Dialnet

User's Manual

Part No. 950-1150B Printed in U.S.A. 8/92

REVISION HISTORY

First Edition.....Release 2.2.....10/91

Second Edition.....Release 2.3..... 8/92

NOTICE

The information contained in this document is subject to change without notice.

Livingston Enterprises, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Livingston Enterprises, Inc. shall not be liable for errors contained herin or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Livingston Enterprises, Inc.

Copyright © 1991, 1992 by Livingston Enterprises, Inc.

TRADEMARKS

PortMaster is a trademark of Livingston Enterprises, Inc.

OpenWindows is a trademark of Sun Microsystems, Inc.

Open Look and UNIX are trademarks of AT&T.

Table of Contents

1 Introduction

Features..... 1-1
About This Manual..... 1-4
Requesting Assistance..... 1-5

2 Quick Start Guide

Common Uses of Dialnet..... 2-1
Connecting Networks..... 2-3
Dial-In Hosts..... 2-6
Hardwired Connections..... 2-9

3 Administration

PortMaster Console..... 3-1
Dialnet with X Windows..... 3-2
Global Configuration..... 3-2
Serial Port Configuration..... 3-6
Port Parameters..... 3-7
Hardwired Dialnet Parameters.....3-10
Dial Out Configuration Parameters..... 3-11
Enabling Serial Port Parameters..... 3-12
Managing the Location Table.....3-12
Location Table Parameters..... 3-14
Managing the Passwords Table..... 3-19
Password Parameters..... 3-22
The Hosts Table.....3-24
The Routes Table.....3-25
Dialing Out to Remote Locations.....3-25
Monitoring Dialnet Connections..... 3-28
Dialnet with an ASCII Terminal.....3-31
Global Configuration.....3-31
Serial Port Configuration.....3-36
Port Parameters.....3-38
Hardwired Dialnet Parameters.....3-40
Dial Out Configuration Parameters..... 3-42
Enabling Serial Port Parameters.....3-42
Managing the Location Table.....3-42
Location Table Parameters..... 3-44
Managing the Passwords Table..... 3-49
Password Parameters..... 3-52
The Hosts Table.....3-55
The Routes Table.....3-56
Dialing Out to Remote Locations.....3-56
Monitoring Dialnet Connections..... 3-58
Command Prompt Interface.....3-62
The User Commands..... 3-62
Using the Commands.....3-63
SET Command.....3-63
Setting Global Parameters..... 3-63

Setting Dialnet Port Parameters.....3-64
Setting Hardwired Dialnet Parameters.... 3-65
SHOW Command.....3-67
SAVE Command..... 3-69
HELP Command.....3-70
IFCONFIG Command..... 3-71
DIAL Command..... 3-71
Managing the Location Table.....3-72
ADD Command.....3-72
DELETE Command.....3-72
SET Command.....3-72
SHOW Command.....3-78
SAVE Command..... 3-78
Managing the Passwords Table..... 3-78
ADD Command.....3-80
DELETE Command.....3-80
SET Command.....3-80
SHOW Command.....3-83
SAVE Command..... 3-83
The Hosts Table.....3-83
The Routes Table.....3-84

4 Networking Primer

Introduction..... 4-1
What is TCP/IP ?..... 4-2
Description of TCP/IP Protocols..... 4-9
The TCP Level.....4-11
The IP Level.....4-14
The Ethernet Level.....4-15
Well-known Sockets and Applications.....4-17
An example application: SMTP.....4-21
Protocols other than TCP.....4-24
Keeping track of names and information.....4-25
Routing.....4-28
Details about Internet Addresses..... 4-30
Datagram fragmentation and reassembly..... 4-33
Ethernet encapsulation: ARP.....4-34
Getting more information..... 4-35

Appendixes

A UNIX Utilities

Pmcommand..... A-1
Pmdial..... A-1
Pmreset..... A-2

B Command Reference

Introduction_____CHAPTER 1

Dialnet for the PortMaster Communications Server adds advanced wide area networking capability to large and small organizations. By using modems and standard telephone lines, local area networks can be economically and seamlessly connected. Dialnet uses the industry standard protocols; Serial Line Internet Protocol (SLIP) and Point to Point Protocol (PPP) to turn the PortMaster into a full IP router. This makes it compatible with all computers and networks which use the TCP/IP protocol. In addition, by incorporating an on-demand feature, dial-up telephone lines are only in use when there is data to be sent between networks. This reduces overall operating costs and increases the number of locations which can be tied together through a small number of modems.

The PortMaster's advanced communications capabilities which support modems, printers, terminals and other serial data-communications devices are enhanced by Dialnet. With Dialnet a single modem can be used for file transfer (ie. uucp, kermit, ymodem, xmodem), dial-in or dial-out access, and wide area networking.

Features

Dialnet provides the following features:

On-demand Dialing - Dialnet monitors internetwork traffic requirements and automatically dials, connects, and initiates network connections when packets are queued for the remote network. This results in low telephone connect charges by only using telephone services when required.

Scheduled Dialing - Dialnet can be configured to initiate network connections only during specific times of the day or days of the week. This allows network administrators to take advantage of lower telephone rates or higher user demands as needed.

Multi-line Load Balancing - Dialnet can be configured to automatically initiate additional dial lines based on network traffic loads. Once

established, Dialnet provides load balancing across the concurrent links. This allows large file transfers or large numbers of users to achieve faster response from the network. Lines are then automatically disconnected when the traffic levels fall below minimums.

Dynamic Routing - Dialnet uses the industry standard Routing Information Protocol (RIP) to automatically establish correct routing of packets through the network. This feature also supports automatic re-routing of packets if a particular segment of the network goes down. This feature simplifies network administration and provides a high level of reliability for the entire network.

Static Routing - Dialnet provides optional user controlled routing entries. This feature is useful for networks where all systems don't support the RIP protocol, or to limit the availability of network connections to specific systems on the network. In addition, default route support is supplied for routing to external networks.

Selective Route Notification - Each network interface can be configured to either forward routing information to the remote site or operate quietly. When forwarding routing information, all destinations at both sites become visible throughout the network. When operating quietly, only static routing entries will be visible to the network. This can be used for increased network security, or to decrease the size of local routing tables.

Extensive Modem Support - By combining the built-in modem control capabilities of the PortMaster (including hardware flow control) and Dialnet's automated dialing scripts, a wide variety of modems are supported. The dialing scripts use the send-expect approach already found in utilities like UUCP. These features allow the user to select the appropriate modem for their specific application, and often allows users to continue using modems they have already purchased.

Full Login Security - For Dial-in network access, Dialnet provides a full password security system. The remote system or user is prompted for a login name and password. These are validated against a local security file. Once validated, the settings previously defined by the

system administrator become active. These options include; specifically assigning an IP address, disconnecting to begin a dial-back, setting various communications parameters, assinging the session a temporary IP address, or allowing the remote side to negotaite IP addresses. These features assure a high degree of flexibility and security.

Flexible Port Assignments - The mix of dial-in and dial-out services available on each port are fully user configurable. This includes modem pool groups, outbound connections on specific lines, inbound connections on specific lines, and setting the mix of standard terminal server features and network features (ie. login, host device, network dial-in, network dial-out, or hardwired). This allows the user to configure the PortMaster to suit the specific needs of their environment.

Network Information Service - Host name resolution is fully supported by utilizing the Network Information Service (NIS), also known as YP. This service can then be backed up by the industry standard Domain Name Service (DNS). In addition, a local hosts file can be maintained.

Advanced Administration - All features of Dialnet are accessible through the standard PortMaster administration software "PMconsole".

Industry Standard Protocols - Dialnet uses industry standard protocols to allow for interoperability with other vendors products and to deliver advanced technology. These include:

- Routing Information Protocol (RIP)
  - RFC 1058 with RIP Extensions
- Internet Protocol (IP)
  - RFC 791
- Internet Control Message Protocol (ICMP)
  - RFC 792
- User Datagram Protocol (UDP)
  - RFC 768
- Transmission Control Protocol (TCP)
  - RFC 793

- Internet Subnetting Procedures
  - RFC 950
- Address Resolution Protocol (ARP)
  - RFC 829
- Simple Network Management Protocol (SNMP)
  - RFC 1155, 1156, 1157
- Telnet Protocol (TELNET)
  - RFC 854
- Serial Line IP Protocol (SLIP)
  - RFC 1055
- Point to Point Protocol (PPP)
  - RFC 1331,1332

About This Manual

This manual is intended for use by those who are familiar with the use of the PortMaster (see the PortMaster User's Manual), Ethernet, TCP/IP, and networking. The manual is organized into four chapters:

Chapter 1: *Introduction* - describes the features of Dialnet.

Chapter 2: *Quick Start Guide* - describes the steps to follow to implement several of the most common Dialnet configurations.

Chapter 3: *Administration* - describes the process used for configuring the communications ports and Dialnet specific tables. This includes:

- The X windows PortMaster Console program.
- The ASCII Terminal PortMaster Console program.
- The Command Prompt Interface.

Chapter 4: *TCP/IP Networking Primer* - provides an overview on TPC/IP networking as an aid to system administrators who are new to wide area networking or TCP/IP in general.

Appendix A: *UNIX Utilities* - Describes extra utilties provided with the PortMaster software distribution.

Appendix B: *Command Reference* - Summary of the commands available in the Command Prompt Interface.

Requesting Assistance

If you have any questions about Dialnet or the PortMaster, please use the following, toll-free number:

1-800-458-9966

Outside of the United States call:

510-426-0770

Quick Start Guide _____CHAPTER 2

This guide provides a step by step checklist for the three most common types of Dialnet connections. By selecting the type of connection which best represents your planned use of Dialnet, and then following the checklist, you can bring Dialnet to an operational state in the fewest number of steps.

Before continuing with this section you should be familiar with TCP/IP, wide area networking, and basic routing strategies. If you are not familiar with these concepts Chapter 4 of the manual is a TCP/IP Networking Primer. The primer has three sections directly related to using Dialnet in a wide area network. They are:

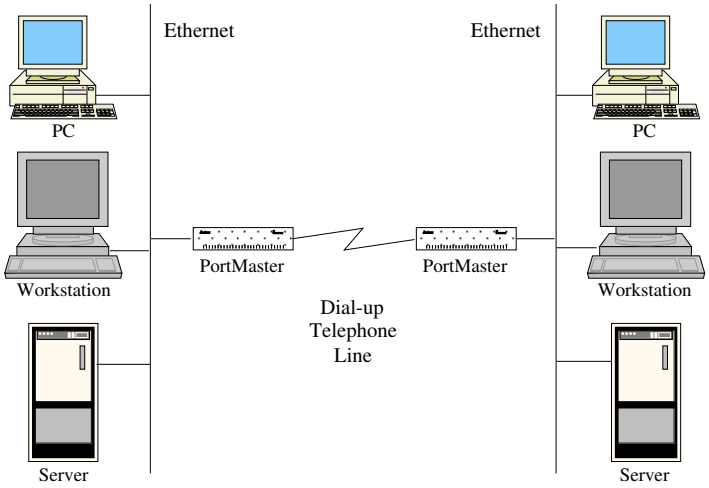
The IP Level - How datagrams are forwarded to the remote computer.

Routing - How gateways forward IP datagrams.

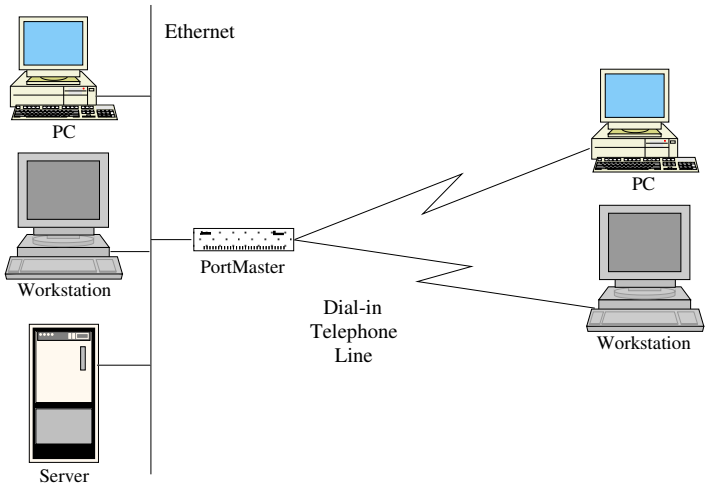
Details about Internet addresses - How Internet addresses are selected and used.

Common Uses of Dialnet

Connecting Networks - One of the most common uses of Dialnet is to establish wide area network connections between local area networks at geographically dispersed sites. The diagram in figure 2-1 shows this configuration.



Dial-in Hosts - Dialnet can be used for individual hosts to gain access to their central network. This is common for UNIX Workstations, or PC's which support TCP/IP and the SLIP or PPP protocol. These are typically used for home systems (ie. tele-commuting) or for portable systems which need dial-up access to electronic mail or network file transfer. The diagram in figure 2-2 shows this configuration.



Hardwired Connections - Either of the previous types of connections can be configured to run over dedicated leased lines. In this mode, it may be desirable to eliminate dialing and login security steps and assume a direct connection is always available. This mode of operation also decreases the configuration requirements for the connection.

Each of the above connection types is described in the following three sections. Proceed to the section which best matches your implementation requirements and follow the recommended installation steps. If none of these configurations matches your requirements, use them as a reference and proceed to Chapter 3 "Administration" for a detailed description of each configuration option.

Connecting Networks

This section describes the steps for connecting two networks together over dial-up telephone lines for on-demand usage. On-demand means that the PortMaster automatically dials and establishes the network link when packets are queued for the remote site. This section assumes that a PortMaster will be used at each site, and modems which are compatible with each other will be used. This example can be used for connecting more than two sites together by simply following the same steps for each site.

Internet Addresses

When connecting two sites which have never directly communicated with each other using TCP/IP it is important to perform some general Internet Address Administration. The steps for this are:

❏ Verify that every computer on both networks has a unique Internet Address.

❏ Verify that the network (or subnet) numbers at each site are unique. If the two locations are using the same network numbers datagram routing will not work correctly. In this case, one of the networks must be changed.

❏ Create a common "hosts" file for both sites so users can access the

computers at the remote location by name. In a hosts file, no two computers should have the same name. Alternatively, Domain Name Service can be implemented. Check your workstation, or TCP/IP vendor's manuals for information on this service.

Hardware Configuration

The PortMaster should be connected to the local ethernet network and the installation steps in Chapter 2 of the PortMaster User's Manual should be followed. The modem should be attached to one of the ports on the PortMaster with a standard modem cable. The modem should also be connected to a direct dial telephone line.

Every modem manufacturer supports a wide variety of configuration settings. Experience has shown that the following configuration parameters are the most important, however the modem manufacturer's manual should be carefully reviewed if the following parameters are not sufficient. The modem should be configured as follows:

❑ Fixed baud rate (usually the fastest speed the modem will run up to 115,200 baud).

❑ RTS/CTS flow control.

❑ 8 databits, 1 stop bit, no parity.

❑ Normal carrier detect.

❑ Hang-up and reset when DTR drops.

Global Configuration

The PortMaster Global Configuration should be checked. This should be done using the PMconsole program. See the section on Global Configuration in Chapter 3 for instructions on editing the global configuration.

❑ Routing should be set to "Broadcast" and "Listen".

❑ The broadcast address should be checked against the one being used by other gateways on the local ethernet.

❑ Dialnet should be "Enabled".

Port Configuration

Each Serial port which will has a modem attached, and will be used for Dialnet connections should be configured. See the section on Setting the Serial Port Configuration in Chapter 3 for instructions on editing the serial ports.

❑ The serial port should be set to "Extended" Mode.

❑ The Port Type should be set to "Dialnet".

❑ The Dialnet Service should be set to "Dial In&Out".

❑ If the port will be used for other services (ie. TwoWay) this port type should be selected in addition to "Dialnet".

❑ The Baud Rates, Modem Control, and Flow Control should be set to match the modem.

❑ The Dial Group should be set to "0" (the default).

Password Table Entry

The Password Table is used for all dial-in connection establishment. A password entry should be created on the PortMaster's at each site to establish login security for the remote PortMaster. Follow the instructions on "Managing the Password Table" in Chapter 3 to set up a new password.

Location Table Entry

The Location Table is used for dialing out from the local PortMaster to a remote site. A location entry should be created on each PortMaster referencing the PortMaster (and network) at the other site. Follow the instructions on "Managing the Location Table" in Chapter 3 to set up a new location. For testing purposes, the location type should be set to manual until the dial command script has been properly tested.

Testing

To test a new connection:

❑ Attempt to dial from one PortMaster to the other PortMaster using

the Dialer screen or the dial command. The use of this screen is described in Dialing Out to Remote Locations in Chapter 3. The dial command is described in the Command Prompt Interface in Chapter 3.

❑ Make any necessary changes to the Dial Command Script and retry dialing until the connection succeeds.

❑ Once the connection is successful verify that the remote PortMaster is accessible through the network (ie. "ping" the remote PortMaster).

❑ Verify that a computer at the remote location is accessible (ie. use rlogin, ping, or telnet).

❑ Reset the serial port to drop the line and perform the above steps dialing in the opposite direction.

❑ With the serial ports on each PortMaster reset (ie. no active connection) switch the location types to On-Demand.

Dial-In Hosts

This section describes the steps for enabling dial-in network connections for remote computers. Users who wish to use this type of connection must have software on their workstation or PC which allows them to issue dial commands to their modem, respond to login and password prompts from the PortMaster for security, and switch over to the SLIP or PPP protocol for sending and receiving TCP/IP datagrams.

Internet Addresses

When connecting a remote computer into the network through a PortMaster, that computer must be assigned a valid Internet address on the same network to which the PortMaster is attached. For example if the PortMaster is on network 192.200.1 and has the internet address 192.200.1.20, the remote computer should be assigned a free address on the 192.200.1 network (ie. 192.200.1.21).

❑ Verify that the remote computer has a unique Internet Address.

❑ Update the global "hosts" file so users can access the computer at the

remote location by name and install the global "hosts" file on the remote computer. Alternatively, Domain Name Service or Network Information Service can be implemented. Check your workstation, or TCP/IP vendor's manuals for information on these service.

Hardware Configuration

The PortMaster should be connected to the local ethernet network and the installation steps in Chapter 2 of the PortMaster User's Manual should be followed. The modem should be attached to one of the ports on the PortMaster with a standard modem cable. The modem should also be connected to a direct dial telephone line.

Every modem manufacturer supports a wide variety of configuration settings. Experience has shown that the following configuration parameters are the most important, however the modem manufacturer's manual should be carefully reviewed if the following parameters are not sufficient. The modem should be configured as follows:

❑ Fixed baud rate (usually the fastest speed the modem will run up to 115,200 baud).

❑ RTS/CTS flow control.

❑ 8 databits, 1 stop bit, no parity.

❑ Normal carrier detect.

❑ Hang-up and reset when DTR drops.

Global Configuration

The PortMaster Global Configuration should be checked. This should be done using the PMconsole program. See the section on Global Configuration in Chapter 3 for instructions on editing the global configuration.

❑ Routing should be set to "Broadcast" and "Listen".

❑ The broadcast address should be checked against the one being used by other gateways on the local ethernet.

❏ Dialnet should be "Enabled".

Port Configuration

Each Serial port which will has a modem attached, and will be used for Dialnet connections should be configured. See the section on Setting the Serial Port Configuration in Chapter 3 for instructions on editing the serial ports.

❏ The serial port should be set to "Extended" Mode.

❏ The Port Type should be set to "Dialnet".

❏ The Dialnet Service should be set to "Dial In".

❏ If the port will be used for other services (ie. User Login) this port type should be selected in addition to "Dialnet".

❏ The Baud Rates, Modem Control, and Flow Control should be set to match the modem.

❏ The Dial Group should be set to "0" (the default).

Password Table Entry

The Password Table is used for all dial-in connection establishment. A password entry should be created on the PortMaster to establish login security for the remote computer. Follow the instructions on "Managing the Password Table" in Chapter 3 to set up a new password.

Testing

To test a new connection:

❏ Attempt to dial from the remote computer into the PortMaster. Monitor the serial port, and the network statistics window using PMconsole during the login process. The use of this screen is described in Monitoring the PortMaster in Chapter 3.

❏ Verify that the remote computer has correct routing table entries. This can be done by creating a default route or using dynamic routing (RIP) to the PortMaster. Default routes are the easiest to use and are generally settable through the "route" command in

UNIX and other implementations.

❏ Once the connection is successful verify that the PortMaster is accessible to the remote computer through the network (ie. "ping" the PortMaster).

❏ Verify that a computer at the central site is accessible (ie. use rlogin, ping, or telnet).

Hardwired Connections

This section describes the steps for enabling hardwired SLIP or PPP connections for remote computers or networks. Users who wish to use this type of connection must have a PortMaster at each end of the connection or a PortMaster on one end and a computer on the remote end which supports the SLIP or PPP protocol.

Internet Addresses

When connecting a remote computer into the network through a PortMaster, that computer must be assigned a valid Internet address on the same network to which the PortMaster is attached. When connecting two sites which have never directly communicated with each other using TCP/IP it is important to perform some general Internet Address Administration.

❏ Verify that every computer on both networks has a unique Internet Address.

❏ If two networks will be connected, verify that the network (or subnet) numbers at each site are unique. If the two locations are using the same network numbers datagram routing will not work correctly. In this case, one of the networks must be changed.

❏ Create a common "hosts" file for both sites so users can access the computers at the remote location by name. In a hosts file, no two computers should have the same name. Alternatively, Domain Name Service can be implemented. Check your workstation, or TCP/IP vendor’s manuals for information on this service.

Hardware Configuration

The PortMaster should be connected to the local ethernet network and the installation steps in Chapter 2 of the PortMaster User's Manual should be followed. The modem should be attached to one of the ports on the PortMaster with a standard modem cable. The modem should also be connected to a direct dial telephone line.

Every modem manufacturer supports a wide variety of configuration settings. Experience has shown that the following configuration parameters are the most important, however the modem manufacturer's manual should be carefully reviewed if the following parameters are not sufficient. The modem should be configured as follows:

❑ Set the type of telephone line to dedicated leased line.

❑ Fixed baud rate (usually the fastest speed the modem will run up to 115,200 baud).

❑ RTS/CTS flow control.

❑ 8 databits, 1 stop bit, no parity.

❑ Normal carrier detect.

❑ Hang-up and reset when DTR drops.

Global Configuration

The PortMaster Global Configuration should be checked. This should be done using the PMconsole program. See the section on Global Configuration in Chapter 3 for instructions on editing the global configuration.

❑ Routing should be set to "Broadcast" and "Listen".

❑ The broadcast address should be checked against the one being used by other gateways on the local ethernet.

❑ Dialnet should be "Enabled".

Port Configuration

Each Serial port which will has a modem attached, and will be used for

Dialnet connections should be configured. See the section on Setting the Serial Port Configuration in Chapter 3 for instructions on editing the serial ports.

❑ The serial port should be set to "Extended" Mode.

❑ The Port Type should be set to "Dialnet".

❑ The Dialnet Service should be set to "Hardwired".

❑ The protocol, mtu, destination, netmask, and routing should be set to match the remote location.

❑ The Baud Rates, Modem Control, and Flow Control should be set to match the modem.

Testing

To test a new connection:

❑ Monitor the serial port, and the network statistics window using PMconsole to verify that the connection is active. The use of these screens is described in Monitoring the PortMaster in Chapter 3.

❑ Verify that routing tables are properly configured. (ie. static routes or dynamic routing (RIP). If using static routing, static route entries will need to be entered into the Routes Table. See the section Managing the Routes Table in Chapter 3 for more information.

❑ Verify that the PortMaster is accessible to the remote computers through the network (ie. "ping" the PortMaster).

❑ Verify that computers at each site are accessible (ie. use rlogin, ping, or telnet).

Administration_____CHAPTER 3

PortMaster Console

PortMaster Console (PMconsole) is the PortMaster administration program. The program uses one of two user interfaces; X windows, or a full screen ASCII terminal. The graphical interface provides several windows through which you can configure the PortMaster. The terminal interface provides multiple screens for performing the same functions.

If your site does not have a host computer which supports the PMconsole application, the Command Prompt Interface may be used. This interface is accessible by using the telnet protocol to login as the administrator via the network, or by attaching a terminal or modem to the PortMaster.

This chapter details the use of PMconsole when configuring and monitoring Dialnet connections. It is intended to augment the information already contained in the PortMaster User's Manual. You should already be familiar with the general use of the PMconsole administration software prior to beginning this section. Refer to the PortMaster User's Manual for information on using the PortMaster as a terminal or device server. The last section in this chapter provides information on using the Command Prompt Interface.

The next three sections detail the use of each of the three available user interfaces. Please move ahead to the section which matches your operating environment.

Dialnet with X Windows

The X windows graphical interface provides several windows through which you can configure the PortMaster ports, monitor the performance, and install other PortMasters on your network.

Each window in the PortMaster Console has a set of buttons. To select a button, position the mouse cursor inside the button you wish to select and press the left mouse button. Buttons with a triangle have pull-down menus. To display the menu, press and hold down the right mouse button. The menu will be displayed. Move the mouse pointer over the desired selection and release the button. Windows can be re-positioned as with other X windows applications.

Global Configuration

The Global configuration window is used to set parameters which are common to all ports and operations of the PortMaster. This window should be configured prior to setting any other Dialnet configuration parameters. From the PortMaster Console, do the following:

1. Position the cursor on the Network icon (upper right corner, above the Device P0), and click the left mouse button. The Network icon will change to reverse video.

2. Select the EDIT button. The window in Figure 3-1 will be displayed.

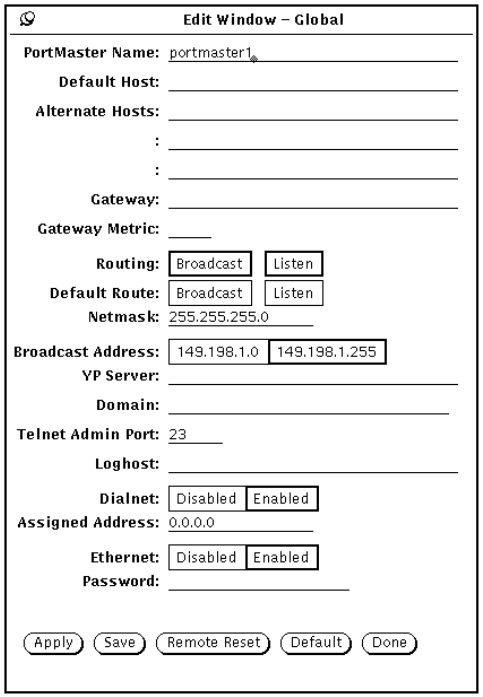


Figure 3-1. Edit Window - Global

3. Enter the following information, terminated by pressing RETURN:

- **PortMaster Name** - name of PortMaster as entered during the installation procedure.
- **Default Host** - This field is not used by Dialnet, however, when using Dialnet in combination with the PortMaster terminal server features, this should be the name of the primary host which the PortMaster will be connected to during operation.
- **Alternate Hosts** - This field is not used by Dialnet, refer to the PortMaster User's Manual for more information on the use of this field.
- **Gateway Name** - If the PortMaster is being connected to a network with gateway systems, then the name of the default gateway on the

same network should be entered. The PortMaster automatically routes network packets through gateways which are running the Routing Information Protocol (RIP), but will use the default gateway if no RIP gateways are found.

- **Gateway Metric -** If the Gateway field above is being used, this value (from 1-15) is the cost associated with using the default gateway route. This is useful if the PortMaster is configured to listen for default routes coming from other routers. Otherwise, this field can be left blank.
- **Routing** - This indicates whether the PortMaster should use the RIP protocol through the Ethernet Interface. Any combination of "Broadcast" and "Listen" can be used. If "Broadcast" is selected, the Portmaster will send RIP information to other computers attached to the local Ethernet. If "Listen" is selected, the PortMaster will accept RIP information from other routers on the local Ethernet. The default is to both Broadcast and Listen.
- **Default Route** - This option works in conjunction with the "Routing" field above. Any combination of "Broadcast" and "Listen" can be used. If "Broadcast" is selected, the Portmaster will send default route information as part of its normal RIP messages. If "Listen" is selected, the PortMaster will accept default route information coming from other routers on the local Ethernet. The default is to not Broadcast or Listen .
- **Netmask** - defaults to 255.255.255.0. This should only be modified if your site uses subnetting or Class A or B IP addresses. Consult you network administrator for this information.
- **Broadcast Address** - This is the IP address which the PortMaster will interpret and use as the local broadcast address. This is used primarily by the RIP protocol and some diagnostic protocols. This value should match other computers on the same local Ethernet.
- **YP Server** - For networks which use Sun's Network Information Service (NIS), also known as YP, the PortMaster can be configured to use this service for host name matching. If this service is being used, enter the hostname of the YP server which the PortMaster

should use.

- **Domain** - If a YP Server is being used, the correct YP domain name must be entered.
- **Telnet Admin Port** - The PortMaster allows the system administrator to access all of the PortMaster administration functions through a command line interface. These functions can be reached by logging onto the PortMaster with a telnet session. This field identifies the specific TCP port number for the PortMaster to listen on for incoming telnet administration sessions. The default value is "23". Any number from 0 to 65536 may be used. If 0 is selected, the PortMaster will disable the telnet administration function.
- **Loghost** - The PortMaster supports the "syslog" network accounting function. If a host is entered in this field, the PortMaster will log user authentication information to this host using the system facility "auth" and the priority of "info" (auth.info). Consult your system's syslogd and /etc/syslog.conf manual pages for more information on these services.
- **Dialnet** - This field enables the Dialnet features of the PortMaster. The configuration options for the serial ports are modified by enabling or disabling Dialnet.
- **Assigned Address** - This field is used for network login users using the PPP protocol. If a network user in the passwords table has been set to use an "Assigned Address", the PortMaster must allocate a temporary IP address to be used for their active session. This IP address is retrieved from a pool of addresses maintained by the PortMaster. The IP address entered in this field will be used as the first free address in the pool of addresses. The PortMaster will allocate as many addresses as there are ports configured for Dialnet access.
- **Ethernet** - This field enables the Ethernet interface of the PortMaster. It should only be disabled when the PortMaster is not being attached directly to the local Ethernet.
- **Password** - PortMaster administration password.

As each of the first three items is entered, it will be validated against your systems global host table (/etc/hosts or Yellow Pages). These were established by PMinstall if this is a stand-alone installation, or by your network administrator. The Netmask should not be changed unless you are using sub-nets and understand their operation. Refer your UNIX system documentation for more information on sub-nets. The password entry requires being entered twice, to ensure that the password was entered correctly. The password can be changed at any time by following the procedures of this section.

4. Select the SAVE button to store the data in the PortMaster. At the bottom of the window will be a message:

Configuration successfully saved.

5. Select the DONE button. At this point, the PortMaster has all the information necessary to begin normal communications with the network.

Serial Port Configuration

The PortMaster serial ports can be configured for Dialnet operation only, terminal server features only, or a combination of the two. These features are chosen by selecting the appropriate port type for each of the serial ports. User Login, Host Device, and TwoWay operation are described in the PortMaster User's manual.

To set the configuration for the serial ports, use the following general procedure:

1. From the PortMaster Console window, select the desired serial port icon by placing the mouse cursor on the icon and clicking the left button.

2. Select the EDIT button. The Edit Window for the serial port will

appear. From this window you can modify the serial port configuration parameters.

3. Once the Edit Window for the port has been displayed, the appropriate port type may be selected. If Dialnet is selected, the window in figure 3-2 will be displayed.

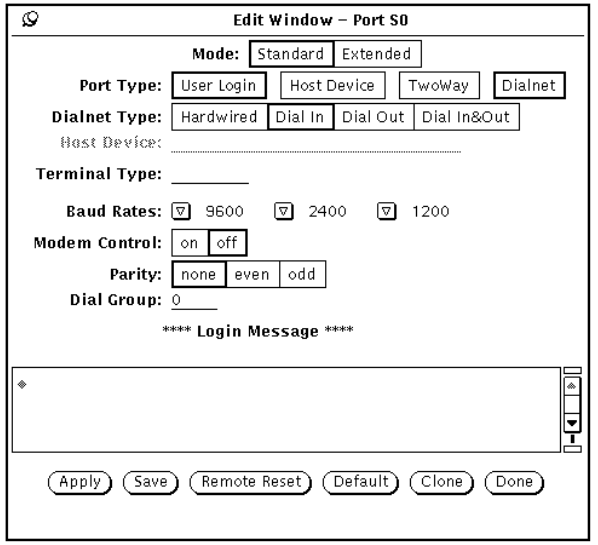


Figure 3-2. Edit Window - Serial Ports

Port Parameters

This window shows the port parameters and the current values. Some selections have a small box with a triangle. Point to the box, click and hold the right mouse button. A menu of choices will be displayed. As you move the cursor through the menu, each item, in turn, will become reverse video. When you are positioned at the value you wish to select, release the mouse button.

Some selection show all options in a horizontal line. The option which is selected has a bold box surrounding it. To select an alternate option, simply click your left mouse button on the desired option.

The edit window supports two modes of operation; "Standard" and "Extended". When the port is being used only for hardwired Dialnet services, the ports have the same values for both standard and extended modes. When Dialnet is mixed with terminal server features, additional field will be displayed on the window.

These are the parameters which can be selected from this window:

Port Type

User Login - (default) the specified port will display a login prompt to the user. After successful baud rate matching and receiving a login name, the user will be transferred directly to a login session on the host Workstation.

Host Device - the port will operate as a host controlled device on the Workstation. The Host Device field must be entered.

Two-Way - (or bidirectional) the port will operate in User Login mode if carrier is detected on pin 8 of the 25-pin connector. Otherwise it can be accessed as a host controlled device on the Workstation. The Host Device field must be entered.

Dialnet - The port becomes available for running the Serial Line Internet Protocol (SLIP) or the Point to Point Protocol (PPP) for establishing network connections between remote sites over modems. The specific services are defined in the following "Dialnet Service" field.

Dialnet Service

Hardwired - The port is being used for a dedicated network connection between two sites. No modem dialing or authentication is required. The port immediately begins running the specified protocol. In this mode, none of the other port types can be used in combination with Dialnet.

Dial In - In this mode, the port will accept dial-in only network connections. The remote system (or user) will be required to enter a user id and password. Dial In network connections are controlled by

the Password Table described later in this chapter.

Dial Out - In this mode, the port becomes available for dialing to remote destinations and initiating network connections to those destinations. Dial Out network connections are controlled by the Location Table described later in this chapter.

Dial In&Out - In this mode, the port will accept dial-in network connections, as well as being available for dial out to remote destinations.

In any of the three Dial modes the port can also be configured to support other port operations concurrently. If one of the other port types is selected, both that port type and Dialnet will be highlighted. If one of the other port types is selected, additional configuration parameters become available which are described in the PortMaster User's Manual.

Baud Rates

All three baud rates should be set to the fixed baud rate of the attached modem.

Modem Control

On or off may be selected. The default is "on". If "on" is selected the PortMaster uses carrier detect to indicate that the connection to the remote system is active. The device will be idle until the carrier detect line is asserted.

Flow Control

Dialnet connections support two types of flow control:

None - In this mode, no flow control will be used. This mode should only be used for modems which support a constant data rate equal to the configured baud rate for the port. Modems which use data compression cannot use this option.

RTS/CTS - This is for devices where hardware flow control is required. When the PortMaster is able to receive data from the attached device, it raises Request to Send (RTS) on pin 4 of the RS-232 connector.

Output from the PortMaster will only occur if the Clear to Send (CTS) modem line on pin 5 of the RS-232 connector is raised by the attached device.

Hardwired Dialnet parameters

The following parameters are only available for ports which are configured for Hardwired Dialnet connections.

Destination

The Internet Address or a valid host name of the system at the remote end of the point to point connection should be entered.

Netmask

The netmask of the system at the remote end of the point to point connection should be entered.

Routing

This indicates whether the PortMaster should use the RIP protocol through the Interface. Any combination of "Broadcast" and "Listen" can be used. If "Broadcast" is selected, the Portmaster will send RIP information to the remote computer. If "Listen" is selected, the PortMaster will accept RIP information from the remote computer.

Protocol

This indicates which protocol to use for encapsulating packets into. The options are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MTU

This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum

Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (i.e. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

Compression

This indicates whether the PortMaster should use Van Jacobson TCP/IP header compression. This option can increase the performance of interactive TCP sessions if both sides of the link support this option. When configuring SLIP connections, both sides must be configured identically. When configuring PPP connections, if compression is enabled and the remote side does not support compression, the PortMaster will dynamically disable this feature.

PPP Async Map

The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments should be set to zero to achieve maximum throughput.

Dial Out Configuration Parameters

Dial Group

This numerical value is used to assign serial ports to modem pools or groups. When the PortMaster initiates a dial-out connection to a remote site, it will only allocate a port which is in the same dial group as has been specified for that location in the Location Table. Valid dial groups are numbered from 0 to 99. The default is 0.

Enabling Serial Port Parameters

If you would like to try a new port configuration, without saving to the PortMaster permanent configuration:

1. Select APPLY

2. Select REMOTE RESET

The new port configuration will now be in effect, until power is removed from the PortMaster or changed again.

If you want to save the new configuration in the permanent PortMaster configuration, select SAVE.

Managing the Location Table

The Location Table is used for all dial out network connections to remote locations. Each entry specifies a remote location, the operational parameters for that location, and information on how to establish a connection to that location.

To view or edit the PortMaster location table, log in to the PortMaster and select the "Tables" button from the main console with your right mouse button. A pop-up menu will be displayed. Select "Locations" from the menu. The following window will be displayed:

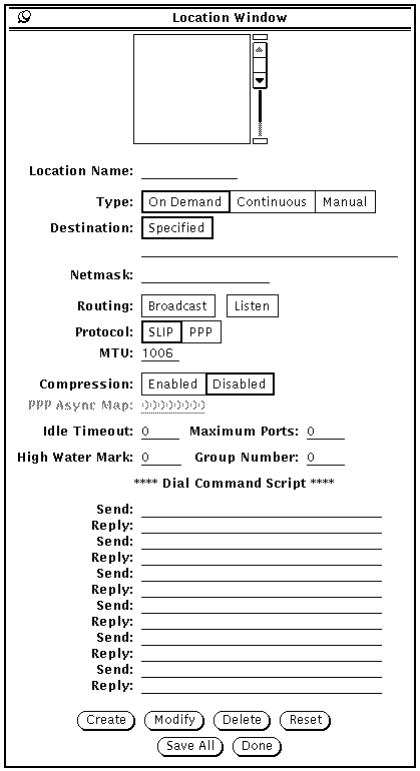


Figure 3-3. Locations Window

The mouse buttons across the bottom of the Locations window are use to access the available functions. The buttons are:

- Create - Creates a new location table entry based on the fields filled in on the location window.
- Modify - Changes the parameters for the location selected in the top location list window to match those filled in on the location window.
- Delete - removes the selected location from the location table.
- Reset - Clears the fields on the location table window.
- Save All - Saves all created, modified, and deleted location table

entries into the PortMaster permanent configuration file.

To create a new location, simply fill in all of the fields in the location window and select the "Create" button. Then select the "Save All" button.

To modify an existing entry, first select that entry with your mouse (by clicking on the entry with the left button) from the selection list. Change the necessary parameters. Then select "Modify" and "Save All".

To delete an existing entry, first select that entry with your mouse (by clicking on the entry with the left button) from the selection list. Then select "Delete" and "Save All".

Location Table Parameters

These are the parameters which can be selected and edited from this window:

Location Name

The name of the remote location. This can be any 12 character name to identify the remote location. It is generally not the name of a single system, but rather an identifier which represents the entire location (ie. the city or the company name at that location).

Type

On Demand - This selection notifies the PortMaster to dial and establish a connection to the remote location only when packets are queued for that location. In this mode the PortMaster creates a network interface and the appropriate routing information to notify the attached local area network of the connectivity to the remote site. This is done even though there may not be an actual physical connection to that site at the time.

Continuous - This selection notifies the PortMaster to always keep the dial-up connection to the remote site active. If the telephone line is lost, the PortMaster will automatically initiate a re-dial to that location.

Manual - This selection notifies the PortMaster to only dial to the remote location when an operator requests the connection. This can be done from the PMconsole Dialer Window or it can be invoked from the "pmdial" program installed on one of the UNIX systems on the network. This also allows connections to be scheduled by using the UNIX cron scheduler to call pmdial.

Note: When switching a location from Manual to On-demand make sure the dial-out connection has been closed (ie. Reset the serial port) before updating the location table.

Destination

For SLIP connections, the IP Address or a valid host name of the system at the remote end of the dial-up connection should be entered.

For PPP connections, the Destination can be negotiated. In this mode, the PortMaster can automatically learn the IP address of the remote system. This feature should only be used for manual or continuous connections. On-Demand locations must use a specified address.

Netmask

The netmask of the system at the remote end of the dial-up connection should be entered.

Routing

This indicates whether the PortMaster should use the RIP protocol through the Interface. Any combination of "Broadcast" and "Listen" can be used. If "Broadcast" is selected, the Portmaster will send RIP information to the remote computer. If "Listen" is selected, the PortMaster will accept RIP information from the remote computer.

Protocol

This indicates which protocol to use for encapsulating packets into. The options are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MTU

This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

Compression

This indicates whether the PortMaster should use Van Jacobson TCP/IP header compression. This option can increase the performance of interactive TCP sessions if both sides of the link support this option. When configuring SLIP connections, both sides must be configured identically. When configuring PPP connections, if compression is enabled and the remote side does not support compression, the PortMaster will dynamically disable this feature.

PPP Async Map

The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments should be set to zero to achieve maximum throughput.

Idle Timeout

For Manual and On-Demand type locations this specifies the number of minutes the line should be idle (no network traffic to the remote site) before disconnecting the telephone connection.

Maximum Ports

The number of dial-out ports the PortMaster can use for this network connection. If 0 is selected dialing to this location will be disabled. If more than 1 is selected the PortMaster will use the "High Water Mark" to decide when to open and dial additional lines. If more than one line is open to the remote location the PortMaster will automatically use load balancing to achieve the maximum throughput of all lines combined. If multiple lines are open the "Idle Timeout" value will be used to decide when to disconnect unneeded lines.

High Water Mark

The number of bytes of queued network traffic required to open an additional dial-out line to the remote site. This value is only used when the "Maximum Ports" is greater than 1 and there are available modems on the PortMaster. Setting a very small number will cause the PortMaster to quickly use all of the ports specified in "Maximum Ports". Generally, interactive terminal traffic has no more than a few hundred bytes queued at any one time. However, file transfers (ie. ftp) will queue several thousand bytes. These size differences should be used when deciding what number to use for the High Water Mark.

Group Number

The modem pool group to which this location is assigned. Only modem ports which are configured with this Group Number will be available for dial-out to this location. This can be used to reserve modems for dial-up to specific locations or to differentiate different types of modems which would be compatible with the remote location.

Dial Command Script

This script is used to issue commands to the modem (or other similar equipment) to dial the correct number and establish the link. In addition, this script is used for authentication (password security) at the remote site. Each "Send" string is issued out of the PortMaster serial port to the modem or remote computer. Each "Reply" string is used by the PortMaster to verify that the previous "Send" string was properly

received by the modem or remote computer and the PortMaster should transmit the next "Send" string.

Send - This string can be up 30 characters in length. Any printing ASCII character can be placed in this string. In addition, the following special characters are available:

\r - This is replaced by the ASCII carriage return character.

\n - This is replaced by the ASCII line feed character.

\0XX - This is replaced by the octal digit specified in the XX.

\\ - This is replaced by the single backslash (\) character.

Generally Send strings would end with the "\r" character.

Reply - This string can be up to 30 characters in length. Any printing ASCII character can be placed in this string. In addition, any of the special characters from above can be used. Generally, a valid reply can be determined by checking for the last word received from the remote system. For example if the remote system is prompting for a login id, it may issued the string:

```
Please enter your login:
```

The Reply string in the Dial Command Script could simple look for "login:".

It is important that the last entry in the Dial Command Script be a Reply string which indicates that the remote location is ready to begin receiving network packets. This is used to activate the TCP/IP protocol coming from the PortMaster. When connecting to a remote PortMaster the final reply string to look for should be "SL/IP" for SLIP connections and "PPP" for PPP connections. For other products, consult that vendors users manual.

The following Dial Command Script is an example of how to establish a connection between two PortMasters which have modems supporting the "AT" dial command syntax:

```
 Send: ATDT18005551212\r
Reply: CONNECT
 Send: \r
Reply: login:
 Send: my_location_name\r
Reply: ssword:
 Send: my_password\r
Reply: SL/IP
 Send:
Reply:
 Send:
Reply:
```

The "18005551212" would be replaced by the actual telephone number of the remote modem. "my_location_name" would be replaced by the actual user name for your location. "my_password" would be replaced by the actual password set up at the remote site which corresponds to the user name.

Managing the Passwords Table

The Passwords Table provides a security and network parameter initialization scheme for locations or individual computers dialing into the PortMaster to establish remote network connections. Modem ports configured to accept Dialnet Dial-in sessions initially prompt for a login user name. After receiving the login name, the PortMaster checks the name against the list of names in the local Passwords Table. If found, the PortMaster initiates password authentication of the remote system. If successfully validated (the password received matches the password stored in the Passwords Table) the PortMaster uses the configuration parameters for the user to initiate a Dialnet connection. If the user has been defined as a "Dialback" network user, the PortMaster will disconnect the line (by dropping DTR) and begin

dialing to the pre-defined dialnet location. If the user is a "Normal" network user, the PortMaster will respond with one of the following messages:

SL/IP session from (149.198.9.1) to remote beginning....

PPP session from (149.198.9.1) to remote beginning....

For SLIP connections, the PortMaster then enables the network interface and any associated routing information as indicated in the Passwords Table. PPP connections are negotiated using the standard PPP option negotiation prior to enabling the network interface. Finally it begins accepting and sending IP packets over the connection.

If the login name does not match an entry in the Passwords Table and the port is configured to support User Logins it automatically forwards the session to the designated host on the local network for password authentication.

To view or edit the PortMaster Passwords Table, log in to the PortMaster and select the "Tables" button from the main console with your right mouse button. A pop-up menu will be displayed. Select "Passwords" from the menu.

The following window will be displayed:

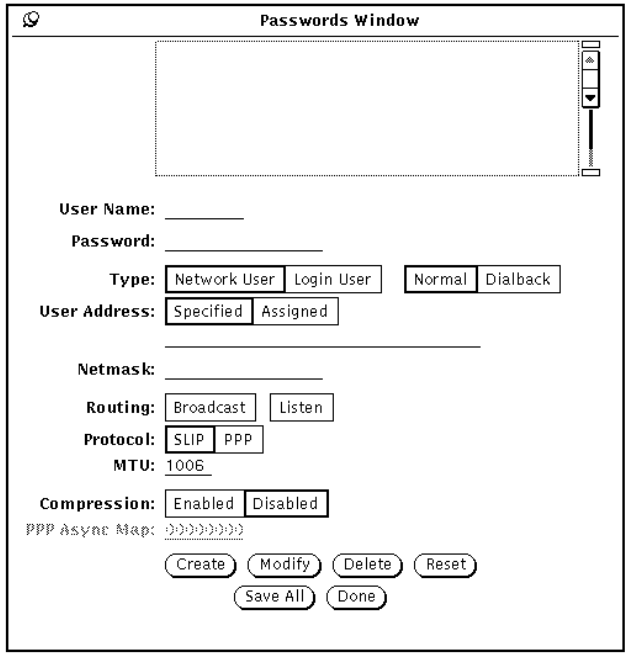


Figure 3-4. Passwords Window

The mouse buttons across the bottom of the Passwords window are use to access the available functions. The buttons are:

- Create - Adds the currently displayed user to the PortMaster Passwords Table.
- Modify - Applies modifications to the user displayed on the Passwords Window.
- Delete - Deletes the currently displayed user from the PortMaster Passwords Table.
- Reset - Clears the Passwords Window to allow the entry of a new user.
- Save All - Stores all creations, modifications, and deletions to the PortMaster Passwords Table into non-volatile storage in the PortMaster.

● Done - Removes the Passwords Window from the display.

Password Parameters

The Passwords table has 10 fields for each normal network user and 4 fields for dialback network users. Consult the PortMaster User's Manual for more information on "Login Users". The fields are:

User Name

The login name of the user. This name can be up to 8 characters in length and is the user name to be supplied at the "login:" prompt from the PortMaster.

Password

The password for the user. This is a string of up to 16 ASCII characters.

Type

All SLIP or PPP users should be set to a type of "Network User". "Login User" is specifically for non-dialnet use. Consult the PortMaster User's manual for information on login users. Network users can be set to "Normal" or "Dialback". Normal users will be allowed to immediately begin a Dialnet session after properly entering their user name and password. Dialback users will be disconnected by the PortMaster and then re-dialed. Re-dialing of dialback users is accomplished through the PortMaster Location Table.

Normal Users

User Address

This is the IP address of the remote computer. Valid Address types are:

Specified - The specific IP address of the remote computer must be entered in the window directly below the "Specified" label. The remote computer will only be allowed to operate with this IP address.

Assigned - The PortMaster will assign a temporary IP address for this

session. This address will be retrieved from the pool of addresses being maintained by the PortMaster. See the Global Configuration window for more information on Assigned Addresses.

Negotiated - This option is valid only for PPP sessions. The PortMaster will attempt to learn the IP: address of the remote computer automatically.

Netmask

The netmask of the system at the remote end of the dial-up connection should be entered.

Routing

This indicates whether the PortMaster should use the RIP protocol through the Interface. Any combination of "Broadcast" and "Listen" can be used. If "Broadcast" is selected, the Portmaster will send RIP information to the remote computer. If "Listen" is selected, the PortMaster will accept RIP information from the remote computer.

Protocol

This indicates which protocol to use for encapsulating packets into. The options are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MTU

This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance

while running concurrent file transfers.

Compression

This indicates whether the PortMaster should use Van Jacobson TCP/IP header compression. This option can increase the performance of interactive TCP sessions if both sides of the link support this option. When configuring SLIP connections, both sides must be configured identically. When configuring PPP connections, if compression is enabled and the remote side does not support compression, the PortMaster will dynamically disable this feature.

PPP Async Map

The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments should be set to zero to achieve maximum throughput.

Dialback Users

Location Name

This is the name of the location stored in the PortMaster Location Table. After authenticating the user, the PortMaster will disconnect the active connection. It will then schedule an automatic dial to the specified location.

The Hosts Table

Each device attached to the Ethernet (with uses TCP/IP) is assigned a unique internet address. Internet addresses are four byte numerical values. To make it easier to remember these values, most systems on

the network support a way of aliasing the internet address to some unique and easy to remember name. This is done on UNIX systems in the /etc/hosts file. The PortMaster also supports the same capability by maintaining a local hosts table.

The PortMaster hosts table should be used when any of the serial ports is set to prompt the user for a host name on the network. This table is then consulted to determine the correct internet address of the host.

If the hostname specified by the user is not found in the hosts table, then the PortMaster will attempt to match the hostname using the Network Information Service (NIS), also known as "YP". See the section "Global Configuration" for more information on setting the YP server and YP domain.

For information on managing the PortMaster Hosts Table, consult the PortMaster User's Manual.

The Routes Table

The PortMaster Routes table is used when static network routes are required. Static routes are required when the user chooses not to run the Routing Information Protocol (RIP) on the PortMaster, or when hosts or networks attached to the PortMaster do not have the ability to run the RIP protocol. For information on managing the Routes Table, consult the PortMaster User's Manual.

Dialing Out to Remote Locations

On-demand or Continuous dial-out locations automatically initiate dialing to a remote location. Manual connections require a system administrator or a scheduler to initiate dial-out connections. PMconsole provides a facility for the system administrator to initiate the dial-out operation and if necessary to monitor the progress of the dial-out operation.

When setting up a new On-demand location it is recommended to first set up the location as a manual dial-out site for testing purposes and then later switch the location type to On-demand.

Note: When switching a location from Manual to On-demand make sure the dial-out connection has been closed (ie. Reset the serial port) before updating the location table.

To initiate dial-out connections via a scheduler (ie. cron in UNIX systems) or from a UNIX command line prompt see the "pmdial" command in the UNIX Utilities section of this manual.

To dial to a remote manual location from PMconsole, select the "View" button from the main console with your right mouse button. A pop-up menu will be displayed. Select "Dialer" from the menu. The window in figure 3-5 will be displayed.

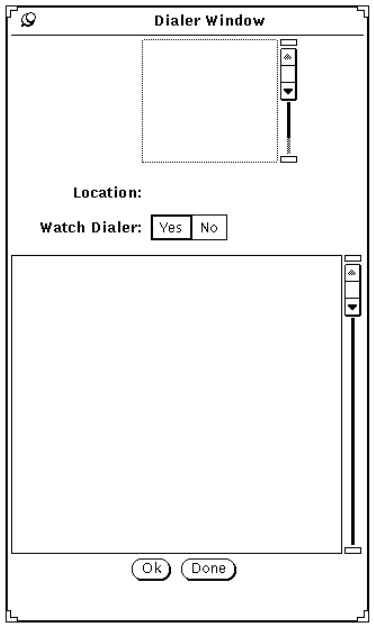


Figure 3-5. Dialer Window

The scrollable list at the top of the window shows all location names which have been set up as "Manual" dial-out sites.

Below the list of locations are two fields:

Location

This is the location to dial. This field is filled in by selecting one of the locations from the scrollable list with the left mouse button.

Watch Dialer

Yes - Causes the PortMaster to display the actual progress of the Dial Command Script in the window below.

No - Causes the PortMaster to start dialing the location and immediately returns control to PMconsole for another operation.

The mouse buttons across the bottom of the Dialer window are use to access the available functions. The buttons are:

- Ok - Begin dialing the specified location.

To begin dialing to a remote location select that entry with your mouse (by clicking on the entry with the left button) from the selection list. The location name will be filled in. Then select "Yes" or "No" as the monitoring option. Finally select "Ok" from the buttons at the bottom of the window.

If "Yes" was selected in the Watch Dialer field the message window will show the location being dialed and the modem port being used. It will then show the execution of the Dial Command Script. The "sendthem" message shows the exact string of characters being sent out the serial port to the modem or remote system enclosed in parenthesis. The "expect" message shows the response string the PortMaster is expecting as a reply enclosed in parenthesis. Immediately following the "expect" string are the characters actually being received by the PortMaster from the modem or remote system. When the PortMaster successfully receives the string is is searching for it will display ":got it" followed by the next send message.

If the dial script completes successfully, the message "Connection Succeeded" will be displayed. Otherwise, after timing out, the message "Connection Failed" will be displayed.

Monitoring Dialnet Connections

The PortMaster allows you to monitor two sets of data communications information. The port statistics or the network statistics can be monitored. Port statistics allow you to view input/output information for each of the ports. This includes things like input bytes, output bytes, errors, active baud rate, and other active parameters. Network statistics allow you to monitor information for each network interface. This includes the Ethernet interface and each of the active Dialnet interfaces. Network statistics include input packets, output packets, packet errors, and active interface values.

From the PortMaster Console window, select the "View" button from the main console with your right mouse button. A pop-up menu will be displayed. Select either "Ports" or "Netstats" from the menu.

Port Statistics

After selecting "Ports" from the pop-up menu, the window in figure 3-6 will be displayed.

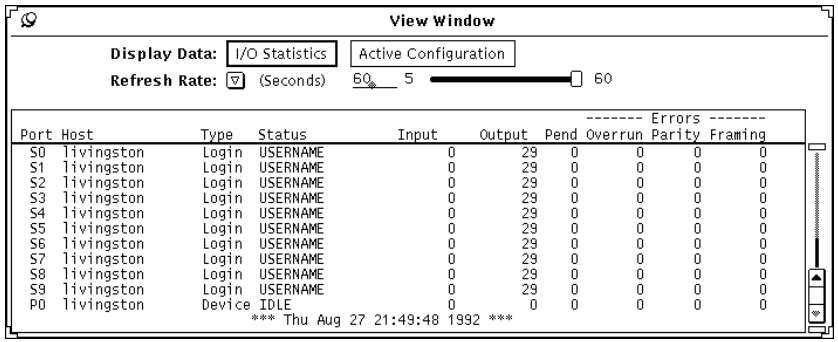


Figure 3-6. View Window

This window allows you to view the configuration and Status of each port. These values are refreshed at a specific rate as defined by the

"Refresh Rate" values. Use the cycling-arrow icon or the slide bar to set the Refresh Rate. The Refresh Rate slide bar can be set to correspond to either seconds, minutes or hours by selection with the mouse.

Two sets of data can be displayed in the view window. These are the "Active Configuration" and the "I/O Statistics". Both modes display the port, its hostname (if configured to support User Logins) or its active network interface if it is currently being used for a Dialnet session, the port type, and the current status of the port.

When "Active Configuration" is selected, the window displays data communications parameters such as line speed, databits and parity.

When "I/O Statistics" is selected the window displays input and output information. There are three columns labeled: Input, Output, and Pending. These relate to the number of bytes transferred in or out the port and the number of bytes in the PortMaster output buffer which are pending to be output. There are also three error counts displayed: Overrun, Parity and Framing errors.

Network Statistics

After selecting "Netstats" from the pop-up menu, the window in figure 3-7 will be displayed.

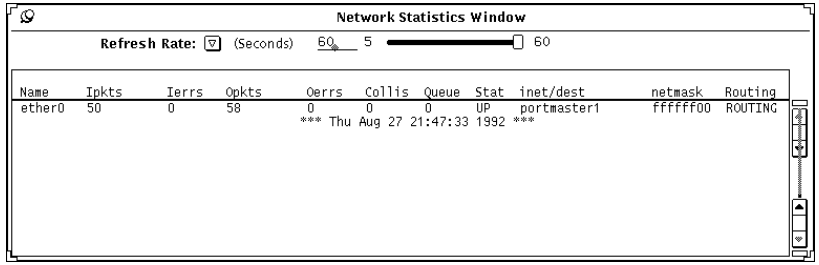


Figure 3-7. Netstats Window

This window allows you to view the Status of each network interface in the PortMaster. These values are refreshed at a specific rate as defined by the "Refresh Rate" values. Use the cycling-arrow icon or the slide

bar to set the Refresh Rate. The Refresh Rate slide bar can be set to correspond to either seconds, minutes or hours by selection with the mouse.

The values which are displayed are:

- Name - The name of the network interface. The Ethernet interface is named "ether0" and each interface through the serial ports are named "ptp", followed by a uniquely assigned number.
- Ipkts - The number of input packets received by the interface.
- Ierrs - The number of input packets received by the interface which were dropped due to errors.
- Opkts - The number of output packets sent by the interface.
- Oerrs - The number of output packets which were not sent by the interface due to errors.
- Collis - The number of collisions experienced while trying to send packets out the interface.
- Queue - The number of packets currently queued for transmission on the interface.
- Stat - Either "UP" or "DOWN" indicating whether the interface is active.
- inet/dest - For the Ethernet interface, the internet address of the PortMaster. For the ptp interfaces, the internet address of the remote system.
- netmask - The netmask of the inet/dest field.
- Routing - "Routing", "Sending", "Listen", or "Quiet" indicating whether RIP protocol packets are being transmitted or received by the interface.

Dialnet with an ASCII Terminal

The ASCII terminal interface provides several screens through which you can configure the PortMaster ports, monitor the performance, and configure Dialnet parameters. This can be run from any ASCII terminal logged onto a Workstation which supports full screen addressing and has an entry in the /etc/termcap file or the terminfo database. Before running pmconsole verify that your TERM environment variable matches your terminals entry in the /etc/termcap file. If the screen display is unreadable, the wrong terminal type has probably been specified. Press the "Control" key and the letter "C" to exit the program and consult with your system administrator on the proper setting of your terminal type.

Global Configuration

The Global configuration screen is used to set parameters which are common to all ports and operations of the PortMaster. This screen should be configured prior to setting any other Dialnet configuration parameters. From the PortMaster Console, do the following:

1. Select the EDIT command. A new menu listing all of the ports and the Network will be displayed. Highlight the "NETWORK" command using the arrow keys and press <RETURN>.

2. The screen in Figure 3-8 will be displayed.

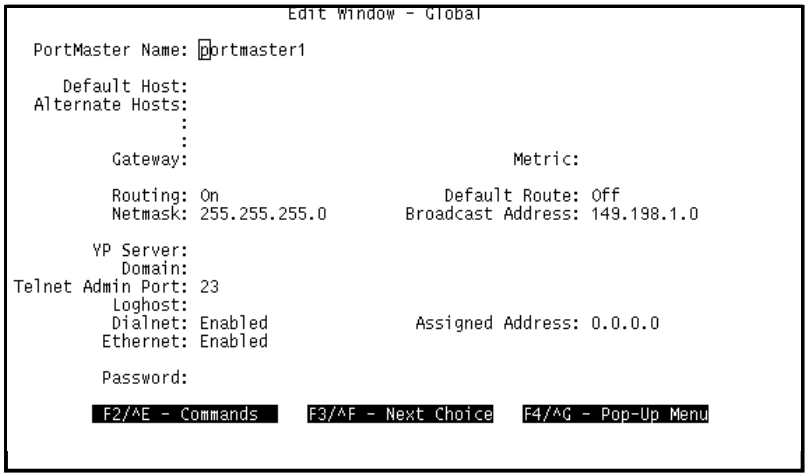


Figure 3-8. Edit screen - Global

3. Enter the following information, terminated by pressing RETURN:

- PortMaster Name - name of PortMaster as entered during the installation procedure.
- Default Host - name of PortMaster host. This is the name of the primary host which the PortMaster will be using for login or device services.
- Alternate Hosts - up to three alternate host names. Ports which are configured for User Login will automatically switch to an alternate host if the Default Host is unavailable (ie. system is down or network is unreachable). Login connections are attempted in the order in which the alternate host names are entered. These are optional and may be left blank.
- Gateway Name - If the PortMaster is being connected to a network with gateway systems, then the name of the default gateway on the same network should be entered. The PortMaster automatically routes network packets through gateways which are running the Routing Information Protocol (RIP), but will use the default gateway if no RIP gateways are found.

- Gateway Metric - If the Gateway field above is being used, this value (from 1-15) is the cost associated with using the default gateway route. This is useful if the PortMaster is configured to listen for default routes coming from other routers. Otherwise, this field can be left blank.
- Routing - This indicates whether the PortMaster should use the RIP protocol through the Ethernet Interface. The following choices are available by using the "Next Choice" or "Pop-Up Menu" function keys:

On - The PortMaster will send RIP information to other computers attached to the local Ethernet and will accept RIP information from other routers on the local Ethernet.

Broadcast - The PortMaster will send RIP information to other computers attached to the local Ethernet.

Listen - The PortMaster will accept RIP information from other routers on the local Ethernet.

Off - The PortMaster will not send RIP information to the local Ethernet and it will ignore all RIP messages it receives on the local Ethernet.

- Default Route - This option works in conjunction with the "Routing" field above. The following choices are available by using the "Next Choice" or "Pop-Up Menu" function keys:

On - The PortMaster will send default route information as part of its normal RIP messages and will accept default route information coming from other routers on the local Ethernet.

Broadcast - The PortMaster will send default route information as part of its normal RIP messages.

Listen - The PortMaster will accept default route information coming from other routers on the local Ethernet.

Off - The PortMaster will not send default route information to the local Ethernet and it will ignore all default route messages it receives on the local Ethernet.

- Netmask - defaults to 255.255.255.0. This should only be modified

if your site uses subnetting or Class A or B IP addresses. Consult you network administrator for this information.

- Broadcast Address - This is the IP address which the PortMaster will interpret and use as the local broadcast address. This is used primarily by the RIP protocol and some diagnostic protocols. This value should match other computers on the same local Ethernet. This can be selected by using the "Next Choice" function key.
- YP Server - For networks which use Sun's Network Information Service (NIS), also known as YP, the PortMaster can be configured to use this service for host name matching. If this service is being used, enter the hostname of the YP server which the PortMaster should use.
- **Domain** - If a YP Server is being used, the correct YP domain name must be entered.
- **Telnet Admin Port** - The PortMaster allows the system administrator to access all of the PortMaster administration functions through a command line interface. These functions can be reached by logging onto the PortMaster with a telnet session. This field identifies the specific TCP port number for the PortMaster to listen on for incoming telnet administration sessions. The default value is "23". Any number from 0 to 65536 may be used. If 0 is selected, the PortMaster will disable the telnet administration function.
- **Loghost** - The PortMaster supports the "syslog" network accounting function. If a host is entered in this field, the PortMaster will log user authentication information to this host using the system facility "auth" and the priority of "info" (auth.info). Consult your system's syslogd and /etc/syslog.conf manual pages for more information on these services.
- **Dialnet** - This field enables the Dialnet features of the PortMaster. The configuration options for the serial ports are modified by enabling or disabling Dialnet. This can be selected by using the "Next Choice" function key.
- **Assigned Address** - This field is used for network login users using the PPP protocol. If a network user in the passwords table has been

set to use an "Assigned Address", the PortMaster must allocate a temporary IP address to be used for their active session. This IP address is retrieved from a pool of addresses maintained by the PortMaster. The IP address entered in this field will be used as the first free address in the pool of addresses. The PortMaster will allocate as many addresses as there are ports configured for Dialnet access.

- **Ethernet** - This field enables the Ethernet interface of the PortMaster. It should only be disabled when the PortMaster is not being attached directly to the local Ethernet. This can be selected by using the "Next Choice" function key.
- **Password** - PortMaster administration password.

As each of the first three items is entered, it will be validated against your systems global host table (/etc/hosts or Yellow Pages). These were established by PMinstall if this is a stand-alone installation, or by your network administrator. The Netmask should not be changed unless you are using sub-nets and understand their operation. Refer to your workstation documentation for more information on sub-nets.

4. You will notice a function key label at the bottom of the screen. This indicates that by pressing function key F2 or by pressing down the "Control" key and typing the letter "E" the commands menu will be displayed. To save the configuration data in the PortMaster permanent configuration, press F2 and select SAVE from the command list. At the bottom of the screen will be a message:

Configuration parameters successfully saved.

5. Select the DONE command. At this point, the PortMaster has all the information necessary to begin normal communications with the network.

Serial Port Configuration

The PortMaster serial ports can be configured for Dialnet operation only, terminal server features only, or a combination of the two. These features are chosen by selecting the appropriate port type for each of the serial ports. User Login, Host Device, and TwoWay operation are described in the PortMaster User's manual.

To set the configuration for a serial port, use the following general procedure:

1. Select the EDIT command. A new menu listing all of the ports and the Network will be displayed. Highlight the desired Serial Port using the arrow keys and press <RETURN>.

2. The Edit screen for the serial port (Figure 3-9) will appear. From this screen you can modify the serial port configuration parameters.

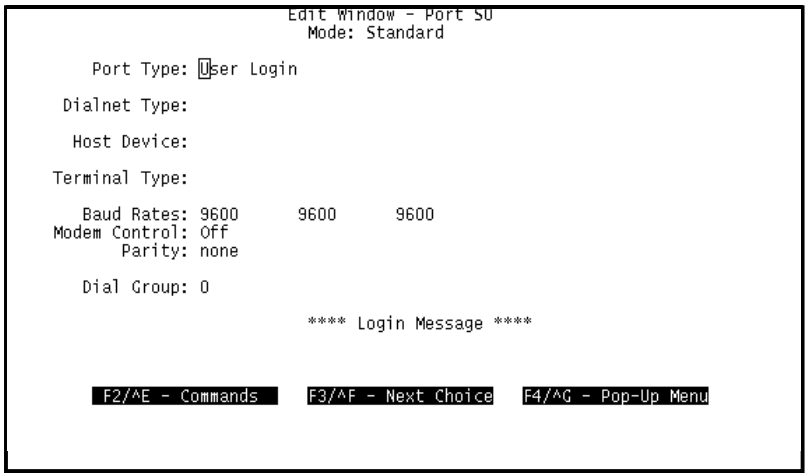


Figure 3-9. Edit screen - Serial Port

Upon entry, you will be in edit mode. This means that you can move around the screen with the arrow keys or by pressing <RETURN>. In addition, there is a set of function key labels across the bottom of the screen. Any of these function keys (or their associated Control characters) can be pressed at any time. The function keys and their meanings are:

F2 or "Control-E" - Commands. When this key is pressed, the function keys will be replaced by a menu of commands which can be executed. The available commands are:

- APPLY - send the configuration to PortMaster memory. Does not take effect until you select REMOTE RESET. These values are temporary until you select SAVE. If power is removed from the PortMaster, the new parameter values will no longer be stored in the PortMaster.
- SAVE - performs an APPLY and stores the port configuration in the PortMaster permanent configuration.
- RESET - reset the serial port using the values stored by the APPLY or SAVE selection.
- GENERIC - set the on-screen values to the port default configuration.
- CLONE - copy the port configuration from another port.
- EDIT - returns you to edit mode. The cursor moves back up onto the screen and the function key labels re-appear.
- DONE - return to the PortMaster Console screen.

F3 or "Control-F" - Next Choice. When this key is pressed, and the cursor is on a menu field, the next available choice for that menu will be displayed.

F4 or "Control-G" - Pop-up Menu. When this key is pressed, and the cursor is on a menu field, a menu of available choices will be displayed.

This screen displays the serial port parameters and the current values. Some selections have menus. To use the menu, move your cursor to the field to be changed with the <RETURN> or arrow keys. Press the

F4 function key (or Control-G) and a menu of choices will be displayed. Alternately the selections can be rotated through with the F3 function key (or Control-F). Other items can be edited directly from your keyboard by typing in new choices.

The serial port edit screen supports two modes of operation; "Standard" and "Extended". Most users will find that the "Standard" mode will meet all of their needs. "Extended" mode is for users who wish to customize the operation of the serial port more extensively. The desired mode can be selected by choosing a **Mode** option at the top of the edit screen.

Port Parameters

These are the parameters that can be selected from this screen:

Port Type *(menu field)*

User Login - (default) the specified port will display a login prompt to the user. After successful baud rate matching and receiving a login name, the user will be transferred directly to a login session on the host Workstation.

Host Device - the port will operate as a host controlled device on the Workstation. The Host Device field described below must be entered.

Two-Way - (or bidirectional) the port will operate in User Login mode if carrier is detected on pin 8 of the 25-pin connector. Otherwise it can be accessed as a host controlled device on the Workstation. The Host Device field described below must be entered.

Dialnet - The port becomes available for establishing network connections between remote sites over modems. The specific services are defined in the following "Dialnet Service" field.

Dialnet Service *(menu field)*

Hardwired - The port is being used for a dedicated network connection between two sites. No modem dialing or authentication is required. The port immediately begins running the Slip or PPP protocol. In this mode, none of the other port types can be used in combination with Dialnet.

Dial In - In this mode, the port will accept dial-in only network connections. The remote system (or user) will be required to enter a user id and password. Dial In network connections are controlled by the Password Table described later in this chapter.

Dial Out - In this mode, the port becomes available for dialing to remote destinations and initiating network connections to those destinations. Dial Out network connections are controlled by the Location Table described later in this chapter.

Dial In&Out - In this mode, the port will accept dial-in network connections, as well as being available for dial out to remote destinations.

In any of the three Dial modes the port can also be configured to support other port operations concurrently. If one of the other port types is selected, both that port type and Dialnet will be selected.

If one of the other port types is selected, additional configuration parameters become available which are described in the PortMaster User's Manual.

Baud Rates *(menu field)*

All three baud rates should be set to the fixed baud rate of the attached modem.

Modem Control *(menu field)*

On or off may be selected. The default is "off". If "on" is selected the PortMaster uses the modem carrier detect line. The device will be idle until the carrier detect line becomes true.

Flow Control *(menu field)*

Dialnet connections support two types of flow control:

None - In this mode, no flow control will be used. This mode should only be used for modems which support a constant data rate equal to the configured baud rate for the port. Modems which use data compression cannot use this option.

RTS/CTS - This is for devices where hardware flow control is required. When the PortMaster is able to receive data from the attached device, it raises Request to Send (RTS) on pin 4 of the RS-232 connector. Output from the PortMaster will only occur if the Clear to Send (CTS) modem line on pin 5 of the RS-232 connector is raised by the attached device.

Hardwired Dialnet parameters

The following parameters are only available for ports which are configured for Hardwired Dialnet connections.

Destination *(data entry field)*

The Internet Address or a valid host name of the system at the remote end of the point to point connection should be entered.

Netmask *(data entry field)*

The netmask of the system at the remote end of the point to point connection should be entered.

Routing *(menu field)*

This indicates whether the PortMaster should use the RIP protocol through the Interface. Any combination of "Broadcast" and "Listen" can be used. If "Broadcast" is selected, the Portmaster will send RIP information to the remote computer. If "Listen" is selected, the PortMaster will accept RIP information from the remote computer.

Protocol *(menu field)*

This indicates which protocol to use for encapsulating packets into. The options are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MTU *(data entry field)*

This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

Compression *(menu field)*

This indicates whether the PortMaster should use Van Jacobson TCP/IP header compression. This option can increase the performance of interactive TCP sessions if both sides of the link support this option. When configuring SLIP connections, both sides must be configured identically. When configuring PPP connections, if compression is enabled and the remote side does not support compression, the PortMaster will dynamically disable this feature.

PPP Async Map *(data entry field)*

The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character

"NULL" and so on. Most environments should be set to zero to achieve maximum throughput.

Dial Out Configuration Parameters

Dial Group *(data entry field)*

This numerical value is used to assign serial ports to modem pools or groups. When the PortMaster initiates a dial-out connection to a remote site, it will only allocate a port which is in the same dial group as has been specified for that location in the Location Table. Valid dial groups are numbered from 0 to 99. The default is 0.

Enabling Serial Port Parameters

If you would like to try a new port configuration, without saving into the permanent configuration:

1. Select F2 (or Control-E) to view the Commands Menu.

Select APPLY from this menu.

2. Select REMOTE RESET from the Commands Menu.

The new port configuration will now be in effect, until the PortMaster is rebooted, the power is removed from the PortMaster or the parameters are changed again and applied.

If you want to save the new configuration in the permanent PortMaster configuration, select SAVE from the Commands Menu.

Managing the Location Table

The Location Table is used for all dial out network connections to remote locations. Each entry specifies a remote location, the operational parameters for that location, and information on how to establish a connection to that location.

To view or edit the PortMaster location table, log in to the PortMaster and select the "Tables" item from the main console menu and press <RETURN>. A pop-up menu will be displayed. Select "Locations"

from the menu and press <RETURN>.

The following screen will be displayed:

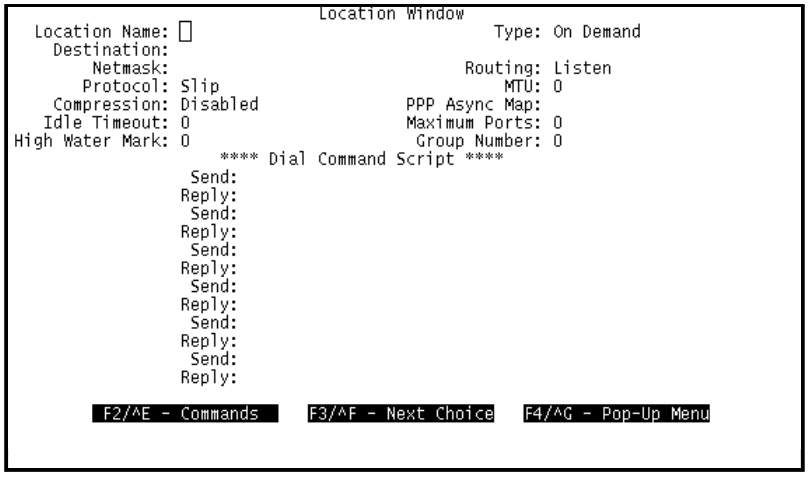


Figure 3-10. Locations Screen

Upon entry, you will be in edit mode. This means that you can move around the screen with the arrow keys or by pressing <RETURN>. In addition, there is a set of function key labels across the bottom of the screen. Any of these function keys (or their associated Control characters) can be pressed at any time. The function keys and their meanings are:

F2 or "Control-E" - Commands. When this key is pressed, the function keys will be replaced by a menu of commands which can be executed. The available commands are:

- Create - Creates a new location table entry based on the fields filled in on the location screen.
- Modify - Changes the parameters for the location specified in the Location Name field.
- Purge - removes the selected location from the location table.
- Reset - Clears the fields on the location table window.

- Save All - Saves all created, modified, and deleted location table entries into the PortMaster permanent configuration file.
- Edit - returns you to edit mode. The cursor moves back up onto the screen and the function key labels re-appear.
- Done - return to the PortMaster Console screen.

F3 or "Control-F" - Next Choice. When this key is pressed, and the cursor is on a menu field, the next available choice for that menu will be displayed.

F4 or "Control-G" - Pop-up Menu. When this key is pressed, and the cursor is on a menu field, a menu of available choices will be displayed.

To create a new location, simply fill in all of the fields in the location screen and select the F2 (Control-E) function key. Select "Create" from the menu. Then select "Save All" from the menu.

To modify an existing entry, first select that entry from the pop-up menu on the Location Name field. Change the necessary parameters. Then select "Modify" and "Save All" from the Commands menu.

To delete an existing entry, first select that entry from the pop-up menu on the Location Name field. Then select "Delete" and "Save All" from the Commands menu.

Location Table Parameters

These are the parameters which can be selected and edited from this window:

Location Name *(menu and data entry field)*

The name of the remote location. This can be any 12 character name to identify the remote location. It is generally not the name of a single system, but rather an identifier which represents the entire location (ie. the city or the company name at that location).

Type *(menu field)*

On Demand - This selection notifies the PortMaster to dial and establish a connection to the remote location only when packets are queued for that location. In this mode the PortMaster creates a network interface and the appropriate routing information to notify the attached local area network of the connectivity to the remote site. This is done even though there may not be an actual physical connection to that site at the time.

Continuous - This selection notifies the PortMaster to always keep the dial-up connection to the remote site active. If the telephone line is lost, the PortMaster will automatically initiate a re-dial to that location.

Manual - This selection notifies the PortMaster to only dial to the remote location when an operator requests the connection. This can be done from the PMconsole Dialer Screen or it can be invoked from the "pmdial" program installed on one of the UNIX systems on the network. This also allows connections to be scheduled by using the UNIX cron scheduler to call pmdial.

Note: When switching a location from Manual to On-demand make sure the dial-out connection has been closed (ie. Reset the serial port) before updating the location table.

Destination *(data entry field)*

For SLIP connections, the IP Address or a valid host name of the system at the remote end of the dial-up connection should be entered.

For PPP connections, the Destination can be negotiated. In this mode, the PortMaster can automatically learn the IP address of the remote system. Enter the word "Negotiated" in this field to enable this feature. This should only be used for manual or continuous connections. On-Demand locations must use a specified address.

Netmask *(data entry field)*

The netmask of the system at the remote end of the dial-up connection should be entered.

Routing *(menu field)*

This indicates whether the PortMaster should use the RIP protocol through the Interface. Any combination of "Broadcast" and "Listen" can be used. If "Broadcast" is selected, the Portmaster will send RIP information to the remote computer. If "Listen" is selected, the PortMaster will accept RIP information from the remote computer.

Protocol *(menu field)*

This indicates which protocol to use for encapsulating packets into. The options are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MTU *(data entry field)*

This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

Compression *(menu field)*

This indicates whether the PortMaster should use Van Jacobson TCP/IP header compression. This option can increase the performance of interactive TCP sessions if both sides of the link support this option. When configuring SLIP connections, both sides must be configured identically. When configuring PPP connections, if compression is enabled and the remote side does not support compression, the PortMaster will dynamically disable this feature.

PPP Async Map *(data entry field)*

The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments should be set to zero to achieve maximum throughput.

Idle Timeout *(data entry field)*

For Manual and On-Demand type locations this specifies the number of minutes the line should be idle (no network traffic to the remote site) before disconnecting the telephone connection.

Maximum Ports *(data entry field)*

The number of dial-out ports the PortMaster can use for this network connection. If 0 is selected dialing to this location will be disabled. If more than 1 is selected the PortMaster will use the "High Water Mark" to decide when to open and dial additional lines. If more than one line is open to the remote location the PortMaster will automatically use load balancing to achieve the maximum throughput of all lines combined. If multiple lines are open the "Idle Timeout" value will be used to decide when to disconnect unneeded lines.

High Water Mark *(data entry field)*

The number of bytes of queued network traffic required to open an additional dial-out line to the remote site. This value is only used when the "Maximum Ports" is greater than 1 and there are available modems on the PortMaster. Setting a very small number will cause the PortMaster to quickly use all of the ports specified in "Maximum Ports". Generally, interactive terminal traffic has no more than a few hundred bytes queued at any one time. However, file transfers (ie. ftp) will queue several thousand bytes. These size differences should be used when deciding what number to use for the High Water Mark.

Group Number *(data entry field)*

The modem pool group to which this location is assigned. Only modem ports which are configured with this Group Number will be available for dial-out to this location. This can be used to reserve modems for dial-up to specific locations or to differentiate different types of modems which would be compatible with the remote location.

Dial Command Script *(data entry field)*

This script is used to issue commands to the modem (or other similar equipment) to dial the correct number and establish the link. In addition, this script is used for authentication (password security) at the remote site. Each "Send" string is issued out of the PortMaster serial port to the modem or remote computer. Each "Reply" string is used by the PortMaster to verify that the previous "Send" string was properly received by the modem or remote computer and the PortMaster should transmit the next "Send" string.

Send - This string can be up 30 characters in length. Any printing ASCII character can be placed in this string. In addition, the following special characters are available:

\r - This is replaced by the ASCII carriage return character.

\n - This is replaced by the ASCII line feed character.

\0XX - This is replaced by the octal digit specified in the XX.

\\ - This is replaced by the single backslash (\) character.

Generally Send strings would end with the "\r" character.

Reply - This string can be up to 30 characters in length. Any printing ASCII character can be placed in this string. In addition, any of the special characters from above can be used. Generally, a valid reply can be determined by checking for the last word received from the remote system. For example if the remote system is prompting for a login id, it may issued the string:

```
Please enter your login:
```


The Reply string in the Dial Command Script could simple look for "login:".

It is important that the last entry in the Dial Command Script be a Reply string which indicates that the remote location is ready to begin receiving network packets. This is used to activate the TCP/IP protocol coming from the PortMaster. When connecting to a remote PortMaster the final reply string to look for should be "SL/IP" for SLIP connections and "PPP" for PPP connections. For other products, consult that vendors users manual.

The following Dial Command Script is an example of how to establish a connection between two PortMasters which have modems supporting the "AT" dial command syntax:

```
 Send: ATDT18005551212\r
Reply: CONNECT
 Send: \r
Reply: login:
 Send: my_location_name\r
Reply: ssword:
 Send: my_password\r
Reply: SL/IP
 Send:
Reply:
 Send:
Reply:
```

The "18005551212" would be replaced by the actual telephone number of the remote modem. "my_location_name" would be replaced by the actual user name for your location. "my_password" would be replaced by the actual password set up at the remote site which corresponds to the user name.

Managing the Passwords Table

The Passwords Table provides a security and network parameter initialization scheme for locations or individual computers dialing into

the PortMaster to establish remote network connections. Modem ports configured to accept Dialnet Dial-in sessions initially prompt for a login user name. After receiving the login name, the PortMaster checks the name against the list of names in the local Passwords Table. If found, the PortMaster initiates password authentication of the remote system. If successfully validated (the password received matches the password stored in the Passwords Table) the PortMaster uses the configuration parameters for the user to initiate a Dialnet connection. If the user has been defined as a "Dialback" network user, the PortMaster will disconnect the line (by dropping DTR) and begin dialing to the pre-defined dialnet location. If the user is a "Normal" network user, the PortMaster will respond with one of the following messages:

SL/IP session from (149.198.9.1) to remote beginning....

PPP session from (149.198.9.1) to remote beginning....

For SLIP connections, the PortMaster then enables the network interface and any associated routing information as indicated in the Passwords Table. PPP connections are negotiated using the standard PPP option negotiation prior to enabling the network interface. Finally it begins accepting and sending IP packets over the connection.

If the login name does not match an entry in the Passwords Table and the port is configured to support User Logins it automatically forwards the session to the designated host on the local network for password authentication.

To view or edit the PortMaster Passwords Table, log in to the PortMaster and select the "Tables" item from the main console menu. A pop-up menu will be displayed. Select "Passwords" from the menu.

The following screen will be displayed:

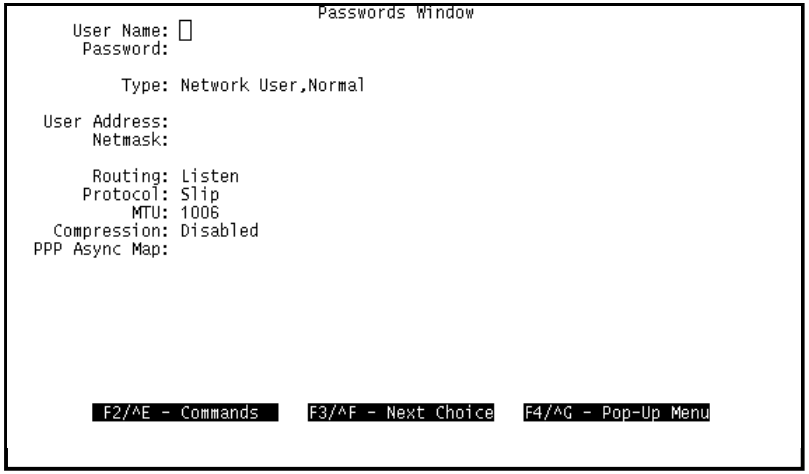


Figure 3-11. Passwords Screen

Upon entry, you will be in edit mode. This means that you can move around the screen with the arrow keys or by pressing <RETURN>. In addition, there is a set of function key labels across the bottom of the screen. Any of these function keys (or their associated Control characters) can be pressed at any time. The function keys and their meanings are:

F2 or "Control-E" - Commands. When this key is pressed, the function keys will be replaced by a menu of commands which can be executed. The available commands are:

- Create - Creates a new user based on the fields filled in on the passwords screen.
- Modify - Changes the parameters for the user specified in the User Name field.
- Purge - removes the selected user from the passwords table.
- Reset - Clears the fields on the screen.
- Save All - Saves all created, modified, and deleted users into the PortMaster permanent configuration file.

- Edit - returns you to edit mode. The cursor moves back up onto the screen and the function key labels re-appear.
- Done - return to the PortMaster Console screen.

F3 or "Control-F" - Next Choice. When this key is pressed, and the cursor is on a menu field, the next available choice for that menu will be displayed.

F4 or "Control-G" - Pop-up Menu. When this key is pressed, and the cursor is on a menu field, a menu of available choices will be displayed.

To create a new user, simply fill in all of the fields in the passwords screen and select the F2 (Control-E) function key. Select "Create" from the menu. Then select "Save All" from the menu.

To modify an existing entry, first select that entry from the pop-up menu on the User Name field or enter the users name. Change the necessary parameters. Then select "Modify" and "Save All" from the Commands menu.

To delete an existing entry, first select that entry from the pop-up menu on the User Name field or enter the users name. Then select "Delete" and "Save All" from the Commands menu.

Password Parameters

The Passwords table has 10 fields for each normal network user and 4 fields for dialback network users. Consult the PortMaster User's Manual for more information on "Login Users". The fields are:

User Name *(data entry and menu field)*

The login name of the user. This name can be up to 8 characters in length and is the user name to be supplied at the "login:" prompt from the PortMaster.

Password *(data entry field)*

The password for the user. This is a string of up to 16 ASCII characters.

Type *(menu field)*

All SLIP or PPP users should be set to a type of "Network User". "Login User" is specifically for non-dialnet use. Consult the PortMaster User's manual for information on login users. Network users can be set to "Normal" or "Dialback". Normal users will be allowed to immediately begin a Dialnet session after properly entering their user name and password. Dialback users will be disconnected by the PortMaster and then re-dialed. Re-dialing of dialback users is accomplished through the PortMaster Location Table.

Normal Users

User Address *(data entry and menu field)*

This is the IP address of the remote computer. Valid Address types are:

Specified - The specific IP address of the remote computer must be entered in the User Address field. The remote computer will only be allowed to operate with this IP address.

Assigned - The PortMaster will assign a temporary IP address for this session. This address will be retrieved from the pool of addresses being maintained by the PortMaster. This option can be selected with the menu function keys or the word "Assigned" can be entered directly. See the Global Configuration window for more information on Assigned Addresses.

Negotiated - This option is valid only for PPP sessions. The PortMaster will attempt to learn the IP: address of the remote computer automatically. This option can be selected with the menu function keys or the word "Negotiated" can be entered directly.

Netmask *(data entry field)*

The netmask of the system at the remote end of the dial-up connection should be entered.

Routing *(menu field)*

This indicates whether the PortMaster should use the RIP protocol through the Interface. The following options are available:

On - The Portmaster will send RIP information to the remote computer and will accept RIP information from the remote computer.

Broadcast - The Portmaster will send RIP information to the remote computer, but will not accept RIP information from the remote computer.

Listen - The Portmaster will not send RIP information to the remote computer, but will accept RIP information from the remote computer.

Off - The Portmaster will not send RIP information to the remote computer, and will not accept RIP information from the remote computer.

Protocol *(menu field)*

This indicates which protocol to use for encapsulating packets into. The options are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MTU *(data entry field)*

This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

Compression *(menu field)*

This indicates whether the PortMaster should use Van Jacobson TCP/IP header compression. This option can increase the performance of interactive TCP sessions if both sides of the link support this option. When configuring SLIP connections, both sides must be configured identically. When configuring PPP connections, if compression is enabled and the remote side does not support compression, the PortMaster will dynamically disable this feature.

PPP Async Map *(data entry field)*

The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments should be set to zero to achieve maximum throughput.

Dialback Users

Location Name *(data entry field)*

This is the name of the location stored in the PortMaster Location Table. After authenticating the user, the PortMaster will disconnect the active connection. It will then schedule an automatic dial to the specified location.

The Hosts Table

Each device attached to the Ethernet (with uses TCP/IP) is assigned a unique internet address. Internet addresses are four byte numerical values. To make it easier to remember these values, most systems on the network support a way of aliasing the internet address to some unique and easy to remember name. This is done on UNIX systems in the /etc/hosts file. The PortMaster also supports the same capability by

maintaining a local hosts table.

The PortMaster hosts table should be used when any of the serial ports is set to prompt the user for a host name on the network. This table is then consulted to determine the correct internet address of the host.

If the hostname specified by the user is not found in the hosts table, then the PortMaster will attempt to match the hostname using the Network Information Service (NIS), also known as "YP". See the section "Global Configuration" for more information on setting the YP server and YP domain.

For information on managing the PortMaster Hosts Table, consult the PortMaster User's Manual.

The Routes Table

The PortMaster Routes table is used when static network routes are required. Static routes are required when the user chooses not to run the Routing Information Protocol (RIP) on the PortMaster, or when hosts or networks attached to the PortMaster do not have the ability to run the RIP protocol. For information on managing the Routes Table, consult the PortMaster User's Manual.

Dialing Out to Remote Locations

On-demand or Continuous dial-out locations automatically initiate dialing to a remote location. Manual connections require a system administrator or a scheduler to initiate dial-out connections. PMconsole provides a facility for the system administrator to initiate the dial-out operation and if necessary to monitor the progress of the dial-out operation.

When setting up a new On-demand location it is recommended to first set up the location as a manual dial-out site for testing purposes and then later switch the location type to On-demand.

Note: When switching a location from Manual to On-demand make sure the dial-out connection has been closed (ie. Reset the serial port)

before updating the location table.

To initiate dial-out connections via a scheduler (ie. cron in UNIX systems) or from a UNIX command line prompt see the "pmdial" command in the UNIX Utilities section of this manual.

To dial to a remote manual location from PMconsole, select the "View" item from the main console menu. A pop-up menu will be displayed. Select "Dialer" from the menu. The window in figure 3-12 will be displayed.

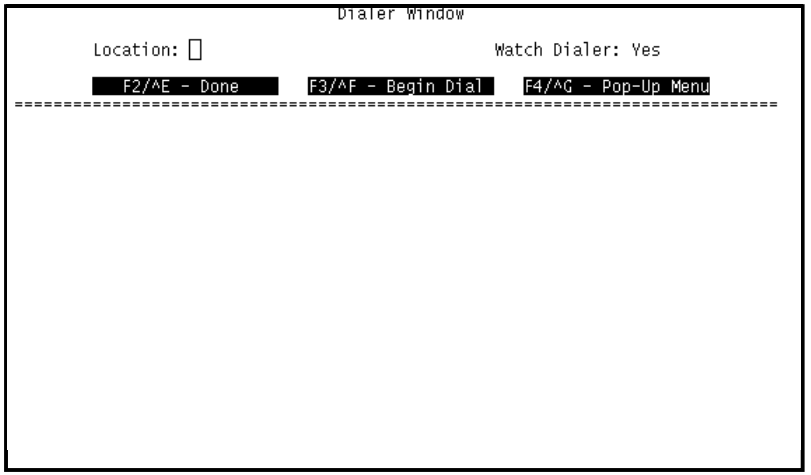


Figure 3-12. Dialer Screen

The Dialer Window has two fields at the top of the screen:

Location *(menu and data entry field)*

This is the location to dial. This field is filled in by typing the name of a location or by selecting one of the locations from the pop-up menu available on the location field (press F4).

Watch Dialer *(menu field)*

Yes - Causes the PortMaster to display the actual progress of the Dial Command Script in the window below.

No - Causes the PortMaster to start dialing the location and immediately returns control to PMconsole for another operation.

To begin dialing to a remote location enter the name in the location field. Then select "Yes" or "No" as the monitoring option. Finally press F3 (or Control-F) to begin dialing.

If "Yes" was selected in the Watch Dialer field the lower portion of the screen will show the location being dialed and the modem port being used. It will then show the execution of the Dial Command Script. The "sendthem" message shows the exact string of characters being sent out the serial port to the modem or remote system enclosed in parenthesis. The "expect" message shows the response string the PortMaster is expecting as a reply enclosed in parenthesis. Immediately following the "expect" string are the characters actually being received by the PortMaster from the modem or remote system. When the PortMaster successfully receives the string is is searching for it will display ":got it" followed by the next send message.

If the dial script completes successfully, the message "Connection Succeeded" will be displayed. Otherwise, after timing out, the message "Connection Failed" will be displayed.

Monitoring Dialnet Connections

The PortMaster allows you to monitor two sets of data communications information. The port statistics or the network statistics can be monitored. Port statistics allow you to view input/output information for each of the ports. This includes things like input bytes, output bytes, errors, active baud rate, and other active parameters. Network statistics allow you to monitor information for each network interface. This includes the ethernet interface and each of the active Dialnet interfaces. Network statistics include input packets, output packets, packet errors, and active interface values.

From the PortMaster Console screen, select the View command. A pop-up menu will be displayed. Select either "Ports" or "Netstats" from the menu.

Port Statistics

After selecting "Ports" from the pop-up menu, a screen similar to Figure 3-13 will be displayed.

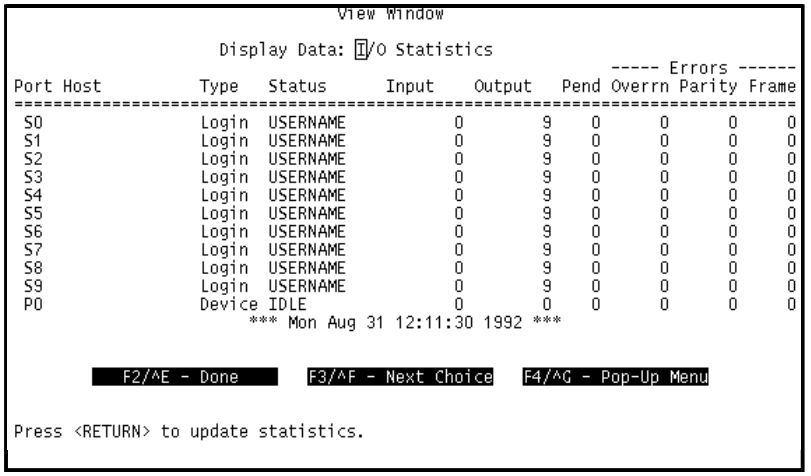


Figure 3-13. View screen

This screen allows you to view the configuration and Status of each port. These values are refreshed each time you press the <RETURN> key.

Two sets of data can be displayed in the view screen. These are the "Active Configuration" and the "I/O Statistics". Both modes display the port, the port type, and the current status of the port.

When "Active Configuration" is selected (using the F3 function key), the screen displays data communications parameters such as baud rate, databits and parity.

When "I/O Statistics" is selected the screen displays input and output information. There are three columns labeled: Input, Output, and Pending. These relate to the number of bytes transferred in or out the port and the number of bytes in the PortMaster output buffer which are pending to be output. There are also three error counts displayed:

Overrun, Parity and Framing errors.

If the PortMaster has more ports than can be displayed on a single screen, the F4 function key can be used to page between different lists of ports.

Network Statistics

After selecting "Netstats" from the pop-up menu, the screen in figure 3-14 will be displayed.

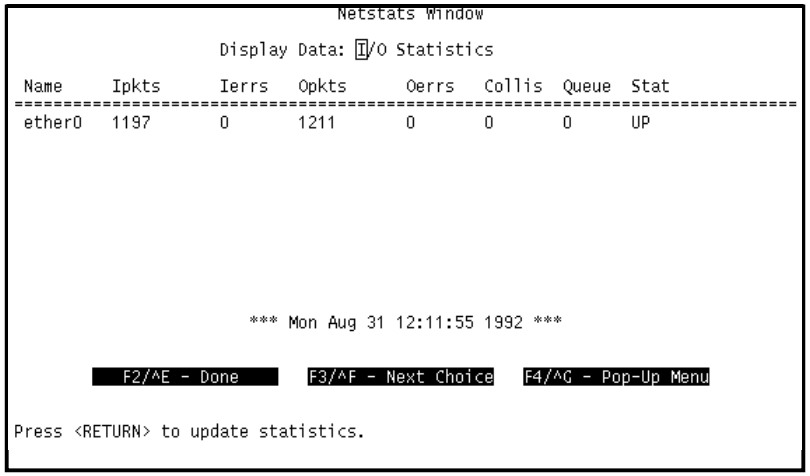


Figure 3-14. Netstats Screen

This screen allows you to view the Status of each network interface in the PortMaster. These values are refreshed each time you press the <RETURN> key. Two sets of data can be displayed in the netstats screen. These are the "Active Configuration" and the "I/O Statistics". Both modes display the name and the current status of the network interface.

When "I/O Statistics" is selected, the values displayed are:

- Name - The name of the network interface. The Ethernet interface is named "ether0" and each interface through the serial ports are named "ptp", followed by a uniquely assigned number.

- Ipkts - The number of input packets received by the interface.
- Ierrs - The number of input packets received by the interface which were dropped due to errors.
- Opkts - The number of output packets sent by the interface.
- Oerrs - The number of output packets which were not sent by the interface due to errors.
- Collis - The number of collisions experienced while trying to send packets out the interface.
- Queue - The number of packets currently queued for transmission on the interface.
- Stat - Either "UP" or "DOWN" indicating whether the interface is active.

When "Active Configuration" is selected, the values displayed are:

- Name - The name of the network interface. The Ethernet interface is named "ether0" and each interface through the serial ports are named "ptp", followed by a uniquely assigned number.
- inet/dest - For the Ethernet interface, the internet address of the PortMaster. For the ptp interfaces, the internet address of the remote system.
- netmask - The netmask of the inet/dest field.
- Routing - "Routing", "Sending", "Listen", or "Quiet" indicating whether RIP protocol packets are being transmitted or received by the interface.
- Stat - Either "UP" or "DOWN" indicating whether the interface is active.

Command Prompt Interface

Any of the previously described administration activities of the PortMaster can be performed using an ASCII terminal connected to a PortMaster serial port or remotely using the telnet protocol to establish an administrative session on the PortMaster. When using an ASCII terminal set the communication parameters of the terminal to match the ones which have been previously set for that port or the default parameters if no settings have been made (Default: 9,600 baud, 8 data bits, 1 stop bit, and no parity).

The following sections describe the commands specific to Dialnet. The PortMaster User's Manual describes commands specific to the Terminal Server functions of the PortMaster.

The User Commands

The following commands are available:

● SET Command - set the port and global parameters. Set values for entries in several tables.

● SHOW Command - shows the status of each port , all ports and global data, or entries in several tables.

● REBOOT Command - reboot the PortMaster to the currently saved configuration.

● RESET Command - reset a port to the currently stored configuration.

● VERSION Command - displays the version number of the PortMaster software.

● QUIT, DONE, EXIT Commands - leave the Command mode.

● SAVE Command - save the current configuration to the PortMaster permanent configuration.

● HELP Command - provides information on each of the PortMaster commands including usage and syntax.

● ADD Command - adds entries to one of the local PortMaster tables.

● DELETE Command - deletes entries from one of the local PortMaster tables.

● DIAL Command - begins dialing to one of the Dialnet Locations.

● IFCONFIG Command - Similar to the ifconfig command on UNIX systems, this provides configuration values for all active interfaces. It also allows modification of active values on the fly.

Using the Commands

To use the Command Prompt Interface, you must login to the PortMaster using a special administrators user name and password. At the login prompt enter:

```
login: !root
```

The PortMaster will request that you enter a password. This is the PortMaster administration password. After initial installation, there is no password. Upon entering the correct password, the program will display the prompt:

```
Command>
```

It is at this point that you can begin to use the PortMaster commands. Each command responds to the minimum number of characters, at least two, that are unique to that command. Command entry is not case sensitive.

SET Command

The SET command is used to set each of the configuration parameters, including global parameters, and port parameters.

Setting Global Dialnet Parameters

The syntax for setting global parameters is:

SET *parameter value*

Where *parameter* is one of:

DIALNET - Whether to enable the Dialnet features of the PortMaster.

Value is on or off.

BROADCAST - Whether to use the high or low broadcast address for the ethernet network. *Value* is high or low.

ROUTING - Whether to use dynamic routing with the RIP protocol. *Value* is On, Broadcast, Listen, or off.

DEFAULT - Whether to broadcast or listen to default routes. *Value* is On, Broadcast, Listen, or off.

ASSIGNED - This value is used for network login users using the PPP protocol. If a network user in the passwords table has been set to use an "Assigned Address", the PortMaster must allocate a temporary IP address to be used for their active session. This IP address is retrieved from a pool of addresses maintained by the PortMaster. The IP address entered will be used as the first free address in the pool of addresses. The PortMaster will allocate as many addresses as there are ports configured for Dialnet access.

Setting Dialnet Port Parameters

Port Type

The syntax for setting the port type is:

SET *port* **[login] [device** *devname***] [twoway] [network** *type***]**

Where *port* is one of S0 - S29. **login, device,** and **twoway** are described in the PortMaster User's manual and set terminal server features for the port.

The **network** option enables Dialnet services of *type* on the port. Valid *types* are:

hardwired - The port is used for a dedicated network connection between two sites. No modem dialing or authentication is required. The port immediately begins running the SLIP or PPP protocol. In this mode, none of the other port types can be used.

dialin - In this mode, the port will accept dial-in only network connections. The remote system (or user) will be required to enter a user id and password. Dial In network connections are controlled by the Password Table described later in this chapter.

dialout - In this mode, the port becomes available for dialing to remote destinations and initiating network connections to those destinations. Dial Out network connections are controlled by the Location Table described later in this chapter.

twoway - In this mode, the port will accept dial-in network connections, as well as being available for dial out to remote destinations.

In any of the three Dial modes the port can also be configured to support other port operations concurrently. If one of the other port types is specified in the set command, both that port type and Dialnet will be enabled.

Dial Group

The Dial Group is used to assign serial ports to modem pools or groups. When the PortMaster initiates a dial-out connection to a remote site, it will only allocate a port which is in the same dial group as has been specified for that location in the Location Table. Valid dial groups are numbered from 0 to 99. The default is 0. The syntax for setting the port group is:

SET *port* **group** *value*

Where *port* is one of S0-S29.

value is a number between 0 and 99.

Setting Hardwired Port Parameters

The syntax for setting port parameters is:

SET *port parameter value*

Where *port* is one of S0-S29.

Parameter and *value* is one of:

ADDRESS - The internet address of the directly attached remote PortMaster or computer. Value is a valid host internet address in the form:

XXX.XXX.XXX.XXX

where the XXX is a decimal number between 0 and 255. If the destination is using subnetting, a netmask should be entered following the destination internet address using the same format as the internet address above. If the destination address is set to 0.0.0.0, the port will be disabled.

ROUTING - This indicates whether the PortMaster should use the RIP protocol through the Interface. The following options are available:

On - The Portmaster will send RIP information to the remote computer and will accept RIP information from the remote computer.

Broadcast - The Portmaster will send RIP information to the remote computer, but will not accept RIP information from the remote computer.

Listen - The Portmaster will not send RIP information to the remote computer, but will accept RIP information from the remote computer.

Off - The Portmaster will not send RIP information to the remote computer, and will not accept RIP information from the remote computer.

COMPRESS - Whether to enable Van Jacobson TCP/IP header compression. Valid values are: ON or OFF.

PROTOCOL - The protocol to use for encapsulating packets. Valid values are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MAP - The PPP Async map. This value should be entered in hex

notation (ie. 0x00000000). The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments should be set to zero to achieve maximum throughput.

MTU - This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

SHOW Command

The SHOW command allows you to look at the current settings for any of the serial ports and for the parallel port. It also lets you list the contents of several tables.

The syntax for the show commands are:

port **-** Input/output, active & default configuration for the port.

ALL - Show global configuration and summary of all ports.

table - Show summary of the specified table. Valid values are: hosts, location, or password.

memory - Show system memory usage.

netconns - Show system wide TCP and UDP network connections.

routes - Show the active and static routing table. Fields displayed include; destination - the remote host or network, gateway - the next hop system, metric - the hop count to the destination, interface - which network interface to use, flags -

"H" - A host route

"N" - A network or subnet route

"S" - A static route

"L" - A route to a directly attached network or host

"D" - A dynamic route

"C" - A route which has recently changed

"O" - An obsolete route (scheduled for deletion)

netstat - Input/output statistics summary for all network interfaces.

user - The configuration settings for a user in the password table.

location - The configuration settings for a dialnet location.

For example:

show all

Shows the settings of all global parameters and a summary of each of the individual ports. This display is similar to the VIEW command in PMconsole described earlier in this chapter.

show s1

Displays the full set of configuration values for serial port S1. If "Extended" mode is turned on (described above), a more detailed set of information will be displayed.

show table hosts

Displays the currently stored hosts table.

REBOOT Command

Reboots the PortMaster, reloading the currently saved parameters. All ports will be set to the state of the last saved configuration. The syntax is:

REBOOT

RESET Command

Reset a port to it's currently stored values. Any terminal that is logged in through the PortMaster will be disconnected. You will need to log in again. The syntax is:

RESET port

- or -

RESET ALL

Where port is one of S0 - S29 or P0

QUIT, DONE, EXIT Command

These commands will log you out of the PortMaster command mode and will return you to login mode.

SAVE Command

Saves the current port configurations or configuration tables in the PortMaster non-volatile storage. The syntax is:

SAVE ALL

- or -

SAVE port

- or -

SAVE ROUTES

- or -

SAVE PASSWORD

- or -

SAVE LOCATION

Where port is one of S0 - S29 or P0. Using SAVE ALL will store all global parameters in the PortMaster.

Using SAVE ROUTES will save static routes (Routes which were created using the ADD ROUTE command).

Using SAVE PASSWORD will save the password (Passwords which were created using the ADD NETUSER or ADD USER command).

Using SAVE LOCATION will save the location table (Locations which were created using the ADD LOCATION command).

HELP Command

Provides information on the usage and syntax of the PortMaster commands. For example:

HELP SET

displays message:

```
Valid SET options are:
  s0-s9,p0  - Set configuration for a port
  address   - Set the PortMaster Internet Address
  assigned  - Set first pooled address for dialin users
  broadcast - Set the broadcast IP address
  debug     - Set the console debugging level
  default   - Set the propagation of default routes
  dialnet   - Enable or disable Dialnet functions
  domain    - Set the NIS domain
  ethernet  - Enable or disable the ethernet interface
  gateway   - Set the default route
  host      - Set the default login host
  location  - Set value for a dialnet location
  loghost   - Set the network syslog host
  netmask   - Set the ethernet interface netmask
  password  - Set the administration password
  routing   - Set the ethernet routing options
  snmp      - Enable or disable SNMP, or set community
  telnet    - Set the TCP port for telnet administration
  user      - Set value for a user in the password table
  ypserver  - Set the NIS server
Use "help set option" for further assistance
```


IFCONFIG Command

When the IFCONFIG command is entered, the active configuration of each network interface is displayed. Each interface is displayed in two lines. The first line has the name of the interface (ie ether0, ptp1) and various flags for that interface. Possible flags are:

UP - The interface is UP and running.

DOWN - The interface is DOWN, it is not in use.

BROADCAST - The network is a broadcast network (ie. ethernet).

POINT_TO_POINT - The network is a point to point connection.

PRIVATE - No routing information will be transmitted on this interface. If private is not shown as a flag, routing information will be transmitted to the interface.

SUSPENDED - This On-Demand Dialnet interface is available for use, but does not have an active telephone connection to the remote site.

The second line will show:

inet - the internet address of the interface if it is an ethernet interface.

dest - the destination internet address if it is a dialnet interface.

netmask - The netmask for the "inet" or "dest".

broadcast - The broadcast address if it is an ethernet interface.

DIAL Command

Initiates dialing to a remote location in the location table. The syntax is:

DIAL *location_name* **[-x]**

Where *location_name* is the name of a location in the location table. If the **-x** option is specified, the PortMaster will display the actual execution of the dial script to the user if the user is logged in on one of the serial ports.

Managing the Location Table

The Location Table is used for all dial out network connections to remote locations. Each entry specifies a remote location, the operational parameters for that location, and information on how to establish a connection to that location.

Adding and Deleting Locations

Two commands have been established for adding and deleting locations:

- ADD Command - adds entries to the local Locations table.
- DELETE Command - deletes entries from the local Locations table.

ADD Command

Add an entry to the local Locations table. The syntax is:

ADD location *location_name*

Where:
location_name - The name of the remote location. This can be any 12 character name to identify the remote location. It is generally not the name of a single system, but rather an identifier which represents the entire location (ie. the city or the company name at that location).

DELETE Command

Delete an entry from the local Locations table. The syntax is:

DELETE location *location_name*

Where:
location_name - is the name of a previously created location.

SET Command

Once a location has been created, all of the configuration parameters associated with that location can be set or modified with the SET command. The syntax is:

SET *location_name parameter value*

Where:
location_name - is the name of a previously created location.

parameter is one of:

ON_DEMAND - This parameter notifies the PortMaster to dial and establish a connection to the remote location only when packets are queued for that location. In this mode the PortMaster creates a network interface and the appropriate routing information to notify the attached local area network of the connectivity to the remote site. This is done even though there may not be an actual physical connection to that site at the time. There is no value required for this setting.

Note: When switching a location from Manual to On-demand make sure the dial-out connection has been closed (ie. Reset the serial port) before updating the location table.

CONTINUOUS - This parameter notifies the PortMaster to always keep the dial-up connection to the remote site active. If the telephone line is lost, the PortMaster will automatically initiate a re-dial to that location.

MANUAL - This parameter notifies the PortMaster to only dial to the remote location when an operator requests the connection. This can be done with the DIAL command or it can be invoked from the "pmdial" program installed on one of the UNIX systems on the network. This also allows connections to be scheduled by using the UNIX cron scheduler to call pmdial.

ADDRESS - The Internet Address of the system at the remote end of the dial-up connection. A valid IP address using standard dot notation should be entered as the value.

NETMASK - The netmask of the system at the remote end of the dial-up connection. A valid netmask using standard dot

notation should be entered as the value.

ROUTING - This indicates whether the PortMaster should use the RIP protocol through the Interface. The following options are available:

On - The Portmaster will send RIP information to the remote computer and will accept RIP information from the remote computer.

Broadcast - The Portmaster will send RIP information to the remote computer, but will not accept RIP information from the remote computer.

Listen - The Portmaster will not send RIP information to the remote computer, but will accept RIP information from the remote computer.

Off - The Portmaster will not send RIP information to the remote computer, and will not accept RIP information from the remote computer.

COMPRESS - Whether to enable Van Jacobson TCP/IP header compression. Valid values are: ON or OFF.

PROTOCOL - The protocol to use for encapsulating packets. Valid values are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

MAP - The PPP Async map. This value should be entered in hex notation (ie. 0x00000000). The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments

should be set to zero to achieve maximum throughput.

MTU - This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

IDLETIME - For Manual and On-Demand type locations this specifies the number of minutes the line should be idle (no network traffic to the remote site) before disconnecting the telephone connection. A value between 0 and 240 can be specified. If 0 is chosen, the connection will not time out.

MAXPORTS - The number of dial-out ports the PortMaster can use for this network connection. If 0 is selected dialing to this location will be disabled. If more than 1 is selected the PortMaster will use the "High Water Mark" to decide when to open and dial additional lines. If more than one line is open to the remote location the PortMaster will automatically use load balancing to achieve the maximum throughput of all lines combined. If multiple lines are open the "Idle Timeout" value will be used to decide when to disconnect unneeded lines. A value between 0 and 10 may be entered.

HIGH_WATER - The number of bytes of queued network traffic required to open an additional dial-out line to the remote site. This value is only used when the "Maximum Ports" is greater than 1 and there are available modems on the PortMaster. Setting a very small number will cause the PortMaster to

quickly use all of the ports specified in "Maximum Ports". Generally, interactive terminal traffic has no more than a few hundred bytes queued at any one time. However, file transfers (ie. ftp) will queue several thousand bytes. These size differences should be used when deciding what number to use for the High Water Mark. A value between 0 and 65535 may be entered.

GROUP - The modem pool group to which this location is assigned. Only modem ports which are configured with this Group Number will be available for dial-out to this location. This can be used to reserve modems for dial-up to specific locations or to differentiate different types of modems which would be compatible with the remote location. A value between 0 and 100 may be entered.

SCRIPT - The dial command script is used to issue commands to the modem (or other similar equipment) to dial the correct number and establish the link. In addition, this script is used for authentication (password security) at the remote site. Three values must be entered with the SCRIPT parameter. The syntax is:

SET *location_name* **SCRIPT** *line_no send reply*

Where *line_no* is a number between 1 and 6. *send* is a string of up to 30 characters surrounded by double quotes. *reply* is a string of up to 30 characters surrounded by double quotes. Each line_no represents one send-reply combination.

Each "Send" string is issued out of the PortMaster serial port to the modem or remote computer. Each "Reply" string is used by the PortMaster to verify that the previous "Send" string was properly received by the modem or remote computer and the PortMaster should transmit the next "Send" string.

Any printing ASCII character can be placed in the send or reply strings. In addition, the following special characters are available:

\r - This is replaced by the ASCII carriage return.

\n - This is replaced by the ASCII line feed.

\0XX - This is replaced by the octal digit in the XX.

\\ - This is replaced by the single backslash (\).

Generally Send strings would end with the "\r" character.

It is important that the last entry in the Dial Command Script be a Reply string which indicates that the remote location is ready to begin receiving network packets. This is used to activate the TCP/IP protocol coming from the PortMaster. When connecting to a remote PortMaster the final reply string to look for should be "SL/IP". For other products, consult that vendors users manual.

The following Dial Command Script is an example of how to establish a connection between two PortMasters which have modems supporting the "AT" dial command syntax:

Send	Reply
`ATDT18005551212\r`	`CONNECT`
`\r`	`login:`
`my_location_name\r`	
`ssword:`	
`my_password\r`	`SL/IP`

The "18005551212" would be replaced by the actual telephone number of the remote modem. "my_location_name" would be replaced by the actual user name for your location. "my_password" would be replaced by the actual password set up at the remote site which corresponds to the user name.

SHOW Command

List a summary of the locations table or the detail of a specific location.

The syntax for the show command is:

SHOW TABLE LOCATION - to display a summary of the location table.

SHOW *location_name* - where *location_name* is the name of a previously created location. This will display all of the configuration information for the location.

SAVE Command

Save Locations into the permanent PortMaster configuration. The syntax is:

SAVE LOCATION

Managing the Passwords Table

The Passwords Table provides a security and network parameter initialization scheme for locations or individual computers dialing into the PortMaster to establish remote network connections. Modem ports configured to accept Dialnet Dial-in sessions initially prompt for a login user name. After receiving the login name, the PortMaster checks the name against the list of names in the local Passwords Table. If found, the PortMaster initiates password authentication of the remote

system. If successfully validated (the password received matches the password stored in the Passwords Table) the PortMaster uses the configuration parameters for the user to initiate a Dialnet connection. If the user has been defined as a "Dialback" network user, the PortMaster will disconnect the line (by dropping DTR) and begin dialing to the pre-defined dialnet location. If the user is a "Normal" network user, the PortMaster will respond with one of the following messages:

SL/IP session from (149.198.9.1) to remote beginning....

PPP session from (149.198.9.1) to remote beginning....

For SLIP connections, the PortMaster then enables the network interface and any associated routing information as indicated in the Passwords Table. PPP connections are negotiated using the standard PPP option negotiation prior to enabling the network interface. Finally it begins accepting and sending IP packets over the connection.

If the login name does not match an entry in the Passwords Table and the port is configured to support User Logins it automatically forwards the session to the designated host on the local network for password authentication.

Five commands have been established for managing the local passwords table:

- ADD Command - adds entries to the local passwords table.
- DELETE Command - deletes entries from the local passwords table.
- SET Command - modifies parameters for a user.
- SHOW Command - shows the entries in the local passwords table.
- SAVE Command - saves the passwords in the permanent configuration.

Consult the PortMaster User's Manual for information on managing non-dialnet users.

ADD Command

Add an entry to the local passwords table. The syntax is:

ADD netuser *name* **password** *password*

Where:

name - is the login name of the remote system requesting a Dialnet dial-in session. This name can be up to 8 characters in length and is the user name to be supplied at the "login:" prompt from the PortMaster. If the PortMaster is configured to support User Logins in addition to SLIP sessions, the SLIP login name must not conflict with a login name on the designated login host on the network.

password - is the password for the dialnet user. This is a string of up to 16 ASCII characters.

DELETE Command

Delete an entry from the local passwords table. The syntax is:

DELETE user *name*

Where:
name - is the login name of an existing password entry.

SET Command

There are 9 parameters which can be set for Dialnet users. The syntax is:

SET user *name* **parameter** *value*

The name of a previously added dialnet user should be entered as the name. The parameters and their values are:

password - The password for the user. This is a string of up to 16 ASCII characters.

address - This is the IP address of the remote computer. Valid

Addresses are:

The specific IP address of the remote computer. The remote computer will only be allowed to operate with this IP address.

"Assigned" - The PortMaster will assign a temporary IP address for this session. This address will be retrieved from the pool of addresses being maintained by the PortMaster. See the Global Configuration window for more information on Assigned Addresses.

"Negotiated" - This option is valid only for PPP sessions. The PortMaster will attempt to learn the IP address of the remote computer automatically.

netmask - The netmask of the system at the remote end of the dial-up connection should be entered.

routing - This indicates whether the PortMaster should use the RIP protocol through the Interface. The following options are available:

On - The Portmaster will send RIP information to the remote computer and will accept RIP information from the remote computer.

Broadcast - The Portmaster will send RIP information to the remote computer, but will not accept RIP information from the remote computer.

Listen - The Portmaster will not send RIP information to the remote computer, but will accept RIP information from the remote computer.

Off - The Portmaster will not send RIP information to the remote computer, and will not accept RIP information from the remote computer.

protocol - This indicates which protocol to use for encapsulating packets into. The options are:

SLIP - Use the Serial Line Internet Protocol.

PPP - Use the Point to Point Protocol.

mtu - This is the desired Maximum Transmission Unit to be used for the Interface. PPP connections can be set to values from 100-1500 and the remote computer can negotiate a smaller value than the setting entered. SLIP connections can be set to values from 100-1006. Maximum Transmission Unit defines the largest "frame" or "packet" which can be sent through this connection. If a packet is too large to fit, it will be automatically fragmented into smaller pieces by the PortMaster. Setting this to smaller (ie. less than 300) values can be useful when using multi-line load balancing or for enhanced interactive performance while running concurrent file transfers.

compress - This indicates whether the PortMaster should use Van Jacobson TCP/IP header compression. This option can increase the performance of interactive TCP sessions if both sides of the link support this option. When configuring SLIP connections, both sides must be configured identically. When configuring PPP connections, if compression is enabled and the remote side does not support compression, the PortMaster will dynamically disable this feature. Valid values are ON or OFF.

map - The PPP protocol supports the escaping of non-printing ASCII in the datastream. Escaping means that specified characters will not be sent through the line, but instead will be replaced by a special set of characters which the remote site will interpret as the original character. The PPP Async Map is a bit-map of characters which should be escaped. The first (or zero) bit corresponds to the first ASCII character "NULL" and so on. Most environments should be set to zero to achieve maximum throughput. This value should be entered in hex notation (ie. 0x00000000).

dialback - This indicates that this is a dialback user. The name of the location stored in the PortMaster Location Table should be entered as the value. After authenticating the user, the

PortMaster will disconnect the active connection. It will then schedule an automatic dial to the specified location.

SHOW Command

List the contents of the active passwords table or a single dialnet user.

The syntax for the show command is:

SHOW table password

- or -

SHOW user *name*

SAVE Command

Save passwords into the permanent PortMaster configuration. The syntax is:

SAVE password

The Hosts Table

Each device attached to the Ethernet (with uses TCP/IP) is assigned a unique internet address. Internet addresses are four byte numerical values. To make it easier to remember these values, most systems on the network support a way of aliasing the internet address to some unique and easy to remember name. This is done on UNIX Workstations in the /etc/hosts file. The PortMaster also supports the same capability by maintaining a local hosts table.

The PortMaster hosts table should be used when any of the ports are set to prompt the user for a host name on the network. This table is then consulted to determine the correct internet address of the host.

If the hostname specified by the user is not found in the hosts table, then the PortMaster will attempt to match the hostname using the Network Information Service (NIS), also known as "YP". See the

section "Setting Global Parameters" for more information on setting the YP server and YP domain.

Consult the PortMaster User's Manual for information on managing the Hosts Table.

The Routes Table

The PortMaster Routes table is used when static network routes are required. Static routes are required when the user chooses not to run the Routing Information Protocol (RIP) on the PortMaster, or when hosts or networks attached to the PortMaster do not have the ability to run the RIP protocol.

For information on managing the Routes Table consult the PortMaster User's Manual.

TCP/IP Networking Primer __CHAPTER 4

This chapter is intended as an aid to system administrators who are new to wide area networking or TCP/IP in general. It is helpful for understanding how TCP and IP function on a network and provides an overview of routing concepts. This information will help the reader understand some of the elements of proper internet address assignment and the underlying mechanisms which Dialnet uses to perform IP routing.

The remainder of this chapter was developed and written at Rutgers State University of New Jersey. It was selected for use in this manual because it is commonly used by members of the Internet community (the largest wide area TCP/IP network in the world) to become familiar with using the Internet.

Livingston Enterprises makes no warranty as to the correctness or suitability of the following text for a specific purpose, but does believe the following information will be helpful to network users and administrators.

Introduction to TCP/IP Networking

RUTGERS
The State University of New Jersey
Center for Computers and Information Services
Laboratory for Computer Science Research
3 July 1987

This is an introduction to the Internet networking protocols (TCP/IP). It includes a summary of the facilities available and brief descriptions of the major protocols in the family.

Copyright (C) 1987, Charles L. Hedrick. Anyone may reproduce this document, in whole or in part, provided that: (1) any copy or republication of the entire document must show Rutgers University as the source, and must include this notice; and (2) any other use of this material must reference this manual and Rutgers University, and the

fact that the material is copyright by Charles Hedrick and is used by permission.

This document is a brief introduction to TCP/IP, followed by advice on what to read for more information. This is not intended to be a complete description. It can give you a reasonable idea of the capabilities of the protocols. But if you need to know any details of the technology, you will want to read the standards yourself. Throughout the text, you will find references to the standards, in the form of "RFC" or "IEN" numbers. These are document numbers. The final section of this document tells you how to get copies of those standards.

What is TCP/IP?

TCP/IP is a set of protocols developed to allow cooperating computers to share resources across a network. It was developed by a community of researchers centered around the ARPAnet. Certainly the ARPAnet is the best-known TCP/IP network. However as of June, 87, at least 130 different vendors had products that support TCP/IP, and thousands of networks of all kinds use it.

First some basic definitions. The most accurate name for the set of protocols we are describing is the "Internet protocol suite". TCP and IP are two of the protocols in this suite. (They will be described below.) Because TCP and IP are the best known of the protocols, it has become common to use the term TCP/IP or IP/TCP to refer to the whole family. It is probably not worth fighting this habit. However this can lead to some oddities. For example, I find myself talking about NFS as being based on TCP/IP, even though it doesn't use TCP at all. (It does use IP. But it uses an alternative protocol, UDP, instead of TCP. All of this alphabet soup will be unscrambled in the following pages.)

The Internet is a collection of networks, including the Arpanet, NSFnet, regional networks such as NYsernet, local networks at a number of University and research institutions, and a number of military networks. The term "Internet" applies to this entire set of networks. The subset of them that is managed by the Department of Defense is referred to as the

"DDN" (Defense Data Network). This includes some research-oriented networks, such as the Arpanet, as well as more strictly military ones. (Because much of the funding for Internet protocol developments is done via the DDN organization, the terms Internet and DDN can sometimes seem equivalent.) All of these networks are connected to each other. Users can send messages from any of them to any other, except where there are security or other policy restrictions on access. Officially speaking, the Internet protocol documents are simply standards adopted by the Internet community for its own use. More recently, the Department of Defense issued a MILSPEC definition of TCP/IP. This was intended to be a more formal definition, appropriate for use in purchasing specifications. However most of the TCP/IP community continues to use the Internet standards. The MILSPEC version is intended to be consistent with it.

Whatever it is called, TCP/IP is a family of protocols. A few provide "low-level" functions needed for many applications. These include IP, TCP, and UDP. (These will be described in a bit more detail later.) Others are protocols for doing specific tasks, e.g. transferring files between computers, sending mail, or finding out who is logged in on another computer. Initially TCP/IP was used mostly between minicomputers or mainframes. These machines had their own disks, and generally were self-contained. Thus the most important "traditional" TCP/IP services are:

File Transfer

The file transfer protocol (FTP) allows a user on any computer to get files from another computer, or to send files to another computer. Security is handled by requiring the user to specify a user name and password for the other computer. Provisions are made for handling file transfer between machines with different character set, end of line conventions, etc. This is not quite the same thing as more recent "network file system" or "netbios" protocols, which will be described below. Rather, FTP is a utility that you run any time you want to access a file on another system. You use it to copy the file to your own system. You then work with the local copy. (See RFC 959 for

specifications for FTP.)

Remote Login

The network terminal protocol (TELNET) allows a user to log in on any other computer on the network. You start a remote session by specifying a computer to connect to. From that time until you finish the session, anything you type is sent to the other computer. Note that you are really still talking to your own computer. But the telnet program effectively makes your computer invisible while it is running. Every character you type is sent directly to the other system. Generally, the connection to the remote computer behaves much like a dialup connection. That is, the remote system will ask you to log in and give a password, in whatever manner it would normally ask a user who had just dialed it up. When you log off of the other computer, the telnet program exits, and you will find yourself talking to your own computer. Microcomputer implementations of telnet generally include a terminal emulator for some common type of terminal. (See RFC's 854 and 855 for specifications for telnet. By the way, the telnet protocol should not be confused with Telenet, a vendor of commercial network services.)

Computer Mail

This allows you to send messages to users on other computers. Originally, people tended to use only one or two specific computers. They would maintain "mail files" on those machines. The computer mail system is simply a way for you to add a message to another user's mail file. There are some problems with this in an environment where microcomputers are used. The most serious is that a micro is not well suited to receive computer mail. When you send mail, the mail software expects to be able to open a connection to the addressee's computer, in order to send the mail. If this is a microcomputer, it may be turned off, or it may be running an application other than the mail system. For this reason, mail is normally handled by a larger system, where it is practical to have a mail server running all the time. Microcomputer mail software then becomes a user interface that retrieves mail from the mail server. (See RFC 821 and 822 for specifications for computer

mail. See RFC 937 for a protocol designed for microcomputers to use in reading mail from a mail server.)

These services should be present in any implementation of TCP/IP, except that micro-oriented implementations may not support computer mail. These traditional applications still play a very important role in TCP/IP-based networks. However more recently, the way in which networks are used has been changing. The older model of a number of large, self-sufficient computers is beginning to change. Now many installations have several kinds of computers, including microcomputers, workstations, minicomputers, and mainframes. These computers are likely to be configured to perform specialized tasks. Although people are still likely to work with one specific computer, that computer will call on other systems on the net for specialized services. This has led to the "server/client" model of network services. A server is a system that provides a specific service for the rest of the network. A client is another system that uses that service. (Note that the server and client need not be on different computers. They could be different programs running on the same computer.) Here are the kinds of servers typically present in a modern computer setup. Note that these computer services can all be provided within the framework of TCP/IP.

Network File Systems

This allows a system to access files on another computer in a somewhat more closely integrated fashion than FTP. A network file system provides the illusion that disks or other devices from one system are directly connected to other systems. There is no need to use a special network utility to access a file on another system. Your computer simply thinks it has some extra disk drives. These extra "virtual" drives refer to the other system's disks. This capability is useful for several different purposes. It lets you put large disks on a few computers, but still give others access to the disk space. Aside from the obvious economic benefits, this allows people working on several computers to share common files. It makes system maintenance and backup easier, because you don't have to worry about updating and backing up copies on lots of different machines. A number of vendors now offer

high-performance diskless computers. These computers have no disk drives at all. They are entirely dependent upon disks attached to common "file servers". (See RFC's 1001 and 1002 for a description of PC-oriented NetBIOS over TCP. In the workstation and minicomputer area, Sun's Network File System is more likely to be used. Protocol specifications for it are available from Sun Microsystems.)

Remote Printing

This allows you to access printers on other computers as if they were directly attached to yours. (The most commonly used protocol is the remote lineprinter protocol from Berkeley Unix. Unfortunately, there is no protocol document for this. However the C code is easily obtained from Berkeley, so implementations are common.)

Remote Execution

This allows you to request that a particular program be run on a different computer. This is useful when you can do most of your work on a small computer, but a few tasks require the resources of a larger system. There are a number of different kinds of remote execution. Some operate on a command by command basis. That is, you request that a specific command or set of commands should run on some specific computer. (More sophisticated versions will choose a system that happens to be free.) However there are also "remote procedure call" systems that allow a program to call a subroutine that will run on another computer. (There are many protocols of this sort. Berkeley Unix contains two servers to execute commands remotely: rsh and rexec. The man pages describe the protocols that they use. The user-contributed software with Berkeley 4.3 contains a "distributed shell" that will distribute tasks among a set of systems, depending upon load. Remote procedure call mechanisms have been a topic for research for a number of years, so many organizations have implementations of such facilities. The most widespread commercially-supported remote procedure call protocols seem to be Xerox's Courier and Sun's RPC. Protocol documents are available from Xerox and Sun. There is a public implementation of Courier over TCP as part of the

user-contributed software with Berkeley 4.3. An implementation of RPC was posted to Usenet by Sun, and also appears as part of the user-contributed software with Berkeley 4.3.)

Name Servers

In large installations, there are a number of different collections of names that have to be managed. This includes users and their passwords, names and network addresses for computers, and accounts. It becomes very tedious to keep this data up to date on all of the computers. Thus the databases are kept on a small number of systems. Other systems access the data over the network. (RFC 822 and 823 describe the name server protocol used to keep track of host names and Internet addresses on the Internet. This is now a required part of any TCP/IP implementation. IEN 116 describes an older name server protocol that is used by a few terminal servers and other products to look up host names. Sun's Yellow Pages system is designed as a general mechanism to handle user names, file sharing groups, and other databases commonly used by Unix systems. It is widely available commercially. Its protocol definition is available from Sun.)

Terminal Servers

Many installations no longer connect terminals directly to computers. Instead they connect them to terminal servers. A terminal server is simply a small computer that only knows how to run telnet (or some other protocol to do remote login). If your terminal is connected to one of these, you simply type the name of a computer, and you are connected to it. Generally it is possible to have active connections to more than one computer at the same time. The terminal server will have provisions to switch between connections rapidly, and to notify you when output is waiting for another connection. (Terminal servers use the telnet protocol, already mentioned. However any real terminal server will also have to support name service and a number of other protocols.)

Network-oriented Window Systems

Until recently, high- performance graphics programs had to execute on a computer that had a bit-mapped graphics screen directly attached to it. Network window systems allow a program to use a display on a different computer. Full-scale network window systems provide an interface that lets you distribute jobs to the systems that are best suited to handle them, but still give you a single graphically-based user interface. (The most widely-implemented window system is X. A protocol description is available from MIT's Project Athena. A reference implementation is publically available from MIT. A number of vendors are also supporting NeWS, a window system defined by Sun. Both of these systems are designed to use TCP/IP.) Note that some of the protocols described above were designed by Berkeley, Sun, or other organizations. Thus they are not officially part of the Internet protocol suite. However they are implemented using TCP/IP, just as normal TCP/IP application protocols are. Since the protocol definitions are not considered proprietary, and since commercially- supported implementations are widely available, it is reasonable to think of these protocols as being effectively part of the Internet suite. Note that the list above is simply a sample of the sort of services available through TCP/IP. However it does contain the majority of the "major" applications. The other commonly-used protocols tend to be specialized facilities for getting information of various kinds, such as who is logged in, the time of day, etc. However if you need a facility that is not listed here, we encourage you to look through the current edition of Internet Protocols (currently RFC 1011), which lists all of the available protocols, and also to look at some of the major TCP/IP implementations to see what various vendors have added.

General description of the TCP/IP protocols

TCP/IP is a layered set of protocols. In order to understand what this means, it is useful to look at an example. A typical situation is sending mail. First, there is a protocol for mail. This defines a set of commands which one machine sends to another, e.g. commands to specify who the sender of the message is, who it is being sent to, and then the text of the message. However this protocol assumes that there is a way to communicate reliably between the two computers. Mail, like other application protocols, simply defines a set of commands and messages to be sent. It is designed to be used together with TCP and IP. TCP is responsible for making sure that the commands get through to the other end. It keeps track of what is sent, and retransmitts anything that did not get through. If any message is too large for one datagram, e.g. the text of the mail, TCP will split it up into several datagrams, and make sure that they all arrive correctly. Since these functions are needed for many applications,
they are put together into a separate protocol, rather than being part of the specifications for sending mail. You can think of TCP as forming a library of routines that applications can use when they need reliable network communications with another computer. Similarly, TCP calls on the services of IP. Although the services that TCP supplies are needed by many applications, there are still some kinds of applications that don't need them. However there are some services that every application needs. So these services are put together into IP. As with TCP, you can think of IP as a library of routines that TCP calls on, but which is also available to applications that don't use TCP. This strategy of building several levels of protocol is called "layering". We think of the applications programs such as mail, TCP, and IP, as being separate "layers", each of which calls on the services of the layer below it. Generally, TCP/IP applications use 4 layers:

- an application protocol such as mail

- a protocol such as TCP that provides services need by many

applications

- IP, which provides the basic service of getting datagrams to their destination

- the protocols needed to manage a specific physical medium, such as Ethernet or a point to point line.

TCP/IP is based on the "catenet model". (This is described in more detail in IEN 48.) This model assumes that there are a large number of independent networks connected together by gateways. The user should be able to access computers or other resources on any of these networks. Datagrams will often pass through a dozen different networks before getting to their final destination. The routing needed to accomplish this should be completely invisible to the user. As far as the user is concerned, all he needs to know in order to access another system is an "Internet address". This is an address that looks like 128.6.4.194. It is actually a 32-bit number. However it is normally written as 4 decimal numbers, each representing 8 bits of the address. (The term "octet" is used by Internet documentation for such 8-bit chunks. The term "byte" is not used, because TCP/IP is supported by some computers that have byte sizes other than 8 bits.) Generally the structure of the address gives you some information about how to get to the system. For example, 128.6 is a network number assigned by a central authority to Rutgers University. Rutgers uses the next octet to indicate which of the campus Ethernets is involved. 128.6.4 happens to be an Ethernet used by the Computer Science Department. The last octet allows for up to 254 systems on each Ethernet. (It is 254 because 0 and 255 are not allowed, for reasons that will be discussed later.) Note that 128.6.4.194 and 128.6.5.194 would be different systems. The structure of an Internet address is described in a bit more detail later.

Of course we normally refer to systems by name, rather than by Internet address. When we specify a name, the network software looks it up in a database, and comes up with the corresponding Internet address. Most of the network software deals strictly in terms of the address. (RFC 882 describes the name server technology used to handle this lookup.)

TCP/IP is built on "connectionless" technology. Information is transfered as a sequence of "datagrams". A datagram is a collection of data that is sent as a single message. Each of these datagrams is sent through the network individually. There are provisions to open connections (i.e. to start a conversation that will continue for some time). However at some level, information from those connections is broken up into datagrams, and those datagrams are treated by the network as completely separate. For example, suppose you want to transfer a 15000 octet file. Most networks can't handle a 15000 octet datagram. So the protocols will break this up into something like 30 500-octet datagrams. Each of these datagrams will be sent to the other end. At that point, they will be put back together into the 15000-octet file. However while those datagrams are in transit, the network doesn't know that there is any connection between them. It is perfectly possible that datagram 14 will actually arrive before datagram 13. It is also possible that somewhere in the network, an error will occur, and some datagram won't get through at all. In that case, that datagram has to be sent again.

Note by the way that the terms "datagram" and "packet" often seem to be nearly interchangable. Technically, datagram is the right word to use when describing TCP/IP. A datagram is a unit of data, which is what the protocols deal with. A packet is a physical thing, appearing on an Ethernet or some wire. In most cases a packet simply contains a datagram, so there is very little difference. However they can differ. When TCP/IP is used on top of X.25, the X.25 interface breaks the datagrams up into 128-byte packets. This is invisible to IP, because the packets are put back together into a single datagram at the other end before being processed by TCP/IP. So in this case, one IP datagram would be carried by several packets. However with most media, there are efficiency advantages to sending one datagram per packet, and so the distinction tends to vanish.

The TCP level

Two separate protocols are involved in handling TCP/IP datagrams.

TCP (the "transmission control protocol") is responsible for breaking up the message into datagrams, reassembling them at the other end, resending anything that gets lost, and putting things back in the right order. IP (the "internet protocol") is responsible for routing individual datagrams. It may seem like TCP is doing all the work. And in small networks that is true. However in the Internet, simply getting a datagram to its destination can be a complex job. A connection may require the datagram to go through several networks at Rutgers, a serial line to the John von Neuman Supercomputer Center, a couple of Ethernets there, a series of 56Kbaud phone lines to another NSFnet site, and more Ethernets on another campus. Keeping track of the routes to all of the destinations and handling incompatibilities among different transport media turns out to be a complex job. Note that the interface between TCP and IP is fairly simple. TCP simply hands IP a datagram with a destination. IP doesn't know how this datagram relates to any datagram before it or after it.

It may have occurred to you that something is missing here. We have talked about Internet addresses, but not about how you keep track of multiple connections to a given system. Clearly it isn't enough to get a datagram to the right destination. TCP has to know which connection this datagram is part of. This task is referred to as "demultiplexing." In fact, there are several levels of demultiplexing going on in TCP/IP. The information needed to do this demultiplexing is contained in a series of "headers". A header is simply a few extra octets tacked onto the beginning of a datagram by some protocol in order to keep track of it. It's a lot like putting a letter into an envelope and putting an address on the outside of the envelope. Except with modern networks it happens several times. It's like you put the letter into a little envelope, your secretary puts that into a somewhat bigger envelope, the campus mail center puts that envelope into a still bigger one, etc. Here is an overview of the headers that get stuck on a message that passes through a typical TCP/IP network:

We start with a single data stream, say a file you are trying to send to some other computer: .....

TCP breaks it up into manageable chunks. (In order to do this, TCP has to know how large a datagram your network can handle. Actually, the TCP's at each end say how big a datagram they can handle, and then they pick the smallest size.)

....

TCP puts a header at the front of each datagram. This header actually contains at least 20 octets, but the most important ones are a source and destination "port number" and a "sequence number". The port numbers are used to keep track of different conversations. Suppose 3 different people are transferring files. Your TCP might allocate port numbers 1000, 1001, and 1002 to these transfers. When you are sending a datagram, this becomes the "source" port number, since you are the source of the datagram. Of course the TCP at the other end has assigned a port number of its own for the conversation. Your TCP has to know the port number used by the other end as well. (It finds out when the connection starts, as we will explain below.) It puts this in the "destination" port field. Of course if the other end sends a datagram back to you, the source and destination port numbers will be reversed, since then it will be the source and you will be the destination. Each datagram has a sequence number. This is used so that the other end can make sure that it gets the datagrams in the right order, and that it hasn't missed any. (See the TCP specification for details.) TCP doesn't number the datagrams, but the octets. So if there are 500 octets of data in each datagram, the first datagram might be numbered 0, the second 500, the next 1000, the next 1500, etc. Finally, I will mention the Checksum. This is a number that is computed by adding up all the octets in the datagram (more or less - see the TCP spec). The result is put in the header. TCP at the other end computes the checksum again. If they disagree, then something bad happened to the datagram in transmission, and it is thrown away. If we abbreviate the TCP header as "T", the whole file now looks like this:

T.... T.... T.... T.... T.... T.... T....

In order to make sure the datagram has arrived at its destination, the recipient has to send back an "acknowledgement". This is a datagram

whose "Acknowledgement number" field is filled in. For example, sending a packet with an acknowledgement of 1500 indicates that you have received all the data up to octet number 1500. If the sender doesn't get an acknowledgement within a reasonable amount of time, it sends the data again. The window is used to control how much data can be in transit at any one time. It is not practical to wait for each datagram to be acknowledged before sending the next one. That would slow things down too much. On the other hand, you can't just keep sending, or a fast computer might overrun the capacity of a slow one to absorb data. Thus each end indicates how much new data it is currently prepared to absorb by putting the number of octets in its "Window" field. As the computer receives data, the amount of space left in its window decreases. When it goes to zero, the sender has to stop. As the receiver processes the data, it increases its window, indicating that it is ready to accept more data. Often the same datagram can be used to acknowledge receipt of a set of data and to give permission for additional new data (by an updated window). The "Urgent" field allows one end to tell the other to skip ahead in its processing to a particular octet. This is often useful for handling asynchronous events, for example when you type a control character or other command that interrupts output. The other fields are beyond the scope of this document.

The IP level

TCP sends each of these datagrams to IP. Of course it has to tell IP the Internet address of the computer at the other end. Note that this is all IP is concerned about. It doesn't care about what is in the datagram, or even in the TCP header. IP's job is simply to find a route for the datagram and get it to the other end. In order to allow gateways or other intermediate systems to forward the datagram, it adds its own header. The main things in this header are the source and destination Internet address (32-bit addresses, like 128.6.4.194), the protocol number, and another checksum. The source Internet address is simply the address of your machine. (This is necessary so the other end knows where the datagram came from.) The destination Internet address is the address of

the other machine. (This is necessary so any gateways in the middle know where you want the datagram to go.) The protocol number tells IP at the other end to send the datagram to TCP. Although most IP traffic uses TCP, there are other protocols that can use IP, so you have to tell IP which protocol to send the datagram to. Finally, the checksum allows IP at the other end to verify that the header wasn't damaged in transit. Note that TCP and IP have separate checksums. IP needs to be able to verify that the header didn't get damaged in transit, or it could send a message to the wrong place. For reasons not worth discussing here, it is both more efficient and safer to have TCP compute a separate checksum for the TCP header and data. If we represent the IP header by an "I", your file now looks like this:

IT.... IT.... IT.... IT.... IT.... IT.... IT....

Again, the header contains some additional fields that have not been discussed. Most of them are beyond the scope of this document. Flags and fragment offset fields are used to keep track of the pieces when a datagram has to be split up. This can happen when datagrams are forwarded through a network for which they are too big. (This will be discussed a bit more below.) A time to live is a number is included that is decremented whenever the datagram passes through a system. When it goes to zero, the datagram is discarded. This is done in case a loop develops in the system somehow. Of course this should be impossible, but well-designed networks are built to cope with "impossible" conditions.

At this point, it's possible that no more headers are needed. If your computer happens to have a direct phone line connecting it to the destination computer, or to a gateway, it may simply send the datagrams out on the line (though likely a synchronous protocol such as HDLC would be used, and it would add at least a few octets at the beginning and end).

The Ethernet level

However most of our networks these days use Ethernet. So now we

have to describe Ethernet's headers. Unfortunately, Ethernet has its own addresses. The people who designed Ethernet wanted to make sure that no two machines would end up with the same Ethernet address. Furthermore, they didn't want the user to have to worry about assigning addresses. So each Ethernet controller comes with an address builtin from the factory. In order to make sure that they would never have to reuse addresses, the Ethernet designers allocated 48 bits for the Ethernet address. People who make Ethernet equipment have to register with a central authority, to make sure that the numbers they assign don't overlap any other manufacturer. Ethernet is a "broadcast medium". That is, it is in effect like an old party line telephone. When you send a packet out on the Ethernet, every machine on the network sees the packet. So something is needed to make sure that the right machine gets it. As you might guess, this involves the Ethernet header. Every Ethernet packet has a 14-octet header that includes the source and destination Ethernet address, and a type code. Each machine is supposed to pay attention only to packets with its own Ethernet address in the destination field. (It's perfectly possible to cheat, which is one reason that Ethernet communications are not terribly secure.) Note that there is no connection between the Ethernet address and the Internet address. Each machine has to have a table of what Ethernet address corresponds to what Internet address. (We will describe how this table is constructed a bit later.) In addition to the addresses, the header contains a type code. The type code is to allow for several different protocol families to be used on the same network. So you can use TCP/IP, DECnet, Xerox NS, etc. at the same time. Each of them will put a different value in the type field. Finally, there is a checksum. The Ethernet controller computes a checksum of the entire packet. When the other end receives the packet, it recomputes the checksum, and throws the packet away if the answer disagrees with the original. The checksum is put on the end of the packet, not in the header. If we represent the Ethernet header with "E", and the Ethernet checksum with "C", your file now looks like this:

EIT....C EIT....C EIT....C EIT....C EIT....C

When these packets are received by the other end, of course all the headers are removed. The Ethernet interface removes the Ethernet header and the checksum. It looks at the type code. Since the type code is the one assigned to IP, the Ethernet device driver passes the datagram up to IP. IP removes the IP header. It looks at the IP protocol field. Since the protocol type is TCP, it passes the datagram up to TCP. TCP now looks at the sequence number. It uses the sequence numbers and other information to combine all the datagrams into the original file.

The ends our initial summary of TCP/IP. There are still some crucial concepts we haven't gotten to, so we'll now go back and add details in several areas. (For detailed descriptions of the items discussed here see, RFC 793 for TCP, RFC 791 for IP, and RFC's 894 and 826 for sending IP over Ethernet.)

Well-known sockets and the applications layer

So far, we have described how a stream of data is broken up into datagrams, sent to another computer, and put back together. However something more is needed in order to accomplish anything useful. There has to be a way for you to open a connection to a specified computer, log into it, tell it what file you want, and control the transmission of the file. (If you have a different application in mind, e.g. computer mail, some analogous protocol is needed.) This is done by "application protocols". The application protocols run "on top" of TCP/IP. That is, when they want to send a message, they give the message to TCP. TCP makes sure it gets delivered to the other end. Because TCP and IP take care of all the networking details, the applications protocols can treat a network connection as if it were a simple byte stream, like a terminal or phone line.

Before going into more details about applications programs, we have to describe how you find an application. Suppose you want to send a file to a computer whose Internet address is 128.6.4.7. To start the process,

you need more than just the Internet address. You have to connect to the FTP server at the other end. In general, network programs are specialized for a specific set of tasks. Most systems have separate programs to handle file transfers, remote terminal logins, mail, etc. When you connect to 128.6.4.7, you have to specify that you want to talk to the FTP server. This is done by having "well-known sockets" for each server. Recall that TCP uses port numbers to keep track of individual conversations. User programs normally use more or less random port numbers. However specific port numbers are assigned to the programs that sit waiting for requests. For example, if you want to send a file, you will start a program called "ftp". It will open a connection using some random number, say 1234, for the port number on its end. However it will specify port number 21 for the other end. This is the official port number for the FTP server. Note that there are two different programs involved. You run ftp on your side. This is a program designed to accept commands from your terminal and pass them on to the other end. The program that you talk to on the other machine is the FTP server. It is designed to accept commands from the network connection, rather than an interactive terminal. There is no need for your program to use a well-known socket number for itself. Nobody is trying to find it. However the servers have to have well-known numbers, so that people can open connections to them and start sending them commands. The official port numbers for each program are given in "Assigned Numbers".

Note that a connection is actually described by a set of 4 numbers: the Internet address at each end, and the TCP port number at each end. Every datagram has all four of those numbers in it. (The Internet addresses are in the IP header, and the TCP port numbers are in the TCP header.) In order to keep things straight, no two connections can have the same set of numbers. However it is enough for any one number to be different. For example, it is perfectly possible for two different users on a machine to be sending files to the same other machine. This could result in connections with the following parameters:

	Internet addresses	TCP ports
connection 1	128.6.4.194, 128.6.4.7	1234, 21
connection 2	128.6.4.194, 128.6.4.7	1235, 21

Since the same machines are involved, the Internet addresses are the same. Since they are both doing file transfers, one end of the connection involves the well-known port number for FTP. The only thing that differs is the port number for the program that the users are running. That’s enough of a difference. Generally, at least one end of the connection asks the network software to assign it a port number that is guaranteed to be unique. Normally, it’s the user’s end, since the server has to use a well-known number.

Now that we know how to open connections, let’s get back to the applications programs. As mentioned earlier, once TCP has opened a connection, we have something that might as well be a simple wire. All the hard parts are handled by TCP and IP. However we still need some agreement as to what we send over this connection. In effect this is simply an agreement on what set of commands the application will understand, and the format in which they are to be sent. Generally, what is sent is a combination of commands and data. They use context to differentiate. For example, the mail protocol works like this: Your mail program opens a connection to the mail server at the other end. Your program gives it your machine’s name, the sender of the message, and the recipients you want it sent to. It then sends a command saying that it is starting the message. At that point, the other end stops treating what it sees as commands, and starts accepting the message. Your end then starts sending the text of the message. At the end of the message, a special mark is sent (a dot in the first column). After that, both ends understand that your program is again sending commands. This is the simplest way to do things, and the one that most applications use.

File transfer is somewhat more complex. The file transfer protocol involves two different connections. It starts out just like mail. The user’s program sends commands like "log me in as this user", "here is my password", "send me the file with this name". However once the command to send data is sent, a second connection is opened for the

data itself. It would certainly be possible to send the data on the same connection, as mail does. However file transfers often take a long time. The designers of the file transfer protocol wanted to allow the user to continue issuing commands while the transfer is going on. For example, the user might make an inquiry, or he might abort the transfer. Thus the designers felt it was best to use a separate connection for the data and leave the original command connection for commands. (It is also possible to open command connections to two different computers, and tell them to send a file from one to the other. In that case, the data couldn't go over the command connection.)

Remote terminal connections use another mechanism still. For remote logins, there is just one connection. It normally sends data. When it is necessary to send a command (e.g. to set the terminal type or to change some mode), a special character is used to indicate that the next character is a command. If the user happens to type that special character as data, two of them are sent.

We are not going to describe the application protocols in detail in this document. It's better to read the RFC's yourself. However there are a couple of common conventions used by applications that will be described here. First, the common network representation: TCP/IP is intended to be usable on any computer. Unfortunately, not all computers agree on how data is represented. There are differences in character codes (ASCII vs. EBCDIC), in end of line conventions (carriage return, line feed, or a representation using counts), and in whether terminals expect characters to be sent individually or a line at a time. In order to allow computers of different kinds to communicate, each applications protocol defines a standard representation. Note that TCP and IP do not care about the representation. TCP simply sends octets. However the programs at both ends have to agree on how the octets are to be interpreted. The RFC for each application specifies the standard representation for that application. Normally it is "net ASCII". This uses ASCII characters, with end of line denoted by a carriage return followed by a line feed. For remote login, there is also a definition of a "standard terminal", which turns out to be a half-duplex

terminal with echoing happening on the local machine. Most applications also make provisions for the two computers to agree on other representations that they may find more convenient. For example, PDP-10's have 36-bit words. There is a way that two PDP-10's can agree to send a 36-bit binary file. Similarly, two systems that prefer full-duplex terminal conversations can agree on that. However each application has a standard representation, which every machine must support.

An example application: SMTP

In order to give a bit better idea what is involved in the application protocols, I'm going to show an example of SMTP, which is the mail protocol. (SMTP is "simple mail transfer protocol.) We assume that a computer called TOPAZ.RUTGERS.EDU wants to send the following message.

```
Date: Sat, 27 Jun 87 13:26:31 EDT
From: hedrick@topaz.rutgers.edu
To: levy@red.rutgers.edu
Subject: meeting

Let's get together Monday at 1pm.
```

First, note that the format of the message itself is described by an Internet standard (RFC 822). The standard specifies the fact that the message must be transmitted as net ASCII (i.e. it must be ASCII, with carriage return/linefeed to delimit lines). It also describes the general structure, as a group of header lines, then a blank line, and then the body of the message. Finally, it describes the syntax of the header lines in detail. Generally they consist of a keyword and then a value.

Note that the addressee is indicated as LEVY@RED.RUTGERS.EDU. Initially, addresses were simply "person at machine". However recent standards have made things more flexible. There are now provisions for systems to handle other systems' mail. This can allow automatic forwarding on behalf of computers not connected to the Internet. It can

be used to direct mail for a number of systems to one central mail server. Indeed there is no requirement that an actual computer by the name of RED.RUTGERS.EDU even exist. The name servers could be set up so that you mail to department names, and each department's mail is routed automatically to an appropriate computer. It is also possible that the part before the @ is something other than a user name. It is possible for programs to be set up to process mail. There are also provisions to handle mailing lists, and generic names such as "postmaster" or "operator".

The way the message is to be sent to another system is described by RFC's 821 and 974. The program that is going to be doing the sending asks the name server several queries to determine where to route the message. The first query is to find out which machines handle mail for the name RED.RUTGERS.EDU. In this case, the server replies that RED.RUTGERS.EDU handles its own mail. The program then asks for the address of RED.RUTGERS.EDU, which is 128.6.4.2. Then the mail program opens a TCP connection to port 25 on 128.6.4.2. Port 25 is the well-known socket used for receiving mail. Once this connection is established, the mail program starts sending commands. Here is a typical conversation. Each line is labelled as to whether it is from TOPAZ or RED. Note that TOPAZ initiated the connection:

```
RED 220 RED.RUTGERS.EDU SMTP Service at 29 Jun 87 05:17:18
TOPAZ HELO topaz.rutgers.edu
RED 250 RED.RUTGERS.EDU - Hello, TOPAZ.RUTGERS.EDU
TOPAZ MAIL From:<hedrick@topaz.rutgers.edu>
RED 250 MAIL accepted
TOPAZ RCPT To:<levy@red.rutgers.edu>
RED 250 Recipient accepted
TOPAZ DATA
RED 354 Start mail input; end with <CRLF>.<CRLF>
TOPAZ Date: Sat, 27 Jun 87 13:26:31 EDT
TOPAZ From: hedrick@topaz.rutgers.edu
TOPAZ To: levy@red.rutgers.edu
TOPAZ Subject: meeting
TOPAZ
TOPAZ Let's get together Monday at 1pm.
TOPAZ .
RED 250 OK
TOPAZ QUIT
```



```
RED 221 RED.RUTGERS.EDU Service closing transmission
channel
```

First, note that commands all use normal text. This is typical of the Internet standards. Many of the protocols use standard ASCII commands. This makes it easy to watch what is going on and to diagnose problems. For example, the mail program keeps a log of each conversation. If something goes wrong, the log file can simply be mailed to the postmaster. Since it is normal text, he can see what was going on. It also allows a human to interact directly with the mail server, for testing. (Some newer protocols are complex enough that this is not practical. The commands would have to have a syntax that would require a significant parser. Thus there is a tendency for newer protocols to use binary formats. Generally they are structured like C or Pascal record structures.) Second, note that the responses all begin with numbers. This is also typical of Internet protocols. The allowable responses are defined in the protocol. The numbers allow the user program to respond unambiguously. The rest of the response is text, which is normally for use by any human who may be watching or looking at a log. It has no effect on the operation of the programs. (However there is one point at which the protocol uses part of the text of the response.) The commands themselves simply allow the mail program on one end to tell the mail server the information it needs to know in order to deliver the message. In this case, the mail server could get the information by looking at the message itself. But for more complex cases, that would not be safe. Every session must begin with a HELO, which gives the name of the system that initiated the connection. Then the sender and recipients are specified. (There can be more than one RCPT command, if there are several recipients.) Finally the data itself is sent. Note that the text of the message is terminated by a line containing just a period. (If such a line appears in the message, the period is doubled.) After the message is accepted, the sender can send another message, or terminate the session as in the example above.

Generally, there is a pattern to the response numbers. The protocol defines the specific set of responses that can be sent as answers to any given command. However programs that don't want to analyze them in

detail can just look at the first digit. In general, responses that begin with a 2 indicate success. Those that begin with 3 indicate that some further action is needed, as shown above. 4 and 5 indicate errors. 4 is a "temporary" error, such as a disk filling. The message should be saved, and tried again later. 5 is a permanent error, such as a non-existent recipient. The message should be returned to the sender with an error message.

(For more details about the protocols mentioned in this section, see RFC's 821/822 for mail, RFC 959 for file transfer, and RFC's 854/855 for remote logins. For the well-known port numbers, see the current edition of Assigned Numbers, and possibly RFC 814.)

Protocols other than TCP: UDP and ICMP

So far, we have described only connections that use TCP. Recall that TCP is responsible for breaking up messages into datagrams, and reassembling them properly. However in many applications, we have messages that will always fit in a single datagram. An example is name lookup. When a user attempts to make a connection to another system, he will generally specify the system by name, rather than Internet address. His system has to translate that name to an address before it can do anything. Generally, only a few systems have the database used to translate names to addresses. So the user's system will want to send a query to one of the systems that has the database. This query is going to be very short. It will certainly fit in one datagram. So will the answer. Thus it seems silly to use TCP. Of course TCP does more than just break things up into datagrams. It also makes sure that the data arrives, resending datagrams where necessary. But for a question that fits in a single datagram, we don't need all the complexity of TCP to do this. If we don't get an answer after a few seconds, we can just ask again. For applications like this, there are alternatives to TCP.

The most common alternative is UDP ("user datagram protocol"). UDP is designed for applications where you don't need to put sequences of

datagrams together. It fits into the system much like TCP. There is a UDP header. The network software puts the UDP header on the front of your data, just as it would put a TCP header on the front of your data. Then UDP sends the data to IP, which adds the IP header, putting UDP's protocol number in the protocol field instead of TCP's protocol number. However UDP doesn't do as much as TCP does. It doesn't split data into multiple datagrams. It doesn't keep track of what it has sent so it can resend if necessary. About all that UDP provides is port numbers, so that several programs can use UDP at once. UDP port numbers are used just like TCP port numbers. There are well-known port numbers for servers that use UDP. Note that the UDP header is shorter than a TCP header. It still has source and destination port numbers, and a checksum, but that's about it. No sequence number, since it is not needed. UDP is used by the protocols that handle name lookups (see IEN 116, RFC 882, and RFC 883), and a number of similar protocols.

Another alternative protocol is ICMP ("Internet control message protocol"). ICMP is used for error messages, and other messages intended for the TCP/IP software itself, rather than any particular user program. For example, if you attempt to connect to a host, your system may get back an ICMP message saying "host unreachable". ICMP can also be used to find out some information about the network. See RFC 792 for details of ICMP. ICMP is similar to UDP, in that it handles messages that fit in one datagram. However it is even simpler than UDP. It doesn't even have port numbers in its header. Since all ICMP messages are interpreted by the network software itself, no port numbers are needed to say where a ICMP message is supposed to go.

Keeping track of names and information: the domain system

As we indicated earlier, the network software generally needs a 32-bit Internet address in order to open a connection or send a datagram. However users prefer to deal with computer names rather than

numbers. Thus there is a database that allows the software to look up a name and find the corresponding number. When the Internet was small, this was easy. Each system would have a file that listed all of the other systems, giving both their name and number. There are now too many computers for this approach to be practical. Thus these files have been replaced by a set of name servers that keep track of host names and the corresponding Internet addresses. (In fact these servers are somewhat more general than that. This is just one kind of information stored in the domain system.) Note that a set of interlocking servers are used, rather than a single central one. There are now so many different institutions connected to the Internet that it would be impractical for them to notify a central authority whenever they installed or moved a computer. Thus naming authority is delegated to individual institutions. The name servers form a tree, corresponding to institutional structure. The names themselves follow a similar structure. A typical example is the name BORAX.LCS.MIT.EDU. This is a computer at the Laboratory for Computer Science (LCS) at MIT. In order to find its Internet address, you might potentially have to consult 4 different servers. First, you would ask a central server (called the root) where the EDU server is. EDU is a server that keeps track of educational institutions. The root server would give you the names and Internet addresses of several servers for EDU. (There are several servers at each level, to allow for the possibly that one might be down.) You would then ask EDU where the server for MIT is. Again, it would give you names and Internet addresses of several servers for MIT. Generally, not all of those servers would be at MIT, to allow for the possibility of a general power failure at MIT. Then you would ask MIT where the server for LCS is, and finally you would ask one of the LCS servers about BORAX. The final result would be the Internet address for BORAX.LCS.MIT.EDU. Each of these levels is referred to as a "domain". The entire name, BORAX.LCS.MIT.EDU, is called a "domain name". (So are the names of the higher-level domains, such as LCS.MIT.EDU, MIT.EDU, and EDU.)

Fortunately, you don’t really have to go through all of this most of the time. First of all, the root name servers also happen to be the name

servers for the top-level domains such as EDU. Thus a single query to a root server will get you to MIT. Second, software generally remembers answers that it got before. So once we look up a name at LCS.MIT.EDU, our software remembers where to find servers for LCS.MIT.EDU, MIT.EDU, and EDU. It also remembers the translation of BORAX.LCS.MIT.EDU. Each of these pieces of information has a "time to live" associated with it. Typically this is a few days. After that, the information expires and has to be looked up again. This allows institutions to change things.

The domain system is not limited to finding out Internet addresses. Each domain name is a node in a database. The node can have records that define a number of different properties. Examples are Internet address, computer type, and a list of services provided by a computer. A program can ask for a specific piece of information, or all information about a given name. It is possible for a node in the database to be marked as an "alias" (or nickname) for another node. It is also possible to use the domain system to store information about users, mailing lists, or other objects.

There is an Internet standard defining the operation of these databases, as well as the protocols used to make queries of them. Every network utility has to be able to make such queries, since this is now the official way to evaluate host names. Generally utilities will talk to a server on their own system. This server will take care of contacting the other servers for them. This keeps down the amount of code that has to be in each application program.

The domain system is particularly important for handling computer mail. There are entry types to define what computer handles mail for a given name, to specify where an individual is to receive mail, and to define mailing lists.

(See RFC’s 882, 883, and 973 for specifications of the domain system. RFC 974 defines the use of the domain system in sending mail.)

Routing

The description above indicated that the IP implementation is responsible for getting datagrams to the destination indicated by the destination address, but little was said about how this would be done. The task of finding how to get a datagram to its destination is referred to as "routing". In fact many of the details depend upon the particular implementation. However some general things can be said.

First, it is necessary to understand the model on which IP is based. IP assumes that a system is attached to some local network. We assume that the system can send datagrams to any other system on its own network. (In the case of Ethernet, it simply finds the Ethernet address of the destination system, and puts the datagram out on the Ethernet.) The problem comes when a system is asked to send a datagram to a system on a different network. This problem is handled by gateways. A gateway is a system that connects a network with one or more other networks. Gateways are often normal computers that happen to have more than one network interface. For example, we have a Unix machine that has two different Ethernet interfaces. Thus it is connected to networks 128.6.4 and 128.6.3. This machine can act as a gateway between those two networks. The software on that machine must be set up so that it will forward datagrams from one network to the other. That is, if a machine on network 128.6.4 sends a datagram to the gateway, and the datagram is addressed to a machine on network 128.6.3, the gateway will forward the datagram to the destination. Major communications centers often have gateways that connect a number of different networks. (In many cases, special-purpose gateway systems provide better performance or reliability than general-purpose systems acting as gateways. A number of vendors sell such systems.)

Routing in IP is based entirely upon the network number of the destination address. Each computer has a table of network numbers. For each network number, a gateway is listed. This is the gateway to be used to get to that network. Note that the gateway doesn't have to connect directly to the network. It just has to be the best place to go to get there. For example at Rutgers, our interface to NSFnet is at the John

von Neuman Supercomputer Center (JvNC). Our connection to JvNC is via a high-speed serial line connected to a gateway whose address is 128.6.3.12. Systems on net 128.6.3 will list 128.6.3.12 as the gateway for many off-campus networks. However systems on net 128.6.4 will list 128.6.4.1 as the gateway to those same off-campus networks. 128.6.4.1 is the gateway between networks 128.6.4 and 128.6.3, so it is the first step in getting to JvNC.

When a computer wants to send a datagram, it first checks to see if the destination address is on the system's own local network. If so, the datagram can be sent directly. Otherwise, the system expects to find an entry for the network that the destination address is on. The datagram is sent to the gateway listed in that entry. This table can get quite big. For example, the Internet now includes several hundred individual networks. Thus various strategies have been developed to reduce the size of the routing table. One strategy is to depend upon "default routes". Often, there is only one gateway out of a network. This gateway might connect a local Ethernet to a campus-wide backbone network. In that case, we don't need to have a separate entry for every network in the world. We simply define that gateway as a "default". When no specific route is found for a datagram, the datagram is sent to the default gateway. A default gateway can even be used when there are several gateways on a network. There are provisions for gateways to send a message saying "I'm not the best gateway -- use this one instead." (The message is sent via ICMP. See RFC 792.) Most network software is designed to use these messages to add entries to their routing tables. Suppose network 128.6.4 has two gateways, 128.6.4.59 and 128.6.4.1. 128.6.4.59 leads to several other internal Rutgers networks. 128.6.4.1 leads indirectly to the NSFnet. Suppose we set 128.6.4.59 as a default gateway, and have no other routing table entries. Now what happens when we need to send a datagram to MIT? MIT is network 18. Since we have no entry for network 18, the datagram will be sent to the default, 128.6.4.59. As it happens, this gateway is the wrong one. So it will forward the datagram to 128.6.4.1. But it will also send back an error saying in effect: "to get to network 18, use 128.6.4.1". Our software will then add an entry to the routing

table. Any future datagrams to MIT will then go directly to 128.6.4.1. (The error message is sent using the ICMP protocol. The message type is called "ICMP redirect.")

Most IP experts recommend that individual computers should not try to keep track of the entire network. Instead, they should start with default gateways, and let the gateways tell them the routes, as just described. However this doesn't say how the gateways should find out about the routes. The gateways can't depend upon this strategy. They have to have fairly complete routing tables. For this, some sort of routing protocol is needed. A routing protocol is simply a technique for the gateways to find each other, and keep up to date about the best way to get to every network. RFC 1009 contains a review of gateway design and routing. However rip.doc is probably a better introduction to the subject. It contains some tutorial material, and a detailed description of the most commonly-used routing protocol.

Details about Internet addresses: subnets and broadcasting

As indicated earlier, Internet addresses are 32-bit numbers, normally written as 4 octets (in decimal), e.g. 128.6.4.7. There are actually 3 different types of address. The problem is that the address has to indicate both the network and the host within the network. It was felt that eventually there would be lots of networks. Many of them would be small, but probably 24 bits would be needed to represent all the IP networks. It was also felt that some very big networks might need 24 bits to represent all of their hosts. This would seem to lead to 48 bit addresses. But the designers really wanted to use 32 bit addresses. So they adopted a kludge. The assumption is that most of the networks will be small. So they set up three different ranges of address. Addresses beginning with 1 to 126 use only the first octet for the network number. The other three octets are available for the host number. Thus 24 bits are available for hosts. These numbers are used for large networks. But there can only be 126 of these very big networks. The Arpanet is one,

and there are a few large commercial networks. But few normal organizations get one of these "class A" addresses. For normal large organizations, "class B" addresses are used. Class B addresses use the first two octets for the network number. Thus network numbers are 128.1 through 191.254. (We avoid 0 and 255, for reasons that we see below. We also avoid addresses beginning with 127, because that is used by some systems for special purposes.) The last two octets are available for host addesses, giving 16 bits of host address. This allows for 64516 computers, which should be enough for most organizations. (It is possible to get more than one class B address, if you run out.) Finally, class C addresses use three octets, in the range 192.1.1 to 223.254.254. These allow only 254 hosts on each network, but there can be lots of these networks. Addresses above 223 are reserved for future use, as class D and E (which are currently not defined).

Many large organizations find it convenient to divide their network number into "subnets". For example, Rutgers has been assigned a class B address, 128.6. We find it convenient to use the third octet of the address to indicate which Ethernet a host is on. This division has no significance outside of Rutgers. A computer at another institution would treat all datagrams addressed to 128.6 the same way. They would not look at the third octet of the address. Thus computers outside Rutgers would not have different routes for 128.6.4 or 128.6.5. But inside Rutgers, we treat 128.6.4 and 128.6.5 as separate networks. In effect, gateways inside Rutgers have separate entries for each Rutgers subnet, whereas gateways outside Rutgers just have one entry for 128.6. Note that we could do exactly the same thing by using a separate class C address for each Ethernet. As far as Rutgers is concerned, it would be just as convenient for us to have a number of class C addresses. However using class C addresses would make things inconvenient for the rest of the world. Every institution that wanted to talk to us would have to have a separate entry for each one of our networks. If every institution did this, there would be far too many networks for any reasonable gateway to keep track of. By subdividing a class B network, we hide our internal structure from everyone else, and save them trouble. This subnet strategy requires special provisions in

the network software. It is described in RFC 950.

0 and 255 have special meanings. 0 is reserved for machines that don't know their address. In certain circumstances it is possible for a machine not to know the number of the network it is on, or even its own host address. For example, 0.0.0.23 would be a machine that knew it was host number 23, but didn't know on what network.

255 is used for "broadcast". A broadcast is a message that you want every system on the network to see. Broadcasts are used in some situations where you don't know who to talk to. For example, suppose you need to look up a host name and get its Internet address. Sometimes you don't know the address of the nearest name server. In that case, you might send the request as a broadcast. There are also cases where a number of systems are interested in information. It is then less expensive to send a single broadcast than to send datagrams individually to each host that is interested in the information. In order to send a broadcast, you use an address that is made by using your network address, with all ones in the part of the address where the host number goes. For example, if you are on network 128.6.4, you would use 128.6.4.255 for broadcasts. How this is actually implemented depends upon the medium. It is not possible to send broadcasts on the Arpanet, or on point to point lines. However it is possible on an Ethernet. If you use an Ethernet address with all its bits on (all ones), every machine on the Ethernet is supposed to look at that datagram.

Although the official broadcast address for network 128.6.4 is now 128.6.4.255, there are some other addresses that may be treated as broadcasts by certain implementations. For convenience, the standard also allows 255.255.255.255 to be used. This refers to all hosts on the local network. It is often simpler to use 255.255.255.255 instead of finding out the network number for the local network and forming a broadcast address such as 128.6.4.255. In addition, certain older implementations may use 0 instead of 255 to form the broadcast address. Such implementations would use 128.6.4.0 instead of 128.6.4.255 as the broadcast address on network 128.6.4. Finally, certain older implementations may not understand about subnets. Thus

they consider the network number to be 128.6. In that case, they will assume a broadcast address of 128.6.255.255 or 128.6.0.0. Until support for broadcasts is implemented properly, it can be a somewhat dangerous feature to use.

Because 0 and 255 are used for unknown and broadcast addresses, normal hosts should never be given addresses containing 0 or 255. Addresses should never begin with 0, 127, or any number above 223. Addresses violating these rules are sometimes referred to as "Martians", because of rumors that the Central University of Mars is using network 225.

Datagram fragmentation and reassembly

TCP/IP is designed for use with many different kinds of network. Unfortunately, network designers do not agree about how big packets can be. Ethernet packets can be 1500 octets long. Arpanet packets have a maximum of around 1000 octets. Some very fast networks have much larger packet sizes. At first, you might think that IP should simply settle on the smallest possible size. Unfortunately, this would cause serious performance problems. When transferring large files, big packets are far more efficient than small ones. So we want to be able to use the largest packet size possible. But we also want to be able to handle networks with small limits. There are two provisions for this. First, TCP has the ability to "negotiate" about datagram size. When a TCP connection first opens, both ends can send the maximum datagram size they can handle. The smaller of these numbers is used for the rest of the connection. This allows two implementations that can handle big datagrams to use them, but also lets them talk to implementations that can't handle them. However this doesn't completely solve the problem. The most serious problem is that the two ends don't necessarily know about all of the steps in between. For example, when sending data between Rutgers and Berkeley, it is likely that both computers will be on Ethernets. Thus they will both be prepared to handle 1500-octet datagrams. However the connection will at some point end up going

over the Arpanet. It can't handle packets of that size. For this reason, there are provisions to split datagrams up into pieces. (This is referred to as "fragmentation".) The IP header contains fields indicating the a datagram has been split, and enough information to let the pieces be put back together. If a gateway connects an Ethernet to the Arpanet, it must be prepared to take 1500-octet Ethernet packets and split them into pieces that will fit on the Arpanet. Furthermore, every host implementation of TCP/IP must be prepared to accept pieces and put them back together. This is referred to as "reassembly".

TCP/IP implementations differ in the approach they take to deciding on datagram size. It is fairly common for implementations to use 576-byte datagrams whenever they can't verify that the entire path is able to handle larger packets. This rather conservative strategy is used because of the number of implementations with bugs in the code to reassemble fragments. Implementors often try to avoid ever having fragmentation occur. Different implementors take different approaches to deciding when it is safe to use large datagrams. Some use them only for the local network. Others will use them for any network on the same campus. 576 bytes is a "safe" size, which every implementation must support.

Ethernet encapsulation: ARP

There was a brief discussion earlier about what IP datagrams look like on an Ethernet. The discussion showed the Ethernet header and checksum. However it left one hole: It didn't say how to figure out what Ethernet address to use when you want to talk to a given Internet address. In fact, there is a separate protocol for this, called ARP ("address resolution protocol"). (Note by the way that ARP is not an IP protocol. That is, the ARP datagrams do not have IP headers.) Suppose you are on system 128.6.4.194 and you want to connect to system 128.6.4.7. Your system will first verify that 128.6.4.7 is on the same network, so it can talk directly via Ethernet. Then it will look up 128.6.4.7 in its ARP table, to see if it already knows the Ethernet address. If so, it will stick on an Ethernet header, and send the packet. But suppose this system is not in the ARP table. There is no way to

send the packet, because you need the Ethernet address. So it uses the ARP protocol to send an ARP request. Essentially an ARP request says "I need the Ethernet address for 128.6.4.7". Every system listens to ARP requests. When a system sees an ARP request for itself, it is required to respond. So 128.6.4.7 will see the request, and will respond with an ARP reply saying in effect "128.6.4.7 is 8:0:20:1:56:34". (Recall that Ethernet addresses are 48 bits. This is 6 octets. Ethernet addresses are conventionally shown in hex, using the punctuation shown.) Your system will save this information in its ARP table, so future packets will go directly. Most systems treat the ARP table as a cache, and clear entries in it if they have not been used in a certain period of time.

Note by the way that ARP requests must be sent as "broadcasts". There is no way that an ARP request can be sent directly to the right system. After all, the whole reason for sending an ARP request is that you don't know the Ethernet address. So an Ethernet address of all ones is used, i.e. ff:ff:ff:ff:ff:ff. By convention, every machine on the Ethernet is required to pay attention to packets with this as an address. So every system sees every ARP requests. They all look to see whether the request is for their own address. If so, they respond. If not, they could just ignore it. (Some hosts will use ARP requests to update their knowledge about other hosts on the network, even if the request isn't for them.) Note that packets whose IP address indicates broadcast (e.g. 255.255.255.255 or 128.6.4.255) are also sent with an Ethernet address that is all ones.

Getting more information

This directory contains documents describing the major protocols. There are literally hundreds of documents, so we have chosen the ones that seem most important. Internet standards are called RFC's. RFC stands for Request for Comment. A proposed standard is initially issued as a proposal, and given an RFC number. When it is finally accepted, it is added to Official Internet Protocols, but it is still referred to by the RFC number. We have also included two IEN's. (IEN's used

to be a separate classification for more informal documents. This classification no longer exists -- RFC's are now used for all official Internet documents, and a mailing list is used for more informal reports.) The convention is that whenever an RFC is revised, the revised version gets a new number. This is fine for most purposes, but it causes problems with two documents: Assigned Numbers and Official Internet Protocols. These documents are being revised all the time, so the RFC number keeps changing. You will have to look in rfc-index.txt to find the number of the latest edition. Anyone who is seriously interested in TCP/IP should read the RFC describing IP (791). RFC 1009 is also useful. It is a specification for gateways to be used by NSFnet. As such, it contains an overview of a lot of the TCP/IP technology. You should probably also read the description of at least one of the application protocols, just to get a feel for the way things work. Mail is probably a good one (821/822). TCP (793) is of course a very basic specification. However the spec is fairly complex, so you should only read this when you have the time and patience to think about it carefully. Fortunately, the author of the major RFC's (Jon Postel) is a very good writer. The TCP RFC is far easier to read than you would expect, given the complexity of what it is describing. You can look at the other RFC's as you become curious about their subject matter. Here is a list of the documents you are more likely to want:

rfc-index list of all RFC's

rfc1012 somewhat fuller list of all RFC's

rfc1011 Official Protocols. It's useful to scan this to see what tasks protocols have been built for. This defines which RFC's are actual standards, as opposed to requests for comments.

rfc1010 Assigned Numbers. If you are working with TCP/IP, you will probably want a hardcopy of this as a reference. It's not very exciting to read. It lists all the offically defined well-known ports and lots of other things.

rfc1009 NSFnet gateway specifications. A good overview of IP routing and gateway technology.

rfc1001/2 netBIOS: networking for PC's

rfc973 update on domains

rfc959 FTP (file transfer)

rfc950 subnets

rfc937 POP2: protocol for reading mail on PC's

rfc894 how IP is to be put on Ethernet, see also rfc825

rfc882/3 domains (the database used to go from host names to Internet address and back -- also used to handle UUCP these days). See also rfc973

rfc854/5 telnet - protocol for remote logins

rfc826 ARP - protocol for finding out Ethernet addresses

rfc821/2 mail

rfc814 names and ports - general concepts behind well-known ports

rfc793 TCP

rfc792 ICMP

rfc791 IP

rfc768 UDP

rip.doc details of the most commonly-used routing protocol

ien-116 old name server (still needed by several kinds of system)

ien-48 the Catenet model, general description of the philosophy behind TCP/IP The following documents are somewhat more specialized.

rfc813 window and acknowledgement strategies in TCP

rfc815 datagram reassembly techniques

rfc816 fault isolation and resolution techniques

rfc817 modularity and efficiency in implementation

rfc879 the maximum segment size option in TCP

rfc896 congestion control

rfc827,888,904,975,985 EGP and related issues

Unix Utilities_____APPENDIX A

Three utility programs are provided with the PortMaster software distribution which support Dialnet operations. They are:

pmcommand

The pmcommand utility allows the administrator to download an ASCII command file to the PortMaster. The syntax of this command is:

pmcommand [-nostop] PM_Name PM_Password Command_File

Where:

-nostop - tells pmcommand not to stop executing if there is an error in the command script. Without this option, any errors detected in commands will cause the script to immediately stop executing.

PM_Name - is the name or IP address of the PortMaster.

PM_Password - is the administration password for the PortMaster.

Command_File - is the name of the file containing portmaster commands. All commands described in the "Command Prompt Interface" can be placed in this file.

pmdial

This utility allows a UNIX user to request that the PortMaster initiate dialing to a Dialnet Manual Dial Location (as found in the PortMaster Location Table). This utility is useful for scheduling network connections to remote sites, to allow users to start network connections, or for custom dial-back applications.

Syntax

pmdial [-v] *portmaster_name password location_name*

Where:

-v - This option causes the PortMaster to display the actual execution

of the dial command script to the user.

portmaster_name - is the name of the PortMaster as shown in the local hosts file.

password - is the PortMaster administration password for the named PortMaster.

location_name - is the name of a dial-out location in the PortMaster Location Table.

Operation

After checking syntax, pmdial establishes a network connection to the PortMaster and issues the command to dial the specified location. Pmdial then monitors the PortMaster to verify that the attempt to dial was successful. Pmdial returns after the dial attempt succeeds, or after the PortMaster times out and fails to complete the connection. Pmdial displays its results on the standard output.

Exit Codes

Pmdial exits with a 0 return code if the dial attempt succeeded. Otherwise, pmdial returns -1.

Limitations

Pmdial uses the TCP administration port of the PortMaster to request and monitor dial operations. Therefore, PMconsole cannot be running at the same time as PMdial.

pmreset

This utility allows a UNIX user to request that the PortMaster reset an active serial port. This would terminate an active Dialnet connection on that port.

Syntax

pmreset *portmaster_name password portname*

Where:

portmaster_name - is the name of the PortMaster as shown in the local hosts file.

password - is the PortMaster administration password for the named PortMaster.

portname - is the name of the serial or parallel port on the PortMaster (ie S0-S9 or P0).

Operation

After checking syntax, pmreset establishes a network connection to the PortMaster and issues the command to reset the specified port. Pmreset displays its results on the standard output.

Exit Codes

Pmreset exits with a 0 return code if the reset attempt succeeded. Otherwise, pmreset returns -1.

Limitations

Pmreset uses the TCP administration port of the PortMaster to request the reset operation. Therefore, PMconsole cannot be running at the same time as PMreset.

Command Reference _____ APPENDIX B

Command List

add	- Add an entry to one of the PortMaster tables
delete	- Remove and entry from one of the PortMaster tables
dial	- Initiate dialing to one of the Dialnet Locations
done	- End administration session with the PortMaster
exit	- End administration session with the PortMaster
help	- provide a brief listing of commands
ifconfig	- View or configure values for the active interfaces
quit	- End administration session with the PortMaster
reboot	- Terminate all sessions, and restart the operating system
reset	- Terminate a session on a port and restart the port
save	- Save configuration parameters or tables
set	- Set configuration values and parameters in tables
show	- Show active and configured values
version	- Display the current version of the operating system

Command Options

add

host	- Add a host to the local hosts table
route	- Add a route to the static routing table
location	- Add a new Dialnet dialout location
snmphost	- Add a host to the SNMP access list
netuser	- Add a SLIP or PPP user to the password table
user	- Add a login user to the password table

delete

host	- Delete a host from the local hosts table
route	- Delete a route from the static routing table
location	- Delete a Dialnet dialout location
snmphost	- Delete a host from the SNMP access list
netuser	- Delete a SLIP or PPP user from the password table
user	- Delete a login user from the password table

reset

all	- Reset all PortMaster ports
s0-s29	- Reset a PortMaster serial port
p0	- Reset the PortMaster parallel port
console	- Stop sending console messages to any of the ports
nic	- Reset the PortMaster ethernet controller
dialer	- Checks all active interfaces against the location table

save

s0-s29	- Save configuration values for a serial port
p0	- Save configuration values for the parallel port
all	- Save all global and port configuration values
location	- Save the entire location table
snmp	- Save the SNMP configuration table
password	- Save the entire password table
host	- Save the entire hosts table
console	- Save the currently active system console
routes	- Save the static routing table

set - global values

password	- Set the PortMaster administration password
host	- Set the default login host for the Portmaster
gateway	- Set the default router for the PortMaster
address	- Set the PortMaster's Internet Address
netmask	- Set the netmask of the ethernet interface
debug	- Set the console message debugging level
domain	- Set the NIS domain for the Portmaster
ypserver	- Set the NIS server for the PortMaster
routing	- Set the routing options for the ethernet interface
dialnet	- Enable or disable Dialnet functions
ethernet	- Enable or disable the ethernet interface
broadcast	- Set the broadcast Internet Address
snmp	- Enable or disable SNMP, or set community names
loghost	- Set the network host to receive syslog messages
assigned	- Set the first assigned address for dialin network users
telnet	- Set the TCP port for telnet administration

set - port values

speed - Set the port speed
parity - Set the parity
databits - Set the number of databits
stopbits - Set the number of stopbits
device - Set the port to be used as an outbound device
login - Set the port to be used for login sessions
twoway - Set the port for both inbound and outbound connections
username - Set the Autolog username
modem - Enable or disable hardware carrier detect
xon/xoff - Enable or disable Xon/Xoff flow control
rts/cts - Enable or disable RTS/CTS flow control
message - Set the welcome message
console - Set the system console
disabled - Disable the parallel port
host - Set the login host for the port
termtype - Set the terminal type for the user's environment
prompt - Set the login prompt for the port
override - Enable or disable host overrides of datacomm parameters
extended - enable or disable extended port features
disconnect - Set the seconds to wait before disconnecting a port
service_login - Set the default login service to use for the port
service_device - Set the default device service to use for the port
network - Enable or disable SLIP or PPP use of the port
routing - Set type of routing for hardwired network connection
address - Set the IP address for hardwired network connection
group - Set the Dialnet group number for the port
compress - Enable or disable header compression for connection
protocol - Set the network protocol for hardwired connection
map - Set the PPP async map for hardwired connection
mtu - Set the Maximum transmisson unit for connection
security - Enable or disable pass-through login access

set - password table values

address - Internet address for the network user
netmask - Netmask for the network user
protocol - Protocol (ie. SLIP or PPP) for the network user

routing - Type of routing for the network user
mtu - Maximum transmission unit for the network user
map - PPP async map for the network user
compression - Enable or disable header compress for the network user
password - Set the user's password
dialback - Set the dialback location or telephone number
host - Set the login host for the user
service - Set the login service for the user

set - location table values

idletime - Set the idle timeout (in minutes) for the location
routing - Set the type of routing to use for this location
address - Set the Internet address of this location
script - Set the Chat script for dialing to this location
manual - Set this to a manual dial location
continuous - Set this location to auto redial if the connection drops
on_demand - Set this location to be dialed On-Demand
netmask - Set the netmask for this location
group - Set the dial group for this location
maxports - Set the maximum number of connections to this location
high_water - Set the high water mark for dialing additional lines
protocol - Set the protocol to use for this location (SLIP or PPP)
compression - Enable or disable header compression for this location
mtu - Set the maximum transmission unit for this location
map - Set the PPP async map for this location

show

s0-s29 - Show input/output, active and default configuration
p0 - Show input/output, active and default configuration
all - Show summary status for all ports
table - Show summary listing of a data table
memory - Show system memory utilization
netconns - Show all TCP and UDP network connections
routes - Show active and static routing table
netstat - Show network interface statistics
user - Show configuration for a user in the password table
location - Show the configuration values for a dialnet location