineering order wire (not implemented in every ALE system).
- ‡ Usually employs MIL-STD-188-110A serial-tone modem and sometimes ALE to choose channel.

Notes

1. PacTOR, GTOR, CLOVER, PacTOR II, FED-STD-1052 and SHAPE TC are automatically adaptive.
2. CLOVER II average throughput is probably higher with models that can use 16 signalling elements.
3. Assessed linking probabilities: high (CLOVER II, ALE); moderate (AMTOR, PacTOR, GTOR); low (NOS TCP/IP at night, AX.25 packet at night).

On-air Measurements of MIL-STD-188-141A ALE Data Text Message Throughput over Short Links

Ken Wickwire
kwick@mitre.org

For the past six months a colleague WILLIAM and I have conducted automated measurements of throughput when ASCII text files are sent over short or “tactical” HF paths using the ALE Data Text Message (DTM) engineering orderwire (EOW). This is a preview of our results.

Tactical HF links generally use either surfacewave or near-vertical-incidence skywave (NVIS). Surfacewave works out to about 50 miles and NVIS to about 300 miles. Multipath, D-layer absorption and interference usually affect NVIS, which usually has lower throughput than communications over “standard” (i.e., long) one-hop skywave paths.

ALE systems’ ability to measure channel quality, and use it to make choices of good channels, now offers improved performance over tactical paths, ALE systems employ a slow but robust waveform that uses interleaving and two kinds of forward error correction (FEC) to combat the fading, noise and interference of HF channels.

ALE standards prescribe three engineering orderwire protocols for half-duplex data transfer: the Automatic Message Display (AMD) mode, the Data Text Message (DTM) mode and the Data Block Message (DBM) mode. DTMs can transfer ASCII text using the ALE waveform and an ARQ protocol.

We have carried out more than 200 measurements of throughput (in char/s and char/s/Hz) using DTMs on a 35-mile path. We have used 125-watt Harris RF5022 ALE radios, which implement DTMs of constant size (300 bytes) and “memory ARQ,” in which up to six erroneous repeats of a message segment sent as a DTM are stored and compared when necessary in an attempt to construct an error-free segment. The RF5022’s ALE firmware segments messages longer than 300 bytes into 300-byte DTMs. These DTM-segments are ACKed one at a time. Experiments suggest that messages 300 to 1000 characters long produce the highest throughput consistent with shortest run time.

Our two stations used broadband sloping longwires that allowed the radios to drive the antennas without tuners. The antennas have both vertical and horizontal components, so that they can launch both surfacewave and NVIS signals.

The ALE modems were programmed to try frequencies between 2 and 16 MHz. (IONCAP runs suggested that any link above 8 or 9 MHz probably used surfacewaves, which were chosen frequently at night, when interference was heavy.) The tests covered seven months from September, 1995, when the average sunspot number was near the bottom of its cycle.

Our measurements were automated by two C-programs. The first runs throughput tests. At the start of a test, and between DTM transfers, the receiving station is scanning the set of programmed frequencies and will stop upon hearing a call. If a link occurs, the calling and receiving stations negotiate a DTM transfer and the sender begins sending the DTM.

The program usually starts by performing a link quality assessment (LQA) exchange, which gives both stations up-to-date info on channel quality. After the exchange, the calling radio links with the receiving one. When the calling radio informs the program

that a link has been established, the program commands the local radio to list the channel number and the corresponding LQA scores for later analysis.

When the LQA scores for the linking channel have been logged, the program prepares the local radio to send an orderwire. It then uploads a standard English ASCII text file for transfer. After sending commands that require a particular response, the program invokes a function called `chkresponse ()` that scans the serial-port input from the local radio for the appropriate response (“LINKED”, “MESSAGE RECEIVED”, etc.).

As the program runs, timers set by the computer’s clock measure “link time” and “message transfer time.” Message transfer time is the time between start of character-by-character uploading of the file to the local radio and receipt of the MESSAGE RECEIVED notification. Transfer time does not include link time. The program calculates throughput by dividing the number of characters sent by the message transfer time. Since the message transfer time includes the few seconds needed for the receiving station to send the MESSAGE RECEIVED frame, the throughput measurements are slightly pessimistic.

The throughput-measuring program writes its results to an archive file. The archive file stores appended, time-stamped data in abbreviated format that can be analyzed off-line by a statistics program. Here is abbreviated statistical output for all the tests run up to 6 May 1996:

```
no_ALE_links = 235
E(link_time_ALE) = 25.96 s, sd(link_time_ALE) = 21.42 s
E(transfer_time_ALE) = 139.9 s, sd(transfer_time_ALE) = 98.7 s
E(no_file_chars_ALE) = 770.4, sd(no_file_chars_ALE) = 615.1
E(tput_ALE) = 5.21 cps, sd(tput_ALE) = 1.16 cps, sd(mean_tput_ALE) = 0.08 cps
max_thruput_ALE = 6.60 cps, E(thruput_ALE/Hz) = 0.003 cps/Hz
```

```
Linking histogram:
Channel 1 (2.394 MHz): 1 link
Channel 2 (2.824 MHz): 70 links
Channel 3 (3.166 MHz): 29 links
Channel 4 (4.565 MHz): 8 links
Channel 5 (5.031 MHz): 56 links
Channel 6 (6.870 MHz): 4 links
Channel 8 (7.850 MHz): 1 link
Channel 9 (9.305 MHz): 7 links
Channel 10 (10.330 MHz): 3 links
Channel 11 (10.523 MHz): 5 links
Channel 12 (13.692 MHz): 45 links
Channel 13 (15.487 MHz): 6 links
```

`E()` and `sd()` stand for the expectation (average) and standard deviation of a measurement. About two-thirds of a set of measurements will be within one standard deviation of their mean and over 90% will be within two. The `sd (mean_tput_ALE)` here suggests that our sample sizes are big enough to give us high confidence that if we collected more throughput measurements under roughly the same conditions, we would not get average throughputs that differed from the one above by more than a tenth of a character per second. One should keep in mind that our “conditions” correspond to winter and spring operations at low sunspot numbers. Average throughput in other seasons and at much higher sunspot numbers will probably be different.

To calculate the average throughputs per Hertz [`E (thruput_ALE/Hz)`], we divided the average throughput by the ALE signaling bandwidth. For the latter we used the formula

for “necessary telegraphy bandwidth” (from the 1992 Dept. of Commerce *RF Management Handbook*):

$$BW = R / \log_2(N_T) + f_{\max} - f_{\min},$$

where R (= 375 bits/s) is the channel rate, N_T (= 8) is the number of MFSK tones in the ALE waveform, f_{\max} (= 2500 Hz) is the highest tone and f_{\min} (= 750 Hz) is the lowest tone. For these values, the signaling bandwidth is 1875 Hz. At the end of its output, the statistics program prints histogram values of the frequencies chosen for linking by the sending ALE radio (more on these below).

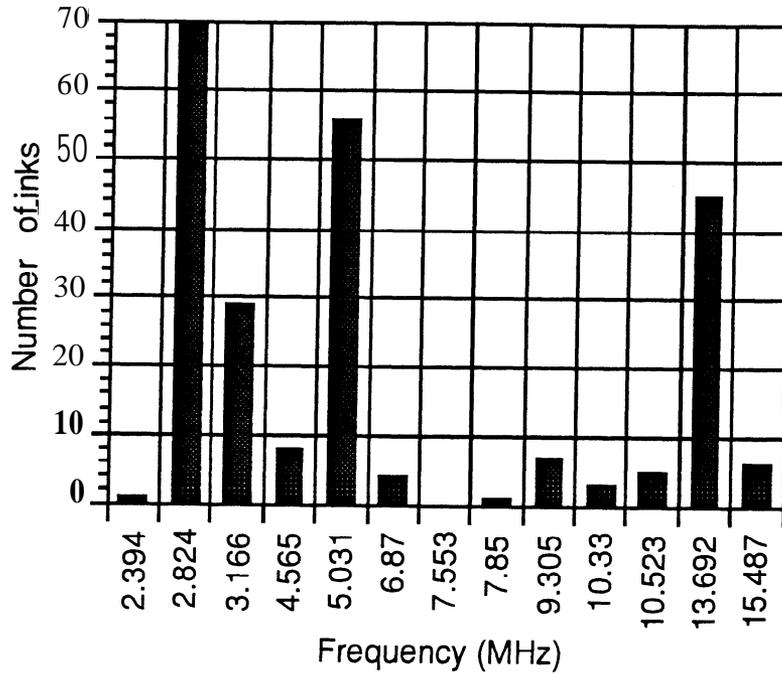
Because of the relatively low channel rate used by the ALE waveform (375 bits/s or 125 symbols/s), and the high overhead used to provide the waveform’s robust forward error correction (about five-sixths of the channel rate), average DTM throughput (about 5 char/s) is modest. What distinguishes ALE DTM transfers from ASCII transfers using several other ARQ protocols, like the AMTOR, PacTOR, GTOR and AX.25 packet protocols, is the fact that on tactical links, DTM transfers are much more frequently successful, especially at night. The main reason for this is ALE’s ability to look for another frequency for linking if the previously tried frequency fails. File transfers were successful on roughly 90% of the automated attempts.

Our ALE systems have probably often linked on surfacewave and possibly E-layer frequencies rather than on NVIS frequencies, which lie near the bottom of the HF band. Surfacewave frequencies appear to have been chosen often at night. (Most traditional propagation prediction programs do not suggest surfacewave frequencies for night time or any other operation.) One reason for avoiding NVIS communications at night is that there is almost always more interference on NVIS frequencies at night than during the day. For short-range communications, it is important to use antennas that can launch both NVIS and surfacewave signals; that is, antennas with both vertical and horizontal components.

The relatively small standard deviation of DTM throughput reflects the DTM protocol’s restricted ability to adapt to changing conditions. The low variability of throughput over short paths and the reliability of transfer are also reflected in the fairly small difference between average and maximum (6.6 cps) observed throughput.

Average link time was about 26 seconds, with a standard deviation of about 21 s. This standard deviation implies that establishing a link required at least two attempts fairly often during our tests. (A single successful link handshake takes about 20 seconds.)

The histogram data produced by the statistics program are graphed below. The histogram shows that most DTMs were transferred at 2.824, 3.166, 5.031 and 13.692 Mhz. The middle two frequencies were chosen mostly during the day and probably supported only NVIS. The 2.824 MHz frequency appears to support both surfacewave and NVIS. (2.394 MHz may have had antenna-matching problems that caused its poor performance.) The transfers on 13.692 MHz were mostly at night and were probably by surfacewave. It is unlikely that an inexperienced communicator would have tried 13 MHz for night-time operation over this link.



These on-air results suggest that although the ALE DTM engineering orderwire mode is relatively slow, on tactical links it can perform ASCII file transfers more reliably than several other ARQ systems in current use.

Acknowledgments.

I am grateful to John Morgan of the Harris Corporation for providing the Harris RF5022 radio systems with which these measurements were made, and to Bob Levreault (W1IMM) for maintaining one of the ALE stations used for the tests.

CLOVER - The Technology Grows and Matures

Lessons Learned and Pleasant Surprises

Bill Henry, K9GWT

The “CLOVER” waveform idea was first presented to radio amateurs by Ray Petit in the July, 1990 issue of ~~QEX~~ narrow-bandwidth mode that required custom radio transmitter and receiver design soon evolved into a more universal waveform for use with any HF SSB transceiver - called “CLOVER-II”.

Development of CLOVER-II was not as simple or as fast as we had expected. It seemed like we were **frequently** “re-inventing the wheel” and development schedules were re-written every week. After a few false starts and a couple “wrong turns”, CLOVER-II was finally shipped and put on-the-air in late 1992. Since then, thousands of CLOVER modems have found their way into ham shacks around the world. CLOVER-II is now available in several different versions of the **PCI-4000**, the P38, and the DSP-4 100 modems as well as in customized systems for commercial and government customers. CLOVER has now been used in every conceivable HF radio application. This includes ham radio, ship **traffic**, aircraft communications, bank data transfer, computer file transfer, image transmission, and even digital voice. Virtually anything that can be digitized and stored in a PC has been sent somewhere via HF radio and CLOVER, **often** at data throughput rates that match or exceed those we see on 1200 baud VHF packet radio.

Early CLOVER Lessons:

CLOVER started with a great burst of enthusiasm. We had a bunch of new ideas and a “hundred or so” ways to implement each one. The new DSP architecture promised unheard of design freedom. This was our first product where virtually *every detail* was set by software. We soon became quite familiar with the phrases: “It’s only software.” - usually followed by - “It will take *how long* to make that change?” We demonstrated the “gas laws” **often**, especially the one that states “gas expands to fill the available volume”. Our version is “Software expands to fill available memory space and consume all processor time”.

CLOVER itself experienced many evolutionary changes. The first version had a bandwidth of 100 Hz, one tone and used only one phase **shift** modulation mode. During development, the CLOVER waveform soon expanded to four tones, a bandwidth of 500 Hz, and a total of 160 different modulation modes - a real “knob-twister’s paradise”! Coming **from** the **HF** packet world, we also intended to fix some of the problems we’d seen with AX.25 on 20 Meters. The CLOVER ARQ protocol therefore includes selective block repeat, in-block error correction coding, bi-directional data flow (no “OVER” command), and adaptive modulation control. While some or all of these ideas had been

used previously, CLOVER was the first to combine the multi-level, multi-mode modulation waveform with an ARQ protocol specifically designed for use over HF radio links. No doubt about it, CLOVER is a *complicated* mode.

Predicting software completion turned out to be very risky business. The first target release date was “April, 1991”. This quickly slipped to “fall, 1991” to “April, 1992” and finally to “November, 1992 (actually the *last* day of November, 1992). AND - we weren't done! During the first 6 months of CLOVER's commercial life, we issued 8 no-cost software up-grades. We've since provided almost **50 software** changes to CLOVER and DSP software, the most recent occurring this summer - July, 1996.

While the hardware schematic and block diagram have been very stable, parts cost and availability have been continuing problems. Price and delivery promises made in 1990 by suppliers have proven to be “optimistic dreams”. This is particularly true of the Motorola DSP components. The cost of these parts remains high and periodically they become “non-deliverable”. In early 1995, frustrated with this high cost and unpredictable delivery, we redesigned the modem to use the much less expensive and more available **TMS320C25** processor and related parts. The **P38** modem is the result, the first and so far only DSP modem with a list price under \$400 (it usually sells in the “mid-\$300 range).

Finally, CLOVER has experienced the typical problems of a pioneering mode. Correctly tuning the radio receiver to exactly match a CLOVER signal takes practice. Virtually all radios made since 1990 can meet the **+10 Hz** tuning increment requirement, but you need a “**soft touch**” on the knob and patience to make small corrections and then wait. Once learned, tuning a CLOVER signal is easy - BUT - it's not like any other mode you've ever used! Likewise, tuning the transmitter for no ALC and less than “Max-Smoke” output takes personal discipline, particularly hard for us old-time RTTY types. These are not new problems in HF radio. The early SSB pioneers faced the very same problems in the 1950's. Tuning-in an SSB signal was a lot harder than tuning an AM signal. 40 years later, we think nothing of tuning an SSB signal - “No big deal”! Likewise, we all soon learned that adjusting a linear amplifier was a lot different than tuning the AM finals for “cherry-red plates”. Inaccurate tuning and incorrect transmitter adjustment severely limit CLOVER performance and have frequently been the underlying problem behind “it doesn't work” complaints. **After** 5 years, CLOVER is finally tuning the corner where we are getting comfortable with using it and can also say “no big deal” about these problems.

Marine CLOVER:

Ships at sea have used the “**SITOR**” mode for **HF** data communications for **30** years. Special frequency channels are allocated for ship-to-shore *Narrow Bandwidth Direct Printing Telegraph (NBDP)* use. “Paired frequencies” are used with ships transmitting on one set of channels and shore stations on another set. Within the “ship” or “shore” **sub-**bands, the channels are spaced exactly 500 Hz apart. Particularly in the case of shore station allocations, every 500 Hz wide channel is in active use, usually by very powerful transmitters (1 **kW** to 10 **kW**). It is therefore very important that adjacent channel

interference be prevented. HAL has teamed with Globe Wireless of Half Moon Bay, California to develop a special version of CLOVER that is tailored to meet ship-to-shore requirements. While the original CLOVER-II waveform has a frequency spectra that is exactly 500 Hz wide (**@ -50 dB**), this leaves no “guard band” between adjacent channels for tuning error. A marine version was therefore developed that has a **-50 dB** bandwidth of 400 Hz, well within the FCC Part 80 limits for use on NBDP HF channels. We call this version “CLOVER-400”. The spectra of CLOVER-400 and the FCC limits are shown in Figure 1.

Globe Wireless has created a “Global Radio Network” of public coast stations distributed around the world. All public coast stations are tied into the global network via whatever common carrier service is most cost effective for that location (wire line, satellite, Internet, etc.). Regardless of its location at sea, a ship can establish **HF** radio contact with one or more network coast stations. When not actively sending data, each ship receiver constantly scans the frequency list of Globe stations, listening to traffic or **SITOR** “free signals”, and logging signal quality data for each station heard. At any given time, the shipboard computer therefore “knows” which frequency and station is optimum for communications. Each ship acts as a “passive sounder”, obtaining “**LQA**” data (Link Quality Assessment) for each usable coast station *without the need to transmit*.

All coast stations are equipped to use either **SITOR** or CLOVER-400 on NBDP HF channels. To maintain compatibility with older vessels, communications start in **SITOR** mode and then switch to CLOVER-400 when the shore station recognizes a **CLOVER**-equipped ship. Use of CLOVER not only increases the speed at which data is delivered, it also provides error-corrected **8-bit** data transfer of any computer file, be it text, data, image, or executable **software**. In fact, new software for the CLOVER modem on each ship is passed via CLOVER on HF radio - the system upgrades itself. All this is accomplished at a fraction of the cost the user would otherwise have to pay for satellite communications.

Voice Bandwidth CLOVER:

While amateur data modes emphasize narrow bandwidth (500 Hz or less) to conserve limited spectrum, commercial and military HF allocations are virtually all for “voice bandwidth” channels (ship-to-shore excepted). The U.S. Civil Air Patrol (CAP) has made **full** use of the four “tone channels” of CLOVER-II to expand the capacity of their limited number of HF channel allocations. The CAP technique is to share an HF voice channel between four non-interfering and independent CLOVER ARQ links. This application takes advantage of CLOVER’s exceptionally high stop-band suppression. Selection of the CLOVER-II “tone channel” is a user-set feature included in all **PCI-4000** and **DSP-4100** modems (not available in the **P38** modem). The CAP CLOVER “multiplex” of a voice channel is shown in Figure 2.

CLOVER-400 Frequency Spectra

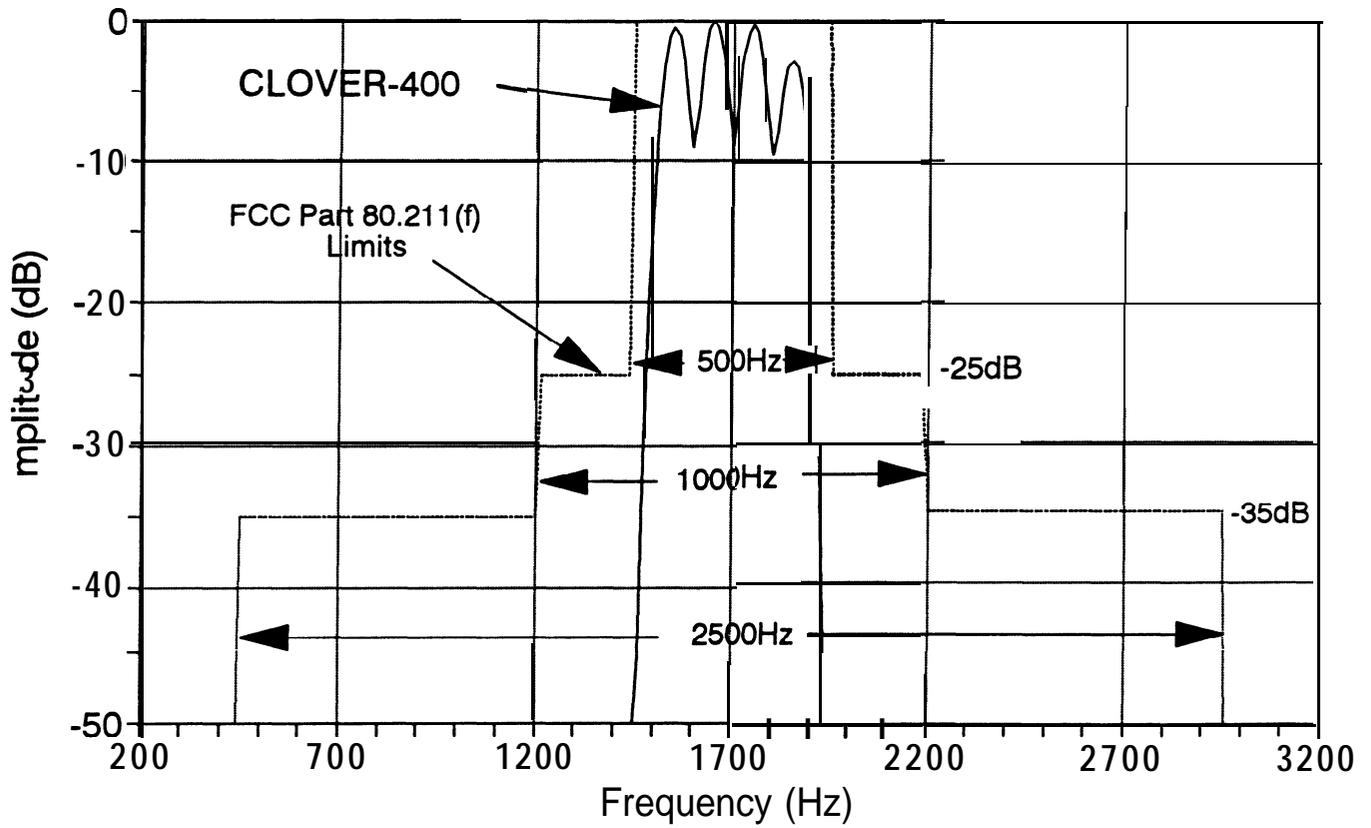


Figure 1.

CIVIL AIR PATROL CLOVER-II Multiplex

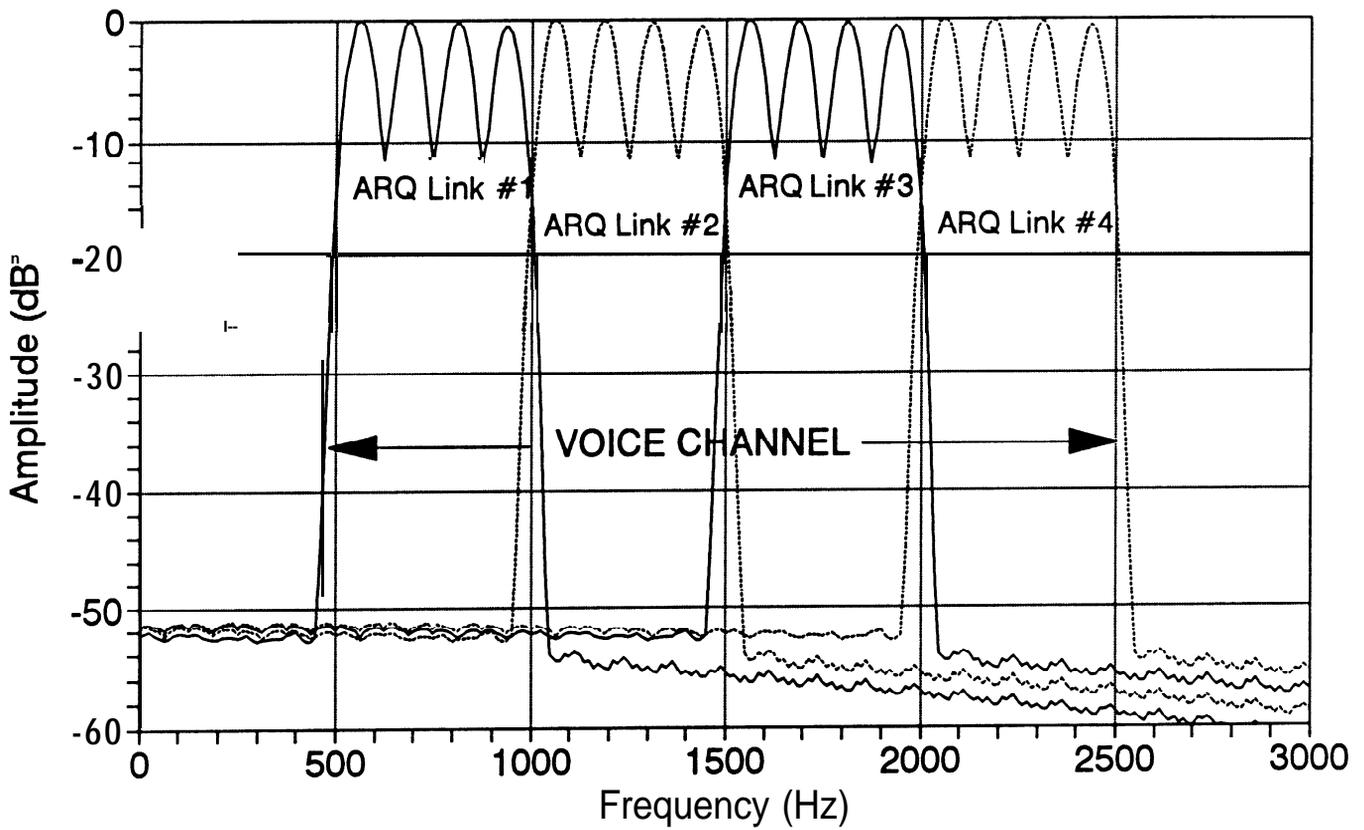


Figure 2.

CLOVER technology has also been expanded to gain higher data throughput by using more tones and a higher symbol rate. "CLOVER-2000" uses 8 tones, a symbol rate of 62.5 Baud, occupies a bandwidth of 2000 Hz, and has a data throughput rate that is approximately 4 times that of CLOVER-II - up to 250 **bytes/sec.** (2000 **bits/sec**). The spectra of CLOVER-2000 is shown in Figure 3, typical throughput characteristics in Figure 4, and ARQ timing information in Figure 5.

On-the-air testing of CLOVER-2000 has shown that it is very robust and often provides better communications than simple scaling of CLOVER-II performance might predict. Over an 800 mile path on 10 MHz at mid-day, CLOVER-2000 has reliably provided data transfer at an average rate of 1000 bits-per-second, and **frequently** running at 1500 to 2000 **bits/sec** for 5 to 10 minute periods. The wider bandwidth and faster symbol rate of CLOVER-2000 (compared to CLOVER-II or **CLOVER-400**) do not cause excessive block failure or repeats. In fact, the short ARQ frame of 5.5 seconds makes CLOVER-2000 very responsive to changes in the ionosphere.

CLOVER Modems:

There are now three modem products that support one or more versions of CLOVER - the PCI-4000, the P38, and the DSP-4100. There have been 4 major versions of the **PCI-4000**, two of which are still in current production. The "standard" PCI-4000, first introduced in 1993, will support CLOVER-II and FSK modes (**RTTY**, **AMTOR**, **Pactor**) but cannot be used with CLOVER-400 or CLOVER-2000. CLOVER-400 capable modems (called "GL-4000") are only available from Globe Wireless. Use of CLOVER-2000 (or CLOVER-400) requires a faster 68000 processor and additional memory. This hardware version is called the "PCI-4000 Plus". All versions of the PCI-4000 use **on-board DSP** and 68000 software that is upload via the PC data bus. Software up-dates can be quickly and easily obtained **from** the **TECHLINE** BBS.

A low cost DSP modem was introduced in 1995 to provide CLOVER technology to radio amateurs at minimum cost. As noted earlier, it was necessary to change the DSP system from the Motorola DSP56001 to the TI **TMS320C25** (plus related components). The TI 'C25 is a "previous generation" device and does not have the speed or convenient instruction set of the 56001. Fitting CLOVER-II into the 'C25 was a real "squeeze". As might be expected, this required a few trade-offs in performance and reduction of some of the deluxe features of the "high horsepower" **PCI-4000**. In particular, the P38 does not have processing power to receive the highest two amplitude modulation modes - **8P2A** and **16P4A**. The P38 will, however, communicate via CLOVER-II with any other P38, PCI-4000, or **DSP-4100** equipped station. Like the PCI-4000, DSP and 68000 **software** for the P38 is loaded via the PC bus and up-dates are readily available from **TECHLINE**.

CLOVER-2000

Frequency Spectra

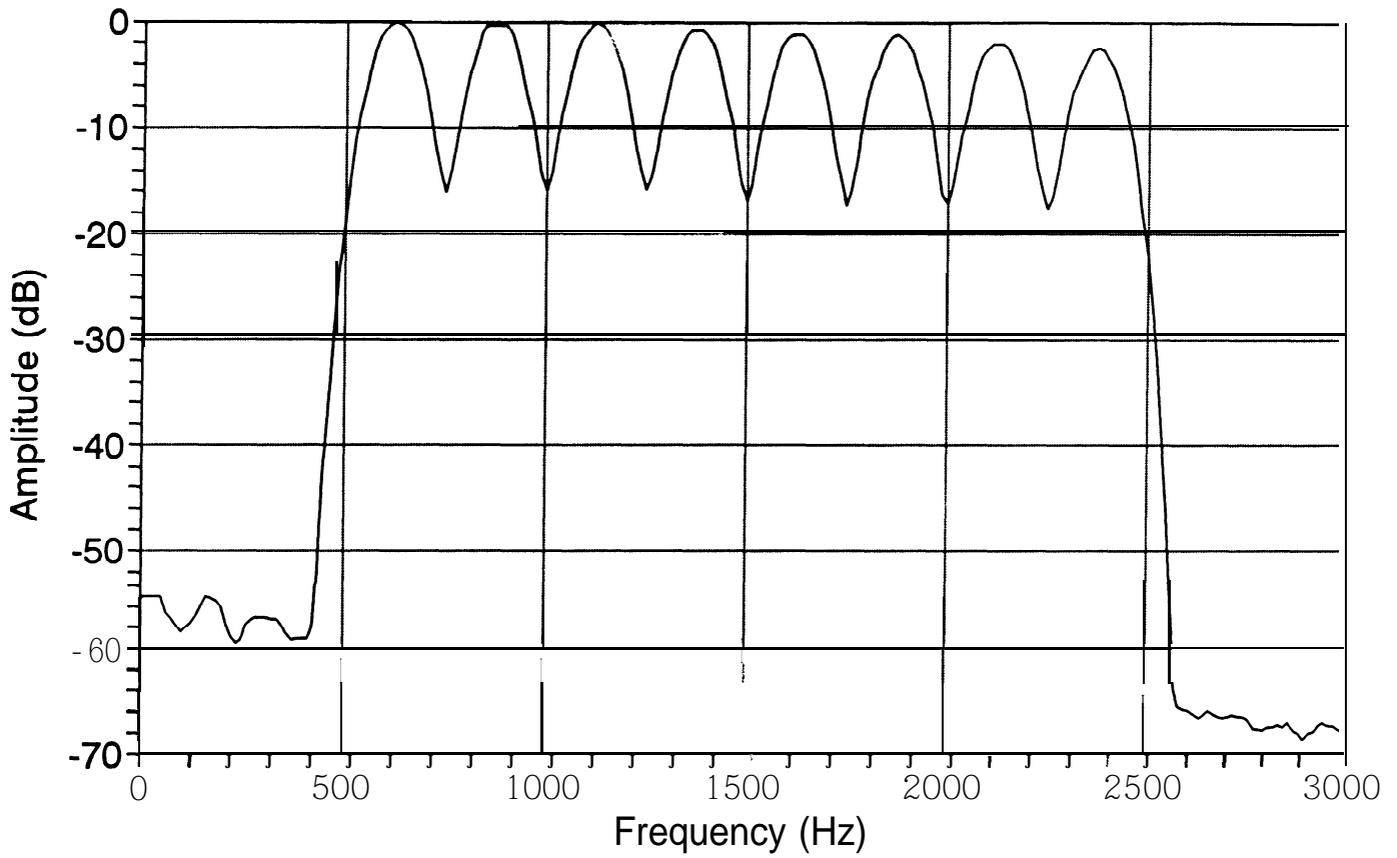


Figure 3.

CLOVER-2000 THROUGHPUT

Bias Comparison

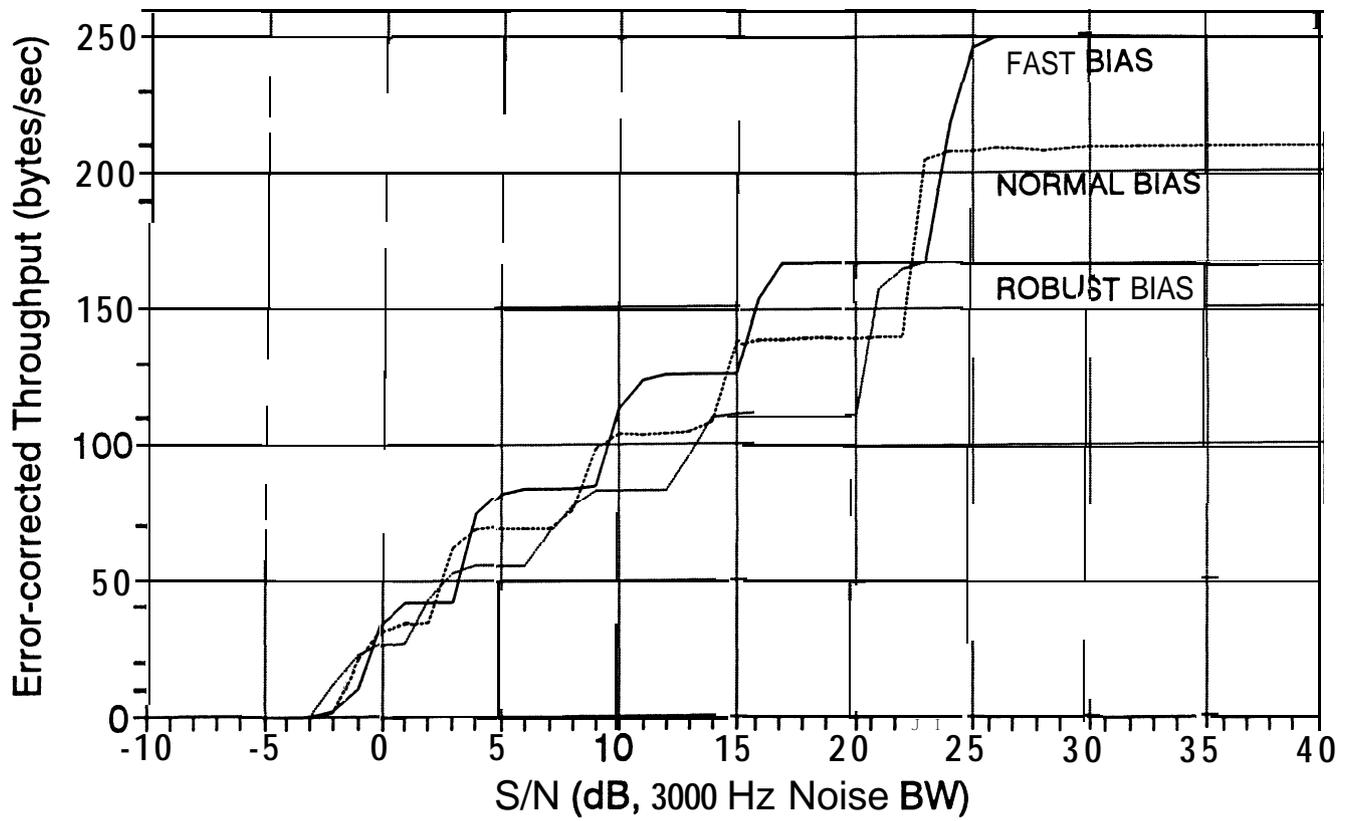
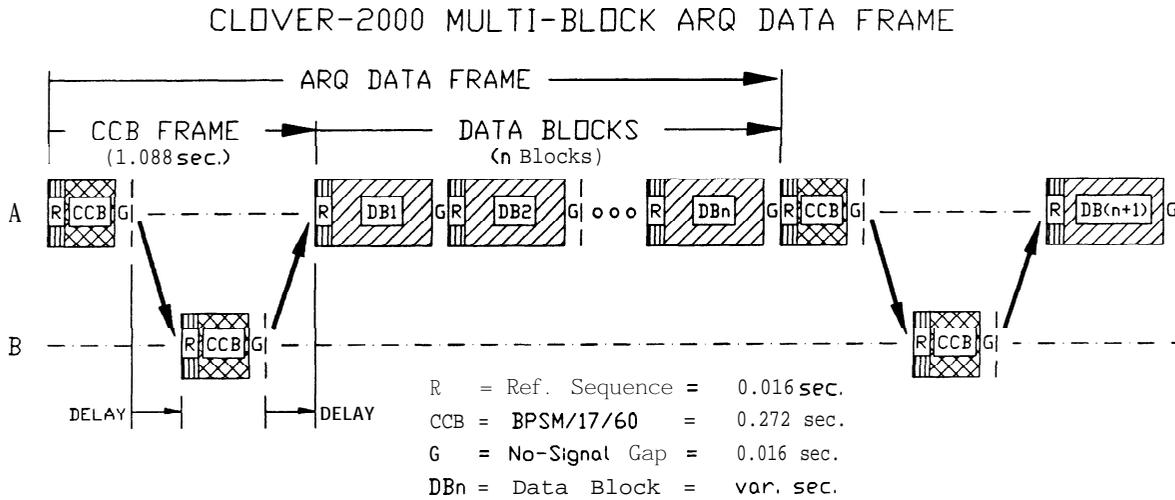


Figure 4.



ROBUST BIAS (60%)

RATE	MOD	BLOCK	BYTES/ FRAME	MAX ERRORS	BLOCK TIME	BLKS/ FRAME	ARQ FRAME TIME	THRU-PUT BYTES/SEC
165	16P4A	255	900	300	0.688 sec	6	5.440 sec	165.4
110	8P2A	255	600	200	1.024 sec	4	5.440 sec	110.3
83	8PSM	255	450	150	1.360 sec	3	5.440 sec	82.7
55	QPSM	255	300	100	2.048 sec	2	5.440 sec	55.5
28	BPSM	255	150	50	4.080 sec	1	5.440 sec	27.6

NORMAL BIAS (75%)

RATE	MOD	BLOCK	BYTES/ FRAME	MAX ERRORS	BLOCK TIME	BLKS/ FRAME	ARQ FRAME TIME	THRU-PUT BYTES/SEC
207	16P4A	255	1128	186	0.688 sec	6	5.440 sec	207.4
138	8P2A	255	752	124	1.024 sec	4	5.440 sec	138.2
104	8PSM	255	564	93	1.360 sec	3	5.440 sec	103.7
69	QPSM	255	376	62	2.048 sec	2	5.440 sec	69.1
35	BPSM	255	188	31	4.080 sec	1	5.440 sec	34.6

FAST BIAS (90%)

RATE	MOD	BLOCK	BYTES/ FRAME	MAX ERRORS	BLOCK TIME	BLKS/ FRAME	ARQ FRAME TIME	THRU-PUT BYTES/SEC
249	16P4A	255	1356	72	0.688 sec	6	5.440 sec	249.3
166	8P2A	255	904	48	1.024 sec	4	5.440 sec	166.2
125	8PSM	255	678	36	1.360 sec	3	5.440 sec	124.6
83	QPSM	255	452	24	2.048 sec	2	5.440 sec	83.1
42	BPSM	255	226	12	4.080 sec	1	5.440 sec	41.5

Figure 5.

The DSP-4 100 modem was introduced in late 1995. Unlike the PCI-4000 and P3 8, the DSP-4100 is not a plug-in card for a PC. Rather, the DSP-4100 operates from 12 VDC, handles data and commands via an RS-232 serial I/O port, and has status lamps on the front panel. In short, the DSP-4100 looks and acts much like a standard phone line modem, but it is used with **HF** radio systems. The DSP-4100 is proving to be the most popular configuration for commercial use of CLOVER particularly in applications where battery power and/or lap-top computers must be used. The **DSP-4100** uses non-volatile Flash ROM which also may be upgraded by loading new DSP or 68000 software via the serial port. Thus, software in all three DSP modems can be easily upgraded without opening a cabinet and replacing EPROM IC's.

In The Future:

CLOVER continues to grow and improve. Present software gives performance and features that were not available when CLOVER started. The "it's only software" development problem is finally working in our favor. Within limits, virtually every CLOVER modem sold can be upgraded in the field at little cost to the user. Custom versions of the hardware, terminal software, and CLOVER itself have been and continue to be developed to meet the needs of its users. CLOVER-2000 and the DSP-4100 have greatly extended the horizon for use of CLOVER technology. CLOVER modem hardware and software is already being installed inside some models of HF transceivers. In these systems, there are no outward signs that CLOVER is being used unless you **turn up** the receiver volume control and hear the distinctive **4- or 8-tone** "CLOVER twitter".

The future direction of CLOVER technology - waveform, protocol, software, and hardware - will be determined by the user. CLOVER provides the "missing link" that makes HF radio an attractive cost-effective alternative to satellite or wire-line data communications.

CLOVER BIBLIOGRAPHY

- Townsend, Jay: "The HAL **Communications** P38 HF Radio DSP Modem," (New Product Review), CQ, October, 1995, p. 30, CQ Publishing, Hicksville, NY.
- Vinson, Glen: "HAL P38 vs AEA PK-232", Digital Journal, **September**, 1995, pp **12 & 13**, **International Digital Radio Association ("IRDA"; formerly American Digital Radio Society - "ADRS")**, Box 2550, Goldenrod, FL 32733-2550.
- Ford, Steve: "Product Review -HAL **Communications** P38 HF Modem", **QST**, **August**, 1995, pp **71-73**, **American Radio Relay League (ARRL)**, **Newington, CT**.
- Blegen, Hal: "The HAL P38 DSP", **Digital Journal (RTTY Journal)**, August, 1995, pp 8 & 22, **International Digital Radio Association ("IRDA"; formerly American Digital Radio Society - "ADRS")**, Box 2550, Goldenrod, FL 32733-2550.
- Schulz, Fred: "P38, die neue CLOVER-Platine" **SWISS ARTG Bulletin (6/95)**, pp **14-17**, **SWISS Amateur Radio Teleprinter Group, c/o Arturo Dielet, HB9MIR (Secretary)**, **Blauenwig 8, 4335 Laufenburg, Switzerland (in German)**.
- Walder, **Stephan**: "Neu bei der SWISS-ARTG: **P38** von HAL", **SWISS ARTG Bulletin (4/95)**, pp 8 & 13, **SWISS Amateur Radio Teleprinter Group, c/o Arturo Dielet, HB9MIR (Secretary)**, **Blauenwig 8, 4335 Laufenburg, Switzerland (in German)**.
- Hollingworth, Jack, "The New HF Data Mode: CLOVER II", **RADIO COMMUNICATIONS**, November, 1993 (pp 52-54), December, 1993 (pp 62-64), January, 1994 (pp 68-70), **RSGB, Lambda House, Cranborne Road, Potters Bar, Herts. EN63JE England**
- Schilling, H.J., and Paul Williams: "Clover ein neus digitales **Funk-Verfahren fur Kurwelle**", CQ DL, June, 1993, pp 385-387, **Deutschen Amateur-Radio-Clubs @ARC, Postfach 11 55, 3507 Baunatal, Germany (in German)**.
- Healy, **James W.**: "HAL **Communications PCI-4000 CLOVER-II Data Controller**" (**Product Review**), **QST**, May, 1993, pp **71-73**, **American Radio Relay League (ARRL)**, **Newington, CT**.
- Townsend, Jay: "CLOVER - **PCI-4000**", **RTTY Journal**, **April**, 1993 (pp 3-4) May/June, 1993 (p 20), 1904 **Carolton Lane, Fallbrook, CA 9202846 14**.
- Walder, **Stephan**: "**CLOVER - Eine neue Betriebsart**", **SWISS ARTG Bulletin (1/93)**, pp S-1 2, **SWISS Amateur Radio Teleprinter Group, c/o Arturo Dietlet, HB9MIR (Secretary)**, **Blauenwig 8, 4335 Laufenburg, Switzerland (in German)**.
- Schulz, Fred: "**CLOVER** ist eingetroffkn", **SWISS ARTG Bulletin (1/93)**, pp **13-15**, **SWISS Amateur Radio Teleprinter Group, c/o Arturo Dietlet, HB9MIR (Secretary)**, **Blauenwig 8, 4335 Laufenburg, Switzerland (in German)**.
- HAL **Communications, PCI-4000 CLOVER-II Data Modem Reference Manual and PC-CLOVER Operator's Manual**, November, 1992, **HAL Communications Corp., Urbana, IL (\$25.00)**.
- Henry, George W., Ray C. Petit: "CLOVER - **Transmision de datos rapida** en HF", CQ (Spanish Language edition), November., 1992, pp **45-48**, CQ Publishing, Hicksville, NY (in Spanish).
- Petit, Ray C.: "The CLOVER-II Communication Protocol - Technical **Overview**", **ARRL 1 lth Computer Networking Conference Proceedings (October, 1992)**, **American Radio Relay League (ARRL)**, **Newington, CT**.
- Henry, George W., Ray C. Petit: "HF Radio Data **Communications: CW to CLOVER**", **Communications Quarterly**, Spring, 1992, pp **11-24**; CQ Publishing, Hicksville, NY.
- Henry, George W., Ray C. Petit: "CLOVER: Fast Data on HF Radio", CQ, May, 1992, pp. **40-44**; CQ Publishing, Hicksville, NY.
- Horzepa, Stan (ed); George W. Henry & Ray C. Petit: "CLOVER Development Continues", "Gateway", QEX, **March**, 1992, pp. **12-14**, **American Radio Relay League (ARRL)**, **Newington, CT**.
- Henry, George W., Ray C. Petit: "CLOVER Status Report", **RTTY Journal**, **January**, 1992, pp. 8-9, **Fountain Valley, CA**
- Henry, George W., Ray C. Petit: "Digital Communications for HF Radio - **AMTOR & CLOVER**"; paper **presented** at **Amateur Radio Digital Communication setninar**, **St. Louis, Mo., October 26, 1991**.
- Petit, Ray C.: "**CLOVER-II: A Technical Overview**", **ARRL 10th Computer Networking Conference Proceedings (1991)**, pp. **125-129**; **American Radio Relay League (ARRL)**, **Newington, CT**.
- Petit, Ray C.: "**CLOVER** is Here", **RTN Journal**, **Fountain Valley, CA; January**, 1991, pp. **16-18**; February, 1991, pp. **12-13**; **March**, 1991, pp. **16-17**; April, 1991, p 10.
- Petit, Ray C.: "The CLOVERLEAF **Performance-Oriented HF Data Communication System**", **QEX**, July, 1990, pp. **9-12**; reprinted in **ARRL/CRRL 9th Computer Networking Conference Proceedings (1990)**, pp. 191-194; **American Radio Relay League (ARRL)**, **Newington, CT**.

Constructing a Worldwide HF Data Network

Problems and Solutions

Introduction

The construction of a data communications system presents a wide range of challenges to the network designer, depending upon the needs of the users. Choosing High Frequency (HF) radio as the communications medium adds significant challenges of its own – challenges that are known to radio amateurs and shortwave listeners worldwide.

This paper describes the design choices made by a commercial company, Globe Wireless, while setting up a Global Radio Network to serve the maritime industry. The technical choices were based, at least in part, on methods and practices first developed in the amateur radio community. Perhaps the explanations that follow will assist amateurs when faced with similar choices.

History

The maritime transportation community has always had a need to communicate with ships at sea for commercial reasons. In addition, sailors have felt an understandable need for reliable communications in the event of an emergency at sea. Since early in this century, HF radio signals have been used for both purposes. More recently, satellite systems have also carried this traffic, to the point that many of the installed radio systems have not been modernized.

In the past, every seagoing nation of the world established its own Public Coast Station to serve the needs of ships in its vicinity, as well as that nation's fleet while it was on the high seas. This led to the development of, and competition between, large, high powered HF radio stations that attempted to cover the globe from a single location. About fifteen years ago, the first communications satellite to serve ships at sea was launched. Since that time the use of HF radio by ships has declined in favor of satellite systems. Today, nearly half of the world's fleet is equipped with an Inmarsat system, and that percentage is increasing.

Today, public coast stations offer two basic messaging services, radiotelegraph (Morse Code) and radio telex (using SITOR protocol). Some public coast stations also offer voice services but that is an entirely different subject and beyond the scope of this paper.

Morse Code is still the primary means of communication for a surprisingly large number of ships at sea. We estimate that 10,000 ships still communicate exclusively via Morse Code. The use of Morse Code by ships at sea is declining and will cease sometime **after** the full implementation of the Global Maritime Distress and Safety System (GMDSS) in February 1999. This will happen because GMDSS mandates the use of modern, automatic communications systems and eliminates the requirement for an experienced radio officer (RO) aboard ship.

Radio telex is the other means of message communication with ships using HF radio. It was originally developed to link shore based Telex machines with those aboard ship. The mechanical machines of the past have given way to personal computers, but the messages are still limited to the upper case only character set of the SITOR protocol as defined in CCIR Recommendations 476-4 and 625. This mode is nearly identical to **AmTOR** as used by hams.

Four years ago coastal radio station KFS in California was fighting a losing battle trying to compete with the Inmarsat satellite network using CW and SITOR. A single HF station, no matter how powerful, can not offer the global coverage of the four Inmarsat **geo-stationary** satellites. In order to compete with the Inmarsat system it was obvious that major changes had to be made. A new service was required that had to be equivalent, and less expensive than Inmarsat. HF radio, using the ionosphere, is certainly is less expensive than satellites with their high construction and launching costs. How, then to make the new service equivalent in other respects?

Requirements

The first step was to define the next generation HF radio maritime communications service. After considerable market research, these basic requirements were identified:

1. **Ease of Use** The system had to easily usable by the typical bridge officer aboard a ship at sea. No technical knowledge of radio or propagation could be assumed. Shipboard **software** must be as easy to use as a typical office **Email** interface – Lotus cc:Mail and Microsoft Mail, for example. This dictated a fully automatic system in respect to the actual sending and receiving of messages and the control of the radio equipment. The program aboard ship had to maintain close contact, via HF radio, with a central computer system ashore so that messages could be exchanged in either direction in a matter of minutes.
2. **High Capacity** The system had to accommodate the complete printable ASCII character set for messages, and allow any binary data – spreadsheets, word-processing documents or pictures, for example – to be included as an attached file. Because of

the increased traffic load that this would generate, a throughput of several times the six characters per second maximum of SITOR was essential. A new, interoperable, data communications protocol and modem were needed. The system chosen should allow high speed data transmission within the bandwidth of a standard ITU Narrow Band Direct Printing (NBDP) channel. The modem must also be able to utilize the traditional SITOR protocol on these same channels.

3. **Global Coverage & Twenty-Four Hour Availability** A ship had to be able to send and receive messages no matter where it was located and at any time of day or night. It was determined that a Global Radio Network of HF stations could meet this goal. The stations in this network should be located so as to provide twenty-four hour coverage to any point on the oceans of the world. Locations were chosen using a “Cellular” model so that the resultant overlapping coverage would provide a level of redundancy to the network. Every location has transmitters and receivers on several frequency bands to overcome changes in coverage due to daily and seasonal shifts in ionospheric propagation.

Design Challenges

This design goal – round-the-clock, automated, electronic mail connectivity with ships sailing on any ocean of the world – required using modern technical solutions to replace the traditional methods of the maritime radio industry. Significant technical problems were identified:

- **Radio Path Selection** The choice of which one of several available shore stations to call, and on what frequency to make the call, was a significant technical problem to solve. This choice would need to be made automatically by the shipboard software at any time of day or night, from any position on the world’s oceans and in spite of changing propagation and interference.
- **Modulation Protocol** A modern modulation method and modem that could coexist with the current SITOR system was needed. Frequency assignments in the Narrow Band Direct Printing (NBDP) portions of the maritime bands are on 500 Hertz channel centers, with no interference allowed to adjacent channels. So a severe limit on occupied bandwidth and a requirement for highly efficient use of the available spectrum were added to the list.
- **Network Interconnection** A reliable, yet affordable means was needed to back-haul messages and control information from the various coastal radio sites to the network control center, to be located in California.

Solutions Considered

Our procedure to finding solutions to the above problems was straightforward. Using brainstorming techniques, we identified any and all possible solutions that we could think

of. Then each technique was examined for practicality given our needs. A small number from the original list of ideas were selected for further detailed evaluation. Described below are the results of those evaluations.

Radio Path Selection

To provide automatic radio path selection four methods, taken from military, commercial and amateur practice, were evaluated.

Automatic Link Establishment (ALE), as defined in MIL STD-188-141A, was considered in detail. Unfortunately this system suffers from high equipment costs and wasted spectrum capacity when pinging the various channels available.

Theoretical Prediction software programs from the simple MINIMUF algorithm to the complex IONCAP system were evaluated. Even given that solar data could be provided to every ship at sea in a timely manner, the accuracy of these prediction systems was just not up to our needs. Knowing that a particular ship would have an 80% chance of a **useable** path to a particular coast station during a 2 hour period was not nearly good enough. We needed the system to know how to move a message NOW.

The **Chirpsounder** system, developed many years ago for the US military by BR Communications was investigated. Chirp transmitters were installed at several public coast stations and the information from associated receivers evaluated. This system is very good at measuring the ionosphere in great detail, but provided much more information than we needed. It would often tell us that the FOT for a particular path was between 9 and 10 Megahertz. That information is not too useful to a maritime service provider whose nearest allocations were at either 8 or 12 Megahertz. In addition, we were concerned about interference to the safety of life at sea (**SOLAS**) channels in the maritime bands potentially caused by the use of a large number of chirp transmitters.

So we went back to the basics. A system that eventually was called **Channel Sounding** had been in practical use since the early days of maritime communications. It relied on the tradition that all CW coast stations, when not occupied with traffic, broadcast a “CQ wheel” on every available channel. This allowed the shipboard RO to simply tune his receiver until he found a coast station on a frequency that he could hear clearly before calling and sending his message. This seemed like a clean and simple solution to the problem, but we had to make it automatic.

Choosing a modulation protocol

High speed, single tone modems using phase modulation were available from several manufacturers. They had originally been developed for military applications (MIL STD-188-1 1 OA for instance) so size and cost were not considered important design requirements. When evaluated, these systems performed fairly well, but we needed a modem that could be had for hundreds, not thousands, of dollars.

Amateurs had developed several new systems over the years, and **all** available were evaluated in our search. We considered using another case shift character to add lower letters case to **SITOR**, as both **G3PLX** and **AEA** had done with **AmTOR**, but that still did not solve the binary file problem. **AX.25** Packet protocol was considered because it can transfer eight bit data. This protocol, which performs very well on stable VHF circuits, falls apart very quickly on any but the very best HF path due to a high symbol rate. **PacTOR** showed initial promise but had two flaws. It can only transmit 7 bit data, and in the 200 baud mode it occupies considerably more than the 500 Hertz NBDP channel bandwidth. Both **PacTOR II**, an extension of **PacTOR**, and **GTOR**, which uses **Golay** encoding, were still in development and not ready for inclusion in our testing. In retrospect it appears that both of these modes also suffer the problem of excessive occupied bandwidth.

CLOVER-II, invented by Ray Petit, a ham, came very near to what we were looking for. It can transmit binary files – and at high speeds compared to **SITOR**. It performs well under poor HF conditions. A modem that included **SITOR** (actually **AmTOR**) on the same board, the HAL Communications PCI-4000, was available at a very reasonable cost. There was just one problem – the occupied bandwidth of **CLOVER-II** is exactly 500 Hertz, albeit with very clean skirts. Trying to stuff that signal into a 500 Hertz channel with no room for a little radio drift or calibration error seemed too much to risk. “Close, *but no cigar*,” we thought.

An affordable back-haul method

We actually considered using HF radio to connect our network sites with the central computer system, but rejected the idea fairly quickly. We needed to use the available spectrum efficiently and using radio for the back-haul wasted it on a two-to-one ratio. Also, the number of transmitters, receivers and antennas would increase by as much as two-to-one.

An obvious solution was to use dedicated **landline** circuits, rented from the local telephone company. In the US this actually method works fairly well due to the competitive telecommunications environment. However it seemed that every time another international border was crossed the price went up, and up, and up.

We investigated setting up our own satellite (VSAT) network. The costs of this approach exceeded those of dedicated **landline** circuits. Besides, we were trying to compete with satellite on the ship-to-shore side.

Current amateur Packet radio BBS forwarding practice included using **both dial-up landline** and Internet Mail systems as an alternative to on-the-air exchange of messages. Could one of these ideas be used?

Solutions Adopted

For radio path selection, we built upon tradition. We had already planned to use ITU NBDP channels where, the 'CQ wheels' of Morse had given birth to the transmission of a digital 'free signal' on channels not in use. We decided to use this signal as a beacon. A scanning technique, then in use by **APLink** and **WinLink** sysops on the ham bands, was adopted. We programmed the shipboard software to scan the ship's receiver through all the available network channels, evaluating any 'free signal' it found. This information was saved in a data base. The result was that the optimum path to the shore was always immediately available. This system came to be called **Automatic Channel Sounding**.

We were wrong about not finding the ideal modulation protocol for our needs. After just a few days of sitting in the lush corn fields of Southern Illinois and thinking about it, the engineers at **HAL Communications** called us back. They had found a way to scale the original **CLOVER** protocol down just a bit and had invented a mode they called **CLOVER-400** just for us. The throughput remained almost the same and the reduced occupied bandwidth of 400 Hertz left a comfortable guard band of 50 Hertz on each side. This meant that it could meet ITU requirements for use on maritime Narrow Band Direct Printing (NBDP) channels.

Within the US we found that we could indeed afford dedicated landlines to connect our network for station control and message exchange. For overseas locations our own software team came up with a proprietary packet protocol scheme that is used over the Internet backbone. We have implemented this now on several stations where a dedicated line to the local Internet provider is affordable and reliable. Back-up systems for these locations include either dial-up modems or **X.25** packet switch connections as available. Again, an idea from amateur radio was adopted for our needs.

Summary

Twelve HF stations, located in eight countries will comprise the Global Radio Network by the end of this year. A map that gives an idea of the coverage is included in Figure 2. Six coastal radio stations – **KEJ** in Hawaii, **KFS** in California, **SAB** in Sweden, **VCT** in Newfoundland, **WNU** in Louisiana and **ZLA** in New Zealand – are now on the air and operating as part of the network. Four network nodes are under construction – **A9M** in Bahrain, **8P0** in Barbados, **VIP** in Perth, **ZSC** in South Africa. Both **A9M** and **VIP** will be on the air very shortly with **8P0** and **ZSC** soon following. Two more stations, **KPH** in San Francisco & **WCC** in Massachusetts are awaiting FCC approval of the license transfer from MCI before being added to the network. Discussions for additional nodes are underway in Asia, Europe and South America.

All of the stations in the network are capable of both **SITOR** and **CLOVER-400** operation on the NBDP channels in use. Ships from several worldwide fleets are using our system, called **GlobeEmail** daily. They send and receive electronic mail messages and attached files using shore systems such as **MCI Mail**, **cc:Mail**, **Microsoft Mail** and the Internet.

The Global Radio Network is currently connected to the central computers in California using a combination of dedicated leased lines, shared leased lines, Internet packets and X.25 packets. Plans call for phasing out many of the dedicated lines in favor of Internet where available, and ATM or X.25 otherwise.

Future Enhancements

Engineers at Globe Wireless and HAL Communications have developed an extension of the CLOVER protocol. Called CLOVER-2000 because it occupies 2,000 hertz of bandwidth, it offers a five-fold increase in throughput over the CLOVER-400 mode now in use on NBDP channels. On-air tests have confirmed this level of performance.

Globe Wireless plans to implement CLOVER-2000 in very spectrum efficient manner. The details are still under development, but consider that CLOVER-2000 alone is not able to fully utilize the spectrum of a 2.8 kilohertz maritime facsimile or voice channel.

Beyond everything above, HAL's DSP engineers have identified changes to both types of CLOVER modulation that have the potential of doubling the throughput again, without increasing the occupied bandwidth. Those guys in Urbana sure don't stand still!

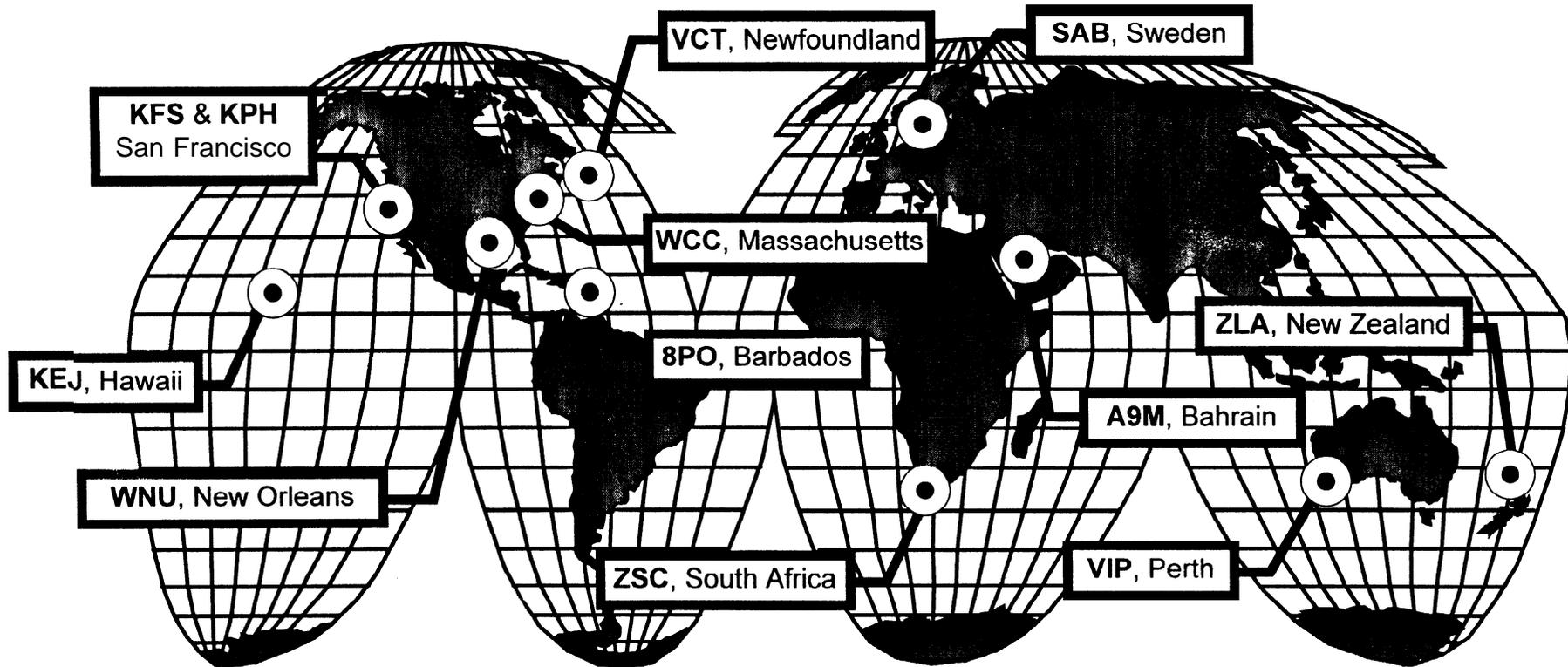


Figure 1 The Global Radio Network

Notes