Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Part II

Managing Local PC/TCP Tasks

Part III

Setting Up PC/TCP Windows Servers

Part IV

Setting Up PC/TCP DOS Servers

Part V

Troubleshooting PC/TCP

Part VI

Appendices

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 2 PC/TCP Installation Options

2.5 Manually Copying PC/TCP Files

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 3 Installing PC/TCP with Windows for Workgroups

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 4 Installing PC/TCP with Novell NetWare

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 6 Installing PC/TCP with Banyan VINES

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 7 Installing PC/TCP with PATHWORKS

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 8 Configuring NetBIOS

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 9 Configuring PC/TCP Remotely Using DHCP or Bootp

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 10 Using PC/TCP with Memory Managers

10.8 Using 386MAX

10.8.1 Using 386MAX with Windows

10.9 Related Information About Using Memory Managers

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 13 Configuring and Tuning the Kernel

13.6 Related Information About Configuring and Tuning the Kernel

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 14 Managing PC/TCP Servers with Server Control

Chapter 15 Configuring a PC as a Windows SNMP MIB-II Agent

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 17 Configuring a PC as a DOS LPD Server

17.8 Troubleshooting the LPD Server

17.9 Related Information About Configuring a PC as a DOS LPD Server

Chapter 18 Configuring Your PC as a DOS SNMP Agent

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 19 Troubleshooting Host Connections

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 20 Troubleshooting Name Resolution

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 21 Troubleshooting the Network Interface Card

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Chapter 22 Troubleshooting the Kernel and Driver Configuration

22.6 Isolating Software Interrupt Conflicts

22.7 Related Information About Kernel Configuration

Chapter 23 Troubleshooting Windows Integration

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

[PCTCP GENERAL]

[PCTCP HOST]

[PCTCP IDPRINT print_session]

[PCTCP IDRIVE]

[PCTCP IDRIVE filesys]

[PCTCP IDRIVE-RESTORE]

[PCTCP IDRIVE-SERVERS]

[PCTCP IDRIVE-USER]

[PCTCP IDRIVE-VXD]

[PCTCP interface n]

[PCTCP IP-SECURITY]

[PCTCP IP-SECURITY n]

[PCTCP KERBEROS]

[PCTCP KERNEL]

[PCTCP KERNEL-VXD]

[PCTCP LPD]

[PCTCP LPD lpd_printer_name]

[PCTCP LPR]

[PCTCP LPR print_session]

[PCTCP LWPE]

[PCTCP NETBIOS]

[PCTCP NETBIOS-VXD]

[PCTCP NNTP]

[PCTCP PCMAIL]

[PCTCP PCTCPNET]

[PCTCP PING]

[PCTCP POP2]

[PCTCP POP3]

[PCTCP PRINT]

[PCTCP PRINT-PROGRAMS]

[PCTCP QUERY]

[PCTCP SCREEN]

[PCTCP SERIAL n]

[PCTCP SESSIONS]

[PCTCP SMTP]

[PCTCP TAR]

[PCTCP 3270]

[PCTCP TIME]

[PCTCP TN]

[PCTCP VMAIL]

[PCTCP VPCTCP]

[PCTCP VT]

[PCTCP VXDINIT]

[PCTCP WMSG]

[CONTINUATION-FILE drive:\path\filename]

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Appendix B Writing Scripts with SLANG

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Appendix C The Standard MIB for SNMP Agents

Managing PC/TCP

PC/TCPâ OnNetä 1.1 and PC/TCPâ Network Software 3.0 (July 1994)

Appendix D Downloading Programmable VT Function Keys

Chapter 1

Overview of TCP/IP Network Information

This chapter provides an overview of the TCP/IP network information typically managed by a system administrator. The users, PCs, client applications, and servers of your network depend on this information to operate. Before you can configure PCs on your network to run PC/TCP“ for example, you must assign names and addresses to them that enable them to access the network and allow you to monitor their network activity more easily.

With this chapter, you can learn

- What you are expected to know before you start to manage network information.
- What protocols are included in the TCP/IP protocol family.
- How both TCP/IP and PC/TCP compare to the International Organization for Standardization’s Open Systems Interconnection (ISO/OSI) model.

- The conventions and syntax of host identification on a network.
- What UNIX network server files are commonly used by system administrators to manage network information.
- Where to find related information about managing network information.

Note: This chapter is not intended to replace other UNIX and TCP/IP system administration manuals that you may use as guides. Use this chapter to get started with basic TCP/IP network administration tasks.

1.1 Before You Start Managing Network Information

This chapter assumes that you have planned for or have prepared for use

- The hardware (computers, cables, connectors, and so on) of your network.
- The routers, bridges, gateways, and other network components for routing information over your network, or between your network and other networks.
- Personal computers (PCs) for your users.
- Computers to provide network services (network servers).

In addition, this chapter assumes that you have read or are familiar with the information contained in Getting Started. If you are unfamiliar with the prerequisite tasks listed in this section, refer to your UNIX or TCP/IP network administration guides for more information.

1.2 The TCP/IP Protocol Family

Networking protocols control communications between connected computers. The TCP/IP group of protocols allows networks and machines with different operating systems and machine architectures to communicate, making it a widely used standard on the Internet. Figure 1-1 shows the TCP/IP architecture, including applications, transport protocols, and the Internet Protocol (IP).

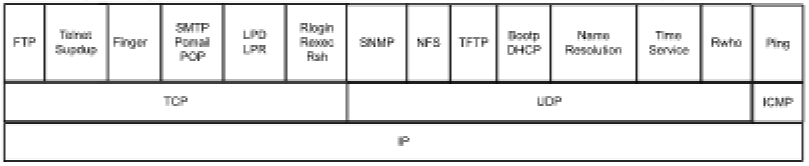


Figure 1-1 TCP/IP Protocols

Communication protocols are often grouped into hierarchies called “stacks,” “suites,” or “families.” When two network hosts communicate, their protocol stacks perform specified functions on data received from the other machine. Each protocol in the stack p IP

relies on the protocols beneath it to perform specific functions or services on data received from the other host, before relaying the data up the stack to the next level. The hierarchical levels of the stack are referred to as “layers.” At each layer, at least one protocol performs a service related to the specified function of that layer. These layers include:

P	Desc
ro	riptio
to	n
c	
ol	
A	TCP/
p	IP

plications	programs include file transfer, terminal emulation, mail,

network management, printing, and other services. Appli catio

ns

gene

rally

rely

on

either

the

Trans

missi

on

Contr

ol Protocol (TCP) or the User Datagram Protocol (UDP

) to

trans

port

data

to

their

peers

on

other

netw

ork

hosts

	.
T	Trans
C	missi
P	on
	Contr
	ol
	Proto
	col
	provi
	des
	conn
	ectio

n-oriented, reliable, sequenced data transfer betw een network hosts. TCP guarantees that data reaches its desti nation, and retransmits any data that fails to do so.

U User

D Data

P gram

Proto

col

provi

des

“unre

liable

”

data

trans

port

between local or remote hosts. Unlike TCP, UDP does

not

track

and

retra

nsmit

data

that

does

not

reach

its

	destination.
ICMP	Internet Control Message Protocol send

s

error

and

contr

ol

mess

ages

from

route

rs or

hosts

to the

message originator. ICMP also generates and replies to echo request packets (such as those used by the ping progr am). ICMP uses the basic support of the Internet Protocol

	(IP), and is also an integral part of IP.
IP	Internet Proto

col

recei

ves

data

bits

from

the

uppe

r

layer

s,

asse

mble

s the

bits

into

an

“IP

datag

ram”

pack

et (if

not

addr

essed to the local host), and chooses the best route to

	send the packet to its destination.
ARP	Address Resolution

Protocol determines the mappings between the physical hardware address, such as an Ethernet address, and the

IP address. Local area network (LAN) PCs use ARP as a method for locating each other on the LAN.

1.3 The ISO/OSI Model

The ISO/OSI (Open Systems Interconnection) model was developed by the International Organization for Standardization in the late 1970s to define a “universal” framework for a standard means of communication between different types of computers. The model’s seven “layers” define functions that take place when two machines communicate.

The ISO/OSI layers are described in Table 1-1:

Table 1-1 The ISO/OSI Layers

L NPur

a apos

y me

e e

r

7 AIncl

pude

ps

li user

c prog

aram

ti s,

o suc

nh as

elec

troni

c

mail

or

file

tran

sfer

prog

ram

s.

6 PRes

r pon

esible

sentation

for formatting application data so it can be displ aye

d or

print

ed

on

the

net

wor

k.

Con

verti

ng

encryption code or translating character sets are examples of the activities at this layer.

5	Session	Provides connection services, such as establishing

and maintaining a session. NetBIOS is a Ses sion

-

laye

r

prot

ocol

.

4 TProv

r ides

auser

n-to-

suser

port

communication and can perform error checkin

g. This layer is concerned with transmission relia bility. Protocols that operate at layers 4 and

3 are often referred to as “communicatio

n protocols.” TCP is an example of a Tran

		sport-layer protocol.
3	Netwo	Responsible for mak

r ing

k cert

ain

that

pac

kets

of

infor

mati

on

get

from

the source to the destination. IP is an exampl e of

a

Net

wor

k-

laye

r

prot

ocol

.

2 DProv

aides

t host

a-to-

Lhost

i deliv

nery

kacro

ss a

LAN

.

The

Dat

a

Link

laye

r

ass

emb

les

data

into

fram

es

and

tran

smits the m. It may also provide low-level error and flow control. This layer actually consists of two sublayers: the Media Access Control (MA

C)

laye

r

and

the

Logi

cal

Link

Con

trol

(LL

C)

layer. Protocols that operate at this layer include Ethernet, IEEE 802.5, token ring, and

PPP (Point-to-Point Protocol).

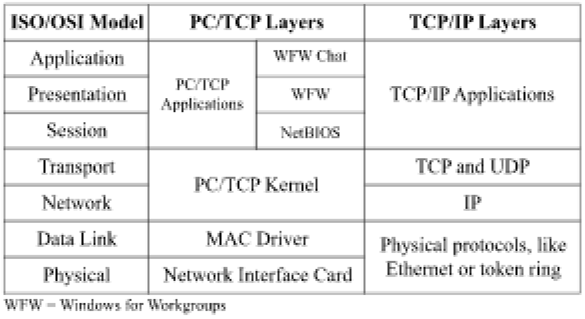
1 Phy Involves the

sical

electrical process of getting data from one host or point in a network to another. Examples of layer 1 devices include the transmission med ium, network interface card, or transceiver.

Since its development, the ISO/OSI model has become a prevalent way of classifying networking functions; it often provides a helpful frame of reference when looking at such protocol architectures as TCP/IP and PC/TCP.

Figure 1-2 shows how the TCP/IP and PC/TCP layers correspond to the ISO/OSI model:



ISO/OSI Model	PC/TCP Layers		TCP/IP Layers
Application	PC/TCP Applications	WFW Chat	TCP/IP Applications
Presentation		WFW	
Session		NetBIOS	
Transport	PC/TCP Kernel		TCP and UDP
Network			IP
Data Link	MAC Driver		Physical protocols, like Ethernet or token ring
Physical	Network Interface Card		

WFW = Windows for Workgroups

Figure 1-2 TCP/IP, the ISO/OSI Model, and PC/TCP

TCP/IP applications correspond to the top three layers (Session, Presentation, and Application) of the ISO/OSI model. TCP and UDP correspond to the ISO/OSI Transport layer, and the Internet Protocol is similar to the ISO/OSI Network layer.

PC/TCP contains driver, kernel, and application components that span the Data Link layer through the Application layer of the ISO/OSI model.

1.4 Identifying Hosts on the Network

Every system or host that you want to connect to your network must have a unique identification, so that data can be routed accurately to the host. On TCP/IP networks, all network hosts have an IP address and hostname. This section describes

- Obtaining a network number and domain name.
- Preparing a network addressing scheme.
- Creating subnets and specifying the subnet mask.
- Specifying the network broadcast address.
- Preparing a host naming scheme.
- Creating a local host table.
- Using a Domain Name System (DNS).

1.4.1 Obtaining a Network Number and Domain Name

The InterNIC is an organization in the United States that assigns network addresses to networks, allowing the networks to connect to the Internet. (There are world organizations that perform a similar function in other countries.) If a host is part of the Internet, its address must be unique on the entire Internet, so that any packets of data routed to the host can reach their proper destination.

Even when a network is not connected to the Internet, the administrators responsible for its addressing scheme usually comply with the Internet addressing conventions. If your network uses its own network addresses, you cannot connect your network to any network that is connected to the Internet, unless your network address is not already in use on the Internet. If, as is usually the case, your network address is not unique, you have to reconfigure every address on your network before connecting to the Internet. In addition, your network will be unable to receive packets from the Internet unless your network number has been registered in the Internet routing database. For these reasons, many administrators obtain an official network number application form from the InterNIC, then submit the completed form to the InterNIC to obtain their network addresses.

When the network number application is approved, the InterNIC assigns a unique “network number” to your network. The network number ensures that your network can be uniquely identified, and that your network’s hosts can be identified as belonging to it.

If you apply to the InterNIC for a network number, you should also obtain a domain name application form from the InterNIC, and return the completed form to the InterNIC for approval. A “domain name” is a unique network name that is associated with your network. (For example, FTP Software, Inc.’s domain name is ftp.com.) If the InterNIC approves your domain name application, the InterNIC assigns your network a unique domain that is associated with your network’s number, and that no other network can use.

Once you obtain a network number and domain name, you can assign IP addresses within the range of your assigned network number, and subdomain names as needed by your network. Information about contacting the InterNIC to obtain application forms appears in the Master Index, Glossary, Bibliography.

1.4.2 Preparing a Network Addressing Scheme

Each host machine on your network must be assigned a unique IP address to provide its network identification, which the host uses when communicating with any other network host. The address represents the host’s connection to a particular network. If you were to disconnect a machine from one network and move it to another network, you would have to give it a different IP address that accurately identifies its new location.

IP addresses are 32-bit numbers. By convention, IP addresses are represented by four sequential fields of decimal integers, separated by dots (.). The value of each field, referred to as an “octet,” or “byte,” can be from 0 to 255. Some examples of IP addresses are 128.127.50.55, 128.127.51.30, and 10.0.0.21.

Each IP address contains the identity of both the network that a host belongs to (the network ID or number) and the host itself (the host ID or number). Hosts on the same network have the same network number. For example, the IP addresses 128.127.50.55 and 128.127.51.30 belong to hosts in the same network, because their addresses begin with the same network number (128.127). The host with the IP address 10.0.0.21 belongs to a different network.

Class A Internet Addresses

The InterNIC has defined five IP address forms, known as “classes,” denoted by letters A through E. Each byte in an IP address signifies either a network or a host, depending on the address class to which the IP address belongs. The format for a Class A address appears in Figure 1-3.

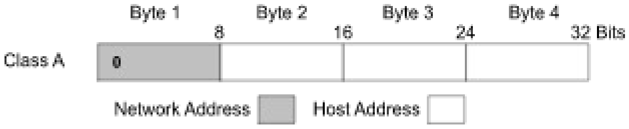


Figure 1-3 Class A Internet Address

In a Class A address, the first byte represents the network number, and the remaining three bytes specify the host number. The highest order bit in the first byte is set to 0; the next seven bits identify the network; and the remaining 24 bits (the second, third, and fourth bytes) identify local hosts. This format allows 128 (27) Class A networks, and over 16 million hosts on each of those networks. The InterNIC reserves numbers in this class for extremely large networks. The MILNET and some large commercial and university networks have Class A addresses.

The network portion of a Class A address (first byte) may use numbers 1 through 126 (the InterNIC reserves numbers 0 and 127) and its first bit must be set to 0. An example of a Class A address is 11.254.254.253.

Class B Internet Addresses

The format for a Class B address appears in Figure 1-4.

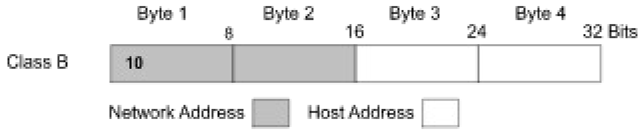


Figure 1-4 Class B Internet Address

In a Class B address, the first two bytes represent the network number, and the remaining two bytes specify the host number. The highest order bits in the first byte are set to 10; the next 14 bits specify the network address; and the remaining 16 bits (the third and fourth bytes) specify the host address. It is possible to have over 16,000 Class B networks (214), and over 64,000 hosts on each of those networks.

Class B network numbers, in the range 128.1 through 191.254, are used for the first two bytes of a Class B address. An example of a Class B address is 128.127.50.101. In this example, the network number is 128.127, and the host's number is 50.101.

Class C Internet Addresses

The format for a Class C address appears in Figure 1-5.

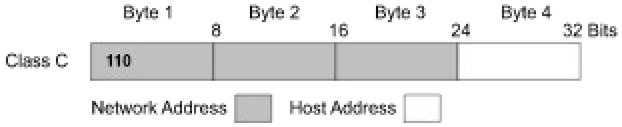


Figure 1-5 Class C Internet Address

In a Class C address, the first three bytes represent the network number, and the fourth byte represents the host number. The highest-order bits of the first byte are set to 110; the next 21 bits specify the network number; and the remaining 8 bits (fourth byte) specify the host number. This allows just over 2 million Class C network numbers (221) and 254 hosts on each of those networks.

Class C network addresses have the numbers 192.0.1 through 223.255.254. An example of a Class C address is 192.32.5.35. In this example, the network's portion is 192.32.5, and the host's portion is 35. Small networks often use class C addresses.

Class D and Class E Internet Addresses

The format for a Class D address appears in Figure 1-6.

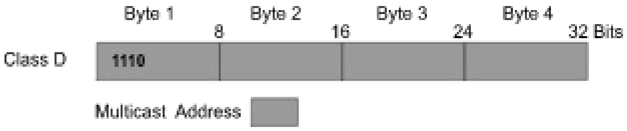


Figure 1-6 Class D Internet Address

The highest order bits of the first byte are set to 1110, and the remaining 28 bits specify the “multicast address.” IP multicasting is the transmission of one or more packets at approximately the same time to a “multicast group,” a set of one or more hosts identified by a special destination address (the multicast address). For more information on using IP multicast addresses, refer to RFC 1112 or your TCP/IP administration guide. The InterNIC does not assign Class D addresses.

Although the InterNIC has defined Class E addresses, with the highest-order bits set to 1111, no Class E addresses are in use. The InterNIC reserves these addresses for future use.

Assigning Host Addresses

Once you obtain a network number from the InterNIC, you can assign unique 32-bit addresses within the number’s range to the hosts in your network. For example, if the InterNIC assigns a Class B network number to your network, you assign an IP address to each of your network’s hosts with the same network number (the same first two bytes), but the last two bytes of each IP address must be unique for each host.

Note that the range of each field in the IP address is between 0 and 254 (255 is a broadcast address). You can assign your IP addresses one at a time, or you can assign a block of addresses to an organization at your site, and let that group assign the individual addresses to its machines. However you plan to perform this task, make certain that you avoid duplication of addresses, and that you record the addresses correctly on your network’s domain name servers.

1.4.3 Creating Subnets on the Network

You can use part of the host number to identify “subnets” within the larger network. For efficient operation, you can partition a large physical network into separate logical parts, or subnets, and connect them to the rest of the network with IP routers. (IP routers translate data link or media protocols to allow different network media to communicate.)

You create subnets on networks to serve various purposes, which include

- Connecting different physical networks. The networks become subnets of a larger internetwork connected by routers.

- Distinguishing between different network LANs.

- Isolating parts of the network. You may want to isolate an unstable part of the network from other parts, or you may want to restrict the traffic of one subnet from the others, to keep data secure.

- Delegating network administration to different groups. One way to delegate administrative tasks is to assign administrators to various subnets.

To create addresses for a network that will contain subnets, decide on the number of subnets that you need and how many hosts you plan to attach to each subnet. These numbers indicate how many bits of an IP address you can use for the subnet and host numbers.

For example, in a class B address, the last 2 bytes, or 16 bits, represent host numbers. Without subnets, it is possible to support over 64,000 unique addresses on one network (216). However, you can decide to use the first 8 bits of the host portion of the address for subnet numbers, and the remaining bits for host numbers. This format allows you to create 254 subnets with over 250 hosts on each subnet. In addition, you do not have to use all 8 bits to represent the subnet number; you can use just a portion of the bits, as in a 2-bit or 4-bit subnet number.

Machines on the same subnet share the same network and subnet numbers of their addresses. In the following list of examples,

128.127.50.2

128.127.50.36

128.127.52.7

if the entire third byte of each address is used as the subnet number, all three hosts belong to the same network (128.127), but only the first two hosts belong to subnet 50; the third host belongs to subnet 52.

Note: Do not use subnet numbers with bits set to 0000 0000 (0) or 1111 1111 (255); the former number was once used as a broadcast address, and the latter is now used as a broadcast address. For more information about the network broadcast address, refer to section 1.4.5, Specifying the Network Broadcast Address.

1.4.4 Specifying the Subnet Mask

In a network composed of subnets, each network host uses the same “subnet mask.” A subnet mask simplifies the routing of packets of information in a network by

- Concealing the host number of an IP address (all bits representing the host number are turned off, which gives them the value 0).
- Revealing the network and subnet bytes (all bits representing the network and subnet numbers are turned on).

A sending host compares its own network and subnet numbers to the revealed portion (the network and subnet numbers) of the destination’s subnet mask. If by comparison the two addresses differ, PC/TCP sends the packet to the appropriate IP router for forwarding to another subnet. If the revealed sections for the two addresses are the same, indicating that the destination is on the same subnet as the sending host, PC/TCP sends the packet directly to its destination.

When defining a subnet mask, note that the subnet bits must be both contiguous and adjacent to the network number in the address. Once you have defined the subnet mask, make certain that all hosts in the network have it, and make certain that all users specify the correct subnet mask in the network driver interface section of their PCTCP.INI files. For example, to specify an 8-bit subnet mask for a Class B address, a user would enter the following information into the [PCTCP ifcust 0] section:

```
[PCTCP ifcust 0]

subnet-mask=255.255.255.0
```

In this example, the 16-bit network number is turned on (255.255), the 8-bit subnet mask is turned on (255) and the host number is turned off (0).

1.4.5 Specifying the Network Broadcast Address

The “network broadcast address” is used to send a packet to every host on the network. In a network broadcast address, all bits in the byte(s) pertaining to the host number in the address are turned on (set to 255). For example, in the network 128.127, the broadcast address might be

```
128.127.255.255
```

The default broadcast address for PC/TCP is 255.255.255.255.

All hosts on the network should use the same network broadcast address. Make certain that, if you plan to use a broadcast address other than the default, your network users enter the broadcast address in the network interface section of their PCTCP.INI files.

For example,

```
[PCTCP ifcust 0]
```

broadcast-address=128.127.255.255

1.4.6 Preparing a Host Naming Scheme

Hostnames are assigned to machines in the network because people can remember names more easily than IP addresses. Once you obtain a domain name from the InterNIC, you can assign hostnames within that domain. All hostnames must be unique within a domain.

You can also create “subdomains” within your network’s domain. A subdomain is a domain contained within another domain. For example, company domains often fall into the Internet domain “com,” while military domains are categorized in the domain “mil.” Within the domain xyz.com, you may want to have subdomains named admin.xyz.com, sales.xyz.com, and shipping.xyz.com. Within the subdomain sales.xyz.com, you may create more subdomains, such as eastern.sales.xyz.com, southern.sales.xyz.com, france.sales.xyz.com, and asia.sales.xyz.com. All subdomains must be unique within a domain, and all hostnames must be unique within a subdomain.

A “fully qualified” domain name begins with a hostname and ends with a domain name. For example, in the name chocolate.sales.xyz.com, the host’s name is chocolate, the subdomain is sales, and the domain is xyz.com. Subdomains are optional additions to hostnames.

Note that subdomains are not associated with subnets. A subdomain may include information about hosts from different networks or subnets.

When choosing a host name, try to use a word that is easy to remember, easy to type, and will not have to be changed once it is assigned (unless you must change the name, such as when you give the machine to a different user). Refer to RFC 1178 for guidelines on choosing host names.

1.4.7 Creating a Local Host Table

A “host table” is a file that lists hostnames and their corresponding IP addresses. When the kernel is configured to have a local host table (there is a local host table path specified in the [pctcp kernel] section of the PCTCP.INI file), PC/TCP uses the local table to resolve hostnames first, then requests hostname resolution from a DNS server on the network if the local host table cannot satisfy the request.

In the PC/TCP host table, each entry describes one host by listing its IP address, one or more hostnames, and possibly any “aliases” associated with that address. (An alias is a short form of a fully qualified hostname. Both name servers and local host tables permit their use.) In this example,

```
#This machine is located in the network laboratory.

128.127.50.1 chocolate.xyz.com choc
```

the item

#This machine... Is a comment (a line of text that you do not intend to be interpreted as anything else), if preceded by a number sign (#).

128.127.50.1 Is the IP address of the host.

chocolate.xyz.com Is the hostname associated with the listed IP address. The hostname must be fully qualified. Note that hostnames are case insensitive.

choc Is the additional name, or alias, associated with the host’s IP address. The alias can change frequently as users add and remove informal names for their machines.

Separate items in a host table by spaces or tabs. You can list a maximum of 20 names (or 100 characters) for each IP address in the table.

When an entry appears in the table twice, PC/TCP returns the address for the first entry. If a hostname is not fully qualified in this table, PC/TCP may not be able to resolve the associated address.

Network administrators often rely on domain name servers to distribute host information throughout the network. Users can rely on local host tables if their domain name servers are temporarily unavailable, but these tables must be manually updated and kept consistent with the information on the domain name servers.

1.4.8 Using a Domain Name Service

A Domain Name System (DNS) server can also map hostnames to IP addresses. DNS often distributes current host information to network applications, like electronic mail programs or the ftp program, on remote hosts. Domain name servers keep the host information current for their domains, and share it with other domain name servers on a network or on other networks. DNS often makes the task of distributing host information on a network faster and easier, but the job of administering a DNS server is more complex than maintaining a local host table file.

For more information about guides on administering Domain Name System servers, refer to the Master Index, Glossary, Bibliography.

1.5 Files for Managing Network Information

After you have configured your network addresses and domain names, you can proceed to handle the administrative tasks involved in providing network services. The following table lists typical UNIX server files that an administrator uses most frequently during daily administrative tasks. If you are uncertain as to how to use these files, refer to your UNIX system administration manual for more information.

Note: The location and, in some cases, the names of these files may vary on your system. On some UNIX systems, the files can be found in the /usr or /etc directories.

Ad	UN
mini	IX
strat	Se
ive	rve

Task	r File Used
Configure the Internet dae	/etc/inetd.conf

mon

that

start

s

vari

ous

net

wor

k

serv

ices

for

your

net

wor

k.

Set /

sub etc

net /

mas net

ks. ma

sks

Map/

net etc

wor /

k net

addrwo

ess rks

es

to

net

wor

k

nam

es.

Assi/

gn	etc
and /	
stor	ser
e	vic
serv	es
er	
and	
dae	
mon	
port	
infor	
mati	

on.	
Configure system logging.	/etc/syslog.conf
Assign numbers	/etc/pro

to protocols.

Define network groups; assi

tocols

/etc/group

gn

sec

ond

ary

grou

p

me

mbe

rshi

p to

a

user

.	
Define network host names and address	/etc/hosts

es;

add

a

new

host

to

your

net

wor

k.

Allo /

w etc

rem /

ote ho

acc sts

ess .eq

to a uiv

host

with

out

requ

iring

a

pas

swo

rd.

Assi /

gn etc

and /

stor pa

e ss

userwd

acc

ount

infor

mati

on

(pas

swo

rds

are

encr

ypte

d).

Con /

trol etc

whic/

h ex

loca por

l file ts

syst

ems

are

avai

labl

e to

rem

ote

host

s

using NFS. Configure the file system. Ent

/etc/filesystems /

er	etc
rem	/
ote	fst
file	ab,
syst	/et
ems	c/v
into	fst
the	ab
file	
syst	
em	
conf	

igur	
atio	
n	
file.	
Con	/
figur	etc
e	/
term	get
inal	tyd
line	efs
s.	,
	/et

c/g

ett

yta

b, /

etc

/ter

mc

ap,

/et

c/tt

ys,

/et

	c/tt
	yta
	b
Con	/
figur	etc
e	/
print	pri
ers	ntc
on	ap
the	
net	
wor	

k.

List /

the etc

prog/

ram rpc

s,

by

proc

ess

nam

e

and

program number, that respond to remote procedur

e	
calls	
.	
Mai	/
ntai	usr
n a	/
list	lib/
of	ali
elec	as
troni	es
c	
mail	

aliases for the network and its mailing lists.

Control default system and user settings.	/etc/default
Keep a	/etc

list of users who are prevented from using ftp	/ftpusers

to

get

acc

ess

to

the

serv

er.

Kee /

p a etc

data/

bas ut

e of mp

user

s

curr

entl

y

logg

ed

in to

the

serv

er.

Record system logins and logouts.	/etc/wtmp
Send a mes	/etc/

sag mo

e of td

the

day

to

all

user

s

whe

n

they

log

in to

the

serv

er.

1.6 Related Information About Managing Network Information

For more information about managing network information, refer to the following sources:

Topic	Source
Basic concepts of PC/	Getting Started

TCP Guides and manuals that explain: TCP/IP

Master Index, Glossary, Bibliog

prot rap

ocol hy

s

ISO

/

OSI

Mod

el

Con

tacti

ng

the

Inte

rNI

C

Inte

rnet

add

ress

clas

ses

Net

wor

k

add

ress

ing

sch

eme

s

Net

wor

k

dom

ains

and

subdomains Host naming schemes DNS

serv

ers

1.4.8 Using a Domain Name Service

1.5 Files for Managing Network Information

1.6 Related Information About Managing Network Information

Chapter 2

PC/TCP Installation Options

PC/TCP offers a variety of installation options for the system administrator. Using the procedures defined in this chapter, you can customize the PC/TCP installation program to meet the needs of your group or site.

Note that the default PC/TCP installation procedure is defined in Getting Started.

Using the procedures in this chapter, you can

- Customize the installation program using the SETUP.INF file.
- Set up a network install.
- Specify a central network configuration method.
- Manually copy PC/TCP files.

This chapter also contains an overview of the network operating system chapters contained in Part II, Setting Up Special PC/TCP Installations.

2.1 Before You Choose an Installation Option

Before you use an installation procedure in this chapter, you should

- Decide how users at your site will install or upgrade PC/TCP software.
- Read the instructions in the SETUP.INF file on disk 1 of the PC/TCP installation diskettes.

2.2 Customizing the Installation Program for Your Site

This section contains procedures for customizing the PC/TCP installation program using the SETUP.INF file. This file is located on disk 1 of your PC/TCP installation disks.

Using the SETUP.INF file, you can configure

- Defaults for required installation information, including which Packet Driver to install, IP addresses, and domain names.
- The installation program dialog boxes shown to the end user.
- The PC/TCP components that the end user can choose from and the sets of files copied for each component.
- The contents of Windows Program Manager groups.
- A no-questions-asked batch install.

- Multiple installation programs (multiple SETUP.INF files), each tailored to the needs of a specific user (client) group.

Note that any changes that you make to the PC/TCP installation program affect the accuracy of the client installation procedures defined in Getting Started. You may want to supply the clients at your site with only the Express Install card or appropriate sections of the Getting Started guide.

Refer to the instructions in the SETUP.INF file for more information about the options described in this section.

2.2.1 Specifying Default Installation Information

You can provide default installation information, the answers to the questions that the installation program asks the user, by configuring the [answers] section of the SETUP.INF file.

The following example shows how this section can be used to specify a PC/TCP destination directory for the PC/TCP Express installation method (in Windows and DOS):

```
[answers]
destdir="c:\newpctcp"
```

You can also specify a default Packet Driver using the network-driver= parameter. You may want to use this parameter in conjunction with a batch install, if many or all of the systems at your site use the same network interface card.

The following example shows an [answers] section with the corresponding driver section for a 3C503 packet driver:

```
[answers]

destdir="c:\newpctcp"

network-driver=pd$3c503
```

Note that the card you specify must be defined in the [netcard] section of the NETCONF.INF file (which can also be found on the PC/TCP installation disks).

2.2.2 Defining Installation Program Dialog Boxes

You can control the installation program dialog boxes that the user sees using parameters in the [options] section of the SETUP.INF file. Each dialog box has a parameter that determines if that dialog is shown. By default, all installation program dialog boxes are shown.

To “hide” a dialog box, set the show-screen-dialog parameter for that screen to no. The following example shows a configuration where the user does not see a destination directory or kernel type dialog:

```
show-license-dialog=yes

show-directory-dialog=no

show-kernel-type-dialog=no

show-components-dialog=yes
```

```
show-driverlist-dialog=yes

show-ipconfig-dialog=yes

show-dnsconfig-dialog=yes
```

You might use these parameters to hide a dialog box for which you have provided default information in the [answers] section of the SETUP.INF file, such as the destination directory or the Packet Driver to install.

Note: If you do not show an installation screen, you must provide the information required by that dialog, either by adding other entries to the SETUP.INF file or configuring user files manually after installation. Refer to the parameter descriptions in the SETUP.INF file for more information.

2.2.3 Specifying PC/TCP Components and Copied Files

The PC/TCP installation program provides a Custom installation that lets the user select which PC/TCP components, or file groups, they want to copy to their system (in addition to a base set of files that provide basic network connectivity). Each component contains a default set of files.

To change default PC/TCP components

Edit the [components] section of the SETUP.INF file. This section lets you control which components the user can choose from, and the name of those components. The following example shows the default components and labels:

```
[components]

nfs-client= ”Using Network Files and Printers (NFS Client)”

email= ”Exchanging Mail and News”
```

```
remotelogin= ”Logging in to a Remote Host (Telnet and Rlogin)”

filetrans= ”Transferring Files (FTP)”

printing= ”Printing {LPR)”

backup= ”Archiving and Restoring Files (TAR and RMT)”
```

To change the files copied for a component

Edit the [install] or appropriate component section in the SETUP.INF file. These sections determine which of the files specified in FILES.INF will be copied during installation. FILES.INF is located on PC/TCP installation disk 1.

The [install] section defines the files copied as part of the base set of PC/TCP files (which all users receive by default, regardless of other selected components).

Other sections define the files copied for a particular PC/TCP component. The entries in these sections are used only if the user selects the appropriate component when installing. For example, if a user selects the Remote Login and Printing components, the installation program uses the following sections to determine the appropriate files to copy:

```
[remote login]
files=drlogin, @dosfiles
files=wrlogin, @winfiles
files=rlogin
files=keymap
[printing]
files=dprint, @dosfiles
files=wprint, @winfiles
files=pctcpnet, @winfiles
```

2.2.4 Configuring Program Manager Groups and Icons

To configure which PC/TCP applications appear in Windows Program Manager groups, edit the [groups] and appropriate application section ([dosapps] or [winapps]) . You can also use these sections to revise program group or item names, or to remove PC/TCP program groups altogether.

In the following example, the [groups] and [winapps] sections have been configured to install only the WinApps program group and the Windows Statistics and Ping program items within that group:

```
[groups]

dosapps= ”PC/TCP DOSApps”,dosapps,no

winapps= ”PC/TCP WinApps”,winapps,yes

[winapps]
```

```
winet= ”Statistics”,yes,@targetdir\winet

wping= ”Ping”,yes,@targetdir\wping

wdial= ”Dialer”,no,@targetdir\wdial

wtar= ”Archiver”,no,@targetdir\wtar
```

2.2.5 Specifying Multiple SETUP.INF Files for Multiple User Groups

You can configure multiple SETUP.INF files if you want to provide different installation programs for different groups of users. This option is especially useful when setting up a network install.

To use multiple SETUP.INF files

1. Create and configure the number of .INF files that you need and name each file uniquely. For example, GROUP1.INF, GROUP2.INF, and so on.
2. Instruct the end user to specify the /h parameter and the appropriate .INF file when they enter the command to run the PC/TCP installation program.

The following example shows a command line to run the PC/TCP Setup program in Windows with an alternate .INF file:

```
a:\setup /h a:\GROUP1.INF
```

2.2.6 Configuring a Batch Install

To configure an installation program that does not require user input, set the parameter batch-install= to yes in the [options] section of SETUP.INF. (The default is no).

When you specify a batch install, the installation program searches the [answers] and [components] section of your SETUP.INF file for required information. Any required information not specified in the SETUP.INF must be entered manually (to the appropriate system file) before you can use PC/TCP.

The batch install completes the installation program based on the value specified for the batch-end= parameter. Possible values for this parameter are

exit	End the program without rebooting (the default)
exit-win	End the program and exit Windows.

Note: If you select this option, PC/TCP will not delete temporary files created during installation; you must delete these files manually.

reboot End the program and reboot the user’s system.

The following example shows the configuration for a batch install that reboots the user’s PC upon completion:

```
[options]

batch-install=yes

batch-end=reboot
```

2.3 Specifying a Central Configuration Method

If you intend to set up a server from which your users can obtain network and PC/TCP configuration information, you must set up the server and specify a central configuration method in the [OPTIONS] section of SETUP.INF.

To enable central configuration

1. Configure a Bootp or DHCP server to provide configuration information.
2. Specify the central configuration method appropriate for your server in the [OPTIONS] section of SETUP.INF. Possible values are bootp, dhcp, and none (the default).

The following example shows the appropriate entry to enable use of a Bootp server:

```
[options]

central-configuration=bootp
```

For client configuration procedures, see Chapter 9, Configuring PC/TCP Remotely Using DHCP or Bootp.

2.4 Setting Up a Network Installation

This section defines the procedure for configuring a network installation. You may choose to have your users install over a network in order to simplify and standardize the installation process. Users at sites that are licensed for multiple installations, and that have installed previous versions of PC/TCP and InterDriveâ, may do upgrades from a common server using InterDrive.

To make this possible, the network administrator must copy all the files from the distribution diskettes to the remote directory, then instruct each remote user to connect to that drive and directory and run the installation program.

To set up a network install

1. At the DOS prompt, enter the command to create a new directory.

The following example creates the directory \NEWPCTCP:

```
mkdir n:\newpctcp
```

2. Copy the PC/TCP distribution diskettes into a directory on a remote machine.

You should use the DOS XCOPY command to copy any subdirectories. Repeat the following procedure for each PC/TCP distribution diskette:

```
xcopy /s a:\*.* n:\newpctcp
```

3. Delete any existing FILES.INF file, then rename the file NETFILES.INF to FILES.INF:

```
del n:\newpctcp\files.inf

ren n:\newpctcp\netfiles.inf n:\newpctcp\files.inf
```

4. Instruct your end users (or clients) to configure (using Interdrive) drive N to point to the remote machine, then mount that drive.

5. Instruct your users to enter the following command lines.

For Windows users:

From the Windows Program Manager, select Run from the File Menu, then enter

```
n:\newpctcp\setup
```

For DOS users:

Enter the following at the DOS command prompt.

```
n:\newpctcp\install
```

This procedure also applies to other network operating systems, such as Banyan VINES, Novell NetWare, and Microsoft LAN Manager.

2.5 Manually Copying PC/TCP Files

If you need to copy additional files from the PC/TCP distribution diskettes, use the following procedure.

Note: FTP Software recommends that you use the PC/TCP installation program to copy PC/TCP files.

To copy a file from the PC/TCP installation disks

1. Place the disk containing the file to be copied in your disk drive.

2. To copy files in Windows — Choose Run from the Program Manager. Type the drive letter and setup /z For example,

```
a:\setup /z a:\filename .ex_ c:\pctcp\filename .exe
```

To copy files in DOS — Type the drive letter and install at the DOS prompt. For example,

```
a:\install /z a:\filename .ex_ c:\pctcp\filename .exe
```

Chapter 2 PC/TCP Installation Options

Chapter 3

Installing PC/TCP with Windows for Workgroups

Windows for Workgroups is a network operating system developed by Microsoft Corporation.

This version of PC/TCP is compatible with Windows for Workgroups 3.11.

PC/TCP provides TCP transport and RFC NETBIOS transport in both TSR (terminate-and-stay-resident) and VxD (Windows virtual device driver) implementations.

Windows for Workgroups works with PC/TCP's VxD kernel via NDIS (Network Driver Interface Specification) over either NDIS3, NDIS2, or ODI (Open Datalink Interface) drivers.

Using this chapter you can learn the following:

- How to configure Windows for Workgroups with PC/TCP for use on a single LAN (local area network).

This system configuration supports NETBIOS services using Windows for Workgroups NETBEUI only. It does not use PC/TCP NetBIOS. Use this configuration with systems that do not need to support NETBIOS services across a router (that is, if all NETBIOS traffic is on a single LAN).

- How to configure Windows for Workgroups with PC/TCP NetBIOS for use over a router.

These system configurations provide RFC NETBIOS functionality beyond a local subnet (for example, with remote Windows for Workgroups workstations).

You can configure your system to use NETBEUI and PC/TCP NetBIOS (TCP/IP) in parallel, or you can configure Windows for Workgroups to use PC/TCP NetBIOS instead of NETBEUI.

To configure Windows for Workgroups with PC/TCP, you must first configure Windows for Workgroups and PC/TCP to run over NDIS or ODI drivers. For detailed procedures see section 3.2, Configuring PC/TCP and Windows for Workgroups to Run over an NDIS Driver.

If you need to support NETBIOS services over a router, follow the additional configuration instructions in section 3.5, Configuring Windows for Workgroups to Use PC/TCP NetBIOS.

3.1 Before You Install PC/TCP with Windows for Workgroups

Verify Windows for Workgroups connectivity over NDIS drivers before you install PC/TCP.

This chapter assumes that you are configuring your system with NDIS3 drivers.

Note: You do not need to reinstall Windows for Workgroups if it is already configured on your system.

To use Windows for Workgroups with NetBEUI transport and NDIS drivers, you should configure Windows for Workgroups with the Real and Enhanced Mode (NDIS2/NDIS3) option.

To use Windows for Workgroups with PC/TCP NetBIOS and NETBEUI in parallel, select the LAN Manager option under the Options for Enterprise Networking menu in the Startup Settings dialog box of the Network icon of the Control Panel.

For more information on how to configure your Windows for Workgroups over NDIS drivers, see the Windows for Workgroups technical documentation.

3.2 Configuring PC/TCP and Windows for Workgroups to Run over an NDIS Driver

With this configuration, Windows for Workgroups stack sends NETBEUI packets and the PC/TCP stack sends IP packets.

Figure 3-1 illustrates the layering of the protocol stacks when you configure your system to use PC/TCP with Windows for Workgroups over an NDIS3 driver.

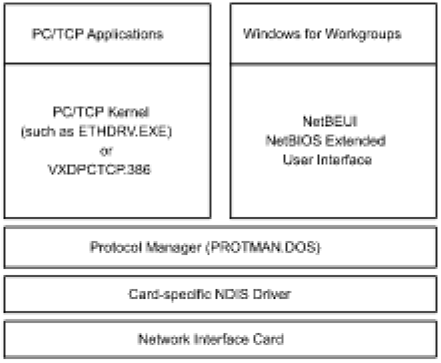


Figure 3-1 Windows for Workgroups and PC/TCP over an NDIS Driver

PC/TCP and Windows for Workgroups installation programs both modify the following system initialization files:

- AUTOEXEC.BAT

- CONFIG.SYS
- PROTOCOL.INI
- SYSTEM.INI

Before you proceed to use PC/TCP with Windows for Workgroups, you should verify that each of these files is correct.

To review the AUTOEXEC.BAT file

Examine your AUTOEXEC.BAT file to verify that

- The Microsoft netbind command is deleted or commented out.
- The net start command precedes the line that any TSR programs (such as the TSR kernel or the VXDINIT.EXE VxD initialization TSR).

This example starts Windows for Workgroups with the PC/TCP ETHDRV TSR kernel:

```
PATH C:\PCTCP
```

```
SET PCTCP=C:\PCTCP.INI

C:\WINDOWS\NET START

C:\PCTCP\ETHDRV
```

This example starts Windows for Workgroups with the PC/TCP VxD kernel:

```
PATH C:\PCTCP

SET PCTCP=C:\PCTCP.INI

C:\WINDOWS\NET START

C:\PCTCP\VXDINIT
```

Note: This example does not load PC/TCP NetBIOS. It assumes that vnebp=no in the [pctcp vxdinit] section of your PCTCP.INI file. (This is the default.)

For detailed procedures for loading PC/TCP NetBIOS and the PC/TCP Interdrive NFS client, see section 3.6, Loading PC/TCP NetBIOS and InterDrive.

To review the CONFIG.SYS file

Examine your CONFIG.SYS file to verify that

- These device statements are deleted or commented out:

– The Windows device driver

– The PC/TCP NDIS2 packet-driver converter (DIS_PKT.GUP)

– The protocol manager (PROTMAN.DOS), and

– Any NDIS card-specific driver

- This device statement appears.

```
device=c:\windows\ifshlp.sys
```

To review the PROTOCOL.INI file

Examine the PROTOCOL section of the PROTOCOL.INI file and verify that includes

the appropriate binding.For example,

```
[PKTDRV]

drivername=PKTDRV$

BINDINGS=MS$ELNKII

intvec=0X60

chainvec=0x65
```

Where BINDINGS= identifies the name of the Windows for Workgroups driver section in this file.

Note: BINDINGS must be capitalized, and there should be no spaces around the equals sign.

To review the SYSTEM.INI file

Examine your SYSTEM.INI file to verify that

Th Doe

is	s
se	this
cti	
on	
[b	Spe
oo	cifie
t]	s the
	Win
	dow
	s for
	Wor
	kgro

ups network driver, followed by a line identifying the

PC/

TCP

netw

ork

drive

r.

For

exa

mple

,

[boot

]

```
netw

ork.d

rv=w

fwne

t.drv

seco

ndne

t.drv

=c:

\pctc

p\pct

cpne
```

	t.drv
[b	Des
oo	cribe
t	s the
de	Win
sc	dow
rip	s for
tio	Wor
n]	kgro
	ups
	3.11
	drive

r.

For

exa

mple

,

[boot

.des

cripti

on]

netw

ork.d

rv=M

icros

oft

Win

dow

s for

Wor

kgro

ups

3.11

[3	Doe
86	s not
En	cont

h] ain

entri

es

for

WF

WFT

P.38

6 or

VPC

TCP.

386.

Rem

ove (or comment out) these entries if they appear.

[network drivers]	Contains entries specifying the name of your network

driver. For example, [network drivers] netcard=

your

_ND

IS_d

river

_file

nam

e.ext

trans

port

=ndi

shlp.

sys,*

netb

eui

Load

RM

Driv

ers

= No

The

aster

isk

(*) in

the

NETBEUI file name is a valid character.

Note:

Only if you are using NDIS2 and the TSR kern el,

add

the

NDI

S2

pack

et-

drive

r-

conv

erter

(\pat

hna

me\d

is_p

kt.gu

p) to

the

trans

port

=

line,

and

set

Load

RM

Driv

ers=

Yes.

3.3 Configuring PC/TCP and Windows for Workgroups to Run over an ODI Driver

ODI is a driver specification developed by Novell to provide the coexistence of NetWare and protocols other than Novell's proprietary IPX (such as TCP/IP and AppleTalk).

PC/TCP can communicate with NetWare 386 servers (3.x and above) using PC/TCP DIX Ethernet and IEEE 802.5 Token Ring kernels.

Note: If you want NETBIOS connectivity over routers (with PC/TCP and Windows for Workgroups over ODI drivers), you must configure the PC/TCP TSR kernel and the PC/TCP NetBIOS TSR.

For more information about configuring PC/TCP NetBIOS, see section 3.5, Configuring Windows for Workgroups to Use PC/TCP NetBIOS.

This is a sample AUTOEXEC.BAT file:

Ent To

er
this
line
SE
T
PC
TC
P=
C:
\P
CT
CP

loa
d
this
PC/
TC
P
con
figu
rati
on
file
pat

\P hna

CT me.

CP.

INI

C: Star

\WI t

ND the

O net

WSwor

\N k.

ET

ST

AR

T

C:

\N

W

CLI

EN

T\L

SL.

CO

M

Link

Sup

port

Lay

er.

(Thi

s

file

co

me

s

with

the

Net

War

e

dist

ribu

tion

.)

C: MLI

\N D

W
CLI
EN
T\3
C5
03.
CO
M

layer. (This is the ODI driver that comes with

you

r

net

wor

k

ada

pter

car

d.)

Thi

s

pro

gra

m

usu

ally

has

the

sa

me

na

me

as

you

r

net

wor

k

inte

rfac

e

with

a .C

OM

ext

	ensi
	on.
C:	Nov
\N	ell
W	prot
CLI	ocol
EN	stac
T\I	k
PX	file.
OD	
I.C	
OM	

C:\PCTCP\ODIPKT.COM	ODI-to-packet driver converter. You can use

co

mm

and

line

opti

ons

to

furt

her

spe

cify

the

bin

din

gs

that

ODI

PKT

.CO

M

sho

uld

perf

orm

.

For example,

odi

pkt

[?]

[unl

]

[sint

=

xx]

[mli

d=

na

me]

[fra

me

=

fra

mestri

ng]

Her

e is

an

exa

mpl

e of

loa

din

g a

driv

er

at

inte

rrup

t

65:

C:

\>o

dipk

t

sint

=65

mlid

=3c

503

fra

me

=ET

HE

RN

ET_

SN

AP

C: Net

\N War

W e

CLI redi

EN rect

T\V or.

LM Not

.EX e:

E

You

can

alte

rnat

ivel

y

loa

d

the

NE

T X

Net

War

e

shel

l

	file.
C:	For
\P	the
CT	PC/
CP	TC
\ET	P
HD	TS
RV.	R
EX	ker
E	nel.
C:	Win
\WI	dow

ND	s
O	for
WS	Wor
\O	kgr
DI	oup
HL	s
P.E	ODI
XE	inte
	rfac
	e.

For detailed procedures for configuring PC/TCP over ODI drivers, see section 4.2,

Configuring NetWare and PC/TCP over ODI Drivers.

3.4 Using PC/TCP Network Drivers with Other Network Operating Systems

The PC/TCP Network Driver (PCTCPNET.DRV) allows you to access remote files and printing services through the Windows File Manager, Control Panel, and Print Manager. If you use other network operating systems concurrently with PC/TCP software, these systems must be connected to enable access to each system’s services. The configuration of these network operating systems is referred to as “chaining” one network to or from another.

The PC/TCP installation program detects other network operating systems on your system and makes the appropriate changes to the [BOOT] section of your SYSTEM.INI file, as outlined in the following table:

Net	SY
wor	ST

k Operating System(s) PC/TCP only	EM.INI entries [BOOT] ne

tw

or

k.d

rv

=p

ctc

pn

et.

dr

v

PC/ [B

TC O

P and Windows for Workgroups

OT]network.drv=wfwnet.drv

se

co

nd

ne

t.d

rv

=p

ctc

pn

et.

dr

v

PC/TCP, Windows for Workgroups, and

[BOOT] network.drv=wfwne

another network operating system (such	t.drv second net.drv=pctcpnet.

as	dr
Novv	
ell	pct
Net	cp
War	ch
e)	ain
	=n
	et
	wa
	re.
	dr
	v

Note: You should verify that the [386EnH] section does not contain entries for

WFWFTP.386 or VPCTCP.386.Remove (or comment out) these entries if they appear.

3.5 Configuring Windows for Workgroups to Use PC/TCP NetBIOS

Configuring Windows for Workgroups with PC/TCP NetBIOS gives you RFC NETBIOS functionality over routers.

Note: PC/TCP supports RFC NetBIOS transport in both VxD and TSR implementations.

You can configure Windows for Workgroups to use PC/TCP NetBIOS in two ways:

- With PC/TCP NetBIOS and Windows for Workgroups NETBEUI in parallel.

Use this configuration to support a non-homogeneous NETBIOS environment. For example, if you need to interoperate with NETBEUI-compatible applications.

- With PC/TCP NetBIOS instead of Windows for Workgroups NETBEUI.

This configuration is most efficient if you do not need to interoperate with NETBEUI-compatible applications. Using this configuration you can reclaim memory that was reserved for NETBEUI.

Note: In general, the default PC/TCP NetBIOS configuration is adequate for most operating environments. If necessary, you can customize PC/TCP NetBIOS by editing the [pctcp netbios] section of the PC/TCP configuration file (PCTCP.INI). For more information, see Chapter 8, Configuring NetBIOS.

3.5.1 Configuring Your System to Use PC/TCP NetBIOS and NETBEUI in Parallel

Before you begin, follow the steps in section 3.2, Configuring PC/TCP and Windows for Workgroups to Run over an NDIS Driver.

Figure 3-2 illustrates the protocol stack when you configure Windows for Workgroups (NETBEUI) and PC/TCP NetBIOS in parallel. Using this configuration, you can support a non-homogeneous NETBIOS environment.

Windows for Workgroups sends/receives NETBIOS traffic via

- RFC NETBIOS (using PC/TCP NetBIOS).
- NETBEUI (using Windows for Workgroups).

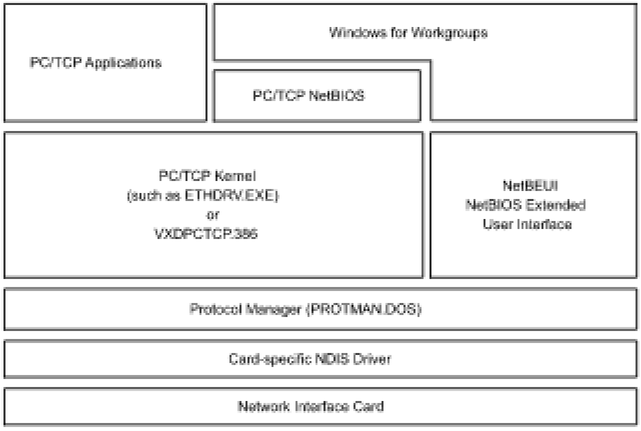


Figure 3-2 Windows for Workgroups (NETBEUI) and PC/TCP NetBIOS in Parallel

Follow these steps to edit your system initialization files to use PC/TCP NetBIOS with Windows for Workgroups(NETBEUI) in parallel:

1. Revise entries in your PROTOCOL.INI file to use PC/TCP NetBIOS.

In the [network.setup] section, change the lana0= entry to lana n, where n is the next available adapter number. For example,

```
[network.setup]

lana2=ms$elnk3,1,ms$netbeui
```

In the [ms$netbeui] section, change LANABASE=0 to LANABASE= n, where n is the same number as the adapter number specified in the [network.setup] section. For example,

```
[ms$netbeui]
```

```
lanabase=2
```

2. Revise entries in your PCTCP.INI file to configure PC/TCP NetBIOS.

If you need to support NETBIOS access over routers, you must configure PC/TCP NetBIOS to specify a name file and a broadcast file (using the namefile= and broadcastfile= parameters in the [pctcp netbios] section of your PCTCP.INI file).

For example,

```
[netbios]

namefile=c:\pctcp\namefile.txt

broadcastfile=c:\pctcp\broadcast.txt
```

Where namefile.txt and broadcast.txt contain entries that identify NETBIOS hosts beyond your local LAN.

For more information about customizing your NetBIOS configuration, see Chapter 8,

Configuring NetBIOS.

Note: When you configure your system to use PC/TCP NetBIOS, you may need to configure the kernel with additional TCP connections. As a general rule, one NetBIOS session requires one TCP connection. For a detailed procedure for configuring the kernel with additional TCP connections, see section 13.3.3, Adjusting the Number of TCP and UDP Connections.

3.5.2 Configuring Your System to Use PC/TCP NetBIOS (instead of NETBEUI)

This configuration lets you use PC/TCP NetBIOS instead of Windows for Workgroups (NETBEUI).

Figure 3-3 illustrates the layering of the software stack.

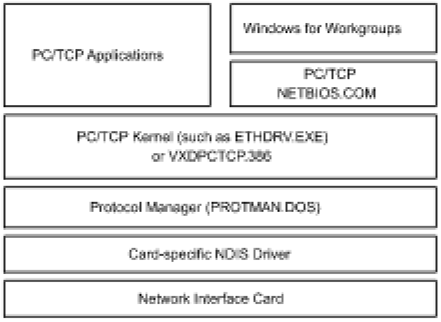


Figure 3-3 Windows for Workgroups with PC/TCP NetBIOS (only)

Follow these steps to configure your system to use PC/TCP NetBIOS (instead of NETBEUI):

1. Follow the steps for configuring your system with Windows for Workgroups over NDIS drivers. For more information, see section 3.2, Configuring PC/TCP and

Windows for Workgroups to Run over an NDIS Driver.

2. Follow the steps for configuring your system to use PC/TCP NetBIOS. For more information, see section 3.5.1, Configuring Your System to Use PC/TCP NetBIOS and NETBEUI in Parallel.

3. Edit the netmisc= and transport= entries in the [386Enh] section of your SYSTEM.INI file to remove references to netbeui.386.For example,

```
[386Enh]

netmisc=ndis.386,ndis2sup.386

transport=nwlink.386,nwnblink.386
```

3.6 Loading PC/TCP NetBIOS and InterDrive

This section describes how you can edit your AUTOEXEC.BAT file to automatically load PC/TCP NetBIOS and the PC/TCP InterDrive NFS client each time that you start your PC.The loading procedures differ based on your kernel implementation (TSR or VxD).

To load the NetBIOS TSR

1. Ensure that the AUTOEXEC.BAT file contains a line that loads your PC/TCP TSR kernel.

2. Add a line to your AUTOEXEC.BAT file to load PC/TCP NetBIOS.

   Load PC/TCP NetBIOS (NETBIOS.COM) before you start the network (net start workstation) and after you load the PC/TCP kernel.

This example loads the PC/TCP DIX Ethernet TSR kernel and the PC/TCP NetBIOS TSR (NETBIOS.COM):

```
C:\WINDOWS\NET START

PATH C:\PCTCP

SET PCTCP=C:\PCTCP\PCTCP.INI

ETHDRV

NETBIOS.COM

NET START WORKSTATION
```

If you are also using InterDrive, you must start the workstation (and log in, if you are using user security) before you load InterDrive.

To load the NetBIOS VxD

Ensure that the AUTOEXEC.BAT file contains a line specifying the VXDINIT VxD initialization program. For example

```
PATH C:\PCTCP
```

```
SET PCTCP=C:\PCTCP\PCTCP.INI

NET START

VXDINIT +n
```

Alternatively, you can use the vxdinit command without the +n option and set vnbep=yes in the [pctcp vxdinit] section of your PCTCP.INI file.

For more information on using PC/TCP NetBIOS, refer to Chapter 8, Configuring NetBIOS.

To load the InterDrive TSR

The InterDrive TSR is configured by default when you install the PC/TCP TSR kernel.

Before you proceed, you should verify that the idrive command appears after the command that loads the PC/TCP kernel in your AUTOEXEC.BAT file.

The following AUTOEXEC.BAT file fragment includes commands that set the path of the PCTCP.INI file, load the PC/TCP TSR kernel for a DIX Ethernet network, and start InterDrive:

```
PATH C:\PCTCP
SET PCTCP=C:\PCTCP\PCTCP.INI
C:\PCTCP\3C503.COM 0x60 3 0x280
C:\PCTCP\ETHDRV.EXE
C:\PCTCP\IDRIVE
```

To load the InterDrive VxD

The InterDrive VxD is configured by default when you install the PC/TCP TSR kernel.

Before you proceed, you should verify the vxdinit command appears in your AUTOEXEC.BAT file before you start Windows. This command starts the VxD kernel and VxD InterDrive (by default). For example,

```
PATH C:\PCTCP

SET PCTCP=C:\PCTCP\PCTCP.INI

VXDINIT
```

This assumes that vidrive=yes in the [pctcp vxdinit] section of your PCTCP.INI file. (This is the default.)

For more information on using the InterDrive client program, refer to Using PC/TCP in Windows   Chapter 6, Sharing Network Files.

3.7 Related Information About Installing PC/TCP with Windows for Workgroups

Refer to the following sources for more information about installing PC/TCP with Windows for Workgroups:

Topic	Source
Installing and configuri	Getting Started

ng

PC/

TC

P

ove

r an

NDI

S

driv

er.

Loa Ch

dingapt

and configuring PC/TCP NetBIOS.

Installin

er 8, Configuring NetBIOS

Docu

g	me
and	nta
conf	tion
iguri	sup
ng	plie
Win	d
dow	wit
s	h
for	you
Wor	r
kgr	dist
oup	rib

s. utio

n

of

Wi

nd

ow

s

for

Wo

rkg

rou

ps

Chapter 3 Installing PC/TCP with Windows for Workgroups

Chapter 4

Installing PC/TCP with Novell NetWare

To use PC/TCP and NetWare concurrently, you can use shared drivers to run a NetWare stack and a PC/TCP stack simultaneously on a single card. You can also use PC/TCP and NetWare concurrently using NetWare/IP.

With this chapter, you can learn how to configure to use NetWare and PC/TCP concurrently

- Over an ODI (Open Data Link Interface) Ethernet or token ring driver.
- Over an ASI (Adapter Support Interface) token ring driver.

You can also learn how to configure your system to run PC/TCP and NetWare concurrently

- Over the Internet Protocol using NetWare/IP.
- Over an Ethernet packet driver.

- Over an NDIS (Network Driver Interface Specification) driver.

4.1 Before You Install PC/TCP with Novell NetWare

Before you install PC/TCP with Novell NetWare, you or your system administrator

should

- Ensure that you meet all of the PC/TCP system requirements in Getting Started.
- Install Novell NetWare and verify that the installation is operational.

4.2 Configuring NetWare and PC/TCP over ODI Drivers

ODI is a driver specification developed by Novell to provide the coexistence of NetWare and protocols other than Novell’s proprietary IPX (such as TCP/IP and AppleTalk).

PC/TCP can communicate with NetWare 386 servers (3.x and above) using PC/TCP DIX Ethernet and IEEE 802.5 Token Ring kernels.

The ODI architecture consists of the Multiple Link Interface Driver (MLID) layer and the Link Support Layer (LSL) layers:

- The MLID handles the sending and receiving of packets to and from a physical or logical media.
- The LSL handles the communication between protocol stacks and MLIDs by determining which protocol stacks to pass a packet to, when one is received by the MLID.

PC/TCP requires an ODI-to-packet driver converter (ODIPKT.COM) to provide an interface between the Link Support Layer and the PC/TCP kernel. The ODI converter supports the DIX Ethernet, IEEE Ethernet, and IEEE 802.5 token ring kernels.

ODI drivers do not require a NetWare shell. Using ODI, you also do not need to modify the Ethernet frame type. The workstation sends DIX encapsulation from the PC/TCP side, and 802.3 packets from the Novell side.

Figure 4-1 shows how PC/TCP and NetWare can work together with ODI. The left half of the figure illustrates the general layering of ODI drivers within the PC/TCP architecture.

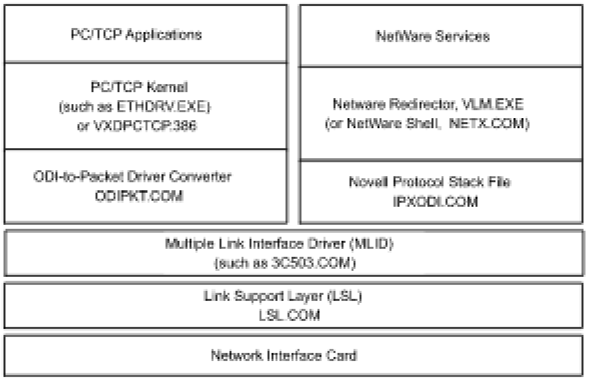


Figure 4-1 Layering of ODI Drivers with PC/TCP

4.2.1 Before You install PC/TCP over ODI Drivers

Make sure that you

1. Install NetWare and verify the installation.

2. Perform the PC/TCP installation steps described in Getting Started Chapter 1,

Installing PC/TCP Software.

Select one of these PC/TCP kernels: DIX Ethernet, IEEE 802.3 Ethernet, or IEEE 802.5 Token Ring.

4.2.2 Configuring NetWare and PC/TCP to Run over ODI Drivers — Procedure

These are the steps to run PC/TCP over ODI drivers:

1. Create or update your NET.CFG file.

   This file contains configuration information for each ODI module. The NET.CFG file may be supplied with your driver.

2. Configure the [pctcp interface n] section of your PCTCP.INI file.

3. Edit the AUTOEXEC.BAT file to load the appropriate drivers.

Reboot your system and test your configuration by communicating with a remote host on the network using the ping command.

To create the NET.CFG file

The file consists of main headings followed by the entries for the section. These are the main headings: PROTOCOL, LINK SUPPORT, and LINK DRIVER.

NET.CFG configuration file entries take the following form:

```
 link driver drivername
```

Note: Precede each configuration entry line with spaces or a tab.

This is a description of the lines in the NET.CFG file:

link driver drivername   Specifies the filename of your vendor-supplied MLID driver.

int int_vectorChanges the interrupt (IRQ)to the number you specify.

port port_#   Changes the board’s base I/O address from the default to the number you specify (in hexadecimal).

mem base_mem_addr   Specifies the base memory address for this interface.

frame type   The “frame” entry (sometimes called “envelope type”) specifies your network media type. The frame type can be any of the following:

P P Net

C/C Wa

T / re

C T fra

P C me

k P typ

er fr e

n a

el m

e

ty

p

e

E E Eth

T T ern

H H et_

D E 80

R R 2.3

V N

E

T

_I

I

IE E Eth

E T ern

E H et_

D E 80

R R 2.3

V N an

E d

T Eth

_ ern

S et_

N 80

A 2.2

P

T T Tok

O O en-

K K Rin

D E g_

R N 80

V - 2.5

R

I

N

G

–

S

N

A

P

If they are needed for other tasks, you can use both the NetWare and the PC/TCP frame type entries. Specify the NetWare frame type before the PC/TCP frame type in the NET.CFG file.

Note: Some network interface adapter cards may require additional parameters (such as, interrupt vector, port number, base memory address, or frame type). For more detailed information about the syntax of NET.CFG file entries, see the Novell NetWare technical documentation.

Here is a sample NET.CFG file for Ethernet 802.3 running over a 3Com 3C503 adapter:

```
link driver 3c503
        int 3
```

```
    port 280
    mem D8000
    frame ETHERNET_802.3
    frame ETHERNET_802.2
    frame ETHERNET_II
```

Note that ODIPKT.COM consumes two protocol stack entries in the internal LSL tables. By default, LSL supports four protocol stacks. If you need more than two protocol stacks, add a line in the following form to the LINK SUPPORT section of your NET.CFG file:

```
 max stacks int
```

where int is the number of stacks that LSL should support.

Note: If you are using a Cabletron network interface card, you must set max stacks to

6.

To configure the network interface in the PCTCP.INI file

Use a text editor to modify the [pctcp ifcust n] section of the PCTCP.INI file so that it describes your PC/TCP kernel interface. For example,

```
[pctcp ifcust 0]
ip-address= 128.127.50.1
router=128.127.50.2
subnet-mask=255.255.255.0
```

If you are using the VxD kernel, add these lines to specify the network interface and frame type. For example, these lines specify the packet driver interface using the DIX Ethernet frame type:

```
frame-type= DIX-Ethernet
```

interface-type= PKTDRV

To modify the AUTOEXEC.BAT file

Use a text editor to load the following programs in the order indicated.

The following lines load both PC/TCP DIX Ethernet kernel and NetWare over ODI on a

3Com 3c503, Ethernet network interface card:

En	To
ter	load
thi	this
s	
lin	
e	
C:	Link

\NWCLIENT\LSL.COM	Support Layer. (This file comes with the NetWar

e

distr

ibuti

on.)

C: MLI

\N D

W laye

CL r.

IE (Thi

NT s is

\3 the

C5 ODI

03. driv

C er

O that

M com

es

with

your

net

wor

k

ada

pter

card.) This program usually has the same nam e as

your

net

wor

k

inter

face

with

a .C

OM

exte

nsio

n.

C: Nov

\N ell

W prot

CL ocol

IE stac

NT k

\IP file.

XO

DI.

C

O

MC:\PCTCP\ODIPKT.COM

ODI-to-packet driver converter. You can

use command line options to further specify the bindings that ODIPKT.COM should perform.

For

exa

mpl

e,

odip

kt

[?]

[unl]

[sint

=

xx]

[mli

d=

nam

e]

[fra

me=

med

ia_t

ype]

Whe

re

media_type is one of the following: ETHER

NET

_II,

ETH

ER

NET

_SN

AP,

or

TOK

EN-

RIN

G_S

NAP.

Note:

Enter the frame media_type

in upper case. Here is an example of load ing

a

driv

er at

inter

rupt

65:

C:

\>od

ipkt

sint

=65

mlid

=3c

503

fram

e=E

THE

RN

ET_

SNA

P

For

a

brief

des

cript

ion

of

eac

h

ODI

PKT

.CO

M

opti

on enter odipkt -? at the DOS command

	line
	pro
	mpt.
C:	Net
\N	War
W	e
CL	redir
IE	ecto
NT	r.
\V	Not
LM	e:
.E	

XE You

can

alter

nati

vely

load

the

NET

X

Net

War

e

shell

file.

C: For

\P the

CT PC/

CP TCP

\E TSR

TH kern

DRel.

V.

For

EX

the

E

PC/

– TCP

or–VxD

C: .

\P This

CT ass

CP ume

\V s

XD that

INI fram

T e-

type

=ET

HD

RV

and

inter

face

-

type

=PK

TDR

V in

the

[pct

cp

ifcus

t 0]

secti

on

of

your

PCT

CP.I

NI

file.

For more information about loading the PC/TCP InterDrive NFS client, see section 4.9, Loading InterDrive.

For more information about configuring the VxD kernel, see Chapter 13, Configuring and Tuning the Kernel.

4.3 Configuring NetWare and PC/TCP over Token Ring and ASI Drivers

NetWare and PC/TCP can work together over token ring through the IBM LAN Support program and the PC/TCP token ring kernel IBMTR.EXE. The IBM LAN Support program is an implementation of the ASI (Adapter Support Interface) specification.

Like the packet driver interface, the ASI driver can identify packets from different protocol types and pass them to the appropriate protocol stack.

Figure 4-2 shows how PC/TCP and NetWare work over token ring.

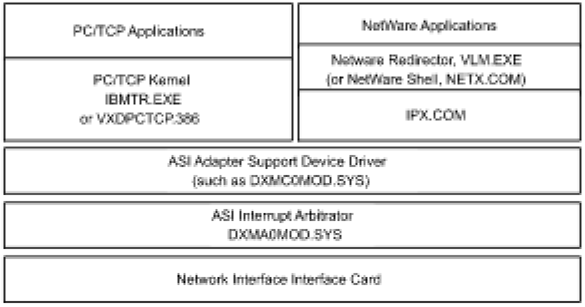


Figure 4-2 PC/TCP and NetWare over Token Ring and ASI

Follow these steps to configure PC/TCP and NetWare over an ASI driver:

1. Install the IBM LAN Support program according to instructions in Chapter 22, Troubleshooting the Kernel and Driver Configuration.

2. Install PC/TCP using the procedures in Getting Started.

3. Generate a NetWare shell.

To use NetWare and PC/TCP over the same network interface card, you must generate a new NetWare IPX shell for the packet driver interface. For more information, see section 4.2, Configuring NetWare and PC/TCP over ODI Drivers.

Note: The NetWare shell generationand linker program is called WSHGEN.EXE for NetWare 3.x and later.

4. Add statements to your CONFIG.SYS file to support PC/TCP and NetWare over token ring. The following is an example:

```
device=c:\asi\dxma0mod.sys
```

```
device=c:\asi\dxme0mod.sys

device=c:\asi\dxmt0mod.sys ES=2 O=Y
```

The ES= option makes the driver allocate more Service Access Points (SAP). Each protocol type running over a token ring board needs a SAP. By default, each driver is allocated one SAP. If you use more than one protocol over a board, set ES= to a number greater than or equal to the number of protocols used.

Before you proceed, you must verify that the shell is loaded properly, and that the IPX.COM loader is running for the ASI driver.

4.4 Configuring NetWare/IP over PC/TCP

PC/TCP provides emulation of the LAN WorkPlace TCP/IP application programming interface (API) using the PC/TCP LWPE.COM program. This TSR (terminate-and-stay-resident) program allows you to run LAN WorkPlace applications, including Novell's NetWare/IP application, over any PC/TCP TSR kernel.

The LWPE program works by intercepting calls from a LAN WorkPlace application or NetWare/IP and redirecting these calls to a PC/TCP kernel. To support NetWare/IP, LWPE.COM must be run over the ODIPKT driver. This is due to dependencies between NetWare/IP, the NetWare redirector (VLM), and the Link Support Layer (LSL).

Figure 4-3 illustrates the protocol stack when you configure PC/TCP with NetWare/IP support.

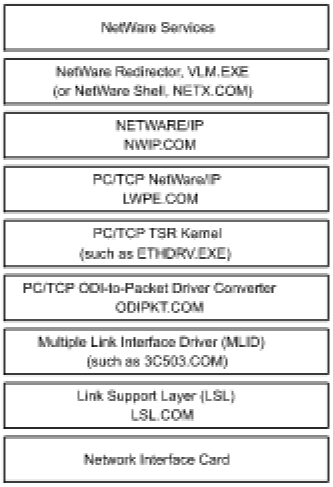


Figure 4-3 PC/TCP with NetWare/IP

As with the standalone NetWare/IP product, the NWIP.COM module will not load unless it finds both a Domain Name System (DNS) server and a Domain SAP Server (DSS) on the network.

For NetWare 3.11 and later servers, PC/TCP and NetWare can run concurrently over any network media supported by PC/TCP through an IP tunnel. The tunnel lets PC/TCP encapsulate, or carry NetWare IPX packets inside IP.

For a description of how to configure an IP tunnel with PC/TCP, consult the anonymous

FTP server ftp.ftp.com.

4.4.1 Before You Configure PC/TCP LWPE to Work over NetWare/IP

Make sure that

- You install the Novell NetWare/IP software on your system.

For more information on NetWare/IP and NetWare commands, refer to the NetWare/IP and NetWare client manuals.

- You follow the steps defined in section 4.2, Configuring NetWare and PC/TCP over ODI Drivers to install PC/TCP and Novell NetWare with ODI drivers.

4.4.2 Configuring PC/TCP LWPE to Work over NetWare/IP — Procedure

Follow these steps to configure PC/TCP LWPE to work over NetWare/IP:

1. Edit your AUTOEXEC.BAT file to load the appropriate programs in the order indicated.

Note: Comment out or delete any line that starts the NetWare/IP STARTNET.BAT file. This line was added to your AUTOEXEC.BAT file by the NetWare/IP client installation procedure.

The following lines load both PC/TCP DIX Ethernet TSR kernel (ethdrv.exe) and NetWare over ODI on a 3Com 3c503, Ethernet network interface card. You should modify these lines to specify your network interface.

EntTo

er load

thi this

s

lin

e

C: LSL

\N laye

W r.

CL

IE

NT

\LS

L.

CO

M

C: MLI

\N D

W laye

CL r.

IE This

NT pro

\3 gra

C5 m

03. usu

COally

M has the same name as your network interface

	with
	a .C
	OM
	exte
	nsio
	n.
C:	ODI
\P	-to-
CT	pac
CP	ket
\O	driv
DI	er

PK con

T.Cvert

O er.

M

C: PC/

\P TC

CT P

CP TS

\E R

TH kern

DR el.

V.

EX

E

C: PC/

\P TC

CT P

CP LW

\L PE.

W CO

PE M

.C TS

O R

M pro

gra

m

that

ena

bles

you

to

run

Nov

ell

Net

War

e IP

over

the

PC/

TC

P

stac

k.

C: Net

\N War

W e/IP

CL appl

IE icati

NT on

\N pro

WI gra

P. m.

CO

M

C: Net

\N War

W e

CL redi

IE rect

NT

\VL

M.

CO

M

or.

Not

e:

You

can

alter

nati

vely

load

the

NE

T X

Net

War

e

shel

l

file.

For more information about loading the PC/TCP InterDrive NFS client, see section 4.9,

Loading InterDrive.

2. Edit the CONFIG.SYS file to includes the following line:

```
LASTDRV=Z
```

3. Edit your PCTCP.INI file to contains a [PCTCP LWPE] section that at least identifies

the pathname of your RESOLV.CFG file (config-path=). You can customize your LWPE configuration by entering new default values for other [ptcp lwpe] parameters.

This example shows the default [pctcp lwpe] settings:

```
[pctcp lwpe]

rcbs=40

sockets=32

loadhigh=yes

config-path=C:\your_RESOLV.CFG_filename
```

4. Verify that your NET.CFG file contains an NWIP_DOMAIN entry.

5. Edit your RESOLV.CFG file to specify your name server environment.

This is a sample RESOLV.CFG file.

```
domain abc.xyz.com
```

```
nameserver 128.127.50.123

nameserver 128.127.51.789
```

For more information on the file syntax, see your NetWare/IP technical documentation.

4.5 Configuring NetWare and PC/TCP over Ethernet and a Packet Driver

PC/TCP and NetWare work together through shared drivers that let both protocols share a single interface card. One of these shared driver specifications is the packet driver.

Figure 4-4 shows the protocol stack for this configuration.

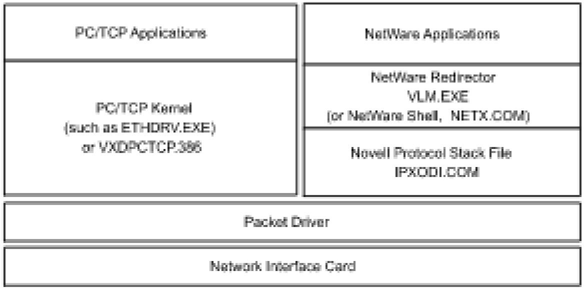


Figure 4-4 PC/TCP and NetWare over a Packet Driver

To use PC/TCP and NetWare concurrently over Ethernet, you must follow these steps:

1. Copy the packet driver software to your PC.

Tested and supported drivers reside on the “Supported Packet Driver” disk of the

PC/TCP distribution. You may also use a packet driver provided by the interface card manufacturer.

Unsupported packet drivers are available via anonymous FTP from host ftp.ftp.com.

2. Follow the instructions in Getting Started to install PC/TCP with the ETHDRV DIX Ethernet kernel.

3. Verify that your AUTOEXEC.BAT file loads these programs in this order:

Ent	To
er	loa
this	d
line	this
3C	Th
503	e

.CO	ap
M	pro
0x6	pri
0 5	ate
0x2	pac
80	ket
0xC	driv
800	er.
	For
	mo
	re
	info

rm

atio

n

ab

out

pac

ket

driv

er

opti

ons

,

	see
	its
	tec
	hni
	cal
	doc
	um
	ent
	atio
	n.
C:	Th
\N	e

WC	Net
LIE	Wa
NT\	re
PDI	pro
PX.	toc
CO	ol
M	sta
	ck
	file
	tha
	t
	you

generated specifically for the packet

	driv
	er.
C:	Th
\N	e
WC	Net
LIE	Wa
NT\	re
VL	red
M.	irec
CO	tor.
M	Not
	e:

You can alternatively load the NE

	T X
	Net
	Wa
	re
	she
	ll
	file.
C:	Th
\PC	e
TC	PC
P\E	/TC
TH	P

DR
V.E
XE
–
or–
C:
\PC
TC
P\V
XDI
NIT

TS
R
ker
nel.
For
the
PC
/TC
P
Vx
D
ker

nel. This assumes that you have set

fra

me

-

typ

e=

an

d

inte

rfa

ce-

typ

e=

in

the

[pct

cp

ifcu

st

0]

sec

tion

of

you

r

PCTCP.INI file. For more information

about configuring the PC/TCP kernel,

see Chapter 13, Configuring and Tu

ning the Kernel.

Note: You may need to modify the server or packet driver Ethernet frame type. Refer to section 4.7, Modifying the Ethernet Frame Type on the NetWare Server for instructions.

For more information about loading the PC/TCP InterDrive NFS client, see section 4.9, Loading InterDrive.

4.6 Configuring PC/TCP and NetWare over NDIS Drivers

The NDIS driver is another shared driver interface that can support multiple protocols. Version 1.15 and later of the NDIS-to-Packet Driver converter program, DIS_PKT.GUP, supports the BYU (Brigham Young University) and Intel shells so that you can use PC/TCP and Novell NetWare concurrently over NDIS drivers.

Follow these steps to configure your system to use NDIS Drivers with NetWare:

1. Perform the PC/TCP installation steps described in Getting Started Chapter 1, Installing PC/TCP Software.

2. Install and configure the NDIS driver according to instructions in Getting Started.

3. Generate a BYU shell, as described in section 4.5, Configuring NetWare and PC/TCP over Ethernet and a Packet Driver.

To use NetWare and PC/TCP over the same network interface card, you must generate a new NetWare IPX shell for the packet driver interface. You can use either an unsupported shell utility from BYU (Brigham Young University), or a supported shell utility from Intel.

You can obtain this software via anonymous FTP from host ftp.ftp.com.

Note: The NetWare shell generationand linker program is called WSHGEN.EXE for NetWare 3.x and later.

4. If you are using Ethernet, modify your CONFIG.SYS file to change the Ethernet packet type so the NDIS driver will accept packets from the NetWare server.

Version 1.30 of the DIS_PKT.GUP converter program provides an /N option that lets you receive NetWare frames. Add the following line to your CONFIG.SYS file:

```
device=C:\pctcp\dis_pkt.gup /N
```

5. Add commands to your AUTOEXEC.BAT file to automatically load both PC/TCP

and NetWare over an NDIS driver when you boot your PC.

Here is a sample set of commands:

```
    NETBIND.EXE
    IPX.COM
    VLM.COM
    ETHDRV.EXE
```

For more information about loading the PC/TCP InterDrive NFS client, see section 4.9, Loading InterDrive.

4.7 Modifying the Ethernet Frame Type on the NetWare Server

The Ethernet frame type (also referred to as a “packet type”) causes incompatibility between PC/TCP and NetWare. Novell uses IEEE 802.3 Ethernet, but without standard 802.2 headers. PC/TCP uses DIX Ethernet as defined in RFC 894. As shown in Figure 4-5, these two frame types do not carry the same information. For the two systems to communicate, both must recognize the same frame type.You can correct this on the client side or on the server side.

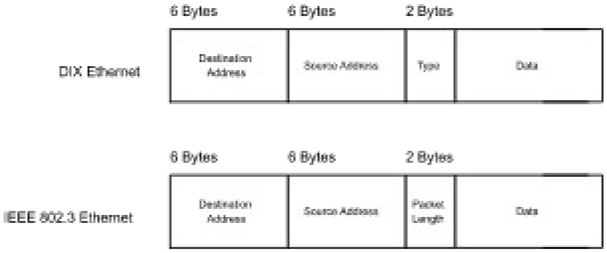


Figure 4-5 Comparison of DIX and 802.3 Ethernet

4.7.1 Modifying the Frame Type on a NetWare Client

From the client side, you can configure the Crynwr packet driver to receive both Novell 802.3 Ethernet packets and DIX packets. Starting with Version 7.1, the Crynwr packet drivers have an -n switch that lets you use the driver without having to change the frame type on NetWare server. If you load the packet driver with the -n option, the driver accepts standard Novell 802.3 Ethernet packets when communicating with workstations running NetWare, or DIX type 8137 packets when communicating with TCP/IP.

4.7.2 Modifying the Frame Type through a NetWare 386 Server

This procedure supports NetWare versions 3.1x and later.

To use DIX 8137 type frames with a NetWare 386 server, modify the frame type when you load the LAN driver. Follow these steps:

1. At the Novell server prompt, enter the load command, specifying the .LAN file for a network interface card installed on the server.

2. Select the appropriate frame type from the list of available frame types displayed by the server software.

3. Bind the LAN drivers to a communication protocol after you modify the frame type. Use the bind command at the NetWare prompt.

This supports a NetWare environment in which all workstations run both PC/TCP and NetWare 386, using both IP and IPX packets. If you also have workstations on the network that run NetWare 386 and not PC/TCP, you must configure an additional LAN driver with an IEEE 802.3 frame type. Load the same LAN driver again, this time selecting Ethernet_802.3 as the frame type.

4.8 Using PC/TCP Network Drivers with Other Network Operating Systems

The PC/TCP Network Driver (PCTCPNET.DRV) allows you to access remote files and printing services through the Windows File Manager, Control Panel, and Print Manager. If you use other network operating systems concurrently with PC/TCP software, these systems must be connected to enable access to each system’s services. The configuration of these network operating systems is referred to as “chaining” one network to or from another.

The PC/TCP installation program detects other network operating systems on your system and makes the appropriate changes to the [BOOT] section of your SYSTEM.INI file, as outlined in the following table:

Net wor	SY ST

k	E
Op	M.
erat	INI
ing	en
Sys	tri
tem	es
(s)	
PC/	[B
TC	O
P	O
onl	T]
y	ne

tw

or

k.d

rv

=p

ctc

pn

et.

dr

v

PC/ [B

TC O

P
and
ano
ther
net
wor
k
ope
rati
ng
syst
em

O
T]
ne
tw
or
k.d
rv
=p
ctc
pn
et.
dr

(su	v
ch	pct
as	cp
Nov	ch
ell	ain
Net	=n
War	et
e)	wa
	re.
	dr
	v
PC/	[B

TCP, Windows for Workgroups, and ano

OOT] network.drv=wfwnet.d

ther network operating system (such as	rvsecondnet.drv=pctcpnet.dr

Novv

ell pct

Net cp

Warch

e) ain

=n

et

wa

re.

dr

v

Note: You should verify that the [386EnH] section does not contain entries for

WFWFTP.386 or VPCTCP.386.Remove (or comment out) these entries if they appear.

4.9 Loading InterDrive

This section describes how you can edit your AUTOEXEC.BAT file to automatically load the PC/TCP InterDrive NFS (Network File System) client. The loading procedures differ based on your kernel implementation (TSR or VxD).

To load the InterDrive TSR

The InterDrive TSR is configured by default when you install the PC/TCP TSR kernel. Before you proceed, you should verify that the idrive command appears after the command that loads the PC/TCP TSR kernel in your AUTOEXEC.BAT file.

The following AUTOEXEC.BAT file fragment includes commands that set the path of the PCTCP.INI file, load the PC/TCP TSR kernel for a DIX Ethernet network, and start InterDrive:

```
PATH C:\PCTCP
```

```
SET PCTCP=C:\PCTCP\PCTCP.INI

C:\PCTCP\3C503.COM 0x60 3 0x280

C:\PCTCP\ETHDRV.EXE

C:\PCTCP\IDRIVE
```

To load the InterDrive VxD

The InterDrive VxD is configured by default when you install the PC/TCP TSR kernel.

Before you proceed, verify that the vxdinit command appears in your AUTOEXEC.BAT file before you start Windows. This command starts the VxD kernel and VxD Interdrive (by default). For example,

```
PATH C:\PCTCP

SET PCTCP=C:\PCTCP\PCTCP.INI

C:\PCTCP\VXDINIT
```

For more information on using the InterDrive client program, refer to Using PC/TCP in

Windows Chapter 6, Sharing Network Files.

4.10 Related Information About Installing PC/TCP with NetWare

For more information about configuring PC/TCP with Novell NetWare, refer to the following sources:

Topic	Source
Installing PC/TCP	Getting Started

with a Packet Driver. Installing and configurin

Documentatio

ng NetWare.	supplied with your distribution of

	No
	vell
	Net
	Wa
	re
Con	Net
figu	Wa
ring	re
Eth	38
ern	6
et	Sy
fra	ste

metypes.

msAdministration Guide

Chapter 4 Installing PC/TCP with Novell NetWare

4.7.1 Modifying the Frame Type on a NetWare Client

4.7.2 Modifying the Frame Type through a NetWare 386 Server

4.8 Using PC/TCP Network Drivers with Other Network Operating Systems

4.9 Loading InterDrive

4.10 Related Information About Installing PC/TCP with NetWare

Chapter 5

Installing PC/TCP with LAN Manager

LAN Manager is a network operating system used by many network products such as 3Com 3+Open, AT&T StarGROUP, the Hewlett-Packard LAN Manager, and others. You can use PC/TCP together with any of these LAN Manager products.

The PC-210 kernel lets you use both LAN Manager and PC/TCP over Network Driver Interface Specification (NDIS) drivers. This lets PC/TCP and a LAN Manager product work together over network topologies based on IEEE 802.3 Ethernet, DIX Ethernet, and 802.5 Token Ring.

PC/TCP is compatible with LAN Manager with NDIS3 (and NDIS2) drivers.

PC/TCP provides TCP transport and RFC NETBIOS transport in both TSR (terminate-and-stay-resident) and VxD (Windows virtual device driver) implementations.

To configure LAN Manager with PC/TCP, you must first configure LAN Manager and

PC/TCP to run over NDIS drivers.

If you need to support NETBIOS services over a router, follow the additional configuration instructions in section 3.5, Configuring Windows for Workgroups to Use PC/TCP NetBIOS.

Using this chapter, you can learn the following:

- How to configure LAN Manager (and NETBEUI) over NDIS drivers without PC/TCP NetBIOS.

This configuration supports NETBIOS services using LAN Manager NETBEUI only. It does not use PC/TCP NetBIOS. Use this configuration with systems that do not need to support NETBIOS services across a router (that is, if all NETBIOS traffic is on a single LAN).

- How to configure LAN Manager with PC/TCP NetBIOS.

These configurations provide RFC NETBIOS functionality over a router.

You can configure your system to use NETBEUI and PC/TCP NetBIOS (TCP/IP) in parallel, or you can configure LAN Manager to use PC/TCP NetBIOS instead of NETBEUI.

To verify your installation, refer to Getting Started   Chapter 1, Installing PC/TCP Software.

5.1 Before Configuring PC/TCP with LAN Manager

Verify LAN Manager connectivity over NDIS drivers before you install PC/TCP.

This chapter assumes that you are configuring your system with NDIS2 drivers.

Note: You do not need to reinstall LAN Manager if it is already configured on your system.

For more information about how to install LAN Manager over NDIS drivers, see the LAN Manager technical documentation.

5.2 Configuring PC/TCP with LAN Manager over an NDIS Driver

With this configuration, the LAN Manager stack sends NETBEUI packets and the PC/TCP stack sends IP packets. Figure 5-1 illustrates the protocol layers using PC/TCP with LAN Manager over an NDIS2 driver.

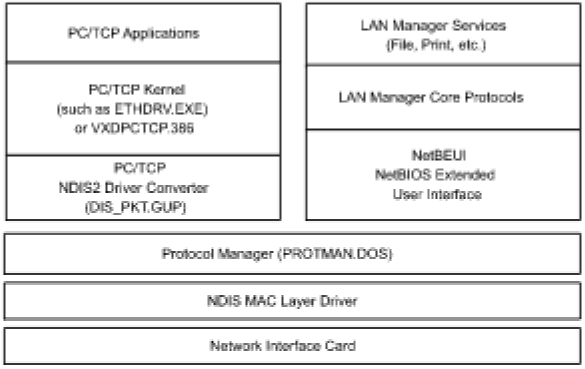


Figure 5-1 Using PC/TCP and a LAN Manager over an NDIS Driver

Note: This chapter assumes that you are configuring your system with NDIS2 drivers. When you configure PC/TCP to use NDIS2 drivers, you must load the DIS_PKT.GUP packet-driver-converter after you load the packet driver and before you load the PC/TCP kernel.

PC/TCP and LAN Manager installation programs both modify these system initialization files:

- AUTOEXEC.BAT
- CONFIG.SYS
- PROTOCOL.INI

If you are using LAN Manager in Windows, you should also review your SYSTEM.INI file.

Before you proceed, you should verify that each of these files is correct.

To verify your configuration, make sure that your kernel is loaded and that you have established connectivity by using the ping command. For more information, refer to Getting Started   Chapter 1, Installing PC/TCP Software.

To review the AUTOEXEC.BAT file

Examine your AUTOEXEC.BAT file to verify that the PC/TCP kernel loads before you start the network (net start workstation).

This example illustrates using the PC/TCP Ethernet TSR kernel (ETHDRV.EXE):

```
c:\windows;c:\;c:\dos;c:\pctcp
set pctcp=c:\pctcp\pctcp.ini
ethdrv
rem ==== lanman 2.2 === do not modify between these lines === lanman 2.2 ===
set path=c:\lanman.dos\netprog;%path%
load netbeui
net start workstation
rem ==== lanman 2.2 == do not modify between these lines === lanman 2.2 ===
```

To load the PC/TCP VxD kernel, replace ethdrv in the preceding example with vxdinit

```
c:\windows;c:\;c:\dos;c:\pctcp

set pctcp=c:\pctcp\pctcp.ini

vxdinit

rem ==== lanman 2.2 === do not modify between these lines === lanman 2.2 ===

set path=c:\lanman.dos\netprog;%path%

load netbeui

net start workstation

rem ==== lanman 2.2 == do not modify between these lines === lanman 2.2 ===
```

Note: This example does not load PC/TCP NetBIOS. It assumes that vnebp=no in the [pctcp vxdinit] section of your PCTCP.INI file. (This is the default.)

For detailed procedures for loading PC/TCP NetBIOS and the PC/TCP Interdrive NFS client, see section 5.4, Loading InterDrive.

To review the CONFIG.SYS file

Examine your CONFIG.SYS file to verify that these device statements appear in the proper sequence.

In this configuration, you load LAN Manager over NETBEUI, and you do not need to run the NDIS driver program NETBIND. Loading NETBEUI performs the NETBIND operation automatically.

This is a sample of relevant entries in your CONFIG.SYS file:

```
DEVICE=C:\LANMAN\PROTMAN.DOS /I:C:\LANMAN

DEVICE=C:\LANMAN\SMARTND.DOS

DEVICE=C:\PCTCP\DIS_PKT.GUP
```

If you are using NDIS3, drivers delete the DIS_PKT.GUP device statement.

To review the PROTOCOL.INI file

Examine the PROTOCOLS section of the PROTOCOL.INI file and verify that it includes the appropriate driver section (such as [PKTDRV]) with the appropriate network driver and binding. For example,

```
[PKTDRV]
drivername=PKTDRV$
BINDINGS=MS$ELNKII
intvec=0X60
chainvec=0x65
```

Where BINDINGS= identifies the name of the LAN Manager driver section in this file.

Note: BINDINGS must be capitalized, and there should be no spaces around the equals sign.

To review your SYSTEM.INI file

If you are using LAN Manager and PC/TCP in Windows, you should verify that your SYSTEM.INI file contains these entries:

This section	Has this entry
[386Enh]	TimerCriticalSection=5000
[boot]	network.drv=lanman22.drv

Note: The PC/TCP installation program replaces the existing network driver with its own network driver. If you want to make connections with LAN Manager, you must specify the LAN Manager driver instead. Use a semicolon to comment out any other networkdrv= statements in the file.

5.2.1 Using PC/TCP Network Drivers with Other Network Operating Systems

The PC/TCP Network Driver (PCTCPNET.DRV) allows you to access remote files and printing services through the Windows File Manager, Control Panel, and Print Manager. If you use other network operating systems concurrently with PC/TCP software, these systems must be connected to enable access to each system’s services. The configuration of these network operating systems is referred to as “chaining” one network to or from another.

The PC/TCP installation program detects other network operating systems on your system and makes the appropriate changes to the [BOOT] section of your SYSTEM.INI file, as outlined in the following table:

Net	SY
wor	ST

k Operating System(s) PC/TCP only

EM.INI entries [BOOT] ne

tw

or

k.d

rv

=p

ctc

pn

et.

dr

v

PC/ [B

TC O

P
and
ano
ther
net
wor
k
ope
rati
ng
syst
em

O
T]
ne
tw
or
k.d
rv
=p
ctc
pn
et.
dr

(su	v
ch	pct
as	cp
LA	ch
N	ain
Ma	=l
nag	an
er)	m
	an
	21
	.dr
	v

Note: You should verify that the [386EnH] section does not contain entries for WFWFTP.386 or VPCTCP.386.Remove (or comment out) these entries if they appear.

5.3 Configuring PC/TCP and LAN Manager to Use PC/TCP NetBIOS

Configuring LAN Manager with PC/TCP NetBIOS gives you RFC NETBIOS functionality over routers.

PC/TCP NetBIOS can communicate only with other systems that also have versions of NetBIOS that comply with RFC 1001 and RFC 1002.

Note: The PC/TCP NetBIOS VxD cannot run in parallel with the NETBEUI TSR. To configure NetBIOS and NETBEUI in parallel, you must configure the PC/TCP TSR kernel.

You can configure LAN Manager to use PC/TCP NetBIOS in two ways:

- With PC/TCP NetBIOS and LAN Manager NETBEUI in parallel.

Use this configuration to support a non-homogeneous NETBIOS environment. For example, if you need to interoperate with NETBEUI-compatible applications.

- With PC/TCP NetBIOS instead of Windows for Workgroups NETBEUI.

This configuration is most efficient if you do not need to interoperate with NETBEUI-compatible applications. Using this configuration, you can reclaim memory that was reserved for NETBEUI.

Note: In general, the default PC/TCP NetBIOS configuration is adequate for most operating environments. If necessary, you can customize PC/TCP NetBIOS by editing the [pctcp netbios] section of the PC/TCP configuration file (PCTCP.INI). For more information, see Chapter 8, Configuring NetBIOS.

After you are satisfied that these files are correct, you can configure PC/TCP NetBIOS and boot your system.

5.3.1 Configuring Your System to Use PC/TCP NetBIOS and NETBEUI in Parallel

You can use LAN Manager and PC/TCP simultaneously through NetBIOS. The NetBIOS application program interface (API) is standard with PC/TCP. It provides a session layer interface that lets a LAN Manager product communicate with the PC/TCP kernel.

Before you begin, follow the steps in section 5.2, Configuring PC/TCP with LAN Manager over an NDIS Driver.

Figure 5-2 illustrates the protocol stack when you configure LAN Manager (NETBEUI) and PC/TCP NetBIOS in parallel. Using this configuration, you can support a non-homogeneous NETBIOS environment. LAN Manager sends/receives NETBIOS traffic via

- RFC NETBIOS (using PC/TCP NetBIOS), and

- NETBEUI (using LAN Manager).

Figure 5-2 LAN Manager (NETBEUI) and PC/TCP NetBIOS in Parallel

Follow these steps to edit your system initialization files to use PC/TCP NetBIOS with Windows for Workgroups(NETBEUI) in parallel:

1. Add a line to your AUTOEXEC.BAT file to load PC/TCP NetBIOS.

Load PC/TCP NetBIOS before you start the network (net start workstation) and after you load the PC/TCP kernel.

Do not attempt to load NetBIOS within a DOS Windows session.Load NetBIOS before you load Windows.

If you are using InterDrive, you must start the workstation (and log in, if you are using user security) before you load InterDrive. For more information about loading InterDrive, see Using PC/TCP in DOS Chapter 7, Sharing Network Files with InterDrive.

This example loads the PC/TCP DIX Ethernet TSR kernel and the PC/TCP NetBIOS TSR (NETBIOS.COM):

```
C:\WINDOWS\NET START
C:\PCTCP\ETHDRV
C:\PCTCP\NETBIOS.COM
C:\WINDOWS\NET START WORKSTATION
```

2. Revise entries in your PROTOCOL.INI file to use PC/TCP NetBIOS.

In the [network.setup] section, change the lana0 entry to lana n, where n is the next available adapter number. For example,

```
[network.setup]

lana2=ms$elnk3,1,ms$netbeui
```

In the [ms$netbeui] section, change LANABASE=0 to LANABASE= n, where n is the same number as the adapter number specified in the [network.setup] section. For example,

```
[ms$netbeui]

lanabase=2
```

3. If you are using a version of LAN Manager prior to 2.1, edit the net1= parameter in your LANMAN.INI file to specify PC/TCP NetBIOS:

```
net1=netbios$, 0
```

4. Revise entries in your PCTCP.INI file to configure PC/TCP NetBIOS.

If you need to support NETBIOS access over routers, you must configure PC/TCP

NetBIOS to specify a name file and a broadcast file (using the namefile= and broadcastfile= parameters in the [pctcp netbios] section of your PCTCP.INI file). For example,

```
[netbios]
namefile=c:\pctcp\namefile.txt
broadcastfile=c:\pctcp\broadcast.txt
```

Where namefile.txt and broadcast.txt contain entries that identify NETBIOS hosts beyond your local LAN.

For more information about customizing your NetBIOS configuration, see Chapter 8, Configuring NetBIOS.

Note: When you configure your system to use PC/TCP NetBIOS, you may need to configure the kernel with additional TCP connections. As a general rule, one NetBIOS session requires one TCP connection. For a detailed procedure for configuring the kernel with additional TCP connections, see section 13.3.3, Adjusting the Number of TCP and UDP Connections.

5.3.2 Configuring Your System to Use PC/TCP NetBIOS (instead of NETBEUI)

This configuration lets you use PC/TCP NetBIOS instead of LAN Manager (NETBEUI).

Figure 5-3 illustrates the layering of the software stack.

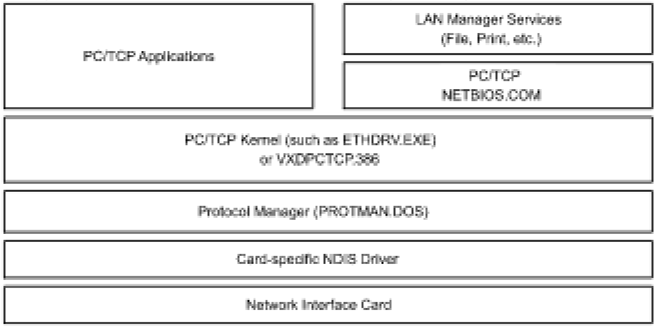

PC/TCP Applications
LAN Manager Services (File, Print, etc.)
PC/TCP NETBIOS.COM
PC/TCP Kernel (such as ETHDRV.EXE) or VXDPCTCP.386
Protocol Manager (PROTMAN.DOS)
Card-specific NDIS Driver
Network Interface Card

Figure 5-3 LAN Manager with PC/TCP NetBIOS (only)

Follow these steps to configure your system to use PC/TCP NetBIOS (instead of NETBEUI):

1. Follow the steps for configuring your system with Windows for Workgroups over NDIS drivers. For more information, see section 5.2, Configuring PC/TCP with LAN Manager over an NDIS Driver.

2. Follow the steps for configuring your system to use PC/TCP NetBIOS. For more information, see section 5.3.1, Configuring Your System to Use PC/TCP NetBIOS and NETBEUI in Parallel.

3. Edit your system initialization files to remove references to NETBEUI, and to replace it with PC/TCP NetBIOS.

To edit the AUTOEXEC.BAT file

Examine your AUTOEXEC.BAT file and, if necessary, modify the file to remove lines that load LAN Manager’s reference NETBEUI.

This sample AUTOEXEC.BAT file illustrates using PC/TCP NetBIOS with the Ethernet TSR kernel (ETHDRV):

```
set pctcp=c:\pctcp\pctcp.ini
ethdrv
```

```
netbios.com

rem === lanman 2.2 === do not modify between these lines === lanman 2.2 ====

set path=c:\lanman.dos\netprog;%path%

rem load netbeui

net start workstation

rem === lanman 2.2 === do not modify between these lines === lanman 2.2 ====
```

To edit the PROTOCOL.INI file

Revise entries in your PROTOCOL.INI file to remove references to NETBEUI.

1. In the [network.setup] section, remove ms$netbeui from the lana n = entry. For example,

```
[network.setup]

; lana2=ms$elnk3,1,ms$netbeui
```

```
lana2=ms$elnk3,1
```

2. Remove the [ms$netbeui] section.

To edit the LANMAN.INI file

You can use the adapter= parameter in the PCTCP.INI file to configure NetBIOS to use the alternative 0x2A interrupt interface (in addition to the 0x5C default interrupt interface). This option lets you remove MINSES.EXE from the LAN Manager network services list (thus saving approximately 2K of conventional memory).

To remove MINSES.EXE from your LAN Manager network services list, edit your LANMAN.INI file to delete minses from the netservices= entry.

To edit the SYSTEM.INI file

If you are using LAN Manager in Windows, edit the netmisc= and transport= parameters in the [386Enh] section to remove references to NETBEUI.

5.4 Loading InterDrive

The InterDrive TSR is configured by default when you install the PC/TCP TSR kernel.

Before you proceed, you should verify that the idrive command appears after the command that loads the PC/TCP kernel in your AUTOEXEC.BAT file. The loading procedures differ based on your kernel implementation (TSR or VxD).

The following AUTOEXEC.BAT file fragment includes commands that set the path of the PCTCP.INI file, load the PC/TCP TSR kernel for a DIX Ethernet network, and start InterDrive:

```
PATH C:\PCTCP

SET PCTCP=C:\PCTCP\PCTCP.INI

C:\PCTCP\3C503.COM 0x60 3 0x280

C:\PCTCP\ETHDRV.EXE
```

```
C:\PCTCP\IDRIVE
```

To load the InterDrive VxD

The InterDrive VxD is configured by default when you install the PC/TCP TSR kernel.

Before you proceed, you should verify the vxdinit command appears in your AUTOEXEC.BAT file before you start Windows. This command starts the VxD kernel and VxD InterDrive (by default). For example,

```
PATH C:\PCTCP
SET PCTCP=C:\PCTCP\PCTCP.INI
VXDINIT +n +i
```

Alternatively, you can use the vxdinit command without command line options, and set vnbep=yes, and vidrive=yes (the default) in the [pctcp vxdinit] section of your PCTCP.INI configuration file.

5.5 Related Information About Installing PC/TCP with LAN Manager

For more information about installing PC/TCP with LAN Manager, see the following sources:

Topic	Source
Installing LAN Ma	Documentation sup

nager.

plied with your distribution of LA

Setting NetBIOS configuration	NManager Chapter 8, Configuring

n Net

opti BI

ons OS

in

the

PC

TCP

.INI

conf

igur

atio

n

file.	
Usi	Usi
ng	ng
the	PC
Inte	/TC
rDri	P
ve	in
clie	Wi
nt	nd
pro	ow
gra	s
m.	Ch

apter 6, Sharing Network Files

Chapter 5 Installing PC/TCP with LAN Manager

Chapter 6

Installing PC/TCP with Banyan VINES

VINES is the network operating system developed by Banyan Systems, Inc. FTP Software, Inc. and Banyan have cooperated to produce versions of PC/TCP and VINES that are completely compatible.

PC/TCP requires Banyan VINES version 2.10 or greater.

Using the procedures in this chapter, you can configure your system to

- Use the PC/TCP EBANYAN kernel to run over Banyan Ethernet.
- Use the PC/TCP IBANYAN kernel to send encapsulated IP packets within VINES transport services.
- Use the PC/TCP ETHDRV or IEEEDRV kernel and VINES with an NDIS (Network Driver Interface Specification) driver.
- Use the PC/TCP IBMTR kernel and VINES over Token Ring with an ASI (Adapter

Support Interface) driver.

With any of these schemes, you can alternate between VINES services and PC/TCP applications without rebooting your PC.

Use the procedures in this chapter to verify your configuration.

6.1 Before You Install PC/TCP with Banyan VINES

Before you install PC/TCP with VINES, you or your system administrator should ensure that

- Banyan VINES software version 2.10 or greater is loaded on both the PC and the file server.
- A network interface card and drivers supported by the VINES are installed.
- The Banyan ban command has been run on this PC to initialize the interface with the VINES driver. Refer to the documentation provided by Banyan Systems.
- You follow the steps in Getting Started for installing PC/TCP. The installation program automatically detects your network drivers and edits your initialization files appropriately.

6.2 Installing PC/TCP with Banyan Ethernet

FTP Software and Banyan support a PC/TCP kernel (EBANYAN) that lets VINES and PC/TCP share a single Ethernet network interface card. With this configuration you can use PC/TCP to access TCP/IP services and VINES to access VINES services.

For this configuration, you must have Banyan VINES (version 2.10 or later) loaded on both the PC and the file server.

Figure 6-1 illustrates the general layering of a Banyan Ethernet driver within the PC/TCP architecture.

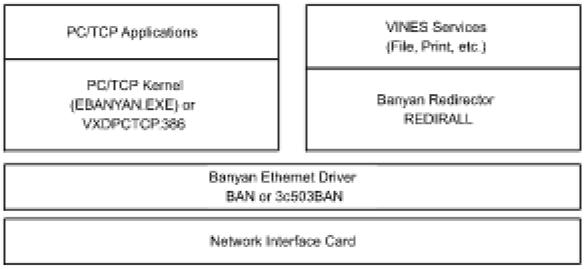

PC/TCP Applications
VINES Services (File, Print, etc.)
PC/TCP Kernel (EBANYAN.EXE) or VXDPCTCP.386
Banyan Redirector REDIRALL
Banyan Ethernet Driver BAN or 3c503BAN
Network Interface Card

Figure 6-1 Layering of a Banyan Ethernet Driver with PC/TCP

Follow these steps to verify your system configuration:

This procedure shows how to configure your system to run the PC/TCP kernel to run Banyan VINES over Ethernet.

1. Examine the AUTOEXEC.BAT file to ensure that it loads the appropriate Banyan VINES drivers and the PC/TCP kernel.

   This example shows how to load the EBANYAN.EXE PC/TCP TSR kernel:

```
set path=c:\pctcp;%path%

set pctcp=c:\pctcp\pctcp.ini

cd \drivers\banyan

ban

cd \

ebanyan
```

   To instruct Windows to load the PC/TCP VxD kernel instead, replace the ebanyan statement with

```
vxdinit
```

This loads the VxD kernel when you start Windows.

Note: This assumes that you have set the appropriate network frame-type and interface-type in the [pctcp ifcust 0] section of your PCTCP.INI file.

2. Examine your PCTCP.INI configuration file to ensure that it includes entries describing your kernel and network interface.

Verify that the [pctcp ifcust 0] section specifies the appropriate IP addresses and subnet mask. For example,

```
[pctcp ifcust 0]

ip-address = 128.127.50.100

subnet-mask = 255.255.255.0
```

```
router = 128.127.50.6
```

To configure the PC/TCP VxD kernel you must also specify the appropriate network frame-type and interface-type.

```
frame-type=DIX-Ethernet

interface-type=EBANYAN
```

Verify that the [pctcp kernel] section identifies the appropriate network interface section inthe PCTCP.INI file. For example,

```
[pctcp kernel]

interface=ifcust 0
```

The CONFIG.SYS file is not modified for this configuration.

6.3 Tunneling TCP/IP Packets within VINES Transport

The IBANYAN kernel developed by FTP Software and Banyan Systems encapsulates PC/TCP network packets within VINES network packets. This product lets PC/TCP and VINES run concurrently on any media supported by VINES (such as DIX Ethernet or IEEE 802.5 Token Ring).

Encapsulated PC/TCP is only available from Banyan Systems under a server-based license. For information about installing this software, refer to your Banyan Systems documentation. For this installation, you must have Banyan VINES (version 3.0 or later) loaded on both the PC and the file server.

- The IBANYAN kernel encapsulates TCP/IP packets within VINES packets and transports them between the network and the PC/TCP applications.
- The Banyan redirector passes the IP packets over the VINES network to the VINES

server specified in the PCTCP.INI file. The VINES server then routes the TCP/IP packets to their final destination. It also allows communication between the PC/TCP stack and the VINES services.

Figure 6-2 illustrates the protocol stack with a Banyan driver and the IBANYAN kernel.

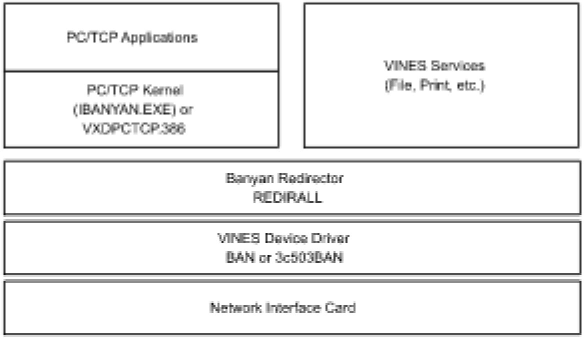


Figure 6-2 PC/TCP Running over a VINES Redirector

This procedure describes how to configure your system with the PC/TCP kernel running over a BANYAN VINES redirector.

1. Examine the AUTOEXEC.BAT file to ensure that it loads the appropriate Banyan VINES drivers and the PC/TCP kernel:

This example shows how to load the IBANYAN TSR kernel:

```
set path=c:\pctcp;%path%

set pctcp=c:\pctcp\pctcp.ini

cd \drivers\banyan

ban

cd \

ibanyan
```

To instruct Windows to load the PC/TCP VxD kernel instead, replace the ibanyan statement with

```
vxdinit
```

This loads the VxD kernel when you start Windows.

Note: This assumes that you have set the appropriate network frame-type and interface-type in the [pctcp ifcust 0] section of your PCTCP.INI file.

2. Examine your PCTCP.INI configuration file to ensure that it includes entries describing your kernel and network interface.

Add these statements	To specify this
[pctcp	The

ibanyan] bserver=myserver [pctcp	name of the Banyan server. The

ifcu	IP
st	ad
0]	dre
	ss,
	su
ip-	bn
add	et-
res	ma
s =	sk,
128	an
.12	d
7.5	rou

0.100 subnet-mask = 255.255.255.0

ter address. For more information

router =128.127.50.6 frame-type=ibanyan

about configuring your IP and rou

interface-type=ibanyan	ter addresses, see the Banyan TC

P/IP Guide. The network frame

an

d

int

erf

ac

e

typ

es

(if

yo

u

are

using the PC/TCP VxD).

[pctcpkernel] The PC TC

interface=ifcust 0

P.INI file section that describes

this kernel interface.

6.4 Using an NDIS Ethernet Driver with PC/TCP

One way to use PC/TCP and VINES together is with an NDIS driver. The NDIS driver provides a common interface that both PC/TCP and VINES can use over a single network card. With this scheme, data does not have to go through a VINES server or router to access other TCP/IP hosts. Your PC transmits TCP/IP packets with no VINES encapsulation.

This installation requires Banyan VINES (version 4.1 or later) loaded on both the PC and the file server.

PC/TCP is compatible with NDIS3 and NDIS2 drivers. NDIS drivers are available from your interface card supplier. In addition, FTP Software, Inc. has a repository of NDIS drivers available via anonymous FTP from the host ftp.ftp.com in the subdirectory /NDIS.

Before you begin, you should refer to the VINES technical documentation to configure the ban driver to use NDIS drivers.

Figure 6-3 illustrates the protocol stack when you configure the NDIS driver with VINES and PC/TCP.

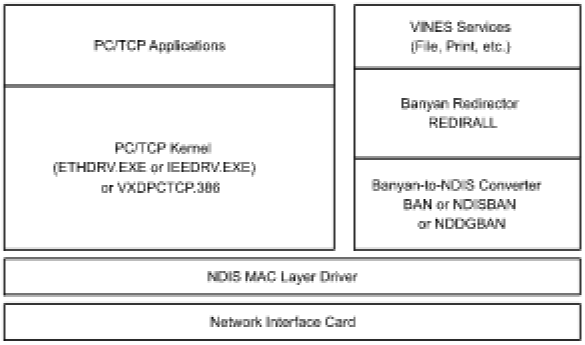


Figure 6-3   PC/TCP and VINES with an NDIS Driver

Follow these steps to configure the DIX Ethernet kernel and NDIS drivers:

1. Examine the AUTOEXEC.BAT file to ensure that it loads the appropriate Banyan VINES drivers and the PC/TCP kernel.

This example shows how to load the DIX Ethernet PC/TCP TSR kernel:

```
set path=c:\pctcp

set PCTCP=c:\pctcp\pctcp.ini

cd \drivers\banyan

ban

cd \

ethdrv
```

Note: You do not need the NETBIND program. The VINES ban command performs the same function as NETBIND.

To instruct Windows to load the PC/TCP VxD kernel instead, replace the ethdrv statement with vxdinit.

This loads the VxD kernel when you start Windows.

Note: This assumes that you have set the appropriate network frame-type and interface-type in the [pctcp ifcust 0] section of your PCTCP.INI file.

2. Examine your PCTCP.INI configuration file to ensure that it includes entries describing your kernel and network interface.

Verify that the [pctcp ifcust 0] section specifies the appropriate IP addresses and subnet mask. For example,

```
[pctcp ifcust 0]
ip-address = 128.127.50.100
subnet-mask = 255.255.255.0
router = 128.127.50.6
```

To configure the PC/TCP VxD kernel, you must also specify the appropriate network frame-type (IEEE, DIX-Ethernet, or Token-Ring) and interface-type(PKTDRV).

```
frame-type=IEEE
```

```
interface-type=PKTDRV
```

Verify that the[pctcp kernel] section identifies the appropriate network interface section in the PCTCP.INI file. For example,

```
[pctcp kernel]

interface=ifcust 0
```

Note: The installation procedure does not modify your CONFIG.SYS or PROTOCOL.INI files.

6.5 Using VINES with the PC/TCP Token Ring Kernel

Banyan Systems makes a version of VINES that runs over the implementation of the ASI specification for Token Ring. You can use this with the FTP Software IBMTR 802.5 kernel to let PC/TCP and VINES run simultaneously over Token Ring.

- The IBM LAN Support program ASI provides an interrupt arbitrator and a device driver to provide communication between the IBMTR kernel and the Token Ring network card.
- The IBMTR kernel is designed to run over the IBM LAN Support program to allow communication between the PC/TCP applications and connecting hosts on the Token Ring network.

Before you begin, you should refer to the VINES technical documentation to configure the BAN driver to use TOKUIBAN.

Figure 6-4 illustrates how PC/TCP and VINES coexist with the ASI drivers and Token Ring.

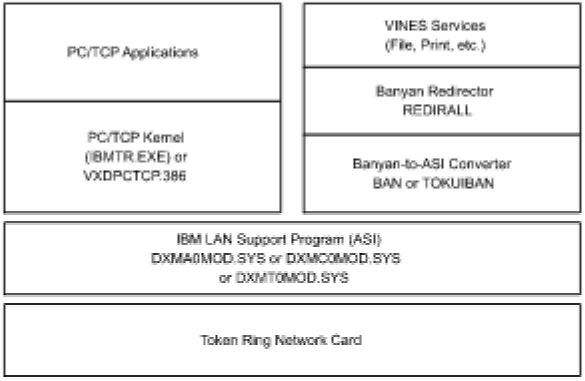

PC/TCP Applications
PC/TCP Kernel (IBMTR.EXE) or VXDPCTCP.386
VINES Services (File, Print, etc.)
Banyan Redirector REDIRALL
Banyan-to-ASI Converter BAN or TOKUIBAN
IBM LAN Support Program (ASI) DXMA0MOD.SYS or DXMC0MOD.SYS or DXMT0MOD.SYS
Token Ring Network Card

Figure 6-4 VINES over an ASI Driver

Follow these steps to configure the IBM Token Ring (IBMTR) kernel:

1. Examine the AUTOEXEC.BAT file to ensure that it loads the appropriate drivers and the PC/TCP kernel.

This example shows how to load the IBM Token Ring PC/TCP TSR kernel:

```
set path=c:\pctcp

set PCTCP=c:\pctcp\pctcp.ini
```

```
cd \drivers\banyan

ban

cd \

ibmtr
```

To instruct Windows to load the PC/TCP VxD kernel instead, replace the ibmtr statement with

```
vxdinit
```

This loads the VxD kernel when you start Windows.

Note: This assumes that you have set the appropriate network frame-type and interface-type in the [pctcp ifcust 0] section of your PCTCP.INI file.

2. Examine the PCTCP.INI file to ensure that it configures the kernel and the IBMTR kernel interface.

Add these statements	To specify this
[pctcp kernel] serial-	The kernel co

number = your_serial_number authentication-

nfiguration.

key=

your

_aut

henti

catio

n_ke

y

interf

ace

=ibm

tr 0

wind

ow =

2048

low-

wind

ow =

0

[pctc Th

p e

ifcus Vx

t 0] D

fram ke

e- rn

type	el
=IB	int
MTR	erf
interfac	
ace-	e.
type	
=IB	
MTR	
[pctc	Th
p	e
ibmtr	IB
0]	M

ip-
addr
ess=
128.
127.
50.1
broa
dcas
t-
addr
ess=
255.

T
R
ke
rn
el
int
erf
ac
e
ch
ar
ac

255.255.255 router=128.127.50.6 subnet-mask=25

teristics. Substitute the ad

5.25 dr

5.25 es

5.0 se

s,

su

bn

et

m

as

k,

et

c.,

as appropriate.

3. Examine the CONFIG.SYS file to verify that it includes the appropriate device drivers.

To run multiple protocols over the IBM LAN Support program, you must allocate an additional Service Access Point (SAP) for each protocol. To do this:

– Add device statements to load the parts of the IBM LAN Support program:

DXMA0MOD.SYS, DXMC0MOD.SYS, DXMT0MOD.SYS.

– Set the Extra SAPs variable to add Service Access Points for each protocol that you are running.

You must load the DXMT0MOD.SYS driver with the Extra SAPs (ES) variable set to a number greater than or equal to the number of protocols used.

To set the ES variable, use a text editor to add a line to your CONFIG.SYS file in the following form, after loading the interrupt arbitrator (DXMA0MOD.SYS):

```
device= drive:\directory\dxmt0mod.sys ES= n
```

where n stands for the number of additional SAPs that you want to allocate.

This is a sample section of your CONFIG.SYS file:

```
    device=c:\asi\dxma0mod.sys
    device=c:\asi\dxmc0mod.sys
    device=c:\asi\dxmt0mod.sys ES=2
```

6.6 Loading InterDrive

This section describes how you can edit your AUTOEXEC.BAT file to automatically load PC/TCP InterDrive NFS client.The loading procedures differ based on your kernel implementation (TSR or VxD).

To load the InterDrive TSR

The InterDrive TSR is configured by default when you install the PC/TCP TSR kernel. Before you proceed, you should verify that the AUTOEXEC.BAT file loads the appropriate programs in the appropriate order. The loading procedures differ based on your kernel implementation (TSR or VxD).

Before you proceed, you should verify that the idrive command appears after the command that loads the PC/TCP kernel in your AUTOEXEC.BAT file.

The following AUTOEXEC.BAT file fragment includes commands that set the path of the PCTCP.INI file, load the PC/TCP TSR kernel for a DIX Ethernet network, and start InterDrive:

```
PATH C:\PCTCP
SET PCTCP=C:\PCTCP\PCTCP.INI
C:\PCTCP\3C503.COM 0x60 3 0x280
C:\PCTCP\ETHDRV.EXE
C:\PCTCP\IDRIVE
```

To load the InterDrive VxD

The InterDrive VxD is configured by default when you install the PC/TCP TSR kernel.

Before you proceed, you should verify the vxdinit command appears in your AUTOEXEC.BAT file before you start Windows. This command starts the VxD kernel and VxD Interdrive (by default). For example,

```
PATH C:\PCTCP

SET PCTCP=C:\PCTCP\PCTCP.INI

VXDINIT
```

For more information on using the InterDrive client program, refer to Using PC/TCP in Windows Chapter 6, Sharing Network Files.

6.7 Using PC/TCP Network Drivers with Other Network Operating Systems

The PC/TCP Network Driver (PCTCPNET.DRV) allows you to access remote files and printing services through the Windows File Manager, Control Panel, and Print Manager. If you use other network operating systems concurrently with PC/TCP software, these systems must be connected to enable access to each system’s services. The configuration of these network operating systems is referred to as “chaining” one network to or from another.

The PC/TCP installation program detects other network operating systems on your system and makes the appropriate changes to the [BOOT] section of your SYSTEM.INI file, as outlined in the following table:

Net	SY
wor	ST

k Operating System(s) PC/TCP only	EM.INI entries [BOOT] ne

tw

or

k.d

rv

=p

ctc

pn

et.

dr

v

PC/ [B

TC O

P and another network operating system

O T] ne tw or k.d rv =p ctc pn et. dr

(su	v
ch	pct
as	cp
Ban	ch
yan	ain
VIN	=vi
ES)	ne
	s.d
	rv
PC/	[B
TC	O
P,	O

Windows for Workgroups, and another net

T]network.drv=wfwnet.drvse

wor	co
k	nd
ope	ne
rati	t.d
ng	rv
syst	=p
em	ctc
(su	pn
ch	et.
as	dr
Ban	v
yan	pct

VIN cp

ES) ch

ain

=vi

ne

s.d

rv

Note: You should verify that the [386EnH] section does not contain entries for

WFWFTP.386 or VPCTCP.386.Remove (or comment out) these entries if they appear.

6.8 Related Information About Installing PC/TCP with VINES

For more information about installing PC/TCP with VINES, refer to the following sources:

Topic	Source
Installing the Banyan	Banyan VINES document

VIN	atio
ES	n
soft	
war	
e.	
IP	VIN
add	ES
res	TC
sin	P/I
g	P
and	Opt
rou	tion

ing packets across networks.

Guide

Chapter 6 Installing PC/TCP with Banyan VINES

Chapter 7

Installing PC/TCP with PATHWORKS

PATHWORKS is a network operating system produced by Digital Equipment Corporation. This network operating system works on top of DECnet protocols as well as supporting other protocol implementations. such as TCP/IP. This chapter discusses how to install PC/TCP with PATHWORKS and a network interface card supplied by Digital Equipment Corporation.

Using procedures in this chapter, you can

- Load and configure the appropriate scheduler and DLL driver software.
- Install and configure the PC/TCP DEPCA kernel.

Note: PC/TCP supports a DEPCA TSR kernel only. (PC/TCP does not provide a DEPCA VxD kernel.)

Alternatively, you may install PC/TCP with PATHWORKS over NDIS drivers using a

DIX Ethernet kernel. For detailed procedures for installing PC/TCP over NDIS drivers, see Getting Started.

Figure 7-1 illustrates the general layering of a DLL driver within the PC/TCP architecture using the DEPCA kernel.

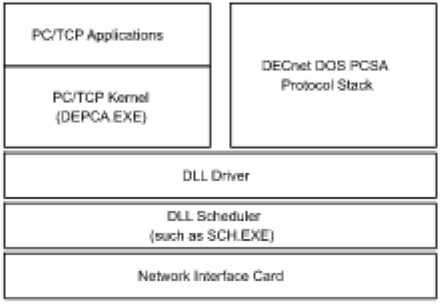


Figure 7-1 Layering of a DLL Driver with PC/TCP

The DLL scheduler provides timing services and background multitasking under DOS.

The Data Link Layer (DLL) driver is a software device driver that provides an interface between the Ethernet controller on the computer and higher software levels. The type of data link depends on the Ethernet controller. A DLL driver is not included with

PC/TCP; you must supply the DLL driver.

The DEPCA kernel is specially built for use with a Data Link Layer (DLL) driver. The

PC/TCP kernel DEPCA.EXE exchanges data with the DLL driver.

7.1 Before Installing the PC/TCP DEPCA Kernel

Ensure that

- You meet all of the requirements in Getting Started.
- PATHWORKS is installed and operational.

7.2 Installing PC/TCP to Run with PATHWORKS — Procedure Overview

Follow these steps to configure your system to run PATHWORKS with PC/TCP:

1. Load the scheduler.

2. Load the DLL driver.

3. Install and configure PC/TCP.

7.3 Loading the Scheduler for the DLL Driver

The scheduler SCH.EXE is a terminate-and-stay-resident (TSR) program that is loaded into memory once, then remains. The scheduler is included in the STARTNET.BAT file that comes with the DLL driver distribution. If you have a STARTNET.BAT file, and you are going to use it to load the scheduler, make sure the line startnet is in your AUTOEXEC.BAT file. This loads the scheduler automatically every time that you start your system.

If you are not using a STARTNET.BAT file, copy the SCH.EXE that came with your DLL driver distribution into any directory in your path:

```
C:\>copy a:sch.exe
```

Note that the scheduler comes with Digital Equipment Corporation DECnet-DOS and PATHWORKS.

To load the scheduler at system startup

Put a line of the following form in your AUTOEXEC.BAT file:

```
sch /option
```

You can use the following options with the sch command. You can use these options whether you issue the command from the DOS prompt, the AUTOEXEC.BAT file, or the STARTNET.BAT file.

/A Configures the SCH.EXE timer automatically. The SCH program uses the CMOS real-time clock if your machine is an IBM PC AT or compatible. Otherwise, the default is to use the system clock as the timer.

/H Selects the hardware timer and uses hardware interrupt 08h, the default of the STARTNET.BAT file.

/N Specifies that the Ethernet vector is not examined for interrupt activity during loading.

/S Specifies that the system clock is used as the timer, without determining the system type.

The following line loads the scheduler from the DOS prompt, and does not examine the Ethernet vector for interrupt activity:

```
C:\> sch /N
```

7.4 Loading the DLL Driver

A DLL driver is a TSR program that is loaded into memory once, then remains. Follow these instructions to load the DLL driver at system startup.

1. Locate the DLL driver on the diskette from your card vendor, or on the PATHWORKS distribution diskettes.

   Below is a partial list of DLL drivers for various interface cards:

   – DLLDEPCA.EXE

   – DLL3C503.EXE

   – DLL3C501.EXE

   – DLL3C523.EXE

   – DLLMICOM.EXE

   Select the file that matches the card installed in your system. For example, if you are using a 3COM 3c503 card, you need the DLL3C503.EXE file.

2. Copy the file to your system and change the name to DLL.EXE.

For example, copy the 3c503 data link file with a command like this:

```
A:\>copy dll3c503.exe C:\pctcp\dll.exe
```

3. To load the DLL driver at system startup, add a line of the following form to your STARTNET.BAT (or AUTOEXEC.BAT) file:

```
dll /option
```

The /option refers to options that you can use with the DLL driver.

The following are the options that you can use:

/ADAPTER:name Identifies the adapter type.

/D:n Indicates the value set for the DMA channel. Do not change the default DMA channel unless a conflict exists.

/FAST Loads the DLL into conventional memory rather than ROM. This action enhances performance but consumes more memory. This option is specific to the DEPCA interface.

/IRQ:n Selects the interrupt channel for the DLL software driver. The value of n should match interrupt number for your interface. Refer to the vendor’s technical documentation.

/M:n Indicates the type of data transfer used by your computer, where n is the value of the type of data transfer. Use this switch if you have not changed the default jumper setting for the memory base address. This switch is optional; if you do not specify the /M switch, the DLL program selects a suitable default.

/MEM:address Specifies the memory base address.

/PORT:n Selects the I/O base address for the DLL software driver. The value of n should match that specified on the settings on the interface card. Refer to the vendor’s technical documentation.

/T:n Indicates the transceiver type, where n is either 1 for BNC (thin wire) or 2 for DIX (standard).

Some DLL drivers (such as DLL3C503.EXE) require that you sp[ecify this switch. See your card vendor’s technical documentation for more information about your DLL driver.

Note: The options may vary, depending on your particular driver software.

7.5 Installing and Configuring PC/TCP to Use with PATHWORKS

To install and configure the depca kernel to run with PATHWORKS, follow these steps:

1. Follow all of the installation steps listed in Getting Started Chapter 1, Installing PC/TCP Software.

2. Examine the resultingPCTCP.INI fine and ensure that there is a [pctcp ifcust n] section similar to this example, substituting the addresses and subnet mask appropriate for your system. (If you have one card, n is 0; if you have two network interface cards, the second interface section is 1, and so on.)

```
[pctcp ifcust 0]

ip-address = 128.127.50.100

subnet-mask = 255.255.255.0

router = 128.127.50.6
```

3. Ensure that you have an entry like the following in the [pctcp kernel] section of your PCTCP.INI file:

```
[pctcp kernel]

interface=ifcust 0
```

Note: If you are familiar with an older version of PC/TCP, and you used the DEPCA kernel, you may have used a [pctcp depca 0] section in the PCTCP.INI file. This changed in PC/TCP version 2.2 to use the [pctcp ifcust 0] section instead.

4. Examine your AUTOEXEC.BAT and ensure that the environment variable is set, as follows:

```
set PCTCP=c:\pctcp\pctcp.ini
```

5. Modify your AUTOEXEC.BAT file to load the scheduler, the DLL driver, and the kernel every time that you start your system. These lines should use the following form:

```
drive:\path\sch /option

drive:\path\dll /options

drive:\path\DEPCA.EXE
```

Here is an example:

```
C:\etc\sch /N

C:\etc\dll /MEM:d000 /IRQ:5 /ADAPTER:DE2000

C:\pctcp\depca.exe
```

This example illustrates using the DEPCA kernel with the 3Comn 3C503 network interface card:

```
c:\drivers\dll\sch.exe

c:\drivers\dll\dll /IRQ:3 /ADAPTER:3c503 /D:1 /T:1

c:\pctcp\depca.exe
```

6. After you complete these steps, start your system and verify the network connection by using the ping command as described in Getting Started section 2.1, Using the PC/TCP Configuration Utility.

7.6 Loading InterDrive

This section describes how you can edit your AUTOEXEC.BAT file to automatically load PC/TCP InterDrive NFS client.

To load the InterDrive TSR

The InterDrive TSR is configured by default when you install the PC/TCP TSR kernel. Before you proceed, you should verify that the idrive command appears after the command that loads the PC/TCP kernel in your AUTOEXEC.BAT file.

The following AUTOEXEC.BAT file fragment includes commands that set the path of the PCTCP.INI file, load the PC/TCP TSR kernel for a DIX Ethernet network, and start InterDrive:

```
PATH C:\PCTCP

SET PCTCP=C:\PCTCP\PCTCP.INI
```

```
C:\PCTCP\3C503.COM 0x60 3 0x280

C:\PCTCP\ETHDRV.EXE

C:\PCTCP\IDRIVE
```

For more information on using the InterDrive client program, refer to Using PC/TCP in Windows   Chapter 6, Sharing Network Files.

7.7 Using PC/TCP Network Drivers with Other Network Operating Systems

The PC/TCP Network Driver (PCTCPNET.DRV) allows you to access remote files and printing services through the Windows File Manager, Control Panel, and Print Manager. If you use other network operating systems concurrently with PC/TCP software, these systems must be connected to enable access to each system’s services. The configuration of these network operating systems is referred to as “chaining” one network to or from another.

The PC/TCP installation program detects other network operating systems on your system and makes the appropriate changes to the [BOOT] section of your SYSTEM.INI file, as outlined in the following table:

Net	SY
wor	ST

k Operating System(s) PC/TCP only	EM.INI entries [BOOT] ne

tw

or

k.d

rv

=p

ctc

pn

et.

dr

v

PC/ [B

TC O

P O

and T]

ano ne

ther tw

net or

wor k.d

k rv

ope =p

rati ctc

ng pn

syst et.

em dr

(su	v
ch	pct
as	cp
PAT	ch
HW	ain
OR	=
KS)	ms
	ne
	t.d
	rv
PC/	[B
TC	O

P, O

Win T]

do ne

ws tw

for or

Work.d

kgr rv

oup =w

s, fw

and ne

ano t.d

ther rv

network operating system (such as PAT	second net.drv=pctcpnet.drv

HW pct

OR cp

KS) ch

ain

=

ms

ne

t.d

rv

Note: You should verify that the [386EnH] section does not contain entries for

WFWFTP.386 or VPCTCP.386.Remove (or comment out) these entries if they appear.

7.8 Troubleshooting the PC/TCP DEPCA Kernel Installation

Trailers are a nonstandard method of encapsulating data for Ethernet (and ProNET-10) installations. Trailers are ofter used by 4.2 Berkeley Standard Distribution (BSD) UNIX systems.

DLL drivers offer only eight portals and PC/TCP requires two of them. PC/TCP uses three more portals to support 4BSD UNIX trailer packet formats, if additional ones are available.

If you experience any of the following problems, trailer encapsulation may be the cause:

- An error message about “portals” appears.

A “portals” error message indicates that other software is also using portals, and there are not enough left for PC/TCP to use in supporting trailers.

- FTP get operations are slow, but FTP put operations are not.
- Telnet (tn) screen updates are slow when transferring large amounts of text from a remote host, but local screen updates and character echos occur quickly.

If you experience any of these problems, ask your UNIX system administrator to turn off trailers.

7.9 Related Information About Installing PC/TCP with PATHWORKS

Refer to the following documentation for more information about the installation:

Topic	Source
Configuring the DEPCA.C	Documentation supplied

OM packet driver.

on the “Supported Packet Driver” disk. If you

are

usi

ng

the

PC/

TC

P

CD-

RO

M

dist

y.

ribu

tion

,

see

the

/CR

WN

YR

sub

dire

ctor

y.

Installing a driver if you are using driver	The documentation supplied by your

software from an interface card ven	interface board vendor

dor. Installing PATHWORKS for DOS.

The PATHWORKS for DOS Client Inst

alla

tion

and

Co

nfig

urat

ion

Gui

de

that

is

sup

plied by your interface board vendor.

Chapter 7 Installing PC/TCP with PATHWORKS

Chapter 8

Configuring NetBIOS

PC/TCP NetBIOS is a session layer interface for local area networks (LANs) that lets you run applications over a variety of network protocols. It provides wide area connectivity for network operating systems (NOS), such as Microsoft LAN Manager. Note that unless you plan to run a network operating system that uses the NetBIOS transport protocol, you do not need to load PC/TCP NetBIOS. A NetBIOS host can communicate only with other RFC-compliant hosts.

PC/TCP NetBIOS uses TCP/IP and is fully compliant with RFCs 1001 and 1002, and with the IBM NetBIOS Application Development Guide.

When you run NetBIOS with a compatible network operating system, your PC can

- Share resources with machines both within a LAN and across a TCP/IP network, using options specified in the PCTCP.INI configuration file.

- Interact with non-PC hosts that implement TCP/IP NetBIOS.

With this chapter, you can learn

- What you must do before you can use PC/TCP NetBIOS.
- How the default configuration addresses most user needs.
- How to start NetBIOS.
- How to set PC/TCP NetBIOS configuration options.
- How to tune PC/TCP for NetBIOS.
- What a NETBIOS programmer needs to know about PC/TCP NetBIOS.
- How to troubleshoot NetBIOS.
- Where to find more information about PC/TCP NetBIOS.

8.1 Before You Start Using PC/TCP NetBIOS

Perform these tasks before you use PC/TCP NetBIOS:

1. Install, configure, and load your network operating system.

2. Install and configure PC/TCP. (The distribution includes PC/TCP NetBIOS.)

Note: To support your network environment, you may need to configure your PC/TCP kernel with additional TCP connections. See section 8.6, Tuning PC/TCP for NetBIOS.

3. Verify that any other NetBIOS product that you are using is RFC-compliant and is compatible with PC/TCP NetBIOS (such as LAN Manager).

8.2 Using NetBIOS in Windows

PC/TCP provides a NetBIOS TSR (terminate-and-stay-resident program) and a NetBIOS VxD (virtual device driver). The PC/TCP installation procedures install and configure the appropriate PC/TCP NetBIOS for your system environment. As a rule, if you are using the PC/TCP VxD kernel, you are also using the VxD NetBIOS. Otherwise, you are using the PC/TCP NetBIOS TSR.

Both PC/TCP NetBIOS implementations provide the same functionality. However, some configuration procedures may differ depending on which implemention is installed.

8.3 Configuring NetBIOS

The PC/TCP installation procedure provides a default NetBIOS configuration. In general, this initial configuration is adequate, and you should not need to override the default settings.

You must unload and reload PC/TCP NetBIOS for new configuration option settings to take effect.

The appropriate configuration option settings may differ depending on the characteristics of your PC/TCP operating environment. Before you begin, you should use the Statistics application (or the inet version command) to determine the type of kernel (TSR or VxD) that you are running.

8.4 Starting NetBIOS

This section provides detailed instructions for starting the NetBIOS VxD and the NetBIOS TSR.

NetBIOS displays these configuration settings when you load it:

```
Memory occupied: 15664, Network adapter number: 0
Local names: 016, Sessions: 010, CBs: 058, Name cache size: 010 names,
NetBIOS name scope ""                                          000 names
from name file
NetBIOS domain scope ""                                        000 addr-s
from bcast file
```

You can start and use LAN Manager or other NetBIOS applications after you load NetBIOS.

8.4.1 Starting the NetBIOS VxD

If your system is configured to use the VxD kernel and NetBIOS, you should enter the vxdinit command into your AUTOEXEC.BAT file.

Before you start Windows, you need to create an entry in your PCTCP.INI file for starting the NetBIOS VxD.

```
[pctcp vxdinit]
vnbep=yes
```

To start the NetBIOS VxD

1. Before you start Windows, verify that the vnbeb=yes parameter is set in the [pctcp vxdinit] section of your PCTCP.INI file.

   This instructs Windows to load the PC/TCP NetBIOS VxD when you start Windows.

2. If the vnbep=yes parameter does not exist, create it.

3. Enter inet unload to unload the PC/TCP VxD kernel.

4. Enter vxdinit to restart the kernel with the new configuration.

5. Restart Windows.

The kernel loads the NetBIOS VxD, and any other VxDs specified in the [pctcp vxdinit] section.

8.4.2 Loading and Unloading the NetBIOS TSR

Use these procedures to load and unload the NetBIOS TSR. PC/TCP NetBIOS loads into upper memory by default.You can use the loadhigh=no parameter in the PCTCP.INI file to load the NetBIOS TSR into conventional memory.

Note: You need to unload and reload NetBIOS for configuration changes to take effect.

To load the NetBIOS TSR

You load the PC/TCP NetBIOS TSR (NETBIOS.COM) either from the DOS prompt or from your AUTOEXEC.BAT file. When you load the PC/TCP NetBIOS TSR with other applications, you should load NetBIOS

- After you load the PC/TCP kernel.
- Before you load Windows (if you are using Windows).
- Before you load InterDrive.

If you are using InterDrive and LAN Manager, you must start the workstation (and log in, if you are using user security) before you load InterDrive. For more information about loading InterDrive, see Chapter 11, Managing the InterDrive Client.

- Before you start any NetBIOS application (for example, before you start a LAN Manager workstation).

If you are using a third-party NetBIOS application, you should start it after you load the PC/TCP kernel and PC/TCP NetBIOS.

This is a sample AUTOEXEC.BAT file section:

```
PATH C:\WINDOWS;C:\;C:\DOS;C:\LOCAL;C:\PCTCP;C:\LANMAN\NETPROG

SET PCTCP=C:\PCTCP\PCTCP.INI

ETHDRV

NETBIOS.COM
```

To unload the NetBIOS TSR

Before you unload NetBIOS, you should stop all network applications that you loaded after NetBIOS, and close any active connections to remote machines made through LAN Manager.

To unload NetBIOS (NETBIOS.COM), enter netbios -u at the system prompt:

```
C:\> netbios -u
```

The following message appears if you attempt to unload PC/TCP NetBIOS while there are still active connections:

```
Unable to unload NetBIOS.
```

To avoid possible TSR problems, unload the applications in the reverse order in which they were loaded.

8.5 Setting PC/TCP NetBIOS Configuration Options

PC/TCP NetBIOS configuration options are applied when you load PC/TCP NetBIOS. A set of standard configuration option default values are supplied with PC/TCP NetBIOS. For most network environments these defaults should be sufficient, and you should not need to reconfigure NetBIOS.

PC/TCP NetBIOS configuration option defaults are specified in the [pctcp netbios] section of your PC/TCP configuration file (C:\PCTCP\PCTCP.INI, by default). You can change PC/TCP NetBIOS options using the standard PC/TCP configuration applications. For more information, see Getting Started   section 2.1, Using the PC/TCP Configuration Utility.

This is a sample PC/TCP NetBIOS configuration file section. When you do not specify a value for a configuration parameter, NetBIOS uses a default value.

```
[pctcp netbios]

adapter-number =

broadcastfile =

cache-elements =

domain-scope =

names =

loadhigh =

namefile =

ncbs =

scope =

sessions =

timeout =
```

You can use PC/TCP NetBIOS configuration options to

- Use multiple NetBIOS adapters.
- Logically partition your NetBIOS subnet.
- Communicate across routers.
- Identify hosts to automatically receive broadcast messages.
- Specify NetBIOS names.
- Use a Domain Name System (DNS) to resolve NetBIOS names.

8.5.1 Using Multiple NetBIOS Drivers

Some network operating systems allow their NetBIOS drivers to coexist with PC/TCP NetBIOS. If you are using multiple NetBIOS stacks, you must set the adapter-number= parameter in the [pctcp netbios] section of your PCTCP.INI file to specify which adapter to use for PC/TCP NetBIOS. For example,

```
adapter-number = 1
```

Note: Some NetBIOS-based network operating system installation programs insert information about their own NetBIOS drivers into the CONFIG.SYS or AUTOEXEC.BAT file. Generally, you should remove any reference to a third-party NetBIOS network driver to avoid conflicts with PC/TCP NetBIOS.

8.5.2 Partitioning Your Network with Scope

Scope is a naming scheme that NetBIOS uses to form a session-layer subnetwork on the LAN. This subnetwork (which differs from an IP subnet) divides hosts into discrete logical groups and limits those that receive broadcasts.

Your PC can communicate only with other NetBIOS hosts that share your scope. For example, if your PC is assigned to the kafka scope and you send a print request to a print server within the yeats scope, you receive the following error message:

```
Network name not found.
```

By default, PC/TCP NetBIOS names have no scope (commonly called a “null” scope). Many NetBIOS vendors (such as 3Com, Bridge, and Excelan) also use a null scope because they design applications for single groups of users on small LANs.

Note: The default scope setting for previous releases was scope=FTP.

You can specify a name scope by editing the scope= parameter in the [pctcp netbios] section of your PCTCP.INI file. To specify a kafka scope, you would enter

```
scope=kafka
```

To specify the null scope, delete the entry from your configuration file, or leave the entry blank. Do not use empty quotation marks.

When you attempt to contact another NetBIOS host, PC/TCP NetBIOS appends your scope (with a leading dot (.)) to your local NetBIOS name (as defined in the computername= entry in your LAN Manager LANMAN.INI file). For example, if you set computername= chris in your LANMAN.INI file, and scope=kafka in your PCTCP.INI file, your host would identify itself as chris.kafka.

8.5.3 Communicating Across Routers

Generally, NetBIOS B-node implementations do not broadcast across IP routers.

PC/TCP NetBIOS provides the following mechanisms to transmit broadcasts across routers:

• Broadcast file	See section 8.5.4, Creating and Using a Broadcast File.
• Name file	See section 8.5.5, Creating and Using a Name File.
• Domain scope	See section 8.5.6, Using Domain Scope.

To configure PC/TCP NetBIOS to communicate across routers

1. Create a broadcast file that identifies the remote hosts to receive NetBIOS messages.

   For example, this broadcast file (C:\PCTCP\NB-BCAST) identifies two remote hosts:

```
128.127.50.11

128.127.50.12
```

For detailed information about how to create and use a NetBIOS broadcast file, see section 8.5.4, Creating and Using a Broadcast File.

2. Create a NetBIOS name file to associate host Internet addresses with a name.

For example, this NetBIOS name file (C:\PCTCP\NB-NAMES) associates names with the hosts identified in the broadcast file:

```
SOMEHOST 128.127.50.11

ANOTHER1 128.127.50.12
```

For detailed information about how to create and use a NetBIOS name file, see section 8.5.5, Creating and Using a Name File.

3. Edit the [pctcp netbios] section of your PCTCP.INI configuration file to identify

your NetBIOS broadcast and name files:

```
[pctcp netbios]

broadcastfile=nb-bcast

namefile=nb-names
```

8.5.4 Creating and Using a Broadcast File

A “broadcast file ” is an ASCII file containing a list of Internet host names or addresses to contact whenever you send information to the network. The broadcast file is read into memory when you load NetBIOS. To change the list of hosts in the broadcast file, you must edit the file and then unload and reload PC/TCP NetBIOS.

When you load PC/TCP NetBIOS, it displays the number of broadcast file entries. Note that PC/TCP NetBIOS does not recognize identical entries as duplicates.

Note: Use this file carefully. NetBIOS sends a message to all local hosts on your LAN and to all hosts in the broadcast file whenever it sends broadcast message or attempts to resolve a name. This can cause heavy network traffic.

This is a sample PC/TCP NetBIOS broadcast file:

```
128.127.55.103
```

```
lanmanserver1

lanmanserver2

128.127.55.120

alpha
```

Specify the path- or filename of the broadcast file by editing the broadcastfile= parameter in the [pctcp netbios] section of your PCTCP.INI configuration file. For example, use this entry to specify the C:\BFILES\MILLER broadcast file:

```
broadcastfile=C:\bfiles\miller
```

If you specify only a filename, PC/TCP NetBIOS searches the directory where your PCTCP.INI file resides. (Use set at the DOS prompt to see the value of your PCTCP environment variable.)

To create a broadcast file

1. Make a list of the names or IP addresses of the remote hosts that you want to add to a broadcast file.

2. Use a text editor to create a new file.

3. Create a name file entry for each name that you want to add.

Each broadcast file entry consists of one line that identifies the hostname or IP address. The maximum number of entries for a broadcast file is 128.

Do not enter a local IP address (such as your LAN Manager server’s address) in the NetBIOS broadcast file.

For example, the following broadcast file (C:\PCTCP\MILLER) contains two entries. The first host is specified by IP address, and the other is specified by a fully qualified Internet hostname.

```
128.127.50.1
```

```
anais.xyz.com
```

4. Verify that the broadcastfile= parameter in your [pctcp netbios] section of your PCTCP.INI configuration file specifies the appropriate file- or pathname.

For example, this entry identifies the miller broadcast file in your PC/TCP home directory (C:\PCTCP, by default):

```
broadcastfile=miller
```

If you specify the miller broadcast file, NetBIOS sends all broadcasts, name resolution queries, and registration packets to the address 128.127.50.1 and the host anais.xyz.com.

8.5.5 Creating and Using a Name File

A “name file” is an ASCII file that associates a list of NetBIOS names with their corresponding fully qualified hostnames or IP addresses. A name file does not consume network resources like a broadcast file does, because NetBIOS sends information to only one remote host at a time. The name file is read into memory when you load NetBIOS.

PC/TCP NetBIOS displays the number of name file entries when you load it. Note that it does not recognize identical entries as duplicates.

This is a sample NetBIOS name file:

```
LMN                 128.127.55.103
LAN1ROOT            lanmanserver1
LAN2ROOT            lanmanserver1
```

```
LAN1PUBLIC         lanmanserver2
LAN2PUBLIC         lanmanserver2
PQR           128.127.55.120
STUV          128.127.55.120
```

Specify the path- or filename of the name file by editing the namefile= parameter in the [pctcp netbios] section of your PCTCP.INI configuration file. This entry identifies the C:\NFILES\PYTHON name file:

```
namefile=C:\nfiles\python
```

If you specify only a filename, PC/TCP NetBIOS searches the directory where your PCTCP.INI file resides. (Use set at the DOS prompt to see the value of your PCTCP environment variable.)

To create a name file

1. Make a list of the hostnames or IP addresses of the remote hosts that you want to add to a name file, together with the NetBIOS names that you want to assign to each.

A NetBIOS name is usually a synonym that you can use instead of a fully qualified hostname or Internet address.

2. Use a text editor to create a new file.

3. Create a name file entry for each NetBIOS name that you want to add.

For each entry, enter the NetBIOS name, press one or more spaces or tabs, then enter the IP address (or fully qualified hostname). Press Enter after each full entry.

To insert hexadecimal values in a NetBIOS name, use a text editor to insert the equivalent quoted control character. For example, to add the hexadecimal value 0x01 to a name, use your text editor to insert Ctrl+A.

Note: NetBIOS names in the name file are case-sensitive. If you are using this name file with LAN Manager, all names in the name file must be uppercase.

The following name file, PYTHON, contains name file entries for two different hosts:

```
idle 128.127.55.100

cleese monty.xyz.com
```

If you are using this name file with LAN Manager, the hostnames must be uppercase:

```
IDLE 128.127.5.100

CLEESE monty.xyz.com
```

NetBIOS names that contain spaces should be enclosed in double (or single) quotation marks:

```
"MY HOST" 128.127.55.100
```

4. Verify that the namefile= parameter in your [pctcp netbios] section of your

PCTCP.INI configuration file specifies the appropriate pathname.

This example identifies the PYTHON name file:

```
namefile=C:\python
```

To use a NetBIOS name

Enter your network command (such as net view or net print), specifying the NetBIOS name associated with the remote hostname.

The following example shows how a LANtastic network user (after loading LANtastic) views the available resources on a remote host. This command connects the user to the remote host MONTY.XYZ.COM using the cleese NetBIOS name defined in the PYTHON name file:

```
C:\> net view \\cleese
```

8.5.6 Using Domain Scope

If PC/TCP NetBIOS cannot resolve a hostname on the local network or through the broadcast file (broadcastfile=) or name file (namefile=) mechanisms, NetBIOS attempts to resolve the hostname by combining an encoded NetBIOS name with the specified domain scope string.

Using a domain scope is most practical if you communicate using a UNIX machine with the Domain Name System (DNS). You can store and update all the NetBIOS names for your network in one host table.

Specify a domain scope using the domain-scope= parameter in the [pctcp netbios] section of your PCTCP.INI configuration file. This entry specifies the xyz.com domain scope:

```
domain-scope=.xyz.com
```

To specify a null domain scope, delete the entry from your configuration file, or leave the entry blank. Do not use empty quotation marks.

PC/TCP NetBIOS uses an extension to the DNS to resolve NetBIOS names. This is an overview of how PC/TCP NetBIOS resolves names:

1. It searches the local PC/TCP NetBIOS name cache.

   Initially, when you load PC/TCP NetBIOS, this cache is empty. When a NetBIOS name is resolved, PC/TCP NetBIOS adds the name to the local name cache.

2. It queries the local network for the NetBIOS name with the scope suffix. (See the scope= parameter in the [pctcp netbios] section of the PCTCP.INI configuration file.)

3. It searches for a match for the NetBIOS name in the name file. The name file associates a NetBIOS name with a fully qualified Internet name. (See the namefile= parameter in the [pctcp netbios] section of the PCTCP.INI configuration file.)

4. It queries the DNS for the encoded and then the unencoded NetBIOS name (if the domain scope suffix was specified).

PC/TCP NetBIOS DNS queries add the domain scope suffix to the NetBIOS name (see the domain-scope= parameter in your PCTCP.INI configuration file). These queries do not use the scope= suffix.

Domain name servers are identified in the domain-name-server= entries in the [pctcp addresses] section of your PCTCP.INI configuration file.

If you do not have a PC/TCP domain name server configured, PC/TCP NetBIOS does not query the DNS to resolve NetBIOS names.

The following example illustrates how to configure PC/TCP NetBIOS to communicate with the hosts, SOMEHOST and ANOTHER1 (by name) across routers.

This is a sample of the relevant entries in your PCTCP.INI file:

```
[pctcp netbios]

broadcastfile=nb-bcast

namefile=nb-names
```

This is a sample C:\PCTCP\NB-BCAST file:

```
128.127.50.10

128.127.50.11
```

This is a sample C:\PCTCP\NB-NAMES file:

```
SOMEHOST 128.127.50.10

ANOTHER1 128.127.50.11
```

To use NetBIOS domain scope

Enter your network command (such as net print or net use), specifying the NetBIOS name associated with the remote hostname.

In this example, a LANtastic network user has defined the domain scope as .xyz.com.

The following command sequence sends the report1 file to printer1, which is attached to the network print server fast.xyz.com:

```
C:\> net use lpt2 \\fast##printer1

C:\> net print report1 lpt2:
```

NetBIOS appends the domain scope .xyz.com to fast and the file is printed on the remote printer using the server fast.xyz.com. (Note that the syntax of the print command varies with each network operating system.)

8.6 Tuning PC/TCP for NetBIOS

Out of the total number of TCP connections configured for the PC/TCP kernel, NetBIOS must allot the following:

- One TCP connection for NetBIOS to monitor incoming session requests from remote hosts.
- One TCP connection for each active session with another host running NetBIOS. For example, a remote host accessing a service on your machine requires one of your TCP connections.

If you are using the PC as a file or print server, you can rapidly use up the available TCP connections. This message (that appears when you load NetBIOS) indicates that you need to reconfigure the PC/TCP kernel with additional TCP connections:

```
Warning: The number of NetBIOS sessions was reduced to n.
```

If you want to have more sessions, configure more connections at kernel startup.

For a detailed description of the procedure for increasing the number of TCP connections, see section 13.3.3, Adjusting the Number of TCP and UDP Connections.

Note: If you are using the NetBIOS TSR, first make sure that you stop all network applications that you loaded after PC/TCP NetBIOS, and close any active connections to remote machines made through a LAN manager or other NetBIOS applications. (Unload the TSRs in the reverse order in which they were loaded.)

8.7 Programming with PC/TCP NetBIOS

The following information assumes that you are a NetBIOS programmer. Read it if you are writing an application program to run over PC/TCP NetBIOS and need to know details about the PC/TCP NetBIOS implementation.

To do this	You need to know that
Use	TCP/

asynchronous I/O	IP NetBIOS allows both synchronous

(blo

cki

ng)

and

asy

nch

ron

ous

(no

nbl

ock

ing)

I/O modes. It indicates completion by

two methods: by upcalling the post rout

ine

s or

by

perf

orm

ing

co

mpl

etio

n

cod

e

verifications in the Network Control Blo

	ck (NCB).
Interpret the INTERFACE	The PC/TCP kernel and Net

_BUSY error	BIOS are not fully re-entrant. The kernel

cannot respond to a requests while it is ser

vici

ng

ano

ther

req

ues

t. If

PC/

TC

P

Net

BIO

S

rec

eiv

es

a

req

ues

t

whil

e

ano

ther request is in process, it rejects the NC

B

and

retu

rns

the

INT

ER

FA

CE

_B

US

Y

err
or.
If
you
r
app
lica
tion
rec
eiv
es
this

err

or

on

a

WA

IT

NC

B, it

mu

st

try

the

	command later.
Interpret nonstandard	Some NetBIOS applica

return values	tions provide utilities that let you list the

status of a NetBIOS session. PC/TCP

Net

BIO

S

rep

orts

AD

AP

TE

R_

ST

AT

US

and

SE

SSI

ON

_ST

AT

US

val

ues

on

pen

din

g sessions, packets sent, packets

received, and retransmissions. (The equ

ival

ent

fiel

ds

dis

pla

yed

on

the

scr

een

for

the

se

stat

us

con

diti

ons

var

y

wit

h

the

utili

ty.)

The following table describes the nonstandard PC/TCP NetBIOS return values:

N Descr

et iption

BI

O

S

C

o

n

dit

ion	
Pending sessions	The number of open and listening sessions.

Packets sent	The number of successful session and datagram send comm

	ands.
Packets received	The number of successful session and datagram packe

	ts received.
Retransmissions	The number of name query retries (unsuccessful

querie

s).

Note: In the Sytek NetBIOS specification, the ADAPTER_STATUS function returns the hardware address of the network interface card. However, PC/TCP NetBIOS returns the IP address of the local PC.

For more information about NetBIOS application programming, refer to the IBM NETBIOS Application Development Guide.

8.7.1 NetBIOS Software Features and Limitations

These are some PC/TCP NetBIOS software limitations:

- PC/TCP does not support the NetBIOS UNLINK command.
- There can be no more than one NO WAIT instance of any other function pending at a time.
- There can be no more than 31 simultaneous sessions (established through CALL or LISTEN functions).
- Datagrams can contain no more than 512 bytes of user data.
- The ADAPTER_STATUS function returns the first 335 bytes of adapter status information, which can contain no more than 16 names.

8.8 Troubleshooting NetBIOS

The following provides solutions to problems that you can have while using PC/TCP NetBIOS:

NetBIOS does not load under the PC LAN Support program.

Remove the IBM NetBIOS device driver from the CONFIG.SYS file. See Getting Started for instructions.

TCP connection messages are displayed.

Increase the number of TCP connections. See section 8.6, Tuning PC/TCP for NetBIOS.

You cannot access a mounted network drive.

Increase the number of TCP connections. See section 8.6, Tuning PC/TCP for NetBIOS.

You see the message Conflicting drives.

Delete persisting connections in your LAN Manager LMUSER.INI file.

You see the message Network name not found.

Verify that the scope= parameter is set correctly in the [pctcp netbios] section of your PCTCP.INI configuration file. See section 8.5.2, Partitioning Your Network with Scope.

You see the message Protocol not loaded.

Confirm that PC/TCP NetBIOS is installed/loaded.

Set ncbs=32 and sessions=32 in the [pctcp netbios] section of your PCTCP.INI configuration file, and configure PC/TCP NetBIOS. See section 8.5, Setting PC/TCP NetBIOS Configuration Options.

Increase the number of TCP connections. See section 8.6, Tuning PC/TCP for NetBIOS.

You see the message Unable to unload NetBIOS.

Stop any active sessions.

You see the message Warning: The number of NetBIOS sessions was reduced to n.

Configure your PC/TCP kernel with additional TCP connections. See section 8.6,

Tuning PC/TCP for NetBIOS.

the

8.9 Related Information About Configuring NetBIOS

Refer to the following sources for more information about configuring NetBIOS:

Topic	Source
Testing the kernel for the	Chapter 13, Configurin

optimal number of TCP connections	g and Tuning the Kernel

and packet buffers. Loading the PC/TCP

Net	
BIO	
S	
Vx	
D.	
Usi	Ap
ng	pen
ker	dix
nel	A,
and	PC/
Net	TC
BIO	P

S configuration file entries.	Configuration Parameters Reference
Usi	Co

ng	mm
net	and
bio	Ref
s	ere
and	nce
vxdi	
nit	
co	
mm	
and	
opti	
ons	

. Using the NetBIOS standard, such	RFC 1001 and RFC 1002

as definitions of broadcast packets and

scope. Programming NetBIOS application

IBM NetBIOS Application De

s. vel

op

me

nt

Gui

de

Chapter 8 Configuring NetBIOS

8.5.5 Creating and Using a Name File

8.5.6 Using Domain Scope

8.6 Tuning PC/TCP for NetBIOS

8.7 Programming with PC/TCP NetBIOS

8.7.1 NetBIOS Software Features and Limitations

8.8 Troubleshooting NetBIOS

8.9 Related Information About Configuring NetBIOS

Chapter 9

Configuring PC/TCP Remotely Using DHCP or Bootp

PC/TCP allows you to keep your PC’s network configuration on a server. By using one of the client configuration programs, DHCP (Dynamic Host Configuration Protocol) or Bootp (Bootstrap protocol), you can automatically get a PC’s network configuration from a server each time that you start the PC.

With either of these programs, you can obtain PC network configuration, regardless of the PC’s location. These programs also let you configure a PC that cannot maintain a local configuration either for security or hardware reasons. By centralizing administration for several PCs, this method of configuration makes information easier to update.

To get configuration information from a server, your network administrator must have set up a DHCP or Bootp server. This determines which protocol you use. DHCP is usually preferred because it is more flexible.

The procedures in this chapter explain how to

- Obtain a permanent or temporary IP address from a server.
- Obtain network configuration information, such as server addresses, printer information, default routers, Domain Name System (DNS) addresses, and a subnet mask.

With this chapter, you can learn

- How your system must be set up before you can use DHCP or Bootp.
- How DHCP and Bootp work.
- How to configure a DHCP client.
- How to start a DHCP client.
- How to configure a Bootp client.

- What to do when you encounter an unexpected situation.

- Where to find more information about DHCP and Bootp.

9.1 Before You Start Using DHCP and Bootp

Ensure that

- You have installed a network interface card in your PC, that it is correctly configured, and that it passes the diagnostic tests supplied by the card vendor.
- You have installed a correct network card driver.
- You or a network administrator have configured a Bootp or DHCP server to provide network information.
- You have installed PC/TCP and have a PCTCP.INI file for your system.

9.2 Understanding a Bootp and DHCP Client

When a DHCP or Bootp client starts on your PC, it sends a broadcast request for configuration information out to the network. If a DHCP or Bootp server is available, it responds to the client request. (A DHCP or Bootp client can also specify an address of a known server that it wants to use for configuration.) The client request contains several values from your system determined by the DHCP client software, including your PC's

- Media Access Control (MAC) address. This address specifies the physical address of the client's system.
- Subnet address.
- IP address field. If the kernel does not have an IP address configured for the PC, this field is 0.

After a server receives the configuration request, it searches its database to find the appropriate configuration for your PC. After the server determines the configuration values and IP address, it sends this information back to your PC. The information received updates the kernel and the PCTCP.INI file on your system. These values can change each time that you start your system and their accuracy is verified every time that you run DHCP or Bootp.

The IP address that you receive from a DHCP server can be a predetermined, permanent address or it can be dynamically allocated from a pool of available addresses. Bootp servers can only return predetermined, permanent addresses. Dynamic allocation allows automatic reuse of an address that is no longer needed by the client to which it was assigned. Dynamic allocation is particularly useful for assigning an address to a client that connects to the network only temporarily.

Dynamically allocated addresses have a lease associated with them. A “lease” determines the length of time that you can keep the address. The DHCP client automatically tries to renew its lease before it expires. If the server does not let the client renew its lease, the server unloads the client’s PC/TCP kernel.

A DHCP client can get network information from either a Bootp server or a DHCP server. During installation, you can select the protocol to use.

DHCP supports clients for both Windows and DOS environments.

The DHCP client for Windows enables you to

- Configure your system.
- View your network configuration.
- Monitor your system status.
- Easily choose network options.

The DOS client uses the command line interface. Because of this, it is more difficult to request specific information, examine your configuration values, and monitor the process.

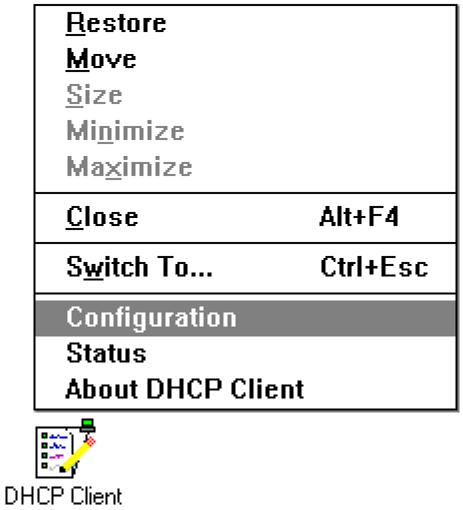
9.3 Configuring a DHCP Client in Windows

If you configured your PC as a DHCP client during installation of PC/TCP, the DHCP client application does not require any additional configuration.

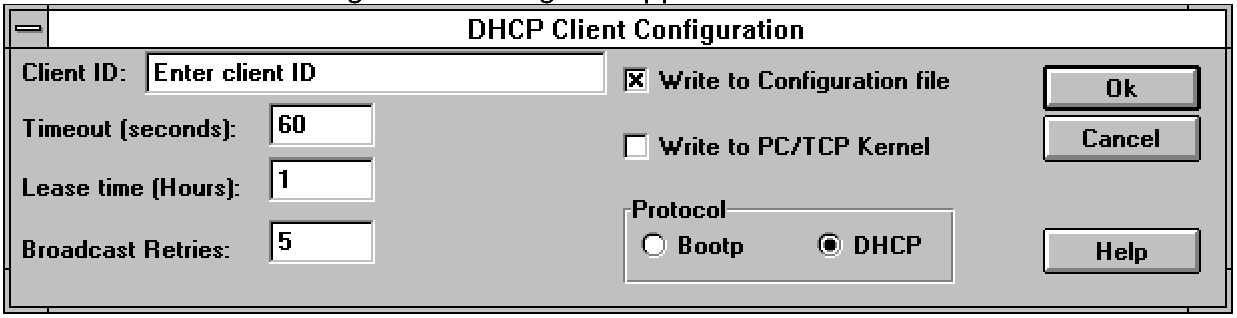
If you did not configure your PC as a DHCP client during PC/TCP installation, or if you need to change configuration values or reconfigure your DHCP client, follow the steps in this section.

To configure your PC as a DHCP client in Windows

1. From the DHCP client’s System menu, choose the Configuration command.



The DHCP Client Configuration dialog box appears.



DHCP Client Configuration

Client ID: Enter client ID

Timeout (seconds): 60

Lease time (Hours): 1

Broadcast Retries: 5

☒ Write to Configuration file

□ Write to PC/TCP Kernel

Protocol

○ Bootp ● DHCP

When you open the DHCP Client Configuration dialog box, the hardware address

(MAC address) of your system is automatically provided by DHCP.

2. If the DHCP server assigns profiles through client IDs, type your client ID.

Specifying a client ID is optional. Ask your network administrator for a valid client ID or

the local policy for entering an ID.

The client ID helps the DHCP server determine your configuration and IP address if

you have previously received configuration information from the server. The DHCP

server also uses the MAC address or the subnet to determine configuration information; however, if you specify a client ID, it overrides these values. This lets you change hardware addresses and still be able to get the same configuration.

You can request a previously assigned IP address in this box instead of a client ID. If available, you will receive the IP address and the same configuration.

3. Choose a Timeout interval.

The timeout interval specifies how long the DHCP client waits for a response. The default timeout interval is 60 seconds.

4. In the Lease Time box, specify the number of hours that you want to keep your IP address.

You need to specify a lease only for a dynamically assigned IP address.

A lease is the amount of time that you are allocated an IP address. If you request more time than the server’s default lease time, you receive the server’s default lease time and a message of explanation. A lease usually ranges in time from 8 to 3600 hours. Your system administrator sets the default lease time. Ask you administrator for your local lease policies.

5. Choose the number of broadcast retries.

This is the number of times that the client attempts to contact a server for a response. The default is 5.

6. In you want to save configuration information in your PCTCP.INI file, select Write to Configuration File.

The only options that cannot be saved to the PCTCP.INI are the IP address, the Serial Number, and the Authentication Key. These values are never written to the configuration file and are always written directly to the kernel. This protects the system from starting with an incorrect configuration for the network.

7. If you want to save configuration information to the kernel, select Write to PC/TCP Kernel.

If you choose to save information to the kernel, DHCP only writes the most important options for configuration. The options kept by the kernel are domain name, DNS server address, router address, IP address, subnet mask, serial number, and authentication key.

Note: If you do not choose to save your information in the PCTCP.INI file or the kernel, your network information is not updated, but you can examine the configuration parameters returned by the server in the DHCP Client Status dialog box.

8. Select either DHCP or Bootp as your protocol.

Your selection depends on the server that provides network configuration information.

The default selection is DHCP. If you are unsure of which protocol to use, ask your network administrator.

9. Choose OK.

You are prompted to save your configuration.

10. Choose OK.

This saves your PC’s information to the server. You are now ready to start the DHCP client.

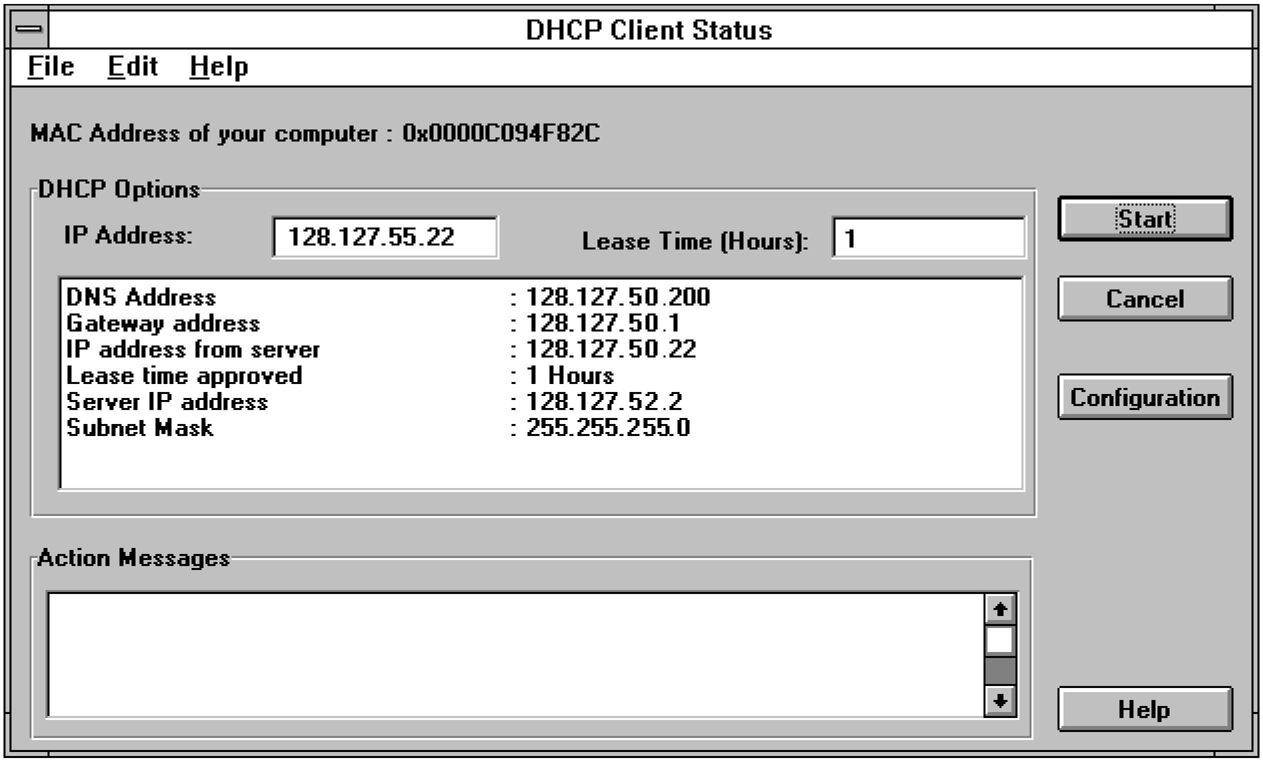
9.3.1 Starting the DHCP Client in Windows

After you have configured the DHCP client, you can use the DHCP Status windows to start the client and send for network configuration values. By default, the DHCP client is located in your Windows Startup directory, so it begins every time that you start Windows.

To start the DHCP client

1. From the DHCP client’s System menu, choose the Status command.

The DHCP Client Status dialog box appears.



MAC Address of your computer : 0x0000C094F82C

IP Address: 128.127.55.22

Lease Time (Hours): 1

```
DNS Address                  : 128.127.50.200
Gateway address              : 128.127.50.1
IP address from server       : 128.127.50.22
Lease time approved          : 1 Hours
Server IP address            : 128.127.52.2
Subnet Mask                  : 255.255.255.0
```

2. Choose Start.

The DHCP client sends your request with your configuration information to a server.

The Action Messages box shows the progress. The server returns the client’s configuration.

If you receive new configuration values, they overwrite older values. Values that were present in past DHCP responses, but are not present in the current response, are invalidated (set to zero or the line is removed). The only information that is not handled in this fashion is the IP address and the serial number and authentication key pair. These values are never written to the configuration file if they are received from the server with configuration information and are always written directly to the kernel (provided that you have chosen to have information written to the kernel). This protects the system from starting with an incorrect configuration for the network.

If you want to make any changes to your configuration after you have started DHCP, simply open the Client Configuration dialog box, make the changes, and choose the OK button again.

Once you have configured the DHCP client for Windows, it starts up with Windows. The application always runs minimized. You must keep DHCP running if you have a dynamically assigned (temporary) IP address so that the server can monitor your lease time.

If you close the application, you will unload the PC/TCP software if you have a dynamically assigned IP address. DHCP gives you a warning before this happens. If you have a permanent IP address, you can quit the application after you have received PC/TCP configuration information.

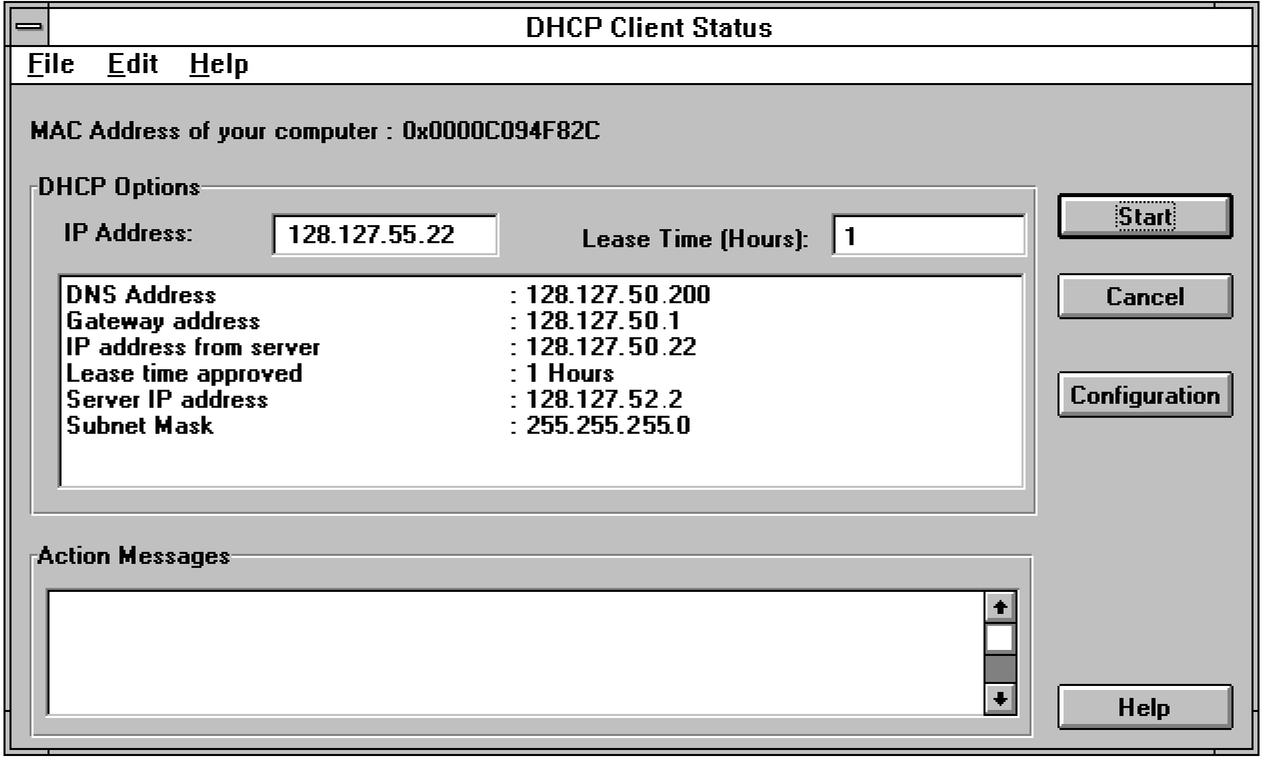
9.3.2 Monitoring DHCP Client Status in Windows

Once you have configured your PC for DHCP or Bootp and started the DHCP client application, you can view your network configuration information.

To view your network information

From the System menu of the DHCP Client icon, choose the Status command.

The DHCP Client Status dialog box appears.



MAC Address of your computer : 0x0000C094F82C

DHCP Options

IP Address: 128.127.55.22

Lease Time (Hours): 1

```
DNS Address                    : 128.127.50.200
Gateway address                : 128.127.50.1
IP address from server         : 128.127.50.22
Lease time approved            : 1 Hours
Server IP address              : 128.127.52.2
Subnet Mask                    : 255.255.255.0
```

This dialog box shows your PC’s IP address, MAC address, lease time, and messages.

If you want to view your network configuration information, choose the Configuration button.

9.4 Configuring a DHCP Client in DOS

You can obtain network configuration information in DOS with the dhcp command. You can type this command at the DOS prompt or you can put the command in your AUTOEXEC.BAT file so that it runs when you start your PC.

The command line interface for the dhcp command is very similar to that of the bootp command (discussed later in this chapter). The primary difference between the commands stems from the way that they handle address assignments. With bootp, each client is identified by a hardware address that is bound to an IP address so that an address for each client must already be assigned and entered with the server. With dhcp you can receive an IP address dynamically or you can retrieve a preassigned address. The dhcp command has several extra command line options to support this additional functionality. The additional options handle lease time, protocol selection, and dynamic address assignment.

9.4.1 DHCP Command Interface for DOS Clients

The dhcp command provides DHCP client capability for DOS clients. The dhcp command requires that your PC has a PCTCP.INI file so that the command can store the information it receives from a DHCP server.

The dhcp command has the format

```
dhcp[options] [config file]
```

With the dhcp command, you can select a communication protocol to use either DHCP or Bootp servers. For a complete listing of dhcp options, see the Command Reference.

To select a protocol

Use the -p protocol option and specify either BOOTP or DHCP for the protocol. Enter the letter d for DHCP or the letter b for Bootp. The default protocol is DHCP.

DHCP also lets you request a lease time for an IP address. This is the amount of time that the client can use the IP address.

To specify a lease time

Use the -l lease option specify the length of time you want to keep your IP address for the lease. Specify the time in hours.

Usually, leases range in length from 8 hours to 3600 hours. Ask your network administrator about your local lease policy. If you ask for too long of a lease, the server assigns you the default length for a lease.

To prevent DHCP from entering configuration information in the PCTCP.INI file

Use the dhcp command with the -nv or the -k command options.

The -nv options let you view the information obtained from the server’s configuration file without writing the information to the PCTCP.INI file or the kernel. The -k option updates only the resident kernel, not the PCTCP.INI file.

To specify the IP address of the DHCP server

Use the dhcp command with the -d address option.

The address specified with this option overrides the server-address= parameter’s setting in the [pctcp bootp] section of the PCTCP.INI file. If the PC’s address is not configured, the DHCP client broadcasts its request to the network and disregards this option.

You can also specify the complete path of a PCTCP.INI file, from which DHCP reads information and to which DHCP writes information by providing the PCTCP.INI file name on the command line. This optional setting overrides the PC/TCP environment variable’s setting.

To see detailed information about the server’s reply

Use the dhcp command with the -v option.

As an example of the dhcp command, you could use the following command line to get configuration information from a DHCP server at the IP address 128.127.50.55 for 40 hours:

```
dhcp -l 40 -d 128.127.50.55 -p DHCP
```

9.5 Configuring a Bootp Client

A Bootp client must be run on a DOS command line or placed in the AUTOEXEC.BAT file to be run when you start up your PC. You can use either the bootp command or the dhcp command with the -p option to get information from a Bootp server.

The bootp command has the format

```
bootp[options] [config file]
```

The bootp command requires that your PC has a PCTCP.INI file so that the command can store the information that it receives from a Bootp server.

To prevent the command from entering configuration information in the PCTCP.INI file

Use the bootp command with the -nv or the -k command options.

The -nv options let you view the information obtained from the server's configuration file (BOOTPTAB) without writing the information to the PCTCP.INI file or the kernel. The

-k option updates only the resident kernel, not the PCTCP.INI file.

The Bootp server can return the vendor network configuration information using different formats, depending on what is specified as the value of the :vm tag in the BOOTPTAB file. The formats control how the information is transported in the 64-byte extension of the Bootp reply packet. Bootp supports replies in the format designed at and named for Carnegie Mellon University (CMU), or in the format compliant with RFC 1533. (If :vm=auto is specified, the server’s default format is used.)

A Bootp reply in the CMU format contains a subset of the information in a reply that conforms to RFC 1048, and uses only predefined tags in the BOOTPTAB file. A Bootp reply in the RFC 1048 format allows BOOTPTAB file tags that are defined by users. However, both formats of the reply packet extension may use no more than 64 bytes to transport the vendor information.

To get the information in the Bootp reply packet in the CMU format

Use the bootp command with the -c command option.

To specify the IP address of the Bootp server

Use the bootp command with the -d address option.

The address specified with this option overrides the server-address= parameter's setting in the [pctcp bootp] section of the PCTCP.INI file. If the PC's address is not configured, the Bootp client broadcasts its request to the network and disregards this option.

To replace your PC's current IP address with a new one from the Bootp server

Use the bootp command with the -f option.

Typically, a server does not give an IP address to a client that already has one. If the server does not reply with a new IP address, the old address stays in use. The bootp command disregards this option if you also use the -n option.

Caution: Never use the -f option with the bootp command if you have network applications already running on your PC; your applications may lose their TCP connections.

You can also specify the complete path of a PCTCP.INI file by providing the PCTCP.INI file name on the command line. The DHCP client uses the file that you specify to store configuration information. This optional setting overrides the PC/TCP environment variable’s setting.

To see detailed information about the server’s reply

Use the bootp command with the -v option (for a verbose listing).

For a complete listing of bootp options, see the Command Reference.

9.6 Troubleshooting DHCP Clients

This section provides possible solutions to situations that you may encounter when you use the DHCP client to get network configuration information. The situation is described first followed by the recommended course of action.

Could not locate server.

Verify that a DHCP or Bootp server is configured and running. If you know the server’s address, try to send a request directly to that address.

Lease is expiring.

The DHCP client automatically tries to renew its lease for its IP address. If the DHCP server does not respond to the client’s request to renew its lease, the client’s PC/TCP kernel will be unloaded. If the DHCP does not renew your lease, check with your network administrator to ensure the server is active and to verify your network’s lease policies.

9.7 Related Information About the DHCP or Bootp Clients

Refer to the following sources for more information about DHCP and Bootp:

Topic	Source
The dhcp and bootp com	Command Referenc

mane

ds

and

their

opti

ons

Boo RF

tp C

prot 95

ocol 1

defi

nitio

n	
DH	RF
CP	C
defi	15
nitio	41
n	
Boo	RF
tp	Cs
exte	15
nsio	32,
ns	15
and	33,

inter15

ope 34

ratio

n

bet

wee

n

Boo

tp

and

DH

CP

Chapter 10

Using PC/TCP with Memory Managers

PC/TCP provides several new features that increase your memory management options and let you optimize PC memory usage.

You can choose to load the kernel, InterDrive (network file system (NFS) client), and NetBIOS TSRs (terminate-and-stay-resident) into upper memory. You can also choose to load portions of the kernel and InterDrive TSRs into expanded memory (EMS).

You can use memory managers with PC/TCP if your PC has more than 1 MB of Random Access Memory (RAM).

PC/TCP complies with the LIM 3.2 Expanded Memory Specification (EMS). (LIM 4.0 applications are compatible with LIM 3.2 if there is a contiguous 64K page frame configured.)

PC/TCP has been tested with the following memory managers:

- Standard memory managers for DOS (EMM386.EXE and HIMEM.SYS)
- QEMM-386 (Quarterdeck Expanded Memory Manager 386)
- 386MAX
- NETROOM

This chapter shows how to

- Exclude your network interface card buffer space.
- Conserve conventional memory.
- Use memory managers with Microsoft Windows.
- Use the memory managers for DOS.
- Use QEMM-386.
- Use 386MAX.

10.1 Before You Use PC/TCP with a Memory Manager

Before you begin, you should

- Understand PC memory management terminology and concepts.
- Install PC/TCP and verify that you have network connectivity.
- Back up your PCTCP.INI, CONFIG.SYS, and AUTOEXEC.BAT files so that you can easily restore your PC’s current configuration (if necessary).
- Make sure that your memory manager is compatible with PC/TCP.

– Verify that it is LIM 3.2-compliant.

– If you load PC/TCP TSRs into upper memory, verify that your memory manager provides UMB (upper memory block) support.

– Verify that your memory manager software interrupt (usually 0x67) does not conflict with the PC/TCP kernel or packet driver interrupt number.

- Review your PC’s memory configuration and requirements.

You need to know what addresses are in use and what addresses are free so that you can load TSRs into memory more efficiently.

If your memory manager is already installed, you can use the report facility it provides to display your memory configuration. For MS-DOS versions 5.0 and later, you can display your PC’s upper memory configuration using the mem command at the system prompt (after first exiting from Windows).

Note: Programs that display your RAM configuration do not generally report memory addresses that are configured for use by special-purpose hardware devices (such as your network interface card).

- Identify any RAM address space to exclude from memory management.

Some software products (such as Windows) and some hardware devices (such as network interface cards) require buffer space in RAM. To run PC/TCP with memory managers you must configure the memory manager to exclude these addresses from management.

To identify RAM addresses to exclude for your network interface cards, see Getting Started.

To identify what areas to exclude if you are using Windows, see section 10.5, Using Memory Managers with Microsoft Windows.

10.2 Understanding Basic PC Memory Management Concepts

This section presents an introduction to basic memory management terms and concepts. Many memory managers use conflicting terminology. You should review these terms before you proceed.

The RAM address space on your DOS PC is partitioned into these regions:

Conventional memory area The 640K memory area 00000h–9FFFFh. DOS typically loads your regular programs and TSR programs into this area.

Upper memory block (UMB) The memory between the end of conventional memory (640K) and 1 MB. DOS typically uses this area for video or network drivers.

Extended memory area RAM at 1024K (1 MB) and above.

High Memory Area (HMA) The first 64K of extended memory (addresses 1024K–1088K).

Note: UMB is also frequently referred to as “high memory,” “upper memory,” and “adapter segment memory.” In this document the term “high memory” refers exclusively to the HMA, the 64K extended memory region just above the UMB. The term “upper memory” refers exclusively to the UMB memory.

The extended memory area can be managed as extended (XMS) memory (using the Extended Memory Specification as defined by Lotus, Intel, and Microsoft), or as expanded (EMS) memory (using the LIM 3.2 Expanded Memory Specification).

- Extended memory managers (XMM) allocate extended memory to XMS-compliant programs and control access to the HMA, a 64K segment that begins at 10000h. You can use extended memory on 286 or higher PCs running in protected mode.
- Expanded memory managers (EMM) allocate memory in 16K pages to EMS-compliant programs. A program can reference up to 64K (four 16K pages). This is

known as an “EMS page frame.” The PC/TCP kernel and some PC/TCP applications run in expanded memory.

Most expanded memory managers can emulate expanded memory using extended memory. This lets an EMS-compliant program access the extended memory blocks through the EMS page frame as if they were expanded memory.

This lets you use extended memory area for

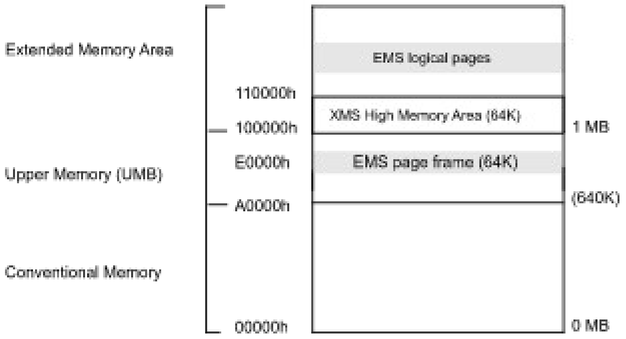
- Expanded memory pages.
- An extended memory RAM disk.
- Microsoft Windows disk cache, through the SMARTDRV program.
- Some program debuggers.

To emulate expanded memory, an expanded memory manager

- Creates the high memory area out of the first 64K of the extended memory area.

- Defines all or part of memory remaining in the extended memory area as extended memory blocks.
- Creates a 64K EMS page frame in the upper memory area. (This reduces the amount of available UMB.)
- Makes some or all of the extended memory blocks look like expanded memory pages.

This figure illustrates a RAM configuration using an extended memory manager to emulate expanded memory (EMS):



10.3 Excluding Your Network Interface Card Buffer Space

Some hardware devices (such as network interface cards) require buffer space in RAM. To avoid interrupt conflicts and to run PC/TCP with memory managers, you must configure the memory manager to exclude these addresses from management.

To configure your memory manager with a network interface card that uses RAM buffers

1. Identify the upper memory addresses configured for your network interface buffer.

The network interface buffer address may be configured with software or with jumper settings on your network interface card. For detailed information about the settings for your network interface card, see Getting Started.

To maximize the amount of free upper memory space, FTP recommends that you choose a network interface base address that is close to video RAM (A000h-C7FFh).

2. Use your memory manager’s command syntax to make the network interface buffer addresses unavailable for use by other devices or TSRs.

For example, for QEMM-386, use the Exclude= parameter; for 386MAX, use the RAM= parameter.

3. If you are using PC/TCP in Windows, you should also instruct Windows to exclude the network interface buffer space by editing the EMMExclude= entry in the [386Enh] section of your SYSTEM.INI file.

This SYSTEM.INI entry reserves the addresses C800h–CC00h:

```
[386Enh]

EMMExclude=C800-CC00
```

10.4 Conserving Conventional Memory

The PC/TCP TSR kernels and other PC/TCP TSRs (such as the InterDrive NFS client) support options that provide greater memory management flexibility and help to conserve conventional memory.

PC/TCP kernel memory requirements differ depending on which kernel you are using and on your configuration option settings. All PC/TCP TSRs report their memory usage when you load them.

PC/TCP options let you

Load into UMB You can load the PC/TCP kernel, and the InterDrive and NetBIOS TSRs into upper memory (UMB).

Load into EMS You can configure the PC/TCP kernel and InterDrive to use expanded memory for some code or data segments.

Reduce kernel memory usage You can affect the amount of memory that the kernel uses by setting kernel configuration options. For more information about reducing the size of the kernel, see section 13.4, Conserving Conventional Memory Usage.

Note: Loading PC/TCP TSRs into expanded memory, or using PC/TCP kernel options to reduce the amount of memory required, both generally result in reduced performance.

Memory managers attempt to arrange your TSRs to make the most efficient use of upper memory. In general, memory managers create an arrangement that is adequate for most machine configurations. After you configure your memory manager, you should review your upper memory configuration to assure that your upper memory usage is optimal.

The PC/TCP kernel and InterDrive TSRs are partitioned into code and data segments.

You can load the segments into upper or expanded memory independently. When you load the PC/TCP kernel TSR, it determines how best to use available upper and expanded memory. If you have enough combined available expanded and UMB memory, you can load the entire PC/TCP kernel above 640K (and not use any conventional memory).

10.4.1 Loading PC/TCP Kernels and TSRs into Upper Memory (UMB)

You can conserve conventional memory by loading the PC/TCP kernel and other PC/TCP TSRs into upper memory. Loading PC/TCP TSRs into UMB requires a memory manager that provides access to upper memory (such as EMM386).

PC/TCP TSRs load into upper memory by default if there is enough upper memory available.

FTP recommends that, in general, you use PC/TCP TSR commands and options to load the PC/TCP TSRs into expanded memory (and not the load high command that your memory manager provides).

Note: To load the PC/TCP TSR kernel into upper memory, you must use PC/TCP load high options.

You can use the standard PC/TCP configuration utilities to specify memory loading options.

This is a summary of the relevant PCTCP.INI file parameter settings:

F UWit

ors h

th e this

is thcon

T isfigu

S c rati

R o on

mopti

mon

a sett

n ing

d

k k [pct

ere cp

n r ker

el n nel]

elloa

dhi

gh

=ye

s

In id[pct

te ri cp

r v idri

D e ve]

ri loa

v dhi

e gh

=ye

s

N n [pct

et etcp

BIbinet

O o bio

```
S s.s]
   c loa
   o dhi
   mgh
      =ye
      s
```

You can use the loadhigh= parameter in the [pctcp vxdinit] section to load the PC/TCP VxD loader into upper memory.

Depending on the configuration of available upper memory, the PC/TCP TSR kernel can load into one or two upper memory blocks. The InterDrive and NetBIOS TSRs each load into one upper memory block (UMB). You can reduce the size of the upper memory block required by the PC/TCP kernel and InterDrive TSRs by also using

PC/TCP expanded memory options (use-emm=yes). PC/TCP TSRs use expanded memory primarily for data buffering. For more information about loading PC/TCP TSRs into expanded memory, see section 10.4.2, Loading PC/TCP Kernels and TSRs into Expanded Memory (EMS).

The following example shows how to load the PC/TCP generic token ring kernel into upper memory using the PC/TCP kernel command. The example assumes that loadhigh=yes in the [pctcp kernel] section of your PCTCP.INI configuration file. Note that PC/TCP TSRs report their memory usage when you load them.

```
C:\> tokdrv

Kernel interrupt vector is 0x61

Code Segment occupies 49.6K of high memory (UMB)

Data Segment occupies 22.8K of high memory (UMB)
```

```
Packet Driver found at vector 0x60

name:

version: 30, class: 17, type: 57, functionality: 6

ifcust (PC/TCP Generic Token Ring packet driver) initialized

5 free packets of length 2048, 5 free packets of length 160

The Resident Module occupies 0 bytes of conventional memory
```

10.4.2 Loading PC/TCP Kernels and TSRs into Expanded Memory (EMS)

If you have an expanded memory manager configured, you can load PC/TCP TSRs into expanded memory. FTP recommends that, in general, you use PC/TCP TSR commands and options to load the PC/TCP TSRs into expanded memory (and not the load command that your memory manager provides).

This table summarizes the PC/TCP expanded memory loading options:

F U or

or s this

th e con

is thfigu

T is rati

S c on

R o opti

mon

msett

a ing

n

d

k k [pct

ere cp

n r ker

el n nel]

eluse

- -

mem

m=

yes

In id [pct

te ri cp

r v idri

D e ve]

ri use

v -

e em

m=

yes

The PC/TCP kernel uses expanded memory for some code segments. Loading the PC/TCP kernel into expanded memory can significantly reduce the amount of conventional or upper memory that the kernel occupies. For example, you can reduce the upper memory required for the DIX Ethernet kernel by approximately 47K if you use the use-emm=yes and loadhigh=yes configuration options.

Note: The PC/TCP kernel can load itself into two non-contiguous upper memory blocks. When you use the expanded (use-emm=yes) and upper memory (loadhigh=yes) options, your kernel can use a smaller upper memory block.

This example shows how to load the PC/TCP DIX Ethernet kernel into expanded memory using PC/TCP command line options:

```
C:\> ethdrv -m
```

The PC/TCP InterDrive TSR uses expanded memory for data buffers by default. When you do not use expanded memory (that is, if you set use-emm=no), the InterDrive TSR uses an additional 7K of conventional memory.

10.5 Using Memory Managers with Microsoft Windows

If you are running Windows over DOS 5.0 (or greater), you must use the HIMEM.SYS and EMM386.EXE drivers supplied with DOS rather than the HIMEM.SYS and EMM386.EXE supplied with Windows.

These are some guidelines for using memory managers with Windows.

To use a third-party memory manager

Verify that your memory manager provides the same XMS capabilities as HIMEM.SYS (the one supplied with Windows). QEMM-386 and 386MAX both provide these XMS services.

To use programs that require expanded memory when Windows is not running

Use a memory manager that provides expanded memory services outside of Windows. (The Windows expanded memory emulation capability is in effect only when Windows is running.)

To use programs that require expanded memory in Windows

Use your memory manager command syntax to place the EMS page frame in the address range supported by Windows (A000h–EFFFh).

10.6 Using the Memory Managers for DOS

You load the memory managers for DOS (HIMEM.SYS and EMM386.EXE) using device= statements in your CONFIG.SYS file.

This is a sample of the relevant entries in your CONFIG.SYS file:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE 512 RAM FRAME=D000 X=C800-CC00
DOS=HIGH,UMB
```

T Does

hi this

s

p

ar

a
m
et
er
D
E
VI
C
E
=

Load
s the
HIM
EM.S
YS
exten
ded
mem

ory

driver

,

then

the

EMM

386.

EXE

expa

nded

mem

ory

emulation program.

512 Configures EMM386.EXE to use

512K of expanded memory. In general, specify 256K more than your TSRs require. This value is adequate

for

the

PC/T

CP

kern

el,

Inter

Drive

, and

NetB

IOS

TSR

	s
	(only
	).
R	Enab
A	les
M	EMS.
	Use
	the
	NOE
	MS
	para
	mete

r instead if you do not need to support expanded

	memory.
FRAME=	Starts the 64K EMS page frame at the specified

address. If you are using Windows, this must be within the

	supported address range (A000h–E000h).
X =	Excludes

the specified addresses from memory management

.

Use

this

para

mete

r to

exclu

de

any

netw

ork

interf

ace buffer space. For the specific addresses to exclude for your PC configuration, see the technical

	docu ment ation for your netw ork card.
D O S =	Uses the HIG H

HI optio

G n to

H, load

U most

M of

B DOS

into

the

high

mem

ory

area.

Uses the UMB option to maintain a link between conventio nal memory and the upper memory area.

To load TSRs into upper memory

You can load TSRs into upper memory using the memory manager loadhigh command or using PC/TCP command and configuration options.

Note: You must use only PC/TCP commands and options to load the PC/TCP kernel into upper memory. Do not use the loadhigh command to load the PC/TCP kernel into upper memory.

This example shows how to load PC/TCP SNMP using the DOS memory manager loadhigh command:

```
C:\> loadhigh C:\PCTCP\snmpd.exe
```

For more detailed information on how to use the DOS memory manager commands and options, see the DOS Technical Reference.

This example shows how to load the PC/TCP DIX Ethernet TSR kernel into upper memory using the PC/TCP kernel command:

```
C:\> ethdrv
```

This assumes that loadhigh=yes in the [pctcp kernel] section of your PCTCP.INI configuration file.

10.7 Using QEMM-386

QEMM-386 is an EMS, XMS, and high memory manager. It automatically converts XMS to EMS memory as needed.

Note: QEMM-386 Stealth mode was not reliable in QEMM-386 versions prior to 7.01.

You load QEMM-386 using device= statements in your CONFIG.SYS file. Here is a sample of the relevant entries in your CONFIG.SYS file:

```
DEVICE=C:\QEMM\QEMM386.SYS EXCLUDE=C800-CC00 RAM
DOS=HIGH
```

T Does

hi this

s

p

ar

a

m

et

er

E Instr

X ucts

C QEM

L M-38

U 6 not

D to

E man

= age addresses in the specified range. Use this para mete

r to

exclu

de

any

netw

ork

interf

ace

buffe

r

spac

e. For the specific addresses to exclude for your

PC configuration, see the technical documentation for

	your network card.
RAM=	Reserves upper memory for loadi

	ng
	TSR
	s and
	drive
	rs.
D	Load
O	s
S	DOS
=	into
HI	the
G	high
H	mem

ory

area.

To load TSRs into upper memory

You can load TSRs into upper memory using the memory manager loadhi command or using PC/TCP command and configuration options.

Note: Use only PC/TCP commands and options to load the PC/TCP kernel into upper memory. Do not use the loadhi command.

This example shows how to load PC/TCP SNMP into upper memory using the QEMM-386 loadhi command:

```
C:\> loadhi C:\PCTCP\snmpd.exe
```

For more detailed information on how to use QEMM-386 commands and options, see the QEMM-386 User’s Guide.

This example shows how to load the PC/TCP InterDrive TSR into upper memory using the PC/TCP idrive command:

```
 C:\> idrive
```

PC/TCP InterDrive loads into upper memory by default. To load it into conventional memory, enter loadhigh=no in the [pctcp idrive] section of your PCTCP.INI configuration file.

10.7.1 Using QEMM-386 with Windows

This section describes how to customize QEMM-386 memory management when you use Windows.

To use programs that require expanded memory in Windows

Edit the QEMM-386 Frame= parameter in your CONFIG.SYS file to have your memory manager place the EMS page frame in the address range supported by Windows (A000h–E000h). For example,

```
DEVICE=C:\QEMM\QEMM386.SYS RAM R:3 X=A000-AFFF FRAME=E000
```

10.8 Using 386MAX

The 386MAX installation program places a device statement like this in your CONFIG.SYS:

```
DEVICE=C:\386MAX\386MAX.SYS PRO=C:\386MAX\386MAX.PRO
```

Use the PRO= option to identify the pathname of the 386MAX.PRO file that contains 386MAX memory management configuration options.

By default, 386MAX places the EMS page frame at E000h.

A sample 386MAX.PRO file looks like this:

```
RAM=C800-CC00 ; Exclude the network interface card buffer area
FRAME=E000
```

T Does

hi this

s

p

ar

a

m

et

er

R Instr

A ucts

M 386

= MAX

not

to manage addresses in this range. Use this para mete

r to

exclu

de

any

netw

ork

interf

ace

buffe

r

spac

e. For the specific addresses to exclude for your

PC configuration, see the technical documentation for

your

netw

ork

card.

This

infor

matio

n is

also

listed

in the

“Net

work

Interf

ace

Card

Infor

matio

n

secti

on of

the

PC/T

CP

Configuration Checklist in Getting Started.

F If you

R are

AME =	using Windows, instructs your memory manager to place

the

EMS

page

fram

e in

the

addr

ess

rang

e

supp

orted

by Windows (A000h–E000h).

To load TSRs into upper memory

You can load TSRs into upper memory using the 386MAX 386load command or using PC/TCP command and configuration options.

This example shows how to load PC/TCP SNMP into upper memory using the 386load command:

```
C:\> 386load SIZE=86864 FLEXFRAME PROG=c:\pctcp\snmpd.exe
```

Specify the size of the TSR (including its initialization code) in decimal.

For more information about 386MAX command line options, see the 386MAX Technical Reference.

This example shows how to load the PC/TCP NetBIOS TSR into upper memory using the PC/TCP netbios command:

```
C:\> netbios.com
```

PC/TCP NetBIOS loads into upper memory by default. To load NetBIOS into conventional memory, enter loadhigh=no in the [pctcp netbios] section of your PCTCP.INI configuration file.

To determine how much upper memory a TSR requires using 386MAX

1. Run 386load with the GETSIZE= parameter (instead of SIZE=):

```
C:\> 386load GETSIZE PROG=c:\pctcp\snmpd.exe
```

2. Run 386load with the /s option to display the Program Memory Summary:

```
C:\> 386load /s
```

Use the reported Initial Size as the value for the 386load SIZE= parameter.

10.8.1 Using 386MAX with Windows

This section describes how to customize 386MAX memory management when you use Windows.

To use programs that require expanded memory in Windows

Edit the 386MAX Frame= parameter in your CONFIG.SYS file to place the EMS page frame in the address range supported by Windows (A000h–EFFFh).

If the 386MAX maximize program places your EMS page frame in an area that is not supported by Windows, you must rearrange your upper memory configuration to create a 64K block available in the supported range.

To run Windows in standard or enhanced mode

Edit the 386MAX RAM= parameter in your CONFIG.SYS file to exclude the translation buffer areas from management.

10.9 Related Information About Using Memory Managers

For more information about managing memory, refer to the following sources:

Topic	Source
Configuring Net BIOS.	Chapter 8, Configuri

	ng NetBIOS
Configuring InterDrive.	Chapter 11, Managing

	the
	Int
	erD
	rive
	Cli
	ent
Dia	Ch
gno	apt
sing	er
PC/	22,
TC	Tro
P	ubl

problems.

eshooting the Kernel and Driver Config

	uration
Configuring PC/TCP TSRs to use	Appendix A, PC/TCP Config

me	ura
mor	tion
y	Par
opti	am
ons.	ete
	rs
	Ref
	ere
	nce
Co	Co
mm	m
and	ma

line	nd
opti	Ref
ons	ere
for	nce
load	
ing	
PC/	
TC	
P	
TS	
Rs.	
Ide	Get

ntifying network interface RAM buffer	ting Started Chapter 1, Installing PC

to exclude from memory management.	/TCP Software

Installing and troubleshooting your PC’s

Your memory manager’s technical

me doc

mor um

y ent

confatio

igur n

atio

n.

Chapter 10 Using PC/TCP with Memory Managers

10.8.1 Using 386MAX with Windows

10.9 Related Information About Using Memory Managers

Chapter 11

Managing the InterDrive Client

InterDrive®software makes it possible for you to connect to a Network File System (NFS) server from your PC and use remote files and printers as if they were local. InterDrive can run as either a terminate-and-stay resident (TSR) program or a virtual device driver (VxD). The TSR is necessary for DOS users. Windows users can choose between the TSR and the VxD; see section 11.2, Choosing the InterDrive TSR or VxD for details.

Figure 11-1 shows the relationship between applications, the InterDrive TSR, and the NFS server in a DOS environment. You can use InterDrive commands, such as idnet, idmnt, and idutil, to configure and manage InterDrive. Once you have established connections to file systems and printers, the client TSR manages the interaction between DOS user applications and the NFS server.

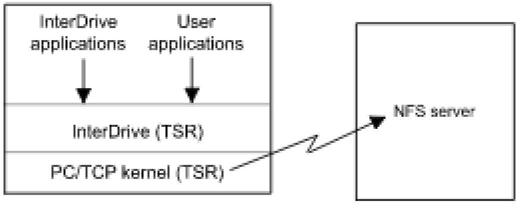


Figure 11-1 InterDrive in DOS

In Windows, an additional component, called the PC/TCP Network Driver, is necessary for access to network files and printers. Figure 11-2 shows the relationship between applications and PC/TCP file and print sharing components in a Windows environment. To configure and manage file and print sharing services, you can use the standard Windows File Manager, Print Manager, and Control Panel interfaces, or alternately, the PC/TCP Network Control application. Once you have established connections to file systems and printers, the PC/TCP Network Driver and InterDrive manage the interaction between user applications and the NFS server.

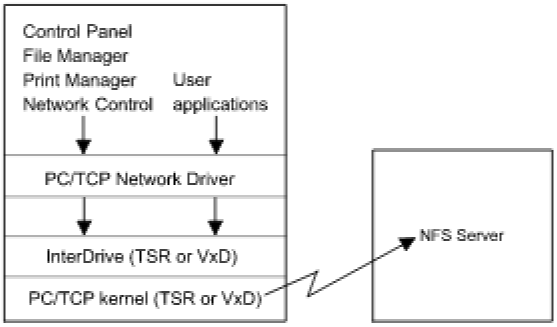


Figure 11-2 InterDrive in Windows

This chapter is intended for people who need to have a deeper understanding of interactions between InterDrive and the NFS server in order to manage InterDrive, either individually or for a large network of end users. With this chapter, you can learn

- What you must do before you start managing InterDrive.
- How to choose between the TSR and VxD versions of InterDrive.
- How to start and stop InterDrive.
- How to create a configuration file that you can adapt for multiple users.
- How to use a centralized authentication server for multiple clients.

- Factors to consider in managing InterDrive’s use of memory.

- Factors to consider in tuning InterDrive network performance.

- What to do when you encounter unexpected situations.

- Where you can find more information about using InterDrive.

11.1 Before You Start to Manage InterDrive

Make sure that

- You have read the related file and print sharing chapters in Using PC/TCP in DOS. See also Using PC/TCP in Windows for information about file and print sharing in Windows. For a detailed list of related documentation, see section 11.9, Related Information About Managing InterDrive.
- You know and understand the environment on the server side, including the type of operating system, the type of NFS services available, and any resulting requirements or limitations.
- You know whether you are running the TSR or VxD version of the PC/TCP kernel, and why.

11.2 Choosing the InterDrive TSR or VxD

InterDrive can run as a terminate-and-stay-resident program (TSR) or as a virtual device driver (VxD). The version of InterDrive configured for your system depends on your choice of PC/TCP kernel. When you choose the VxD kernel at installation time, you get the VxD InterDrive. Likewise, when you install the TSR kernel, you get the TSR InterDrive.

If your working environment is mainly Windows, FTP recommends that you use the VxD kernel and InterDrive because of their memory advantage. In addition to the Windows interface for NFS printing and file sharing, the VxD supports the DOS InterDrive commands when you use them from a DOS session within Windows.

The InterDrive VxD loads when you start Windows. When you exit from Windows, the VxD versions of the kernel and InterDrive unload, and InterDrive closes any network connections that you made while in Windows. If you need network connectivity before you start Windows, or if you need to preserve network connections when you exit from Windows, use the TSR kernel and InterDrive. With the TSR, you avoid having to reconfigure the network software and reconnect drives and printers every time you move between DOS and Windows. For additional considerations and information about choosing TSR or VxD solutions, see the Getting Started guide.

11.3 Starting and Stopping InterDrive

If you select the TSR kernel and InterDrive at installation time, the install program inserts the idrive command in your AUTOEXEC.BAT file and InterDrive should start automatically each time that your system starts.

If you select the VxD kernel and InterDrive, the install program adds the vxdinit command to your AUTOEXEC.BAT file and inserts the vidrive=yes parameter in the [pctcp vxdinit] section of your PCTCP.INI file. The vxdinit command prepares your system to load the VxD kernel and other VxD components when Windows starts. The vidrive=yes parameter indicates that you want to load the InterDrive VxD.

You can use these commands to start and stop InterDrive at times other than system startup or Windows startup. For example, after you change certain InterDrive parameters, you need to stop and restart InterDrive for the change to take effect.

To stop the InterDrive TSR

Enter idutil -u at the system prompt.

InterDrive disconnects all mounted file systems and printers and unloads from memory.

To start the InterDrive TSR

Enter idrive at the system prompt.

To stop the InterDrive VxD

1. Exit from Windows.

The kernel and all VxD components unload.

2. If you have changed parameters and you want the new parameters to take effect,

at the system prompt, enter vxdinit -u to unload the VxD loader.

To start the InterDrive VxD

1. At the system prompt, enter vxdinit to restart the VxD loader.

2. Start Windows. The VxD kernel and InterDrive load with new parameters.

11.4 Creating a Universal Configuration File

If you are a network administrator in charge of setting up NFS file and print sharing for a number of users, you can create default PCTCP.INI configuration file sections that each user can copy and modify as appropriate. Use the TEMPLATE.INI file in the directory containing your PC/TCP files to obtain a skeleton copy of the InterDrive configuration file sections.

If you want the PCTCP.INI file to be shared by more than one user, you can specify user= ? in each section defining a specific file system or print session. This causes InterDrive to prompt for a username during an attempt to mount the file system or printer.

Use the [pctcp idrive] section to specify parameters that apply globally for all InterDrive connections. Use the [pctcp idrive filesys] and [pctcp idprint print_session] sections to define specific file system and print connections. You can override some global parameters for specific file systems. For details, see Appendix A, PC/TCP Configuration Parameters Reference.

11.5 Using a Centralized Authentication Server

The PCNFSD is the server that authenticates your username and password when you try to mount a file system or printer. The PCNFSD also controls the mounting of printers, so it must be running on any system whose printer you are trying to mount. However, when you mount file systems, it is possible to use a different server for authentication. For example, you might mount file systems on servers named Green, Blue, and Red, with server Green authenticating mount requests for all systems. A centralized authentication server is useful if

- You want to store and maintain password information in one secure, centralized place.
- PCNFSD software is not available on all NFS servers in your network.

By default, InterDrive requests authentication from the same server whose files or printers you are trying to mount. You can use the pcnfsd= parameter in the [pctcp idrive] section of your PCTCP.INI file to specify the hostname or IP address of another authentication server.

11.6 Managing InterDrive’s Use of Memory

When it starts, InterDrive automatically tries to make the most efficient use of memory that it can. If you are using the InterDrive VxD, you do not need to worry about how it uses memory. If you are using the TSR, there are some things that you can do to improve its efficiency.

This section gives a brief overview of how InterDrive uses memory. It assumes that you have a basic understanding of memory management concepts. For introductory memory management concepts and additional guidelines, see Chapter 10, Using PC/TCP with Memory Managers.

11.6.1 Understanding How the VxD Uses Memory

A major advantage of the InterDrive VxD is that you do not have to understand much about how it uses memory. The VxD relies on Windows memory management, which does not conflict with the needs of other Windows applications. Therefore, the VxD can allocate as much space in memory as it needs.

For the most part, the VxD uses the same configuration parameters and defaults as the InterDrive TSR, reading them from the [pctcp idrive] section of your PCTCP.INI file. Because of the increased memory capabilities of Windows, the VxD overrides some defaults in the [pctcp idrive] section in favor of larger values. For example, if you are using the VxD InterDrive, you can mount the maximum of 16 file systems and 7 printers by default.

The VxD stores values specific to its operation in the [pctcp idrive-vxd] section. For more information, see Appendix A, PC/TCP Configuration Parameters Reference.

11.6.2 Adjusting How the TSR Uses Memory

The amount of memory that the InterDrive TSR occupies varies depending on a number of factors, including

- Whether or not the program can load into upper memory.
- Whether or not an expanded memory manager (EMM) is running on your system.
- The number of default printers and file systems that you configure.
- The settings for FILES and FCBS in your CONFIG.SYS file.
- Whether and how you use caching.

By default, when you start the InterDrive TSR, it tries to load into an upper memory block if

- A memory manager that provides access to upper memory (such as

EMM386.EXE) is present, and

- There is enough room in upper memory on your system.

InterDrive also checks for the presence of an EMM when it loads. If an EMM is running, InterDrive allocates space in expanded memory for the maximum number of print entries, lookup cache buffers, name-mapping cache buffers, and read-write cache buffers.

You can disable the use of upper memory and expanded memory by setting loadhigh= no and use-emm= no in the [pctcp idrive] section of your PCTCP.INI file. However, remember that the use of these features can save a great deal of conventional memory.

You can do even more to reduce the TSR’s use of memory by configuring the minimum resources that you need for printing and file sharing. The print-entries= parameter in the [pctcp idrive] section of your PCTCP.INI file specifies how many printers you want to be able to mount concurrently. The default number of print entries is 1 without an EMM and 7 with one. For every print entry that you configure, InterDrive allocates print buffer space in memory. Therefore, you can conserve memory by limiting the print-entries= value to the maximum number of printers that you need, or specifying print-entries= 0 if you do not use NFS printing.

InterDrive also allocates space based on the maximum number of files that can be open at one time on your system. This number is determined by the sum of the FILES= and FCBS= entries in your CONFIG.SYS file, or by a command issued to applications such as the Quarterdeck Office Systems FILES.COM or FCBS.COM programs. To conserve memory, ensure that the values that you have set for these entries are reasonable estimates representing the number of files that you plan to use concurrently.

When the InterDrive TSR loads with an EMM running, it allocates four pages (a page frame) for each of the following:

- Lookup cache
- Name mapping cache
- Printing and read-write cache

Although memory usage is not as critical an issue with an EMM, you can reduce the amount of extended memory that InterDrive uses by disabling features that you do not need.

For example, if you do not need read-write caching, you can disable it by specifying rw-cache-buffers= 0. Disabling read-write caching and setting print-entries= 0 reduces extended memory use by four pages.

11.7 Tuning InterDrive Network Performance

InterDrive automatically configures certain parameters with values that should work best for the environment in which it is running. However, if you experience slow response time when using network files or printing to an NFS printer, you may benefit by tuning some of these parameters. Remember that increasing the values for parameters only on the client side does not guarantee improved performance. Network performance is influenced by a number of factors, including

- The type of network connecting the client and server, and whether or not there are intervening gateways or routers.
- The server’s ability to process requests and exchange data.
- The method that an application uses to read and write data.

This section describes how you can improve performance, with those considerations in mind, by

- Changing the packet read and write size.
- Changing the streaming value.
- Using read-write caching.
- Using TCP connections.

11.7.1 Adjusting Packet Read and Write Size

“Read size” is the size of packets that InterDrive gets and processes from the NFS server. An example of a read is when you copy a file from the server to your local PC.

“Write size” is the size of packets that InterDrive formats and sends to the server. An example of a write is when you save a file to a network drive.

By default, InterDrive sets the write size to the transfer size requested by the server (usually 8K) and selects the largest possible packet read-size based on these factors:

- The transfer size requested by the server.
- Whether or not the PC/TCP kernel has any huge packets allocated to it. The VxD kernel allocates 25 huge packets by default; you can set the TSR kernel value with the huge-packets= parameter in the [pctcp kernel] section of your PCTCP.INI file.
- Whether or not streaming is enabled for the mounted file system.

Because large packets carry more data per transaction, fewer packets need to be sent, which usually results in better network performance. However, sometimes large packets have an adverse effect on performance.

For example, if there is a router or gateway between the client and server, the router breaks down (fragments) packets into the largest size it can handle, adding time to network transactions. So, if a gateway fragments packets larger than 512 bytes, for instance, you should set the InterDrive read and write sizes to 512.

Another reason for poor performance could be that the NFS server does not process large packets well. Many older NFS servers take longer processing read sizes that are not 1024 bytes or multiples of 1024.

To verify the current read and write size

Use the idutil -f command.

The output displays current settings for each mounted file system, including the current read and write sizes. It also shows statistics for “current round trip time” and “base round trip time” so that you can analyze whether network transactions are taking longer than they should be.

To change the read size

1. Set the read-size= parameter in your PCTPC.INI file. Specify a global value in the [pctcp idrive] section or a value for a specific file system in a [pctcp idrive filesys] section. The minimum value is 512 bytes; the maximum is 8192. The new value takes effect when you restart InterDrive or remount the specified file system.

   –or–

   Use the idconfig -r drive: command to set the read size for a specified file system for the current InterDrive session only.

2. If you increase the read size to its maximum (8K), verify that at least one huge packet buffer is allocated for the PC/TCP kernel to process large packets.

You set this with the huge-packets= parameter in the [pctcp kernel] section of your PCTCP.INI file. The default value is 25 buffers for the VxD kernel and 0 for the TSR kernel.

3. Unload and restart the kernel for a new huge-packets= value to take effect.

To change the write size

Set the write-size= parameter in the [pctcp idrive] section of your PCTCP.INI file.

Specify a global value in the [pctcp idrive] section or a value for a specific file system in a [pctcp idrive filesys] section. The minimum value is 512 bytes; the maximum is 8192.

The new value takes effect when you restart InterDrive or remount the specified file system.

–or–

Use the idconfig -w drive: command to set the write size for a specified file system for the current InterDrive session only.

11.7.2 Tuning Stream Values

In a typical network exchange, one system sends a packet, then waits for a response from the remote host before sending the next packet. InterDrive can do what is known as “streaming”; that is, it sends several packets at a time without waiting for a response for each one sequentially. Because several packets are being processed concurrently, writes and reads can take much less time than waiting for each packet to travel through the network and be acknowledged by the remote host. InterDrive does require an eventual response for each packet that it sends. Consequently, streaming can improve performance without affecting reliability.

Streaming can improve performance

- In networks with long transmission delays, where sequential processing of each packet takes a long time.

- When the server has high processing capacity and can handle streaming well.

Sometimes streaming can actually degrade performance. Decreasing streaming, or disabling it altogether, can improve performance in the following situations:

- When the InterDrive read size is larger than the maximum transmit unit (MTU) for the network.
- When the server is not able to process streamed packets well.
- If there is a router in the network path between client and server.
- If you are using a serial line connection to the remote host.

One indicator that streaming is not working well is a high number of retransmitted packets.

To see current statistics on retries sent

Use the idutil -s command.

The command produces output similar to the following:

```
---------------------InterDrive Statistics------------------------

9696 RPC requests sent

9650 RPC replies received

119 retries sent

73 bad XIDs received

174 NFS errors (these can be ignored)

0 stale filehandle errors

0 send errors

0 receive errors

Last net I/O error = 0 (No error)

Last NFS error = 248 (Unexpected end of file or directory)
```

```
 Last DOS error = 18 (No more files)
```

A high value for retries sent can indicate a performance problem due to streaming. If InterDrive is busy retransmitting unacknowledged packets, and the server is overloaded trying to process all of the retransmitted data, the result is degraded performance.

A high number of retries can also indicate other network problems. For example, the bad XID count indicates the number of duplicate replies that the server sent in response to retransmitted requests. A very low bad XID value in comparison to the retries value could indicate that packets are being dropped; the client keeps resending packets, but the server does not receive them or respond. A high retry count also can indicate an uneven round trip time between the client and server due to network congestion or router problems. These factors require adjustment to your network configuration that is outside of the scope of InterDrive tuning.

To change the stream value

Edit the stream= parameter in the [pctcp idrive] section of your PCTPC.INI file. The new value takes effect when you restart InterDrive. Change the value by increments of 1 until you reach a setting that provides adequate performance. You can also set this value for a specific file system in a [pctcp idrive filesys] section.

–or–

Use the idconfig -q command to change the stream value for the current InterDrive session only. The command can change the setting either globally or for a specified file system.

Note: InterDrive automatically sets the upper limit for streaming to one less than the number of ÿÿÿÿÿÿaming can actually degrade performance. Decreasing streaming, or disabling it altogether, can improve performance in the following situations:□

c