

D-Link DL2000-based Gigabit Ethernet Adapter Installation

May 23, 2002

Compatibility List

Adapter Support:

- D-Link DGE-550T Gigabit Ethernet Adapter.
- D-Link DGE-550SX Gigabit Ethernet Adapter.
- D-Link DL2000-based Gigabit Ethernet Adapter.

The driver support Linux kernel 2.4.7 later. We had tested it on the environments below.

. Red Hat v6.2 (update kernel to 2.4.7) . Red Hat v7.0 (update kernel to 2.4.7) . Red Hat v7.1 (kernel 2.4.7) . Red Hat v7.2 (kernel 2.4.7-10)

Quick Install

Install linux driver as following command:

```
1. make all
2. insmod dl2k.ko
3. ifconfig eth0 up 10.xxx.xxx.xxx netmask 255.0.0.0
                        ^^^^^^^^^^^^^^^^^^      ^^^^^^^^^^
                        IP                        NETMASK
```

Now eth0 should active, you can test it by "ping" or get more information by "ifconfig". If tested ok, continue the next step.

4. `cp dl2k.ko /lib/modules/`uname -r`/kernel/drivers/net`
5. Add the following line to `/etc/modprobe.d/dl2k.conf`

```
alias eth0 dl2k
```
6. Run `depmod` to updated module indexes.
7. Run `netconfig` or `netconf` to create configuration script `ifcfg-eth0` located at `/etc/sysconfig/network-scripts` or create it manually.
[see - Configuration Script Sample]
8. Driver will automatically load and configure at next boot time.

Compiling the Driver

In Linux, NIC drivers are most commonly configured as loadable modules. The approach of building a monolithic kernel has become obsolete. The driver can be compiled as part of a monolithic kernel, but is strongly discouraged. The remainder of this section assumes the driver is built as a loadable module. In the Linux environment, it is a good idea to rebuild the driver from the source instead of relying on a precompiled version. This approach provides better reliability since a precompiled driver might depend on libraries or kernel features that are not present in a given Linux installation.

The 3 files necessary to build Linux device driver are `dl2k.c`, `dl2k.h` and `Makefile`. To compile, the Linux installation must include the `gcc` compiler, the kernel source, and the kernel headers. The Linux driver supports Linux Kernels 2.4.7. Copy the files to a directory and enter the following command to compile and link the driver:

CD-ROM drive

```
[root@XXX /] mkdir cdrom
[root@XXX /] mount -r -t iso9660 -o conv=auto /dev/cdrom /cdrom
[root@XXX /] cd root
[root@XXX /root] mkdir dl2k
[root@XXX /root] cd dl2k
[root@XXX dl2k] cp /cdrom/linux/dl2k.tgz /root/dl2k
[root@XXX dl2k] tar xfvz dl2k.tgz
[root@XXX dl2k] make all
```

Floppy disc drive

```
[root@XXX /] cd root
[root@XXX /root] mkdir dl2k
```

```
[root@XXX /root] cd dl2k
[root@XXX dl2k] mcopy a:/linux/dl2k.tgz /root/dl2k
[root@XXX dl2k] tar xfvz dl2k.tgz
[root@XXX dl2k] make all
```

Installing the Driver

Manual Installation

Once the driver has been compiled, it must be loaded, enabled, and bound to a protocol stack in order to establish network connectivity. To load a module enter the command:

```
insmod dl2k.o
```

or:

```
insmod dl2k.o <optional parameter> ; add parameter
```

example:

```
insmod dl2k.o media=100mbps_hd
```

or::

```
insmod dl2k.o media=3
```

or::

```
insmod dl2k.o media=3,2 ; for 2 cards
```

Please reference the list of the command line parameters supported by the Linux device driver below.

The `insmod` command only loads the driver and gives it a name of the form `eth0`, `eth1`, etc. To bring the NIC into an operational state, it is necessary to issue the following command:

```
ifconfig eth0 up
```

Finally, to bind the driver to the active protocol (e.g., TCP/IP with Linux), enter the following command:

```
ifup eth0
```

Note that this is meaningful only if the system can find a configuration script that contains the necessary network information. A sample will be given in the next paragraph.

The commands to unload a driver are as follows:

```
ifdown eth0
ifconfig eth0 down
rmmod dl2k.o
```

The following are the commands to list the currently loaded modules and to see the current network configuration:

```
lsmod
ifconfig
```

Automated Installation

This section describes how to install the driver such that it is automatically loaded and configured at boot time. The following description is based on a Red Hat 6.0/7.0 distribution, but it can easily be ported to other distributions as well.

Red Hat v6.x/v7.x

1. Copy `dl2k.o` to the network modules directory, typically `/lib/modules/2.x.x-xx/net` or `/lib/modules/2.x.x/kernel/drivers/net`.
2. Locate the boot module configuration file, most commonly in the `/etc/modprobe.d/` directory. Add the following lines:

```
alias ethx dl2k
options dl2k <optional parameters>
```

where `ethx` will be `eth0` if the NIC is the only ethernet adapter, `eth1` if one other ethernet adapter is installed, etc. Refer to the table in the previous section for the list of optional parameters.

3. Locate the network configuration scripts, normally the `/etc/sysconfig/network-scripts` directory, and create a

configuration script named ifcfg-ethx that contains network information.

- Note that for most Linux distributions, Red Hat included, a configuration utility with a graphical user interface is provided to perform steps 2 and 3 above.

Parameter Description

You can install this driver without any additional parameter. However, if you are going to have extensive functions then it is necessary to set extra parameter. Below is a list of the command line parameters supported by the Linux device driver.

mtu=packet_size	Specifies the maximum packet size. default is 1500.
media=media_type	Specifies the media type the NIC operates at. autosense Autosensing active media.
	10mbps_hd 10Mbps half duplex.
	10mbps_fd 10Mbps full duplex.
	100mbps_hd 100Mbps half duplex.
	100mbps_fd 100Mbps full duplex.
	1000mbps_fd 1000Mbps full duplex.
	1000mbps_hd 1000Mbps half duplex.
	0 Autosensing active media.
	1 10Mbps half duplex.
	2 10Mbps full duplex.
	3 100Mbps half duplex.
	4 100Mbps full duplex.
	5 1000Mbps half duplex.
	6 1000Mbps full duplex.
	By default, the NIC operates at autosense. 1000mbps_fd and 1000mbps_hd types are only available for fiber adapter.
vlan=n	Specifies the VLAN ID. If vlan=0, the Virtual Local Area Network (VLAN) function is disable.
jumbo=[0 1]	Specifies the jumbo frame support. If jumbo=1, the NIC accept jumbo frames. By default, this function is disabled. Jumbo frame usually improve the performance int gigabit. This feature need jumbo frame compatible remote.
rx_coalesce=m	Number of rx frame handled each interrupt.
rx_timeout=n	<p>Rx DMA wait time for an interrupt. If set rx_coalesce > 0, hardware only assert an interrupt for m frames. Hardware won't assert rx interrupt until m frames received or reach timeout of n * 640 nano seconds. Set proper rx_coalesce and rx_timeout can reduce congestion collapse and overload which has been a bottleneck for high speed network.</p> <p>For example, rx_coalesce=10 rx_timeout=800. that is, hardware assert only 1 interrupt for 10 frames received or timeout of 512 us.</p>
tx_coalesce=n	Number of tx frame handled each interrupt. Set n > 1 can reduce the interrupts congestion usually lower performance of high speed network card. Default is 16.
tx_flow=[1 0]	Specifies the Tx flow control. If tx_flow=0, the Tx flow control disable else driver autotdetect.
rx_flow=[1 0]	Specifies the Rx flow control. If rx_flow=0, the Rx flow control enable else driver autotdetect.

Configuration Script Sample

Here is a sample of a simple configuration script:

```
DEVICE=eth0
USERCTL=no
ONBOOT=yes
BOOTPROTO=none
BROADCAST=207.200.5.255
NETWORK=207.200.5.0
NETMASK=255.255.255.0
IPADDR=207.200.5.2
```

Troubleshooting

Q1. Source files contain ^ M behind every line.

Make sure all files are Unix file format (no LF). Try the following shell command to convert files:

```
cat dl2k.c | col -b > dl2k.tmp  
mv dl2k.tmp dl2k.c
```

OR:

```
cat dl2k.c | tr -d "\r" > dl2k.tmp  
mv dl2k.tmp dl2k.c
```

Q2: Could not find header files (*.h)?

To compile the driver, you need kernel header files. After installing the kernel source, the header files are usually located in `/usr/src/linux/include`, which is the default include directory configured in Makefile. For some distributions, there is a copy of header files in `/usr/src/include/linux` and `/usr/src/include/asm`, that you can change the `INCLUDEDIR` in Makefile to `/usr/include` without installing kernel source.

Note that RH 7.0 didn't provide correct header files in `/usr/include`, including those files will make a wrong version driver.