

Linux DECnet Networking Layer Information

1. Other documentation....

- Project Home Pages - <http://www.chygwyn.com/> - Kernel info - <http://linux-decnet.sourceforge.net/> - Userland tools - <http://www.sourceforge.net/projects/linux-decnet/> - Status page

2. Configuring the kernel

Be sure to turn on the following options:

- CONFIG_DECNET (obviously)
- CONFIG_PROC_FS (to see what's going on)
- CONFIG_SYSCTL (for easy configuration)

if you want to try out router support (not properly debugged yet) you'll need the following options as well..

- CONFIG_DECNET_ROUTER (to be able to add/delete routes)
- CONFIG_NETFILTER (will be required for the DECnet routing daemon)

Don't turn on SIOCGIFCONF support for DECnet unless you are really sure that you need it, in general you won't and it can cause ifconfig to malfunction.

Run time configuration has changed slightly from the 2.4 system. If you want to configure an endnode, then the simplified procedure is as follows:

- Set the MAC address on your ethernet card before starting `_any_` other network protocols.

As soon as your network card is brought into the UP state, DECnet should start working. If you need something more complicated or are unsure how to set the MAC address, see the next section. Also all configurations which worked with 2.4 will work under 2.5 with no change.

3. Command line options

You can set a DECnet address on the kernel command line for compatibility with the 2.4 configuration procedure, but in general it's not needed any more. If you do set a DECnet address on the command line, it has only one purpose which is that it's added to the addresses on the loopback device.

With 2.4 kernels, DECnet would only recognise addresses as local if they were added to the loopback device. In 2.5, any local interface address can be used to loop back to the local machine. Of course this does not prevent you adding further addresses to the loopback device if you want to.

N.B. Since the address list of an interface determines the addresses for which "hello" messages are sent, if you don't set an address on the loopback interface then you won't see any entries in `/proc/net/neigh` for the local host until such time as you start a connection. This doesn't affect the operation of the local communications in any other way though.

The kernel command line takes options looking like the following:

```
decnet.addr=1,2
```

the two numbers are the node address 1,2 = 1.2 For 2.2.xx kernels and early 2.3.xx kernels, you must use a comma when specifying the DECnet address like this. For more recent 2.3.xx kernels, you may use almost any character except space, although a `.` would be the most obvious choice :-)

There used to be a third number specifying the node type. This option has gone away in favour of a per interface node type. This is now set using `/proc/sys/net/decnet/conf/<dev>/forwarding`. This file can be set with a single digit, 0=EndNode, 1=L1 Router and 2=L2 Router.

There are also equivalent options for modules. The node address can also be set through the `/proc/sys/net/decnet/` files, as can other system parameters.

Currently the only supported devices are ethernet and `ip_gre`. The ethernet address of your ethernet card has to be set according to the DECnet address of the node in order for it to be autoconfigured (and then appear in `/proc/net/decnet_dev`). There is a utility available at the above FTP sites called `dn2ethaddr` which can compute the correct ethernet address to use. The address can be set by `ifconfig` either before or at the time the device is brought up. If you are using RedHat you can add the line:

```
MACADDR=AA:00:04:00:03:04
```

or something similar, to `/etc/sysconfig/network-scripts/ifcfg-eth0` or wherever your network card's configuration lives. Setting the MAC address of your ethernet card to an address starting with "hi-ord" will cause a DECnet address which matches to be added to

the interface (which you can verify with `iproute2`).

The default device for routing can be set through the `/proc` filesystem by setting `/proc/sys/net/decnet/default_device` to the device you want DECnet to route packets out of when no specific route is available. Usually this will be `eth0`, for example:

```
echo -n "eth0" >/proc/sys/net/decnet/default_device
```

If you don't set the default device, then it will default to the first ethernet card which has been autoconfigured as described above. You can confirm that by looking in the `default_device` file of course.

There is a list of what the other files under `/proc/sys/net/decnet/` do on the kernel patch web site (shown above).

4. Run time kernel configuration

This is either done through the `sysctl/proc` interface (see the kernel web pages for details on what the various options do) or through the `iproute2` package in the same way as IPv4/6 configuration is performed.

Documentation for `iproute2` is included with the package, although there is as yet no specific section on DECnet, most of the features apply to both IP and DECnet, albeit with DECnet addresses instead of IP addresses and a reduced functionality.

If you want to configure a DECnet router you'll need the `iproute2` package since its the only way to add and delete routes currently. Eventually there will be a routing daemon to send and receive routing messages for each interface and update the kernel routing tables accordingly. The routing daemon will use `netfilter` to listen to routing packets, and `rtnetlink` to update the kernels routing tables.

The DECnet raw socket layer has been removed since it was there purely for use by the routing daemon which will now use `netfilter` (a much cleaner and more generic solution) instead.

5. How can I tell if its working?

Here is a quick guide of what to look for in order to know if your DECnet kernel subsystem is working.

- Is the node address set (see `/proc/sys/net/decnet/node_address`)
- Is the node of the correct type (see `/proc/sys/net/decnet/conf/<dev>/forwarding`)
- Is the Ethernet MAC address of each Ethernet card set to match the DECnet address. If in doubt use the `dn2ethaddr` utility available at the [ftp archive](#).
- If the previous two steps are satisfied, and the Ethernet card is up, you should find that it is listed in `/proc/net/decnet_dev` and also that it appears as a directory in `/proc/sys/net/decnet/conf/`. The loopback device (`lo`) should also appear and is required to communicate within a node.
- If you have any DECnet routers on your network, they should appear in `/proc/net/decnet_neigh`, otherwise this file will only contain the entry for the node itself (if it doesn't check to see if `lo` is up).
- If you want to send to any node which is not listed in the `/proc/net/decnet_neigh` file, you'll need to set the default device to point to an Ethernet card with connection to a router. This is again done with the `/proc/sys/net/decnet/default_device` file.
- Try starting a simple server and client, like the `dnping/dnmirror` over the loopback interface. With luck they should communicate. For this step and those after, you'll need the DECnet library which can be obtained from the above [ftp sites](#) as well as the actual utilities themselves.
- If this seems to work, then try talking to a node on your local network, and see if you can obtain the same results.
- At this point you are on your own... :-)

6. How to send a bug report

If you've found a bug and want to report it, then there are several things you can do to help me work out exactly what it is that is wrong. Useful information (most of which is essential) includes:

- What kernel version are you running ?
- What version of the patch are you running ?
- How far though the above set of tests can you get ?
- What is in the `/proc/decnet*` files and `/proc/sys/net/decnet/*` files ?
- Which services are you running ?
- Which client caused the problem ?
- How much data was being transferred ?
- Was the network congested ?
- How can the problem be reproduced ?
- Can you use `tcpdump` to get a trace ? (N.B. Most (all?) versions of `tcpdump` don't understand how to dump DECnet properly, so including the hex listing of the packet contents is essential, usually the `-x` flag. You may also need to increase the length grabbed with the `-s` flag. The `-e` flag also provides very useful information (ethernet MAC addresses))

7. MAC FAQ

A quick FAQ on ethernet MAC addresses to explain how Linux and DECnet interact and how to get the best performance from your hardware.

Ethernet cards are designed to normally only pass received network frames to a host computer when they are addressed to it, or to the broadcast address.

Linux has an interface which allows the setting of extra addresses for an ethernet card to listen to. If the ethernet card supports it, the filtering operation will be done in hardware, if not the extra unwanted packets received will be discarded by the host computer. In the latter case, significant processor time and bus bandwidth can be used up on a busy network (see the NAPI documentation for a longer explanation of these effects).

DECnet makes use of this interface to allow running DECnet on an ethernet card which has already been configured using TCP/IP (presumably using the built in MAC address of the card, as usual) and/or to allow multiple DECnet addresses on each physical interface. If you do this, be aware that if your ethernet card doesn't support perfect hashing in its MAC address filter then your computer will be doing more work than required. Some cards will simply set themselves into promiscuous mode in order to receive packets from the DECnet specified addresses. So if you have one of these cards its better to set the MAC address of the card as described above to gain the best efficiency. Better still is to use a card which supports NAPI as well.

8. Mailing list

If you are keen to get involved in development, or want to ask questions about configuration, or even just report bugs, then there is a mailing list that you can join, details are at:

http://sourceforge.net/mail/?group_id=4993

9. Legal Info

The Linux DECnet project team have placed their code under the GPL. The software is provided "as is" and without warranty express or implied. DECnet is a trademark of Compaq. This software is not a product of Compaq. We acknowledge the help of people at Compaq in providing extra documentation above and beyond what was previously publicly available.

Steve Whitehouse <SteveW@ACM.org>