# Linux PPP FAQ

Al Longyear, longyear@netcom.com v1.12, 12 January 1996.

This document contains a list the most Frequently Asked Questions (FAQ) about PPP for Linux (and their answers). It is really *not* a HOWTO, but is in 'classical' Question / Answer form. We have a different document which represents the PPP-HOWTO. It is written by Robert Hart.

## 1 Preface

Please send any corrections to longyear@netcom.com.

This is but one of the Linux HOWTO/FAQ documents. You can get the HOWTO's from sunsite.unc.edu:/pub/Linux/docs/HOWTO (this is the 'official' place) or via WWW from *the Linux Documentation home page* <http://sunsite.unc.edu/mdw/linux.hmtl>. You cannot rely on the HOWTO's being posted to comp.os.linux.answers, as some news feeds have complained about their size.

Throughout this document, I have used the word 'remote' to mean 'the system at the other end of the modem link'. It is also called 'peer' in the PPP documentation. Another name for this is called the 'gateway' when the term is use for routing. Its IP address will show as the 'P-t-P' address if you use ifconfig.

Microsoft is a registered trademark of Microsoft Corporation. Morning Star is a registered trademark of Morning Star Technologies Incorporated. All other products mentioned are trademarks of their respective companies.

## 2 General information

### 2.1 What is PPP?

PPP, or Point-to-Point Protocol, is a recognized 'official' Internet protocol. It is a protocol used to exchange IP frames (and others) over a serial link. The current RFC for PPP is 1661. There are many related ones.

Contrary to what some people think, it does not mean "Peer to Peer Processing"; although you may do peer-peer communications using TCP/IP over a PPP link.

### 2.2 My university (company) does not support PPP. Can I use PPP?

In general, no. A 'classical' PPP implementation requires that you make changes to the routes and network devices supported by the operating system. This may mean that you will have to rebuild the kernel for the remote computer.

This is not a job for a general user. If you can convince your administration people that PPP is a 'good thing' then you stand a chance of getting it implemented. If you can't, then you probably can't use PPP.

However, if you are using a system which is supported by the people who are marketing the "TIA" (The Internet Adapter) package, then there is hope. I do not have much information on this package, however, from what I have found, they plan to support PPP in "the next version". (My information may be old. Contact them directly. Information on TIA is available at `ftp.marketplace.com` in the `/pub/tia` directory.)

If your system is not supported by TIA, and you choose not to use slirp, and you can't convince the admin group to support PPP then you should use the '`term`' package. Some service providers will object to you running '`term`'. They have many different reasons, however the most common is 'security concerns'.

There is a version of TIA for Linux.

In addition to TIA, Danny Gasparovski wrote a program called `slirp` which will perform functions similar to TIA. The program is currently available with the source code from the ftp site `blitzen.canberra.edu.au:/pub/slirp`. You should obtain the code if you wish additional information about this program. From the initial examination, it is seems to be an excellent contender to the commercial TIA program.

## 2.3 Where is PPP?

It is in two parts. The first part is in the kernel. In the kernels from 1.1.13, the driver is part of the network system drivers.

The second part is the 'daemon' process, `pppd`. This is a **required** process. The source to it is in the file `ppp-2.2.0e.tar.gz` located on `sunsite.unc.edu` in the `/pub/Linux/system/Network/serial` directory.

Version 2.2 and above are designed to be used only with the 1.2 and later kernels. Please don't use this version with the 1.1 series kernels as they are out of date for either the tty driver or the networking software.

## 2.4 I just obtained PPP. What do I do with it?

**R**ead **T**he **F**ine **M**aterial available.

Start by reading the `README` file and then the `README.linux` file. The documentation sources are listed below.

## 2.5 Where are additional sources of information for PPP? (Where's the documentation? Is there a HOWTO?, etc.)

There are several sources of information for the PPP protocol as implemented under Linux.

- The `README` file in the source package.
- The `README.linux` file in the source package.
- The `Net-2-HOWTO` document.
- The `PPP-HOWTO` document.

- The Network Administration Guide.
- The `pppd` man page.
- The FAQ document for the comp.protocols.ppp newsgroup.

The HOWTO and this FAQ are stored in the usual place for the Linux HOWTOs. That is currently on `sunsite.unc.edu` in the directory `/pub/Linux/docs/HOWTO`.

The Network Administration Guide is available in the `/pub/Linux/docs/LPD/network-guide` directory on sunsite. It is also published by O'Riellly and Associates. So, if you want a really professional document, then buy a copy from your local bookstore.

The '`man`' pages are included in the source package. You will probably have to move them to the normal man directory, `/usr/man/man8` before the `man` command may find them. Alternately, you may use `nroff` and `more` to view them directly.

The FAQ for comp.protocols.ppp describes the PPP protocol itself and the various implementations. You will find the FAQ for the usenet news group, `comp.protocols.PPP`, archived on `rtfm.mit.edu` in the `/usenet` directory. It is in eight parts at the present time.

## 2.6 Would someone please send me scripts for PPP so that I may see how they are written?

There are a few scripts which are included with the source package for pppd. It will cover the normal types of access where you are requested to enter a UNIX login and password.

Specific 'scripts' for specific systems are not included. If you have problems with a specific connection then you should contact the help desk for your site, the local news group at the site, or the general usenet groups for Linux. Unfortunately, time does not permit me to answer questions for help on supplying a script for your specific system.

## 2.7 Where should I post questions about PPP?

The primary usenet group for the PPP implementations is `comp.protocols.PPP` or `comp.os.linux.setup`. Use this group for general questions such as "How do I use pppd?" or "Why doesn't this work?".

Questions such as "Why wont pppd compile?" are generally linux related and belong on the comp.os.linux.networking group.

Please don't use comp.os.linux.help; even if your site should still carry this obsolete news group.

## 2.8 The PPP software doesn't work. HELP!!!

This is one of the most sickening questions. I realize that this is a plea for help. However, it is practically useless to post this message **with no other information**. I, and most others, will only ignore it.

Please see the question regarding errors which normally occur at the modem's disconnection. They are not the cause of a problem, only a symptom. Posting a message with only those errors is also meaningless.

What is needed is the output of the system log (syslog) when you run the `pppd` program with the option '`debug`'. In addition, if you are using chat then please use the '`-v`' option to run the sequence with verbose output.

Please include the output from the kernel's startup. This shows the various kernel hardware information such as your UART type, PPP version, etc.

Please include all information that you can relating to the problem. However your system configuration, disk drive configuration, terminal type, mouse location and button status, etc. are irrelevant. What is important is the system to which your are trying to contact, the PPP (or terminal server) that they are using, the modem types and speed that you are using, etc.

Take care and go through the output. Remove the references to the telephone number, your account name, and the password. They are not important to analyzing the problem and would pose a security risk to you if you published them to usenet. Also discard the lines which neither come from the kernel nor pppd.

Do **NOT** run the pppd program with the option '`kdebug 31`' and post that!

If the problem warrants examining the data stream, then you will be contacted by email and asked to mail the trace. Usenet already costs too much for too many people.

Information is written to various levels. The debug information is written to the debug level. The informational messages are written to the info level. The errors are written to the error level. Please include all levels the the '`local2`' group which come from the `pppd` process.

In addition, please do not delete the time stamp information. It is important.

## 2.9 How do I use PPP with a system which uses dynamic IP assignments? It assigns a different IP address to me with each call.

The assignment of the local IP address is a function of the options given to pppd and the IPCP protocol. You should use the 'magic' IP address of 0.0.0.0 if you must specify the local IP address. Most people simply leave the local IP address out of the option list.

The other option which is closely tied to this is called 'noipdefault'. The noipdefault option instructs the pppd process to not attempt to guess the local IP address from your hostname and the IP addresses in the /etc/hosts file. Most people use this option when the IP address is dynamically assigned. However, this option does not mean 'use dynamic IP addresses'. The use of dynamic IP addresses is automatic when the local IP address is not given.

## 2.10 How do I know what IP address was given to me when it is dynamically assigned?

Use the /etc/ppp/ip-up hook. The local IP address is the fourth parameter. This will be executed when pppd knows the IP address for the local system. The fifth parameter is the remote IP address if you should wish to know this value as well.

If you are curious about the value assigned then you may use the `ifconfig` program to display the current settings. It will show you the current values for both the local IP address and the IP address assigned to the remote under the P-t-P heading.

### 2.11 Can I use the same local IP address for each line of a PPP server?

Yes. The local address is not significant to the local system. You must have a unique `remote` IP address. The routing is performed based upon the remote IP address and not the local IP address.

## 3 Other implementations

### 3.1 Do you know of a implementation for PPP other than Linux? I would like one for HP-UX, or AIX, or ... (you fill in the blank) ?

Check the PPP FAQ document mentioned above.

HP-UX is supported by the commercial Morningstar package. AIX is in the current 2.2 pppd package.

If you don't find one listed then post to the `comp.protocols.PPP` group and not the Linux group.

(Please don't mail me asking for "Do you know of a PPP package for ..."? These requests will now be 'appropriately' filed. ;-))

The pppd package placed on sunsite does not contain the code which would use the some of the ports which use the streams interface. This is due to the reason that the streams interface contains a restrictive copyright which prevents the commercial packaging of the source which contains the module. We, the people who have been working on the pppd package, have tried to contact the author of the original module for streams in an attempt to have the copyright changed. He was un-responsive at first. Now he can not be located.

For this reason, and due to the fact that the sunsite site is for Linux, I decided to remove the AIX, Next, and any other port of pppd which involved the original streams code. The SunOS and Solaris ports `are` included since their streams implementation has been rewritten. You should continue to find the BSD variation as well as the Linux form in the package. If you wish the pppd code for a system which uses streams then you will have to consult the PPP-FAQ for the location of the pppd archive site near you. Alternately, you can use archie. Just don't use the mirrors for sunsite as they will not have the code.

### 3.2 Did you know that there is a program called 'dp'?

Yes, we know. The `dp` package was considered very early in the development stage quite a few months back. It is nice. It supports 'demand dial'. It also only works with systems which support streams. This is primarily the SunOS (Solaris) operating systems.

The question of demand dial is covered later in this document.

Linux, at the present time, does not supports streams.

There are several other packages for PPP available on the 'net'. The 'portable PPP' package is very much like the TIA code. There is another package called simply 'PPP'. There is code for PPP in the KA9Q package.

The `slirp` and `TIA` code will do PPP as well.

Of all of the packages available, the pppd package was the closest to the requirements and functions of Linux to warrant the port.

(If you want more information about these other packages, ask in the `comp.protocols.PPP` group!)

### 3.3 What RFCs describe the PPP protocol?

The current implementation of PPP is a mixture of several. The major portion of the PPP code is written against the RFCs 1331 and 1332. These RFCs were later obsoleted. 1331 was replaced by 1548 and that, in turn, was obsoleted by 1661 six months later.

Most implementations of PPP will be happy to talk to the Linux PPP code.

A complete list is in the PPP faq.

[to quote the FAQ document]:

> All of 1134, 1171, and 1172 (and 1055, for that matter `:-)` have been obsoleted. They're interesting only if you want to debug a connection with an ancient PPP implementation, and you're wondering why (e.g.) it asked you for `IPCP` option 2 with a length of only 4, and Compression-Type `0x0037`. (There's a lot of that still running around - be careful out there.)

Linux PPP will automatically detect this condition and compensate for it.

## 4 Compatibility

### 4.1 Can PPP talk to a SLIP interface?

No. SLIP works with SLIP. PPP works with PPP.

Some vendors may offer products which work both as SLIP and PPP. However, they must be configured to run in one mode or the other. There is no present method to determine, based upon the protocol passed at the time of a connection, which combination of SLIP protocols or PPP is being requested.

### 4.2 Which is better? PPP or SLIP?

**IT DEPENDS UPON MANY FACTORS**. The people who post this type of question have usually not read the `Net-2-HOWTO` document.

A good technical discussion is available at Morning Star's www server, `www.morningstar.com`.

### 4.3 Is CHAP or PAP better for authentication?

If you have the choice, use CHAP. Failing that, PAP is better than nothing.

### 4.4 What about CHAP which Microsoft uses with Windows NT?

CHAP is a cryptographic authentication protocol. It means that it takes some form of a key and will encrypt the response using a one-way encryption algorithm. The algorithm is negotiated at the time that the CHAP protocol is requested. The most common is called MD5. It has an encryption code of 05 in the CHAP request.

Microsoft uses a DES algorithm which, until recently, was incompatible with the pppd process. If you wish to connect to a Windows NT server, there are a set of patches which are included with the pppd source code to support the DES style used by Microsoft.

`Contrary to what some un-informed people believe at Stanford University believe, Microsoft did not just go against the recommendations of the IETF working group. The code values were properly requested and the implementation has been fully documented.`

## 5 Authentication files

### 5.1 What goes into the /etc/ppp/pap-secrets file? Do you have a sample?

The PAP protocol is most often implemented as your user name and password. You need to include the name of the remote system, your account name, and the password. If the user on abbot wishes to call costello, the entry would be similar to the following.

```
#account   remote     password     IP address list
abbott     *          firstbase
```

To use PAP authentication with the simplest case, you should also include the 'user' option to specify which of the pap-secrets file entries is to be used. The option is explained in the pppd man page. However, the simplest for this example is:

user abbott

### 5.2 What goes into the /etc/ppp/chap-secrets file? Do you have a sample?

The most common problem is that people don't recognize that CHAP deals with a pair of secrets. Both computers involved in the link must have both secrets to work.

For example, if `abbot` wants to talk to `costello`, then `abbot`'s file would have:

```
#remote    local      secret       IP address list
abbott     costello   firstbase    10.10.10.2
costello   abbott     who          10.10.10.1
```

And costello's file would have:

```
#remote    local      secret       IP address list
abbott     costello   firstbase    10.10.10.2
costello   abbott     who          10.10.10.1
```

# 6 Construction problems

## 6.1 I get compile errors when I try to compile the kernel

The 2.2 package contains instructions which describe the steps needed to install the package. Briefly, you need to run the configure command. It will generate the links to the Makefile. Next, you need to run 'make kernel'. This will install the needed pieces should the need to be updated.

Once the pieces have been installed, please rebuild the kernel at this time. Do this even if you have previously constructed the kernel to support PPP. The driver shipped with the 1.2 and early 1.3 kernels is not compatible with the 2.2 version of pppd.

Once you have rebuilt the kernel then you may resume to build the pppd process, chat, and pppstats.

There are problems with practically every version of the kernel and the pppd package. In the linux/Other.Patches directory of the source archive you will find a series of patches. Two of them need to be applied to the 1.2.13 kernel. The only one which you do not apply to the 1.2.13 kernel is the patch labeled "1.3".

If you have an early version of the 1.3 kernel then you need to apply the two patches for the 1.3 kernel and the slhc files.

After you have applied the patch, if you are using an old version of the include files, then you will have problems with the compilation of the structures in the ppp_defs.h header file. Edit this file and change the `"#if 0"` to be `"#if 1"`. This is due to the definition of the u_int32_t typedef.

The package will build with the 1.3.58 kernel and the 5.2.18 version of the libc and include files without the need for any patches.

# 7 Problems running pppd

## 7.1 pppd says that version 0.0.0 is out of date

You are attempting to run the 2.2 pppd process and you haven't rebuilt the drivers in the kernel.

## 7.2 pppd says that that the kernel is not configured for PPP. I know that I enabled the option and built the kernel.

Make sure that you did rebuild the kernel and that you are running it.

Make sure that you don't have an old copy of pppd on your disk and you are running that version. The previous version of pppd was stored on /usr/lib/ppp. Many people objected to this location. The 2.2 code has moved the `pppd`, `chat`, and `pppstats` to the /usr/sbin directory. If your scripts still reference **/usr/lib/ppp** then you will probably run the old code.

## 7.3 `pppd` wont run unless you are root

The pppd process needs to make changes to the networking system and this can only be done if you are the root user. If you wish to run `pppd` from other than the root user then the pppd program needs to be secured 'suid to root'.

```
chown root pppd
chmod 4755 pppd
```

If you wish to control the pppd access to a select group of people, then make the `pppd` process owned by the group and do not permit all others to run the program.

## 7.4 `unable to create pid file: no such file or directory`

You need to create the directory `/var/run`. On earlier Slackware distributions, this was a symbolic link to the `/etc` directory.

This is a warning. The PPP software will work normally in spite of this message. However, the `ppp-off` script depends upon this file. It is a good idea to create the directory or make the link to the appropriate location.

The posix header, `paths.h`, defines the location for the pid file under the name "`_VAR_RUN`". If you wish to use a different directory for PPP and others, change the value for this define and rebuild the software.

## 7.5 `/etc/ppp/options: no such file or directory`

You need to create the directory `/etc/ppp` and have a file called '`options`' in that directory. It needs to be readable by the `pppd` process (root).

The file may be empty. To make an empty file use the '`touch`' command.

See the `pppd` man page, `pppd.8`, for a description of this file.

## 7.6 `Could not determine local IP address`

This happens with many configurations of the Telebit Netblazer. The problem is not the terminal server, but the site which has not configured the terminal server with a set of IP addresses.

The Netblazer does not have your IP address. You do not have your IP address. The link will not work unless both IP addresses are known.

- The Netblazer does not have your IP address and you do not have your IP address.
- The Netblazer does know its IP address and you do not have its IP address.

The link will not work unless both IP addresses are known.

You must tell the Netblazer the IP addresses to be used. Use the local IP address and the remote IP address as a parameter to the pppd process.

Use the pppd option format of:

local_ip:remote_ip

(That is the local IP address, a colon, and the remote IP address.)

### 7.7 `Could not determine remote IP address`

See the previous answer.

### 7.8 I keep getting the message to the effect that the magic number is always NAKed. The system will not connect.

There is a one in over four billion chance that the two systems have chosen the same magic number. If you get a continual failure about the magic number, the chances that this is a fluke will geometrically reduce.

The two most common reasons for this failure are:

- The remote PPP software is not running when you think it is. Is the remote system configured to run PPP? Is the PPP process in the expected location? Is the privileges suitable so that you may run it? This would indicate that the shell is doing the local echo of the data. This is the more common reason.
- The modem has disconnected immediately upon making the connection and logging you on to the remote. Most modems are configured to echo the data sent to them and you are seeing the local echo from the modem.

In either case, the Linux system is sending data to the remote which is being fed immediately back into the serial receiver. This is not an acceptable condition. You have what is called a "`loop`".

### 7.9 `protocol reject for protocol fffb`

This usually occurs when you are trying to connect to a Xyplex terminal server. Version 5.1 of the Xyplex terminal server software, according to Xyplex, has numerous problems with PPP. It is strongly recommended that you update the Xyplex software to at least version 5.3.

If you must use Xyplex version 5.1, then use the pppd option "`vj-max-slots 3`" to limit the number of slots to three. The problem on the Xyplex server is that it will accept the request for the default 16 slots, but fail to operate beyond the third slot. It should have return a NAK frame with the limit, but it does not.

Alternately, you can disable the Van Jacobson header compression with the option "`-vj`".

## 7.10 The PPP software connects, sends quite a few frames, but still does not seem to connect. Why is that?

Linux does not support RPI modems. If your modem is RPI then you will have to find a different modem. This is not likely to change in the future given the statements made by Rockwell's management.

Examine the system log when you use the "`debug`" option. (You will need the system log data anyway if you are going to ask for help.) If the trace shows that it is sending the `LCP`-request frame over and over again and the id number is not incrementing then you are not exchanging frames with the remote PPP software.

Three common reasons for this are:

- You don't have the PPP software running on the other end. You are sending the PPP frames to some other program which is probably saying "What is this #$%^ ?" Please make sure that you have the PPP software started on the other end before you enter the PPP protocol sequence. Try to use a normal modem program and go through the logon sequence that you would normally do. Do you see the PPP frames being sent to you?

  The PPP frames are fairly distinctive. They will be about 40 characters in length and contain several { characters. They should not have a carriage return character after them and are sent out in a burst with a pause between the bursts.

- The line is not "eight bit clean". This means that you need to have eight data bits, no parity, and one stop bit. The PPP link absolutely requires eight data bits. The pppd software will automatically put the line into eight data bits, no parity, and one stop bit. The remote must match this configuration or framing and parity errors may occur.

  PPP will escape characters. It is not possible for it to escape bits as kermit does. PPP will *not* work with a seven bit communications link.

- The remote is configured to require authentication such as `PAP` or `CHAP`. You have not configured the local system to use this feature. Therefore, the remote is discarding all of your frames until it sees a valid authentication frame from you. Since you are not configured to generate the frames, the `IPCP` frames which you send are being ignored. In this case, either configure the remote to not expect authentication or configure the local system to do authentication and supply the proper secrets.

  Examine the receipt of the LCP configure frame. If it shows an 'auth' type, then the remote is configured for authentication.

## 7.11 The /etc/ppp/ip-up scripts wont work.

The pppd process launches the program at the location /etc/ppp/ip-up when the IP layer goes up. It gives it parameters which define the line status. Such things include the device name, communications speed, and IP addresses.

However, what may not be clear is that it treats this file as a `program`. It is not a script. The program is started by using the exec() function of Linux.

What this means is that if you wish to use a script for these programs, then you must do two things.

- You need to have the file marked as executable with chmod. The proper mode for the file should be mode 100. Mode 500 is acceptable if you wish to read the file and mode 700 is acceptable if you wish to write to the file. The file should be owned by the root user.
- The file must have as the first line the sequence:

  ```
  #!/bin/sh
  ```

  The # character must be in the first character position of the very first line of the file. The interpreter program, /bin/sh in this case, may be any program which is expected to run the script. Most people will use the Bourne shell for this purpose. It is commonly stored in the location /bin/sh. Other commonly used interpreters are perl and csh. What is important is that the first two characters of the file be the # and ! characters respectively.

### 7.12 I can't connect to the merit network.

Some users of the merit network have indicated that it needs PAP. Did you try PAP authentication?

# 8 DIP

### 8.1 DIP does not have support for PPP's mode

The current version of dip-uri supports PPP in that it will execute the pppd process when you execute 'mode PPP'. However, there are many options which are needed for the proper operation of pppd. Since dip does not pass these to the program, they must be stored in the /etc/ppp/options file.

The dip program controls the establishment of the SLIP link. It controls the SLIP link with the aid of slattach, ifconfig, and route. These programs may be used to establish a SLIP link. They are not useful for the establishment of a PPP link.

The dip program may be used to dial the telephone and start the `PPP` software on the remote system. It is best used in this mode as the parameter to the '`connect`' option. However, you have the option to use dip to control the link. It is not important how pppd be executed to run the PPP link. It is only important that it be executed as it is a mandatory program for the PPP protocol.

While this is not a FAQ for dip, there is a common problem with dip and pppd. The dip process has the absolute pathname to the pppd process embedded within it. Until recently, the location for pppd was /usr/lib/ppp/pppd. It has moved to the /usr/sbin/pppd location. So, if you are unable to get dip to start pppd then check the pathnames in dip.

Additional information about the dip process is in the Net-2-HOWTO document.

# 9 Process termination

## 9.1 Is there a 'dip -k' for PPP?

No. There is no '`dip -k`'.

In the chat directory, there is a '`PPP-off`' script. This will stop the PPP link in the same manner as the '`dip -k`'.

I have included it below. (Cut it out. Store it in its own file. Make the file executable with chmod.)

```
\#!bin/sh
DEVICE=ppp0
\#
\# If the ppp0 pid file is present then the program is running. Stop it.
if [ -r /var/run/\$DEVICE.pid ]; then
        kill -INT `cat /var/run/\$DEVICE.pid`
\#
\# If the kill did not work then there is no process running for this
\# pid. It may also mean that the lock file will be left. You may wish
\# to delete the lock file at the same time.
        if [ ! "\$?" = "0" ]; then
                rm -f /var/run/\$DEVICE.pid
                echo "ERROR: Removed stale pid file"
                exit 1
        fi
\#
\# Success. Let pppd clean up its own junk.
        echo "PPP link to \$DEVICE terminated."
        exit 0
fi
\#
\# The PPP process is not running for ppp0
echo "ERROR: PPP link is not active on \$DEVICE"
exit 1
```

## 9.2 PPP does not hangup the modem when it terminates

There are several reasons for this.

- Did you use the pppd '`modem`' parameter? This parameter controls whether or not the `pppd` process is to control and honor the signals reflecting the modem status. This parameter is explained in the man page for `pppd`.

- Do you have the modem presenting the DCD signal and honoring DTR? The Hayes sequence for this is usually "&C1". If you reset the modem during the connection sequence with "ATZ" then ensure that your modem is configured correctly. The DTR signal is generated by the computer and instructs the modem to disconnect. Hayes sequence for this is usually "&D1" or "&D2" with "&D2" being the preferred setting for PPP. Many manufacturers will ignore the DTR condition in their 'factory defaults' setting.
- Did you use a cheap cable which does not pass the DCD signal? Macintosh 'Classic' cables are notorious for this problem. The Macintosh Classic does not use this signal.
- For dial-in connections, did you exec the pppd process properly?

  The pppd process should be 'exec'ed from the script rather than simply executed. If you attempt to simply run the pppd process then it will be the shell which will receive the SIGHUP hangup signal and not the pppd process. The 'shell' script should have a format similar to the following:

```
#!/bin/sh
exec pppd -detach modem ...
```

- The use of **dip** and **diald** has, on occasion, interfered with the ability of pppd to sense the loss of the carrier. In this case, you should use the lcp-echo-request and lcp-echo-failure options to detect the loss of the connection in-band.

# 10 Data Transfer related issues

## 10.1 The ftp transfers seems to die when I do a 'put' operation. They will work correctly if I 'get' a file. Why?

Do you have the flow control enabled? Flow control is set by the pppd option `crtscts` for RTS/CTS and `xonxoff` for XON/XOFF. If you don't enable the flow control then you will probably overrun the modem's buffers and this will prove to be disastrous with vj header compression.

## 10.2 How do I use XON/XOFF for flow control?

The better flow control is CTS/RTS. However, if you can not do the hardware flow control with the signals CTS and RTS, then use XON/XOFF. The following three steps need to be performed.

- You need to specify the pppd option `xonxoff`. This tells the pppd process to configure the serial device for XON/XOFF flow control and to load the two characters into the tty driver.
- You need to specify the XON and XOFF characters in the pppd parameter `asyncmap`. This tells the remote system that is should quote the XON and XOFF characters when it wishes to send them to you. It is normally specified as the pppd parameter '`asyncmap a0000`'.
- Of course, don't forget to tell the modem to use XON/XOFF flow control. My `ZyXEL` modem uses a sequence '&R1&H4' to do this.

## 10.3 The modem seems to always connect at a strange rate. When I use minicom, the modem will always use 14400. However, PPP is using 9600 or 7200 or even 2400. How do I fix this?

Put the desired rate as an option to the pppd process. If you don't put the rate, then pppd process will use whatever rate is set currently at the time. Not all programs will restore all of the parameters to the previous settings properly upon exit. This may lead to strange rates configured for the serial device.

Linux does not support modems which use the RPI (**R**ockwell **P**rotocol **I**nterface) proprietary specification. Given the proprietary nature of the specification (even if you signed a NDA Rockwell will not release the code needed to interface to the modem) it is **extremely** unlikely that Linux will **ever** support this modem. The only solution, should you have a RPI modem, is to take it back to the dealer and get one which does not use RPI.

Some of the catch phrases to avoid are modems which are marked as having error correction in software, "windows" compatible, or "requiring a special driver" for full operation. These usually indicate that the modem uses RPI.

## 10.4 The ftp transfers seems to be very slow when I do a 'get' operation. The 'put' operation is much faster. Why?

Did you specify the option:

**asyncmap 0**

when you ran pppd? If you forgot the option, the peer must quote (double) all of the control characters in the range from 00 to 1F (hex). This will result in a statistical loss of about 12.5% in speed for all of the data which you receive.

Did you configure the remote system? If so, did you forget flow control on its modem?

## 10.5 The proxyarp function fails to find the hardware address.

Use the `ppp-2.1.2d.tar.gz` package. The `pppd` process was erroneously compiled with the 1.1.8 kernel and it used `Net-3` rather than `Net-2` definitions.

Additionally, you should refer to the proxy-ARP mini-HOWTO about the requirements for using proxy-ARP.

The 2.1 package had a limit of 64 network devices. At the the that the proxyarp function was written, 64 seemed to be a very likely limit as most people had one or two ethernet controllers. This is no longer the case when we consider that some systems routinely have 128 network devices.

The 2.2 package has raised the limit to 256 network devices. It is a compile-time define in the sys-linux.c module.

# 11 Routing and other problems

## 11.1 My route to the remote keeps disappearing! It last for about 3 minutes and then the route just goes away. Help!

This is not a question for PPP.

Hint: **DON'T RUN `routed`!**

## 11.2 I would like to attach my other computers on my network to the Internet through my PPP connection. I have only the one IP address which is assigned to me from my service provider. (It may even have been dynamically assigned.) How may I do this?

You may not. At least, you can't do it in the manner that you would normally want to do it. The problem is that your provider would not know about the IP addresses of your local network and therefore wont route the frames to your local system.

There are other solutions, however.

- You may telnet to your one computer running pppd and then use telnet or ftp to reach out to the rest of the Internet. This is not really much better then just using the computer directly, but it does work for simple things.
- You may run a recent 1.3 kernel and use the "IP Masquerade" option. For instructions on how to use this facility you should join the linux-net developer list or refer to the Net-2-HOWTO document.
- You may run the `socks` program on your PPP system. This will perform the same facility as the IP Masquerade but it will take modified clients. The advantage is that the socks program has been around for some years and many clients will understand the concept of a 'proxy' server which is needed to work with socks.

## 11.3 I can reach the remote server, but I can not get anywhere else.

Did you forget the '`defaultroute`' parameter to pppd? This parameter adds a default route into your routing system so that frames to all other IP addresses will be sent to the PPP device.

The PPP software will not replace the default route if you have one already set when you run pppd. This is done to prevent people from destroying their default route to the ethernet routers by accident. A warning message is written to the system log if the defaultroute parameter is not performed for this reason.

## 11.4 I have a default route and I still can't get anywhere else! Now what?

The problem then is not with the local Linux system. It most likely is routing problem on the remote end.

The remote system is not configured for '`IP forwarding`'. It is an RFC requirement that this option **NOT** be enabled by default. You must enable the option. For Linux systems, you will need to build the kernel and specify that you want IP forwarding/gatewaying.

The remote computers need a route back to you just as you need a route to them. This may be accomplished by one of four methods. Each has advantages and limitations. You need to do one and only one of these.

- Use a host route. At each host on the remote system, add a host route to your Linux IP address with the gateway being the terminal server that you use for your local access. This will work if you have a small number of host systems and a simple network without bridges, routers, gateways, etc.
- Use a network route. Subdivide the remote IP addresses so that your local Linux IP address and the remote terminal server address and the remote terminal server's ethernet address is on the same IP network. This will work if you have the IP addresses to spare. It will work very well if you have a Class-B IP network and can afford to put the all of the remote addresses on the same IP network. Then add a network route on each of the gateways and routers so that any address of the remote network is sent to the terminal server. Most configurations have many hosts but few routers. (At `sii.com`, we have over 300 active host systems with only 3 routers.)
- Use `gated` on all of the gateways and on the terminal server. This will cause the terminal server to broadcast to the gateways that it can accept the frames for your IP address. Since the hosts will have a default route to one of the gateways, the gateways will generate the ICMP re-direct frame and the specific host will automatically add its host route.
- Use proxy ARP on the terminal server. This will only work if your remote IP address is in the same IP domain as one of the domains for the network cards.

There is no clear solution. You must choose one of these.

If your remote router requires to receive RIP frames in order to update the route to your system then you should use the `bcastd` program on sunsite.unc.edu. This will generate the RIP frames without actually running gated.

## 11.5 I can not ping my local IP address

You are not able to do this because you wont normally have a route to the address. This is the normal operating environment.

If you wish to ping your own system then use the loopback address of 127.0.0.1.

You may be able to ping the remote address. However, some terminal servers may not allow this as the address may be 'phony' to them. It depends upon their environment.

In general, don't try to ping either address. Choose a third address which is well known to be available on the remote network such as one of your name server IP address.

While the PPP software will not perform this task, you may add the route table entry yourself once the link has been established. The syntax for the route statement is:

```
route add -host 192.187.163.32 lo
```

where the local IP address is represented as 192.187.163.32 in this example. This will tell the network software to route all frames destined to your local IP address to the loopback adapter. Once you add the appropriate route to the local IP address then you may use this address as the target to IP frames.

You will be responsible for deleting the route when the link goes down.

# 12 Interactions with other PPP implementations

## 12.1 I am using a Trumpet (for MSDOS) and the connection simply terminates. Why is this happening?

Trumpet does not like any VJ header compression. Use the pppd option "-vj" to turn it off.

## 12.2 I am using dp-3.1.2 (with SunOS) and the system will not allow me to use anything but ping, or nslookup. Why is this happening?

There is a bug in the **3.1.2** version of dp. Please get the **3.1.2a** or later file from the dp ftp home site harbor.ecn.purdue.ecu. Until you can put the patch into dp, disable the vj header compression.

## 12.3 I can not connect to/with my Windows NT code

Microsoft has chosen to support a non-standard authentication protocol with Windows NT. That is their right to do so provided that they have registered the protocol number with the IANA. (They have.) If the 'accept only Microsoft encrypted authentication' check box is set in the phone book entry, the connection will not complete. This setting mandates that the Windows NT system only exchange PPP authentication with another Microsoft PPP implementation.

The current pppd 2.2.0e code has patches to support this authentication protocol. However, the support is available in one direction only. It will connect to a Windows NT server. It will not permit a Windows NT server to dial-up the Linux system and connect with this protocol.

However, these patches for the Microsoft authentication require the use of a DES library. Please respect the governmental regulations concerning the distribution of this form of 'munitions'.

If you can not use this package and you have the option of changing the settings on the Windows NT system then go to the Windows NT Phone Book settings, advanced, security settings and ensure that the 'Accept any authentication including clear text' box is **checked** and the 'accept only Microsoft encrypted authentication' is **not checked**. The other check boxes may be checked or not as you see fit.

Then use PAP on the Linux side. Put your Windows NT account name and password into the /etc/ppp/pap-secrets file.

The Microsoft authentication sequence is a PAP style authentication with their DES encryption algorithm for the passwords. Normal PAP sends the passwords in clear text. This would violate their C2 security goals.

Versions of the Linux PPP code earlier than 2.1.2c have a flaw in their decoding of the authentication request. They will not work with a Windows NT system as they will not negotiate the proper authentication. Please used 2.1.2c or later if you wish to connect to Windows NT. The current version, 2.2 or 2.1.2d if you need 1.1 kernel support, should be used if possible.

Scott Hutton <shutton@habanero.ucs.indiana.edu> sent me the following:

Basically, NT RAS (Remote Access Services) will drop your connection if you REJ anything critical (i.e., authentication protocol). So, the trick was to create a *mostly* bogus chap-secrets file. Mine has

```
"" * ""
```

in it. This causes pppd to send a NAK rather than a REJ. With the SPAP registry key removed, the next protocol attempted is PAP (which is what I'm using).

Other points are to make sure that *only* TCP/IP services are enabled in RAS (not NetBEUI nor IPX [`Ed: IPX is being addressed. Until it is installed properly, this is probably a good thing to disable as well.`]). I also had to fiddle with a couple of other registry keys to kill timeouts (which are problematic when you're only doing TCP/IP):

```
HKEY_LOCAL_MACHINE\(\backslash\)SYSTEM\(\backslash\)CurrentControlSet\(\backslash\)Services\(\backslash\)Remo
    Autodisconnect: REG_DWORD: 0
```

and to get my routing to work correctly:

```
HKEY_LOCAL_MACHINE\(\backslash\)SYSTEM\(\backslash\)CurrentControlSet\(\backslash\)Services\(\backslash\)RasA
    DisableOtherSrcPackets: REG_DWORD: 0
```

For completeness, the key that needs to be disabled to eliminate SPAP:

```
HKEY_LOCAL_MACHINE\(\backslash\)SYSTEM\(\backslash\)CurrentControlSet\(\backslash\)Services\(\backslash\)RasM
```

# 13 Other messages written to the system log

## 13.1 `Alarm`

This is not a problem. It means that a timer has expired. Timers are a necessary part of the protocol establishment phase.

## 13.2 SIGHUP

The pppd process has received a HUP signal. The HUP signal is generated by the tty software when the remote system has disconnected the modem link. It means that the modem has put the 'telephone receiver back on the hook', or, 'Hung UP' the connection.

The kill program may also be used to send this signal to the pppd process.

The pppd process will terminate the link in an orderly fashion when it receives this signal.

## 13.3 SIGINT

The pppd process has received an INT signal. The INT signal is generated by the console software when you press the Ctrl-C key combination and pppd is the foreground process.

The kill program may also be used to send this signal to the pppd process. In fact, the recommended method to terminate the pppd link is to send the process an INT. See the question relating to "dip -k" for a script which will perform this task.

The pppd process will terminate the link in an orderly fashion when it receives this signal.

## 13.4 Unknown protocol (c025) received!.

The remote wishes to exchange Link Quality Reporting protocol with the Linux system. This protocol is presently not supported. This is not an error. It is merely saying that it has received the request and will tell the remote that "I can't do this now. Don't bother me with this!"

The Morning Star PPP package will always try to do LQR protocol. This is normal.

## 13.5 Unknown protocol (80fd) received!.

The remote wishes to exchange Compression Control Protocol with the Linux system. This type of protocol is layered upon the basic data protocol and will, if successfully negotiated, result in a fewer number of bytes transmitted for the frame. This means that the transfer will be quicker.

However, there are many types of compressors which are used under the general 'umbrella' of a Compression Control Protocol. The 2.2 PPP package understands only one; the BSD compressor. This compressor works very similar to the Unix 'compress' program and uses a LZW compressor. Depending upon the size of the code, there can be a significant amount of kernel space needed to hold the compression and decompression dictionaries. This should not be used if you have a limited memory space and should not even be contemplated if you have 8Meg or less real (RAM) memory. In those cases you should invest in a decent modem which support compression.

Unless both sides can agree upon the type of compression the compression will not be used.

Another common compressor is called Predictor-1. This will take less memory and run faster. However, its compression is not as good in that it will send a little more data than the equivalent frame given to the BSD compressor.

Many commercial terminal servers will employ a compressor called "Stacker(TM) LZW" or LZS protocol. This is a commercial compression agent. Apparently Stacker will give you a license for no charge. However, a specific license is required and that will usually prevent it being included with the pppd process.

The 2.3 package will additionally include the compressor known as 'deflate'. It is a variation of the common package called 'gzip'.

## 13.6 The connection fails with errors "ioctl(TIOCGETD): I/O error" or "ioctl(PPPIOCSINPSIG): I/O error". What now?

Look at the boot messages when you boot the kernel. If it says "`PPP version 0.1.2`" then you have an old version of the `PPP.c` driver.

If it says "`PPP version 0.2.7`" then you have the current driver, for the 2.1.2 package however, it was not built with the same set of defines for the ioctl numbers. Ensure that you have only one file called "`if_ppp.h`". It should be located in the kernel's `include/linux` directory. Once you have done this, rebuild the kernel and the pppd process.

If it says "`PPP version 2.2.0`" then you are using the driver for the 2.2.0 package. This version of the driver will only work with the 2.2 series of the pppd package. The 2.2 pppd program will only work with this version of the driver.

## 13.7 Sometimes the messages "ioctl(PPPIOCGDEBUG): I/O error", "ioctl(TIOCSETD): I/O error" and "ioctl(TIOCNXCL): I/O error" occur. Why?

The remote system has disconnected the telephone. The tty drivers will re-establish the proper tty discipline and these errors are the result of the `pppd` process trying to do the same thing. These are to be expected.

## 13.8 My `ifconfig` has strange output for PPP.

Usually the ifconfig program reports information similar to the following:

```
ppp0      Link encap UNSPEC  HWaddr 00-00-00-00-00-00-00-00 ...
          inet addr 192.76.32.2  P-t-P 129.67.1.65  Mask 255.255.255.0
          UP POINTOPOINT RUNNING  MTU 1500  Metric 1
```

The information is for display purposes only. If you are using a recent kernel then update the nettools package with the current one on `sunacm.swan.ac.uk` in the directory `/pub/Linux/networking/nettools`.

## 13.9 The file `/proc/net/dev` seems to be empty

Did you just issue the command "`ls -l /proc/net`" and are wondering why the size is zero? If so, this is normal. Instead, issue the command:

cat /proc/net/dev

You should not find the file empty. The size is always shown as zero, but that is the 'proc' file system. Don't believe the size. Do the command.

The 'more', 'less', and 'most' programs may not be used to view the file directly. If you wish to use these programs, use it as follows:

cat /proc/net/dev | less

### 13.10 The kernel reports that the PPP devices are being unlinked when the system is being started.

This is not a problem. The message is the result of the ppp driver calling the procedure 'unregister_netdev'. This permits the ppp driver to dynamically allocate the devices as they are needed. There is no real limit to the number of devices which may be created. For the sake of setting a limit, the value of 256 was chosen as the maximum number of devices. Should you find that this is too small then you may change the define in the ppp.c code to make it any value that you wish or supply the value when you use the dynamically loaded module.

If you are receiving these messages then you have an old version of the ppp.c driver. Please consider upgrading the driver as the old scheme of calling dev_unlink is 'bad'.

### 13.11 I just checked /proc/net/dev and there are no PPP devices. Where did they go?

They went nowhere. They were all unlinked during the startup of the system. Please see the previous question for additional information.

## 14 Network routing issues (using PPP as a 'cheap' bridge)

### 14.1 `Slattach` and `ifconfig` don't work like SLIP

Do not use `slattach` and `ifconfig` with PPP. These are used for SLIP. The `pppd` process does these functions at the appropriate time. These must occur after the `LCP` and `IPCP` protocols have been exchanged.

You can not replace `pppd` with `slattach` and `ifconfig`. Most of the protocol support for PPP is in the `pppd` process. Only the IP (and `IPX` when it is completed) processing is in the kernel.

The host route to the remote system will be automatically added by pppd. There is no option to NOT add the route. The pppd process will terminate if the route could not be added.

The default route may or may not be added. This is controlled by the option '`defaultroute`'. If you have a default route, it will not be changed.

If you must do routing for an entire network, then put the route command into the `/etc/ppp/ip-up` script. The parameters to the script are:

```
$0 - name of the script (/etc/ppp/ip-up or /etc/ppp/ip-down)
$1 - name of the network device (such as ppp0)
$2 - name of the tty device (such as /dev/cua0)
$3 - speed of the tty device in Bits Per Second (such as 38400)
$4 - the local IP address in dotted decimal notation
$5 - the remote IP address in dotted decimal notation
$6 - the value of the ipparam parameter
```

### 14.2 I want the route to the network and not the route to the host.

On `sunsite` there is a package called `devinfo.tar.gz`. It contains some useful little programs which will extract the data from the device and to do various things with the dotted IP addresses.

The documentation is in the man pages in the file.

For example, if you want to route the entire IP domain to the remote, the following may be used in `/etc/ppp/ip-up`.

Of course, if the values are not variable, then simply use the appropriate entry in the route command.

```
# Obtain the netmask for the ppp0 (or whatever) device
NETMASK = `devinfo -d $1 -t mask`

# Obtain the IP domain (without the host address by removing the extra bits)
DOMAIN = `netmath -a $5 $NETMASK`

# Do the network route now that the IP domain is known
route -net add $DOMAIN gw $5
```

## 15 Other features and protocols

### 15.1 What about support for 'demand dial'

Use the **diald** package. This is on sunsite in the same directory as the PPP source, `/pub/Linux/system/Network/serial`.

### 15.2 What about 'filtering'

There are no plans to put filtering into the PPP code. The 1.3 kernel supports a firewall option and you should use that rather than attempt to find a method of putting firewall logic into a network device driver. Use either the `ipfw` or `ipfwadm` programs to define the rules for the firewall code in the kernel.

### 15.3 How about IPX?

It is in the 2.2.0e package.

### 15.4 How about NETBIOS?

There is a netbios PPP protocol. However, your better solution would be to use TCP/IP and the '`samba`' code.

Microsoft and others have used Netbios PPP protocol.

The nbfcp protocol is a public document and available from several sources. The Netbios protocol is not a valid address family at the present time for Linux. Until Linux supports the protocol, there is little need to support Netbios over PPP for Linux.

### 15.5 I need ISDN support. Is there any?

ISDN support revolves around having a working ISDN driver. The present design of the PPP driver does not lend itself well to the concept of a block of data being received. This is being changed. A driver for the Sonix interface is being developed.

### 15.6 I would like multi-point support. Is there any support?

Multi-point would be nice. I am not aware of anyone working on multi-point support at the present time.

### 15.7 How about just standard synchronous PPP?

There are small changes needed to support a serial interface which uses synchronous communications. The redesign of the PPP driver will help with this function as well. Kate Marika Alhola has expressed an interest in writing such a synchronous driver for her hardware. You should contact her at kate@iti.fi or kate@nic.funet.fi for further information.

She informs me that the current status of sync ppp is, that I have had it few months in "production" use talking with Cisco(TM) in speeds 64K and 256K. The source is under the GPL license and it may be found in ftp://nic.funet.fi/pub/Linux/kernel/xnet-sync-driver-1.0.tar.gz.

## 16 Miscellaneous

### 16.1 Do you have a PPP compatible mail reader?

Huh? You have the wrong group if you want MSDOS. PPP has nothing to do with the mail user agent. All of the mail agents are compatible with PPP.

### 16.2 How about a news reader?

Refer to the previous answer.

# 17 Questions about chat

## 17.1 My modem wont dial when I run chat

The modem is required to be in the command mode to issue dial commands. If your modem is 'online' then characters sent to the modem will be sent to the remote system.

If possible, configure the modem to monitor the DTR signal and to return to the command mode when the DTR signal dropps. This will permit the computer to force the modem back to the command mode when the pppd process terminates at the end of a connection. It will then be in the proper state when the next execution attempts to dial the telephone.

If you cant do this then you should change the dial sequence so that it is similar to the following. It will ensure that the modem is in the command state prior to attempting to send the dial sequence.

```
TIMEOUT 3 "" \rAT OK-+++\c-OK AT\&D2\&C1 TIMEOUT 60 OK ATDT555-1212 CONNECT
```

The commands will change the timeout period to three seconds. This accommodates the `guard` time period used by many modems. It will then send **AT** to the modem and look for its response of **OK**. If it is not received in the three seconds, it will send the +++ sequence to the modem and wait for the modem to present the expected **OK** response. Once it receives the valid response it will configure the modem and dial the telephone number.

## 17.2 The modem dials only on every second attempt

Please refer to the above answer. It is usually the same issue.

## 17.3 The chat script stops after sending the account name and it never receives the password prompt.

Some systems, notably SCO, will flush the receive buffers after writing the prompts for user name and password. The chat program normally transmits the response immediately upon seeing the prompt. The result is that the reply from chat is flushed by SCO. The chat program continues to wait for the password prompt. However, the remote system is still waiting for the user to enter the account name.

The solution is simple. Slow down the responses from chat so that there is time for the remote system to flush the receive buffer before chat starts to send the response. Chat supports this with the parameter. Change the response strings similar to the following:

```
ogin:--ogin: \d\daccount assword: \d\dhello2u2
```