Chapter 5

SLIP and PPP

This chapter introduces the Silicon Graphics implementation of the Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP). SLIP and PPP are protocols for TCP/IP networking over a serial line. SLIP and PPP can be used for connecting remote systems to a local area network, or for connecting two networks together.

The Silicon Graphics implementation provides both RFC 1144 data compression and its own proprietary data compression, which compresses header framing, checksum, and TCP/IP information to three bytes. SLIP is part of the *eoe2* software subsystem, and is installed via the *eoe2.sw.slip* package. Use the *versions*(1M) command to check to see if you have SLIP installed on your system.

PPP offers substantially the same features as SLIP, but is more flexible and robust. In general, PPP is preferable to SLIP. However, you will have to use SLIP if the system you want to connect to supports SLIP and not PPP.

PPP is part of the *eoe2* software subsystem, and is installed via the *eoe2.sw.ppp* package. Use the *versions*(1M) command to check to see if you have PPP installed on your system.

The following sections are included in this chapter:

- "NFS Over SLIP or PPP" on page 95
- "File Transfer Over SLIP or PPP" on page 96
- "Troubleshooting SLIP and PPP Links" on page 96

Overview

Usually, a SLIP or PPP connection serves one of the following purposes:

- To connect a single remote system to a network. Here the systems have a client-server relationship, with the single system being the client, and the system directly connected to the network being the server. The server handles all of the routing.
- To connect two networks together. In this case, the two systems usually act as peers.

A given SLIP or PPP link can be set up so that one system always initiates the connection (by dialing the other system), or it can be set up so that either system can initiate the connection. In the case of a client-server connection, the client usually initiates the connection by dialing the server. However, the connection can be bi-directional as well. In the case of a connection between two networks, you can either choose one system to initiate connection, or allow either system to initiate the connection as needed.

Setting up an individual system as a SLIP or PPP client requires the following steps:

1. Install the software.
2. Select and install the appropriate hardware (modem and cable).
3. Select an IP address.
4. Configure the software for dial-out.
5. Configure routing behavior.

Setting up a system as a SLIP or PPP server requires the following steps:

1. Install the software.
2. Select and install the appropriate hardware (modem and cable).
3. Configure the software for dial-in (set up configuration information for each client).
4. Configure routing behavior.

Connecting two networks using SLIP or PPP requires the following steps:

1. Install the software on both systems.
2. Select and install the appropriate hardware (modem and cable) on both systems.
3. Configure the software for dial-out access on one or both systems.
4. Configure the software for dial-in access on one or both systems.
5. Configure routing behavior.

Installing the Software

Make sure you have the appropriate software installed: to use SLIP, you need to have the *eoe2.sw.slip* and *eoe2.sw.uucp* subsystems installed. To use PPP, you need to have the *eoe2.sw.ppp* and *eoe2.sw.uucp* subsystems installed. You should also install the corresponding reference pages (*eoe2.man* subsystems). To verify whether these packages are installed, use the *versions* command. For example, to check whether you have PPP installed, you could use the following command:

```
% versions eoe2.\*.{ppp,uucp}
```

If PPP and UUCP are installed, *versions* should produce output something like this:

```
I = Installed, R = Removed

   Name                 Date      Description

I  eoe2                 03/24/95  IRIX Execution Environment 2, 5.3 with XFS
I  eoe2.man             03/24/95  IRIX Execution Environment Man Pages
I  eoe2.man.ppp         03/24/95  Point-to-Point Protocol Man Pages
I  eoe2.man.uucp        03/24/95  UNIX-to-UNIX Copy Man Pages
I  eoe2.sw              03/24/95  IRIX Execution Environment Software
I  eoe2.sw.ppp          03/24/95  Point-to-Point Protocol Software
I  eoe2.sw.uucp         03/24/95  UUCP Utilities
```

For instructions on installing software, see IRIX Admin: Software Installation and Licensing. After installing the SLIP or PPP software, you must reboot your system to reconfigure the kernel.

Selecting a Modem

A modem capable of at least 9600 bits per second (bps, sometimes referred to as "baud") is required for use with SLIP or PPP. However, at this speed, your link will be quite slow. A good modem choice is any modem that supports the V.32bis or V.34 standards and a speed of 14,400 bps or greater. A half-duplex modem can be used but is less desirable for interactive tasks.

Silicon Graphics does not manufacture these modems and cannot be responsible for changes to the modems. Silicon Graphics cannot guarantee modem compatibility or the quality of the telephone line used with SLIP or PPP.

If your modem supports both v.32 and PEP™ modes (or another batch-oriented mode), you can take advantage of these capabilities while using SLIP or PPP. The v.32 mode is preferred for most uses, but switching to PEP before using a file-transfer utility such as *ftp* or *rcp* yields higher performance.

If you are using a high-speed modem, be sure it supports hardware flow control using RTS/CTS. With such a modem, use the *ttyf** device name.

Note: Silicon Graphics supports DSI™, Intel®, Telebit®, ZyXEL™, U.S. Robotics®, Hayes®, and most Hayes compatible modems. While many others work, configuring them may be much more complicated and their operation is not guaranteed.

For instructions on installing modems and configuring modems, see "Installing a Modem" in IRIX Admin: Peripheral Devices.

IP Addresses for SLIP and PPP Clients

Each SLIP or PPP client needs an IP address and hostname for its SLIP or PPP interface. If it is also connected to one or more local area networks, it will also require an address and hostname for each of its other network interfaces. If you have had an address and hostname assigned to you by your system administrator or network service provider, you can go ahead and use this information to configure your system.

Some network service providers use dynamic addressing, where each client is given a dynamically allocated IP address when it connects. IRIX PPP has support for dynamic addressing. IRIX SLIP does not support dynamic addressing. See "Using Dynamic

Address Allocation With PPP" on page 93 for more information about dynamic addressing.

If you are setting up a SLIP or PPP server on your LAN, or are connecting two networks over a serial link, you should see "SLIP and PPP Routing and Address Allocation" on page 90 for planning information.

Configuring a System for Dial-Out

Configuring SLIP or PPP for dial-out requires approximately the same steps:

1. Add a line to the */etc/uucp/Systems* file describing the connection.
2. Add a line for the modem in the */etc/uucp/Devices* file.
3. If your modem is not already listed in the */etc/uucp/Dialers* file, add an entry for it.
4. Verify that the modem line is correctly configured in the */etc/inittab* file.

PPP requires one additional step, setting up the */etc/ppp.conf* file.

Configuration Files for Dial-Out

This section describes the various configuration files for SLIP and PPP:

- */etc/uucp/Systems*
- */etc/uucp/Devices*
- */etc/uucp/Dialers*
- */etc/uucp/inittab*
- */etc/ppp.conf*

/etc/uucp/Systems

The */etc/uucp/Systems* file contains the information your system needs to dial up another system. The remote station's node name and telephone number, as well as the local modem's speed and a password, are all kept here and are used to log in to the remote station. The format for a line representing a SLIP or PPP connection in the systems file is *system* Any *type speed phone login-script*

system specifies the name of the remote system. *type* gives the name of an entry in the *Devices* file, specifying a line and modem type. *speed* specifies the speed of the connection between the system and the modem. *phone* specifies the phone number for the remote system. The *login-script* tells SLIP or PPP how to log into the remote system.

In this example, the local station "wenders" uses this line in its */etc/uucp/Systems* file to call station lynch. The connection is made at 38,400 bps. The password is "hopper." SLIP logs in to the remote station by responding with "slip-wenders" to the login prompt, and "hopper" to the password prompt.

```
lynch Any ACUSLIP 38400 5551212 "" \r\c ogin:--ogin: slip-wenders \
asswd: hopper SLIP
```

The information must be in one continuous line. The last string, "SLIP," forces the local station to wait for the remote station to announce that it is starting the SLIP protocol.

The third field in the systems file (in this case, "ACUSLIP") specifies the modem line to be used to call the remote station. There must be at least one entry matching this field in the */etc/uucp/Devices* file.

For more information on the */etc/uucp/Systems* file, see "The Systems File" on page 138.

/etc/uucp/Devices

The file */etc/uucp/Devices* is used to configure the desired device, modem speed, and dialer program for SLIP on your IRIS station. The correct format for a line appropriate for SLIP in */etc/uucp/Devices* is:

type *device* `null` *speed* `212 x` *dialer*

The first field, *type*, can be any string you like, but it should correspond with the type specified in the Systems file. If you have multiple modems that can be used interchangeably, they should all have the same *type* name—the system automatically selects one of these modems when making a call. The *type* name ACUSLIP is commonly used to designate a modem for use with SLIP or PPP.

device can be any flow-control device associated with a port not currently in use. *speed* should be the speed of the connection between your system and the modem. *dialer* can be any dial program listed in */etc/uucp/Dialers*.

If you want to configure more than one modem speed, use a different port, or use a modem that supports a different command set, create a new line in the devices file that reflects the change.

For example, the following line configures SLIP to use a Telebit™ T2500 modem at 38,400 bps on the serial port 2 hardware flow control device:

```
ACUSLIP ttyf2 null 38400 212 x t25slip
```

For more information on the */etc/uucp/Devices* file, see "The Devices File" on page 131.

/etc/uucp/Dialers

The */etc/uucp/Dialers* file contains entries for various types of modems. The *dialer* field in the *Devices* file should refer to one of these entries.

You shouldn't need to add an entry to the dialers file, unless you are installing a type of modem not supported by IRIX. If you do need to add or modify an entry in the dialers file, see "The Dialers File" on page 135.

/etc/inittab

The port you specified in */etc/uucp/Devices* must be configured for dial-out or dial-in/dial-out use in the */etc/inittab* file. If you set up your modem for dial-out or dial-in/dial-out use as directed in "Installing a Modem" in IRIX Admin: Peripheral Devices, the */etc/inittab* file should already be set up correctly.

For example, if you want to use *ttyf2* for dial-out SLIP, the line for *ttyf2* in */etc/inittab* should read

```
t2:23:off:/etc/getty ttyf2 co_38400         # port 2
```

This line turns off the *getty* program on port number 2.

If the link can be initiated by either station, you must turn on *uugetty.* For example, to configure a symmetric link using Telebit T2500 modems, change the line to

```
t2:23:respawn:/usr/lib/uucp/uugetty -Nt 60 -it25in,conn ttyf2 dx_38400
```

Changes made to */etc/inittab* are acted upon when *init* reexamines the */etc/inittab* file. To make *init* reexamine */etc/inittab* immediately, use this command:

```
/etc/telinit q
```

For more information, see inittab(4).

/etc/ppp.conf

The */etc/ppp.conf* file is used to specify options for PPP connections. Each entry in the file consists of a host name and a set of options. For example:

```
salad     out remotehost=dial-in.salad.com
          localhost=caesar.salad.com
          quiet add_route
```

This example specifies a connection between the local host caesar.salad.com and a remote machine called dial-in.salad.com. The **out** keyword specifies that this is an outgoing connection. The **quiet** keyword specifies a demand-dialed connection (see "Demand Dialing" on page 95). The **add-route** keyword tells PPP to set up a default route through dial-in.salad.com.

For a complete list of options for the *ppp.conf* file, see the ppp(1M) reference page.

Sample SLIP Configuration for Dial-Out

This section shows a sample SLIP configuration for dial-out. This example sets up a connection between tuna.salad.com and dial-in.salad.com, a server on the salad.com corporate network.

/etc/uucp/Systems

```
salad Any ACUSLIP 38400 5551212 "" \r\c ogin:--ogin: slip-tuna \
asswd: celery SLIP
```

/etc/uucp/Devices

```
ACUSLIP ttyf2 null 38400 212 x t25slip
```

/etc/inittab

```
t2:23:off:/etc/getty ttyf2 co_38400         # port 2
```

To start SLIP using the configuration in this example, enter the following command:

```
% /usr/etc/slip -o -p comp -r salad
```

Sample PPP Configuration for Dial-Out

The main configuration file for PPP is */etc/ppp.conf*. This file is described in detail in the ppp(1M) reference page. Here is an example of a *ppp.conf* file:

```
salad     out remotehost=dial-in.salad.com
          localhost=caesar.salad.com
          quiet add_route
```

This entry describes an outgoing connection from a standalone system to a remote host named "dial-in.salad.com," which acts as a gateway to the spice.com network. The entry specifies demand-dialed (**quiet**) mode. Stand-alone clients should usually include the **add_route** keyword to set up a default route through the PPP server. The entry *salad* should match an entry in the */etc/uucp/Systems* file, such as the following:

```
salad Any ACUSLIP 38400 555-1212 "" @\r\c ogin--ogin: ppp-caesar \
ssword: mypasswd PPP
```

IRIX PPP sends out the message "starting PPP" before starting the protocol; therefore, the chat script shown above waits to receive the string "PPP" to ensure that the login has succeeded. If you are connecting to a non IRIX system, you may need to remove the "PPP" string from this example.

There must be at least one ACUSLIP entry in the */etc/uucp/Devices* file:

```
ACUSLIP ttyf2 null 38400 212 x t25slip
```

This entry specifies that *ttyf2* is connected to a Telebit 2500 modem.

You should also have your */etc/inittab* file set up to expect the modem port speed you are using (in this case, 38400 bps) and to have *getty* or *uugetty* turned off. Specific details on editing the */etc/inittab* file and restarting *telinit* are found in IRIX Admin: Peripheral Devices. The following entry works with the above listed file entries for PPP:

```
t2:23:off:/etc/uucp/uugetty ttyd2 dx_38400  # ppp modem
```

You may use dial-out PPP with the above-listed entry, but for dial-in PPP, you need to configure *uugetty* to answer the line, as described in IRIX Admin: Peripheral Devices.

It is important to note that the above-listed entries are simply examples of one configuration that works for one example site. The same entries may not yield satisfactory results in every case, due to other differences in site configuration and modem manufacturer and model. For example, the PPP site you are dialing into may require different settings in the */etc/ppp.conf* file, and the entry in the *Devices* file assumes a specific brand and model of modem. Also, the example assumes you have configured the modem as described in “Installing a Modem” in IRIX Admin: Peripheral Devices.

When you have configured the files, you must issue the *ppp* command as *root*. The following command works for the above-listed example file entries:

```
ppp -r salad
```

For complete information on the *ppp* command and its options, see the ppp(1M) reference page.

Configuring a System for Dial-In

To configure any system for dial-in, you must have at least one port set up for dial-in use (or combination dial-in/dial-out use). Each system that can dial in requires an entry in the */etc/passwd* file. In addition, SLIP connections may require an entry in the */usr/etc/remoteslip* file, and PPP connections may require an entry in */etc/ppp.conf*.

Configuring SLIP for Dial-In

SLIP requires an entry in */etc/passwd* in order to log in. The user ID and group ID must both be zero (0). Instead of the shell specified at the end of a normal entry in */etc/passwd*, SLIP uses the file */usr/etc/remoteslip*. To allow tuna.salad.com to log in as “slip-tuna,” dial-in.salad.com should have this line in */etc/passwd*:

```
slip-tuna:3RsB768WRAN2.:0:0:slip for tuna:/:/usr/etc/remoteslip
```

An entry like this one is necessary for each station that calls in by using SLIP. For maximum system security, the home directory for SLIP accounts should only be writable by the superuser. Using open directories such as */tmp* opens your system up to a variety of threats.

Note: The encrypted password in the example does not represent a real password. You must use the *passwd* command to set the password for the SLIP login. See the passwd(1) reference page for information on using *passwd*.

/usr/etc/remoteslip

In the */etc/passwd* entry for slip-tuna, the login shell is specified as the file */usr/etc/remoteslip*. This file is used to invoke SLIP on a remote station. In */usr/etc/remoteslip*, the *slip* command can specify the remote station's name and any other options appropriate to that connection.

/usr/etc/remoteslip is a Bourne shell script. You can add to the case statement as you would to any Bourne shell script case statement. The sh(1) reference page contains detailed information about shell script programming. Each SLIP connection should have an entry in the following format:

```
slip-nodename )
    exec /usr/etc/slip options
    ;;
```

nodename is the name of the remote station. Options for *slip* are detailed in the slip(1M) reference page.

The */usr/etc/remoteslip* file might contain this entry on the station dial-in.salad.com:

```
# Edit the case statement as required.

case $USER in

        slip-tuna)
                exec /usr/etc/slip -p comp -i -r tuna
                ;;
        *)
                exec /usr/etc/slip -i -r $USER
                ;;
esac
```

The connection between dial-in.salad.com and tuna uses the Silicon Graphics proprietary header prediction and compression for faster data transfer because the **-p comp** option is specified. To use RFC 1144 compression, use **-p cslip** instead. The option tells SLIP that the session is input from another station. The *slip* option, **-r tuna**, specifies the remote station's name. Note that this example also supplies a default case. If all of your SLIP clients use the same parameters, you can just modify the default case, instead of adding entries for each client.

Configuring PPP for Dial-In

Like SLIP, PPP requires an entry in */etc/passwd* in order to log in. The user ID and group ID must both be zero (0). Instead of the shell specified at the end of a normal entry in */etc/passwd*, PPP uses the command */usr/etc/ppp*. To allow the system caesar.salad.com to log in as "ppp-caesar," dial-in.salad.com should have this line in */etc/passwd*:

```
ppp-caesar:3RsB768WRAN2.:0:0:PPP for caesar:/:/usr/etc/ppp
```

An entry like this one is necessary for each station that calls in by using PPP. For maximum system security, the home directory for PPP accounts should be writable only by the superuser. Using open directories such as */tmp* exposes your system to a variety of threats.

Note: The encrypted password in the example does not represent a real password. You must use *passwd* to set the password for the PPP login. See the passwd(1) reference page for information on using *passwd*.

If the default parameters are acceptable, you don't even need an entry in */etc/ppp.conf* for a client. You may want to add a minimal entry, such as this:

```
ppp-caesar     in remotehost=caesar.salad.com
```

Note: You should never use the **add-route** keyword by itself on a server system.

SLIP and PPP Routing and Address Allocation

There are three basic ways to deal with routing over a SLIP or PPP link, depending on the circumstances.

- If you have a few standalone clients connecting to a server on your main network, you'll probably want to assign clients addresses from the server's network, and use proxy-ARP routing.
- If you have a lot of standalone clients connecting to your server, you're better off setting aside a subnet for SLIP and PPP client addresses. In this case, the server appears to the main network as a gateway to the SLIP/PPP "net."
- If you're connecting two networks together using SLIP or PPP, each network should have its own network number, and the systems that form the endpoints of the link should both run *routed*.

In all cases, the server should run the routing daemon, *routed*, unless you have a very small network and use static routing. For more information, see the routed(1M) and route(1M) reference pages. You can turn *routed* on using the *chkconfig* command:

```
# chkconfig routed on
```

Standalone clients should not run *routed*.

Proxy-ARP Routing

If you have a small number of standalone hosts connecting to a SLIP or PPP server, you can use Proxy-ARP routing. In this system, each of the standalone hosts is assigned an Internet address from the server's network. Each client sets up a default route through the server, and the server advertises each of the client's addresses using the Address Resolution Protocol (ARP). To set up a default route through the server, you can use the *route* command after a SLIP connection is established:

```
route add net default server-address 1
```

You should add this command to the script you use to start SLIP. For a PPP link, you can accomplish the same thing by adding the "add-route" keyword to the *ppp.conf* file.

For each client, the server should issue an *arp* command:

```
arp -s client-hostname server--ethernet-address pub
```

Note that the server's Ethernet address is not the same as its IP address. To determine the server's Ethernet address, run the *arp* command from another system on the same Ethernet:

```
% arp dial-in.salad.com
dial-in.salad.com (192.26.79.7) at 8:0:69:9:4f:ef
```

The string of hexadecimal numbers separated by colons (8:0:69:9:4f:ef) is the server's Ethernet address.

The server can be set up to run the *arp* commands at boot time by placing the commands in a local network script. The following example shows a local network script that sets up ARP entries for a set of clients:

```
#!/bin/sh
#
# starting up local networking stuff
```

```
#

IS_ON=/etc/chkconfig
CONF=/etc/config
SERV_ADDR=8:0:69:9:4f:ef

if $IS_ON verbose ; then
        ECHO=echo
        VERBOSE=-v
else            # For a quiet startup and shutdown
        ECHO=:
        VERBOSE=
fi

case "$1" in
  'start')
      # setup proxy ARP for the dialin hosts
      # if this host has more than one interface,
      # you will need to hard-code the Ethernet MAC address
      # instead of letting it be determined at run time.
      # note that 4DDN (among others) may change the MAC address from
default!
      # and some AppleTalk packages change the output of `netstat
-ian`!
      arp -s client1 $SERV_ADDR pub
      arp -s client2 $SERV_ADDR pub
      arp -s client3 $SERV_ADDR pub
      ;;
  'stop')
      # be nice and delete the ARP entries
      arp -d client1
      arp -d client2
      arp -d client3
      ;;
  *)
        echo "usage: $0 {start|stop}"
        ;;
esac
exit 0
#
```

Assume the preceding file was named */etc/init.d/network.local*, and you want it to run just after networking started, then you would use the following commands to create the startup and shutdown links:

```
ln -s /etc/init.d/network.local /etc/rc2.d/S31netlocal
ln -s /etc/init.d/network.local /etc/rc0.d/K39netlocal
```

SLIP/PPP Subnet

If you have a lot of standalone clients, it would be cumbersome to issue *arp* commands for all of them. A better strategy is to allocate client addresses from a special subnet set aside for SLIP and PPP clients. The server acts as a gateway to this net. The server must run *routed*, and it should use the **-F** option to reduce unnecessary network traffic. For example, if the SLIP/PPP subnet was 192.26.80, the string "-F 192.26.80" should be added to the */etc/config/routed.options* file.

The clients should set up default routes through the server, just as with proxy-ARP routing. In addition, if clients are to communicate with one another, they must have their Ethernet interfaces turned off:

```
% chkconfig network off
```

If clients have their Ethernet interfaces enabled, they will try to reach other clients on the same "net" through Ethernet, rather than through the server.

Connecting Two Networks With SLIP or PPP

Connecting two networks using SLIP or PPP is probably the simplest case from a routing standpoint. Each network should have its own network number, and the addresses of the server and client should be allocated from their respective networks. Both client and server should run *routed*.

Routers on the main network may need to be configured to recognize and route to the new network.

Using Dynamic Address Allocation With PPP

If you are connecting to a service provider who uses PPP with dynamic address allocation, use the keywords **localhost=0,0** and **add_route** in the */etc/ppp.conf* file, to allow the remote system to assign an IP address, and set up a default route through the remote system.

Configuring a Bidirectional Link

A simple SLIP or PPP link, in which one station must always initiate the connection, can be changed to allow either station to initiate the connection.

The first step is to establish a link in one direction. Once you've gotten that link working correctly, set up the link in the reverse direction. You should be able to use the same *ppp.conf* entry for both dial-in and dial-out.

Starting SLIP or PPP at Boot Time

To have a SLIP or PPP connection between networks start automatically, create a local network script, */etc/init.d/network.local*, for that purpose. The script resides on the station where you want to initiate the link and should be linked to appropriate files in the */etc/rc2.d* and */etc/rc0.d* directories. For example, to automatically start the demand-dialed PPP link described in "Sample PPP Configuration for Dial-Out" on page 87, you would create a local network script on caesar.salad.com. The script should start PPP when called with the argument **start**, and terminate it when called with the argument **stop**.

```
#!/bin/sh
# ppp boot startup script
case $1 in
   start)
       /etc/killall ppp
       if /etc/chkconfig ppp && test -x /usr/etc/ppp -a -s /etc/ppp.conf
       then
           /usr/etc/ppp -r salad &
       fi
       ;;
   stop)
       /etc/killall -TERM ppp
       ;;
   *)
       echo "usage: $0 {start|stop}"
       ;;
esac
```

This script should be linked to the appropriate filenames in the */etc/rc2.d* and */etc/rc0.d* directories:

```
ln -s /etc/init.d/network.local /etc/rc2.d/S31netlocal
ln -s /etc/init.d/network.local /etc/rc0.d/K39netlocal
```

Demand Dialing

If you have a SLIP or PPP link that is used irregularly but frequently, it is likely to be economical for you to make that link a demand-dialing link. With demand dialing, the system makes the telephone connection as needed, when there is network traffic to be transmitted, and it drops the connection when there is no traffic. By default, much of the non-essential network traffic does not cause a link to be established. For example, the traffic generated by the time daemon (*timed*) would not cause a call to be placed, but a request for file transfer would cause the connection to be made.

To use demand-dialing, add the **-q** option to your *slip* command on system that initiates the connection. Demand-dialing mode is also known as "quiet" mode. For complete information on this and other options, see the slip(1M) reference page.

PPP can be configured to use demand dialing by adding the "quiet" keyword to the *ppp.conf* file.

When using demand-dialing, it makes sense to start SLIP or PPP automatically at boot time, as described in "Starting SLIP or PPP at Boot Time" on page 94.

Note that the link is initiated only when there is traffic on the initiating end of the link.

NFS Over SLIP or PPP

You can run NFS over a SLIP or PPP link. NFS will be very slow because of the amount of information transferred in NFS transactions. You may be able to improve performance with these measures:

- Use NFS with modems faster than 9600 bps.
- If your modem offers a choice between v.32 mode and "batch-oriented" or "half-duplex" mode (for example, PEP), use the batch-oriented mode.

  Some dialers require v.32 or PEP mode. When choosing a new modem mode, verify that the dialer you specify uses the mode you want. Dialer modes are noted in the file */etc/uucp/Dialers.*
- Use one of the SLIP header prediction and compression options.

  For more information regarding the SLIP compression options, **comp** and **cslip**, see the slip(1M) reference page.

- Adjust the NFS options **rsize**, **wsize, timeo**, and **retrans** when mounting NFS file systems.

  To improve performance, read and write smaller blocks and specify longer timeouts. See the fstab(4) reference page for more information on NFS filesystem options.

File Transfer Over SLIP or PPP

File transfer over a serial link may be slow if other demanding utilities share the link. For faster file transfer, try these solutions:

- If your modem offers a choice between v.32 mode and batch-oriented mode (for example, PEP), use the batch-oriented mode.
- Use *uucp* if you do not want to share the line with other utilities.

Troubleshooting SLIP and PPP Links

If your SLIP or PPP link seems to be connecting, but you can't reach systems on the remote network, you might have a routing problem. Try to reach the remote system with *ping*:

```
% ping -c 10 dial-in.salad.com
```

If the connection is working, you should see output something like this:

```
PING dial-in.salad.com (192.26.79.52): 56 data bytes
64 bytes from 192.26.79.52: icmp_seq=0 ttl=255 time=2 ms
64 bytes from 192.26.79.52: icmp_seq=1 ttl=255 time=1 ms
64 bytes from 192.26.79.52: icmp_seq=2 ttl=255 time=1 ms
64 bytes from 192.26.79.52: icmp_seq=3 ttl=255 time=1 ms
64 bytes from 192.26.79.52: icmp_seq=4 ttl=255 time=1 ms
64 bytes from 192.26.79.52: icmp_seq=5 ttl=255 time=1 ms
64 bytes from 192.26.79.52: icmp_seq=6 ttl=255 time=1 ms
64 bytes from 192.26.79.52: icmp_seq=7 ttl=255 time=1 ms
64 bytes from 192.26.79.52: icmp_seq=8 ttl=255 time=2 ms
64 bytes from 192.26.79.52: icmp_seq=9 ttl=255 time=1 ms


----dial-in.salad.com PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
```

If the connection is not working, you should see output like this:

```
PING dial-in.salad.com (192.26.79.52): 56 data bytes


----dial-in.salad.com PING Statistics----
10 packets transmitted, 0 packets received, 100% packet loss
```

If you can contact the remote host, but not other systems on the network, you probably have a routing problem. Check that your routing is set up as described in "SLIP and PPP Routing and Address Allocation" on page 90.

If you aren't getting any connection at all, test the line with another utility. If you are familiar with *uucp*, you may want to establish a *uucp* link between the stations as a means of testing the connection. Most users will find *cu* an easier way to debug the link.

Note: *cu* requires a *direct* entry in the */etc/uucp/Devices* file. Refer to "The Devices File" on page 131 for more details.

When debugging a SLIP connection, check each station separately. First check the port and modem on the local station by using a *cu* command like this:

`cu -d -s speed -l port`

For example, to test the port and modem installed on the station tuna.salad.com, you would use this *cu* command:

`cu -d -s 38400 -l ttyf2`

(It may be necessary to turn off the *(uu)getty* first by changing **respawn** to **off** on the line for the port in the file */etc/inittab*.)

The modem should respond. Many modems respond by printing `AT`. If the modem does not respond as expected, review the SLIP configuration procedure for the local station and review the modem configuration and documentation. If the modem does respond as expected, disconnect from *cu* by typing a tilde followed by a dot:

`~.`

Connect the stations you want to link as you would for the SLIP link. On the local station, use *cu* to call the remote station through the port and connection you have already verified with *cu*.

Check the connection to the remote station with a *cu* command like this:

```
cu -d -sspeed telno
```

For example, to test the connection between tuna and dial-in.salad.com, you would call dial-in from tuna with this *cu* command:

```
cu -d -s38400 5552002
```

You should see the local station tell the modem to call the remote station. Eventually, you should see the login prompt. Type the *send* strings from the */etc/uucp/Systems* file in response to the *expect* strings.