Modem Sharing mini−HOWTO

Friedemann Baitinger

fb@baiti.net

2001−08−22

Revision History

Revision v1.12	2001−08−22	Revised by: gjf
Corrected author's email		
Revision v1.11	2001−07−12	Revised by: aeg
Converted to DocBook SGML		
Revision v1.10	1999−09−11	Revised by: fb
Added "Feedback from the Users" section		
Revision v1.01	1997−06−12	Revised by: fb
Converted source to HTML		

Describes how to setup a Linux system to share a modem with other systems over a TCP/IP network.

Table of Contents

1. Legal Notice

2. The Server Side

It is assumed that the server is a Linux system with either:

- a modem attached to a `/dev/ttySx` device
- an 'isdn4linux'−emulated modem mapped to a `/dev/ttyIx` device

The easiest setup I can think of uses a five lines perl script to implement a modem daemon in `/usr/sbin/modemd`:

```
#!/usr/bin/perl
select((select(STDOUT), $| = 1)[$[]);
select((select(STDIN), $| = 1)[$[]);
exec 'cu −s 115200 −l /dev/ttyS1';
die '$0: Cant exec cu: $!\n';
```

The modem daemon is started by the **inetd** process if a client connects to the appropriate port as described below. The **modemd** simply connects the socket handle with STDIN and STDOUT of the **cu** command and lets **cu** handle the actual modem device.

The existence of the modem daemon must be made known to the **inetd** process by updating its configuration file, usually `/etc/inetd.conf` like:

```
#
# modem daemon
#
modem stream tcp nowait root /usr/sbin/tcpd /usr/sbin/modemd /dev/ttyS1
```

In order to make this work, an entry to `/etc/services` needs to be added like:

```
modem           2006/tcp        modemd
```

This associates a symbolic name with an explicit port, 2006 in the example. The portnumber could be any number not already assigned to an existing service. After these changes have been made, a signal must be sent to the inetd process in order to let **inetd** re−read and process its configuration file:

```
bash# ps | grep inetd
194  ?  S     0:00 /usr/sbin/inetd

bash# kill −HUP 194
```

Now the server side is ready to accept requests from clients. The correct function can be verified by:

```
bash$ telnet localhost modem

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

You are now connected to the modem. You can now issue **AT** commands in order to verify the setup:

```
atz
```

```
atz
OK

ati1
ati1
Linux ISDN
OK

^]
telnet>quit
bash$
```

Instead of using the Perl script as a modem server, there is also a program named Masqdialer available at http://w3.cpwright.com/mserver/.

With Masqdialer you can export any number of modems connected to your server to any host that can connect the server via TCP/IP on a given port with a binary data stream.

2.1. Masqdialer Installation

Before compiling check config.h for compile time options:

- set the path for the config file to your liking
- set the path for the lock file to your liking

Do **make all**.

Copy the binaries (`mserver` and `tcpconn`) into a suitable directory such as /usr/local/sbin/. Copy `mserver.conf` into the path that you specified in `config.h`.

Masqdialer could be started from one of your system startup scripts. A simple **/usr/local/sbin/mserver** will run it as a daemon.

2.2. Masqdialer Configuration

A line in `mserver.conf` could look like this:

```
5800 /dev/ttyS1      115200,8,N,1    *.foo.org,192.168.2.1
```

which would mean that a modem connected to `/dev/ttyS1` can be connected via port 5800 from anywhere in the domain foo.org and from host 192.168.2.1. Any other hosts are rejected. Pitfall: If you don't specify hosts then *ANY* host will be allowed to connect. The serial line settings are fixed and cannot be changed from the client side. You can export several modems on a single port. Masqdialer only locks the modem devices by use of UUCP style lock files when they are actually in use thus allowing other programs to take advantage of them.

3. The Client Side

At this time, only Windows client setups are described here. On the client PC, a COM−port redirector for TCP/IP is required. The best program for this purpose I have found is DialOut/IP from Tactical Software for Windows 3.1 and Windows 95. (The Windows 3.1 version can be used under Windows NT for 16−bit applications only. A 32−bit version for Windows NT is due late summer 1997.)

DialOut/IP presents the shared modem on a new virtual COM port that it adds to Windows. This virtual COM port can be used by Windows programs as if the shared modem is directly connected. Most client applications (including Windows 95 dial−up networking) accept this and work as if there were a real COM port and modem, with the general exception being fax applications or any others that need access to UART control lines. DialOut/IP can be configured to provide telnet protocol processing, but that feature applies to certain modem pool products and not to the Linux setup described in this file. Note that, despite its name, DialOut/IP can be used also by applications that wait for incoming calls.

On http://www.tactical−sw.com/ there is a page for downloading a fully functional evaluation version that times out in 1−2 weeks. Installation and configuration is handled by a setup program, with installation details in the `README.TXT` file. When you run DialOut/IP, you enter the IP address and port number of the shared modem.

DialOut/IP is a commercial product that is licensed on a per−modem basis, that is, the price depends on the number of modems that you are sharing. The license states that you can install the software on any number of PC's that access the shared modems.

---

4. Security Considerations

If you have only one modem for all your hosts in your local area network, there is probably no reason to worry about security here. However, if any one or more of the hosts in your LAN are connected to the Internet by other means than using the modem we have just setup as a modem server, then security considerations are required, otherwise anybody can do a **telnet your_host modem** and dial out long distance or even international calls at will.

I suggest to install and configure tcp−wrappers in order to protect the modem server against unauthorized access.

5. Examples

I am using the setup as described in The Server Side and The Client Side to run Quicken on my Windows 95 ThinkPad and do home banking with the modem attached to my Linux machine. The "modem" in my case is not even a real modem, it is an emulated modem on an ISDN−So card. Quicken just sees a COM port, it doesn't know that the device attached to the COM port is actually at the other end of my Ethernet LAN, nor does it know that it is not a standard analog modem but an ISDN device which happens to understand **AT** commands.

6. Feedback From Users

Since the first release of this document in June 1997 I have received many email messages related to the subject. In most of the messages people were seeking more help to get the modem sharing configured and running.

Recently I received an interesting feedback from Karsten.Hilbert@gmx.net. Karsten pointed out that although DialOut/IP may be a good client program he'd like to have a GPL'ed client. Karsten mentioned the Software Bazaar <http://visar.csustan.edu/bazaar/> and he volunteered to put in an offer. Here is what Karsen wrote:

```
Date: Fri, 27 Aug 1999 17:46:39 +0200 (CEST)
From: Karsten Hilbert <med94ecz@studserv.uni-leipzig.de>
Reply-To: Karsten Hilbert <Karsten.Hilbert@gmx.net>
To: fb@baiti.net
Subject: Windows-Modemsharing-Howto

Hi !

The howto mentions DialOut/IP as a good tool to connect Windows
clients to a linux server sharing a modem.

I agree. However, it would be more attractive to have a free,
GPL'ed client, wouldn't it ? This I thought and decided to offer
some money for the implementation of such a client on the
Software Bazaar. If someone grabs the project I will pay him a
certain amount of money if it is completed and functional. After
that the client would be GPL'ed.

Now, I can only offer so much money :) But other people could
join in and offer some, too, thus increasing the incentive.
Wouldn't your howto be a perfect place to mention this
possibility ?

The Bazaar can be found at:

 http://visar.csustan.edu/bazaar/

Thought I could mention this to you.


Karsten
```