NET User Reference Manual (NOS Version)

Phil Karn, KA9Q

1. The NET.EXE Program

The MS-DOS executable file **net.exe** provides Internet (TCP/IP), NET/ROM and AX.25 facilities. Because it has an internal multitasking operating system, **net.exe** can act simultaneously as a client, a server and a packet switch for all three sets of protocols. That is, while a local user accesses remote services, the system can also provide those same services to remote users while also switching IP, NET/ROM and AX.25 packets and frames between other client and server nodes.

The keyboard and display is used by the local operator to control both host and gateway level functions, for which a number of commands are provided.

1.1. Installation

Net.exe uses the following directory structure:

```
/spool
/spool/help
/spool/mail
/spool/mqueue
/spool/rqueue
/spool/news
```

By default, the /spool directory is placed in the root directory of the current drive. However, a subdirectory may be specified with the **-d** command-line option described below. If a subdirectory is given, the **alias, autoexec.net, dialer, domain.txt** and **ftpusers** configuration files must also be located there.

The "/spool" directory and its sub-directories are used by the bbs, SMTP and NNTP services. The **areas, forward.bbs, history, mail.log, rewrite** and **signatur** configuration files are located here.

The **alias, forward.bbs** and **rewrite** files are described in the document "maildoc", which should be found at the same location as this file.

1.2. net [-b] [-s <sockets>] [-d <directory>] [<startup file>]

1.2.1. -b

The **-b** option specifies the use of BIOS for console output; the default is to write directly to the video display buffer. Use this option if you are running under a windowing package and have trouble with output "bleeding through" on top of other windows.

1.2.2. -s

The **-s** option specifies the size of the *socket* array to be allocated within **net.exe**. This limits the number of network connections that may exist simultaneously. The default is 40.

1.2.3. -d

The **-d** option allows the user to specify a directory for the configuration and spool files; it defaults to the root directory of the system.

1.2.4. Startup file

After all command-line options, the name of a startup file may be specified. If no startup file is specified, **net.exe** attempts to open a file named **autoexec.net** in the configuration directory of the current drive. If the file exists, it is read and executed as though its contents were typed on the console as commands. (See the **Commands** chapter.) This feature is useful for attaching communication interfaces, configuring network addresses, and starting the various services.

2. Console modes

The console may be in one of two modes: *command mode* and *converse mode*. In *command mode*, the prompt **net>** is displayed and any of the commands described in the **Commands** chapter may be entered. In *converse mode*, keyboard input is processed according to the *current session*.

Sessions come in many types, including *Telnet, FTP, AX25, NETROM, Ping, More, Hopcheck* and *Tip*. In a Telnet, AX25, NETROM, or Tip session, keyboard input is sent to the remote system and any output from the remote system is displayed on the console. In a FTP session, keyboard input is first examined to see if it is a known local command; if so it is executed locally. If not, it is "passed through" to the remote FTP server. (See the **FTP Subcommands** chapter). In a Ping session the user may test the path to a remote site, and in a More session, the user may examine a local file. A Hopcheck session is used to trace the path taken by packets to reach a specified destination. A Tip session provides a "dumb terminal" service that bypasses all network protocols.

The keyboard also has *cooked* and *raw* states. In *cooked* state, input is line-at-a-time; the user may use the line editing characters ^U, ^R and backspace to erase the line, redisplay the line and erase the last character, respectively. Hitting either return or line feed passes the complete line up to the application. In *raw* state, each character is immediately passed to the application as it is typed.

The keyboard is always in *cooked* state in command mode. It is also *cooked* in converse mode on an AX25, FTP or NET/ROM session. In a Telnet session it depends on whether the remote end has issued (and the local end has accepted) the Telnet WILL ECHO option (see the **echo** command).

On the IBM-PC, the user may escape back to *command mode* by hitting the F10 key. On other systems, the user must enter the *escape* character, which is by default control-] (hex 1d, ASCII GS). (Note that this is distinct from the ASCII character of the same name). The *escape* character can be changed (see the **escape** command).

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

3. Commands

This chapter describes the commands recognized in command mode, or within a startup file such as **autoexec.net**. These are given in the following notation:

command
command literal_parameter
command subcommand <parameter>
command [<optional_parameter>]
command a | b

Many commands take subcommands or parameters, which may 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 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. Optional parameters are shown enclosed in [brackets], and a parameter enclosed in <angle brackets> should be replaced with an actual value or string. For example, the notation **<hostid>** denotes an actual host or gateway, which may be specified in one of two ways: as a numeric IP address in dotted decimal notation (eg. 44.0.0.1), or as a symbolic

name listed in the file **domain.txt**.

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.

Certain FTP subcommands (eg. **put, get, dir**, etc) are recognized only in converse mode with the appropriate FTP session; they are not recognized in command mode. (See the **FTP Subcommands** chapter.)

Note that certain commands may have been configured out of a given copy of **net.exe** to save disk and memory. If a command has been configured out, it will not appear in the list produced by the "?" command, nor will it be recognized by the command interpreter.

3.1. <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, **net.exe** remains in command mode.

3.2. !

An alias for the **shell** command.

3.3.

Commands starting with the hash mark (#) are ignored. This is mainly useful for comments in the **autoexec.net** file.

3.4. abort [<session #>]

Abort a FTP **get, put** or **dir** operation in progress. If issued without an argument, the current session is aborted. (This command works only on FTP sessions.) 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. When sending a file, **abort** sends a premature end-of-file. Note that 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.

3.5. 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 (eg. Ethernet, 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.

3.5.1. arp add <hostid> ethernet | ax25 <ethernet address> | <ax25_address>

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.

3.5.2. arp publish <hostid> ethernet | ax25 <ethernet address> | <ax25_address>

This command is similar to the **arp add** command, but system will also respond to any ARP request it sees on the network that seeks the specified address. *Use this feature with great care.*

3.5.3. arp drop <hostid> ax25 | ethernet

Remove the specified entry from the ARP table.

3.5.4. arp flush

Drop all automatically-created entries in the ARP table. Permanent entries are not affected.

3.6. asystat

Display statistics on attached asynchronous communications interfaces (8250 or 16550A), if any. The display for each port consists of three lines. 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, and the speed in bits per second. (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.)

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.

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.)

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).

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).

3.7. attach <hw type> ...

Configure and attach a hardware interface to the system. Detailed instructions for each driver are in the **Attach Commands** chapter. An easy way to obtain a summary of the parameters required for a given device is to issue a partial attach command (eg. **attach packet**). This produces a usage message giving the complete command format.

3.8. ax25 ...

These commands are used to control the AX.25 amateur radio link level protocol.

3.8.1. ax25 blimit [<count>]

Display or set the 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 the backoff reaches the blimit setting it is held at that value, which defaults to 30. 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.

3.8.2. ax25 dest

Display the AX25 destination monitoring database. Each callsign seen in the destination field of an AX25 frame is displayed (most recent first), along with the time since it was last referenced. The time since the same callsign was last seen in the source field of an AX25 frame on the same interface is also shown. If the callsign has never been seen in the source field of a frame, then this field is left blank. (This indicates that the destination is either a multicast address or a "hidden station".)

3.8.3. ax25 digipeat [on | off]

Display or set the digipeater enable flag.

3.8.4. ax25 flush

Clear the AX.25 "heard" list (see **ax25 heard**).

3.8.5. ax25 heard

Display the AX.25 "heard" list. For each interface that is configured to use AX.25, a list of all callsigns heard through that interface is shown, along with a count of the number of packets heard from each station and the interval, in hr:min:sec format, since each station was last heard. The list is sorted in most-recently-heard order. The local station is included in the listing; the packet count reflects the number of packets transmitted. This count will be correct whether or not the modem monitors its own transmissions.

3.8.6. 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 value is in milliseconds. The actual round trip time will be learned by measurement once the connection has been established.

3.8.7. ax25 kick <axcb>

Force a retransmission on the specified AX.25 control block.

3.8.8. 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.

3.8.9. ax25 mycall [<call>]

Display or set the local AX.25 address. The standard format is used (eg. KA9Q-0 or WB6RQN-5). This command must be given before any **attach** commands using AX.25 mode are given.

3.8.10. ax25 paclen [<size>]

Limit the size of I-fields on new AX.25 connections. If IP datagrams or fragments larger than this 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. To have any effect on IP datagrams, this parameter should be less than or equal to the MTU of the associated interface.

3.8.11. 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 timeout 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 timeout 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 timeout 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 timeout might have been caused by a lost acknowledgement. This is obviously a tradeoff, so experiment with the poll threshold setting. The default is 128 bytes, one half the default value of **paclen**.

3.8.12. ax25 reset <axcb>

Delete the AX.25 connection control block at the specified address.

3.8.13. ax25 retry [<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 this fails **retry** times, then the connection is abandoned and all queued data is deleted. A value of 0 means "infinity"; the retry limit is disabled. retry

3.8.14. ax25 route

Display the AX.25 routing table that specifies the digipeaters to be used in reaching a given station.

3.8.14.1. ax25 route add <target> [digis ...]

Add an entry to the AX.25 routing table. An automatic **ax25 route add** is executed if digipeaters are specified in an AX25 **connect** command, or if a connection is received from a remote station via digipeaters. Such automatic routing table entries won't override locally created entries, however.

3.8.14.2. ax25 route drop <target>

Drop an entry from the AX.25 routing table.

3.8.15. 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 are dumped in more detail. Note that the send queue units are frames, while the receive queue units are bytes.

3.8.16. ax25 t3 [<milliseconds>]

Display or set the AX.25 idle "keep alive" timer. Value is in milliseconds.

3.8.17. ax25 version [1 | 2]

Display or set the version of the AX.25 protocol to attempt to use on new connections. The default is 1 (the version that does not use the poll/final bits).

3.8.18. ax25 window [<size>]

Set the number of bytes that can be pending on an AX.25 receive queue beyond which I frames will be answered with RNR (Receiver Not Ready) responses. This presently applies only to suspended interactive AX.25 sessions, since incoming I-frames containing network (IP, NET/ROM) packets are always processed immediately and are not placed on the receive queue. However, when an AX.25 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.

3.9. BOOTP

The bootp client and server are added to KA9Q to provide automatic configuration capabilities. With this suite of extensions, a KA9Q host can automatically configure its IP address, subnet mask, broadcast address, host name, the default gateway, the name servers, and default boot file. This simplifies host configuration.

The bootp server supports dynamic IP address assignment. If a bootp request is made by a host to the server, and the server doesn't have a static record for the PC making the request, an IP address may be assigned from a list of dynamic addresses. This simplifies server configuration, so that machines don't require prior IP address assignment. This is useful in environments such as university dormitories, where network service is provided, and the computers configurations change frequently. When the server list of free addresses reaches a minimum threshold, it will begin attempts to reclaim the address.

The bootp client and server code are written according to RFC 951 and 1048.

3.9.1. bootp [<net_name>] [silent] [noisy]

Send a request to a bootp server, and wait for a reply. On receipt of the server reply, the information is used to configure the host. If a reply is not received, the command will time out. Without arguments, **bootp** sends a request to the first interface in the interface list.

This command requires that there exist a routing entry for the IP broadcast address 255.255.255.255 pointing to the appropriate interface. If the interface uses ARP, there must also be an ARP entry that maps that address to the appropriate link level broadcast address. For example, if you have an Ethernet interface named "ethernet", use the following commands before the **bootp** command:

route add 255.255.255.255 ethernet

arp add 255.255.255.255 ether ff:ff:ff:ff:ff:ff

The following **bootp** subcommands are available:

3.9.1.1. bootp <net_name>

Send a request over the specified network.

3.9.1.2. bootp silent

Set bootp so that it will not print the configuration.

3.9.1.3. bootp noisy

Set bootp so that it will print the configuration.

3.9.2. bootpd [start] [stop] [dns] [dynip] [host] [rmhost] [homedir] [defaultfile] [logfile] [logscreen]

This command starts and stops the bootp server, and sets the configuration for the information it will provide in replies. If the file **bootptab** exists, it will read the file for configuration information. On receipt of a request, if **bootptab** has been changed, the server will reread the file for the changed configuration. The following subcommands are available:

3.9.2.1. bootpd start

Start the bootp server, reading from the file bootptab for configuration information.

3.9.2.2. bootpd stop

Stop the bootp server.

3.9.2.3. bootpd dns

Print the address of the domain name servers supplied in replies.

3.9.2.4. bootpd dns <IP addr of domain name server>...

Set the addresses.

3.9.2.5. bootpd dynip

Print the range and use of the dynamic IP address.

3.9.2.6. bootpd dynip <net_name> <IP address> <IP address>

Set the range of IP address to be used for network netname. These address will be supplied to hosts that are not found in the static record.

3.9.2.7. bootpd dynip <netname> off

Turn off dynamic ip for network interface netname.

3.9.2.8. bootpd host

Print the information in the static host table.

3.9.2.9. bootpd host <hostname> ethernet|ax25 <ethernet addr>|<ax25 addr> <ip addr> [boot file]

Add a host to the host table. The LANSTAR packet drivers provide an Ethernet interface to upper layer applications, so configure a LANSTAR network as an Ethernet.

3.9.2.10. bootpd rmhost <hostname>

Remove host <hostname> from the static host tables.

3.9.2.11. bootpd homedir

Print the default directory for the bootp file name used when the bootp file is not specified in the static host record, and when dynamic addresses are supplied. Default is the null string.

3.9.2.12. bootpd homedir <directory name>

Set the default directory.

3.9.2.13. bootpd defaultfile

Print the default file for the bootp file name used when the bootp file is not specified in the static host record, and when dynamic addresses are supplied. Default is the null string.

3.9.2.14. bootpd defaultfile <filename>

Set the default file.

3.9.2.15. bootpd logfile

Print the status of logging to a log file.

3.9.2.16. bootpd logfile <filename | default> on|off

Sets the file for logging to <filename> or the default, bootplog. Turn logging to that file on or off.

3.9.2.17. bootpd logscreen

Print the status of logging to the screen.

3.9.2.18. bootpd logscreen on|off

Turn logging to the screen on or off.

3.10. cd [<dirname>]

Change the current working directory, and display the new setting. Without an argument, **cd** simply displays the current directory without change. The **pwd** command is an alias for **cd**.

3.11. close [<session>]

Close the specified session; without an argument, close the current session. On an AX.25 session, this command initiates a disconnect. On a FTP or Telnet session, this command sends a FIN (i.e., initiates a close) on the session's TCP connection. This is an alternative to asking the remote server to initiate a close (**QUIT** to FTP, or the logout command appropriate for the remote system in the case of Telnet). When either FTP or Telnet sees the incoming half of a TCP connection close, it automatically responds by closing the outgoing half of the connection. Close is more graceful than the **reset** command, in that it is less likely to leave the remote TCP in a "half-open" state.

3.12. connect <iface> <callsign> [<digipeater> ...]

Initiate a "vanilla" AX.25 session to the specified call sign using the specified interface. Data sent on this session goes out in conventional AX.25 packets with no upper layer protocol. The de-facto presentation standard format is used, in that each packet holds one line of text, terminated by a carriage return. A single AX.25 connection may be used for terminal-to-terminal, IP and NET/ROM traffic. The three types of data are automatically separated by their AX.25 Level 3 Protocol IDs.

Up to 7 optional digipeaters may be given; note that the word **via** is NOT needed. If digipeaters are specified, they are automatically added to the AX25 routing table as though the **ax25 route add** command had been given before issuing the **connect** command.

3.13. delete <filename>

Delete a **filename** in the current working directory.

3.14. detach <iface>

Detach a previously attached interface from the system. All IP routing table entries referring to this interface are deleted, and forwarding references by any other interface to this interface are removed.

3.15. dialer <iface> <seconds> <hostid> <pings> <dialer-file>

Setup an autodialer session for the interface. Whenever the interface is idle for the interval in <seconds>, the autodialer will ping the <hostid>. If there is no answer after <pings> attempts, the autodialer will execute the special commands contained in the <dialer-file>.

If the interval in <seconds> is zero, a previous dialer command process will be removed. If the number of <pings> is zero, the <dialer-file> will be executed without pinging the <hostid>.

The file may have any valid name, and must be located in the configuration directory (see the **Installion** section). The commands in the file are described in the **Dialer Subcommands** chapter.

3.16. dir [<dirname>]

List the contents of the specified directory on the console. If no argument is given, the current directory is listed. Note that this command works by first listing the directory into a temporary file, and then creating a **more** session to display it. After this completes, the temporary file is deleted.

3.17. disconnect [<session #>]

An alias for the **close** command (for the benefit of AX.25 users).

3.18. domain ...

These commands control the operation of the Internet Domain Name Service (DNS).

3.18.1. domain addserver <hostid>

Add one or more domain name server(s) to the list of name servers.

3.18.2. domain dropserver <hostid>

Remove one or more domain name server(s) from the list of name servers.

3.18.3. 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.

3.18.4. domain query <hostid>

Send a query to a domain server asking for all resource records associated with this <hostid>, and list the records.

3.18.5. domain retry [<count>]

Display or set the number of attempts to reach each server on the list during one call to the resolver. If this count is exceeded, a failure indication is returned. If set to 0, the list will cycle forever; this may be useful for unattended operation. The default is 3.

3.18.6. domain suffix [<domain suffix>]

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 (eg. **telnet foo.bar** would NOT be turned into **foo.bar.ampr.org**).

3.18.7. domain trace [on | off]

Display or set the flag controlling the tracing of domain server requests and responses. Trace messages will be seen only if a domain name being sought is not found in the local cache file, **domain.txt**.

3.18.8. domain cache ...

These commands are used for the use of the resource record file **domain.txt**, and the local memory cache.

3.18.8.1. domain cache clean [on | off]

Display or set the flag controlling the removal of resource records from the **domain.txt** file whose time-to-live has reached zero.

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.

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

3.18.8.2. domain cache list

List the current contents of the local memory cache.

3.18.8.3. domain cache size [<count>]

Display or set the nominal maximum size of the local memory cache. The default is 20.

(Note: The cache may be temporarily larger when waiting for new records to be written to the **domain.txt** file.)

3.18.8.4. domain cache wait [<seconds>]

Display or set the interval in seconds to wait for additional activity before updating the **domain.txt** file. The default is 300 seconds (5 minutes).

3.19. echo [accept | refuse]

Display or set the flag controlling client Telnet's response to a remote WILL ECHO offer.

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

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 (eg. 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. 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 congestion control algorithms in TCP). 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.)

3.20. eol [unix | 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 unix mode, carriage returns are translated to line feeds. This command is not necessary with all UNIX systems; use it only when you find that a particular system responds to line feeds but not carriage returns. Only SunOS release 3.2 seems to exhibit this behavior; later releases are fixed.

3.21. escape [<char>]

Display or set the current command-mode escape character in hex. (This command is not provided on the IBM-PC; on the PC, the escape char is always F10.)

3.22. etherstat

Display 3-Com Ethernet controller statistics (if configured).

3.23. exit

Exit the **net.exe** program and return to MS-DOS.

3.24. finger <user@hostid> [<user@hostid> ...]

Issue a network finger request for user **user** at host **hostid**. This creates a client session which may be interrupted, resumed, reset, etc, just like a Telnet client session.

3.25. ftp <hostid>

Open an FTP control channel to the specified remote host and enter converse mode on the new session. Responses from the remote server are displayed directly on the screen. See the **FTP Subcommands** chapter for descriptions of the commands available in a FTP session.

3.26. help

Display a brief summary of top-level commands.

3.27. hop ...

These commands are used to test the connectivity of the network.

3.27.1. hop check <hostid>

Initiate a *hopcheck* session to the specified host. This uses a series of UDP "probe" packets with increasing IP TTL fields to determine the sequence of gateways in the path to the specified destination. This function is patterned after the UNIX *traceroute* facility.

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

3.27.2. hop maxttl [<hops>]

Display or set the maximum TTL value to be used in hop check sessions. This effectively bounds the radius of the search.

3.27.3. hop maxwait [<seconds>]

Display or set the maximum interval that a hopcheck session will wait for responses at each stage of the trace. The default is 5 seconds.

3.27.4. hop queries [<count>]

Display or set the number of UDP probes that will be sent at each stage of the trace. The default is 3.

3.27.5. hop trace [on | off]

Display or set the flag that controls the display of additional information during a hop check session.

3.28. 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 that it does NOT set the system's IP address.

If <name> is the same as an <iface> (see the **Attach commands** chapter), this command will search for a CNAME domain resource record which corresponds to the IP address of the <iface>.

3.29. hs

Display statistics about the HS high speed HDLC driver (if configured and active).

3.30. icmp ...

These commands are used for the Internet Control Message Protocol service.

3.30.1. icmp echo [on | off]

Display or set the flag controlling the asynchronous display of ICMP Echo Reply packets. This flag must be on for one-shot pings to work (see the **ping** command.)

3.30.2. icmp status

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

3.30.3. icmp trace [on | off]

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 option should be turned off before using the **hop check** facility because it relies on ICMP Time Exceeded messages, and the asynchronous display of these messages will be mingled with **hop check** command output.

3.31. ifconfig

Display a list of interfaces, with a short status for each.

3.31.1. ifconfig <iface>

Display an extended status of the interface.

3.31.2. ifconfig <iface> broadcast <address>

Set the broadcast address for the interface. The <address> takes the form of an IP address with 1's in the host part of the address. This is related to the **netmask** sub-command. See also the **arp** command.

3.31.3. ifconfig <iface> encapsulation <name>

Not fully implemented.

3.31.4. ifconfig <iface> forward <forward-iface>

Set a forwarding interface for multiple channel interfaces. To remove the forward, set <forward-iface> to <iface>.

3.31.5. ifconfig <iface> ipaddress <hostid>

Set the IP address for this interface. It is standard Internet practice that each interface has its own address. For hosts with only one interface, the interface address is usually the same as the host address. See also the **hostname** and **ip address** commands.

3.31.6. ifconfig <iface> linkaddress <hardware-dependant>

Set the hardware dependant address for this interface.

3.31.7. ifconfig <iface> mtu <mtu>

Set the MTU for this interface. See the **Setting ... MTU, MSS and Window** chapter for more information.

3.31.8. ifconfig <iface> netmask <address>

Set the sub-net mask for this interface. 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. This is related to the **broadcast** sub-command. See also the **route** command.

3.31.9. ifconfig <iface> rxbuf <?>

Not yet implemented.

3.32. ip ...

These commands configure the Internet Protocol (IP) service.

3.32.1. ip address [<hostid>]

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.

3.32.2. ip rtimer [<seconds>]

Display or set the IP reassembly timeout. The default is 30 seconds.

3.32.3. ip status

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

3.32.4. ip ttl [<hops>]

Display or set the time-to-live value placed in each outgoing IP datagram. This limits the number of switch hops the datagram will be allowed to take. The idea is to bound the lifetime of the packet should it become caught in a routing loop, so 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 the official recommended value for the Internet.

3.33. isat [on | off]

Display or set the AT flag. Currently, there is no sure-fire way to determine the type of clock-chip being used. If an AT type clock is in use, this command will allow measurement of time in milliseconds, rather than clock ticks (55 milliseconds per clock tick).

3.33.1. kick [<session>]

Kick all sockets associated with a session; if no argument is given, kick the current session. Performs the same function as the **ax25 kick** and **tcp kick** commands, but is easier to type.

3.34. log [stop | <filename>]

Display or set the **filename** for logging server sessions. If **stop** is given as the argument, logging is terminated (the servers themselves are unaffected). If a file name is given as an argument, server session log entries will be appended to it.

3.35. mbox

Display the status of the mailbox server system (if configured).

3.36. memory ...

These commands are used to display memory allocation statistics.

3.36.1. memory free

Display the storage allocator free list. Each entry consists of a starting address, in hex, and a size, in decimal bytes.

3.36.2. memory ibuffs

Display or set the number of buffers on the interrupt buffer pool. The default is 5.

3.36.3. memory ibufsize

Display or set the size of each buffer on the interrupt buffer pool. Since the interrupt buffer pool consists of fixed-size buffers, the value chosen must be large enough to satisfy the needs of the most demanding driver. The default is 2048.

3.36.4. memory sizes

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

3.36.5. memory status

Display a summary of storage allocator statistics. The first line shows the base address of the heap, its total size, the amount of heap memory available in bytes and as a percentage of the total heap size, and the amount of memory left over (i.e., not placed on the heap at startup) and therefore available for **shell** subcommands.

The second line shows the total number of calls to allocate and free blocks of memory, the difference of these two values (i.e., the number of allocated blocks outstanding), the number of allocation requests that were denied due to lack of memory, and the number of calls to free() that attempted to free garbage (eg. by freeing the same block twice or freeing a garbled pointer).

The third line shows the number of calls to malloc and free that occurred with interrupts off. In normal situations these values should be zero. The fourth line shows statistics for the special pool of fixed-size buffers used to satisfy requests for memory at interrupt time. The variables shown are the number of buffers currently in the pool, their size, and the number of requests that failed due to exhaustion of the pool.

3.37. mkdir <dirname>

Create a sub-directory in the current working directory.

3.38. mode <iface> [vc | datagram]

Control the default transmission mode on the specified AX.25 interface. In **datagram** mode, IP packets are encapsulated in AX.25 UI frames and transmitted without any other link level mechanisms, such as connections or acknowledgements.

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 connections are opened if necessary.

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

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) are 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).

3.39. more <file> [<file> ...]

Display the specified file(s) a screen at a time. To proceed to the next screen, press the space bar; to cancel the display, hit the 'q' key. The **more** command creates a session that you can suspend and resume just like any other session.

3.40. param <iface> [<param> [value]] ...

Invoke a device-specific control routine. The following parameter names are recognized by the parameter command, but not all are supported by each device type. Most commands deal only with half-duplex packet radio interfaces.

TxDelay - transmit keyup delay
Persist - P-persistence setting
SlotTime - persistence slot time setting
txTail - transmit done holdup delay
FullDup - enable/disable full duplex
Hardware - hardware specific command
TxMute - experimental transmit mute command
DTR - control Data Terminal Ready (DTR) signal to modem
RTS - control Request to Send (RTS) signal to modem
Speed - set line speed
EndDelay
Group
Idle
Min
MaxKey
Wait
Down - drop modem control lines
Up - raise modem control lines
Return - return a KISS TNC to command mode

Depending on the interface, some parameters can be read back by omitting a new value. This is not possible with KISS TNCs as there are no KISS commands for reading back previously sent parameters.

On a KISS TNC interface, the **param** command generates and sends control packets to the TNC. Data bytes are treated as decimal. For example, **param ax0 txdelay 255** will set the keyup timer (type field = 1) on the KISS TNC configured as ax0 to 2.55 seconds (255 x .01 sec). On all **asy** interfaces (slip, kiss/ax25, nrs, ppp) the **param <iface> speed** command allows the baud rate to be read or set.

The implementation of this command for the various interface drivers is incomplete and subject to change.

3.41. ping <hostid> [<length> [<seconds> [<incflag>]]]

Ping (send ICMP Echo Request packets to) the specified host. By default the data field contains only a small timestamp to aid in determining round trip time; if the optional **length** argument is given, the appropriate number of data bytes (consisting of hex 55) are added to the ping packets.

If interval is specified, pings will be repeated indefinitely at the specified number of seconds; otherwise a single, "one shot" ping is done. Responses to one-shot pings appear asynchronously on the command screen, while repeated pings create a session that may be suspended and resumed. Pinging continues until the session is manually reset.

The **incflag** option causes a repeated ping to increment the target IP address for each ping; it is an experimental feature for searching blocks of IP addresses for active hosts.

3.42. ppp ...

These commands are used to configure Point to Point Protocol interfaces.

This implementation of PPP is designed to be as complete as possible. Because of this, the number of options can be rather daunting. However, a typical PPP configuration might include the following commands:

```
attach asy 0x3f8 4 ppp pp0 4096 1500 9600
dial pp0 30 <hostid> 3 dialer.pp0
#
ppp pp0 lcp local accm 0
ppp pp0 lcp local compress address on
ppp pp0 lcp local compress protocol on
ppp pp0 lcp local magic on
ppp pp0 lcp open active
#
ppp pp0 ipcp local compress tcp 16 1
ppp pp0 ipcp open active
#
route add default pp0
```

3.42.1. ppp <iface>

Display the status of the PPP interface.

3.42.2. ppp <iface> lcp ...

These commands are used for the LCP [Link Control Protocol] configuration.

3.42.2.1. ppp <iface> lcp close

Shutdown the PPP interface.

3.42.2.2. ppp <iface> lcp local ...

These commands control the configuration of the local side of the link. If an option is specified, the parameters will be used as the initial values in configuration requests. If not specified, that option will not be requested.

For each of these options, the **allow** parameter will permit the remote to include that option in its response, even when the option is not included in the request. By default, all options are allowed.

3.42.2.2.1. ppp <iface> lcp local accm [<bitmap> | allow [on | off]]

Display or set the Async Control Character Map. The default is 0xffffffff.

3.42.2.2.2. ppp <iface> lcp local authenticate [pap | none | allow [on | off]]

Display or set the authentication protocol. The default is **none**.

3.42.2.2.3. ppp <iface> lcp local compress address/control [on | off | allow [on | off]]

Display or set the option to compress the address and control fields of the PPP HLDC-like header. This is generally desirable for slow asynchronous links, and undesirable for fast or synchronous links. The default is off.

3.42.2.2.4. ppp <iface> lcp local compress protocol [on | off | allow [on | off]]

Display or set the option to compress the protocol field of the PPP HLDC-like header. This is generally desirable for slow asynchronous links, and undesirable for fast or synchronous links. The default is off.

3.42.2.2.5. ppp <iface> lcp local magic [on | off | <value> | allow [on | off]]

Display or set the initial Magic Number. The default is off (zero).

3.42.2.2.6. ppp <iface> lcp local mru [<size> | allow [on | off]]

Display or set the Maximum Receive Unit. The default is 1500.

3.42.2.2.7. ppp <iface> lcp local default

Reset the options to their default values.

3.42.2.3. ppp <iface> lcp open active | passive

Wait for the physical layer to come up. If **active**, initiate configuration negotiation. If **passive**, wait for configuration negotiation from the remote.

3.42.2.4. ppp <iface> lcp remote ...

These commands control the configuration of the remote side of the link. The options are identical to those of the local side. If an option is specified, the parameters will be used in responses to the remote's configuration requests. If not specified, that option will be accepted if it is allowed.

For each of these options, the **allow** parameter will permit the remote to specify that option in its request. By default, all options are allowed.

3.42.2.5. ppp <iface> lcp timeout [<seconds>]

Display or set the interval to wait between configuration or termination attempts. The default is 3 seconds.

3.42.2.6. ppp <iface> lcp try ...

These commands are used for the various counters.

3.42.2.6.1. ppp <iface> lcp try configure [<count>]

Display or set the number of configuration requests sent. The default is 10.

3.42.2.6.2. ppp <iface> lcp try failure [<count>]

Display or set the number of bad configuration requests allowed from the remote. The default is 5.

3.42.2.6.3. ppp <iface> lcp try terminate [<count>]

Display or set the number of termination requests sent before shutdown. The default is 2.

3.42.3. ppp <iface> ipcp ...

These commands are used for the IPCP [Internet Protocol Control Protocol] configuration.

The **close, open, timeout** and **try** sub-commands are identical to the LCP (described above).

3.42.3.1. ppp <iface> ipcp local ...

These commands control the configuration of the local side of the link. If an option is specified, the parameters will be used as the initial values in configuration requests. If not specified, that option will not be requested.

For each of these options, the **allow** parameter will permit the remote to include that option in its response, even when the option is not included in the request. By default, all options are allowed.

3.42.3.1.1. ppp <iface> ipcp local address [<hostid> | allow [on | off]]

Display or set the local address for negotiation purposes. If an address of 0 is specified, the other side of the link will supply the address. By default, no addresses are negotiated.

3.42.3.1.2. ppp <iface> ipcp local compress [tcp <slots> [<flag>] | none | allow [on | off]]

Display or set the compression protocol. The default is **none**.

The **tcp** <slots> specifies the number of "conversation" slots, which must be 1 to 255. (This may be limited at compilation time to a smaller number.) A good choice is in the range 4 to 16.

The **tcp** <flag> is 0 (don't compress the slot number) or 1 (OK to compress the slot number). KA9Q can handle compressed slot numbers, so the default is 1.

3.42.3.2. ppp <iface> ipcp remote ...

These commands control the configuration of the remote side of the link. The options are identical to those of the local side. If an option is specified, the parameters will be used in responses to the remote's configuration requests. If not specified, that option will be accepted if it is allowed.

For each of these options, the **allow** parameter will permit the remote to specify that option in its request. By default, all options are allowed.

3.42.4. ppp <iface> pap ...

These commands are used for the PAP [Password Authentication Protocol] configuration.

The **timeout** and **try** sub-commands are identical to the LCP (described above). However, the terminate counter is unused.

3.42.4.1. ppp <iface> pap user [<username> [<password>]]

Display or set the username (the password may be set, but not displayed). When the username is specified, but no password is supplied, the **ftpusers** file is searched for the password. When a username/password is unknown or rejected, a session will appear at the console to prompt for a new username/password.

3.42.5. ppp <iface> trace [<flags>]

Display or set the flags that control the logging of information during PPP link configuration.

The flag value is 0 for none, 1 for basic, and 2 for general. Values greater than 2 are usually not compiled, and are described in the appropriate source files where they are defined.

3.43. ps

Display all current processes in the system. The fields are as follows:

PID - Process ID (the address of the process descriptor).

SP - The current value of the process stack pointer.

stksize - The size of the stack allocated to the process.

maxstk - 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 reaches or exceeds the stksize figure, the system is almost certain to crash; the **net.exe** program should be recompiled to give the process a larger allocation when it is started.

event - The event this task is waiting for, if it is not runnable.

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.)

3.44. pwd [<dirname>]

An alias for the **cd** command.

3.45. record [off | <filename>]

Append to **filename** all data received on the current session. Data sent on the current session is also written into the file except for Telnet sessions in remote echo mode. The command **record off** stops recording and closes the file.

3.46. remote [-p <port>] [-k <key>] [-a <kickaddr>] <hostid> exit | reset | kick

Send a UDP packet to the specified host commanding it to exit the **net.exe** program, reset the processor, or force a retransmission on TCP connections. For this command to be accepted, the remote system must be running the **remote** server and the port number specified in the **remote** command must match the port number given when the

server was started on the remote system. If the port numbers do not match, or if the remote server is not running on the target system, the command packet is ignored. Even if the command is accepted there is no acknowledgement.

The **kick** command forces a retransmission timeout on all TCP connections that the remote node may have with the local node. If a connection is idle, a current ACK packet (without data) is sent. If the -a option is used, connections to the specified host are kicked instead. No key is required for the kick subcommand.

The **exit** and **reset** subcommands are mainly useful for restarting the **net.exe** program on a remote unattended system after the configuration file has been updated. The remote system should invoke the **net.exe** program automatically upon booting, preferably in an infinite loop. For example, under MS-DOS the boot disk should contain the following in **autoexec.net**: