THIS IS A PREVIEW COPY OF A DOCUMENT UNDER DEVELOPMENT

DRAFT - USE WITH CAUTION (REPORT ERRORS)

JNOS-2

“J” version of Network Operating System
This document rev:0 is matched to software rev 2.0

A software manual for the owner / administrator / user

compiled
by Skip VerDuin K8RRA [k8rra@arrl.net]
validated against JNOS2.0d [tbd]
by Maiko Langelaar VE4KLM

based on work of
MANY OTHERS
(based in part on the NOS Reference Manual,
by Phil Karn, KA9Q and Gerard van der Grinten, PA0GRI
for jnos-1.11)

DISCLAIMER

The authors make no guarantees, explicit or implied,
about the functionality or any other aspect of this product.

Refer to the manuals provided by the manufacturer
of your equipment for installation procedures.

<u>This document is a compilation of the documentation of others</u>. Selected documents applicable to JNOS software are included, the content is then validated specifically against the referenced software on the Linux platform, the text is modified as required to be faithful to the application design intent. Variances on other platforms are shown as differences. Permission from original authors is sought, and all inclusions are displayed in the two part bibliography.

Table of Contents

INTRODUCTION

This document addresses JNOS in a comprehensive manner. This author considers it appropriate to describe the *nos application in one searchable document that permits the owner to obtain, compile, configure, and use jnos. To that end, many documents are sought, collected, inserted, somewhat rearranged, and validated to permit a (hopefully)straightforward approach to the task of application management. Special attention is given to the prerequisites for functionality. When this document is viewed in a web enabled environment, cross reference material is made available thru anchors to remote sites.

The organization of this document is perhaps unusual. The author has many manuals built in chronological order from purchase to use. Over time most use of the manual is in the back pages. This document begins with the use of the software and progresses toward the requirements which must be in place to support such use. It is believed that a good installation results from thorough grounding in user expectation – thus software compilation (the early step) remains in the rear of the manual.

This document is based on Linux platform testing and variances are shown for other suitable platforms. Also, this document addresses the commands available in the target version of software per the version indicated on the title page. Some variances in material are noted as >deprecated< or >superseded< if the feature is planned to be removed, in addition if the revision is noted also then it will have been deleted, yet remain in the text for reference.   Information about commands or other modules which are not included in the target software is often found thru bibliography research.

This document is prepared and distributed (as though it is software) under the GPL for use and modification as the end user desires. First it is widely distributed in PDF form for the purpose of defining the status over time – not to be modified. Second it is privately distributed in computer readable form presently using Open Office tools for an open source format document thus avoiding a proprietary formated product. Conforming to GPL, this work may be modified and redistributed so long as the credits and bibliography remains in tact. Questions, remarks and suggestions about this document are welcome and should be sent to the author identified on the cover sheet

Corrections (comments) to the documentation must include the following information:

1. Document ID (See the Title Page)
2. Page Number
3. Text as it exists [This does not have to be the complete text. But it must be enough to ensure unambiguous identification of the area under discussion.]
4. Text as it is proposed to be or an explanation of the problem which I will convert into appropriate text.

The documents have been prepared using Open Office V2. Submittal using many other word processor formats are easily handled, the authors request care in selecting tools that permit direct conversion into OO.

DO NOT send a copy of the whole document with revisions scattered throughout the text. Limited consideration will be given to whole documents when “compare document” option makes finding changes easy and foolproof. Send the corrections as directed above. If it comes to the Street Address, please ensure you send it on CD in standardized format.

TERMINOLOGY

Here are some of the abbreviations and terminology used throughout this manual.

AREA is a segment of all BBS data available to a user, also called “user area”. Areas are normally associated with a user or administrator created public data.

CALLSIGN is the licenced amateur call assigned by the regulating agency. For the purposes of this document when an example provided the call of the owner is "j8nos" and a remote site is "r1nod", "r2nod", and so forth.

HOSTNAME is the tcp/ip name of a computer or packet system. For the purposes of this document when an example reference is to the owner the host is shown as myjnos, and when another remote system it is shown as "[tbd]".

INTERNET is a worldwide adhoc computer network. It has thousands of computers at schools, companies, and amateur packet radio systems connected to it.

MTU [*Maximum Transmission Unit*] is the maximum data size in one packet. Most often the data referred to by MTU is the transported data, i.e. data frame in a network connection. With tcp/ip, the size of the tcp/ip frame inside the ax.25 packet is the MTU; with net/rom, the size of the data inside the netrom packet is the MTU.

NODE and **MAILBOX** are terms used interchangeably for the user interface when connected to the system.

NRS [*Net/Rom Serial protocol*] is what TNCs with Net/Rom or TheNet eproms talk on the serial port.

PACLEN [*packet length*] is most often used to refer to data size in a link packet. The data in an ax.25 packet can be up to paclen bytes.

PORT or **INTERFACE** means the physical connection to a radio or other system (i.e., radio port or serial interface). These two terms are used interchangeably.

PPP [*point-to-point protocol*] is an alternative to SLIP, that has the advantage of automatically configuring IP addresses, compression, and MTU.

RFCs [*Requests For Comment*] are standard papers used on Internet to discuss and propose new networking protocols and other related topics.

RSPF [*Radio Shortest Path First*] is a tcp/ip routing protocol especially targeted at radio environments.

RTT [*Round Trip Time*] indicates the time needed for data to be sent and acknowledged.

SLIP [*Serial Line IP*] is a way to send IP frames over a serial port without using ax.25 or ethernet to carry the data. You can use SLIP to connect to PCs or Unix systems also running SLIP, and interchange tcp/ip data.

"CTRL-X" *[operator special character]* means to hold down the <CTRL> key and press the <X> key simultaneously. There are several special control characters available and they are described as their utility is required.

[tbd] [*to be determined*] means perhaps an OOPS by the authors not having caught an issue discovered but not resolved prior to publication. The author(s) hope you don't find one. {on the other hand - if you know the answer...}

[Please note that example text is now placed at the beginning of the line. Earlier manuals in ASCII format indented simply so that gateways which handle this message do not complain at having a line beginning with command text; this convention is followed throughout this documentation. It is recommended to NOT return this work into ASCII format at a later date.]

BBS [MAIL & GATEWAY] USER

NOS is structured to support multiple simultaneous users with services such as conferencing and email.

The user interface for JNOS is very similar to most of the well-known BBS programs. This is to provide an easy transition for users. The user interface in the JNOS program is commonly called the 'mailbox' or bbs. These terms will be used interchangeably throughout this document when describing commands.

This manual considers the BBS user and administrative user separately. The BBS user may be local to the node and access the BBS via the console status display, or remote and access the BBS via one of the ports or via Internet. Also, the node itself will be considered as static and interconnected to Internet or mobile and interconnected only via radio. Further, the operating environments considered are AX.25 BBS, TCP/IP Internet, and [tbd] Winlink 2000 email connected.

As a user, you can discover the neighborhood from the login text, using the JHEARD, IHEARD, IPROUTE, and other commands at your disposal, and chatting with other users or the administrator. Once you discover the environment, you can make choices in how to proceed.

BBS ACCESS

Access to jnos BBS mailbox is attainable using a variety of methods. Here are some of the options available:

From jnos console: jnos> telnet localhost the default localhost is jnos BBS.

From jnos console: jnos> bbs is a shortcut to login.

From a user session: ...$ telnet 192.168.2.2 where the IP address is provided for tun in the file “autoexec.nos”. Here it is important to note that access may be widely available via Internet depending on structure and permissions...

From a user terminal: ...$ telnet jnos where “jnos” has been defined in the /etc/hosts file as a static IP on the network.

From remote radio via AX.25: connect vhf station-ssid The syntax and station ID will vary by the remote software in use and your own autoexec.nos.

From remote radio via IP: telnet

The first prompt for IP access will be: “login: “. The request is for your userid and it is typical to use your own call as your userid. The second prompt will be “passwd: “. A password is used to validate authenticity, it is unencripted thruout the entire process so don’t put great stock in secrecy.

Upon a users first login, a registration process is available to collect pertinent data. The prompt string might be: “Area: station (#0) >” where “station” is the current area being viewed and “#0” is the number of mail messages waiting to be read. In all cases the BBS allows access to mail, bulletins, etc, per the permissions established by the owner / administrator of the node.

Access to remote BBS's from jnos BBS is attainable via a variety of methods. Here are some of the options available:

To a remote radio via AX.25: connect vhf station-ssid where station is known or might be found in the jheard station list.

To a remote radio via NET/ROM routing: c node where node is found in the node list.

Upon reaching the remote station the session continues in a manner dictated by the station software and mode of connection.

BBS [USER] COMMANDS

The following commands are available to the users connected to the mailbox. This file is also available separately as mboxcmds.txt, and in brief form in the appendix. Some commands may not be available by choice of the administator at compile time. The typical response of jnos is “Huh?” when a command is not processable as entered by the user. The command vocabulary is:

1. **AREA;** The Area command lists the mail areas that contain messages you may read. To read messages in one of the areas, type 'A <areaname>'. You will then be told how many new, not previously listed messages there are in this area. You can send mail to any of the listed areas as 'S <areaname>'
   1. A gives a short listing, whereas
   2. AF gives a full listing with descriptions
   3. AN shows areas that have new mail since you last logged off.
2. **BYE;** The Bye command is used to close access to the JNOS MBOX session. Control returnes to the process in place prior to access. Bye will close your mailbox file and remove any messages that you have marked “K[ill]”.
3. **CONNECT;** The Connect command establishes an AX.25 protocol connection with a remote station. Use the "ports" or "jheard" command while in the mailbox to discover the proper id of port. It has the following modes:
   1. C[onnect] [port] [callsign] via [<digipeater> . . .]connects to station 'callsign' over interface 'port', possibly via (note the use of 'via' is optional!) 'digipeater...' a list of digipeaters seperated by space.

   2. C[onnect] [node] connects to a remote node with 'node' as either node-call or node-alias over netrom determined path.

4. **CONV;** This is a roundtable discussion feature. 'channel' allows specifying the conference channel you wish to join. Channel default = 0. This command is optionally implimented and the default is to not define the command. To enable the command, compile with #defined [tbd].
   1. conv [<channel>] puts you in converse mode.
5. **DOWNLOAD;** Display a file from the remote system on the local terminal. Note that if the terminal in use includes a capture feature that is spooling display data, then the file is actually saved in that destination. The full path_name is added to the filename if the desired file is not in the working directory. Use the W[hat] command to explore the directory tree of the remote node.
   1. D[ownload] [/][<path_name>/]filename sends a plain ASCII text file. Please note that the character used to separate the path and filename is a "/" (forward slash).
   2. DM download the motd which is otherwise unavailable once you get into the mbox.
   3. DU [/][<path_name>/]filename downloads binary files converted to UUENCODED ASCII. You will need the "uudecode" utility to convert this ASCII file back to binary. Source code, in various languages, for uudecode is widely available on the Internet. Look for uudecode.bas, uudecode.pas, and uudecode.c.
6. **ESCAPE;** The escape command, when entered by itself, will display the character that is current set as the escape character, and whether escape processing is enabled. This character is what will be used if you want to exit from the current session . For instance, if you have started a "chat" session, and you don't get a response from the operator after waiting a few minutes, you can enter the escape character, followed by a <RETURN> or <ENTER>, and the session will be terminated. You will then be returned to the MBOX prompt.
   1. E[scape] [<char> | <integer> | off|on] The escape character may be changed to one of your preference by entering "escape" followed by a <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). Alternatively, an <integer> corresponding to the decimal code for the escape character may be specified. Escape processing can be enabled or disabled by specifying "on" or "off" as the only argument..
   2. EXAMPLES
      1. escape ^Z (the ASCII character <CTRL>Z)
      2. escape X (the character "x" is the new escape)
      3. escape 120 (the character "x" is the new escape)
      4. escape off (suspend escape character processing)
      5. escape on (resume escape character processing)
7. **FINGER** The finger command retrieves personal information about users of a system.
   1. F[inger] displays a list of known users on the current system.
   2. F[inger] [<user_name>] display information about if and when the user last logged in, as well as any information which may be set in the user's finger-file.
   3. F[inger] [<user_name>][@<host>] Perform the same functions detailed above on

another TCP/IP host connected to the network. To get a list of the users on a remote system, enter "finger" followed by a <SPACE> and an "@", then the host name. To get information about a remote user, insert the user name before the "@".

4. EXAMPLES
   1. finger (list the known users on this system)
   2. f sysop (list info about the local user "sysop")
   3. f @wg7j (list the known users at host "wg7j")
   4. f johan@wg7j (display info about "johan" at host "wg7j")
   5. f help[@jnoshost] (shows which pseudo-user names are available or obtaining JNOS system info via f finger command).

8. **HELP** Get on-line assistance for user commands
   1. ? displays a list of the commands that have help descriptions available for them: area bye connect download escape finger help info jheard kill list mboxuser nodes nroutes operator ports read send telnet upload verbose xpert what zap
   2. H <command> Displays help for a specific command.
   3. Example: Display the help text for the command 'connect'.
      1. 'h connect'
9. **IHEARD** The IHeard command shows the tcp/ip systems recently heard. For ax.25 interfaces (ports), show all tcp/ip activity heard, even when this system was not involved in it. For other interfaces, show those systems that we actively routed packets for (i.e.. systems that talked to us.)
   1. I[heard] Show tcp/ip activity for all ports.
   2. I[heard] [<port>] Show tcp/ip activity for <port>.
10. **INFO** Sysop-supplied site information.
11. **IPROUTE** IP[route] shows the available TCP/IP routes the system has configured. It shows the interfaces and gateways involved in the routes, and also the expiration timer (if applicable).
    1. This could be a LONG list if the system has a lot of ip routes. Please ask the sysop for more about the information given in the display.
12. **JHEARD;** The jheard command will display a list of all the station callsigns that have been received as sending packet traffic on the channel, the time since the station was heard last, and the total number of packets received. Warning: if this system has been on the air for very long, and the channels are very active, the "heard" list could be extremely long.
    1. J[heard] displays the "heard" list for all interfaces.
    2. J[heard] [<port>] displays a list of the stations heard on a particular channel. See the Ports command for determining which channel is heard on which port.
13. **KILL** K[ill] <message_number_or_range> [<message_number_or_range> . . .]The kill command allows you to delete messages from the current mailbox (if you have been given that permission by the operator). The kill command only applies to messages in the current mail "area". The current mail area can be checked and

modified with the "A[rea]" command. At least one message number must be supplied. The message numbers you can select from may be displayed with the "L[ist]" command. The second parameter on each line of the list is the <message_number>. You may specify a range of message numbers to kill by placing a "-" between the first and last message numbers of the range. No intervening spaces are allowed. For example:

1. KM will delete all messages in the area.
2. KU will un-kill a message that was previously marked for killing.
3. KA will delete all messages in the area.
4. K 4-7 is equivalent to K 4 5 6 7.

14. **LIST** L[ist] [<starting_msg_number> [<ending_msg_number>]] List prints a list of the messages from the current mailbox (or "area"). For each message, the list contains the number, 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 msg 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.

    1. L by itself will display the headers for all unread messages, if any.
    2. LA list all messages, read or unread
    3. LL display the last <number> of message headers.
    4. LM is the same as 'L'
    5. LB list all bulletins
    6. LS [subject] list messages in the current 'area' with [subject] in the subject line.
    7. LT list all traffic
    8. L> xxx list all messages that have the string 'xxx' in the To: address including numeric strings.
    9. L< xxx list all messages that have the string 'xxx' in the From: address including numeric strings.

15. **Mailbox** Mailbox is an array of seven commands to display connected users and mailbox contents in various ways.

    1. M will display a list of all the current users, how they connected, and their current activity.
    2. MC <path, area> (sysop) copies the current message to the path or area indicated. This command accepts a range of messages, e.g., mc 5-26 junk will copy messages 5 through 26 to area 'junk'.
    3. ML will list all past users of the system, when they were last on and how many times they've connected.
    4. ML n will show the last n users of the system
    5. ML call will list when 'call' last logged in
    6. MM <path, area> (sysop) moves the current message to the path or area desired. This command accepts a range of messages, e.g., mm 5-26 junk will move messages 5 through 26 to area 'junk'.

   7. MS will give some info on the number of messages handled since the system has been up

16. **NODES**   Node prints a list of NET/ROM nodes that are known to this system and for which the nodeids do not begin with '#'. A '>' printed in front indicates that the route has been used in the past 60 seconds
    1. N *   will give info on all known nodes including "hidden" nodes (those with IDs beginning with '#').
    2. N <nodename> displays information about routes (paths) available to <nodename>
    3. NR[oute] command will list all known netrom neighbor stations, with a listing of the path quality to them, number of destinations the neighbor knows, and the obsolescence count.
17. **OPERATOR** O[perator] allows you to "talk" keyboard-to-keyboard with the operator of this NOS system if the system is attended.
    1. When you wish to terminate the chat session, type the escape character on your keyboard, and then press <ENTER> or <RETURN>. The default escape character is "CTRL-X". This escape character may be changed to whatever you prefer by using the "E[scape]" command.
18. **PING** <host>   Check if <host> is alive. Returns RTT.
19. **PORTS** P[orts] prints a list of AX.25 interfaces (ports) that are installed in this system. A description of the port is also given if one has been setup for that port. These ports can be used to make outgoing AX.25 connections with the "C[onnect]" command.
20. **QUERY** This queries the BuckMaster CDRom callbook server for info about the calls given. Example: Q <call> [<call> . . .] More then one call per query is allowed. This command is optionally implemented and the default is to not define the command. To enable the command, compile with #defined [tbd].
21. **READ** Read a message (or messages) from the current mail area. To read a specific message, you may either type "read #" or just the number by itself. You can also simply advance sequentially through the messages by just pressing the <ENTER> or <CR> key. This will display the next message in order. The "read" command displays only an abbreviated portion of the mail headers. If you want to display all the header lines, use the V[erbose] command instead.
    1. #   Reads the one message numbered “#” > the “R” is presumed.
    2. R[ead] #   The full form of the read command.
    3. R[ead] <msg_number_or_range> [<msg_number-or_range> . . .] If there is a specific list of messages you are interested in (determined by the use of the L[ist] command, for instance), you may enter the list of message numbers (separated by spaces) on the "read" command-line.
    4. RM display without interruption all unread messages.
    5. EXAMPLES:
        1. read 3 5 (Display only messages 3 and 5)
        2. 4   (Display message 4)
        3. <CR>   (Display next message)
22. **SEND;** The send command allows you to create a message and jnos will move it into a user mailbox at either this system, or some other system on the network. The

"S" command may also be specialized for other message type or handling (e.g. SP wb7xxx @ n7xxx).

1. S[end] <user>[@ <host>] [< <from_addr>] [$<bulletin_id>]is the general form of the send mail command.
    1. <user> is the id of the recipient in the target mail system. For BBS users, the name is typically the call sign of the BBS user.
    2. <host> is the target system ID. For BBS users, the BBS ID including the ssid define the target system.
    3. <from_addr> may be added if a response is to be to another system. This field is for special use and won't be covered here.
    4. <bulletin_id> may be added and is the subject of [tbd] This field is for special use and won't be covered here.
2. SB <user>[@<host>] (Send Bulletin) As above, but ANY <user> may read the message from the mailbox.
    1. <User> is usually a category rather than an individual user when sending bulletins.
    2. <host> directs the message to one specific host, and if not provided it will cause the message to be delivered to all known hosts. USE WITH CAUTION
3. SC <user>[@ <host>] [< <from_addr>] [$<bulletin_id>] Send a message to more than one user. The system will prompt with "Cc: ", which allows you to add more users to be sent 'carbon copies' of the message. Separate users on the Cc: line with commas.
4. SF <user>[@ <host>] [< <from_addr>] [$<bulletin_id>] The "SF" command will forward a copy of the current message to the user specified.
5. SP <user>[@ <host>] (Send Personal) As in the "S" version, but only the addressee (<user>) may read the message from the mailbox.
6. SR [msg_number] "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 or from the "reply to" field of the message header.
    1. msg_number refers to the message as displayed by the "L"ist command at the time of posting the response.
7. ST <user>[@ <host>] The "ST" command allows you to send "traffic" to <user> specified. Traffic has special meaning to AX.25 specification and is not covered here.
8. EXAMPLES
    1. send kf7xx   (Send a message to the local user, kf7xx)
    2. s kf7xx @ wb7xxx   (Send a message to kf7xx at the wb7xxx host)
    3. sr 3   (Reply to message number 3)
    4. sf n7aaa%n7bbb@w7ccc (Forward current msg to n7aaa at n7bbb via w7ccc)
    5. sc wg7j   (Send with Carbon copy to others) Cc: ka7ehk, n7dva@n7dva

23. **TELNET** T[elnet] <hostname> [<port_number>] [cronly] The telnet command allows you to initiate a TCP connection from the NOS BBS out across the network to access another host. If the remote host is another NOS node it will present a login request to that BBS. This allows an AX.25 user with nothing more than a terminal,

TNC, and radio, to gain access to the TCP/IP network. To quit the session at any time, enter the escape character (<CTRL>X by default, can be changed with the E[scape] command).

1. <hostname> is the remote system address. Enter:
   1. an IP address like: "44.106.132.20".
   2. a qualified name that DNS services can find like: "k8rra.ampr.org".
2. <portnumber> By including the optional port_number, you can connect to any TCP server at the remote host. The default is to be connected to the "telnet" server, which in the case of JNOS software, is the MBOX.
3. "cronly" The qualifying parameter "cronly" may be used to compensate for end-of-line character variation.

24.**UPLOAD;** Create an ASCII file from the terminal keyboard on the working directory at the connected host. Note that if the terminal in use locally has a spooling feature, then effectively a file-to-file transfer can be initiated. Include the full path_name if the file is to be created in a subdirectory of the working directory on the remote system. The working directory is determined at login time from configuration data set by the administrator. Use the W[hat] command to explore the directory tree of the remote node.

1. U[pload] [/][<path_name>/]<filename> The transfer proceeds line-by-line until you terminate transfer by entering either a "<CTRL>Z" or "/ex" as the first item on a blank line.
2. EXAMPLES:
   1. upload kepler.txt
   2. u /public/satelite/oscar13.txt

25.**VERBOSE** This command allows you to read a message (or messages) from the current mail area, and it includes all the header lines for display. The R[ead] command operates the same way, but with abbreviated header lines.

1. V[erbose] <msg_number_or_range> [<msg_number_or_range> . . .] View a specific message or a list of messages with all headers.
2. VH, 'verbose held' is verbose-read-held mesages (sysop only).
3. VM, 'verbose mine' Display, without interruption, all unread messages in the area.

26.**WHAT** W[hat] [/][<path_name>] Generate a sorted directory listing of the current directory or the one specified by the optional path_name. The listing includes the filename (or subdirectory name if there is a "/" appended), the file size in bytes, creation time, and date.

1. EXAMPLES
   1. what (Displays a directory listing of the "current" dir)
   2. w /nos/pub (Display a list of files contained in the "/nos/pub" dir)

27.**XPERT** The Xpert command toggles the prompts that the system gives. The states set as following are remembered at logout and used at next login.

1. X - toggles the prompt between using long and short prompts.
2. XA - toggles the 'current area' indication on or off.
3. XG - effective with 1.11a, addr/#bits - establishes an encapped route for addr,

if permitted. The XG command, when allows by the 0x2000000 permission bit in ftpusers, lets the sender register as a gateway for the indicated host or subnet, for a period of time. This could be used by systems connecting with a dynamic IP address, to facilitate tcp forwarding of bulletins.

4. XN - toggles the 'netrom id' prompt on or off
5. XM - shows the number of lines before -more- prompting occurs in lists
6. XM n - sets the number of lines ...
7. XP - toggles LINEMODE-style prompting for input (telnet/tip users).
8. XR - shows if 'Reply-to' line is added when sending mail. You need to have set an email address when you registered.

28. **ZAP** Z[ap] [/][<path_name>/]<filename> The zap command allows you to delete a file in the current directory of one you specify with the optional path_name. Use of this command requires that permission be granted by the operator of this system.
    1. EXAMPLES
        1. zap myfile.txt   (Deletes myfile.txt in the current dir)
        2. z /nos/mydir/myfile.txt   (Deletes myfile.txt in /nos/mydir)

These commands have been validated on a variety of systems. As differences in use for each platform become known they will be documented below.

LINUX [tbd]

MAC – OS-X [tbd]

DOS [tbd]

BBS SERVICES

Now to expand on the commands a bit with some examples of use. There are many more than found here, it is the enginuity of the user and his/her ability to ferret out new twists from the Internet that will build on the basics presented herein:

How to Connect to Another Node

One of the key features of BBS's in general is the ability to change hosts in a hop-scotch manner and obtain services of remote hosts. The process is generally known as connecting to other hosts. The varieties include basic manual, NET/ROM, etc, and the paths of interconnection include radio RF, and Internet.

Manually connect to another station via RF on HF

NB: The ability to manually connect to another station via radio link was supposed to be the same for HF and VHF/UHF links and available from BOTH the BBS prompt and the sysop console. I would avoid using the console at this time[tbd], since the console method is still not working properly, and is actually quite unstable. Sorry for the trouble .

For example, starting from the BBS prompt after gaining access and logging in (and registering at the initial contact) to the BBS, if you want to connect to VE7DHM on your dxp modem connected to port “dxp38”, you could enter the command, 'c dxp38 ve7dhm', resulting in something like the following:

```
 You have 0 messages.
 Area: ve4klm Current msg# 0.
 ?,A,B,C,D,E,F,H,I,IH,IP,J,K,L,M,N,NR,O,P,PI,R,S,T,U,V,W,X,Z >
 c dxp38 ve7dhm
 Trying... Escape sequence is: +++<CR>
```

Give it a few seconds (DXP38 needs some initializations), then you should see the modem start to make connect requests.

BE CAREFULL !!! THE ABOVE COMMAND WILL CAUSE THE DXP38 TO START SWITCHING YOUR HF RADIO TRANSMIT ON AND OFF AT REGULAR INTERVALS. IN OTHER WORDS, MAKE SURE YOU HAVE YOUR HF EQUIPMENT SETUP PROPERLY. I'M NOT TO BLAME FOR DAMAGE CAUSED BY YOU FAILING TO FOLLOW PROPER PROCEDURES FOR SETTING UP YOUR HF EQUIPMENT. YOU HAVE BEEN WARNED !!!

If the DXP38 is not able to connect after a while (the time could be 18 minutes using the default parameters provided to get started), it will stop and you will see something like the following example:

```
 You have 0 messages.
 Area: ve4klm Current msg# 0.
 ?,A,B,C,D,E,F,H,I,IH,IP,J,K,L,M,N,NR,O,P,PI,R,S,T,U,V,W,X,Z >
 c dxp38 ve7dhm
 Trying... Escape sequence is: +++<CR>
 *** failure with ve7dhm
 Area: ve4klm Current msg# 0.
 ?,A,B,C,D,E,F,H,I,IH,IP,J,K,L,M,N,NR,O,P,PI,R,S,T,U,V,W,X,Z >
```

If the DXP38 is able to connect, you will see something like this instead:

```
 Area: ve4klm Current msg# 0.
 ?,A,B,C,D,E,F,H,I,IH,IP,J,K,L,M,N,NR,O,P,PI,R,S,T,U,V,W,X,Z >
 c dxp38 vo1epc
```

```
Trying... Escape sequence is: +++<CR>
*** connected to vo1epc
[AirMail-3.1.936-B2FHIM$]
(amlem:h3,g:GN37pn)
Welcome to VO1EPC ...
St. John's , Newfoundland
No Traffic
VE4KLM de VO1EPC>
```

Once you are connected, you may proceed to process on the connected remote host. At ANY time you can issue the escape sequence +++ followed by a carriage return, which *should* disconnect you from the station, and return you to the BBS prompt.

After completing whatever you are up to at the remote host, you would enter the command, "Bye" at the prompt “>”, and technically, a disconnect should happen properly, at which point you would be returned back to the former BBS prompt. If the connection does not seem to terminate, you'll have to issue the '+++' command to do so. In the case where you are connected to an AirMail system like above, the available command list is not prompted yet it waits for your command.

The same look and feel applies to using the PTC modem, it's just a different port name for the most part. For example, if you want to connect to WU3V, you would enter a command like 'c ptcpro wu3v', resulting in something like (OK, the “>” prompt example below is the “X”pert version):

```
[JNOS-2.0c4-BFHIM$]
You have 10 messages  -  0 new.
Area: ve4klm (#1) >
c ptcpro wu3v
Trying... The escape sequence is: +++<CR>
```

Give it a few seconds (the PTC also needs a few initializations), then you should see the modem start to make connect requests. The rest is more or less similar to what was shown earlier on in this section.

Posting E-Mail

Mail to other users of the node is simply "S call" where the call is the users call sign and login ID. Listing and reading mail to yourself is intuitive from the "mail:" commands shown by the "?" command.

Forward to another BBS

When forwarding is configured for the jnos node, then mail posted locally will be passed on to another node. Forwarding function is typically an automatic scheduled task for jnos. To determine the configuration of the local node [tbd]

How to get JNOS to forward to WinLink 2000 (WL2K) systems

To summarize, I was able to forward a message from JNOS to Steve's (K4CJX) winlink server over a tcpip telpac connection, then pick off the response he left for me later. Pretty neat.

What's even more cool, is that I can email ve4klm AT winlink DOT org from my commercial email account, then use 'mbox ki K4cjx' and pick it off Steve's WL2K server over the telpac forward. What's more important to note about the WinLink 2000 system is that you can telpac or HF forward to ANY of the WL2K servers to pick your message off.

That particular feature of not needing to access a particular box to get your messages is VERY powerful in my opinion, and I've been thinking about adopting the same feature for a future version of JNOS. Very nice indeed !

Here is what you need to do

1. Configure as per the section: "Configure to forward to WinLink 2000 (WL2K) systems" found later in this manual.
2. Operationally it went like this:
   1. From the BBS prompt I composed a message 'SP K4CJX@K4CJX', then I waited for the SMTP to deliver it to the jnos mail directory for kc4cjx. You can send to anyone else registered on wl2k as well, for example 'SP VE4KLM@K4CJX'.
   2. From the sysop prompt then I did a 'mbox ki k4cjx'. NOTE: Presently in the USA the FCC limits unattended operation so consider the law if you put this on a clock.
   3. Later on in the day I repeated 'mbox ki K4cjx' !>note the capital K which forces reverse forwarding<!, and I received a short reply from him. I'll be able to pick it off any of the WL2K servers either using telpac forwarding or with my DXP38 on HF :-)
3. >OPTIONALLY: At the JNOS sysop console, enter 'mbox trace on' for full debugging to see how the session moves along. If you have problems, that information is very important for me to see if I am to fix any issues you run into.

That's all for now. Remember, this is experimental.

Please REPORT any problems so that I can fix them. Thank you.

Saturday Evening, March 11, 2006 - Maiko Langelaar / VE4KLM

Transer of ASCII File Material

Jnos provides the "Upload" and "Download" commands to move blocks of ASCII data, and "What" to explore the directory structure where files reside. Further, the transfer is to/from the users terminal, thus file-to-file moves require terminal (emulators?) that include a spooling feature. The transfer mechanism is restricted to ASCII text, but by using Binary-to/from-ASCII utility tools most forms of data can actually be moved. While this may seem burdensome, other tools are implemented in the TCP/IP components of jnos that simplify these tasks at the expense of requiring network connectivity at the users disposal.

SYSTEM ADMINISTRATION [SYSOP] USER

This section is devoted to the administrative operator of the system. The administrator starts, controls progress, adjusts characteristics, and terminates jnos sessions. Management of drive space (content), review of logs, analysis of traces, and directing change for users, is the perview of the administrative user.

This Commands Manual contains the commands and their descriptions for using and operating a JNOS tcp/ip and ax.25 packet switch, BBS, and network node. Also contained in this manual is information about the FORWARD.BBS, REWRITE, ALIAS, and USERS files used in a JNOS installation.

STARTING JNOS MANUALLY

On the linux platform it is typical to use a multi-session console to start jnos and serve as the status display.

As user “root”, change directory to the root jnos directory, typically: 'cd /jnos'. As administrator, you might take this moment to check file conditions and perform some of the routine stuff. Then issue the command:

```
 ...# ./jnos options
```

There are several command line options which can be exercised when starting JNOS. These options are used to set environment variables, select configuration and autoexec files, and other functions. Options should be separated by tabs or spaces. If there is an option argument, there should NOT be any whitespace between the option and the argument. The only option not preceded by '-' is the alternate startup file. [tbd] Some are platform specific:

ALL PLATFORMS:

```
 -b     : Use direct video for the screen output.
 -C     : allow core files to be created following certain errors
 -c#    : Set the number of COLUMNS on the screen to #.
 -D     : disallow the interval timer...testing vestige from 1.09 JNOS
 -drootdir  : Set the root directory for the configuration file path.  This may be
        : overwritten by the file specified in the -f option configuration file.
 -e     : Pause after each error line in autoexec.nos
 -fnos.cfg  : Set JNOS config file and path names as indicated in the 'nos.cfg'
        : file.  This overrides the -d option.
 -gn    : Set trace colors, when tracing to console (ANSI.SYS needed):
        :  n=0 => none (default)
        :  n=1 => tailor to gray-scale monitor
        :  n=2 => tailor to color monitor
 -i     : always re-index mail files at startup
 -I     : never re-index mail files at startup (instead, test at 1st access)
 -l     : Do NOT delete .lck files in the mail and news subdirectories
 -mn    : Set the default screen swap mode.
        :   n = 0 Use EMS  (If compiled in and available.)
        :   Default is EMS Available
        :   n = 2 Use memory.  (Default if NO EMS available.)
        :   n = 3 Use a temporary disk file
 -n     : No trace session
 -r#    : Set the number of ROWS on the screen to #
 -t     : trace the autoexec.nos file. You will be asked before each if you want to
        : execute it. 'y' accepts, anything else skips the line.
 -u#    : set the number of status lines, valid values are 0-3
 -v     : Verbose.  Print each line from autoexec.nos before parsing.
 -wf+b  : Set foreground/background colors for system status display
        : default: white on magenta
 -xf+b  : Set foreground/background colors for session status
        : default: white on blue
 -yf+b  : Set foreground/background colors for 'main' window
        : default: the system colors when JNOS starts.(most often ltgray on black)
 -zf+b  : Set foreground/background colors for 'split' window
        : default: white on green
        : 0=black, 1=blue, 2=green, 3=cyan, 4=red, 5=magenta, 6=brown, 7=white/gray
 file   : Name of the startup file. Default: 'autoexec.nos' or set with -fnos.cfg
LINUX (UNIX) PLATFORM (the ./ context):
 -Smgr:o : set the default session manager to <mgr>, with options <o>
 -Tmgr:o : set the default trace session manager to <mgr>, with options <o>
DOS PLATFORM (the C:\ context):
[tbd]
MAC PLATFORM:
[tbd]
```

As JNOS starts, selected parameters will be displayed on the status display. Upon completion of the startup, the status display will prompt “jnos>” for sysop commands.

Environmental Variables

JNOS uses a few environment variables, two of them are required for the proper working of JNOS at all, a few are used if you compiled in the support for callbook CD's, and others are totally optional. Recall that environment variables are established by the DOS "set" command, typically in autoexec.bat, and have the syntax:

set varname=value

Required Variables:

COMSPEC

TZ This is the TIME ZONE variable. It is used throughout the code to determine the proper time with respect to what the PC clock reports. You may wish to set your PC clock to UTC time, and then set TZ=GMT0. This is the simplest method, but it may be annoying to have to convert the time when looking at your status line, or when using other software than JNOS. A benefit is, you do not have to change anything when daylight savings comes in effect or stops. An alternate method requires you to set the true local time (and to modify your pc time when daylight savings becomes effective, since Borland C uses the USA rules for when daylight time starts and ends), and then set TZ following this scheme:

TZ=aaabcddd

aaa = SOLAR time zone name, e.g.: CET

b = '+' or '-' see the following

c = UTC - Localtime (solar). If the value is negative b='-'

If the value is positive b='+'

ddd = Daylight savings zone name, e.g.: CDT

TMP This is the TEMPORARY FILES directory. JNOS puts here some files it uses for various purposes (incoming mail, screen output when changing session, etc.). It is wise to set this to a directory that can be happily erased in case something goes wrong. Do not set TMP=c:\dos ! You would end up with a lot of rubbish. I personally have created a c:\temp directory and have TMP point there. It may be useful also to point TMP to a ramdrive if you have one in XMS or EMS memory, but it must be large enough to accommodate the space demands of JNOS work files. It should be at least twice as large as the biggest SMTP message you expect to receive. Also, don't forget to take into account that JNOS can be doing many tasks "simultaneously". If you do not define TMP, JNOS will store it's work files in the root directory.

Variables required by the callbook code:

QRZDRV The QRZ callbook server needs this variable to know where the CD is. For example: set QRZDRV=f:

QRZPATH Also needed for the QRZ callbook server, it should point to the directory (on QRZDRV) where the callbkc.idx and callbkc.dat files are. For example: set QRZPATH=\callbk

SAMAPI This is required by the SAM callbook code, and specifies the decimal integer value of the samapi tsr software's multiplex id. Example: set SAMAPI=98

Optional Variables:

TERM You can set the TERMINAL TYPE you want to be used for the rlogin client. Also the extended telnet options (not compiled in the standard distributions) use this variable. Example: set TERM=vt100

USER Sets the USER NAME default to be used in rlogin and ftp client connections by overriding default value. The Ident protocol also uses this variable to identify the console user. Example: set USER=ik5pvx

MSGUSER   If you compiled in the mailmsg command, it will use this variable to form the From: field of the message. If undefined, it will use 'sysop'. Example: set MSGUSER=ik5pvx

NOSSYSOP, NOSPASSWD, NOSPATH, NOSPRIVS   These are used to allow the sysop to log in case the ftpusers file gets corrupted. NOSSYSOP is the username and NOSPASSWD is the corresponding password. NOSPATH defaults to the root of the current disk and can be used to specify where NOSSYSOP can go, in the same format as used in the ftpusers file. NOSPRIVS specifies the access privileges. It can be a decimal or a hexadecimal number, again with the same meanings as in ftpusers. It defaults to 0x407f, i.e. can read, erase, use the gateways, be sysop and expert.

NOSSYSOP   Sets username.

NOSPASSWD   Sets the password

NOSPATH Sets the path. If not set, it will be the root directory on the current drive.

NOSPRIVS   Sets the Privileges. If not set it will be FTP_READ + FTP_CREATE + NETROM _CMD + TELNET_CMD + SYSOP_CMD + IS_EXPERT.

NOSPRIVS accepts both decimal and hexidecimal if "0x" precedes the value.

CONSOLE: STATUS DISPLAY

JNOS now has a (up to) 3-line status display which shows (lines 1 and 2 are the 'system' window):

On the first line: time, heap and core free memory, number of connections to the different servers. Then a list of active sessions, where sessions with data waiting are blinking.

The 2nd line shows: the users connected to the bbs. {Eg: BBS: *w0rli johan #ka7ehk !n7ifj}   A status symbol in front shows one of the following:

prefix	*user is*
none	idle
*	a bbs
@	in sysop mode
!	gatewayed out
#	reading or sending mail
=	transferring data (up/download)
^	in convers or sysop-chat mode
?	in none of the above, but not idle...

On the 3rd line is: data depending on the current session.

Always displayed are the current session number and type.

If the sessions are network connections, displayed are remote connection name, tx-queue (bytes for tcp, packets for ax.25), and state of the connection. It then shows the retry timer, with current time left, and initial value.

In 'repeat', 'more' or 'look' sessions, the 3rd line show the command, filename or user/socket for the session.

The number of status lines displayed can be set with the '-uX' commands line option.

'-u0' turns it off. Default is '-u3'

NOTE: if tracing is enabled, this will 'bleed' through the status window. This is NOT a bug, but is inherent to the way tracing works in JNOS. (It would take a major rewrite to fix.) The status window is rebuilt about twice a second to overcome this.

SYSTEM [SYSOP] COMMANDS

This section describes jnos administrative commands. The syntax is one of:

```
command

command literal_parameter

command subcommand <parameter>

command [<optional_parameter>]

command a | b
```

Many commands take subcommands, parameters, or both, which may in turn be optional or required. In general, if a required subcommand or parameter is omitted, an error message will summarize the available subcommands or required parameters. (Giving a '?' in place of the subcommand will also generate the message. This method is useful when the command word alone is a valid command.) If a command takes an optional value parameter, issuing the command without the parameter generally displays the current value of the variable. (Exceptions to this rule are noted in the individual command descriptions.) Two or more parameters separated by vertical bar(s) '|' denote a choice between the specified values.

When one of the choices is default, that choice will be in UPPERCASE. Optional parameters are shown enclosed in [brackets]. Parameters enclosed in <angle brackets> should be replaced with actual values or strings. The generic notation for number values is <nnnn>, and for string values, it is <string_id>. For example, the notation <hostname> means the actual host or gateway callsign or id. Numerical defaults are explicitly stated as such, e.g., "Default = 7."

All commands and many subcommands may be abbreviated. You only need type enough of a command's name to distinguish it from others that begin with the same series of letters. Parameters, however, must be typed in full.

Commands are printed below in bold (if you have the version of this document that supports fancy formatting), and many commands have an example following the textual description of the commands.

The following section contains the comprehensive set of commands for JNOS. Some commands may not be available depending on administrator choice at compile time.

1. **<CR>;** Entering a carriage return (empty line) while in command mode puts you in converse mode with the current session. If there is no current session, JNOS remains in command mode and reissues the 'jnos>' prompt.

2. **!;** An alias for the shell command.

3. **#;** Text starting with the hash mark (#) is ignored. The hash mark is mainly useful for comments in the autoexec.nos file.

4. **?;** Same as the 'help' command.

5. **abort [<session #>];** Abort an FTP session. This command works only on FTP

sessions to terminate a get, put, or dir, operation in progress. If issued without an argument, the current session is aborted.

5.1 NOTE: When receiving a file, abort simply resets the data connection; the next incoming data packet will generate a TCP RST (reset) response to clear the remote server.

5.2 NOTE: When sending a file, abort sends a premature end-of-file.

5.3 NOTE: In both cases abort will leave a partial copy of the file on the destination machine, which must be removed manually if it is unwanted.

6 **arp;** Display the Address Resolution Protocol table that maps IP addresses to their subnet (link) addresses on subnetworks capable of broadcasting. For each IP address entry the subnet type (e.g., AX.25), subnet address and time to expiration is shown. If the link address is currently unknown, the number of IP datagrams awaiting resolution is also shown.

6.1 arp add <hostid> ax25 | netrom <callsign> <iface>OR arp add <hostid> ether | ax25 | netrom | arcnet <ether_addr>|<callsign> <iface> Add a permanent entry to the table. It will not time out as will an automatically created entry, but must be removed with the 'arp drop' command.

6.1.1 arp add 44.26.0.19 ax25 wg7j-2 port1

6.2 arp drop <hostid> ax25 | netrom <iface>or arp drop <hostid> ether | ax25 | netrom | arcnet <iface> Delete a permanent entry from the arp table.

6.2.1 arp drop 44.26.0.19 ax25 port1

6.3 arp eaves [<iface>] [on | OFF] Display or set the 'arp eavesdrop' function per interface. If set, all arp replies overheard on the interface will be logged in the arp table. This speeds up arp discovery, but might build a huge arp table taking up lots of memory. Default for each interface is off.

6.3.1 # Set arp eavesdrop on interface port1

6.3.2 arp eaves port1 on

6.4 arp flush Drop all automatically-created entries in the ARP table; permanent entries are not affected.

6.5 arp maxq [n] Display or set the maximum number of packets to be buffered waiting for an arp resolution to finish. Default = 5.

6.5.1 arp maxq 5

6.6 arp poll [<iface>] [on | off] Display or set the 'arp keepalive polling' per interface. If set, when an arp entry expires, a query will be sent for the address. This keeps the arp table fresh, but possibly retains many unneeded entries.

6.6.1 arp poll port1 on

6.7 arp publish <hostid> ax25|netrom <callsign> <iface> This command is similar to the 'arp add' command, but the system will also respond to any ARP request it sees on the network that seeks the specified address. (Use this feature with great care!)

6.7.1 arp publish 44.26.1.19 ax25 wg7j-2 port1

6.8 arp sort [ON | off] Sorts the arp display

7 **asyconfig** <iface> <parameter> [<value>]; Display or set asy configuration parameters on a UNIX (eg, Linux) implementation of Jnos. These parameters can be changed at any time.

7.1 <iface> is any asy interface name, as given to the attach command.

7.2 <parameter> is one of:

7.2.1 **bufsize** Receive buffer size [default = attach bufsize]

7.2.2 **rxqueue** Receive flow-control threshold [default = 1] After "rxqueue" packets are put in the network "hopper" by the asy receive task, it blocks;

7.2.3 **txqueue** Transmit flow-control threshold [default = 1] after "txqueue" packets are transmitted by the asy transmit task or if EAGAIN/EWOULDBLOCK is returned on a write, it blocks

7.2.4 **status** Display asystat info for <iface>

7.3 <value> is optional, and if specified, provides a new setting for the parameter. If omitted, the current setting is displayed.

7.4 Underlying process: The "rxqueue" and "txqueue" parameters support a simple form of inter-task flow control. For better i/o performance, at the expense of other Jnos tasks, increase the values of rxqueue and txqueue. Increase the rxqueue value if you find other systems retry packets that your system has received but not acked.

7.5 For example; I have both set to 5 on my asy links and to 10 on my SLIP-to-Linux pty link, and it seems to be a fairly good compromise.

8 **asystat;** Display statistics on attached asynchronous communications interfaces (8250 or 16550A), if any.

8.1 Example:

8.1.1 tnc: [NS16550A] [trigger 0xc0] [rlsd line control] 19200 bps [@ 3f8,4]

8.1.1.1 RX: 929462 int, 3683653 chr, 0 hw over, 6 hw hi, 17210 fifo TO, 0 sw over, 126 sw hi

8.1.1.2 TX: 946381 int, 7400204 chr, 0 q, 79401 MS int, 40 THRE TO

8.1.2 tnc2: [NS16550A] [trigger 0xc0] 9600 bps [@ 2f8,3]

8.1.2.1 RX: 722895 int, 2865467 chr, 0 hw over, 8 hw hi, 13075 fifo TO, 0 sw over, 248 sw hi

8.1.2.2 TX: 246420 int, 1889156 chr, 0 q, 1 MS int, 16 THRE TO

8.2 The display for each port consists of three lines.

8.2.1 The first line gives the port label and the configuration flags; these indicate whether the port is a 16550A chip, the trigger character if any, whether CTS flow control is enabled, whether RLSD (carrier detect) line control is enabled, the speed in bits per second, and the port address and IRQ number in hexadecimal. (Receiving the trigger character causes the driver to signal upper layer software that data is ready; it is automatically set to the appropriate frame end character for SLIP, PPP and NRS lines.)

8.2.2 The second line of the status display shows receiver (RX) event counts: the total number of receive interrupts, received characters, receiver overruns (lost characters) and the receiver high water mark. The high water mark is the maximum number of characters ever read from the device during a single interrupt. This is useful for monitoring system interrupt latency margins as it shows how close the port hardware has come to overflowing due to the inability of the CPU to respond to a receiver interrupt in time. 8250 chips have no FIFO, so the high water mark cannot go higher than 2 before overruns occur. The 16550A chip, however, has a 16-byte receive FIFO which the software programs to interrupt the CPU when the FIFO is one-quarter full. The high water mark should typically be 4 or 5 when a 16550A is used; higher values indicate that the CPU has at least once been slow to respond to a receiver interrupt.

8.2.2.1 When the 16550A is used, a count of FIFO timeouts is also displayed on the RX status line. These are generated automatically by the 16550A when three character intervals go by with more than 0 but less than 4 characters in the FIFO. Since the characters that make up a SLIP or NRS frame are normally sent at full line speed, this count will usually be a lower bound on the number of frames received on the port, as only the last fragment of a frame generally results in a timeout (and then only when the frame is not a multiple of 4 bytes long.)

8.2.2.2 Finally, the software fifo overruns and high water mark are displayed. These indicate whether the <bufsize> parameter on the attach command needs to be adjusted (see the Attach Commands chapter).

8.2.3 The third line shows transmit (TX) statistics, including a total count of transmit interrupts, transmitted characters, the length of the transmit queue in bytes, the number of status interrupts, and the number of THRE timeouts. The status interrupt count will be zero unless CTS flow control or RLSD line control has been enabled. The THRE timeout is a stopgap measure to catch lost transmit interrupts, which seem to happen when there is a lot of activity (ideally, this will be zero).

8.3 The "asystat" command for UNIX JNOS is somewhat different from the DOS version.

8.3.1 Example:

8.3.1.1 144.99: cua0, 9600 bps, packet size 255, RTS/CTS disabled, carrier disabled

8.3.1.1.1 RX: ints 2 chars 73 puts 2 rxqueue 5 qlen 0 ovq 0 block 0 buf 1024

8.3.1.1.2 TX: ints 1 chars 57 gets 0 txqueue 2 qlen 0 ovq 0 block 0

8.3.1.2 linux: ttype, 38400 bps, non-blocking, RTS/CTS disabled, carrier disabled

8.3.1.2.1 RX: ints 0 chars 0 puts 0 rxqueue 2 qlen 0 ovq 0 block 0 buf 1024

8.3.1.2.2 TX: ints 0 chars 0 gets 0 txqueue 2 qlen 0 ovq 0 block 0

8.4 The parameters shown are:

8.4.1 ints is the number of times pwait() returned control to the rx or tx task, indicating that select() detected pending data on input or a packet became available for output.

8.4.2 chars   is the number of characters read/written.

8.4.3 puts   number of times received data was placed in the packet Hopper.

8.4.4 gets   number of times data was taken out of the send queue and written to   the device.

8.4.5 rxqueue is described in the asyconfig command helpfile.

8.4.6 txqueue is described in the asyconfig command helpfile.

8.4.7 qlen   is the instantaneous queue size (number of packets).

8.4.8 ovq is the number of times queue flow control (described in asyconfig) was triggered.

8.4.9 block   is the number of times read/write tried to block (in the case of   reads, the number of times it tried to block after data was claimed to be available).

8.4.10 buf is the current receive buffer size, as specified in the attach statement or the "asyconfig" command.

9 **at;** The 'at' command is used to provide automatic starting of other JNOS commands at predetermined times.

9.1 at This form displays each scheduled at command and its id number, so that 'at kill n' can be used (see below).

9.2 at time <cmd>

9.2.1 time takes the form

9.2.1.1 yymmddhhmm   (specific date/time)

9.2.1.2 hhmm   (next occurrence of this time)

9.2.1.3 whhmm   (next occurrence of this time on

9.2.1.4 specified weekday. 0=Sun,1=Mon, ..., 6=Sat)

9.2.1.5 mm   (mm mins after next hour)

9.2.1.6 now+hhmm   (offset from present time)

9.2.2 <cmd> is any legal JNOS command. Multiple word commands must be enclosed in double quotes (" "). Commands which invoke the DOS shell must include '!' or shell as the first word in order for JNOS to recognize that an external command is to be invoked. If the external command requires the command processor, then "/c" may be given as the second word. Example: "! /c x.bat>> e:foo". To automatically reissue the at command when the timer matures, append a "+" character to the <cmd> string.

9.2.3 Example: at 0130 "! cleanup+"

9.2.4 Example: at now+0100 "ax25 flush+"

9.2.5 Example: at 0600 "writeall \"Il est 6h00 GMT\"+"

9.3 at k <id_num> [<id_num>...] This form of the 'at' command kills (i.e., deletes) jobs <id_num>...

10 attach; Configure and attach a hardware interface to the system. Detailed instructions for each driver are in the Attach Commands chapter of [tbd]. An on-line way to obtain a summary of the parameters required for a given device is to issue a partial attach command (e.g., attach asy). This produces a message giving the complete command format.

10.1 attach <hw_type> <label> <mtu> is the common form of the attach command. Later sections describe the command in detail because the remaining parameters vary somewhat for each device. Generally used fields are:

10.1.1 <hw_type> is the kind of I/O device being attached to the system. Each valid device driver is described briefly below.

10.1.1.1 **asy** a standard PC asynchronous I/O port using the National 8250, 16450, or 16550(A) chip or equivalent to the system.

10.1.1.2 **axip** a RFC1226 compatible AX.25 frame encapsulator for transmission of AX.25 frames over the IP.

10.1.1.3 **bpq** allows JNOS to attach interfaces directly to bpqcode without having to use the packet driver interface and the nodedrv4.com TSR. JNOS will talk to bpqcode using the bpq_host interrupt code in version 4.00 and up of bpqcode.

10.1.1.4 **kiss** another kiss interface on a previously attached serial port. This command allows the use of multiport TNCs. The first port must be attached by a prior "attach asy ..." command with <mode> set to ax25 or pkiss. Use mode pkiss if G8BPQ polled kiss mode is desired.

10.1.1.5 **netrom** This makes available a pseudo interface to enable NET/ROM operations.

10.1.1.6 **packet** Driver for use with separate software "packet drivers" which conform to the FTP Software, Inc., Software Packet Driver specification. See the Crynwr (TM) packet driver collection at ftp.crynwr.com, if your hardware (e.g. ethernet card) came without a packet driver.

10.1.1.7 **scc** & **escc** G8FSL/PE1CHL driver for generic Z8530 cards, including DRSI cards.

10.1.1.8 **tun** This is a pseudo device driver to connect the jnos tcp/ip stack to the host network tcp/ip stack. It is called a tunnel, and supersedes the same function in the earlier slip interface.

10.1.1.9 **3c500, pc100, dsri, eagle, hapn, 0, hs, pi, tsync,**>deprecated ?< are all coded in jnos as hw_devices but are untested and are perhaps unsupported.

10.1.2 <label> defines the name by which the interface will be known to various commands, such as "connect", "route", "trace", etc. In general label may be any name, only in a couple circumstances are label names restricted to differentiate between hw_devices, see for example configuring the SCS PTC Pro TNC.

10.1.3 <mtu> is the Maximum Transmission Unit size in bytes. See the section 'Of PACLEN, MTU, MSS, and More' in the Appendix for a discussion of the effect of MTU on system performance.

10.2 attach asy <io_addr> <vector> <mode> <label> <bufsize> <mtu> [<speed>]

[<flags>] Configure and attach an asynchronous serial device. Additional driver information is provided:

10.2.1 The generalized version (valid on DOS platform) is as follows:

10.2.1.1 <io_addr> is the base address of the control registers for the device driver, or the directory entry for the device. Both the address and the vector must be in hexadecimal. You may put "0x" in front of the numbers, but they will be interpreted in hex even without the prefix.

10.2.1.2 <vector> is the comm port IRQ value. If it is suffixed by "_C" then the interrupt service routine for this port is chained to the server list for this IRQ. e.g. com1 = 0x3f8

10.2.1.3 <mode> controls how IP datagrams are to be encapsulated in the device's link level protocol. JNOS does not initialize device mode, thus the physical device must be preset prior to starting jnos. Choices include:

10.2.1.3.1 **ax25** An AX.25 header and a KISS mode TNC control header are added to the front of the datagram before SLIP encoding. Either UI (connectionless) or I (connection-oriented) AX.25 frames can be used.

10.2.1.3.2 **hfdd** specific to SCS and DXP TNCs support pactor protocol. This jnos mode supervises operation of the TNC operating in HOST mode in addition to passing data. This mode appeared first in jnos2.0e[tbd].

10.2.1.3.3 **pkiss** Similar to ax25, except that packets are checksummed and the interfaced devices are queried (polled) regularly for data. Used by G8BPQ polled-kiss software.

10.2.1.3.4 **nrs** Similar to ax25, except that packets are transmitted only when permitted by flow-control signals. Typical of Netrom node stacks using Netrom serial protocol.

10.2.1.3.5 **ppp** Similar to slip, except the protocol is different and typically is capable of compression.

10.2.1.3.6 **ptc** Has similarities to both ax25 and ppp and is specific to the propriatary SCS PTC Pro TNC and pactor (I, II, & III) protocol.

10.2.1.3.7 **slip** Encapsulates IP datagrams directly in SLIP frames without a link header. This is for operating point-to-point lines and is compatible with 4.2BSD UNIX SLIP.

10.2.1.4 <label> Only in mode hfdd are there restrictions to the name of the interface. For an SCS ptc-pro TNC, the first three characters must be "ptc" followed by 1 to 3 user selected characters. For the PTC device, the label must begin "dxp". See also the configuration section below.

10.2.1.5 <bufsize> sets buffer size.

10.2.1.5.1 For ASYNCHRONOUS PORTS, specifies the size of the ring buffer in bytes to be statically allocated to the receiver, incoming bursts larger than <bufsize> may cause data to be lost.

10.2.1.5.2 >deprecated jnos2.0: For ETHERNET, specifies how many PACKETS may be queued in the receive queue at one time. Excess packets will be discarded as they are received. This is useful to prevent the system from running out of memory should another node suddenly develop a case of diarrhea.<

10.2.1.6 <speed> sets the Baud rate to match the device setting. AUTOBAUD is not supported. Default is [tbd].

10.2.1.7 <flags> a string of characters specifying options:

10.2.1.7.1 c - do BPQ checksumming of ax.25 kiss packets
10.2.1.7.2 v - do Van Jacobson TCP header compression on slip intfc
10.2.1.7.3 f - force use of 16550a uart features (eg UM16C550 chip)
10.2.1.7.4 fN - set 16550a trigger level to N (1,4,8, or 14. Default=4)
10.2.1.7.5 n - use NRS-CTS protocol on nrs interface

10.2.1.8 It is well to validate that the #define flags needed to support the option has been compiled into config.h .

10.2.1.8.1 c ax25 AX25 and POLLEDKISS
10.2.1.8.2 v slip SLIP and VJCOMPRESS
10.2.1.8.3 n nrs NRS
10.2.1.8.4 f (any) ASY

10.2.2 The UNIX version of "attach asy" is slightly different from the DOS version, such that:

10.2.2.1 <io_addr> is the Unix device name of a tty type device from the device directory, "/dev/" is prepended to complete the full path. e.g., cua0 or ttyS2.

10.2.2.2 <vector> is built into the device, enter "-" as a place holder.

10.2.2.3 <flags> Flag "f" is not required in a jnos2.0+ Linux environment. <superseded: The fifo trigger level flag, "fN", is reinterpreted as providing a packet size. The value can range from 0 to 255; if it is 0, the original input mechanism is used, otherwise blocking reads are used with termios VMIN set to the specified value. At the moment there is no error checking; if you set the buffer size smaller than VMIN you could lose incoming data. Under some unix versions, setting it to anything other than a multiple of VMIN could cause problems. The tradeoff here is that when the VMIN mechanism is used, input could be delayed by up to 1/5 second on a mostly quiet channel, but on an active channel JNOS will use *much* less CPU time and generally will be much more efficient.>

10.3 attach axip <label> <mtu> <ipaddress> [<callsign>] This pseudo device is used to establish a point-to-point 'tunnel' between two systems over the AX.25 network. Suitable software configured on the other end include jnos, [tbd]. An example: 'attach axip axip1 256 44.26.1.19 WG7J-15'

10.3.1 <ipaddress> is the address of the system on the other side of the 'tunnel,

10.3.2 <callsign> is the optional AX.25 callsign this station is listening on for frames to digipeat. Note that if you want cross-tunnel digipeating to work, each attached axip interface should listen to a different callsign. These should also be different from other callsigns used on this station.

10.4 attach bpq init <vec> <stream> OR attach bpq <port> <label> [<mtu> [<callsign>]] The JNOS BPQ driver reduces memory and processing time used by nodedrv4.com to send packets between JNOS and bpqcode.

10.4.1 The "attach bpq init" sub-command attaches the bpq_host TSR driver, which is accessed via software interrupt <vec> (in hex) for BPQ stream <stream> (in decimal). "attach bpq init" may only be called once.

10.4.2 The second form of the "attach bpq" sub-command is called once per <port>. It associates a BPQ <port> number with an interface name <label>. <mtu> may not exceed 256, and defaults to the ax.25 paclen value. <callsign> defaults to the ax25 mycall value, and establishes the callsign used by this BPQ port.

10.5 attach kiss <asy_iface_label> <port> <label> [mtu] This device establishes a second feature over an already established "asy" device. An example might look like first an 'attach asy 03f8 4 ax25 port1 512 256 9600' followed by an 'attach kiss port1 1 port2'. Again - the TNC must be initialized prior to attaching it.

10.5.1 <asy_iface_label> is the name of the serial port interface previously defined.

10.5.2 <port> is the port number (1-15) to use, and probably should be 1. (the original asy port is automatically port 0 !) It is up to the tnc's firmware to select which port is at what baud. For example, the KPC9612 uses the MYDROP command for this purpose.

10.5.3 <label> is the name for this second kiss port, and does not duplicate the first name.

10.5.4 <mtu> is an optional mtu, if different from the mtu on the first kiss port.

10.6 attach netrom This command must preceed the 'start netrom' command if it is used. >deprecated jnos2.0: This command is automatically executed when the netrom server is started with the 'start netrom' command.<

10.7 attach packet <softintr#> <label> <maxqueue> <mtu> [ipaddress] This is [tbd] [ipaddress] NOTE that the packet driver is often loaded as a TSR in config.sys or autoexec.bat. It is the driver that must be supplied with information such as hardware IRQ, i/o address, and software IRQ. Example: 'attach packet 0x7e eth0 5 1500'

10.7.1 <softintr#> JNOS only needs to know the software IRQ value to interact with this driver. You must pick a software IRQ that is otherwise unused. Values between 0x78 and 0x7f are often suitable.

10.7.2 <maxqueue> [tbd]

10.7.3 <ipaddress> is an optional ip address for the interface. If not set, the system 'ip address' will be used.

10.8 attach [ssc | escc]... Use escc if the Z85230 chip is installed, otherwise use scc. Specify only one 'easy' or one 'init' command, followed by as many 'mode' commands as required to completely define the card.

10.8.1 attach escc <board label> baycom|drsi|opto <base address> <interrupt number> [t<timing channel>] Is the "easy" form of the command (obviously).

10.8.1.1 <board label>

10.8.1.2 <base address>

10.8.1.3 <interupt number>

10.8.1.4 <timing channel>

10.8.2 attach escc <board label> <#chips> init <base address> <spacing between SCC chip base addresses> <offset to channel A control register> <offset to channel B control register> <offset from each channels control register to data register> <address of INTACK/Read Vector port. (0 to read from RR3A/RR2B)> <CPU interrupt vector number> <clock frequency (PCLK/RTxC) prefix with "p" for PCLK> <optional hardware type> <optional parameter for special hardware> <optional t<tick chan> specification to use a channel to improve timing> is the "init" form of the command.

10.8.2.1 <all> for the moment you and your specification sheet for the board are on your own... [tbd]

10.8.3 attach escc <board label> <SCC channel number to attach, 0/1 for first chip A/B><mode> <label> <maximum transmission unit, bytes> <interface speed, e.g, "1200". prefix with "d" when an external divider is available to generate the TX clock.> <buffer size> <optional callsign used on radio channel> <optional s flag to specify software-derived DCD detection>is the "mode" form of the command.

10.8.3.1 <mode> may be one of: "ax25", "kiss", "nrs", or "slip"

10.8.3.2 <all> as above...

10.8.4 Example follows: (see sccg8fsl.txt and scc.c, in JNOS src zipfile, for more info)

10.8.4.1 attach scc scc0 drsi 300 7

10.8.4.2 attach scc scc0 1 init 300 16 2 0 1 0 7 p4915200 8

10.8.4.3 attach scc scc0 0 ax25 vhfa 235 d1200 512 k5arh-3 s

10.8.4.4 attach scc scc0 1 ax25 vhfb 235 d1200 512 k5arh-2 s

10.9 attach tun <label> <mtu> <devid> This device interconnects stacks and supports TCP/IP traffic to the host beyond which lies the Internet. To reach Internet the host stack must forward packets - see tun configuration later in this manual.

10.9.1 <devid> in examples is "0" for the first instantiation.

11 **attended** [off | on]; Displays or sets the global "I am present" flag in JNOS. This flag is used in the welcome header for incoming ttylink connections.

12 **axui** <iface> [<unproto_call> [digipeater_string]] This command allows the console user to create a UI (unproto) session. The administrator sends UI packets from lines entered at the keyboard, and displays received UI packets on the status display. A split-screen session is created where possible. Only one axui session is permitted at a given time. (AXUISESSION must be #define'd in config.h to enable this command / not defined in default 2.0d).

12.1 <iface> use this ax25 interface <iface>

12.2 <unproto call> Unproto supports an address. The default <unproto_call> is "ID".

12.3 [digipeater string] is optional digi calls.

12.4 An session allows a few commands, which begin with '/' in column 1:

12.4.1 /c newcall change the unproto call to <newcall>

12.4.2 /i iface change to interface <iface>

12.4.3 /? or /h display a help message

12.4.4 /t toggle display of the time a packet was received

12.4.5 /q, /b, /e close the axui session

12.5 Example: To use digipeater n5knx-5 for a local ARES unproto net, type:

12.5.1 axui ax0 ares n5knx-5

13 **ax25 <subcommands>;** All AX.25 parameters are configurable per interface. Commands of the form 'ax25 <command>' set the default or global values. Use the 'ifconfig <iface> ax25 <command>' form to set or show the interface-specific values.

13.1 NOTE:THIS IS A CHANGE FROM THE BEHAVIOR EXHIBITED PRIOR TO JNOS VERSION 1.10X16, RELEASED 08FEB94. To set the system default ax.25 parameters, you must do so BEFORE attaching interfaces. After attachment, you must use the 'ifconfig <iface> ax25' command form to show or change values for that interface.

13.2 ax25 alias <aliascall> The alias command shows or sets the system's alias call. If netrom is enabled, this modifies the same call as the 'netrom alias' command. The 'ax25 alias' command is NOT needed in that case! If netrom is not used, this command allows an alias name to be set such that users can connect to it. The alias is typically a short string and should not be a valid callsign, since any SSID is ignored when matching against <aliascall> (6 char MAX). Example: ax25 alias knx

13.3 ax25 bbscall [<bbs_call>] The bbscall subcommand sets or shows the current bbs callsign. Connects to this callsign, when set properly, will "jump-start" the mailbox. That is, after the connect no additional packet need be sent to obtain the mailbox greeting. This subcommand will automatically set the bbscall for all currently-attached AX.25-class interfaces with no bbscall set. For bbscall to function properly, it must differ from the system alias/netrom alias, as well as the link address of the interface (usually set by 'ax25 mycall'). See also 'ifconfig <iface> bbscall <bbs_call>'. Note that the bbscall value is also used for the source address of ax.25 connections initiated from the console, provided that the ax25 ttycall is not set, or that JNOS was *not* compiled with TTYCALL_CONNECT #define'd.

13.3.1 #Example: (in the following order)

13.3.1.1 'ax25 mycall N5KNX-1'

13.3.1.2 'ax25 alias KNX'

13.3.1.3 'ax25 bbscall N5KNX'

13.3.1.4 'attach <(all interfaces)>' or 'ifconfig <name> bbscall <bbscall>' sets only iface <name>

13.4 ax25 bc <iface> The bc command forces an immediate broadcast on the given interface. The particular interface or port must have been enabled with the ax25 bcport command first. If this is so, the ID will be broadcast as set with the ax25 bctext commands.

13.4.1 ax25 bc port1

13.5 ax25 bcinterval [<seconds>] The bcinterval displays or sets the time in seconds between broadcasts. On display, both the interval and the countdown values are shown. Default = 600 (10 minutes).

13.6 ax25 bcport [<iface> [on | OFF]] Display or set the active interfaces for ax.25 broadcasting (i.e. beacons). If no interface is given, all interfaces with ax.25 beaconing enabled will be listed. If interface is given, the status of beaconing for that interface is shown, or set according to the following on or off option. Initial state is off.

13.6.1 ax25 bcport port1 on

13.7 ax25 bctext ["broadcast text"] Display or set the default text to be sent for broadcast messages sent out every ax25 bcinterval seconds. To compose a multi-line message, use \r between the text of each line, ie, "line1\rline2". See also 'ifconfig <iface> bctext ["bctext"].

13.7.1 ax25 bctext "This is the beacon text!"

13.8 ax25 blimit [<value>] Default: 30 Display or set the default AX25 retransmission backoff limit. Normally each successive AX25 retransmission is delayed by twice the value of the previous interval; this is called binary exponential backoff. When 2^(retrycount) reaches or ex-ceeds the blimit <value>, the retry interval is no longer increased. To prevent the possibility of "congestive collapse" on a loaded channel, blimit should be set at least as high as the number of stations sharing the channel. Note that this is applicable only on actual AX25 connections; UI frames will never be retransmitted by the AX25 layer. See also ax25 maxwait, and ax25 timertype.

13.8.1 #Set ax25 blimit to 15

13.8.1.1 ax25 blimit 15

13.9 ax25 dest [<iface>] Display the destinations saved in the heard list, for all interfaces with heard list maintenance enabled, or for the just the specified interface. See also "ax25 heard" and "ax25 filter N".

13.10 ax25 digipeat [<iface> [ON | off]] Display or set digipeating per interface. If no interface is given, all interfaces with digipeating enabled will be listed. If interface is given, the status of digipeating for that interface is shown, or set according to the following on or off option. If cross-band or AXIP digipeating is to be allowed, digipeating must be enabled on both interfaces involved. Default is on.

13.10.1 # Display digipeat status of port1

13.10.1.1 ax25 digipeat port1

13.11 ax25 filter N Default: 0 Sets the ax.25 heard list filtering value. This is a global value that affects all ports with heard list maintenance set to ON.

13.11.1 0 => log both source and destination callsigns

13.11.2 1 => do not log source callsign

13.11.3 2 => do not log destination callsign

13.11.4 3 => do not log any callsign, i.e., disable all heard list updates

13.12 ax25 flush Clears the AX.25 "heard" list (see ax25 heard and ax25 hport)

13.13 ax25 heard [<iface>] Display the AX.25 "heard" list. For each interface that is configured to use AX.25 heard listing (see 'ax25 hport'), a list of all ax25_source addresses heard on that interface is shown, along with a count of the number of packets heard from each station and the time since each station was last heard. The maximum length of the heard table can be set with the 'ax25 hsize' command. If interface is given, only the heard list for that interface is displayed.

13.13.1 ax25 heard port1

13.14 ax25 hearddest [<iface>] Same as "ax25 dest [<iface>]".

13.15 ax25 hport [<iface> [ON | off]] Display or set the status of the ax.25 heard feature. If no interface is given, all interfaces with ax.25 heard enabled will be listed. If interface is given, the status of ax.25 heard for that interface is shown, or set according to the following on or off option. Default is on.

13.15.1 #Display port1 status

13.15.1.1 ax25 hport port1

13.16 ax25 hsize [<size>] Set or display the size of the heard list table. Default is 0 which means no limit.

13.17 ax25 irtt [<milliseconds>] Display or set the initial value of smoothed round trip time to be used when a new AX25 connection is created. The actual round trip time will be learned by measurement once the connection has been established. Default is 5000ms.

13.17.1 #Set irtt to 10 seconds (10000 milliseconds)

13.17.1.1 ax25 irtt 10000

13.18 ax25 kick <axcb> Force a retransmission on the specified AX.25 control block. The control block address can be found with the ax25 status command. This is useful to reactivate connections that have long time-out values.

13.19 ax25 maxframe [<count>] Establish the maximum number of frames that will be allowed to remain unacknowledged at one time on new AX.25 connections. This number cannot be greater than 7. Without <count> it will display the current setting. Note that the maximum outstanding frame count only works with virtual connections. UI frames are not affected. Also note that for optimal performance, a value of 1 should be used. Default is 1 frame.

13.20 ax25 maxwait [<msec>] Sets a limit (in msec) to the retry timeout values. Default = 30000 (30 secs). A value of 0 disables maxwait.

13.21 ax25 mycall [<ax25call>] Display or set the default local AX.25 address. The standard format is used, (e.g. WG7J or KA7EHK-5). This command must be given before any attach commands using AX.25 mode are given.

13.21.1 ax25 mycall wg7j-3

13.22 ax25 paclen [<size>] This sets the default paclen used when attaching interfaces that will carry AX.25 connections. See also 'ifconfig <iface>

13.22.1 ax25 paclen'. Default is 256 bytes.

13.22.2 This parameter limits the size of I-fields on new AX.25 connections. If IP datagrams or fragments of datagrams larger than paclen are transmitted, they will be transparently fragmented at the AX.25 level, sent as a series of I frames, and reassembled back into a complete IP datagram or fragment at the other end of the link. IP datagrams will not be affected if this parameter is greater than or equal to the MTU of the associated interface.

13.22.3 If NET/ROM communication is configured, the NetRom MTU value should be Paclen - 20. !!! The Net/Rom header takes 20 bytes, and is part of the AX25 data. Default netrom mtu is 236.

13.22.4 Note1: the AX.25 Level 2 Version 2 definition specifies a maximum paclen of 256 bytes. Some systems are not equipped to handle larger packets (e.g. G8BPQ based systems), so be careful when using this parameter.

13.23 ax25 pthresh [<size>] Display or set the poll threshold to be used for new AX.25 Version 2 connections. The poll threshold controls retransmission behavior as follows. If the oldest unacknowledged I-frame size is less than the poll threshold, it will be sent with the poll (P) bit set if a time-out occurs. If the oldest unacked I-frame size is equal to or greater than the threshold, then a RR or RNR frame, as appropriate, with the poll bit set will be sent if a time-out occurs. The idea behind the poll threshold is that the extra time needed to send a "small" I-frame instead of a supervisory frame when polling after a time-out is small, and since there's a good chance the I-frame will have to be sent anyway (i.e., if it were lost previously) then you might as well send it as the poll. But if the I-frame is large, send a supervisory (RR/RNR) poll instead to determine first if retransmitting the oldest unacknowledged I-frame is necessary; the time-out might have been caused by a lost acknowledgment. This is obviously a tradeoff, so experiment with the poll threshold setting. The default is 128 bytes, one half the default value of <paclen>

13.24 ax25 reset <axcb> Delete the AX.25 connection control block at the specified address. This deletes a connection and everything associated with it. The control block address can be found with the 'ax25 status' command.

13.25 ax25 retries [<count>] Limit the number of successive unsuccessful retransmission attempts on new AX.25 connections. If this limit is exceeded, link re-establishment is attempted. If the link can't be re-established in <count> times, then the connection is abandoned and all queued data is deleted. Default is 5.

13.26 ax25 route [<subcommand>] Without optional subcommands, display the AX.25 routing table that specifies the digipeaters to be used in reaching a given station.

13.26.1 ax25 route add <target> <iface> [[via] digis ...]OR ax25 route perm <target> <iface> [[via] digis ...] Add an entry to the AX.25 routing table. The route added may be replaced automatically by a new one, unless "perm" is used instead of "add". Replacement may occur when digipeaters are specified in an AX25 link from the node or a connection is received from a remote station via digipeaters. Such automatic replacement is usually desirable; use "route add perm ..." to prevent this where necessary.

13.26.1.1 <target> is the destination call to reach via digipeaters.

13.26.1.2 <iface> is the interface to use for this route, i.e. JNOS allows different digi routes for different interfaces.

13.26.1.3 [digis...] is a list of one or more digipeaters, separated by spaces and/or commas. The keyword "via" is optional.

13.26.1.4 Example: ax25 route add k7uyx-1 port1 wg7j wa7tas n7dva

13.26.1.5 Example: ax25 route perm k7uyx-1 port1 wg7j wa7tas n7dva

13.26.2 ax25 route drop <target> <iface> Drop an entry for <target> from the AX.25 routing table.

13.26.2.1 ax25 route drop k7uyx-1 port1

13.26.3 ax25 route mode <target> <iface> [mode] This sets the mode ip links to the destination call should used when established. If nothing is given for a certain destination, the interface default mode is datagram.

13.26.3.1 <target> the destination call

13.26.3.2 [mode] Sets the interface ip mode to one of datagram | interface | vc for the target. The optional mode spec is one of the following:

13.26.3.2.1 datagram is unconnected mode (AX25 UI frames), and is the default mode.

13.26.3.2.2 interface is to use the mode as set by the 'mode' command. (See also the 'mode' command)

13.26.3.2.3 vc is a virtual circuit (ax25 connected mode, meaning that ip frames are sent using ax.25 connections)

13.26.3.3 Example: ax25 route mode k7uyx-1 port1 vc

13.27 ax25 status [<axcb>] Without an argument, display a one-line summary of each AX.25 control block. If the address of a particular control block is specified, the contents of that control block is shown in more detail. Note that the send queue units are frames, while the receive queue units are bytes.

13.28 ax25 t2 [<milliseconds>] Default: 1000 Display or set the AX.25 "resptime" timer. Value is in milli-seconds. Default is 1 sec, and minimum is 0ms. This timer lets JNOS eliminate some redundant transmissions and optimize what it sends by possibly adding ACKs to I-frames, by delaying the transmission of packets. A value of zero disables the use of the t2 timer, as does the use of datagram mode (UI) [see mode command].

13.29 ax25 t3 [<milliseconds>] Default: 0 Display or set the AX.25 idle "keep alive" timer. Value is in milliseconds. Default is 0, i.e. no 'keep-alive' polling.

13.30 ax25 t4 [<seconds>] Default: 300 Display or set the AX.25 Link "redundancy" timer. Value is in seconds. When no exchange has been had during this time the link is reset and closed. Default = 300 seconds (5 minutes).

13.31 ax25 timertype [type] Sets or displays the type of timer used for retransmission and recovery. Default is linear.

13.31.1 [type] is one of the following:

13.31.1.1 **Linear** means that each retry will use a time-out that is RTT greater than the previous time-out. I.e. 4sec, 8sec, 12sec, 16sec, etc.

13.31.1.2 **Exponential** means that each retry will use a time-out that is twice the previous time-out. I.e. 4sec, 8sec, 16sec, 32sec, etc.

13.31.1.3 **Original** means that each retry will use a time out that is twice the RTT, I.e. 4sec, 8sec, 8sec, 8sec, etc.

13.31.2 Example: ax25 timertype exponential

13.32 ax25 ttycall [ttycall] Set or display the tty-link callsign for direct keyboard access. Set both 'attended on' and 'mbox attend on' to use this function. The ttycall value is also used for the source address of ax.25 connections initiated from the console, provided that JNOS was compiled with TTYCALL_CONNECT #define'd.

13.33 ax25 version [n] Default: 2 Display or set the version of the AX.25 protocol to attempt in making new connections. Version 1 is the version that does not use the poll/final bits.

13.34 ax25 window [<size>] Default: Linux=2048 DOS=512 bytes Display or set the number of bytes that can be pending on an AX.25 receive queue. When exceeded, I frames will be answered with RNR (Receiver Not Ready) responses. RNR presently applies only to suspended interactive AX.25 sessions, since incoming I-frames containing network (IP, NET/ROM) packets are processed immediately and are not placed on the receive queue. IMPORTANT: when a connection carries both interactive and network packet traffic, an RNR generated because of backlogged interactive traffic will also stop network packet traffic from being sent.

14 **bbs;** Switches the session on the terminal from administrator to BBS user and logs into your own mailbox system automatically. This command is a shortcut to typing: "telnet localhost" followed by login process.

14.1 JNOS is configured for this shortcut by:

14.1.1 compiled with both MD5AUTHENTICATE and EXPEDITE_BBS_CMD #define'd.

14.1.2 add an entry to "net.rc" file with the systems hostname, the desired userid, and the corresponding password.

14.1.2.1 Example entry: n5knx.ampr.org n5knx passemoi

15 **bulletin <subcommand>;** The 'bulletin' command establishes how JNOS treats incoming bulletins based on their R: line headers. These commands are available when JNOS was compiled with RLINE #define'd, and are highly recommended for BBS operations. While many defaults are OFF for historical reasons, I think most BBS operators will want to turn these features ON!

15.1 bulletin check [on | OFF] Displays or sets a flag indicating bulletin forwarding is to be optimized. If <ON>, an incoming bulletin's R: lines are scanned for BBSes to which we forward. If any are found, the bulletin is marked as having already been forwarded to that BBS. When invoked with no options, the list of BBSes to which we forward is also displayed. Note: only the first NUMFWDBBS (currently 8) BBSes in forward.bbs are checked in the R: lines, hence the most active BBSes should occur first in forward.bbs.

15.2 bulletin date [on | OFF] Displays or sets whether the date associated with a message is the originate date (if ON), or the date received (if OFF). The originate date is obtained from the last R: line, and should yield better operation of the bulletin expire mechanism. See also 'bulletin holdold'.

15.3 bulletin holdold [#days] Displays or sets the number of days since message origination, above which an incoming bulletin is placed into a hold state. A held bulletin is visible only to the sysop, and has a "X-BBS-Hold: Age" header added. This feature only works when 'bulletin date ON' is in effect.

15.4 bulletin loophold [<count>] Default: 2 Displays or sets a counter which, if exceeded by the number of times our call appears in a message's R: line headers, causes the message to be Held to avoid a message-forwarding loop. A held message is visible only to the sysop, and has a "X-BBS-Hold: Loop" header added.

15.5 bulletin return [on | OFF] Displays or sets how a message's return address is obtained. If ON, the return address is obtained from the last R: header line when available. If OFF, sender%forwarding.bbs@mycall is used.

15.6 Note that for the mailbox SR (send-reply) command to work, the system's rewrite file must be capable of handling the '%' symbol. A recommended rewrite rule is: *%*@YOURCALL* $1@$2 r

16 **callserver hostname [hostname>];** Sets the host address of the system to be querried by the "callbook" or mailbox "Q" command. The specified server system responds at tcp port 1235. <hostname> may be that of the local system if a server has been started with the "start" command. See the "readme.now" file for further information that pertains to QRZ, SAM, and Buckmaster cdrom databases.

17 **comm <asy_iface> <"text_string">;** Sends "text_string", followed by a CR character, to the specified asynch interface. Normally, this command is used to place a TNC into KISS mode when JNOS is started, but it may also be useful to send AT commands to attached phone modems.

17.1 Example 1: Ensure kiss mode is on:

17.1.1 comm tnc "kiss on"

17.1.2 comm tnc "restart"

17.1.3 pause 4

17.1.4 param tnc txdelay 10

17.2 Example 2: have telephone modem answer on first ring:

17.2.1 param dialup up

17.2.2 comm dialup "atz e0 s0=1"

17.2.3 pause 2

17.2.4 start tip dialup modem 360

18 **connect <iface> <destination> [<digi1,digi2...digi3>];** Initiate an ax25 connection thru interface with a remote call. Note that there is only a 'space' between <destination> and <digi...> although a "v" or "via" will be ignored if it is used per various other sources of command syntax. Use "ax h" if needed to identify interface and active remote stations. ***NOTE: As of jnos2.0d1 this command fails regularly for HF connections thru ptc... and dxp... TNCs and should be avoided until the bug is repaired and this note is removed***

18.1 <iface> is a port name assigned by the attach label.

18.2 <destination> may be either callsign-ssid or an alias.

18.3 <digi...> list of digipeater node call signs or aliases seperated by space.

18.4 Examples:

18.4.1 c ax0 n5knx-2   -> connect to n5knx-2 on port ax0

18.4.2 c ax0 n5knx-2 MKG pix kc8ke -> as before thru the digi sequence

19 **convers <subcommands>;** These commands configure the network conference server. These are available if jnos is compiled with #defined [tbd]

19.1 convers channel [<default_chan_number>] Default: 0   Displays or sets the default channel number, that is, the initial channel to which new users are assigned.

19.1.1 convers channel 2

19.2 convers drop [<addr>]   Drop the remote convers link to <addr>. See also 'convers link'. If <addr> is not given, all links are dropped.

19.2.1 convers drop 44.26.1.19

19.3 convers filter   Default: refuse (nobody)   Set how the convers node will respond to connect requests.

19.3.1 convers filter mode [accept | refuse] Sets or displays the filter mode. 'filter mode accept' allows links from only the hosts in the filter list. 'filter mode refuse' allows links from all hosts except those in the list.

19.3.2 convers filter [ipaddress | hostname] Builds the filter list used in conjunction with the 'convers filter mode' command.

19.4 convers host <name>   Displays or set the convers hostname as will be used when announcing the system to conference users or remote links. Maximum length is 10 chars, but if you want to stay compatible with JNOS.EXE based convers servers use a maximum of 8 character for the convers host name (unless the system runs JNOS-v1.04 or later).

19.4.1 If the 'hostname' gets set and the 'convers host' isn't set yet, it will be set to the first 10 chars of the 'hostname'. After this, if any sub domains (i.e. periods) exist in the hostname, the convers hostname will be terminated at the right-most period. e.g. If 'converse host' is not set, and 'hostname jnos.wg7j.ampr.org' is given, then after this the converse hostname will be 'jnos.wg7j'.

19.4.1.1 convers host Corvallis

19.5 convers interface [<iface>] [on|OFF]   Displays or sets the active convers interfaces. This command needs to be given for each interface that which will allow connections to the conference call (see 'convers mycall'); e.g., this command can be used to allow conference call access only on the user ports but not on the backbone/linking ports. This can also be useful to avoid confusion when different nodes have the same conference call. (Locally, we use the call 'QSO' for the conference server for different nodes, and ran into problems when a user tried to connect to it from a backbone node. All of a sudden two nodes were answering the connect !) Default is off.

19.5.1 convers interface port1 on

19.6 convers link [<addr> [port] [name]] OR convers xlink [<addr> [port] [name]] If no <addr> is given, display the list of linked nodes with link status and statistics indicated. If <addr> is given, add a convers link to another (remote) conference server. The link is LZW-compressed if xlink, instead of link, is specified (and XCONVERS was #define'd at compile-time).

19.6.1 <addr> is the ip address or hostname of the remote server to which to link.

19.6.2 [port] is the tcp port number to use for the server connection. [port] defaults to 3600 if the link command is used, and 3601 if the xlink command is used.

19.6.3 [name] is the optional name that will show up in the links listing shown with the '/links' command if the link has not yet been established. [name] can be a maximum of 10 characters; the first character must be a non-integer.

19.6.4 After the link has been established, the name will be set to the name with which the remote system introduced itself.

19.6.5 convers link 44.26.1.19 Testing

19.7 convers [u|h]maxq [<bytes>] Display or set the upper limit for the number of bytes that can be queued up waiting for transmission on a connection to another server. If there is more data than this limit, the connection to the other server will be closed.

19.7.1 You are able to set individual limits for users and hosts with 'convers hmaxq' and 'convers umaxq'. If set to 0, there is no limit, otherwise connections will be reset if there is more than the []maxq value data outstanding on the connection. The connections will be RESET instead of gracefully closed.

19.7.2 Default values are umaxq of 1024 and hmaxq of 5120. Note that any changes will only affect new connections, not existing connections.

19.8 convers maxwait [<seconds>] Default: 10800 Display or set the upper limit for the time the system will wait to reestablish a disconnected convers link that originated at this system. Time is given in seconds.

19.8.1 convers maxwait 600

19.9 convers motd ["<yourmessage>"] Set or show the message of the day for the convers server. This message is displayed when users connect to the server. NOTE: this option is obsolete in recent convers implementations. Instead, the contents of the file /spool/convmotd.txt are displayed. This filename can be changed by setting the ConvMotd string in nos.cfg.

19.10 convers mycall <mycall> Display or set the 'conference call'. <mycall> is a separate ax.25 callsign. If set, users can connect to it to get immediately connected to the conference bridge. However, each port or interface that this call should be allowed on should be enabled with the 'convers interface' command. Conference call connections bypass the regular node interface. This is independent from the settings of 'mbox convers' or whether the network conference server has been started. See also 'convers t4'.

19.10.1 convers mycall QSO

19.11 convers online [long | call | @host] Display a list of convers users known to the convers server. This is the same report as a /who listing made from within the convers facility. The default report is a "quick" format listing of the connected users. The "long" option specified a long-format report, which can be restricted to a particular "call" or "host".

19.11.1 Example: conv on @luzana -or- conv on wu3v -or- conv on

19.12 convers setinfo [yes | NO] Display or the set the ability of conference users to change their personal info as stored in /finger/dbase.dat. This sub-command is only available when CNV_CHG_PERSONAL was #define'd when JNOS was compiled.

19.13 convers t4 [<seconds>] Default: 7200 Display or the set the conference call connection T4 timer. t4 is the 'redundancy timer' for ax.25 connections to the conference server. This allows you to set a different inactivity time-out for ax25 node and conference connections. Default is 7200, i.e. 2 hours.

19.13.1 convers t4 900

19.14 convers tdisc [<seconds>] Default: 0 Display or the set the conference call general redundancy timer that applies to all connections to the conference server. Connections which are idle longer than <seconds> will be disconnected. Default is 0, i.e. disable idle timeouts.

19.14.1 convers tdisc 1200

19.15 NOTE: Converse users often wonder why the /help report is not complete. This is because the default compilation options for convers.c include "noblocking". This results in JNOS dropping data destined for any output queue that exceeds certain length limits, and thus ensures that JNOS will not run low on buffer space due to the inability of a slow RF link to process the output queue fast enough. The limits at which lossage occurs depends on how a user connects to JNOS: telnet users are limited by the tcp window size, ax.25 users are limited by the JNOS 'ax25 window' setting, and the local console is limited by the LOCSFLOW constant (2048). The convers.c source could be edited to #undef noblocking, and recompiled in an attempt to rectify this limitation, but (worse?) instability might result!

20 **delete <filename>;** Deletes the specified file. <filename> may include a complete path. Functions the same as the DOS Delete command.

21 dialer;

21.1 dialer <interface> [<dialer_file> [<seconds> [<pings> [<hostid>]]]] Set up an autodialer session for the interface. Whenever the interface is idle for the interval in <seconds>, the autodialer will ping the <hostid> if specified, or send a link-layer echo request if possible and no ping target was given. If there is no incoming data after <pings> attempts, the autodialer will execute the special commands contained in the <dialer_file>.

21.2 If no <dialer_file> is specified, a previous dialer command process will be removed. If the number of <pings> is omitted, the <dialer_file> will be executed without first pinging the <host>.

21.3 The <dialer_file> may have any valid name, and is located in the JNOS root directory unless a full pathname is provided. The dialer commands in the file are described below.

21.3.1 >> Examples: dialer sl0 ns9tel.dia 30 10 ns9tel

21.4 DIALER FILE COMMANDS

21.4.1 control down | up

21.4.1.1 Control the 'asy' interface. The 'down' option drops DTR (and RTS except in the Unix version). The 'up' option asserts DTR (and RTS except in the Unix version).

21.4.1.1.1 >> Example: control down

21.4.1.2 Actually, other options beside down and up are allowed, provided they are supported by the param command for the dialer's interface.

21.4.1.2.1 >> Example: control dtr 1

21.4.2 send "<string>" [<milliseconds>] This dialer command will write the specified string to the interface. The string quote marks are required, and the string may not contain embedded control characters. However, the standard C string escape sequences are recognized (but \0 should not be used). If <milliseconds> is specified, the <string> characters are sent with a <milliseconds> inter-character delay, useful for ancient Micom switches!

21.4.2.1 >> Example: send "atdt555-1212"

21.4.3 speed [115200|57600|38400|19200|9600|4800|2400|1200|300] This command sets the speed of the interface to one of the available speeds. If the speed argument is missing, the speed will be displayed in the dialer session window.

21.4.3.1 >> Example: speed 1200

21.4.4 wait <milliseconds> ["test_string" [speed|ipaddress]]

21.4.4.1 If only the time is specified, the dialer pauses for the desired number of milliseconds. Otherwise, the dialer reads until the <test_string> is detected on the interface.

21.4.4.2 If the string is not detected within the desired time, the autodialer will reset. The string quote marks are required, and the string may not contain embedded control characters. However, the standard C string escape sequences are recognized (but \0 may not be used).

21.4.4.3 If the "speed" keyword is specified, the dialer will continue to read characters until a non-digit is detected. The string read is converted to an integer, and used to set the interface speed. If the trailing non-digit is not detected within the desired time, or the integer value is not a valid speed, the autodialer will reset.

21.4.4.4 If the "ipaddress" keyword is specified, the dialer will continue to read characters until a dotted-quad IP address is detected. The numeric address is used to set the interface IP address. If a trailing non-digit is not detected within the specified time, or the address is invalid, the autodialer will reset. This option is only available when SLIP was #define'd at compile time, since PPP protocol supports address negotiation.

21.4.4.4.1 >> Example: wait 45000 "CONNECT " speed

21.4.4.4.2 >> Example: wait 5000 "Assigned IP address is" ipaddress

21.4.5 DIALER FILE EXTENDED COMMANDS

21.4.5.1 failmode [on | OFF] 'failmode' establishes whether the dialer should continue after a failed dialer command. <off> implies abort the dialing script, while <on> means continue the script, which in effect enables the 'ifok' and 'iffail' commands.

21.4.5.2 begin 'begin' starts a block of commands, and is typically used after an 'ifok' or 'iffail' command.

21.4.5.3 end 'end' terminates a block of commands, which extends to the previous unpaired 'begin'.

21.4.5.4 exit [<return_code>] 'exit' ends the dialer script, with the result code set to that of the previous dialer command unless <return_code> is specified.

21.4.5.5 status [up | down] 'status' is similar to the 'control' command, except that the iostatus() routine is notified.

21.4.5.6 ifok <cmd> 'ifok' invokes the dialer command <cmd> if the previous command was successful.

21.4.5.7 iffail <cmd> 'iffail' invokes the dialer command <cmd> if the previous command was not successful.

21.4.5.8 verbose [ON | off] 'verbose' sets the verbosity level of the dialer, that is, whether the dialer echoes the script commands as they are read and displays output received during the wait command. The "off" setting is recommended for those well-debugged scripts used with the ping/redial option. The verbose setting is retained across dialer invocations.

21.5 DIALER FILE EXAMPLE > Found in the Appendix

22 **domain <subcommand>;** The domain commands control and show the working of the name to Internet address mapping software (referred to as DNS: Domain Name Service). JNOS has both a DNS client and a server. The server will answer queries from data in the domain cache, and from information stored in the DOMAIN.TXT file. The DNS server is only available when DOMAINSERVER is #define'd in config.h, but the DNS client is always available.

22.1 domain addserver <hostid> [<timeout>] Add a domain name server to the list of name servers. <timeout> is an optional timeout setting in seconds for this server. If <timeout> is not included in the command, the value defaults to 3 * (tcp_irtt). Servers are queried in the opposite order in which they were added. Examples:

22.1.1 domain addserver wg7j.ece.orst.edu 30

22.1.2 domain addserver 128.193.48.1

22.1.3 domain addserver ucsd.edu

22.2 domain cache <subcommand> Following commands work on the domain cache. These are resource records held in memory. (described in RFC1033/1034)

22.3 domain cache clean [<yes | NO>] Displays or sets the discard of expired resource records. expired records have their time-out value decremented to zero. Normally resource records get a default time-out value of 1800 seconds. After this time they are considered "old" and if referenced again the domain name resolver should be inquired again.

22.3.1 When clean is off (the default), expired records will be retained; if no replacement can be obtained from another domain name server, these records will continue to be used.

22.3.2 When clean is on, expired records will be removed from the file whenever any new record is added to the file.

22.3.3 Example: domain cache clean yes

22.4 domain cache dump   Immediately clears the domain cache. This amounts to being a "flush".

22.5 domain cache list   This command shows the current content of the in-memory cache of resource records.

22.6 domain cache size [<size>] Display or set the maximum size of the local in-memory domain cache. Default is 5. Example:

22.6.1 domain cache size 10

22.7 domain cache wait [<secs>]   Default: 300   Display or set the number of seconds which must elapse before the domain.txt file is updated from the resource records stored in the domain cache. Upldating is controlled by the 'domain update' command (see also).

22.8 domain dns [on|off] Display or toggle the state of the Domain Name Server. If on, the system is active as a Domain Name Server. The system will then answer queries from other tcp/ip hosts regarding hostname to ip-address, and ip-address to hostname translations. The dns subcommand is only available when DOMAINSERVER was #define'd when JNOS was compiled.

22.9 domain dropserver <hostid> Remove a domain name server from the list of name servers. You are warned when you delete the last name server.

22.9.1 domain dropserver ece.orst.edu

22.10 domain listservers List the currently configured domain name servers, along with statistics on how many queries and replies have been exchanged with each one, response times, etc.

22.11 domain look <search_text> This command searches domain.txt and displays records matching <search_text>. The suppoed <search_text> must match exactly, i.e., case is significant. This subcommand is available only if MORESESSION was #define'd when JNOS was compiled.

22.12 domain maxclients [<N>];   Command to limit loading, N=number of clients, while acting as a Domain Name Server. Default is 6.

22.13 domain maxwait [<time-out>] This sets a time-out value (1 to 255 seconds) to a query or domain name server. This is not set for an already defined server but will be used for a newly defined name server. Also the value is used for domain name lookups (E.g. when a user does a telnet to a host with the mailbox'T host' command). Note that (PC based) name servers can have trouble finding records in a large database. Default is 60 seconds.

22.13.1 domain maxwait 10

22.14 domain query [<hostid>] Displays the results of a DNS query for <hostid>.

22.15 domain retries [<retries>] Default is 2. The retry value (number) limits the number of queries sent out to remote domain name resolvers before giving up and telling you that host xyzzy.ampr.org does not exist. The total time lost with a query is (retries * time-out * number of domain servers defined); i.e., the delay between requesting a hostname to ip-address translation and getting the answer can become very long if you use many servers, and set the timeouts/retries high !

22.15.1 domain retries 1

22.16 domain subnet [ON | off] This command works in conjunction with 'domain translate' to allow or disallow translation of any address ending in 0 or 255. On systems which have a lot of subnets, turning off subnet translation can result in a considerable speedup when displaying routes with 'domain translate on'.

22.17 domain suffix [<domain suffix> | none] Display or specify the default domain name suffix to be appended to a host name when it contains no periods. For example, if the suffix is set to "ampr.org." and the user enters 'telnet ka9q', the domain resolver will attempt to find 'ka9q.ampr.org.' If the host name being sought contains one or more periods, however, the default suffix is NOT applied if the last part of the name is less than 5 characters and contains only letters; e.g., 'telnet foo.bar' would NOT be turned into 'foo.bar.ampr.org.' 'telnet foo.ka9q' will be turned into 'foo.ka9q.ampr.org.' Note that a trailing dot (.) is required for the suffix. If the suffix is the string 'none' (without trailing period), the current suffix is cleared and forgotten. Default is "ampr.org."

22.17.1 domain suffix ece.orst.edu.

22.18 domain trace [on| OFF] Display or set the flag controlling the tracing of domain server requests and responses. This only works when console is enabled. Default is off.

22.18.1 domain trace on

22.19 domain translate [on | OFF] Display or set the flag that controls the translation of ip addresses in dot notation into symbolic names. The translation process makes heavy use of reverse domain name lookups. Do not set this flag unless you have a good and fast connection to a domain name server.

22.19.1 domain translate on

22.20 domain ttl [ttl] Select a default 'ttl' value to be applied to server responses than contain none.

22.21 domain update [on | off] Controls whether or not domain.txt file is updated with server responses.

22.22 domain verbose [on | off] Display or set the flag controlling the return of a full name (on) or only the first name (dot delimiter) (off). This is for IP address to name translation only. If off, home.wg7j.ampr.org. will show as 'home.wg7j', whereas if on it will show as 'home.wg7j.ampr.org'

22.22.1 domain verbose off

23 dump <hexaddress | .> [range]; The dump command shows memory in hex and ascii. Hex-address is a 32-bit value split into page address and page offset. A splitting colon is not used nor accepted. If decimal-range is not given , 128 bytes are displayed. 'dump .' displays memory starting at the end of a previous dump

command. This command is primarily useful for debugging.

23.1 dump 0x12fe0008

24 echo [accept|refuse]; Default: accept Display or set the flag controlling client Telnet's response to a remote WILL ECHO offer.

24.1 The Telnet presentation protocol specifies that in the absence of a negotiated agreement to the contrary, neither end echoes data received from the other. In this mode, a Telnet client session echoes keyboard input locally and nothing is actually sent until a CR is typed.

24.2 Local line editing is also performed: backspace deletes the last character typed, while control-U deletes the entire line.

24.3 When communicating from keyboard to keyboard the standard local echo mode is used, so the setting of this parameter has no effect. However, many timesharing systems (e.g. UNIX) prefer to do their own echoing of typed input. (This makes screen editors work right, among other things). Such systems send a Telnet WILL ECHO offer immediately upon receiving an incoming Telnet connection request.

24.4 If 'echo accept' is in effect, a client Telnet session will automatically return a DO ECHO response. In this mode, local echoing and editing is turned off and each key stroke is sent immediately (subject to the Nagle tinygram algorithm in TCP).

24.5 While this mode is just fine across an Ethernet, it is clearly inefficient and painful across slow paths like packet radio channels. Specifying 'echo refuse' causes an incoming WILL ECHO offer to be answered with a DONT ECHO; the client Telnet session remains in the local echo mode. Sessions already in the remote echo mode are unaffected. (Note: Berkeley Unix has a bug in that it will still echo input even after the client has refused the WILL ECHO offer. To get around this problem, enter the 'stty - echo' command to the shell once you have logged in).

25 edit [<filename>] ; An ascii text editor is included in the JNOS distribution. A clone of the UNIX "ed" editor, is invoked from the console with the path to the file to be edited:

25.1 edit c:/ftpusers

25.2 This option is enabled by compiling with EDITOR and ED #define'd.

25.3 A full-screen console-only editor is available if JNOS is compiled with both EDITOR and TED #define'd.

26 eol [standard | null]; Default: standard Display or set Telnet's end-of-line behavior when in remote echo mode. In 'standard' mode, each key is sent as is. In 'null' mode, a NUL character is sent immediately after sending a CR.

26.1 This command is not necessary with all UNIX systems; use it only when you find that a particular system requires two CRs to end a line. In any case, the eol setting is only pertinent when in remote- echo mode, since otherwise a CR is translated to LF by JNOS' tty driver.

27 errors [ON | off]; Set whether the system will send messages about system errors and permission infringements to user 'sysop'. Default is on.

28 escape [<literal_char>]; Default: 'CTRL-]' Display or set the current command-

mode escape character. The character has to be entered as a literal character. The escape character returns control to the "jnos>" prompt, and functions the same way as F10.

29 etelnet <host> [<port_number>] loginid password; The 'etelnet' command is similar to the telnet command (which see), but it accepts the login id and password to be provided to the <host> system. The etelnet command is available when JNOS was compiled with MD5AUTHENTICATE #define'd. Etelnet will encrypt the password if the <host> system supports MD5 authentication (as indicated by a [hex_number] challenge value in its password prompt). If no such challenge is provided, the password is sent "in the clear".

29.1 If you are telnetting to a port/service that does not provide a password prompt, then 'telnet' should be used instead of 'etelnet'.

30 ettylink <host> [<port_number>] loginid password; The 'ettylink' command is similar to the ttylink command ,but it accepts the login id and password to be provided to the <host>system should it request a login id and password. The default port, 87, on a JNOS system does NOT request these, so using ettylink to this port/application is not recommended. The ettylink command is available when JNOS was compiled with MD5AUTHENTICATE #define'd. Ettylink will encrypt the password if the <host> system supports MD5 authentication (as indicated by a [hex_number] challenge value in its password prompt). If no such challenge is provided, the password is sent "in the clear".

31 exit [return_code]; Default: 0 Causes the JNOS program to terminate when at the JNOS> prompt. When shelled to DOS, causes a return to the JNOS> prompt. When terminating the program, an "Are you sure?" query is given. Enter "y(es) <cr>" to end the program. Any other response returns to JNOS.

31.1 If a <return_code> numeric value is provided, this value is returned to the caller of JNOS. This is typically a batch file, which may then take action depending on this code. For example:

```
31.1.1 :rerun
31.1.2 jnos110i  -u1 -g2
31.1.3 if errorlevel 100   goto remote_exit
31.1.4 if errorlevel 99    reboot
31.1.5 if errorlevel 1     goto rerun
31.1.6 goto exit
31.1.7 :remote_exit
31.1.8 :exit
```

31.2 In this example, issuing "exit 99" would run the reboot command, "exit 1" would restart JNOS, and "exit" or "exit 0" would exit the batch file that invoked JNOS. The remote command (c.v.) when given the exit parameter, will use a return code of 100.

32 expire <subcommand>; The 'expire' command is used to invoke the process which deletes old BBS messages. The mailboxes or newsgroups to be expired, and the age (in days) after which a message may be deleted, are specified in the Expirefile (defaults to /spool/expire.dat).

32.1 expire [now | interval] Begin the expire process immediately, if <now> is specified, or every <interval> hours in the future (but no immediate expire is done). If no arguments are provided, the current expire interval is displayed.

32.2 Expire may also be scheduled at startup. Example: To run expire at 04:30 each morning, place these commands into your autoexec.nos file

32.2.1 at 0430 "expire 24"

32.2.2 at 0431 "expire now"

32.3 If the number of days is omitted, 21 is used. Also, if newsgroups are specified, the associated History file is expired as well, but the number of days used is the maximum of all the days provided for newsgroups in the Expirefile. Example:

32.3.1 allusa 7

32.3.2 amsat 10

32.3.3 !swap 10

32.3.4 !rec.radio 7

32.4 See Also: oldbid

33 finger <username[@host]> [<username[@host]> ...]; Issue a network 'finger' request for <username> at <host>. Finger is typically used to find out specific information about users on local or remote hosts. As our network expands, this application will help hams find out information about each other quickly and efficiently. By fingering a user, you can find out such information as a user's name, his mailing address, telephone number, QSL information, and other useful facts. This information is kept in a separate text file for each user.

33.1 The finger command under NOS can be issued in any of the following three ways:

33.1.1 finger <username> The first form of the command is used to find out information about a user at the local host, namely your own system. It is useful for testing 'finger' on a system that you know is running. >> Examples: finger n8fow

33.1.2 finger <username>@<host> The second form of the command is used to find out information about a user at a remote host. >> Example: finger n8fow@n8fow

33.1.3 finger @<host> If you don't know the name of a particular user at a remote host, you can use the third form of the command. This command returns a list of all 'finger' files on the remote system. >> Example: finger @n8fow

33.2 To enable the finger server so that others may query the users on your system, you must give the 'start finger' command. The finger files that provide information on a <username> are located by default in \finger (see Fdir and Fdbase in nos.cfg), and are ordinary ASCII files created by the sysop. Also, if the SAM or QRZ callbook server is configured, <username> is looked up in the callbook and displayed if the search is successful.

33.3 Certain <username> strings are taken to mean that a JNOS function should be invoked to display system information, depending on what configuration options were used to build the server JNOS:

<username>	config_opt	output_same_as
conf	CONVERS	conference bridge /WHO
links	CONVERS	conference bridge /LINKS
mbxinfo	MAILBOX	'I cmd in mailbox'
mhold	HOLD_LOCAL_MSGS	'mbox holdlocal'
mstat	MAILBOX	'mbox mailstat'
mpast	MAILBOX	'mbox past'
users	MAILBOX	'mbox status'
usersdat	USERLOG	'finger x' forall users in users.dat
mailfor	MAILFOR	'mbox mailfor'
info	ALLCMD	'info'
ax25	AX25	'ax25 stat'
aheard	AX25	'ax25 heard'
netrom	NETROM	'netrom stat'
iheard	all	'ip heard'
memstat	all	'mem stat'
socket	all	'socket'
tcpview	all	'tcp view bytes'
asystat	ASY	'asystat'
pkstat	PACKET	'pkstat'
rip	RIP	'rip stat'

34 fkey; The 'fkey' command allows you to program the function keys and several other cursor control keys.

34.1 fkey This command produces a listing of the currently defined function keys.

34.2 fkey <key_number> [<value> | "<string>"] Display or define a new setting for a function key.

34.3 Control characters can be included in the string by prefixing with the ^ character (SHIFT 6 on most keyboards); e.g. CR is entered as ^M. To insert a ^ in the string, enter ^^.

34.4 Note: If the first character of a function key definition is '~' then JNOS switches to the Command session and processes the rest of the definition (if any).

34.5 >> Examples:

34.5.1 fkey 87 "trace tnc0 211^M" (SHIFT-F4 turns trace on)

34.5.2 fkey 72 "" (disable up arrow)

34.5.3 fkey 113 "~" (Alt-F10 switches to the Command session)

34.6 FKEY TABLE Pressing the selected key produces the accompanying number. Double keys are S=shift, C=ctrl, A=alt.

key	***num***	***key***	***num***	***key***	***num***	***key***	***num***	***key***	***num***
F1	59	Stab	15	SF1	84	CF1	94	AF1	104
F2	60			SF2	85	CF2	95	AF2	105
F3	61	home	71	SF3	86	CF3	96	AF3	106
F4	62	up	72	SF4	87	CF4	97	AF4	107
F5	63	pgup	73	SF5	88	CF5	98	AF5	108
F6	64	left	75	SF6	89	CF6	99	AF6	109
F7	65	right	77	SF7	90	CF7	100	AF7	110
F8	66	end	79	SF8	91	CF8	101	AF8	111
F9	67	down	80	SF9	92	CF9	102	AF9	112
F10	68	pgdn	81	SF10	93	CF10	103	AF10	113
F11	133	ins	82	SF11	135	CF11	137	AF11	139
F12	134	del	83	SF12	136	CF12	138	AF12	140
Cprnt	114	Cpgdn	118	Cend	117	Chome	119	Cpgup	132

35 ftp <hostname> [<scriptfile>]; The ftp command is used to make a TCP connection with <hostname>, and then use File Transfer Protocol to exchange data between the systems. Once the connection is established, a small set of commands is used to manage the file exchange. A command is executed locally if it is a local client command. Otherwise, the command is sent to <hostname> for execution. If <scriptfile> is provided, all commands are obtained from the indicated file; otherwise, they are read from the console. The following are commands supported by JNOS clients and servers:

35.1 ? Display all available command names.

35.2 ascii Treat the data to be transferred as ASCII text, so that line endings are used suitable to the receiving system.

35.3 batch [y|n] Query the state of the command batching flag, or set it if <y|n> is given. Batching involves sending as many commands as possible before waiting for responses from <hostname>.

35.4 binary Treat the data to be transferred as binary data, that is, verbatim data not to be changed while storing on the receiving system.

35.5 cd path Change to directory <path> on system <hostname>.

35.6 cdup Change to immediately-superior directory on system <hostname>.

35.7 dele file Delete <file> on the remote system.

35.8 dir spec List the contents of the current directory on the remote system, in a verbose manner. If <spec> is given, the subset that matches this file specification is listed.

35.8.1 Example: dir *.exe

35.9 get file   Transfer <file> from remote system TO the local system.

35.10 hash [y|n] Query the state of the hashmark flag, or set it if <y|n> is given. Hashmarks are written to the screen for each 1000 bytes written to the local file system.

35.11 help   Same as ?

35.12 lcd path   Change to directory <path> on the local system.

35.13 ldir spec Same as the dir command, but applied to the local system.

35.14 listSame as dir command.

35.15 lmkdir dir Create directory <dir> on the local system.

35.16 ls spec   List just the names in the current directory on the remote system. If <spec> is given, the subset that matches this file specification is listed.

35.16.1 Example: ls *.exe

35.17 mdtm file Display the modified-time in GMT for <file> as yyyymmddhhmmss.

35.18 mget spec Transfer all files matching <spec> from the remote system TO the local system.

35.19 mkdir dir Create directory <dir> on the remote system.

35.20 mput spec Transfer all files matching <spec> to the remote system FROM the local system.

35.21 nlst   Same as ls command.

35.22 put file   Transfer <file> to the remote system FROM the local system.

35.23 quit   Close the TCP connection to <hostname> and exit the ftp cmd.   See also the JNOS "abort" command.

35.24 reclzw [y|n] Query the state of the ftp client LZW-supported flag, or set it if <y|n> is supplied. LZW compression is only supported for ASCII-type transfers.

35.25 rename from to Rename the file named <from> to the name <to> on the remote system.

35.26 reget file The RFC959-endorsed way to restart an interrupted get. No check is made to assure the file is consistent between systems; the size of the local <file> is provided to the remote system and a get is issued relative to this point.

35.27 restart pos The RFC959-endorsed way to restart a transfer is to first establish a starting offset position and then issue a transfer command to resume at that position. The dir command would be useful to obtain the offset to specify in the restart command, and then a put <file> would cause <file> to be sent starting from the given position.

35.28 resume file Restart an interrupted transfer of <file> from the remote system to the local system. Checks are made to assure the file is consistent between systems. This is a JNOS extension to the FTP standard. Other non-compatible equivalents exist in other implementations (c.f. wu-ftpd).

35.29 rmdir dir Delete (remove) directory <dir> on the remote system.

35.30 rput file Resume an interrupted transfer of <file> to the remote system FROM the local system. Checks are made to assure the file is consistent between systems. This is a JNOS extension to the FTP standard. Other non-compatible equivalents exist in other implementations (c.f. wu-ftpd).

35.31 sendlzw [y|n] Query the state of the ftp server LZW-supported flag, or set it if <y|n> is supplied. LZW compression is only supported for ASCII-type transfers.

35.32 type [a|b|l] Query the current transfer type value, or set it if <a|b|l> is given. Use <ascii>, <binary> (or <image>). <logical 8> is also supported.

35.33 view file Transfer <file> from the remote system TO the local system's console screen. The file is assumed to be an ASCII file!

35.34 verbose [n] Query verbosity of error handler, or set it if integer <n> is given. 0 => error msgs only, 1 => final msg only, 2 => control msgs too, 3 => control msgs + hash marks, 4 => control msgs + byte counts.

35.35 Since unrecognized commands are sent to the remote ftp server for tion, additional commands may be available, depending upon the ftp server implementation. For example, the WU-FTPD may accept site-written extensions, and thus allow: site exec <extended_cmd> <cmd_args>.

35.36 The remote ftp server will require a login name and password. These values may be provided by a file called "net.rc" by default (see Hostfile in nos.cfg). The file entry for password may be omitted to instead have the client ftp prompt for it.

35.37 Many systems, including JNOS, will reduce the amount of extraneous messages sent, if the password is prepended with a '-'. However, JNOS does not support this when an anonymous login occurs via an MD5 authentication exchange.

36 gate <subcommands>:   The gate command manages the tcpgate server, which accepts connects to certain tcp ports, and logically redirects them to another host and/or tcp port. Thus it is only one half of a proxy server! This command requires JNOS to be compiled with TCPGATE #define'd.

36.1 When invoked without arguements, gate displays the current state of any active connections through the tcpgate code. On the left is the originating address and port, on the right is the destination address and port, as determined by the 'start gate...' command, and in the center is the port that was used by tcpgate to trap the connection.

36.2 gate status   Displays the manner in which incoming connections are mapped to what outgoing ones. It also displays the number of client connections allowed and the inactivity timeout.

36.3 gate maxcli   Determines how many clients may connect through the tcpgate at any one time.

36.4 gate tdisc <seconds>   Determines the inactivity timeout (in seconds) as measured by data SENT to the client. A value of zero disables this feature.

36.5 start gate port hostname [port]   This is used to start up the tcpgate server on a particular port. The server listens for connection on that port and accepts them. It then makes a connection to the nominated hostname (with optionally a new port number - otherwise the same port number as is listened for is used). The 'gate status' command may be used to determine what ports have already been configured. This command generates a failure if that port number is already in use.

37 **help;** The 'h' command provides help for the help command. The JNOS '?' command provides a list of all the top-level JNOS commands.

37.1 To obtain more help on a particular command, type 'help' followed by the command name. Or type the command name followed by a question mark. Examples:

37.1.1 help mbox displays the help file for the mbox command

37.1.2 mbox ?   displays the subcommands for the mbox command

38 **hfdd server <iface> [start | stop]** control the hf RF server for inbound tcp/ip connections.

38.1 <iface> is a defined interface name - tested with ptcpro and dxp38 hw_devices.

38.2 start/stop instantiates and terminates server service >>>per jnos2.0d1 the stop command does not [tbd] function (exit and restart jnos is the only known "stop" method) and does not give an error msg.

39 **history [<N>];** Default: 10 The history command displays the number and contents of previous commands retained in a circular list, or sets the depth <N> of the history list. These commands can be recalled by the UP-arrow and DOWN-arrow keys. Press return to execute a recalled command, or backspace to erase from the right side. No further editing is currently supported. Setting the list size to 0 (zero) disables this feature.

40 **hop;** The 'hop' commands are used to test the connectivity of the network.

40.1 hop check [-n] <host> Initiate a hop check session to the specified host. This uses a series of UDP "probe" packets with increasing IP time-to-live (TTL) fields to determine the sequence of gateways in the path to the specified destination. This function is patterned after the UNIX 'traceroute' facility. If the -n option is used, no DNS lookups are done, so no names are listed for each hop.

40.1.1 ICMP message tracing should be turned off before this command is executed (see the 'icmp trace' command).

40.2 hop maxttl [<hops>] Default: 30 Display or set the maximum TTL value to be used in hop check sessions. This effectively limits the radius of the search.

40.3 hop maxwait [<seconds>] Default: 5 Display or set the maximum interval that a hop check session will wait for responses at each stage of the trace.

40.4 hop queries [<count>] Default: 3 Display or set the number of UDP probes that will be sent at each stage of the trace.

40.5 hop trace [on | OFF] Default: off Display or set the flag that controls the display of additional information during a hop check session.

41 **hostname [<name>];** Display or set the local host's name. By convention this should be the same as the host's primary domain name. This string is used only in the greeting messages of the various network servers. NOTE: it does NOT set the system's IP address.

41.1 If <name> is the same as an <interface> defined in an 'attach' command, this command will search for a CNAME domain resource record which corresponds to the IP address of the <interface>. This is commonly used for to set your hostname to that of a dynamically assigned IP address dial-in line.

41.2 Example: hostname crv.kuyx.ampr.org.

42 **http <subcommand>;** Controls the operation of the JNOS HTTP server.

42.1 http absinclude [on | OFF] Set or show the value of the flag that determines if the html specification 'include file="path"' is acceptable. If absinclude is off, the include command results in an error reported to the client. If on, the file at "path" is inserted into the html document being prepared for transmittal to the client.

42.2 http always [ON | off] This displays or sets the always-send flag. If on, html files will always be sent regardless of their modification time. If you use SSIs in your html files, then the file modification time of the html file is not indicative of the content change, and setting always to ON may prove useful.

42.3 http dontlog [str1|str2|...|strn] This displays or sets a list of strings which, if contained in a URL, will prevent detailed logging of that access. This command is available when #define'd HTTP_EXTLOG at compile time.

42.3.1 Example: http dontlog junkdir/|.gif|.jpg

42.4 http maxcli [<number>] Default: 10 Display or set the maximum number of http client connections allowed. When this limit is reached, or if available memory drops below 'mem threshold', new connections are immediately refused. See also "http simult".

42.5 http multihomed [on | OFF] Display or set the flag which allows the http server to report the name associated with the interface used to access it. If off, the system hostname is used.

42.6 http simult [<number>] Default: 5 Display or set the number of simultaneous active http client connections allowed. When this limit is reached, new connections will immediately block until an older client connection terminates. See also "http maxcli".

42.7 http status Display information about the http server(s).

42.8 http tdisc [<#secs>] Default: 180 Display or set the number of seconds of idle time allowed a client before it is disconnected for inactivity.

43 **icmp <subcommands>;** These commands are used for the Internet Control Message Protocol service.

43.1 icmp echo [ON | off] Display or set the flag controlling the asynchronous display of ICMP Echo Reply packets. This flag must be on for pings to work. Default is on.

43.2 icmp quench [ON | off] With 'icmp quench off', when a packet is received and memory available < threshold, the packet will be dropped (i.e., no quench or anything.) The higher protocol layers will keep track of re-transmitting the dropped packets.

43.2.1 With 'icmp quench on', when packets are received and the high water mark for dynamically allocatable storage has been exceeded, JNOS submits an ICMP Source Quench to the originator. Usually, before the originator will have reacted to the source quench, JNOS's dynamically allocatable storage will have been exhausted. What happens after that is uncertain, but it is assumed to be unfavorable. Many tcp/ip implementations don't even respond to Source Quenches at all. See also 'memory threshold command.' Default is ON.

43.3 icmp status Display statistics about the Internet Control Message Protocol (ICMP), including the number of ICMP messages of each type sent or received.

43.4 icmp timeexceed [<ON | off>] Allows 'time exceeded' message to be sent when the ttl of an ip packet to be routed becomes zero. When turned OFF, no message is sent which allows the system to become invisible for 'traceroutes', etc.

43.5 icmp trace [0 | 1 | 2] Display or set the flag controlling the display of ICMP error messages. These informational messages are generated by Internet routers in response to routing, protocol or congestion problems. This only functions when in console mode. Default is 0 (off). A trace value of 1 traces all icmp packets, while a value of 2 traces only icmp types known to JNOS.

44 **ifconfig [<subcommand>];** When no subcommand is given, display a list of interfaces, with a short status for each. See the 'ifconfig <iface>' command for a description of the display. If a valid subcommand is given, it will be executed (see below).

44.1 ENGINEERING CHANGE NOTE: Useage of ifconfig, ax25, and tcp, has changed for JNOS V1.1 per the following rules:

44.1.1 ALL ax25 and MOST tcp parameters are now configurable per interface. The 'ax25 <cmd>' commands set the system default values and the 'ifconfig <iface> ax25 <command>' commands set or show the interface specific value(s). The 'tcp <cmd>' commands work in the same manner.

44.1.2 As a result of this change, 'ifconfig' NO LONGER takes multiple commands on one line. 'ifconfig ln0 netmask ffffff00 broadcast 255.255.255.255' is invalid. The command line must be separated into two commands as: 'ifconfig ln0 netmask ffffff00' and 'ifconfig ln0 broadcast 255.255.255.255'

44.2 ifconfig <iface> [<subcommands>] When only iface is given, the interface status is displayed. Interface status shows:

44.2.1 IP addr - the ip address assigned to this interface

44.2.2 MTU - the maximum transmission unit for this interface.

44.2.3 Link encap - the type of link protocol to send packets with over this interface (AX.25, NETROM etc.)

44.2.4 Paclen - if the interface is an AX.25 interface, this is the Paclen used for connections on this interface

44.2.5 flags - interface flags, the sum of all the options set with the various commands. Interface flag values are the sums of the following options, and can be set or unset (i.e. toggled) with the following commands (See their individual descriptions for more)

value	*ID*	*command*	*description of flag*
0x1	DATAGRAM_MODE	mode iface	Send datagrams in raw link frames
0x2	CONNECT_MODE		Send datagrams in connected mode
0x4	IS_NR_IFACE	netrom interface	Activated for netrom use
0x8	NR_VERBOSE		broadcast routes verbose
0x10	IS_CONV_IFACE	convers interface	Activated for conference call access
0x20	AX25_BEACON	ax25 bport	Broadcast AX.25 beacons
0x40	HIDE_PORT	mbox hide	Don't show port in mbox 'P' comman
0x80	AX25_DIGI	ax25 digi	Allow digipeating
0x100	ARP_EAVESDROP	arp eaves	Listen to ARP replies
0x200	ARP_KEEPALIVE	arp poll	Keep arp entries alive after time-out
0x400	LOG_AXHEARD	ax25 hport	Do ax.25 heard logging for interface
0x800	LOG_IPHEARD	ip hport	Do IP heard logging on this interface
0x1000	NO_AX25	mbox noax25	No ax.25 mbox for interface
0x2000	BBS_ONLY	mbox bbsonly	BBSes only on this iface
0x4000	USERS_ONLY	mbox usersonly	Users only on this iface
0x8000	SYSOP_ONLY	mbox sysoponly	Sysops only on this iface

44.2.6 netmask - the ip network mask. See elsewhere for a discussion.

44.2.7 broadcast - the ip broadcast address on this interface. Used when doing arp, etc.

44.2.8 sent ip - the number of ip packets sent on the interface

44.2.9 sent tot- the total number of packets sent (i.e. ip, ax.25, etc.)

44.2.10 sent idle - the elapsed time this interface hasn't transmitted any data.

44.2.11 recv ip - the number of ip packets received on the interface

44.2.12 recv tot- the total number of packets received (i.e. ip, ax.25, etc.)

44.2.13 recv idle- the elapsed time this interface hasn't received any data.

44.2.14 descr - a description of the interface

44.3 ifconfig <iface> ax25 [<subcommand> [arguments]] Displays and/or sets the value for 'subcommand'. Most subcommands display or reset values for the specific interface that were set previously. See the "ax25" for descriptions and default values.

44.3.1 'ifconfig <iface> ax25 ?' displays the valid list of accepted subcommands.

44.3.2 ifconfig <iface> ax25 cdigi <call> Set the 'crossband digipeater only' callsign. If this call is set, digipeating works independently from the 'ax25 digipeat' setting. Connections cannot be made to the cdigi call. JNOS will search for another ax.25 interface with the same Cdigi call as that of the interface receiving the digipeated packet, and transmit the digipeated packet via that interface. >> Example that follows would allow someone to use digipeater call "2x70" to automatically transfer between the two interfaces. Note in this example you would probably wish to use beacons to ID the interfaces periodically, since the transmissions don't contain your callsign. This might argue for using a valid call-ssid as the cdigi value!:

44.3.2.1 ifconfig 2m ax25 cdigi 2x70

44.3.2.2 ifconfig 70cm ax25 cdigi 2x70

44.3.3 ifconfig <iface> ax25 paclen <num> Set the AX.25 paclen for this interface. This is useful if you want to use a value different from the default as set with the 'ax25 paclen' command; e.g., if you have a port with an HF link, you might want to set it to 128. You can also set it to greater than 256 if you have a high speed port. (This command only works for interfaces that can carry AX.25 connections, i.e., it is not for SLIP interfaces, etc.)

44.3.3.1 NOTE1: The AX.25 V2 specification specifies a MAXIMUM of 256 for paclen. If you have a paclen > 256, you may run into problems when interfacing to other non-NOS systems (in particular G8BPQ- based systems.)

44.3.3.2 NOTE2: The value of paclen influences NETROM behavior if the interface is activated for netrom with the 'netrom interface' command! If the paclen for this interface is smaller than any other (netrom active) paclen, the netrom mtu value will be set to this paclen - 20 ! This is to assure that you will not get fragmentation at the ax.25 level when trying to send large data packets over netrom connections. AX.25 V2.1 fragmentation is presently handled only by NOS and derived code as far as is known. Other systems, such as TheNet, BPQ, MSYS, etc., may not include proper handling of V2.1 fragmentation.

44.3.3.3 What the preceding means is, if you have a VHF port with paclen 256, and an Hf port with paclen 128, and BOTH are active with netrom, the netrom mtu will be 108 !

44.4 ifconfig <iface> broadcast <addr> Set the ip broadcast address of interface <iface> to <addr>.

44.5 ifconfig <iface> description "descr" This command sets the interface description to the string specified. If no descr is supplied (i.e. ""), the current description will be cleared. The description is displayed with the mailboxP command (if the interface wasn't hidden from that display). It is also shown in the ifconfig command.

44.6 ifconfig <iface> encapsulation <mode> Sets the encapsulation for interface iface to slip or ax25. This should never be needed, since it is automatically executed when interfaces are attached.

44.7 ifconfig <iface> forward <iface-2> When a forward is defined, all output for interface <iface> is redirected to <iface-2>. To remove the forward, set <iface-2> to <iface>.

44.8 ifconfig <iface> ipaddress <addr> Set the IP address to <addr> for this interface. Normally the ip address is assigned from the system ip address when the interface is first attached. However, it might be necessary to change it when a system acts as a ip-gateway.

44.9 ifconfig <iface> linkaddress <linkaddr>Set the hardware dependent address for this interface. For AX.25 this is the callsign. If you want to allow cross band digipeating, give each port a different ax.25 call with this command.

44.10 ifconfig <iface> mtu <num> Set the maximum transfer unit to <num> bytes.

44.11 ifconfig <iface> netmask <address>Set the sub-net mask for <iface>. The <address> takes the form of an IP address with 1's in the network and subnet parts of the address, and 0's in the host part of the address. Sample: 'ifconfig ec0 netmask 0xffffff00' for a class C network (24 bits). This is related to the 'broadcast' subcommand. See also the 'route' command.

44.12 ifconfig <iface> rxecho <iface-2> When a rxecho interface is defined, all input from interface <iface> is also copied (echoed) to <iface-2>. To remove rxecho, set <iface-2> to "off". This feature requires #define'd RXECHO at compile time.

44.13 ifconfig <iface> tcp [<command>] Sets or displays <command> TCP parameter for <iface>. System default parameters are set at compile time, or are reset PRIOR TO attaching interfaces using 'tcp subcommand'. After attaching an interface, use the 'ifconfig <iface> tcp' commands to set interface parameters.

44.13.1 OUTGOING tcp connections get the values for the interface on which the initial sync packet ('connect request') is routed out.

44.13.2 INCOMING tcp connections get the values for the interface the initial request arrives on.

44.13.3 'ifconfig <iface> tcp ?' displays the valid list of parameters for access: blimit, irtt, maxwait, mss, retries, syndata, timertype, window

45 **index [<areaname>];** Causes the mailindex program to be run and re-establish the ndexes for the mailbox. 'index *' indexes ALL mailbox files.

46 **info;** The 'info' command displays information about the NOS package.

47 **ip <subcommand>;** These commands are used for the Internet Protocol service.

47.1 ip access <permit|deny|delete> <proto> <sourceaddr[/bits]|all> <destaddr[/bits]|all> <iface> [loport | all [hiport]]Display or set ip access controls. The ip access command controls packet routing via the specified <iface> by determining which source ip addresses <sourceaddr> are routed to which destination ip addresses <destaddr>. If no ip access commands are issued for <iface>, the default behavior is to permit all sources to access all destinations. But once an IP access command is entered for <iface>, all routes via <iface> that are not specifically permitted by an ip access command, will be denied.

47.1.1 Execution of this subcommand will add or delete an access control entry in an internal table. Incoming packets that would be routed via <iface> are compared with the table entries for <iface>, in the order that they were added, to determine if access will be granted (and routing take place). Access will be granted only if an entry matching <destaddr> and <sourceaddr> is found with "permit" set before either a match with "deny" set is found, or the end of the table is reached. The optional /bits suffix to the ipaddr specifies how many leading bits in the ipaddr are to be considered significant in the routing comparisons. If not specified, 32 bits (i.e., full significance) is assumed. All addresses can be specified by "all". Access can be made protocol dependent via the <proto> parameter. <proto> may be 'a' for any, 't' for TCP, 'u' for UDP, 'i' for ICMP, or the IP protocol number. For UDP and TCP protocols, loport and hiport specify the port or range of TCP or UDP ports for which the access control command applies. If none or all is specified, all ports are assumed.

47.1.2 "ip access" will display the table of current access control entries.

47.1.3 Access commands should be entered from the most specific to the least specific, since the first match (permit or deny) encountered for a given interface in the internal table is definitive.

47.1.4 #Example: allow a specific AMPRnet host access to the internet ip access permit any 44.76.1.199 all eth0 but deny all others except UDP (eg, DNS) access ip access permit udp 44/8 all eth0 all permit only AMPRnet hosts access to RF port

47.1.4.1 ip access permit any 44/8 44/8 2m

47.2 ip address [<addr>] Display or set the default local IP address. This command must be given before an 'attach' command if it is to be used as the default IP address for the interface.

47.3 ip encap [4 | 94] Display or set the packet ID code used for transmitted IP-IP encapsulated packets. As of 1 March 1995, the default pid is 4.

47.4 ip heard Display the ip-heard list. This shows the recently heard tcp/ip systems and may be set on netrom interfaces. This See also the 'ip hport' and 'ip flush' commands.

47.5 ip flush Clear the ip-heard list. See 'ip heard' and 'ip hport'.

47.6 ip hport [<iface> [ON | off]] Display or set the ip-heard facility. If no argument is given, show the interfaces on which ip-heard is currently active. If <iface> is given, shows the status of the ip-heard flag for the given interface. If <iface> <on|off> is given, it will set the flag on or off. Default is on.If this flag is on, ip heard frames will be logged in a table. This table can be shown with the 'ip heard' command or with the mailbox'IHeard' command. Ip-heard logging on ax.25 interfaces logs all ipstations heard on the port, even if the system wasn't directly involved in the ip activity. For non-ax.25 interfaces, only ip frames that we were actively involved in (i.e. that we routed) are logged. (this difference is due to code internals)

47.6.1 ip hport port1 off

47.7 ip hsize [n] Default: 16 Display or set the maximum size of the Ip heard table. 0 means no limit.

47.8 ip rtimer [<seconds>] Default: 30 Display or set the IP reassembly time-out.

47.9 ip status Display Internet Protocol (IP) statistics, such as total packet counts and error counters of various types.

47.10 ip ttl [<hops>] Display or set 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. You should make the value slightly larger than the number of hops across the network you expect to transit packets. The default is set at compilation time to 255, the official recommended value for the Internet.

48 **lock [password <"password_string">];** Lock the keyboard or define a password string. Setting the password and locking the keyboard can only been done at the system console keyboard or in the /autoexec.nos startup file. The password can not been displayed.

48.1 If 'password' is given then <password_string> is saved as the unlock string.

48.2 If no parameters are supplied, the keyboard becomes locked when a password was specified earlier.

48.3 If the keyboard is locked, the password is requested. If a correct password password_string is supplied, the keyboard becomes unlocked.

49 **log [ON | off];** Turn on or off system logging. Log files are created daily and stored in the logs directory.

50 **look [user | socket#] (/<cmd>);** This command allows peeking into a user session on the BBS or into any non-local socket. 'look [user | socket#]' brings up a window which follows the specified user or socket.

50.1 When looking in on a bbs user, you can initiate a 'chat' session with that user. There are a few commands available:

50.1.1 /? or /h - will show a sort help line

50.1.2 /m <msg> - send a message to user. (only if 'looking' at user) User sees: '<sysop>: message text'

50.1.3 /c - initiate a 'chat' with user. The bbs will suspend while you talk with the user. Note: if not in 'chat' mode, all non-command strings will be ignored.

50.1.4 /q or /b -

50.1.4.1 if in 'chat' mode, finishes it and returns the user to the bbs.

50.1.4.2 if in 'look' mode, finishes looking at user/socket.

50.2 When looking at non-bbs sockets might be helpful debugging things or seeing what an ftp user or a smtp user is doing. NOTE: often you won't see what you expect: e.g., ftp user sockets don't show the data transferred including directory listings because data transfer occurs on a separate socket. (Not seeing it is NOT a bug, for once...8))

51 **lzw;** Lzw (Lempel-Ziv-Welch) is the data compression capability for some ASCII-mode sockets. At present, LZW compression is supported by JNOS SMTP, FTP, NNTP and POP3.

51.1 lzw mode [fast | compact] Default: compact Display or set the compression method used in data compression for specific sockets.

51.2 lzw bits [<number>] Default: 9 Display or set the number of bits used for the compression size. The more bits defined the larger the table space needed. Range is 9 to 16.

51.3 lzw trace [on | off] Default: off Display or set the lzw trace flag.

If set on, compression statistics will be displayed on the console when a compressed socket is closed.

52 **mailmsg <to_addr> ["<subject>"] "msg";** Mail a message to <to_addr> from <msguser>@<Hostname>. <msguser> can be set in the DOS environment (Set MSGUSER=<any_user_name>), and defaults to "sysop" if not set in the environment. <subject> is optional, and must be quoted if it contains whitespace characters. <msg> is either a string to be placed in the body of the message (quoted if it contains white- space characters) or the path to a file to be placed in the message body. A path must begin with forward or backward slashes, or a drive specification, otherwise it is taken to be a <msg> string. This command is available when JNOS is compiled with #define'd MAILMSG.

52.1 Example #1: mailmsg n5knx "don't forget!" "replenish the Cardhu"

52.2 Example #2: at 0400 "mailmsg jpd@usl.edu /stats.out+"

52.3 Note: the bm.exe program is often used to originate smtp messages in an automatic fashion, but requires that JNOS be shelled-out to run, and it is possible that memory may be insufficient to do this. An alternative might be to use the mailbox "send" command, perhaps in conjunction with the upload command (to send a file), but this can't be easily automated. Hence the introduction of the mailmsg command.

53 **mbox [<subcommands>];** Without a subcommand, display the current users of the mailbox. With a valid subcommand, it will execute the following commands.

53.1 mbox alias [<alias>] ["cmd"] Set or show alias commands for the mailbox. To use an alias, the user must type the full alias which will then cause "cmd" to be executed. If the user types 'A', a list of sysop- defined aliases will be displayed. 'alias' is 6 characters maximum. "cmd" is 64 characters maximum. Note that "cmd" is in double quotes, and is interpreted as a mailbox command unless it begins with '@', in which case it is a console command. Only console commands that don't spawn sessions, and that don't require special privileges, are allowed. See also "mbox showalias", below.

53.1.1 'mbox alias' displays the current aliases.

53.1.2 'mbox alias <alias>' displays only <alias> if it is defined.

53.1.3 'mbox alias <alias> <cmd>' adds, deletes, or redefines <alias> Examples:

53.1.3.1 To add a new BBS command - mbox alias bbs "c port bbscall"

53.1.3.2 To delete an alias - mbox alias myalias ""

53.1.3.3 To redefine an alias - mbox alias bbs "c newport newbbscall"

53.1.3.4 To invoke the console pkstat command - mbox alias pkst "@pkstat"

53.2 mbox attend [ON | off] This displays or sets the attend flag. If set, users can initiate chat with the sysop via the O)perator mailbox command. (You need to have started the ttylink server for this !)

53.3 mbox bbsonly [iface] [on | OFF] Specify whether iface is only accessible to stations which have the BBS flag set.

53.4 mbox convers [ON | off] Display or set the access to the conference bridge. The mailbox or mailbox'CONV' command is disallowed when off. Default is ON.

53.5 mbox fbb [0 | 1 | 2] Display or set the flag which enables standard WA7MBL-style forwarding (0), FBB batched-style forwarding (1), or FBB compressed and batched forwarding (2). The flag is used to determine which forwarding capabilities is proposed to a another station. The default value is determined by whether JNOS was compiled with #define'd FBBCMP and/or #define'd FBBFWD. In any case, the forwarding method actually used is the intersection of capabilities of both stations involved in forwarding. Batched forwarding allows the forwarding direction to be reversed every 5 messages. Compressed FBB forwarding requires a completely transparent connection and uses Lempel Ziv Huffman compression.

53.6 mbox fwdinfo [string] Displays or sets the string that is used in the BBS R: header when forwarding mail to other BBSes.

53.7 mbox haddress [string] Displays or sets your system's hierarchical bbs address. As of version 1.08, you must include the bbs call with the hierarchical address. Example: mbox haddress wg7j.or.usa.na

53.8 mbox header [on | OFF] Explicitly turns the R: line in the message header on or off. With mbox header off, regular users can forward from JNOS to another system designated as a full-service bbs without leaving a trace to their system. This avoids having a downstream bbs improperly identify the originating station as a bbs. The elements of the R: header are optional as of JNOS107. The line has this format (where undefined elements will be dropped): R:yymmdd/hhmmz @:haddress [qth] fwdinfo #:mid $:bid Z:zip NOTE: 'mbox header off' also turns off the mbox third-party flag.

53.9 mbox hideport [<iface>] [on | OFF] Display or set a port that should not be displayed when the 'P' command in the mailboxis given. Display is always available to users with sysop permissions. This command is usefor for ports that should be backbone linking only, etc. Note that currently users can still do ax.25 connections via that port, if they know the name. Default is OFF, the port is displayed.

53.9.1 mbox hideport port1 on

53.10 mbox holdlocal [on | OFF] Display or set the flag indicating if locally-originated messages, or messages arriving via an SMTP-protocol connection are to held for sysop review before being forwarded or readable. Only messages written to areas listed in /spool/holdlist are affected. The sysop would typically use the LH and RH commands to review held messages and approve the removal of the hold status.

53.10.1 This command is enabled only with #define'd HOLD_LOCAL_MSGS at compile-time.

53.10.2 If invoked to display the hold flag status, and the status is ON, then all areas listed in the Holdlist file that contain held messages, are displayed, along with the total number of held messages.

53.11 mbox ifilter [on | OFF] Display or set the spurious I-frame filter flag, which when set will prevent unexpected I-frames sent to mycall from spawning a mailbox session. The effect of setting this flag is to deny access to the mailbox unless our ax25 bbscall, ax25 ttycall, convers mycall, ax25 alias or netrom alias is used. Mbox ifilter is necessary to send IP, Net/ROM, and segmentation frames over ROSE circuits.

53.12 mbox kick [<bbscall>] This is an immediate command. It forces the system to start a new BBS forwarding cycle. All BBSes in forward.bbs are examined unless <bbscall> is given, in which case only that entry in forward.bbs is considered. If <bbscall> is given in uppercase, a poll will be done; otherwise, a connect is attempted only when there is traffic to send. Example:

53.12.1 at 35 "mbox kick w5ddl+" kicks just w5ddl

53.12.2 at hh:35 mbox kick K5ARH forces an immediate poll

53.13 mbox mailfor [interval] At the end of each interval, the system reads all non-area mailboxes and checks them for unread mail. Next, a beacon is sent on all active interfaces. (see 'mbox mport' command). The beacon is addressed to the AX.25 address 'MAIL'. The data in this packet contains a list of the mailboxes with unread mail in the format 'Mail for: <user> ... '. Systems such as LAN-LINK can trigger mail-snatches from this beacon. Interval is in seconds. If <interval> is the string "now" an immediate check is done.

53.13.1 mbox mailfor exclude [areaname areaname ...] In certain cases you might not want a private area to show up in the mail beacon; e.g., private areas to be forwarded to another bbs. You can exclude such areas from the beacon with the 'mbox mailfor exclude' command.

53.13.2 mbox mailfor watch [call_1 call_2 ...] will watch for new mail arriving for the specified personal mailboxes, and add a blinking MAIL to the first status line if new mail exists. To remove the MAIL from the status line, read the last message in the appropriate mailboxes, and force a run of the mailfor process by 'mbox mailfor now'. The sysop will typically want to watch his personal mailbox, and perhaps a "check" mailbox to which unusual mail is rewritten.

53.13.3 Examples follow:

53.13.4 'mbox mailfor' will show the status of the timer and list mailboxes with unread mail.

53.13.5 A simple 'mbox mailfor exclude' shows the list.

53.13.6 'mbox mailfor exclude area1 area2 area3 ...' will disable these area names from the beacon. You can give multiple exclude commands to build the list.

53.13.7 'mbox mailfor watch call ...'

53.14 mbox mailstats This command shows how many messages have been read and sent by users and how many messages have been received from and forwarded to BBSes. Also shown is the current amount of "core" memory un-used, and the number of logins, current user count, and number of different users seen since JNOS was booted.

53.15 mbox maxusers [N] Default: 0 Display or set the number of simultaneous mailbox users permitted. A value of zero disables this feature. A mailbox connection which would exceed the maximum permitted user count, is disconnected with the error message "Too many mailbox sessions". Note that both incoming and outgoing connections, even forwarding sessions, are counted.

53.16 mbox mport [<iface>] [on | off] Displays or sets the interfaces for 'MAIL' beacons. Set this flag for each port you want the mail beacon to be sent on.

53.17 mbox newmail [ON | off] When ON, users will be notified during login which areas have new mail since the last time that user logged out. This function uses a time stamp on a status file for each message area. The information shown can later be repeated with the 'AN' mailbox command.

53.18 mbox noax25 [iface] [on | OFF] Specify whether iface is only accessible to stations connecting by other than the ax.25 protocol. Connections to the converse call are not affected by this command.

53.19 mbox nobid [on | OFF] Set whether to accept or refuse bulletins with NO BID. Off means refuse if there is no BID. On means accept regardless of BID. Default is OFF (refuse)

53.20 mbox nrid [ON | off] When set, the first time a user logs onto the system, the system will add the netrom node id in NODE:CALL format to the prompt. Users can then change it with the mailbox XN command. The system will use the new prompt format as set by the user the next time the user logs in.

53.21 mbox past [<N> | flush] Displays the users that have logged on since the system has been running. If an integer <N> is provided, only the most-recent N users are displayed. If "flush" is provided, the list of past users is cleared.

53.22 mbox password <newpassword> This sets a new remote sysop password. A remote sysop is a user whose entry in the ftpusers file has the SYSOP_CMD bit set. When a remote sysop enters the '@' command to the JNOS mailbox, and there is a non-null mbox password established, five random numbers are displayed. The remote sysop is expected to then transmit the letters corresponding to these numbers, taken as zero-relative positions in the password string. Several lines of five letters can be sent, only one of which need be correct. The last line sent must be empty, i.e., just a CR. If the response is correct, the remote sysop is then given the JNOS command-line prompt, and may issue most JNOS console commands. Commands which would require creation of a new session are disallowed. Use the "exit" command to exit from the JNOS command level.

53.23 mbox qth [location] This displays or sets the location of your system, and uses it in the R: headers when doing bbs forwarding.

53.24 mbox register [ON | off] Enable or disable user registration. If the user gives an e-mail address during registration, messages sent will include a 'Reply- To:' header with that e-mail address. Setting this command to OFF will prevent the 'please register' message from being sent to new users.

53.25 mbox reset [login_id_list] Force a disconnect for the named user(s), or list the names of the users currently connected if no argument is provided.

53.26 mbox secure [on | OFF] This displays or sets the mailbox secure flag. If set, only users coming on over netrom and ax25 connections can make gateway connects if they have the right privileges. If not set, anyone with the right privs can do a telnet connect, provided that their login name passes the valid-callsign tests within JNOS.

53.27 mbox sendquery [ON | off] If on, users will be queried if they really want to send the message after they have typed the ^Z or /ex when sending a message. 'N' or 'n' will abort, anything else will send the message.

53.28 mbox showalias [on | OFF] Set or display the flag which determines whether aliases (if defined) are listed in the mbox prompt.

53.29 mbox smtptoo [on | OFF] This displays or sets the smtp header flag. If set, the system will include most smtp headers when forwarding the message via bbs forwarding. When not set, the headers will be stripped (default).

53.30 mbox status Displays the current users of the system {NOS}.

53.31 mbox sysoponly [iface] [on | OFF] Specify whether port is accessible only by stations with SYSOP flag set. Default is off.

53.32 mbox timer [<nnnn>] Default = 0 (No forwarding) Entering only the subcommand displays the forwarding setting and countdown timer value. [<nnnn>] sets the interval between forwarding attempts in seconds. When the timer matures, the forward.bbs file is scanned for entries that match the time constraints, and either have msgs to forward or have the poll flag set on their entry. A process is started to initiate forwarding with each matching entry, all processes begin at once! Finer control of forwarding times can be obtained by keeping the timer at 0, and using the at command to issue repeated 'mbox kick' commands for the desired forwarding partners. For example, at 05 "mbox kick k5arh+" will try to forward to just k5arh at five minutes after each hour.

53.33 mbox tdisc [<nnnn>] This command sets the mailbox inactivity timer in seconds. If no user input has been received for nnnn seconds while connected to the mailbox, the user will be disconnected. Default = 0 = no timeout.

53.34 mbox tmsg [<string>] Display or set the 'telnet message'. This is the message sent to telnet connections before the login prompt is sent!

53.35 mbox trace [on | OFF] This displays or sets the value of the trace flag. If set, the mailbox is verbose about certain aspects of the forwarding cycle.

53.36 mbox usersonly [iface] [on | OFF] Specify whether iface is only accessible to stations which don't have the BBS flag set.

53.37 mbox zipcode [<nnnnn>] Set or display the system's postal zipcode. This is used in the R: line when forwarding.

54 **memory <subcommands>;** These commands are used for memory allocation.

54.1 memory freelist Display the storage allocator free list. Each entry consists of

54.2 a starting segment, in hex, and a size, in decimal bytes.

54.3 memory ibufsize [<size>] Display or set the size of the buffers in the interrupt buffer pool. The size should be set to the largest type of buffer plus a header size of 8. For example: If your ax.25 is the only interface and a packet length of 512 is defined, the ibufsize should be 512 + 72 + 8 = 592 . The 72 is the ax.25 header (source , destination, 8 digipeaters, 1 control byte and 1 pid byte). Default is obtained from the value of IBUFSIZE in config.h.

54.4 memory minalloc [<bytes>] Set the minimum number of bytes to allocate per malloc() call. Setting a small value (32 or 64) might allow the realloc scheme to work more effectively by preventing excessive memory fragmentation. Default = 0, no minimum allocation size.

54.5 memory nibufs [<number>] Display or set the number of interrupt buffer pool buffers. If the number of buffers is set, the statistics in the 'memory status' display are reset for number of interrupt buffer fails. The minimum available value is set to the requested number of buffers. A rule of thumb for the number of buffers is to watch the statistics and keep a minimum of 2 free buffers. Increase or decrease as required. Default is 10. See also the section on INTERFACE BUFFERS.

54.6 memory sizes Display a histogram of storage allocator requested sizes. Each histogram bin is a binary order of magnitude (i.e., a factor of 2).

54.7 memory status Display a summary of storage allocator statistics. heap size 52560, avail 12880 (24%), morecores 150, coreleft 5872

54.8 The first line shows the total size of the internal heap, the amount of memory available on the internal heap with the percentage of the total heap size, next the number of times memory has been requested from the Operating System, and the amount of memory the OS has left over.

54.9 allocs 16706, frees 16389 (diff 317), alloc fails 0, invalid frees 0

54.10 Next, the number of times memory has been allocated, and has been freed is shown. The difference is the number of buffers currently allocated. Alloc fails show up when the system is running out of memory resources. Invalid frees mean that memory was overwritten, and indicates the system is about to lose sanity...

54.11 garbage collections yellow 0, red 0

54.12 Garbage collections free memory from network control structures that could not have otherwise been freed. Yellow garbage collections are started when the total available memory, i.e. avail+coreleft, becomes smaller then memthresh. Red garbage collections indicate that available memory got below memthresh/2

54.13 interrupts-off calls to malloc 0, free 0

54.14 These should never be other then 0. They indicate calls to the memory allocator with interrupts off. These requests should be handled by the interrupt buffer pool. If these values are non-zero, you most likely have a problem.

54.15 Intqlen 9, Ibufsize 600, Iminfree 9, Ibuffail 0

54.16 This line shows the current number of interrupts buffers in the interrupts buffer pool, the size of each buffer, and the minimum number of free buffers. If this last number gets close to, or becomes zero, you should increase the buffer pool size with the 'memory nibuf' command. The statistics are reset when this command is executed.

54.17 memory threshold [<size>] Displays or sets the memory threshold size in bytes. If free memory gets below this value, no more new connections can be started and no new connections will be accepted. In addition, incoming packets are dropped. This is an attempt to preserve the system's sanity.

55 **mkdir <directory>;** Create a sub-directory in the current working directory.

56 **mode <iface> [vc | datagram];** Display or set the default transmission mode on the specified AX.25 interface. [TBD]The defaults can be overridden with the type-of-service (TOS) bits in the IP header. Turning on the "reliability" bit causes I frames to be used, while turning on the "low delay" bit uses UI frames. The effect of turning on both bits is undefined and subject to change.

56.1 In datagram mode, IP packets are encapsulated in AX.25 UI frames and transmitted without any other link level mechanisms, such as connections or acknowledgments.

56.2 In vc (virtual circuit) mode, IP packets are encapsulated in AX.25 I frames and are acknowledged at the link level according to the AX.25 protocol. Link level (i.e. AX.25) connections are opened as necessary.

56.3 In both modes, ARP is used to map IP to AX.25 addresses.

56.4 In both modes, IP-level fragmentation is done if the datagram is larger than the interface MTU. In Virtual Circuit mode, however, the resulting datagram (or fragments) is further fragmented at the AX.25 layer if it (or they) is still larger than the AX.25 <paclen> parameter. In AX.25 fragmentation, datagrams are broken into several I frames and reassembled at the receiving end before being passed to IP. This is preferable to IP fragmentation whenever possible because of decreased overhead (the IP header isn't repeated in each fragment) and increased robustness (a lost fragment is immediately retransmitted by the link layer).

57 **more <filename> [<text> ...];** Display the specified file(s) a screen at a time. The 'more' command creates a session that you can suspend and resume just like any other session.

57.1 To proceed to the next line, hit the CR key.

57.2 To proceed to the next screen, press the space bar.

57.3 To proceed to the next console screen by first clearing the current screen, press TAB. This may be faster than the scrolling action used by the space-bar technique.

57.4 To cancel the display, hit the 'q' key. If the text is given after the filename, each line in the file containing that text is displayed.

58 **motd ["message"];** Display or set the current message of the day. This message is shown to users logging in via telnet to port 87 (the ttylink service), or via connections to the system's ttycall alias (set by the 'ax25 ttycall' command).

59 **netrom <subcommands>;** These commands influence netrom behavior.

59.1 netrom acktime [<milliseconds>] Displays or sets the ack delay timer, similarly to ax25 t2. Default is 8000ms (i.e. 8 seconds).

59.2 netrom alias <aliascall> This sets the netrom alias call for this station. Other stations can connect to the ax25 callsign and to the NetRom alias (when set). The alias is broadcast with a NetRom broadcast. If netrom is not activated, you can use the 'ax25 alias' command to set the alias callsign, such that users can still connect to the alias, even though netrom activities are not allowed.

59.3 netrom bcnodes <iface> Initiates an immediate broadcast of nodelist on <iface>. Verbose behavior is controlled by the 'netrom interface' command.

59.4 netrom bcpoll <iface> Initiates a poll sent to the named interface. This poll will request a netrom routes broadcast from other nodes, so that the routing table can be updated. This is automatically done any time an interface is activated (or changed) for netrom. This should speed up route discovery at startup.

59.5 netrom call <call> Displays or sets the call to be used by the netrom interface. Note: this is a shortcut for the 'ifconfig netrom linkaddress' command. It defaults to the 'ax25 mycall' value.

59.6 netrom choke [<milliseconds>] Display or set the time breaking a send choke. Choke is the term netrom uses for flow control conditions. Default is 180000 ms (180 seconds.)

59.7 netrom connect <node> Starts a new JNOS session and attempts to establish a netrom transport connection with <node>, which must be a known node call or alias.

59.8 netrom derate [ON|off] Display or set automatic derating of netrom routes on link failure. Default is on.

59.9 netrom g8bpq {on | OFF] Display or set the G8BPQ-emulation behaviour. If set, transport layer conreq frames carry two extra bytes to indicate round-trip time, and conack frames have one extra byte to contain TTL (hopcount).

59.10 netrom hidden [on|OFF] Display or set the hidden node flag. Nodes with aliases starting with the '#' character are displayed using the command 'N *'. Default is OFF.

59.11 netrom interface [<iface> <quality> [n]] Display or activate <iface> as a netrom interface. 'netrom interface' will show the currently active netrom interface. Each time an interface is activated or changed, a broadcast poll will be sent out on the interface. This minimizes the route discovery time at startup, and will update routes when the interface quality has changed.

59.11.1 <iface> is an attached port name. If <iface> has already been activated as a netrom interface, re- issuing the command will reset <quality> and [n] to the new values.

59.11.2 <quality> can be set between 1 (very bad) and 255 (very good).

59.11.3 Interfaces are activated using verbose routes broadcasting by default, meaning that they broadcast all known routes. This can be changed by adding the optional 'n' parameter to the command. Then only the system itself will be announced in a broadcast, not the known routes.

59.11.4 NOTE: If the paclen of the interface is smaller then the netrom mtu + 20, then the netrom mtu will be set to paclen-20 . This is to avoid fragmentation, causing incompatibilities with non-JNOS based NetRom nodes. See the 'ifconfig <iface> paclen' command for more.

59.12 netrom irtt [<milliseconds>] Display or set the initial round trip time. Default is 45000ms, i.e. 45 seconds.

59.13 netrom kick <nrcb> Give the control block a kick to get activity going again.

59.14 netrom load Retrieves the last set of netrom destination data saved with the 'netrom save' command.

59.15 netrom maxclients [<count>] >deprecated jnos2.0: Displays or sets the maximum number of simultaneous outgoing netrom sessions that will be allowed. The default is 10. Reduce <count> if network congestion is a problem.<

59.16 netrom minquality [<minqual>] Display or set the minimum quality for recognizing a node entry. Entry's below this value are not considered valuable for usage. Default is 10 >superceeds 50<. You are strongly urged to set this value much higher, so as to minimize the space needed to store the nodes list.

A consequence of too many nodes listed is memory depletion, leading to lockups or reboots.

59.17 netrom neighbour Display all known net/2rom neighbors.

59.18 netrom nodefilter <subcommands> Manipulate node filtering.

59.18.1 netrom nodefilter add <neighbor> <iface> Add <neighbor> on port <iface> to the filter table. See the 'netrom nodefilter mode' command to determine the manner to handle node updates from <neighbor>.

59.18.2 netrom nodefilter mode [none|accept|reject] Display or set the initial node filter scheme. 'none' accepts all netrom routes and is the default. 'accept' accepts routes only from nodes defined with the 'netrom nodefilter add' command. 'reject' does not accept routes from any nodes defined with 'netrom nodefilter add'

59.18.3 netrom nodefilter drop <neighbor> <iface> Delete the node <neighbor> on interface <iface> from the filter table.

59.19 netrom nodetimer [<seconds>] Display or set the interval to transmit the nodes list. If you want to use other than the default, you must first attach the netrom interface with 'attach netrom' and then set the new nodetimer value. Default is 1800 seconds (half an hour).

59.20 netrom obsotimer [<seconds>] Display or set the time a node obsolescence count gets decremented. If you want to use other than the default, you must first attach the netrom interface with 'attach netrom' and then set the new obsotimer value. Default is 1800 seconds.

59.21 netrom promiscuous [OFF|on] Enables nodes with a path quality greater than defined with minquality. If on, all nodes are received regardless of nodefilter mode. Default is off.

59.22 netrom qlimit [<nnnn>] Display or set the maximum queue limit for choke to occur. Similar to ax25 window. Default is 512 bytes.

59.23 netrom reset <nrcb> Remove the control block. You can find the control block with the 'netrom status' or 'socket' commands.

59.24 netrom retries [<nn>] Display or set the maximum number of retries on connect, disconnect or data. Default is 3.

59.25 netrom route <subcommands> NetRom routing commands.

59.25.1 Routes can be marked as 3 types in the various route displays:

59.25.1.1 'P' - a permanent route set with the 'netrom route add' command

59.25.1.2 'B' - a route received through a nodes broadcast from a neighbor node

59.25.1.3 'R' - a recorded route from an incoming packet from a not previously known netrom node

59.25.2 netrom route add <alias> <call> <iface> <quality> <neighbor> Add a permanent netrom route. The new route is to netrom system <alias> with call <call>, and the route is on interface <iface> with quality <quality> via the neighbor <neighbor>.

59.25.2.1 netrom route add salem af7s-1 port1 178 k7uyx-1

59.25.2.2 A route to a direct neighbor looks like:

59.25.2.3 netrom route add crv k7uyx-1 port1 192 k7uyx-1

59.25.3 netrom route drop <destination> <neighbor> <iface> Delete the netrom route to call <destination> via neighbor <neighbor> on <iface>.

59.25.4 netrom route info [<destination>] Display the route a packet would take to get to <destination>. If <destination> is not given, information about all known netrom nodes is displayed. 'netrom route info' and 'netrom route info *' are equivalent commands.

59.26 netrom save Saves (i.e., writes) the netrom node table to Netromfile (default is /netrom.sav) for later use by the 'netrom load' command.

59.27 netrom split <node> Same as a 'netrom connect <node>', except a split-screen is used so that keyboard input appears in its own window.

59.28 netrom status Display all netrom connections.

59.29 netrom tdisc [secs] Display or set the NetRom Link "redundancy" timer. Value is in seconds. When no data exchange has happened during this time the link is reset and closed. Default is 900 seconds (15 minutes).

59.30 netrom timertype [exponential|LINEAR] Displays or sets the type of backoff used on netrom retries. Default is linear.

59.31 netrom ttl [<hops>] Display or set the maximum number of hops a frame might take before being discarded. Default is 10.

59.32 netrom user <usercall> Display or set the user call to be used by 'netrom connect'. The default call is the ax25 mycall.

59.33 netrom window [<frames>] Display or set the size of the sliding window. This is the largest send and receive window we might negotiate. Default is 2.

60 **nntp;** The 'nntp' commands control the operation of the Network News Transfer Protocol (NNTP). The nntp features are defined at compile-time. Two NNTP modules are available, "NNTP" which is an NNTP client only, that stores news in a mailbox-compatible form, and another, "NNTPS" which is both an NNTP client and server, that stores news articles in a form inaccessible to mailbox users.

60.1 We first describe the "NNTP" nntp client features:

60.1.1 nntp addserver <nntpserver_host> <interval> [<range>] [<groups>] Add an NNTP news server to query every <interval> seconds for new articles in the specified <groups>.

60.1.2 <range> specifies the time-of-day limits when the queries will be made, in the form hh:mm-hh:mm where start time precedes end time.

60.1.3 Multiple 'nntp addserver' commands may be used to concatenate groups (up to a maximum of 512 bytes).

60.1.4 Example: nntp add w5ddl.ampr.org 3600 18:00-06:00

60.1.5 nntp directory [<News_spool_dir> [News_control_dir>] Display or set the spool directory for spooling news articles. Default is /spool/mail. Optionally set a new control directory. The default control dir is /spool/news.

60.1.6 nntp directory old=new Establish a newsgroup name mapping, so that a newsgroup name beginning with <old> is changed to one beginning with <new>, which may be a null string. To delete a mapping, use <old>==. The mapping scan continues until the list is exhausted, in the same order the nntp dir commands were specified.

60.1.6.1 Example:

60.1.6.1.1 nntp dir rec.radio.=

60.1.6.1.2 nntp dir amateur.=

60.1.6.1.3 nntp dir shortwave=swl

60.1.6.1.4 nntp dir equipment=eq

60.1.6.2 will map rec.radio.amateur.policy to policy, rec.radio.swap to swap, rec.radio.shortwave to swl, and rec.radio.amateur.equipment to eq.

60.1.7 nntp dropserver <nntpserver_host> Drop the specified NNTP server from the list of servers to contact.

60.1.8 nntp firstpoll [<#days>] Default: 5 Sets or shows the number of days of old news that is requested in the initial poll to a new server.

60.1.9 nntp groups <group> [<group> ...] Default: All groups Display or set the currently set USEnet newsgroup(s). The group names are separated by spaces or commas. The '*' and '!' metacharacters (meaning 'all' and 'not' respectively) are supported.

60.1.10 nntp kick <nntpserver_host> Kick the local NNTP client to get in touch with the named server.

60.1.11 nntp listservers List the currently defined servers.

60.1.12 nntp lzw <ON | off> Turn on or off attempts to request LZW compression be used by the server when sending articles.

60.1.13 nntp maxcleints [<count>] Displays or sets the maximum number of simultaneous NNTP client sessions that will be allowed. The default is 0, that is, no limit (except that imposed available memory).

60.1.14 nntp trace <level> Default: 1 Sets or shows the current trace level for NNTP traffic.

60.1.14.1 Level

60.1.14.1.1 0: No tracing.

60.1.14.1.2 1: Display serious errors only

60.1.14.1.3 2: Display serious and transient errors

60.1.14.1.4 3: Display serious and transient errors, plus session progress

60.1.14.1.5 4: Display serious and transient errors, session progress and actual received articles

60.1.14.1.6 5: Display errors.

60.1.15 nntp quiet <yes | NO> Default: no Sets or shows the current arrival-notification setting for NNTP traffic. The notification will include a BEL character if "smtp quiet no" is in effect.

60.2 We now describe the "NNTPS" client/server commands. Remember that the 'start nntp' command must be used to allow the nntp server to accept connects from other nntp clients.

60.2.1 nntp active Displays the active file, which shows configured newsgroups. See 'nntp create'.

60.2.2 nntp access [on | OFF] Displays or sets whether access permissions are enforced. When enabled, the file /spool/news/access is scanned to determine the access permissions for a client host. Each line of this file has fields of the form: host:permissions: where host is explicit hostname (FQDN) or starname, eg, *.aara.org and permissions are a string of chars: R => read, P => post, none=>deny access. When access is turned on, hosts not mentioned are DENIED nntp access.

60.2.3 nntp add <nntpserver_host> <interval> [<range>] [<groups>] Add an NNTP news server to query every <interval> seconds for new articles in the specified <groups>. If no <groups> are specified, all groups found in /spool/news/active are checked.

60.2.3.1 <range> specifies the time-of-day limits when the queries will be made, in the form hh:mm-hh:mm where start time precedes end time.

60.2.3.2 Multiple 'nntp add' commands may be used to concatenate groups (up to a maximum of 512 bytes).

60.2.3.3 Example: nntp add news.usl.edu 3600 usl.test,rec.radio.swap

60.2.4 nntp create <news.group.name> [y|n] Updates the /spool/news/active file, which must have an entry for each news group you wish to receive. Choose y to permit posting to this group, or n to deny posting. y is assumed if nothing is specified. The /spool/news/pointer file is also updated with the path to the directory which will contain the articles. Articles will be stored as separate files, named by an integer corresponding to their arrival order.

60.2.5 nntp drop <nntpserver_host> Drop the specified NNTP server from the list of servers to contact.

60.2.6 nntp dump <newsgroup> [<areaname>] Dump all the news articles in <newsgroup> to the JNOS area called <areaname>. This would allow mailbox users to read news, but no provisions are made to dump just new articles. If <areaname> is omitted, then <newsgroup> is used as the area name. Note: this command is unavailable if JNOS was compiled with #define'd NEWS_TO_MAIL (see note 4 below).

60.2.7 nntp firstpoll [<#days>] Default: 5 Sets or shows the number of days of old news that is requested in the initial poll to a new server.

60.2.8 nntp ihave [<val>] Default: 0 Set or display the IHAVE nntp-protocol behavior. The IHAVE protocol tells the server the message-ids of articles stored here; it is used to forward articles off this system.

60.2.8.1 0 => IHAVE disabled (default)

60.2.8.2 1 => IHAVE reports only for newsgroups associated with serverhost

60.2.8.3 2 => IHAVE reports for all newsgroups

60.2.9 nntp kick <nntpserver_host> Kick the local NNTP client to get in touch with the named server.

60.2.10 nntp list List the currently defined servers.

60.2.11 nntp lzw <ON | off> Turn on or off attempts to request LZW compression be used by the server when sending articles.

60.2.12 nntp post Posts to a local newsgroup. A session is created, and the console user is queried for UserName (unless established by a prior 'nntp profile' command), Newsgroup, Subject, and message body. While entering the msg, a line consisting of ".u" or ".r" will then prompt for a file name, which is inserted into the article being built. "/EX", "***END" or "." will end the article when found alone on a line. When the message body is terminated, the prompt [Send, Abort, Exit, List] is displayed. Enter the first letter of the desired choice. Note that Exit quits the post subcommand, while Abort (or Send) allows you to post another article.

60.2.13 nntp profile {fullname|host|organization|reply|sig|user} string_value Profile establishes values for the header fields of posts originating here. Options include:

60.2.13.1 sig path_to_signature_file

60.2.13.2 host FQDN Defaults to our 'hostname'

60.2.13.3 fullname "First Mi. Lastname"

60.2.13.4 organ "organization name desired"

60.2.13.5 user "user name"

60.2.13.6 reply reply-to-address

60.2.14 nntp read <newsgroup> [<article_number>] Reads the local <newsgroup>, beginning with the first unread article unless <article_number> is also provided. A session is created for displaying the articles. After each article, a prompt "Read next/previous? (n/p/q) " allows the console user to easily progress through the articles.

60.2.15 nntp quiet [<val>] Default: 0 Dispays or sets the value if the quiet behavior flag. Nntp will display a message and/or beep when a new article arrives:

60.2.15.1 0 => beep only (default)

60.2.15.2 1 => beep and display msg

60.2.15.3 2 => no msg or beep

60.2.16 NOTES:

60.2.16.1 See the 'expire' command for information on removing old articles.

60.2.16.2 The TO: addresses, when present, are stripped from article headers by the NNTP client. This was done to prevent loops should the area be forwarded to another JNOS system, since the TO: address would cause the msg to be routed back to the mail-to-news daemon. If you want to forward an area, give the TO: address on the line in forward.bbs that lists the area. Example: rec.radio.swl all@swl

60.2.16.3 The NNTPS software includes a mail-to-news feature, such that email with a To: address that begins with "!" is passed to the NNTPS module. The remainder of the To: address is interpreted as a newsgroup name, with the name truncated at the first occurrence of one of "%@.,/", and with "!" translated to "." and "+" to ",". An alias is usually used to provide this special name. For example, to route all ALLUS bulletins to both the allus area, and the ampr.allus newsgroup, use the alias: allus   allus !ampr!allus To do the above task, and also post to local.allus, use: allus allus !ampr!allus+local!allus

60.2.16.4 The NNTPS software includes a news-to-mail feature, such that news articles can be emailed to local or remote destinations after they are processed by nntp. This would allow, for example, emailing to a public area, so that BBS users too could read news articles. JNOS must be compiled with #define'd NNTPS and #define'd NEWS_TO_MAIL and a file /spool/news/gateway must exist to define the mapping from a newsgroup to an SMTP To: address. Each non-comment line in the gateway file must begin with a newsgroup name (starnames OK), followed by spaces or tabs, followed by the email To: address. Examples:

60.2.16.4.1 # comment line

60.2.16.4.2 rec.radio.swapsale

60.2.16.4.3 rec.radio.shortwaveswl

60.2.16.4.4 rec.radio.amateur.* ham

61 **nrstat;** Displays the netrom serial interface statistics. This is only valid if the serial port was attached in NRS mode. See the 'attach' command for more.

62 **oldbid;** The 'oldbid' command is used to invoke the process which deletes old message-id's (or bids) in the Historyfile. The default location of Historyfile is /spool/history. Do not confuse this file with the NNTP History database, stored in /spool/news/history by default!

62.1 oldbid [interval age_in_days] Begin the oldbid process every <interval> hours, and delete those bids which are older than <age_in_days> days. If no arguments are provided, the current oldbid interval and age are displayed.

62.2 Example: To expire bids every 24 hours, that are 30 days old, use: oldbid 24 30

63 **param <iface> [<param> [<value>]];** Param invokes a device-specific control routine. A simple 'param <iface>' will give a list of available parameters and their current values, for the interface <iface>. <param> can be the literal name of the parameter, or it can be its numeric index (see below). <value> is either a boolean value (such as ON or OFF) converted to 1 or 0, or an integer.

63.1 On a serial interface, the param command sends control packets over the serial port. For example, 'param port1 txdelay 255' will set the keyup timer (called txdelay) on the KISS TNC configured as port1 to 2.55 seconds (255 x .01 sec). In most TNC KISS implementations, a 10ms tick count is used, so that (for example) 30 means 300ms, and a limit of 255 is imposed on <value>.

63.2 The SCC driver, on the other hand, allows a limit of 65535 for <value>.

63.3 Currently supported <params> include:

63.3.1 Name (index) Definition

63.3.2 TxDelay (1) keyup delay before sending data
63.3.3 Persist (2) the csma persistence (range 0-255) (probability of success
63.3.4 SlotTime (3) the channel access slottime (how often we throw the dice)
63.3.5 TxTail (4) time to keep transmitter keyed up after end of packet
63.3.6 FullDup (5) simultaneous TX and RX enabled when ON
63.3.7 Hardware (6)
63.3.8 TxMute (7)
63.3.9 DTR (8) Assert DTR when true, de-assert when false.
63.3.10 RTS (9) Assert RTS when true, de-assert when false.
63.3.11 Speed (10) async baud, etc.
63.3.12 EndDelay (11)
63.3.13 Group (12) SCC iface group id and TX-interlock style
63.3.14 Idle (13)
63.3.15 Min (14)
63.3.16 MaxKey (15) maximum time to allow transmitter to be keyed (0 - 65000)
63.3.17 Wait (16)
63.3.18 Pactor (17) Pactor mode = 1,2,3
63.3.19 Down (129) De-assert RTS and DTR on serial port
63.3.20 Up (130) Assert RTS and DTR on serial port
63.3.21 Blind (131) Ignore CTS and DSR transitions on serial port
63.3.22 RcvMode (253) set packet driver receive mode
63.3.23 Return2 (254) TNC partially (!!) exits KISS mode
63.3.24 Return (255) TNC exits KISS mode

63.4 Additional information for selected <params>:

63.5 The GROUP param specified an integer whose low-order 8 bits provide a transmit-group number (0 implies no group membership), and whose remaining bits provide flags describing that group's behavior: 1 => only transmit when all receive channels in this group are clear. 2 => don't transmit simultaneously on interfaces in this group.

64 **pause <seconds>;** Pauses for the specified number of seconds. This command is commonly used in the autoexec.nos file when loading in large routing tables or to pause after various time critical commands.

65 **ping <host> [<length> [<repeat_ms> [<incflag>]]];**Verify a host is alive, by sending it ICMP Echo Request packets.

65.1 <host> is the address to ping.

65.2 <length> is the number of bytes to use in the data portion of the packet being built. <length> defaults to a small value, sufficient to contain a timestamp to facilitate determining round-trip timing. Any additional data bytes will contain 0x55 characters ('U').

65.3 <repeat_ms> specifies that instead of sending a single ping, a session is to be established to do repeated pings every <repeat_ms> MILLISECONDS. A running display of attempts, round-trip times, and mean deviation of round-trip times is maintained in this session. To terminate the session, press F10 and then use the reset command.

65.4 <incflag> indicates that the host IP address is to be incremented by one at each ping attempt. It is intended to help search blocks of IP addresses for active hosts. It should be used sparingly if at all.

66 popmail <subcommands>; Popmail is a mail client compatable with POP2 or POP3 servers (depending on the jons compilation configuration) which will solicit mail from selected hosts carrying the TO address: "mailbox@host". Popmail may be scheduled or may be kicked to initiate service. New mail is deposited into spool/mail directory available to BBS mail agent. Please do note that jnos client will delete the source message upon successful transfer, thus if more than one client is accessing the server then mail may be missed by the other client.

66.1 NOTES relative to POP services in JNOS:

66.1.1 A POP server is also available in JNOS host. The administrator need only 'start' the pop2 or pop3 daemon, and configure the popusers file, to make his/her mail accessible to remote pop clients.

66.1.2 The standard of pop2 is superseded by pop3. If #define'd MD5AUTHENTICATE, JNOS pop3 supports the apop technique, which avoids, were possible, sending a plain-text password.

66.2 popmail addserver <host> [<seconds>] [hh:mm-hh:mm] <protocol> <mailbox> <username> <password> Add host IP to the pop server table. Note: On entering this command the host name is looked up. If nonexistent or unreachable, an error message is displayed.

66.2.1 When seconds is given, a timer is started to query the host with that interval for mail. If not specified no quering to the pop host will be started. You have to do that manually with a kick.

66.2.2 When hh:mm is given then only in that inclusive time frame are queries to the host made (allowed).

66.2.3 Protocol is either POP2 or POP3, depending on the mail service the host is providing.

66.2.4 Mailbox is the mailbox name on the host where mail has to be picked up.

66.2.5 Username and password are this system's validation parameters for the host.

66.3 popmail dropserver <host> Drops host from the table of pop servers to be queried. All references to the entry are deleted from the current system.

66.4 popmail kick <host> Starts a pop session with host to retrieve mail. This command is needed when no interval is specified with the popmail addserver command.

66.5 popmail list Lists the current popmail server table.

66.6 popmail lzw [on|off] Enables, disables lzw compression for popmail.

66.7 popmail quiet <yes|no> Displays or sets the notification of new mail arriving via pop.

66.8 popmail trace <level> Displays or sets the trace level of pop sessions. Note that tracing output goes to the log file. Current trace levels are:

66.8.1 0 - No tracing

66.8.2 1 - Serious errors reported

66.8.3 2 - Transient errors reported

66.8.4 3 - session progress reported

66.9 popmail t4 [<#seconds>] Default: 0 Displays or sets the pop client's idle timeout value (in seconds). Zero means no timeout, otherwise the value should be at least 300 seconds. This command is only available when #define'd POPT4 at compile time.

67 **prompt [on | OFF];** The 'prompt' command sets or displays the value of the MSDOS-Style prompt switch. If set ON, and status lines are not enabled, the current directory is printed as the first part of the console command prompt.

68 **ps;** Display process status information. The first line shows the time the system has been running, the active stack segment, the interrupt stack usage, and the JNOS psp segment. Note that the DOS JNOS code is loaded at segment address psp+0x10. Next, ps displays all processes in the system. The fields are as follows:

68.1 PID - Process ID (the segment of the process descriptor).

68.2 SP - The current value of this process' stack pointer.

68.3 maxstk - The size of the stack allocated to this process.

68.4 stksize - The apparent peak stack utilization of this process. This is done in a somewhat heuristic fashion, so the numbers should be treated as approximate. If this number is close to the maxstk figure, the system is likely to crash. Please notify the author if you find such a situation. (The program should be recompiled to give the process a larger allocation when it is started.)

68.5 event - The event this process is waiting for, if it is not runnable. This is actually a pointer value.

68.6 fl - Process status flags. There are three: I (Interrupts enabled), W (Waiting for event) and S (suspended). The I flag is set whenever a task has executed a pwait() call (wait for event) without first disabling hardware interrupts. Only tasks that wait for hardware interrupt events will turn off this flag; this is done to avoid critical sections and missed interrupts. The W flag indicates that the process is waiting for an event; the 'event' column will be non-blank. Note that although there may be several runnable processes at any time (shown in the 'ps' listing as those without the W flag and with blank event fields) only one process is actually running at any one instant (The Refrigerator Light Effect says that the 'ps' command is always the one running when this display is generated.)

69 **pwd [<directory>];** An alias for the 'cd' command.

70 **rarp;[tbd]** The 'rarp' commands are associated with the Reverse Address Resolution Protocol (RARP). RARP is used where a station knows its own Ethernet address or call sign but does not know its own IP address.

70.1 rarp Display RARP statistics.

70.2 rarp query <interface> <ether_address|callsign> [<ether_address|callsign> ...] Issue a RARP request for an IP address for <ether_address> or <callsign>, via <interface>.

71 **rdate <subcommand>;** Remote Date (rdate) is used to set the time of the local JNOS system based on the time obtained from a remote system's time server. The Unix time (port 37) protocol is used.

71.1 rdate offset [<+|-><hour_offset>] Causes the time read from the remote system to be adjusted by adding in the indicated <hour_offset>, which may be negative. If no <hour_offset> is provided, the current offset is displayed. Note: the time server provides the time in UTC. If an environment variable, TZ, is properly maintained, then an offset of zero should suffice. Otherwise, use the number of hours difference between local and UTC time.

71.2 This offset command IS OBSOLETE and is no longer compiled into JNOS after release 1.10M.

71.3 rdate server <hostname> Causes a connect to <hostname> TCP port 37, from which a time is obtained, adjusted by an offset if one was given, and then used to set the time on the local system.

72 **record [off | <filename>];** Opens <filename> and appends to it all data received or sent in the current session. (This includes up/downloading to a mailbox).

72.1 For example, if you are in a Telnet session 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 <filename>" will be displayed and another "JNOS>" prompt will be issued. Enter CR on a blank line and you will return to the Telnet session with recording activated. The command 'record off' stops recording and closes the file; this is done automatically when the associated session is closed.

73 **remark;** Writes its arguments to the current output stream. Similar to the MS-DOS 'echo' command. Example:

73.1 remark "do: source \dial_usl.nos...to dial USL and start PPP."

73.2 Writes the quoted string to the current output stream.

74 **remote...** The remote command can be invoked in any one of three ways. First in the form 'remote [-s <syskey>] [-g <gwkey>]', to set a key phrase used by the remote server process to validate commands sent to it. Second in the form 'remote [-p <port>] [-k <key>] [-a <kickaddr>] <host> kick | exit | reset', to send a UDP packet to a specified host commanding it to exit JNOS, reset the processor, or force a retransmission on TCP connections. Third in the form 'remote [-p <port>] [-k <key>] [-r addr/#bits] <host> add | drop', to send a UDP packet to a specified host commanding it to either add or drop an encapped route to a specified host or subnet via the sending system's IP address, that is, register ourselves as a gateway for a specified host or subnet for a finite period.

74.1 For the second and third forms of the remote command to be accepted, the remote system must be running the remote server. Also, the port number specified in the remote command must match the port number given when the server was started on the remote system. Further, if the remote system established a key phrase, then that phrase must also be provided via the -k option, so the remote system can check that it matches.

74.2 If the port numbers or keywords do not match, or if the remote server is not running on the target system, the command packet is ignored.

74.3 Even if the command is accepted there is no acknowledgement.

74.4 The 'kick' subcommand forces a retransmission timeout on all TCP connections that the remote node may have with the local node.

74.5 If the '-a' option is used, connections to the specified host are kicked instead. No key is required when using the 'kick' subcommand.

74.6 The 'exit' and 'reset' subcommands are mainly useful for restarting JNOS on a remote unattended system after the configuration file has been updated. The remote system should invoke JNOS automatically upon booting, preferably in an infinite loop.

74.7 The add or drop subcommands allow a JNOS system with a dynamically-assigned IP address, to register as a gateway with a cooperating system having a known IP address. The duration of the route thus established is controlled by the remote system, typically 15 minutes. Since UDP packets are not guaranteed to be delivered, it might be wise to send remote commands more frequently than the timeout. See the at command, well suited to this purpose.

74.8 remote -s [<key>] The 'exit' and 'reset' subcommands of 'remote' require a password. The password is set on a given system with the '-s' option, and it is specified in a command to a remote system with the '-k' option.

74.9 If no password is set with the '-s' option, then the 'exit' and 'reset' subcommands are disabled.

74.10 remote -g [<gwkey>] The 'add' and 'drop' subcommands of remote may require a password. The password is set on a given system with the '-g' option, and it is specified in a command to a remote system with the '-k' option.

74.11 Note that 'remote' is an experimental feature in JNOS; it is not yet supported by any other non-KA9Q-derived TCP/IP implementations, although a version of the remote command for BSD Unix exists.

75 **rename <oldfilename> <newfilename>;** Renames <oldfilename> to <newfilename>. This performs the same function as the DOS rename command.

76 **repeat [milliseconds] command;** Starts a new session screen with the output of the command updated every interval in milliseconds. If you hit F10 it ends the session.

76.1 Example: repeat 1000 tcp status

76.2 repeats every second 'tcp status' command

77 **reset [<session>];** Will cancel and reset the the specified session; if no argument is given, reset the current session. To complete the process, return to the now reset session and enter <CR> and the session will be cleared totally.

77.1 This command should be used with caution since it does not reliably inform the remote end that the connection no longer exists. In TCP a reset (RST) message will be automatically generated should the remote TCP send anything after a local reset has been done. In AX.25 the DM message performs a similar role. Both are used to get rid of a lingering half-open connection after a remote system has crashed.

78 **rewrite <address>;** Will show the result of the mail rewrite process, that is, how <address> is changed based on the contents of Rewritefile (default location is /spool/rewrite). The result of rewriting determines whether an incoming message is stored in an area (and, which area receives the message), or whether it is stored into /spool/mqueue for handling by SMTP. The rewrite command is useful for testing modifications to Rewritefile.

78.1 Example: rewrite n8fow@n8fow.ampr.org

78.2 shows the mail rewriting that will be done for mail addressed to n8fow@n8fow.ampr.org.

79 **rip <subcommand>;** The NOS implementation implements all features of the normal RIP protocol (RFC 1058) and all features of the RIP-2 protocol (RFC 1388) except multicasting (which NOS does not currently implement) and Route Tags (NOS does not implement any EGPs). The subcommands given first are used for RIP followed by the list for RIP-2. The RIP-2 implementation includes compatibility with RIP-1. The sets of commands are separated here to improve clarity.

79.1 >>>>>>>>>>>>> RIP-1 <<<<<<<<<<<<<<<<

79.1.1 rip accept <gateway> Remove the specified gateway from the RIP filter table, allowing future broadcasts from that gateway to be accepted.

79.1.2 rip add <hostid> <seconds> <flags> Add an entry to the RIP broadcast table. The IP routing table will be sent to <hostid> every interval of seconds. If <flags> is specified as 1, then "split horizon" processing will be performed for this destination. That is, any IP routing table entries pointing to the interface that will be used to send this update will be removed from the update. If split horizon processing is not specified, then all routing table entries except those marked "private" will be sent in each update. (Private entries are never sent in RIP packets). If flags is 2, the broadcast will also advertise a route to the system itself. Flags are accumulative, i.e. a value of 3 will mean both "split horizon" and "me too". See also the 'route' command.

79.1.3 Triggered updates are always done. That is, any change in the routing table that causes a previously reachable destination to become unreachable will trigger an update that advertises the destination with metric 15, defined to mean "infinity".

79.1.4 Note that for RIP packets to be sent properly to a broadcast address, there must exist correct IP routing and ARP table entries that will first steer the broadcast to the correct interface and then place the correct link-level broadcast address in the link-level destination field. If a standard IP broadcast address convention is used (e.g. 44.26.0.0 or 44.26.255.255) then chances are you already have the necessary IP routing table entry (unusual subnet or cluster-addressed networks may require special attention!) However, an 'arp add' command will be required to translate this address to the appropriate link level broadcast address; For example, arp add 44.255.255.255 ax25 qst-0 for an AX25 packet radio channel. (If there are multiple AX25 interfaces, make a unique address for each interface.)

79.1.5 rip drop <dest> Remove an entry from the RIP broadcast table.

79.1.6 rip kick Immediate command to send a rip update.

79.1.7 rip merge [on|OFF] This flag controls an experimental feature for consolidating redundant entries in the IP routing table. When rip merging is enabled, the table is scanned after processing each RIP update. An entry is considered redundant if the target(s) it covers would be routed identically by a less "specific" entry already in the table. That is, the target address(es) specified by the entry in question must also match the target addresses of the less specific entry and the two entries must have the same interface and gateway fields. For example, if the routing table contains

```
79.1.8 Dest    Len  Interface  Gateway   Metric  P Timer  Use
```

79.1.9 44.2.3.4   32   ax0   44.96.1.2   1   0   0 0

79.1.10 44.2.3.0   24   ax0   44.96.1.2   1   0   0 0

79.1.11 then the first entry would be deleted as redundant since packets sent to 44.2.3.4 will still be routed correctly by the second entry. Note that the relative metrics of the entries are ignored.

79.1.12 rip refuse <gateway> Refuse to accept RIP updates from the specified <gateway> by adding the gateway to the RIP filter table. It may be later removed with the 'rip accept' command.

79.1.13 rip request <gateway> Send a RIP Request packet to the specified <gateway>, causing it to reply with a RIP Response packet containing its routing table.

79.1.14 rip status Display RIP status, including a count of the number of packets sent and received, the number of requests and responses, the number of unknown RIP packet types, and the number of refused RIP updates from hosts in the filter table. A list of the addresses and intervals to which periodic RIP updates are being sent is also shown, along with the contents of the filter table.

79.1.15 rip trace [0|1|2]   Default is 0.   This variable controls the tracing of incoming and outgoing RIP packets. Setting it to 0 disables all RIP tracing. A value of 1 causes changes in the routing table to be displayed, while packets that cause no changes cause no output. Setting the variable to 2 produces maximum output, including tracing of RIP packets that cause no change in the routing table.

79.1.16 rip ttl <seconds> Displays or sets the time to live timer to 'seconds'. Normal time-out value is 240 seconds. This is not the ttl in a rip broadcast (16 = infinite). Set this timer before starting rip. Change this timer only in cooperation with your surrounding nodes.   Default is 240 seconds. End of RIP-1 commands.

79.2 >>>>>>>>>>>>>>>> RIP-2 <<<<<<<<<<<<<<<<<<<<

79.2.1 The following text is provided by N0POY who did the NOS implementation of RIP-2. This document covers the implementation of RIP-2 (RFC 1388) in NOS. Specifically the WG7J version of NOS. RIP-2 is an enhanced version of the RIP protocol (RFC 1058). RIP and RIP-2 are an interior gateway protocol (IGP). RIP-2 for NOS was implemented by Jeff White, N0POY.

79.2.2 This documentation is for the beta release V0.9 of RIP-2

79.2.3 RIP-2 Features include:

79.2.3.1 Routing Domains

79.2.3.2 Authentication

79.2.3.3 Proxy routing

79.2.3.4 Filtering of naughty nodes

79.2.3.5 Optional refusal of a default route

79.2.3.6 Enhanced logging and tracing

79.2.3.7 Route subnet masks correctly maintained

79.2.3.8 Optional refusal to accept older RIP version broadcasts

79.2.3.9 Mixing of RIP-1 and RIP-2 support

79.2.4 NOS RIP COMMANDS

79.2.4.1 RIP ACCEPT <gateway> The RIP ACCEPT command resumes the acceptance of RIP broadcasts from a specific node given in the <GATEWAY> field. Examples are:

79.2.4.1.1 RIP ACCEPT 192.55.248.1 or

79.2.4.1.2 RIP ACCEPT skeggi.tcman.ampr.org

79.2.4.2 RIP ADD <DEST> <INTERVAL> [<FLAGS>] [<RIPVER>] [AUTH <PASSWORD>] [RD <routing domain>] The RIP ADD command adds a node to the list of stations that are to be broadcast to with the local nodes routing table.

79.2.4.2.1 <DEST> is the destination node, usually a broadcast address.

79.2.4.2.2 <INTERVAL> is the number of seconds between broadcasts.

79.2.4.2.3 <FLAGS> are the RIP flags used (see below for the flags), it is a hexadecimal number.

79.2.4.2.4 <RIPVER> is the version of the RIP broadcasts. This may be a 1 or 2. The AUTH identifier precedes the authentication password to be included with the RIP broadcasts to this destination.

79.2.4.2.5 The RD identifier precedes the routing domain number. This number must range from 0 to 65535.

79.2.4.2.6 The authentication fields and routing domain fields are only valid with RIP-2 broadcasts. The password must be 16 characters or fewer. Printable ASCII characters are recommended, but not required.

79.2.4.3 RIP <FLAGS>:

79.2.4.3.1 0x01 Do 'split horizon' processing

79.2.4.3.2 0x02 Include ourselves in the routing broadcast

79.2.4.3.3 0x04 Broadcast RIP packets (default type)

79.2.4.3.4 0x08 Multicast RIP packets (not implemented) (RIP-2)

79.2.4.3.5 0x10 Poisoned Reverse on

79.2.4.3.6 0x20 Authentication data to be included in broadcast (RIP-2)

79.2.4.3.7 Recommend flags are Split Horizon, and Poisoned Reverse or 0x11. Authentication and routing domain data entered here only applies to the outgoing RIP broadcasts. See RIP AUTHADD and RIP AUTHDROP for entering acceptable passwords and routing domains. Example:

79.2.4.4 RIP ADD SKEGGI.TCMAN.AMPR.ORG 30 0x31 2 AUTH frodo RD 2

79.2.4.5 RIP ADD BIGGUS.TCMAN.AMPR.ORG 300 0x11 1

79.2.4.6 RIP PROXY <SRC> <DEST> <INTERVAL> [<FLAGS>] [AUTH <PASSWORD> [RD <ROUTING DOMAIN>]

79.2.4.6.1 The RIP PROXY command adds a node to the list of stations that are to be broadcast to with the local nodes routing table.

79.2.4.6.2 <SRC> is the node that the broadcast will "point" to.

79.2.4.6.3 <DEST> is the destination node, usually a broadcast address.

79.2.4.6.4 <INTERVAL> is the number of seconds between broadcasts.

79.2.4.6.5 <FLAGS> are the RIP flags used (see above for the flags), it is a hexadecimal number.

79.2.4.6.6 The AUTH identifier precedes the authentication password to be included with the RIP broadcasts to this destination.

79.2.4.6.7 The RD identifier precedes the routing domain number. This number must range from 0 to 65535.

79.2.4.6.8 The authentication fields and routing domain fields are only valid with RIP-2 broadcasts. The password must be 16 characters or fewer. Printable ASCII characters are recommended, but not required.

79.2.4.6.9 Proxy RIP is tricky, complex and not needed for normal use. Do NOT use proxy rip unless you understand what you are doing. Proxy RIP's primary use would be to advertise routes to another machine that is acquiring routing information via another routing protocol. See RFC 1388 for further details.

79.2.4.7 RIP DROP <dest> [<DOMAIN>] RIP DROP removes a routing broadcast entry. If a RIP-2 broadcast was entered, the correct routing domain needs to be entered, since it is possible to broadcast multiple routing domains to the same address. Example:

79.2.4.7.1 RIP DROP SKEGGI.TCMAN.AMPR.ORG 2

79.2.4.8 RIP AUTHADD <interface> <routing domain> [<password>]RIP AUTHADD adds an acceptable routing domain and optionally a password to a specific interface. Example:

79.2.4.8.1 RIP AUTHADD ax0 2 frodo

79.2.4.8.2 RIP AUTHADD en0 3

79.2.4.9 RIP AUTHDROP <interface> <routing domain>RIP AUTHDROP removes an acceptable routing domain (and password if any) from a specific interface. Example:

79.2.4.9.1 RIP AUTHDROP ax0 2

79.2.4.10 RIP REJECT <version> RIP REJECT is used to ignore older RIP broadcasts, as they may cause undesirable routing table alterations. The version number is the version number and below that are ignored. RIP version 0 (XNS RIP) is always ignored. The default is 0. To ignore RIP-1 broadcasts: RIP REJECT 1 would do the job.

79.2.4.11 RIP FILTER <on|OFF> RIP FILTER ON will cause advertisements to the default route (0.0.0.0) to be tossed and ignored. The default is OFF. This can serve as a LID filter. Default routes should NOT be advertised, unless there is a specific reason (i.e. this machine is a gateway to the rest of the Internet).

79.2.4.12 RIP MERGE <on|OFF> RIP MERGE ON will cause overlapping routing entries to be merged into one routing entry. The default is OFF. For example N0BEL.TCMAN.AMPR.ORG is a route to 192.133.30.0/28, and

192.133.30.16/28, with merging on this would become a single entry of 192.133.30.0/27.

79.2.4.13 RIP REFUSE <gateway> RIP REFUSE will reject all RIP broadcasts from the GATEWAY station. RIP ACCEPT is the opposite. By default all stations are accepted.

79.2.4.14 RIP REQUEST <GATEWAY> RIP REQUEST asks the gateway station to send a routing table now, rather than waiting for periodic updates.

79.2.4.15 RIP STATUS RIP STATUS will display various statistics for RIP-1 and RIP-2, RIP broadcasts, RIP refusals, and acceptable Interface, Domain and Password combinations. It also displays the refusing version level. The DEFAULT interface is for every interface. Thus unless removed, and RIP-2 broadcast with a domain of 0 does not require a password and will be accepted.

79.2.4.16 RIP TRACE <level> [<FILE>] RIP TRACE will begin tracing RIP operations. The higher the level, the more detailed the logging. Level 9 is the useful maximum, with level 0 (the default) being no logging. If a file is specified, logging will go to that file, else logging appears on the console.

79.2.4.17 RIP TTL <time-To-LIVE> RIP TTL sets the time-to-live before RIP entries expire from the routing tables. The default should work for almost all cases.

80 **rlogin <host>;** Sets up an 'rlogin' session via port 511 to a Unix-compatible host. Default terminal is a vt100-compatible terminal (with keyboard mappings as defined with the 'fkeys' command).

81 **rmdir <directory>;** Remove a sub-directory from the current working directory. The sub-directory must be empty before 'rmdir' can be used.

82 **route [<subcommand>];** With no arguments 'route' displays the IP routing table. NOTE: Attempting tcp connections to an address without an existing route fails immediately.

82.1 Here are some routing rules:

82.1.1 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.

82.1.2 There is one built-in interface: loopback. Loopback is generally for internal purposes, although destinations explicitly routed via this interface result in the packet being dropped, i.e., loopback can be used as a bit-bucket!

82.2 route add <desthostid>[/bits] | default <iface> [<gatewayhostid> | direct] [<metric>] This command adds an entry to the routing table. It requires at least two more arguments, the

82.2.1 <desthostid> IP address of the target destination.

82.2.1.1 The optional /bits suffix to the destination host id specifies how many leading bits in the host id 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, ARPA Class A, B and C networks would use suffixes of /8, /16 and /24 respectively. E.g. the command

82.2.1.1.1 'route add 44/8 ax0 44.64.0.2' causes any IP addresses beginning with "44" in the first 8 bits to be routed to 44.64.0.2; the remaining 24 bits are "don't- cares".

82.2.1.2 The special destination 'default' is used to route datagrams to addresses not matched by any other entries in the routing table; it is equivalent to specifying a /bits suffix of /0 to any destination hostid. 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 (TTL) fields expire. (Routing loops for specific addresses can also be created, but this is less likely to occur accidentally).

82.2.2 <iface> interface name to which its packets should be sent

82.2.2.1 If the interface is a point-to-point link, then <gatewayhostid> may be omitted even if the target is non-local because this field is only used to determine the gateway's link level address, if any.

82.2.2.2 If the destination is directly reachable, <gatewayhostid> is also unnecessary since the destination address is used to determine the interface link address.

82.2.2.3 If <rspf> is used and the system is a switch / router to multiple routes, the keyword 'direct' can be used instead of a <gatewayhostid> to set the metric higher than the default of 1. This way routes advertised by other rspf stations can be cheaper and get selected.

82.2.2.3.1 If 'direct' is given but <metric> not, a new algorithm is used to set the metric dependent on the number of subnet mask bits.

82.2.3 <gatewayhostid> when the destination is not local, the gateway host IP address should also be specified.

82.2.4 <metric> when added a 1 value is close and a 31 is far in computing a route choice when there are multiple paths or tie destinations.

82.2.5 Here are some examples of the route add command:

82.2.5.1 **route add 44.0.0.3 sl0** # Route datagrams to IP address 44.0.0.3 to port sl0 (happens to be a SLIP line #0). No gateway is needed because SLIP is point-to point.

82.2.5.2 **route add default ec0 44.0.0.1** # Route all default traffic to the gateway on the local Ethernet with IP address 44.0.0.1

82.2.5.3 **route add 192.4.8/24 ec0** # The local Ethernet has an ARPA Class-C address assignment; route all IP addresses beginning with 192.4.8 to it

82.2.5.4 **route add 44.0.0.10 ax0** # The station with IP address 44.0.0.10 is on the local AX.25 channel

82.2.5.5 **route add 44.64.0.0/16 encap 192.4.8.129(4[tbd])** # An encapsulation link to 192.4.8.12 where subnet 44.64.0.0 is accessible. The DNS services does not know where 44... is so we need to place packets into sites that can process them.

82.3 route addprivate <dest hostid>[/bits] | default <iface> [<gateway hostid> [<metric>]] This command is identical to 'route add' except that it also marks the new entry as private; it will never be included in outgoing RIP updates. It will also not be shown in the mailbox 'IProute' command.

82.4 route drop <dest hostid>[/bits] Delete an entry from the table. If a packet arrives for the deleted address and a default route is in effect, it will be used.

82.5 route flush Delete (drop) all temporary routes, that is, routes added with an expiration timer (eg, by RIP).

82.6 route look <hostname> Display just the routing table entry used to reach <hostname>.

82.7 route sort [off|ON] Display or set the route display sort flag. When set, the route command will sort its report, which tends to display similar routes together. When reset, the report is by decreasing number of significant bits used in comparing host addresses.

83 **rspf <subcommand>;** RSPF is the Radio Shortest Path First protocol. Each station listens for RRH (Router to Router Hello) messages. When such a RRH message is received, it will figure out if the link is bi-directional by pinging the other station. The protocol is described in the RSPF 2.1 specification.

83.1 rspf interface <interface> <quality> <horizon><interface> is the required interface rspf should use. quality is from 1 to 127, horizon is between 1 to 255. End nodes should have the quality set to 1. Immediate nodes normally set the quality to 8. The normally used value for horizon is 32.

83.2 rspf mode [vc | datagram | none] Display the preferred mode for RSPF. Modes are vc (Virtual Circuit) and datagram. none resets the preferred mode.

83.3 rspf rrhtimer [seconds] Display or set the rrh timer value.

83.4 rspf suspecttimer [<seconds>] Display or set the suspect timer value.

83.5 rspf timer [<seconds>] Display or set the update timer value.

84 **Session [<session#>] [arguments];** Without arguments, displays the list of current sessions,including session number, associated socket, the session screen-swap mode, the session type and state, the send and receive queue sizes, and the remote TCP or AX.25 address. An asterisk (*) is shown next to the current session.

84.1 Entering a session number as an argument to the session command will put you in converse mode with that session. If you have started the TTYLINK server, and have set Attended mode, an incoming telnet con- nect to the ttylink port will automatically create a split-screen session and switch you to that session.

84.2 session <session#> flowmode [on | off] <flowmode> displays or enables / disables setting of --more-- handling for <session#>. This is handy for long directory listings coming from an ftp session, for example . Escaping to command mode before issuing the dir command and entering "session # flowmode on" gives a page at a time to look at. At any time you can escape out again and switch flowmode off. Note that a ftp session has it's own flow command now built in. See FTP commands later in this manual.

84.3 To avoid the --more-- prompt when the output of console commands exceeds the screen length, turn flowmode off for session 0. This session number always exists, but is not shown in the session listing. Some commands, such as "more" or "dir", will ignore flowmode since they manage their own output pagination. But, these commands will accept a 'Q' to quit (abort) further output. Example: session 0 flowmode off

84.4 session <session#> split [on | off] <split> displays or enables / disables the split-window capability of session <session#>. This is useful to separate what you enter from the keyboard (shown in a small window at the bottom of the screen) from incoming data, which is shown in a large window at the top of the screen. For example, a ttylink session is begun in split mode.

84.5 session swapmode [ems | xms | memory | file]<swapmode> displays or defines how session screens are saved when the associated session is not current. The initial setting is determined by the -m command-line argument value and/or compilation options.

85 sessmgr [<subcommands>]; The sessmgr command sets or displays aspects of the Jnos unix session manager and existing sessions.

85.1 Depending on what compile-time options were selected, Jnos supports curses, raw, and dumb, managers.

85.1.1 Curses supports options "tty", "notty", "bold", "nobold", "8bit" and "no8bit". "ansi" is a synonym for "notty" and "noansi" is a synonym for "tty". The "tty" option turns off VT102/ANSI emulation, resulting in a "dumb crt" mode of operation. The "bold" option changes text highlighting from reverse-video to high-intensity display. The "8bit" option passes character codes > 127 to the display; the default "no8bit" mode only allows some of the line-drawing charcter codes > 127 to be accepted.

85.1.2 The Dumb session manager doesn't interpret escape sequences, and since it has no function keys, it interprets ^C as F10 and ^T as F9, that is, ^C switches to the command session, and ^T switches to the trace session. Use the session command in the command session to switch to other sessions.

85.1.3 The Raw session manager does not provide VT102 emulation, and so relies on the Linux tty driver to interpret escape sequences.

85.2 sessmgr create <sessmgr> <jnos_command> [<arguments>] Creates a new session for <jnos_command>, managed by <sessmgr>. This is equivalent to typing <jnos_command> and any arguments, having set the default session manager to <sessmgr>. Presently not all session-creating Jnos commands are available.

85.3 sessmgr default [<sessmgr>] Displays or sets the value of the default session manager. As mentioned above, <sessmgr> is one of "curses", "raw", or "dumb". The default session manager applies to sessions created by typing their associated command names in the command window. Note that options can be associated with <sessmgr> by appending a colon (:) followed by a comma-separated list of options. Example: curses:8bit,tty

85.4 sessmgr options <s#> [<options>] Displays or sets the options for session number <s#>.

85.5 sessmgr reparent <session> <sessmgr> Change the session manager to <sessmgr> for session <session>, which must be uniquely specified by number, name, or type, or may instead be the name of a session manager, which results on all sessions having that session manager being reparented to the new session manager <sessmgr>.

85.6 sessmgr status Display info about the available session managers, and which sessions exist, with their managers and options indicated.

86 **shell [cmd [args...]];** Suspends JNOS and executes a sub-shell ("command processor" under MS-DOS). When the sub-shell exits, NOS resumes (under MS-DOS, enter the exit command).

86.1 Note: see the COMSPEC environment variable.

86.2 Background activity (FTP servers, etc.) is also suspended while the subshell executes. Note that this will fail unless there is sufficient unused memory for the subshell and whatever command the user tries to run. When shelled out, Mailbox Operator connects and ttylink incoming connections are refused. A 'system unattended' message is sent to the "connector" of that socket.

86.3 Here are some comments particular to the MSDOS system: if no arguments are provided to the JNOS shell command, the MSDOS command processor is invoked to process commands entered from the console. Use the exit command to return to JNOS. If arguments are provided, the first one is assumed to be the program name to invoke. If the program is a batch file, you must precede it's name with /c so as to invoke the command processor explicitly to process the batch file commands:

86.3.1 shell /c ibackup.bat \spool\mail \bkup

86.4 The /c prefix is also useful to allow i/o redirection to be interpreted by DOS:

86.4.1 shell /c FC /A /C autoexec.bat autoexec.sav >outfile

87 **skick <socket#>;** This is a shorthand for the various 'kick' subcommands. This one searches the socket for the correct type and kicks the transport layer.

88 **smtp <subcommand>;** These commands are used for the Simple Message Transport Protocol service (that is, mail).

88.1 smtp batch [yes | no] If set smtp will batch the commands into one frame. When off only one command is sent and a response is waited for. Some old and flaky smtp servers cannot handle more than one command at a time. NOS can handle multiple. If you are not hindered by an old smtp server, setting batch reduces bandwidth. However, if you obtain bounced email containing: “>>> DATA <<< 503 Need MAIL before RCPT” you should turn batching off!

88.2 smtp dtimeout [<hours>] Default: 0 Displays or sets the number of hours a message will remain in the smtp mqueue before being returned to sender. Delivery attempts are made at "smtp timer" intervals. If <hours> is zero, the message remains in mqueue indefinitely.

88.3 smtp gateway [<hostid> | none] Displays or sets the host to be used as a "smart" mail relay. Any mail sent to a host not in the domain.txt file or not found via a nameserver query, will instead be sent to the gateway for forwarding (if #define'd). To undefine the gateway, specify "none".

88.4 smtp hopper [ON | off] Displays or sets the flag used to enable the "hopper" feature. This feature, available when JNOS is compiled with #define'd HOPPER, will attempt to deliver mail to a router which serves the mail's destination address. The router address is used even when an 'smtp gateway' is defined. This feature should not be enabled when X1J or other routers are used which don't accept SMTP connections. Note that if ##define'd HOPPER, then JNOS starts with smtp hopper enabled by default. This feature is inherently "dangerous"!

88.5 smtp kick [<hostname>] Run through the outgoing mail queue and attempt to deliver any pending mail to all systems, or just to <hostname> if specified. This command allows the user to "kick" the mail system manually. Normally, this command is periodically invoked by a timer whenever NOS is running.

88.6 smtp kill <jobid> Kill <jobid> and delete the message. If the job is "locked" by the smtp client process, you will be asked to confirm the removal.

88.7 smtp list List the current jobs in the mqueue. An "L" means locked and in progress. It is wise to add in autoexec.bat a "del /spool/mqueue/*.lck" command, since JNOS will not remove any pre-existing locks (it assumes other tasks share the mqueue, and dare not remove their locks).

88.8 smtp maxclients [<count>] Displays or sets the maximum number of simultaneous outgoing SMTP sessions that will be allowed. The default is 10. Reduce <count> if network congestion is a problem.

88.9 smtp maxservers [<count>] Displays or sets the maximum number of simultaneous incoming SMTP sessions that will be allowed. The default is 0, i.e., no limit other than the amount of memory available.

88.10 smtp mode [queue | ROUTE] Sets the smtp delivery mode. If 'queue', all messages are left in /spool/rqueue for external forwarding and handling. If 'route', messages are handled and, if for local, appended to a mailbox, or if remote they are forwarded. Default = 'route'

88.11 smtp quiet [YES | no] Enables or disables the inclusion of a beep in the message that new mail arrived at this system. See 'smtp trace' below, to enable the printing of a new-mail-arrived message.

88.12 smtp t4 [<seconds>] Displays or sets a t4 timer for smtp client (outgoing) sessions so that they will disconnect after a period of inactivity and prevent lockups. Default = 0, i.e., no disconnect timeout.

88.13 smtp tdisc [<seconds>] Displays or sets a disconnect timer for smtp server (incoming) sessions so that they will disconnect after a period of inactivity and prevent lockups. Default = 0, i.e., no disconnect timeout.

88.14 smtp timer [<seconds>] Displays or sets the interval, between scans of the outbound mail queue. For example, smtp timer 600 will cause the system to check for outgoing mail every 10 minutes and attempt to deliver anything it finds, subject of course to the smtp maxclients limit. Setting a value of zero disables queue scanning altogether. This value is recommended for standalone IP gateways that never handle mail, since it saves wear and tear on the (floppy) disk drive. Default = 0

88.15 smtp trace [<value>] Displays or sets the trace flag in the SMTP client, allowing you to watch SMTP's conversations as it delivers mail. Zero (the default) disables tracing. A trace value of 1 just enables the "new mail for n5knx from <k5arh@w5ddl.ampr.org>". Larger values produce more voluminous trace output.

88.16 smtp usemx [yes | NO] Displays or sets a flag enabling or disabling MX record lookups. This can be enabled if a domain server is available in the near distance (reachable). It should be disabled (default) if no domain server is in reach to satisfy the MX query. Note that MX record handling is limited to the five highest-preference hosts. Also, the smtp t4 timer must be set, in order to timeout on non-responsive hosts.

89 **socket [<socket#>];** Without an argument, displays all active sockets, giving their index and type, the address of the associated protocol control block and the and owner process ID and name. If the index to an active socket is supplied, the status display for the appropriate protocol is called. For example, if the socket refers to a TCP connection, the display will be that given by the 'tcp status' command with the protocol control block address.

90 **source <script_filename>;** The 'source' command runs a set of NOS commands which are in <script_filename>. This is a very convenient way of executing a series of commands without having to enter them individually at the keyboard.

91 **split <iface> <call>;** Same as the 'connect' command but uses a split screen mode. Do a 'help connect' for more info.

92 **start <servers...>;** Start the specified server. Do a 'start ?' for a list of available servers (this won't be displayed to remote sysops).

92.1 Servers often take an argument which specifies an alternate port number from which to listen. Example:

92.1.1 start callbook [port#] Default port: 1235

92.2 Default port numbers are listed in socket.h.

92.3 Exceptions:

92.4 start tip iface [modem | terminal] [timeout_secs] starts a mailbox server which listens for connections on <iface>. If <modem> is used, CD (carrier detect) must be asserted before a connection is recognized. A <timeout> value specifies how many seconds of inactivity must occur before a disconnect is initiated.

92.5 A tip about using the tip server to provide async dialup access to JNOS: the following commands demonstrate one way to set it up.

92.6 # source this file to configure COM1 for dialup access to JNOS

92.7 attach asy 0x3f8 4 ax25 dialup 2048 256 9600

92.8 #

92.9 ifconfig dialup descrip "dialup access"

92.10 param dialup up

92.11 # Send any desired config cmds here, eg, answer phone on 1st ring

92.12 comm dialup "atz e0 s0=1"

92.13 pause 2

92.14 #start tip dialup terminal 360

92.15 start tip dialup modem 360

92.16 # If the modem obeys DTR (must be asserted to answer the phone)

92.17 # the modem might be permed with S0=1, and then the answering could

92.18 # be controlled by param dialup up, param dialup down, commands

92.19 # possibly done daily by a repeating at command...

92.20 start http [port#] [drive] [rootdir] starts an http server. The default port is 80, the default disk drive is C, and the default rootdir (for html file access) is /wwwroot. See the http helpfile for more information, especially in the notes section.

93 **status [on|off];** The 'status' command displays general system information, attended flag, and a table of open files (when possible). This is useful for monitoring file transfer activity.

93.1 If STATUSWIN and STATUSWINCMD were #define'd when JNOS was compiled, the status command can take an optional argument, which determines whether the status window is turned on or off. The status window must be enabled when JNOS is started (see the -u, -w and -x command- line argument descriptions). The 'status on|off' command can then be used to disable (via "off"), or re-enable (via "on"), the status window at the top of the screen. When the status line state is changed, existing sessions will have their screen cursor repositioned.

93.2 The status window can have up to 3 lines. The first line shows the time, heap/dos available memory, number of connections to various services, and then a list of active sessions, where sessions with data waiting to be read are blinking. A blinking MAIL shows when the 'mbox mailfor watch' command detects unread email. When JNOS contains a mailbox server, the next (second) line shows the users connected to the bbs. A status symbol precedes the user's id:

93.2.1 none - user is idle

93.2.2 * - user is a bbs

93.2.3 @ - user is in sysop mode

93.2.4 ! - user has gatewayed out

93.2.5 # - user is reading or sending mail

93.2.6 = - user is transferring data (up/download)

93.2.7 ^ - user is in convers or sysop-chat mode

93.2.8 ? - use is in none of the above, but not idle...

93.3 The final line, when present, shows data depending on the current session. The current session number and type are always displayed. If the session is a network connection, the remote connection name, tx-queue (bytes for tcp, packets for ax.25), and state of the connection are displayed, followed by the retry timer, with current time left, and initial value. In repeat, more or look sessions, the third line shows the command, filename or user/socket for the session.

94 **stop <server...>;** Stop the specified server, rejecting any further remote connect requests. Existing connections are allowed to complete normally.

95 **strace [on|off]** Displays or controls trace output display. Without argument, strace displays the present setting. The "off" argument causes trace output to display on the F10 command session, the "on" argument causes trace output to be displayed under F9 session.

96 **tail <filename>;** Tail displays the last 20 (twenty) lines from <filename>. Tail does not include buffered text prior to being written to the file. See also the taillog command.

97 **taillog;** Displays the last 20 (twenty) lines from the log file.

98 **tcp <subcommand>;** These commands are used for the Transmission Control Protocol service. All TCP parameters are configurable per interface. Use the 'ifconfig <iface> tcp <command>' form to set or show the interface-specific values.

98.1 Commands of the form 'tcp <command>' set default or global values prior to attachment of ports. To set the system default TCP parameters, you must do so BEFORE attaching interfaces. After attachment, you must use the 'ifconfig <iface> tcp' command form to show or change values for that interface.

98.2 Notes: Attempting outgoing connections to addresses without an existing route results in Error number 219.

98.3 tcp access <permit|deny|delete> <ipaddr[/bits]|all> [loport [hiport]] >deprecated ?[tbd]: Display or set tcp access controls, which determine which TCP services (ports) are accessible to which IP addresses. If no tcp access commands are issued, the default behavior is to permit all hosts to access all ports. But once a TCP access command is entered, all other ports and addresses are denied unless specifically permitted by subsequent tcp access commands.

98.3.1 This subcommand adds or deletes an access control entry maintained in an internal table. Incoming tcp packets are compared with the table entries, in the order that they were added, to determine if access will be granted. Access is granted only if an entry with matching ipaddr or range, and ports, is found with "permit" set before either a match with "deny" set if found, or the end of the table is reached. The optional /bits suffix to the ipaddr specifies how many leading bits in the ipaddr are to be considered significant in the address comparisons. If not specified, 32 bits (i.e., full significance) is assumed. All addresses can be specified by "all". Loport and hiport specify the port or range of TCP ports for which the access control command applies. If "all" is given as the loport, or if no port range is specified, all ports are assumed, i.e., 1 to 65534.

98.3.2 "tcp access" will display the table of current access control entries.

98.3.3 Access commands should be entered from the most specific ipaddr to the least specific, since the first match (permit or deny) encountered in the internal table is definitive.

98.3.4 #Examples:

98.3.4.1 **tcp access permit 44.76.1.199 25** #Allow a specific AMPRnet host SMTP access

98.3.4.2 **tcp access deny 44.76.1.199** #but deny all other services to him

98.3.4.3 **tcp access permit 44.76.1/24** all #Allow all other AMPRnet hosts full access to TCP services

98.3.4.4 **tcp access permit 23.1.46/24 1 25** #Allow a specific subnet access to ports 1 through 25, which includes echo, discard, ftp, telnet, and smtp.

98.3.4.5 #Note that all other hosts not matched above, are denied access<

98.4 tcp blimit [<value>] Default: 31 Display or set the default tcp retransmission backoff limit. Normally each successive tcp retransmission is delayed a time value that increases exponentially or linearly. The backoff limit <value> serves to set the maximum backoff delay allowed. See also tcp timertype and tcp maxwait.

98.5 tcp clean Reset all tcp connections that are in a "FIN wait 2" state. This is useful to release memory resources held by JNOS for connections that were not properly closed.

98.6 tcp irtt [<milliseconds>] Display or set the initial round trip time estimate, in milliseconds, to be used for new TCP connections until they can measure and adapt to the actual value. The default is 5000 milliseconds (5 seconds). Increasing irtt when operating over slow channels will avoid the flurry of re-transmissions that would otherwise occur as the smoothed estimate settles down at the correct value. Note that this command should be given before servers are started in order for it to have effect on incoming connections.

98.6.1 TCP also keeps a cache of measured round trip times and mean deviations (MDEV) for current and recent destinations. Whenever a new TCP connection is opened, the system first looks in this cache. If the destination is found, the cached IRTT and MDEV values are used. If not, the default IRTT value mentioned above is used, along with a MDEV of 0. This feature is fully automatic, and it can improve performance greatly when a series of connections are opened and closed to a given destination (e.g. a series of FTP file transfers or directory listings).

98.7 tcp kick <tcb_addr> If there is unacknowledged data on the send queue of the specified TCB, this command forces an immediate retransmission. <tcb addr> can be found with the 'tcp status' command.

98.8 tcp maxwait [<msec>] Set or show the maximum time for retry timeout in milliseconds. Default = 0, no maximum.

98.9 tcp mss [<size>] Display or set the TCP Maximum Segment Size in bytes that will be sent on all outgoing TCP connect request (SYN segments). This tells the remote end the size of the largest segment (packet) it may send. Changing MSS affects only future connections; existing connections are unaffected.

98.10 tcp reset <tcb_addr> Deletes the TCP control block at the specified address.

98.11 tcp retries [<num>] Display or set the number of retries before a tcp connection will be reset. Default is 10. This is useful to eliminate idle connections that have not been properly shut down. If set to zero, there is no maximum, i.e. a connection will never retry out.

98.12 tcp rtt <tcb_addr> <milliseconds> Replaces the automatically computed round trip time in the specified TCB with the rtt in milliseconds. This command is useful to speed up recovery from a series of lost packets since it provides a manual bypass around the normal backoff retransmission timing mechanisms.

98.13 tcp status [<tcb_addr> | all] Without arguments, displays several TCP-level statistics, plus a summary of all existing TCP connections, including TCB address, send and receive queue sizes, local and remote sockets, and connection state. If <tcb addr> is specified, a more detailed dump of the specified TCB is generated, including send and receive sequence numbers and timer information. If "all" is given, the summary will also include TCBs in a listening state (awaiting a connection). In this case, a (S) will indicate that a server process is to be spawned when a connection occurs.

98.14 tcp syndata [yes | NO] Display or set the tcp syn + data piggybacking flag. Some tcp systems cannot handle syn + data together.

98.15 tcp timertype [linear | exponential] Display the current setting or set the timer type backoff algorithm. Default is linear.

98.16 tcp trace [yes | NO] Display or set the tcp trace flag on or off.

98.17 tcp view Display socket activity.

98.18 tcp window [<size>] Displays or sets the default receive window size in bytes to be used by TCP when creating new connections. Existing connections are unaffected.

99 **Telnet;** The 'telnet' command allows you to initiate a connection using the Telnet protocol. The end result is much the same as doing an AX.25 connect in most cases, but you'll be taking advantage of the attributes of the TCP/IP protocols. See also the descriptions of the "echo" and "eol" commands.

99.1 telnet <host> [<port_number>] connects to host on host port number specified.

99.2 host is a name compatible with DNS services. If a route can not be established, the result is error 219? [tbd].

99.3 port_number is a TCP/IP port on the target machine. Default is [tbd] telnet server.

99.4 parameter list: recognizes "cronly", telpac. [tbd]

100 **Term;** The term command is used to configure TCP access to local async ports. It is only available if JNOS was compiled with TERMSERVER #define'd. After starting satisfactorily, whatever is received from the tcp connection is sent to the interface, and whatever is received from the interface is sent to the tcp connection.

100.1 term iface [<iface> options...] Displays the list of interfaces accessible via term, or establishes a term interface and its operating parameters.

100.1.1 term iface <iface> 7bit [OFF | on] Displays or changes the settings of the flag which causes term to apply a mask of 0x7F to all characters read during the term session.

100.1.2 term iface <iface> break [<integer>] Displays or changes the value of the character code which, when read from the tcp input stream, causes term to send a BREAK to the associated serial device. The default is -1, i.e., disabled.

100.1.2.1 Example: 'term iface mdm1 break 3' will interpret ASCII ^C as a send-break character.

100.1.3 term iface <iface> cronly [OFF | on] Displays or changes the setting of the flag which causes term to ignore a newline (LF) read immediately after a CR is read from the tcp input stream. Note that if the nlcr option is in effect, a CRLF sequence is translated into a CRCR sequence, since the nlcr option is applied before the cronly option.

100.1.4 term iface <iface> drop Deletes the interface <iface> from the list of interfaces accessible to term. The interface must not be in use by a term process.

100.1.5 term iface <iface> flushwait [<#ms>]   Default: 0   Displays or changes the number of milliseconds after which any non-newline-terminated input from the serial port, is flushed so that it becomes visible to the term user. The default value is 0, meaning that no flushing is done. A flushwait value of 500 ms is a good value to use when it is important to see, for example, login prompts that are not followed by a CR.

100.1.6 term iface <iface> nlcr [OFF | on] Displays or changes the setting of the flag which causes term to translate a newline read from the tcp input stream, into a CR to be sent to the serial interface.

100.1.7 term iface <iface> noecho [OFF | on] Displays or changes the setting of the flag which causes telnet echoing to be turned off for the duration of a term session.

100.1.8 term iface <iface> noopt [OFF | on] Displays or changes the setting of the flag which causes telnet option processing to be turned off for the duration of the a term session.

100.1.9 term iface <iface> winkdtr [OFF | on] Displays or changes the setting of the flag which causes DTR to be deasserted for one second, at the start of a term session.

100.2 term password string   Sets the term facility password to the provided string. The default is no password required.

100.3 start term [port#]   When 'start term' is issued, connects to TCP port 5000 (default), will be asked the term password if one was defined, then asked an interface name (unless just one term interface was defined).

101 tip <interface>   Start a Terminal Interface Protocol (TIP) session that connects to the specified interface**(asy only!)** in "dumb terminal" mode. This feature is primarily useful for manually communicating with a serial device (e.g. a TNC operating in native "cmd:" mode), or for establishing SLIP connections via modems.

101.1 The interface must have already been attached with the 'attach' command.

101.2 All data subsequently transmitted and received via the interface are raw characters, without any protocol envelope. Any packet traffic (IP datagrams, etc) routed to the interface while a 'tip' session exists will be discarded.

101.3 To close a 'tip' session, use the 'reset' command. The interface will then revert to normal slip, nrs or kiss mode operation.

102 **trace [<iface> [off | <btio> [outfile]];** Controls packet tracing by the interface drivers. Tracing is controlled on a per-interface basis. Without arguments, 'trace' gives a list of all defined interfaces and their tracing status.

102.1 <iface> spec is the name given in the attach command. Tracing status output can be limited to the single interface by specifying the name.

102.2 <btio> Flag names are abbreviated as "BTIO". Specific bits enable tracing of the various interfaces and the amount of information produced. With this field blank, tracing status is printed. When this field is "off", tracing is immediately terminated. When flags are specified the trace is initiated or switched from the previous flag setting. The flags are given as a 4-digit hexadecimal number interpreted as follows:

102.2.1 B - Broadcast & Raw Dump selector [sum bits 1, 2, 4]:

102.2.1.1 1 - Broadcast filter flag. If set, only packets specifically addressed to this node will be traced; broadcast packets will not be displayed.

102.2.1.2 2 - Raw flag. If this bit is set, a "raw" dump style is selected, for those interfaces which support it (e.g., ppp).

102.2.1.3 4 - Poll flag. If this bit is set, include polls in the output trace of polled-kiss interfaces.

102.2.2 T - Controls type of tracing (select # 0 thru 3):

102.2.2.1 0 - Protocol headers are decoded, but data is not displayed

102.2.2.2 1 - Protocol headers are decoded, and data (but not the headers themselves) are displayed as ASCII characters, 64 characters per line. Unprintable characters are displayed as periods.

102.2.2.3 2 - Protocol headers are decoded, and the entire packet (headers AND data) is also displayed in hexadecimal and ASCII, 16 chars per line.

102.2.2.4 3 - A minimal display of headers and data is produced, provided JNOS was compiled with MONITOR #define'd.

102.2.3 I - Enable tracing of input packets if 1, disable if 0

102.2.4 O - Enable tracing of output packets if 1, disable if 0

102.3 <outfile> specifies the path and file name for output. If outfile is left blank, then trace output is presented to the screen session specified by the "strace" command.

102.4 Examples:

102.4.1 **trace port1 0x0111** Trace all packets on port1 and display with headers on screen.

102.4.2 **trace lan 0x1210 lan_ipt.trc** Trace all non-broadcast received packets on lan to the file named "lan_ipt.trc" in the jnos root directory.

103 **ttylink <host> [<port_number>]** Default: 87 The 'ttylink' command starts a tcp protocol session with <host> using the split screen mode. Also see the telnet command.

104 **udp status;** Show the status of active udp control blocks

105 **write <username|sock#> <message>;** Send a message to a particular user. <message> is the message, if "more then one word, put it in quotes". <username|sock#> can be either the user name of a mailboxuser, or a valid socket number. The latter allows you to send a message to a network conference user etc. See also 'writeall'. For Example:

105.1 The message from console: 'write wg7j "this is a test!"'

105.2 Will be shown as: '<bell>*** Message from sysop this is a test!

106 **writeall <message>;** Send a message to all mailbox and conference users. This command isuseful to warn users of an impending shutdown, and might be suitable forinclusion in your onexit.nos script file. Example:

```
    106.1 writeall "N5KNX BBS shutting down...bye!"
```

As a parting note, the above commands have been tested successfully on a variety of platforms. As variations become clear the differences will be included below:

Linux

MAC – OS-X

DOS

ROUTINE TASKS

Certain data and tasks are required for operation of a jnos node. Once installed and running file and user management becomes a focus.

User passwords and permissions

Use the following field descriptions in creating the "userfile" (default name: ftpusers) file. The generalized format of each line is:

```
  <name> <passwd> <dir>;<dir> <perm> [<dir>;<dir> <perm>] ...
```

<name> is the userid, normally a callsign for amateur radio use. In addition, some reserved names carry special connotation. The <name> "univperm" should be included in the ftpusers file to allow anyone not otherwise found in the ftpusers file to logon with "guest" status. Protocol-specific permissions can be allowed by using the following service names;

- tcpperm - telnet login to mailbox
- ax25perm - ax.25 login to mailbox
- nrperm - netrom login to mailbox
- confperm - convers signin
- pppperm - ppp's call to userlogin
- ftpperm - ftp login
- tipperm - tip login to mailbox

<passwd> is the pass word key the user must supply at login for access to the system.

- If <password> is set to '<string>', then <string> must be presented.
- If <password> is set to '*', then any entry will satisfy password.

<dir> is the highest directory in the system tree the user may access. It becomes the users root directory. Subdirectories under <dir> may be accessed by the user. More than one <dir> may be given separated by “;”. Full paths are specified for <dir>. Paths beginning with "/" reference the root filesystem, paths without the leading "/" begin at "/jnos".

> **<drive:/>** is the drive letter prefix needed for DOS platform. The full syntax is "<drive:/<dir>>". Note the "/" rather than the "\" is used in this context.

<perm> is a hex or decimal number that is the sum of the values which defines what the user is allowed to do while logged onto the system. The following is a list of the user permission values assignable for the “ftpusers” file. To set a list of options, simply add values and use the sum in either number base.

Name:	***(hex)***	***(dec)***	***Permit or Deny Feature or Action***
FTP_READ	0x1	1	Read files
FTP_CREATE	0x2	2	Create new files
FTP_WRITE	0x4	4	Overwrite or delete existing files
AX25_CMD	0x8	8	AX.25 gateway operation allowed
TELNET_CMD	0x10	16	Telnet gateway operation allowed
NETROM_CMD	0x20	32	NET/ROM gateway operation allowed
SYSOP_CMD	0x40	64	Remote sysop access allowed
EXCLUDED_CMD	0x80	128	This user is banned from the BBS
PPP_ACCESS_PRIV	0x100	256	bit for PPP connection
PPP_PWD_LOOKUP	0x200	512	Priv bit for peerID/pass lookup
NO_SENDCMD	0x400	1024	Disallow send command
NO_READCMD	0x800	2048	Disallow read command
NO_3PARTY	0x1000	4096	Disallow third-party mail
IS_BBS	0x2000	8192	This user is a bbs
IS_EXPERT	0x4000	16384	This user is an expert
NO_CONVERS	0x8000	32768	Disallow convers command
NO_ESCAPE	0x10000	65536	Default is no escape char
NO_LISTS	0x20000	131072	No lists displayed from mailbox
NO_LINKEDTO	0x40000	262144	Disable '*** linked to'
NO_LASTREAD	0X80000	524288	Ignore last read in <area>.usr (shared accts)
NO-FBBCMP	0x100000	1048576	Avoid FBB compression
XG_ALLOWED	0X200000	2097152	Allow XG (dynip route) cmd

The initial directory (that is, your starting directory after an ftp session is established) is the first directory listed, UNLESS one of the directories in the list is preceded by an "=" to flag it as the initial directory. Example:

```
  anonymous * /pub/wr_only 0x0002  /pub/rw_del 0x0007 =/pub 0x0001
```

You may provide access to more than one set of directories with different permissions for each. This allows a user to access a personal directory with complete read/write/delete access and a public directory with read permissions only, or any other combination you may desire.

```
       univperm * /public 138283
  (or: univperm * /public 0x21c2b)
```

gives anyone not otherwise known login permission as a guest who can read or create (upload) new files on FTP connections, access ax25 or netrom stations, but has no mbox send, read, 3rd_party, or list functions.

```
    wg0b doug c:/wg0b 0x407f /public;/nts 0x407b
```

defines two different sets of permissions for three different paths.

Initiate forwarding (and/or reverse forwarding)

BE CAREFULL !!! THE NEXT COMMANDS WILL CAUSE THE DXP OR SCS TO START SWITCHING YOUR HF RADIO TRANSMIT ON AND OFF AT REGULAR INTERVALS. IN OTHER WORDS, MAKE SURE YOU HAVE YOUR HF EQUIPMENT SETUP PROPERLY. I'M NOT TO BLAME FOR DAMAGE CAUSED BY YOU FAILING TO FOLLOW PROPER PROCEDURES FOR SETTING UP YOUR HF EQUIPMENT. YOU HAVE BEEN WARNED !!!

ALSO, BECAUSE THIS IS EXPERIMENTAL, I DISCOURAGE UNATTENDED FORWARDS ![tbd]

Note : Stay away from the generic 'mbox kick' and instead keep the kicks to individual stations (for the HF forwarding anyways).

To initiate a forward (if you have messages or bulletins waiting to be sent to the remote station), then simply use something like this :

```
...> mbox kick wu3v
```

If you have no traffic to forward TO your remote station, but you want to do a reverse forward (to pick up anything from THEM), then use this :

```
...> mbox kick Vo1xc
```

Note : note the first letter is CAPITALIZED in the callsign. That is how JNOS does a reverse forward. Capitalize the first character of the remote station.

Believe it or not, that's all there is to it !

HF server side, accepting connections from REMOTE stations

I have written a server side for JNOS 2.0 that will let you run a server on any of the HF ports. The only drawback is that once the server is running for a particular HF port, then that port can no longer be used to make outgoing connections, and you can not 'mbox kick' to forwarding partners that are configured for that particular HF port.

This version of the server code has no 'stop' command at this time. The only way to break out of the server at this time is to restart the JNOS program. [tbd]

To start the HF server on a particular HF port, use a command like this :

```
...> hfdd server ptcpro start
or
...> hfdd server dxp38 start
```

You can put this command at the end of your autoexec.nos if you want.

Watching the JNOS log and reporting problems

Be warned, the JNOS log will get filled up with the HF stuff. It's there on purpose to help me see how things are running. If you run into problems, try and explain them to me as detailed as possible, and include the JNOS log for me to analyze.

Another hint while you are forwarding with a remote station (AFTER you connect to them, or they have connected to you through the HF server), you can use the command, 'look <callsign>', from the sysop (F10) console, and actually follow the entire FBB forward session as it moves along. This is a great way to learn the FBB protocol. Sometimes this information can be helpful to me as well.

This is far from perfect, but I'm quite happy with the progress so far. That's it for now. Have fun, let me know what you think, feedback is very welcome. I don't care if it's good, bad, or ugly, or critical of design, or whatever. It won't get better if people don't bother telling me what the issues are.

Trace data management

There are a lot of personal preference and physical space constraints behind this issue. Need-to-know drives the selection of trace setup. ./trace is the default destination directory intended for storage of trace data, and user familiarity with platform utilities and search tools aids in where-to-find the information contained in the trace files.

For the purposes of this manual, it is suggested to trace the minimum data consistent with need-to-know, and place review on a calendar to manage file size and space available. Files that are too large become a burden to analyse so begin tracing into new files as tools dictate. Drive space can become a premium so delete old stuff where digging into performance history is no longer feasible.

File transfers via FTP

When FTP server is configured, file transfers in and out are possibility using the techniques that follow. [tbd]

INSTALLATION & CONFIGURATION

To prepare for use, JNOS requires installation and configuration as discussed in this section. The view from this section is services are defined by data in files, and jnos is started with references to files and parameters. JNOS communicates with the outside world thru I/O ports of various types, and services are provided by a collection of servers within jnos. Server parameters are largely supplied at start-up time from the file “autoexec.nos”. Jnos code must be compiled to support the services provided, and that is the topic of the final section.

Internet configuration considerations include selected interfaces with other jnos nodes, and local interfaces with an Internet Service Provider (ISP).

An important issue to consider in troubleshooting an installation is the effect of other services on jnos ability to communicate and access files. A firewall can stop internet communications cold. It typically works to disable the firewall while testing a configuration and re-enable the firewall after everything works and retest with the firewall in effect. SELinux also manages security and can cause jnos to not work. Tripwire can be employed to safeguard the platform against intrusion via external links that jnos supports. If any of these type services are to coexist with the jnos application, then administration must apply to both jnos and the operating system.

APPLICATION SOFTWARE AND FILE STRUCTURE

For those not using the Linux Installer, or issues deeper than the installer resolves, this section specifies the directory structure and software requirements. This section is built on the default configuration as delivered in pre-compiled and soruce. The compilation process may be used to modify this if desired by the owner.

THE LINUX INSTALLER

Installing JNOS from scratch can be a complicated procedure, and winds up scaring away first time users. This is my attempt to make JNOS installs as easy as windows. The user will not have to edit configuration files, they will simply be asked a series of questions, and the installer will do the rest. NOTE - this is thee prototype (experimental) and for LINUX only !

The installer is designed to provide the first time user with a very simplistic, yet runnable JNOS environment. It assumes you have only one TNC (already in KISS mode), and provides network connectivity to the linux box on which JNOS is running. Once installed, you can connect to JNOS from RF, and you can telnet to JNOS from the linux box. From the linux side, you will be able to telnet to JNOS and from there connect to any packet station on RF using the single TNC. Like I said, this is very simplistic, but I think a good starting point for a first time user. The installer is text based, no graphics libs are required.

Click here to down a compressed tar file of the installer package. Save it to the '/tmp' directory. Now login or 'su' to ROOT user, and uncompress the file using 'gunzip installer.tar.gz', then extract the contents using 'tar xvf installer.tar', which will create a directory called '/tmp/installer'. Change to the '/tmp/installer' directory, then run the command './jnosinstaller' as ROOT user. Just answer the questions as they come. When finished, keep in mind that JNOS has to be run as the ROOT user.
I've tested the installer and run the resulting JNOS configuration on both FC2 and FC3 (kernel 2.6.11) systems without any problems. The JNOS binary itself is compiled on a Slackware 9.1 system, and seems to run fine on FC2 and FC3. This configuration depends on the 'tun' kernel module (which is loaded on most systems anyways).

More information and updates will be available as time permits. Click here to continue to the main page

I/O CONFIGURATION

Help files – help, spool/help / Help files install from [tbd] and require a little cusomization. For the file spool/help/info use your favorite text editor to describe the system configuration for your system.

Startup – autoexec.nos

Welcome message files – motd / The login message in motd.txt requires editing to welcome users to the system.

User ID and permission files – spool/users, ftpusers, popusers

Forwarding files – forward.bbs, rewrite, alias

RF Ports

JNOS2.0 will allow users to connect to remote stations using the following hardware;

- AX.25 packet standard on TNCs that support KISS standards and trancievers in HF, VHF, and UHF frequencies at speeds from 300 to 9600B
- Clover and P-Mode (Pactor) modes on a DXP38 (and similar) TNC in HF bands
- Pactor (1, 2 and 3) modes on a SCS PTC II Pro (and similar) TNC in HF bands

Remote stations can also connect to JNOS2.0 on all hardware. The Clover and Pactor requires the custom HF server to support the protocols.

JNOS2.0 also supports TCP/IP protocol over AX.25 interconnections.

Configuring the TNC with standard KISS

[tbd]

Configuring the Halcomm DXP38 Modem

Even though I am using a DXP38, the DXP 4100 apparently has the same command set, so I would not be surprised if you could use it as well. Also, I don't see why one could not use a P38 (internal card) as well. If you have these other units, please try them out for me, and let me know how it works.

Connect the DXP38 to a free serial port, power the DXP38 off and on to make sure factory defaults have been initialized. Put the following lines in your autoexec.nos configuration file :

```
      attach asy ttyS2 - ax25 dxp38 4096 256 9600 f
      ifconfig dxp38 description "HF Clover Port"
```

Make sure you replace 'ttyS2' with the actual serial port being used.

The following two entries MUST exist at this time. This will not be necessary in future releases, but for now you need to do it this way. Failure to disable the 'ax25' and 'mailfor' beacons will mess up communications to the DXP unit.

```
      ax25 bcport dxp38 off
      mbox mport dxp38 off
```

The following entry :

```
      param dxp38 17 1
```

is only needed if you want the DXP put into P-Mode. If the entry is not present, the DXP will default to Clover mode.

Note : any DXP modem interface names MUST start with the 3 letters, 'dxp'. If you are using multiple units, you could call them 'dxp1', 'dxp2', and so on.

Configuring the SCS PTC II Pro Modem

Some of you may not like this, but before you use the SCS PTC with JNOS 2.0, you will need to initialize the modem using 'minicom', 'hyperterminal', or some other terminal interface program. In a tight fit, JNOS itself has a very simple terminal interface (use the 'tip <interface>' command), but in order to exit from it, you will have to restart JNOS itself.

To me it's not a big issue, I run alot of my modems that way. Once they are set, they stay that way for months at a time, and I don't see the need to have to constantly switch between modes just because I need to restart JNOS for a simple fix or configuration change.

Note : The JNOS 2.0 driver does NOT do any init codes, it assumes that the PTC is already in host mode. This driver uses JHOST1 mode, NOT the extended host mode at this time. Here's what I do :

1) connect the SCS to a serial port.

2) run 'minicom' on the SCS serial port, with baud rate of 115200 if you can, then go to the next step (don't type anything, just leave it).

3) power the SCS off and on, then hit ENTER key on 'minicom', the SCS will do an 'autobaud' sequence, and you should get the standard banners, followed by the 'cmd:' prompt.

4) with the 'cmd:' prompt in front of you, enter these commands (replace my callsign with your callsign of course) :

```
mycall ve4klm
maxerr 40
pduplex 0
listen 0
remote 0
box 0
```

5) finally, put the TNC into host mode. Issue the following command :

```
JHOST1
```

you will not get any prompt back. Don't hit any more keys, just exit out of the minicom or other terminal program, DO NOT reset the modem on the way out. Once you're back at the command prompt, you're ready to use it.

Put the following lines in your autoexec.nos configuration file :

```
attach asy ttyS2 - ax25 ptcpro 4096 256 115200 f
ifconfig ptcpro description "HF Pactor Port"
```

Note : the baud rate you specify in the above 'attach' command MUST match that which the SCS modem *autobauded* to if this is to work. That's not a big deal, but you should be aware of this.

Note : make sure you replace 'ttyS2' with the actual serial port being used.

The following two entries MUST exist at this time. This will not be necessary in future releases, but for now you need to do it this way. Failure to disable the 'ax25' and 'mailfor' beacons will mess up communications to the PTC modem.

```
      ax25 bcport ptcpro off
      mbox mport ptcpro off
```

The following entry :

```
      param ptcpro 17 1
```

MUST exist for the PTC modems. Without it, the changeovers will not work.

Note : any SCS interface names MUST start with the 3 letters, 'ptc'. If you are using multiple units, you could call them 'ptcpro', 'ptcII', and so on.

If you ever switch off the SCS PTC modem, you will need to detach the interface or quit JNOS 2.0, then rerun 'minicom' as per the instructions, but this time you will only need to run the 'JHOST1' command to put it back into hostmode.

The modem remembers all the other parameters (commands) you had put in, unless you do a factory reset type of thing, or unless you temporarily had used the PTC with some other software like AirMail or FBB, or whatever, then you will have to enter them all again.

Customized AX25 cross port digipeating rules

This is a New feature for JNOS 2.0c5 [Updated: June 21, 2005]- callsign substitution (version 2.0c5a)

```
 Basic crossport digi rules
 --------------------------
```

A few people have requested some forms of crossport digipeating that fall outside the usual JNOS crossport digipeating capabilities. For example, let's say we have 2 interfaces (ports) A and B, on the JNOS system whose callsign is VE4KLM. Let's say that A is an HF port, and that B is a VHF port for local packet.

The sysop wants to be able to define a digipeat rule that says if we get a packet on port A that looks like this :

```
       HFUSER>APRS,WIDE
```

then retransmit it out port B looking like this :

```
       HFUSER>APRS,VE4KLM*,WIDE
```

One can now use the new 'ax25 xdigi ...' command to support this type of behavior. In the above example, you would use a command something like the following :

```
    ax25 xdigi A B WIDE
```

* WARNING : IF you choose to use WIDEN-N for the digipeater callsign, you should disable the 'aprs flags +digi' feature. At this time, there is a conflict between the cross port digipeating rules and the experimental 'WIDEN-N' digipeater code built-in to NOSaprs.

I am sorry for any inconvenience this poses - I'm looking into it.[tbd]

NOTE : this is for ALL packet apps, not just APRS. I used APRS merely as an example, since requests came mostly from those that use APRS. These crossport digi rules apply to AX25, so there is no reason why a conventional packet user can not put in their own customized crossport digi rules to help them connect from one system to the next over a bewildering array of ax25 interfaces (ports).

you can add AS MANY RULES AS YOU WANT !!!

The type of ports have no bearing, as long as they are AX25 type ports - ie, axudp, axip, kiss serial links, attach asy - ax25.

Callsign substitution during crossport digi

Taking it one step further, someone had suggested that when the original packet get's retransmited, it would actually go out port B looking like this instead (note - WIDE changed to WIDE3-3):

```
       HFUSER>APRS,VE4KLM*,WIDE3-3
```

This would be cross port digipeating with callsign substitution, which is now implemented in version 2.0c5a (note the 'a'). To use the feature, use the same command as earlier in this document, but this time add the callsign you wanted substituted in - at the end of the command, for example :

```
    ax25 xdigi A B WIDE WIDE3-3
```

FEEDBACK is welcome !!!

This is just the start. If this feature does not quite do what you want, let me know please, and I'll make it so. Thank you.

Internet Ports

JNOS2.0 supports Internet connectivity.

Configuring the "tun" bridge

On a Linux platform with an existing IP stack, jnos provides the tun device to bridge between the system stack and the jnos stack. Tun is configured by the autoexec.nos startup process.

How to replace SLATTACH with TUN in JNOS 2.0

For those familiar with the "slattach" bridge, here is a tutorial by Maiko to present the differences in approach. Slattach is depricated in jnos2.0x[tbd]. I will explain how to do this, using configurations which I have run, or am currently running. Note, I use Slackware 9.1 as my linux platform. I mention that, just incase you are wondering why the path of my 'rc.local' file does not match that of your 'rc.local' file. I use Jnos 2.0 as my NOS platform.

I will do the 'before' verses 'after' approach using examples ...

```
2) Here is my existing SLIP connection (BEFORE)
   -------------------------------------------
```

Most existing JNOS systems interface to the linux networking using a slip connection between linux and JNOS. On the linux side the program, 'slattach', is run (usually from the /etc/rc.d/rc.local or similar file). On the JNOS side, an 'attach' command is placed in the autoexec.nos file to do the slip.

1) /etc/rc.d/rc.local

```
#
# Initialize the SLIP interface for the JNOS router
#
/usr/sbin/slattach -s 38400 -p slip /dev/ptyp0 &
#
sleep 3
#
/sbin/ifconfig sl0 192.168.1.130
/sbin/ifconfig sl0 netmask 255.255.255.224
/sbin/ifconfig sl0 pointopoint 192.168.1.131
/sbin/ifconfig sl0 mtu 1500
/sbin/ifconfig sl0 up
#
```

2) /jnos/autoexec.nos

```
#
attach asy ttyp0 - slip sl0 4096 256 38400
#
ifconfig sl0 ipaddress 192.168.1.131
ifconfig sl0 netmask 255.255.255.0
ifconfig sl0 mtu 1500
#
```

```
3) Here is my new TUN connection (AFTER)
   ------------------------------------
```

First of all, there is no entry in the 'rc.local' file anymore. You should either comment out or remove the SLIP related stuff. With tun, it is all done from the JNOS side, there is no need to run the 'slattach' anymore, it's all done in the autoexec.nos now :

```
#
#attach asy ttyp0 - slip sl0 4096 256 38400
#
attach tun tun0 1500 0
#
ifconfig tun0 ipaddress 192.168.1.131
ifconfig tun0 netmask 255.255.255.0
ifconfig tun0 mtu 1500
#
# sleep a second to make sure TUN driver is up
pause 1
#
# note, NOS creates the TUN device, so NOS needs to do a postconfig
shell ifconfig tun0 192.168.1.130 pointopoint 192.168.1.131 mtu 1500 up
#
```

A couple of things to note. The attach for the slip is commented out, and there is a new attach command for the tun device. In the above example, I've renamed the sl0 interface to tun0, but you don't have to of course. The three ifconfig lines don't change from before. There are two new lines, one to give the tun device a chance to come up (probably not necessary, but doesn't hurt), and secondly a shell out to linux to do a linux side configuration of the new tun device.

4) IMPORTANT CONCEPT TO UNDERSTAND

The tun device is created by NOS itself, when it does an open on the kernel tun device (/dev/net/tun). BEFORE you run JNOS, you won't be able to access the device. For example, using ifconfig to create and configure the device will not work, because it won't exist YET. Once JNOS is up and running, the device then becomes available.

Also, for packets to reach beyond the host stack then switch ip_forward must be turned "on". For example on a Linux FC-4 platform, to check the switch issue: '...# cat /proc/sys/net/ipv4/ip_forward' and expect to see "1" response. To set the switch if you get a "0" response use: '# echo 1 > /proc/sys/net/ipv4/ip_forward'. The switch is supposed to be retentive, thus this process need be done only once. If it proves to be a recurring problem, consider placing the "turn on" command in /etc/rc.d/... for each system boot.

I don't believe this to be a big issue for most installations, and it works fine for my setup here at home.

How to use IP-in-UDP encapsulation in JNOS 2.0

1) Introduction

Gone are the days where it was easy to pass 44 traffic over the internet, or where IPIP was a protocol that saw little hinderance. A lot of internet service providers and (from what I understand) a lot of router equipment by default now blocks this type of traffic. More accurately, many internet service providers block outgoing traffic originating from within their network IF the ip address is not from within their own allocation of numbers - meaning that IP packets having a source address of 44 will never make it out. Using both IPIP and IPUDP as the encapsulation methods can help us to overcome those obstacles, however for the new generation of home user, even IPIP will not work, since most home users are now behind a firewall or NAT router of some type. That leaves us with IPUDP, which is a simple, non complicated, and NOS friendly solution.

Now that cheap consumer firewall and NAT routers have appeared on the market, we can now make use of these nifty devices to improve AmprNet connectivity again, and at the same time keep our systems as secure as can be possible.

IPUDP is where IP packets are encapsulated in UDP datagrams. For all intensive purposes, IPUDP is the same as IPIP, except that IPUDP will pass through those cheap firewall or NAT routers, where in most cases IPIP will not.

K2MF (Barry), N1URO (Brian), and I discussed this some time ago. Barry should get the main credit on this one, he has implemented this on MFNOS, and then I followed up, by implementing it for JNOS 2.0.

2) Requirements

In order for you to make this work, you will need a static host that also has support for IPUDP. My static host is N1URO at this time, HOWEVER ...

I am actually in the process of actively getting the mirrorshades system to support this new protocol, so that IPUDP can be considered a formal gateway to which mirrorshades will route direct to as it does with IPIP, but that's still in the works (who knows - it may or may not happen).

I'm optimistic, and have submitted an RFC of sorts to the powers that be.

3) Configuration

When adding a 44 route, typically one uses something like this :

```
 route add 44.0/8 encap a.b.c.d
```

The above command implies that IP-in-IP encapsulation should be used. If you want to force the use of IP-in-UDP encapsulation instead, then simply add the word 'udp' to the end of the route command, for instance :

```
 route add 44.0/8 encap a.b.c.d udp
```

Our implementation of IP-in-UDP uses UDP port 94 (for both the source and destination ports). Just make sure your firewall or NAT router is configured properly to pass this port. It also makes sense to restrict access to that port to only those IPUDP hosts you link with. If your firewall or NAT router lets you do that, great ! If not, then you could use IPTABLES on linux to do it instead.

That's all there is it.

Date: Mon, 16 May 2005 21:42:31 -0500 (CDT)
From: maiko <...>
To: TAPR xNOS Mailing List <...>

PACKET ROUTING

Routing is a fairly complex subject made up of a few simple concepts. For the purposes of this document routing will be discussed in three segments. Some automation is implemented in routing, but largely routing for 44... net is static and manual.

For the process of fixing broken routes, various tools are handy. Tracing activity on I/O ports is available on screen with the F9 key, and trace data may be archived to file. Use the jnos command "trace" to start and stop tracing. On a Linux host, the "tcpdump" command displays activity on the host and can also display live or archive data on file. The "ping" command on both jnos and the host measure roundtrip time for packets to the remote host and back. The syntax and operation is somewhat different for command line, BBS, and sysop, commands. The jnos "hop check" and Linux host "traceroute" commands display the route available for communication to remote nodes.

LAN and NODE Configuration

It is common to place the computer hosting jnos on a LAN supported by the owner or an organization. For this document, the default configuration will be a private LAN using Class C IP addresses 192.168... with a bridge to Internet, and containing a small cluster of various computers. The ISP will be assumed to provide dynamic IP service.

WAN Configuration

There are several sources for Wide Area Network configuration. On a global scale, AMPR.net provides a wealth of connectivity. More locally as an example, MI-DRG administers a network in Michigan, USA for development and emergency preparedness. There are many others, but they all provide information and coordination for the purpose of Wide Area Networking. These networks also have the characteristic of combining point-to-point connectivity via both Amateur RF and public Internet.

Static IP, VPN, ENCAP IP, DNS

RF Path Configuration

Amateur radio introduces some unique considerations on routing. Frequency coordination, prorogation of signal, rig tuning, AX.25 vs TCP/IP, and other things will be discussed here.

SERVER / CLIENT SERVICES

A list of servers found in JNOS may be found from the “start” command output. The list is effected by the compilation options when the executable is built. This section covers configuration requirements for specifically started services.

APRS Server

[tbd]

AX25 Server & BBS

The AX.25 server is an integral component of the BBS. Upon using packet to enter the jnos application, the login completes by placing the user in the set of BBS services. No destinction is made between AX.25 and TCP access to the bbs, the command set is the same.

The "start AX25" command enables enables packet services. Each port capable of AX.25 communication becomes a path to the BBS. [tbd]

CONVerse Server

[tbd]

FINGER User Information Server

[tbd]

HTTP Server

[tbd] Notes:

106.2 To start an http server, use the start command. The syntax is:

106.2.1 start http [port#] [drive] [rootdir] The default http server port is 80, the default disk drive is C, and the default rootdir is /wwwroot. This root directory MUST contain a file called "root.htm". If a client specifies an explicit path to a directory below this root, a reference to "welcome.htm" in that directory is assumed. If welcome.htm does not exist, a directory listing is prepared and sent. "welcome.nhd" can be used instead of "welcome.htm", to cause the file contents to be sent without headers. This feature can be utilized to do unusual things like automatic redirection (probably not necessary anymore with today's intelligent browsers). The JNOS default http root dir can be changed by defining a new value for HttpDir in nos.cfg (and starting JNOS with -f nos.cfg).

106.2.2 UNIX note: the drive letter is required, but ignored! Also, the recognized filename suffix is ".html", not ".htm".

```
106.2.3 Client URL:     JNOS file fetched:

106.2.4 http://your.system.name/    <rootdir>/root.htm

106.2.5 http://your.system.name/dir <rootdir>/dir/welcome.htm (if exists)
          otherwise, a listing of <rootdir>/dir contents

106.2.6 http://your.system.name/X/file  <rootdir>/X/file
```

106.3 Multiple http servers, each on a different port, may be started, up to a limit of 5 (established at compile time by MAXPORTS). The stardard port is

80, but a non-standard one can be given in a URL, such as "http://n3yco.ampr.org:99/".

106.4 The JNOS /spool directory must contain a file called "access.www" which controls HTTP access rights. Lines either specify a directory path at and below which access is denied, or specify a {path, realm, encoded_user:passwd} sequence to which access is permitted only if the encoding matches that provided by the client. This file MUST exist, even if empty, to allow any http accesses. Note that access.www limits what can be specified on your system by a URL from an http client. It does NOT limit access to included files specified in local html files (but see 'http absinclude').

106.4.1 Example:

106.4.2 #type1: path-relative-to-wwwroot   realm   encoded-"user:password"

106.4.3 /pub/jnos/www/unzipped/secret SecretPlace Z3Vlc3Q6aGVsbG8=

106.4.4 #type2: directory at or below which access is denied:

106.4.5 /pub/jnos/private

106.4.6 To produce an encoding of a "userid:password" combination to be used in access.www, use the base64.exe utility (produced by 'make base64' in the JNOS source directory): base64 userid:password > encoded.txt   Then edit the resulting file to yield a line for insertion into access.www.

106.5 If JNOS http was compiled with HTTP_EXTLOG #define'd, then by default the directory /wwwlogs will contain a record of accesses. This file, created daily, will grow VERY large if your server is very busy! See http://mvmpc9.ciw.uni-karlsruhe.de for information on Karl-Heinz Weiss' cleanlog utility. The log directory can be changed by defining a new value for HLogsDir in nos.cfg (and starting JNOS with -f nos.cfg).

106.6 Counters are maintained in /wwwstats, but this directory can be changed by defining a new value for HttpStatsDir in nos.cfg (and yes, starting JNOS with -f nos.cfg).

106.7 Server Side Include (SSI) support is patterned after those in the NCSA httpd. An SSI has the form: <!--# cmdname tag="value" --> echo var="s" displays the value associated with variable <s>:

106.7.1 DATE_LOCAL   - Current localtime

106.7.2 DATE_GMT - Current time in GMT

106.7.3 HOSTNAME - Server's hostname

106.7.4 DOCUMENT_URI - Resource Identifier of the current doc.

106.7.5 DOCUMENT_NAME- File name of the current doc.

106.7.6 LAST_MODIFIED- Modified date/time of the current doc.

106.7.7 TOTAL_HITS   - Total hits on this server.

106.7.8 REQ_FROM - From: header of requestor (if available)

106.7.9 REQ_REFERER - Referring URL (if given by browser)

106.7.10 REQ_AGENT   - Client browser's name (if given)

107 echo dcount="filename" displays the named counter.

108 echo icount="filename" increment and displays the named counter.

109 echo scount="filename" increment the named counter.

110 include file="/absolute/path" inserts the referenced file or dir, provided "http absinclude" is set on.

111 include virtual="path" inserts the referenced file or dir, where "path" is relative to the http root dir. "path" must end in '/' to be interpreted as a directory.

112 exec cgi="name?arglist" executes the cgi "name" which must be compiled into JNOS. Two CGIs are presently available:

113 counter.xbm (if CGI_XBM_COUNTER was #define'd) and postlog (if CGI_POSTLOG was #define'd).

114 counter.xbm produces an X-bitmap display of the counter name provided as the argument. The counter is maintained in /wwwstats. Additional arguments are "inv" for inverting the display colors, and "noinc" to not increment the counter before it is displayed. Because this counter is returned as a bitmap it should be referenced as an image so a browser will handle it correctly: <IMG src="/counter.xbm?tcount.dat+noinc">

115 postlog demos the POST html command; see http.c for details.

116 Note that a URL may specify a CGI as if it were a filename under the root directory. Example: http://localhost/counter.xbm?cntrname.ext

117 7) More information on writing html documents can be found at:

117.1 http://www.visualogic.com/http_1.0/index.html

MAIL Forwarding Server

Mail Forwarding is an optional service for jnos. Mail forwarding may be a manual and or automatic server/client function compiled into jnos per #defined parameters [tbd]. There are two flavors, one remaining on the air over RF paths using AX.25 standards, and one using Internet SMTP services.

Any HF forwarding of mail and bulletins between this version of JNOS and any F6FBB system that I have tested with, has been almost flawless, and worked quite well to date. I'm very pleased with the results.

HF forwarding with AirMail and Winlink 2000 systems still needs alot of work. My efforts to forward traffic with these HF systems has shown mixed results. I have actually successfully forwarded mail and bulletins to these systems on several occassions, but not consistently.

One goal of this project is to see JNOS to JNOS forwarding over HF, using these new modes that I have added to JNOS 2.0. That's what I would really like to see in the end. If you want to arrange live HF testing with me, then please contact me, and let's try it out. Thank you.

Configure to partner(s) via AX.25

This section of the NOS docs deals with the intricacies of mail forwarding. You should read and understand this documentation thoroughly before attempting to forward mail through your NOS box to the AX.25 BBS world, otherwise you might grossly misconfigure your system and be the unhappy recipient of flames from BBS sysops.

This section does NOT deal with the minutiae of the mailbox and its various commands; it assumes that you understand concepts such as user areas (both public and private) and how to list and send mail. If you need help with these, please look elsewhere in the NOS docs.

1. Introduction

Apart from the usual domain.txt and other files necessary for ordinary functionality of NOS, three files are important in the mail forwarding process. These are :

1. spool/forward.bbs,
2. alias
3. spool/rewrite

The contents of these will now be addressed individually.

2. /spool/forward.bbs

The forwarding partners are defined in '/jnos/spool/forward.bbs' file. The detail of the forward file describes the actions taken by NOS in forwarding to remote partners. The file contains a series of forwarding records, each record being separated by a line containing two or more hyphens. The template for each forwarding record is:

```
●  BBS-callsign time-specs poll-flag
●  Connection spec with route
●  Connection command script      <zero or more lines>
●  List of areas to be forwarded  <one per line>
●  ------------                   <end of record>
```

File content example to forward two areas:

```
-------------
ve4wws 0007 P
ax25 430 ve4wws
ve4wws
mb
-------------
```

2.1. BBS callsign

This is simply the ordinary call of the remote BBS. A typical (but not random!) entry might be simply the line:

```
sm0rgv
```

The callsign may be followed, on the same line, by a comma separated list of valid intervals when forwarding is to take place. Each valid interval is a four digit number: the first two digits are the beginning hour of the valid interval, the last two digits are the final hour of the valid interval. For example, if the first line of a forwarding record looks like:

```
sm0rgv 0006,1414
```

then forwarding to sm0rgv will take place only during hours numbered 00, 01, 02, 03, 04, 05, 06 and 14. Ticks of the mbox timer outside of these times will not cause mail to be forwarded to sm0rgv. The default interval for forwarding is 0023.

If you desire to force a connect to occur, even if we have no traffic to transfer, then use the poll flag, P, in addition to any time specs.

2.2. Connection route

This is the method by which communication is to be established with the remote BBS. The first token on the line is the type of protocol to be used. This is one of ax25, netrom, tcp, hf, or file. Following this is whatever further information the chosen protocol requires to make the connection, perhaps a digipeater route. An example connection route for a simple ax25 connection on interface ax0 is:

```
ax25 ax0 g3dlh
```

A more complex one is:

```
ax25 ax0 g3dlh via k5arh uv-3 kb5ogn-1
```

JNOS can forward to an export file. To specify forwarding to a file, use this syntax in the connect-line:

```
file <path_to_export file>
```

2.3. Connection command script

Connection commands may, optionally, follow the connection route. These take the form of a single character command followed by suitable argument strings. Supported commands are:

- .text text (with CR suffix) is sent to remote.
- any search string is reset.
- ! eschew FBB compression
- +text establish a search string.
- @NN read a line with timeout of NN secs, and fail unless it contains the search text established with the + command.
- * NN read and discard lines until a line matching desired string is encountered or until NN secs has elapsed.
- # comment_text comment text is ignored

For example, suppose that we wish to establish a netrom connection with sm0rgv-2, through the netrom node #sth67. Then the connection route and connection command portion of the record would look like:

```
netrom #sth67
.c sm0rgv-2
```

Another example, this time we use ax.25 protocol to connect with a netrom node, k7uyx-1, and then connect to another netrom node:

```
ax25 ax0 k7uyx-1    <- initial connection to netrom node
```

```
.c rlimb            <- ask for a netrom connect from this node
+Connected          <- if we don't get this, things went wrong
@60                 <- maximum one minute wait !
```

If the station is reached through digipeating, then the digipeater callsigns should either be specified in the connect line (described above) or be in the ax25 route to the destination callsign. For example, if you wish to forward traffic to r2nod, implicitly using r1nod as a digipeater, then you should have the line:

```
ax25 route add r2nod r1nod
```

in your autoexec.nos file.

2.4. List of areas to be forwarded

This is a list, one per line, of entries in the /spool/mail directory which will be forwarded to the remote BBS. An entry of the form:

```
r1nod
```

will cause forward server to scan the file /spool/mail/r1nod.txt for unread messages. Any such messages are sent to the remote BBS and deleted from the file.

One can also forward user areas using this mechanism. To do this, simply place a line containing the name of the area in the record. So, to forward amsat bulletins to the BBS, one would have a line:

```
amsat
```

This will search the /spool/mail/amsat.txt file; any messages contained therein which have not been forwarded to the BBS in question will be forwarded. They will NOT be deleted. The determining factor as to whether or not entries are deleted is that if the filename is present in the /spool/areas file, then there is NO deletion, otherwise there is.

Please note that ONLY FILES IN /spool/mail are checked. In particular, the outbound SMTP mail queue is NOT checked.

2.4.1 Changing the recipient address

Normally, NOS uses the information in the To: header line to determine the parameters used by the "S" command during BBS forwarding. Occasionally, one might want to change this behavior. In this case, a line of the form:

```
area  newaddress
```

in the list of areas to be forwarded will replace the originally typed destination with the string newaddress instead.

3. /alias

The alias file is used to map LOCAL names to other names, which may be either local or remote; additionally, from a single input message, the alias file permits one to produce multiple output messages. Thus, typical uses for the /alias file are:

- converting one local name to another,

- converting a local name to a remote name,
- exploding a mail message so that it is passed on to several recipients.

The format of a record in the alias file requires no separation between records in the /alias file other than a newline.

The aliasname is a local username; that is, it does not contain an "@" symbol. When the alias file is processed, if the destination of the message matches precisely the aliasname, then the mail is redirected to ALL of the aliased recipients.

Scanning of the /alias file is performed by the SMTP server. The SMTP timer (which controls the SMTP client) is kicked whenever the mailbox or SMTP server queues something for delivery by SMTP. Mail transport within a single NOS system is performed through the SMTP client/server mechanism. The result of these facts is that as soon as a piece of mail is entered to the mailbox, the SMTP client is kicked and attempts to deliver the mail (which has already been scanned by the rewrite mechanism - see below). If the mail is local to the NOS system (i.e. no "@" sign in the address), then the /alias file will be scanned and the name mappings take place.

A few lines in the /alias file might look something like:

```
bdale   bdale@n3eua
local   fred@k0yum bdale@n3eua bill@ai0c.ampr.org n5op@n5op jim@k0jtz n0esg@n0esg
g4bki   g4bki@gb7bil.ampr.org
```

The system MUST know how to deliver traffic to each of the individual addresses in the style in which they are entered in the /alias file. If the system does not know how to deliver one of the new addresses, then it will send it to the SMTP gateway station defined by the 'smtp gateway' command.

Note that it is reasonable, and sometimes desirable, to have alias records of the form:

```
area    area dest1 dest2 ...
```

As the /alias file is scanned only once (see below), this does not result in an infinite recursion.

4. /spool/rewrite

The rewrite file is used to perform a one-to-one mapping between destination addresses as received by NOS and destination addresses as actually used by NOS. Each record within the rewrite file comprises a single line, containing either two or three entries separated by spaces.

- The first field is the template field
- If a destination address matches the template, the address is replaced by the second field.
- The third field, which is optional, is the single letter "r", which, if present, tells NOS to rescan the rewrite file, using the new destination address to attempt to match against the templates.

Special characters:

- A template may contain one or more "*". These stand for a match of any number of characters (including zero).
- In the second field, the character "$", followed by a single digit in the range

1 to 9, represents the string that matched the respective "*" in the template.

By way of example, suppose that there is a line in the rewrite file which looks like:

```
*@* $1%$2@g1emm.ampr.org
```

Then, any traffic reaching the system through the mailbox or the SMTP server, but which is supposed to go to a remote system, will be redirected to go through g1emm.ampr.org. Suppose that a user logs on, and sends a message to n0gbe@nq0i. Then the rewrite file attempts to match "n0gbe@nq0i" against the entry *@*. It matches, and assigns $1 the value n0gbe, and $2 the value nq0i. The mail file as written to the disk will no longer be to n0gbe@nq0i, but, rather, to n0gbe%nq0i@g1emm.ampr.org. [The nomenclature station1%station2@station3 means the final destination is station1@station2, and this traffic is to be routed through the gateway station3.]

As soon as a template match is found, the conversion is performed and scanning is stopped, unless the third "r" field is present, in which case scanning restarts from the top of the file.

Note: It is a good idea to have a line of the form:

```
*@*.ampr.org $1@$2.ampr.org
```

at the beginning of your rewrite file. This will cause all amprnet traffic to be caught early in the rewrite scan, and no further scanning (and, hence, no unexpected substitutions) will take place.

Configure to partner(s) via hf

File content is similar to how they are defined for regular ax25 connects. The only real difference is that instead of using 'ax25' or 'c', you would use the 'h' command, ie:

```
-------------
ve4wws
ax25 430 ve4wws
ve4wws
mb
-------------
wu3v
h ptcpro wu3v
wu3v
packet
-------------
ve4gls
h dxp38 ve4gls
ve4gls
   -------------
```

Also, your rewrite file may possibly have some entries specific to your forwarding partners. For example, in mine I have to have something similar to these below (if sp wu3v @ wu3v is to work).

```
*@ve4wws* ve4wws
*@wu3v* wu3v
```

```
*@ve4gls* ve4gls
```

Configure to WinLink 2000 (WL2K) network via telpac

1) This feature is tested OK in March 2006 for use and requires a source code revision patch installed against jnos2.0d1 using the compilation methods of the next major section in this manual. Click here to download the modified 'forward.c' source, put it into your JNOS source directory, then do the 'make' command to generate a new JNOS binary. You MUST do this or else it will not work! As per jnos2.0e the patch is included in the distribution.

The telpac connections seem to be like the early XFBB sessions, only the CR (carriage return) is used at the end of line, which can be dealt with easily, since JNOS already has a 'cronly' option that can be passed in the forward.bbs file.

So why not just use the 'cronly' option? Because it seems that WL2K also does not like FA proposals, so I needed a way to force JNOS to use FB proposals. As a result, I created a new 'telpac' option to deal with both issues.

2) Add the following entry to your '/jnos/spool/forward.bbs' file:

```
-------------
k4cjx
tcp <IP> 12001 telpac
@
@
@
+Callsign :
@
..<wl2kid>
+Password :
@
.<wl2kpwd>
k4cjx
-------------
```

Where you MUST make the following substitutions:

- "kc4cjx" is the winlink Radio Message Server (formerly called PMBO)to forward to.
- "<IP>" is the server IP address, available from [tbd].
- "<wl2kid>" is your own winlink system login ID.
- "<wl2kpwd>" is your own password for the winlink network.

!!* try and keep experimental forwarding to a minimum, since it is a *production* system used worldwide ... Please show Steve (K4CJX) some courtesy and change the k4cjx to another winlink server you want to forward with. If you don't have a particular winlink server in mind, available servers are listed in the Telpac documents at Winlink site.

3) In my '/jnos/spool/rewrite' file, I added the following (example) entry :

```
*@k4cjx* k4cjx
```

Where you must make the following substitutions:

- "k4cjx" is the winlink server to forward to.

Scanning procedure

The two files which are used to determine the disposition of traffic are scanned under slightly different circumstances. Note that neither the /alias nor the /spool/rewrite scan makes any actual changes to the contents of the traffic. In particular, the To: field remains exactly as it was first entered into the system, with one exception: if the message is to be stored into a local area, and if the To: field ends with @our_hostname, then this suffix is removed, and any rightmost % symbol is changed to an @ symbol, to facilitate forwarding to ax.25 networks.

There are four possible entry routes for traffic into the system:

- SMTP
- through the mailbox by a user
- through the mailbox by a BBS
- via an external program (like BM) or creation of the files manually

NOS determines if a piece of traffic was entered into the system by a BBS by looking for a BBS system ID (like the "[JNOS-IHM$]" block issued by NOS) on the incoming connection prior to messages being uploaded.

5.1. Traffic received by SMTP server

1. The rewrite file is scanned and any changes applied (unless the traffic was received through the local mailbox; in that case, this step does not occur);
2. If the traffic appears to be local then the alias file is scanned and any changes or explosions applied.
3. Any copies local to the system are delivered; copies for remote delivery are placed in the SMTP queue.

5.2. Traffic received by mailbox from user

1. The rewrite file is scanned and any changes applied;
2. The traffic is passed to the SMTP client.

5.3. Traffic received by mailbox from BBS

1. The rewrite file is scanned and any changes applied;
2. The traffic is passed to the SMTP client.

5.4. Traffic entered by external mechanism

1. No scanning occurs;
2. The traffic is passed to the SMTP client.

6. Headers

Appropriate RFC-822 headers are added to all incoming traffic. Traffic entering through the mailbox receives a full complement of RFC-822 headers; traffic coming through the SMTP server has only a "Received:" header applied. On forwarding to a BBS, if an item of traffic contains BBS R: headers, the RFC-822 header is converted to an appropriate R: line at the time that NOS forwards the message. (This change only occurs for BBS forwarding; forwarding by SMTP retains the RFC-822 headers.)

7. Bulletin Identifiers (BIDs)

The AX.25 BBS system has evolved a reasonably efficient way of reducing overhead when forwarding bulletins. When a bulletin is originated on a BBS, it is given a unique bulletin identifier (BID). This BID should (theoretically) travel with the bulletin, and should never be changed during the distribution of the bulletin. Each system keeps track of all received BIDs. If a forwarding station wishes to forward a bulletin to a BBS, then the receiving station checks its local list of known BIDs and informs the transmitting station if it already possesses the bulletin in question. The NOS mailbox conforms to this protocol. Received BIDs are stored in the file /spool/history, and are encoded in the Message-ID: header line of the message by NOS. Messages forwarded from areas listed in the /areas file will have their BID (re)generated from the Message-ID: line. Note that ALL messages from public areas are forwarded with a BID, whether or not the message was produced with the "SB" command. Like other BBSes, NOS will inform a transmitting station not to transmit a bulletin if it is one that NOS already has locally; likewise, it understands similar messages from other stations to which it tries to forward.

Note that the BID mechanism is not a part of the SMTP world. If you are forwarding bulletins through SMTP, there is no certain mechanism by which the receiving station can reject the attempted delivery of a bulletin, even if it already exists on the recipient system. JNOS will generate a bid that will match the bids that other JNOS systems generate from the same message, but this convention is not universally followed. (Note that another possible workaround is to deliver bulletins to TCP/IP stations using TCP instead of SMTP. Alternatively, one could use NNTP, as NNTP commands utilize the Message-ID: line, from which the BID is derived.) The BID is preserved no matter which mechanism is used to deliver the bulletin.

8. Traffic in practice

Now, the big question is, how does one set up these various files to perform intelligent manipulation of mail? A number of examples follow. Note that, often, there is more than one way to accomplish an objective. The following are merely examples (and not necessarily the most efficient method possible for any given case). The format used will be:

typed destination -> intended destination

followed by the necessary entries in the alias (/alias), rewrite (/spool/rewrite) and forwarding (/spool/forward.bbs) files.

8.1. Using familiar names - SMTP destination

```
bdale -> bdale@n3eua.ampr.org

alias:
bdale   bdale@n3eua.ampr.org

rewrite:
forward:
```

8.2. Exploding local mail

```
sysops -> nq0i, n5op@n5op.ampr.org

alias:
sysops  nq0i n5op@n5op@ampr.org
```

```
rewrite:
forward:
```

8.3. Using familiar names - BBS forwarding

```
g4bki -> g4bki@gb7bil.2712.gbr.eu, to be forwarded by ai0c

alias:
rewrite:
forward:
ai0c
ax25 ax1 ai0c
g4bki g4bki@gb7bil.2712.gbr.eu
ai0c
```

8.4. Handling incoming bulletins by subject

```
tcpip@* -> nq0i, tcpip, bdale@n3eua.ampr.org, ai0c@ai0c [a BBS]

alias:
tcpip   nq0i tcpip bdale@n3eua.ampr.org ai0c

rewrite:
tcpip@* tcpip

forward:
ai0c
ax25 ax1 ai0c
ai0c
```

Let's walk through the above example. An incoming item comes in addressed to TCPIP@ALLUS. A scan is made through the rewrite file, and a match is found. The item is redirected to tcpip. The alias file is scanned; a total of four copies of the item exist after this, three in local areas tcpip, nq0i and ai0c, and one on the SMTP queue (for bdale@n3eua.ampr.org). When the mailbox timer next ticks, the mail in the local ai0c area will be forwarded on the ax1 interface to ai0c.

8.5. Routing based on Hierarchical addressing

```
Wyoming -> KE7VS (SMTP)
Nebraska -> AG0N (BBS over the NETROM, NETROM ID WNBBS)
Europe -> W0LJF (BBS over AX.25)

alias:
rewrite:
*.noam    $1.na r
*.us      $1.usa.na r
*.usa     $1.usa.na r

*.ne      $1.ne.usa.na r
*.wy      $1.wy.usa.na r

*@*.wy.usa.na $1%$2.wy.usa.na@ke7vs
```

```
*.ne.usa.na   ag0n
*.eu      w0ljf

forward:
ag0n
netrom ax0 wnbbs
ag0n
----------
w0ljf
ax25 ax1 w0ljf
w0ljf
----------
```

Why is the example rewrite file apparently so complicated? This is to handle poorly constructed hierarchical addresses in a reasonable way. A full U.S. hierarchical address has the form: callsign@BBS.#localid.state.usa.na. Many states have no #localid field. In the example rewrite file above, the first three lines convert non-standard, but frequently used, U.S. designators to the more standard format. It is common for users not to use a full hierarchical address if the destination is relatively local. For example, a user might easily use only .wy instead of the full .wy.usa.na if he is geographically close to Wyoming. The second grouping of two lines handles this problem. Note the third, "r", field in all the entries so far.

The remainder of the file handles properly formatted hierarchical addresses.

8.6. General bulletin handling

The details of bulletin handling will vary somewhat from place to place, as there are several distinct styles of bulletin handling currently in use in the AX.25 BBS world. In general, it is necessary to arrange one's system so that it accepts bulletins from BBSes, forwards them to one or more stations, and also handles intelligently bulletins input by users into NOS.

Suppose that we wish to handle bulletins @JUNK. We are to deposit them locally in the junk area, and also forward to BBS g4bki. We also know that we generally receive @JUNK bulletins from g4amj, a local BBS which handles much bulletin traffic.

```
alias:
rewrite:
*@junk   junk

forward:
g4bki
ax25 ax1 g4bki
g4bki
junk
----------
g4amj
ax25 ax1 g4amj
g4amj
junk
----------
```

All incoming @JUNK traffic is written to the junk area (which should be an explicit entry in the /spool/areas file). Each tick of the mailbox timer, NOS scans the junk area for traffic not forwarded to g4bki or g4amj and attempts to deliver unforwarded bulletins. Usually, g4amj will respond with a "Have it" message and the bulletin will not be forwarded. Any bulletins @JUNK deposited locally by users will automatically be sent to both g4bki and g4amj.

Questions and Answers

Q. Under what circumstances does NOS request reverse forwarding from a BBS?

A. NOS requests a reverse forward after completing any forwards of its own to the BBS. If no traffic was queued for a given BBS, then no connection is attempted, so no reverse forward request is issued, UNLESS the P (poll) flag is in a forward.bbs time-spec field. Reverse forwarding can be forced by specifying the BBS callsign in upper case, i.e., mb kick N5KNX

Q. What kinds of message types does the NOS mbox support?

A. Basically, NOS supports all two letter commands starting with an "S". If the mailbox has not received an SID banner (the "[NOS-H$]") from a connected station, then an SF command will forward the current message to the address specified on the command line. The SR command will send a reply to the current message. One can also issue the command "SR <number>", where <number> is the number of the message to which you want to generate a reply. All other variations cause an X-BBS-Msg-Type: header to be added to the message. When a message with such a line is forwarded to a BBS, it is sent to the BBS with the appropriate message type as the second letter in the "S" command to the BBS.

If NOS has received a valid SID, then ALL S commands are handled by the X-BBS-Msg-Type: mechanism outlined above.

Logic map of the mailbox

[tbd] [insert "flow" dwg here...]

FTP Services

[tbd]

NET/ROM services

originated in the 1980's with WA8DED and W6ISU. Configuration begins with autoexec.nos and continues with adjustments to files that support netrom services. First 'attach netrom' in autoexec.nos, to permit jnos to pass net/rom frames at level 3 (network). This is often done so the frames are successfully recognised even when no local process is involved.

Next, if local processing is desired, then configure netrom and 'start netrom' server to handle level 4 (transport) frames.. Example configuration is: [tbd]

Manual entry of neighbors is done [tbd]

NET/ROM services may create a “./netrom.sav” file to preserve netrom nodes for later use. To create the file use 'netrom save' and to restore the data use 'netrom load” commands at the admin session. The load operation may be built into autoexec.nos at administrator discression.

POP Mail services

[tbd]

REMOTE Services

[tbd]

SMTP configuration notes

Hi all,

Well it took me long enough to figure this out, but here is a slightly lengthy, but hopefully useful explanation of how JNOS does mail delivery to the outside world.

I was forced to go through this exercise because I could not get JNOS to deliver mail to any point past my ISP's subnet.

1) The 'smtp ga' option.

Use a domain name and not an IP address. Many ISP now rotate their smtp servers on a regular basis. For example, 'smtp.a.b.com' could easily point to a different server every few times it gets resolved by JNOS for mail delivery. My isp rotates among mx1, mx2, mx3, etc, and it can get interesting how some ISPs do this stuff.

For example, instead of using this :

```
    smtp ga 1.2.3.4
```

Use a domain name instead, for example :

```
    smtp ga smtp.someisp.com
```

2) My ISP does NOT allow SMTP to leave their subnet. If I want to have mail delivered outside my ISP (to the rest of the world), I have no choice but to send it to the smtp server of my ISP.

To FORCE JNOS to send mail to the 'smtp ga' entry, you MUST switch the 'smtp usemx' option OFF !!! In other words, make sure :

```
     smtp usemx off
```

This causes ALL mail deliveries to go via the default gateway that you have specified in the 'smtp ga' command.

Note : One possible issue with the above, is if my ISP does not like the 'HELO ve4klm.ampr.org' it sees coming from my box that itself actually resolves to my commerical ip address. Does not seem to be a problem for my ISP anyways, and if you look at the headers at the destination it shows coming FROM 've4klm.ampr.org', followed by the *real* domain name and ip address. Some destination mail filters may not like that, your success will probably vary.

Hopefully the above paragraph makes sense to most of you. Sorry if it doesn't :-)

IF I were to have 'smtp usemx on', then MOST (not all) of my emails will try and be delivered by JNOS using direct SMTP from my box to the end user. BUT - My ISP blocks outgoing SMTP past their border routers, so the mail will never get delivered.

IF 'smtp usemx on' is being used, delivery will ONLY occur if there is no MX record found for the recipient's email address. Delivery in that case of course would be via the ISP default smtp gateway. This is not consistent, so I would not count on the 'usemx' option to be reliable in this case.

Summarizing what I do to ensure that JNOS can deliver outgoing mail. I have the following entries defined in my autoexec.nos configuration :

a) make sure I specify a domain name and not an ip address for the smtp server of my ISP, for example :

```
smtp  ga  smtp.myisp.somewhere.com
```

b) make sure that I do NOT use MX records, for example :

```
smtp  usemx  off
```

I hope this is helpful to some. Comments and corrections welcome !

TELNET Server & BBS

The telnet server is an integral component of jnos. Upon using telnet to enter the jnos application, the telnet login completes by providing the owner with sysop services and other users with BBS services. No destinction is made between AX.25 and TCP access to the BBS, the command set is the same.

The "start telnet" enables access to JNOS. Each port capable of TCP/IP communication becomes a path for user access. [tbd]

TIP SESSION

[tbd]

ttylink Services

[tbd]

SOURCE CODE COMPILATION

Greetings JNOS users, I present to you JNOS 2.0d - now compilable up to 'gcc version 4.0.1'

1) There is NO incremental update from previous versions, too much has changed, and a mass compile is required anyways.

Put the compressed tar file into an empty directory, then from within that directory, issue the following commands :

```
    gunzip jnos2.tar.gz
    tar xvf jnos2.tar
    cd jnos2
```

2) Edit the config.h and set your preferences. For the more experienced people, you can provide your own config.h, but I suggest you make a backup copy of the one included in this source, ie :

```
    cp config.h config.h.dist
```

IMPORTANT: Previous versions of Makefile contained the -DAPRS -DJNOSAPRS if users wanted to include APRS services. That is no longer in use. Every thing is now defined in the config.h header file (where it belongs).

3) There is only one makefile now, so all that remains to be done is :

```
    make clean
    make
```

Please note that warnings will happen, I know about them already, so please ignore them. There is NO need to run 'make depend', even though it does work now, I never use it, and don't see the need right now.

4) That's it, if all goes well, you should have a new 'jnos' binary.

CONFIGURATION METHODOLOGY

Detail application configuration is done by setting various switches at compile time. The file config.h contains the array of configuration flags, and files.c contain the default file locations used by the compiler. As distributed these files suggest a starting point. There are also other examples provided with jnos20d distribution to be considered during tuning, they are:

- config.h – typical jnos2 pre-compiled distributions (may not be exact) at VE4KLM
- asyconf.h - another set used by WG7J
- distconf.h - the set used for WG7J precompiled version: [tbd]
- gwconfig.h - the set used for wg7j.ece.orst.edu gateway
- bbsconf.h - configuration for wa7tas.or.usa.noam BBS
- unixconf.h - n5knx full Linux configuration for Jnos
- users.h - an end user configuration suggestion

- homeslip.h - home.wg7j.ampr.org site configuration

What follows is taken from the basic file to demonstrate settable options for compilation. It is suggested that a new owner might initially compile with the distribution config.h if only to validate development on the host system. Once the development process is proven, then begin the steps leading to the appropriate options for use on the target install. It follows that cataloging various versions of config.h is recommended by this author.

Application SERVICES Structure

The following options presented in alphabetical order in groups of defined [ON], undef [OFF], logic, and comment text, from “config.h”. Some descriptive text is provided by developers in keeping with good documentation practices.

```
#define ALLCMD             /* include dump,fkey,info,mail,motd,record,tail,
#define APRSD
#define ASY                /* Asynch driver code */
#define ATCMD              /* Include timed 'at' execution */
#define AX25               /* Ax.25 support */
#define AX25SERVER         /* Ax.25 server */
#define AX25SESSION        /* Connect and (if SPLITSCREEN) split commands */
#define AX25_XDIGI         /* May 2005, new custom cross port digi rules */
#define AXIP               /* digipeater via ip port 93 interface */
#define BBSSESSION         /* bbs (same as telnet localhost) */
#define _CONFIG_H
#define DEFAULT_SIGNATURE   /* 23Jul2005, Maiko, default sig for all mail */
#define DIRSESSION         /* dir cmd */
#define DOMAINSERVER  /* Udp Domain Name Server */
#define DOSCMD             /* Include cd,copy,del,mkdir,pwd,ren,rmdir commands */
#define DQUERYSESSION /* Include "domain query" cmd */
#define DYNGWROUTES   /* Enable use of dynamic gateways in 'route add' */
#define ED                 /* editor uses Unix ed syntax; OK for remote sysops. ~13KB */
#define ENCAP              /* Include IP encapsulation code */
#define ETHER              /* Generic Ethernet code */
#define EXPEDITE_BBS_CMD /* Use MD5 and net.rc to autologin console to bbs */
#define EXPIRY             /* Include message and bid expiry */
#define FBBCMP             /* add FBB LZH-Compressed forwarding code */
#define FBBFWD             /* add enhanced FBB Forwarding code (no compression) */
#define FILECMDS           /* Include D,U,W,Z commands */
#define FINGERSERVER    /* Tcp finger server */
#define FINGERSESSION   /* finger cmd */
#define FOQ_CMDS           /* Include Finger, Operator, Query
```

```
#define FTP_REGET              /* add RFC959 ftp client reget&restart cmds */
#define FTP_RENAME             /* add RFC959 ftp client rename command */
#define FTP_RESUME             /* add Jnos ftp client resume&rput cmds */
#define FTPSERVER              /* Tcp ftp server */
#define FTPSESSION             /* ftp,abort,ftype, and iff LZW: ftpclzw,ftpslzw cmds */
#define GATECMDS               /* Include gateway releated commands C,E,N,NR,P,PI,T */
#define GWTRACE                /* Log all gateway connects to the logfile */
#define HFDD                   /* 15Jan2005, Not everyone will want or be able to do HF */
#define HOPCHECKSESSION             /* IP path tracing command */
#define IBUFSIZE  2048         /* Size of interrupt buffers */
#define INP3                   /* 22Sep2005, Maiko, Support some INP3 stuff */
#define JNOSAPRS
#define KISS                   /* Multidrop KISS TNC code for Multiport tnc */
#define LOCK                   /* Include keyboard locking */
#define LOOKSESSION            /* follow user activity on the bbs */
#define LZW                    /* LZW-compressed sockets */
#define MAILBOX                /* Include SM0RGV mailbox server */
#define MAILCLIENT
#define MAILCMDS               /* Include mail commands, S, R, V etc */
#define MAILFOR                /* Include Mailbox 'Mail for' beacon */
#define MAIL_HDR_TRACE_USER_PORT          /* 26Jul2005, Maiko, MH Trace info */
#define MAILMSG                /* Include mailmsg command */
#define MAXSCC      4          /* Max number of SCC+ESCC chips (< 16) */
#define MBFWD                  /* Include Mailbox AX.25 forwarding */
#define MONITOR                /* Include user-port monitor trace mode */
#define MORESESSION            /* more - view ASCII file page by page */
#define MTHRESH   16384        /* Default memory threshold */
#define NAX25       24         /* Number of axip interfaces (if defined) */
#define NETROM                 /* NET/ROM network support */
#define NETROMSERVER           /* Net/rom server */
#define NETROMSESSION          /* netrom connect & split iff NETROM defined */
#define NEWS_TO_MAIL           /* NNTPS emails per gateway file */
#define NIBUFS      5          /* Number of interrupt buffers */
#define NN_REMOVE_R_LINES           /* remove R: lines from incoming email */
#define NNTPS_TIMEOUT  3600         /* idle-timeout #secs for nntp server */
#define NNTP_TIMEOUT 900            /* idle-timeout #secs for nntp client (both versions)*/
#define NN_USESTAT             /* Try GROUP/STAT cmds if NEWNEWS fails */
#define NR4TDISC               /* Include Netrom L4 timeout-disconnect */
```

```
#define NROWS        25     /* Number of rows on screen */
#define NRS                 /* NET/ROM async interface */
#define NSESSIONS  10       /* Number of interactive clients */
#define PACKET              /* FTP Software's Packet Driver interface */
#define PINGSESSION         /* ping cmd */
#define POP3CLIENT          /* POP3 client -- IAB draft standard */
#define POP3SERVER          /* POP3 server -- IAB draft standard */
#define POP_TIMEOUT 600             /* pop server idle timeout value, in secs (>= 600) */
#define RDATECLI            /* Time Protocol client */
#define REDIRECT            /* Allow cmd [options] > outfile. Use >> to append */
#define REGISTER    /       * Include User Registration option */
#define REMOTECLI           /* remote UDP kick/exit/reset */
#define REMOTESERVER  /* Udp remote server */
#define REPEATSESSION  /* repeat cmd */
#define RLINE               /* Include BBS R:-line interpretation code */
#define SERVERS             /* Include TCP servers */
#define SESSIONS
#define SHELL               /* Include shell command */
#define SLIP                /* Serial line IP on built-in ports */
#define SM_CURSES
#define SMTP_DENY_RELAY     /* Refuse to relay msgs from hosts not in our subnets */
#define SMTPSERVER          /* Tcp smtp server */
#define SPLITSCREEN         /* Needed for split, netrom split, and ttylink cmds */
#define STATUSWIN           /* Up to 3 line status window */
#define TELNETSERVER   /* Tcp telnet server */
#define TELNETSESSION  /* telnet cmd */
#define TIPSERVER           /* Serial port tip server */
#define TIPSESSION          /* tip - async dumb terminal emulator */
#define TRACE               /* Include packet tracing code */
#define TTYCALL             /* Include AX.25 ttylink call */
#define TTYLINKSERVER  /* Tcp ttylink server */
#define TTYLINKSESSION          /* ttylink cmd - split-screen chat */
#define TUN                 /* Enable use of the TUN linux kernel module for networking */
#define UDP_DYNIPROUTE          /* Support dynamic IPaddr encap routes via UDP/remote */
#define USERLOG             /* Include last-msg-read,prompt-type user tracking */
#define USERLOG             /* need to remember (new)lastread */
#undef AGGRESSIVE_GCOLLECT /* exit 251 when availmem < 1/4 of 'mem threshold'*/
#undef APPLETALK           /* Appletalk interface (Macintosh) */
```

```
#undef APRSC
#undef ARCNET             /* ARCnet via PACKET driver */
#undef AUTOROUTE          /* Include AX.25 IP auto-route code(causes problems when VC mode is
used for ip) */
#undef AX25PASSWORD       /* Ask ax.25 users for their passwords */
#undef AX25SERVER
#undef AX25SESSION
#undef AXIP
#undef AXUISESSION        /* Ax.25 unproto (axui) command */
#undef BOOTPCLIENT        /* Include BootP protocol client */
#undef BOOTPSERVER        /* Include BootP protocol server */
#undef BPQ                /* include Bpqhost interface */
#undef BUCKTSR            /* Buckmaster callsign DB via bucktsr.exe (April 95) */
#undef CALLBOOK
#undef CALLCLI            /* Include only callbook client code (to query remote server) */
#undef CALLSERVER         /* Include BuckMaster CDROM server support */
#undef CGI_POSTLOG        /* HTTP: Add a POST demo via CGI */
#undef CGI_XBM_COUNTER         /* HTTP: Add an X-bitmap counter via CGI */
#undef CNV_CALLCHECK           /* Convers only allows callsigns */
#undef CNV_CHAN_NAMES          /* Convers named channels */
#undef CNV_CHG_PERSONAL        /* Allow users to change personal data permanently */
#undef CNV_ENHANCED_VIA        /* If convers user is local, "via" gives more info */
#undef CNV_LINKCHG_MSG         /* Send link-change messages in convers */
#undef CNV_LOCAL_CHANS         /* Convers local channels and msg-only channels */
#undef CNV_TIMESTAMP_MSG       /* Add hh:mm prefix to msgs sent to local users */
#undef CNV_TOPICS         /* Convers channel topics are gathered */
#undef CNV_VERBOSE
#undef CNV_VERBOSE_CHGS        /* Default to /VERBOSE yes. Use this judiciously! */
#undef CNV_VERBOSE        /* Verbose msgs */
#undef CONVERS            /* Conference bridge (babble-box :-) */
#undef DIALER             /* SLIP/PPP redial code */
#undef DISCARDSERVER      /* Tcp discard server */
#undef DRSI               /* DRSI PCPA slow-speed driver */
#undef EAGLE              /* Eagle card driver */
#undef ECHOSERVER         /* Tcp echo server */
#undef EDITOR             /* include internal ascii editor */
#undef EMS                /* Include Expanded Memory Usage */
#undef ESCAPE             /* Allow Unix style escape on PC */
```

```
#undef EXPIRY
#undef FBBVERBOSELOG /* log more data for FBB-protocol transfers */
#undef FILECMDS
#undef FOQ_CMDS
#undef FTPDATA_TIMEOUT          /* ftp server timeout on recvfile */
#undef FTPSERV_MORE     /* ftp server supports RNFR, RNTO, MDTM commands */
#undef FWD_COMPLETION_CMD /* run a forwarding-completed command if set in script */
#undef FWDCTLZ          /* Use a CTRL-Z instead of /EX to end message forwarding */
#undef FWDFILE          /* Include forwarding-to-file (export) feature */
#undef GATECMDS
#undef HAPN             /* Hamilton Area Packet Network driver code */
#undef HOLD_LOCAL_MSGS          /* Hold locally-originated msgs for review by sysop */
#undef HOPPER           /* Include SMTP hopper code by G8FSL */
#undef HP95             /* hp95-style uart handling */
#undef HS               /* High speed (56kbps) modem driver */
#undef HTTP_EXTLOG      /* HTTP: Add detailed access logging in "/wwwlogs" dir */
#undef HTTP             /* Selcuk Ozturk's HTTP server on port 80 */
#undef ICALL            /* Buckmaster's international callsign database April '92 */
#undef IDENTSERVER      /* RFC 1413 identification server (113/tcp) */
#undef IPACCESS         /* Include IP access control code */
#undef KICK_SMTP_AFTER_SHELLCMD         /* kick smtp client after each shell cmd */
#undef KISS
#undef LINK             /* permit this convers node to be linked with others*/
#undef MAILCMDS
#undef MAILCMDS
#undef MAILERROR        /* Include Mail-on-error option */
#undef MBOX_DYNIPROUTE          /* Add XG mbox cmd to route dynamic IPaddr via encap*/
#undef MBOX_FINGER_ALLOWED              /* undef=>telnet permission needed for mbox finger */
#undef MBOX_PING_ALLOWED /* undef=>telnet permission needed for mbox ping */
#undef MBXTDISC
#undef MD5AUTHENTICATE          /* Accept MD5-Authenticated logins */
#undef MEMLOG           /* include alloc/free debugging to MEMLOG.DAT file? */
#undef MONITOR
#undef MONSTAMP         /* add time stamp to monitor-style trace headers */
#undef MORESESSION
#undef MULTITASK        /* Include Dos shell multi-tasker */
#undef NN_INN_COMPAT /* send "mode reader" cmd after connecting to server */
#undef NNTP             /* Netnews client */
```

```
#undef NNTPS            /* Netnews client and server */
#undef NR4TDISC
#undef NRPASSWORD       /* Also ask NetRom users for passwords */
#undef NRS
#undef PACKET           /* FTP Software's Packet Driver interface */
#undef PACKETWIN        /* Gracilis PackeTwin driver */
#undef PC100            /* PAC-COM PC-100 driver code */
#undef PC_EC            /* 3-Com 3C501 Ethernet controller */
#undef PI               /* VE3IFB pi dma card scc driver */
#undef POLLEDKISS       /* G8BPQ Polled Multidrop KISS TNC code */
#undef POP2CLIENT       /* POP2 client -- IAB not recommended */
#undef POP2SERVER       /* POP2 server -- IAB not recommended */
#undef POPT4            /* add 'pop t4' command to pop3 client, setting timeout */
#undef PPP_DEBUG_RAW /* Additional PPP debugging code...see pppfsm.h */
#undef PPP              /* Point-to-Point Protocol code */
#undef PRINTEROK        /* OK to name a printer as an output device */
#undef PS_TRACEBACK     /* ps <pid> option enabled to do a back-trace */
#undef QRZCALLB         /* QRZ callbook server. Note that you can NOT have */
#undef RARP             /* Include Reverse Address Resolution Protocol */
#undef RDATESERVER      /* Time Protocol server */
#undef RELIABLE_SMTP_BUT_SLOW_ACK  /* smtp server delays msg ack until filing completed */
#undef RIP98            /* Include RIP98 routing */
#undef RIP              /* Include RIP routing */
#undef RLOGINSESSION    /* Rlogin client code */
#undef RSPF             /* Include Radio Shortest Path First Protocol */
#undef RSYSOPSERVER     /* Tcp telnet-to-mbox-as-sysop server */
#undef RXECHO           /* Echo rx packet to another iface - WG7J */
#undef SAMCALLB         /* SAM callbook server. Note that you can NOT have */
#undef SCC              /* PE1CHL generic scc driver */
#undef SEND_EDIT_OK     /* Send cmd offers (E)dit option to mbox users */
#undef SHOWFH           /* show free filehandles in status line */
#undef SHOWIDLE         /* show relative system-idle in status line */
#undef SLFP             /* SLFP packet driver class supported */
#undef SM_CURSES
#undef SM_DUMB
#undef SM_RAW
#undef SMTP_REFILE      /* smtp server rewrites to addr according to from|to */
#undef SMTP_VALIDATE_LOCAL_USERS /* local user must be in ftpusers/popusers/users.dat */
```

```
#undef STATUSWINCMD   /* status off|on command to modify status window */
#undef STKTRACE       /* Include stack tracing code */
#undef SWATCH         /* stopWATCH code */
#undef TCPACCESS      /* Include TCP access control code */
#undef TCPGATE        /* TCPGATE redirector server */
#undef TED            /* editor uses TED syntax; local console only */
#undef TERMSERVER      /* Async serial interface server */
#undef TIMEZONE_OFFSET_NOT_NAME /* smtp headers use hhmm offset from GMT */
#undef TIPSERVER
#undef TN_KK6JQ       /* add more telnet options support */
#undef TRACESERVER    /* remote interface trace server */
#undef TRANSLATEFROM          /* smtp server rewrites from addrs too */
#undef TTYCALL
#undef TTYCALL_CONNECT        /* SABM pkt uses TTYCALL, not BBSCALL, as src call */
#undef TTYLINK_AUTOSWAP       /* ttylink server automatically swaps to new session */
#undef UNIX_DIR_LIST  /* Unix-style output from DIR and ftp DIR cmds. */
#undef USERLOG
#undef VJCOMPRESS     /* Van Jacobson TCP compression for SLIP */
#undef WELCOMEUSERS   /* 05Jun2005, Maiko, Extraneous welcome info */
#undef WS_FTP_KLUDGE  /* ftp server lies to please ws_ftp winsock app */
#undef XCONVERS       /* LZW Compressed convers server and links */
#undef XMODEM         /* xmodem file xfer for tipmail  */
#undef XMS            /* Include Extended Memory Usage */
```

In addition to the options listed above, some automated processing is included in config.h to aid a sensible configuration and save some labor by automation. While setting options it is will to consider the following testing and programmer notes. To determine the detail of the process, please read config.h and act accordingly.

```
#ifdef DIRSESSION
#if defined(ARCNET) || defined(SLFP)
#if defined(BUCKTSR)
#if defined(CALLSERVER)
#if defined(EDITOR) && defined(ED) && defined(TED)
#if defined(NNTP) && defined(NNTPS)
#if defined(NRS)
#if defined(PC_EC) || defined(PACKET)
#if defined(POP2CLIENT) || defined(POP3CLIENT)
#if defined(QRZCALLB)
#if defined(SAMCALLB)
```

```
#ifdef POLLEDKISS
#ifdef STATUSWIN
#ifdef UNIX
#ifndef AX25
#ifndef _CONFIG_H
#ifndef CONVERS
#ifndef FBBFWD
#ifndef LZW
#ifndef MAILBOX
#ifndef MAILCMDS
#ifndef MBFWD
#ifndef NETROM
#ifndef SM_CURSES
#ifndef SM_DUMB
#ifndef SM_RAW
#ifndef SMTPSERVER
#ifndef TIPSERVER
#ifndef TRACE
#if NIBUFS == 0

#error Cannot #define both ED and TED
#error Cannot #define both NNTP and NNTPS
#error NIBUFS should never be zero
```

Just to demystify where the above information came from, and to assist the owner in obtaining like information for his own documentation, use the command line utilities “sort” and “gedit” (or your favorite editor) to process your adjusted config.h at any time to get a side-by-side comparison.

Host FILE Structure - name, path, & format

After services are specified, support data files must be configured. JNOS contains a couple methods for assigning file location. The first is to modify "files.c" prior to compilation to set new defaults, the second is to provide "nos.cfg" file for use during application startup to override defaults. Below are the defaults used in the standard distribution.

NAME	*PATH*	*DIRECTORY CONTENTS*
CmdsHelpdir	./help	Console/Sysop commands help
Spoolqdir	./spool	spool

NAME	***PATH***	***DIRECTORY CONTENTS***
Mailspool	./spool/mail	local mail gets delivered
Fdir	./finger	finger files go
LogsDir	./logs	logs containing system activity
	./trace	I/O port activity trace
	./public	FTP transfer root (subdirectories OK)
	./aprs	APRS status
Helpdir	./spool/help	BBS help
Mailqdir	./spool/mqueue	incoming mail gets queued for the smtp daemon
Routeqdir	./spool/rqueue	if you route mail queue for the smtp daemon
	./spool/log	[tbd]
Signature	./spool/signatur	user signatures
Newsdir	./spool/news	NNTP
HttpDir	./wwwroot	Http server root
HttpStatsDir	./wwwstats	Http server statistics
HLogsDir	./wwwlogs	Http server extended logs

NAME	***PATH***	***FILE CONTENTS***	***USE***
	./nos.cfg	Override directory structure and file locations	Administrator
Startup	./autoexec.nos	system setup (startup) commands in JNOS format.	Administrator
Onexit	./onexit.nos	these commands get executed on exit	Administrator
Userfile	./ftpusers	user permission	Security
Hostfile	./net.rc	ftp host file for auto-login	Administrator
Alias	./alias	mail recipient alias	BBS
Dfile	./domain.txt	domain.txt	
Motdfile	./spool/motd.txt	BBS message of the day	BBS
Mreg	./spool/mreg.txt	Registration help	Security
Arealist	./spool/areas	list of public areas on the system	BBS
Maillog	./spool/mail.log	mail log	BBS
Mailqueue	./spool/mqueue/*.wrk		BBS
Historyfile	/spool/history	Bulletin ID's	BBS
Popusers	./popusers	POP server users are listed	Security

NAME	*PATH*	*FILE CONTENTS*	*USE*
Ftpmotd	./spool/ftpmotd.txt	FTP message of the day	FTP
Rewritefile	./spool/rewrite	otherwise Rewritefile does TO address	mail rewrite rules
Translatefile	./spool/translat	Translatefile does FROM adrs	mail rewrite rules
Refilefile	./spool/refile	Refilefile does the TO addr as a function of "FROM| TO"	mail rewrite rules
Forwardfile	./spool/forward.bbs	forwarding rules	BBS mail forward
Fdbase	./finger/dbase.dat	finger database	Finger server
Pdbase	./spool/names.dat	mail daemon searches for full user names	Finger server
Holdlist	./spool/holdlist	areas and mailboxes subject to msg-hold on locally-originated msgs	Finger server
Cinfo	./finger/dbase.dat	Convers user info; Convers user personal info / notice that default is the same as Fdbase !	Converse server
Cinfobak	./finger/dbase.bak	Convers user info backup / Previous Cinfo file (after update)	Converse server
ConvMotd	./spool/convmotd.txt	Convers MOTD file / Connect greeting.	Converse server
Channelfile	./spool/channel.dat	Convers channel numbers to names	Converse server
?access.rc?[tbd]		DOS access perm file (see Dave)	
Netromfile	./netrom.sav	saved netrom routes	NET/ROM
Expirefile	./spool/expire.dat	expire command	NET/ROM
NInfo	./news/info	NNTP info	NNTP
Naccess	./news/access	NNTP access	NNTP
Active	./news/active	NNTP active	NNTP
Pointer	./news/pointer	NNTP pointer	NNTP
Nhelp	./news/help	NNTP help	NNTP
History	./news/history	NNTP message history	NNTP
Forward	./news/forward	NNTP forward	NNTP
Poll	./news/poll	NNTP poll	NNTP
Newstomail	./news/gateway	NNTP news to mail mapping	NNTP
UDefaults	./spool/users.dat	user defaults	automatic update
UDefbak	./spool/users.bak	backup of the above	automatic update

nos.cfg # Format: one line containing "<internal name> = <file path>"

Userfile # Format: <name> <password> [<drive:></rootdir>;</root2>] <#perms>] [=]...

Hostfile # Format: remote_hostname login_name password

Alias # Format: aliasname recipient1[<tab> or <SP>]recipient2 recipient3 recipient4 ... recipientN

Cinfo # format: <name|call> <personal data, i.e., QTH, etc.>

Channelfile # format: <channel_number> <channel_name>

# NOTE: UDefaults and UDefbak have to be ON THE SAME DRIVE !!!

Mailqueue # this should have same path as smtpSERVER

Expirefile # Format:
  # comment line
  #< mbox_name> <number_of_days_to_retain_messages>
  # <mbox_path> <number_of_days_to_retain_messages>
  # <!news.group.name> <number_of_days_to_retain_articles>

Why is the binary so large AND what to do about crashing

Note that my makefile has debugging turned on, so any 'jnos' binary you compile will be large (because it contains debuggin info). If you are not at all interested in debugging, then you can reduce the size of the 'jnos' binary using the following command :

```
    strip jnos
```

which will strip out the debugging information and symbol table.

If you ARE interested in helping me fix bugs, then I encourage you to run the GDB debugger that comes with most linux distros, AND make sure you do NOT strip the 'jnos' binary, since it contains important info for the debugger to use when a crash occurs.

Using GDB is easy. With JNOS running already, find out it's pid, using the linux command, 'ps -ef | grep jnos'. Once you know what the pid is, then run the gdb debugger something like this :

```
    gdb -p pid
```

GDB will load, JNOS will hang temporarily, and GDB will suddenly give you a prompt. Enter the command, 'continue', at the prompt, and JNOS will continue to run again.

When a crash occurs, GDB will break out to the prompt again, and JNOS will hang. Take a screen shot of what GDB printed out, then type in the command, 'where' or 'back', at the GDB prompt, and note the info that appears. Please send all of that information to me, with a brief explanation of what might have been going on at the time.

If you have JNOS logging in effect, please send the log file at the time of the crash if you don't mind. Logs are under /jnos/logs/ directory (by default).

In order to capture operator interface data for documentation (and when things go wrong), I found a couple ways that function pretty well. I find them most useful as

I replicate problems since malfunctions are fairly infrequent, and so I don't always operate with the following stuff in place. After capturing the problem thru reenactment, I find it pretty easy to use my favorite text editor to select the portion of the screen text that relates to the bug, and prepare a file to use as an email attachment.

To capture material for debugging purposes, my standard mode of operating (on Linux FC-4) is to:

1. init jnos from multi-session "F4" as root (#) so I have a separate jnos console immediately available
2. init gdb from multi-session "F3" as root (#) so I have an independant console for trace control and capture.
3. init telnet from multi-session "F5" as my normal user ($) to access jnos BBS services
4. use the GUI tools for most everything else on multi-session "F7" (as a normal uid)

I don't have jnos established as a service for 24/7 operation (yet) since radio availability is an issue and sometimes the TNC needs TLC if there has been power interruption. I often trace interface traffic using standard jnos features and catalog the results for back-tracking purposes.

My favorite is to capture using "tee", a Linux (Unix) basic utility. To load the file "screen.txt" with data presented on the terminal, use the command:

```
...$ telnet 192.168.2.2 | tee screen.txt
```

After telnet exits, edit the screen.txt file as needed to display what was seen during the session. One caveat - the command line echos (that I type) are missing from the output. All-in-all, this works pretty well to collect text for documentation purposes - I use it to help me remember details about remote stations.

My second favorite for deeper analysis in HEX when special characters are involved, telnet has a built-in trace facility. There is a utility to re-display the hex data in character form, but I have not needed it to date. If the plan is to capture data, then:

```
...$ telnet 192.168.2.2 -n screen.dump
```

While in the session and before the problem, escape and enable the trace with:

```
^]

telnet> toggle netdata
```

After the problem (presuming telnet has not crashed) repeat the above to stop collection. After the session is complete then edit the file "screen.dump" with that favorite editor. When there is a surprise, and the "-n file" paramater is not given to telnet upon start-up, the file may be set during the session by:

```
^]

telnet> set tracefile screen.dump
```

That needs to be done before toggling netdata to keep the trace data off the screen. There are more options to this found in the man pages for telnet.

My third favorite requires root (#) user, so I don't do it often... Quoting Maiko:

" I found this a short time ago -

http://www.faqs.org/docs/Linux-HOWTO/Keyboard-and-Console-HOWTO.html#s20

I hope that helps. "

This uses the command:

```
      ...# setterm -dump 5
```

In this case, I use the "F1" console as root to issue the command to capture the text existing at the moment on console "F5". Notice that by changing the "-dump #" that a screen may be captured from 2=administrator, 3=debugger, 5=user. This works when finding a suspected condition to document (assuming the condition fits one screen full or less) and the user can skip over to another multi-console to do the capture... This does not trace, it snapshots the screen and places the content into the file "screen.dump". Again my favorite editor helps me make a "show-and-tell" display for whatever purpose. NOTE: I have found the normal user can also do this BUT it must be preceded by a su and chmod to set /dev/vcsa5 with o+rw permissions.

I hope someone else finds this useful too...

BIBLIOGRAPHY

The bib is presented in two sections. Where particular versions or revisions apply, they are noted.

The first section are works compiled into this document – either in total or has a portion extracted. The original works may be somewhat modified depending on results of the testing program on the jnos version identified in the title sheet. Some attempt is made to remain current with subsequant modifications by the original author, but no formal program is undertaken.

The second section are works worthy of consideration in deeper research of NOS technology.

DOCUMENTS COMPILED HEREIN

JNOS 1.1 COMMANDS MANUAL [ID: JN-CMD1.11] Rel 1.11
A derivative of the JNOS Commands Manual
by Johan K. Reinalda and Douglas E. Thompson
as modified by Robert Fahnestock [WH6IO] and James P. Dugal [N5KNX]

1) INSTALLING ON LINUX Release JNOS2.0d
2) HF Support for DXP and SCS-PCT-II Pro Modens – JNOS2.04c
3) How to use IP-in-UDP Encapsulation – JNOS2.0
4) SMTP Configuration notes – JNOS2.0 & 1.11f
5) How to replace SLATTACH with TUN – JNOS2.0
6) Customized AX.25 Cross Port Digipeating Rules – JNOS2.0c5a
7) Compilation instructions
all by Maiko Langelaar

NOS mail docs
BY G4AMJ/NQ0I and SM0RGV (Rev. 3) {ed 2006 Note: NQ0I is now N7DR}
Slight mods for JNOS 1.11
by N5KNX (Rev. 3a).

Mailbox Command Cryptsheet (7/22/95 FOR JNOS 1.10L)

by Johan. K. Reinalda, WG7J/PA3DIS

DOCUMENTS REFERENCED

This list is not exhaustive; there are many other interesting articles, but these are the ones most relevant to NOS and TCP/IP.

ARRL Computer Networking Conference Proceedings
Available from ARRL HQ, Newington CT. Send mail to info@arrl.org for an automatic response pointing at more information about the ARRL. Some of these papers are available online in the directory ftp.ucsd.edu:/hamradio/packet/tcpip/docs.

NOS Overviews and Documentation

```
 NOS Command Set Reference
   Ian Wade G3NRW   10th (1991)

 NOSVIEW: The On-Line Documentation Package for NOS
   Ian Wade G3NRW   11th (1992)

 The KA9Q Internet (TCP/IP) Package: A Progress Report
   Phil Karn KA9Q   6th (1987)

 Amateur TCP/IP: An Update
   Phil Karn KA9Q   7th (1988)

 Amateur TCP/IP in 1989
   Phil Karn KA9Q   8th (1989)
```

This should be considered to be required reading.

```
 MACA - A New Channel Access Method for Packet Radio
   Phil Karn, KA9Q   9th (1990)

 A Duplex Packet Radio Repeater Approach to Layer One
 Efficiency
   Robert Finch, N6CXB  6th (1987)
   Scott Avent, N6BGW

 A Duplex Packet Radio Repeater Approach to Layer One
 Efficiency, Part Two
   Scott Avent, N6BGW   7th (1988)
   Robert Finch, N6CXB
```

Services and Protocols

```
 The Design of a Mail System for the KA9Q Internet protocol
   Bdale Garbee, N3EUA  6th (1987)
   Gerard van der Grinten, PA0GRI

 Finger - A User Information Lookup Service
   Michael T. Horne, KA7AXD  7th (1988)
 Callsign Server for the KA9Q Internet Protocol Package
   Doug Thom, N6OYU     8th (1989)
   Dewayne Hendricks, WA8DZP

 The Network News Transfer Protocol and its Use in Packet Radio
   Anders Klemets, SM0RGV   9th (1990)

 A Routing Agent for TCP/IP: RFC 1058 Implemented for the KA9Q
 Internet Protocol Package   7th (1988)
   Albert G. Broscius, N3FCT

 Thoughts on the Issues of Address Resolution and Routing in
 Amateur Packet Radio TCP/IP Networks
   Bdale Garbee, N3EUA  6th (1987)

 Another Look at Authentication
   Phil Karn KA9Q   6th (1987)

 LZW Compression of Interactive Network Traffic
   Anders Klemets, SM0RGV   10th (1991)

 PACSAT Protocol Suite -- An Overview
   Harold Price, NK6K   9th (1990)
   Jeff Ward, G0/K8KA

 BULLPRO -- A Simple Bulletin Distribution Protocol
   Tom Clark, W3IWI     9th (1990)
```

Macintosh

```
 KA9Q Internet Protocol Package on the Apple Macintosh
   Dewayne Hendricks, WA8DZP 8th (1989)
   Doug Thom, N6OYU

 Status Report on the KA9Q Internet Protocol Package for the
 Apple Macintosh
   Dewayne Hendricks, WA8DZP 9th (1990)
   Doug Thom, N6OYU

 Higher Speed Amateur Packet Radio using the Apple Macintosh
 Computer
   Doug Yuill, VE3OCU   10th (1991)
```

Network design

```
 The Implications of High-Speed RF Networking
   Mike Chepponis, K3MC  8th (1989)
   Glenn Elmore, N6GN
   Bdale Garbee, N3EUA
   Phil Karn, KA9Q
   Kevin Rowett, N6RCE

 Design of a Next-Generation Packet Network
   Bdale Garbee, N3EUA  8th (1989)
```

More and Faster Bits: A Look at Packet Radio's Future
Bdale Garbee, N3EUA 7th (1988)

Physical Layer Considerations in Building a High Speed Amateur
Radio Network
Glenn Elmore, N6GN   9th (1990)

Spectral Efficiency Considerations for Packet Radio
Phil Karn, KA9Q   10th (1991)

Network Implementation

Packet Radio at 19.2 kB -- A Progress Report
John Ackermann, AG9V 11th (1992)

Implementation of a 1Mbps Packet Data Link
Glenn Elmore, N6GN   8th (1989)
Kevin Rowett, N6RCE

Hubmaster: Cluster-Based Access to High-Speed Networks
Glenn Elmore, N6GN   9th (1990)
Kevin Rowett, N6RCE
Ed Satterthwaite, N6PLO

Recent Hubmaster Networking Progress in Northern California
Glenn Elmore, N6GN   9th (1990)
Kevin Rowett, N6RCE

The 56 kb/s Modem as a Network Building Block: Some Design
Considerations
Barry McLarnon, VE3JF   10th (1991)

Digital Networking with the WA4DSY Modem - Adjacent Channel
and Co-Channel Frequency Reuse Considerations
Ian McEachern, VE3PFH   10th (1991)

A Full-Duplex 56kb/s CSMA/CD Packet Radio Repeater System
Mike Chepponis, K3MC 10th (1991)
Lars Karlsson, AA6IW

A High Performance, Collision-Free Packet Radio Network
Phil Karn KA9Q   6th (1987)

Adaptation of the KA9Q TCP/IP Package for Standalone Packet
Switch Operation
Bdale Garbee, N3EUA 9th (1990)
Don Lemley, N4PCR
Milt Heath

Hardware

The KISS TNC: A Simple Host-to-TNC Communications Protocol
Mike Chepponis, K3MC 6th (1987)
Phil Karn, KA9Q

```
The Ottawa Packet Interface (PI) A Synchronous Serial PC
Interface for Medium Speed Packet Radio
  Dave Perry, VE3IFB   10th (1991)

HAPN-2: A Digital Multi-Mode Controller for the IBM PC
  John Vanden Berg, VE3DVV 11th (1992)

The PackeTen system - The Next Generation Packet Switch
  Don Lemley, N4PCR    9th (1990)
  Milt Heath
```

APPENDIX

Appendicies follow. The first two are intended to be printed and used as cheat-sheets or reference cards for moderately seasoned operators. Others also contain topics that may warrent printing seperately for particular task notes.

APPENDIX i: JNOS Administrator Command Crib_sheet

```
<CR>
!
#
?
abort [<session #>]
arp
asyconfig <iface> <parameter>
asystat
at
attach
attended [off | on]
axui <iface> [<unpronto call>][digipeater string]]
ax25 <subcommands>
bbs
bulletin<subcommands>
callserver2 [<hostname> <port>]
comm <asy_iface> <"text-string">
connect <iface> <destination> [<digi1,digi2...digin>]
convers <subcommands>
delete <filename>
dialer<subcommand>
domain <subcommand>
DOS Environmental Variables
dump <hexaddress | .> [range]
echo [accept|refuse]
edit [<filename>]
eol {standard | null]
errors [ON | off]
etelnet <host> [<port_number>] loginid password
ettylink
exit [<return_code>]
expire <subcommand>
finger <username [@host] > [<username [@host]>...]
Fkey
Ftp <hostname> [<scriptfile>]
gate <subcommands>
help  or  ?
history [<N>]
hop <subcommands>
hostname [<name>]
http <subcommand>
icmp <subcommands>
ifconfig [<subcommand>]
index [<areaname>]
info
ip <subcommand>
lock [password "<password_string">]
log [on|off]
look [user | socket#] (/<cmd>) (
lzw
mailmsg <to_addr> ["<subject"] <msg | /path>
mbox [<subcommands>]
```

```
memory <subcommands>
mkdir <directory>
mode <iface> [vc | datagram]
more <filename> [<text>...]
motd ["message"]
netrom <subcommands>
nntp
nrstat
oldbid <interval>
param <iface> [<param>]
pause <seconds>
ping <host> [<length>] [<repeat_ms> [incflag>]]]
popmail<subcommand>
prompt [on|OFF]
ps
pwd <directory>
rarp
rdate <subcommand>
record  [off | <filename>]
remark
remote
rename <old filename> <new filename>
repeat [milliseconds]  command
reset [<session>]
rewrite <address>
rip <subcommand>
rlogin <host>
rmdir <directory>
route [<subcommand>]
rspf <subcommand>
session [<session#>]
sessmgr <subcommand>
shell [cmd [args]]
skick <socket#>
smtp <subcommand>
socket [<socket#>]
source <script_filename>
split <iface><call>
start <servers...>
status [on | off]
stop <server...>
tail <filename>
taillog
tcp <subcommand>
telnet <host>
term
tip <interface>
trace [<iface> [off | <btio>]
ttylink <host> [<port_number>]126
udp status
write <username|sock#> <message>
writeall <message>
```

APPENDIX ii: JNOS BBS Command Crib_sheet

<cr>	Hitting a return causes the next message (if any) to be read
?	Give short list of all commands
A	Give list of message areas without descriptions
AF	Give the areas list with descriptions (if set)
AF name	Give more information about the specified area (if available)
AN	Show the areas that have new mail since you last logged off
A name	Change to area 'name'
ALI	Shows command 'aliases' as set by the sysop. (abbreviation)
B	Bye (ie. disconnect)
C node	Netrom connection, if permitted and available
C port call	AX.25 connection, if permitted
CA	Connect to callbook server (if available)
CONV [chan]	Convers bridge (like Dx-CLUSTER), if available [Default is channel 0]
D filename	Download textfile 'filename'
DU filename	Download a uuencoded file (for binary files)
DX filename	Download with xmodem (for tip connects)
DM	Download (ie, display) message-of-the-day
E	Show the escape character
E char	Set the escape character to 'char' (one character!)
F	Finger users on this system or remote tcp systems 'F conf' will show conference bridge users. 'F iheard' will show the ip-heard list. For local users, this will now show when the user was last logged in to the system. Eg 'F wg7j' For remote tcp systems 'F @host' shows all users on 'host' or 'F user@host' shows 'user' on system 'host'
H	Help
H x	Get help on command x
I	Info on system. Shows a message set by the sysop
IH	IHeard, shows recently heard tcp/ip stations
IP	IProute, shows the tcp/ip routes the system maintains
J	Just heard, on all sysop-enabled ports
J port	Just heard on 'port'
K n n	Kill message # n (one or more numbers or ranges is accepted)
KM	Kill Mine, kills all read messages
KU	Un-kills messages previously marked to be killed
L	List new messages
LA	List ALL messages in current area
LB	List messages with message type B (bulletins)
LH	List Held messages. All msgs in current area are checked.
LM	List mine, lists new messages
LL n	List last n messages
LT	List message with message type T (traffic)
LS xyz	List messages with string 'xyz' in subject
L> xyz	List only messages with string 'xyz' in the To: field
L< xyz	List only messages with string 'xyz' in the From: field
M	Shows current mailbox users
MC name	Copy the current message to area/path 'name' (SYSOP only)
MC x [y,z] name	Copy messages x,y and z in current area to area/path 'name'
ML	Shows all past users since system startup
ML n	Shows the n past users since startup
ML call	Shows when 'call' logged on last

```
MM name         Move the current message to area/path 'name' (SYSOP only)
MM x [y,z] name Move messages x,y and z in current area to area/path 'name'
MS              Show message and system status
N               Alphabetical listing of known netrom nodes (if available)
NR              Shows all netrom neighbour nodes
N nodename/call Shows information about the route to the node
N *             Shows infomation about all nodes (Long!)
O               Chat with operator (if system is attended)
P               Give a list of the ax.25 ports of the system (and description,  if
set by sysop)
PI host         Ping a tcp/ip host (ie. check if it is 'alive').  This shows the
round-trip-time to the system.
R n             Read message n (a list of numbers or ranges is accepted)
RH              Read all Held messages (SYSOP only).
RM              Read Mine, reads all unread messages
REG             Register as a user on this bbs
S               Send message (defaults to private)
SB              Send bulletin (be carefull!)
SC              Send with Carbon Copy to others
SF              Forward current message to someone else
SP              Send private message
ST              Send traffic message
SR              Send a reply to the current message
SR n            Send a reply to message number n
T host [port]   Telnet to 'host' (if permitted) (the optional port specifies
                a tcp port if other then the telnet port)
U file          Upload a textfile
UX file         Upload a textfile with xmodem (for Tip users)
V n             Verbose read of message n (a list or range is accepted)
VH              Verbose read of all Held messages (SYSOP only).
VM              Verbose read of all unread messages
W               What files are in the current directory
W path          Listing of the directory 'path'. Wildcards ie, *, are accepted
X               Toggle Expert status
XA              Toggle current area-indication in prompt
XM              Show number of lines before more-prompt
XM n            Set more-prompt to n lines
XN              Toggle the 'Netrom look-alike' prompt on/off
XP              Toggle LINEMODE-style prompting for input (telnet/tip users)
XR              Show if a 'reply-to: your-email-address' is added when sending
                mail. You need to have set an email address with 'REG'
XR on|off       Sets the 'reply-to' state to on or off
Z file          Zap (delete) a file (if permitted)
```

APPENDIX A: REWRITE FILE

The Rewrite file is used to perform a one-to-one mapping between a message's destination (To:) addresses as received by JNOS, and the destination addresses as actually used by JNOS. Each record within the rewrite file comprises a single line, containing either two or three fields separated by spaces. The first field is the template field; if a destination address matches the template, it is replaced by the second field after variable substitution. The rewrite process terminates after the first template match, with JNOS using the resultant value as the new destination address, unless there is an optional third field which contains the single letter "r". In this case, JNOS rescans the rewrite file from the beginning, attempting to match the new destination address with one of the templates. If no template matches the destination, it is used unchanged as the result.

If the address resulting from rewrite contains an "@" it is sent via smtp. Otherwise it should be the name of an area. There is one special case: "refuse" as the destination causes the message to be rejected.

A template contains a combination of verbatim ascii text, and special characters "?*+\". To treat a special character as an ordinary character, precede it by '\'. The '?' character matches any single character, '*' matches any number of consecutive characters (including zero), and '+' matches a sequence of one or more characters. In the second field, the character "$", followed by a single digit in the range 1 to 9, represents the string that matched the respective '*' or '+' in the template. For example,

```
    tcpip@* tcp
    *@tcpip tcp
```

rewrites all addresses beginning or ending with "tcpip" to the tcp area. The two-character sequence, $H, in the second field is replaced by the hostname of the system, as set by the hostname command.

SAMPLE REWRITE FILE (edit to suit your call and neighbors):

```
# Sample rewrite file for K5ARH pbbs, also known as w5ddl.ampr.org
#
# Special destinations: refuse  => send NO (ax.25) or Bad host (smtp)
# (a.k.a. "areas")  nts*    => mailbox writable by all (i.e., kill cmd works)
#           Also, ST used in forwarding.
#       help    => never flag as already read, so L shows all
# take x%y@me and reprocess as x@y
*%*@w5ddl* $1@$2 r
*%*@localhost* $1@$2 r
# For dual-id'ed BBSes, treat @one as @other:
*@k5arh* $1@w5ddl r
#
# Next, pass real FQDN's unchanged
*@*.edu $1@$2.edu
*@*.com $1@$2.com
*@*.gov $1@$2.gov
*@*.org $1@$2.org
*@*.net $1@$2.net
*@*.mil $1@$2.mil
```

```
#
# Now addresses we discard with impunity (junk, useless stuff from gatewayed systems)
#eg, astro@* refuse
#eg, *@dist9 refuse
dx@ww refuse
#

# Handle local sysop, and sysop bulls
sysop k5arh
sysop@w5ddl* k5arh
sysop@allla allla
sysop@* sysop
*@sysop sysop
#

# Now pass specific bulletins on to our areas
# NOTE: It is important that the lines below are kept in that order
# (Ie TO sorting FIRST, then AT sorting !!)
# Otherwize something like 'amsat@allusa' will end up in the 'allusa' area
# instead of the 'amsat' area where I prefer it.
tcpip@* tcpip
wanted@* wanted
want@* wanted
need@* wanted
wp@* btrbbs
sale@* sale
4sale@* sale
trade@* sale
swap@* sale
dx@* dx
races@* races
fcc@* fcc
amsat@* amsat
arrl@* arrl
ares@* ares
nasa@* nasa
gulf@* gulf
keps@* keps
larc@* larc
kd5sl@* btrbbs
#
*@gulf gulf
*@nasa nasa
*@amsat amsat
*@ares* ares
*@arrl arrl
*@arl arrl
*@la allla

*@allla allla
*@allus* allus
*@allbbs* allus
*@us* allus
*@usa allus
*@franca franca
```

```
*@ww ww
laoep@* laoep
*@laoep* laoep
#

# LARC is local Lafayette distribution
*@larc larc
*@test test
#

# Anything left @w5ddl is private mail to my users
*@w5ddl* $1
w5ddl@* k5arh

#
# Handle local hosts I forward to:
*@n5knx* n5knx
*@wu3v* wu3v
# Now BBSes we pass north/east/westward
*@wb5bke* bkebbs
*@kb5ogn* ognbbs
*@ka5nmn* ognbbs
*@k4ry* k4ry
*@ka4pkb* ka4pkb
*@w5ksi* norbbs
*@kb4gbs* norbbs
*@kk4cq* norbbs
*@n5ssy* norbbs
*@w4iax* norbbs
*@n5uxt* norbbs
*@wd5ghw* btrbbs
*@kd5sl* btrbbs
*@ae5v* aexbbs
*@ke5l* aexbbs
*@wg5w* aexbbs
*@kb5bfv* lchbbs
*@n2ktq* lchbbs
*@wb5txn* lchbbs
*@kb5tbb* cnbbbs
*@kb5vjy* aexbbs
*@wa4imz* cnbbbs
*@ka5kth* cnbbbs
*@wb5fro* cnbbbs
*@f6cnb* cnbbbs
*@w0gvt* cnbbbs
ka3zyp@* cnbbbs
#
# Handle frequent mistakes or special cases or some other reason I now forget:
71*@* aexbbs
kb5bfv lchbbs
kb5bfv@* lchbbs
ae5v@* aexbbs
kc5gwh@* bkebbs
kc5gwh bkebbs
n5ssy norbbs
```

```
ka5nmn@* ognbbs
ka5ydj@* norbbs
n5ssy@* norbbs
wb5vtn@* norbbs
n5eqo@* norbbs
# <<<others here>>>
# NTS local, in-state, outside
*@705* ntslocal
*@71* aexbbs
*@706* lchbbs
*@70* btrbbs
*@ntstx* cnbbbs
*@nts* nts
# Now state-specific forwarding (if any)
*@*.tx* cnbbbs

#
# Continents via HF at west f6cnb
*@*.eu cnbbbs
*@*.oc cnbbbs
*@*.noam cnbbbs
*@*.na cnbbbs
*@*.usa cnbbbs
*@*.as cnbbbs
*@*.af cnbbbs
*@*.sa cnbbbs

#
# Anything else means we must add more, above
*@* check

#
# Try to handle addressing mistakes by mbox users!
*/* check
*\* check
*&* check
*.* check
```

APPENDIX B: DIALER file

The following dialer script will perform these steps:

```
   1. drop DTR & RTS to force a hangup
   2. wait 2 seconds and then raise DTR & CTS
   3. set the port speed to 9600 baud and initialize the modem
   4. dial a number
   5. turn on continue-after-error mode
   6. wait for the modem to return a CONNECT message
   7. abort if BUSY was received instead of CONNECT, or dial-out failed
   8. try three times to send a CR and obtain a Login: prompt
   9. send my login name, password, wait 5 seconds and then exit
Let's assume a BUSY will always be detected within 10s, and a connect ALWAYS takes
longer.  Also, if we invoked dialer with the ping options, this gives us an eventual
busy redial.  -- n5knx
```

```
# verbose off [enable when well-debugged \ eliminate most output]
control down
wait 2000
control up
speed 9600
send "atz\ratm0l0e0\r"
wait 1000
send "atdt555-1212\r"
failmode on
wait 10000 "BUSY"
ifok exit 1
wait 60000 "CONNECT"
iffail exit
wait 2000
send "\r"
wait 5000 "ogin"
iffail begin
 send "\r"
 wait 5000 "ogin"
 iffail begin
  send "\r"
  wait 5000 "ogin"
  iffail begin
   control down
   exit 1
  end
 end
end
wait 1000
send "myname\r"
wait 5000 "assword"
  iffail begin
   control down
   exit 1
  end
wait 1000
send "mypassword\r"
wait 5000
exit
```

APPENDIX C: Of PACLEN, MTU, MSS, and More

Note: This appendix is taken from the JNOS40 Config Manaual written by Johan Reinalda and William Thompson c. 1994.

Setting Bufsize, Paclen, Maxframe, MTU, MSS and Window

Many NOS users are confused by these parameters and do not know how to set them properly. This appendix will first review these parameters and then discuss how to choose values for them. Special emphasis is given to avoiding interoperability problems that may appear when communicating with non-Nos implementations of AX.25.

1. AX25 Parameters

1.1. Paclen

Paclen limits the size of the data field in an AX.25 I-frame. This value does not include the AX.25 protocol header (source, destination, digipeater addresses, control and pid bytes). The AX.25 V2 protocol specifies a maximum of 256 bytes for the paclen. Be aware that some AX.25 implementations can not handle paclen greater then this, however NOS derived systems can handle this situation. This may cause interoperability problems. Even Nos may have trouble with certain KISS TNCs because of fixed-size buffers. The original KISS TNC code for the TNC-2 by K3MC can handle frames limited in size only by the RAM in the TNC, but some other KISS TNCs cannot. Since unconnected-mode (datagram) AX.25 uses UI frames, the paclen value has no effect in unconnected mode. IE. if you run IP in Datagram mode, you can still have an MTU > Paclen ! (As long as your receive buffers can handle the mtu size; see INTERFACE BUFFERS...)

In JNOS40, and JNOS v1.05 (and greater), paclen can be set on a per interface basis with the 'ifconfig <iface> paclen ###' command. Thus you can have a paclen of 256 on one interface and a smaller or larger on other interfaces. When you first attach an interface, the paclen defaults to the value of the 'ax25 paclen' setting (this defaults to 256.) If you want to carry netrom data over ax.25 links, the system needs to make sure the paclen is large enough to handle the netrom MTU sized data. Net/rom mtu is a maximum of 236; with a 20 byte header for the routing, this gives a data size of 256 to be send with ax.25 packets.

In most versions of NOS.EXE, you can get problems when you use netrom and set the paclen < 256. When the ax.25 layer gets to send netrom frames, it cannot handle the whole data in one packet. It will then split it up in several smaller frames, using the AX.25 Version 2.1 fragmentation scheme. However, TheNet, Net/Rom, G8BPQ, MSYS and other nodes CAN NOT handle this type of fragmentation, resulting in impossible connections ! However, if you are running JNOS40, or JNOS v1.05 (or greater) for the PC, you need not worry about this. These 2 version of NOS have been modified to never provide more then paclen sized netrom data to the ax.25 layer ! Thus the above problem will never happen...

1.2. Maxframe

This parameter controls the number of I-frames that may be send on an AX.25 connection before it must stop and wait for an acknowledgement. Since the AX.25/LAPB sequence number field is 3 bits wide, this number cannot be larger than 7. It can be shown that the optimal value is 1.

Since unconnected-mode (datagram) AX.25 uses UI frames that do not have sequence numbers, this parameter does not apply to unconnected mode.

2. IP and TCP Parameters

2.1. MTU

The MTU (Maximum Transmission Unit) is an interface parameter that limits the size of the largest IP datagram that it may handle. The MTU is the sum of the size of the data inside the IP header (TCP of UDP most likely), and the size of the IP header itself. IP datagrams routed to an interface that are larger than its MTU are each split into two or more IP fragments. Each fragment has its own IP header and is handled by the network as if it were a distinct IP datagram, but when it arrives at the destination it is held by the IP layer until all of the other fragments belonging to the original datagram have arrived. Then they are reassembled back into the complete, original IP datagram. The minimum acceptable interface MTU is 28 bytes: 20 bytes for the IP (fragment) header, plus 8 bytes of data. There is no default MTU in Nos; it must be explicitly specified for each interface as part of the 'attach' command. This is not the case for the netrom interface. Since ip data is carried inside a 'regular' netrom frame, the ip data should never be larger then the maximum data size the netrom protocol can handle. The default mtu here is 236, which is the protocol maximum.

If you plan on running IP in Datagram mode (the default), you can have an MTU that is larger then the paclen for that interface. However, you should still be aware of the constraints of the receiver buffer size! (See INTERFACE BUFFERS)

If you plan on using IP in Virtual Connect, or VC, mode, you should be aware of the following. If you set the IP MTU larger then paclen for the interface, you will hand a packet to the ax.25 layer that is larger then what it can send in one packet. Thus you will get AX.25 V2.1 fragmentation. If you are exchanging ip data with NOS based stations only, you have no problems (other then the fragmentation).

If you are interfacing with stations other then NOS bases systems, you again will have troubles, like with netrom. They will most likely not be able to handle the packets correctly. Thus be aware that in this case you should set the mtu to equal the paclen, to avoid these problems.

2.2. MSS

MSS (Maximum Segment Size) is a TCP-level parameter that limits the amount of data that the remote TCP will send in a single TCP packet. MSS values are exchanged in the SYN (connection request) packets that open a TCP connection. In the Nos implementation of TCP, the MSS actually used by TCP is further reduced in order to avoid fragmentation at the local IP interface. That is, the local TCP asks IP for the MTU of the interface that will be used to reach the destination. It then subtracts 40 from the MTU value to allow for the overhead of the TCP (20 bytes) and IP (20 bytes) headers. If the result is less than the MSS received from the remote TCP, it is used instead.

2.3. Window

This is a TCP-level parameter that controls how much data the local TCP will allow the remote TCP to send before it must stop and wait for an acknowledgment. The actual window value used by TCP when deciding how much more data to send is referred to as the effective window. This is the smaller of two values: the window advertised by the remote TCP minus the unacknowledged data in flight, and the congestion window, an automatically computed time-varying estimate of how much data the network can handle.

2.4. Discussion

2.4.1. IP Fragmentation vs. AX.25 Segmentation

IP-level fragmentation often makes it possible to interconnect two dissimilar networks, but it is best avoided whenever possible. One reason is that when a single IP fragment is lost, all other fragments belonging to the same datagram are effectively also lost and the entire datagram must be retransmitted by the source. Even without loss, fragments require the allocation of temporary buffer memory at the destination, and it is never easy to decide how long to wait for missing fragments before giving up and discarding those that have already arrived. A reassembly timer controls this process. In Nos it is (re)initialized with the 'ip rtimer' parameter (default 30 seconds) whenever progress is made in reassembling a datagram (i.e., a new fragment is received). It is not necessary that all of the fragments belonging to a datagram arrive within a single time-out interval, only that the interval between fragments be less than the time-out.

Most commercial sub networks that carry IP have MTUs of 576 or more, so interconnecting them with sub networks having smaller values can result in considerable fragmentation. For this reason, IP implementors working with links or subnets having unusually small packet size limits are encouraged to use transparent fragmentation, that is, to devise schemes to break up large IP datagrams into a sequence of link or subnet frames that are immediately reassembled on the other end of the link or subnet into the original, whole IP datagram without the use of IP-level fragmentation.

Such a scheme is provided in AX.25 Version 2.1. It can break a large IP or NET/ROM datagram into a series of paclen-sized AX.25 segments (not to be confused with TCP segments), one per AX.25 I-frame, for transmission. Subsequently, it can reassemble them into a single datagram at the other end of the link before handing it up to the IP or NET/ROM module.

Unfortunately, the segmentation procedure is a new feature in AX.25 and is not yet widely implemented; in fact, Nos and derived software, is so far the only known implementation. For regular NOS systems this can create some interoperability problems between Nos and non-Nos nodes. However, JNOS40 and JNOS 1.05 (or later) have provisions built in to avoid these problems. This problem is discussed further in the section on setting the MTU.

2.4.2. Setting MTU

TCP/IP header overhead considerations similar to those of the AX.25 layer when setting paclen apply when choosing an MTU. However, certain sub network types supported by Nos have well-established MTUs, and these should always be used unless you know what you're doing: 1500 bytes for Ethernet, and 508 bytes for ARCNET. The MTU for PPP is automatically negotiated, and defaults to 1500. Other subnet types, including SLIP and AX.25, are not as well standardized.

SLIP has no official MTU, but the most common implementation (for BSD UNIX) uses an MTU of 1006 bytes. Although Nos has no hard wired limit on the size of a received SLIP frame, this is not true for all other systems. Interoperability problems may therefore result if larger MTUs are used in Nos.

Choosing an MTU for an AX.25 interface is more complex. When the interface operates in datagram (UI-frame) mode, the paclen parameter does not apply. The MTU effectively becomes the paclen of the link. However, when using AX.25 CONNECTIONS to send IP packets, data will be automatically segmented into I-frames no larger than paclen bytes. Unfortunately, as also mentioned earlier, Nos is so far the only known implementation of the new AX.25 segmentation procedure. Thus, if you are exchanging IP over connections (i.e. VC mode) with none-NOS based systems, it might be needed to avoid AX.25 segmentation. Thus you should make sure that packets larger than paclen are never handed to AX.25. In order to assure this, you should not set the MTU greater then the paclen.

On the other hand, if you do not send IP over connections, but in datagram mode, you can use a larger MTU. Here, you are limited by the receive buffer size (and the tolerance of other network users for your large packets and proportionally greater bandwidth share !).

For connections carrying IP between NOS systems (i.e. NOS-NOS VC mode), you can let AX.25 segmentation do it's thing. If you choose an MTU on the order of 1000-1500 bytes, you can largely avoid IP-level fragmentation and reduce TCP/IP-level header overhead on file transfers to a very low level. And you are still free to pick whatever paclen value is appropriate for the link.

2.4.5. Setting MSS

The setting of this TCP-level parameter is somewhat less critical than the IP and AX.25 level parameters already discussed, mainly because it is automatically lowered according to the MTU of the local interface when a connection is created. Although this is, strictly speaking, a protocol layering violation (TCP is not supposed to have any knowledge of the workings of lower layers) this technique does work well in practice. However, it can be fooled; for example, if a routing change occurs after the connection has been opened and the new local interface has a smaller MTU than the previous one, IP fragmentation may occur in the local system.

The only drawback to setting a large MSS is that it might cause avoidable fragmentation at some other point within the network path if it includes a "bottleneck" subnet with an MTU smaller than that of the local interface. (Unfortunately, there is presently no way to know when this is the case.

There is ongoing work within the Internet Engineering Task Force on a "MTU Discovery" procedure to determine the largest datagram that may be sent over a given path without fragmentation, but it is not yet complete.) Also, since the MSS you specify is sent to the remote system, and not all other TCPs do the MSS-lowering procedure yet, this might cause the remote system to generate IP fragments unnecessarily.

On the other hand, a too-small MSS can result in a considerable performance loss, especially when operating over fast LANs and networks that can handle larger packets. So the best value for MSS is probably 40 less than the largest MTU on your system, with the 40-byte margin allowing for the TCP and IP headers. For example, if you have a SLIP interface with a 1006 byte MTU and an Ethernet interface with a 1500 byte MTU, set MSS to 1460 bytes. This allows you to receive maximum-sized Ethernet packets, assuming the path to your system does not have any bottleneck subnets with smaller MTUs.

2.4.6. Setting Window

A sliding window protocol like TCP cannot transfer more than one window's worth of data per round trip time interval. So this TCP-level parameter controls the ability of the remote TCP to keep a long "pipe" full. That is, when operating over a path with many hops, offering a large TCP window will help keep all those hops busy when you're receiving data. On the other hand, offering too large a window can congest the network if it cannot buffer all that data. Fortunately, new algorithms for dynamic controlling the effective TCP flow control window have been developed over the past few years and are now widely deployed. Nos includes them, and you can watch them in action with the 'tcp status <tcb>' or 'socket <#>' commands. Look at the cwind (congestion window) value.

In most cases it is safe to set the TCP window to a small integer multiple of the MSS, (e.g. 4times), or larger if necessary to fully utilize a high bandwidth*delay product path. One thing to keep in mind, however, is that advertising a certain TCP window value declares that the system has that much buffer space available for incoming data. Nos does not actually pre-allocate this space; it keeps it in a common pool and may well overbook" it, exploiting the fact that many TCP connections are idle for long periods and gambling that most applications will read incoming data from an active connection as soon as it arrives, thereby quickly freeing the buffer memory. However, it is possible to run Nos out of memory if excessive TCP window sizes are advertised and either the applications go to sleep indefinitely (e.g. suspended Telnet sessions) or a lot of out-of-sequence data arrives. It is wise to keep an eye on the amount of available memory and to decrease the TCP window size (or limit the number of simultaneous connections) if it gets too low.

Depending on the channel access method and link level protocol, the use of a window setting that exceeds the MSS may cause an increase in channel collisions. In particular, collisions between data packets and returning acknowledgments during a bulk file transfer may become common. Although this is, strictly peaking, not TCP's fault, it is possible to work around the problem at the TCP level by decreasing the window so that the protocol operates in stop-and-wait mode. This is done by making the window value equal to the MSS.

2.5. Summary

In most cases, the default values provided by JNOS40 for each of these parameters will work correctly and give reasonable performance. Only in special circumstances such as operation over a very poor link or experimentation with high speed modems should it be necessary to change them.

Interface Buffers

There are two different types of buffers associated with attaching interfaces. The first type is the ring buffer or fifo (first in, first out) that is used when attaching the serial port. It is used for receiving only. This buffer is allocated just once, and is used throughout the life of the interface. The asynchronous (or serial port) receiver interrupt code puts characters in this buffer in a circular fashion. When the end of the buffer is reached, the next character is stored at the beginning and continues through the buffer again.

The receiver process for the serial interface reads the characters from this fifo-buffer into memory buffers, or mbufs, that are used internally to handle and pass around data. Setting the size of the fifo buffer is an empirical process. A good place to start is to set it to twice the size of the packet length used over the serial port. Once you have the interface running, you can monitor the usage of the fifo buffer with the 'asy' command. This will show you the 'buf hi' value which is the same as 'sw hi' for PC based NOS. 'buf hi' is the highest value of the number of characters that were waiting in the fifo buffer to be read by the receiver process. If 'buf hi' is close to the fifo buffer size, or if the 'asy' command shows buffer overflows ('buf over' for JNOS40 code), you should increase the buffer size. If however the number is significantly smaller, you could decrease the buffer size.

The second type of interface buffer is also a receiver buffer. However, this type of buffer is allocated, then released as needed. This buffer almost always is allocated during interrupt service routines, i.e., when the interrupts are off! In order to keep the service routine short, the buffer is allocated from a special 'interrupt buffer queue' because a regular memory allocation would take far too long.

The two internal modem drivers use this second type of buffer. Since one interrupt buffer pool services all the drivers that need buffers during interrupt, the size and number of these buffers are quite critical to a system's 'well-being'. A buffer acquired from the interrupt buffer pool needs to be large enough to handle the largest packet received from any of the internal modem interfaces.

A good rule to estimate the size of the interrupt buffers needed is as follows:

Set 'memory ibufsize' to the (largest + 'extra') of:

1 - the largest ax.25 paclen parameter of the internal modem interfaces, OR

2 - the largest ip mtu parameter of the internal modem interfaces.

PLUS 'extra' which accounts for the longest possible ax.25 header (source, destination, control and digipeaters). Use 80 bytes for' extra' as a general rule.

The MTU is important because on ax.25 interfaces where the MTU is larger than the paclen, there is a possibility of receiving larger ip frames when IP traffic is carried in Datagram mode. This possibility does not exist if IP traffic is carried in Virtual Connect (VC) mode, since the data will be split in several paclen-sized ax.25 packets. (This is AX.25 V2.1 fragmentation at work for you!) When using VC mode you can ignore the MTU values when finding the ibuf size.

The above discussion assumes that paclen and mtu values are coordinated between all users in your area packet network. If this is not the case, someone else might send packets larger then what your system can handle. Such packets will cause receive buffer overflow and will be dumped!
If you have an ax.25 interface with paclen of 256, mtu of 256, and another with paclen of 256 and mtu of 512, you should set the 'memory ibufsize' to at least 512 + 80 ! JNOS40 ibufs default to 600 bytes, and should suffice for paclen's or mtu's up to 512. NOS.EXE has a default ibufsize of 2k (ie 2048 bytes), wich suffices for the standard Ethernet MTU of 1500 bytes.
Determining the number of buffers needed is another empirical process. Start with, say, ten buffers (ie 'memory nibufs 10'). If you get a lot of memory ibuffails in the 'mem stat' display, you should increase the number of buffers.

NOTE: if you are not using one or more of the internal modems or any drivers that require interrupt buffers, there is no need to keep them around. In that case, simply set 'mem nibuf 0'.

APPENDIX C: Information Files via web

JNOS is available via the Internet fromftp://pc.usl.edu/pub/ham/jnos. This directory will contain the latest source zipfiles, executables for evaluation under MSDOS, as well as documentation files. The file README.1ST will describe the files found in the JNOS directory. The docs111.zip file is an archive containing useful JNOS information, such as:

- HELP.ZIP- Mailbox-user command help files, normall extracted into /spool/help.
- NOS.CFG- Use with JNOS -f option to override default file names/locations.
- README.NOW- History of changes in each JNOS release plus information on compiling JNOS, and memory configuration.
- MBOXCMDS.TXT - A quick summary of JNOS mailbox comands.
- CONFIG.ZIP- Sample configurations volunteered by users.
- JDOC_ASC.ZIP- JNOS 1.11 manual in (raw) ASCII format.
- JDOCW6.ZIP- JNOS 1.11 manual in Microsoft Word 6.0 format.
- README.1ST- (This appendix as a stand-alone file).
- MAILBOX.DOC- Info on BBS forwarding and related config files. This information is contained in Appendix C.
- ENV_VAR.DOC- Info on DOS environment variables used by JNOS. This information is also contained in the JNOS 1.11 manuals.
- CMDSHLPx.ZIP- Online console command help, normally extracted into /help.