SLIP SERVER - USING DIALUP IP

versjon 0.81β
revidert 18.10.93

INTRODUCTION

This paper discusses various aspects of setting up and maintaining a SLIP connection. The primary audience is first time or novice users, as well as more experienced users who are not familiar with networking. The main focus will be on setting up and configuring a server to accept SLIP/PPP connections. This article grew out of my own work to set up SLIP, and then PPP, and then SLIP once again. It is my hope that no-one has to go through what I did. I see a real need for a comprehensive document which discusses many aspects of SLIP setup, and since I have a little spare time, well...

As this is a first rough working copy, I expect that there are omissions, inconsistencies, and perhaps even bugs (!), so PLEASE consult an expert if you have access to one. I am not an expert, although I have learned quite a lot by working with SLIP and PPP. If you have corrections or additional comments you feel should be included in this guide, please e-mail/fax me and we'll see what you have. My intention is to update and expand this document in the future to include many popular platforms, thus serving as a reference for SLIP installation for non-guru type users. At this point I have written specifically for the NeXT computer and SLIP, but the principles should apply to other platforms/software packages. I have purposely issued this document in PostScript form to keep some degree of control over it - I like the idea of one updated and revised publicly available version better than ten copies varying in accuracy and availability. If you would like this document in an editable form, e-mail me and we'll discuss it.

This document and its contents are based on a SLIP configuration for NeXT computers, using the popular public domain package TransSys Dialup-IP™ by Louis Mamakos <louie@TransSys.COM>. It may be applicable to other implementations and other platforms, but don't count on it. No guarantee of this guides fitness for any application or use is implied or expressed in any way, shape, or form. Use at your own risk. You are on your own.

The latest version of this document may be obtained via anonymous ftp to the following archives:

ftp.sunet.se /pub/network/papers/SlipServer.ps.Z
sonata.cc.purdue.edu /pub/next/docs/SlipServer.ps.Z
+ various other public archives,

or by e-mailing me directly.

Table of Contents

I. WHAT IS SLIP/PPP

SLIP stands for Serial Line IP (Internet Protocol), and is a method of sending data normally traversing ethernet and broadband lines over serial (i.e. telephone) lines. In other words, when your SLIP connection is established (see below), you are an interactive part of the Internet (or whichever net you have dialed into). SLIP has been described as a hack, and while that may be true, it is supported by many vendors, and provides (at least on the NeXT) performance and ease of use equalling or exceeding PPP. PPP stands for Point to Point Protocol, and is a bit more complex. PPP supports multiple protocols, while SLIP supports only IP. PPP includes many options which can enhance it's function, SLIP may be regarded as 'bare bones.' PPP is an approved standard (RFC-1331, W. Simpson, "The Point-to-Point Protocol (PPP) for the Transmission of Multi-protocol Datagrams over Point-to-Point Links", May 1992.) which was created to take the place of the 'nonstandard' SLIP, considered an interim solution (RFC 1055, J. Romkey, "Nonstandard for transmission of IP datagrams over serial lines: SLIP", June 1988.). Both can be used in similar situations - either dedicated (leased) lines, or dialup lines. If you need more information on the evolution and comparison of SLIP and PPP vs. other technologies, I recommend reading John Landwehr's paper "Transparent Remote Network Connectivity: Internet Protocol Over Serial Lines." (see Other Resources)

There are two versions of SLIP: SLIP and CSLIP. CSLIP is SLIP with Van Jacobson header compression (typically yielding a reduction in TCP/IP header data size from ca. 40 bytes to 3 for telnet/ftp, 5 for general use), giving added performance because of reduced data traversing the line. The added performance is most noticeable in interactive use. (RFC 1144, V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links", February 1990.)

II. WHAT YOU NEED

Lets start with the obvious. You need two computers, one of which must be correctly configured and connected to a functioning network, ideally an Internet site if you want to be on the Internet.

You will also need two modems. Not just any modems - high speed modems. According to the popular definition of 'high speed,' this implies 9600 baud or greater. If you want to do more than a simple telnet or rlogin or ftp you want a fast modem. My definition of fast starts at 14.4 baud. The faster things go, the more you can do, and the quicker things happen. Before you start dreaming of NFS and similar things, stop, and be thankful for good old telnet, rlogin, and ftp. Along with the two modems and computers you will need appropriate cables and serial ports. On the NeXT this means you have to have a NeXT specific serial cable (no Macintosh cables!), one that supports hardware flow control. If you (still) have a 030 board you cannot use flow control, nor can you use a 040 cable. You can still try the configuration and see if things work anyway. SUN's have been known to have problems with their standard serial ports, the hardware flow control is broken. The modem can request that the machine slow down transmission, but the machine cannot request the same of the modem, with the result that the modem stuffs the serial port faster than the port can service the data, causing packets to be dropped. This gives an error "`zs0: silo overflow`" at speeds as low as 19.2. Hardware flow control seems to function properly with add-on serial boards, and a patch for the standard serial ports appears to be in the works. In the meantime, SLIP may be run on SUNS, but it is recommended that the machine not be used for much else if serial port errors are to be avoided. Jonathan Hardwick <jch@cs.cmu.edu> notes, "You can reduce the problem by not trying to do much else if the serial port is in use; a humble Sun3/50 running cslip-2.6 (see later) at 38.4 can go a whole day without silo overflows, as long as it's just being used as an glorified X terminal."

My modem is a ZyXEL, and the servers modem is a Hayes, therefore the configurations listed will apply to these make and model modems. If your modem is different, refer to your reference guide (you have a manual, right...?) to enter the equivalent commands.

Thirdly, you need some kind soul, philanthropic university, or commercial service provider to allow you to dial in, and to assign your machine it's very own IP number and machine name. Without these two things you won't get too far...

Lastly, you need software to configure and control the SLIP interface at both ends of the connection. My configuration uses Louis Mamakos' public domain package: TransSys Dialup-IP, version 19920904A. Although this package is running at both ends, in theory it should work with other SLIP programs as well. Personally I haven't tested it.

I'm assuming you have a NeXT. If you have a Macintosh or PC, you'll need to get appropriate SLIP/PPP software from an archive, a friend, or a commercial supplier. If you have a SUN you can try cslip-2.6 (SunOS 4.1.1 and earlier), partially authored by Van Jacobsen. SLIP will be included in the next release of Solaris 2. If you have a DEC you're out of luck... (just kidding). HP includes SLIP in their software distributions.

III. WHERE TO GET IT

Publicly available (PD) software for SLIP and PPP are available at a variety of places (archives) for different hardware/OS combinations. Please consult the list under “Other Networking Programs” for a starting point as to where you can obtain a SLIP or PPP implementation for your machine. Commercial programs are also available, and have the added advantage of support. Sometimes this far outweighs the price. Consider it if you are dependant upon a reliable and trouble free SLIP/PPP connection.

IV. HOW TO SET IT UP

There are many steps needed in setting up SLIP, and I will try to break things down as simply as possible.

Software

The first thing to do is to install the software, which thanks to Mr. Mamakos is childs-play. Click twice on the SLIP_920904-A.tar file. WorkspaceManager will automagically bring up an Inspector panel. Click on Unarchive (Return) on the Inspector panel. If you have installed Opener.app or have specified a different default App for files with *.tar extensions, you will have to follow that application's normal untarring routine. If you prefer to use a terminal, give the following command (what you type is in **bold**):

`myhost%` **`tar -xvf SLIP_920904-A.tar`**

When you look in the folder again you will have an Installer package (SLIP_920904.pkg). Before you do anything more, you need to check to see that you are logged in as root the SLIP package cannot properly be installed otherwise. Log in as root, and click two times on the Installer package. (you can also su to root and start Installer.app from the command line if you want). The Installer panel should come up click “Install” and accept the default path as O.K. (/usr/dialupip). Installation is quick and painless. When installation is finished, quit Installer.app, logout of the root account, login as yourself, and double click on /usr/dialupip/Documentation.bshlf. Click on List Titles, and double click on Manual.wn. Read it. Read it once more. (Note to users who aren't installing on NeXTs - Louis Mamakos TransSys Dialup-IP for the NeXT is a wonderful package, and should be downloaded and checked out by all SLIP/PPP users just for the documentation. Better yet, find a friend with a NeXT and read it there. Mr. Mamakos' documentation is excellent, and the dist includes man files in standard man troff format for most of the files in the package, as well as extensive installation instructions. Recommended reading!)

If you have installed the TransSys Dialup-IP package previously, there is no danger in installing it again - none of your configuration files will be overwritten. If you are unsure of your first installation, might as well do it once more, just to be 100% sure everything is there.

If you are not installing on a NeXT, you will need to download the SLIP/PPP software from a ftp archive site or get it from a friend. Uncompress/unarchive it as necessary, and install it in the proper location (I assume that other distributions include at least a README file with installation instructions) on your machine. Normally this will be the machine connected to the Internet and with a modem properly attached - i.e. the server. PPP allows the software to be installed on machines which are not physically connected to a modem/modem pool by routing the incoming call over the net to the server with the PPP software (this is called tunneling). I am not aware if this is possible with SLIP. For installation of CSLIP-2.6 on SUNS, the following installation notes were gratefully provided by Jonathan Hardwick:

In general, the README file is enough to get you going. Here are a couple of notes.

- SunOS 4.1.1 (at least) has TCP code similar to that in 4.3-tahoe BSD, so there's no need to throw it away as suggested in section (a) of the README.
- The idea in section (b) seems to be to set the default TCP buffer sizes to approximately twice the MTU of the connection (which is set by default to 552 in if_sl.c).
- If you don't have access to kernel sources to make the retransmit timer modification in section (c), you can use adb to directly patch netinet/tcp_input.o. The modification doesn't seem to make *that* much of a difference, so if you don't know adb and can't find a guru, don't worry.
- The hardest part of the installation is getting the /etc/remote and login.script.unix files right. However, these are only needed if/when you want to automate the process of bringing up the slip slink (so that just typing e.g. “tip -s slip” will dial, establish a connection, and bring up SLIP). You can use “slattach” from the tools subdirectory to check that SLIP itself is working.
- If you don't need to set up your machine to accept dial-in SLIP, you can omit step (6).

Great, now all the software for SLIP is in place. Please note that for the changes made during configuration, you will need root access. If you don't have root access you shouldn't be doing this.

a) Software Configuration

[I'll assume that your software is properly configured on the client side...]
For the server, there are fewer files which need to be edited (I'm assuming that the server *will not* be doing dial out). Please bear in mind that the files mentioned are parts of the TransSys Dialup-IP package, although the same information has to be configured for any SLIP implementation.

First (as root), edit */usr/dialupip/config/config.slip*. You need to edit the following lines:

SLIP0LOCAL={your (i.e. the servers) slip interface IP address here}
SLIP0REMOTE={the clients (i.e. who shall be calling in) slip interface IP address here}
SLIP0NETMASK=255.255.255.0 {shouldn't need to be changed}
SLIP0CONFIG={SLIP if you are using the PD package without header compression,

CSLIP if you have purchased a license to header compression}
SLIP0DEFAULT=NO {I assume you *do not* want default traffic going out over the servers serial port... For a *client* not on any local network, this should be set to YES}

The file includes rather clear instructions as to what each of the parameters mean, in case the above is unclear or doesn't suit your installation. It should work for most.

Repeat the above information as necessary, substituting the '0' in SLIP0 for the interface number. The SLIP package is setup for two SLIP ports, if you want more (up to 10 allowed), you need to replace the existing *cslip_reloc* kernel driver with *cslip_reloc10* (rename *cslip_reloc* to *cslip_reloc.orig* and link/copy *cslip_reloc10* to *cslip_reloc*). On other installations using other software packages, please consult your documentation to see how many SLIP ports are supported, and how they should be configured. For special set ups involving dynamically assigned IP addresses at login time, see Advanced Configuration. (Have you installed SLIP/PPP on a SUN? DEC? HP? Write me!)

An example follows:

Client (home) machine has IP number 127.42.100.3 for interface slip0
Server machine has IP number 127.42.100.1 for interface slip0
The servers /usr/dialupip/config/config.slip should have the following lines (among others):

SLIP0LOCAL=127.42.100.1
SLIP0REMOTE=127.42.100.3

The client should look the opposite (i.e. LOCAL refers to the *local* machine - home machine in client case, server in server case)

Next, make sure that */usr/dialupip/config/diald.conf* exists. I deleted it thinking it wasn't needed since the server wouldn't be doing dial out. I was wrong. All the lines can be commented out, but the tcldial daemon has to find it or SLIP won't start up properly.

Make a copy of */usr/dialupip/bin/dudisc*, and name it */usr/dialupip/bin/dudisc_slip0* (i.e. the same as the interface name, slip0 for the first SLIP interface, slip1 for the second, etc). Use chmod to make it SUID.

Use nu or UserManager to create a new user account. The name is unimportant, the login shell should be set to /usr/dialupip/bin/dudisc_slip0 (or the appropriate SLIP interface if you have several). When the client SLIP program makes a connection, it will log in with this name, dudisc_slip0 will be executed and will establish the SLIP connection between machines, transparently sending packets between the two. Include a .hushlogin file to avoid stray data getting mixed up in the connection.

Ensure that /usr/dialupip/config/keyfile exists. This should be either a link to or a copy of either keyfile.slip (for standard PD SLIP) or keyfile.demo (for demo of CSLIP). It must *not* be touched or modified in any way shape or form, otherwise SLIP will not function at all. (This is specific to the TransSys package).

b) Network Configuration - Server

[Again, it is assumed that the client is properly set up]
Here there are several points which are important to remember. First, the SLIP-server is acting as a gateway. A gateway is a machine (device?) that has more than one interface configured, and is used to transfer network traffic from one interface (network) to another. Therefor it has to know about the interfaces available, and it has to know how to send traffic (packets) from one location to another. It does this with help of routing tables maintained in a database. On the NeXT this is in NetInfo, optionally in (). So, if you want Internet access, your server had better already be properly connected to the Internet at some point (I hope that this is obvious).

Each interface must be assigned it's own IP address, and may also be assigned it's own domain if desired. Normally the SLIP interface is assigned a separate subnet. Next, check the existing routing tables on the machine with netstat and ifconfig. If the configuration files are correct, and the machine has been rebooted (or optionally the scripts started by hand) there should be appropriate address entries added by the configuration script, routing traffic from one interface to the other. You should see clearly that the gateway for packets destined for client machines (client SLIP addresses, or an entire subnet) is the SLIP IP address of the server, and that there exist routes to and from other machines, gateways, etc on the ethernet (en0) interface. (see Troubleshooting for en example of how it should look).

If, after rebooting, none of the entries show up, check your /etc/rc.local file to see that the needed lines are present to start SLIP, that the line refers to files that exist (i.e. /usr/dialupip/config/rc.slip), and that the syntax is correct (best to copy it out of the TransSys installation manual). Try starting it by hand as root with /bin/sh. If netstat still doesn't see the addresses check your () file in (). You can also try modifying the routing tables directly (this assumes that the SLIP interface *is* present, but not the proper routing information) with route (see the man page) by adding the appropriate entries for the SLIP interface. If this works, there is most likely an error in the start-up or configuration scripts - check for misspellings or other subtle problems!

As the server is already on the Net, your current /etc/resolv.conf and /etc/hostconfig files should not need modification. NeXT's have a graphical hierarchical network information database known as NetInfo, so on a NeXT the /etc/hosts file doesn't require change. On the other hand, it's handy to have the information duplicated in /etc/hosts, and won't interfere with anything as long as NetInfo is running. For other types of machines all I can suggest (at the moment) is to read the documentation for adding network hosts.

Lastly, /etc/ttys needs to be modified to allow dialing in, and to start the login process when a connection is established. The tty which is used for the serial port dial-in on a

NeXT is ttyd[ab] without hardware flow control, and ttydf[ab] with. Use the ttydf[ab] (hardware flow control) entry if at all possible. The line in /etc/ttys which looks like this:

```
ttydfb"/usr/etc/getty D9600"      unknown    off
```

should be changed to this:

```
ttydfb"/usr/etc/getty D38400"     vt100      on
```

This instructs init to start getty with the D38400 speed entry (double check /etc/gettys that the appropriate speed entry exists) as a vt100 terminal type when the modem answers, and get the login name and start login. The above entry is for hardware flow control on serial port b, for port a change the last letter of ttydfb to an `a', for a 030 board or a cable not supporting hardware flow control, remove the `f' in ttydfb.

Other machines have other naming schemes for terminal devices, check the manual and enable the proper serial port at the proper speed.

SUMMARY

In essence, setting up a machine (either server or client) entails largely the same steps needed to enable any client or server as a network host. IP numbers must be assigned, machine names, domain names, and the appropriate files relating to the network tools in use at the installation - be it DNS, NIS, or NetInfo - must be modified accordingly. In addition, software directly affecting the SLIP interface (as most machines come without a configured SLIP interface) must be configured to create and maintain the necessary interface(s), as well as proper configuration of one or more modems.

c) Modem Configuration

Modem configuration can be tricky business - “Hayes compatible” means less and less each day :-(

Some things you need to configure the modem for:

- 8 bits, no parity
- RTS (hardware) flow control, if your server supports it. Remember to reference the proper device file in /dev to use the hardware flow control! (i.e. /dev/cufa for serial port a instead of /dev/cua, etc)
- Follow DTR

Generally speaking, you want the modem to pass the data as quickly as possible with a minimum of modification - no xon/xoff, no error control, etc, because the IP protocol takes care of all this anyway. Some users recommend programming the modem to not accept retrain requests, while others feel that this is better than risking a dropped connection (in the event of a very bad line). If the modems support both fallback and fallforward, it might be good idea to enable retrain, therefor enabling optimal throughput as line conditions change. Since there are so many different modems on the market, I refer the reader to the modems manual for the actual setup string. If you encounter difficulties with the modems (quite probable, I'm afraid), check and double-check the connection and setup strings. Try using tip or kermit to make a connection. Check that /etc/gettytab includes an entry for the proper speed you have configured the modems for, and that /etc/ttys refers to the proper device, program and terminal speed. Please please please *do not* enable root login on any device that allows remote calling, you are merely inviting trouble.

Here is a list of configurable modem setup options for a ZyXEL 1496E, as included in the TransSys package:

```
# configure modem with proper parameters
# S2=128 - turn off escape into command mode
# &K4- V.42/V.42bis or MNP4/MNP5
# &N0- auto-negotiate highest possible link rate
# M0- speaker off
# M1- speaker on until connect
# V1- verbose responses
# Q0- display responses
# X5
# &C1- CD tracks carrier presence
# &D3- hang up and reset to profile 0 when DTR dropped
# &H3- hardware (RTS/CTS) flow control
# &J0- single phone line RJ11 jack
# &L0- normal phone line (not leased)
# &M0- async mode
# &R1- ignore RTS, assume always on
# &S0- DSR override, assume always on
# N1- ring volume
# *Q0- no response to poor signal quality
```

So a sample configuration string might look like this:
AT S2=128 &K4 &N0 M1 V1 Q0 X5 &C1 &D3 &H3 &J0 &L0 &M0 &R1 &S0 N1 *Q0

d) Advanced Configuration

Dynamic IP address allocation

Dynamically assigning IP addresses is usual at larger installations such as universities and commercial service providers, as the SLIP service is open to many users who may share a server machine or port on a terminal server. As I haven't done this, I have no idea how to set it up. All I can say is “see the manual.” If anyone has info on this, I would appreciate hearing from you.

e) Miscellaneous Configuration

Mail

For mail to function properly the sendmail.cf file *may* need to be modified, on the client end of things. On the server, as long as the client is registered in the network database as a valid client everything should work. I'm no mail guru - find one if you need help. One problem that can occur is that some mail will bounce, as the SLIP connection isn't up all the time. Mail tries to deliver mail and finds that the destination is down because the SLIP link is down/inactive. Most mail programs try for three days before giving up, and the mail admin can configure the frequency that mail delivery is attempted. An alternative is to use a POP mail client to fetch mail from the mail server. NuPOP on the PC is a relatively convenient and stable program to handle mail stored on a remote server. There are also POP implementations for other platforms (Eudora(?) for the Mac, plus POP programs for NeXT and SUN). If the connection is often up, or used over a leased line, mail should function as expected for any client on the network.

News

For clients to be able to read Usenet News, they need to be included in the news servers news_clients file. Setup should be just the same for “normal” network clients. It is usually much faster for clients to simply use telnet or rlogin to a server and run rn/tin/trn/gnus in a terminal window instead of setting up a newsfeed to the SLIP client. Windows based (i.e. NewsGrazer, I forgot the name of the X windows news reader) newsreaders work, but on a slow (9600) link it can take a long time compared to a terminal based approach.

V. TROUBLESHOOTING

Follows is a list of common problems, their symptoms, and probable causes and solutions.

SLIP device not installed

Upon booting, you should see boot messages indicating installation and initialization of the SLIP interface. If you don't, check /etc/rc.local (or appropriate addition to /etc/rc.-boot or /etc/rc) for the invocation of the rc.slip script which installs the SLIP interface at boot time. One easy way to check if the SLIP interface is installed and enabled is to check the current configured interfaces with netstat, i.e.:

```
myhost% netstat -i
Name   Mtu  Network     Address    Ipkts Ierrs Opkts   Oerrs   Coll
en0    1500 next-defaul slp-gw     37    0     37      0       0
lo0    1536 loopback    localhost  7155  0     7155    0       0
slip0  1006 127.42.100  127.42.10024176  153   23831   0       0
slip1  1006 none        none       0     0     0       0       0
en0    1500 none        none       37    0     37      0       0
```

One may also try using ifconfig for interface configuration information:

```
myhost% ifconfig slip0
slip0: flags=11<UP,POINTOPOINT>
        inet 127.42.100.1 --> 127.42.100.3 netmask ffffff00
```

Routing tables not installed/correct

Is the proper/appropriate routing information in the routing tables? Again, use netstat to confirm that the SLIP interface is active and is routed properly:

```
myhost% netstat -r
Routing tables
Destination     Gateway          Flags    Refs     Use  Interface
localhost        localhost         UH          3     1323  lo0
127.42.100.1     localhost         UH          0        0  lo0
127.42.100.3     127.42.100.1      UG          0    23871  slip0
+ all of the usual ethernet (interface en0) entries...

myhost% netstat -rn
Routing tables
Destination     Gateway          Flags    Refs     Use  Interface
127.0.0.1        127.0.0.1         UH          3     1323  lo0
127.42.100.3     127.0.0.1         UH          0        0  lo0
default          127.42.100.1      UG          0    23856  slip0
+ all of the usual ethernet (interface en0) entries...
```

If you don't have entries for a SLIP interface, check your configuration files - for the Trans-Sys package this means /usr/dialupip/config/rc.slip, config.slip and /etc/rc.local. Watch the console at boot time, you should see the rc.local script being executed, which starts the rc.slip script going. It echoes each interface configured and invokes ifconfig, route, and duioctl to configure the interface according to the configuration information in config.slip. If you have accidentally configured identical addresses for local and remote (not hard to do :-), it will *remove* the routing information for that interface.

Modem setup wrong

The modem should answer on the first ring, and establish a connection at the highest supported speed without fallback, error correction, etc. If you are having troubles with the modem after it answers, check by logging in as a regular user (i.e. don't use SLIP autodialing, but dial in from another machine and attempt to log in). If you get garbage instead of a login string, it could be a speed problem. Ensure on both ends that the modems are set up identically for speed, parity, etc. If you don't get any prompt, or if the modem doesn't answer despite setting the auto answer bit, check your cables. Can you use the modem to dial out? Try exchanging with other cables that you know work.

VI. FUTURE ADDITIONS

Other (vendor specific) server configurations (SUN, Hewlett Packard, etc).
Cisco configuration for SLIP and PPP.
Removal of the sarcastic comments relative to non-NeXT platforms.
Editing to improve readability and language usage.

VII. OTHER RESOURCES

Usenet News: comp.dcom.modems
Usenet News: comp.protocol.ppp
Usenet News: comp.sys.next.sysadmin/hardware/programmer

Landwehr, J. (1992). *Transparent Remote Network Connectivity: Internet Protocol Over Serial Lines*.
Available via anon ftp to: sonata.cc.purdue.edu /pub/next/docs/SLIP_PPP_Paper.ps.Z

Rempe, Glenn D. (1993). *Connecting to the Internet*.
Available via:

RFCs
Available via anon ftp to: nic.ddn.mil
Available via electronic mailing service: service@nic.ddn.mil

VIII. NETWORKING PROGRAMS

KA9Q	(MS-DOS)	anon ftp: ucsd.edu
Merit PPP collection	(PPP for various platforms)	anon ftp: merit.edu
Morning Star PPP	(PPP for various platforms)	e-mail: marketing@morningstar.com
Ohio PPP collection	(PPP for various platforms)	anon ftp: archive.cis.ohio-state.edu
Marble Teleconnect	(SLIP for NeXT)	?
TransSys Dialup-IP	(SLIP/CSLIP for NeXT)	anon ftp: sonata.cc.purdue.edu
CSLIP-2.6	(CSLIP for SUN)	anon ftp: ftp.ee.lbl.gov

IX. ACKNOWLEDGEMENTS
Thanks to the following for providing invaluable help in the form of private e-mail, Usenet News, FAQ files, and various articles and papers (see the Other Resources section):

Kjell Rune Abbedissen	<abbedissen@edb.uib.no>
Pierce Cantrell	<pierce@cantrell.tamu.edu>
Karl Denninger	<karl@ddsw1.mcs.com>
Dwight Everhart	<biff@biff.jpunix.com>
Mark L. Fausett	<mfausett@LOCUTUS.BBN.COM>
Jeff Hallgren	<jhall@tahiti.umhc.umn.edu>
Jonathan Hardwick	<jch@cs.cmu.edu>
Alan Hinds	<U32472%UICVM.bitnet@livid.uib.no>
Pete Holsberg	<pjh@mccc.edu>

Kenneth Høstland	<hostland@edb.uib.no>
Hans Morten Kind	<kind@edb.uib.no>
Edwin Kremer	<Edwin.Kremer@NOC.cs.ruu.nl>
Caroline Lambert	<caroline@stp.Stanford.EDU>
John Landwehr	<jland@next.com>
Markus Lankeit	<markus@u.washington.edu>
James S MacKinnon	<jmack@skye.Phys.UAlberta.Ca>
Louis A. Mamakos	<louie@transsys.com>
Dr. Andreas Mueller	<afm@mathi.uni-heidelberg.de>
Glenn D.Rempe	<glenn@iridium.digex.net>
Walter Roberson	<roberson@hamer.ibd.nrc.ca>
Bob Sutterfield	<bob@MorningStar.Com>

...and certainly others whom have asked not to be mentioned or I have lost their address (sorry!).

All copyrights and trademarks are property of the respective companies (NeXT, SUN, IBM, etc...) named. Inclusion of a specific product or manufacturer is coincidental and in no way represents an endorsement of said product.

And in case anyone wonders, no - I am not a computer science major ;-). I am a student in the graduate programme of psychology at the University of Bergen, Norway. I may be contacted at the following address:

Thor Legvold
NorNeXT User Group
Postboks 749
5001 Bergen
Tlf/Fax:+47 55 90 21 66
email: Legvold@edb.uib.no (all mail accepted)
edmtl@fiol.uib.no (NeXTmail accepted)
thor@sushi.uib.no (fan mail accepted)