HOWTO.3

PPP Configuration setup for serial TCP/IP phone or wireless connections between a local and remote Linux Box .

A. Phone connection

Note: It is a good idea to set up a basic dial-up connection and check that the serial link and mgetty program are functioning properly. Follow steps in *HOWTO.2* and make sure the remote computer is sending a clean login prompt to your screen.

Setup procedure for PPP server (dial-in):

Step 1:

```
File:           /etc/inittab
Line to add:    d1:2345:respawn:/sbin/mgetty -D /dev/ttyS#
```

Where # is the number of the port which will be monitored by the mgetty process. This port should be dedicated to incoming calls and not be used to interface with any other devices (such as a UPS serial connection).

Note that the -D option is important as it forces mgetty to treat the modem as a DATA modem. No fax initialization is attempted.

Re-boot the machine and the mgetty process will be started automatically by the inittab master process.

Step 2:
Using the user configuration panel, create a new user ppp. Set the password, user information, and create the /home/ppp directory. Do not make any changes to the default shell at this time. Close the configuration panel and activate the changes.

Edit the /etc/passwd file and replace the default shell with /usr/sbin/pppd. This is not a recognized shell by the user configuration control panel and this is why the /etc/passwd file has to be edited separately. After the ppp login authentication process has completed, the remote server will start the pppd process automatically instead of the normal shell.

Note: Do *not* edit the /etc/passwd file to create the ppp account. Edit the file only to modify the shell, after the account has been created through the control panel.

Step 3:
Create the file .ppprc in /home/ppp and add the following lines

```
-detach
modem
lock
crtscts
proxyarp
localhostIP:remotehostIP
```

Note: If the PPP server is networked, then it should already have an IP address, and you must replace the string localhostIP with it. If it is going to be used as a stand-alone machine, then make up a dummy IP address. Replace the string remotehostIP with the IP address of the PPP client. If the PPP client calling the server already has a static IP address on some remote network, make sure the dummy IP address assigned to the server will not conflict with another valid IP on the PPP client's network. The server may

allow clients with different IP addresses to dial in by adding more lines of the form `localhostIP:remotehostIP` to the .`pppr`c file.
If you wish to enable any client to establish a PPP connection with the server, do not include the address of the PPP client in line 6 of the `.ppprc` file. Only use the string `localhostIP:`

Step 4:
File: /etc/rc.d/rc.local
Line 1 to add: `chmod u+s /usr/sbin/pppd`
Line 2 to add: `chmod a+rw /dev/ttyS*`
This gives system permission to any logged user to run the `pppd` daemon and ensures that the device are accessible by everyone.

Step 5:
If your PPP server is a Linux box on the local Ethernet, and you want your standalone PPP client to be able to see machines behind the server (i.e. you can ping any valid IP address), you must enable IP forwarding by the server. Edit the file `/etc/sysconfig/network` and change the line that says FORWARD_IPV4=no, to FORWARD_IPV4=yes. This is absolutely essential for a seamless connection to the internet. The other parameter of importance is the `proxyarp` option above (which sets up a proxy ARP entry in the server's ARP table which says 'send all packets destined to the PPP client to me'. This is the easiest way to set up routing to a single PPP client.

Setup procedure for PPP client (dial-out):

Step 1:
Copy the chat scripts from /`usr/docs/ppp.SOMEVERSION/`scripts to the user directory. You may create a separate `ppp` directory for the scripts.

The only scripts that are needed for a standard `ppp` connection are `ppp-on, ppp-on-dialer`, and `ppp-off`. Make sure that all three files are executable by issuing the command `chmod a+x  <file>` for each file.

Edit the file `ppp-on` and make the required changes to the lines shown below:

```
TELEPHONE=telephone number of PPP server
ACCOUNT=ppp
PASSWORD=ppp12345
LOCAL_IP=xxx.xxx.xxx.xxx
REMOTE_IP=xxx.xxx.xxx.xxx
NETMASK=255.255.255.0


DIALER_SCRIPT=/ppp_scripts_directory/ppp-on-dialer

exec /usr/sbin/pppd debug lock modem crtcts /dev/ttyS# 19200 \
$LOCAL_IP:$REMOTE_IP \
netmask $NETMASK defaultroute connect $DIALER_SCRIPT
```

Notes:
The shell variable `REMOTE_IP` is the IP address of the dial-in PPP server. If the server was set up to allow any dial-in connections, then leave the IP address blank (i.e. `REMOTE_IP=`) in the connection setup parameters, and leave out the `$REMOTE_IP` variable from the command exec (i.e. `$LOCAL_IP:`).
The `defaultroute` parameter adds a default route to the client's routing system. If the PPP client is establishing a connection to a networked PPP server, and you want to be able to see machines beyond the

server, the IP address assigned to the client should belong to the same subnet as that of the server (since we are using a netmask of 255.255.255.0). Choose a number between 1 and 255 that is not already assigned to a machine on the server's subnet.

Step2:
Issue the command `chmod u+s /usr/sbin/pppd`
This gives system permission to any logged user to run the `pppd` daemon.

Step3:

Modify the `ppp-on-dialer` file to conform with the "chat" strings exchanged by the local modem – remote modem and computer. Each line of the dialer script consists of an "expect string" "send string" pair.

Note: The script below (default template) will work without modification for a U.S. Robotics Courier V. Everything modem connecting to a PPP server running RedHat Linux 5.2. Note that the "expect string" consists of the standard "login" string prompt sent by the remote computer. Keep in mind that if the client connects to a server such as an Internet Service provider, the "expect string" may be different. The remote server may be sending a "username" string instead of the "login" string". Other "expect strings" sent by the remote server may have to be inserted in the `ppp-on-dialer` script as well. The easiest way to determine which strings are sent by your provider is to use the Linux communication program `cu` to call the provider directly, and record the strings echoed to the screen by the remote computer, as well as the ones you have to type in, all the way to the password prompt.

```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
exec chat -v                                     \
      TIMEOUT           3                        \
      ABORT       '\nBUSY\r'                     \
      ABORT       '\nNO ANSWER\r'                \
      ABORT       '\nRINGING\r\n\r\nRINGING\r'   \
      ''          \rAT                           \
      'OK-+++\c-OK'     ATH0                     \
      TIMEOUT           30                       \
      OK                ATDT$TELEPHONE           \
      CONNECT           ''                       \
      ogin:--ogin:      $ACCOUNT                 \
      assword:          $PASSWORD
```

Initiating the connection:

From the PPP client, issue the command `./ppp-on` to invoke the script
The connection process can be monitored on the PPP client by opening another xterm and typing `tail -f /var/log/messages` to see all the diagnostic messages sent by the client's PPP daemon. On the server side, the same command can be used to capture the messages sent by its own daemon once it gets started. Additionally, `tail -f /var/log/mgetty.log/ttySx` can be invoked to check on the status of the serial connection itself.

To confirm that a valid PPP connection exists, type `ifconfig` on the client to see the *ppp0* network interface and relation information in addition to the *lo* (local host) network interface. Pinging the remote server shall instill further confidence that the connection has been made successfully.

Troubleshooting:

a).The remote server disconnects immediately after the password authentication process has completed.
-You didn't set system permission for the `pppd` daemon on the server. By default only root can start the `pppd`!
-The login and/or password string(s) sent by the client do(es) not match that (those) expected by the server. If login/password seem to match, make sure that you can log in as PPP by using the basic dial-up connection through the `cu` communication program. It could be that the server is pre-empting user logins for some reason (which will happen if the file `/etc/nologin` is found)

b) Link is established as seen by `the /var/log/messages` file on the client ; however `ifconfig` shows that a PPP connection does not really exist, or the server cannot be seen by pinging the client.
-IP addresses for the client and/or server do not match.

c) PPP fails before the modem even begins dialing.
-Make sure that the `ppp-dialer` script has *execute* permission for all.
-The dialer script is not configured properly.

d) Once the link is established the server is very slow in sending a prompt to the client following a telnet or ftp request.
-This will happen if the remote server is set up as a networked computer but the network to which it is attached is down. The prompt may take several minutes to be sent to the client. To remedy the situation, deactivate the eth0 interface on the server and disable boot-time activation if the server is not going to be connected to a networked for most of the time. A similar time-out may occur if there is a network configuration conflict between the server and the client.

e) Serial link is not 8-bit clean, or time-out sending configuration requests seen in the file /var/log/messages for the client
-Make sure that the `/usr/sbin/pppd` daemon is the default shell for the ppp user.

B. Wireless connection

Note: It is a good idea to set up a basic connection and check the serial link with the `mgetty` program running on the server. Follow steps in *HOWTO.2* for the wireless connection and make sure the remote computer is sending a clean login prompt to your screen.

Setup procedure for PPP server :

Once you have tested connectivity with the server running the mgetty daemon on the serial port, you can replace the mgetty program with the pppd daemon in the /etc/inittab file

File: `/etc/inittab`
Line to add:
```
d1:2345:respawn:/sbin/pppd –detach lock crtscts /dev/ttyS#
<LOCAL_IP>:<REMOTE_IP>  <speed>
```

`where`
`<LOCAL_IP>` is the IP address of the server.

<REMOTE_IP> is the IP address of the PPP client. If you wish to enable any client to establish a PPP connection with the server, leave this field blank.
<speed> is the connect speed desired.

Re-boot the machine and the pppd process will be started automatically by the inittab master process.

The defaultroute parameter adds a default route to the client's routing system. If the PPP client is establishing a connection to a networked PPP server, and you want to be able to see machines beyond the server, the IP address assigned to the client should belong to the same subnet as that of the server (since we are using a netmask of 255.255.255.0). Choose a number between 1 and 255 that is not already assigned to a machine on the server's subnet.

Where # is the number of the port which will be monitored by the mgetty process. This port should be dedicated to incoming calls and not be used to interface with any other devices (such as dial-out modem, or UPS serial connections).

Setup procedure for PPP client:

When you are ready to establish a PPP connection with the server (you can automate this with a cron job), issue the command
Once you have tested connectivity with the server running the mgetty daemon on the serial port, you can replace the mgetty program with the pppd daemon in the /etc/inittab file

```
/sbin/pppd –detach crtscts lock /dev/ttyS# <LOCAL_IP>:<REMOTE_IP>
<speed> &
```

This command should be run in the background and your connect speed should match the setting on the PPP server. You do not need to validate the connection using username/password pairs as for a dialup connection since you have physical control of both machines.

Initiating the connection:

As outlined above, as soon as pppd is initiated on the client, it will bring up the link and you have access to the standard TCP/IP application programs.
The connection process can be monitored on the PPP client by opening another xterm and typing `tail –f /var/log/messages` to see all the diagnostic messages sent by the client's PPP daemon. On the server side, the same command can be used to capture the messages sent by its own daemon once the connection is established.
To confirm that a valid PPP connection exists, type `ifconfig` on the client to see the *ppp0* network interface and relation information in addition to the *lo* (local host) network interface. Pinging the remote server shall instill further confidence that the connection has been made successfully.

Note:

The pppd command with the above options will bring up the link between two non-networked computers. No routing as been specified yet. If the PPP server is connected to a local network, you should add the command-line option proxyarp to pppd started by the inittab process. This option sets up a proxy ARP entry in the server's ARP table which says 'send all packets destined to the PPP client to me'. This is the easiest way to set up routing to a single PPP client. Furthermore, if you want your standalone PPP client to be able to see machines behind the server (i.e. you can ping any valid IP address), you must enable IP forwarding by the server. Edit the file /etc/sysconfig/network and change the line that says FORWARD_IPV4=no, to FORWARD_IPV4=yes. This is absolutely essential for a seamless connection to the internet.

On the client side, you must add the option `defaultroute` to the `pppd` command. The `defaultroute` parameter adds a default route to the client's routing system. Also, if the PPP client is establishing a connection to a networked PPP server, and you want to be able to see machines beyond the server, the IP address assigned to the client should belong to the same subnet as that of the server (since we are using a netmask of 255.255.255.0). Choose a number between 1 and 255 that is not already assigned to a machine on the server's subnet.

If the PPP client calling the server already has a static IP address on some remote network, make sure the dummy IP address assigned to the server will not conflict with another valid IP on the PPP client's network.