

X.25 support within isdn4linux

This is alpha/beta test code. Use it completely at your own risk. As new versions appear, the stuff described here might suddenly change or become invalid without notice.

Keep in mind:

You are using several new parts of the 2.2.x kernel series which have not been tested in a large scale. Therefore, you might encounter more bugs as usual.

- If you connect to an X.25 neighbour not operated by yourself, ASK the other side first. Be prepared that bugs in the protocol implementation might result in problems.
- This implementation has never wiped out my whole hard disk yet. But as this is experimental code, don't blame me if that happened to you. Backing up important data will never harm.
- Monitor your isdn connections while using this software. This should prevent you from undesired phone bills in case of driver problems.

How to configure the kernel

The ITU-T (former CCITT) X.25 network protocol layer has been implemented in the Linux source tree since version 2.1.16. The isdn subsystem might be useful to run X.25 on top of ISDN. If you want to try it, select

"CCITT X.25 Packet Layer"

from the networking options as well as

"ISDN Support" and "X.25 PLP on Top of ISDN"

from the ISDN subsystem options when you configure your kernel for compilation. You currently also need to enable "Prompt for development and/or incomplete code/drivers" from the "Code maturity level options" menu. For the x25trace utility to work you also need to enable "Packet socket".

For local testing it is also recommended to enable the isdnloop driver from the isdn subsystem's configuration menu.

For testing, it is recommended that all isdn drivers and the X.25 PLP protocol are compiled as loadable modules. Like this, you can recover from certain errors by simply unloading and reloading the modules.

What's it for? How to use it?

=====

X.25 on top of isdn might be useful with two different scenarios:

- You might want to access a public X.25 data network from your Linux box. You can use i4l if you were physically connected to the X.25 switch by an ISDN B-channel (leased line as well as dial up connection should work).

This corresponds to ITU-T recommendation X.31 Case A (circuit-mode access to PSPDN [packet switched public data network]).

NOTE: X.31 also covers a Case B (access to PSPDN via virtual circuit / packet mode service). The latter mode (which in theory also allows using the D-channel) is not supported by isdn4linux. It should however be possible to establish such packet mode connections with certain active isdn cards provided that the firmware supports X.31 and the driver exports this functionality to the user. Currently, the AVM B1 driver is the only driver which does so. (It should be possible to access D-channel X.31 with active AVM cards using the CAPI interface of the AVM-B1 driver).

- Or you might want to operate certain ISDN teleservices on your linux box. A lot of those teleservices run on top of the ISO-8208 (DTE-DTE mode) network layer protocol. ISO-8208 is essentially the same as ITU-T X.25.

Popular candidates of such teleservices are EUROfile transfer or any teleservice applying ITU-T recommendation T.90.

To use the X.25 protocol on top of isdn, just create an isdn network interface as usual, configure your own and/or peer's ISDN numbers, and choose x25iface encapsulation by

```
isdnctrl encap <iface-name> x25iface.
```

Once encap is set like this, the device can be used by the X.25 packet layer.

All the stuff needed for X.25 is implemented inside the isdn link level (mainly isdn_net.c and some new source files). Thus, it should work with every existing HL driver. I was able to successfully open X.25 connections on top of the isdnloop driver and the hisax driver. "x25iface"-encapsulation bypasses demand dialing. Dialing will be initiated when the upper (X.25 packet) layer requests the lapb datalink to be established. But hangup timeout is still active. Whenever a hangup occurs, all existing X.25 connections on that link will be cleared. It is recommended to use sufficiently large hangup-timeouts for the isdn interfaces.

In order to set up a conforming protocol stack you also need to specify the proper l2_prot parameter:

To operate in ISO-8208 X.25 DTE-DTE mode, use

```
isdnctrl l2_prot <iface-name> x75i
```

To access an X.25 network switch via isdn (your linux box is the DTE), use

```
isdnctrl l2_prot <iface-name> x25dte
```

To mimic an X.25 network switch (DCE side of the connection), use

```
isdnctrl l2_prot <iface-name> x25dce
```

However, x25dte or x25dce is currently not supported by any real HL level driver. The main difference between x75i and x25dte/dce is that x25d[tc]e uses fixed lap_b addresses. With x75i, the side which initiates the isdn connection uses the DTE's lap_b address while the called side used the DCE's lap_b address. Thus, l2_prot x75i might probably work if you access a public X.25 network as long as the corresponding isdn connection is set up by you. At least one test was successful to connect via isdn4linux to an X.25 switch using this trick. At the switch side, a terminal adapter X.21 was used to connect it to the isdn.

How to set up a test installation?

=====

To test X.25 on top of isdn, you need to get

- a recent version of the "isdnctrl" program that supports setting the new X.25 specific parameters.
- the x25-utils-2.X package from
ftp://ftp.hes.iki.fi/pub/ham/linux/ax25/x25utils-
(don't confuse the x25-utils with the ax25-utils)
- an application program that uses linux PF_X25 sockets (some are contained in the x25-util package).

Before compiling the user level utilities make sure that the compiler/preprocessor will fetch the proper kernel header files of this kernel source tree. Either make /usr/include/linux a symbolic link pointing to this kernel's include/linux directory or set the appropriate compiler flags.

When all drivers and interfaces are loaded and configured you need to ifconfig the network interfaces up and add X.25-routes to them. Use the usual ifconfig tool.

```
ifconfig <iface-name> up
```

But a special x25route tool (distributed with the x25-util package) is needed to set up X.25 routes. I.e.

```
x25route add 01 <iface-name>
```

will cause all x.25 connections to the destination X.25-address "01" to be routed to your created isdn network interface.

README.x25.txt

There are currently no real X.25 applications available. However, for tests, the x25-utils package contains a modified version of telnet and telnetd that uses X.25 sockets instead of tcp/ip sockets. You can use those for your first tests. Furthermore, you might check <ftp://ftp.hamburg.pop.de/pub/LOCAL/linux/i41-eft/> which contains some alpha-test implementation ("eftp4linux") of the EUROfile transfer protocol.

The scripts distributed with the eftp4linux test releases might also provide useful examples for setting up X.25 on top of isdn.

The x25-utility package also contains an x25trace tool that can be used to monitor X.25 packets received by the network interfaces. The /proc/net/x25* files also contain useful information.

- Henner