Linux and the 3Com EtherLink III Series Ethercards (driver v1.18c and higher)

This file contains the instructions and caveats for v1.18c and higher versions of the 3c509 driver. You should not use the driver without reading this file.

release 1.0

28 February 2002

Current maintainer (corrections to):
        David Ruggiero <jdr@farfalle.com>

Introduction

The following are notes and information on using the 3Com EtherLink III series ethercards in Linux. These cards are commonly known by the most widely-used card's 3Com model number, 3c509. They are all 10mb/s ISA-bus cards and shouldn't be (but sometimes are) confused with the similarly-numbered PCI-bus "3c905" (aka "Vortex" or "Boomerang") series. Kernel support for the 3c509 family is provided by the module 3c509.c, which has code to support all of the following models:

- 3c509 (original ISA card)
- 3c509B (later revision of the ISA card; supports full-duplex)
- 3c589 (PCMCIA)
- 3c589B (later revision of the 3c589; supports full-duplex)
- 3c579 (EISA)

Large portions of this documentation were heavily borrowed from the guide written the original author of the 3c509 driver, Donald Becker. The master copy of that document, which contains notes on older versions of the driver, currently resides on Scyld web server: http://www.scyld.com/.

Special Driver Features

Overriding card settings

The driver allows boot- or load-time overriding of the card's detected IOADDR, IRQ, and transceiver settings, although this capability shouldn't generally be needed except to enable full-duplex mode (see below). An example of the syntax for LILO parameters for doing this:

```
ether=10,0x310,3,0x3c509,eth0
```

This configures the first found 3c509 card for IRQ 10, base I/O 0x310, and transceiver type 3 (10base2). The flag "0x3c509" must be set to avoid conflicts with other card types when overriding the I/O address. When the driver is loaded as a module, only the IRQ may be overridden. For example, setting two cards to IRQ10 and IRQ11 is done by using the irq module option:

```
options 3c509 irq=10,11
```

Full-duplex mode

The v1.18c driver added support for the 3c509B's full-duplex capabilities. In order to enable and successfully use full-duplex mode, three conditions must be met:

(a) You must have a Etherlink III card model whose hardware supports full- duplex operations. Currently, the only members of the 3c509 family that are positively known to support full-duplex are the 3c509B (ISA bus) and 3c589B (PCMCIA) cards. Cards without the "B" model designation do *not* support full-duplex mode; these include the original 3c509 (no "B"), the original 3c589, the 3c529 (MCA bus), and the 3c579 (EISA bus).

(b) You must be using your card's 10baseT transceiver (i.e., the RJ-45 connector), not its AUI (thick-net) or 10base2 (thin-net/coax) interfaces. AUI and 10base2 network cabling is physically incapable of full-duplex operation.

(c) Most importantly, your 3c509B must be connected to a link partner that is itself full-duplex capable. This is almost certainly one of two things: a full- duplex-capable Ethernet switch (*not* a hub), or a full-duplex-capable NIC on another system that's connected directly to the 3c509B via a crossover cable.

Full-duplex mode can be enabled using 'ethtool'.

> **Warning**
>
> Extremely important caution concerning full-duplex mode

Understand that the 3c509B's hardware's full-duplex support is much more limited than that provide by more modern network interface cards. Although at the physical layer of the network it fully supports full-duplex operation, the card was designed before the current Ethernet auto-negotiation (N-way) spec was written. This means that the 3c509B family ***cannot and will not auto-negotiate a full-duplex connection with its link partner under any circumstances, no matter how it is initialized***. If the full-duplex mode of the 3c509B is enabled, its link partner will very likely need to be independently _forced_ into full-duplex mode as well; otherwise various nasty failures will occur - at the very least, you'll see massive numbers of packet collisions. This is one of very rare circumstances where disabling auto- negotiation and forcing the duplex mode of a network interface card or switch would ever be necessary or desirable.

Available Transceiver Types

For versions of the driver v1.18c and above, the available transceiver types are:

0	transceiver type from EEPROM config (normally 10baseT); force half-duplex
1	AUI (thick-net / DB15 connector)
2	(undefined)
3	10base2 (thin-net == coax / BNC connector)
4	10baseT (RJ-45 connector); force half-duplex mode
8	transceiver type and duplex mode taken from card's EEPROM config settings
12	10baseT (RJ-45 connector); force full-duplex mode

Prior to driver version 1.18c, only transceiver codes 0-4 were supported. Note that the new transceiver codes 8 and 12 are the *only* ones that will enable full-duplex mode, no matter what the card's detected EEPROM settings might be. This insured that merely upgrading the driver from an earlier version would never automatically enable full-duplex mode in an existing installation; it must always be explicitly enabled via one of these code in order to be activated.

The transceiver type can be changed using 'ethtool'.

Interpretation of error messages and common problems

Error Messages

eth0: Infinite loop in interrupt, status 2011. These are "mostly harmless" message indicating that the driver had too much work during that interrupt cycle. With a status of 0x2011 you are receiving packets faster than they can be removed from the card. This should be rare or impossible in normal operation. Possible causes of this error report are:

- a "green" mode enabled that slows the processor down when there is no keyboard activity.
- some other device or device driver hogging the bus or disabling interrupts. Check /proc/interrupts for excessive interrupt counts. The timer tick interrupt should always be incrementing faster than the others.

No received packets

If a 3c509, 3c562 or 3c589 can successfully transmit packets, but never receives packets (as reported by /proc/net/dev or 'ifconfig') you likely have an interrupt line problem. Check /proc/interrupts to verify that the card is actually generating interrupts. If the interrupt count is not increasing you likely have a physical conflict with two devices trying to use the same ISA IRQ line. The common conflict is with a sound card on IRQ10 or IRQ5, and the easiest solution is to move the 3c509 to a different interrupt line. If the device is receiving packets but 'ping' doesn't work, you have a routing problem.

Tx Carrier Errors Reported in /proc/net/dev

If an EtherLink III appears to transmit packets, but the "Tx carrier errors" field in /proc/net/dev increments as quickly as the Tx packet count, you likely have an unterminated network or the incorrect media transceiver selected.

3c509B card is not detected on machines with an ISA PnP BIOS.

While the updated driver works with most PnP BIOS programs, it does not work with all. This can be fixed by disabling PnP support using the 3Com-supplied setup program.

3c509 card is not detected on overclocked machines

Increase the delay time in id_read_eeprom() from the current value, 500, to an absurdly high value, such as 5000.

Decoding Status and Error Messages

The bits in the main status register are:

value	description
0x01	Interrupt latch

value	description
0x02	Tx overrun, or Rx underrun
0x04	Tx complete
0x08	Tx FIFO room available
0x10	A complete Rx packet has arrived
0x20	A Rx packet has started to arrive
0x40	The driver has requested an interrupt
0x80	Statistics counter nearly full

The bits in the transmit (Tx) status word are:

value	description
0x02	Out-of-window collision.
0x04	Status stack overflow (normally impossible).
0x08	16 collisions.
0x10	Tx underrun (not enough PCI bus bandwidth).
0x20	Tx jabber.
0x40	Tx interrupt requested.
0x80	Status is valid (this should always be set).

When a transmit error occurs the driver produces a status message such as:

```
eth0: Transmit error, Tx status register 82
```

The two values typically seen here are:

0x82

Out of window collision. This typically occurs when some other Ethernet host is incorrectly set to full duplex on a half duplex network.

0x88

16 collisions. This typically occurs when the network is exceptionally busy or when another host doesn't correctly back off after a collision. If this error is mixed with 0x82 errors it is the result of a host incorrectly set to full duplex (see above).

Both of these errors are the result of network problems that should be corrected. They do not represent driver malfunction.

Revision history (this file)

28Feb02 v1.0 DR New; major portions based on Becker original 3c509 docs