

# **PC/IP User's Guide**

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**

**Laboratory For Computer Science**

**Network programs based on the DoD Internet Protocol  
for the IBM Personal Computer**

**PC/IP release of March, 1986; document updated April 14, 1986**

**by: Jerome H. Saltzer  
John L. Romkey**

Copyright 1984, 1985, 1986 by the Massachusetts Institute of Technology

Permission to use, copy, modify, and distribute these programs and their documentation for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies and supporting documentation, the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the programs without written prior permission, and notice be given in supporting documentation that copying and distribution is by permission of M.I.T. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

## CREDITS

The PC/IP packages are built on the work of many people in the TCP/IP community, both at M.I.T. and elsewhere. Following are some of the people who directly helped in the creation of the packages.

Network environment--John L. Romkey

Terminal emulator and customizer--David A. Bridgham

Initial TFTP--Karl D. Wright

Initial telnet--Louis J. Konopelski

Telnet model--David D. Clark

Tasking package--Larry W. Allen

Development system--Christopher J. Terman

Development environment--Wayne C. Gramlich

Administrative Assistant--Muriel Webber

October 3, 1985. This document is in cover.mss



# Table of Contents

1. Overview of PC/IP network programs	1
1.1. Software environment	1
1.2. Hardware environment	3
1.3. Customization	4
2. Technical Notes on PC/IP	5
2.1. Device Drivers	5
2.2. IP	5
2.3. UDP	6
2.4. TCP	6
2.5. Hostname resolution	7
2.6. TFTP	7
2.7. LPR	7
2.8. TERMINAL EMULATOR	8
3. Other documentation	9
4. Changes From The Last Release	13
5. Changes In Prior Releases	19
6. Software Installation	25
7. Hardware Installation	29
7.1. Pronet jumper selection	30
7.2. Interlan NI5010 jumper selection	30
7.3. 3COM Etherlink Interface	31
7.4. Memory expansion card flaw	33
8. Ethernet Etiquette	35
9. Host Names and Internet Addresses	37
10. Debugging options	39
11. Dialup line file transfer	43
12. Custom	45
12.1. Standard customization parameters	45
12.2. Site customization of network level parameters	46
12.3. Personal customization of terminal emulation options	48
12.4. Other parameters	48
12.5. On-the-fly error correction	49
12.6. Temporary customization	49
13. Iprint	51
13.1. Customization	51
13.2. Notes	51
14. Lpr	53
14.1. Customization	55
14.2. Notes	55
15. Monitor	57
15.1. Control file	58
15.2. User commands	59

## PC/IP User's Guide

15.3. Display modes	60
15.4. Bugs	60
15.5. Customization	60
16. Netname	61
16.1. Customization	62
17. Netwatch	63
17.1. Filtering	65
17.2. Combining Filters	65
17.3. Performance	66
18. Nicname	67
19. Onetname	69
19.1. Customization	69
20. Onhook and Offhook	71
21. Ping	73
21.1. Optional features	73
21.2. Using ping to isolate network trouble	74
21.3. Using ping to evaluate a serial line	74
22. Setclock	77
22.1. Customization	78
23. Telnet	79
23.1. Closing connections	80
23.2. Terminal emulation conventions and compatibility	81
23.3. Heath H19 features not emulated	82
23.4. File transfer with PC/telnet	82
23.5. Nested calls to the DOS shell	83
23.6. Customization	84
23.7. DOS note	84
23.8. Debugging note	84
24. Term	85
24.1. Operating instructions	85
24.2. Configuration customization	86
25. TFTP	87
25.1. Notes	87
26. TFTP Server	91
27. Whols	93
28. Status	95
29. PC/IP bugs, tasks, problems, projects, and bright ideas	97

# 1. Overview of PC/IP network programs

The PC/IP network programs are a set of commands that operate under the DOS operating system for the IBM Personal Computer. They provide a set of facilities that make the PC a directly attachable network host rather than a simulated intelligent terminal.

The PC/IP programs have their origin in an M.I.T. research project on protocol effectiveness. In consequence, they incorporate several second- and third-generation protocol implementation techniques and algorithms. These techniques and algorithms:

- reduce the processing load on the computer
- maximize opportunities for parallel operation of the network and the computer
- minimize unnecessary retransmission of data
- eliminate certain pathological situations in which two technically compatible machines communicate very inefficiently.

One result of application of these techniques is that a complete host implementation can operate with high effectiveness in a machine as small as the PC.

## 1.1. Software environment

The PC/IP programs are a set of commands that operate under IBM PC-DOS versions 2.0, 2.1, 3.0, or 3.1. A device driver must be installed, but no changes are required to the operating system. Ethernet versions of the PC/IP programs have been verified to work under TopView.

These programs all use the ARPANET standard end-to-end Internet Protocol, IP, and can be used to communicate with any other host that also uses that protocol. Individual commands use various higher-level ARPANET standard protocols from the IP family, as appropriate:

TCP                    for reliable byte stream transmission

## PC/IP User's Guide

UDP                for datagram service

Telnet             for remote login

ICMP               for control messages

TFTP               for file transfer

name lookup service protocol

error and event logging protocol

time-of-day and calendar service protocol

Thus these programs are directly useful only in a network environment where there are other hosts that implement one or more IP-family protocols [Thus the programs are not useful on a network where all other hosts implement only the SNA LU6.2, A.I. Laboratory CHAOS, DECNET, or Xerox NS protocol families.]

There is one limitation in the PC/IP protocol implementation that may affect usage in some environments: reassembly of fragmented packets is not supported. If one anticipates communications with a host that is accessible only via networks that require small packets, this limitation may be a problem.

In addition to the protocols mentioned above, PC/IP network programs make use, if available, of several network services commonly found in IP network environments.

These services include:

- name-to-host-address translation service
- IP gateways to other IP networks
- time-of-day and calendar services
- error logging service
- printer service

If any of these services is not available, it is still possible to use the PC/IP network programs, though with loss of certain convenience features.

## 1.2. Hardware environment

The PC/IP programs operate on a standard IBM PC, PC/XT, or PC/AT. They have also been reported to work on a COMPAQ IBM-compatible PC. Under DOS 2.0, they require 128 kbytes of memory, one disk drive, 80-column display (the IBM Monochrome display card, Color Graphics Adapter or Professional Graphics Adapter) and any of the following kinds of hardware network attachment: RS-232 port, Interlan NI5010 Ethernet<sup>1</sup>, 3COM Etherlink<sup>2</sup> Ethernet (not the new, smart card), or Proteon proNET<sup>3</sup>.

When an RS-232 port is used, the other end of the line can go (either by dialup or by direct wiring) to another PC or a gateway that forwards packets to and from a local area network. The link-level protocol used is a non-standard one designed to allow flow control, buffer management, and packet-to-packet redundancy compression on a full-duplex line. To simplify forwarding of packets destined for an RS-232 attached PC, the gateway assigns the Internet host address of the PC. The PC asks the gateway for its assigned address using another non-standard protocol. The port may be used at any standard data rate from 300 bits/sec. to 19.2 kbits/sec. Note, however, that highly interactive services, such as character-at-a-time remote echoing, are not very effective at data rates below 9.6 kbits/sec. The RS-232 port does not work under TopView at speeds of 9.6 kbits/sec. and higher.

The Ethernet versions of the PC/IP programs provide a driver for the Interlan NI5010 interface and the 3COM Etherlink interface<sup>4</sup>. Since Ethernet addresses do not map directly into Internet host addresses, the Ethernet driver uses ARP, the IP standard Ethernet-to-Internet address translation protocol. If this protocol is not implemented by other hosts, it is possible, by use of a customization option, to supply manually a limited number of Ethernet-to-Internet address bindings.

---

<sup>1</sup>*Ethernet* is a registered trademark of the Xerox Corporation.

<sup>2</sup>*Etherlink* is a registered trademark of 3COM Corporation.

<sup>3</sup>*proNET* is a registered trademark of Proteon, Inc.

<sup>4</sup>The PC/IP Etherlink driver does not use the 3COM software or device driver, and it does not require that hardware switches be set to simulate availability of four disk drives. However, if the environment contains the 3COM software or switch settings, the PC/IP Ethernet driver will still operate correctly.

The proNET versions of the PC/IP programs provide a driver for the Proteon, Inc., proNET ring interface.

There are four versions of each PC/IP command, one version for each of kind of network interface supported.

### **1.3. Customization**

A customization program, named **custom**, sets certain parameters in a DOS device driver that is used by each PC/IP network program. Some of these parameters, such as serial line speed, cannot otherwise be discovered by the software. Others, such as the preferred modes of operation of the remote login program, depend on characteristics of the distant host most often used. Still others, such as the Internet addresses of name servers, are site-dependent. Details of which parameters may be set for each program are found in the descriptions of the individual programs and in the description of **custom**.

28 September 1985. This document is in file `overview.mss`

## 2. Technical Notes on PC/IP

This section discusses technical details about the implementation of PC/IP and interactions between PC/IP and other TCP/IP implementations (especially Berkeley 4.2 Unix). Casual users of PC/IP can skip this section unless interested; people installing PC/IP at a site should definitely read this section.

### 2.1. Device Drivers

Both the Interlan and 3COM ethernet drivers use the Address Resolution Protocol (ARP), as described in NIC RFC 826. The proNET driver does not support ARP.

Some TCP/IP implementations encapsulate IP packets in a non-standard fashion called a *trailer*, as specified in NIC RFC 893. PC/IP does not support trailers with any of its drivers; instead, it supports only the standard form of encapsulation as specified in NIC RFC 894. The only implementations that are known to send trailers are 4.2 UNIX<sup>5</sup> derivatives (Wollongong's VMS TCP/IP, for instance, is derived from 4.2's). The 4.2 Unix command *ifconfig(8)* can control trailer usage on a per-interface basis.

The maximum length packet PC/IP is prepared to send or receive is 820 bytes long, including the local net header.

### 2.2. IP

The IP layer does not implement packet fragmentation or reassembly. If a packet fragment is received, it is discarded. Options are never sent, and incoming options are ignored. Type of service is ignored.

Incoming destination unreachables and other errors will be printed on the display if the NETERR or PROTERR debugging flags are turned on (see the section on debugging). IP protocol or TCP or UDP socket, unless those packets were broadcast. Packets are determined to be broadcast by the old convention of having the host field be all 0's.

---

<sup>5</sup>UNIX is a trademark of American Telephone and Telegraph Co."

Routing is done according to RFC 950. The user specifies the PC's network address and the number of bits in its subnet field with the `PC/custom` command. This program computes a mask that can be used to separate the net/subnet portion of the address from the host portion. Two machines are determined to be on the same physical network if the net/subnet parts of both their addresses are identical. If PC/IP tries to send to a machine that is not on the same physical network, it routes the packet through the default gateway, also specified with `custom`. If PC/IP receives an ICMP redirect from the gateway, it records the redirect in internal routing tables, which it scans before using the default gateway.

### 2.3. UDP

PC/IP has a complete UDP implementation, including checksums.

The 4.2 Unix UDP had a number of problems that have been fixed in 4.3. Among other things, checksums were computed incorrectly, so PC's would not accept packets from 4.2 machines.

### 2.4. TCP

The TCP is a single connection implementation tailored to Telnet. It sends `MAXBUFSIZE` options setting the maximum buffer size to 511 bytes to defeat trailers (discussed above; trailer packets have a multiple of 512 bytes of data). It ignores incoming `MAXBUFSIZE` options.

A number of TCP problems have been fixed since the January 1985 release. One of the most noticeable was an incompatibility between 4.2 Unix's TCP and PC/IP's. The 4.2 TCP sent probing messages called "keepalives" when a connection was otherwise inactive. PC/IP did not respond to these messages the way 4.2 expected, and 4.2 would decide that the PC was down and close the connection. PC/IP's TCP now responds as 4.2 expects.

### 3. Other documentation

This section provides an annotated list of other documents that describe or pertain to PC/IP.

1) Saltzer, J.H., et al., "The Desktop Computer as a Network Participant," IEEE Selected Areas in Communications, May, 1985, pp 468-478.

Discussion of the implementation may be found in:

2) Romkey, John L., "PC/IP Programmer's Manual"

The following undergraduate thesis describes the first implementation of a file transfer protocol package. Although that package has been superseded, there are still several points of design strategy that carry over into various PC/IP packages.

3) Wright, Karl D., "A File Transfer Program for a Personal Computer," S. B. Thesis, M.I.T. Department of Electrical Engineering and Computer Science, April, 1982. Also available as M.I.T. L.C.S. Technical Memorandum TM-217.

The following undergraduate thesis describes the TCP/Telnet package. This package is still in use, though the thesis describes an early implementation.

4) Konopelski, Louis J., "Implementing Internet Remote Login on a Personal Computer," S. B. Thesis, M.I.T. Department of Electrical Engineering and Computer Science, December, 1982. Also available as M.I.T. L.C.S. Technical Memorandum TM-233.

Much of the PC/IP implementation was influenced by the ideas of David D. Clark documented in the "Internet Protocol Implementation Guide," August, 1982, SRI International, Menlo Park, California. Five parts of this document are of particular interest:

- 5) Window and Acknowledgement Strategy in TCP (RFC 813)
- 6) Names, Addresses, Ports and Routes (RFC 814)

## PC/IP User's Guide

- 7) IP Datagram Reassembly Algorithms (RFC 815)
- 8) Fault Isolation and Recovery (RFC 816)
- 9) Modularity and Efficiency in Protocol Implementation (RFC 817)

The protocols used in the PC/IP packages are specified in the "Internet Protocol Transition Workbook", March, 1982, available from SRI International. The particular protocol documents are:

- 10) Internet Protocol (RFC-791)
- 11) Internet Control Message Protocol (RFC-792)
- 12) User Datagram Protocol (RFC-768)
- 13) Transmission Control Protocol (RFC-793)
- 14) Telnet Protocol (RFC-764)
- 15) Trivial File Transfer Protocol (RFC-783)
- 16) Name Server Protocol (IEN-116)
- 17) Time Server Protocol (RFC-868)
- 18) Ncname/Whois server (RFC-812)
- 19) Echo Protocol (RFC-862)

One other protocol is described in the ARPANET Protocol handbook of January, 1978:

- 20) Finger protocol (NIC-42758 or RFC-742)

The domain name system and name resolution protocol are described in the following documents:

- 21) Domain Names - Concepts and Facilities (RFC-882)
- 22) Domain Names - Implementation Specification (RFC-883)

The Supdup remote login protocol is described by Mark Crispin in:

- 23) Supdup (RFC 734)

The 4.2 Unix printer spooling protocol is described in the PC/IP Programmer's Manual mentioned above, and is also described by Ralph Campbell in:

- 24) 4.2BSD Line Printer Spooler Manual (Unix documentation)

The subnet routing scheme used by PC/IP is described by Jeff Mogul and Jon Postel in:

25) Internet Standard Subnetting Procedure (RFC 950)

The method for encapsulating IP packets on an ethernet is as specified by Charles Hornig in:

26) A Standard for the Transmission of IP Datagrams over Ethernet Networks (RFC 894)

The Address Resolution Protocol, used only by the ethernet drivers, is as specified by David Plummer in:

27) An Ethernet Address Resolution Protocol (RFC 826)

The protocol used to send files to the Imagen print server is described in:

28) Imprint-10 Programmer's Manual, Imagen Corp. April, 1984.

The following document describes a transcription of PC/IP into Pascal, for use on the Apple Macintosh computer and Applebus:

29) Sherman, Mark, "A Network Package for the Macintosh using DoD Internet Protocols," Department of Mathematics and Computer Science, Dartmouth College, New Hampshire.

29 December 1985. This document is in file otherdoc.mss



## 4. Changes From The Last Release

This section describes user-visible changes since the January, 1985, release of the PC/IP packages.

### A. Changes to user commands.

#### PC/custom

1. Now allows the user to select transmit and receive DMA channels.
2. Accepts interface base I/O address in hexadecimal instead of decimal.
3. A new flag was added to specify the preferred output radix for IP addresses (decimal or octal).
4. Now recomputes the subnet mask when either the number of subnet bits changes or the internet address changes (address could have changed class).
5. Permits the user to separately control debug tracing of different protocol levels.
6. The office and phone number fields were removed and the space reused.
7. Added a domain name field and the addresses of up to three domain name servers. The number of old-style name servers was reduced to two.

#### PC/netwatch

1. Can now match on several layers of types, and can match on protocol source and destination addresses.
2. Now has an explicit *pause* command and also pauses when printing help message.
3. Packets are displayed in color when possible.
4. Records the number of packets accepted of each type.
5. Records the number of packets accepted by length (in quanta of 64 bytes).
6. Records the number of broadcast packets.
7. Ethernet versions can display the manufacturer of an ethernet interface when printing the address.

## PC/IP User's Guide

8. Can print local net addresses when printing packets symbolically.
9. Works with proNET driver.
10. Can match on source, destination or source/or/destination address (hardware or protocol).

### PC/setclock

1. PC/setclock now automatically adjusts for daylight savings time according to 1985 U.S. law.

### PC/telnet

1. Changed user interface to the debugging and statistics printing commands, allowing a larger repertoire. All debugging commands now are obtained by F10/control-something; a list of them is displayed in response to F10/control-h.

### PC/tftp

1. Greeting message now mentions the mode of the transfer, so that default of netascii does not trap unwitting user.
2. Packet buffer allocation now is preceded by flushing out broadcast packets to allow TCP packet processing. Formerly, if a lot of broadcasts or character-at-a-time messages from a hyperactive UNIX filled up the packet buffers during initialization of tftp, it couldn't get any free buffers and would have to give up.
3. Disk writes are now collected into a 10 KByte buffer rather than performed once per arriving packet. Improves performance on Ethernet transfers to floppy disks by a factor of three. A new option ("spool") to the server turns off this buffering, so that the tftp server can be used as an interface to a print spooler.

### PC/whois

1. Order of opening connection, receiving data, and closing connection modified to avoid bug in BSD 4.2 finger server, which would forget about sending more than one packet if the connection is closed from the originator's end.
2. All bare ASCII LF characters are changed to LF/CR, because BSD 4.2 finger server doesn't send network standard ASCII. Since a bare LF is never legal in network standard ASCII, this change doesn't cause trouble with servers that do netascii right.

## B. Changes to protocol implementations

## TCP (affects telnet, nickname, and whois)

1. Fixed error in which TCP failed to react when rereceiving old data. This error had two effects: First, if an ACK was lost, the connection would hang, the other end would give up, and the next thing to be typed would cause a foreign reset. Second, it caused TCP to ignore UNIX "keepalives", which lead to a foreign reset if the connection isn't used for ten minutes. TCP now responds to "keepalives".
2. Fixed error in TCP close/reset sequence, eliminating occasional message "Bad TCP State" on exit.
3. Performance improvement of about 50%, accomplished by calling client with larger blocks.
4. Implemented TCP maxsegsize option, set to 511 bytes. This feature prevents the foreign system from sending packets larger than the PC can handle. It also has the side effect of preventing Berkeley 4.2 systems from sending packets with trailers to the PC.
5. Changed checksum calculation to accept either FFFF or 0000, to compensate for the ambiguous TCP specification as to which form of one's complement zero is expected. Some implementations use one, some the other. TCP now works with either kind of implementation.
6. Connection close sequence now acknowledges properly when the close is initiated by the PC.
7. A deadlock in the algorithm for window opening was eliminated.
8. Closing a connection when it is only partially opened no longer triggers an "unexpected state" error message.

## ICMP

1. No longer sends "destination unreachable" in response to IP broadcast packets. Helps avoid avalanching the network.
2. Bad echo sequence number messages are now displayed only when debugging switches are on.

## IP

1. Properly returns an error if the user tries to send a packet to an address not on the local network, but the gateway address hasn't been customized. (Previously, PC/IP erroneously sent the packet to address 0.0.0.0.)

## PC/IP User's Guide

### UDP

1. The name resolver has been updated to use new ARPANET standard domain name resolution protocol.
2. Debug tracing of incoming packets now includes packet size.

### C. Other changes

#### Debugging features

1. Separated debugging trace flags for the application, transport, internet, and driver levels. This change permits user to control the volume of debugging messages much more effectively. Any individual debugging flag can be turned on or off, either with **PC/custom** or dynamically when running **PC/telnet**.

#### Terminal emulator (affects **PC/term** and **PC/telnet**)

1. Special version created for use with IBM Professional Graphics Display, which mis-emulates the cursor motion of the Color Graphics Display. (used in **PC/pgatn** and **PC/pgaterm**)
2. Several bugs that produced incorrect line fill and background colors have been fixed. These were noted mostly when using the "vi" editor via **PC/telnet**.
3. Hercules color card (as well as some others that have more than 2K of display memory) now works.

#### Timer package

1. A misdeclared variable in the timer package led to an error every 65535 times the timer was used. The most noticeable symptom of this bug was that the line-25 clock in **PC/telnet** stopped ticking after 18 hours.

#### 3COM Etherlink Ethernet driver

1. Properly initializes ARP cache to all zeros. Previous initialize loop terminated early, left garbage, and occasionally caused improper hit.
2. Driver now saves and restores state of interrupt handlers and masks, so that it can be reused, for example, while calling a shell from **telnet**.

### D. New Packages

1. **PC/lpr** allows users to send files to be printed by a 4.2 Unix printer spooler.
2. **PC/monitor** is a new command that repeatedly tests a list of network services and keeps a display of the result.
3. New device driver for the Interlan NI5010 Ethernet card.
4. A domain name resolver was integrated with the old-style name resolver.

17 January 1986. This document is in file changes.mss



## 5. Changes In Prior Releases

This section describes changes between the January, 1985 and the February 1, 1984, releases of the PC/IP packages.

### WARNING

A major structural change in the installed device driver appeared in the January 1985 release. As a result, a single PC must run either all 1984 release programs or else all 1985 release programs. Users of the 1984 release who want to switch to the 1985 release must perform the installation and customization procedures just as if they had never before used these programs. Note, however, that if one PC runs the 1984 release and another PC runs the 1985 release on the same network they can communicate.

#### I. Changes that affect all packages

- 1) Name user upgraded to check responses to make sure they are for the current request rather than for an earlier one.
- 2) Improved error messages throughout system.
- 3) The sources of PCIP were modified to compile and assemble with the latest release of the microcomputer development C compiler, which now handles identifiers of longer than 8 characters correctly. In addition, a new C library is now in use. (Neither of these changes should cause any user-visible effects.)
- 4) An error in initialization that caused some programs to crash when run on machines with more than 512K of memory was fixed.
- 5) Class B and Class C Internet addresses now print properly.
- 6) All commands now return an error code to DOS as they terminate, so that the DOS ERRORLEVEL feature can be used.
- 7) An error in the NETDEV device driver that prevented operation under TopView was fixed.

## PC/IP User's Guide

8) The Ethernet device driver is now substantially more reliable, and it works with the PC/AT.

### II. Changes to protocol implementations

#### A) ICMP

1) Time-exceeded debugging messages now include information about the packet that got in trouble.

#### B) IP

1) A bug in interpretation of bit fields in the IP header that caused the "do-not-fragment" bit to act as the "this is a fragment" bit was fixed.

#### C) UDP

1) An error in length interpretation was fixed, which eliminates some bogus checksum errors.

#### D) TCP

1) TCP now provides an entry that allows an aborting command to reset a connection before exiting.

### III. Changes in specific packages

#### A) PC/telnet

1) The escape sequences F10-u and F10-U enable and disable the 25th line clock.

2) A new feature allows the user to specify that the **tftp** server of **telnet** should not ask the user for permission to do file transfers.

3) The escape sequence F10-A (which turned on tracing of TCP activity) is now invoked by F10-P.

4) An experimental feature allows the user to call a command interpreter while running **telnet**.

B) Terminal emulator (used in **PC/term** and **PC/telnet**)

1) Certain escape sequences are not emulated; the emulator simply discards them. Formerly, the emulator discarded only the escape sequence but not its following arguments. Now, the arguments are discarded also.

2) In certain scrolling situations, the wrong attribute byte was used at the end of newly scrolled lines. The bug affected only color displays, where text near the bottom of the screen was filled to the right with light blue background. Screen filling is now done correctly.

C) **PC/whois**

1) Replaced messages using the old command name **finger** with the name **whois**.

2) The user can now abort the command by typing "q".

D) **PC/tftp**

1) An error in **PC/tftp** sometimes caused outgoing packets to contain a header with a zero-length field. Although the resulting packet was, according to protocol, strictly legal, a bug in BSD4.2 UNIX caused UNIX to go into a loop in the kernel whenever it received such a packet. The error in **PC/tftp** was fixed.

2) An error in server **tftp** caused one packet buffer to be lost each time it was used to send a file from the PC. In addition, the error caused the wrong data packet to be resent when a timeout occurred, thereby assuring failure of that transfer. The error was fixed. (This error affects both **PC/tftp** and **PC/telnet**.)

3) A bug in calculating the length of error packets was fixed. Other hosts should no longer receive error packets with extraneous junk at the end.

## PC/IP User's Guide

### E) Ethernet driver

1) The 3COM Etherlink card for the PC locks up when it receives successive runt packets. It remains locked up until either the card is reset or the PC tries to send a packet. The effect is to disable any program (such as ping, tftp, or netwatch) that acts as a server. The Ethernet driver program was modified to watch for this condition and reset the Etherlink card if necessary.

2) An (apparently) hardware bug causes the 3COM Etherlink card to occasionally fail to respond to DMA requests on the PC/AT. The Ethernet driver program now loops in a busy wait to insure that DMA is completed, rather than depending on an interrupt. It now initiates DMA with a sequence that works on PC's with an expansion chassis.

3) Zero-length packets (a common occurrence) are no longer reported as protocol errors.

4) The software can now be configured to use any DMA channel and I/O base address that the 3COM Etherlink card can be configured to use. (But PC/custom does not yet allow setting DMA channel.)

5) Lost interrupts are now picked up by a timer. This addition improves reliability on Ethernets that have a large traffic load.

### F) PC/custom

1) Upgraded to allow flexible choice of I/O base address for Ethernet interface. Also allows setting of user name, office location, telephone number, and printer service address. Ability to set inverse video mode in display removed.

### G) PC/netwatch

1) A new "symbolic" format option displays IP, CHAOS, and Ethernet ARP interpretation of received packets, as an alternative to simple hexadecimal contents.

2) Packet buffer area reduced from 1000 to 512 undisplayed packets.

**H) PC/hostname**

- 1) Built-in table of name servers brought up to date.

**I) PC/ping**

- 1) Ethernet version now gives intelligible error messages when pinging a non-responding host on the same local net.

**H) New packages**

- 1) **PC/nickname:** A command to send requests to the ARPANET Network Information Center name server.

- 2) **PC/iprint:** A command to send files to an Imagen printer service.

18 January 1985. This document is in file oldchanges.mss



## 6. Software Installation

This section describes how to install the PC/IP commands and how to do initial customization.

The first step is to determine whether a serial line, an Ethernet, or a Pronet interface will be used for network attachment. One should obtain a diskette containing the proper versions of the set of PC/IP commands.

The distribution diskette is designed to be a read-only master copy, and it does not contain any parts of DOS. Thus you should start by copying the files you intend to use from the distribution diskette onto a formatted, DOS-containing working diskette or hard disk. You can then put the distribution diskette away in a safe place.

The next step is customization of the PCIP system for your environment. To do customization a few key facts about the environment must be collected for input to the customizer. If you are using an Ethernet or Pronet attachment:

1. Someone must assign an Internet address for this PC.
2. If you plan to communicate with hosts not directly attached to the same physical net, you must know the Internet address of a gateway that is attached to the Ethernet or Pronet.
3. If you are using an Ethernet and other hosts on your Ethernet do not use the proposed standard Ethernet-to-Internet address translation protocol, you must obtain a list of Ethernet-to-Internet address translations for the other hosts on your Ethernet.
4. Figure out which DMA channel and which interrupt vector will be used by your network interface. (See the hardware installation section for details.)

If you are using a serial line attachment, you do not need any of the above pieces of information. Instead, you must know the data rate of the serial line you plan to use.

That is the minimum repertoire of information needed for customization. In addition, you will probably want to make use of time, name, and printer services, so you should also obtain a list of names of up to five name servers and time servers, and one print

server. The names are all you need if your local net is linked to the ARPANET; you will be able to use the **PC/hostname** command to discover the Internet addresses of those services.

The next step is to customize the network device, a file named *netdev.sys* on the working diskette or hard disk, using the minimum set of facts collected above. See the writeup of command **PC/custom** for details on how to customize *netdev.sys*.

The customization of *netdev.sys* does not take effect until you install it as a DOS device driver. The reason is that *netdev.sys* is a file that describes a device driver rather than the device driver itself. Installation is automatic when DOS is bootloaded, that is either when the PC power is turned on or when control-alt-delete is typed. However, there is one detail: In order for the device driver to be installed automatically, the bootload diskette or disk must also contain a file named *config.sys* and that file must contain a line such as:

```
DEVICE=NETDEV.SYS
```

that names the file containing the device driver. If you already use a *config.sys* file you should make a copy of it and add this line, using a text editor. If you do not already have a *config.sys* file, you can use the one found on the distribution diskette. The DOS reference manual provides more information about the **DEVICE** command and about the file *config.sys*.

You should now have a *config.sys* file containing a "device" command that names *netdev.sys*, you should have customized *netdev.sys* with the minimum information, and after you bootload DOS (type control-alt-delete) you will be ready to try a **PC/IP** command.

Try, for example, **PC/ping**, specifying the IP address of some host that you should be able to address, to see what happens. If customization is not correct, some error message should appear that may give a clue as to what is wrong.

The next step is to use **PC/hostname** to obtain the Internet addresses of some time

and name servers, add them to the customization, and reboot to check them out. **PC/setclock** can be used to verify that both time service and name service customization are working: if **PC/setclock** succeeds when invoked with no argument, at least one time service address is correct; if it can obtain the time from a named time service, at least one name service address is correct.

8 April 1985. This document is in file soft-inst.mss



## 7. Hardware Installation

This section describes the installation procedures for network interface hardware supported by PC/IP, and also notes a problem found with some old memory cards.

Before you install your network interface, you need to select a DMA channel (note that the proNET card can use two separate DMA channels, one for transmitting packets and one for receiving them) for it and an interrupt vector. The board may need to be reconfigured to use your selections, and you also have to inform the software via the `PC/custom` command. It is important that the DMA channel and interrupt vector you select are not being used by any other hardware in the machine. Below is a chart showing what channels and vectors the network interfaces supported by PC/IP can use, and what channels and vectors are already in use in a PC, PC/XT and PC/AT.

Network Interfaces	Interrupts	Dma Channel
Pronet	2, 3, Or 4	1, 2, or 3
3COM Etherlink (old revs)	3 or 5	1 or 3
3COM Etherlink (newer)	2, 3, 4, 5, 6, or 7	1, 2, or 3
Interlan NI5010	3 or 5	1 or 3
PC Devices		
COM1 (same for AT)	4	none
COM2 (same for AT)	3	none
Printer 1 (same for AT)	7	none
Printer 2 (same for AT)	5	none
Floppy disk	6	3
PC/XT hard disk	5	2
AT Devices		
Floppy and hard disk	6	2

If you cannot, or choose not to, use a DMA channel, you should set the DMA channel (via `PC/custom`) to 0. The driver will then use a tight loop to transfer data from the card. We recommend using DMA, however.

If you install more than one network interface in your machine, or have some other unusual hardware, it is also important to make sure that the base I/O address of the interface does not conflict with any other hardware. Refer to the documentation for the appropriate interface for details on setting the base I/O address.

## 7.1. Pronet jumper selection

There are both switch-settable and jumper-selectable options on the proNet p1300 ring interface. See the p1300 "User's Guide" that came with the interface for more details.

### Switches

The node address switch (S39) must be set to an address different from every other station on your ring network, and that address is the same as the value in the last field of the Internet address that your machine has been assigned (see step one under software installation.) Note that the node address switch uses the "on" position to denote the binary value "0" and the "off" position to denote the binary value "1". The board/rom address switch (S22) should be set to all zeros (all "on".)

### Jumpers

The interrupt vector jumper and the two DMA jumpers (one for input, one for output) must agree with the configuration that the software will assume, and must not conflict with other installed I/O devices. The card can be set to use any of interrupt vector positions 2, 3, or 4, and any of DMA channels 1, 2, and 3. The PC/IP software requires that the two DMA jumpers be configured to use the same DMA channel. The card is usually shipped with the jumpers set for a configuration that will work with a standard PC or PC/XT, using interrupt vector 2 and DMA channel 1 for both input and output.

## 7.2. Interlan NI5010 jumper selection

All options on the Interlan NI5010 interface are jumper selectable. For more information, refer to the "NI5010 Installation and Programming Guide", supplied with the card by Mlcom-Interlan.

### Interrupts

The NI5010 card can be configured to use interrupt 3 or interrupt 5, controlled by jumper W7. The B position of the jumper selects interrupt 3; the A position selects interrupt 5.

### DMA

The NI5010 card can be configured to use DMA channel 1 or 3, or no DMA at all, depending on the positions of jumpers W8 and W14.

channel	W8	W14
none	remove	remove
1	A	B
3	B	A

### 7.3. 3COM Etherlink interface

Over the years, 3COM has produced a variety of different versions of the Etherlink Ethernet card for the IBM PC. All older cards, and in the current product line, models 500 and 501, work with PC/IP. There is one model, the 505 "smart card," that PC/IP does not support.

#### Jumper selection

There are several jumper-selectable hardware options on 3COM Etherlink Ethernet cards. Older cards, identifiable by the label "Rev. A" or "Rev. x.y" where x and y are integers, are usually green in color and have cryptic jumper labels. The newer cards have completely different labels for their option setting jumpers, and a few additional settings. The option sets shown below are known to work with the PC/IP Ethernet packages.

The choice of which DMA channel and which interrupt vector to use depends on what other I/O equipment is attached to the PC. For example, on an IBM PC/XT the hard disk, floppy disk, and printer are configured to use interrupt vector positions five, six, and seven, leaving two, three, and four for other attached devices. The Ethernet commonly uses interrupt vector position three.

3COM Etherlink card option settings:

New label	Old label	suggested jumper position
DMA REQ	jp1	channel 1 (must match software)
DMA ACK	jp2	Must match DMA REQ or jp1
Interrupt	jp3	vector 3 (must match software)
I/O address bit 9	n/a	right (1)
I/O address bit 8	jp4	right (1)
I/O address bit 7	jp5	left (0)
I/O address bit 6	jp6	left (0)
I/O address bit 5	jp7	left (0)
I/O address bit 4	jp8	left (0)
Memory address bit 19	n/a	right (1)
Memory address bit 18	jp9	right (1)
Memory address bit 17	jp10	right (1)
Memory address bit 16	n/a	left (0)
Memory address bit 15	n/a	right (1)
Memory address bit 14	jp11	right (1)
Memory address bit 13	jp12	left (0)
Memory address bit 12	jp13	left (0)
Memory enable	jp14	right (disable)
n/a	sw1	Left for onboard transceiver, right for external transceiver
U11/U10	n/a	Plug goes in socket U10 for onboard, socket U11 for external transceiver

Older Etherlink cards can be set only to interrupt vector positions three and five, and therefore must use position three in an XT. Similarly, the PC/XT uses DMA channel three for the hard disk and DMA channel two for its floppy disk, so the Ethernet must use DMA channel one. In recent shipments the 3COM Ethernet card has been configured at the factory with exactly these two settings.

**3COM Ethernet external transceiver incompatibility**

Certain combinations of 3COM Etherlink cards that are labeled "Revision B" with external transceivers generate improper Ethernet waveforms. These improper waveforms can be decoded without trouble by any 3COM interface, but some Ethernet interfaces from other manufacturers cannot decode these improper signals. This problem may be the cause when a PC can communicate with some, but not other, hosts on the same Ethernet. If switching the 3COM Etherlink card to use the internal transceiver clears up the difficulty, then the transceiver incompatibility problem is almost certainly the cause. Contact a 3COM sales representative for information on an engineering change that corrects the problem.

#### 7.4. Memory expansion card flaw

Some IBM 64K/256K memory expansion cards have a design flaw that causes trouble when an I/O device uses DMA channel 1. (The PC/IP software usually uses DMA channel 1 for the Ethernet or the Pronet.) The symptom of this design flaw when running PC/IP software is a catastrophic crash with the screen displaying the message PARITY CHECK 2. The crash usually occurs within the first hundred or so packets transmitted to or from the network.

If this problem appears, one should check the memory expansion card to see whether or not it has this design flaw, and whether or not it has been field-upgraded. Flawed cards have a connection between pins nine and ten of chip U49. (The connection is a very small printed circuit stripe on the underside of the card.) Repaired cards have had that connection severed, and two new wires added, from chip U33 pin eight to chip U33 pin eleven, and from chip U33 pin ten to chip U49 pin nine. Note that making changes such as these must be done with care to avoid damaging the card, and may void any warranties. If you have a flawed card it may be appropriate to inquire of your dealer what action should be taken. Alternatively, the network can be operated using DMA channel 3 if that channel is not already in use for some other device such as a hard disk, or can operate without DMA by selecting channel 0.

14 April 1986. This document is in file hard-inst.mss



## 8. Ethernet Etiquette

The Ethernet is a shared communication system, and certain actions can unintentionally disrupt it. This section describes practices and procedures that can minimize troubles seen by other users of the same Ethernet.

1. Most PC's are attached using "thin Ethernet," which means that the Ethernet wire comes down to the back of the PC where it connects to the PC with a T-connector. The continuity of the Ethernet (and of service to other users) depends on the integrity of the connection through the T-connector. If you disconnect your PC from the Ethernet, you should:
  - a. make certain that the continuity of the thin Ethernet through the T-connector is maintained.
  - b. Make certain that the T-connector is not touching anything metallic or conductive.
  - c. If the disconnection will be for more than a few minutes, replace the T-connector with a barrel connector. (Unterminated T-connectors provide an opportunity for noise, radiation, and echoes; one such opportunity won't necessarily bring down an Ethernet, but a large number of them can.)

If you are at one end of an Ethernet segment, you will find that one side of your T-connector has the Ethernet coming in, while the other side has a terminator attached. Continuity from the Ethernet to the terminator is just as important as continuity from one section of cable to the next, so if you disconnect your PC from the Ethernet, you should make certain that the Ethernet continues to be terminated, using either the T-connector or a barrel connector.

2. The Internet address used for your PC when it is attached to the Ethernet must be unique, and it must be manually assigned. Thus some care is needed to insure that two PC's don't accidentally try to operate using the same Internet address. Each PC/IP command has this Internet address embedded in it (as part of customization). You should not change the Internet address that your PC uses without coordinating the change with the central registry of addresses of other PC's. Also, if you exchange diskettes containing network programs with other PC/IP users be sure that you recustomize the internet address before using the programs.

23 October 1983. This document is in file ethernet.mss



## 9. Host Names and Internet Addresses

A brief description of the syntax of host names and internet addresses, and the method by which host names are resolved.

When PC/IP network programs accept a hostname argument it may be in either of two standard forms:

**Internet address** An internet address is four octal integers separated by commas, for example:

22,11,0,127

or four decimal integers separated by periods, for example:

18.9.0.87

Each integer represents one byte of a 32-bit standard internet address, in the order "network," "subnet," etc. When the user supplies an internet address the PC/IP network program uses it as is, depending on nothing else for name resolution.

**Host name**

When a PC/IP network program encounters a string that does not appear to be an internet address, it interprets the string as a host name and it attempts to resolve the name by appeal to one or more name servers via the network. The program sends inquiries to as many as three domain name servers and two old-style name servers whose internet addresses are embedded in the *netdev.sys* file. (The user may change the number of name servers and their internet addresses by use of the PC/IP program **custom**.)

If a text name is given, first up to three domain name servers are polled in succession. If the name is a fully qualified domain name then it is passed intact to the domain name servers; otherwise the domain specified with PC/**custom** is appended to the end of the name. If none of the domain name servers can resolve the name (or if no domain server addresses are specified) the old-style name resolver is used.

The old-style name resolve can produce three outcomes, since name servers may reply with an internet address, reply with a "host name unknown" response, or may not reply at all. To increase availability several name servers are polled, and the following rules merge the resulting replies:

1. If one or more name servers respond with an internet address, the program uses the first such response received and ignores all later responses.

## PC/IP User's Guide

2. If one or more name servers respond, but all the responses are "host name unknown," the program displays that error message and exits.
3. If no response arrives from any name server within five seconds, the program gives up, displays the error message "name servers not responding," and exits.

Note that if different name servers give different responses to the same inquiry, the user may see erratic results depending on which name servers are up and which respond more quickly. If one suspects that name servers are not responding or are not in agreement, the commands *netname* and *onetname* may help isolate the trouble.

17 January 1986. This document is in file *nameres.mss*

## 10. Debugging options

This section explains the operation and usefulness of the debugging option switches that can be set using the customizer.

The PC/IP packages have built in as part of their design a large number of error and progress report messages, but these messages do not appear on the display screen unless specifically requested. The debugging option switches control which messages the packages display. When troubles appear in the use of network programs, it is often not immediately apparent whether the cause is a problem in the local computer system, in some distant server, or in some network in between. The tracing that is controlled by the debugging switches has as its primary value that it can allow fairly rapid trouble isolation in such circumstances.

The arrangement of the debugging option switches in the PC/IP packages has evolved as the requirements for tracing have become better understood; this evolution is incomplete and there are quite a number of cases where different packages and different levels of network protocol do not yet follow consistent conventions.

The debugging switches can be set ON or OFF as customization options. The usual technique is to customize the debugging options to the ON position in the *netcust:* device so that they apply only to the current session. However, as is described below some users may find it helpful or interesting to customize the first few of the switches permanently ON (in the file *netdev.sys*) to allow monitoring of network status and problems. Each debugging option switch is described here and in the customizer by a symbolic name.

Here are the message categories controlled by each debugging switch:

- |                |  |
|----------------|--|
| <b>NETERR</b>  | Reports all recoverable errors detected by the local network (Ethernet, proNET, or serial line) driver. Can be left ON during normal operation to monitor appearance of network troubles.                |
| <b>PROTERR</b> | Reports all packets received that seem to be inappropriate for the protocol being used, or that represent some other trouble at the protocol level. Primarily useful for debugging other implementations |

or discovering incompatibilities between implementations on different computer systems. Can be left ON during normal operation to serve as a warning that one has contacted a host that isn't following protocol in the expected way.

**TIMEOUT** Reports all timeouts waiting for the other end of a connection to respond. Can be left ON during normal operation to monitor frequency of timeout-triggered retries.

**APTRACE** Provides a trace of the activities of the application level protocol. For example, in PC/tftp, APTRACE produces a one-line message for each file block that is sent or received. Can be left on during normal operation if progress reports are important or useful, but tends to fill the screen with tracing messages.

The following debugging options are primarily useful for finding problems in the interactions between the PC network protocol implementation and those of other machines. They generally produce so much output that they are best left off unless they are really needed.

**TCTRACE** Provides a trace of the activities of the transport level protocol, such as UDP or TCP. Produces a one-line message for each packet that is sent or received at the transport level.

**INTRACE** Provides a trace of the activities of the Internet protocol level, IP or ICMP. Produces a one-line message for each packet that is sent or received by the Internet level.

**NETRACE** Provides a trace of the activities of the local network driver. Produces a one-line message for each packet that is sent or received on the local network.

**DUMP** Whenever an incoming packet seems to have something wrong with it, this switch causes its contents to be displayed in hexadecimal format. In conjunction with NETRACE, INTRACE, or TCTRACE, will produce a symbolic dump at the appropriate level. (Note that the time required to display a complete packet contents may exceed the timeout/retransmit time of some hosts, so setting this switch ON can significantly alter the sequence of packets received and sent.)

**INFOMSG** Triggers a long list of informational and progress report messages. Used primarily to find out how far a PC/IP package got before it crashed.

**BUGHLT** Displays a message whenever the network level code of PC/IP detects

a gross application error of some kind. (Not actually used very much.)

The **PC/telnet** command has a special tracing feature that is useful for tracking interactions with a remote time-sharing host. The **PC/telnet** escape F10/control-A toggles the **APTRACE** debugging switch described above. When **APTRACE** is ON, **PC/telnet** displays on line 25 a cryptic progress report (updated once per second) on the connection to the other host. This report appears as follows:

```
Sent: N1(N2)N3  Rcvd:  N4(N5)N6  Window: N7
```

with the following interpretation:

- N1            Number of bytes sent by the PC to the other host.
- N2            Number of sent bytes not yet acknowledged by the other host.
- N3            Number of packets resent to the other host in hope of eliciting an acknowledgement.
- N4            Number of bytes received from the other host.
- N5            Number of received bytes not yet acknowledged to the other host.
- N6            Number of packets rereceived (that is, duplicates) from the other host.
- N7            Number of bytes that PC/IP has authorized the other host to send. (TCP window size.)

Note that while ON, **APTRACE** also triggers a one-line-per-block message from the tftp server if it used from within **PC/telnet**.

**PC/telnet** can be asked to toggle any debugging switch at any time, using F10 followed by the appropriate control- character. Several other debugging and maintenance toggles and displays are also available. F10/control-H displays a list of possibilities.

16 September 1985. This document is in file debugging.mss.



## 11. Dialup line file transfer

One use of the PC/IP commands is to transfer files to and from some network-attached system over a dial-in line. Two scenarios for that use are described here. For either scenario, the description of commands **PC/onhook** and **PC/tftp** should be read before proceeding. These scenarios require that the serial line version of **PC/tftp** be used.

- Scenario with manually-controlled dialling:

1. Type the command **offhook**.
2. Dial the telephone number of the PC concentrator. When it answers, switch the modem to data.
3. (Optional, but recommended) Type the command **setclock** to verify that the network connection is operational and also to set the PC clock so that the date and time attached to newly transferred files will be correct.
4. Issue the **tftp** command to get or put the file wanted.
5. Repeat the previous step until all files are transferred.
6. Type the command **onhook**.
7. Switch the modem to voice and disconnect it from the telephone line.

- Scenario for use with a terminal-controlled dialling modem:

1. Type the command **term**.
2. Using the terminal emulator, instruct the modem to dial as desired.
3. Continue with step three, above.

27 November 1983. This document is in file dialup.mss



## 12. Custom

**PC/custom**, version 2.2

A command to customize the PC/IP environment, allowing setting of parameters that describe the network environment and preferred option settings.

Usage:

**custom** netdev [model]

begins customization of the device description found in the file named *netdev.sys*. When finished customizing, custom rewrites the file *netdev.sys* with the new parameters in place. The customizer is menu-driven and self-explanatory. If a second argument is given, custom reads the values of the customization parameters of the command found in the file *model.sys* and uses them as initial values. It then enters the usual starting menu so that the user can review the result.

For simplicity and uniformity, the one device driver contains the customization parameters for all network levels and all commands. For example, one can set serial line parameters even though the PC/IP commands to be used contain an Ethernet driver. It is not necessary to specify values for customization parameters that will not be used. For example, if the command PC/**setclock** will not be used, one need not specify the Internet addresses of time servers.

Note that customizing the file *netdev.sys* will have no effect until the next time DOS is bootloaded. See the writeup entitled "software installation" for more details.

### 12.1. Standard customization parameters

There are several customization parameters that are applicable to all or several different PCIP commands. Customization parameters that apply to just one command are described in the writeup of that command.

Site customization to match network hardware options, switch settings, and parameters:

## PC/IP User's Guide

- Serial line speed. Can be set to any baud rate from 110 to 19,200. (Needs to be set only for serial line use.)
- Interrupt vector for network interface. Should be set to correspond to the interrupt vector number that the hardware interface will use. The PC/IP serial line driver ignores this entry.
- Receive and transmit DMA channels for network interface. Should be set to correspond to the DMA channel that the hardware interface will use. Most hardware only supports a single DMA channel for both receive and transmit; if that is the case, both should be set to the same value. The proNET interface *does* support separate channels. The DMA channel can also be set to 0 if no DMA should be used; a tight loop will be used instead to copy data to or from the interface. (The serial line driver does not use DMA at all). Under some circumstances, using a copy loop can be faster than using DMA (for instance, on the PC/AT).
- Network interface I/O address. Should be set to match the I/O base address used by the network hardware. Default is 0300 (Hex).
- Ethernet address. One can set the Ethernet address in one of three ways:
  1. to the Ethernet address found on the network interface card,
  2. to an Ethernet address derived from the Internet address by concatenating 16 leading zero bits,
  3. to an arbitrary Ethernet address specified to the customizer.

One should normally use the first option; the others are available to deal with non-standard Ethernet environments.

- Number of network interfaces. This parameter is currently not used; it is provided for future implementation of multi-network attachment.

### 12.2. Site customization of network level parameters

- Internet address of this computer. (Not needed for serial line use.)
- Internet address of default IP gateway. (Not needed for serial line use.)
- Internet addresses of up to two IP non-domain name servers. These are old-style, IEN-116 name servers.
- Internet addresses of up to three IP domain name servers.
- The text name of the machine's domain. This name is used by the domain

name resolver. When asked to resolve a name that is not fully qualified (no domain is specified), the resolver will append this name to given name.

- Internet addresses of up to five time servers. The servers are polled at two second intervals in the order they were set by the customizer, so one may place preferred services nearer the head of the list. (Needed only by PC/setclock.)
- Internet address of a print server. (Needed only by PC/iprint and PC/lpr.)
- Internet address of an IP log server. If this address is non-zero, some PC/IP commands send error-logging or statistics-gathering packets to this address. For privacy, the address may be set to 0,0,0,0, in which case all logging is suppressed.
- Up to three initial values for Ethernet-to-Internet address cache. (Needs to be set only for Ethernet use and when the environment does not provide the proper protocol.) IP addresses are entered in standard octal or decimal form. Ethernet addresses are entered as 6 octal byte values (each between 0 and 377) separated by commas.
- TCP window size and low window level. These two parameters affect the performance and smoothness of flow of data in commands such as Telnet. The window size is the count, in bytes, of the maximum amount of data that another host should send to the PC without waiting for authorization to send more. If not customized, its default value is 450 bytes. One might make this value smaller if there is a gateway with limited buffering ability in the pathway between the PC and a commonly-used host, or if talking to a host on the same local area network. The low window level is the trigger point at which the PC sends authorization to send more data to the other host. If not customized, its default value is 200 bytes. If there is a long round-trip delay to a commonly-used host, one might adjust this value so as to just fill up the pipeline from that host. The low window level must be less than the window size, and the window size must be less than 2000 bytes.
- Telnet transmission trigger. Can be set to send every character as it is typed (necessary if using a character-based remote echo system) or to send a batch of typed characters only when a newline character is typed (less demanding on the remote system.)
- First RVD drive. This parameter is provided for a future feature.
- Number of subnet bits. This parameter determines how many bits, following the network part of an IP address, are used to identify the attached subnetwork. PC/custom displays an octal "subnet mask" that is derived from the IP address and the number of subnet bits. This feature is used in a simple way, as follows: when an IP packet is to be sent from the PC, its IP destination address is masked with the subnet mask. The part of the

destination address that is revealed by the subnet mask is then compared with the corresponding part of the PC's own IP address. If the revealed sections of the two addresses are different, the destination is assumed to be on another subnetwork, and the routing layer sends the packet to the default gateway. If the revealed sections are the same, the destination is assumed to be on the same local area network as the PC itself, and the concealed portion of the destination address is used by the network layer to construct the proper physical address (perhaps using an Address Resolution Protocol). If subnetwork routing is not in use, an extent of zero is appropriate. At M.I.T. the subnetwork routing scheme uses 8 bits for subnetwork identification.

### 12.3. Personal customization of terminal emulation options

- Action on lines too long to fit on screen. Discard mode places all excess characters in column 80. Wraparound mode places excess characters on next line.
- Swap interpretation of backspace with control-backspace. (See telnet description of emulation conventions for further information.)

### 12.4. Other parameters

- User's name. This is a character string that is included in error logging messages and is placed in headers of files sent to a print server. May be set to (or left) blank.
- Internet address output radix. This parameter controls whether numeric Internet addresses are printed in decimal (for instance, 18.10.0.65) or octal (22,32,0,101). Numeric addresses can always be input in either radix.
- Debug options. There are several options that turn on various degrees of progress reports, tracing, and otherwise suppressed error messages. These options are of interest primarily to system programmers. One normally sets them to "all off". (See the section *Debugging Options* for more details.)
- Local standard time offset, in minutes before GMT. West of GMT the value is positive, east of GMT the value is negative. For EST the value is +300. For SET the value is -60 in the winter.
- Local standard time designation string. Three letters, such as EST, EDT, or SET. (Note that if the middle letter of the time zone designation string is "s" or "S" PC/`setclock` will automatically do daylight-savings-time conversion.)

## 12.5. On-the-fly error correction

Errors in typing names and addresses can be corrected with the following common editing conventions: The backspace key discards the last character typed, while Control-U discards the entire name or address typed so far, allowing one to start over.

## 12.6. Temporary customization

The command

```
custom netcust
```

will recustomize the currently active device driver, which is named *netcust*. Customization of *netcust* takes effect immediately, rather than at next bootload, and is lost when the next bootload takes place. Temporary customization of the active device driver is sometimes useful in debugging, for example to turn on a tracing option for a while. Note that for temporary customization to work there must be an already-present active device driver, previously loaded by DOS.

17 January 1986. This document is in file custom.mss



## 13. Iprint

PC/iprint, version 1.1

A program to send a text file to an Imagen print server.

Usage

**iprint** filename  
or  
**iprint -n** filename  
or  
**iprint -q** filename

sends the file *filename* to the default print server, using the standard protocol for the Imagen family of print servers. **PC/iprint** specifies a default format that simulates an 80-character, 10-pitch line printer. It arranges for a header line containing the name of the file and the current date and time to appear at the top of each page, and it attaches a cover sheet containing the user's name. If the option "-n" is given, the header line is omitted. If the option "-q" is given, the **iprint** command displays no messages unless it encounters an error.

### 13.1. Customization

The following parameters of **iprint** can be customized with the **PC/custom** command:

1. Internet address of the print server. Note that **lpr** also uses the print server address when connecting to a Unix printer spooler. The print server address cannot be customized separately for **iprint** and **lpr**.
2. Name of the user, for the cover sheet.

### 13.2. Notes

All PCIP packages follow the DOS convention of returning an **ERRORLEVEL** value when they exit. For use in batch files, **PC/iprint** returns **ERRORLEVEL=0** if the printer service accepts the file, and **ERRORLEVEL=1** otherwise.

**PC/IP User's Guide**

**23 January 1986. This document is in file lprint.mss**

## 14. Lpr

PC/lpr, version 3.0

A program to send a text or graphics file to a local or remote printer. Emulates the Unix lpr command to a limited extent.

### Usage

```
lpr [ -Pprinter ] [ -Sserver ] [-pgvqw] filename
```

causes the file *filename* to be spooled to a printer. The -P option may be used to force output to a specific printer, and the -S option may be used to specify a print server.

If the option "-P" is given, the word following the "-P" is taken as the name of the printer to be used. If the "-P" option is missing, the name of the printer to be used is taken from the PRT environment variable. If this variable has not been set, the name of the printer is taken to be *lp*. (Note that the option may be written "-P printer" OR "-Pprinter", as in the UNIX convention for the **lpr** command.)

The name of the printer, whether taken from the "-P" option, the environment variable, or defaulted, is significant. The following names have special meanings:

**local**                    If the printer name is "local", the file is printed with the DOS "PRINT" command. There must be a printer attached to the PC issuing the command, and the normal messages issued by "PRINT" will be displayed and should be responded to.

**prn,lpt1,lpt2,com1,com2**

If the printer name is any of these names, which have special meaning to DOS, PC/lpr assumes that the print server is another PC on the network that is running a PC/tftp print server and possibly a spooler. PC/lpr will send the file to that PC by means of the tftp protocol, using the given printer name as the target device. If the server PC has a printer attached under that device name, it will print the file there.

**all other names**        If the printer name is any other value, PC/lpr assumes it to be the name of a UNIX printer controlled by a UNIX/lpd daemon, running

the 4.2bsd printer protocol. In these cases, **PC/lpr** creates a cover sheet with the user's name (taken from the **USER** environment variable), the name of the file, and the office number (taken from **PC/IP** customization information). It then sends the file to the **UNIX/lpd** daemon for printing.

If the option **"-S"** is given, the word following the **"-S"** is taken as the Internet name or address of the print server to be used. If the **"-S"** option is missing, the identity of the print server to be used is taken from an environment variable named **"SERVER"** or if there is no environment variable, from the customization parameter **"default print server"**. (Note that the option may be written **"-S server"** OR **"-Sserver"**.)

If the option **"-q"** is given, **PC/lpr** displays no messages unless it encounters an error.

If the option **"-w"** is given, and the file was sent to a **UNIX** print server, **PC/lpr** will wait until the server closes the network connection before exiting. If the printer is directly attached to the server, this is a way to wait until the file has started printing.

The following single letter options are used to specify that a filter is to be used on the file before printing it. In all cases, a temporary file will be created, modified by the filter, spooled to the printer, and then erased:

- p**            **pr** is used to format the file. If the print server is running **Unix**, it will process the file with the normal **Unix pr** command. If the print server is running **DOS**, **PC/lpr** will run the file through a **DOS pr** filter before printing. In either case, the file is printed in pages with a five-line margin at the top and bottom, and a header line consisting of the date, time, name of file, and page number, on the third line of each page.
- g**            **prntmeta** is used to format the file. This is a **DOS-only** graphics filter which translates **GKS** meta files for printing on the **IBM Graphics Printer**.
- v**            this option is reserved for future use for files in printer-specific formats. There is currently no support for this option.

### 14.1. Customization

The following parameters of `PC/lpr` can be customized with the `PC/custom` command:

1. Internet address of a remote print server.
2. Name of the user, for the cover sheet.

The following parameters of `PC/lpr` can be customized by setting DOS environment variables:

1. printer name, set with "set prt=name"
2. user name, set with "set user=name"
3. server name, set with "set server=name"

### 14.2. Notes

All `PC/IP` packages follow the DOS convention of returning an `ERRORLEVEL` value when they exit. For use in batch files, `PC/lpr` returns `ERRORLEVEL=0` if the printer service accepts the file, and `ERRORLEVEL=1` otherwise.

28 February 1986. This document is in file `lpr.mss`



## 15. Monitor

**PC/monitor**, version 1.4

A program that monitors availability of network services, keeping a display that shows which are currently responding and which are not.

Usage:

**monitor** *filename*

**PC/monitor** reads the control file *filename* to determine the list of services to be monitored. It then tests each service in the list. Following each such test it displays the name of the host of that service in a form that indicates the outcome of the test. After completing a round of tests, **PC/monitor** waits for 60 seconds, then performs another round of tests. An asterisk on the display indicates which service is currently being tested.

Whenever a service responds normally, **PC/monitor** displays the host's name using normal display mode. If a service that responded on the previous test fails to respond, **PC/monitor** displays the host's name in intensified mode. If two or more successive tests of a service fail, **PC/monitor** changes the display of that host's name to blinking intensified mode, and sounds an audible alarm once. The user can acknowledge having seen such a warning by hitting the space bar, which causes **PC/monitor** to change currently blinking names to normal intensity on the next round of tests.

If the service responds but the response is incorrect, its host's name is underlined (or on a color monitor, in blue).

**PC/monitor** switches off all debugging switches just before it starts to display test results. If it notices some error while trying to invoke a service, it displays the host's name in inverse video.

To stop the tests and exit from **PC/monitor**, type "q". To start another round of tests without waiting for completion of the 60-second timeout, type "g".

PC/monitor can test the following kinds of services:

1. UDP time service. PC/monitor sends a standard time service request and watches for a time response from that server. It does not check the value of the result.
2. UDP domain name service. PC/monitor sends a domain name service request for a name specified in the input file. It checks the response to verify that it is the one expected.
3. UDP name service (IEN-118). PC/monitor sends an old-style name service request for a name specified in the input file. It does not check value of the response. (N.B. Both name service and domain name service test results appear in the same column of the display.)
4. ICMP echo service. PC/monitor sends a standard echo request containing 20 bytes of random data, and watches for an echo response containing those 20 bytes of random data.
5. RVD-control service. PC/monitor sends a shutdown control request with the password "x" (in anticipation that "x" is not the maintenance password) and watches for a response from that server, but does not check that response for correctness.

## 15.1. Control file

The format of the control file is as follows:

1. The file is ASCII, so it may be prepared with an ordinary text editor.
2. White space (blanks, tabs, or new-lines) separates control inputs in the control file. A control input consists of a control identifier followed by an equal sign, followed by control parameters separated by semicolons. (Recommendation: put one token on a line, so the result is easy to read and modify.)
3. Following is an example of a control input describing a service to be tested:

```
service=echo;multics;10.0.0.6
```

The first parameter, "echo" in that example, could be replaced by "domain", "name", "rvd", "time", "time1", or "time2". (The use of "time1" and "time2" is explained in point 4, below.) The second parameter is the name to be displayed of the host that runs the service to be tested. This name must be eleven or fewer characters in length. The third parameter, containing the internet address of that host, is optional. If absent, PC/monitor uses the customized name services to resolve the displayed name. If present, it can be in either octal form (with commas) or decimal form (with decimal points).

4. The display limits the number of services of any one type to 20; the service types "time", "time1", and "time2" place the result of a time test in three different columns of the display, and thus increase the limit on the number of time services to 60.)
5. To comment out a token, insert the letter # as the first character.
6. The time between passes through the service test is normally 60 seconds. This time can be changed by a control line of the form
 

```
pause=15
```

 where the number of seconds to pause is an integer less than 65535.
7. Name service tests are performed by sending a request for the name provided in a control line of the form
 

```
nametest=multics.mit.edu;10.0.0.6
```

 Where the name must be fewer than 30 characters in length. (But for a domain name test, it must be a complete domain name.) For checking the correctness of a domain name server, the corresponding internet address may be given in either decimal form (with periods) or octal form (with commas).
 

If no nametest control line is provided, PC/monitor uses the default name "athena.athena.mit.edu" and looks for the response 18.58.0.1.
8. After processing the input file, PC/monitor pauses for five seconds, to permit review of any non-fatal warning messages that occurred during that processing.

## 15.2. User commands

Summary of user requests accepted by PC/monitor:

q	("quit") Exit to DOS
g	("go") Start another round of tests.
c	("clear") Redisplay the screen contents, in case they have been messed up by an error message.
space	("acknowledge") Change all current blinking intense fields to blinking normal.

### 15.3. Display modes

Summary of display modes and their meanings:

normal	Latest test of this service was successful.
intense	Latest test of this service failed; previous one was successful.
intense blinking	Two or more tests in succession failed.
normal blinking	space bar hit since two or more failed.
underlined (blue)	service responded, but with wrong answer.
inverse video	trouble encountered in trying to do this test.

### 15.4. Bugs

1. RVD service availability should be tested by sending server-status-request packets, not shutdown requests.
2. If more than 20 services of one type appear in the control file, PC/monitor muddles the display rather than reporting an error.
3. After several hours of operation, catastrophic errors begin to appear, first muddling the display, and then crashing the monitor.

### 15.5. Customization

The following parameters of PC/monitor can be customized with the custom command:

1. Internet addresses of up to five name servers. The name servers are used to resolve those names found in the control file that are not accompanied with internet addresses.

23 January 1986. This document is in file monitor.mss

## 16. Netname

PC/netname, version 1.0

A package to look up the Internet address that corresponds to a character string name, using the UDP domain name protocol.

Usage:

```
netname [-a] [-t timeout] name [domain-name-server]
```

Where *name* is the character string host name to be resolved. If no domain name server is specified, **netname** sends an inquiry to each customized domain name server, and displays all responses. The optional argument **-a** causes netname to display application trace information that may help in discovering obscure problems in name server tables. The optional argument **-t** causes the next argument to be used as the timeout, in seconds, before giving up on the name server. The default timeout is 20 seconds.

If *name* contains at least one period character, **netname** assumes it to be a complete domain name and sends it unchanged to the domain server. If *name* contains no period characters, **!b[netname]** appends to it the customized default domain name for this PC.

There are three possible results of a name inquiry:

1. An Internet address.
2. The response "name not known"
3. No response.

Each network node has a primary name, and may also have any number of secondary names. **PC/netname** accepts inquiries for both primary and secondary names. If the name requested is a secondary one, **netname** reports the primary name as part of the response.

In the second form, above, *domain-name-server* is an optional argument that identifies

a specific domain name server that is to be invoked to resolve the name *name*. *domain-name-server* may be either a character string name (in which case it is resolved using the customized name servers) or an Internet address in standard form.

The section on host names, domain names, and Internet addresses provides more information on the resolution of host names. This command is useful primarily for trouble isolation when one suspects that name tables may be inconsistent or incorrect.

### 16.1. Customization

- Default domain name. Appended to any name that contains no period characters.
- List of domain name servers.
- Application trace. If the APTRACE debugging flag is on, **PC/netname** displays details of the name server response.

22 February 1986. This document is in file netname.mss

## 17. Netwatch

### PC/netwatch, version 7.0

A program to monitor the attached local network. It is useful primarily for debugging network operations on a broadcast network.

Usage:

#### **netwatch**

No arguments are required. **PC/netwatch** listens to the attached local broadcast network and displays one line of information for every packet that goes by. This information consists of the "to" and "from" local network addresses, the packet length, the value of the protocol type field, and 8 selected contiguous bytes of the packet contents. While **netwatch** is running one may type commands to it. The commands either display collected information, change **netwatch**'s operating mode, or tell it to filter for specific types of packets. The commands are:

- a Match all packets. Turns off all packet filtering.
- c Display packet type counts. Prints a list of all packet types that are built in to **netwatch** and how many of each type it has accepted and displayed. Some counts are misleading because some protocols have two type indicators in their headers (TCP and UDP packets have two socket numbers).
- d Match on destination. Prompts the user to input a destination address and only accepts packets going to that address. See the section below on filtering for more information.
- h Display packet length histogram. Displays a list of packet lengths in 64 byte increments and a count of how many packets of each length have been accepted and displayed by **netwatch**.
- l Clear screen.
- m Toggle using manufacturer info in hardware addresses. This command is only useful on Ethernet **netwatches**. The first three bytes of an Ethernet address can be used to determine the manufacturer of the Ethernet card the address is associated with. This command toggles

whether or not **netwatch** prints the first three bytes as hexadecimal numbers or symbolically as the name of the manufacturer.

- n Toggle normal and symbolic modes. Switches between the mode where **netwatch** simply dumps packets in hex and the mode where **netwatch** unparses packet headers and displays a symbolic representation of the contents of the packet.
- p Pause. Waits for the user to type something before proceeding.
- q Quit. Return to PC-DOS.
- r Reset packet count. Resets the count of accepted packets to 0.
- s Match on source. Prompts the user to input a source address and only accepts packets coming from that address. See the section below on filtering for more information.
- t Match on packet type. Prompts the user to input a packet type specification (see below on filtering) and only accepts packets of that type.
- u Match only on unknown packets. **Netwatch** will only accept and display packets of types it does not know. This feature interacts incorrectly with some of the internal counters in **netwatch**; some packet counters will still increment on packets that do not get displayed.
- w Match all packets coming to or from an address. Prompts the user to input an address and accepts and displays only packets coming from or going to that address. See below on filtering.
- A Shows more application layer information when displaying packets.
- H Display histogram of packet lengths. This command shows a bar graph of packet lengths and counts.
- I Shows more internetwork layer information when displaying packets.
- L Toggle displaying local net addresses. When displaying local net address, **netwatch** will display hardware source and destination addresses even when in symbolic mode.
- N Shows more network layer information when displaying packets.

- S            Print statistics.
- T            Shows more transport layer information when displaying packets.
- ?            Print command summary.

## 17.1. Filtering

**Netwatch** allows the user to specify filters for packets. Only packets matching those filters will be accepted and displayed. **Netwatch**'s internal counters (length and counts of different packet types) only count accepted packets.

The simplest filter is on packet type. Packet types can be symbolic or numeric. For instance, on an Ethernet, one can match on packet type "ip" or packet type "800". Higher level protocols can be matched on as well; you can match on "ip tcp" or "ip tcp telnet". A '?' anywhere in the type specification will cause **netwatch** to respond with a list of acceptable types.

**Netwatch** can also filter on addresses. There are three types of address matching: source, destination and watching (source or destination). Both hardware and protocol addresses can be specified. On a proNET ring, you might type "99" as a hardware address. On any hardware, you could specify "ip 18.26.0.65" as an Internet protocol address or "chaos 15101" as a Chaosnet protocol address. You can specify the hardware broadcast address as "\*" or the numeric address.

## 17.2. Combining Filters

Filters can be combined: you can look for all "ip tcp telnet" packets coming from host "ip 18.26.0.65", or you can combine hardware addresses and protocol addresses. There are a few catches to be wary of.

None of the filters clear the old filters when you start. Therefore you should normally use the 'a' command to accept all packets before you change the type or address **netwatch** is matching on. Some combinations make no sense, for instance, watching for packets coming to or from "ip 18.26.0.65" and then watching for packets coming to or from "chaos 15101". In this case, **netwatch** will probably never see *any* packets.

because it is looking for packets with the correct bytes in right positions for *both* of those addresses.

Also, watching on both hardware and protocol addresses might miss some combinations of both.

### **17.3. Performance**

The limitations of the monitor in high-traffic situations should be understood. The monitor can handle a burst rate of about 200 packets per second. Packets arriving faster than that are missed (but counted in the statistics of the network driver). In addition, the display rate is about 25 packets per second and there is a buffer that can hold 512 undisplayed packets. If packets arrive faster than the display rate for a long enough time to fill up the buffer, the monitor discards overflow packets.

Note: When the proNET version of **netwatch** is used, a jumper must be set in the hardware to permit the interface to accept all packets.

2 October 1985. This document is in file netwatch.mss

## 18. Nicname

PC/nickname, version 2.0

A program to invoke the ARPANET Network Information Center name directory service.

Usage:

**nickname** name

sends a request to the ARPANET Network Information Center (assumed to be located at IP address 12,0,0,63) name resolution service, inquiring about "name". The NIC normally responds with a text string, which **nickname** then displays on the screen of the PC.

If name is "-help" **nickname** displays some hints on how to use it. If name is "help", **nickname** forwards the request to the NIC, which responds to that name with a screenful of information on making more sophisticated inquiries.

If the Network Information Center is not forthcoming with a response, PC/**nickname** will give up after about 20 seconds. Typing the letter "q" will cause PC/**nickname** to abort the operation immediately. This feature is useful if one discovers that the output from the NIC is more extensive than anticipated.

4 December 1984. This document is in file nickname.mss



## 19. Onetname

PC/**onetname**, version 5.0

A package to look up the Internet address that corresponds to a host's character string name, using the (now obsolete) IEN-116 UDP/ICMP name server protocol. (See the command PC/netname for name lookup using the newer domain name service.)

Usage:

```
onetname name
or
onetname name nameserver
```

Where *name* is the character string host name to be resolved. If no nameserver is specified, **onetname** sends an inquiry to every known name server, and displays all responses.

*nameserver* is an optional argument that, if provided, identifies a specific name server that is to be invoked to resolve the name *name*. *nameserver* may be either a character string name (in which case it is resolved using the customized name servers) or an Internet address in standard form.

The section on host names and Internet addresses provides more information on the resolution of host names. This command is primarily useful for trouble isolation when one suspects that name tables may be inconsistent or incorrect.

### 19.1. Customization

PC/**onetname** has no special customization parameters of its own. The names and Internet addresses of several ARPANET IEN-116 name servers are built in to PC/**onetname**, and are changeable only by recompiling or patching the program. Name server addresses provided by customization are used only for resolution of name server names. See the description of custom for explanation of these parameters.

9 January 1985. This document is in file onetname.mss



## 20. Onhook and Offhook

### PC/offhook PC/onhook

Programs to connect or disconnect (in telephone jargon, "place off-hook or on-hook" and in common parlance, "pick up or hang up") the telephone line attached to a serial port on the PC.

Usage:

**offhook**  
**onhook**

These commands are provided for the scenario in which several different PC/IP commands are to be used in a single session, via a dial-up serial line. The **offhook** command instructs the attached modem that the computer is prepared to use the serial line (by turning on the signal "data terminal ready" in the modem interface.) By convention, PC/IP commands that use the serial line always restore the on-hook/off-hook status of that serial line as they exit. At the completion of the session, the user may issue an **onhook** command to disconnect the telephone line.

Note that the terminal emulator command, PC/**term**, is also useful in management of on-hook and off-hook status, especially in the case where either the modem or the computer at the other end of the serial line is controlled by sending ASCII characters. Rather than starting a session with the **offhook** command one starts with **term**, using the emulator to tell the modem and switches how to connect things up. Then, the user exits **term** with F10/q, which leaves the serial port in off-hook status. At the completion of the session, the user issues the **onhook** command as usual.

The commands PC/**onhook** and PC/**offhook** operate on serial port number one, known in PC documentation as "COM1:".

23 October 1983. This document is in file onhook.mss



## 21. Ping

PC/ping, version 5.0

A program to send an echo request to another host and watch for a response, using the ICMP/IP protocol. It is used primarily to isolate trouble in an internetwork environment.

Usage:

**ping** hostname

Where *hostname* is either a character-string name of the target or an Internet address in standard form. (See the section on hostnames in the network overview for more details.) The hostname *me* will send an echo request addressed to the computer on which the command is typed.

**Ping** reports success with a message such as "Host x,y,z,w responding" where x,y,z,w is the Internet address of the target. It may also report one of several failures:

- Host not responding
- Host responded but the returned echo packet was defective
- The initial echo request packet could not be sent

When exiting, **ping** also prints an array of statistics about its operation. These statistics are in two categories: details of local network usage and details of packets processed. These statistics often provide clues about network problems to a network specialist.

### 21.1. Optional features

**ping -t** hostname

will go into a loop continually sending echo requests to host *hostname*, each time waiting for a response before sending the next request. To exit this loop, type the single letter "q". When in looping mode **ping** reports all echo failures, and also maintains a summary line of trials and successes.

## **ping -s**

starts an echo server, a program that will respond to echos sent to this PC from elsewhere in the network. (Note that all PC/IP programs, including **ping**, always act as echo servers whenever they are in control of the PC.)

### **21.2. Using ping to isolate network trouble**

When a host fails to respond, it may mean either that that host is down or that some network or gateway in the path from the user to the host is down. (It could also mean that the host does not implement the IP/ICMP echo request protocol.) Further **ping** experiments can usually determine which (and thus whom to call for repair.) A successful **ping** directed to another host on the same network as the original host usually means that the original host is down or not listening to the network. Failure to get echos from any host on that network means that the trouble is along the path somewhere. A **ping** directed to the gateway into the network in question is the next step. One can continue to work back from the target toward the originator until the point where communication breaks down is found.

The echo request sent by the **ping** command is dispatched using a low-level protocol that does not try to guarantee delivery. As a result, there is a possibility that any one echo request may be accidentally lost for some reason such as temporary overload in some gateway. Thus one cannot be confident that a particular network or gateway has failed unless a series of **ping** experiments consistently succeed in getting to the point in question and consistently fail to get beyond that point.

### **21.3. Using ping to evaluate a serial line**

Since **ping** contains a built-in echo server it can be used to test or evaluate a serial line in two ways. If a gateway is attached at the other end of the serial line, the command "**ping me**" exercises the serial line in both directions as well as the gateway. Alternatively, if one loops back the other end of the serial line so that all data sent down the line comes immediately back to the PC, the command "**ping me**" will still work, using an Internet address chosen by **ping**. In both cases, the command form "**ping -t me**" is appropriate to start a continual test of the line. Any packets damaged

ping

In transit will lead to error reports; the summary of tries and successes provides a picture of the total effect of line noise.

26 October 1984. This document is in ping.mss



## 22. Setclock

PC/setclock, version 6.1

A program to obtain a clock reading from a network time service and set the PC date and time accordingly.

Usage:

**setclock** [time-server]

where *time-server* is either a character-string name or an Internet address of a network host that provides an UDP time service. PC/setclock sends a request, using the standard UDP time service protocol, to *time-server*. If the name *time-server* is omitted, PC/setclock sends requests to a default list of Internet addresses of the known time servers. This list is stored within PC/setclock and can be set or changed with the customizer. PC/setclock takes the first response, converts the calendar clock reading found therein to the local date and time and displays it. Finally, PC/setclock calls the standard PC-DOS entry points to set the system date and time.

If the second letter of the customized time zone label is either 's' or 'S', from the last Sunday in April to the last Sunday in October, PC/setclock adjusts the local time one hour forward for Daylight Savings Time and changes the 's' to 'd' or 'S' to 'D'.

If no time server responds, or the network is not operational, PC/setclock displays a message to that effect and leaves the current date and time settings of PC-DOS unchanged.

PC/setclock is designed for use either as a stand-alone command or as a command invoked by an *autoexec.bat* batch file. There are two advantages to using PC/setclock in an *autoexec.bat* batch file. First, DOS does not ask the user to type the date and time on every bootload operation. Second, it provides an immediate test of whether or not the network connection is operational. If setclock receives at least one response, it returns to DOS with the DOS variable `ERRORLEVEL=0`; otherwise `ERRORLEVEL=1`.

## 22.1. Customization

The following parameters of `PC/setclock` can be customized with the `PC/custom` command:

- Local standard time offset, in minutes before GMT. West of GMT the value is positive, east of GMT the value is negative. For EST the value is +300. For SET the value is -60.
- Local standard time designation string. Three letters, such as EST, EDT, or SET. If the second letter is 's' or 'S' then `PC/setclock` automatically provides Daylight Savings Time during the appropriate part of the year.
- Internet addresses of up to five time servers. The servers are polled at two second intervals in the order they were set by the customizer, so one may place preferred services nearer the head of the list.

21 May 1985. This document is in file `setclock.mss`

## 23. Telnet

### PC/telnet, Version 8.0

A remote login program for the IBM PC, using the TCP/IP protocol and emulating a display terminal.

Usage:

**telnet** hostname  
or  
**telnet -p** portno hostname

Where *hostname* is either a character-string name of the target host, or an internet address in standard form. (See the section on hostnames in the network overview for more details.) When used with the "-p" option, the argument *portno* is used as a port number at the target machine. This feature is used to connect with certain **telnet**-like services available on some hosts.

From the point of view of the target host, **PC/telnet** emulates a standard "network virtual terminal". From the point of view of the keyboard user, **PC/telnet** emulates a Heath H19 terminal. The terminal emulation is only approximate. A set of conventions and list of incompatibilities appears on the next page.

Typing the command with the name or internet address of a target host causes **PC/telnet** to try to establish a connection. When that connection is successful, the target host should display its greeting banner. The following conventions apply to the translation between H19 emulation and network virtual terminal emulation:

1. Function key F10 is an escape used to invoke **PC/telnet** functions. F10 followed by a question mark displays a list of escape sequences. Others are F10 followed by:
  - a Send "Are You There?" inquiry to the target host.
  - b Send "break" to the target host.
  - c Close the connection and exit from **telnet**.

- e Send to target on every typed character.
  - l Local echo. (PC/telnet echos typed input.)
  - E Send to target only when End-Of-Line is typed.
  - q exit from telnet without closing connection.
  - r Remote echo. (Target host echos typed input.)
  - x Send any outstanding data now.
  - U turn on the line-25 clock and status report. (default is on)
  - u turn off the line-25 clock and status report.
2. Function key F10 is also used to change the mode of operation of the terminal emulator within PC/telnet. These escapes are:
- B Backspace key sends BS, control-backspace sends DEL.
  - D Backspace key sends DEL, control-backspace sends BS.
  - d If output line too long, discard extra characters.
  - w If output line too long, wrap around to next line.
3. The PC "Print-Screen" feature, triggered by key "PrtSc", can be used from within PC/telnet, but immediately preceding its use one must restore the display buffer to the format expected by PrtSc. Function key F10 typed twice does this format adjustment.
4. F10 also provides some TFTP server commands, discussed below.

### 23.1. Closing connections

At the end of a login session, some hosts will close the connection, in which case PC/telnet exits, returning to PC-DOS. Other hosts issue an invitation for another login. In the latter case, type F10 followed by "c" to close the connection and exit from PC/telnet. Other methods of exiting, such as F10 followed by "q", or powering down the PC, will leave a dangling TCP/Telnet connection that some hosts may not clean up properly. A later attempt to login to that host from the same PC may encounter interference from the unclosed previous connection.

If you close a connection without logging out, most hosts will deal with the situation in the same way they handle telephone line hangups. If you exit **telnet** without either logging out or closing the connection, the host may not realize you are gone, and there is no way to pick up the connection again. (The host, noticing lack of activity for a long time, may eventually log you out and close the connection.)

If you try to open a connection to a host that does not respond, **PC/telnet** will try eight times, then display an error message and exit. Note that this message may mean either that the target host is not listening to the network or that some network or gateway in the communication path to that host has failed. (The command **PC/ping** may be useful in isolating the trouble.)

Versions of **PC/telnet** are available for both local area networks and serial lines. On a serial line, at speeds below 9600 baud, the combination of remote echo and send-on-every-character modes causes display to fall far behind typed input. Local echo and send-on-newline modes are recommended for operation at lower line speeds.

## 23.2. Terminal emulation conventions and compatibility

The following conventions allow the PC keyboard to behave like that of a Heath H19:

1. There is no repeat key. To repeat any key, hold it down.
2. The function keys are keys F1-F5.
3. The color keys are F6(blue), F7(red) and F8(gray).
4. The H19 has separate keys for ASCII "Carriage Return" and ASCII "Line Feed". These two functions are combined on the PC "Enter" key. To send an ASCII CR, type "enter". To send an ASCII LF, type "control-enter".
5. The H19 has separate keys for ASCII "Backspace" and ASCII "Delete". These two functions are combined on the PC "back-space" key. To send ASCII DEL, type "backspace". To send ASCII BS, type "control-backspace". A customization option and an F10 escape allow interchanging backspace and control-backspace. For convenience, the keypad key labeled "Del" also sends an ASCII DEL.
6. Note that, like it or not, the emulator exactly emulates the Heath H19 line wraparound feature. That is, in line wraparound mode, the emulator automatically goes to the next line after placing a character in column 80.

rather than waiting to see if the program or typist will try to put something in column 81.

### 23.3. Heath H19 features not emulated

hold screen/scroll	keyclick disable
graphics	keyboard disable
shifted keypad	block cursor
alternate keypad	AutoCR
Identify as VT-52	AutoLF
transmit page	transmit 25th line
offline/online switch	parity enable
XOFF/XON flow control	most ANSI escapes
control-suppression of transmitting display mgt codes	
restore power-up configuration	
ESC x setting of parameters	
control-space does not send null (but control-2 does)	

### 23.4. File transfer with PC/telnet

The **PC/tftp** server package can be invoked while using **PC/telnet**. With this feature, one can use **PC/telnet** to log in to a remote host, and then move files between that host and the PC, using the other host's tftp command to control the transfer. Compared with initiating the transfer from the PC, this method has two advantages:

1. Because the user authenticates himself upon logging in to the distant host he can transfer any files to which he has access, not just publicly accessible files.
2. The user can invoke other commands on the distant host in conjunction with the transfer. (E.g., compile and load a program before sending it.)

Seven functions of **PC/telnet** support tftp service. They are invoked by typing function key F10 followed by:

- |   |  |
|---|--|
| T | Enable the tftp server. (This is the default mode of operation when Telnet starts.)  |
| t | Disable the tftp server (upon completion of any transfer currently in progress).   |
| I | Send this PC's internet address, in decimal format, as if it had been typed on the keyboard. This function simplifies the issuing of tftp commands on other systems. |

- o Same as l, but send in octal format.
- I Display this PC's internet address on line 25 in octal and decimal formats. (For use if the other system needs this address in some odd format.)
- y Accept a file transfer request.
- n Refuse a file transfer request.
- A Accept all file transfer requests, without asking. Typing F10/A again returns to the normal mode of operation.

When another host requests a file transfer to or from this PC, the PC/tftp server asks the PC/telnet user for permission to accept the request. (Type F10/y or F10/n.) For a successful transfer, the user must accept the request before the remote host loses patience, times out, and aborts the transfer. Hosts commonly have a 10 to 30 second timeout.

Further information on file transfer may be found in the description of PC/tftp service, and in the description of tftp usage for the remote host. Some hosts have a tftp command that is similar to the PC/tftp command, so that writeup in this manual may offer some help if no other documentation is available.

### 23.5. Nested calls to the DOS shell

The PC/telnet escape F10/! invokes a nested DOS command interpreter, permitting the user to invoke other DOS commands locally on the PC without shutting down the telnet connection. This feature requires a configuration of at least 192K bytes of memory, and while running the nested command interpreter, network commands cannot be used. One should not stay in DOS for extended periods because while in DOS, arriving messages are ignored, and if the host at the other end of the telnet tries to send such a message it may become impatient with the lack of response from the PC, and close the connection. To return to PC/telnet, issue the DOS command EXIT.

## 23.6. Customization

The following parameters of **telnet** can be customized with the **PC/custom** command:

1. The parameters for TCP window size and TCP low window are of particular interest to **PC/telnet** users. If one is communicating with a mainframe time-sharing system on the same local area network as the PC, it is recommended that a window size no larger than 350 bytes be used, with a low window of 150 bytes. Use of larger windows may lead to pauses when displaying large quantities of output. See the description of **PC/custom** for more explanation of these parameters.
2. Start in line-at-a-time mode or character-at-a-time mode.

## 23.7. DOS note

The DOS feature of redirecting output to a file cannot be used for **PC/telnet** display output.

## 23.8. Debugging note

There are several debugging features built in to **PC/telnet** that can be useful in tracing network problems. See the section "debugging options" for more information. Function key F10, followed by control-H, will produce a display of a list of those options.

16 September 1985. This document is in file **telnet.mss**

## 24. Term

### PC/term

A terminal emulator for the IBM Personal Computer. This program is not strictly a part of the PC/IP network software, since it makes no use of the IP protocol family. It turns the PC into a terminal so that it may be used to log in to hosts that provide dial-up login facilities. It is included as part of the PC/IP package because it uses a terminal emulation package that is identical to the one used in PC/telnet.

### 24.1. Operating instructions

- To run the emulator type the DOS command TERM. It will immediately activate "data-terminal-ready", which notifies the attached modem or host computer system of its presence. It then clears the screen and begins emulation, without any greeting messages.
- To exit the emulator without deactivating "data-terminal-ready", type F10, followed by "q". This method of exit leaves the modem or attached computer system with the impression that the terminal is still attached. (exit off-hook)
- To exit the emulator and deactivate "data-terminal-ready", type F10, followed by "c". This method of exit tells the modem or attached computer system that the terminal has been disconnected. (exit on-hook)
- To exit the emulator with "data-terminal-ready" restored to the value it had when the emulator command was first typed, type break (control-scrlllock).
- To send a break type F9.
- To set configuration options type F10, follow instructions.
- Standard terminal configuration options:  
     full/half duplex    line discard/wrap    baud rate
- Extra emulator configuration options:  
     reverse BS/DEL    normal/Inverse video    select serial port
- Type F10 again to exit from the option-setting menu.
- Type F10 twice before using the PC Print-Screen feature, to restore the screen buffer to the format expected by PrtSc.

## 24.2. Configuration customization

The "power-up configuration" may be customized by using the DOS DEBUG command as follows:

1. While in DOS, type the command "DEBUG TERM.COM"
2. To the debugger, type command "g" to enter the emulator.
3. Hit key F10 and set the desired power-up configuration.
4. Choose menu item "q" to return to the debugger.
5. Type command "w" to save the new configuration on the disk.
6. Type command "q" to return to DOS.

For a list of keyboard conventions and terminal emulation limitations, see the writeup of PC/telnet, which uses the same terminal emulator.

9 June 1983. This document is in file term.mss

## 25. TFTP

### PC/tftp, Version 7.3

A file transfer package for the IBM PC, using the UDP/IP protocol.

Usage:

```
tftp [ get ] local-file-name hostname foreign-file-name [octet]
      [ put ]
```

where

- The first argument should be *get* to move a file from another machine to the PC, or *put* to move a file from the PC to another machine.
- *local-file-name* is the name of the file in the file system of the PC.
- *hostname* is either a standard character-string name of the other computer, or the Internet address of that computer. See the section on hostnames for more details on this argument.
- *foreign-file-name* is the name of the file in the file system of the other computer. Note that the foreign computer may require that this file name be "fully qualified," that is it may need to include a directory name in idiosyncratic syntax in order that the foreign system can identify the wanted file. If the foreign file name syntax requires use of characters reserved by PC/DOS, then the name must be surrounded by double-quote marks. (The PC/DOS reserved characters are greater-than, less-than, and reverse slash.)
- The optional argument *octet* instructs **tftp** to move the file literally, byte-by-byte, from one computer to the other. If this argument is omitted, the file is assumed to be a text file, and **tftp** automatically performs any necessary character set conversions to and from the network standard character set representation, known as *netascii*. For compatibility, PC/tftp also accepts the argument *image* with the same meaning as *octet*.

### 25.1. Notes

Not all hosts implement TFTP service. It is currently available on most Multics, PDP-11 UNIX, VAX-UNIX, Alto, IBM PC, and TOPS-20 machines attached to the network.

TFTP does not demand a password from the user, so most foreign hosts are not willing to let just any file be transferred. As a general rule, one can move a file from a foreign host if that file is publicly accessible on that host. If it is protected from public access, it is usually protected also from TFTP *get* operations. Similarly, a file may be moved to directory in a foreign host only if that host would normally permit anyone to put files in that directory. An important restriction that most hosts enforce is that one may not *put* a file on top of an already-existing file of the same name. This restriction is especially important to understand if for some reason a *put* operation fails or is aborted. Despite the failure, the foreign host may have created an empty or partial file, with the name specified. Another attempt to *put* the file with the same name will then fail because of the access-control restriction.

It is possible to send a file to a printer on a remote PC that is running the tftp server, by giving a name such as "PRN" or "LPT1" as the foreign file name. See the writeup of the tftp server for more details.

All PCIP packages follow the DOS convention of returning an ERRORLEVEL value when they exit. In the case of PC/tftp, the value zero means that the file was successfully transferred, while the value one means that some error prevented completion of the transfer. The ERRORLEVEL feature is primarily of use if PC/tftp is invoked as a command from a DOS batch file.

The version of TFTP distributed with Berkeley 4.2 UNIX contains two defects that are often noticed only by PC/IP users. First, it ignores the using computer's specification of netascii or octet mode, and performs all transfers in octet mode. Thus when a text file is transferred to or from a PC the resulting file is not translated, and end-of-line characters are not properly represented. Second, if a single packet sent to the PC gets lost during the transfer, the 4.2 UNIX TFTP never retries and it ignores retries from the PC. Thus the loss of a single packet guarantees failure of that file transfer. A new TFTP which does not have these problems is available on the PC/IP source release.

See also the writeups of PC/tftp service and dialup line file transfer.

tftp

16 September 1985. This document is in file tftp.mss



## 26. TFTP Server

PC/tftp server, version 7.3

An implementation of a file transfer server for the IBM PC.

Usage:

```
tftp serve
or
tftp serve spool
```

The PC/tftp server package allows users at other network hosts to initiate file transfers to and from this PC. The option *spool* disables write blocking, to allow the tftp server to be used as a print spooler.

Notes:

- Server **tftp** can also be invoked from within the **telnet** command, while logged in to another host. See the writeup of PC/telnet for usage instructions.
- While server **tftp** is running, no other use can be made of the PC. To turn server **tftp** off, type "q". If a file transfer is already in progress, server **tftp** will shut down immediately, leaving the host at the other end of the transfer wondering where it went.
- There is no access control whatever. The **tftp** server allows a remote host to initiate a *get* or *put* operation for any file on any accessible disk. (The version of the **tftp** server that is invoked from PC/telnet asks the user for confirmation of each file transfer request that it receives.)
- The PC-DOS operating system is not designed for unattended use, so leaving a PC alone with the tftp server running does not work very well. For example, if the distant host tries to initiate a put to a write-protected diskette or unreaded disk drive, PC-DOS will stop in its tracks and ask the operator of this PC what to do. Until someone answers this query, the tftp server appears to be dead.
- In initiating file transfers from other hosts, the user at the other host must know the IP address of the PC that is running server tftp. This IP address may not be associated with any name table name. [In Berkeley UNIX 4.2, one can learn the IP address of the host originating a telnet connection by using the command "who am I". This feature simplifies transferring files back to the PC from which one originated a telnet connection.]

- PC-DOS will prefix any file name supplied by the foreign host with the default drive and the default working directory for that drive. To override these defaults, the foreign tftp initiator can supply a full drive descriptor and path name. However, because of the special characters (colons and backslashes) appearing in fully qualified PC-DOS file names, one may have to use some quoting convention on the foreign host to type the file name at command level. [For example, on another PC, path names should be enclosed in double quotes. On UNIX, back-slash characters should be doubled or replaced with forward-slash characters, which PC/tftp will accept instead.]
- The tftp server permits only one file transfer at one time. If any host requests a transfer while one is already in operation, the tftp server will refuse the second request.
- The tftp server can be used as a print spooler, simply by telling the tftp user to send files to the appropriate device file name (such as PRN or LPT1). When used this way, the usual write blocking done by the tftp server sometimes interferes, since the tftp server accumulates up to 10K bytes of transferred data before initiating the first write to the device. On trying to send the next block of data, the tftp client may then time out and give up because the server PC will concentrate all its attention on the printer for a long time. The server should be started with the option *spool* to disable write blocking.

See also the writeups of PC/tftp and PC/telnet.

16 September 1985. This document is in file tftps.mss

## 27. Whois

PC/whois, version 6.0

A program to obtain directory information about a registered user of another network host, using the TCP/IP *finger* protocol.

Usage:

**whois** name@host

Where *name* is the character string name of a registered user at the target host, and *host* is either a character-string name of the target host or an Internet address in standard form. (See the section on hostnames in the Introduction for more details.) If *name* is omitted, some hosts will respond with a list of currently logged-in users.

The **whois** command sends an inquiry, and displays the answer, if any. The form and contents of the answer are determined entirely by the target host. Note that some hosts do not respond to whois requests. They may either ignore the request (in which case PC/whois displays the message "...host not responding") or reject it (in which case PC/whois displays the message "Closed: foreign reset".)

If the target host is not forthcoming with a response, PC/whois will give up after about 20 seconds. Typing the letter "q" will cause PC/whois to abort the operation immediately. This feature is useful if one discovers that the quantity of output is more extensive than anticipated.

4 December 1984. This document is in file whois.mss



## 28. Status

This is a list of serious known bugs and features that, although described in this manual, are not actually implemented yet.

**PC/term:** control-scrolllock exits on-hook, rather than with DTR restored to its original value.

**PC/term:** at data rates of 4800 bits/second and below, when two-character sequences are transmitted in response to a single keyboard key, (such as for function keys and cursor controls) one of the characters is sometimes lost.

**Internet Protocol:** Because the PC implementation does not currently reassemble fragmented packets, none of the PC/IP packages can be used with hosts that gratuitously fragment large packets or through gateways that fragment packets. Currently MIT-Multics is the only known ARPANET host that gratuitously fragments large packets, making **tftp** service unusable for files larger than 128 bytes. (PC/**telnet** is usable with MIT-Multics, because Multics TCP never tries to send large packets.) Within the M.I.T. environment, large packets are fragmented only when they traverse the CHAOS network.

**PC/telnet:** If, while using the **tftp** server, a disk problem occurs that leads DOS to display a message (e.g., "Disk not ready: abort, retry or ignore?") DOS attempts to display the message without realizing that PC/**telnet** is operating the screen with offset pointers. Thus the DOS message may appear in a random place on the screen, cut apart in two pieces, or even not appear at all. If the user types a response to the question, the response will be accepted by DOS and (assuming that the DOS file operation is successful) the display returns to normal.

**PC/AT:** All PC/IP programs have been checked on the PC model AT using both the Ethernet and serial line drivers. Although all appear to work correctly, some problems that may be symptoms of lost interrupts have been noted. The most serious symptom is that while transferring files to or from diskettes, the diskette drive occasionally appears

## PC/IP User's Guide

to fall out of the ready status. (A retry always finds that the disk is actually ready.) Note that when this problem occurs, the error message that DOS produces triggers an instance of the previous problem.

16 September 1985. This document is in file status.mss

## 29. PC/IP bugs, tasks, problems, projects, and bright ideas

23 January 1986

\*\*\* means critical, needed immediately

\*\* means important to have soon

\* means would improve quality of operation significantly

(no stars) means would be nice to do when time permits

Asynchronous line driver:

1. Doesn't check size of incoming packets, can fail.
2. Exits if PC gateway doesn't respond. (Should return error.)\*
3. Port number and interrupt line should be a customizable configuration option.
4. Loses transmitted characters on escape sequences at low data rates.\*

proNET p1300 driver: (no known problems)

Interlan NI5010 driver: (no known problems)

3COM Etherlink driver: (no known problems)

IP protocol handler:

1. Doesn't reassemble fragmented packets.\*
2. Doesn't reply to time stamp or information requests.
3. Doesn't upcall on destination unreachable.\*

UDP protocol handler: (no known problems)

TCP protocol handler: (no known problems)

PC/telnet version 8.0

1. F10/close doesn't work while connection is being opened. (Should ensure that half-opened connection isn't completed, then exit.)\*

## PC/IP User's Guide

2. Output buffer full condition damages multi-character sequences.
3. Should catch DOS exit call on file errors and fix screen before DOS tries to display its error message.

### PC/ping version 5.0:

In test mode, line 25 should contain

- time since test started\*
- name and Internet address of ping target
- identification of the program in use
- number of packets sent and number lost

### PC/tftp version 7.3:

1. Shouldn't touch PC file until other site confirms willingness to try transfer.\*
2. Need way to shut off a transfer that isn't wanted.

### PC/tftp server version 7.3:

1. Should allow multiple connections.
2. Needs graceful shutdown after current transfer is complete.
3. Crashes when aborting after a disk write protect error.
4. When responding to a get, runs at half the speed of a put.

PC/onetname version 5.0: (no known problems)

PC/netname version 5.0: (no known problems)

PC/setclock version 5.0: (no known problems)

PC/custom version 2.2: (no known problems)

PC/term command:

1. Doesn't work on port 2.\*
2. Should display modem status register contents in F10 mode.
3. full/half duplex, line discard/wrap options should be per port.
4. F10/b should send break and return to emulation, to match **telnet**.
5. Two-character output sequences are lost at speeds of 4800 baud and below.
6. Control-scroll lock should exit with DTR restored to entry value.

**PC/onhook** and **PC/offhook** commands: (no known problems)

**PC/whois** command version 6.0: (no known problems)

**PC/nickname** version 2.0: (no known problems)

**PC/lpr** version 4.0: (no known problems)

**PC/monitor** version 1.4: (no known problems)

**PC/iprint** version 1.1

Needs control of Imagen font, etc.

**PC/netwatch** version 7.0

1. Should confirm source address, destination, type, length match in line 25.

Other general projects:

1. Need a canned response for "whois" requests. Need a polling "whois" to find out who is on PC's.

2 October 1985. This document is in file tasks.mss

# PC/IP User's Guide