5 Configuring Network Services

This chapter describes the required and optional network services used by NCD terminals. The contents of this chapter are:

The following network services are discussed in other chapters because they are used only when booting: address discovery and subnet mask discovery (Chapter 3, Booting—Address Discovery) and X server download service (Chapter 4, Booting—X Server Loading).

Which Network Services Are Used by NCD Terminals

NCD terminals require that host computers residing on the network provide the following services. Some of the following are optional or depend on the network configuration:

- ❑ Address discovery service—provides terminals with their network addresses and other information at boot time. You can use BOOTP/DHCP or RARP for address discovery, or you can store addresses in NVRAM.

  For information on address discovery services and storing information in NVRAM, see Chapter 3, Booting—Address Discovery.
- ❑ Subnet mask discovery service—provides the terminal with the subnet mask (if your network uses subnetting) when it boots. Most versions of BOOTP/DHCP allow you to enter the subnet mask in the database file. The alternatives to BOOTP/DHCP are ICMP (Internet Control Message

Protocol), which is included in the TCP/IP protocol family, or storing the subnet mask in NVRAM.

For information on the subnet mask, see Chapter 3, Booting—Address Discovery.

- X server download service—NCD terminals use TFTP or NFS for downloading an X server file.

  For information about downloading X servers from the network, see Chapter 4, Booting—X Server Loading. For basic information on starting and configuring TFTP and NFS, see "Configuring How a Terminal Accesses Files" on page 5-13.

  For information on loading an X server locally from a PCMCIA card, see the *NCDware System Administrator's Guide for UNIX Systems*.

- File service—NCD terminals use TFTP or NFS for accessing files on network hosts during normal operation and at boot time.

  For information on configuring the file service, see "Configuring How a Terminal Accesses Files" on page 5-13.

  For information on accessing local files on a PCMCIA card or diskette, see the *System Administrator's Guide*.

- Address resolution service—The Address Resolution Protocol (ARP) translates between hardware addresses and IP addresses. Translations can also be configured manually.

  For information on address resolution, see "Configuring the ARP Cache (Resolved Addresses)" on page 5-5.

- Name service—By using a name service, you can specify hostnames instead of network addresses in terminal configuration parameters or in commands that require a network address. NCD terminals can use either the IEN 116 (Internet Engineering Notes) name service or the Domain Name System (DNS). You can also store translations in a terminal's local name cache.

  For information on name service, see "Using a Name Service" on page 5-7.

Summary of Network Service Defaults and Alternatives

Table 5-1 lists the default network services setup of an NCD terminal, the alternatives, and where to get more information. When you add a terminal to the network using ***ncdinstall***, the defaults are in effect.

Table 5-1 Default Network Services and Alternatives

Default	Alternatives	References
Address Resolution Service		
The terminal uses ARP for address resolution. Entries are added to the terminal's ARP cache as addresses are resolved.	Add entries manually to the ARP table.	"Configuring the ARP Cache (Resolved Addresses)" on page 5-5
ARP table entries last for 20 minutes	Change the timeouts for complete and incomplete entries.	
Name Service		
The terminal uses IEN 116 protocol for name service unless NVRAM is set to factory defaults and BOOTP/DHCP supplies name server information.	Use DNS instead of IEN 116.	"Using a Name Service" on page 5-7
	Configure the local name cache manually.	
The terminal uses the boot host as the name server host.	Specify other name servers.	
The fully qualified domain name must be specified when referring to network hosts.	Specify the domain name suffix.	
The terminal does not send a reverse name request to discover its own hostname when it boots.	Configure the terminal to send a reverse name request.	"Using a Name Service" on page 5-7
Name cache parameters are set to their default values.	Redefine the name cache parameter values.	

Table 5-1 Default Network Services and Alternatives (Continued)

Default	Alternatives	References
File Service		
File service is from the boot host.	Specify initial file servers.	"Configuring How a Terminal Accesses Files" on page 5-13
	Configure the file service table to add other hosts and file systems.	
Routing		
Routes are automatically placed into the routing table.	Manually configure the routing table.	"Configuring Routing (Accessing Remote Networks)" on page 5-25
The boot host is the default gateway.	Specify default gateways.	
Router discovery is used to discover neighboring gateways.	Turn off router discovery.	
TCP Performance		
TCP performance parameters have default values.	Customize the TCP performance parameters.	"Setting TCP Performance Parameters" on page 5-31

Configuring the ARP Cache (Resolved Addresses)

A terminal attempting to contact another host broadcasts the IP address of the host via the ARP protocol and receives the host's Ethernet address. These resolved addresses are maintained in the terminal's ARP cache, a local table of resolved addresses. The terminal checks its ARP cache before attempting to contact a host to see if the address has already been resolved.

The **tcpip-arp-cache** parameter contains all of the addresses that ARP has resolved or attempted to resolve (Setup ⇒ Change Setup Parameters ⇒ ARP ⇒ ARP Cache). Table 5-2 lists the entries in a row in each row of the table.

You can manually add entries to the ARP cache if necessary.

Table 5-2 tcpip-arp-cache Table Entries

Table Entry	Possible Values	Result
ethernet-address	default	00:00:00:00:00:00
	ethernet address	The Ethernet address the host supplies in response to the ARP request from the terminal.
ip-address	default	0.0.0.0
	IP address	The IP address broadcast by the terminal.
type	default	incomplete
	incomplete	The IP address could not be resolved to an Ethernet address.
	dynamic	This completed entry is subject to automatic deletion after the timeout elapses.
	static	This entry is not subject to automatic deletion.
time-since-last-use	default	0
	integer	The amount of time (in minutes) since this entry was used by the terminal. Range: 0 - 255.

The lifetimes of the dynamic and incomplete entries in the **tcpip-arp-cache** table are governed by the following parameters.

The **tcpip-arp-complete-entry-timeout** parameter specifies how long a dynamic ARP table entry should be allowed to exist without being used before it is automatically deleted (Setup ⇒ Change Setup Parameters ⇒ ARP ⇒ Complete Entry Timeout).

Table 5-3 tcpip-arp-complete-entry-timeout Parameter

Possible Values	Result
default	20
integer	How long to wait (in minutes) before deleting an unused complete entry. Range: 1 - 255.

The **tcpip-arp-incomplete-entry-timeout** parameter specifies how long an incomplete ARP table entry should be allowed to exist before it is automatically deleted (Setup ⇒ Change Setup Parameters ⇒ ARP ⇒ Incomplete Entry Timeout).

Table 5-4 tcpip-arp-incomplete-entry-timeout Parameter

Possible Values	Result
default	1
integer	How long to wait (in minutes) before deleting an incomplete entry. Range: 1 - 255.

Using a Name Service

A name service translates between IP addresses and hostnames. Name service is optional, but you must use it if you want to specify hosts by their hostnames instead of their IP addresses. It is simpler and more meaningful to specify hostnames in remote configuration files, Setup menus, and commands. If you try to specify a hostname without using a name service, the terminal cannot find the host.

NCD terminals can use both DNS (Domain Name System) and IEN 116 name services.

Making Sure a Name Service is Running on the Local Network

To make sure DNS is available on the name server host:

- ❏ Verify that the daemon (***named*** or ***in.named***) is configured in the relevant startup file on the name server host. You can use a command similar to the following to find the command line starting up the name daemon:

```
# grep named /etc/rc*
/etc/rc.local: if [-f /usr/etc/in.named -1 -f /etc/named.boot]; then
in.named; echo -n ' named')  > /dev/console
```

- ❏ Make sure that the name server host's DNS database files are set up.

To make sure IEN 116 name service is available on the name server host:

- ❏ Verify that the daemon (most commonly, ***tnamed***) is configured on the name server host. You can use a command similar to the following to find the entry starting the daemon:

```
# grep tnamed /etc/inetd.conf
name dgram udp wait root /user/etc/in.tnamed in.tnamed
```

- ❏ IEN 116 uses the **/etc/hosts** file as its database. If the terminal is listed, no further database configuration is necessary.

Making Sure the Terminal Uses the Name Service

You should always place name service parameters at the beginning of a remote configuration file, before any parameters that use hostnames. In addition, insert an **apply** command after the name service parameters to make

sure the name service is in effect for subsequent parameters that use hostnames.

Selecting the Name Service Protocol

Set the **tcpip-name-server-protocol** parameter to the name service protocol you are using (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Name Server Protocol).This parameter is saved in NVRAM and takes effect immediately if set interactively.

Table 5-5 tcpip-name-server-protocol Parameter

Possible Values	Result
default	ien-116 (If the terminal's NVRAM is set to the factory defaults and the BOOTP/DHCP reply contains DNS name servers, the default value is "dns.")
ien-116	The terminal uses the IEN 116 name service method.
dns	The terminal uses DNS.
both	The terminal uses both IEN 116 and DNS.

Specifying Name Server Hosts

You can specify as many name server hosts as you need. If you do not specify a name server, the terminal uses the boot host for name service.

Enter the IP addresses of hosts offering name service into the **tcpip-name-servers** table (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Name Servers). If you enter an address of 0.0.0.0, the boot host is used for name service.

If you use BOOTP/DHCP and specify name servers for this terminal in the **bootptab** file, those name servers are placed in the table automatically. This parameter is saved in NVRAM.

Table 5-6 tcpip-name-servers Parameter

Table Entry	Possible Values	Result
server	default	0.0.0.0
	IP address	The name server with this IP address is used to map host IP addresses to symbolic names.

For example:

```
tcpip-name-servers = {
          { 192.43.150.001 }
          { 192.43.150.005 }
          }
```

Specifying the Default Domain Suffix for DNS

If you are using DNS and set this parameter, you do not have to specify a fully qualified domain name when specifying hostnames. Set the **tcpip-dns-default-domain** parameter to the default domain suffix to be appended to hostnames in searches (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ DNS Default Domain). The suffix is not appended to fully qualified names (names that contain all the components of the domain name).

The default domain suffix is the same as the domain name obtained from the Internet authority. For example, the parameter setting specifying NCD's default domain suffix is:

```
tcpip-dns-default-domain = ncd.com
```

Table 5-7 tcpip-dns-default-domain Parameter

Possible Values	Result
default	nil
nil	No suffix is applied to hostnames in name service searches.
domain suffix	The suffix applied to hostnames in name service searches.

Configuring the Name Translation Table (Local Name Cache)

Translations are automatically placed in the local name cache. If your site does not run a name service, you can place translations into the table manually.

The terminal maintains the local name cache in the **tcpip-name-local-cache** parameter (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Local Name Cache). Table 5-8 lists the entries in each row of the table.

Table 5-8 tcpip-name-local-cache Table Entries

Table Entry	Possible Values	Result
name	default	nil
	hostname	Hostname discovered through the name service or added manually.
address	default	0.0.0.0
	IP address	IP address corresponding to the hostname.
lifetime	default	0
	integer	The time (in seconds) (relative to the time the terminal was booted) at which the entry becomes invalid.

By default, case is considered when the terminal searches for a name in the local name cache. To ignore case, set the **tcpip-name-cache-ignore-case** parameter to "true." (Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Ignore Case on Name Cache Lookups).

To customize the local name cache, you can change the following parameters; however, the defaults work well for most sites.

Setting the Name Cache Entry Lifetime

The **tcpip-name-cache-max-lifetime** parameter controls the maximum amount of time that an entry in the cache is used before it is deleted automatically (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Name Cache Max Lifetime).

Table 5-9 tcpip-name-cache-max-lifetime Parameter

Possible Values	Result
default	1800
integer	The maximum lifetime (in seconds) of the name cache. Range: 0 - 4294967295.

Setting the Name Cache Size

The **tcpip-name-cache-max-size** parameter sets the maximum number of entries allowed in the name cache (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Name Cache Max Size). When the maximum size is reached, the oldest entry is discarded when a new entry is added. You may want to restrict the size of the cache to save memory for other uses.

Table 5-10 tcpip-name-cache-max-size Parameter

Possible Values	Result
default	32
integer	The maximum number of entries in the name cache. Range: 0 - 4294967295 or until all free memory is consumed.

Setting Name Service Timeouts

The **tcpip-name-server-retransmission-timeout** parameter determines how long the terminal waits before sending a retransmission if the name server does not respond (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Name Server Retransmission Timeout).

Table 5-11 tcpip-name-server-retransmission-timeout

Possible Values	Result
default	2
integer	How long to wait (in seconds) before retransmitting a name service request. Range: 1 - 4294967295.

The **tcpip-name-server-transaction-timeout** parameter determines how long the terminal waits for a response from the name server before failure is declared (Setup ⇒ Change Setup Parameters ⇒ TCP/IP Name Service ⇒ Name Server Transaction Timeout). The value of this parameter should be larger than the retransmission timeout.

Table 5-12 tcpip-name-server-transaction-timeout

Possible Values	Result
default	10
integer	How long (in seconds) to attempt a name service request before declaring a failure condition. Range: 1 - 4294967295.

Discovering the Terminal's Hostname when Booting (Reverse Name Request)

The **unit-query-for-name-at-boot** parameter controls whether, at boot time, the terminal sends a reverse name request to the DNS name servers to find the terminal's host name (Setup ⇒ Change Setup Parameters ⇒ Unit ⇒ Query for Unit Name at Boot).

Table 5-13 unit-query-for-name-at-boot Parameter

Possible Values	Result
default	none
none	The terminal does not attempt a reverse name query at boot.
tcpip	The terminal tries a reverse query to the TCP/IP name servers. If no name servers are defined, the terminal tries the TCP/IP boot server, if the boot server is defined. The name service protocol must be DNS.
ncdnet	The terminal tries a reverse query to the MOP boot server, if the boot server is defined.

Configuring How a Terminal Accesses Files

This section describes accessing files (other than the X server file) that are located on a network host. For information on local file service (accessing files on a PCMCIA card or local diskette), see the *System Administrator's Guide*.

Configuring the Initial File Servers

The initial file servers are used for loading configuration files, fonts, and the **rgb.txt** file when the terminal boots and for accessing files while the terminal is running. The initial file servers are automatically entered into the file service table described in "Configuring the File Service Table" on page 5-15.

The **file-initial-server-1** and **file-initial-server-2** remote configuration parameters permit you to define the primary and secondary initial file servers (Setup ⇒ Change Setup Parameters ⇒ File Service ⇒ Initial File Server 1, Initial File Server 2).

The **file-initial-protocol-1** and **file-initial-protocol-2** parameters specify the file access method for the initial file servers (Setup ⇒ Change Setup Parameters ⇒ Initial Protocol 1, Initial Protocol 2). The default file access protocol is TFTP. All of these parameters can be saved in NVRAM.

If you specify both file servers and the primary server is not available, the terminal tries to load its configuration file from the secondary server.

If one of the initial file servers is set to IP address 0.0.0.0, the boot host is used as the file server and is automatically entered into the file service table. Normally, you cannot use the boot host for both initial file servers. If the IP addresses of both initial file servers are set to 0.0.0.0, the secondary initial file server is ignored unless they are using different file service protocols.

Table 5-14 file-initial-server-1 Parameter

Possible Values	Result
default	0.0.0.0
0.0.0.0	The boot host is the initial file server.
IP address or ***hostname***	The primary initial file server.

Table 5-15 file-initial-server-2 Parameter

Possible Values	Result
default	0.0.0.0
0.0.0.0	The boot host is the secondary file server.
IP address or ***hostname***	The secondary initial file server.

Table 5-16 file-initial-protocol-1 Parameter

Possible Values	Result
default	tftp
tftp	Use the TFTP protocol for file access.
nfs	Use NFS for file access (using the UDP protocol).
nfs/tcp	Use NFS for file access (using the TCP protocol).
ncdnet	Use DAP for file access.

Table 5-17 file-initial-protocol-2 Parameter

Possible Values	Result
default	tftp
tftp	Use the TFTP protocol for file access.
nfs	Use NFS for file access (using the UDP protocol).
nfs/tcp	Use NFS for file access (using the TCP protocol).
ncdnet	Use DAP for file access.

Configuring the File Service Table

After loading an X server, the terminal uses its file service table, defined in the **file-service-table** parameter, for all file access (Setup ⇒ Change Setup Parameters ⇒ File Service ⇒ File Service Table). This table maps the default file locations known to the X server to the actual locations of files on file server hosts. The entries in each row of the file service table are described in Table 5-19.

By default, the terminal uses the boot host as the initial file server on which it searches for files (such as configuration files) during the booting process. After booting, the terminal also uses the boot host by default for all ongoing file requests.

If you have defined initial file servers, as explained in “Configuring the Initial File Servers” on page 5-13, the initial file servers are automatically placed in the file service table. If the terminal is accessing files only from these hosts and the boot host and the files are in their default locations, no further configuration of the file service table is necessary.

If files required by the terminal are not on the boot host or designated initial file servers or are not in their default locations, configure the file service table to map the default file access points known by the X server to the actual file access points and actual host.

The default file locations known to the X server are listed in Table 5-18.

Table 5-18 Default File Locations

File Type	Default Directory
Remote configuration files	**/usr/lib/X11/ncd/configs**
Color definition file (**rgb.txt**)	**/usr/lib/X11/ncd**
Fonts	**/usr/lib/X11/ncd/fonts**
Diagnostic log file	No default location
Keysym file (**XKeysymDB**)	**/usr/lib/X11/ncd**

Each entry in the table specifies a file server host, the file access point used by the terminal, the actual file access point on the file server, the protocol used, the retransmission and transaction timeout periods, and the amount of data transmitted on each read and write operation.

To change the file access point and the host:

1. Find the entry in the **file-service-table** parameter (see Table 5-19) that has the default location in the local-unix-mount-point field. For example, if you are placing the remote configuration files in a non-standard location, look for the default location **/usr/lib/X11/ncd/configs** in the local-unix-mount-point field. (In Setup ⇒ Change Setup Parameters ⇒ File Service ⇒ File Service Table, look for `Local UNIX Mount Point` with the default location.)
2. In the server mount point field, enter the actual file access point on the host. (In the File Service Table, click on the `Server Mount Point` entry you want to change, then type the actual file access point in the text entry box.)
3. If the actual file access point is on a host other than the boot host or an initial file server, enter the name or IP address of the host in the server field. (In the File Service Table, click on the `Server` entry you want to change, then type the name or IP address of the host in the text entry box.)

Note Local file systems are not entered into the file service table.

Table 5-19 file-service-table Parameter

Table Entries	Possible Values	Result
local-unix-mount-point	default	nil
	pathname	The terminal's local UNIX-style pathname for this file service access point.
local-vms-mount-point	default	nil
	pathname	The terminal's local VMS-style pathname for this file service access point.
server	default	nil
	network address or ***hostname***	The file server host.

Table 5-19 file-service-table Parameter (Continued)

Table Entries	Possible Values	Result
protocol	default	tftp
	tftp	TFTP is used for accessing files through this access point.
	nfs	NFS/UDP is used for accessing files through this mount point.
	nfs/tcp	NFS/TCP is used for accessing files through this mount point.
	ncdnet	NCDnet is used for accessing files through this mount point.
server-mount-point	default	nil
	pathname	Pathname for this file service access point on the file server host.
file-name-type (This field is not used if the protocol field is "nfs" or "nfs/tcp.")	default	unknown
	unknown	This value works for TFTP or DAP.
	unix	The file server uses UNIX-style filenames.
	vms	The file server uses VMS-style filenames.
retransmission-timeout	default	3
	integer	The amount of time (in seconds) between successive transmissions of a file service request. This is only used with file service protocols running over connectionless transports (for example, NFS or TFTP). Range: 0 - 4294967295.
transaction-timeout	default	30
	integer	The amount of time (in seconds) to attempt a file service request before a failure situation is declared. Range: 0 - 4294967295.

Table 5-19 file-service-table Parameter (Continued)

Table Entries	Possible Values	Result
read-size[1]	default	8192
	integer	The amount of data (in bytes) requested in a single read request from the terminal. This parameter is used with NFS, NFS/TCP, and TFTP. Values below 512 bytes cause noticeably slow performance. Range: 0 - 8192.
write-size [1]	default	8192
	integer	The amount of data (in bytes) requested in a single write request from the terminal. This parameter is only used with NFS or NFS/TCP. Values below 512 bytes cause noticeably slow performance. Range: 0 - 8192.

[1] If the terminal is having trouble reading files with NFS across gateways, try decreasing read-size and write-size to 1024 bytes.

An example file service table follows:

```
file-service-table = {
     {/usr/lib/X11/ncd/ nil eagle tftp /usr/local/lib/X11/ncd/ unknown 3
30 8192 8192}
     {/var/tmp nil eagle nfs /var/tmp unknown 3 30 8192 8192}
     }
```

Configuring the Matching Method

When attempting a file access, the terminal compares the file request with the local mount points in the file service table. By default, the terminal tries only the longest matching pathname (or pathnames, if there are matches of equal length). The longest match is the most complete match, the one that matches most or all of the elements in the pathname. You can configure the terminal to try all matching pathnames instead.

The **file-try-all-matches-on-open** parameter (Setup ⇒ Change Setup Parameters ⇒ File Service ⇒ Try All Matches on Open) controls how the terminal uses file service table entries when trying to access a file.

Table 5-20 file-try-all-matches-on-open Parameter

Possible Values	Result
default	false
false	The terminal tries only the longest matches.
true	The terminal tries all matching pathnames, beginning with the longest match.

The two methods of matching are explained in more detail in the following subsections.

Trying Only the Longest Matches

By default, the terminal tries only the longest matches. For example, assume that the pathname of a font requested by a client program is **/usr/lib/X11/ncd/fonts/pcf/100dpi/10x20.pcf**, and the file service table contains the following local mount points:

```
/usr/lib/X11/ncd
/usr/lib/X11/ncd
/usr/lib/X11/ncd
/usr
```

The first three local mount points match the request. The terminal tries the hosts in the order in which they are listed in the file service table, until it succeeds in opening the font file.

You may wish to have several longest matches to ensure that the terminal can always find the font or other data it needs.

Trying All Matches

If the terminal is configured to try all matches, the terminal first finds all the matches. After finding all the matching paths, the terminal sorts the mount points by length and tries the longest path first. If the file is not found there, the next longest is tried and so on. The root directory (/) matches any request.

For example, assume that the pathname requested by a client program is **/usr/lib/X11/ncd/fonts/100dpi/10x20.snf**, and the following local mount points are in the file service table:

```
/usr/lib/X11/ncd/fonts/100dpi
/usr/lib/X11/ncd/fonts
/usr/lib/X11/ncd
/usr
/
/ncd
```

The first five mount points match this request and the terminal.

Configuring File Access through TFTP

Terminals can use TFTP to download the X server and other files at boot and for ongoing file access.

NCD does not recommend using TFTP for writing to diagnostic log files.

TFTP is implemented by a daemon program, ***tftpd***(8), and configured in the boot host's **/etc/inetd.conf** file.

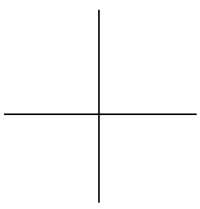
Secure versus Non-Secure TFTP

TFTP can run in two modes: secure mode (also called restricted mode) and non-secure mode.

Secure (Restricted) TFTP

Secure TFTP enhances security because it requires that the host perform a *change root* operation (***chroot***[8]) to the directory specified when TFTP is invoked. The directory specified when TFTP is invoked is TFTP's default home directory (usually **/tftpboot**). Because of the ***chroot***, all files to be accessed using secure TFTP (including X servers, fonts, and remote configuration files) must be physically installed under the directory and in the same file system partition. Symbolic links do not work.

If installing all files in the secure directory makes the directory too large, you can mount a file system partition, using the secure directory as the mount point. You could also use the secure directory only for X servers and use NFS as the access method for other files and fonts.



Non-Secure TFTP

Use non-secure TFTP when extra security is unnecessary. Non-secure TFTP is more flexible because ***chroot*** is not used. With non-secure TFTP, you can put X servers and modules in any directory. Note that when you use a non-standard directory for the X server or server modules, you must configure the terminal to find the files and configure the booting process to place the X server and modules in the desired location(s).

Make Sure TFTP Is Enabled on the Host

Consult your vendor documentation on how to make sure that TFTP is enabled. On some systems, you can use the following procedure:

1. Make sure the ***tftpd*** daemon has been installed and enabled. Usually, the daemon is enabled in the file **/etc/inetd.conf**; for example:

   ```
   tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd
   ```

   If a comment symbol (#) appears at the beginning of the entry, remove it. Always specify "wait" instead of "no wait." Otherwise, each ***tftpd*** request starts a new process, which can cause the host to start processes until it cannot start any more. If you specify "wait," each request is processed before another is serviced.

   Usually, ***tftpd*** runs under the user ID *root* as indicated in the example command line.

2. Make sure that the X server and module directories and other required files are world-readable.

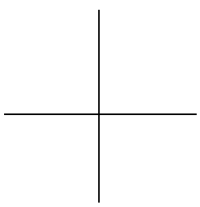
3. If you make any changes to the **/etc/inetd.conf** file, restart the ***inetd*** daemon to force it to reread the configuration file and start ***tftpd*** running.You can restart the daemon by finding its process id and sending it a hangup signal. For example:

   ```
   # ps -acx | grep inetd
   17601 ?  I 0:12 inetd
   # kill -HUP 17601
   ```

   On some systems, the command is **`ps -ef | grep inetd`**.

4. If you are using secure TFTP, make sure that all files to be accessed through TFTP are installed in the directory specified by the TFTP entry in the **/etc/inetd.conf** file. For example, on SunOS systems, the enabling line in **/etc/inetd.conf** for secure TFTP is:

```
tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd -s /tftpboot
```

This line makes it impossible for the NCD terminal to access fonts and configuration files because secure TFTP cannot reach **/usr/lib/X11/ncd**. This directory is outside the secure directory, which TFTP treats as its root (/) directory. One solution is to change `-s /tftpboot` to `-s /usr/tftpboot`. Restart the ***inetd*** daemon as directed in Step 3. Then move the X servers to **/usr/tftpboot** and move **/usr/lib/X11/ncd** to **/usr/tftpboot/usr/lib/X11/ncd**.

On HP-UX systems after Version 7, TFTP is secure; the TFTP daemon's home directory is the secure directory **/usr/tftpdir**. Any files that the terminal accesses via TFTP should be placed in this directory.

Configuring File Access through NFS

The terminal can use NFS for accessing all files and for downloading an X server. When accessing files through NFS, the X server temporarily mounts the file system onto its internal path.

Configuring the Host for NFS File Access

For files to be available through NFS, the host directories must be exported. This ensures that NFS clients, such as NCD terminals, can access the directories.

For example, on SunOS:

1. To export the default directory for X server files, add a line describing the directory in the **/etc/exports** file. For example:

   `/tftpboot/`

   or

   `/usr/tftpboot`

   Files can be exported to specific terminals, exported to everyone, or exported to *unknown*, the default name for an NCD terminal.
2. On the host where the directory resides, enter the following command:

   `# exportfs -a`

Setting User and Group IDs for NFS File Access

If the host exporting the file systems restricts mount requests to certain user or group IDs, set the **file-nfs-uid** and **file-nfs-gid** parameters to the relevant user ID (UID) and group ID (GID). These parameters are not available in the Setup menus.

The default value for both parameters is "-2", which corresponds to *nobody*. NFS handles requests that do not have a valid UID and GID by mapping them to the anonymous user. By default, the anonymous user is *nobody*. With user and group IDs of -2, files and directories must be world-readable and world-writable.

Table 5-21 file-nfs-uid Parameter

Possible Values	Result
default	-2
-2	Access is the same as the *world* permissions.
integer	The user ID of the requestor.

Table 5-22 file-nfs-gid Parameter

Possible Values	Result
default	-2
-2	Access is the same as the *world* permissions.
integer	The group ID of the requestor.

Setting the Unmount Timer for NFS File Access

The **file-nfs-unmount-timeout** parameter (Setup ⇒ Change Setup Parameters ⇒ File Service ⇒ NFS Unmount Timeout) controls how long to wait before unmounting file systems because of inactivity. The default is 1800 seconds (30 minutes). An unmounted file system is remounted the next time the terminal tries to access a file.

Table 5-23 file-nfs-unmount-timeout Parameter

Possible Values	Result
default	1800
integer	Timeout (in seconds) before file systems are unmounted due to inactivity. Range: 1 - 3600.

Changing the Timeout for Failed File Servers

The **file-failed-server-ignore-timeout** parameter (Setup ⇒ Change Setup Parameters ⇒ File Service ⇒ Failed Server Ignore Timeout) controls how long the terminal ignores a file server that has failed because of a network timeout error. When the terminal attempts to open a new file, it skips over the ignored servers.

The default timeout period is 120 seconds. A long timeout speeds up booting and session reset when the primary initial file server has failed.

Table 5-24 file-failed-server-ignore-timeout Parameter

Possible Values	Result
default	120
integer	The amount of time (in seconds) to ignore a file server that has failed because of a network timeout error. Range: 1 - 600.

Issuing Extended File Service Diagnostic Messages

The **file-extended-diagnostics** parameter (Setup ⇒ Change Setup Parameters ⇒ File Service ⇒ Extended Diagnostics) controls the extent of the file service diagnostics messages issued by the terminal. By default, a minimum number of messages are issued.

If you are having problems with the terminal accessing files, you can arrange to display more specific messages by setting this parameter to “true.”

Table 5-25 file-extended-diagnostics Parameter

Possible Values	Result
default	false
false	Minimal file service diagnostic messages are issued.
true	Extended file service diagnostic messages are issued.

Configuring Routing (Accessing Remote Networks)

If the terminal is communicating with remote networks, make sure that routes to other networks are set up and the subnet mask is set properly.

For most sites, you need only specify the default gateways described in this section. The terminal maintains current routes in the routing table described in "The IP Routing Table" on page 5-26.

Specifying Default Gateways

Default gateways are a reliable way to contact hosts outside the local network. If the terminal cannot find a usable route in the routing table, it contacts the default gateways. You specify the default gateways in the **ip-initial-default-gateway-1** and **ip-initial-default-gateway-2** parameters (Setup ⇒ Change Setup Parameters ⇒ IP ⇒ Initial Default Gateway 1, Initial Default Gateway 2). These parameters can be saved in NVRAM.

If the host named in the **ip-initial-default-gateway-1** parameter is not available, the terminal tries the host listed in **ip-initial-default-gateway-2**.

If one of the default gateways is 0.0.0.0, the terminal uses the boot host as the default gateway. If both gateways are set to 0.0.0.0, the second is ignored.

The default gateways are automatically placed in the routing table and are the initial entries in the table.

Although you can designate only two gateways using these parameters, you can specify as many additional gateways as you need by entering them into the routing table after the initial configuration file is loaded.

Table 5-26 ip-initial-default-gateway-1 Parameter

Possible Values	Result
default	0.0.0.0
0.0.0.0	The gateway is the boot host.
IP address or ***hostname***	A default gateway.

Table 5-27 ip-initial-default-gateway-2 Parameter

Possible Values	Result
default	0.0.0.0
0.0.0.0	The gateway is the boot host.
IP address or ***hostname***	A default gateway.

The IP Routing Table

NCD terminals maintain an internal routing table that contains current routes to remote hosts and networks. When attempting to reach a host outside the local network, the terminal tries the following methods of finding a route in the order given:

1. A route to the specific host
2. A route to the network or subnet the host is on
3. The default route, either as specified in **ip-initial-default-gateway** or obtained via router discovery
4. Proxy ARP, if enabled by **ip-use-proxy-arp**

The routing table can contain multiple routes to a single destination. If there is more than one matching route, the terminal uses the route with the greatest preference value.

The routing table changes over time due to normal operation. Routes are placed in the table by:

- ❏ Actions of network protocols (proxy ARP, router discovery, and ICMP redirects). See “Finding Routes to Hosts through Proxy ARP” on page 5-29 and “Discovering Neighboring Gateways through Router Discovery” on page 5-30.
- ❏ Default gateway parameter settings. These are the first entries in the table after the terminal boots. For more information about the default gateway parameters, see “Specifying Default Gateways” on page 5-25.
- ❏ The system administrator entering routes into the table manually. Manual changes to the routing table take effect immediately.

The IP routing table is defined in the **ip-routing-table** parameter (Change Setup Parameters ⇒ IP ⇒ Routing Table). Routing table entries include read-only

values, which you cannot modify, as well as read / write values. The entries in each row of the routing table are defined in Table 5-28.

Entries with the destination 0.0.0.0 are created from the **ip-initial-default-gateway-1** and **ip-initial-default-gateway-2** parameter settings.

Table 5-28 ip-routing-table Parameter

Table Entry	Possible Values	Result	Field Type
destination	default	0.0.0.0	read / write
	0.0.0.0	The entry is one of the default gateways.	
	IP address or ***hostname***	Address of the host network or name of the host.	
gateway	default	0.0.0.0	read / write
	IP address or ***hostname***	The IP address of the next hop on this route. If the route is bound to an interface that is realized through a broadcast medium, this field contains the agent's IP address on the interface.	
preference	default	0	read / write
	0	The midpoint of the preference range.	
	integer	Determines which route is preferred when there are multiple routes to a destination. Router discovery messages convey this information dynamically; otherwise, you can configure it statically. The terminal tries higher-numbered routes first. Range: -2147483648 to 2147483647.	

Table 5-28 ip-routing-table Parameter (Continued)

Table Entry	Possible Values	Result	Field Type
type	default	static	read / write
	static	The system administrator created the route and it cannot be deleted or marked unusable.	
	dynamic	The network discovered the route (by the proxy ARP, ICMP, or router discovery protocols) and it can be deleted or marked unusable if the terminal detects failures when using the route.	
creation-method	default	snmp	read / write
	snmp	SNMP set the route.	
	icmp	ICMP set the route.	
	local	The system administrator created the route.	
	proxy-arp	Proxy ARP created the route.	
birth	default	0	read-only
	integer	Amount of time (in seconds) after booting that the route was created.	
time-to-live	-1	The route should not be automatically deleted.	read / write
	integer	How long (in seconds) before the route is deleted. This information is conveyed in router discovery messages. Range: 1 - 2147483647	
destination-type	default	network	read / write
	network	The destination is a network. Most routes are network routes.	
	host	The destination is a host. Proxy ARP routes and the routes used for SLIP (Serial Line Internet Protocol) connections are host routes.	

Table 5-28 ip-routing-table Parameter (Continued)

Table Entry	Possible Values	Result	Field Type
route-mask	***hexadecimal number***	A hexadecimal value indicating the bits in the destination address used to determine the route. The mask is logically AND-ed with the destination address before being compared to the value in the gateway field. This field is used by SNMP (Simple Network Management Protocol).	read-only

An example routing table follows:

```
ip-routing-table = {
    { 0.0.0.0 gateway1.ncd.com -1 dynamic local 42 -1 network }
    { 127.0.0.1 127.0.0.1 0 static local 582 -1 host }
    { 0.0.0.0 eagle.ncd.com -1 dynamic local 42 -1 network }
    { 192.40.157.0 ncdu11.ncd.com 0 static local 39 -1 network }
    }
```

Finding Routes to Hosts through Proxy ARP

Proxy ARP resolves routes to hosts for which there are no routes in the IP routing table. It redirects the terminal's request to communicate with a host on another network to the gateway that provides the route to the host. Routes discovered through proxy ARP are automatically placed into the routing table and identified as dynamic routes. The terminal uses hosts specified in the default gateway parameters before resorting to proxy ARP.

To configure the terminal to use proxy ARP, set the **ip-use-proxy-arp** parameter to "true" (Setup ⇒ Change Setup Parameters ⇒ IP ⇒ Use Proxy Arp). This parameter is saved in NVRAM.

Table 5-29 ip-use-proxy-arp Parameter

Possible Values	Result
default	false
false	The terminal does not use proxy ARP to locate gateways.
true	The terminal uses proxy ARP to locate gateways.

Discovering Neighboring Gateways through Router Discovery

Router discovery is an extension to ICMP that enables hosts attached to multicast or broadcast networks to discover the IP addresses of neighboring routers (gateways). If the router discovery daemon is running on your network, you can use this method of discovering routes. The terminal automatically places the routes discovered in the routing table as dynamic routes.

To configure a terminal to use router discovery, make sure the **ip-use-router-discovery** parameter is set to "true" (the default) (Setup ⇒ Change Setup Parameters ⇒ IP ⇒ Use Router Discovery). This parameter can be saved in NVRAM.

Table 5-30 ip-use-router-discovery Parameter

Possible Values	Result
default	true
true	The terminal modifies its IP routing table with information received from router discovery messages.
false	The terminal does not modify its routing table with information received from router discovery messages.

If the terminal should use router discovery to solicit for routing information at boot time, make sure **ip-use-router-solicit** is set to "true" (the default) (Setup ⇒ Change Setup Parameters ⇒ IP ⇒ Use Router Solicit). This parameter can be saved in NVRAM.

Table 5-31 ip-use-router-solicit Parameter

Possible Values	Result
default	true
true	The terminal solicits for routing information.
false	The terminal does not solicit for routing information.

Setting TCP Performance Parameters

To customize the terminal's TCP interactions, you can change the TCP performance parameters described in this section.The default settings work properly for most installations. You should not need to adjust these parameters.

Caution

Setting these parameters incorrectly might cause your terminal to stop working and lead to excess network loading.

Adjusting the TCP Send and Receive Buffers

You can adjust the buffers used by the terminal in sending and receiving TCP packets. The default of 4096 bytes works well for sending images. For text-oriented clients, 2048 bytes works better.

These parameter settings should correspond to the TCP windows advertised by the host.

The **tcp-receive-buffer-size** parameter specifies the maximum amount of received data that a TCP connection buffers in the terminal (Setup ⇒ Change Setup Parameters ⇒ TCP ⇒ Receive Buffer Size). This corresponds to the TCP receive window advertised by the terminal to the peer (device on the other end of the connection).

Table 5-32 tcp-receive-buffer-size Parameter

Possible Values	Result
default	4096
integer	Size of the receive buffer (in bytes). Range: 1024 - 65535.

The **tcp-send-buffer-size** parameter specifies the maximum amount of data awaiting transmission that a TCP connection buffers in the terminal (Setup ⇒ Change Setup Parameters ⇒ TCP ⇒ Send Buffer Size). This corresponds to the maximum amount of the peer's TCP send window that is used by the terminal.

Table 5-33 tcp-send-buffer-size Parameter

Possible Values	Result
default	2048
integer	Size of the send buffer (in bytes). Range: 1024 - 65535.

Specifying the TCP Timeout

The **tcp-connect-timeout** parameter specifies the amount of time that must elapse between a TCP connection attempt and a lack of response before failure is declared (Setup ⇒ Change Setup Parameters ⇒ TCP ⇒ Connect Timeout).

Table 5-34 tcp-connect-timeout Parameter

Possible Values	Result
default	75
integer	Elapsed time (in seconds) between TCP connection attempts before failure is declared. Range: 1 - 4294967295.

Specifying TCP Maximum Retransmissions

The **tcp-max-retransmissions** parameter specifies the number of retransmissions on a TCP connection before failure is declared (Setup ⇒ Change Setup Parameters ⇒ TCP ⇒ Maximum Retransmissions).

Table 5-35 tcp-max-retransmissions Parameter

Possible Values	Result
default	12
integer	Retransmissions on a TCP connection before failure is declared. Range: 1 - 4294967295.

Specifying the TCP Linger Time

The **tcp-default-linger-time** parameter specifies the default time interval during which TCP attempts reliable transmission of outstanding data on the connection's transmit queue after local software closes a connection (Setup ⇒ Change Setup Parameters ⇒ TCP ⇒ Default Linger Time). This timer is optional and higher-level software can configure it for each connection.

Table 5-36 tcp-default-linger-time Parameter

Possible Values	Result
default	120
integer	Time (in seconds) that TCP continues to attempt transmission after the local software closes the connection. Range: 1 - 4294967295.

Allowing Larger Segment Sizes

When set to "false," the **tcp-default-mss-for-non-local** parameter allows segment sizes larger than the default to be used when communicating with non-local hosts (that is, hosts on the other side of a gateway). The default segment size is 536 bytes (Setup ⇒ Change Setup Parameters ⇒ TCP ⇒ Use default maximum segment size for non-local hosts).

Table 5-37 tcp-default-mss-for-non-local Parameter

Possible Values	Result
default	true
true	The default segment size, 536 bytes, is used when communicating with non-local hosts.
false	Segment sizes larger than the default can be used when communicating with non-local hosts.