1. Introduction

TCP/IP is the common name for a set of protocols developed to allow cooperating computers to share resources across a network. A protocol is simply a set of conventions or rules which must be adhered to by the communicating computers on a network to ensure that the information being exchanged is received and interpreted correctly. In the amateur radio software implementation of TCP/IP networking, the command set of the TNC is replaced with a very basic set of commands and the protocols are run in the user's computer. This opens up the real power of computer-to-computer networking via packet radio.

TCP/IP provides for file transfer, electronic mail, remote login services, and network file access. It was originally designed for the Department of Defense to connect unlike mainframe computers in the military, government, research institutions, private industry, and universities so that all could share resources on a common network. For more information on TCP/IP, consult "Introduction to the Internet Protocols," by Charles L. Hedrick (Rutgers University, 1987). This document is often available in electronic form on amateur radio packet systems. TCP/IP has been adapted for use on amateur radio networks and it complies with FCC rules on amateur radio digital transmissions. The amateur radio implementation of TCP/IP also provides services for keyboard chats, bulletin boards, AX.25 packet, and NET/ROM networking.

While there are many commands in a TCP/IP software package, the basic commands are not difficult to learn or to use. The user first sends a few software commands to the TNC to substitute its limited command set with an even simpler command set (called "KISS," for "keep it simple stupid") and transfers control to the TCP/IP software in the user's computer. The user can then engage in TCP/IP communications, or, since the software package has the capability of communicating in normal AX.25 packet, the user can operate with AX.25 packet bulletin boards, keyboard connects, digipeaters, or NET/ROM nodes.

TCP/IP has several advantages over normal AX.25 packet. At its lower levels, its strategies for retransmission of packets, exponential backoff in the face of channel congestion, handling of lost and duplicated packets, and packetization of data to be transmitted often lead to better overall channel throughput. It is designed to be a multi-connect, store-and-forward system. With TCP/IP, your local switch (similar to a node on AX.25 or NET/ROM) will hold your mail and on your request, forward the mail to your computer. Mail can also be forwarded to and from an AX.25 PBBS. There is never a "station busy" reply on TCP/IP, since the software provides for multiple sessions, with the switch passing out mail to connected stations much like a dealer dealing cards, while handling a file transfer at the same time. Another advantage of the TCP/IP protocols and software is that the routing of packets through several systems to the eventual destination is simpler for the user than that in AX.25 or NET/ROM forwarding. You do not need to know the full route to the destination. Rather, you set up a routing table to the stations that you can communicate with directly. If all other systems set up accurate routing tables, your packets will be forwarded properly to the desired destination.

The most accurate name for this set of protocols is the "Internet protocol suite," a layered family of protocols. TCP (transmission control protocol) and IP (internet protocol) are two of the lower level protocols. While the end-user of the suite does not often interact with the TCP or IP protocols, they are the best known of the protocols, and it has become common to use the term TCP/IP to refer to the whole family.

This document is a beginner's guide to use of the KA9Q Internet Software Package on the amateur radio packet network (AMPRNET). This software is the result of several years of development by Phil Karn, KA9Q, and an international group of collaborators. Phil developed the earlier NET TCP/IP software in the mid-1980's and it was "officially released" to the amateur radio community in April, 1989. He then began the development of NOS (Network Operating System), based on a multitasking networking kernel. The resulting software is extremely versatile. It was written for the IBM PC and clones, but has been ported to the Apple Macintosh, Atari ST, Commodore Amiga, and to several versions of UNIX machines. It has drivers for several hardware interfaces, allowing communication on wire networks as well as packet radio networks. Since the source code has been made available, there are numerous versions of this software. This beginner's guide documents the use of KA9Q NOS 911229 (written December 12, 1991) as modified in version 2.0h by Gerard van der Grinten, PA0GRI.

The NOS software provides the following services:

keyboard chat — A modified telnet protocol, called "ttylink," as implemented in NOS, allows users to "chat" from keyboard to keyboard.

mail — The simple mail transfer protocol (SMTP) provides services for sending and receiving mail. The sending, receiving, and forwarding computers can run unattended during this transfer, so it is not necessary to log into a PBBS to pick up your mail. A separate program, Bdale's Mailer (BM), first written by Bdale Garbee, N3EUA, is available to compose and read mail messages. It is also documented in this guide.

file transfer — The file transfer protocol (FTP) allows a user on any computer to get files from another computer, or to send files to another computer. Security is handled by requiring the user to specify a user name and password for the other computer. Both text (AASCI) and binary (executables or encoded text) files can be transferred.

mailbox or BBS — A mailbox, with a user interface similar to that of an AX.25 packet BBS, allows both TCP/IP and AX.25 packet users to read and compose mail. It also allows AX.25 users access to many of the TCP/IP services, providing them with an introduction to NOS and possibly an inducement to switch to the use of NOS.

finger — Finger is typically used to find out specific information about users on local or remote hosts. By fingering a user, you can find out such information as a user's name, mailing address, telephone number, QSL information, and other useful facts.

ax25 — Regular AX.25 services are also provided, so that NOS can be used for all of your packet radio activities. You can connect to a friend who is not running TCP/IP and conduct a keyboard chat or you can login to an AX.25 packet BBS.

NET/ROM — NOS also allows your packet system to serve as a NET/ROM node, although this will not be documented in this guide.

There are many other TCP/IP services that may be made available for use on packet radio in the future. Callsign servers are available on some systems. Access to network news via the network news transport protocol (NNTP) is also becoming available. Methods are being developed to set up routing tables and keep them up-to-date automatically, without requiring the user to edit the table. When multitasking computers become commonplace hobbyist machines, you will be able to remotely login to these systems to execute programs on them. Network file systems may become available, so that you can store your files on a remote disk and access them through the packet radio network. Most of these potential services will require higher baud rate networks, as a 1200 baud radio link is just too slow to support them.

It should be noted that the primary focus of many of the users of the TCP/IP software on amateur radio is experimentation with new prototocols and new versions of the software. While there are many full-service TCP/IP systems operating 24 hours a day providing services to the general user community, many others are on the air only sporadically to test new software. Thus, unless you enjoy dealing with constant technological change, you may be disappointed with TCP/IP networking in the amateur community.

1.1. Objectives of This Guide

The objectives of this guide are intentionally limited. The guide is intended to encourage more hams to use the NOS software on the amateur packet radio network (AMPRNET) and thus, it is aimed at beginners. It provides information on installing and using KA9Q NOS version 911229 (PA0GRI v.0f) and version 3.3.2 of BM on an IBM PC or clone, with a serial interface to a TNC running the KISS (Keep

It Simple, Stupid) firmware. Installation and use of other versions of this software running on PCs or other machines is similar to that described in this guide, but no attempt has been made in this guide to deal with the many other versions of the software.

This guide only documents the subset of the NOS commands that are of most use to a beginning end-user of the software and the packet network. Short descriptions of all of the top-level NOS commands are given in Appendix D. Further details are available in the “Network Operating System User Reference Manual,” by Phil Karn, KA9Q, and Gerard van der Grinten, PA0GRI, reflecting version 911229 of NOS 2.0h as released to the public by PA0GRI. This manual is often available on-line on the packet radio network. It should be consulted for details which are beyond the scope of this guide.

1.2. Acknowledgements

Version 1.0 of this guide, dated May 9, 1990, was based on KA9Q NET 890421 and BM 3.3.1 and much of the material in this version was taken from “The KA9Q Internet Software Package,” with permission from the author, Bdale Garbee. Shayne Hughes, N6SPE, Jim Pearce, N6ESV, and Chuck Bland, N6DBT, also provided many useful comments on the manuscript. Much of this material remains intact in this newer version of the guide.

This current version of the guide is based on the “Network Operating System User Reference Manual,” by Phil Karn, KA9Q, and Gerard van der Grinten, PA0GRI. To document the mailbox, I relied on the help files that were recently revised by Jim Mankin, KB3KJ, who couldn’t recall who had written the original files. Dell Tucker, WA6IUK, and Gary James, N6CNG, provided comments on the manuscript.

I would appreciate receiving any comments you have about this guide. I can be contacted at the following addresses:

AX.25 PBBS:	N6GF@WA6NWE.#NOCAL.CA.USA.NA
Internet:	ford@eecs.ucdavis.edu
U.S. Mail:	226 Diablo Ave., Davis, CA 95616

2. Necessary Resources

The purpose of this section of the guide is to describe the necessary resources you must have available to be able to set up an amateur packet radio system running TCP/IP. The hardware requirements of TCP/IP are nearly the same as any AX.25 packet station, although this guide assumes that the host computer is a PC or clone.

2.1. Computer

The computer required to run the version of NOS described in this guide is an IBM PC or clone running the MS-DOS or PC-DOS operating system (to be referred to simply as DOS for the remainder of this guide). The PC can range from the original PC (8086) to the XT, AT (286), 386, or 486 machines. This computer must have at least one serial port and a floppy disk drive. A hard disk is preferred, but is not absolutely necessary.

It is assumed that the reader of this guide is familiar with the basics of DOS. You should be able to set up directories, manage files, and have available and know how to use a text editor.

2.2. TNC

The TNC (terminal node controller) used for TCP/IP must run the KISS firmware. This includes the TAPR TNC-1, TNC-2 and clones produced by several manufacturers, equipped with a ROM running KISS. For the TNC-2 or clones, version 1.1.6 of the firmware, or later, is required. Most of the more recent TNCs also run the KISS firmware.

Attach your TNC to your PC serial port using an RS-232C cable, following your TNC manufacturer's instructions. Set the baud rate between the computer and the TNC as recommended in the TNC instructions. Verify that the TNC works properly in the AX.25 mode, again following the TNC instructions, and then enter the commands to run KISS. Further information on running the KISS mode is given in Appendix A.

2.3. Radio

The majority of the TCP/IP packet operations are on 2 meters, so you will need a 2m FM transceiver. The radio requirements for TCP/IP are the same as those for AX.25 packet. Follow the directions in your TNC manual to interface the transceiver to the TNC.

In most areas in the U.S., TCP/IP packet operations are found in the frequency ranges 144.91-145.09 MHz and 145.71-145.79 MHz. You will have to ask around to find out what frequency is being used in your area. One way to find the TCP/IP operation is to operate your TNC in the AX.25 mode and monitor the frequencies with “MONITOR ON.” The TCP/IP frequency is the one that causes the most “garbage” to be displayed on your screen, although NET/ROM nodes also cause this problem as well. The reason for this is that AX.25 TNCs do not decode the TCP/IP or NET/ROM control information.

2.4. IP Address

IP addresses are 32 bit numbers that uniquely identify a given machine (or “host”) running the TCP/IP protocol suite. All of the possible 32 bit numbers are coordinated by an entity known as the Network Information Center, or NIC. Amateur Radio operators are fortunate in that a “Class A Subnet” consisting of 24 bits of address, in the range 44.X.X.X, has been reserved for our use. Brian Kantor, WB6CYT, of San Diego, CA, serves as the top level administrator of the 44.X.X.X address space, and assigns blocks of addresses to regional coordinators from around the world. The list of these regional coordinators, as of January 1992, is given in Appendix C. This list is updated several times a year and is generally available on AX.25 PBBS systems.

You need to acquire an IP address before you can link in with the local AMPRNET community. To do this, contact the coordinator for your region, listed in Appendix C, or in a more recent list posted on your local PBBS. Your coordinator will assign you an address from the block for your area. If you need help in contacting this coordinator, you should ask around the local packet community for assistance. If all else fails, Brian Kantor can be reached as brian@ucsd.edu on the Internet if you have access to this wire network. He can help to locate your IP address coordinator.

If you have not yet obtained your IP address and want to get on the air immediately, you may temporarily use [44.128.0.*], with “*” replaced by a number between 1 and 255. Try to ensure that no one else in your area is using the same number.

2.5. KA9Q NOS Software

The KA9Q TCP/IP program NOS and the mailer BM are likely to be available in your local area. You should inquire about availability on your local packet BBS. This would not only provide you with the software, but also contact with someone who has used the software and could help you with its installation and use. Further, since the source code to these programs is available, many local versions are available and it is often to your advantage to use these local versions.

The programs NOS and BM must be installed and configured on your computer. The easiest way to do this is to edit the sample configuration files that are often included in distributions of the software. Local distributions may also include configuration information appropriate for your local network, so it is to your advantage to acquire the software locally. If your distribution does not include the sample files, detailed information on installation and configuration is given in Appendix B of this guide. The information in this appendix can also be used to understand the commands given in the sample files.

3. Definitions, Conventions, and Notation

In this section, some terms used in TCP/IP networking are defined and the conventions and notation used in the guide are explained.

Each system on the amateur packet radio network is referred to as a “node” or “host,” terms that are derived from wire networks. Since each TCP/IP node includes a computer, the term “machine” is also used interchangeably with “node” and “host.” As a user, you employ a “local host” and you communicate with a “remote host.” The local host often requests services from the remote host, and as a result, the remote host is known as a “server” and the local host is a “client.” Actually, each host provides a server and a client for each of the supported protocols.

Some hosts are set up not for use by an end user, but rather to forward packets from many users, in some ways similar to the function of an AX.25 digipeater, and to serve as a file repository. These systems are generally operated 24 hours a day and are known as “switches” or “gateways.” Some of these systems are set up to serve as mail gateways to and from AX.25 PBBS systems. Note, however, that end user hosts can also be used to forward TCP/IP packets.

Each host on the AMPRNET is identified by an IP address, as discussed in Section 2.4. Hosts are also known by a symbolic name which is linked to the IP address in the configuration file, DOMAIN.TXT. The convention on AMPRNET is to choose the user’s callsign as the symbolic host name.

The conventions used in this guide are described below. The intent was to produce this guide in the form of a simple ASCII (plain text) file that could be distributed through the AMPRNET. Thus, it was not possible to use bold or italic fonts, changes in point size, or underlining to clarify meaning.

3.1. NOS Commands

NOS has a command line-driven user interface. The commands have the following forms:

```
command
command arguments
command subcommand arguments
```

Commands and subcommands are character strings, usually consisting only of letters, but occasionally also including numbers. The arguments may be optional or required and the conventions for these arguments are given in the next section.

The NOS commands are case-sensitive and must be entered in lower case. However, file names are not case-sensitive and can be entered in either upper case or lower case.

All of the NOS commands and many of the subcommands may be abbreviated. You only need to type enough of a command to distinguish it from other commands that begin with the same letters. Arguments, however, must be typed in full.

If a required subcommand or argument is omitted, an error message will summarize the available subcommands or required arguments. Typing “?” in place of a required subcommand or argument will also generate this error message, which is useful when the command name alone is a valid command. When a command takes an optional value parameter, issuing the command without an argument generally displays the current value of the parameter.

3.2. NOS Command Arguments

NOS command arguments are also referred to as parameters, and are of the following types:

parameter — Literal argument. A character string such as “clear” with no surrounding brackets, is a required portion of a command and is to be typed exactly as shown. Many commands have subcommands which are required parameters.

<parameter> — Variable argument. A command argument enclosed by arrow brackets, such as “<directory>,” is a variable. An appropriate value for the variable must be used in the command. Appropriate values to be substituted are described as needed in this guide.

[parameter] — Optional argument. A command argument enclosed by square brackets, such as “[on],” is an optional argument. The effect of including, or not including, this argument in a command is described in this guide.

[<parameter>] — Optional variable argument. A command argument enclosed by first by arrow brackets and then by square brackets, such as “[<msglist>],” is an optional variable argument.

| — “OR” operator. Command arguments separated by “**|**” mean that either one or the other argument is to be used. For example, “<host>**|**clear” means that you are to either enter the variable <host> or the literal argument “clear.”

3.3. Notation

The following is the notation to be used in describing the commands in this guide.

<ip_addr> — The numeric IP address of a host in dotted decimal notation, optionally enclosed in brackets, e.g. 44.2.0.100 or [44.2.0.100].

<host_name> — The symbolic name of a host, e.g. “n6gf” or “eyolo”.

<host> — Denotes a host, switch, or gateway, which may be specified either as a symbolic name (<host_name>), or as a numeric IP address (<ip_addr>). The mappings between IP addresses and symbolic names are defined in the file “DOMAIN.TXT” described in section B.5 of Appendix B.

<callsign> — An amateur callsign, either in upper or lower case.

<directory> — The name of an existing directory on the host computer. Directory references can either be relative to the current directory, or absolute, beginning at the root (\ or /). Note that either back slashes (\) or forward slashes (/) can be used in directory names in NOS, while DOS requires back slashes (\). To refer to the parent directory, “..” can be used and for the current (working) directory “.” can be used.

<file> — The name of a file, e.g. DOMAIN.TXT. DOS is not case-sensitive, so file names can be given in either upper case or lower case with no differences implied. NOS displays file names in lower case. This guide generally gives file names in upper case when the files must be accessed by DOS commands and in lower case when they are accessed by NOS commands.

. . . — “Continued” notation. A command argument ending with “. . .” means that that multiple arguments of this type may be given. This is often used when multiple file names can be specified.

<value> — An integer number.

<string>	Any string of ASCII characters, surrounded by double quotes (") if the string contains spaces, such as "N6GF, Davis, CA".
<cr>	A carriage return, usually marked “Return” or “Enter” on most keyboards. Note that all commands given in this guide must be followed by a carriage return, although the <cr> notation will not be used in this case. <cr> is only used when the “command” needed is a carriage return on an otherwise empty line.
<F10>	The command key labeled “F10” or “f10” at the top or left side of the keyboard.
<space>	The space bar.
CTRL-X	The CTRL-X character is entered by holding down the “Ctrl” key while pressing the “x” key. Other common combinations of this type are CTRL-Z and CTRL-A.

4. NOS

The program that implements the Internet protocols is NOS.EXE. In this section, information on executing NOS, its command and converse modes, managing multiple sessions, NOS utility commands, escaping back to DOS and exiting NOS are provided. Information on the major NOS commands is given in later sections.

4.1. Executing NOS

If you have installed NOS in your root directory, it is invoked by simply typing the following at a DOS prompt:

nos

If you have installed NOS in a directory other than the root and this directory is specified in the “path” command in your AUTOEXEC.BAT file, NOS is invoked by typing the following at a DOS prompt:

nos -d <directory>

where <directory> is the directory where NOS has been installed. For information on installing NOS, see Appendix B.

When NOS is executed, it first opens the configuration file “AUTOEXEC.NOS” in the NOS directory. This file is described in section B.4 of Appendix B. NOS then displays the following:

```
KA9Q NOS version 911229 (PA0GRI v2.0h)
This version produced by PA0GRI
Copyright 1991 by Phil Karn (KA9Q) and contributors.
net>
```

The last line above is the NOS prompt “net>”. If this message is not displayed, something is wrong. Check your installation to see if you missed something. If you still have troubles, find a NOS user and ask for help.

NOS can also be invoked by typing:

net [-d <directory>] <file>

NOS will first attempt to open <file> as an alternate configuration file, which is read instead of AUTOEXEC.NOS.

4.2. Command and Converse Modes

NOS may be in one of two modes: command mode or converse mode. In command mode, the prompt "net>" is displayed and any of the NOS commands described in this guide may be entered. In converse mode, keyboard input is processed according to the "current session." Sessions include keyboard chats, file transfers (ftp), directory listings, AX.25 connections and several others, which will be described below. Not all NOS commands initiate a new session, but for those which do, this guide indicates that a new session will be created. In several of the sessions, keyboard input is sent to the remote host and any output from the remote host is displayed on the console. In an ftp session, only ftp converse mode commands may be entered. In these sessions, the user remains in converse mode until either the session is terminated (described in the command sections of this guide) or by escaping back to command mode, as described below.

4.2.1. <F10>

The user may escape back to command mode from converse mode by pressing <F10>. The command mode prompt "net>" will be displayed and any of the NOS commands may be entered. The session that the user "escaped" from will remain active.

4.2.2. <cr>

By entering <cr> at a "net>" prompt, the user will return to converse mode in the "current session." If there is no current session, NOS remains in command mode and reissues the "net>" prompt. Multiple sessions can be handled by NOS. For more information on multiple sessions, see the next section.

4.3. Managing Multiple Sessions

The NOS program can handle multiple sessions. For example, you can have an ftp file transfer running at the same time as a keyboard chat session. However, you should limit your use of multiple sessions that are transmitted on a 1200 baud radio channel, as you will cause considerable congestion. Note that not all sessions, such as local host directory listing sessions, will result in packet transmissions.

To start a second session, escape from the first session to the "net>" prompt by pressing <F10>. Start the second session as you would normally (as described in the command sections of this guide).

To monitor the multiple sessions, use the "session" command from the "net>" prompt. The syntax for this command is:

session [<session #>]

Without arguments, "session" displays a list of current sessions, including the active session number(#), the sequential session number since NOS was first executed (S#), the number of characters in the receive and send queues, the state of the session, and the remote socket (host and protocol). An example of a session list:

```
 #  S#  Type      Rcv-Q Snd-Q State        Remote socket
 1  138 AX25          0     0 Connected    n6spe-1 (N6SPE-1 on ax0)
*2  141 FTP          69     0 Established  eyolo (44.2.0.96:ftp)
```

An asterisk (*) is shown next to the "current" session; entering <cr> at a "net>" prompt will put you in converse mode with this session. Entering an active session number as an argument to the session command will put you in converse mode with that session.

Each session (including the "command session") has its own screen. When a new session is created, the command display is saved in memory and the screen is cleared. When the command escape key <F10> is hit, the current session screen is saved and the command screen is restored. When a session is resumed, its screen is restored exactly as it appeared when it was last current.

When a session is closed (through commands which will be described below), the following will be displayed on its screen:

```
Hit enter to continue
```

When you hit enter, you are returned to the command session (“net>” prompt).

The output of many sessions, such as directory listings, finger listings, and others, is managed by the “more” function, described in Section 4.4.2 below. The more function displays one screen at a time, pausing for a command from the user before continuing the output. The more function has been activated when “--More--” is displayed at the bottom of the screen.

4.4. Utility Commands

NOS provides several utility commands to provide help information, display files, and to manage directories and files.

4.4.1. help

Displays a list of the main NOS commands. The command “?” is equivalent to “help.” Note that several commands are listed that are not described in this guide. Concise descriptions of the commands listed by help are given in Appendix D.

4.4.2. more <file> [<file> ...]

Displays the specified file(s) a screen at a time. The prompt “--More--” appears at the bottom of the screen. To display the next screen, press the space bar. To display the next line, hit <cr>.

This command creates a session that can be suspended or resumed. To close the session, hit the “q” key at a “--More--” prompt.

Note that screen output from other sessions is automatically processed with the “more” command. This is indicated by the “--More--” prompt.

4.4.3. tail <file>

Displays the “tail” (the last 18 lines) of the specified file.

4.4.4. pwd

Displays the name of the current directory on the local machine.

4.4.5. cd <directory>

Changes the current directory to <directory>, which must be an existing directory on the local machine. The directory specified can be relative to the current directory, or absolute, with the name beginning at the DOS root (\ or /). The name of the new directory is displayed.

4.4.6. dir [<directory>]

List the contents of the specified directory on the console. If no argument is given, the current directory is listed. This command first lists the directory into a temporary file, then creates a “more” session to display it. After this completes, the temporary file is deleted. For example, a listing of a NOS directory:

```
alias              3,615 21:28  3/02/92   autoexec.nos        805 10:09  3/21/92
bm.exe            33,500  9:59  7/08/90   bm.rc               184 20:39  1/04/92
domain.txt        12,732 18:03  2/29/92   finger/                 21:11 11/17/91
ftpusers              83 22:05 12/30/91   nansi.sys         3,044 19:30  3/20/91
spool/                  20:40 12/12/91   nos.exe         199,573 10:47  3/07/92
```

```
10 files. 30,277,632 bytes free. Disk size 63,809,536 bytes.
```

Note that this 2-column listing gives the file names (a trailing / indicates a directory name), file size in bytes, and the time and date the file was last modified.

4.4.7. rename <oldfile> <newfile>

Renames <oldfile> as <newfile>.

4.4.8. delete <file>

Delete the specified file.

4.4.9. mkdir <directory>

Create a sub-directory <directory> in the current working directory.

4.4.10. rmdir <directory>

Remove the sub-directory <directory> from the current working directory. The sub-directory must be empty before “rmdir” can be used.

4.5. Executing DOS Commands

While running NOS, you may need to execute some DOS commands or to run a program such as a text editor. This can be done by escaping to DOS and then returning to NOS when you are finished. To escape to DOS, enter at a “net>” prompt:

    !

You will be returned to your DOS prompt and you can execute any DOS command. Note that the command “shell” is the same as the “!” command. If the command “multitask” is set to “on”, NOS will continue to run while you are at the DOS prompt, but if it is set to “off”, all NOS sessions will be suspended. For more information on “multitask”, see Section B.4.7 of Appendix B. To return to NOS, enter at an DOS prompt:

    exit

You will be returned to the “net>” prompt.

4.6. Exiting NOS

Before you exit NOS, you should check to see if you have any sessions active and if there is anyone logged into your mailbox. To check both sessions that you initiated as well as sessions initiated remotely, you will need to use the “tcp status” command, described in section 11.1.1. Mailbox activity can be checked with the "mbox status" command, described in section 10.6.

When you are sure you want to exit the NOS program and return to DOS, enter at a “net>” prompt:

    exit

5. Keyboard Chat

The ttylink command allows you to initiate a keyboard chat connection with a user at a remote host and a ttylink server on the local host allows a keyboard chat session to be initiated by a remote user. A split screen interface is used to allow easy conversation. The “telnet” protocol is used to implement this capability, so it is often called a telnet session.

5.1. Chat Session Messages

There are two commands which control messages that are sent to hosts attempting to establish a chat session.

5.1.1. attended [on | off]

This tells the host initiating the chat session if your system is attended or not. The command “attended on” generates the message “The system is attended” and the command “attended off” leads to the message “The system is unattended.” Without an argument, the current attend state is displayed. The default is “attended on” so if you do not regularly attend your system, you should include “attended off” in your AUTOEXEC.NOS file, as described in Appendix B.

5.1.2. motd <string>

The specified string is the “message of the day,” which is sent following the attended message to the host initiating the chat session. If you are not attending your system, you might use the following example:

```
motd "If unattended please leave a message in the mailbox"
```

5.2. Initiating a Chat Session

The command to initiate a keyboard chat session with the specified host and enter chat converse mode is:

ttylink <host>

For example:

ttylink n6spe

A new session will be established and if the desired host is on the air and reachable on the network, a message similar to the following will be displayed:

```
Welcome to TTY-Link at n6spe.ampr.org - The system is attended
```

If the remote system is attended, as indicated in the above example, you can then type back and forth in much the same way as in an AX.25 connection. When you’re done, use the <F10> key to escape back to command mode, and then type “close” to close the connection, as described in section 5.5 below.

5.3. Accepting a Chat Session

If a remote host requests a chat session, a message similar to the following will be displayed on your console:

```
Incoming Telnet session 2 from 44.2.0.96:1026 on Sat Mar 14 23:16:48 1992
```

A session number is automatically assigned. You may then select this chat session and converse with the remote as though you had initiated the session. If you are in command mode when the request arrives, enter <cr> at a “net>” prompt and you will enter converse mode for the chat session. If you are in converse mode in a session, use the <F10> key to escape back to command mode, use the “session” command to list the active sessions, and then use the “session <session#>” command to enter converse mode on the desired telnet session.

5.4. File Upload and Download

The telnet session can be recorded to a file, or an ASCII file can be uploaded instead of entering the information at the keyboard.

5.4.1. record <file> | off

Opens <file> and appends to it all data received or sent on the current ttylink session. If you are in telnet converse mode and want to initiate recording, you will need to use the <F10> key to escape back to command mode to issue the record command. The message “Recording into <file>” will be displayed and another “net>” prompt will be issued. Enter <cr> on a blank line and you will return to the ttylink converse mode with recording activated. The command “record off” stops recording and closes the file.

5.4.2. upload <file>

Opens <file> (must be an ASCII, file, not a binary file) and sends it on the current ttylink session as though it were typed on the terminal. If you are in ttylink converse mode and want to initiate uploading, you will need to use the <F10> key to escape back to command mode to issue the upload command. The uploading is initiated, but the file contents are not displayed on the screen during the uploading and no message is displayed to indicate that the uploading is complete. Enter <cr> on a blank line at the “net>” prompt and you will return to the ttylink converse mode.

5.5. Closing a Chat Session

To close a ttylink chat session, the following command is used:

    close [<session #>]

If you are in ttylink converse mode, press <F10> to escape to the “net>” prompt to issue this command. If you are running only one session, entering close without arguments will close the session. If you have multiple sessions, entering close without arguments will initiate a close on the current session. If you are running multiple sessions, the “session” command will display a list of these sessions. Entering close with a session number argument will initiate a close on the specified session. Only one of the hosts involved in the ttylink session needs to initiate the close. “Disconnect” is functionally the same command as “close.”

6. Mail

One of the most useful features of TCP/IP is electronic mail. Mail can be delivered to your unattended machine and you can read it at your leisure. There is no need to log into a PBBS to pick up your messages. Your messages can also be temporarily stored on your local switch and be delivered to your machine when you run NOS.

NOS sends and receives mail using the simple mail transport protocol (SMTP). There are two basic ways to compose and read these mail messages: through the NOS mailbox or BBS, described in section 8 below or through an external mail program. There are several external mail programs which can be used, but only the simplest of these, the program BM, will be described in this guide. This section concentrates primarily on BM, but a few comments on SMTP are given in subsection 6.5. In addition, a few comments on the post office protocol (POP) for mail store and forward are given in subsection 6.6.

The BM.EXE mail user interface program was created by Bdale Garbee, N3EUA, and despite popular belief, “BM” really stands for “Bdale’s Mailer.” It was later extended by Gerard van der Grinten, PA0GRI, and Dave Trulli, NN2Z. BM version 3.3.2 890801 is documented here.

6.1. Executing BM

BM is a separate program from NOS and thus, it must be executed from a DOS shell or subshell. However, NOS provides a simple command which escapes to a DOS subshell and executes BM. When you quit BM, you also exit the subshell and return to NOS command mode.

If the command “multitask” is set to “on”, NOS will continue to run while the user is in BM. If “multitask” is set to “off”, all NOS sessions will be suspended while the user is in BM. For more information on “multitask”, see Section B.4.7 of Appendix B.

BM is invoked by typing the following from a “net>” prompt:

mail

BM will first read the mail configuration file, BM.RC, described in section B.6 of Appendix B, and then will read the user mailbox defined in this configuration file. Using the BM main menu commands described in section 6.2, you can then compose or read mail messages or manage your mailbox.

If you use the command “!” to escape to a DOS subshell, BM may also be invoked in another way:

bm **-**u <mailbox> **|** **-**f <file>

With the argument “**-**u <mailbox>,” you can specify which mailbox to read, overriding that defined in BM.RC. With the argument “**-**f <file>,” BM will read the messages from <file> instead of a mailbox. This is useful if you have saved previously received messages to a file, using the “s” command described in section 6.2.6.

When you invoke BM, a banner will be displayed, followed by two lines of copyright information and then the mail header information, as described in section 6.2.1 below. Finally, there is a line reminding you to “Type ? for help” and then a prompt, consisting of your user name in quotes followed by an arrow (>).

6.2. BM Main Menu Commands

All BM main menu commands are single letters followed by optional arguments.

A <msglist> is a space-delimited list of message numbers, for example:

1 3 4 5

The message numbers are given in the message headers.

6.2.1. h

Display message headers. The message headers contain the message number, the status indicating whether it has been read or deleted, the sender, size, date, and subject. For example:

```
Mailbox /nos/spool/mail/n6gf.txt - 3 messages, 1 new

 DY  1  n6cng@n6cng               02 Apr 16:31 7198 TCP/IP FTPUSERS
  Y  2  N6SPE@n6spe.ampr.org      02 Apr 17:40  576 Re: eyolo
> N  3  N6SPE@n6spe.ampr.org      02 Apr 17:42  942 Re: NOS log
```

In the first line above, “D” indicates that the message has been marked for deletion and “Y” indicates that it has been read. The message number is 1, the sender is n6cng@n6cng it was sent on April 2 at 16:31, it 7198 characters long and the subject is “TCP/IP FTPUSERS.” In the third line above, “>” indicates that this is the current message and “N” means that it has not yet been read.

6.2.2. ?

Display a help menu for BM commands.

6.2.3. [<msg#>]

Entering a message number from the header listing will cause the message text to be displayed. If a carriage return is entered on a blank line, the current message will be displayed.

6.2.4. d [<msglist>]

Mark messages for deletion. With no argument, the current message (indicated by “>” in the first column of the message header) is marked for deletion. Messages marked for deletion are removed when exiting BM via the “q” command, when changing to an alternate mailbox with the “n” command, or when updating with the “$” command.

6.2.5. u [<msglist>]

Undelete a message that is marked for deletion. The status of a message can be determined by looking at the status field of the message using the “h” command.

6.2.6. s [<msglist>] [<file>]

The “s” command is used to save messages in a file. If no file name is given the file specified by the mbox variable in BM.RC is used. If no message number is supplied then the current message is saved. If <file> does not exist, it will be created. If <file> does exist, the messages will be appended to this file. The messages are stored in the same format as a mailbox file with all mail headers left intact. This file can later be read into BM by invoking BM with the “-f <file>” argument.

6.2.7. w [<msglist>] <file>

The “w” command is used to save messages in a file. Only the message body is saved. All mail headers are removed. If no message number is supplied then the current message is saved.

6.2.8. p [<msglist>]

The “p” command is used to send messages to the printer. This command uses the DOS device PRN for output. If no message number is supplied then the current message is printed.

6.2.9. m [<recipient_list>]

The “m” command is used to compose a message to be mailed to the list of recipients, specified by mail addresses or aliases, which are described in section 6.4. All local recipient addresses (those which don’t contain an “@”) are checked for possible aliases in the ALIAS file, described in section B.7 of Appendix B. Each alias that is found, is expanded into its list of addresses. If no recipient list is supplied with the “m” command, you will be prompted for a recipient list.

When you are composing a message, several commands are available such as invoking an editor or reading in text from other messages or files, as described in section 6.3. To end a message enter a line containing a single period.

It is important to remember that the input line buffer has a 128 character limit. You should format your text by entering a carriage return at the end of each line. Typing excessively long lines may cause data loss due to truncation when passing the message through other hosts. Keeping lines less than 80 characters is always a good idea.

6.2.10. r [<msg#>]

Reply to a message. Reply reads the header information in order to construct a reply to the sender. The destination information is taken from the "From:" or the "Reply-To:" header, if included. If no message number is supplied the current message is used.

6.2.11. f [<msg#>]

The "f" command is used to forward a mail message to another recipient. If no message number is supplied the current message is used. The user is prompted for the recipients and a subject. The mail header is added to the message text while retaining the complete original message in the body. Also see the ˜m command.

6.2.12. b [<msg#>]

Bounce a message. Bounce is similar to forwarding but instead of your user information, the original sender information is maintained. If no message number is supplied the current message is used.

6.2.13. n [<mailbox>]

Display or change the mailbox. The "n" command with no arguments will display a list of mailboxes containing mail. If an argument is supplied, then the current mailbox is closed and a new mailbox is opened.

6.2.14. l

List outbound messages. The job number, the sender, and the destination for each message is displayed. A status of "L" will appear if the SMTP sender has the file locked, meaning the message has begun to be sent to its destination. If you find that a message has been in the locked state for a long period of time, the corresponding SMTP session may be "stuck." See section 11.1 for a description of how to monitor the status of SMTP sessions and to reset them if they get stuck.

6.2.15. k [<msglist>]

Remove an outbound message from the send queue. A message can be removed from the send queue by specifying the job number obtained by the l command. If the message is locked you will be warned that you may be removing a file that is currently being sent by SMTP. It is best not to remove locked messages. They are better handled with the "tcp reset" command, described in section 11.1.2.

6.2.16. $

Update the mailbox, deleting messages marked for deletion. If you have a large number of messages in your mailbox and are cleaning it up by marking unwanted messages for deletion, updating the mailbox will shorten the header list.

6.2.17. x

Exit to DOS without changing the data in the mailbox. Messages marked for deletion will NOT be deleted.

6.2.18. q

Quit to DOS updating the mailbox. Messages marked for deletion WILL be deleted.

6.3. Text Input Commands

The following commands are available while composing a message. The tilde character (˜) must be entered as the first character on a line.

˜r <file>	Read <file> into the message buffer.
˜m <msg#>	Read <msg#> into the message buffer.
˜p	Display the text in the message buffer.
˜e	Invoke the editor defined in BM.RC with a temporary file containing the text in the message buffer.
˜q	Abort the current message. No data is sent.
˜˜	Insert a single tilde character into the message.
˜?	Display the help menu of tilde escape commands.

6.4. Mail Addresses

Mail is addressed to a recipient, which is either a user at a host or an “alias,” which is an alternative name for one or more recipients. Aliases are defined in the ALIAS file, described in section B.7 of Appendix B. These recipients can be on the local host or a remote host.

Mail addressing varies from simple to mildly confusing. The simplest form is:

<recipient>[@<host>]

If @<host> is not included, BM will search to see if <recipient> is in the ALIAS file on the local host. If it is, it will be expanded to the recipient list given for the alias. If <recipient> is not given in the alias file, the message will be sent to <recipient> on the local host (this is probably not what you intended, unless you have more than one user on your host).

If the address <recipient>@<host> is used, the message will be sent to <host>, where <recipient> is first looked up in the ALIAS file on <host>. If <recipient> is found to be an alias, it will be expanded to the recipient list given for the alias, and the message will be forwarded to these recipients. If <recipient> is not found in the alias file, the message will be delivered to <recipient> on <host>.

Host names can be found in your DOMAIN.TXT file, described in section B.5 of Appendix B. Valid mail user names for a given host can be found using the finger command, as described in section 10.2, if the finger files have been set up on the the host. If an ALIAS file has been distributed for your area, mail addresses and aliases will be found there. Otherwise, you will have to contact the intended recipient and ask for his or her address.

If the remote host of the recipient is not on the air when you try to send the message, it will remain in your mail queue until some time when both hosts are on the air at the time you attempt to send the message. To avoid this delay, switches have been set up in many areas which run 24 hours a day and can be used for mail forwarding. If the switch your recipient communicates with is <switch> (a host name) and this switch forwards the mail to your recipient using POP (post office protocol), then the mail can be addressed as:

<recipient>@<switch>

If POP is not used, then the address to be used is:

<recipient>%<host>@<switch>

Your mail will be transferred to <switch> and then forwarded to <recipient>@<host>.

6.5. SMTP

NOS sends and receives mail using the simple mail transport protocol (SMTP). This is handled automatically by NOS, although, you may want to “kick” out your outgoing mail manually, as described below.

When mail is received, SMTP displays the message:

New mail arrived for <user> from <sender>

where <user> is the addressee of the received mail and <sender> is the mail address of the person who sent the mail.

If you have included the “smtp timer” command in your AUTOEXEC.NOS file (see section B.4 of Appendix B), SMTP will check your outbound mail queue at the time interval you have set to determine if there is any new outgoing mail that should be sent. If you haven’t included this command, or you want to send out your mail before the next timer interval, you can manually “kick” out your outgoing mail, as described below.

6.5.1. smtp kick

Reviews the outgoing mail queue and attempts to deliver any pending mail. This allows the user to “kick” the mail system manually. The command can be entered at a “net>” prompt after you have composed mail messages.

6.5.2. smtp list

List the messages in the outgoing mail queue. For example:

```
S  Job    Size Date  Time  Host    From
L  171     244 03/13 07:13 n6spe   n6gf@n6gf
  n6spe%n6spe@eyolo
   173     243 03/13 07:14 n6yuv   n6gf@n6gf
  n6yuv%n6yuv@eyolo
```

The “L” entry in the “S” status column indicates that the message entry is locked by SMTP; in other words, SMTP is attempting to send the message.

6.5.3. smtp kill <job_number>

Kills outgoing message number <job_number>, which is given in the “Job” column in the “smtp list” command, as shown above.

6.5.4. smtp quiet [on|off]

Display or set the SMTP quiet flag. This flag determines whether the bell sounds when the system announces that incoming mail has arrived. When quiet is set to “off, you hear the bell. With no arguments, “smtp quiet” displays the status of the flag.

6.6. POP

The Post Office Protocol (POP) stores mail on a host, then forwards it on demand to a user’s host. It is well-adapted to amateur radio, in which many users are only occasionally active on the network. With the use of POP, the forwarding host does not have to keep trying to establish an SMTP connection to a destination host that is not always on the air, but rather, it waits for the destination host to request that the mail be forwarded.

You will have to inquire if POP service is provided in your area. If so, you will need to include “popmail” commands in your AUTOEXEC.NOS file, as explained in Sections B.4.28 to B.4.30 in Appendix B. The only command you might want to use otherwise “kicks” the mail out of the forwarding host.

6.6.1. popmail kick <host>

Starts a pop forwarding session from the specified host. This command is needed when no interval is specified in the “popmail addserver” command, described in Section B.4.28 in Appendix B.

7. File Transfer

The ftp command provides for the transfer of files using the file transport protocol. It enables you to do the following:

Transfer text and binary files between local and remote host
List directories on a remote host
Delete files on a remote host
Create and remove directories on a remote host

The remote host can be unattended and the ftp server on that host will provide the requested services. File access privileges are defined in the FTPUSERS file, described in section B.8 of Appendix B. This file defines user login names, passwords, directories to be accessed and file access privileges. It is a common convention to allow arbitrary users limited access to files under the special login names “anonymous” or “guest.”

7.1. Ftp Command

The command ftp is used to initiate an ftp session. It is invoked as:

ftp <host>

where <host> is the desired remote host. If the session is established, you will enter converse mode on the new ftp session. Only the ftp commands described below are valid in an ftp session.

When the connection between the two machines is opened, you’ll get a banner from the remote machine, followed by a login prompt and a password prompt. If you’ve negotiated with the manager of the remote machine to have a special login name and password set up for you in the FTPUSERS file, use these names in your login. If not, use one of the special login names, “anonymous” or “guest,” and in these cases, use your call sign as your password. Your login and password are recorded in the log file on the remote host, allowing the manager of that host to keep track of ftp activity. After you have logged in, you will be presented with the ftp prompt: “ftp>”.

7.2. Ftp Converse Mode Commands

The following are the ftp commands that are valid in ftp converse mode, issued at the ftp prompt “ftp>”. Note that you may not have been granted permission to use all of these commands. Refer to Section B.8 of Appendix B for information on the user permissions that are generally granted to remote users.

7.2.1. dir [<remote_file>|<remote_directory> [<local_file>]]

Without arguments, “dir” requests that a full directory listing of the remote server’s current directory be sent to your display. If one argument is given, it is interpreted as a specific file or sub-directory on the remote file system that is to be listed. To refer to the current working directory, use “.” as the directory specification. If two arguments are given, the second is taken as the local file into which the directory listing should be written (instead of being sent to the display). The full listing gives the file names, sizes, and creation dates.

You should request a directory listing when you first log into an unfamiliar machine. There will often be a file named “readme” or “whathere.txt” that will give some information about the files available on the remote machine. This file can then be acquired with a “get” command (described below), and read on your machine to learn more about the files available on the remote host.

7.2.2. ls [<file>|<directory> [<localfile>]]

“ls” is identical to the “dir” command except that an abbreviated directory listing is provided. This listing gives only the file names.

7.2.3. pwd

Displays the name of the current directory on the remote host (“pwd” literally means “print working directory”).

7.2.4. cd <remote_directory>

Changes the current directory on the remote host to the directory indicated by <directory>, which must be an existing directory on the remote host. The directory specified can be relative to the current directory, or absolute, with the name beginning at the root (\ or /).

7.2.5. get <remote_file> [<local_file>]

Asks the remote host to send the file specified in the first argument and to write the file on the local machine. The second argument, if given, will be the name of the file on the local machine; otherwise it will have the same name as on the remote host. See the “type” command below if the file requested is other than an ASCII file. If the file is over 10,000 characters in size, you should only start the transfer when the radio channel is relatively quiet. Use the “abort” command described in section 7.3 below if you want to terminate the transfer before it has been completed.

7.2.6. mget <remote_filename> [<remote_filename> ...]

Send a collection of files from the remote host. File names may include wild card characters; they will be interpreted and expanded into a list of files by the remote host. For example:

```
mget *.txt
```

will send all files having a file name extension of “.txt”. The files will have the same name on the local host that they had on the remote host. See the “type” command below if the files requested are other than ASCII files. Use the “abort” command described in section 7.3 below if you want to terminate the transfer before it has been completed.

7.2.7. put <local_file> [<remote_file>]

Asks the local host to send the file specified in the first argument and to write the file on the remote machine. The second argument, if given, will be the name of the file on the remote host; otherwise it will have the same name as on the local machine. You must have write privilege on the remote host to use this command. Use the “abort” command below if you want to terminate the transfer before it has been completed. See the “type” command below if the file to be sent is other than an ASCII file.

7.2.8. mput <local_filename> [<local_filename> ...]

Send a collection of files to the server. File names may include wild card characters; they will be expanded locally into a list of files to be sent. For example:

```
mput *.txt
```

will transfer all files having the file name extension “.txt” on the local host. The files will have the same name on the server as on the local system. If the files you want to send are other than ASCII files, you must use the “type” command below before initiating the transfer. You must have write privilege on the remote host to use this command.

7.2.9. quit

Terminates the ftp session.

7.2.10. type [a | i]

Tells both the local and remote hosts the type of file that is to be transferred. Without arguments, the current mode is displayed. The default is “a”, which means ASCII (i.e., a text file). In “i” mode, which means “image”, files are sent exactly as they appear in the file system. This mode must be used when exchanging raw binary files (executables, compressed archives, etc). The file type must be set before a “get” or “put” command is initiated. The file type remains in effect until it is changed by a subsequent “type” command.

7.2.11. dele <remote_file>

Deletes a file on the remote machine. You must have delete privilege on the remote host to use this command.

7.2.12. mkdir <remote_directory>

Creates a directory with the name <remote_directory> on the remote machine. You must have write privilege on the remote host to use this command.

7.2.13. rmdir <remote_directory>

Deletes <remote_directory> on the remote machine. The remote directory must be empty before you can remove it. You must have delete privilege on the remote host to use this command.

7.2.14. verbose [0 | 1 | 2 | 3]

Set or display the level of message output in file transfers. Verbose 0 gives the least output, and verbose 3 the most, as follows:

- 0: Display error messages only.
- 1: Display error messages plus a one-line summary after each transfer giving the name of the file, its size, and the transfer time and rate.
- 2: Display error and summary messages plus the progress messages generated by the remote FTP server.
- 3: Display all messages. In addition, a “hash mark” (#) is displayed for every 1,000 bytes sent or received.

7.2.15. hash

A synonym for the “verbose 3” command.

7.2.16. user <username>

Normally you will be asked for your username when you log in to FTP. However, if you mistype your name and the remote system rejects it, you can then try again with the “user” command.

7.2.17. pass <password>

Normally you will be asked for your password when you log into FTP. However, if you mistype your password and the remote system rejects it, you can then try again with the “pass” command.

7.3. Aborting a File Transfer

To abort a get, mget, put, mput, dir or ls operation in progress, it is necessary to escape back to the command mode (“net>” prompt) and use the “abort” command (this command works only on FTP sessions).

7.3.1. abort [<session #>]

Aborts an ftp file transfer operation in progress on the indicated session. Issued without an argument, the current session is aborted. Abort is valid only when a transfer is in progress. When a file transfer operation is aborted, a partial copy of the transferred file will be left on the destination machine, which must be removed manually if it is unwanted. This is also true for a dir or ls operation when the directory listing is written as a local file. After the abort is completed, you can return to the ftp session and use other ftp converse commands.

7.4. Ftp Example

In the ftp example below, the user initiates an ftp session with the remote host “eyolo,” logs in as the user “anonymous,” requests a directory listing, changes to subdirectory, gets a binary file and terminates the session. The text given in parentheses to the right of the commands indicate what has been typed by the user.

```
net> ftp eyolo                                    (ftp eyolo)
Resolving eyolo... Trying 44.2.0.96:ftp...
FTP session 1 connected to eyolo
220 eyolo.ampr.org FTP version 911229 ready at Mon Mar 23 16:16:54 1992
Enter user name: anonymous                        (anonymous)
331 Enter PASS command
Password: n6gf                                    (n6gf)
230 Logged in as anonymous, restrictions apply
ftp> dir                                          (dir)
200 Port command okay
150 opening data connection for LIST /public
docs/              16:42 3/02/92 lists/            16:40 3/02/92
programs/          14:40 3/02/92 utility/          16:42 3/02/92
whathere.txt  1,755 18:22 3/21/92
5 files. 6,959,104 bytes free.  Disk size 10,584,064 bytes.
LIST: 265 bytes in 13 sec (20/sec)
226 File sent OK
ftp> cd programs                                  (cd programs)
257 "/public/programs" is current directory
ftp> dir                                          (dir)
200 Port command okay
150 Opening data connection for LIST /public/programs
bm.exe       41,225 17:55 2/25/92 nos.exe     174,454 17:43 2/25/92
2 files. 6,959,104 bytes free.  Disk size 10,584,064 bytes.
LIST: 135 bytes in 6 sec (23/sec)
226 File sent OK
ftp> type i                                       (type i)
ftp> get bm.exe                                   (get bm.exe)
200 Type OK
200 Port command okay
```

```
150 opening data connection for RETR bm.exe
RETR: 41225 bytes in 1521 sec (27/sec)
226 File sent OK
ftp> quit                                 (quit)
221 Goodbye!
FTP session 1 closed: EOF
Hit enter to continue
```

The user’s callsign was used as the password, which is shown in this example. However, the password is not echoed to the screen by the software. After logging in, the prompt is “ftp>”. The display generated by the “dir” command in this example shows that the user was logged into the /public directory. The listing shows that there is one file, named “whathere.txt,” of size 1755 bytes, created at 18:40 on 3/21/92. There are also four subdirectories, indicated by “/” at the end of their names: “docs/,” “lists/,” “programs/,” and “utility/,” all created on 3/2/92. The dir output is finished with the “226 File sent OK” message. The command “cd programs” is issued to change to the subdirectory programs. A dir command on this subdirectory shows that there are two files, “bm.exe” and “net.exe.” These are executable programs since the file name extensions are “exe” and therefore, they are binary files. The “type i” command is issued so that a binary file can be transferred. Note that due to an apparent bug in the program, the response to this command “200 Type OK” does not get displayed until after the next ftp command is issued. The “get” command is issued and there will be a delay as the “bm.exe” file is retrieved. This is also finished when the message “226 File sent OK” is received.

8. Mailbox or BBS

NOS provides a service known either as a mailbox or a bulletin board system (BBS). This provides services for reading and composing mail, reading and composing bulletins, listing file directories on a remote host, uploading and downloading files, AX.25 connections from a remote host, and other services to be explained below. Note that many of these services are provided more conveniently by other NOS commands. The best use of these services by NOS users is as an alternative to BM for reading and composing mail messages. While I think that BM offers better services for these functions, the mailbox has the advantage that it does not need the additional memory required by the DOS subshell for running BM.

The primary value of the NOS mailbox (BBS) is in providing TCP/IP services for AX.25 users. An AX.25 connection to a host running NOS enters through the mailbox, giving the AX.25 user access to the AMPRNET. This introduction to TCP/IP should encourage AX.25 users to switch to NOS and become part of AMPRNET. Since this service is not expected to be heavily used, the documentation provided here is not very detailed.

8.1. Mailbox Messages

There are two commands that control messages that are sent to users who log into your mailbox.

8.1.1. mbox motd <string>

Send <string> as the message of the day to a user who has just logged into your mailbox. Since many users may be unfamiliar with the use of this mailbox, I use the following message of the day:

> mbox motd "Type ’H<cr>’ for help, ’I<cr>’ for information"

This message can be defined in the AUTOEXEC.NOS file, as described in Appendix B.

8.1.2. mbox attend [on | off]

Display or set the mailbox “attended” flag. This is used to announce in the mailbox whether or not the station manager is present and willing to chat (via the mailbox “Operator” command). If the mailbox attended flag is off, the response to the chat command is the message: “Sorry - the system is unattended” and the mailbox prompt is again displayed. If the mailbox attended flag is on, a chat session is initiated.

8.2. Initiating a Mailbox Session

A mailbox session is initiated with the command

telnet <host>

where <host> is either the local host (for reading and composing your own mail) or a remote host. To initiate a session on the local host, an alternative command is:

bbs

The mailbox session begins with a banner message including “KA9Q NOS” and the name of the host. The user is then prompted for a login name and a password. You should always use your callsign for the login name. If you have not used the mailbox on this host before, you can usually use anything for the password, including simply <cr>. Until you have requested additional permissions from the host manager (sysop), your privileges on this host will be limited. When the sysop gives you added privileges, you will also be given a specific password. Login names, passwords, and user permissions are defined in the FTPUSERS file — see Section B.8 of Appendix B.

After entering your login name and password, some welcome information and then a mailbox prompt is displayed; for example, on my local switch:

```
[NOS-H$]
Welcome n6gf,
to the eyolo.ampr.org TCP/IP Mailbox (911229 (PA0GRI v2.0h))
Type 'H<cr>' for help, 'I<cr>' for information

You have 2 messages  -  1 new
Current msg# 1 :
 ?,A,B,C,D,E,F,H,I,J,K,L,M,N,O,P,R,S,T,U,V,W,X,Z >
```

The line after the “welcome” message is the “mbox motd” described above. You are told how many messages are waiting for you and how many of these are new. Then the mailbox prompt is displayed, indicating the current message number and reminding you of the first letters of all of the mailbox commands.

An AX.25 user initiates a mailbox session by simply connecting to the callsign of the TCP/IP host. The login proceeds as for a NOS user.

8.3. Mailbox Commands

As indicated by the prompt above, there is a long list of mailbox commands. These commands can be typed in full, or abbreviated by the first letter of the command. Note again that you may not have permission to use all of these commands. For details on the permission codes, see Section B.8 of Appendix B.

8.3.1. ?

Displays a summary of mailbox commands:

```
(?)help    (A)rea  (B)ye    (C)onnect (D)ownload (E)scape  (F)inger
(H)elp     (I)nfo  (J)heard (K)ill    (L)ist     (M)busers (N)odes
(O)perator (P)orts (R)ead   (S)end    (T)elnet   (U)pload  (V)erbose
(W)hat     (X)pert (Z)ap
```

8.3.2. Area [<area_name>]

The Area command, without an argument, lists the mail areas (i.e. mailboxes) that contain messages you may read. The list gives the name of each area, followed by a description of the area's contents. When the Area command is followed by a valid <area_name>, your current mail context will be switched to the new area. You may then use the Read and List commands to review messages in the selected area. To add your own message to one of these areas, first select the area with the Area command, and then use the Send command to send mail to that area. The mail areas are defined in the AREAS file, described in section B.10 of Appendix B.

8.3.3. Bye

The Bye command allows you to exit from the NOS mailbox. This closes your mailbox file, removes any messages that you have deleted with the Kill command and terminates the session.

8.3.4. Connect [<port>] [<callsign> | <node>] [<digipeater> . . .]

The connect command allows you to connect to another AX.25 station through the mailbox systems's radio port(s) or through a NET/ROM node known to the mailbox system, or to initiate a keyboard chat with the mailbox system operator.

You connect via AX.25 if you specify a <port> on the command line. You can determine what AX.25 ports are available by using the "Ports" command. The connection will use your callsign with the SSID modified. The station you want to connect to is specified in the "<callsign>" parameter. If you need to reach this station via one or more digipeaters, enter the list list of digipeaters following the destination station's callsign (note that the word "via" is not needed).

If you don't specify a <port> on the connect command, the system will attempt to make a NET/ROM connection to the node given by the "<node>" parameter. You can get a list of nodes known to this system by using the "Nodes" command.

8.3.5. Download [<directory>/] <filename>

The Download command sends an ASCII (plain text) file from the remote host to the current directory of the local host. The optional <directory> is to be included if the desired file is not in the current directory, as determined with the What command.

8.3.6. DU [<directory>/] <filename>

The DU (download uuencoded) command sends a binary file, converted to uuencoded ASCII from the remote host. You will need the UUDECODE.EXE program to convert this ASCII file back to binary after the download.

8.3.7. Escape [<escape_character>]

Without an argument, Escape displays the current escape character. This character is used to exit from a chat session. The escape character may be changed to one of your preference by entering "Escape" followed by <space> and the character that will become the new escape character. This must be a single typed character (the <CTRL> key may be used in addition).

8.3.8. Finger [<username>][@<host>]

The Finger command retrieves personal information about users of a system. When issued without parameters, a list of known users on the mailbox host is displayed. For more information on the finger command, see section 10.2.

8.3.9. Help [<command_name>]

The Help command will display help for a given command. Without a parameter, “Help” will display the following list of command names for which help information is available:

```
area      bye       connect   download  escape    finger
help      info      jheard    kill      list      musers
nodes     operator  ports     read      send      telnet
upload    verbose   xpert     what      zap
```

To provide help information on your own system, you will need to install the mailbox help files, described in Section B.11 of Appendix B.

8.3.10. Info

The Info command provides information about the mailbox host. This information must be made available by the mailbox manager, as described in Section B.11 of Appendix B.

8.3.11. Jheard [<interface>]

The Jheard command displays a list of all the station callsigns that have been heard by the mailbox host packet station on the specified interface. With no argument, the “heard” lists for all interfaces is displayed. The format of the display is the same as that for the “ax heard” command, described in section 10.1.

8.3.12. Kill <message_number> [<message_number> ...]

The Kill command deletes messages from the current mailbox area (if you have been given that permission by the operator). At least one message number must be supplied.

8.3.13. List [<starting_msg_number> [<ending_msg_number>]]

The List command displays a list of the messages in the current mailbox (or “area”). For each message, the list contains the subject header line, the time and date it was created, who it is from, how many bytes long it is, and whether or not it has been read. You may include an optional <starting_msg_number> from which to begin displaying the list. If you specify a starting message number, then you may also specify an ending number as well. This will limit the display for you in case there are a large number of messages in a particular “area” mailbox.

8.3.14. Musers

The Musers command displays a list of all the current users of the mailbox and how they have connected to it.

8.3.15. Nodes [<node>]

The Nodes command without a parameter prints a list of NET/ROM nodes that are known to the mailbox system. For a specified known node, information on the node is displayed. A connection to one of these nodes can be attempted with the “Connect <node>” command.

8.3.16. Operator

The Operator command initiates a keyboard chat with the operator of the mailbox system. When you select this mode, a new window will be opened on the operator’s console, and whatever you type will be visible there. If the operator is present, and types something in return, it will be sent back to you. If the operator is not present (the mailbox attended flag is off on the operator’s host), you will receive the message “Sorry — the system is unattended” and the mailbox prompt will again be displayed.

When you wish to terminate the chat session, type the escape character on your keyboard, and then press <cr>. The default escape character is CTRL-X. This escape character may be changed to whatever you prefer by using the “Escape” command.

8.3.17. Ports

The Ports command prints a list of AX.25 interfaces (ports) that are installed on the mailbox system, along with a description if one has been setup for that port. These ports can be used to make outgoing AX.25 connections with the “Connect” command.

8.3.18. Read <message_number> [<message_number> ...]

The Read command allows you to read a message (or messages) from the current mail area. The read a specific message, you may either type “read <message_number>” or just the number by itself. Typing <cr> at a mailbox prompt displays the message indicated by the “Current msg #” in the prompt.

8.3.19. Send

There are three versions of the Send command, allowing you to compose or forward a mail message:

S[end] <username>[@<host>]
SR [message_number]
SF <user>[@<host>]

The “S” command allows you to compose a message to be sent to the indicated address. The “SR” command allows you to compose a “reply” to either the current message or the message number specified. The subject will be copied and the reply will be sent to the address it was sent from. In composing a message, remember to enter <cr> at the end of each line and to terminate the message with either /EX or CTRL-X followed by <cr>. A message can be aborted with CTRL-A. The “SF” command will forward a copy of the current message to the user specified.

8.3.20. Telnet <hostname>

The Telnet command allows you to initiate a mailbox or BBS session on another host. This allows an AX.25 user to gain access to the TCP/IP network.

8.3.21. Upload [<directory>/]<filename>

The Upload command allows you to transfer an ASCII file from the local host to the mailbox host. A directory other than the current working directory on the mailbox host may be specified. The transfer proceeds line-by-line until the file is complete. The transfer is terminated by entering either a CTRL-Z or /EX and <cr> on an otherwise blank line.

8.3.22. Verbose <message_number> [<message_number> ...]

The Verbose command is the same as the Read command, but all headers on the mail message are displayed. The Read command only displays an abbreviated header.

8.3.23. What [<directory>]

The What command displays a sorted directory listing of the current directory or the specified directory.

8.3.24. Xpert

The Xpert command toggles this mailbox session between using long and short prompts.

8.3.25. Zap [directory>/]<file>

The Zap command deletes <file> in the current directory or the specified directory. You must be granted permission to delete files by the operator of the mailbox system to use this command.

9. AX.25 Services

NOS provides AX.25 services, better known as the standard packet radio protocol that you probably used before you switched to TCP/IP. This allows you to use NOS to move to another frequency and check into the local AX.25 PBBS, or to initiate a keyboard session with a friend who hasn't been convinced to switch to TCP/IP yet.

9.1. Initiating an AX.25 Connection

The connect command is used to initiate an AX.25 connection. The syntax is:

connect ax0 <callsign> [<digipeater> ...]

This initiates an AX.25 session to the specified call sign. Up to 7 optional digipeaters may be given; note that the word "via" is NOT needed. Data sent on this session goes out in conventional AX.25 packets with no upper layer protocol. The de-facto presentation standard format is used, in that each packet holds one line of text, terminated by a carriage return. Two examples are:

connect ax0 n3eua	(connect direct to N3EUA)
connect ax0 n3eua n1fed n0ccz	(connect to N3EUA via N1FED and N0CCZ)

In response to this command, a new session will be initiated, with the following message for the first example above:

Trying N3EUA on ax0...

If the connection is made, the following is displayed:

AX25 session 2 connected to n3eua

At this point, you're connected just like using a plain old TNC, on a session which can be suspended, resumed, or closed as with all NOS sessions.

When you're ready to disconnect, use the <F10> key to escape from the session back to the "net>" prompt, and then type "disconnect", as described in section 9.3.

9.2. File Upload and Download

AX.25 sessions can be recorded to a file and a file can be uploaded in place of typing the information on the keyboard.

9.2.1. record <file> | off

Opens <file> and appends to it all data received or sent on the current AX.25 session. If you are in AX.25 converse mode and want to initiate recording, you will need to use the <F10> key to escape back to command mode to issue the record command. The message "Recording into <file>" will be displayed and another "net>" prompt will be issued. Enter <cr> on a blank line and you will return to the AX.25 converse mode with recording activated. The command "record off" stops recording, displays the message "Recording off" and closes the file.

9.2.2. upload <file>

Opens <file> and sends it on the current AX.25 session as though it were typed on the terminal. If you are in AX.25 converse mode and want to initiate uploading, you will need to use the <F10> key to escape back to command mode to issue the upload command. The uploading is initiated, but the file contents are not displayed on the screen during the uploading and no message is displayed to indicate that the uploading is complete. Enter <cr> on a blank line at the "net>" prompt and you will return to the AX.25 converse mode.

9.3. Terminating an AX.25 Connection

To terminate an AX.25 connection, use the following command:

disconnect [<session #>]

If you are in AX.25 converse mode, press <F10> to escape back to the "net>" prompt to issue this command. If you are running only one session, entering disconnect without arguments will terminate the connection. If you have multiple sessions, entering disconnect without arguments will initiate a close on the current session. If you are running multiple sessions, the "session" command will display a list of these sessions. Entering disconnect with a session number argument will initiate a close on the specified session. After entering disconnect, if you re-enter the disconnected session, you will find the message "AX25 session 1 closed: Normal" and the session will be terminated. Note that "disconnect" is the same as the "close" command and that the two command names can be used fully interchangeably.

10. Network Information and NOS Status

Several commands are available to acquire information about the network and its users and about the status of NOS and its mailbox.

10.1. ax25 heard

Displays the AX.25 "heard" list. For each interface that is configured to use AX.25, a list of all AX.25 addresses (callsigns and host names) heard through that interface is shown, along with a count of the number of packets heard from each station and the interval, in hr:min:sec format, since each station was last heard. This entry is always present even if no packets have been sent. If this list is too long to be displayed on one screen, it will be handled by the "more" function, although a new session will not be created. For example:

```
Interface  Station   Time since send  Pkts sent
ax0        N6GF          0:00:00:11         74
Station   Time since heard Pkts rcvd : Station   Time since heard Pkts rcvd
N6SPE-1       0:00:00:08        328  : N6GF          0:00:00:11         73
KB6KYJ-15     0:00:00:33         17  : HIGH          0:00:00:33        419
K6DOP-15      0:00:00:56         50  : KB6RIH        0:00:16:21        140
N6YUV         0:00:25:07         43  : W6HIR-1       0:00:35:12         10
```

The first two lines indicate that the local host "N6GF" has sent out a total of 74 packets on interface ax0, with the last packet sent 11 seconds ago. The following five lines provide information on eight remote hosts that have been heard on the interface.

To clear the "heard" list and begin accumulating new information, use the command:

ax25 flush

10.2. finger [<finger_file>][@<host>]

This command allows you to “finger” information files on your host or on a remote host. Finger files are described in section B.9 of Appendix B. <finger_file> is the name of the finger information file you wish to query and <host> is the name of the remote host where the file resides.

If you issue the command in the following form:

finger <finger_file>

you can query information from a finger file on the local host, namely your own system. This is useful for testing finger on a system that you know is running.

The command in the following form:

finger @<host>

returns a list of all finger files available on <host>. It also displays a list of users currently logged into the mailbox, in the following form:

```
Current remote users:
User       State    S#  Where
n6spe      CMD      137 44.2.0.100:1033
```

This indicates that user N6SPE is at the mailbox prompt (CMD), in session number 137, on IP address 44.2.0.100, port 1033.

Finally, issuing the command in the form:

finger <finger_file>@<host>

will display <finger_file> from <host>. This initiates a new NOS session and the output displayed is processed by the “more” command, described in Section 4.4.2.

10.3. ping <host> [<length> [<interval>]]

This command is used to see if a remote host is on the air and if so, to determine the quality of the path between the local and remote host. When the command is issued in the form:

ping <host>

the remote host given in the argument is queried once. If it returns an echo, the IP number of the host and the round trip time required are displayed. For example:

44.2.0.96: rtt 4007

In this case, the round trip to the remote host [44.2.0.96] took 4007 ms, or about 4 seconds. If no echo is received, due to the host being off the air, a poor path to the host, or a packet collision, nothing is displayed.

The default is to transmit a short ping packet. The <length> argument sets the byte-length of the transmitted ping packet. This can be used to determine if long packets result in frequent collisions.

If the <interval> parameter is supplied, pings will be repeated indefinitely at the specified interval. This sets up a repetitive test, where the remote host is queried at a time interval given by the number in the second argument, interpreted in milliseconds. Users should be careful not to overdo this testing, as the ping queries will add to channel congestion. The results are displayed in a separate session, which can be suspended or resumed. This session and the accompanying pings are continued until the session is stopped by escaping back to the “net>” prompt and issuing the “close” command.

For example, the following command:

ping eyolo 80 10000

results in the session display:

```
Resolving eyolo... Pinging eyolo (44.2.0.96); data 80 interval 10000 ms:
      sent      rcvd    %       rtt   avg rtt      mdev
         1         1  100      4947      4947         0
         2         2  100      3477      4763       368
         4         3   75      4374      4714       373
         5         4   80      3972      4621       465
```

Five packets were sent to the remote host eyolo and only packet 3 was not received. The column labeled "%" gives the percentage of pings received, "rtt" gives the round-trip time for each packet, "avg rtt" is the running average round-trip time, and "mdev" is the mean deviation of this time.

The round-trip times reported by the "ping" command depend on the setting of the "isat" mode. If "isat" is on, times are reported to the nearest millisecond. If "isat" is off, times are reported in multiples of 55ms. For further information on the "isat" command, see section B.4.6 of Appendix B.

10.4. trace

The trace command is used to monitor the activity on an interface. The syntax of this command is:

trace [<iface> [off | <btio_flags> [<tracefile>]]]

The flags enable or disable tracing and determine the amount of information displayed. Without arguments, trace gives a list of all defined interfaces and their tracing status. When the radio interface <iface> is specified, tracing status is given for this interface only.

The Broadcast/Type/Input/Output (BTIO) flag settings specify the amount of information produced. The BTIO settings are expressed as a hexadecimal number as follows:

```
B=1:  Broadcast filter flag enabled.  Only packets specifically
      addressed to this interface will be displayed.  Broadcast
      packets will be ignored.

B=0:  Broadcast filter flag disabled.

T=2:  Protocol headers are decoded, and the entire packet
      (headers AND data) is also displayed in hexadecimal
      and ASCII, 16 characters per line.  Unprintable characters
      are displayed as dots.

T=1:  Protocol headers are decoded, and data (but not the
      headers themselves) are displayed as ASCII characters,
      64 characters/line.

T=0:  Protocol headers are decoded, but data is not displayed.

I=1:  Enable tracing of input packets.

I=0:  Disable tracing of input packets.

O=1:  Enable tracing of output packets.

O=0:  Disable tracing of output packets.
```

By default, trace output is displayed on the screen. If <tracefile> is given, trace output is saved in the specified file. The trace file is closed by setting the BTIO flags to 0.

For example, to trace the activity on interface ax0, with an ASCII display:

trace ax0 111

To turn off this tracing:

trace ax0 000

10.5. status

The “status” command displays general information about NOS. It is useful for monitoring file access. An example of the display generated by NOS:

```
KA9Q Internet Protocol Package, v911229 (PA0GRI v2.0h)
This version produced by Gerard van der Grinten - PA0GRI

NOS load information:   Code Segment = 0caa - Data Segment = 5949

The system time is Wed Mar 18 06:40:49 1992
NOS was started on Tue Mar 17 21:28:08 1992

Elapsed time => 0 days:09 hours:12 minutes:41 seconds.

The station is currently Unattended.
The 'Message Of The Day' is
Please send email if I am not here -- thanks,  Gary

                 Table of Open Files.
                 --------------------
Name           length   offset hnd acc PSP device type/owner
----           ------   ------ --- --- --- -----------------
README  .TXT      536      536   1 rw 0636 drive C:    [nos.exe]
NOS     .LOG     8757     8757   1 rw 0636 drive C:    [nos.exe]
TMP6    .$$$     3950     2048   1 r  0636 drive C:    [nos.exe]
```

The files listed include README.TXT, which was being displayed by a “more” command, NOS.LOG, which is the NOS log file, discussed in section B.4.26 of Appendix B, and TMP6.$$$, a temporary file created for a directory listing generated by the “dir” command.

10.6. mbox status

Displays information about the current users of the NOS mailbox. An example of the display generated:

```
User        State     S#  Where
n6spe       CMD       145 44.2.0.100:1027
n6cng       LOGIN     149 44.2.0.100:1028
```

This provides the callsigns of the users, the state of their activity, the sequential session number, and the IP address and port under which their session is conducted. The state “CMD” means the user is at the command prompt, and “LOGIN” means the user is in the midst of the login process.

11. Advanced Topics

While this document is intended to be a guide for beginners, there are a few advanced topics of interest to many users that should be mentioned. The presentation of these topics is concise and the reader should consult the “Network Operating System User Reference Manual,” by Phil Karn and Gerard van der Grinten, for details.

Note that the commands included in the NOS configuration file, AUTOEXEC.NOS, and described in Appendix B, can also be used interactively. Further, these commands, when issued without their variable arguments, will report the current values of these arguments. For example, the “route” command, issued without arguments, will display the current routing table. These commands, issued with arguments, can be used to alter the configuration of NOS while it is running. For example, “route add,” described in section B.4.10, can be used to add entries to the routing table, possibly to experiment with alternative routes. To drop routes that are found to be unreliable, the following command can be used:

route drop <dest_host>

where <dest_host> is the IP address or host name for the destination of your packets. Note that if you find better routing methods with this experimentation, you will have to edit the “route add” commands in your AUTOEXEC.NOS file for this routing table to be in effect the next time you run NOS.

11.1. TCP

The TCP command can be used to monitor and control session status at a lower level than provided elsewhere in NOS.

11.1.1. tcp status [<tcb_addr>]

Issued without the optional argument, this command displays the status table for TCP sessions. For example:

```
( 1)tcpRtoAlgorithm            4     ( 2)tcpRtoMin                  0
( 3)tcpRtoMax         4294967295     ( 4)tcpMaxConn        4294967295
( 5)tcpActiveOpens             5     ( 6)tcpPassiveOpens            5
( 7)tcpAttemptFails            0     ( 8)tcpEstabResets             0
( 9)tcpCurrEstab               2     (10)tcpInSegs                109
(11)tcpOutSegs               111     (12)tcpRetransSegs             5
(14)tcpInErrs                  0     (15)tcpOutRsts                 0
    &TCB Rcv-Q Snd-Q  Local socket           Remote socket          State
750f0008     0   335  44.2.0.100:1037        44.2.0.32:smtp         Established
758c0008     0     0  44.2.0.100:1035        44.2.0.96:ftp          Established
74940008     0     0  0.0.0.0:smtp           0.0.0.0:0              Listen (S)
70060008     0     0  0.0.0.0:telnet         0.0.0.0:0              Listen (S)
749f0008     0     0  0.0.0.0:finger         0.0.0.0:0              Listen (S)
6fdb0008     0     0  0.0.0.0:ftp            0.0.0.0:0              Listen (S)
6ff90008     0     0  0.0.0.0:ttylink        0.0.0.0:0              Listen (S)
```

The first seven lines give some TCP-level statistics, including the number of packets received (tcpInSegs), sent (tcpOutSegs), retransmitted due to collisions (tcpRetransSegs), current sessions (tcpCurrEstab), sessions opened by a remote host (tcpPassiveOpens), and sessions opened by the local host (tcpActiveOpens). The table below gives a summary of all existing TCP connections. “&TCB” is the TCP address, “Rcv-Q” and “Snd-Q” are the numbers of characters in the receiving and sending queues, “Local socket” and “Remote socket” give the IP address and port of the local and remote host, “State” gives the state of the session. Note that the remote socket IP address is given as 0.0.0.0 for the listening servers. In this example, there is an established outbound smtp session, assigned to port 1037 on the local host, connected to the smtp server of the host with IP address 44.2.0.32. 335 characters are in the queue ready to be transmitted. There is also an established ftp session, assigned to port 1035 on the local host, connected to the ftp server of the remote host with IP address 44.2.0.96.

Issuing the command with the argument <tcb_addr>, taken from the "&TCB" column in the table, will provide a more detailed table of information on the specified TCP connection. Of particular interest is the last line of the table, which provides information about the retry timer. This is the timer that determines when a packet retransmission will be attempted. It is expressed in the form "running time/threshold time," both given in milliseconds. When the running time reaches the threshold, the pending packet will be retransmitted. If receipt of this packet is not acknowledged, the threshold time will be increased and the running timer will be restarted. When the channel becomes congested, the threshold time becomes very large, and the data throughput drops substantially. This is known as the "exponential backoff" strategy of TCP/IP.

You should display the TCP status before you exit NOS. This will let you know if there are any active connections. For example, you should not exit NOS if someone is running an FTP session with your host. Except for the clatter of your disk drive, you would not be aware of an active FTP session unless you check the TCP status. If there are some active sessions, it is best to leave NOS running so that your TCP/IP node stays on the air. If you think you have "stuck" sessions in which there is no active packet transmission, see section 11.1.3 for information on resetting these sessions.

11.1.2. tcp kick <tcp_addr>

If there is data on the send queue of the TCP connection indicated by the argument <tcb_addr>, taken from the table generated by the "tcp status" command, this will force an immediate retransmission. This can be attempted if the connection appears to be "stuck." This can happen if the route is unreliable, or if there is considerable channel congestion. This command should be used sparingly, as it adds to channel congestion and defeats the TCP strategies to deal with this congestion.

11.1.3. tcp reset <tcp_addr>

Resets the TCP connection indicated by the argument <tcp_addr>, taken from the table generated by the "tcp status" command. This effectively terminates the connection. While this command should not be used indiscriminently, there are situations in which it is useful, primarily when a TCP connection has gotten "stuck." The three common situations in which a TCP connection can get stuck are: remote host has crashed, propagation path of the route to the remote host has deteriorated, and there is extreme congestion on the channel. In each of these situations, tcp status for the corresponding TCP connection will show an increasingly large threshold time.

If you are in a telnet session or an AX.25 connection, you will find times when the channel is so congested that you will wait nearly forever for a response from the remote host. If you want to give up on the session, escape to NOS and close (or disconnect) the session. If there is no response from the remote host, the session can remain half open and should be reset using the tcp reset command.

Similarly, an ftp session can also become stuck. If you are in the midst of a file transfer (get or put operation), escape to NOS and use the "abort" command to terminate the transfer. Then the "quit" command will initiate a close on the session. If there is no response from the remote host, the half open session should be reset using tcp reset.

Another situation in which this command is appropriate for use is the "stuck" smtp session. As discussed in section 6.2.14, when a mail message has begun to be sent, it is "locked." If the route to the destination host is unreliable, the exponential backoff strategy of TCP can cause the transfer to be delayed almost indefinitely. If you want to terminate the transfer, so that you can try again at another time, or using a different route, you can reset the corresponding TCP session. When this is done, the associated mail message is unlocked and can be resent, or killed from BM if you want to give up on the message entirely.

Appendix A. TNC KISS Mode

As explained in section 2.2, your TNC must be in KISS mode to run the NOS program. This appendix provides some information on how to enter the KISS mode.

For the TNC-2 and clones, using a telecommuncations program or an AX.25 packet program to communicate with the TNC, at the “cmd:” prompt, first type “KISS ON” and you will receive the message “KISS WAS OFF” and another “cmd:” prompt. Type “RESTART” and you should note that the CON and STA LEDs will flash three times to indicate you have entered the KISS mode. This command set will then have effect for subsequent power on/off cycles. To return to normal operation, enter the command “param ax0 255” at the “net>” prompt when you are running the KA9Q TCP/IP software package.

For Kantronics TNCs, again using a telecommunications program or an AX.25 packet program to communicate with the TNC, type “KISSMODE ON” while in command mode, followed by “RESET,” to put the TNC in KISS mode. Turning the TNC off and then on will cause the TNC to return to command mode. If you first turn KISSMODE ON and then PERM the value in EEPROM, when the TNC is turned on, it will automatically be in KISS mode.

To put the AEA PK232 into KISS mode, the following commands should be sent to the TNC using a telecommunications program:

```
START 0
STOP 0
XON 0
XOFF 0
XFLOW OFF
CONMODE TRANS
HPOLL OFF
KISS ON
RAWHDLC ON
PPERSIST ON
HOST ON
```

You can also put your TNC in KISS mode using the NOS command “comm” to send straight text to a TNC still in native command mode during NOS startup. The syntax for “comm” is:

```
comm ax0 "<string>"
```

“comm” sends a text-string via <interface>. Note that the string is enclosed by double quote characters to preserve spaces, tabs etc. For example, for the TNC2, you would send the two commands:

```
comm ax0 "KISS ON"
comm ax0 "RESTART"
```

Appendix B. Installation of NOS and BM

As with most software, some effort is needed to get NOS and BM installed and properly configured on your system. This requires setting up the proper directory structure and editing some configuration files. While there are a number of details to attend to, none of this is very difficult.

The configuration files discussed below must be edited with a text editor. Any editor will do, as long as it writes ASCII (simple text) files. A word processing program will work as well, as long as you have it write an ASCII output file.

B.1. Installation Overview

It will be assumed that you have received a disk (or disks) containing at least the files NOS.EXE and BM.EXE. You may also have the files AUTOEXEC.NOS, DOMAIN.TXT, FTPUSERS, and ALIAS, although if these were not included on your distribution, don't worry, as you will be able to create them with an editor.

NOS can be installed in any directory, but for this example, I will assume that it will be installed in \nos, to be referred to as the NOS directory, preferably on your C:\ hard disk drive, or otherwise on your A:\ floppy disk drive. lf you choose to install NOS in another directory, just replace \nos with the chosen directory in the directions below. The steps required for this installation are the following:

1. Create the directory structure described in section B.2.
2. Copy NOS.EXE and BM.EXE to the NOS directory.
3. Edit your AUTOEXEC.BAT file in the root directory to include the NOS directory in the path and to add the environment variables, as described in section B.3.
4. Create the file AUTOEXEC.NOS in the NOS directory and edit as described in section B.4.
5. Create the file DOMAIN.TXT in the NOS directory and edit as described in section B.5.
6. Create the file BM.RC in the NOS directory and edit as described in section B.6.
7. Create the ALIAS file in the root directory and edit as described in section B.7.
8. Create the FTPUSERS file in the NOS directory and edit as described in section B.8.
9. Create the finger file(s) in the directory \nos\finger and edit as described in section B.9.
10. Create the file AREAS in the directory \nos\spool as described in section B.10.
11. Copy or create the mailbox help files in the directory \nos\spool\help as described in section B.11.

Note that the NOS.EXE executable is generally provided in a self-extracting compressed form. It has been compressed to save disk space and transfer time. When you execute the program, it is automatically uncompressed and thus occupies much more memory space than disk space.

B.2. File Structure

Most of the required files will be kept in the NOS directory, but several other directories must also be set up.

\nos\spool	The NOS log file (described in section B.3.26) is normally stored in this directory. The mail log file, MAIL.LOG is created in this directory by smtp and the mailbox AREAS file resides in this directory.
\nos\spool\mail	This directory holds the individual mailboxes for each user name on your system. The extension .txt is added to the user name to form the mailbox name. Mail received by the SMTP server is appended to the mailbox file.
\nos\spool\mqueue	The directory holds the outbound mail jobs. Each job consists of 2 files: an xxx.txt and xxx.wrk file, where xxx is a unique numerical prefix. When the job is being sent, an xxx.lck file is also created. The file sequence.seq is used to keep track of the last message number. The mail transport protocol, SMTP, and the mail program, BM, manage the files in this directory.

\public	This directory and its sub-directories are normally set up to hold the files available for anonymous ftp, described in section B.8.
\nos\finger	This directory holds the finger files, described in sections 10.2 and B.9.
\nos\spool\help	This directory holds the mailbox help files, described in section B.11.

The files that reside in the NOS directory include:

NOS.EXE
BM.EXE
AUTOEXEC.NOS
HOSTS.NOS
BM.RC
ALIAS
FTPUSERS

B.3. DOS Environment — AUTOEXEC.BAT

To set up the DOS environment to run NOS, you will need to make some additions to your AUTOEXEC.BAT file. First, the NOS directory must be included in the PATH command. Consult your DOS manual for information on the PATH command. Second, several environment variables must be set, as described below.

B.3.1. TZ

TZ is the time zone, used in mail headers. An example TZ setting is:

```
set TZ=PST8
```

The first 3 characters are the time zone (PST, the Pacific Standard Time, in the example above) and the fourth character is the number of hours from Universal Coordinated Time (UTC). If TZ is not set, UTC is assumed.

B.3.2. BMRC

BMRC specifies the directory where the BM startup file, BM.RC, is located. This is usually the NOS directory. For example:

```
set BMRC=C:\NOS
```

If BMRC is not set, BM attempts to find BM.RC in the root directory.

B.3.3 TMP

TMP specifies the directory where temporary files are created. For example:

```
set TMP=C:\TMP
```

If TMP is not set, the temporary files are created in the root directory.

B.4. NOS Configuration File — AUTOEXEC.NOS

The AUTOEXEC.NOS file, created in the NOS directory, has a function similar to that of the AUTOEXEC.BAT file in DOS, hence the name. When NOS is executed, it reads AUTOEXEC.NOS and executes all of the commands as if they had been typed in to the program from the keyboard. This provides an easy mechanism for setting up the initial system configuration, including setting the callsign, IP address, hostname, AX.25 parameters, routing table, servers to start, and protocol variables.

An example AUTOEXEC.NOS file is often distributed with the NOS software. This file is fully commented and explains how the example file should be edited for your use. If this file is not available, use a text editor to create the file, following the instructions given in the subsections below. It is suggested that you put the commands into AUTOEXEC.NOS in the order given below.

B.4.1. #

Commands starting with the hash mark (#) are ignored. This is mainly useful for comments in the AUTOEXEC.NOS file. The comments can appear anywhere in the file.

B.4.2. hostname <host_name>

Sets the local host's name (an ASCII string, NOT an IP address). This is usually chosen to be <call>.ampr.org, where <call> is your amateur call sign. The suffix "ampr.org" is officially recognized as meaning an "AMateur Packet Radio" station. Note that in some areas, only <call> is used as the hostname. Your hostname will show up in mail headers and in the greeting messages from the SMTP (mail), FTP (file transfer), and AX.25 mailbox servers. For example:

    hostname n6gf.ampr.org

B.4.3 ax25 mycall <callsign>[-<value>]

Set the local AX.25 address. It does the same thing that "MYCALL" does in your AX.25 TNC. The standard format is used, e.g., KA9Q or WB6RQN-5. The optional dash (-) and <value> following the callsign is the SSID (substation ID), used when it is necessary to distinguish between two or more packet stations with the same callsign. The SSID will be 0 unless explicitly set to another value, which must be a integer number from 0 to 15. This command must be given before any attach command using the AX.25 mode is given. For example:

    ax25 mycall n6gf

B.4.4 ip address <ip_addr>

Sets the local IP address. See section 2.4 for information on acquiring an IP address. For example:

    ip address [44.2.0.100]

B.4.5. attach

The attach command configures and attaches a hardware interface to the system. While many interfaces can be handled by NOS, in this guide, only a single serial interface to a TNC running the KISS protocol will be considered. The general form of the command for our purposes is:

```
attach asy <I/O address> <vector> ax25 <iface> <bufsize> <mtu> <baud>
```

"asy" refers to a standard PC asynchronous interface. Other hardware interfaces are supported by NOS, but will not be covered in this guide.

<I/O address> is the base address of the control registers for the serial interface.

<vector> is the interrupt vector number. Both the I/O address and the vector must be in hexadecimal. (You may put "0x" in front of these two values if you wish, but note that they will be interpreted in hex even if you don't use it).

"ax25" forms IP datagrams to correspond to the AX.25 packet protocol. Other modes are supported by NOS, but will not be covered in this guide.

<iface> is the name by which the interface will be known to the various commands, such as "connect," "route" and "trace". You may choose any name you wish for each interface, although it is common to choose "ax0," "ax1," etc.

<bufsize> specifies the size of the ring buffer in bytes to be statically allocated to the receiver in an asynchronous port; incoming bursts larger than this may (but not necessarily) cause data to be lost. The suggested value is 1024.

<mtu> is the maximum transmission unit size, in bytes. Datagrams larger than this limit will be fragmented at the IP layer into smaller pieces. The suggested value is 256.

<baud> is the baud rate of the serial communication between the computer and the TNC. It must be chosen to be the same value as the baud rate selected on the TNC, as discussed in section 2.3. The suggested value is 4800 for an XT, and 9600 for an AT.

Example 1 — Attach the PC serial card normally known as "com1" (the first controller) to operate in AX.25 mode at 4800 baud with a KISS TNC:

```
attach asy 0x3f8 4 ax25 ax0 1024 256 4800
```

Example 2 — Attach the secondary PC serial card ("com2") to operate in AX.25 mode at 4800 baud with a KISS TNC:

```
attach asy 0x2f8 3 ax25 ax0 1024 256 4800
```

B.4.6. isat [on | off]

Sets the AT clock flag. For a PC/AT, 386, or 486 machine, "isat" should be set to "on." For an original PC or PC/XT (8086 or 8088 machines), "isat" should be set to "off." For example:

```
isat on
```

B.4.7 multitask [on | off]

Set NOS multitasking. When set to "on", NOS continues to run when the user has issued the shell (!) or "mail" command. This enables NOS to process packets and begin sessions even while the user is at the DOS prompt or in BM (or another mail program).

The disadvantage to multitasking is increased memory usage, since memory cannot be allocated by NOS due to the presence of the other program. Thus most versions of NOS grab some memory before actually shelling out and this remains on the heap after NOS is re- entered. Consequently the amount of core memory available drops significantly the first time a shell is used.

When set to "off", all NOS sessions are suspended when the user has issued the shell (!) or "mail" command. The sessions are restarted when the user exits from the DOS subshell, or quits the mail program.

For example:

```
multitask on
```

B.4.8. ip ttl <value>

Sets the default time-to-live value placed in each outgoing IP datagram. This limits the number of switch hops the datagram will be allowed to take. The idea is to bound the lifetime of the packet should it become caught in a routing loop, so the value should be somewhat larger than the diameter of the loop. A suggested value is 16. For example:

```
ip ttl 16
```

B.4.9. param ax0 <value1> <value2>

Param invokes a device-specific control routine. On a KISS TNC interface, this sends control packets to the TNC. Data values are treated as decimal. This command is used to change TNC parameters such as TXDELAY, TXTAIL, PERSIST, and SLOTTIME. This command is TNC-specific, so you must read the documentation for the KISS implementation for your TNC. Most KISS implementations include good default values, so you shouldn't have to use this feature, but if things don't work, you can use the "param" command to try tweaking the TNC.

On a TNC-2, <value1>=1 will set TXDELAY (the time after key down when packet information is transmitted) to <value2> times 0.01 seconds. To set TXDELAY to 0.5 seconds:

```
param ax0 1 50
```

On Kantronics TNCs, <value1>=1 sets TXDELAY to <value2> times 0.01 seconds, as with the TNC-2. For <value1>=2, PERSIST is set to <value2>. For <value1>=3, SLOTTIME is set to <value2> X 0.01 seconds. The default values for these parameters are generally acceptable, so it is usually not necessary to include these commands in AUTOEXEC.NOS.

To properly set up the serial interface to your TNC, you need to include the following commands, which set RTS (ready to send) and DTR (data terminal ready):

```
param ax0 rts 1
param ax0 dtr 1
```

B.4.10. route add

The "route add" command adds an entry to the routing table, defining how your outgoing packets should be routed. Routing can be a complicated issue, so it would be best for you to get some help on this from an experienced local TCP/IP user.

The command syntax is:

```
route add <dest_host>[/bits]|default ax0 [<gateway_host>]
```

Your objective is to route your packets directly to those hosts that you can "hear" and route your packets to the remaining hosts through hosts that can serve as gateways (similar to AX.25 digipeaters). The destination host for your packets is <dest_host>, which is either an IP address or a host name, as defined in the file DOMAIN.TXT, described in section B.4. The gateway is <gateway_host>, also an IP address or host name. Note that it is not necessary to specify the entire sequence of hosts from your system to the destination, but rather only the destination and the first stop on the way. If the routing table on the other hosts has been set up properly, your packets will get to the desired destination. Thus, everyone has to cooperate to keep packets moving.

In my area, N6RQR is over 40 miles away, but he is up in the foothills above me and has a good station, so I can communicate with him directly. His IP address is [44.2.0.54], so I route packets directly to him with the statement:

```
route add [44.2.0.54] ax0
```

The option "/bits" can be used to avoid having to include a route add statement for each and every host you can communicate with directly. To understand this, more details on IP addresses are needed. The IP address is a 32-bit number, with four 8-bit numbers separated with periods (.). Each 8-bit number can range from 0 to 255. The optional "/bits" suffix to the destination host id specifies how many leading bits in the host IP address are to be considered significant in the routing comparisons. If not specified, 32 bits (i.e., full significance) is assumed. With this option, a single routing table entry may refer to many hosts all sharing a common bit string prefix in their IP addresses. For example, [44.0.0.0]/8 would match all addresses in the form [44.*.*.*], where "*" is any number between 0 and 255; the remaining 24 bits are "don't-cares". When an IP address to be routed matches more than one entry in the routing table, the entry with largest "bits" parameter (i.e., the "best" match) is used. This allows individual hosts or blocks of hosts to be exceptions to a more general rule for a larger block of hosts.

The "/bits" option can be used to route packets directly to all the hams on TCP/IP in my town. We have been assigned IP addresses in the range [44.2.0.96] to [44.2.0.111], so our addresses agree in the first 24 bits. Further, 96 decimal is 01100000 binary and 111 decimal is 01101111 binary. Thus, our addresses agree in an additional 4 bits, for a total of 28 bits. The command used is:

```
route add [44.2.0.96]/28 ax0
```

Hams in a neighboring region have been assigned IP addresses of the form [44.4.*.*]. N6RQR can forward packets to them, so we use him as a gateway with the command:

```
route add [44.4.0.0]/16 ax0 n6rqr
```

The special destination "default" is used to route datagrams to addresses not in the routing table; it is equivalent to specifying a /bits suffix of /0 to any destination host. Care must be taken with default entries since two nodes with default entries pointing at each other will route packets to unknown addresses back and forth in a loop until their time-to-live value is exceeded (see "ip ttl" above). (Routing loops for specific addresses can also be created, but this is less likely to occur accidentally).

In my area, we route packets to all other destinations through our local switch, which has the host name "eyolo." The command used is:

```
route add default ax0 eyolo
```

B.4.11. smtp timer <value>

Sets the interval to <value>, in seconds, between scans of the outbound mail queue to determine if there is any new outgoing mail which should be sent. For example, "smtp timer 600" will cause the system to check for outgoing mail every 10 minutes and attempt to deliver anything it finds. For an end-user system that is not normally intended as a mail forwarder, you do not want to set the interval to be too short, as you will be frequently accessing disk needlessly. An interval of 30 minutes (<value>=1800) is probably reasonable. You can also "kick" out the mail manually, as described in section 6.5.1. The suggested command is:

```
smtp timer 1800
```

B.4.12. smtp gateway <host>

Defines the host to be used as a "smart" mail relay. Any mail sent to a host not defined in the file "DOMAIN.TXT" will instead be sent to the gateway for forwarding. You will have to ask locally if there is a host that is used as a mail gateway. If so, include this command in AUTOEXEC.NOS.

B.4.13. tcp mss <value>

Set the TCP maximum segment size in bytes that will be sent on all outgoing TCP connect requests (SYN segments). This tells the remote end the size of the largest segment (packet) it may send. An mss of 216 will force folks to send you packets of 256 characters or less (counting the overhead). Suggested command:

tcp mss 216

B.4.14. tcp window <value>

Sets the window parameter, which establishes the maximum number of bytes that may be outstanding before your system expects an ack. If the window is twice as big as mss, for example, there will be two active packets on the channel at any given time. Large values of window are a problem on the air. Keep mss <= window <= 2*mss. Suggested command:

tcp window 432

B.4.15. start <server>

Starts the specified Internet server, allowing remote connection requests. Suggested commands:

start ftp
start telnet
start ttylink
start smtp
start finger
start ax25

B.4.16. ax25 digipeat [on|off]

Controls whether AX.25 packets addressed to this station as a digipeater will be repeated or not. If you want to operate as a digipeater (for those poor souls not operating TCP/IP), include the following command:

ax25 digipeat on

B.4.17. ax25 filter <value>

Controls whether the list of call signs heard is logged or not. <value> is chosen from:

0: Log source and destination stations
1: Log destination stations only
2: Log source stations only
3: Disable all logging

In this guide, only source station log displays have been covered, so the suggested command is:

ax25 filter 2

B.4.18. ax25 maxframe <value>

Establishes the maximum number of frames that will be allowed to remain unacknowledged at one time on AX.25 connections. This number cannot be greater than 7. Suggested command:

ax25 maxframe 1

B.4.18. ax25 paclen <value>

Limits the AX.25 packet length. This parameter should be less than or equal to the <mtu> defined in the attach command. Suggested command:

ax25 paclen 256

B.4.20. ax25 retry <value>

Limits the number of successive unsuccessful retransmission attempts on AX.25 connections. If this limit is exceeded, link re-establishment is attempted. If this fails “retry” times, then the connection is abandoned and all queued data is deleted. Suggested command:

ax25 retry 10

B.4.21. ax25 window <value>

Sets the number of bytes that can be pending on an AX.25 receive queue. Suggested command:

ax25 window 2048

B.4.22. attended [on | off]

This tells the host initiating the chat session if your system is attended or not. The default is “attend on” so if you do not regularly attend your system, you should use the command:

attended off

B.4.23 motd <string>

Sets the “message of the day,” which is sent when another stations initiates a keyboard chat (ttylink) session with you. For example:

motd "If station is unattended, please leave a mail message"

B.4.24. mbox attended [on | off]

This tells a users logging into the mailbox if your system is attended or not. Keyboard chat sessions are accepted if this flag is on. If you do not regularly attend your system, you should use the command:

mbox attended off

B.4.25. mbox motd <string>

Sets the mailbox “message of the day,” which is sent when a user logs into your mailbox. For example:

mbox motd "Type ’H<cr>’ for help, ’I<cr>’ for information"

B.4.26. mbox expert <on | off>

Sets the prompt displayed in the mailbox. When on, the full prompt shown in Section 8.2 will be displayed. When off, the prompt will be reduced to simply “>.”

B.4.27. log <file>

Defines the name of the file for NOS log entries. Don’t include this command if you don’t want to keep a log. The suggested command is:

log \nos\spool\nos.log

You should read this file occasionally and then discard it, as it can grow to be quite large.

B.4.28 popmail addserver <host> [<seconds>] <protocol> <mailbox> <username> <password>

If you receive your mail through a local switch, you need to contact the administrator of that system to see if “pop” (post office protocol) is used to deliver the mail. If so, you need to include the popmail addserver, kick, and quiet commands in your AUTOEXEC.NOS file.

The “popmail addserver” command defines parameters for receiving mail from a POP server. <host> is the name of the host providing POP service; <seconds> is the interval in seconds between interrogations of the pop server to see if you have mail waiting, which is typically set to 30 minutes (1800 seconds); <protocol> is either POP2 or POP3, depending on the service the host is providing; <mailbox> is the name assigned to your mailbox on the pop host (usually your callsign); and <username> and <password> are the username and password assigned by the administrator of the pop server, which are often both set to your callsign.

For example, N6GF receives pop mail from the host “eyolo.ampr.org” and includes the following command in his AUTOEXEC.NOS file:

popmail addserver eyolo.ampr.org 1800 POP3 n6gf n6gf n6gf

If you regularly receive your mail from a pop server, then the mail address you should give to your correspondents is <user>@<host>, where <user> is your mail username (usually your callsign) and <host> is the name of the pop server. This address should be used in the “reply to” field in the BM.RC file, described in section B.6 of this appendix.

B.4.29. popmail kick <host>

Starts a pop session with the specified host to retrieve mail.

B.4.30. popmail quiet <yes | no>

Set the popmail quiet flag. This flag determines whether you are notified of new mail arriving via pop.

B.4.31. remote <host> kick

If you receive mail from a local switch that is not running a pop server, then you need to include the “remote” command in your AUTOEXEC.NOS file to “kick” any pending mail deliveries. <host> is the hostname of the switch which is forwarding your mail.

If you regularly receive your mail from a switch not running a pop server, then the mail address you should give to your correspondents is <user>%<local_host>@<remote_host>, where <user> is your mail username (usually your callsign), <local_host> is the name of your host, and <remote_host> is the name of the switch which forwards your mail. This address should be used in the “reply to” field in the BM.RC file, described in section B.6 of this appendix.

B.5. The Host Table — DOMAIN.TXT

The file DOMAIN.TXT, residing in the NOS directory, provides a mapping between an IP addresses and symbolic hostnames. It is used by NOS to look up a hostname to figure out the correct IP address to use. These hostnames may be used in establishing a TCP/IP connection, e.g. it is not necessary to enter “telnet [44.2.0.98]”, but merely “telnet n6spe”.

Each entry takes one line, and the fields are separated by tabs. For example:

```
eyolo.ampr.org.        IN       A        44.2.0.96
```

“IN” is the class of the record. It means Internet, and is included in all enties. “A” is the type of the record, and it means that this is an address record. Domain name eyolo.ampr.org therefore has Internet address 44.2.0.96.

Another possible entry is the “CNAME” (Canonical Name) record. For example:

```
pc.n6spe.ampr.org.          IN      CNAME   eyolo.ampr.org.
```

This says that domain name “a.n6spe.ampr.org” is actually an alias for the system with (primary, or canonical) domain name “eyolo.ampr.org.” When a domain name having a CNAME record is given to NOS, the system automatically follows the reference to the canonical name and returns the IP address associated with that entry.

DOMAIN.TXT is a plain text file and it may be easily edited by the user to add, change or delete records. If there is an established group of TCP/IP users in your area, they probably maintain an up-to-date DOMAIN.TXT table that is made available by an ftp file transfer.

B.6. Mail Configuration File — BM.RC

The BM.RC file, created in directory indicated by the BMRC environment variable, provides BM with the configuration needed for the operation of the mailer.

The format for each line in the BM.RC file is:

<variable> <value>

The variables described below are valid in the BM.RC file.

B.6.1. host <host_name>

Sets the local hostname for use in the mail headers. This is a required field. This should match the hostname definition in AUTOEXEC.NOS.

B.6.2. user <username>

Defines the user name of the person who is sending mail. This is also used as the default mailbox for reading mail. On the AMPRNET this is usually set to your call. There is a DOS limit of 8 characters for the user name.

B.6.3. edit <directory>\<editor>

Defines the name of the text editor you wish to use to compose and edit the text of outgoing messages. <editor> is the name of the editor and <directory> is the name of the directory where it is stored on the host computer. The use of edit is optional.

B.6.4. fullname <your full name>

Provides your full name to the mailer for use in the comment portion of the “From:” header line. The use of fullname is optional.

B.6.5. reply <return address>

Defines the address where you wish to receive replies to messages sent. This option is useful if you are operating your pc on a local area network and would like your mail replies sent to a more “well known host,” for example, your local switch. The address specified by reply is used to generate a “Reply-To:” header in outbound mail. The “Reply-To:” header overrides the “From:” header which is the address normally used to reply to mail. This field is optional.

B.6.6. mbox <file>

Specifies the default file to be used for the “save” command, for saving received messages. This file is in the same format as a mailbox and may later be viewed using the -f option of BM. If this option is not used then the default file name is set to mbox. The full pathname of the file must be given, or the file will be created in the root directory.

B.6.7. record <file>

If defined, a copy of each message sent will be saved in <file>.

B.6.8. folder <directory>

If defined folder contains the directory used by the save command. If not defined, files will be saved in the current directory.

B.6.9. smtp <directory>

Defines the directory containing the mailbox files. The default directory is \spool\mail on the current drive.

B.6.10. mqueue <directory>

Defines the directory containing the outgoing mail files. The default directory is \spool\mqueue on the current drive.

B.6.11. Example BM.RC File

Here is an example of a BM.RC file:

```
host n6gf.ampr.org
user n6gf
reply n6gf%n6gf@eyolo
edit c:\bin\vi
fullname Gary Ford
mbox /ford/mail/mbox
folder /ford/mail
smtp /nos/spool/mail
mqueue /nos/spool/mqueue
```

B.7. The Alias File — ALIAS

The ALIAS file provides an easy way to maintain mailing lists. The ALIAS file must be located in the root directory for access from BM, in the NOS directory for access from the mailbox and in both locations for access from both BM and the mailbox. It allows you to send mail to an easily-remembered name, instead of a complicated address. An alias can be any string of characters not containing the “@” symbol. The format for an entry in the alias file is:

```
<alias> <recip1> <recip2> <recip3>
<TAB> <recip4> ...
```

where <alias> is the name you have given to this mailing list, which you will use as a mail address and <recipN> mail address. Note that a long list of aliases can be continued on an additional line by placing a tab or space in the first position of the continuation line. A “#” as the first character on a line indicates a comment.

Some example aliases are:

```
spe n6spe%n6spe@eyolo
dave nn2z@nn2z
# mail to local eyolo users
ey-gang n6gf%n6gf@eyolo n6spe%n6spe@eyolo
```

In the above example, when specifying ey-gang as the recipient, BM will expand the alias into the list of recipients from the alias file.

An alias may not contain another alias from the same file. However, an alias may contain a recipient name that is an alias on the local host or a remote host. For example, if the host “eyolo” contains the following alias in addition to the aliases above:

```
# gang mail
gang se-gang@sesac nc-gang@ncsac ey-gang@eyolo
```

then mail addressed to “gang” will be forwarded to the alias “se-gang” at the host “sesac,” to the alias “nc-gang” at the host “ncsac” and to the alias “ey-gang” on “eyolo” itself. The mail forwarded to “ey-gang” will then be forwarded again, as indicated by the example alias above for “ey-gang.”

The use of an ALIAS file is optional. If you are just getting started with TCP/IP, you might want to wait until you are more familiar with mail addressing before you establish this file. More information on mail addressing is given in section 6.4.

Your local TCP/IP user group may maintain an alias file, so you should inquire as to its availability.

B.8. The FTP and Mailbox Users File — FTPUSERS

Since MS-DOS was designed as a single-user operating system, it provides no access control; all files can be read, written or deleted by the local user. It is usually undesirable to give such open access to a system to remote network users. The ftp server and the mailbox therefore provide their own access control mechanisms.

The file “FTPUSERS,” created in the NOS directory, is used to control remote ftp and mailbox access. The default for ftp is NO file read/write access. Thus, if the FTPUSERS file isn’t created, the ftp server will be unusable. The default for the mailbox is to allow users to read and send mail, but not to have any file read/write access or any access to the mailbox telnet, AX25, or NET/ROM ports. For both ftp and the mailbox, a remote user must first “log in” to the system, giving a login name and password. For ftp, this login name and password must be listed in FTPUSERS for the user to gain access to file transfer commands. For the mailbox, the login name must be a callsign and it and password must be listed in FTPUSERS and have the needed permissions set for the user to transfer files or to use the radio ports.

Each entry in FTPUSERS consists of a single line of the form

```
<username> <password> <path> <permission>
```

There must be exactly one space between each field. Comment lines are begun with “#” in column one.

<username> is the user’s assigned login name.

<password> is the required password. Note that this is transmitted in plain text; therefore it is not a good idea to give general write permission to the root directory. A password of “*” (a single asterisk) means that any password is acceptable.

<path> is the list of directories in which the remote user is permitted to have access to files. The directory names in this list must be absolute, i.e., starting from the root directory (/). Multiple directories are specified by separating them with a “;” character and no spaces around it.

<permission> is a decimal number granting permission for read, write, and overwrite and delete operations for directories in <path> and their subdirectories.

The permission codes are:

1	Read files in directories in <path>
2	Create new files in directories in <path>
4	Overwrite and delete existing files in directories in <path>
8	AX.25 gateway operation allowed from the mailbox
16	Telnet gateway operation allowed from the mailbox
32	NET/ROM gateway operation allowed from the mailbox

A permission of 2 allows a user to write a new file if it does not overwrite an existing file. A permission of 4 allows a user to write a file even if it overwrites an existing file, and to delete files. These permissions are additive. For example, 3 (2 + 1) means that the user is given read and create permission, but not overwrite/delete permission. Similarly, 7 (4 + 2 + 1) means that the user is given read, write, overwrite and delete privileges. Also, 63 (32 + 16 + 8 + 4 + 2 + 1) provides all read, write, overwrite, and delete permissions and gives access to all gateways.

It is a standard convention to keep a repository of downloadable files in the directory \public and to allow users to logon with the username "anonymous" with no password to access these files. In some areas, the public access username is "guest." Every system providing an FTP server is encouraged to provide restricted access to "anonymous" and "guest" users. The appropriate FTPUSERS entries allowing the users "anonymous" and "guest" to read and write files under /public and subdirectories, but not to overwrite or delete any files are:

```
anonymous * /public 3
guest * /public 3
```

You might want to give a friend access to both his or her own directory as well as the public access directory and to have gateway access. For example:

```
n6spe test /users/n6spe;/public 31
```

This gives user "n6spe," with password "test," read, write, overwrite and delete privileges for files under /users/n6spe and /public and their sub-directories, as well as access to your AX.25 and telnet gateways (permission 31 = 1 + 2 + 4 + 8 + 16).

A login name of "univperm" has a special meaning in the user validation mechanism. If univperm is included as a valid login name in FTPUSERS, then any unknown user (not otherwise listed in FTPUSERS) will be given the permissions indicated by the univperm entry. This is primarily useful for giving limited permissions to new mailbox users. For example:

```
univperm * /public 27
```

This gives an unknown user read and write file access and allows operation on the AX.25 and telnet gateways (permission 27 = 1 + 2 + 8 + 16).

B.9. Finger Files

Finger files are information files stored in the \nos\finger directory. It is common practice to set up a file for each of the users on the host. Thus, if n6gf is a user, he should have an information file named "n6gf." This file is edited with a text editor and should include information about the user such as his or her full name, address, and phone number. Additionally, information about the packet system, such as computer, TCP/IP software, TNC, radio, and antenna could be included. Using the finger command, as described in section 10.2, the remote user can access the names of the finger files and then have them displayed on his or her console. Thus, you could have a separate information file for your system and include general information about the files available for downloading and your usual hours of operation. There is no specified format for these files. They will be displayed on the remote host just as they appear in the file.

B.10. Mailbox Areas File

The AREAS file in \nos\spool specifies public message areas which can be accessed from the “areas” command in the mailbox. The area names start in the first column. A line having a space as the first character is taken to be a descriptive comment. The “areas” command in the mailbox, issued without an argument, will display the entire AREAS file to the remote user.

An example AREAS file:

```
            Eyolo Public Message Areas
            ==========================

nos       Discussion of the NOS software
tcpip     General discussion of tcp/ip
packet    General discussion of packet radio

 To enter an area, type "a" followed by the area name.
 For example:"a nos".  To then list the messages
 available to be read, type "l"
```

B.11. Mailbox Help Files

To complete the implementation of the “help” command in the mailbox, a set of help files must be created in the directory \nos\spool\help. These files must be named as follows:

```
area.hlp       finger.hlp     list.hlp       read.hlp       what.hlp
bye.hlp        help.hlp       musers.hlp     send.hlp       xpert.hlp
connect.hlp    info.hlp       nodes.hlp      telnet.hlp     zap.hlp
download.hlp   jheard.hlp     operator.hlp   upload.hlp
escape.hlp     kill.hlp       ports.hlp      verbose.hlp
```

Each file is an ASCII (plain text) file, providing help information for the command indicated by the file name. For example, the file “finger.hlp” provides information about the “finger” command. A set of help files is available and may have been included in the NOS distribution you received, or may be available in the public directories on a local host. If not, you can create these files in a text editor, using the command information provided in this guide. Even if you have acquired a set of help files, you will need to edit the “info.hlp” file, as this file provides information about your packet station.

Appendix C. IP Address Coordinators

Listed below are the coordinators for IP address assignments, as of January 15, 1992. This list is updated several times a year and is generally available on AX.25 PBBS systems.

```
44.002   Bob Meyer          K6RTV    Calif: Sacramento
44.004   Douglas Thom       N6OYU    Calif: Silicon Valley - San Francisco
44.006   Don Jacob          WB5EKU   Calif: Santa Barbara/Ventura
44.008   Brian Kantor       WB6CYT   Calif: San Diego
44.010   Terry Neal         AA6TN    Calif: Orange County
44.012   Steven King        KD7RO    Eastern Washington,Idaho
44.014   John Shalamskas    KJ9U     Hawaii & Pacific Islands
44.016   Jeff Angus         WA6FWI   Calif: Los Angeles - S F Valley
44.017   Dana Myers         KK6JQ    Calif: Antelope Valley/Kern County
44.018   Geoffrey Joy       KE6QH    Calif: San Bernardino & Riverside
44.020   Fred Schneider     K0YUM    Colorado: Northeast
44.022   John Stannard      KL7JL    Alaska
44.024   Dennis Goodwin     KB7DZ    Washington state: Western (Puget Sound)
44.026   Ron Henderson      WA7TAS   Oregon
44.028   Don Adkins         KD5QN    Texas: Dallas
44.030   J Gary Bender      WS5N     New Mexico
44.032   Bdale Garbee       N3EUA    Colorado: Southeast
44.034   Jeff Pierce        WD4NMQ   Tennesee
44.036   Doug Drye          KD4NC    Georgia
44.038   Mike Abbott        N4QXV    South Carolina
44.040   Jeff Jacobsen      WA7MBL   Utah
44.042   Phil Akers         WA4DDE   Mississippi
44.044   Ed Thorne          WB1FEM   Massachusetts: western
44.046   William Simmons    WB0ROT   Missouri
44.048   Jacques Kubley     KA9FJS   Indiana
44.050   Ron Breitwisch     KC0OX    Iowa
44.052   Gary Grebus        K8LT     New Hampshire
44.054   Ralph Stetson      KD1R     Vermont
44.056   Don Hughes         KA1MF    Eastern Mass
44.058   Rich Clemens       KB8AOB   West Virginia
44.060   Howard Leadmon     WB3FFV   Maryland
44.062   Jim Dearras        WA4ONG   Virginia (not DC)
44.064   Dave Trulli        NN2Z     New Jersey: northern
44.065   Bob Applegate      WA2ZZX   New Jersey: southern
44.066   John DeGood        NU3E     Delaware
44.068   Bob Foxworth       K2EUH    New York: Long Island
44.069   Paul Gerwitz       WA2WPI   New York: upstate
44.070   Gary Sanders       N8EMR    Ohio
44.072   Ken Stritzel       WA9AEK   Chicago - North Ill.
44.074   James Curran       KA4OJN   North Carolina (east)
44.075   Charles Layno      WB4WOR   North Carolina (west)
44.076   Kurt Freiberger    WB5BBW   Texas: south
44.077   Rod Huckabay       KA5EJX   Texas: west
44.078   Joe Buswell        K5JB     Oklahoma
44.080   John Gayman        WA3WBU   Pennsylvania: eastern
44.082   Steven Elwood      N7GXP    Montana
44.084   Bob Ludtke         K9MWM    Colorado: Western
44.086   Reid Fletcher      WB7CJO   Wyoming
44.088   Jon Bloom          KE3Z     Connecticut
44.090   Mike Nickolaus     NF0N     Nebraska
```

```
44.092  Pat Davis         KD9UU    Wisconsin, upper peninsula Michigan
44.094  Gary Sharp        WD0HEB   Minnesota
44.096  Don Bennett       K4NGC    District of Columbia
44.098  Garry Paxinos     (waiting)Florida
44.100  Ken Adkisson      WB4FAY   Alabama
44.102  Jeff King         WB8WKA   Michigan (lower peninsula)
44.104  Charles Greene    W1CG     Rhode Island
44.106  Tyler Barnett     N4TY     Kentucky
44.108  James Dugal       N5KNX    Louisiana
44.110  Richard Duncan    WD5B     Arkansas
44.112  Bob Hoffman       N3CVL    Pennsylvania: western
44.114  Steven Elwood     N7GXP    N&S Dakota
44.116  Tom Kloos         WS7S     Oregon: NW&Portland,Vancouver WA
44.118  Jon Andrews       WA2YVL   Maine
44.120  unassigned
44.122  Dale Puckett      K0HYD    Kansas
44.124  David Dodell      WB7TPY   Arizona
44.125  Earl Petersen     KF7TI    Southern Nevada
44.126  Karl Wagner       KP4QG    Puerto Rico
#
# 44.128 is reserved for testing.  Do not use for operational networks.
# You may safely assume that any packets with 44.128 addresses are bogons
# unless you are using them for some sort of testing
#
44.128  TEST
#
# International subnet coordinators by country
#
44.129  Japan             JG1SLY   Tak Kushida, JH3XCU Joly Kanbayashi
44.130  Germany           DL4TA    Ralf D Kloth
44.131  United Kingdom   G6PWY     Chris Baron
44.132  Indonesia        YB1BG     Robby Soebiakto
44.133  Spain             EA4DQX   Jose Antonio Garcia.  Madrid. (EA4DQX @ EA4DQX)
44.134  Italy             I2KFX
44.135  Canada            VE3GYQ   David Toth
44.136  Australia        VK2ZXQ    John Tanner
44.137  Holland           PA0GRI   Gerard Van Der Grinten
44.138  Israel            4X6OJ    Ofer Lapid
44.139  Finland           OH1MQK   Matti Aarnio
44.140  Sweden            SM0RGV   Anders Klemets
44.141  Norway            LA4JL    Per Eotang
44.142  Switzerland      HB9CAT    Marco Zollinger
44.143  Austria           OE1KDA   Krzysztof Dabrowski
44.144  Belgium           ON7LE
44.145  Denmark           OZ1EUI
44.146  Phillipines      DU1UJ     Eddie Manolo
44.147  New Zealand
44.148  Ecuador           HC5K     Ted
44.149  Hong Kong        VS6EL
44.150  Yugoslavia       YU3FK     Iztok Saje
44.151  France            FC1BQP   Pierre-Francois Monet
44.152  Venezuela        OA4KO/YV5 Luis Suarez
44.153  Argentina        LU7ABF    Pedro Converso
44.154  Greece            SV1IW    Manos
```

44.155	Ireland	EI9GL	Paul Healy
44.156	Hungary	HA5DI	Markus Bela
44.157	Chile	CE6EZB	Raul Burgos
44.158	Portugal	CT1DIA	Artur Gomes
44.159	Thailand	HS1JC	Kunchit Charmaraman
44.160	South Africa	ZS6BHD	John
44.161	Luxembourg	LX1YZ	Erny Tontlinger
44.162	Cyprus	5B4TX	C. Costis
44.163	Central America	TI3DJT	Chuck Hast
44.164	Surinam	PZ2AC	Otto Morroy
44.165	Poland	SP5WCA	Andrzej K. Brandt
44.166	Korea	HL9TG	Gary ?
44.167	India	VU2LBW	Lakshman ("Lucky") Bijanki
44.193	Outer Space-AMSAT	W3IWI	Tom Clark

Appendix D. NOS Command Descriptions

Given below are all of the commands listed by the “help” command in NOS, with concise descriptions of these commands. If the command is described in more detail in this guide, the section(s) in which it is referenced is given in parentheses. An asterisk (*) after the section reference indicates that not all of the subcommands have been described in this guide.

!	Escape to DOS (4.5)
abort	Abort an ftp get, put, or dir operation in progress. (7.3)
arp	Address Resolution Protocol. Connects IP addresses with callsigns.
asystat	Display statistics on attached asynchronous communications interfaces.
attach	Configure and attach a hardware interface. (B.4.5)*
attended	Display or set the “host attended” flag. (5.11, B.4.22)
ax25	AX.25 (normal packet) services. (10.1, B.4.3, B.4.16-B.4.21)*
bbs	Enter the local mailbox or BBS (8.2)
cd	Change directory. (4.4.5)
close	Close a session. (5.5, 9.3)
comm	Send a text string via an interface (A)
connect	AX.25 connect request. (9.1)
delete	Delete a file. (4.4.8)
detach	Detach a previously attached interface from the system.
dialer	Set up an autodialer session for an interface
domain	Control and show the working of the name to IP address mapping software.
dump	Shows memory contents in hex and ascii.
dir	List contents of a directory. (4.4.6)
disconnect	Close a session (alias for close). (5.5, 9.3)
echo	Controls telnet keyboard echo.
eol	Controls telnet end of line behavior.
escape	Displays or sets the command-mode escape character in hex.
exit	Exit NOS and return to DOS. (4.6)
finger	Finger information files on remote host. (10.2, B.9)
fkey	Displays or sets values for the programmable keys on the PC keyboard.
ftp	File Transfer Protocol. (4.2, 7.1)
ftype	Displays or sets the default file type (ascii or binary) for ftp.
help	List NOS commands. (4.4.1)
hop	Commands to test the connectivity of the network.
hostname	Display or set hostname. (B.4.2)
icmp	Display or set the flag controlling the ICMP echo reply packets
ifconfig	Display a list of interfaces, with a short status for each.
info	Display information about the version of NOS currently running.
ip	Internet Protocol. (B.4.4, B.4.8)*

isat	Display or set the AT flag. (10.3, B.4.6)
kick	Force an immediate retransmission on a session.
log	Controls logging of server sessions. (B.4.26)
mail	Starts a shell escape to the BM mail program. (6.1)
mbox	Configure and display status of mailbox or BBS. (8.1, 10.6, B.4.24, B.4.25)*
memory	Display and control memory allocation.
mkdir	Create a sub-directory in the current working directory (4.4.9)
mode	Controls transmission mode on AX.25 interfaces.
more	Display file(s) a screen at a time. (4.4.2)
motd	Displays or sets the “message of the day.” (5.1.2, B.4.23)
multitask	Displays or sets the multitask flag (4.5, 6.1, B.4.7)
netrom	Controls NET/ROM services.
nntp	Controls the network news transfer protocol.
nrstat	Displays NET/ROM statistics.
param	Invokes a device-specific control routine. (B.4.9)*
ping	Query a remote host. (10.3)
pop	Controls the post-office protocol for mail delivery.
popmail	Controls the delivery of pop mail. (B.4.28-B.4.30)*
ppp	Controls the point-to-point protocol for serial interfaces.
ps	Display all current processes in the system.
pwd	Display name of current directory. (4.4.4)
rarp	Controls the reverse address resolution protocol.
remote	Send control commands to a remote host (B.4.28)*
rename	Rename a file. (4.4.7)
record	Record telnet or AX.25 session to a disk file. (5.4.1, 9.2.1)
reset	Reset a session.
rip	Controls the RIP service, which manages an IP routing table.
rlogin	Initiates a remote login session.
rmdir	Remove a directory. (4.4.10)
route	Controls the routing table. (11., B.4.10)*
rspf	Controls the radio shortest path first protocol.
session	Controls sessions selection. (4.3, 5.3, 8.4)
sccstat	Display statistics on the SCC driver.
shell	Escape to DOS. Alias for ! (4.5)
smtp	Simple Mail Transport Protocol. (6.5, B.4.11, B.4.12)*
socket	Displays information on active sockets.
source	Reads subsequent commands from a specified file.
start	Start a server. (B.4.15)*

stop	Stop a server.
status	Display NOS status information. (10.5)
tail	Display the contents of the last 18 lines of a file. (4.4.3)
tcp	Transport Control Protocol. (11.1, B.4.13, B.4.14)
telnet	Telnet protocol. (8.2)
test	Start an internal test for an overflow problem.
trace	Monitor packet traffic. (10.4)
ttylink	Initiate a keyboard-to-keyboard chat session. (5.2)
third-party	Restricts third-party traffic.
tip	Creates a session that communicates in “dumb-terminal” mode.
udp	User datagram protocol.
upload	Upload ASCII file to telnet or AX.25 session. (5.4.2, 9.2.2)
watch	Displays current software stopwatch values.
watchdog	Enable or disables the watchdog timer.
?	List NOS commands. Alias for help. (4.4.1)

Beginner’s Guide to TCP/IP on the Amateur Packet Radio Network Using the KA9Q Internet Software

Version 2.0
April 8, 1992

Documenting
KA9Q NOS 911229 (PA0GRI v2.0h)
BM 3.3.2

by

Gary E. Ford, N6GF

Beginner's Guide to TCP/IP on the Amateur Packet Radio Network Using the KA9Q Internet Software

Contents