PPP−2.2/PPP−2.3 for NeXTSTEP FAQs

There are two sets of FAQs for PPP. The first, comp.protocols.ppp Frequently Wanted Information, is maintained by Ignatios Souvatzis <ignatios@cs.uni−bonn.de> and is a FAQ that covers many different PPP implementations for many different computer platforms. It is very general in nature. The second, this document, identifies several common problems that one may encounter when trying to set up PPP on a NeXT system. Select one of these links to jump to the appropriate section:

- *How do I set up my system to send mail*
- *Netinfo Sleeps whenever I try to send mail*
- *How do I set up my system as a PPP server*
- *How can I have getty run at 57600?*
- *I get the error* **ioctl(PPPIOCGUNIT): Inappropriate ioctl for device**
- *pppd says my link is not 8−bit clean*
- *I have an /etc/resolv.conf file but I still can't resolve names*
- *I changed my address to 127.0.0.1 but now my machine won't boot*
- *When I try to run TCPDUMP I get errors about* ***/dev/bpf0*** *not being found*
- *When I start pppd I get the message* ***Sorry − PPP is not available on this system***
- *I stopped getting output to* ***/usr/adm/messages*** *or* ***/usr/adm/ppp2.2.log***
- *Using PPP with TTYDSP on a NeXT cube fails*
- *If I do not include BPF, then compilation fails*
- *I am having troubles creating a* ***chat*** *script*
- *How can I create virtual host interfaces (Multi−home the machine)?*

How do I set up my system to send mail?

François Magnan <magnan@MATHCN.UMontreal.CA> has written an excellent set of instructions for configuring sendmail 8.7.5 for use with a PPP link. Please refer to this document for further details.

Netinfo Sleeps whenever I try to send mail

Problem: Your link appears to come up successfully. However, if you try to send mail or perform any other operation that requires access to netinfo, you get a message similar to:
sendmail[546]: NetInfo timeout connecting to local domain, sleeping

Discussion: This problem is typically a result of an improper routing setup on the PPP client (your host). When PPP starts up, it will dynamically negotiate an IP address for use on the PPP interface (usually ppp0). If you don't specify an IP address to pppd, then the address will usually be provided by the peer. It may change each time that you bring up a PPP link. If you did specify an IP address, pppd will attempt to use that address first. If that fails, it will try to get an address from the peer. Either way, the PPP interface usually has a new IP address. The routing problem can result when trying to access your local host (and netinfo) on the IP address that was dynamically negotiated.

Solution: The solution is straightforward. You must add a route from the IP address that your PPP interface uses to the special loopback IP address 127.0.0.1. The *route* command will allow you to do this. For instance, if you are assigned the address 35.8.74.211 during PPP negotiation, you can add the needed route by entering (as the user *root*):

```
/usr/etc/route add 35.8.74.211 127.0.0.1 0
```

This route needs to be added each time the link comes up. However, one problem with hard coding this command into /etc/ppp/ip−up is that you may get a different IP address each time the link comes up. Fortunately, Bill Bereza <berezaw@river.it.gvsu.edu> submitted this nice script clip that can be pasted into /etc/ppp/ip−up. This will create the correct route entry for you automatically. Place this in /etc/ppp/ip−up:

```
/usr/etc/route add $4 127.0.0.1 0
```

You will also want to add this little clip to /etc/ppp/ip−down:

```
/usr/etc/route delete $4 127.0.0.1
```

This removes the route when your link goes down.

How do I setup my system as a PPP server?

Izumi Ohzawa <izumi@pinoko.berkeley.edu>

writes:

```
Q: How does one set up a PPP server on NS?

A:

Here's a summary of steps to set up a PPP Client <-> Server connection
where *you* have to set up the server side instead of using a PPP
service offered by your university or other Internet providers.

My configuration is as follows but it should work with other NS
platforms and modems.

Home:
  NeXTcube 040 25MHz NS3.2 +USR SportsStar 14.4k +ppp2.2-0.1.9
Server:
  NeXTcube 040 25MHz NS3.2 +ZyXEL U1496E +ppp2.2-0.1.9 (+NXFax 1.04)

This note only illustrates specific scripts and options for PPP
server setup.  It doesn't cover PPP installation itself, e.g.,
expanding ppp installer package, editing /etc/rc.local, etc.
Follow PPP install instructions for that.

PPP server setup described here essentially places your home machine
as one of the local machines on the same subnet as the server, using
the proxyarp mechanism (Server and home machines have the same
subnet number).  This way, nothing has to be configured beyond the
```

PPP server (e.g., in router/gateway), which means one less thing to ask your network administrator.

*** Note in the steps below:
Edit IP addresses to suit your local domain.
128.32.111.sss is the IP address of PPP server "pppserver".
128.32.111.hhh is the IP address for the home machine "homehost".

IMPORTANT:
You should NOT use just any 'hhh' you pick. Make sure that the IP address for your home machine is properly allocated by your network administrator and registered with DNS.

==========
Office/Campus PPP server side setup:
==========

[0] (Skip step [0] and goto [1] if you use NXFax)
First configure dial-up login so you can login to the server machine via modem/serial port as a regular shell user.
I haven't tested this, because I am using NXFax, but the key points are as follows:

(a) modify /etc/ttys
Turn on the 'ttydfa' or 'ttydfb' line, i.e.,

```
ttydfa  "/usr/etc/getty D38400"          unknown         on
```

(b) modify /etc/gettytab
Define D38400, e.g.:

```
D38400:ap:p8:im=\r\n\r\nNeXT (%h) (%t)\r\n\r\r\n\r:sp#38400:
```

(c) kill -HUP 1 (or reboot)

[NOTE: with NXFax, the above is not needed because FaxDaemon will exec getty only when it detects a data call. Enable data call in NXFax.]

[1] Create user "ppp0" in NetInfo, using UserManager.app or "nu -a" and specify /usr/local/bin/pppserve as this user's shell. This way, pppd will be automatically started when user "ppp0" logs in.

Check with:

```
% nidump passwd / | grep ppp
ppp0:BZKIbZ8A3t4aM:121:20:PPP:/Users/ppp0:/usr/local/bin/pppserve
```

[2] Make sure the script pppserve has execute permissions.

```
# chmod a+x /usr/local/bin/pppserve
```

```
# ls -l /usr/local/bin/pppserve
-rwxr-xr-x  1 root wheel   616 Jun 11 17:00 /usr/local/bin/pppserve*
```

```
[3] #### Here's the pppserve script itself
--- This is almost the same as Examples/pppup.remote ------------
% cat /usr/local/bin/pppserve
#!/bin/csh -f
#
# This script is used as the shell for a user account entitled ppp.
# When users log into this account, PPP is automatically started
# (through this script).  The 'echo' command is to help the remote
# process synchronize with this end.
#
echo Starting PPP
/usr/bin/mesg n
/bin/stty -tostop litout
exec /usr/local/bin/pppd
```

```
[4] Setup necessary info for "proxyarp"
Try command:
% nidump -r machines /
```

The output must include something like below for "proxyarp" on the server to work. Otherwise, you must have /etc/ethers file (see "man 5 ethers"). See also "man pppd", "man 8 arp" and "man 4 arp" for info on proxyarp. Proxyarp advertises to the subnet that packets destined to homehost should be sent to the ethernet address of pppserver (pppserver acting as proxy for homehost). Without proxy ARP working, IP packets from homehost can go out to the Internet, but packets from remote sites cannot come back to your home machine. This will result in the condition where TCP/IP stuff works only between homehost and pppserver, but attempts to connect to outside will fail (though you can connect to outside once you rlogin/telnet to pppserver). It is not necessary to execute "arp -s ..." command explicitly on pppserver. Option "proxyarp" to pppd will take care of everything dynamically as the link is brought up and taken down (see step [5]).

```
    name = (pppserver pppserver.berkeley.edu);
    bootfile = mach;
    bootparams = ();
    en_address = 0:0:f:0:c1:3a;
    ip_address = 128.32.111.sss;
    serves = pppserver/local;
```

"en_address" is the ethernet address that is unique among all ethernet interfaces (hostid is typically derived from this). The ethernet address of any NS machine should be recorded in the boot log in /usr/adm/messages, so to find out, do:

```
% grep Ethernet /usr/adm/messages
```

If the PPP server is also a NetInfo config server (master), then it might not have the above entries for itself. Either add it, or create /etc/ethers with a line like

```
0:0:f:0:c1:3a   pppserver
```

To check if "proxyarp" is working (while the PPP link is up), command "arp -a", on pppserver should show:

```
   homehost (128.32.111.hhh) at 0:0:f:0:c1:3a permanent published
```

Command "arp -a" on other hosts on the same subnet as pppserver should show (after "ping" to each):

```
   pppserver (128.32.111.sss) at 0:0:f:0:c1:3a
   homehost (128.32.111.hhh) at 0:0:f:0:c1:3a
```

[5] Prepare options file for pppd (/etc/ppp/options).
Here, I specify the IP address pair for client/server on the server side, so home machine doesn't have to specify them explicitly. This makes it easier to configure the client for multiple server possibilities (e.g., Annex server run by your institution or your own modem on your office machine).

```
% cat /etc/ppp/options
128.32.111.sss:128.32.111.hhh passive proxyarp bsdcomp 10,10
lcp-echo-interval 15 lcp-echo-failure 3
```

[6] Troubleshooting
In case of problems, try logging in via rlogin from another network host as user "ppp0", and check if it starts the script /usr/local/bin/pppserve correctly. PPP won't work, but you should get something like below. If you don't get it, check execute permissions and path for the login shell for user ppp0 in NetInfo:

```
% rlogin pppserver -l ppp0
Password:
Last login: Fri Jun 23 01:49:00 on ttydfa
Starting PPP
~_!j~~_!
 ....(several lines of garbage)...
[Close Terminal window here.]
```

[7] /etc/shells should NOT contain /usr/local/bin/pppserve.
You should not include the login shell for user ppp0 in /etc/shells. Not having it makes the system a bit more secure, especially if you just want to allow Internet access to user ppp0, but do not want ppp0 to see the server's file systems. FTP daemon will deny access to user ppp0 if you leave /etc/shells alone.

```
=====================================================
Home side setup:
=====================================================


[H1] ---- /etc/ppp/options ---------
/dev/cufa 38400 mru 296 mtu 296 lcp-echo-interval 15
lcp-echo-failure 3 crtscts defaultroute modem


[H2] ---- /usr/local/bin/pppup ----------------
#!/bin/sh
#

exec /usr/local/bin/pppd netmask 255.255.255.0 bsdcomp 10,10 connect \
"/usr/local/bin/chat -v -f /etc/ppp/script.office"



[H3] ---- /etc/ppp/script.office -----------------

ABORT BUSY ABORT "NO CARRIER" ABORT "NO DIAL TONE"
ABORT "ERROR" "" ATZ0 "OK" ATDT6431234 CONNECT
"" ogin: ppp0 assword: xxxxxxxx PPP


[H4] Home machine has the same /etc/resolv.conf as that on the server.


----
I just type "pppup" on the home machine to bring up the link.
```

How can I have getty run at 57600?

Problem: The NeXT supplied *getty* operates at a maximum speed of 38400.

Discussion: Those people who allow dialin acess to their NeXT system must enable *getty* on the appropriate serial device. However, one problem with the *getty* supplied with NeXT is that it has a maximum speed of 38400. Those wishing to dial into a NeXT PPP server are thus limited by the DTE speed of the peer. However, thanks to the work of Gert Doering <gdoering@physik.tu–muenchen.de> , there is a replacment getty that will run on NeXT systems. While Gert's *mgetty* is intended for Fax usage (and does work under NeXT, but that is another story), it may be used as a replacement for the NeXT supplied *getty*. *mgetty* will allow operation at 57600.

Solution: Get and install **mgetty–0.22** according to the directions. Before you perform the make, install these Patches curteously supplied by Mark Salyzyn <mark@ve6mgs.ampr.ab.ca> and Christian Starkjohann <cs@ecs.co.at>. Once *mgetty* is installed, you must edit /etc/ttys and change the getty

entry for your device. For my direct connection, my entry looks like:

```
ttyfa "/usr/local/bin/mgetty -x 9 -r -b -s 57600 ttyfa DNXFast"
unknown on secure
```

The *NXFast* entry in /etc/gettytab looks like:

```
DNXFast::ap:im=\r\n\r\nNeXT / NXFast (%h) (%t)\r\n\r\r\n\r:sp#57600:
```

I am not certain that the DNXFast entry is even needed with mgetty.

I get the error: ioctl(PPPIOCGUNIT): Inappropriate ioctl for device

Problem *pppd* bombs out with an error similar to:

```
Jan 26 14:46:25 localhost pppd[256]: Connected...
Jan 26 14:46:26 localhost pppd[256]: ioctl(PPPIOCGUNIT):
Inappropriate ioctl for device
Jan 26 14:46:26 localhost pppd[256]: Exit.
```

Discussion: There are two things that can cause this error. The most common error is that you are using the innappropriate device for communication to your modem. You should be using `/dev/cufa` or `/dev/cufb`. Second, the Loadable Kernel Server (LKS) and the user level daemon (*pppd*) must be of the same version. While unusual, the error above can be the result of using the wrong Loadable Kernel Server (LKS) with *pppd* (or vice versa). Some versions of PPP installed the LKS (ppp_reloc) in /usr/lib/kern_loader/ppp and some in /usr/local/ppp/reloc. The second is going to be the standard place for installation from now on.

Solution: Make sure that you are using the appropriate device for communications with your modem. Further, make sure that your /etc/rc.local is loading the correct version of the LKS and make sure you are really calling the correct pppd for use with the LKS that you loaded.

---

pppd says my link is not 8–bit clean

Problem: Your pppd/chat dials the modem but you cannot get a negotiation to start. The /usr/adm/ppp2.2.log file shows something similar to:

```
Mar 13 12:02:41 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:02:44 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:02:47 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
```

```
<accomp>]
Mar 13 12:02:51 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:02:54 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:02:57 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:03:00 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:03:03 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:03:06 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:03:09 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp>
<accomp>]
Mar 13 12:03:12 crystal pppd[243]: LCP: timeout sending
Config-Requests
Mar 13 12:03:12 crystal pppd[243]: Connection terminated.
Mar 13 12:03:12 crystal pppd[243]: Serial link is not 8-bit clean:
Mar 13 12:03:12 crystal pppd[243]: All received characters had bit 7
set to 0
```

Discussion: This is a common problem. It is typically the result of a failure to properly start the remote PPP process. The problem arises since the local PPP starts sending packets as soon as *chat* exits. Since there is no remote PPP process running to interpret the packets, the remote command line interpreter starts sending error messages for each received packet (considered garbage to the remote CLI). Thus, your local PPP process is receiveing error message text instead of the expected PPP packets. Since error message text is usually ascii (values < 127) PPP believes that the link is not 8−bit clean.

Solution: Make sure you add the '−v' option to chat (in your dial script) and then check the output of /usr/adm/ppp2.2.log to see why chat failed to start the remote PPP process.

I have a /etc/resolv.conf file but I still can't resolve names

Problem: You modified (or created an initial) /etc/resolv.conf file. You believe you put the proper information in for the nameservers. However, you still cannot resolve names. When you tried to fire up *nslookup*, it returned with a line similar to:

```
Default Server: sidney
Address: 0.0.0.0
>
```

The Address 0.0.0.0 was displayed instead of the address of the machine you described in /etc/resolv.conf.

Discussion: Name resolution is a complex process that maps computer names to IP addresses (and

vice versa). When you tell your machine to initiate an FTP session with host *ftp.thoughtport.com*, your machine must somehow figure out the IP address of the machine *ftp.thoughtport.com*. This process is called name resolution and is performed by the resolver.

While the details of this process are a bit tricky, it basically boils down to the fact that certain machines on the internet are designated as *name servers*. The resolver must contact a name server in order to perform the name to IP address translation. If you cannot contact a name server, then resolution will not occur.

The resolver gets the information about which name servers to contact out of the file /etc/resolv.conf (please note the spelling). This file can contain several lines, however the most important are the lines similar to:

```
nameserver <IP Address>
```

When name resolution is properly configured you may use the program called `nslookup` to query the nameservers. As part of this program's initial output, it will describe which nameserver it actually contacted. The IP address of that nameserver should be the same as one of the IP addresses you put in your /etc/resolv.conf file.

The problem shown above, where the IP address is 0.0.0.0, means that `nslookup` was unable to contact a nameserver. If no nameserver is contacted, then name resolution will generally not work.

Solution: There are 2 possible reasons why you may have an /etc/resolv.conf file but still fail to get name resolution.

- You mispelled the file name (/etc/resolv.conf is the correct spelling).
- You mistakenly supplied the actual host name instead of the IP address for your nameserver. Since this host must be contacted before name resolution can occur, you must supply an actual IP address and not a name. If you supply a name, you get into a catch 22 sitution.

When you believe you have a correct /etc/resolv.conf, start up `nslookup` and see if it is able to contact the nameserver. If not, please mail the group (nextppp@listproc.thoughtport.com) for further help.

I changed my address to 127.0.0.1 but now my machine won't boot

Problem: You followed the installation instructions regarding the setup of your host using the Hostmanater.app. However, after setting your address to 127.0.0.1, your machine would not reboot. Your system now times out when trying to access system services and essentially hangs during boot.

Discussion: This problem is caused by the local loopback interface not being properly configured during the boot process. The lo0 interface is not configured as UP at an early enough stage for some

system resources to use.

Solution: As the user root, make a backup copy of the file /etc/iftab. Call it /etc/iftab.prePPP. Then, edit /etc/iftab and place the line:
lo0 inet 127.0.0.1 netmask 0xff000000 −trailers up
on a new line right before the −1− entry. You may now boot your machine as IP address 127.0.0.1.

When I try to run TCPDUMP I get errors about /dev/bpf0 not being found

Problem: You have successfully installed the BPF LKS. The output of /usr/adm/messages verifies that the PPP interfaces were properly attached to BPF. However, you are still unable to start *tcpdump*. You get a message complaining that tcpdump cannot find /dev/bpf0.

Discussion: Proper installation of BPF requires that you not only load the NeXT LKS (called *bpf_reloc*) but also that you create some special devices which *tcpdump* will use for communication with BPF.

Solution: To create the special files, log in as root, and then in a terminal window, execute these five commands:

```
/usr/etc/mknod /dev/bpf0 c 32 0
/usr/etc/mknod /dev/bpf1 c 32 1
/usr/etc/mknod /dev/bpf2 c 32 2
/usr/etc/mknod /dev/bpf3 c 32 3
/usr/etc/mknod /dev/bpf4 c 32 4
```

Assuming that the BPF relocatable is properly installed, you should now be able to run *tcpdump* when a PPP link is up and active.

When I start pppd I get the message Sorry – PPP is not available on this system

Problem: The program *pppd* will not run. It exits with the error message **Sorry – PPP is not available on this system**.

Discussion: *Pppd* requires operating system support to work correctly. This support is provided in a module called a Loadable Kernel Server (LKS). The above error message is generated whenever the system detects that the PPP LKS is not properly loaded.

Solution: The LKS must be loaded each time your system is rebooted. In order to perform this task automatically, an entry must be added to the file called `/etc/rc.local`. This entry will automatically perform the operation each time the system reboots.

As user root, modify this file and add the following lines:

```
#
# Load the PPP LKS
#
if [ -f /usr/local/ppp/reloc/ppp_reloc ]; then
        /usr/etc/kl_util -a /usr/local/ppp/reloc/ppp_reloc
fi
```

This assumes that the PPP LKS is actually stored in the file `/usr/local/ppp/reloc/ppp_reloc`. Further for this entry to work immediately after modifying `/etc/rc.local`, you must execute the *kl_util* command by hand or perform a system reboot.

To check if PPP is currently loaded, you may execute the following command:

```
/usr/etc/kl_util -s
```

If there is a server loaded for PPP you are ready to go.

I stopped getting output to /usr/adm/messages or /usr/adm/ppp2.2.log

Problem: I used to get messages in */usr/adm/ppp2.2.log* or */usr/adm/messages*. However, I don't get them now.

Discussion: This is typically the result of modifying the actual file in question. When your system starts up, the syslog daemon (syslogd) opens the files that are described in */etc/syslogd.conf*. If you remove a file that syslogd has opened, then the daemon will not be able to correctly write to it. This is true even if you recreate the file. The daemon will have a file handle to the old non−existant file even though a new file exists. Further, if the file doesn't exist at syslogd startup time, the daemon will not recognize any new files created.

Solution: Make sure the files listed in /etc/syslogd.conf actually exist. If they do not, then as user *root*, create them using the `touch` command. For instance, if you want to initially create /usr/adm/ppp.2.2.log then execute:

```
/usr/bin/touch /usr/adm/ppp2.2.log
```

Once the file is created you must either reboot your system (safest) or you may send syslogd a 'kill −HUP' signal.

If you want to clear a file that has become too large, you can first copy the file to a backup and then issue the following command (substituting the appropriate file name):

```
cat > /usr/adm/messages
^D
```

The `^D` means you hold the control key down and type the D key. This will successfully clear out a file and simultaneously allow syslogd to continue writing to the file. This operation must be done as user *root*.

Using PPP with TTYDSP on a NeXT cube fails

Problem: When trying to use PPP−2.2 with TTYDSP, the PPP process fails with the message `tcgetattr: invalid argument.`

Discussion: TTYDSP is not compatible with PPP−2.2. PPP−2.2 uses line disciplines that will not interoperate with TTYDSP.

Solution: Currently None. A fix may be possible but would require lots of work. Sources are available for those interested in working on it. I have no plans to pursue this further. However, I might reconsider this decision as a paid consultant. Email me (Stephen J. Perkins <perkins@cps.msu.edu>) if interested.

If I do not include BPF, then compilation fails

Problem: I disabled BPF compilation in the Makefiles. However, the makefile still tries to compile BPF but dies in the process.

Discussion: This is caused by an error in makefile design. At the point where BPF starts to compile, the PPP LKS has already been successfully made (without BPF support). The Makefile shouldn't start trying to make the BPF LKS. The make fails because certain defines required by BPF are not defined when you disable BPF support.

Solution: Ignore the failed message. The PPP LKS (ppp_reloc) is correctly made (without BPF support).

I am having troubles creating a chat script

Problem: I cannot get my chat script to properly work.

Discussion: Chat is complex enough that one FAQ cannot describe all the problems. I'm hoping that some day, someone will volunteer to write a CHAT−HOWTO (in the spirit of the Linux HOW−TOs). For now, the only advice I can give you is to recognize that chat is usually called from within a shell script. You must be very careful to properly quote shell special characters you send to chat. For instance, you must be careful about symbols like `<>&|`.

Solution: Join the Mailing list and ask for help from the many fine people there. They can usually fix you right up!

How can I create virtual host interfaces (Multi–home the machine)?

Problem: I would like to make my machine answer to multiple IP addresses using a single interface.

Discussion: Many people who serve WWW pages want to create *virtual* machines. These are named machines that have independent IP addresses. However, all virtual machines are really represented by one physical machine. The PPP LKS allows you to do this, however, using the LKS gives you a lot of unnecessary baggage. Hideo "Sir MaNMOS" Morishita has written a small LKS that just provides for virtual hosting. This NEXTSTEP Mach loadable device driver allows NCSA httpd 1.5 (and later) and Apache httpd to use their virtual hosting capability.

Solution: Install Hideo Morishita's virutual hosting package. Directions for use are included with the package.

Last Updated July 4, 1996 by SJP

[IMAGE]