

Configuring Linux PPP Dial-in Server

This tech note provides straightforward instructions on how to configure a Red Hat Linux PPP server. Implementation requires one modem to be configured for each dialup serial port on the server. The solution converts the Linux server into a single-hop IP forwarding router for the PPP clients, but not full, multi-subnet access unless the Linux server is configured to be a router.

Solution

Requirements:

- A. An instance of Red Hat Linux 6.1 (only 6.1 was tested), but any hardware platform should work, as long as `pppd` and `mgetty` are installed correctly. This configuration was tested on the following hardware: 486 laptop with 24 Mbytes of RAM and Pentium PC with 32 Mbytes of RAM.
- B. Network interface combinations for the PPP server that may include:
 - PCMCIA Ethernet card and external modem plugged in to the serial port (for laptops)
 - PCMCIA Ethernet card and PCMCIA modem card (do not use the combo card for laptops)
 - PCI Ethernet NIC and ISA interface Microsoft Windows 3.1 compatible modem card or external Hayes-compatible POTS modem (for desktop PCs)

Configuration

- A. Uncomment the `AutoPPP` line in the `/etc/mgetty+sendfax/login.config` file.
- B. Adjust the baud rate to the desired throughput in the `/etc/mgetty+sendfax/mgetty.config` file. The default value is 38400 but it can be set to 57600 for 56K modems.
- C. Add the following line to the bottom of the `/etc/inittab` file:

```
ppp0:345:respawn:/sbin/mgetty -x 3 ttyS0
```

Change `ttyS0` to the port used by the modem. You can query the ports by using the `setserial -g /dev/ttyS*` command. If you are debugging, you may also set the reporting to be `-x 9`, which is extremely verbose. The log file is usually in `/var/log/mgetty.log.ttyS?`.

- D. Put the following options in the `/etc/ppp/options` file:

```
-detach
asyncmap 0
netmask 255.255.255.0
proxyarp
lock
crtscts
modem
noauth
```

Set the `netmask` to your network settings. The keyword `proxyarp` ensures that the PPP client can see the rest of the LAN by telling the Linux server to add an entry to the ARP table and to add the proxy itself as the router for the LAN traffic destined to the PPP client.

- E. Create or edit the `/etc/ppp/options.ttyS?` file, where `ttyS?` is the serial port the modem is connected to. Add host names separated by colons as shown below. These names must exist in the `/etc/host` file, and you need to create one file for each of the serial line.

```
pppserver:pppclient
```

- F. Add the server and client names and IP addresses to the `/etc/host` file if they don't already exist. Then, on the server, also use `linuxconf` to configure the `eth0` port to be the IP address of the server host name. The following is a sample `/etc/host` file:

```
127.0.0.1 local host
129.200.9.2 pppserver loghost
129.200.9.101 pppclient
```

- G. Grant `suid` root access to the `pppd` by typing: `chmod u+s /usr/sbin/pppd`. This allows the `pppd` to have access to the protected network resources.
- H. Force the Linux runtime system to re-read the `/etc/inittab` file by typing: `init q`.
- I. To set the IP forwarding feature so that the server routes the client requests to the rest of the LAN, type:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

By default, the Red Hat 6.1 kernel is compiled with `ip_forward` and can be activated using the command above. This feature may not be turned on by default, in which case you should rebuild the kernel with flags set to `ip_foward` in the `/usr/src/linux` directory. The `ip_forward` and `proxyarp` features in Linux allow it to easily route PPP traffic to the LAN. However, if the network has multiple subnets, then you should configure the Linux box to become a router by using `linuxconf`.

You should now be able to test the system using an analog phone line plugged into the modem that is connected to the PPP server. In some instances, like a trade show, phone lines may be unpredictable and it may be worthwhile to purchase a phone line simulator. Multiple line simulators are also available.

Notes:

Modems

The `mgetty` PPP login facility for Linux is quite sophisticated and actually handles the answering of the phone when the modem rings. Unlike the Solaris™ PPP server configuration, the Linux `mgetty` does not recommend configuring the modem to auto-answer after `N` number of rings (i.e. `ATSO=N`). Allowing for the system to answer the calls rather than the modem can reduce high costs on toll free number PPP dialup systems because if the system does not receive a valid PPP login, the connection is terminated. However, if the modem is auto-answering, it may occasionally hold the line open for some time before timing out and notifying the host server of the failed connection.

For `mgetty`, you can usually accept the factory modem settings and the standard AT&F initialization and ATZ reset strings should work fine. In some cases, if the modem has been used for other purposes and the defaults were changed, you should reset the modem to a more acceptable profile and save the setting. On

<http://solaris.java.sun.com/technotes/linuxppp.html>

Linux, you can use the `minicom` utility or `dip`. On Solaris you can use `tip` or `cu` to configure the modem. Below is a list of AT commands to set the modem. Some of the settings are factory defaults and are used to handle handshake, echo, and speed negotiation.

<code>at&f</code>	Resets active defaults to factory defaults for a clean slate
<code>ate1</code>	Echoes characters in command state
<code>atq2</code>	Returns no result codes when answering
<code>atw1</code>	Returns negotiation progress messages
<code>ats0=1</code>	Automatically answers after 1 ring
<code>ats37=15</code>	Sets maximum modulation speed to 28.8K
<code>ats11=60</code>	Speeds up dialing by shortening inter-DTMF delays
<code>ats70=32</code>	Increases maximum number of frame retransmissions (default is 10)
<code>at&k3</code>	Uses hardware flow control (CTS/RTS) instead of XON/XOFF
<code>at&q5</code>	Sets error control mode
<code>at&d3</code>	Monitors DTR signal and hangup and turns system back on if system shuts off
<code>at&s0</code>	Asserts DSR is always on
<code>at&w</code>	Saves custom settings in the user profile
Note: If <code>at&c1</code> ends the program, restart it and issue the following commands.	
<code>at&c1</code>	Tracks presence of carrier detect signal
<code>at&w</code>	Saves custom settings in the user profile

Power Cycle/ Reboot

On occasion, if the system has been rebooted, the `/proc/sys/net/ipv4/ip_foward` flag will be reset to zero. Repeating [Step 1](#) usually solves the problem.

June 2000