NAME

serial-ip − Serial IP (SLIP or PPP) setup

DESCRIPTION

−−−−−−−

Note: This text and the serial IP code is not finished. Code needs to be added to **nonamed** to allow it to be used both with and without a connection to the Internet, and by now there is a PPP program for standard Minix "out there" that will change everything that is said in this text. So much to do, so little time...

−−−−−−−

This manual page describes the Minix network setup to use serial line IP. The serial IP protocol used can either be the older SLIP by means of the **slip**(8) program, or PPP (Point-to-Point Protocol), the newer and better serial IP protocol implemented by the **ppp**(8) program. Alas standard Minix only supports SLIP.

In the following text all descriptions and examples will name SLIP or the **slip** program, but one may just as well read PPP or **ppp**. Where necessary the differences will be noted.

A typical use of the **slip** program is like this:

> **slip /dev/psip2 </dev/tty01 >/dev/tty01**

The argument of the program, the **/dev/psip2** device, is one of the so-called "Pseudo IP" devices that the Minix TCP/IP driver **inet**(8) offers to implement a virtual network on. On an ethernet IP packets are received or transmitted by the ethernet card, but packets on a pseudo IP network are channeled back to or received from a program running in user space, such as **slip**. Standard input and output are used by **slip** to exchange packets with another SLIP implementation. This is normally through an RS-232 serial line like the second serial line **/dev/tty01** as used in the example above.

If we look at the flow of data over normal ethernet then this is what a TCP connection between two Minix machines, **telnet** for instance, looks like:

```
   [telnet]
      |
  /dev/tcp0
      |
     inet
      |
  [ethernet]
      |
     inet
      |
  /dev/tcp0
      |
 [in.telnetd]
```

One-half (!) of a SLIP connection would look like this:

```
   [telnet]
      |
  /dev/tcp2
      |
     inet
      |
  /dev/psip2
      |
     slip
      |
[serial line]
     ...
```

Configuration for a SLIP network only

It is important to know that as far as **inet** is concerned the pseudo IP network is just another network, nothing special. So you have to convince **inet** that it has to send packets out over that network. One does this by setting a default route that makes **inet** believe that there is a router somewhere on the pseudo-IP network.

Assume your machine has been given the IP address **192.168.0.13** by your service provider. Let's choose another address on that network, **192.168.0.1** for instance. (You can use the address of the SLIP gateway if you want to make it look pretty, but it doesn't really matter, anything "out there" is ok.) To make Minix aware of the situation you have to configure the pseudo IP network. For Minix-vmd you need to look for the **if-then-else-fi** code in **/usr/etc/rc** that tests if **/etc/rc.net** should be run. Copy the lines in the **else** clause that starts network daemons to **/etc/rc.net** and add the following lines to make it look like this:

```
# My SLIP interface address.
ifconfig -h 192.168.0.13 -n 255.255.255.0

# Standard network daemons.
daemonize rarpd $named irdpd rip inetd

# Default route to the outside world.
add_route -g 192.168.0.1
```

For standard Minix one has to edit **/etc/rc** instead at the point of the XXX comments. The **ifconfig** goes at the first XXX, the **add_route** at the second XXX. The result is conceptually the same as the example above. The important thing is the order: Configuration, Daemons, Routes. (First give addresses to the networks, let the daemons meditate over the results and possibly configure more networks (rarpd), then add routes to the configured networks.)

Just one thing left to do. The system uses the first ethernet network (**eth0**, **ip0**, **tcp0**, and **udp0**) as the default network. With the program **netdefault**(8) you have to change the links to the default devices (**eth**/**psip**, **ip**, **tcp**, and **udp**) to point to the first pseudo IP network (**psip2**, **ip2**, **tcp2**, and **udp2**):

netdefault psip2

In **/etc/hosts** list at least **localhost** and the name of your machine with its SLIP address. This way your machine will boot and know its own name. Now you need to find a way to let your system know the addresses of other machines. There are three ways:

List the names and addresses of any other machine you wish to talk to in **/etc/hosts**. Drawback: This will quickly become a pretty long list.

Create an **/etc/resolv.conf** that lists a nameserver at your ISP and **127.0.0.1** (localhost). Drawback: With the SLIP link down it takes 5 to 10 seconds for a name lookup to time out on the remote name server before the local name server is tried.

Install the above **/etc/resolv.conf** when **slip** is started, and remove it when **slip** exits. Drawback: Long running programs only read **/etc/resolv.conf** at startup, so they don't notice it changing.

Run a real Internet name daemon from the **named** package. Drawback: Nontrivial to set up.

Configuration for a SLIP - Ethernet router (simple case)

XXX

Configuration for a SLIP - Ethernet router (complex case)

XXX

FILES

/dev/psip* Pseudo-IP devices for use by **slip** and **ppp**.

SEE ALSO

boot(8), **inet**(8), **netdefault**(8), **term**(1), **chat**(1).

BUGS

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)