

– General Linux 1 –

Configure Modem and Sound cards [2]

(Linux Professional Institute Certification)

a

```
.~.      Prepared by: Andrew Eager  
/V\          Geoffrey Robertson  
// \\  
@._.@      geoffrey robertson  
           geoffrey@zip.com.au
```

```
$Id: gl1.101.3.slides.tex,v 1.3 2003/05/30 04:57:25 waratah Exp $
```

^aCopyright © 2002 Geoffrey Robertson. Permission is granted to make and distribute verbatim copies or modified versions of this document provided that this copyright notice and this permission notice are preserved on all copies under the terms of the GNU General Public License as published by the Free Software Foundation—either version 2 of the License or (at your option) any later version.

Configure Modem and Sound cards [2]

(1.1) 101 Hardware & Architecture [11]

1.101.1 Configure BIOS Fundamental Settings [3]

1.101.3 Configure Modem and Sound cards [2]

1.101.4 Setup SCSI devices [2]

1.101.5 Setup different PC expansion cards [2]

1.101.6 Configure Communication Devices [2]

1.101.7 Set Up USB Devices [2]

Configure Modem and Sound card []

Objective

Ensure devices meet compatibility requirements (particularly that the modem is NOT a win-modem), verify that both the modem and sound card are using unique and correct IRQ's, I/O, and DMA addresses, if the sound card is PnP install and run sndconfig and isapnp, configure modem for outbound dial-up, configure modem for outbound PPP — SLIP — CSLIP connection, set serial port for 115.2 Kbps

Configure Modem and Sound card []

Key files, terms, and utilities

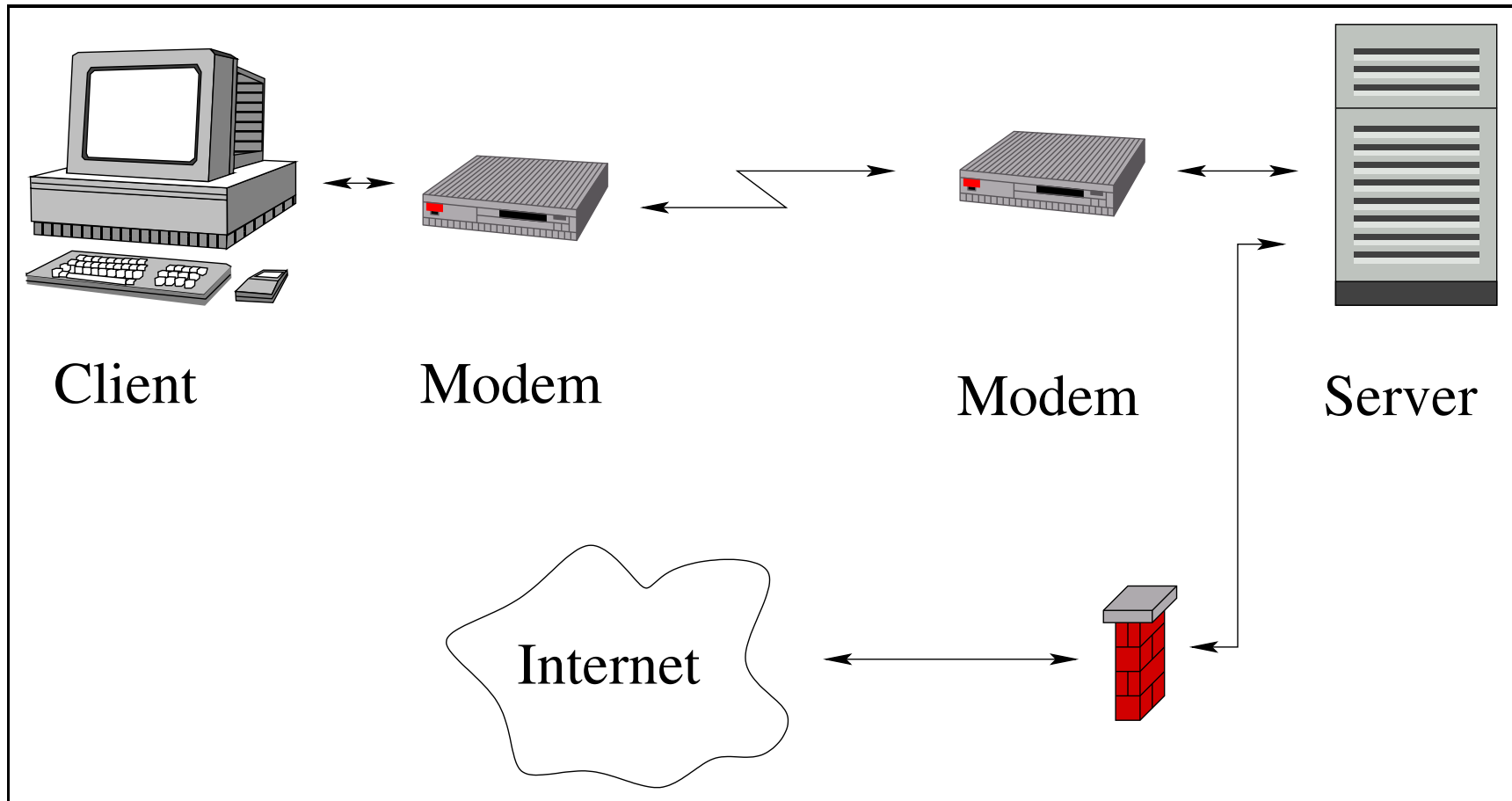
Configure Modem and Sound card []

Resources of interest

:

:

Setting Up a Shell Dialup Service



Inbound Shell Login - Server

Inbound Shell Login - Server

1. Edit `/etc/inittab` to automatically spawn `mgetty`.

```
$ tail -1 /etc/inittab ↔
```

```
T0:2345:respawn:/sbin/mgetty -x0 -s 57600 -D ttyS0
```

(Change `ttyS0` to whatever device your modem is connected to)

Inbound Shell Login - Server

1. Edit `/etc/inittab` to automatically spawn `mgetty`.

```
$ tail -1 /etc/inittab ↔
```

```
T0:2345:respawn:/sbin/mgetty -x0 -s 57600 -D ttyS0
```

(Change `ttyS0` to whatever device your modem is connected to)

2. Connect the modem and phone line.

Inbound Shell Login - Server

1. Edit `/etc/inittab` to automatically spawn `mgetty`.

```
$ tail -1 /etc/inittab ↔
```

```
T0:2345:respawn:/sbin/mgetty -x0 -s 57600 -D ttyS0
```

(Change `ttyS0` to whatever device your modem is connected to)

2. Connect the modem and phone line.

3. Run # **telinit q** ↔ to reread `/etc/inittab`

Inbound Shell Login - Server

1. Edit `/etc/inittab` to automatically spawn `mgetty`.

```
$ tail -1 /etc/inittab ↔
```

```
T0:2345:respawn:/sbin/mgetty -x0 -s 57600 -D ttyS0
```

(Change `ttyS0` to whatever device your modem is connected to)

2. Connect the modem and phone line.
3. Run `# telinit q` ↔ to reread `/etc/inittab`
4. Check that the modem's DTR indicator is on.

Inbound Shell Login - Server

1. Edit `/etc/inittab` to automatically spawn `mgetty`.

```
$ tail -1 /etc/inittab ↵
```

```
T0:2345:respawn:/sbin/mgetty -x0 -s 57600 -D ttyS0
```

(Change `ttyS0` to whatever device your modem is connected to)

2. Connect the modem and phone line.
3. Run `# telinit q` ↵ to reread `/etc/inittab`
4. Check that the modem's DTR indicator is on.
5. Check `mgetty` is there:

```
$ ps aux |grep mgetty ↵
```

```
root ..... /sbin/mgetty -x0 -s 57600 ttyS0
```

Outbound Shell login - Client

Outbound Shell login - Client

1. Run # `minicom -s` ↔ (as root) and setup the serial port to the desired `tty` and `linespeed`.

Outbound Shell login - Client

1. Run # **minicom -s** \leftrightarrow (as root) and setup the serial port to the desired `tty` and `linespeed`.
2. Give normal users access to the modem device.

Outbound Shell login - Client

1. Run # `minicom -s` ↔ (as root) and setup the serial port to the desired `tty` and `linespeed`.
2. Give normal users access to the modem device.
3. Run `minicom` as a normal user. Once the modem has initialised typing `AT` at the `minicom` terminal will prompt the modem to return `OK`:

`AT` ↔

`OK`

and dial your server:

`OK`

`ATDT12345678`

Outbound Shell login - Client

1. Run # `minicom -s` ↔ (as root) and setup the serial port to the desired `tty` and `linespeed`.
2. Give normal users access to the modem device.
3. Run `minicom` as a normal user. Once the modem has initialised typing `AT` at the `minicom` terminal will prompt the modem to return `OK`:

`AT` ↔

`OK`

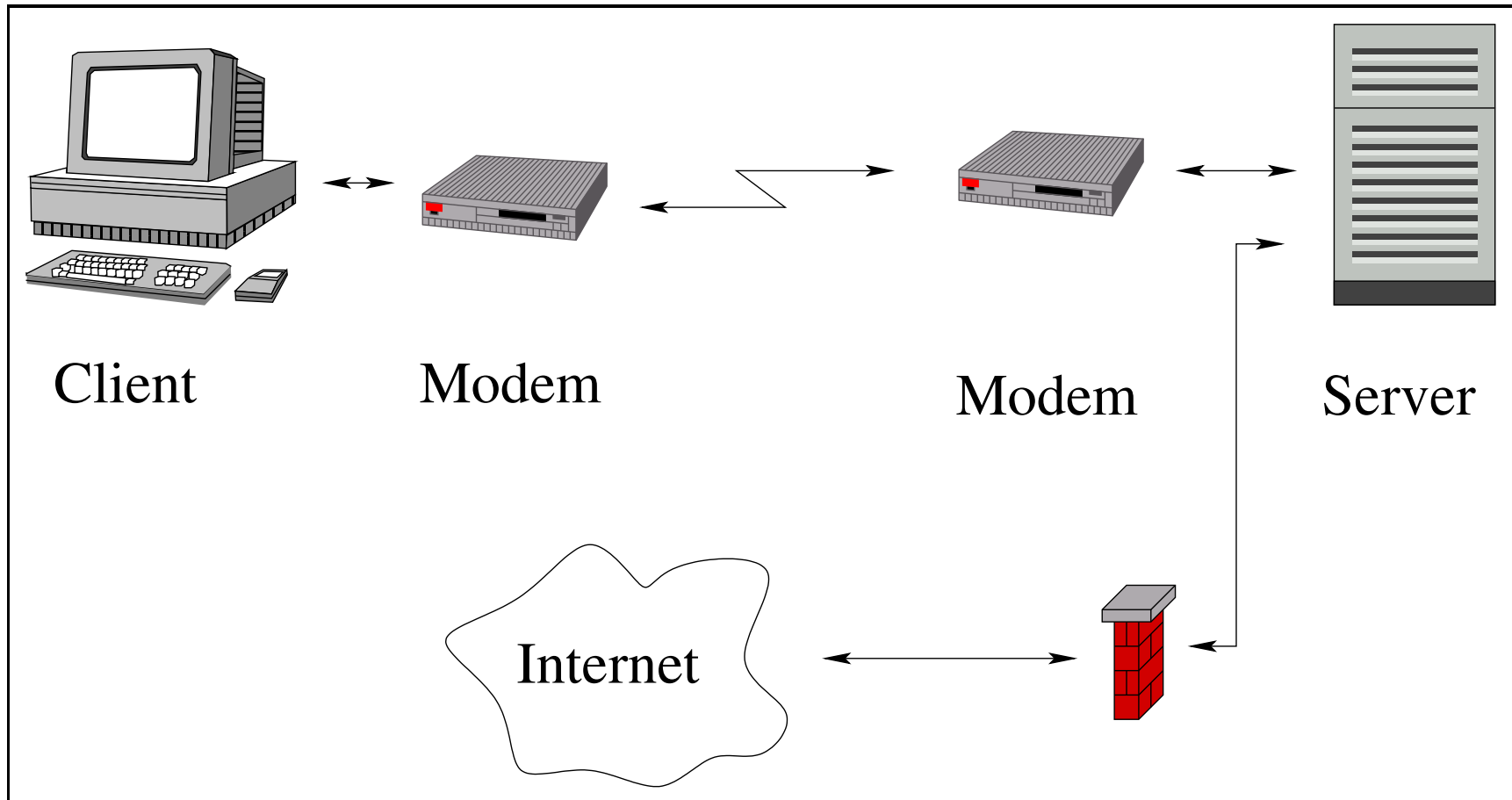
and dial your server:

`OK`

`ATDT12345678`

4. When the modem answers and trains, you should be presented with a login prompt. Login as you normally would.

Setting Up a PPP Dialup Service



Inbound Dialup ppp - Server

1. Edit `/etc/inittab` to run `mgetty`.

```
$ tail -1 /etc/inittab ↔
```

```
T0:2345:respawn:/sbin/mgetty -x0 -s 57600 -D ttyS0
```

Change `ttyS0` to whatever device your modem is connected to.

2. Add the following line (if not already there) to

```
/etc/modules.conf
```

```
alias ppp0 ppp_generic
```

3. Connect modem and phone line.

4. Run # `telinit q` ↔ to reread `/etc/inittab`

5. Edit `/etc/mgetty/login.config` and uncomment the line

```
/AutoPPP/ - a_ppp /usr/sbin/pppd auth -chap +pap login de
```

6. Edit the file `/etc/ppp/options` and uncomment or add the lines:

```
asynmap 0
#auth
crtcts
lock
modem
-detach
proxyarp
```

7. Edit the file `/etc/ppp/options.ttyS0` and uncomment or add the line:

```
192.168.0.253:192.168.0.10
```

The format of this line is `Server IP:Client IP`. Note that these addresses should be on the same network as your server unless you are prepared to setup routing for a new network. However, they should not clash with previously allocated IP addresses.

8. Edit `/etc/ppp/pap-secrets` and add:

```
#user          interface      password      allowed-ip-address
<username>    *              <password>    *
```

This line says: Let user `<username>` with password `<password>` use any ppp interface with any IP address. The username and password can be anything you like and do not have to be in `/etc/passwd`. You will need the username and password pair when you come to set up the client.

Outbound Dialup ppp - Client

NOTE: If your client & server machines are already connected via a LAN you will probably need to bring down the LAN on your client machine. (Or at least remove any route associated with the LAN from the client). To do this: `ifconfig eth0 down` Type `$ /sbin/route -n ↔` and make sure it looks like the line below before configuring ppp on the client:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	*	255.0.0.0	U	0	0	0	lo

Common to both methods: Edit `/etc/modules.conf` and add the following line:

```
alias ppp0 ppp_generic
```

From within X:

1. Run kppp and create an entry for your server, just as you would do for an ISP. The only items that need to be added are:
 - The telephone number of the server
 - The userid you selected in the server pap-secrets file
 - The password you selected in the server pap-secrets file
2. Click the Connect button and you should be away!.

From a terminal using wvdial:

Run # `wvdialconf /etc/wvdial.conf` ↔ . You should end up with a file called `/etc/wvdial.conf` that looks something like:

```
[Dialer Defaults]
Modem = /dev/ttyS0
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
; Phone = <Target Phone Number>
; Username = <Your Login Name>
; Password = <Your Password>
```


From a terminal using wvdial:

Now edit this file and add the following to the end of the file:

```
[Dialer <server-name>]
Username = username    #just as you entered into pap-secrets
Password = passwd      #just as you entered into pap-secrets
Phone = <telephone number>
Inherits = Dialer Defaults
Stupid mode = 1
New PPPD = 1
```

Stupid mode tells wvdial to start pppd as soon as it sees the login prompt and not to log into the server as a normal user first.

From a terminal using wvdial:

Now you can dial your server by doing the following:

```
$ wvdial <server-name>↵
```

Where `<server-name>` is the name you gave to the second dialer stanza above.

Note that `wvdial` will add an entry into `/etc/ppp/pap-secrets` containing the username and password pair automatically.

Adding Automatic DNS setup

You can have the server pass a pair of DNS IP addresses to the client which the client can use to resolve dns queries. If you do this, you should be able to use Windows as a client just like any other ISP.

Server Side:

Add the following line to `/etc/ppp/options`

```
ms-dns <DNS-IPADDR1>
```

```
ms-dns <DNS-IPADDR2>
```

The DNS ip addresses should be whatever your server is using (`$ cat /etc/resolv.conf ↔` to see). If your server is using the local interface (127.0.0.1) then you should set the address to that of eth0 and make sure that bind is configured to listen on that interface. The following line should be in `/etc/named.conf`

```
listen-on <ip-address>
```

Client Side:

Add the following line to `/etc/ppp/options`

```
usepeerdns
```

Now create a file called `/etc/ppp/ip-up.local` which contains:

```
cp /etc/resolv.conf /etc/ppp/resolv.conf.orig  
echo "nameserver $DNS1" > /etc/resolv.conf  
echo "nameserver $DNS2" >> /etc/resolv.conf
```

Create another file called `/etc/ppp/ip-down.local` which contains:

```
cp /etc/ppp/resolv.conf.orig /etc/resolv.conf
```

Congratulations. You now have a server which behaves just like an ISP!

The End