# Linux Ethernet-HOWTO

by Paul Gortmaker v2.7, 5 maggio 1999

Questo è Ethernet Howto, una raccolta di informazioni su quali dispositivi Ethernet possono essere usati con Linux e su come configurarli. Si noti che questo Howto si concentra sull'aspetto hardware e sui driver a basso livello delle schede Ethernet e non tratta l'aspetto software di cose come `ifconfig` e `route`. Si veda il Network Howto per tali informazioni. Traduzione a cura di Lorenza Romano (*titti@dei.unipd.it*) e Giovanni Bortolozzo (*borto@pluto.linux.it*).

## Indice

**1 Introduzione** **8**

1.1 Nuove versioni di questo documento . . . 9

1.2 Come usare Ethernet-Howto . . . 9

1.3 AIUTO – Non funziona! . . . 9

**2 Che scheda si dovrebbe acquistare per Linux?** **11**

2.1 E quali sono i driver stabili? . . . 12

2.2 Schede a 8 bit e schede a 16 bit a confronto . . . 12

2.3 Schede Ethernet (VLB/EISA/PCI) a 32 bit . . . 12

2.4 Schede e driver a 100Mbps disponibili . . . 13

2.5 100VG e 100BaseT a confronto . . . 13

2.6 Tipi di cavo che la propria scheda dovrebbe supportare . . . 13

**3 Domande frequenti** **14**

3.1 Driver alpha – come procurarseli e come usarli . . . 14

3.2 Come usare più di una scheda Ethernet per macchina . . . 14

3.3 Il comando `ether=` non è servito a niente. Perché? . . . 16

3.4 Problemi con schede NE1000/NE2000 (e cloni) . . . 17

3.5 Problemi con le schede SMC Ultra/EtherEZ e WD80*3 . . . 21

3.6 Problemi con le schede 3Com . . . 22

3.7 FAQ non specifiche su una particolare scheda . . . 22

3.7.1 Linux e schede Ethernet ISA di tipo Plug and Play . . . 22

3.7.2 La scheda Ethernet non è rilevata all'avvio . . . 23

3.7.3 `ifconfig` mostra un indirizzo di I/O sbagliato per la scheda . . . 24

3.7.4 La macchina PCI rileva la scheda ma il driver ma no . . . 24

3.7.5 Le schede ISA a memoria condivisa non funzionano in una macchina PCI (`0xffff`) . . . 24

3.7.6 Sembra che la scheda invii dati ma non riceve niente . . . 24

3.7.7 Supporto per Asynchronous Transfer Mode (ATM) . . . 24
3.7.8 Supporto per Gigabyte Ethernet . . . 24
3.7.9 Supporto FDDI . . . 25
3.7.10 Supporto Full Duplex . . . 25
3.7.11 Schede Ethernet per Linux su macchine SMP . . . 25
3.7.12 Schede Ethernet per Linux su piattaforma PCI Alpha/AXP . . . 27
3.7.13 Ethernet per Linux su hardware SUN/Sparc . . . 27
3.7.14 Ethernet per Linux su altro hardware . . . 27
3.7.15 Connettere 10 o 100 BaseT senza un hub . . . 27
3.7.16 SIOCSIFxxx: No such device . . . 28
3.7.17 SIOCSFFLAGS: Try again . . . 28
3.7.18 Usare 'ifconfig' e Link UNSPEC con l'indirizzo hardware 00:00:00:00:00:00 . . . 28
3.7.19 Enorme numero di errori in RX e TX . . . 28
3.7.20 Voci in `/dev/` per le schede Ethernet . . . 28
3.7.21 Linux e i "trailer" . . . 29
3.7.22 Accesso a basso livello al dispositivo Ethernet . . . 29

**4 Suggerimenti per le prestazioni 29**

4.1 Concetti generali . . . 29
4.2 Schede ISA e velocità del bus ISA . . . 30
4.3 Impostare la finestra TCP di ricezione (TCP Rx Window) . . . 30
4.4 Incrementare le prestazioni NFS . . . 31

**5 Informazioni specifiche su rivenditori, produttori e modelli 31**

5.1 3Com . . . 32
5.1.1 3c501 . . . 32
5.1.2 EtherLink II, 3c503, 3c503/16 . . . 32
5.1.3 Etherlink Plus 3c505 . . . 33
5.1.4 Etherlink-16 3c507 . . . 33
5.1.5 Etherlink III, 3c509 / 3c509B . . . 33
5.1.6 3c515 . . . 34
5.1.7 3c523 . . . 34
5.1.8 3c527 . . . 35
5.1.9 3c529 . . . 35
5.1.10 3c562 . . . 35
5.1.11 3c575 . . . 35
5.1.12 3c579 . . . 35
5.1.13 3c589 / 3c589B . . . 36

5.1.14 3c590 / 3c595 . . . 36
5.1.15 3c592 / 3c597 . . . 36
5.1.16 3c900 / 3c905 / 3c905B . . . 36
5.1.17 3c985 . . . 37
5.2 Accton . . . 37
5.2.1 Accton MPX . . . 37
5.2.2 Accton EN1203, EN1207, EtherDuo-PCI . . . 37
5.2.3 Accton EN2209 Parallel Port Adaptor (EtherPocket) . . . 37
5.2.4 Accton EN2212 PCMCIA Card . . . 37
5.3 Allied Telesyn/Telesis . . . 37
5.3.1 AT1500 . . . 37
5.3.2 AT1700 . . . 38
5.3.3 AT2450 . . . 38
5.3.4 AT2500 . . . 38
5.3.5 AT2540FX . . . 38
5.4 AMD / Advanced Micro Devices . . . 39
5.4.1 AMD LANCE (7990, 79C960/961/961A, PCnet-ISA) . . . 39
5.4.2 AMD 79C965 (PCnet-32) . . . 39
5.4.3 AMD 79C970/970A (PCnet-PCI) . . . 40
5.4.4 AMD 79C971 (PCnet-FAST) . . . 40
5.4.5 AMD 79C972 (PCnet-FAST+) . . . 40
5.4.6 AMD 79C974 (PCnet-SCSI) . . . 40
5.5 Ansel Communications . . . 40
5.5.1 AC3200 EISA . . . 40
5.6 Apricot . . . 41
5.6.1 Apricot Xen-II On Board Ethernet . . . 41
5.7 Arcnet . . . 41
5.8 AT&T . . . 41
5.8.1 AT&T T7231 (LanPACER+) . . . 41
5.9 Boca Research . . . 41
5.9.1 Boca BEN (ISA, VLB, PCI) . . . 41
5.10 Cabletron . . . 42
5.10.1 E10**, E10**-x, E20**, E20**-x . . . 42
5.10.2 E2100 . . . 42
5.10.3 E22** . . . 43
5.11 Cogent . . . 43
5.11.1 EM100-ISA/EISA . . . 43

5.11.2 Cogent eMASTER+, EM100-PCI, EM400, EM960, EM964 . . . . . 44
5.12 Compaq . . . . . 44
5.12.1 Compaq Deskpro / Compaq XL (Embedded AMD Chip) . . . . . 44
5.12.2 Compaq Nettelligent/NetFlex (Embedded ThunderLAN Chip) . . . . . 44
5.13 Danpex . . . . . 44
5.13.1 Danpex EN9400 . . . . . 44
5.14 D-Link . . . . . 45
5.14.1 DE-100, DE-200, DE-220-T, DE-250 . . . . . 45
5.14.2 DE-520 . . . . . 45
5.14.3 DE-528 . . . . . 45
5.14.4 DE-530 . . . . . 45
5.14.5 DE-600 . . . . . 45
5.14.6 DE-620 . . . . . 46
5.14.7 DE-650 . . . . . 46
5.15 DFI . . . . . 46
5.15.1 DFINET-300 e DFINET-400 . . . . . 46
5.16 Digital / DEC . . . . . 46
5.16.1 DEPCA, DE100/1, DE200/1/2, DE210, DE422 . . . . . 46
5.16.2 Digital EtherWorks 3 (DE203, DE204, DE205) . . . . . 46
5.16.3 DE425 EISA, DE434, DE435, DE500 . . . . . 47
5.16.4 DEC 21040, 21041, 2114x, Tulip . . . . . 47
5.17 Farallon . . . . . 48
5.17.1 Farallon Etherwave . . . . . 48
5.18 Fujitsu . . . . . 48
5.18.1 Fujitsu FMV-181/182/183/184 . . . . . 48
5.19 Hewlett Packard . . . . . 48
5.19.1 27245A . . . . . 49
5.19.2 HP EtherTwist, PC Lan+ (27247, 27252A) . . . . . 49
5.19.3 HP-J2405A . . . . . 49
5.19.4 HP-Vectra On Board Ethernet . . . . . 49
5.19.5 Schede HP 10/100 VG Any Lan (27248B, J2573, J2577, J2585, J970, J973) . . . . . 49
5.19.6 HP NetServer 10/100TX PCI (D5013A) . . . . . 50
5.20 IBM / International Business Machines . . . . . 50
5.20.1 IBM Thinkpad 300 . . . . . 50
5.20.2 IBM Credit Card Adaptor for Ethernet . . . . . 50
5.20.3 IBM Token Ring . . . . . 50
5.21 Schede Ethernet ICL . . . . . 51

5.21.1 ICL EtherTeam 16i/32 . . . 51
5.22 Schede Ethernet Intel . . . 51
5.22.1 Ether Express . . . 51
5.22.2 Ether Express PRO/10 . . . 51
5.22.3 Ether Express PRO/10 PCI (EISA) . . . 51
5.22.4 Ether Express PRO 10/100B . . . 52
5.23 Kingston . . . 52
5.24 LinkSys . . . 52
5.24.1 Schede LinkSys Etherfast 10/100 . . . 52
5.24.2 LinkSys Pocket Ethernet Adapter Plus (PEAEPP) . . . 52
5.24.3 LinkSys PCMCIA Adaptor . . . 53
5.25 Microdyne . . . 53
5.25.1 Microdyne Exos 205T . . . 53
5.26 Mylex . . . 53
5.26.1 Mylex LNE390A, LNE390B . . . 53
5.26.2 Mylex LNP101 . . . 53
5.26.3 Mylex LNP104 . . . 53
5.27 Novell Ethernet, NExxxx e cloni associati . . . 54
5.27.1 NE1000, NE2000 . . . 54
5.27.2 NE2000-PCI (RealTek/Winbond/Compex) . . . 54
5.27.3 NE-10/100 . . . 55
5.27.4 NE1500, NE2100 . . . 55
5.27.5 NE/2 MCA . . . 55
5.27.6 NE3200 . . . 55
5.27.7 NE3210 . . . 56
5.27.8 NE5500 . . . 56
5.28 Proteon . . . 56
5.28.1 Proteon P1370-EA . . . 56
5.28.2 Proteon P1670-EA . . . 56
5.29 Pure Data . . . 56
5.29.1 PDUC8028, PDI8023 . . . 56
5.30 Racal-Interlan . . . 56
5.30.1 ES3210 . . . 56
5.30.2 NI5010 . . . 57
5.30.3 NI5210 . . . 57
5.30.4 NI6510 (non EB) . . . 57
5.30.5 EtherBlaster (aka NI6510EB) . . . 57

5.31 RealTek . . . 57
5.31.1 Adattatore pocket RealTek RTL8002/8012 (AT-Lan-Tec) . . . 57
5.31.2 RealTek 8009 . . . 58
5.31.3 RealTek 8019 . . . 58
5.31.4 RealTek 8029 . . . 58
5.31.5 RealTek 8129/8139 . . . 58
5.32 Sager . . . 58
5.32.1 Sager NP943 . . . 58
5.33 Schneider & Koch . . . 59
5.33.1 SK G16 . . . 59
5.34 SEEQ . . . 59
5.34.1 SEEQ 8005 . . . 59
5.35 SMC (Standard Microsystems Corp.) . . . 59
5.35.1 WD8003, SMC Elite . . . 59
5.35.2 WD8013, SMC Elite16 . . . 60
5.35.3 SMC Elite Ultra . . . 60
5.35.4 SMC Elite Ultra32 EISA . . . 61
5.35.5 SMC EtherEZ (8416) . . . 61
5.35.6 SMC EtherPower PCI (8432) . . . 61
5.35.7 SMC EtherPower II PCI (9432) . . . 61
5.35.8 SMC 3008 . . . 62
5.35.9 SMC 3016 . . . 62
5.35.10 SMC-9000 / SMC 91c92/4 . . . 62
5.35.11 SMC 91c100 . . . 62
5.36 Texas Instruments . . . 62
5.36.1 ThunderLAN . . . 62
5.37 Thomas Conrad . . . 62
5.37.1 Thomas Conrad TC-5048 . . . 62
5.38 VIA . . . 62
5.38.1 VIA 86C926 Amazon . . . 63
5.38.2 VIA 86C100A Rhine II (and 3043 Rhine I) . . . 63
5.39 Western Digital . . . 63
5.40 Winbond . . . 63
5.40.1 Winbond 89c840 . . . 63
5.40.2 Winbond 89c940 . . . 63
5.41 Xircom . . . 63
5.41.1 Xircom PE1, PE2, PE3-10B* . . . 64

5.41.2 Schede PCMCIA Xircom . . . 64
5.42 Zenith . . . 64
5.42.1 Z-Note . . . 64
5.43 Znyx . . . 64
5.43.1 Znyx ZX342 (DEC 21040 based) . . . 64
5.44 Identificare una scheda sconosciuta . . . 64
5.44.1 Identificare il Network Interface Controller . . . 65
5.44.2 Identificare l'indirizzo Ethernet . . . 65
5.44.3 Suggerimenti per provare ad usare una scheda sconosciuta . . . 66
5.45 Driver per i dispositivi non Ethernet . . . 66

**6 Cavi, coassiali, doppini intrecciati 67**

6.1 Thin Ethernet (thinnet) . . . 67
6.2 Doppino intrecciato (twisted pair) . . . 67
6.3 Thick Ethernet . . . 68

**7 Configurazione del software e diagnotici 69**

7.1 Programmi di configurazione per le schede Ethernet . . . 69
7.1.1 Schede WD80x3 . . . 69
7.1.2 Schede Digital/DEC . . . 69
7.1.3 Schede NE2000+ o AT/LANTIC . . . 69
7.1.4 Schede 3Com . . . 70
7.2 Programmi diagnostici per schede Ethernet . . . 70

**8 Informazioni tecniche 70**

8.1 I/O programmato, memoria condivisa e DMA a confronto . . . 71
8.1.1 I/O programmato (Programmed I/O) (es. NE2000, 3c509) . . . 71
8.1.2 Memoria condivisa (Shared memory) (es. WD80x3, SMC-Ultra, 3c503) . . . 71
8.1.3 Accesso diretto alla memoria (DMA) di tipo slave (normale) (non per Linux!) . . . 71
8.1.4 Accesso diretto alla memoria (DMA) di tipo bus master (es. LANCE, DEC 21040) . . . 71
8.2 Scrivere un driver . . . 72
8.3 Interfaccia dei driver verso il kernel . . . 72
8.3.1 Probe (rilevamento) . . . 72
8.3.2 Interrupt handler (gestore dell'interrupt) . . . 73
8.3.3 Funzione transmit . . . 73
8.3.4 Funzione receive . . . 73
8.3.5 Funzione open . . . 73
8.3.6 Funzione close (opzionale) . . . 73

8.3.7 Funzioni varie . . . 73
8.4 Informazioni tecniche dalla 3Com . . . 73
8.5 Note sulle schede basate su PCnet / LANCE di AMD . . . 74
8.6 Multicast e modalità promiscua . . . 74
8.7 Berkeley Packet Filter (BPF) . . . 75

**9 Networking con computer portatili 76**
9.1 Usare SLIP . . . 76
9.2 Supporto PCMCIA . . . 76
9.3 Schede Ethernet ISA nella docking station . . . 77
9.4 Adattatori pocket (su porta parallela) . . . 77

**10 Miscellanea 77**
10.1 Passare al kernel argomenti Ethernet . . . 77
10.1.1 Il comando `ether` . . . 77
10.1.2 Il comando `reserve` . . . 78
10.2 Usare un driver Ethernet come modulo . . . 79
10.3 Documentazione correlata . . . 80
10.4 Liberatoria e copyright (in inglese) . . . 80
10.5 Chiusura . . . 81

# 1 Introduzione

Ethernet-Howto tratta delle schede che si dovrebbero e non si dovrebbero acquistare; di come configurarle, di come usarne più di una e di altri problemi e quesiti frequenti. Comprende informazioni dettagliate sull'attuale livello di supporto di tutte le più comuni schede Ethernet disponibili.

*Non* comprende l'aspetto software delle cose, che è trattato nel NET-3 Howto. Si noti anche che quesiti generali, non specifici su Linux, riguardanti Ethernet, non trovano (o almeno non dovrebbero trovare) risposta qui. Per quesiti di quel tipo, si vedano le eccellenti informazioni nelle FAQ di *comp.dcom.lans.ethernet*, che possono essere scaricate via FTP da `rtfm.mit.edu` come tutte le altre FAQ dei newsgroup.

Questa revisione tratta i kernel stabili fino alla versione 2.2.7 compresa.

Ethernet-Howto è di:

Paul Gortmaker, `p_gortmaker@yahoo.com`

Fonte principale di informazioni per la versione iniziale di Ethernet-Howto, disponibile esclusivamente in formato ASCII, è stato:

Donald J. Becker, `becker@cesdis.gsfc.nasa.gov`

che dovremmo ringraziare per aver scritto la grande maggioranza dei driver attualmente disponibili per Linux per le schede Ethernet. È anche l'autore dell'originario NFS server. Grazie Donald!

Questo documento è Copyright (c) 1993-1999 di Paul Gortmaker. Si vedano la liberatoria e le informazioni sulla copia alla fine di questo documento (10.4 (copyright)) per informazioni circa la ridistribuzione e le solite questioni legali "non siamo responsabili per ciò che riuscirai a rompere...".

## 1.1 Nuove versioni di questo documento

Nuove versioni di questo documento possono essere reperite all'indirizzo:

*Ethernet-HOWTO* `<http://metalab.unc.edu/mdw/HOWTO/Ethernet-HOWTO.html>`

o per chi desidera usare FTP e/o procurarsi formati non HTML:

*Sunsite HOWTO Archive* `<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/>`

Questo è il sito ufficiale, ma il documento può anche essere trovato nei diversi mirror WWW/ftp. Gli aggiornamenti vengono fatti appena nuove informazioni e/o driver diventano disponibili. Se la copia che si sta leggendo è vecchia di più di 6 mesi, si dovrebbe controllare per vedere se è disponibile una copia aggiornata.

Questo documento è disponibile in diversi formati (postscript, dvi, ASCII, HTML, ecc.). Personalmente consiglio di leggerlo in HTML (attraverso un browser WWW) o in formato Postscript/dvi. Entrambi contengono riferimenti incrociati che non sono inclusi nel formato ASCII.

## 1.2 Come usare Ethernet-Howto

Poiché questa guida sta diventando sempre più grande, probabilmente non si vuole sprecare il resto del pomeriggio leggendola per intero. E la buona notizia è che non la si *deve* leggere tutta. Le versioni HTML e Postscript/dvi hanno un indice che aiuterà senz'altro a trovare ciò di cui si ha bisogno molto più velocemente.

Può essere che si stia leggendo questo documento perché non si riesce a far funzionare le cose e non si sa cosa controllare o verificare. La sezione 1.3 (AIUTO – Non funziona!) è rivolta ai nuovi utenti di Linux e metterà nella direzione giusta.

Tipicamente gli stessi problemi e quesiti sono posti *più e più* volte da diverse persone. Può essere che il proprio problema specifico sia una delle Frequently Asked Questions (domande frequenti) e trova risposta nella sezione FAQ di questo documento (3 (Sezione FAQ)). Tutti dovrebbero dare un'occhiata a questa sezione prima di inviare una richiesta di aiuto.

Se non si possiede una scheda Ethernet, allora si dovrà in primo luogo scegliere una scheda (2 (Che scheda si dovrebbe acquistare...)).

Se si possiede già una scheda Ethernet, ma non si è sicuri di poterla usare con Linux, allora si dovrà leggere la sezione che contiene informazioni specifiche su ogni produttore e le relative schede (5 (Informazioni specifiche su...)).

Se si è interessati ad alcuni degli aspetti tecnici dei driver dei dispositivi per Linux, allora si può dare una scorsa alla sezione contenente questo tipo di informazioni (8 (Informazioni tecniche)).

## 1.3 AIUTO – Non funziona!

Okay, niente panico. Questa sezione vi condurrà per mano nel processo che consente di far funzionare le cose anche se non si hanno precedenti conoscenze di Linux o sull'hardware Ethernet.

La prima cosa da fare è scoprire il modello della propria scheda cosicché si possa determinare se Linux ha un driver per quella particolare scheda. Generalmente schede diverse sono controllate in modo diverso dal computer ospite e il driver per Linux (se ne esiste uno) contiene queste informazioni per il controllo in un

formato che permette a Linux di utilizzare la scheda. Se non si ha un manuale o qualcosa del genere che dia informazioni sul modello della scheda, allora si può provare la sezione di aiuto sulle schede misteriose (si veda la sezione 5.44 (Identificare una scheda sconosciuta)).

Ora che si sa che tipo di scheda si possiede, si leggano da cima a fondo i dettagli a essa relativi nella sezione sulle specifiche delle schede (5 (Informazioni specifiche su...)) che elenca in ordine alfabetico i produttori di schede, i numeri identificativi dei modelli e se c'è o meno un driver per Linux. Se è catalogata come "Non supportata" ci si può pressoché arrendere qui. Se non si riesce a trovare la propria scheda nell'elenco, si controlli per vedere se il suo manuale la cataloga come "compatibile" con un altro tipo di scheda conosciuto. Ci sono per esempio centinaia se non migliaia di schede diverse costruite per essere compatibili con il progetto originario NE2000 della Novell.

Assumendo che si sia scoperto che esiste un driver per Linux per la propria scheda, è ora necessario trovarlo e farne uso. Solo perché Linux ha un driver per la propria scheda ciò *non* significa che esso sia compreso in ogni kernel (il kernel è il nucleo del sistema operativo, la prima cosa caricata all'avvio e contiene, tra le altre cose, i driver per le diverse parti hardware). A seconda di chi ha prodotto la particolare distribuzione di Linux che si sta usando ci possono essere solo alcuni kernel precompilati e un grosso insieme di driver sotto forma di piccoli moduli separati, oppure un sacco di kernel, che coprono un enorme insieme di combinazioni di driver incorporati.

Molte distribuzioni di Linux adesso contengono un gruppo di piccoli moduli, i diversi driver. I moduli necessari tipicamente vengono caricati in un secondo tempo nel processo di avvio o su richiesta non appena serve un driver per accedere ad un particolare dispositivo. Occorrerà inserire questo modulo nel kernel dopo che è stato avviato. Si vedano le informazioni fornite con la propria distribuzione sull'installazione e l'uso dei moduli, oltre alla sezione sui moduli in questo documento (10.2 (Usare i driver Ethernet come moduli)).

Se non si è trovato né un kernel precompilato con il proprio driver, né il driver in forma modulare, è probabile che si possieda una scheda rara e si dovrà compilare il proprio kernel includendo il driver. Una volta installato Linux, la compilazione di un kernel su misura non è affatto difficile. Essenzialmente si risponde sì o no a cosa si vuole che il kernel contenga e poi gli si dice di compilarlo. Esiste un Kernel-Howto che aiuterà a far questo.

A questo punto si dovrebbe essere riusciti in qualche modo ad avviare un kernel con il proprio driver incorporato o a caricare il driver come modulo. Poiché circa la metà dei problemi che ha la gente è dovuta al non avere caricato il driver né in un modo né nell'altro, ora si potrebbe scoprire che le cose funzionano.

Se non funziona ancora allora è necessario verificare che il kernel stia effettivamente rilevando la scheda. Per fare questo, dopo che il sistema si è avviato e sono stati caricati tutti i moduli, fatto il login, digitare `dmesg | more`. Questo permetterà di rivedere i messaggi che il kernel ha fatto scorrere sullo schermo durante il processo di avvio. Se la scheda è stata rilevata si dovrebbe vedere da qualche parte in quell'elenco, un messaggio del driver della propria scheda che inizia con `eth0` e cita il nome del driver e i parametri hardware per i quali è stata configurata (configurazione degli interrupt, indirizzo delle porte di input/output, ecc.). Nota: Linux all'avvio elenca tutte le schede PCI installate nel sistema, senza badare ai driver disponibili, non si scambi questo per la rilevazione dei driver che avviene più tardi.

Se non si vede un messaggio di identificazione del driver di questo tipo, allora il driver non ha rilevato la propria scheda e questo è il motivo per il quale le cose non funzionano. Si vedano le FAQ (3 (Sezione FAQ)) per il da farsi se la propria scheda non viene rilevata. Nel caso si possieda una scheda NE2000 compatibile, nella sezione FAQ vi sono anche alcuni suggerimenti specifici per fare in modo che la scheda venga rilevata.

Se la scheda viene rilevata ma il messaggio di rilevamento riporta un errore di qualche tipo, come un conflitto di risorsa, probabilmente il driver non sarà inizializzato correttamente e la scheda continuerà a non essere utilizzabile. Anche i più comuni messaggi di errore di questo tipo sono elencati nella sezione FAQ insieme ad una soluzione.

Se il messaggio di rilevamento sembra corretto, confrontare bene le risorse della scheda riportate dal driver

con quelle per le quali la scheda è fisicamente configurata (attraverso dei piccoli ponticelli di colore nero sulla scheda o attraverso delle utilità software fornite dal produttore). Queste devono corrispondere esattamente. Per esempio se la scheda è configurata per IRQ 15 e il driver riporta nei messaggi di avvio IRQ 10, le cose non funzioneranno. La sezione FAQ tratta i casi più comuni di driver che non rilevano correttamente le informazioni di configurazione delle diverse schede.

A questo punto si è riusciti a far sì che la propria scheda sia rilevata con tutti i parametri corretti e, se tutto va bene, le cose funzionano. Altrimenti si ha o un errore di configurazione software o un errore di configurazione hardware. Un errore di configurazione software è il non configurare correttamente gli indirizzi di rete usando i comandi `ifconfig` e `route` e dettagli su come fare queste cose sono esaurientemente descritti nel Network HowTo e nella "Network Administrator Guide". Probabilmente entrambi si trovano nel CD-ROM che si è usato per l'installazione.

Un errore di configurazione hardware si ha quando un qualche conflitto di risorsa o errore di configurazione (che il driver non ha rilevato in fase di avvio) impedisce alla scheda di funzionare correttamente. Ciò può essere osservato in parecchie situazioni diverse. (1) Si ha un messaggio di errore quando `ifconfig` tenta di aprire il dispositivo per usarlo, del tipo "SIOCSFFLAGS: Try again". (2) Il driver riporta messaggi d'errore su `eth0` (li si può vedere usando `dmesg | more`) o strane incongruenze ogniqualvolta prova a mandare o ricevere dati. (3) Digitando `cat /proc/net/dev` appaiono numeri diversi da zero in una delle colonne errs, drop, fifo, frame o carrier corrispondenti a `eth0`. (4) Digitando `cat /proc/interrupts` appare un numero di interrupt nullo per la scheda. Anche la maggior parte dei tipici errori di configurazione hardware sono discussi nella sezione FAQ.

Bene, se si è arrivati a questo punto e le cose non funzionano ancora, si legga la sezione FAQ di questo documento, si legga la sezione sulle specifiche dei produttori che descrive la propria scheda, *e se ancora non funziona* allora si dovrebbe riccorrere all'invio di una richiesta di aiuto ad un opportuno newsgroup. Se si invia la richiesta, si descrivano dettagliatamente tutte le informazioni del caso tipo la marca della scheda, la versione del kernel, i messaggi del driver all'avvio, l'output di `cat /proc/net/dev`, una chiara descrizione del problema e naturalmente cosa si è già provato a fare per far funzionare le cose.

Sorprenderebbe sapere quante persone inviano cose inutili del tipo "Può aiutarmi qualcuno? La mia scheda Ethernet non funziona" e nient'altro. I lettori dei newsgroup tendono a ignorare queste richieste stupide, mentre una descrizione dettagliata del problema può consentire a un "Linux-guru" di individuare immediatamente il problema.

# 2 Che scheda si dovrebbe acquistare per Linux?

La risposta a questa domanda dipende molto da cosa si intende esattamente fare con la propria connessione di rete e quanto traffico dovrà sostenere.

Se si prevede un singolo utente che occasionalmente faccia una sessione FTP o una connessione WWW, allora probabilmente anche una vecchia scheda ISA a 8 bit farà al proprio caso.

Se si intende installare un server e si vuole che l'overhead della CPU per la trasmissione e ricezione dei dati sia mantenuto al minimo, probabilmente si deve considerare una delle schede PCI che usano un chip con capacità di bus-mastering, per esempio il chip tulip (21xxx) della DEC o il chip Pcnet-PCI della AMD.

Se si è in una situazione intermedia tra le due citate, una qualsiasi delle schede a basso costo PCI o ISA a 16 bit con driver stabili andrà bene.

## 2.1 E quali sono i driver stabili?

Per schede ISA a 16 bit, i seguenti driver sono piuttosto maturi e non si dovrebbero avere problemi se si acquista una scheda che li utilizza:

SMC-Ultra/EtherEZ, SMC-Elite (WD80x3), 3c509, Lance, NE2000.

Ciò non significa che tutti gli altri driver non siano stabili. Si dà il caso che quelli sopra siano i più vecchi e i più utilizzati fra tutti i driver per Linux e questo li rende l'alternativa più sicura.

Si noti che alcune schede madri economiche possono avere problemi con il bus-mastering fatto dalle schede Lance ISA e che alcuni cloni NE2000 possono avere problemi a essere rilevati all'avvio.

I driver PCI per Linux più comunemente usati sono probabilmente quelli per le Vortex/Boomerang (3c59x/3c9xx) della 3Com, il tulip (21xxx) della DEC e la EtherExpressPro 100 dell'Intel. Anche le diverse schede clone della NE2000 PCI sono estremamente comuni, ma l'acquisto di un clone di questa scheda non è raccomandato a meno che spendere il meno possibile non sia più importante che avere una scheda moderna concepita per elevate prestazioni.

## 2.2 Schede a 8 bit e schede a 16 bit a confronto

Probabilmente non è più possibile acquistare una scheda ISA a 8 bit nuova, ma per i prossimi anni ne salterranno fuori molte, e a prezzi molti bassi, ai raduni in cui avvengono scambi di pezzi per computer. Questo le renderà popolari per i sistemi "Ethernet casalinghi". Quanto sopra è valido adesso anche per le schede ISA a 16 bit visto che ora sono le schede PCI a essere molto comuni.

Alcune schede a 8 bit che forniscono adeguate prestazioni per un uso dal leggero al medio sono la wd8003, la 3c503 e la ne1000. La 3c501 fornisce prestazioni scadenti; queste povere reliquie dei tempi XT, vecchie di 12 anni, dovrebbero essere evitate (si mandino ad Alan, le colleziona...).

Il canale dati a 8 bit non nuoce così tanto alle prestazioni: ci si può aspettare di scaricare via ftp da un host veloce da 500 a 800kB/s con una scheda a 8 bit wd8003 (su un bus ISA veloce). Se la maggior parte del proprio traffico di rete è verso siti remoti allora il collo di bottiglia nel percorso sarà altrove e la sola differenza in velocità di cui ci si accorgerà è durante l'attività di rete nella propria sottorete.

## 2.3 Schede Ethernet (VLB/EISA/PCI) a 32 bit

Si noti che una rete a 10Mbps tipicamente non giustifica la richiesta di una interfaccia a 32 bit. Si veda 8.1 (I/O programmato vs. ...) sul perché avere una scheda Ethernet a 10Mbps su un bus ISA a 8 MHz realmente non è un collo di bottiglia. Anche se avere la scheda Ethernet su un bus veloce non significherà necessariamente trasferimenti più veloci, ciò di solito comporterà una riduzione del carico della CPU e questo è un vantaggio per sistemi multi utente.

Naturalmente per reti a 100Mbps, che sono attualmente una consuetudine, l'interfaccia a 32 bit è una cosa di cui non si può fare a meno per far uso dell'intera banda.

La AMD offre i chip a 32 bit PCnet-VLB e PCnet-PCI. Si veda 5.4.2 (AMD PCnet-32) per informazioni sulle versioni a 32 bit del chip LANCE / PCnet-ISA.

Il chip 21xxx PCI "tulip" della DEC è un'altra alternativa per i più esigenti (si veda 5.16.4 (DEC 21040)). Molti produttori realizzano schede che usano questo chip e il prezzo di queste schede senza nome è di solito abbastanza conveniente.

Un'altra possibilità sono le schede PCI 'Vortex' e 'Boomerang' della 3Com e il prezzo è abbastanza conveniente se si riesce a procurarsene una con un contratto di valutazione, finché dura (si veda 5.1.14 (3c590/3c595)).

Anche le schede PCI EtherExpress Pro 10/100 della Intel si dice funzionino bene con Linux (si veda 5.22.4 (EtherExpress))

Parecchi produttori di cloni hanno iniziato a fare NE2000 PCI basate su chip RealTek o Winbond. Anche queste schede sono supportate dal driver per Linux ne2000 per i kernel versione v2.0.31 e più recenti. Comunque si trae profitto solo dalla interfaccia bus più veloce visto che la scheda usa ancora l'antichissima interfaccia driver ne2000. Dalla versione v2.0.34 (e superiori) è disponibile anche un driver separato specifico per queste schede PCI `ne2k-pci.c` che è più efficiente del driver ISA `ne.c`.

## 2.4 Schede e driver a 100Mbps disponibili

La lista dell'hardware a 100Mbps attualmente supportato è la seguente: schede con chip DEC 21140; schede 3c595/3c90x Vortex; le EtherExpressPro10/100B; le PCnet-FAST; le SMC 83c170 (epic100) e le HP 100VG ANY-LAN.

Per ciascun tipo si dia un'occhiata alle informazioni specifiche sui produttori presenti in questo documento. Può essere necessario dare un'occhiata anche a uno di questi:

*Linux and 100Mbs Ethernet* <http://cesdis.gsfc.nasa.gov/linux/misc/100mbs.html>

*Donald's 100VG Page* <http://cesdis.gsfc.nasa.gov/linux/drivers/100vg.html>

*Dan Kegel's Fast Ethernet Page* <http://alumni.caltech.edu/dank/fe/>

## 2.5 100VG e 100BaseT a confronto

100BaseT è molto più importante di 100VG e il seguente messaggio promazionale di Donald tratto da una delle sue news informative più vecchie inviate su `comp.os.linux` riassume abbastanza bene la situazione: «Per coloro che non ne sono al corrente, esistono due standard Ethernet a 100Mbs in competizione, 100VG (conosciuto anche come 100baseVG e 100VG-AnyLAN) e 100baseT (con tipi di cavo 100baseTx, 100baseT4 e 100baseFx). Lo standard 100VG è entrato per primo sul mercato e penso sia stato strutturato meglio rispetto al 100baseTx. Facevo il tifo affinché vincesse ma certamente non vincerà. HP e altri hanno fatto parecchie scelte sbagliate:

1) Rinviando lo standard così da poter favorire IBM e supportare i frame token ring. Ciò sembrò una buona idea all'epoca visto che avrebbe consentito alle imprese che usavano token ring di aggiornare senza che i loro manager dovessero ammmettere di aver fatto un errore molto costoso commissionando la tecnologia sbagliata. Ma non ci si guadagnò nulla visto che i due tipi di frame non potevano coesistere su una rete, il token ring è un pantano di complessità e comunque IBM passò a 100baseT.

2) Producendo esclusivamente schede ISA e EISA (un modello PCI è stato annunciato solo recentemente). Il bus ISA è troppo lento per 100Mbs ed esistono relativamente poche macchine EISA. All'epoca VLB era comune, veloce e conveniente e PCI una alternativa fattibile. Ma il buon senso "tradizionalista" riteneva che i server si sarebbero fermati ai più costosi bus EISA.

3) Non inviandomi un databook. Sì questa mossa è stata la vera ragione del fallimento di 100VG :-). Ho richiesto ovunque informazioni di programmazione e tutto ciò che ho potuto ottenere è stata una brochure a colori dalla AT&T, di qualche pagina, su carta patinata, che descriveva quanto meraviglioso fosse il chipset Regatta.»

## 2.6 Tipi di cavo che la propria scheda dovrebbe supportare

Se si sta allestendo una piccola rete "personale", si dovrà probabilmente usare thinnet ovvero cavi Ethernet sottili. Cioè la versione con connettori standard BNC. La thinnet o cablaggio Ethernet sottile (cavo coassiale

RG-58) con connettori BNC (di metallo, da spingere e girare per chiudere) è chiamata tecnicamente 10Base2.

Stanno diventando comuni anche moltissime schede Ethernet in una versione "mista" (combo) per soli $10-$20 in più. Queste schede hanno incorporato sia il transceiver per il doppino intrecciato (twisted pair) che per thinnet, consentendo di cambiare idea in seguito.

Il cavo a doppino intrecciato e connettori RJ-45 (giant phone jack – connettore telefonico gigante) è chiamato tecnicamente 10BaseT. Lo si può sentir chiamare anche UTP (Unshielded Twisted Pair – doppino intrecciato non schermato).

La più datata thick (spessa) Ethernet (cavo coassiale da 10mm), che si trova solo nelle installazioni più vecchie, è chiamata 10Base5. La spina a 15 pin, a forma di lettera D, che si trova su alcune schede Ethernet (il connettore AUI) è usato per connettere transceiver esterni e thick Ethernet.

Installazioni aziendali estese useranno probabilmente 10BaseT piuttosto che 10Base2. 10Base2 non offre alcuna possibilità di aggiornamento a una 100Base-qualsiasi.

Si veda 6 (Cavi, coassiali,...) per altre informazioni riguardanti i diversi tipi di cavi Ethernet.

# 3 Domande frequenti

Ecco alcuni dei quesiti posti più frequentemente a proposito dell'uso di Linux con una connessione Ethernet. Alcuni dei quesiti più specifici sono classificati in "base al costruttore". Può essere che il quesito per il quale si ha bisogno di una risposta sia già stato posto da qualcun altro (e abbia trovato risposta!), perciò anche se non si trova la risposta qui, probabilmente si può trovare ciò che di cui si ha bisogno in un archivio di news come *Dejanews* `<http://www.dejanews.com>`.

## 3.1 Driver alpha – come procurarseli e come usarli

Ho sentito che è disponibile un driver aggiornato oppure un driver preliminare o alpha (sperimentale) per la mia scheda. Dove posso procurarmelo?

Il "più nuovo dei nuovi" driver può essere trovato sul sito FTP di Donald: `cesdis.gsfc.nasa.gov` nell'area `/pub/linux/`. Qui le cose cambiano abbastanza frequentemente perciò si cerchi bene. In alternativa, per localizzare il driver che si sta cercando, può essere più facile usare un browser WWW all'indirizzo:

*Don's Linux Home Page* `<http://cesdis.gsfc.nasa.gov/linux/>`

(ci si guardi dai browser WWW che fanno il "munge" (NdT munge: trasformare in modo imperfetto le informazioni [dal Jargon Lexicon]) del sorgente sostituendo i TAB con spazi e così via – se si è incerti usare ftp o almeno un URL FTP per scaricare).

Ora, se il driver è realmente un alpha o pre-alpha, lo si tratti come tale. In altre parole, non si reclami perché non si capisce cosa farne. Se non si riesce a capire come installarlo allora probabilmente non lo si dovrebbe provare. Anche se mette fuori uso la propria macchina, non si reclami. Si mandi invece un rapporto ben documentato del bug o, ancora meglio, una patch!

Si noti che alcuni dei driver sperimentali/alpha "utilizzabili" sono stati inclusi nell'albero dei sorgenti del kernel standard. Una delle prime cose che verranno chieste quando si esegue `make config` è "Prompt for development and/or incomplete code/drivers" (proposta per il codice o driver in fase di sviluppo o incompleti). Si dovrà rispondere "Y" (sì) per richiedere l'inclusione di un qualche driver alpha/sperimentale.

## 3.2 Come usare più di una scheda Ethernet per macchina

Cosa è necessario fare affinché Linux possa utilizzare due schede Ethernet?

La risposta a questo quesito dipende se si sta/stanno usando il/i driver come modulo caricabile o direttamente compilato nel kernel. Adesso moltissime distribuzioni di Linux usano driver modulari. Ciò evita di dover distribuire un mucchio di kernel ciascuno contenente un insieme diverso di driver. Si usa invece un singolo kernel di base e i driver necessari per un particolare sistema utente vengono caricati una volta che l'avvio del sistema è arrivato al punto tale da poter accedere ai file dei moduli dei driver (memorizzati di solito in `/lib/modules/`).

*Con il driver come modulo:* Nel caso di driver PCI, in genere il modulo rileva automaticamente tutte le schede installate di quello specifico modello. Comunque il rilevamento di una scheda non è una operazione sicura per schede ISA, perciò di solito è necessario fornire l'indirizzo base di I/O della scheda affinché il modulo sappia dove guardare. Questa informazione è memorizzata nel file `/etc/conf.modules`.

Come esempio si consideri un utente che ha due schede ISA NE2000, una a `0x300` ed una a `0x240` e le righe da mettere nel file `/etc/conf.modules`:

```
alias eth0 ne
alias eth1 ne
options ne io=0x240,0x300
```

Come agisce: se l'amministratore (o il kernel) fa un `modprobe eth0` oppure un `modprobe eth1` allora dovrebbe essere caricato il driver `ne.o` sia per `eth0` che per `eth1`. Inoltre quando è caricato il modulo `ne.o`, dovrebbe esserlo con le opzioni `io=0x240,0x300` cosicché il driver sa dove cercare le schede. Si noti che `0x` è importante, cose del tipo `300h`, comunemente usate nel mondo DOS, non funzionano. Cambiando l'ordine di `0x240` e `0x300` si cambierà quale scheda fisica finirà in `eth0` e quale in `eth1`.

Per gestire più schede, molti driver modulari ISA possono accettare diversi valori di I/O separati da virgole, come in questo esempio. Comunque, alcuni driver (più vecchi?), come il modulo 3c501.o, attualmente sono in grado di gestire solo una scheda per modulo caricato. In questo caso si può caricare il modulo due volte per far sì che entrambe le schede siano rilevate. Il file `/etc/conf.modules` dovrebbe presentarsi così:

```
alias eth0 3c501
alias eth1 3c501
options eth0 -o 3c501-0 io=0x280 irq=5
options eth1 -o 3c501-1 io=0x300 irq=7
```

In questo esempio l'opzione `-o` è stata usata per dare a ogni istanza del modulo un nome univoco, poiché non è possibile caricare due moduli con lo stesso nome. È anche usata l'opzione `irq=` per specificare la configurazione IRQ hardware della scheda (questo metodo può essere usato anche con moduli che accettano valori di I/O separati da virgole, ma è meno efficiente poiché il modulo finisce per essere caricato due volte anche se non sarebbe realmente necessario).

Come esempio finale, si consideri un utente con una scheda 3c503 a `0x350` e una SMC Elite16 (wd8013) a `0x280`. Avranno:

```
alias eth0 wd
alias eth1 3c503
options wd io=0x280
options 3c503 io=0x350
```

Per le schede PCI, tipicamente sono necessarie solamente le righe `alias` per correlare le interfacce `ethN` con l'appropriato nome del driver, poiché l'I/O base di una scheda PCI può essere rilevato in modo sicuro.

I moduli disponibili sono tipicamente memorizzati in `/lib/modules/‘uname -r‘/net` dove il comando `uname -r` fornisce la versione del kernel (es. 2.0.34). Si può guardare là per vedere quale corrisponde alla propria

scheda. Una volta che si hanno le impostazioni corrette nel proprio file `conf.modules`, si può collaudare il tutto con:

```
modprobe ethN
dmesg | tail
```

dove 'N' è il numero dell'interfaccia Ethernet che si sta collaudando.

*Con il driver compilato nel kernel:* Se si ha il driver compilato nel kernel, allora tutto quel che serve per usare più schede Ethernet lo si ha già. Comunque si noti che al momento, per default, solo *una* scheda Ethernet è rilevata automaticamente. Ciò aiuta a evitare possibili blocchi all'avvio causati dal rilievo di schede "ombrose".

(Nota: dagli ultimi kernel 2.1.x, i rilievi al boot sono stati separati in sicuri ed insicuri, in modo che tutti quelli sicuri (es. PCI e EISA) trovino automaticamente tutte le loro schede. Nei sistemi con più di una scheda Ethernet e almeno una scheda ISA, sarà ancora necessario fare una delle cose che seguono.)

Ci sono due modi per abilitare l'auto-rilevamento della seconda (e la terza, e...) scheda. Il metodo più semplice è di passare al momento dell'avvio, degli argomenti al kernel, solitamente usando LILO. Il rilevamento della seconda scheda può essere ottenuto con un semplice argomento di boot come `ether=0,0,eth1`. In questo caso `eth0` e `eth1` saranno assegnate nell'ordine in cui sono rilevate le schede al boot. Diciamo che si vuole che la scheda a `0x300` sia `eth0` e quella a `0x280` sia `eth1`, allora si potrebbe usare

```
LILO: linux ether=5,0x300,eth0 ether=15,0x280,eth1
```

Il comando `ether=` accetta più della terna IRQ, I/O e nome mostrata sopra. Si veda la sezione 10.1 (Passare argomenti Ethernet...) per la sintassi completa, i parametri specifici delle schede e dritte su LILO.

Questi argomenti di boot possono essere resi permanenti cosicché non si debba reinserirli ogni volta. Si veda l'opzione di configurazione di LILO `append` nel manuale di LILO.

Il secondo metodo (non raccomandato) è di modificare il file `Space.c` e sostituire la voce `0xffe0` per l'indirizzo I/O con uno zero. La voce `0xffe0` dice di non cercare quel dispositivo – rimpiazzandola con uno zero si abiliterà l'autorilevamento di quel dispositivo.

Si noti che, se si intende usare Linux come gateway tra due reti, si dovrà ricompilare un kernel abilitando l'IP forwarding (inoltro degli IP). Solitamente usare un vecchio AT/286 con un software del tipo "kbridge" è una soluzione migliore.

Se si sta leggendo questa cosa mentre si *naviga in rete*, si potrebbe guardare il mini-howto che Donald ha nel suo sito WWW. Si veda *Multiple Ethercards* <http://cesdis.gsfc.nasa.gov/linux/misc/multicard.html>.

## 3.3 Il comando `ether=` non è servito a niente. Perché?

Come descritto sopra, il comando `ether=` funziona *solo* per driver compilati nel kernel. Adesso molte distribuzioni usano i driver in forma modulare, perciò il comando `ether=` è usato ancora raramente (certa documentazione più vecchia non è ancora stata aggiornata per riportare questo cambiamento). Se si vogliono adoperare le opzioni per un driver Ethernet modulare si *devono* fare modifiche al file `/etc/conf.modules`.

Se si sta usando un driver compilato nel kernel e si è aggiunto un comando `ether=` al proprio file di configurazione LILO, si noti che esso non avrà effetto fino a che non si riesegue `lilo` per eseguire il file di configurazione aggiornato.

## 3.4 Problemi con schede NE1000/NE2000 (e cloni)

**Problema:** Una scheda clone PCI NE2000 non è rilevata all'avvio usando una versione del kernel v2.0.x.

**Causa:** Il driver `ne.c`, fino alla versione del kernel v2.0.30, conosce solo il numero identificativo PCI delle schede clone basate su RealTek 8029. Da allora, molti altri hanno distribuito cloni PCI NE2000 con diversi numeri identificativi PCI, perciò il driver non li rileva.

**Soluzione:** La soluzione più facile consiste nell'aggiornare il kernel di Linux alla versione v2.0.31 (o più recente). Essa è a conoscenza dei numeri identificativi di circa cinque diversi chip PCI NE2000 e li rileva automaticamente all'avvio o nella fase di caricamento del modulo. Se si aggiorna alla versione 2.0.34 (o più recente) esiste un driver specifico NE2000 solo PCI che è leggermente più piccolo e più efficiente del driver originario ISA/PCI.

**Problema:** Una scheda clone PCI NE2000 si presenta come una ne1000 (scheda a 8 bit!) all'avvio o quando si carica il modulo ne.o per la versione v2.0.x, perciò non funziona.

**Causa:** Alcuni cloni PCI non implementano l'accesso byte wide (perciò non sono realmente compatibili NE2000 al 100%). Ciò fa pensare al sistema che siano schede NE1000.

**Soluzione:** È necessario aggiornare il kernel alla versione v2.0.31 (o più recente) come descritto sopra. Adesso il driver controlla che non si verifichi questo problema hardware.

**Problema:** Una scheda PCI NE2000 ha prestazioni orribili, anche se si riduce la dimensione della finestra come descritto nella sezione 4 (Suggerimenti per le prestazioni).

**Causa:** Le specifiche per il chip 8390 originale, progettato e commercializzato più di dieci anni fa, osservavano che per ottenere la massima affidabilità, era necessaria una lettura fittizia dal chip prima di ogni operazione di scrittura. Il driver ha i mezzi per farlo ma è stato disabilitato per default sin dai tempi dei kernel v1.2. Un utente ha riferito che riabilitare questa 'mis-feature' ha migliorato le prestazioni ottenute con un clone economico PCI NE2000.

**Soluzione:** Visto che questa soluzione è stata riportata da una sola persona, non ci si illuda troppo. La riabilitazione della lettura prima della scrittura si ottiene semplicemente modificando il file del driver in `linux/drivers/net/`, togliendo il commento alla riga contenente `NE_RW_BUGFIX` e poi ricompilando il kernel o il modulo come al solito. Se funziona si invii una e-mail che descrive la differenza di prestazioni e il tipo di scheda/chip che si possiede (la stessa cosa può essere fatta anche per il driver `ne2k-pci.c`).

**Problema:** Il driver `ne2k-pci.c`, con una scheda PCI NE2000, riporta messaggi di errore del tipo `timeout waiting for Tx RDC` e non funziona bene.

**Causa:** La propria scheda e/o il collegamento tra la scheda e il bus PCI non può gestire l'ottimizzazione I/O a long word usata in questo driver.

**Soluzione:** Prima di tutto, si controllino le impostazioni disponibili nel BIOS/CMOS setup per vedere se alcune di quelle correlate alla temporizzazione del bus PCI, sono troppo stringenti per ottenere operazioni affidabili. Altrimenti, l'uso del driver ISA/PCI `ne.c` (o la rimozione di `#define USE_LONGIO` dal driver `ne2k-pci.c`) dovrebbe permettere di usare la scheda.

**Problema:** Una scheda ISA Plug and Play NE2000 (per esempio RealTek 8019) non è rilevata.

**Causa:** Le specifiche originarie NE2000 (e perciò il driver per Linux NE2000) non hanno il supporto per il Plug and Play.

**Soluzione:** Si usi il disco di configurazione DOS fornito con la scheda per disabilitare PnP e per assegnare la scheda ad uno specifico indirizzo di I/O e IRQ. Si aggiunga una riga a `/etc/conf.modules` del tipo `options ne io=0xNNN` dove `0xNNN` è l'indirizzo di I/O in formato esadecimale a cui la scheda è stata assegnata (ciò assume che si stia usando un driver modulare; se non è così si usi all'avvio un argomento `ether=0,0xNNN,eth0`). Può accadere anche che si debba entrare nel BIOS/CMOS setup e contrassegnare

l'IRQ come Legacy-ISA al posto di PnP. In alternativa, se è necessario lasciare PnP abilitato per la compatibilità con qualche altro sistema operativo, si consideri il pacchetto *isapnptools*. Si provi `man isapnp` per capire se è già installato nel proprio sistema. Se non lo è, si dia un'occhiata al seguente URL:

*ISA PNP Tools* `<http://www.roestock.demon.co.uk/isapnptools/>`

**Problema:** Un driver NE*000 riporta 'not found (no reset ack)' durante il rilevamento all'avvio.

**Causa:** Ciò è collegato alla modifica vista in precedenza. Dopo la verifica iniziale che un 8390 è all'indirizzo di I/O rilevato, si effettua la riinizializzazione (reset). Quando la scheda ha completato tale operazione, si suppone dichiari (acknowlodge) che è completata. La propria scheda non lo fa e così il driver assume che nessuna scheda NE sia presente.

**Soluzione:** Si può dire al driver che si possiede una scheda scadente specificando al momento dell'avvio il valore esadecimale `0xbad`, solitamente non usato, per `mem_end` (NdT: si noti che "bad" significa letteralmente "cattivo, brutto, scarso, ecc."). Quando si usa `0xbad`, si *deve* anche fornire un I/O base diverso da zero per la scheda. Per esempio, una scheda a `0x340` che non dichiara il completamento del reset dovrebbe usare qualcosa del tipo:

```
LILO: linux ether=0,0x340,0,0xbad,eth0
```

Ciò permette che il rilevamento della scheda continui anche se la propria scheda non effettua l'ACK del reset. Se si sta usando il driver come modulo, allora si può fornire l'opzione `bad=0xbad` proprio come si fornisce l'indirizzo di I/O.

**Problema:** Una scheda NE*000 blocca la macchina al primo accesso alla rete.

**Causa:** Questo problema è stato riportato per kernel a partire dalla versione 1.1.57 fino alla corrente. Sembra confinato a poche schede clone configurabili via software. Apparentemente sie aspettano di essere inizializzate in qualche modo speciale.

**Soluzione:** Diverse persone hanno riferito che l'esecuzione del programma DOS di configurazione software e/o l'uso del driver per DOS forniti con la scheda prima di fare il boot "a caldo" di Linux (cioé usando loadlin o con il "saluto a tre dita" (CTRL-ALT-CANC)) consente alla scheda di funzionare. Ciò indicherebbe che queste schede necessitano di essere inizializzate in un modo particolare, leggermente diverso da ciò che fa l'attuale driver per Linux.

**Problema:** Una scheda Ethernet NE*000 a `0x360` non viene rilevata.

**Causa:** La propria scheda ha uno spazio degli indirizzi di I/O ampio `0x20`, il che la fa entrare in collisione con la porta parallela a `0x378`. Altri dispositivi che possono essere lì sono il secondo controller del floppy (se in dotazione) a `0x370` e il controller secondario IDE a `0x376-0x377`. Se la/le porta/e sono già assegnate ad un altro driver, il kernel non permette che avvenga il rilevamento.

**Soluzione:** Si sposti la propria scheda ad un indirizzo del tipo `0x280, 0x340, 0x320` o si compili senza il supporto per la stampante su porta parallela.

**Problema:** La rete "scompare" ogniqualvolta si stampa qualcosa (NE2000).

**Causa:** Il problema è lo stesso visto sopra, ma si ha un kernel più vecchio che non verifica la sovrapposizione delle regioni di I/O. Si usi la stessa soluzione vista prima e ancora meglio ci si procuri un nuovo kernel.

**Problema:** NE*000 ethercard probe at 0xNNN: 00 00 C5 ... not found. (invalid signature yy zz)

**Causa:** Prima di tutto, si ha una scheda NE1000 o NE2000 all'indirizzo 0xNNN? Se sì, l'indirizzo hardware riportato ha l'aria di essere uno valido? Se sì, allora si possiede un clone NE*000 disgraziato. Si suppone che tutti i cloni NE*000 abbiano il valore `ox57` nei byte 14 e 15 della PROM SA sulla scheda. La propria scheda non ce l'ha – essa ha invece 'yy zz'.

**Soluzione:** Ci sono due modi di aggirare l'ostacolo. Il più facile consiste nell'usare un valore di mem_end `0xbad` come descritto sopra per il problema 'no reset ack'. Ciò consentirà di bypassare il controllo della firma, sempre che si fornisca anche un valore I/O base diverso da zero. Questa soluzione non richiede di ricompilare il kernel.

Il secondo metodo (per hacker) comporta la modifica delle stesso driver e la ricompilazione del proprio kernel (o modulo). Il driver (usr/src/linux/drivers/net/ne.c) contiene un elenco "Hall of Shame" (Ndt: gioco di parole con "Hall of Fame"; "fame"= famoso, "shame"=vergogna, disonore) pressappoco alla riga 42. Questo elenco è usato per rilevare cloni disgraziati. Per esempio, le schede DFI usano "DFI" nei primi 3 byte della PROM al posto di usare `0x57` nei byte 14 e 15 (come dovrebbero fare).

**Problema:** La macchina si blocca durante l'avvio giusto dopo il messaggio '8390...' oppure 'WD....'. La rimozione della NE2000 risolve il problema.

**Soluzione:** Si cambi l'indirizzo base della propria NE2000 con qualcosa del tipo `0x340`. In alternativa si può usare l'argomento di boot "reserve=" in combinazione con l'argomento "ether=" per tutelare la scheda da rilievi di altri driver di dispositivi.

**Causa:** Il proprio clone NE2000 non è un clone abbastanza buono. Una NE2000 effettiva è un abisso senza fondo che intrappola ogni driver che stia facendo l'autorilevamento nel suo spazio. Spostare la NE2000 ad un indirizzo meno popolare la porta fuori dalla portata di altri rilievi automatici, consentendo alla macchina di avviarsi.

**Problema:** La macchina si blocca all'avvio durante il rilevamento SCSI.

**Causa:** È lo stesso problema visto precedentemente, si cambi l'indirizzo della scheda Ethernet o si usino gli argomenti di boot reserve/ether.

**Problema:** La macchina si blocca all'avvio durante il rilevamento della scheda sonora.

**Causa:** No, in realtà ciò avviene durante il rilevamento SCSI "silenzioso" ed è lo stesso problema visto precedentemente.

**Problema:** NE2000 non rilevata all'avvio – nessun tipo di messaggio.

**Soluzione:** Non esiste una "soluzione magica" visto che possono essere parecchie le cause per cui non è stata rilevata. Il seguente elenco dovrebbe aiutare a risolvere i possibili problemi.

1) Si compili un nuovo kernel che includa solo i driver dei dispositivi di cui si ha bisogno. Si verifichi che si stia davvero avviando il kernel nuovo. Dimenticare di eseguire lilo, eccetera può portare ad avviare quello vecchio (si guardi attentamente l'ora e la data di compilazione riportati all'avvio). Suona ovvio, ma l'abbiamo fatto tutti in passato. Ci si assicuri che il driver sia effettivamente incluso nel nuovo kernel cercando nel file `System.map` cose tipo `ne_probe`.

2) Si guardino attentamente i messaggi all'avvio. Davvero non accenna mai al fatto che sta facendo un rilevamento ne2k, per esempio 'NE*000 probe at 0xNNN: not found (blah blah)' o fallisce proprio silenziosamente? C'è una grossa differenza. Si usi `dmesg|more` per rivedere i messaggi di avvio dopo aver fatto il login o si digiti Shift-PgUp per scorrere il contenuto dello schermo dopo che l'avvio si è completato e appare il prompt del login.

3) Dopo l'avvio si esegua un `cat /proc/ioports` e si verifichi che l'intero spazio di I/O richiesto dalla scheda sia libero. Se si parte da `0x300`, il driver n2ek richiederà `0x300-0x31f`. Se un qualsiasi altro driver di dispositivo ha occupato anche solo una porta da qualche parte in quell'intervallo, il rilevamento non avrà luogo a quell'indirizzo e continuerà silenziosamente al prossimo degli indirizzi rilevati. Un caso frequente è avere il driver lp che riserva `0x378` o il secondo canale IDE che riserva `0x376`, il che impedisce al driver ne il rilevamento in `0x360-0x380`.

4) Allo stesso modo come sopra per `cat /proc/interrupts`. Ci si assicuri che nessun altro dispositivo abbia

occupato l'interrupt per il quale è stata impostata la scheda. In questo caso, il rilevamento avviene e il driver Ethernet protesta rumorosamente all'avvio perché non riesce a ottenere la linea IRQ desiderata.

5) Se si è ancora perplessi del fallimento silenzioso del driver, allora lo si modifichi aggiungendo alcuni printk() al rilevamento. Per esempio con il ne2k si potrebbero aggiungere/rimuovere righe (contrassegnate rispettivamente con un '+' o '-') in `linux/drivers/net/ne.c` tipo:

```
    int reg0 = inb_p(ioaddr);

+    printk("NE2k probe - now checking %x\n",ioaddr);
-    if (reg0 == 0xFF)
+    if (reg0 == 0xFF) {
+       printk("NE2k probe - got 0xFF (vacant I/O port)\n");
        return ENODEV;
+    }
```

Perciò esso produrrà messaggi di output per ogni indirizzo di porta che esamina e si potrà capire se l'indirizzo della propria scheda è stato o no rilevato.

6) Ci si può anche procurare il diagnostico ne2k nel sito ftp di Don (pure citato nell'howto) e vedere se è in grado di rilevare la propria scheda dopo che si è avviato Linux. Si usi l'opzione '`-p 0xNNN`' per dirgli dove cercare la scheda (l'indirizzo di default è `0x300` e non va a guardare da nessun'altra parte a differenza del rilevamento all'avvio). L'output risultante quando trova una scheda sarà qualcosa del tipo:

```
Checking the ethercard at 0x300.
  Register 0x0d (0x30d) is 00
  Passed initial NE2000 probe, value 00.
8390 registers: 0a 00 00 00 63 00 00 00 01 00 30 01 00 00 00 00
SA PROM  0: 00 00 00 00 c0 c0 b0 b0 05 05 65 65 05 05 20 20
SA PROM 0x10: 00 00 07 07 0d 0d 01 01 14 14 02 02 57 57 57 57

NE2000 found at 0x300, using start page 0x40 and end page 0x80.
```

I propri valori register e PROM saranno probabilmente diversi. Si noti che tutti i valori PROM sono duplicati per una scheda a 16 bit, l'indirizzo Ethernet (00:00:c0:b0:05:65) appare nella prima riga e la firma ripetuta `0x57` appare alla fine della PROM.

L'output risultante quando non c'è nessuna scheda installata a `0x300` sarà qualcosa del tipo:

```
Checking the ethercard at 0x300.
  Register 0x0d (0x30d) is ff
  Failed initial NE2000 probe, value ff.
8390 registers: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
SA PROM      0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
SA PROM 0x10: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

 Invalid signature found, wordlength 2.
```

I valori `0xff` si presentano poiché quello è il valore restituito quando si legge una porta di I/O libera. Si possono vedere dei valori diversi da `0xff` anche se si ha qualche altro tipo di hardware nella regione rilevata.

7) Si provi a fare un warm boot di Linux da un dischetto di avvio per DOS (attraverso loadlin) dopo aver eseguito il driver per DOS o il programma di configurazione forniti. Può essere che faccia una qualche 'magia' extra (cioé non standard) per inizializzare la scheda.

8) Si provi il packet driver di Russ Nelson ne2000.com per vedere se almeno questo riesce a rilevare la propria scheda – se no, allora le cose non vanno bene. Esempio:

```
A:> ne2000 0x60 10 0x300
```

Gli argomenti sono: il vettore degli interrupt software, l'IRQ hardware e l'I/O base. Lo si può trovare in un qualsiasi archivio msdos nel file pktdrv11.zip. La versione attuale può essere più recente della 11.

## 3.5 Problemi con le schede SMC Ultra/EtherEZ e WD80*3

**Problema:** Si ottengono messaggi tipo il seguente:

```
eth0: bogus packet size: 65531, status=0xff, nxpg=0xff
```

**Causa:** C'è un problema di memoria condivisa.

**Soluzione:** La causa più comune è dovuta a macchine PCI che non sono configurate per mappare dispositivi nella memoria ISA. Perciò si finisce per leggere la RAM del PC (tutti valori `0xff`) invece della RAM della scheda che contiene i dati provenienti dal pacchetto ricevuto.

Altri tipici problemi facili da risolvere sono: i conflitti di scheda, avere la cache o la "shadow ROM" abilitata per quella regione o avere il proprio bus ISA che va ad una velocità superiore agli 8 MHz. Si trovano anche un numero sorprendente di guasti di memoria sulle schede Ethernet, perciò si esegua un programma di diagnostica nel caso in cui se ne possieda uno per la propria scheda.

**Problema:** La scheda SMC EtherEZ non funziona nella modalità a memoria non condivisa (PIO).

**Causa:** Le versioni più vecchie del driver Ultra supportavano la scheda solo nella modalità a memoria condivisa.

**Soluzione:** Il driver a partire dalla versione del kernel 2.0 supporta anche la modalità ad I/O programmato. Si aggiorni alla versione v2.0 o più recente.

**Problema:** Le vecchie wd8003 e/o le wd8013 configurabili con i ponticelli ottengono sempre l'IRQ sbagliato.

**Causa:** Le vecchie schede wd8003 e i cloni wd8013 configurabili con i ponticelli non possiedono la EEPROM dalla quale il driver può leggere l'impostazione dell'IRQ. Se il driver non è in grado di leggere l'IRQ allora esso prova a scoprirlo automaticamente con auto-IRQ. Se l'auto-IRQ restituisce il valore zero, il driver non fa altro che assegnare IRQ 5 a una scheda a 8 bit o IRQ 10 a una scheda a 16 bit.

**Soluzione:** Si eviti il codice di auto-IRQ e si comunichi al kernel qual è l'IRQ per il quale la scheda è stata configurata nel proprio file di configurazione dei moduli (o mediante una argomento di boot per i driver compilati nel kernel).

**Problema:** La scheda SMC Ultra è rilevata come wd8013, ma l'IRQ e l'indirizzo base della memoria condivisa sono sbagliati.

**Causa:** La scheda Ultra assomiglia molto a una wd8013 e se il driver Ultra non è presente nel kernel, il driver wd può scambiare la Ultra per una wd8013. Il rilevamento della Ultra avviene prima del rilevamento della wd perciò questo di solito non accade. La Ultra memorizza l'IRQ e l'indirizzo base della memoria in un modo diverso nella EPROM, da cui i valori contraffatti riportati.

**Soluzione:** Si ricompili il kernel con solo i driver di cui si ha bisogno. Se si ha una commistione di schede ultra e wd nella stessa macchina e si stanno usando i moduli allora si carichi per primo il modulo ultra.

## 3.6 Problemi con le schede 3Com

**Problema:** La 3c503 si prende l'IRQ N, ma questo serve per una qualche altro dispositivo che ha bisogno dell'IRQ N (per esempio il driver del CDROM, il modem, ecc.). Si può risolvere la cosa senza ricompilare il kernel?

**Soluzione:** Il driver della 3c503 cerca una linea di IRQ libera nell'ordine {5, 9/2, 3, 4} e dovrebbe scegliere una linea che nessuno sta usando. Il driver decide quando la scheda è attivata con `ifconfig`.

Se si sta usando un driver modulare, si possono usare dei parametri del modulo per impostare molte cose, incluso il valore dell'IRQ.

Ciò che segue imposta l'IRQ9, la locazione base `0x300`, <valori ignorati> e if_port #1 (il transceiver esterno).

```
io=0x300 irq=9 xcvr=1
```

In alternativa, se il driver è compilato nel kernel, è possibile fissare gli stessi valori all'avvio passando i parametri attraverso LILO.

```
LILO: linux ether=9,0x300,0,1,eth0
```

Ciò che segue imposta l'IRQ3, la ricerca della locazione base, <valori ignorati> e il transceiver di default if_port #0 (il transceiver interno).

```
LILO: linux ether=3,0,0,0,eth0
```

**Problema:** 3c503: configured interrupt X invalid, will use autoIRQ.

**Causa:** La sheda 3c503 può usare solo uno degli IRQ{5, 2/9, 3, 4} (solo queste linee sono connesse alla scheda). Se si passa un valore di IRQ che non appartiene al precedente insieme, si otterrà il messaggio sopra indicato. Di solito non è necessario specificare un valore di interrupt per la 3c503. La 3c503 effettuerà l'autoIRQ quando viene usato ifconfig e sceglie uno degli IRQ{5, 2/9, 3, 4}.

**Soluzione:** Si usi uno degli IRQ validi elencati sopra oppure si abiliti l'autoIRQ ma senza specificare la linea IRQ in alcun modo.

**Problema:** I driver per la 3c503 forniti non utilizzano la porta AUI (thicknet). Come si può optare per essa al posto della porta thinnet di default?

**Soluzione:** La porta 3c503 AUI può essere selezionata al momento dell'avvio per i driver compilati nel kernel e all'inserzione del modulo per i driver modulari. La selezione avviene impostando ad 1 il bit meno significativo della variabile attualemente non usata dev->rmem_start, cosicché un parametro all'avvio tipo:

```
LILO: linux ether=0,0,0,1,eth0
```

dovrebbe funzionare per i driver compilati nel kernel.

Per specificare la porta AUI quando si carica il driver come modulo è sufficiente aggiungere `xcvr=1` alla riga delle opzioni del modulo insieme con i propri valori di I/O e IRQ.

## 3.7 FAQ non specifiche su una particolare scheda

### 3.7.1 Linux e schede Ethernet ISA di tipo Plug and Play

Per ottenere i migliori risultati (e il minimo rompimento) si raccomanda di usare il programma (di solito DOS) fornito con la propria scheda per disabilitare il meccanismo PnP e impostarla definitivamente a un

indirizzo di I/O e un IRQ. Ci si assicuri che l'indirizzo di I/O che si utilizza sia rilevato all'avvio o, se si usano i moduli, si fornisca l'indirizzo sotto forma di opzione `io=` in `/etc/conf.modules`. Può darsi che si debba anche entrare nel BIOS/CMOS setup e contrassegnare l'IRQ come Legacy-ISA al posto di PnP (se il proprio computer ha questa opzione).

Si noti che non è necessario avere DOS installato per eseguire un programma di configurazione basato su DOS. Di solito è sufficiente avviare da un dischetto DOS ed eseguire i programmi dal dischetto fornito. Si può anche scaricare gratuitamente OpenDOS e FreeDOS.

Se si necessita, per la compatibilità con qualche altro sistema operativo, che sia abilitato il PnP, allora con Linux si dovrà usare il pacchetto isapnptools per configurare ogni volta la(e) scheda(e) all'avvio. Ci si dovrà ancora assicurare che l'indirizzo di I/O scelto per la scheda sia rilevato dal driver o fornito sotto forma di opzione `io=`.

### 3.7.2 La scheda Ethernet non è rilevata all'avvio

Di solito la causa di ciò e che si sta usando un kernel che non include il supporto per la propria particolare scheda. Per un kernel modulare ciò significa che non si è richiesto di caricare il modulo di cui si ha bisogno o che è necessario specificare al modulo un indirizzo come opzione.

Se si sta usando un kernel basato sui moduli per esempio quelli installati dalla maggior parte delle distribuzioni Linux, allora si provi a usare l'utilità di configurazione della distribuzione, per selezionare il modulo adatto alla propria scheda. Per le schede ISA è una buona idea determinare l'indirizzo di I/O della scheda e aggiungerlo come opzione (per esempio `io=0x340`) se l'utilità di configurazione ne richiede qualcuna. Se non c'è alcuna utilità di configurazione allora si dovrà aggiungere il nome corretto del modulo (e le opzioni) al file `/etc/conf.modules` – si veda `man modprobe` per maggiori dettagli.

Se si sta usando un kernel precompilato che è parte della distribuzione, allora si controlli la documentazione per vedere che kernel si è installato e se è stato compilato con il supporto per la propria scheda. Se non lo è stato, allora o si prova a procurarsene uno con il supporto per la propria scheda, oppure se ne compili uno su misura.

Solitamente è una buona cosa compilare il proprio kernel con solo i driver di cui si ha bisogno, poiché questa cosa non solo riduce la dimensione del kernel (risparmiando memoria RAM preziosa per le applicazioni!) ma riduce anche il numero di rilevamenti che possono incasinare hardware "sensibile". Compilare un kernel non è così complicato come sembra. Si deve solo rispondere sì o no a una serie di domande su quali driver si vogliono, e lui fa tutto il resto.

Un'altra causa importante è l'avere un altro dispositivo che usa una parte dello spazio di I/O di cui ha bisogno la propria scheda. Molte schede usano uno spazio di I/O di 16 o 32 byte. Se la propria scheda è impostata a `0x300` ed usa 32 byte. allora il driver chiederà la regione `0x300-0x31f`. Se un qualsiasi altro dispositivo ha registrato anche solo una porta da qualche parte in quell'intervallo, il rilevamento non avrà luogo a quell'indirizzo e il driver continuerà silenziosamente con il prossimo indirizzo fra quelli che prova. Quindi, dopo il boot, si faccia `cat /proc/ioports` per verificare che l'intero spazio d'I/O che la scheda richiede è libero.

Un altro problema è l'avere la propria scheda impostata con i ponticelli a un indirizzo di I/O che non viene controllato di default. L'elenco di tali indirizzi per ogni driver si trova facilmente proprio dopo i commenti iniziali nel sorgente del driver. Anche se l'impostazione I/O della propria scheda non è nell'elenco degli indirizzi controllati, la si può fornire al boot (per i driver compilati nel kernel) con il comando `ether=` come descritto in 10.1 (Passare argomenti Ethernet...). I driver modulari possono fare uso dell'opzione `io=` in `/etc/conf.modules` per specificare un indirizzo che non è controllato di default.

### 3.7.3 `ifconfig` mostra un indirizzo di I/O sbagliato per la scheda

No, non lo fa. Lo si sta solamente interpretando in modo scorretto. Questo *non* è un bug e i numeri riportati sono corretti. Capita solo che in alcune schede basate su 8390 (wd80x3, smc-ultra, ecc.) il chip 8390 in realtà risiede a un offset rispetto alla prima porta di I/O assegnata. Questo è il valore salvato in `dev->base_addr` ed è ciò che `ifconfig` riporta. Se si vuole vedere l'intero intervallo di porte che la propria scheda usa, allora si provi `cat /proc/ioports` che darà i numeri che ci si aspetta.

### 3.7.4 La macchina PCI rileva la scheda ma il driver ma no

Alcuni BIOS PCI potrebbero non abilitare tutte le schede PCI all'accensione, specialmente se è attivata l'opzione "PNP OS" del BIOS. Questa mis-feature è per supportare la versione corrente di Window che ancora usa alcuni driver in real mode. Si disabiliti questa opzione oppure si provi ad aggiornare a un driver più recente che abbia il codice per abilitare una scheda disabilitata.

### 3.7.5 Le schede ISA a memoria condivisa non funzionano in una macchina PCI (`0xffff`)

Questa cosa solitamente si presenta come una serie di letture di valori `0xffff`. Nessun tipo di scheda a memoria condivisa potrà funzionare in una macchina PCI finché non si configuri correttamente il BIOS PCI. Lo si deve impostare per permettere l'accesso in memoria condivisa dal bus ISA alla regione di memoria che la propria scheda cerca di usare. Se non si capisce quali impostazioni sono appropriate allora si chieda al proprio fornitore o all'esperto di computer locale. Per i BIOS AMI, solitamente c'è una sezione "Plug and Play" dove ci dovrebbero essere delle impostazioni tipo "ISA Shared Memory Size" e "ISA Shared Memory Base". Per le schede come la wd8013 e la SMC Ultra, si modifichi la dimensione predefinita da "Disabled" a 16kB e si metta l'indirizzo della memoria condivisa della propria scheda nell'altra impostazione.

### 3.7.6 Sembra che la scheda invii dati ma non riceve niente

Si faccia un `cat /proc/interrupts`. Il totale corrente del numero di eventi di interrupt generati dalla propria scheda sarà nell'elenco dato dal comando precedente. Se è a zero e/o non aumenta quando si prova a usare la scheda allora probabilmente c'è un conflitto di interrupt con un altro dispositivo installato nel computer (indifferentemente dal fatto che l'altro dispositivo abbia o meno un driver installato/disponibile). Si cambi l'IRQ di uno dei due dispositivi a un IRQ libero.

### 3.7.7 Supporto per Asynchronous Transfer Mode (ATM)

Werner Almesberger sta lavorando al supporto ATM per Linux. Sta lavorando con la scheda ENI155p della Efficient Networks ENI155p (*Efficient Networks* <http://www.efficient.com/>) e la scheda ZN1221 della Zeitnet (*Zeitnet* <http://www.zeitnet.com/>).

Werner sostiene che il driver per la ENI155p è abbastanza stabile, mentre quello per la ZN1221 non è ancora finito.

Si veda lo stato attuale dei driver al seguente URL:

*Linux ATM Support* <http://lrcwww.epfl.ch/linux-atm/>

### 3.7.8 Supporto per Gigabyte Ethernet

C'è un qualche supporto per gigabyte Ethernet sotto Linux?

Sì, attualmente ce ne sono almeno due. Un driver per l'adattatore Gigabit Ethernet PCI G-NIC della Packet Engines è disponibile nei kernel v2.0 e v2.2. Per maggiori dettagli, supporto e driver aggiornati, si veda:

`http://cesdis.gsfc.nasa.gov/linux/drivers/yellowfin.html`

Il driver `acenic.c` disponibile nei kernel v2.2 può essere usato con la scheda Gigabit Ethernet AceNIC della Alteon e con altre schede basate su Tigon come la 3c985 della 3Com. Il driver dovrebbe funzionare anche con la GA620 della NetGear, ma la cosa non è ancora stata verificata.

### 3.7.9 Supporto FDDI

C'è il supporto FDDI per Linux?

Sì. Larry Stefani ha scritto un driver per v2.0 usando le schede DEFEA (FDDI EISA) e DEFPA (FDDI PCI) della Digital. Era stato incluso nel kernel v2.0.24. Attualmente non sono supportate altre schede.

### 3.7.10 Supporto Full Duplex

Il Full Duplex mi darà i 20MBps? Linux lo supporta?

Cameron Spitzer ha scritto quanto segue sulle schede 10Base-T full duplex: «Se se ne connette una a uno switched hub full duplex e il proprio sistema è abbastanza veloce e non sta facendo molto altro, può mantenere la connessione occupata in entrambe le direzioni. Non ci sono cose tipo 10BASE-2 o 10BASE-5 full duplex. Il full duplex funziona disabilitando il rilevamento delle collisione nell'adattatore. Questo è il motivo per cui non lo si può fare con un cavo coassiale; la LAN in questo modo non funzionerebbe. Il 10BASE-T (interfaccia RJ45) usa fili separati per la trasmissione e la ricezione così è possibile utlizzare entrambe le direzioni nello stesso momento. Lo switching hub si fa carico del problema delle collisioni. La frequenza dei segnali è 10Mbps.»

Quindi come si può vedere si sarà in grado di ricevere e trasmettere solo a 10Mbps e così non ci si aspetti un incremento delle prestazioni di un fattore 2. Che sia o meno supportato, la cosa dipende dalla scheda e forse anche dal driver. Alcune schede possono fare l'auto negoziazione, altre hanno bisogno del supporto da parte del driver e per altre ancora è necessario che l'utente selezioni un'opzione nella configurazione EEPROM della scheda. Comunque solamente utilizzatori seri/pesanti noteranno la differenza tra le due modalità.

### 3.7.11 Schede Ethernet per Linux su macchine SMP

Se si spendono soldi in più su un computer multi processore (MP) allora si comperi anche una buona scheda Ethernet. Per i kernel v2.0 non è veramente necessario, ma lo è sicuramente per quelli v2.2. La maggior parte delle schede più vecchie non intelligenti (per esempio progetti per bus ISA PIO e memoria condivisa) non furono mai pensate considerando in alcun modo l'uso con macchina MP. La tendenza attuale è di comperare una scheda intelligente di progettazione moderna e assicurarsi che il driver sia stato scritto (o aggiornato) per gestire le operazioni MP (le parole chiave sono "progettazione moderna": le NE2000 PCI sono semplicemente un progetto di 10 anni fa adattato a un bus moderno). La presenza del testo `spin_lock` nei sorgenti del driver è un buona indicazione che il driver è stato scritto per gestire le operazioni MP. Si veda sotto per i dettagli completi su perché di dovrebbe acquistare una buona scheda per l'uso MP (e cosa accade se non lo si fa).

Nei kernel v2.0, solo un processore alla volta era permesso "in modalità kernel" (ovvero poteva cambiare i dati del kernel e/o eseguire device driver). Quindi dal punto di vista della scheda (e del driver associato) non c'era niente di diverso rispetto ad operazioni uni-processore (UP) e le cose funzionavano come prima (questo era il modo più semplice di ottenere una versione MP di Linux funzionante: un unico grosso lock (blocco, semaforo) attorno a tutto il kernel permette l'accesso ad un solo processore alla volta. In questo modo si sa che non si avranno due processori che provano a cambiare la stessa cosa allo stesso tempo!).

Lo svantaggio nel permettere un solo processore alla volta in modalità kernel è che si ottengono prestazioni di tipo MP solamente se si eseguono programmi indipendenti e che fanno calcoli pesanti. Se i programmi fanno un sacco di input/output (I/O) come la lettura e la scrittura di dati su disco o attraverso la rete, allora tutti i processori tranne uno saranno in stallo in attesa che le loro richieste di I/O siano completate mentre l'unico processore in esecuzione in modalità kernel freneticamente prova a eseguire tutti i device driver per soddisfare le richieste di I/O. Il kernel diventa il collo di bottiglia e poiché c'è solo un processore in modalità kernel, le prestazioni di una macchina MP in presenza di I/O pesante, in questo caso detoriorano velocemente verso quelle di una macchina a singolo processore.

Poiché questa cosa è chiaramente inferiore al caso ideale (specialmente per server di file/WWW, router, ecc.) i kernel v2.2 hanno un lock più fine. Ciò significa che più di un processore alla volta può essere in modalità kernel. Invece di un unico grosso lock attorno all'intero kernel, ci sono un sacco di piccoli lock che proteggono i dati critici dall'essere manipolati da più di un processore alla volta, per esempio un processore può eseguire il driver della scheda di rete, mentre nello stesso momento un altro esegue il driver il disco.

Okay, con tutto questo bene in mente, ecco qui le insidie: il lock più fine significa che si può avere un processore che prova a inviare dati attraverso un driver Ethernet mentre un altro prova ad accedere allo stesso driver/scheda per fare qualcos'altro (per esempio ottenere le statiche della scheda per il comando `cat /proc/net/dev`). Oops, le statistiche della propria scheda sono state appena inviate attraverso il cavo, mentre invece si volevano i dati per le proprie statistiche. Sì, si è un po' confusa la scheda chiedendole di fare due (o più!) cose alla volta ed è probabile che mandi in crash la propria macchina mentre ci prova.

Quindi il driver che funzionava per UP non è più abbastanza buono: necessita di essere aggiornato con i lock che controllano l'accesso alla scheda cosicché i tre processi di ricezione, trasmissione e manipolazione dei dati di configurazione siano serializzati al grado necessario alla scheda per funzionare stabilmente. La parte strana qui è che un driver non ancora aggiornato con lock per operazioni MP stabili, sembrerà probabilmente funzionare su macchine MP in presenza di un leggero carico di rete, ma provocherà il crash della macchine o almeno esibirà uno strano comportamento quando due (o più!) processori proveranno a fare più di uno di questi tre processi nello stesso momento.

Un driver Ethernet aggiornato per MP richiederà (al minimo) un lock attorno al driver che limiti l'accesso ai punti d'ingresso del kernel nel driver in maniera "uno alla volta, prego". Con questa cosa, le cose saranno serializzate cosicché l'hardware sottostante dovrebbe essere trattato come se fosse usato in una macchina UP, e quindi dovrebbe essere stabilee. Lo svantaggio è che un unico lock attorno all'intero driver Ethernet ha le stesse implicazioni negative sulle prestazioni dell'avere un unico grosso lock attorno a tutto il kernel (ma su scala più piccola), ovvero si può avere un solo processore alla volta che usa la scheda. [Nota tecnica: L'impatto sulle prestazioni può anche includere un incremento nelle latenze degli interrupt se i lock che è necessario aggiungere sono del tipo `irqsave` e sono tenuti per un lungo periodo di tempo.]

Possibili miglioramenti a questa situazione possono essere fatti in due modi. Si può provare a minimizzare il tempo tra quando si fa il lock e quando lo si rilascia, e/o si può implementare un lock più fine all'interno del driver (per esempio un lock attorno all'intero driver sarebbe eccessivo se invece fosse necessario solamente un lock o due per proteggere contro l'accesso simultaneo a una coppia di registri/impostazioni delicati della scheda).

Comunque, per le più vecchie schede non intelligenti che non sono mai state progettate pensando all'uso MP, nessuno di questi miglioramenti può essere realizzato. Peggio ancora le schede non intelligenti tipicamente richiedono che il processore sposti dati tra la scheda e la memoria del computer, cosicché nel caso peggiore il lock sarà mantenuto per tutto il tempo necessario per spostari ogni pacchetto da 1.5kB sul bus ISA.

Le più moderne schede intelligenti tipicamente spostano i dati direttamente da e nella memoria del computer senza nessun aiuto da parte di un processore. Questa è una grande vittoria, poiché il lock è mantenuto allora solo per il breve tempo che il processore impiega per dire alla scheda dove prendere/mettere nella memoria il prossimo pacchetto di dati. Inoltre le schede più moderne tendono meno a richiedere un grosso unico lock attorno all'intero driver.

### 3.7.12 Schede Ethernet per Linux su piattaforma PCI Alpha/AXP

Dalla versione v2.0, solo i driver 3c509, depca, de4x5, pcnet32 e tutti quelli 8390 (wd, smc-ultra, ne, 3c503, ecc.) sono stati resi "indipendenti dall'architettura" così da poter funzionare su sistemi basati su CPU DEC Alpha. Possono funzionare anche altri driver PCI aggiornati presenti nella pagina WWW di Donald poiché sono stati scritti appositamente indipendenti dall'architettura.

Si noti che le modifiche richieste per rendere un driver indipendente dall'architettura non sono così complicate. Si deve solo fare quanto segue:

- moltiplicare tutti i valori relativi ai `jiffies` per HZ/100 per tener conte del diverso valore di HZ che usa l'Alpha. (`timeout=2;` diventa `timeout=2*HZ/100;`)
- sostituire tutti deriferimenti dei puntatori nella memoria di I/O (da 640k a 1MB) con le appropriate chiamate readb() writeb() readl() writel() calls, come mostrato in questo esempio.

```
-       int *mem_base = (int *)dev->mem_start;
-       mem_base[0] = 0xba5eba5e;
+       unsigned long mem_base = dev->mem_start;
+       writel(0xba5eba5e, mem_base);
```

-sostituire tutte le chiamate memcpy() che hanno la memoria I/O come sorgente o destinazione con le appropriate `memcpy_fromio()` o `memcpy_toio()`.

I dettagli sulla gestione degli accessi alla memoria in maniera indipendente dall'architettura sono documentati nel file `linux/Documentation/IO-mapping.txt` fornito con i kernel recenti.

### 3.7.13 Ethernet per Linux su hardware SUN/Sparc

Per le informazioni più aggiornate sulla roba Sparc, si provi il seguente URL:

*Linux Sparc* `<http://www.geog.ubc.ca/sparc>`

Si noti che qualche hardware Ethernet Sparc riceve il suo indirizzo MAC dal computer ospite, e quindi ci si può ritrovare con più interfacce allo stesso indirizzo MAC. Se si ha bisogno di mettere più di una interfacia nella stessa rete, allora si usi l'opzione `hw` di `ifconfig` per assegnare un indirizzo MAC univoco.

Le questioni riguardati il porting dei driver PCI su piattaforma Sparc sono simili a quelle citate sopra per la piattaforma AXP. Inoltre ci possono essere un po' di questione relative all'endian, poiché la Sparc è big endian mentre AXP e ix86 sono little endian.

### 3.7.14 Ethernet per Linux su altro hardware

Ci sono parecchie altre piattaforme hardware sulle quali può girare Linux, come Atari/Amiga (m68k). Come nel caso Sparc è meglio controllare nel sito ufficiale del port di Linux per quella piattaforma per vedere cosa attualmente è supportato (sono benvenuti link ai suddetti siti – inviatemeli!)

### 3.7.15 Connettere 10 o 100 BaseT senza un hub

Possono connettere assieme sistemi basati su 10/100BaseT (RJ45) senza un hub?

Senza altri dispositivi/marchingegni si possono collegare facilmente 2 macchine, ma non di più. Si veda 6.2 (Doppino intrecciato), che spiega come farlo. E no, non si può far su un hub semplicemente incrociando un po' di fili assieme e qualcos'altro. È praticamente impossibile gestire bene il segnale di collisione senza duplicare l'hub.

### 3.7.16 SIOCSIFxxx: No such device

Ricevo un sacco di messaggi 'SIOCSIFxxx: No such device' al boot, seguiti da un 'SIOCADDRT: Network is unreachable'. Cosa c'è che non va?

Il proprio dispositivo Ethernet non è stato rilevato al boot o all'inserzione del modulo e quando sono eseguiti `ifconfig` e `route` non hanno nessun dispositivo con cui lavorare. Si usi `dmesg | more` per rivedere i messaggi di boot e vedere se c'è un qualche messaggio sul rilevamento della scheda Ethernet.

### 3.7.17 SIOCSFFLAGS: Try again

Ricevo 'SIOCSFFLAGS: Try again' quando eseguo 'ifconfig'. Eh?

Qualche altro dispositivo si è preso l'IRQ che la propria scheda Ethernet prova ad usare e quindi questa non può usarlo. Non si deve necessariamente riavviare per risolvere ciò, poiché alcuni dispositivi si prendono l'IRQ solo quando ne hanno bisogno e poi lo rilasciano quando hanno fatto. Esempi sono alcuni schede sonore, porte seriali, driver per floppy disk, ecc. Si può digiare `cat /proc/interrupts` per vedere quali interrupt sono attualmente *in uso*. La maggior parte dei driver per schede Ethernet per Linux si prendono l'IRQ solo quando sono attivate per l'uso con 'ifconfig'. Se si riesce a far sì che l'altro dispositivo rilasci la linea IRQ richiesta allora si dovrebbe essere in grado di 'Try again' (riprovare) con ifconfig.

### 3.7.18 Usare 'ifconfig' e Link UNSPEC con l'indirizzo hardware 00:00:00:00:00:00

Quando eseguo ifconfig senza alcun argomento, mi dice che LINK è UNSPEC (invece di 10Mbs Ethernet) e dice anche che il mio indirizzo hardware è di tutti zeri.

Questo è perché si sta usando una versione del programma 'ifconfig' più recente della versione del kernel. Questa nuova versione di ifconfig non è in grado di riportare queste proprietà quando usata con un kernel più vecchio. Si può o aggiornare il proprio kernel, tornare a una versione più vecchia di 'ifconfig' oppure semplicemente ignorare la cosa. Il kernel conosce l'indirizzo hardware e non gli interessa se ifconfig non può leggerlo.

Si possono ricevere strane informazioni se il programma `ifconfig` che si sta usando è molto più vecchio del proprio kernel.

### 3.7.19 Enorme numero di errori in RX e TX

Quando eseguo ifconfig senza alcun argomento, mi dice che ho un enorme numero di errori sia nei pacchetti ricevuti che trasmessi. Tutto sembra funzionare bene. Cosa c'è che non va?

Si guardi bene. Dice `RX packets` *grande numero* **PAUSE** `errors 0` **PAUSE** `dropped 0` **PAUSE** `overrun 0`. E la stessa cosa per la colonna `TX`. Quindi i grandi numeri che si vedono sono il numero totale di pacchetti che la propria macchina ha ricevuto e trasmesso. Se ancora si trova la cosa confusa, si provi invece ad usare `cat /proc/net/dev`.

### 3.7.20 Voci in `/dev/` per le schede Ethernet

Ho /dev/eth0 come link (collegamento) a /dev/xxx. È giusto?

Contrariamente a ciò che si è sentito, i file in /dev/* non sono utilizzati. Si può cancellare qualsiasi `/dev/wd0`, `/dev/ne0` e voce simile.

### 3.7.21 Linux e i "trailer"

Dovrei disabilitare i trailer quando uso 'ifconfig' con la mia scheda?

Non si possono disabilitare i trailer e nemmeno si dovrebbe volerlo fare. I 'trailer' sono degli stratagemmi (hack) per evitare la copia dei dati negli strati di rete. L'idea era di usare un banale header di dimensione fissata 'H', mettere le informazioni dell'header di dimensione variabile alla fine del pacchetto e allocare tutti gli 'H' byte del pacchetto prima dell'inizio della pagina. Sebbene fosse una buona idea, in pratica ha mostrato di non funzionare bene. Se qualcuno suggerisce l'uso di '-trailers', si noti che è l'equivalente del sangue delle capre sacrificali. Non sarà niente per risolvere il problema, ma se il problema si risolve da solo allora qualcuno può affermare una profonda conoscenza delle arti magiche.

### 3.7.22 Accesso a basso livello al dispositivo Ethernet

Come posso accedere a basso livello al dispositivo Ethernet in Linux, senza passare attraverso TCP/IP e company?

```
int s=socket(AF_INET,SOCK_PACKET,htons(ETH_P_ALL));
```

Questo consente di avere un socket che riceve qualsiasi tipo di protocollo. Si facciano chiamate `recvfrom()` a questo socket e lui riempirà sockaddr con il tipo di dispositivo nel campo sa_family e il nome del dispositivo nell'array sa_data. Non so chi abbia originariamente inventato SOCK_PACKET per Linux (c'è da anni) ma è una cosa superba. Si può usare anche per inviare roba grezza attraverso chiamate `sendto()`. Naturalmente per fare entrambe le cose si deve avere l'accesso come root.

# 4 Suggerimenti per le prestazioni

Ecco qua un po' di trucchi che si possono usare se si soffre di una basso throughput Ethernet o per guadagnare un po' di velocità nei trasferimenti ftp.

Il programma `ttcp.c` è un buon test per misurare la velocità di throughput. Un altro modo comunemente usato è di fare un `ftp> get grosso_file /dev/null` dove `grosso_file` è >1MB e risiede nel buffer di Tx della macchina (si faccia il 'get' almeno due volte, poiché la prima volta si approntera la cache del buffer nella macchina di trasmissione). Si deve mettere il file nella cache del buffer perché non si è interessati ad includere nella misura la velocità di accesso ai file dal disco. Questo è anche il motivo per cui si inviano i dati in ingresso a `/dev/null` piuttosto che dentro il disco.

## 4.1 Concetti generali

Anche una scheda a 8 bit è in grado di ricevere pacchetti back-to-back (uno dietro l'altro) senza alcun problema. Le difficoltà nascono quando il computer non riesce a estrarre i pacchetti ricevuti dalla scheda abbastanza velocemente da far spazio ai pacchetti in arrivo. Se il computer non libera abbastanza velocemente la memoria della scheda dai pacchetti già ricevuti, la scheda non avrà spazio per mettere i nuovi pacchetti.

In questo caso o si scartano (drop) i nuovi pacchetti oppure si scrivono sopra a quelli già ricevuti (overrun). In entrambi i casi si interrompe seriamente il flusso regolare del traffico causando/richiedendo la ritramissione e degradando seriamente le prestazioni anche di un fattore 5!

Le schede con a bordo più memoria possono fare la "bufferizzazione" di più pacchetti e quindi gestire grossi picchi (burst) di pacchetti back-to-back senza perderne nessuno. D'altra parte ciò significa che la scheda non

necessita più di una bassa latenza dal computer riguardo all'estrazione dei pacchetti dal buffer per evitare la perdita di pacchetti.

La maggior parte delle schede a 8 bit hanno un buffer da 8kB e la maggior parte di quelle a 16 ne hanno uno da 16kB. La maggior parte dei driver di Linux riserva 3kB del buffer per due buffer Tx, quindi nel caso di una scheda a 8 bit rimangono solo 5kB di spazio per la ricezione. Questo spazio è sufficiente solo per tre pacchetti Ethernet a dimensione massima (1500 byte).

## 4.2 Schede ISA e velocità del bus ISA

Come menzionato in precedenza, se i pacchetti sono rimossi abbastanza velocemente dalla scheda allora la condizione di drop/overrun non avverrà anche se la dimensione del buffer per i pacchetti Rx è piccola. Il fattore che determina la frequenza con la quale i pacchetti sono rimossi dalla scheda e messi nella memoria del computer è la velocità del percorso dati che collega le due, ovvero la velocità del bus ISA (se la CPU è un vecchio e lento 386sx-16 allora anche questo avrà la sua influenza).

Il clock del bus ISA raccomandato è circa 8Mhz, ma molte schede madri e dispositivi periferici possono essere fatti funzionare a frequenze più elevate. La frequenza di clock per il bus ISA solitamente può essere impostata nel CMOS setup, selezionando un divisore della frequenza di clock della scheda madre/CPU. Alcune schede madri ISA e PCI/ISA possono non avere questa opzione e quindi si è incastrati nelle impostazioni di fabbrica.

Per esempio, ecco qui alcune velocità di ricezione misurate dal programma TTCP su un 486 a 40Mhz, con una scheda a 8 bit WD8003EP, a diverse velocità del bus ISA.

| Velocità bus ISA (MHz) | Rx TTCP (kB/s) |
|---|---|
| 6.7 | 740 |
| 13.4 | 970 |
| 20.0 | 1030 |
| 26.7 | 1075 |

Voglio vedere chi riesce a far meglio di 1075kB/s con una *qualsiasi* scheda Ethernet a 10Mb/s usando TCP/IP. Comunque, non ci si aspetti che qualsiasi sistema funzioni a velocità del bus ISA molto elevate. La maggior parte dei sistemi non funzioneranno correttamente a velocità superiori a 13MHz (inoltre, in alcuni sistemi PCI la velocità del bus ISA è fissata a 8 Mhz, cosicché l'utente finale non ha la possibilità di incrementarla).

Oltre a una velocità di trasferimento più elevata, si trarrà anche beneficio da una riduzione nell'uso della CPU dovuta alla durata minore dei cicli di memoria e di I/O (si noti che anche i dischi fissi e le schede video presenti nel bus ISA mostreranno solitamente un aumento delle prestazioni dovuto all'incremento della velocità del bus ISA).

Ci si assicuri di fare un back up dei propri dati prima di fare esperimenti con velocità del bus ISA superiori agli 8Mhz e di controllare accuratamente che tutte le periferiche ISA funzionino correttamente dopo aver fatto qualsiasi incremento alla velocità del bus.

## 4.3 Impostare la finestra TCP di ricezione (TCP Rx Window)

Ancora una volta, le schede con poca memoria RAM a bordo e percorsi dati tra la scheda e la memoria del computer relativamente lenti avranno problemi. L'impostazione predefinita per la finestra di ricezione TCP è di 32kB, il che significa che un computer veloce nella sottorete a cui appartiene la propria macchina, può scaricare in quest'ultima 32kB di dati senza fermarsi per controllare se tutti sono stati ricevuti correttamente.

Le versioni recenti del comando `route` hanno la possibilità di impostare al volo la dimensione di questa finestra. Solitamente la riduzione della dimensione di questa finestra è necessaria solo per la rete locale,

poiché i computer che sono dietro ad un paio di router e gateway sono abbastanza 'bufferizzati' da non porre problemi. Un esempio d'uso potrebbe essere:

```
route add <qualcosa> ... window <dim_fin>
```

dove `dim_fin` è la dimensione della finestra che si vuole usare (in byte). Una scheda 3c503 a 8 bit su un bus ISA che va a 8Mhz o meno dovrebbe funzionare bene con una dimensione della finestra di circa 4kB. Una finestra troppo grande causerà drop e overrun dei pacchetti e una riduzione drastica del throughput. Si può verificare lo stato operativo facendo `cat /proc/net/dev` che mostrerà tutte le situazioni di drop/overrun avvenute.

### 4.4 Incrementare le prestazioni NFS

Alcuni hanno scoperto che l'uso di una scheda a 8 bit in client NFS causa prestazioni più povere di quelle che ci si aspetta, quando si usa la dimensione dei paccheti NFS di 8kB (nativa della Sun).

Una possibile causa per questa cosa potrebbe essere la differente dimensione del buffer a bordo tra schede a 8 e 16 bit. La dimensione massima di un pacchetto Ethernet è circa 1500 byte. Affinché arrivi un pacchetto NFS di 8kB ci vogliono circa sei pacchetti back to back Ethernet di dimensione massima. Sia le schede a 8 bit che quelle a 16 non hanno problemi a ricevere pacchetti back to back. Il problema sorge quando la macchina non rimuove in tempo i pacchetti dal buffer della scheda e il buffer va in overflow. E nemmeno il fatto che le schede a 8 bit necessitano di un ulteriore ciclo di bus ISA per ogni trasferimento aiuta. Quel che si *può* fare se si ha una scheda a 8 bit è di impostare la dimensione del trasferimento NFS a 2kB (o anche 1kB) o provare a incrementare la velocità del bus ISA per far sì che il buffer della scheda venga ripulito più velocemente. Ho scoperto che una vecchia scheda WD8003E a 8MHz (senza altro carico di sistema) può sostenere grosse ricezioni a dimensioni NFS di 2kB, ma non a 4kB, dove le prestazioni si degradano di un fattore tre.

D'altra parte, se l'opzione predefinita di mount è di usare la dimensione di 1kB e si ha almeno una scheda isa a 16 bit, si troverà un incremento significativo nel passare a 4kB (o anche 8kB).

## 5 Informazioni specifiche su rivenditori, produttori e modelli

Nel seguito sono elencate molte schede in ordine alfabetico per nome del rivenditore (produttore) e poi identificativo del prodotto. Oltre all'ID di ciascun prodotto, ci potrà essere "Supportata", "Semi supportata" oppure "Non supportata".

Supportata significa che esiste un driver per la scheda e molti lo stanno usando felicemente e sembra quindi abbastanza affidabile.

Semi supportata significa che esiste un driver, ma è vera almeno una delle descrizioni seguenti: (1) Il driver e/o l'hardware ha dei bug che possono provocare prestazioni scadenti, fallimenti di connessione e persino crash. (2) Il driver è nuovo e la scheda è poco comune e quindi il driver è stato poco usato/collaudato e il suo autore ha quindi ricevuto pochi riscontri. Ovviamente la (2) è preferibile alla (1) e la descrizione di ciascuna scheda/driver dovrebbe chiarire quale delle due valga. In entrambi i casi, probabilmente sarà necessario rispondere 'Y' (sì) alla domanda "Prompt for development and/or incomplete code/drivers?" quando si lancia `make config`.

Non supportata significa invece che attualmente non è disponibile alcun driver per quella scheda. Ciò può essere dovuto alla mancanza d'interesse nell'hardware che è raro/poco comune, o al fatto che il produttore non vuole rilasciare la documentazione sull'hardware necessaria per scrivere un driver.

Si noti che la differenza tra "Supportata" e "Semi supportata" è abbastanza soggettiva ed è basata sui commenti degli utenti osservati nei messaggi inviati ai vari newsgroup e mailing list (dopo tutto, è impossibile per una persona sola collaudare tutti i driver con tutte le schede per ogni versione del kernel!!!). Quindi si faccia attenzione perché può succedere che si trovi una scheda segnata come semi-supportata, che nel proprio caso funziona perfettamente (il che è bene), oppure una scheda indicata come supportata, che in realtà dà una serie infinita di problemi (il che non è proprio bene).

Dopo lo stato, c'è il nome che il driver ha nel kernel di Linux. Questo è anche il nome del modulo del driver che si potrebbe usare nella riga `alias eth0 nome_driver` presente nel file di configurazione dei moduli `/etc/conf.modules`.

## 5.1 3Com

Se non si è sicuri di cosa sia la propria scheda, ma si pensa che sia una scheda 3Com, probabilmente lo si può scoprire dal numero di assemblaggio. La 3Com ha un documento 'Identifying 3Com Adapters By Assembly Number' (ref 24500002) che molto probabilmente farà al proprio caso. Si veda 8.4 (Informazioni tecniche dalla 3Com) per maggiori informazioni su come ottenere documenti dalla 3Com.

Si noti inoltre che la 3Com ha un sito FTP con diverse cose carine: `ftp.3Com.com`.

Quelli che leggono questo documento tramite un browser WWW, possono provare a vedere anche nel sito WWW della 3Com.

### 5.1.1 3c501

Stato: Semi supportata, Nome del Driver: 3c501

Questa obsoleta scheda a 8 bit dell'età della pietra è veramente troppo "demente" per poter essere usata. La si eviti come la peste. Non si compri questa scheda, nemmeno per scherzo. Le sue prestazioni sono orribili ed ha un sacco di problemi.

Per quelli che non sono ancora convinti, la 3c501 può solamente fare una cosa alla volta: mentre sta rimuovendo un pacchetto dal buffer non può ricevere un altro pacchetto, nemmeno può riceverne uno mentre sta caricando un pacchetto trasmesso. Questa cosa andava bene per le reti composte da computer basati su 8088 dove l'elaborazione di un pacchetto e la risposta richiedevano decine di millisecondi, ma le reti moderne inviano pacchetti back-to-back (uno dopo l'altro) praticamente per qualsiasi transazione.

L'autoIRQ funziona, non è usato il DMA, l'autoprobe cerca solo a `0x280` e a `0x300` e il livello di debug è impostato con il terzo argomento al momento del boot.

Ancora una volta, l'uso della 3c501 è *vivamente sconsigliato*! Ancora di più con un kernel con l'IP multicast, ci si inchioderà mentre si ascoltano *tutti* i pacchetti multicast. Si vedano i commenti in cima al codice sorgente per ulteriori dettagli.

### 5.1.2 EtherLink II, 3c503, 3c503/16

Stato: Supportata, Nome del driver: 3c503 (+8390)

La 3c503 non ha un "EEPROM setup" e quindi non è necessario eseguire nessun programma di diagnostica/setup prima di usare la scheda con Linux. L'indirizzo della memoria condivisa della 3c503 è impostato usando i ponticelli in comune con l'indirizzo della PROM di boot. Ciò confonde molta gente che ha familiarità con altre schede ISA, dove di solito si lascia sempre il ponticello impostato per "disabilitare" a meno che non si abbia una PROM di boot.

Queste schede dovrebbero andare alla stessa velocità delle WD80x3 a pari larghezza di bus, ma in realtà sono un po' più lente. Queste schede Ethernet hanno anche un modalità ad I/O programmato che non usa le funzionalità offerte dal 8390 (gli ingegneri della 3Com hanno scoperto troppi bachi!). Il driver 3c503 di Linux funziona anche con la 3c503 in modalità I/O programmato, ma è più lento e meno affidabile rispetto alla modalità a memoria condivisa. Inoltre la modalità ad I/O programmato non viene ben collaudata quando vengono aggiornati i driver. Non si dovrebbe usare la modalità in I/O programmato a meno che non se ne abbia bisogno per compatibilità con MS-DOS.

La linea IRQ della 3c503 è impostata nel software, con nessun suggerimento da parte della EEPROM. Diversamente dal driver MS-DOS, quello per Linux ha la funzionalità di autoIRQ: usa la prima linea IRQ libera fra {5,2/9,3,4}, selezionandola ogni volta che la scheda è configurata con ifconfig (versioni più vecchie del driver selezionano l'IRQ all'avvio). La chiamata ioctl() in 'ifconfig' restituirà EAGAIN se non c'è alcuna linea IRQ disponibile.

Alcuni problemi comuni che la gente ha con la 503 sono discussi in 3.6 (Problemi con...).

Se si intende usare questo driver come modulo caricabile probabilmente si dovrebbe vedere 10.2 (Utilizzare i driver Ethernet come moduli) per informazioni specifiche sui moduli.

Si noti che alcune vecchie workstation diskless (senza dischi) 386 hanno già una 3c503 (fatta dalla 3Com e venduta sotto nome diverso, come 'Bull') ma l'ID di produzione non è un ID 3Com e quindi non sarà rilevata. Maggiori dettagli possono essere trovati nel pacchetto Etherboot, di cui si avrà comunque bisogno per avviare queste macchine diskless.

### 5.1.3 Etherlink Plus 3c505

Stato: Semi supportata, Nome del driver: 3c505

È un driver che era stato scritto da Craig Southeren `geoffw@extro.ucc.su.oz.au`. Queste schede usano anche il chip i82586. Non ci sono poi tante schede di questo tipo in giro. È incluso nel kernel standard, ma viene dichiarato come driver alpha. Si veda la sezione 3.1 (Driver alpha) per importanti informazioni sull'uso con Linux di driver Ethernet in alpha-test.

Se si ha intenzione di usare una di queste schede si dovrebbe leggere anche il file `/usr/src/linux/drivers/net/README.3c505`. Contiene diverse opzioni che si possono abilitare o disabilitare.

### 5.1.4 Etherlink-16 3c507

Stato: Semi supportata, Nome del driver: 3c507

Questa scheda usa uno dei chip Intel e lo sviluppo del driver è strettamente legato allo sviluppo del driver per la Ether Express dell'Intel. Il driver è incluso nel kernel standard, ma come driver sperimentale. Si veda la sezione 3.1 (Driver alpha) per importanti informazioni sull'uso con Linux di driver Ethernet in alpha-test.

### 5.1.5 Etherlink III, 3c509 / 3c509B

Stato: Supportata, Nome del driver: 3c509

Questa scheda è piuttosto economica e ha buone prestazioni per essere una ISA senza bus-master. Gli svantaggi sono che la 3c509 originale richiede una latenza di interrupt molto bassa. La 3c509B non dovrebbe soffrire dello stesso problema, poiché ha un buffer più grande (vedere sotto). Queste schede usano i trasferimenti PIO, analogamente a una scheda ne2000 e quindi, in confronto, una scheda a memoria condivisa come una wd8013 sarà più efficiente.

La 3c509 originale ha un buffer per i pacchetti piccolino (4kB totali, 2kB Rx, 2k Tx), cosicché qualche volta il driver perde dei pacchetti se gli interrupt sono mascherati troppo a lungo. Per minimizzare questo problema, si può provare a non mascherare gli interrupt durante i trasferimenti dei dischi IDE (si veda *man hdparm*) e/o a incrementare la velocità del proprio bus ISA cosicché i trasferimenti dei dischi IDE terminino prima.

Il più recente modello 3c509B ha 8kB sulla scheda e il buffer può essere diviso in 4/4, 5/3 o 6/2 per Rx/Tx. Questa impostazione è modificabile con l'utilità di configurazione DOS ed è salvata nella EEPROM. Questo dovrebbe alleviare il suddetto problema della 3c509 originale.

Gli utilizzatori della 3c509B dovrebbero usare l'utilità DOS fornita per disabilitare il supporto *plug and play* e per impostare il tipo di uscita di cui si ha bisogno. Il driver per Linux attualmente *non* supporta la "Autodetect media setting", quindi si *deve* selezionare 10Base-T, 10Base-2 oppure AUI. Si noti che per disabilitare completamente il PnP, si dovrebbe fare un `3C5X9CFG /PNP:DISABLE` e poi farlo seguire da un reset della macchina per assicurarsi che abbia avuto effetto.

Alcuni chiedono delucidazioni sulle impostazioni "Server or Workstation" e "Highest Modem Speed" presenti nella utilità di configurazione sotto DOS. Donald scrive «Questi sono solo degli hint ai driver e il driver di Linux non usa questi parametri: ottimizza sempre per la massima velocità di trasferimento piuttosto che per la bassa latenza ('Server'). La bassa latenza era di importanza critica per i vecchi throughput IPX non-windowed. Per ridurre la latenza il driver MS-DOS per la 3c509 disabilita gli interrupt per alcune operazioni, bloccando gli interrupt per la porta seriale. Da qui la necessità dell'impostazione 'modem speed'. Il driver per Linux rimuove la necessità di disabilitare gli interrupt per lunghi periodi di tempo operando solo su interi pacchetti, ad esempio non cominciando a trasmettere un pacchetto finché non sia stato completamente trasferito nella scheda.»

Si noti che il rilevamento della scheda ISA usa un metodo diverso rispetto a quello di molte schede. In sostanza, si chiede alla scheda di rispondere inviando dati ad una ID_PORT (le porte da `0x100` a `0x1ff` con intervalli di `0x10`). Questo rilevamento implica che una scheda particolare sarà *sempre* rilevata per prima in una configurazione con più 3c509 ISA. La scheda con il più basso indirizzo Ethernet hardware alla fin fine sarà *sempre* la `eth0`. Questa cosa non dovrebbe preoccupare nessuno, tranne quelli che vogliono assegnare un indirizzo hardware da 6 byte a una particolare interfaccia. Se si hanno più schede 3c509, la cosa migliore è aggiungere i comandi `ether=0,0,ethN` senza specificare la porta di I/0 (ovvero si usi I/O=zero) e permettere al rilevamento di determinare quale scheda è la prima. L'uso di un valore I/O non nullo assicura solamente che non saranno rilevate tutte le schede, quindi non lo si faccia.

Se questa cosa veramente vi risulta noiosa, si dia un'occhiata all'ultimo driver di Donald, in quanto si può essere in grado di usare un valore `0x3c509` nei campi dell'indirizzo di memoria inutilizzati per ordinare il rilevamento ed adattarlo alle proprie necessità.

### 5.1.6 3c515

Stato: Supportata, Nome del driver: 3c515

Questa scheda, nome in codice "CorkScrew", è la scheda a 100Mbps ISA offerta dalla 3COM. Un driver relativamente nuovo di Donald è incluso nei kernel v2.2. Per informazioni più aggiornate, probabilmente si dovrebbe guardare nella pagina relativa alla Vortex:

*Vortex* `<http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html>`

### 5.1.7 3c523

Stato: Semi supportata, Nome del driver: 3c523

Questa scheda su bus MCA usa l'i82586. Chris Beauregard ha modificato il driver ni52 per farlo funzionare con queste schede. Il driver può essere trovato nell'albero dei sorgenti dei kernel v2.2.

Maggiori dettagli possono essere trovati nella pagina di MCA-Linux a `http://glycerine.cetmm.uni.edu/mca/`.

#### 5.1.8 3c527

Stato: Non supportata.

Sì, un'altra scheda MCA. No, non risquote molto interesse. Se si è impelagati con l'MCA maggior fortuna la si ha con la 3c529.

#### 5.1.9 3c529

Stato: Supportata, Nome del driver: 3c509

Questa scheda in realtà usa lo stesso chipset della 3c509. Molto prima che il supporto MCA fosse aggiunto al kernel, Donald ha messo un hook nel driver della 3c509 che cerca le schede MCA dopo aver provato a rilevare quelle EISA e prima di cercare le ISA. Il codice di rilevamento richiesto è incluso nel driver distribuito con i kernel v2.2. Maggiori dettagli nella pagina di MCA-Linux a

`http://glycerine.cetmm.uni.edu/mca/`

#### 5.1.10 3c562

Stato: Supportata, Nome del driver: 3c589 (distribuito separatamente)

Questa scheda PCMCIA è la combinazione di una scheda Ethernet 3c589B con un modem. All'utente finale il modem appare come un modem normalissimo. La sola difficoltà è far sì che i due driver di Linux condividano lo stesso interrupt. C'è il supporto per alcuni nuovi registri e un po' per la condivisione degli interrupt hardware. Si deve usare un kernel v.2.0 o più recenti che ha il supporto per la condivisione degli interrupt.

Grazie ancora a Cameron per aver fornito a David Hinds un campione di questo prodotto e la documentazione. Il supporto lo si cerchi nel pacchetto PCMCIA di David.

Si veda la sezione 9.2 (Supporto PCMCIA) per maggiori informazioni sui chipset PCMCIA, abilitatori di socket, ecc.

#### 5.1.11 3c575

Stato: Sconosciuto.

È in sviluppo un driver per questa scheda PCMCIA e si spera che in futuro sia incluso nel pacchetto PCMCIA di David. Per sapere lo stato attuale è meglio controllare nel pacchetto PCMCIA.

#### 5.1.12 3c579

Stato: Supportata, Nome del driver: 3c509

La versione EISA della 509. La versione EISA attuale usa lo stesso chip a 16 bit invece che un interfaccia a 32 bit, quindi l'incremento di prestazioni non è meraviglioso. Ci si assicuri che la scheda sia configurata per la modalità di indirizzamento EISA. Si legga la sezione precedente sulla 3c509 per informazioni sul driver.

### 5.1.13 3c589 / 3c589B

Stato: Semi-supportata, Nome del driver: 3c589

Molti stanno usando questa scheda PCMCIA da un po' di tempo. Si noti che il supporto non è (al momento) incluso nell'albero dei sorgenti del kernel. La "B" nel nome significa la stessa cosa che nel caso della 3c509.

Sono disponibili dei driver nel sito ftp di Donald e nel pacchetto PCMCIA di David Hinds. Sarà necessario avere anche un controller PCMCIA supportato. Si veda la sezione 9.2 (Supporto PCMCIA) per maggiori informazioni su driver, chipset, abilitatori di socket PCMCIA, ecc.

### 5.1.14 3c590 / 3c595

Stato: Supportate, Nome del driver: 3c59x

Queste schede "Vortex" sono per macchine a bus PCI: '590 è l'offerta 3Com per i 10Mbps mentre la '595 è l'offerta per i 100Mbs. Si noti inoltre che si può usare la '595 come una '590 (cioè in modalità a 10Mbps). Il driver è incluso nei sorgenti del kernel v2.0, ma è continuamente aggiornato. Se si hanno problemi con il driver nel kernel v2.0, si può trovare un driver più aggiornato partendo dall'URL seguente:

*Vortex* `<http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html>`

Si noti che in giro si sono due diverse schede 3c590: i primi modelli con 32kB di memoria sulla scheda e gli ultimi modelli che hanno solo 8kB di memoria. È possibile che non si sarà in grado di trovare una nuova 3c59x sul mercato ancora per molto, poiché è stata sostituita con la 3c90x. Se si vuole comprarne una usata da qualcuno, si provi a trovare la versione con 32kB. La 3c595 ha 64kB, in quanto non si può andare molto lontano con soli 8kB a 100Mbps!

Un ringraziamento a Cameron Spitzer e Terry Murphy della 3Com per aver inviato schede e documentazione a Donald cosicché ha potuto scrivere il driver.

Donald ha messo su una mailing list per il supporto al driver Vortex. Per unirsi alla lista, si faccia semplicemente:

```
echo subscribe | /bin/mail linux-vortex-request@cesdis.gsfc.nasa.gov
```

### 5.1.15 3c592 / 3c597

Stato: Supportate, Nome del driver: 3c59x

Queste sono le versioni EISA della serie di schede 3c59x. Le 3c592/3c597 (altrimenti note come Demon) dovrebbero funzionare con il driver vortex discusso in precedenza.

### 5.1.16 3c900 / 3c905 / 3c905B

Status: Supportate, Nome del driver Name: 3c59x

Queste schede (altrimenti note come 'Boomerang' o EtherLink III XL) sono state realizzate per prendere il posto delle schede 3c590/3c595.

Il supporto per la revisione 'B' (Cyclone) è stato aggiunto solo di recente. Per usare questa scheda con i vecchi kernel v2.0, si deve ottenere il driver `3c59x.c` aggiornato dal sito di Donald a:

*Vortex-Page* `<http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html>`

Per qualsiasi dubbio si veda la pagina WWW suddetta. Donald ha messo su una mailing list per il supporto del driver Vortex, annunci, ecc. Per unirsi alla lista, si faccia semplicemente:

```
echo subscribe | /bin/mail linux-vortex-request@cesdis.gsfc.nasa.gov
```

#### 5.1.17 3c985

Stato: Supportata, Nome del driver: acenic

Questo driver di Jes Sorensen è disponibile nei kernel v2.2. Supporta diverse altre schede Gigabit oltre al modello 3Com.

## 5.2 Accton

### 5.2.1 Accton MPX

Stato: Supportata, Nome del driver: ne (+8390)

Non ci si faccia ingannare dal nome. Questa è una scheda compatibile con la NE2000 e dovrebbe funzionare con il driver ne2000.

### 5.2.2 Accton EN1203, EN1207, EtherDuo-PCI

Stato: Supportata, Nome del driver: de4x5, tulip

Questa è un'altra implementazione del chip PCI 21040 della DEC. La scheda EN1207 ha il 21140 e ha pure un connettore 10Base-2, che si è dimostrato causare un po' di problemi a qualcuno nella selezione di quel mezzo. Per altri ha funzionato bene usando la scheda con 10Base-T e 100Base-T. Quindi come per qualsiasi acquisto, la si dobrebbe provare assicurandosi di poterla restituire nel caso non funzioni.

Si veda 5.16.4 (DEC 21040) per maggiori informazioni su queste schede e la situazione attuale del driver.

### 5.2.3 Accton EN2209 Parallel Port Adaptor (EtherPocket)

Stato: Semi supportata, Nome del driver: ?

È disponibile un driver per questi adattatori su porta parallela ma non fa ancora parte dei sorgenti del kernel 2.0 e 2.1. Si deve scaricare il driver da:

`http://www.unix-ag.uni-siegen.de/~nils/accton_linux.html`

### 5.2.4 Accton EN2212 PCMCIA Card

Stato: Semi supportata, Nome del driver: ?

David Hinds stava lavorando su un driver per questa scheda e quindi per conoscere lo stato attuale la cosa migliore è quindi quella di controllare l'ultima release del suo pacchetto PCMCIA.

## 5.3 Allied Telesyn/Telesis

### 5.3.1 AT1500

Stato: Supportata, Nome del driver: lance

Questa è una serie di schede Ethernet a bassa costo che usa la versione 79C960 del chip LANCE dell'AMD. Sono schede bus-master e quindi fra le più veloci schede Ethernet per bus ISA disponibili.

Informazioni sulla selezione del DMA e la numerazione del chip possono essere trovate in 5.4.1 (AMD LANCE).

Informazioni tecniche aggiuntive sulle schede Ethernet basate su AMD LANCE possono essere trovate nella sezione 8.5 (Note su AMD...).

### 5.3.2 AT1700

Stato: Supportata, Nome del driver: at1700

Si noti che per accedere a questo driver durante il `make config` si deve ancora rispondere 'Y' alla domanda iniziale "Prompt for development and/or incomplete code/drivers?". Ciò è dovuto semplicemente allo scarso responso avuto sulla stabilità del driver a causa della relativa rarità della scheda. Se si hanno problemi con il driver distribuito assieme al kernel, può interessare un driver alternativo reperibile a:

`http://www.cc.hit-u.ac.jp/nagoya/at1700/`

La serie di schede Ethernet AT1700 della Allied Telesis sono basate sul chip MB86965 della Fujitsu. Questo chip usa un'interfaccia ad I/O programmato e una coppia di buffer di trasmissione di dimensione fissa. Ciò permette l'invio back-to-back di piccoli gruppi di pacchetti, con una breve pausa durante lo scambio dei buffer.

Una caratteristica unica è l'abilità di pilotare un cavo STP (Shielded Twisted Pair - doppino intrecciato schermato) da 15 Ohm comunemente utilizzato per il Token Ring, oltre ai cavi UTP (unshielded twisted pair - doppino intrecciato non schermato) da 100 Ohm del 10baseT. Della scheda ne esiste pure una versione per fibra ottica (AT1700FT).

Il chip Fujitsu utilizzato nella AT1700 ha un difetto di progettazione: può essere reinizializzato completamente solo operando un ciclo completo di alimentazione della macchina. Premendo solamente il pulsante di reset non si inizializza l'interfaccia. Questo non sarebbe tanto male, tranne per il fatto che può essere rilevata in modo affidabile solo quando è stata reinizializzata. La soluzione/work-around è di spegnere la macchina se il kernel ha problemi a rilevare l'AT1700.

### 5.3.3 AT2450

Stato: Supportata, Nome del driver: pcnet32

Questa è la versione PCI della AT1500 e non soffre dei problemi cha ha la scheda PCI 79c970 della Boca. Informazioni sulla selezione del DMA e la numerazione del chip possono essere trovate nella sezione 5.4.1 (AMD LANCE).

Ulteriori informazioni tecniche sulle schede Ethernet basate sul LANCE dell'AMD possono essere trovate nella sezione 8.5 (Note su AMD...).

### 5.3.4 AT2500

Stato: Semi supportata, Nome del driver: rtl8139

Questa scheda usa il chip 8139 della RealTek. Si veda la sezione 5.31.5 (RealTek 8139).

### 5.3.5 AT2540FX

Stato: Semi supportata, Nome del driver: eepro100

Questa scheda usa il chip i82557 e quindi può/potrebbe funzionare con il driver eepro100. Se la si prova si è invitati ad inviare un rapporto cosicché queste informazioni possano essere aggiornate.

## 5.4 AMD / Advanced Micro Devices

Carl Ching dell'AMD è stato talmente gentile da fornire descrizioni molto dettagliate su tutti i più importati prodotti Ethernet dell'AMD che mi hanno aiutato a chiarire un po' questa sezione.

### 5.4.1 AMD LANCE (7990, 79C960/961/961A, PCnet-ISA)

Stato: Supportata, Nome del driver: lance

In realtà non ci sono schede Ethernet dell'AMD. Probabilmente si sta leggendo questa sezione perché le sole cose che si possono leggere sopra la propria scheda dicono AMD e uno dei numeri suddetti. Il 7990 è il chip 'LANCE' originale, ma molte cose (tra cui questo documento) fanno riferimento a tutti questi chip simili come chip 'LANCE' (...incorrettamente potrei aggiungere).

Questi numeri fanno riferimento a chip dell'AMD che sono il cuore di molte schede Ethernet. Per esempio la AT1500 della Allied Telesis (si veda la sezione 5.3.1 (AT1500)) e la NE1500/2100 (si veda la sezione 5.27.4 (NE1500)) usano questi chip.

Il 7990/79c90 è stato da tempo rimpiazzato da nuove versioni. Il 79C960 (detto anche PCnet-ISA) essenzialmente contiene il nucleo della 79c90, assieme a tutto l'altro hardware di supporto, che permette una soluzione Ethernet in un unico chip. Il 79c961 (PCnet-ISA+) è una versione Plug and Play senza ponticelli della '960. L'ultimo chip della serie ISA è il 79c961A (PCnet-ISA II), che aggiunge piene funzionalità full duplex. Tutte le schede con uno di questi chip dovrebbero funzionare con il driver lance.c, ad eccezione di quelle veramente vecchie che usano il 7990 originale in una configurazione a memoria condivisa. Queste vecchie schede possono essere identificate dalla mancanza di jumper per il canale DMA.

Un problema comune è la ricezione del messaggio 'busmaster arbitration failure'. Questo è mostrato quando il driver LANCE non può accedere al bus dopo che è passato un ragionevole intervallo di tempo (5us). Solitamente indica che l'implementazione del bus-master della scheda madre ha problemi o qualche altro dispositivo sta intasando il bus, oppure c'è un canale DMA in conflitto. Se il proprio BIOS ha l'opzione GAT (che sta per Guaranteed Access Time – tempo di accesso garantito) si provi ad attivare/alterare questa impostazione per vedere se è di qualche aiuto.

Si noti inoltre che il driver cerca una scheda valida solo in questi indirizzi: `0x300, 0x320, 0x340, 0x360` e qualsiasi indirizzo fornito con l'argomento di boot `ether=` è ignorato silenziosamente (questa cosa verrà corretta). Quindi per ora ci si assicuri che la propria scheda sia configurata per uno dei suddetti indirizzi I/O.

Il driver dovrebbe funzionare ancora bene anche se sono installati più di 16 MB di memoria, in quanto, quando serve, sono usati i 'bounce-buffer' in memoria bassa (i.e. qualsiasi dato sopra il 16 MB è copiato dentro un buffer sotto il 16 MB prima di essere dato alla scheda perché lo trasmetta).

Il canale DMA può essere impostato con i bit meno significativi dell'altrimenti inutilizzato valore di dev->mem_start (PARAM_1) (si veda 10.1.1 (PARAM_1)). Se non impostato è rilevato abilitando a turno tutti i canali DMA liberi e controllando se l'inizializzazione ha successo.

La scheda HP-J2405A è un eccezione: con questa scheda è facile leggere i valori impostati nella EEPROM per IRQ e DMA.

Si veda la sezione 8.5 (Note su AMD...) per maggiori informazioni su questi chip.

### 5.4.2 AMD 79C965 (PCnet-32)

Stato: Supportata, Nome del driver: pcnet32

Questa è la PCnet-32: una versione bus-master a 32 bit del chip LANCE originale per sistemi VL-bus e local bus. Sebbene questi chip possano funzionare con il driver `lance.c` standard, è disponibile anche una versione a 32 bit (`pcnet32.c`) che non si preoccupa mai delle limitazioni ai 16 MB associati con il bus ISA.

### 5.4.3 AMD 79C970/970A (PCnet-PCI)

Stato: Supportata, Nome del driver: pcnet32

Questa è la PCnet-PCI: simile alla PCnet-32, ma progettata per i sistemi basati su bus PCI. Si vedano le informazioni sulla PCnet-32. Si noti che è necessario compilare un kernel abilitando il supporto per il PCI. La '970A aggiunge al progetto originale della '970 il supporto per il full duplex oltre ad altre caratteristiche.

Si noti che l'implementazione Boca della 79C970 fallisce su macchine Pentium veloci. È un problema hardware, in quanto affligge anche gli utenti DOS. Si veda la sezione sulla Boca per maggiori dettagli.

### 5.4.4 AMD 79C971 (PCnet-FAST)

Stato: Supportata, Nome del driver: pcnet32

Questo è il chip a 100Mbit per sistemi PCI della AMD, che supporta anche le operazioni full duplex. È stato introdotto nel giugno 1996.

### 5.4.5 AMD 79C972 (PCnet-FAST+)

Stato: Sconosciuto, Nome del driver: pcnet32

Questa dovrebbe funzionare proprio come la '971 ma la cosa non è ancora stata confermata.

### 5.4.6 AMD 79C974 (PCnet-SCSI)

Stato: Supportata, Nome del driver: pcnet32

Questa è la PCnet-SCSI, che in pratica viene trattata come una '970 dal punto di vista Ethernet. Si vedano anche le informazioni precedenti. Non mi si chieda se è supportata la parte SCSI del chip: questo è *Ethernet-HowTo*, non lo SCSI-HowTo.

## 5.5 Ansel Communications

### 5.5.1 AC3200 EISA

Stato: Semi-supportata, Nome del driver: ac3200

Si noti che per accedere a questo driver durante il `make config` si deve ancora rispondere 'Y' alla domanda iniziale "Prompt for development and/or incomplete code/drivers?". Questa cosa è semplicemente dovuta allo scarso responso da parte degli utenti sulla stabilità del driver a causa della relativa rarità della scheda.

Questo driver è incluso nel kernel attuale con un driver in alpha test. È basata sul comune chip NS8390 usato nelle schede ne2000 e wd80x3. Si veda la sezione 3.1 (Driver sperimentali) in questo documento per importanti informazioni riguardanti i driver sperimentali.

Se la si usa, lo si faccia sapere ad uno di noi, poiché il riscontro da parte degli utenti è stato basso, anche se il driver è nel kernel già a partire dalla versione 1.1.25.

Se si intende utilizzare questo driver come modulo caricabile probabilmente si dovrebbe vedere la sezione 10.2 (Usare i driver Ethernet come moduli) per informazioni specifiche sui moduli.

## 5.6 Apricot

### 5.6.1 Apricot Xen-II On Board Ethernet

Stato: Semi supportata, Nome del driver: apricot

Questa scheda Ethernet on board usa un chip i82596 bus-master. Può essere solo all'indirizzo I/O `0x300`. Guardando nei sorgenti del driver, sembra anche che l'IRQ sia fissato via hardware al 10.

Le prime versioni di questo driver avevano la tendenza a pensare che qualsiasi cosa presente a `0x300` fosse una NIC apricot. Da allora l'indirizzo hardware è verificato per evitare questi falsi rilievi.

## 5.7 Arcnet

Stato: Supportata, Nome del driver: arcnet (arc-rimi, com90xx, com20020)

Con il costo ormai veramente basso e le migliori prestazioni di Ethernet, è possibile che molti posti diano via gratis il loro hardware Arcnet, il che risulta in un sacco di sistemi domestici con Arcnet.

Un vantaggio di Arcnet è che tutte le schede hanno la medesima interfaccia cosicché un unico driver funziona per tutte. Hanno pure una gestione degli errori e quindi si può supporre che non perdano mai pacchetti (bellissimo per il traffico UDP!).

Il driver arcnet di Avery Pennarun è nel kernel dalla versione 1.1.80. Il driver arcnet usa 'arc0' come nome invece del solito 'eth0' dei dispositivi Ethernet. Segnalazioni di bug e racconti di successo possono essere inviati a:

`apenwarr@foxnet.net`

Nel kernel standard ci sono file d'informazioni sull'impostazione dei ponticelli e suggerimenti generali.

Si può supporre che il driver funzioni anche con le schede ARCnet a 100Mbs!

## 5.8 AT&T

Si noti che la StarLAN della AT&T è una tecnologia orfana, come la LattisNet della SynOptics e non può essere usata in un ambiente 10Base-T standard, senza un hub che le 'parla' entrambe.

### 5.8.1 AT&T T7231 (LanPACER+)

Stato: Non supportata.

Queste schede StarLAN usano un interfaccia simile a quella del chip i82586. Ad un certo punto Matthijs Melchior (`matthijs.n.melchior@att.com`) si era messo a giocare con il driver 3c507 e quasi aveva qualcosa di funzionante. Non ho sentito più niente da allora.

## 5.9 Boca Research

Sì, fanno molto più che solo schede seriali multiporta. :-)

### 5.9.1 Boca BEN (ISA, VLB, PCI)

Stato: Supportata, Nome del driver: lance, pcnet32

Questa schede sono basate sui chip PCnet dell'AMD. Gli acquirenti accorti dovrebbero essere avvisati che molti utenti hanno avuto un sacco di problemi con queste schede VLB/PCI. Sono stati colpiti specialmente i possessori di sistemi Pentium veloci. Si noti che questo non è un problema del driver, in quanto colpisce anche gli utenti DOS/Win/NT. Il numero del supporto tecnico della Boca è (407) 241-8088 e può essere raggiunto anche a `75300.2672@compuserve.com`. Le vecchie schede ISA non sembra soffrano dello stesso problema.

Donald ha fatto un test comparativo tra una scheda PCI della Boca e una implementazione PCnet/PCI simile della Allied Telsyn e ha rilevato che i problemi dipendono dall'implementazione Boca del chip PCnet/PCI. Si può accedere ai risultati di questi test nel server www di Don.

*Linux at CESDIS* `<http://cesdis.gsfc.nasa.gov/linux/>`

La Boca sta offrendo una 'warranty repair' (ripazione in garanzia) per i possessori di tali schede che consiste nell'aggiunta di un condensatore, ma sembra che la cosa non funzioni al 100% per molte persone, sebbene aiuti un po'.

Se *ancora* si è convinti di comprare una di queste schede, allora almeno si provi ad ottenere una garanzia incondizionata di restituzione entro 7 giorni, cosicché se non funziona bene nel proprio sistema, la si può sempre restituire.

Per ulteriori infomazioni generali sui chip della AMD si veda 5.4.1 (AMD LANCE).

Informazioni tecniche ulteriori sulle schede Ethernet AMD LANCE possono essere trovate nella sezione 8.5 (Note su AMD...).

## 5.10 Cabletron

Donald scrive: «Sì, un'altra di quelle compagnie che non vuole rilasciare le sue informazioni per la programmazione. Hanno aspettato mesi prima di confermare che tutte le loro informazioni erano proprietarie, sprecando deliberatamente il mio tempo. Se si può si evitino queste schede come la peste. Si noti che ad alcuni che hanno telefonato alla Cabletron sono state dette cose del tipo 'un certo D. Becker sta lavorando su un driver per Linux' – facendo sembrare che io lavori per loro. Questo NON è vero.»

Apparentemente la Cabletron ha cambiato la sua politica riguardo le informazioni per la programmazione (come la Xircom) da quando Donald ha fatto quei commenti parecchi anni fa: si mandi una email a `support@ctron.com` se si vuole verificare questa cosa o chiedere le infomazioni per la programmazione. Comunque, a questo punto, c'è una richiesta minima per driver modificati/aggiornati per le vecchie schede E20xx e E21xx.

### 5.10.1 E10**, E10**-x, E20**, E20**-x

Stato: Semi supportate, Nome del driver: ne (+8390)

Queste sono praticamente dei cloni delle NEx000 che si dice funzionino con i driver standard NEx000, grazie a un controllo specifico durante il rilevamento. Se ci sono dei problemi, difficilmente possono essere risolti, poiché non sono disponibili le informazioni per la programmazione.

### 5.10.2 E2100

Stato: Semi supportata, Nome del driver: e2100 (+8390)

Ancora, non si può fare molto quando le informazioni per la programmazione sono proprietarie. Le E2100 ha un pessimo progetto. Ogniqualvolta mappa la sua memoria condivisa durante un trasferimento di pacchetti, la mappa dentro *un'intera regione di 128K*! Ciò significa che in quella regione *non si può* usare con sicurezza

un altro dispositivo gestito a memoria condivisa, nemmeno un'altra E2100. Funzionerà per maggior parte del tempo, ma tutto un tratto non lo farà più (sì, questo problema può essere evitato disabilitando gli interrupt mentre si trasferiscono i pacchetti, ma quasi certamente si perderanno colpi di clock). Inoltre, se si mal programma la scheda o si ferma la macchina proprio al momento sbagliato, nemmeno il bottone di reset sarà utile. Si *deve* spegnere la macchina e *lasciarla* spenta per almeno 30 secondi.

La selezione del mezzo è automatica, ma volendo lo si può imporre con i bit meno significativi del parametro dev->mem_end. Se veda 10.1.1 (PARAM_2). Gli utilizzatori dei moduli possono specificare un valore `xcvr=N` come `options` nel file `/etc/conf.modules`.

Inoltre, non si scambi la E2100 per un clone della NE2100. La E2100 è un NetSemi DP8390 a memoria a condivisa, "rozzamente" simile alla demente della WD8013, mentre la NE2100 (e la NE1500) usano un AMD LANCE con bus-mastering.

Nel kernel standard è incluso un driver per la E2100. Comunque, poiché non sono disponibili informazioni per la programmazione, non si aspettino le correzioni di eventuali bachi. Non se ne usi una a meno che non la si possegga già.

Se si ha intenzione di usare questo driver come modulo caricabile probabilmente si dovrebbe vedere la sezione 10.2 (Usare i driver Ethernet come moduli) per infomazioni specifiche sui moduli.

### 5.10.3 E22**

Stato: Semi supportata, Nome del driver: lance

Secondo le infomazioni in un bollettino tecnico della Cabletron, queste schede usano il chip PC-Net dell'AMD (si veda 5.4.1 (AMD PC-Net)) e dovrebbero quindi funzionare con il driver lance generico.

## 5.11 Cogent

Ecco come e dove possono essere raggiunti:

```
Cogent Data Technologies, Inc.
175 West Street, P.O. Box 926
Friday Harbour, WA 98250, USA.

Cogent Sales
15375 S.E. 30th Place, Suite 310
Bellevue, WA 98007, USA.

Supporto tecnico:
Telefono (360) 378-2929 tra le 8 e 17 PST
Fax (360) 378-2882
Compuserve GO COGENT
Bulletin Board Service (360) 378-5405
Internet: support@cogentdata.com
```

### 5.11.1 EM100-ISA/EISA

Stato: semi supportata, Nome del driver: smc9194

Queste schede usano il chip SMC 91c100 e potrebbero funzionare con il driver per SMC 91c92, ma la cosa non è ancora stata verificata.

### 5.11.2 Cogent eMASTER+, EM100-PCI, EM400, EM960, EM964

Stato: Supportate, Nome del driver: de4x5, tulip

Queste sono un'altra implementazione del DEC 21040 che quindi dovrebbe funzionare tranquillamente con il driver standard per il 21040.

La EM400 e la EM964 sono schede a quattro porte che usano un bridge DEC 21050 e 4 chip 21040.

Si veda 5.16.4 (DEC 21040) per maggiori informazioni su queste schede e la situazione attuale.

## 5.12 Compaq

La Compaq non è veramente in affari per la costruzione di schede Ethernet, ma un sacco di loro sistemi hanno controller Ethernet integrati nella scheda madre.

### 5.12.1 Compaq Deskpro / Compaq XL (Embedded AMD Chip)

Stato: Supportati, Nome del driver: pcnet32

Le macchine come quella della serie XL hanno un chip PCI 79c97x dell'AMD nella scheda madre che può essere usato con il driver LANCE standard. Ma prima di usarlo, si devono utilizzare alcuni trucchetti per far sì che il BIOS PCI si metta in un posto dove Linux lo possa vedere. Frank Maas è stato così gentile da fornire i dettagli operativi:

«Il problema con questa macchina Compaq è che la directory PCI è caricata in memoria alta, in un punto che il kernel di Linux non può (non vuole) raggiungere. Risultato: la scheda non è mai rilevata e nemmeno usabile (altro effetto: non funziona nemmeno il mouse). Il workaround (come descritto approfonditamente in http://www-c724.uibk.ac.at/XL/) è di caricare MS-DOS, lanciare un piccolo driver che ha scritto la Compaq e poi caricare il kernel di Linux usando LOADLIN. Ok, vi do il tempo per dire 'yuck, yuck', ma per ora questo è il solo metodo che conosco. Il driver semplicemente sposta la directory PCI nel posto dov'è di solito (e dove Linux la può trovare).»

Ulteriori informazioni generali sui chip AMD possono essere trovate in 5.4.1 (AMD LANCE).

### 5.12.2 Compaq Nettelligent/NetFlex (Embedded ThunderLAN Chip)

Stato: Supportata, Nome del driver: tlan

Questi sistemi usano un chip ThunderLAN della Texas Instruments. Informazioni sul driver ThunderLAN possono essere trovare in 5.36.1 (ThunderLAN).

## 5.13 Danpex

### 5.13.1 Danpex EN9400

Stato: Supportata, Nome del driver: de4x5, tulip

Ecco un'altra scheda basata sul chip 21040 della DEC, che si dice funzioni bene e che costa relativamente poco.

Si veda 5.16.4 (DEC 21040) per maggiori informazioni su queste schede e l'attuale situazione del driver.

## 5.14 D-Link

### 5.14.1 DE-100, DE-200, DE-220-T, DE-250

Stato: Supportata, Nome del driver: ne (+8390)

Alcune delle prime schede D-Link non avevano la firma (signature) `0x57` nella PROM, ma il driver ne2000 ne è conscio. Per quanto riguarda le schede configurabili via software, si può scaricare il programma di configurazione da `www.dlink.com`. Le schede DE2** erano le maggiormente segnalate per avere errori spuri di mismatch nell'indirizzo di trasferimento con le prime versioni di Linux. Si noti che esistono anche schede della Digital (DEC) chiamate DE100 e DE200, ma la somiglianza finisce qui.

### 5.14.2 DE-520

Stato: Supportata, Nome del driver: pcnet32

Questa è una scheda PCI che usa la versione PCI del chip LANCE dell'AMD. Informazioni sulla selezione del DMA e la numerazione del chip possono essere trovate nella sezione 5.4.1 (AMD LANCE).

Ulteriori informazioni tecniche sulle schede Ethernet basate sul LANCE dell'AMD le si può trovare nella sezione 8.5 (Note su AMD...).

### 5.14.3 DE-528

Stato: Supportata, Nome del driver: ne, ne2k-pci (+8390)

Apparentemente la D-Link ha cominciato a fare anche cloni PCI della NE2000.

### 5.14.4 DE-530

Stato: Supportata, Nome del driver: de4x5, tulip

Questa è un'implementazione del chip DEC 21040 ed è riportato che funziona con il driver generico tulip per il 21040.

Si veda la sezione 5.16.4 (DEC 21040) per maggiori informazioni su queste schede e sullo stato attuale del driver.

### 5.14.5 DE-600

Stato: Supportata, Nome del driver: de600

Ai possessori di un portatile e a tutti quelli che vorrebbero un metodo veloce per mettere il loro computer su Ethernet piacerà usare questa schede. Il driver è incluso nei sorgenti del kernel standard.

Il driver è stato scritto da Bjorn Ekwall `bj0rn@blox.se`. Ci si aspetti una velocità di trasferimento di circa 180kb/s da questa scheda attraverso la porta parallela. Si dovrebbe leggere il file README.DLINK nell'albero dei sorgenti del kernel.

Si noti che il nome del device da passare a `ifconfig` *ora* è `eth0` e non più il `dl0` usato in precedenza.

Se la propria porta parallela *non* è all'indirizzo standard `0x378` allora si deve ricompilare. Bjorn scrive: «Poiché il driver DE-620 prova a spremere anche l'ultimo microsecondo dal ciclo, ho reso l'irq e l'indirizzo delle costanti piuttosto che delle variabili. Questo consente di ottenere una velocità usabile, ma significa anche che non si possono modificare questi assegnamenti p.es. da lilo; si _deve_ ricompilare...». Si noti

inoltre che alcuni portatitili implementano la porta parallela on board a `0x3bc` che è dove erano/sono le porte parallele sulle schede monocromatiche.

#### 5.14.6 DE-620

Stato: Supportata, Nome del driver: de620

Analoga alla DE-600, con solo due formati d'uscita. Bjorn ha scritto un driver per questo modello per le versioni del kernel pari e superiori alla 1.1. Si vedano le informazioni precedenti sulla DE-600.

#### 5.14.7 DE-650

Stato: Semi supportata, Nome del driver: de650 (?)

Alcuni hanno usato questa scheda PCMCIA per abbastanza tempo nei loro computer portatili. In pratica è un 8390, come una NE2000. Si suppone che anche la scheda PCMCIA LinkSys e la IC-Card Ethernet siano dei cloni della DE-650. Si noti che al momento questo driver *non* è parte del kernel standard, e quindi si devono applicare alcune patch.

Si veda 9.2 (Supporto PCMCIA) in questo documento e, se si può, si dia un'occhiata a:

*Don's PCMCIA Stuff* `<http://cesdis.gsfc.nasa.gov/linux/pcmcia.html>`

### 5.15 DFI

#### 5.15.1 DFINET-300 e DFINET-400

Stato: Supportate, Nome del driver: ne (+8390)

Ora queste schede sono rilevate (sin dal 0.99pl15) grazie a Eberhard Moenkeberg `emoenke@gwdg.de` che ha notato che usano 'DFI' nei primi 3 byte della PROM, invece di usare `0x57` nei byte 14 e 15, che è quello che fanno tutte le schede NE1000 e NE2000 (la 300 è un pseudo-clone della NE1000 a 8 bit e la 400 lo è della NE2000).

### 5.16 Digital / DEC

#### 5.16.1 DEPCA, DE100/1, DE200/1/2, DE210, DE422

Stato: Supportate, Nome del driver: depca

Nel file sorgente 'depca.c' è inclusa della documentazione, che comprende informazioni su come usare più di una di queste schede in una macchina. Si noti che la DE422 è una scheda EISA. Queste schede sono tutte basate sul chip LANCE dell'AMD. Si veda la sezione 5.4.1 (AMD LANCE) per maggiori informazioni. Possono essere usate sino ad un massimo di due schede ISA, perché possono essere impostate solamente per gli indirizzi di I/O base `0x300` e `0x200`. Se si intende farlo, si leggano le note nel file sorgente del driver `depca.c` nell'albero dei sorgenti del kernel standard.

Questo driver funzionerà anche sulle macchine basate su CPU Alpha e ci sono diverse ioctl() con le quali l'utente può giocare.

#### 5.16.2 Digital EtherWorks 3 (DE203, DE204, DE205)

Stato: Supportata, Nome del driver: ewrk3

Queste schede usano un chip proprietario della DEC, invece del chip LANCE utilizzato nelle prime schede come la DE200. Queste schede supportano sia la memoria condivisa che l'I/O programmato, sebbene raggiungano una calo di prestazioni del 50% se si usa la modalità PIO. La dimensione della memoria condivisa può essere impostata a 2kB, 32kB o 64kB, ma con questo driver sono state testate solamente le prime due dimensioni. David dice che le prestazioni sono virtualmente identiche tra il modo a 2kB e quello a 32kB. Maggiori informazioni (tra cui l'uso del driver come modulo caricabile) sono presenti all'inizio del file sorgente `ewrk3.c` e anche nel file `README.ewrk3`. Ambedue i file sono nella distribuzione standard del kernel. Questo driver ha il supporto per le CPU Alpha, come il depca.c.

Il driver standard ha diverse chiamate ioctl() interessanti che possono essere usate per ottenere ed azzerare le statistiche sui pacchetti, leggere/scrivere la EEPROM, cambiare l'indirizzo hardware e cose così. Gli smanettoni possono vedere il codice sorgente per maggiori informazioni su queste possibilità.

David ha scritto anche un programma di configurazione per questa scheda (sulla falsa riga del programma DOS `NICSETUP.EXE`) assieme con altri strumenti. Possono essere trovati in praticamente tutti i siti FTP di Linux nella directory `/pub/Linux/system/Network/management` (si cerchi il file `ewrk3tools-X.XX.tar.gz`).

### 5.16.3 DE425 EISA, DE434, DE435, DE500

Stato: Supportate, Nome del driver: de4x5, tulip

Queste schede si basano sul chip 21040 menzionato sotto. La DE500 usa il chip 21140 per fornire connessioni Ethernet a 10/100Mbs. Si deve leggere la sezione sul 21040 qui sotto per ulteriori informazioni. Ci sono alcuni opzioni da passare in compilazione per le schede non DEC che usano questo driver. Si veda il file `README.de4x5` per i dettagli.

Tutte le schede Digital rileveranno automaticamente il mezzo (tranne, temporaneamente, la DE500 a causa di un brevetto).

Questo driver è pronto anche per le CPU Alpha e può essere caricato come modulo. Gli utenti possono raggiungere l'interno del driver attraverso chiamate ioctl. Si vedano gli strumenti 'ewrk3' e il sorgente de4x5.c per informazioni su come farlo.

### 5.16.4 DEC 21040, 21041, 2114x, Tulip

Stato: Supportate, Nome del driver: de4x5, tulip

Il DEC 21040 è una soluzione Ethernet bus-mastering in un unico chip, simile al chip PCnet dell'AMD. Il 21040 è progettato specificatamente per l'architettura di bus PCI. La nuova scheda PCI EtherPower della SMC usa questo chip. Per le schede basate su questo chip è possibile scegliere tra *due* driver. C'è il driver DE425 discusso in precedenza e driver generico 'tulip' per il 21040.

**Attenzione:** Sebbene la propria scheda possa essere basata su questo chip, *nel proprio caso i driver potrebbero non funzionare*. David C. Davies scrive:

«Non c'è alcuna garanzia che il 'tulip.c' o il 'de4x5.c' funzionino con una qualsiasi scheda basata sui DC2114x oltre a quelle per cui sono stati scritti. PERCHÉ?? Perché c'è un registro, il General Purpose Register (CSR12) che (1) nel DC21140A è programmabile da qualsiasi produttore e tutti lo fanno in modo differente (2) nei DC21142/3 c'è ora un registro di controllo SIA (a la DC21041). Il solo piccolo raggio di speranza è che riusciamo a decodificare la SROM per aiutare l'impostazione nel driver. Comunque, questa non è una soluzione garantita in quanto alcuni produttori (per esempio la scheda SMC 9332) non seguono le raccomandazioni Digital Semiconductor sul formato di programmazione della SROM.»

In termini meno tecnici, ciò significa che se non si è sicuri che una scheda sconosciuta con un chip DC2114x funzionerà con i driver per Linux, allora ci si assicuri di porterla restituire dove la si è comprata *prima* di pagare.

Nella maggior parte delle ultime schede EtherPower della SMC al posto del 21040 si può trovare il più aggiornato chip 21041. Il 21140 è per supportare il 100Base-? e funziona con i driver per Linux per il chip 21040. Per usare il driver `de4x5` di David con una scheda non DEC, si deve guardare il file `README.de4x5` per i dettagli.

Donald ha usato le schede EtherPower-10/100 della SMC per sviluppare il driver 'tulip'. Si noti che il driver presente nel albero standard dei sorgenti al momento non è la versione più aggiornata. Se si hanno problemi con questo driver, ci si dovrebbe procurare la versione più nuova dal sito ftp/WWW di Donald.

*Tulip Driver* `<http://cesdis.gsfc.nasa.gov/linux/drivers/tulip.html>`

Questo URL contiene anche un elenco (non esaustivo) delle diverse schede/vendor che usano il chip 21040.

Si noti inoltre che il driver tulip al momento è ancora considerato un driver sperimentale (si veda la sezione 3.1 (Driver sperimentali)) e dovrebbe essere trattato come tale. Per usarlo, si dovrà modificare il file `arch/i386/config.in` e decommentare la riga per il supporto `CONFIG_DEC_ELCP`.

Donald ha messo su anche una mailing list per gli annunci sul driver tulip, ecc. Per unirsi alla lista si digiti:

```
echo subscribe | /bin/mail linux-tulip-request@cesdis.gsfc.nasa.gov
```

## 5.17 Farallon

La Farallon vende gli adattatori e i transceiver EtherWave. Questo dispositivo permette di mettere in daisy chain diversi dispositivi 10baseT.

### 5.17.1 Farallon Etherwave

Stato: Supportato, Nome del driver: 3c509

È riportato che questo driver sia un clone del 3c509 che include il transceiver EtherWave. La gente l'ha usata con successo con Linux e l'attuale driver 3c509. Sono troppo costose per l'uso comune, ma sono una grande opzione in casi speciali. I prezzi degli hublet partono da $125, e la Etherwave aggiunge $75-$100 al prezzo della scheda

## 5.18 Fujitsu

Diversamente da molti costruttori di chip di rete, la Fujitsu ha fatto e venduto anche alcune schede di rete basate sui loro chip.

### 5.18.1 Fujitsu FMV-181/182/183/184

Stato: Supportato, Nome del driver: fmv18x

Secondo quel che afferma il driver, queste schede sono semplicemente un'implementazione del MB86965 della Fujitsu, il che le rende molto simili alle schede AT1700 della Allied Telesis.

## 5.19 Hewlett Packard

Le schede 272** usano l'I/O programmato, similmente alle schede NE*000, ma la porta del trasferimento dei dati può essere 'spenta' quando non vi si accede, evitando problemi con i driver che fanno l'autorilevamento.

Grazie a Glenn Talbott per l'aiuto fornito nel chiarire la confusione in questa sezione a proposito dei numeri di versione dell'hardware HP.

### 5.19.1 27245A

Stato: Supportata, Nome del diver: hp (+8390)

Una 10BaseT a 8 Bit basata sul 8390, non è raccomandata per tutte quelle ragioni delle 8 bit. È stata riprogettata un paio di anni fa per essere altamente integrata e ciò ha causato alcune modifiche nei tempi di inizializzazione che affliggono solo i programmi di test, non i driver LAN (la nuova scheda non è 'pronta' subito dopo si entra ed esce dalla modalità loopback).

Se si intende usare questo driver come modulo caricabile probabilmente si dovrebbe vedere la sezione 10.2 (Usare i driver Ethernet come modulo) per informazioni specifiche sui moduli.

### 5.19.2 HP EtherTwist, PC Lan+ (27247, 27252A)

Stato: Supportate, Nome del driver: hp+ (+8390)

La HP PC Lan+ è diversa dalla scheda HP PC Lan standard. Questo driver è stato aggiunto all'elenco dei driver del kernel standard durante il ciclo di sviluppo del v1.1.x. Può essere fatta funzionare in modalità PIO come una ne2000, o in modalità a memoria condivisa come una wd8013.

La 47B è una scheda a 16 bit 10BaseT con AUI a 16 Bit basata su 8390 e la 52A è una 16 bit ThinLan con AUI basata su 8390. Queste schede hanno una RAM da 32K per la bufferizzazione dei pacchetti da trasmettere/ricevere invece dei soliti 16KB e entrambe offrono il rilievo automativo del connettore LAN.

Se si intende usare questo driver come modulo caricabile probabilmente si dovrebbe vedere la sezione 10.2 (Usare i driver Ethernet come modulo) per informazioni specifiche sui moduli.

### 5.19.3 HP-J2405A

Stato: Supportata, Nome del driver: lance

Queste schede costano meno e sono un po' più veloci delle 27247/27252A, ma mancano di alcune caratteristiche quali AUI, la connettività ThinLAN e il boot PROM socket. Sono una versione del LANCE abbastanza generica, ma piccole diversità nel progetto le rendono incompatibili con il driver 'NE2100' generico. Il supporto speciale per queste schede (comprensivo della lettura del canale DMA dalla scheda) è incluso grazie alle informazioni fornite da Glenn Talbott dell'HP.

Ulteriori informazioni tecniche sulle schede basate su LANCE possono essere trovate nella sezione 8.5 (Note su AMD...)

### 5.19.4 HP-Vectra On Board Ethernet

Stato: Supportata, Nome del driver: lance

L'HP-Vectra ha un chip PCnet dell'AMD sulla piastra madre. Informazioni sulla selezione del DMA e la numerazione del chip possono essere trovate nella sezione 5.4.1 (AMD LANCE).

Ulteriori informazioni tecniche sulle schede basate su LANCE possono essere trovate nella sezione 8.5 (Note su AMD...)

### 5.19.5 Schede HP 10/100 VG Any Lan (27248B, J2573, J2577, J2585, J970, J973)

Stato: Supportate, Nome del driver: hp100

Questo driver supporta anche alcuni dei prodotti VG della Compex. Poiché il driver supporta schede ISA, EISA e PCI, quando si esegue `make config` sui sorgenti del kernel lo si trova nella sezione relativa alle schede ISA.

#### 5.19.6 HP NetServer 10/100TX PCI (D5013A)

Stato: Supportata, Nome del driver: eepro100

Apparentemente queste sono semplicemente delle schede EtherExpress Pro 10/100B dell'Interl rimarchiate. Si veda la sezione su Intel per maggiori informazioni.

### 5.20 IBM / International Business Machines

#### 5.20.1 IBM Thinkpad 300

Stato: Supportato, Nome del driver: znet

È compatibile con la Z-note della Zenith basata su Intel. Si veda 5.42.1 (Z-note) per maggiori informazioni.

Suppongo che questo sito abbia un esauriente elenco di cosette utili per le nuove versioni del ThinkPad. Ancora non l'ho verificato personalmente.

*Thinkpad-info* <http://peipa.essex.ac.uk/html/linux-thinkpad.html>

Quelli che non possono usare un browser WWW, provino a `peipa.essex.ac.uk:/pub/tp750/`

#### 5.20.2 IBM Credit Card Adaptor for Ethernet

Stato: Semi-supportato, Nome del driver: ? (distribuito separatamente)

Molti stanno usando questa scheda PCMCIA con Linux. Valgono le solite cose, come la necessità di avere nel proprio portatile un chipset PCMCIA supportato e che si deve applicare una patch al kernel standard per il supporto PCMCIA.

Si veda 9.2 (Supporto PCMCIA) in questo documento e se si può si dia un'occhiata a:

*Don's PCMCIA Stuff* <http://cesdis.gsfc.nasa.gov/linux/pcmcia.html>

#### 5.20.3 IBM Token Ring

Stato: Semi supportata, Nome del driver: ibmtr

Il supporto per il token ring richiede molto più che la sola scrittura di un driver per i dispositivi. Richiede anche la scrittura delle routine di instradamento per il token ring. È nella scrittura di queste ultime che si perde la maggior parte del tempo.

Peter De Schrijver ha speso un po' di tempo sul Token Ring. Ha lavorato con schede Token Ring ISA e MCA dell'IBM.

L'attuale codice per il token ring è stato incluso all'inizio dello sviluppo dei kernel della serie 1.3.x.

Peter dice che è stato originariamente testato su schede MCA 16/4 Megabit Token Ring, ma dovrebbe funzionare con altre schede basate sul chip Tropic.

## 5.21 Schede Ethernet ICL

### 5.21.1 ICL EtherTeam 16i/32

Stato: Supportata, Nome del driver: eth16i

Questo driver l'ha scritto Mika Kuoppala (*miku@pupu.elt.icl.fi*) ed è stato introdotto verso il kernel 1.3.4x. Usa il chip MB86965 della Fujitsu, usato anche nelle schede at1700.

## 5.22 Schede Ethernet Intel

Si noti che la nomenclatura delle diverse schede Intel è ambigua e confusa al massimo. Se si hanno dubbi, si veda il numero `i8xxxx` sul chip principale della scheda oppure, per le schede PCI, si usino le informazioni sul PCI nella directory `/proc` e poi si confronti con i numeri elencati qui.

### 5.22.1 Ether Express

Stato: Supportata, Nome del driver: eexpress

Questa scheda usa il i82586 dell'Intel. Le prime versioni di questo driver (nei kernel v1.2) erano classificate come sperimentali e non funzionavano bene per la maggior parte della gente. Quelli che l'hanno provato dicono che il driver nei kernel v2.0 sembra funzionare molto meglio, sebbene il driver sia ancora sperimentale e un po' problematico sulle macchine più veloci.

I commenti all'inizio del sorgente del driver elencano alcuni problemi (e correzioni!) associati con queste schede. Il trucchetto di rimpiazzare nel driver tutti gli `outb` con `outb_p` si dice abbia funzionato per più di un utente.

### 5.22.2 Ether Express PRO/10

Stato: Supportata, Nome del driver: eepro

Bao Chau Ha ha scritto un driver per queste schede incluso nei primi kernel v1.3.x. Può funzionare anche con alcuni sistemi Ethernet integrati della Compaq basati sul chip i82595.

### 5.22.3 Ether Express PRO/10 PCI (EISA)

Stato: Semi supportata, Nome del driver: ? (distribuito separatamente)

John Stalba (stalba@ultranet.com) ha scritto un driver per la versione PCI. Queste schede usano il chip d'interfaccia PLX9036 PCI con un chip i82596 LAN controller della Intel. Se la propria scheda ha il chip i82557, allora `non` si possiede questa scheda ma piuttosto la versione discussa dopo e quindi si deve usare il driver EEPro100.

Si può ottenere il driver sperimentale per la propria scheda PCI PRO/10 assieme alle instruzioni per usarlo a:

*EEPro10 Driver* `<http://www.ultranet.com/~stalba/eep10pci.html>`

Se si ha una scheda EISA, probabilmente si dovrà lavorare un po' sul driver per tener conto delle differenze (PCI vs. EISA) nei meccanismi di rilievo usati nei due casi.

### 5.22.4 Ether Express PRO 10/100B

Stato: Supportata, Nome del driver: eepro100

Si noti che questo driver *non* funzionerà con le vecchie schede 100A. I numeri dei chip elencati nel driver sono i82557/i82558. Per aggiornamenti del driver e/o supporto sul driver, si veda

*EEPro-100B Page* <http://cesdis.gsfc.nasa.gov/linux/drivers/eepro100.html>

Per iscriversi alla mailing list relativa a questo driver, si faccia:

```
echo subscribe | /bin/mail linux-eepro100-request@cesdis.gsfc.nasa.gov
```

Apparentemente Donald ha firmato un accordo di non-disclosure che dice che lui non può effettivamente rivelare il codice sorgente del driver! Che cosa dire di questa insulsaggine da parte di Intel?

## 5.23 Kingston

La Kingston fa diverse schede, tra cui schede basate su NE2000+, AMD PCnet e DEC tulip. La maggior parte di queste schede dovrebbero funzionare bene con i rispetti driver. Si veda la *pagina web della Kingston* <http://www.kingston.com>

La KNE40 basata sul 21041 tulip della DEC si dice funzioni bene con il driver tulip generico.

## 5.24 LinkSys

La LinkSys ha fatto una manciata di diversi cloni NE2000, alcuni sono schede ISA semplici, altre ISA plug-and-play e altri ancora sono cloni PCI della ne2000 basati su uno dei chip PCI ne2000 supportati. Semplicemente ci sono troppi modelli per elencarli tutti qui.

Le LinkSys sono "Linux-friendly" e c'è pure una pagina WWW specifica per il supporto di Linux e hanno anche la parola Linux stampata sulla scatola di alcuni dei loro prodotti. Si veda:

`http://www.linksys.com/support/solution/nos/linux.htm`

### 5.24.1 Schede LinkSys Etherfast 10/100.

Stato: Supportate, Nome del driver: tulip

Si noti che queste schede sono state sottoposte a parecchie 'revisioni' (ovvero sono stati usati diversi chip) tutte sotto lo stesso nome. La prima usava il chip della DEC. La seconda revisione usava il PNIC 82c168 PCI della Lite-On e il supporto per questo era stato aggiunto al supporto per il driver tulip standard (dalla versione 0.83). Ulteriori informazioni sul PNIC sono disponibili a:

`http://cesdis.gsfc.nasa.gov/linux/drivers/pnic.html`

Altre informazioni sulle diverse versioni di queste schede possono essere trovate nella pagina WWW della LinkSys menzionata in precendenza.

### 5.24.2 LinkSys Pocket Ethernet Adapter Plus (PEAEPP)

Stato: Supportata, Nome del driver: de620

Questo è un clone (o supposto tale) nel DE-620 e si dice funzioni bene con quel driver. Si veda 5.14.6 (DE-620) per maggiori informazioni.

### 5.24.3 LinkSys PCMCIA Adaptor

Stato: Supportato, Nome del driver: de650 (?)

Questa è una versione rimarchiata del DE-650. Si veda 5.14.7 (DE-650) per maggiori informazioni.

## 5.25 Microdyne

### 5.25.1 Microdyne Exos 205T

Stato: Semi supportata, Nome del driver: ?

Un'altra scheda basata sul chip i82586. Dirk Niggemann `dirk-n@dircon.co.uk` ha scritto un driver che lui classifica come "pre-alpha" e vorrebbe che la gente testasse. Gli si scriva per maggiori dettagli.

## 5.26 Mylex

La Mylex può essere raggiunta ai seguenti numeri, nel caso qualcuno voglia chiedere qualcosa.

```
MYLEX CORPORATION, Fremont
Sales:  800-77-MYLEX, (510) 796-6100
FAX:    (510) 745-8016.
```

Hanno anche un sito web: *Mylex WWW Site* `<http://www.mylex.com>`

### 5.26.1 Mylex LNE390A, LNE390B

Stato: Supportata, Nome del driver: lne390 (+8390)

Queste sono delle schede EISA piuttosto vecchie che fanno uso di un'implementazione a memoria condivisa simile alla wd80x3. Nella serie attuale 2.1.x del kernel è presente un driver per queste schede. Ci si assicuri di impostare l'indirizzo della memoria condivisa sotto il primo MB o oltre il più alto indirizzo della memoria RAM fisicamente installata nella macchina.

### 5.26.2 Mylex LNP101

Stato: Supportata, Nome del driver: de4x5, tulip

Questa è una scheda PCI basata sul chip 21040 della DEC. È selezionabile con uscita 10BaseT, 10Base2 e 10Base5. Si è verificato che la scheda LNP101 funziona con il driver 21040 generico.

Si veda la sezione sul chip 21040 (5.16.4 (DEC 21040)) per maggiori informazioni.

### 5.26.3 Mylex LNP104

Stato: Semi supportata, Nome del driver: de4x5, tulip

La LNP104 usa il chip 21050 della DEC per gestire *quattro* porte 10BaseT indipendenti. Dovrebbe funzionare con i driver 21040 recenti che sanno come condividere gli IRQ, ma nessuno ha ancora riportato di averci provato (a quanto ne so).

## 5.27 Novell Ethernet, NExxxx e cloni associati

Il prefisso 'NE' viene da Novell Ethernet. La Novell ha seguito il progetto più economico del databook della National Semiconductor e vende i diritti di produzione Eagle, solo per immettere sul mercato schede Ethernet a prezzi ragionevoli (la ormai ubiqua scheda NE2000).

### 5.27.1 NE1000, NE2000

Stato: Supportata, Nome del driver: ne (+8390)

La ne2000 è ora il nome generico del progetto base fatto attorno al chip 8390 della NatSemi. Usano l'I/O programmato piuttosto che la memoria condivisa, il che comporta maggiore facilità di installazione ma prestazioni un po' più basse e un po' di problemi. Alcuni dei problemi più comuni che capitano con le schede NE2000 sono elencati in 3.4 (Problemi con...).

Alcuni cloni della NE2000 usano il chip 'AT/LANTic' 83905 della National Semiconductor, che offre una modalità a memoria condivisa simile a quella della wd8013 e la configurazione via software della EEPROM. La modalità a memoria condivisa permette un minor utilizzo della CPU (cioè è più efficiente) rispetto alla modalità ad I/O programmato.

In generale non è una buona idea mettere un clone della NE2000 all'indirizzo I/O `0x300` perché praticamente *tutti* i driver di dispositivo cercano lì al boot. Alcuni cloni NE2000 "poveri" non la prendono bene ad essere rilevati in aree sbagliate e risponderanno bloccando la macchina. Inoltre anche `0x320` non va bene perché i driver SCSI rilevano a `0x330`.

Donald ha scritto un progamma di diagnostica per NE2000 (ne2k.c) che va bene per tutte le schede ne2000. Si veda la sezione 7.2 (Programmi diagnostici) per maggiori informazioni.

Se si intende usare questo driver con modulo caricabile probabilmente si dovrebbe vedere la sezione 10.2 (Usare i driver Ethernet come moduli) per informazioni specifiche sui moduli.

### 5.27.2 NE2000-PCI (RealTek/Winbond/Compex)

Stato: Supportata, Nome del driver: ne, ne2k-pci (+8390)

Sì, che ci si creda o no, la gente sta facendo schede PCI basate su progetto dell'interfaccia dell'ne2000 che ha ormai più di 10 anni. Al momento praticamente tutte queste schede sono basate sul chip 8029 della RealTek, o sul chip 89c940 della Winbond. Anche le schede Compex, KTI, VIA e Netvin apparentemente usano questi chip, ma hanno un diverso ID PCI.

L'ultimo kernel v2.0 ha il supporto per rilevare automaticamente tutte queste schede ed usarle (se si sta usando un kernel v2.0.34 o più vecchio, lo si dovrebbe aggiornare per assicurarsi che la propria scheda sia rilevata). Ora ci sono due driver tra cui scegliere: l'originale driver ISA/PCI `ne.c` o quello relativamente nuovo solo PCI `ne2k-pci.c`.

Per usare il driver ISA/PCI originale si deve rispondere 'Y' all'opzione 'Other ISA cards' quando si lancia `make config` perché in realtà si sta usando lo stesso driver NE2000 che usano le schede ISA (la qual cosa dovrebbe suggerire che queste schede non sono così intelligenti come può essere, ad esempio, una PCNet-PCI o una DEC 21040...).

Il nuovo driver solo PCI differisce dal driver ISA/PCI nel fatto che tutto il supporto per le vecchie schede a 8 bit NE1000 è stato rimosso e che i dati sono spostati da e nella scheda in blocchi più grandi, senza alcuna pausa tra un'operazione e l'altra che le vecchie NE2000 ISA richiedono per operare in maniera affidabile. Il risultato è un driver più efficiente, ma non ci si ecciti troppo in quanto le differenze non saranno così ovvie nel funzionamento normale (se si vuole veramente la massima efficenza assieme ad un basso utilizzo

della CPU, allora una NE2000 PCI è semplicemente una scelta povera). Aggiornamenti del driver e ulteriori informazioni possono essere trovate a:

`http://cesdis.gsfc.nasa.gov/linux/drivers/ne2k-pci.html`

Se si possiede una scheda PCI NE2000 che *non* è rilevata dalla versione più recente del driver, si contatti il manutentore del driver NE2000 citato nel file `/usr/src/linux/MAINTAINERS` e gli si spedisca l'output di `cat /proc/pci` e `dmesg` dimodoché possa aggiungere il supporto per la scheda.

Si noti inoltre che diversi prodottori di schede sono noti per attaccare l'etichetta 'NE2000 Compatible' sulle scatole dei loro prodotti anche se sono completamente differenti (e.g. PCNet-PCI o RealTek 8139). Se si è in dubbio si controlli il numero del chip principale con quelli in questo documento.

### 5.27.3 NE-10/100

Stato: Non supportata.

Queste sono delle schede ISA a 100Mbps basate sui chip DP83800 e DP83840 della National Semiconductor. Attualmente non c'è alcun driver che le supporta né nessuno ha ancora detto che sta lavorando su un driver. Apparentemente non è disponibile la documentazione sul chip ad eccezione di un file PDF che non fornisce abbastanza dettagli per scrivere un driver.

### 5.27.4 NE1500, NE2100

Stato: Supportate, Nome del driver: lance

Queste schede usano il chip LANCE 7990 originale dell'AMD e sono supportate usando il driver lance di Linux. I cloni NE2100 più nuovi usano il chip aggiornato PCnet-ISA dell'AMD.

Alcune delle prime versioni del driver lance avevano problemi con la determinazione della linea IRQ attraverso l'autoIRQ delle schede Novell/Eagle 7990. Teoricamente questa cosa ora è stata corretta. Se non lo fosse, allora si specifichi l'IRQ usando LILO e si renda noto che questo è ancora un problema.

Informazioni sulla selezione del DMA e la numerazione del chip possono essere trovate nella sezione 5.4.1 (AMD LANCE).

Ulteriori informazioni tecniche sulle schede basate su LANCE si trovano nella sezione 8.5 (Note su AMD...)

### 5.27.5 NE/2 MCA

Stato: Semi supportata, Nome del driver: ne2

C'erano anche un po' di schede NE2000 microchannel fatte da diverse compagnie. Questo driver, disponibile nei kernel v2.2, rileverà le seguenti schede MCS: Novell Ethernet Adapter NE/2, Compex ENET-16 MC/P e Arco Ethernet Adapter AE/2.

### 5.27.6 NE3200

Stato: Non supportata

Questa vecchia scheda EISA usa un 80186 a 8Mhz assieme con un i82586. Nessuno sta lavorando su un driver in quanto non ci sono né informazioni disponibili sulla scheda né una reale richiesta per un driver.

#### 5.27.7 NE3210

Stato: Supportata, Nome del driver: ne3210 (+8390)

Questa scheda EISA è completamente diversa dalla NE3200, poiché usa un chip 8390 della Nat Semi. Il driver può essere trovato nell'albero dei sorgenti del kernel v2.2. Ci si assicuri si impostare l'indirizzo della memoria condivisa sotto il primo MB o superiore all'indirizzo più alto della RAM fisica installata nella macchina.

#### 5.27.8 NE5500

Stato: Supportata, Nome del driver: pcnet32

Queste sono semplicemente delle schede con il chip PCnet-PCI ('970A) dell'AMD. Maggiori informazioni sulle schede basate su LANCE/PCnet possono essere trovate nella sezione 5.4.1 (AMD LANCE).

### 5.28 Proteon

#### 5.28.1 Proteon P1370-EA

Stato: Supportata, Nome del driver: ne (+8390)

Apparentemente questa è un clone NE2000 e funziona bene con Linux.

#### 5.28.2 Proteon P1670-EA

Stato: Supportata, Nome del driver: de4x5, tulip

Questa è un'altra scheda PCI basata sul chip Tulip della DEC. Si dice funzioni bene con Linux.

Si veda la sezione sul chip 21040 (5.16.4 (DEC 21040)) per maggiori informazioni sul driver.

### 5.29 Pure Data

#### 5.29.1 PDUC8028, PDI8023

Stato: Supportata, Nome del driver: wd (+8390)

Le serie di schede PDUC8028 e PDI8023 della Pure Data si dice funzionino, grazie a uno speciale codice di rilevamento fornito da Mike Jagdis `jaggy@purplet.demon.co.uk`. Il supporto è integrato con il driver WD.

### 5.30 Racal-Interlan

La Racal Interlan può essere raggiunta via WWW a `www.interlan.com`. Credo che in passato fosse conosciuta anche come MiCom-Interlan.

#### 5.30.1 ES3210

Stato: Semi supportata, Nome del driver: es3210

Questa è una scheda EISA a memoria condivisa basata su 8390. Con il kernel v2.2 è distribuito un driver sperimentale e si dice funzioni bene, ma il rilevamento EISA dell'IRQ e dell'indirizzo della memoria condivisa non senbra funzionare bene con (almeno) le prime revisioni della scheda (comunque questo problema non è

ristretto al solo mondo Linux...). In quel caso, si devono fornire tali informazioni al driver. Per esempio, per una scheda all'IRQ 5 e memoria condivisa a `0xd0000` che usa un driver modulare, si aggiuga `options es3210 irq=5 mem=0xd0000` a `/etc/conf.modules`. Oppure con il driver compilato nel kernel, all'avvio si passi `ether=5,0,0xd0000,eth0`. L'I/O base è rilevato automaticamente e quindi dovrebbe essere usato il valore 0.

#### 5.30.2 NI5010

Stato: Semi supportata, Nome del driver: ni5010

Solitamente ci si doveva procurare separatamente il driver per queste vecchie schede a 8 bit della MiCom-Interlan, ma ora è distribuito con i kernel v2.2 come driver sperimentale.

#### 5.30.3 NI5210

Stato: Semi supportata, Nome del driver: ni52

Anche questa scheda usa uno dei chip dell'Intel. Michael Hipp ha scritto un driver e ora è incluso nel kernel standard come driver 'alpha'. Michael vorrebbe sentire dei commenti dagli utenti che posseggono questa scheda. Si veda 3.1 (Driver sperimentali) per conoscere importanti informazioni sull'uso dei driver Ethernet sperimentali.

#### 5.30.4 NI6510 (non EB)

Stato: Semi supportata, Nome del driver: ni65

C'è anche un driver per la NI6510 basata su LANCE ed è sempre stato scritto da Michael Hipp. Anche questo è un driver sperimentale. Per qualche ragione questa scheda non è compatibile con il driver LANCE generico. Si veda 3.1 (Driver sperimentali) per conoscere importanti informazioni sull'uso dei driver Ethernet sperimentali.

#### 5.30.5 EtherBlaster (aka NI6510EB)

Stato: Supportata, Nome del driver: lance

Dal kernel 1.2.23, al driver LANCE generico è stato aggiunto un controllo per la firma `0x52, 0x44` specifica della NI6510EB. Comunque alcuni hanno riportato che questa firma non è la stessa per tutte le schede NI6510EB, il che fa sì che il driver LANCE non rilevi la scheda. Se questo succede, si può cambiare il controllo (verso la riga 322 in lance.c) a printk() cosicché stampi i valori per la propria scheda e poi usarli come default invece di `0x52, 0x44`.

Usando il driver lance, probabilmente le schede dovrebbero essere usate in modalità ad 'alte prestazioni' e non in compatibilità NI6510.

### 5.31 RealTek

#### 5.31.1 Adattatore pocket RealTek RTL8002/8012 (AT-Lan-Tec)

Stato: Supportato, Nome del driver: atp

Questo è un adattatore pocket generico e a basso costo, venduto dalla AT-Lan-Tec e (probabilmente) da molti altri fornitori. Nel kernel standard è presente un driver. Si noti che le informazioni più importanti sono contenute nel file sorgente del driver 'atp.c'.

Si noti che nelle prime versioni di questo driver il nome di device da passare a `ifconfig` *non* era `eth0` bensì `atp0`.

#### 5.31.2 RealTek 8009

Stato: Supportata, Nome del driver: ne (+8390)

Questa è un clone ISA della NE2000 e si dice funzioni bene con il driver NE2000 di Linux. Dal sito WWW della RealTek (`http://www.realtek.com.tw`) o via FTP dallo stesso sito può essere scaricato il programma `rset8009.exe`.

#### 5.31.3 RealTek 8019

Stato: Supportata, Nome del driver: ne (+8390)

Questa è la versione Plug and Pray della scheda precedente. Si usi il software per DOS per disabilitare il Pnp ed abilitare la configurazione senza ponticelli; si configuri la scheda ad un indirizzo I/O ed a un'IRQ ragionevoli e si dovrebbe essere pronti per partire (se si usa il driver come modulo, non si dimentichi di aggiungere l'opzione `io=0xNNN` a `/etc/conf.modules`). Dal sito WWW della RealTek (`http://www.realtek.com.tw`) o via FTP dallo stesso sito può essere scaricato il programma `rset8009.exe`.

#### 5.31.4 RealTek 8029

Stato: Supportata, Nome del driver: ne, ne2k-pci (+8390)

Questa è una impletazione PCI a singolo chip di un clone NE2000. Diversi vendor adesso vendono schede con questo chip. Si veda la sezione 5.27.2 (NE2000-PCI) per informazioni sull'uso di una qualsiasi di queste schede. Si noti che questo è un progetto di più di 10 anni fa semplicemente adattato al bus PCI. Le prestazioni non saranno molto migliori rispetto a quelle dell'eqivalente modello ISA.

#### 5.31.5 RealTek 8129/8139

Stato: Semi-supportate, Nome del driver: rtl8139

Una soluzione Ethernet PCI a chip singolo della RealTeak. Un driver per le schede basate su questo chip è stato incluso nella release v2.0.34 di Linux. Al momento nei kernel v2.2 se si vuole avere accesso a questo driver si deve rispondere 'Y' quando viene chiesto se si vogliono i driver sperimentali. Per maggiori informazioni si veda:

`http://cesdis.gsfc.nasa.gov/linux/drivers/rtl8139.html`

### 5.32 Sager

#### 5.32.1 Sager NP943

Stato: Semi supportata, Nome del driver: 3c501

Questa è semplicemente un clone della 3c501, con un diverso prefisso S.A. nella PROM. Suppongo che sia talmente demente quanto la 3c501 originale. Il driver cerca l'ID della NP943 e poi la tratta semplimente come un 3c501. Si veda la sezione 5.1.1 (3Com 3c501) per tutte le ragioni per le quali veramente non si deve usare una di queste schede.

## 5.33 Schneider & Koch

### 5.33.1 SK G16

Stato: Supportata, Nome del driver: sk_g16

Questo driver è stato incluso nei kernel v1.1 ed è stato scritto da PJD Weichmann e SWS Bern. Sembra che la SK G16 sia simile alla NI6510, per il fatto che è basata sulla prima edizione del chip LANCE. Ancora una volta, sembra che pure questa scheda non funzioni con il driver LANCE generico.

## 5.34 SEEQ

### 5.34.1 SEEQ 8005

Stato: Supportata, Nome del driver: seeq8005

Questo driver è stato incluso nei primi kernel v1.3 ed è stato scritto da Hamish Coleman. Nel driver sono incluse pochissime informazioni sulla scheda e quindi ci sono pure poche informazioni da mettere qui. Se si hanno domande, probabilmente la cosa migliore è di scrivere a hamish@zot.apana.org.au.

## 5.35 SMC (Standard Microsystems Corp.)

La divisione Ethernet della Western Digital è stata acquisita dalla SMC molti anni fa, quando la wd8003 e wd8013 erano i prodotti di punta. Da allora la SMC ha continuato a fare schede ISA basate sull'8390 (Elite16, Ultra, EtherEZ) e ha aggiunto alla gamma anche diversi prodotti PCI.

Informazioni per contattare la SMC:

SMC / Standard Microsystems Corp., 80 Arkay Drive, Hauppage, New York, 11788, USA. Supporto tecnico telefonico: 800-992-4762 (USA) o 800-433-5345 (Canada) o 516-435-6250 (Altri nazioni). Richieste di documentazione: 800-SMC-4-YOU (USA) o 800-833-4-SMC (Canada) o 516-435-6255 (Altre nazioni). Supporto tecnico via E-mail: `techsupt@ccmail.west.smc.com`. Sito FTP: `ftp.smc.com`. Sito WWW: *SMC* `<http://www.smc.com>`.

### 5.35.1 WD8003, SMC Elite

Stato: Supportate, Nome del driver: wd (+8390)

Queste sono le versioni ad 8 bit della scheda. Il chip 8003 a 8 bit è leggermente meno costoso, ma vale solo se destinato ad un uso leggero. Si noti che alcune schede senza EEPROM (cloni con i ponticelli o schede we8003 veramente vecchie) non hanno modo di riportare la linea IRQ utilizzata. In questo caso, è usato auto-irq e se fallisce, il driver assegna l'IRQ 5 senza dire niente. Dal sito FTP della SMC si possono scaricare i dischetti di configurazione/driver. Si noti che alcune delle versioni più recenti del rpogramma 'SuperDisk' della SMC falliranno nel rilevare le veramente vecchie schede senza EEPROM. Il file `SMCDSK46.EXE` sembra essere una buona scelta a tutto tondo. Inoltre le impostazioni dei ponticelli per tutte le loro schede si possono trovare in un file testo ASCII nel summenzionato archivo. L'ultima (migliore?) versione può essere ottenuta da `ftp.smc.com`.

Poiché questo sono in pratica analoghe alle loro controparti a 16 bit (WD8013 / SMC Elite16), si dovrebbe vedere la sezione successiva per maggiori informazioni.

### 5.35.2 WD8013, SMC Elite16

Stato: Supportate, Nome del driver: wd (+8390)

Negli anni sono stati aggiunti al progetto altri registri e una EEPROM (le prime schede wd8003 sono apparse circa 10 anni fa!). I cloni solitamente vanno sotto il nome '8013' e tipicamente usano un progetto senza EEPROM (con i ponticelli). Gli ultimi modelli delle schede SMC montano un chip 83c690 della SMC invece dell'originale DP8390 della National presente nelle prime schede. Il progetto a memoria condivisa rende queste schede un po' più veloci delle schede PIO, specialmente con pacchetti più grossi. Più importante, dal punto del vista del driver, si evitano così un po' di bachi nella modalità ad I/O programmato dell'8390, permettendo accessi multi-thread sicuri al buffer dei pacchetti e non si ha il registro dati dell'I/O programmato che pianta la macchina durante il controllo al boot.

Le schede non EEPROM che non possono leggere l'IRQ selezionato proveranno a fare l'auto-irq e se falliscono assegneranno silenziosamente l'IRQ 10 (le versioni a 8 bit assegnano l'IRQ 5).

Le schede con a bordo dimensioni di memoria non standard la possono specificare all'avvio (o come opzione in `/etc/conf.modules` se si usano i moduli). La dimensione standard della memoria è di 8kB per le schede a 8 bit e 16kB per quelle a 16 bit. Per esempio, le schede più vecchie WD8003EBT potevano essere impostate attraverso i ponticelli per 32kB di memoria. Per usare appieno questa RAM, si dovrebbe usare qualcosa di simile a (con I/O=0x280 e IRQ 9):

```
LILO: linux ether=9,0x280,0xd0000,0xd8000,eth0
```

Si veda anche 3.5 (Problemi dell'8013) per alcuni dei problemi più comuni e le risposte alle domande più frequenti.

Se si intende usare questo driver come modulo caricabile probabilmente si dovrebbe vedere la sezione 10.2 (Usare i driver Ethernet come modulo) per informazioni specifiche sui moduli.

### 5.35.3 SMC Elite Ultra

Stato: Supportata, Nome del driver: smc-ultra (+8390)

Questa scheda Ethernet è basata sul chip 83c790 della SMC che ha un po' di nuove caratteristiche rispetto al 83c690. Sebbene possieda una modalità simile alle schede SMC più vecchie, non è completamente compatibile con i vecchi driver WD80*3- Comunque in questa modalità condivide la maggior parte del codice con gli altri driver 8390 anche se funziona leggermente più veloce rispetto ad un clone WD8013.

Poiché parte della Ultra *sembra* una 8013, il controllo per la Ultra si suppone trovi una Ultra prima che il controllo per la wd8013 abbia la possibilità di indentificarla in maniera errata.

Donald ha riferito che è possibile scrivere un driver separato per la modalità 'Altego' della Ultra che permette di concatenare la trasmissione al costo di un uso inefficiente dei buffer di ricezione, ma probabilmente ciò non avverrà.

Gli utilizzatori di adattatori host SCSI bus master prendano nota: Nel manuale distribuito con Interactive UNIX, è riportato che un bug nella SMC Ultra causerà corruzione dei dati con dischi SCSI utilizzati attraverso un adattatore host aha-154X. Questa cosa affligge probabilmente anche le schede aha-154X compatibili, come le schede BusLogic e gli adattatori host AMI-Fastdisk SCSI.

La SMC ha riconosciuto che il problema si presenta con Interactive e con vecchie versioni dei driver per Windows NT. È un conflitto hardware con le prime versioni della scheda che può essere aggirato nel progetto del driver. Il driver Ultra attuale protegge da questo abilitando la memoria condivisa solo durante il trasferimento con la scheda. Ci si assicuri che la propria versione del kernel sia almeno 1.1.84 e che la versione riportata dal driver al boot sia almeno `smc-ultra.c:v1.12`, altrimenti si è vulnerabili.

Se si intende usare questo driver come modulo caricabile probabilmente si dovrebbe vedere la sezione 10.2 (Usare i driver Ethernet come modulo) per informazioni specifiche sui moduli.

### 5.35.4 SMC Elite Ultra32 EISA

Stato: Supportata, Nome del driver: smc-ultra32 (+8390)

Questa scheda EISA ha molto in comune con la sua controparte ISA. Un driver funzionante (e stabile) è incluso sia nei kernel v2.0 che v.2.2. Grazie a Leonard Zubkoff per aver acquistato alcune di queste schede dimodoché sia stato possibile aggiungerne il supporto per Linux.

### 5.35.5 SMC EtherEZ (8416)

Stato: Supportata, Nome del driver: smc-ultra (+8390)

Questa scheda usa il chip 83c795 della SMC e supporta le specifiche Plug 'n Play. Ha anche una modalità *SMC Ultra* compatibile, che le permette di essere usata con il driver Ultra per Linux. Per migliori risultati si usi il programma fornito dalla SMC (disponibile nel loro sito www/ftp) per disabilitare il PnP e configurarla per la modalità a memoria condivisa. Si vedano le informazioni precedenti per alcune note sul driver Ultra.

Per i kernel v1.2, la scheda ha dovuto essere configurata per l'uso a memoria condivisa. Comunque i kernel v2.0 possono usare la scheda in modalità a memoria condivisa o I/O programmato. La modalità a memoria condivisa è leggermente più veloce e usa anche meno risorse di CPU.

### 5.35.6 SMC EtherPower PCI (8432)

Stato: Supportata, Nome del driver: de4x5, tulip

NB: La EtherPower II è una scheda completamente diversa. Si veda sotto! Queste schede sono un'implementazione base del 21040 della DEC, cioè un unico grosso chip e una coppia di transceiver. Donald ha usato una di queste schede per lo sviluppo del driver 21040 generico (meglio noto come `tulip.c`). Grazie a Duke Kamstra, ancora una volta, per aver fornito una scheda sulla quale fare lo sviluppo.

Alcune delle ultime revisioni di questa scheda usano il più recente chip 21041 della DEC, che può causare problemi con versioni più vecchie del driver tulip. Se si hanno problemi ci si assicuri di usare l'ultima versione del driver, che potrebbe non essere ancora stata inclusa nell'attuale albero dei sorgenti del kernel.

Si veda 5.16.4 (DEC 21040) per ulteriori dettagli sull'uso di una di queste schede e sullo stato attuale del driver.

Apparentemente, l'ultima revisione delle scheda, la EtherPower-II usa il chip 9432. Al momento non è chiaro se questa funzionerà con il driver attuale. Come sempre, se non si è sicuri, si verifichi di poter restituire la scheda se non funziona con il driver per Linux *prima* di pagarla.

### 5.35.7 SMC EtherPower II PCI (9432)

Stato: Semi supportata, Nome del driver: epic100

Queste schede, basate sul chip 83c170 della SMC, sono completamente differenti da quelle basate sul TULIP. Un nuovo driver è stato incluso nei kernel v2.0 e v2.2 per supportare queste schede. Per ulteriori dettagli si veda:

`http://cesdis.gsfc.nasa.gov/linux/drivers/epic100.html`

#### 5.35.8 SMC 3008

Stato: Non supportata.

Queste schede a 8 bit sono basate sul Fujitsu MB86950, che è un'antica versione del MB86965 usato dal driver at1700 per Linux. Russ dice che probabilmente si può metter su un driver guardando il codice at1700.c e il suo packet driver DOS per la scheda Tiara (tiara.asm). Non sono molto comuni.

#### 5.35.9 SMC 3016

Stato: Non Supportata.

Queste sono scheda 8390 a 16 bit ad I/O mappato, molto simili ad una generica scheda NE2000. Se si riescono ad ottenere le specifiche dalla SMC, allora il porting del driver NE2000 probabilmente sarà abbastanza facile. Non sono molto comuni.

#### 5.35.10 SMC-9000 / SMC 91c92/4

Stato: Supportata, Nome del driver: smc9194

La SMC9000 è una scheda VLB basata sul chip 91c92. Il 91c92 sembra essere presente anche su un po' di schede di altre marche, ma è abbastanza poco comune. Erik Stahlman (erik@vt.edu) ha scritto questo driver presente nei kernel v2.0, ma non nei più vecchi kernel v1.2. Si dovrebbe essere in grado di mettere il driver nell'albero dei sorgenti v1.2 con poche difficoltà.

#### 5.35.11 SMC 91c100

Stato: Semi supportata, Nome del driver: smc9194

Si pensa che il driver SMC 91c92 funzioni per le schede basate su questo chip 100Base-T, ma al momento la cosa non è stata verificata.

### 5.36 Texas Instruments

#### 5.36.1 ThunderLAN

Stato: Supportata, Nome del driver: tlan

Questo driver gestisce i molti dispositivi Ethernet built-in della Compaq, compresi i gruppi NetFlex e Netelligent. Supporta anche i prodotti Olicom 2183, 2185, 2325 e 2326.

### 5.37 Thomas Conrad

#### 5.37.1 Thomas Conrad TC-5048

Questa è un'altra scheda PCI basata sul chip 21040 della DEC.

Si veda la sezione sul chip 21040 (5.16.4 (DEC 21040)) per maggiori informazioni.

### 5.38 VIA

Probabilmente non si vedrà mai una scheda di rete VIA, in quanto la VIA costruisce diversi chip usati da altri per costruire le loro schede Ethernet. Hanno un sito WWW a:

http://www.via.com.tw/

### 5.38.1 VIA 86C926 Amazon

Stato: Supportato, Nome del driver: ne, ne2k-pci (+8390)

Questo chip è quanto offre la VIA come PCI-NE2000. Si può scegliere tra il driver `ne.c` per ISA e PCI o il driver solo per PCI `ne2k-pci.c`. Si veda la sezione sul PCI-NE2000 per ulteriori informazioni.

### 5.38.2 VIA 86C100A Rhine II (and 3043 Rhine I)

Stato: supportato, Nome del driver: via-rhine

Questo driver relativamente nuovo può essere trovato negli attuali kernel 2.0 e 2.1. È un miglioramento rispetto al chip NE2000 86C926 nel fatto che supporta i trasferimenti bus master, ma gli stretti requisiti sull'allineamento al bit 32 del buffer limitano i benefici guadagnati da ciò. Per maggiori dettagli e driver aggiornati si veda:

http://cesdis.gsfc.nasa.gov/linux/drivers/via-rhine.html

## 5.39 Western Digital

Si veda la sezione 5.35 (SMC) per informazioni sulle schede SMC (la SMC ha comprato la divisione delle schede di rete della Western Digital molti anni fa).

## 5.40 Winbond

La Winbond in realtà non costruisce e vende schede complete al pubblico, piuttosto costruisce soluzioni Ethernet su un unico chip che altre compagnie comprano e mettono nelle loro schede PCI con il loro nome che poi vendono nei negozi.

### 5.40.1 Winbond 89c840

Stato: Semi supportato, Nome del driver: winbond-840

Questo driver attualmente non è distribuito con il kernel, poiché è in fase di test. È disponibile a:

http://cesdis.gsfc.nasa.gov/linux/drivers/test/winbond-840.c

### 5.40.2 Winbond 89c940

Stato: Supportato, Nome del driver: ne, ne2k-pci (+8390)

Questo chip è uno dei due più comunemente presenti sulle schede ne2000 PCI a basso costo vendute da un sacco di costruttori. Si noti che questo è sempre un progetto vecchio di dieci anni adattato ad un bus PCI. Le prestazioni non saranno tanto migliori rispetto a quelle dell'equivalente modello ISA.

## 5.41 Xircom

Per lungo tempo, la Xircom non ha voluto rilasciare la informazioni di programmazione necessarie per scrivere un driver, a meno che non si firmasse per averle. Apparentemente abbastanza utenti Linux li hanno subbissati di richieste di supporto per il driver (affermano di supportate tutti i più popolari sistemi operativi

di rete...) e quindi hanno cambiato la loro politica e hanno permesso il rilascio di documentazione senza dover firmare un accordo di non rivelazione. Ad alcuni hanno detto che avrebbero rilasciato il codice sorgente del driver SCO, mentre ad altri hanno detto che non forniscono più informazioni su prodotti 'obsoleti' come i primi modelli PE. Se si è interessati e si vogliono verificare da sé queste cose, si può contattare la Xircom al 1-800-874-7875, 1-800-438-4526 o +1-818-878-7600.

### 5.41.1 Xircom PE1, PE2, PE3-10B*

Stato: Non supportate.

Non per dare tante speranze, ma se si ha uno di questi adattatori per porta parallela, si può riuscire ad usarlo nell'emulatore DOS con i driver per DOS forniti dalla Xircom. Si deve permettere a DOSEMU di accedere alla propria porta parallela e probabilmente si dovrà giocare un po' con SIG (Silly Interrupt Generator di DOSEMU).

### 5.41.2 Schede PCMCIA Xircom

Stato: Semi-supportate, Nome del driver: ????

Per alcune delle schede PCMCIA della Xircom esistono dei driver disponibili nel pacchetto PCMCIA di David Hinds. Si veda in questo per informazioni più aggiornate.

## 5.42 Zenith

### 5.42.1 Z-Note

Stato: Supportata, Nome del driver: znet

L'adattatore di rete built-in Z-Note è basato su un i82593 dell'Intel ed usa *due* canali DMA. Esiste un driver (sperimentale?) nell'attuale versione del kernel. Come tutti gli adattatori pocket e per notebook, quando si esegue `make config` lo si trova nella sezione 'Pocket and portable adaptors'. Si noti che il ThinkPad 300 dell'IBM è compatibile con Z-Note.

## 5.43 Znyx

### 5.43.1 Znyx ZX342 (DEC 21040 based)

Stato: Supportata, Nome del driver: de4x5, tulip

Si ha la scelta fra *due* driver per le schede basate su questo chip. C'è un driver DE425 scritto da David e il driver generico per 21040 che ha scritto Donanld.

Si noti che dal 1.1.91, David ha aggiunto una opzione di compilazione che può permettere alle schede non DEC (come quella della Znyx) di funzionare con il suo driver. Si veda il file `README.de4x5` per i dettagli.

Si veda la sezione 5.16.4 (DEC 21040) per maggiori informazioni su queste schede e la situazione attuale del driver.

## 5.44 Identificare una scheda sconosciuta

Bene, e così l'amico del vicino del cugino di vostro zio ha un fratello che ha trovato una vecchia scheda Ethernet ISA in un case AT che usava come gabbia per criceti di suo figlio. In qualche modo questa scheda

vi è capitata tra le mani e la volete usare con Linux, ma nessuno ha idea di che scheda sia e non c'è alcuna documentazione.

Per prima cosa, si cerchi un qualsiasi numero del modello che potrebbe essere un indizio. Qualsisi numero di modello che contiene 2000 potrebbe essere un clone della NE2000. Qualsiasi scheda con 8003 o 8013 da qualche parte sarà una scheda Western/Digital WD80x3 o una SMC Elite o un clone di una di esse.

### 5.44.1 Identificare il Network Interface Controller

Si cerchi il chip più grosso nella scheda. Sarà il network controller (NIC) e la maggior parte può essere identificata dal numero. Se si sa quale NIC c'è sulla scheda, quanto segue può aiutare ad scoprire che scheda è.

Probabilmente il NIC ancora più comune è il DP8390 della National Semiconductor, noto anche come NS32490, DP83901, DP83902, DP83905 e/o DP83907. E questi sono solo quelli fatti dalla National! Altre compagnie, come la Winbond e la UMC, costruiscono cloni del DP8390 e del DP83905, come il Winbond 89c904 (clone del DP83905) e l'UMC 9090. Se la scheda ha su una qualche forma di 8390, allora è possibile che sia un clone della ne1000 o della ne2000. Le seconde schede più comuni basate su 8390 sono le schede wd80x3 e cloni. Le schede con un DP83905 possono essere configurate come una ne2000 *o* come una wd8013. Le versioni pià nuove delle schede wd80x3 genuine e delle SMC Elite hanno un 83c690 al posto del DP8390 originale. Le schede SMC Ultra hanno un 83c790 ed usano un driver leggermente diverso dalle schede wd80x3. Le schede SMC EtherEZ hanno un 83c795 ed usano lo stesso driver della SMC Ulta. Tutte le schede con BNC basate su una qualche versione di 8390 o di un suo clone solitamente avranno un chip DIP a 16 pin 8392 (o un 83c692, o un ???392) molto vicino al connettore BNC.

Un altro NIC comune presente sulle schede più vecchie è l'Intel i82586. Le schede che hanno questo NIC includono le 3c505, 3c507, 3c523, Intel EtherExpress-ISA, Microdyne Exos-205T e la Racal-Interlan NI5210.

Il NIC LANCE originale dell'AMD era numerato AM7990 e le revisioni più nuove includono i 79c960, 79c961, 79c965, 79c970 e 79c974. La maggior parte delle schede con uno di questi funzionerà con il driver LANCE di Linux, ad eccezione delle vecchie schede Racal-Interlan NI6510 che hanno il loro driver apposito.

Le schede PCI più nuove che hanno un DEC 21040, 21041, 21140 o un numero simile sul NIC dovrebbero essere in grado di usare il driver tulip o il de4x5.

Altre schede PCI che hanno un grosso chip marchiato RTL8029 o 89C940 o 86C926 sono cloni ne2000 e il driver nelle versioni di Linux 2.0 e superiori dovrebbe automaticamente rilevarle al boot.

### 5.44.2 Identificare l'indirizzo Ethernet

Ogni scheda Ethernet ha il suo indirizzo personale ed unico a sei byte. I primi tre byte di quell'indirizzo sono gli stessi per ogni scheda fatta da un particolare costruttore. Per esempio tutte le schede SMC partono con `00:00:c0`. Gli ultimi tre sono assegnati dal costruttore univocamente ad ogni scheda che produce.

Se la propria scheda ha un etichetta che dà tutti i 6 byte del suo indirizzo si può risalire al costruttore dai primi tre. Comunque è più comune vedere solo gli ultimi tre byte stampati su un'etichetta attaccata alla PROM che non dirà niente.

Si può determinare a quale rivenditore è stato assegnato un determinato indirizzo dall'RFC 1340. Apparentemente in giro ci sono anche elenchi più aggiornati. Si provi a fare una ricerca WWW o FTP di `EtherNet-codes` o `Ethernet-codes` e qualcosa si troverà.

### 5.44.3 Suggerimenti per provare ad usare una scheda sconosciuta

Se non si è ancora sicuri di che scheda sia ma si è un po' ristretta la cerchia delle possibilità, allora si può compilare un kernel con una serie di driver al suo interno e vedere se uno di essi rileva automaticamente la scheda al boot.

Se il kernel non rileva la scheda, potrebbe essere che la scheda non è configurata ad uno degli indirizzi che i driver controllano quando cercano una scheda. Se questo è il caso, si può provare a scaricare `scanport.tar.gz` da un sito ftp di Linux e vedere se può scoprire a che indirizzo è configurata. Scansiona lo spazio I/O ISA da `0x100` a `0x3ff` cercando dispositivi che non sono registrati in `/proc/ioports`. Se trova un dispositivo sconosciuto ad un qualche indirizzo particolare, si può esplicitamente indirizzare il controllo a quell'indirizzo con un comando `ether=` al boot.

Se si è riusciti a far rilevare la scheda, allora solitamente si può scoprire cosa fanno i ponticelli combiandone uno alla volta e vedendo a quale I/O base e IRQ la scheda viene poi rilevata. Le impostazioni dell'IRQ possono essere determinate anche seguendo le tracce sul retro della scheda da dove sono saldati i ponticelli. Contando i contatti dorati nel retro della scheda partendo dalla parte della scheda con la linghetta di metallo si troveranno gli IRQ 9, 7, 6, 5, 4, 3, 10, 11, 12, 15, 14 rispettivamente nei contatti 4, 21, 22, 23, 24, 25, 34, 35, 36, 37, 38. Le schede a 8 bit hanno solo fino al contatto 31.

I ponticelli che sembra non facciano niente solitamente servono per selezionare l'indirizzo di memoria della ROM di boot opzionale. Altri ponticelli posizionati nei pressi dei connettorei BNC o RJ-45 o AUI solitamente servono per selezionare il mezzo d'uscita. Tipicamente questi sono anche vicino allo 'scatolotto nero' del convertitore di tensione marchiato YCL, Valor o Fil-Mag.

Un bella collezione di impostazioni di ponticelli per diverse schede può essere trovata a:

*Ethercard Settings* `<http://www.slug.org.au/NIC/>`

## 5.45 Driver per i dispositivi non Ethernet

Ci sono alcuni altri driver nei sorgenti di Linux che si presentano ai programmi di rete come dispositivi *simil-Ethernet*, anche se non sono veramente Ethernet. Per completezza sono qui brevemente elencati.

`dummy.c` – Lo scopo di questo driver è di fornire un dispositivo al quale far puntare un instradamento, ma che non trasmette realmente pacchetti.

`eql.c` – Load Equalizer (Equalizzatore di carico), schiavizza più dispositivi (solitamente modem) e bilancia il carico di trasmissione tra essi presentandoli come un unico dispositivo ai programmi di rete.

`ibmtr.c` – IBM Token Ring, che non è veramente Ethernet. Broken-Ring necessita dell'instramento source e di altre brutte cose.

`loopback.c` – Dispositivo loopback al quale vanno tutti i pacchetti destinati alla macchina e partenti da questa. Essenzialmente sposta semplicemente il pacchetto dalla coda TX nella coda RX.

`pi2.c` – Interfacce PI e PI2 dell'Ottawa Amateur Radio Club.

`plip.c` - Parallel Line Internet Protocol, permette a due computer di inviarsi pacchetti attraverso le porte parallele connesse in maniera point-to-point (punto a punto).

`ppp.c` – Point-to-Point Protocol (RFC1331), per la trasmissione dei datagrammi multiprotocollo su una connessione Point-to-Point Link (solitamente modem).

`slip.c` – Serial Line Internet Protocol, permette a due computer di inviarsi pacchetti attraverso le porte seriali (solitamente via modem) connesse in maniera point-to-point (punto a punto).

`tunnel.c` – Fornisce un tunnel IP attraverso il quale si può incanalare il traffico di rete in maniera trasparente tra le sottoreti.

`wavelan.c` – Un transceiver radio tipo Ethernet controllato da un coprocessore Intel 82586 usato su altre schede Ethernet come la Intel EtherExpress.

# 6 Cavi, coassiali, doppini intrecciati

Se si sta mettendo su una nuova rete, si dovrà decidere se usare thin Ethernet (cavo coassiale RG58 con connettori BNC) o 10baseT (cavi tipo telco a doppini intrecciati con connettori 'telefonici' RJ-45 a otto vie). La vecchia thick Ethernet, con cavi RG-5 a N connettori, è obsoleta e si vede poco in giro.

Si veda 2.6 (Tipi di cavi...) per una panoramica introduttiva ai cavi. Si noti inoltre che la FAQ di *comp.dcom.lans.ethernet* contiene un sacco di informazioni utili sui cavi e robe simili. Si faccia FTP a rtfm.mit.edu e si cerchi nella directory `/pub/usenet-by-hierarchy/` la FAQ di quel newsgroup.

## 6.1 Thin Ethernet (thinnet)

Il cavo thin Ethernet è abbastanza economico. Se si vuole farsi da soli i cavi, l'RG58A a solid-core costa \$0.27/m mentre l'RG58AU stranded è a \$0.45/m. I connettori BNC twist-on costano meno di due dollari l'uno e altri pezzi vari sono attrettanto economici. È essenziale terminare correttamente ciascun capo del cavo con terminatori da 50 Ohm, che costano due dollari alla coppia. È altrettanto vitale che il cavo non sia formato da più spezzoni: il connettore a 'T' dev'essere attaccato direttamente alle schede Ethernet.

Ci sono due svantaggi principali ad usare la thinnet. Il primo è che è limitata a 10Mb/sec: i 100Mb/sec richiedono il doppino intrecciato. Il secondo svantaggio che se si ha un grande anello di macchine connesse assieme e qualche testone interrompe l'anello staccando un cavo da un connettore a T, l'intera rete va giù perché vede un'impedenza infinita (circuito aperto) invece dei 50 Ohm richiesti come terminazione. Si noti che si può rimuovere la T stessa dalla scheda senza uccidere l'intera sottorete, purché non si stacchino i cavi dalla T. Naturalmente questa cosa disturberà la macchina dalla quale si è tolto il connettore a T 8-) Se se si sta facendo una piccola rete di due macchine, sono *ancora* necessari sia il connettore a T che i terminatori da 50 Ohm: *non si può* semplicemente connettere le due macchine assieme!

Ci sono anche dei simpatici sistemi di cavi che apparentemente *sembrano* un unico cavo, ma in realtà il cavo fa un giro da parte a parte dentro il rivestimento esterno, dando al cavo quella tipica sezione ovale. Nel punto dove torna indietro l'anello è messo un connettore BNC che si può connettere alla propria scheda. Quindi si ha l'equivalente di due cavi e un BNC a T, ma in questo caso è impossibile per l'utilizzatore rimuovere un cavo da un lato della T e disturbare la rete.

## 6.2 Doppino intrecciato (twisted pair)

Le reti a doppini intrecciati necessitano di hub (concentratori) attivi, che costano attorno ai \$50 e il costo reale del semplice cavo può essere più alto di quello necessario per la thinnet. Si può tranquillamente ignorare quelli che dicono che si può usare il cablaggio telefonico già esistente visto che dove c'è è una rara installazione.

D'altra parte, tutte le specifiche Ethernet a 100Mb/sec usano doppini intrecciati e sono usati pure nella maggior parte delle nuove installazioni. Inoltre, Russ Nelson aggiunge che «Le nuove installazioni dovrebbero usare cavi di categoria 5. Qualsiasi altra cosa spreca il proprio tempo, in quanto un 100Base-qualsiasi richiederà cavi categoria 5.»

Se si sta solamente connettendo macchine, è possibile evitare l'uso di un hub, scambiando le coppie Rx e Tx (1-2 e 3-6).

Se si guarda il connettore RJ-45 (come se lo si stesse connettendo nella propria bocca) con il gancetto di bloccaggio in alto, allora i pin sono numerati da 1 a 8 da sinistra verso destra. L'uso dei pin è il seguente:

| Numero Pin | Assegnamento |
|---|---|
| 1 | Output Data (+) |
| 2 | Output Data (-) |
| 3 | Input Data (+) |
| 4 | Riservato per l'uso telefonico |
| 5 | Riservato per l'uso telefonico |
| 6 | Input Data (-) |
| 7 | Riservato per l'uso telefonico |
| 8 | Riservato per l'uso telefonico |

Se si vuole fare un cavo, quanto segue vi tornerà utile. Le coppie di segnali differenziali devono essere nella stessa coppia intrecciata per ottenere la minima impendenza/perdita richiesta di un cavo UTP. Se si guarda la tabella precedente, si vedrà che 1+2 e 3+6 sono i due insiemi di coppie di segnali differenziali. Non 1+3 e 2+6!!!!!! A 10MHz e basse distanze, si può ancora far qualcosa con questi errori. Ma non ci si pensi nemmeno lontanamente a 100Mhz.

Per un normale cavo con terminazioni 'A' e 'B', si deve usare la mappatura diretta pin a pin, con l'ingresso e l'uscita che usano ciascuna una coppia di doppini intrecciati (per esigenze di impedenza). Questo significa che 1A va a 1B, 2A a 2B, 3A a 3B e 6A a 6B. I cavetti che collegano 1A-1B e 2A-2B devono essere un doppino intrecciato. Ed anche i cavetti che collegano 3A-3B e 6A-6B devono essere un'altro doppino intrecciato.

Ora se non si ha un hub e si vuole fare un 'null cable', quel che si deve fare è rendere l'ingresso di 'A' l'uscita di 'B' e l'uscita di 'A' l'ingresso di 'B', senza cambiare la polarità. Questo significa connettere 1A a 3B (out+ di A a in+ di B) e 2A a 6B (out- A a in- B). Queste due connessioni devono essere un doppino intrecciato. Trasportano quel che la scheda/spina 'A' considera l'uscita e quel che è visto come ingresso dalla scheda/spina 'B'. Poi si connetta 3A a 1B (in+ A a out+ B) e si connetta anche 6A a 2B (in- A a out- B). Anche queste due connessioni devono essere un doppino intrecciato. Trasportano quel che la scheda/spina 'A' considera ingresso e quel che la scheda/spina 'B' considera uscita.

Quindi se si considera un normale cavo per avere un cavo 'null', si tagli via uno dei capi, si scambino di posto i doppini intrecciati di Rx e Tx nella nuova spina e la si chiuda. Niente di complicato. Si deve semplicemente portare il segnale Tx di una scheda nel Rx dell'altra e viceversa.

Si noti che prima che il 10BaseT fosse ratificato come standard, c'erano altri formati di rete che usavano connettori RJ-45 e lo stesso schema di connessione appena visto. Esempi sono la LattiNet della SynOptics e la StarLAN dell'AT&T. In alcuni casi (come con le prime schede 3c503) si potevano impostare dei ponticelli per far sì che la scheda dialogasse con hub di diverso tipo, ma la maggior parte delle schede progettate per questi vecchi tipi di reti non funzioneranno con le reti/hub 10BaseT (si noti che se la scheda ha anche una porta AUI, allora non c'è alcuna ragione per non usarla assieme con un transceiver da AUI a 10BaseT).

## 6.3 Thick Ethernet

La thick Ethernet è praticamente obsoleta e solitamente è usata solo per mantenere la compatibilità con un'installazione preesistente. Si può fare uno strappo alle regole e connettere thick e thin Ethernet assieme con un connettore passivo N-to-BNC da 3 dollari, spesso la miglior soluzione per espandere una thicknet già esistente. In questo caso una soluzione corretta (ma più costosa) sarebbe di usare un ripetitore (repeater).

# 7 Configurazione del software e diagnotici

In molti casi, se la configurazione viene fatta via software e salvata poi in una EEPROM, si dovrà avviare DOS e usare il programma per DOS fornito dal rivenditore per impostare IRQ, I/O, indirizzo di memoria e altre robette della scheda. D'altra parte è auspicabile che questa sia una cosa che si dovrà fare solo una volta. Se non si ha il software per DOS della propria scheda, si provi a guardare nel sito WWW del produttore. Se non si conosce il nome del sito si provi ad indovinarlo, per esempio 'www.produttore.com' dove 'produttore' è il nome del produttore della propria scheda. Questa cosa funziona per la SMC, la 3Com e molti *molti* altri produttori.

Per alcune schede esistono le versioni Linux delle utilità di configurazione e sono qui elencate. Donald ha scritto alcuni piccoli programmi per Linux di diagnostica per le schede. La maggior parte di questi sono il prodotto finale di strumenti di debug che ha creato durante la scrittura dei diversi driver. Non ci si aspettino interfacce a menu carine. Per poterli usare nella maggioranza dei casi sarà necessario leggere il codice sorgente. Anche se per la propria particolare scheda non esiste un diagnostico, si possono ottenere comunque alcune informazioni digitando semplicemente `cat /proc/net/dev` (assumendo che all'avvio la propria scheda sia stata almeno rilevata).

In tutti i casi, si dovrà usare la maggior parte di questi programmi come root (per permettere l'I/O nelle porte) e prima di farlo è consigliabile disattivare la scheda Ethernet usando `ifconfig eth0 down`.

## 7.1 Programmi di configurazione per le schede Ethernet

### 7.1.1 Schede WD80x3

Per quanti hanno schede wd80x3, c'è il programma `wdsetup` che può essere trovato nel file `wdsetup-0.6a.tar.gz` nei siti ftp su Linux. Non è mantenuto attivamente e non è aggiornato da un po' di tempo. Se nel proprio caso funziona, allora bene; se non funziona, si usi la versione DOS che si dovrebbe aver ricevuto con la scheda. Se non si ha la versione DOS, si sarà felici di sapere che i dischetti aggiornati di configurazione e dei driver possono essere scaricati dal sito ftp della SMC. Natualmente, si *deve* possedere una scheda con la EEPROM per usare questo programma. Le *vecchie*, ma proprio vecchie, schede wd8003 e alcuni cloni del wd8013 per configurare la scheda usano dei ponticelli.

### 7.1.2 Schede Digital/DEC

La scheda Digital EtherWorks 3 può essere configurata in maniera simile che con il programma DOS `NICSETUP.EXE`. David C. Davies ha scritto, assieme al driver, questo progrmma e altri strumenti per la EtherWorks 3. Si cerchi nella directory `/pub/linux/system/Network/management` nel sito FTP su Linux più vicino, il file chiamato `ewrk3tools-X.XX.tar.gz`.

### 7.1.3 Schede NE2000+ o AT/LANTIC

Alcune implementazioni del DP83905 della National Semiconductor (come le AT/LANTIC e le NE2000+) sono configurabili via software (si noti che queste schede emulano anche una wd8013!). Per configurare queste schede si può scaricare il file `/pub/linux/setup/atlantic.c` dal server ftp di Donald, `cesdis.gsfc.nasa.gov`. Inoltre, con tutte queste schede sembra funzionare anche il programma per le schede DP83905 della Kingston, poiché non fa controlli sul tipo di rivenditore prima di permetterne l'uso. Si segua l'URL seguente: *Kingston Software* `<http://www.kingston.com/download/etherx/etherx.htm>` e si scarichi `20XX12.EXE` e `INFOSET.EXE`.

Si faccia attenzione quando si configurano schede NE2000+, poiché alcune impostazioni errate possono causare problemi. Un esempio tipico è l'abilitazione accindentale della ROM di boot nella EEPROM (anche

se la ROM non è installata) con una impostazione che va in conflitto con la scheda VGA. Il risultato è che il computer semplicemente fa beep quando lo si accende e sullo schermo non succede niente.

Solitamente si può risolvere questa situazione nel modo seguente. Si rimuova la scheda dalla macchina e poi si riavvii e si entri nel menu di configurazione del BIOS. Si cambi la voce 'Display Adapter' in 'Not Installed' e si imposti il disco di avvio di default in 'A:' (il proprio lettore di floppy). Si cambi anche la voce 'Wait for F1 if any Error' in 'Disabled'. In questo modo, il computer dovrebbe avviarsi senza l'intervento dell'utente. Ora si crei un dischetto DOS avviabile ('format a: /s /u') e si copi dentro il floppy il programma `default.exe` dell'archivio `20XX12.EXE` suddetto. Poi si digiti `echo default > a:autoexec.bat` in modo tale che il programma che reimposta i valori predefiniti della scheda sia eseguito automaticamente quando si avvia da questo dischetto. Si spenga la macchina, si reinstalli la scheda ne2000+, si inserisca il nuovo dischetto di boot e la si riaccenda. Probabilmente farà ancora beep, ma alla fine si dovrebbe vedere la lucetta del floppy che si accende quando finalmente fa il boot. Si aspetti un paio di minuti che il floppy si fermi, il che indica che ha finito di eseguire il programma `default.exe` e poi si spenga il computer. Dopo lo si riaccenda ancora e teoricamente si dovrebbe avere ancora un display che funziona, permettendo così di risistemare le impostazioni del BIOS e modificare i valori della EEPROM della scheda come si vuole.

Si noti che se non si ha DOS sotto mano, si può fare tutto quello sopra con un dischetto di avvio di Linux che lancia automaticamente il programma `atlantic` di Donald (con le giuste opzioni d'avvio) invece che con un dischetto di avvio di DOS che lancia automaticamente il programma `default.exe`.

### 7.1.4 Schede 3Com

La famiglia di schede Etherlink III della 3Com (p.es. 3c5x9) può essere configurata usando un'altra utilità di configurazione di Donald. Per configurarle si può scaricare il file `/pub/linux/setup/3c5x9setup.c` dal server ftp di Donald, `cesdis.gsfc.nasa.gov` (si noti che il programma DOS di configurazione 3c5x9B può avere più opzioni pertinenti alla nuova serie "B" della famiglia Etherlink III).

## 7.2 Programmi diagnostici per schede Ethernet

Tutti i programmi diagnostici scritti da Donald possono essere ottenuti da questo URL:

*Ethercard Diagnostics* `<ftp://cesdis.gsfc.nasa.gov/pub/linux/diag/index.html>`

Allied Telesis AT1700: si cerchi il file `/pub/linux/diag/at1700.c` su `cesdis.gsfc.nasa.gov`.

Cabletron E21XX: si cerchi il file `/pub/linux/diag/e21.c` su `cesdis.gsfc.nasa.gov`.

P PCLAN+: si cerchi il file `/pub/linux/diag/hp+.c` su `cesdis.gsfc.nasa.gov`.

Intel EtherExpress: si cerchi il file `/pub/linux/diag/eexpress.c` su `cesdis.gsfc.nasa.gov`.

Schede NE2000: si cerchi il file `/pub/linux/diag/ne2k.c` su `cesdis.gsfc.nasa.gov`. C'è ancora una versione PCI per gli ormai comuni cloni della NE2000-PCI.

RealTek (ATP) Pocket adaptor: si cerchi il file `/pub/linux/diag/atp-diag.c` su `cesdis.gsfc.nasa.gov`.

Tutte le altre schede: si provi a digitare `cat /proc/net/dev` e `dmesg` per vedere quali informazioni utili possiede il kernel sulla scheda in oggetto.

# 8 Informazioni tecniche

Queste informazioni saranno utili a quanto vogliono capire un po' meglio come funziona la loro scheda, oppure vogliono giocare con il driver o anche provare a fare il loro driver personale per una scheda che attualmente non è supportata. Se non si cade in queste categorie, allora si può pure saltare questa sezione.

## 8.1 I/O programmato, memoria condivisa e DMA a confronto

Se già si ricevono e inviano pacchetti back-to-back (NdT letteralmente "uno dopo l'altro"), non si possono mettere altri bit nel cavo. Ogni scheda Ethernet moderna può ricevere pacchetti back-to-back. I driver per Linux per DP8390 (wd80x3, SMC-Ultra, 3c503, ne2000, ecc.) arrivano abbastanza vicino all'invio di pacchetti back-to-back (il limite è dato dalla reale latenza degli interrupt) e l'hardware 3c509 e AT1500 non ha problemi ad inviare automaticamente pacchetti back-to-back.

Il bus ISA può fare 5.3MB/sec (42Mb/sec), che sembra più che sufficiente per una Ethernet a 10Mbps. Nel caso di schede a 100Mpbs, chiaramente è necessario un bus più veloce per sfruttare la larghezza di banda della rete.

### 8.1.1 I/O programmato (Programmed I/O) (es. NE2000, 3c509)

Pro: Non usa nessuna risorsa di sistema riservata, solo alcuni registri di I/O e non ha il limite dei 16M.

Contro: Solitamente più bassa è la velocità di trasferimento, più la CPU attende senza far niente, e l'accesso ai pacchetti interleaved è solitamente difficile se non impossibile.

### 8.1.2 Memoria condivisa (Shared memory) (es. WD80x3, SMC-Ultra, 3c503)

Pro: Più veloce e semplice dell'I/O programmato e inoltre permette l'accesso casuale ai pacchetti. Dove possibile, i driver per Linux calcolano il checksum dei pacchetti IP in ingresso non appena sono copiati fuori dalla scheda. Tale cosa risulta in un'ulteriore riduzione dell'uso della CPU rispetto ad una equivalente scheda PIO.

Contro: Usa maggiore spazio di memoria (molto grande per gli utenti DOS, essenzialmente non problematico sotto Linux) e usa ancora parecchio la CPU.

### 8.1.3 Accesso diretto alla memoria (DMA) di tipo slave (normale) (non per Linux!)

Pro: Libera la CPU durante il reale trasferimento dei dati.

Contro: Il controllo delle condizioni al contorno, l'allocazione di buffer contigui e la programmazione dei registri DMA la rende la tecnica più lenta fra tutte. Inoltre satura facilmente un canale DMA e richiede buffer allineati in memoria bassa.

### 8.1.4 Accesso diretto alla memoria (DMA) di tipo bus master (es. LANCE, DEC 21040)

Pro: Libera la CPU durante il trasferimento dei dati, può legare assieme i buffer, nel bus ISA può richiede una piccola (nulla) perdita del tempo di CPU. La maggior parte dei driver per Linux bus-master usano lo schema 'copybreak' nel quale i grossi pacchetti sono messi direttamente dalla scheda nel buffer di rete del kernel, e i pacchetti piccoli sono copiati invece dalla CPU che li carica in cache per le eleborazioni successive.

Contro: (Applicabile solamente alle schede per bus ISA) Per la scheda sono necessari buffer in memoria bassa e di canali DMA. Qualsiasi dispositivo bus-master avrà problemi con altri dispositivi non bus-master che si appropriano del bus, come alcuni primitivi adattatori SCSI. Alcuni chipset delle schede madri mal progettati hanno problemi con il bus-master. Una ragione per non usare alcun tipo di dispositivo DMA è un processore 486 progettato come rimpiazzo di un 386: questi processori devono scaricare la loro cache ad ogni ciclo di DMA (fra questi ci sono Cx486DLC, Ti486DLC, Cx486SLC, Ti486SLC, ecc.)

## 8.2 Scrivere un driver

La sola cosa che serve per usare una scheda Ethernet con Linux è il driver appropriato. Per questo è essenziale che i costruttori rilascino le informazioni tecniche di programmazione al pubblico senza che qualcuno debba vendersi per vita per averle. Un buon esempio sulla possibilità di ottenere documentazione (oppure se non si sta scrivendo codice, sulla possibilità che qualcun'altro scriverà quel driver di cui si ha veramente bisogno) è la disponibilità del packet driver Crynwr (nato Clarkson). Russ Nelson guida questa operazione ed è stato molto d'aiuto nel supportare lo sviluppo dei driver per Linux. I *net-surfer* possono provare questo URL per vedere il software di Russ.

*Russ Nelson's Packet Drivers* `<http://www.crynwr.com/crynwr/home.html>`

Avuta la documentazione si può scrivere un driver per la propria scheda e usarlo per Linux (almeno in teoria). Si tenga presente che alcuni dei vecchi hardware che erano stati pensati per macchine di tipo XT non funzioneranno molto bene in un ambiente multitasking come Linux. L'uso di uno di questi comporterà non pochi problemi se la propria rete deve sostenere un traffico ragionevole.

La maggior parte delle schede ha dei driver per le interfacce MS-DOS come NDIS o ODI, ma questi sono inutili per Linux. Molti hanno suggerito di fare il link direttamente di questi o utilizzare una traduzione automatica, ma queste sono cose praticamente impossibili. I driver MS-DOS si aspettano di essere in modalità a 16 bit e si basano su 'interrupt software', cose che sono entrambe incompatibili con il kernel di Linux. Questa incompatibilità è in realtà una caratteristica di Linux, in quanto alcuni driver per Linux sono spesso decisamente migliori delle loro controparti MS-DOS. La serie '8390' dei driver, per esempio usa un buffer di trasmissione ping-pong, che ora è stato introdotto anche nel mondo MS-DOS.

(Buffer Tx di tipo ping-pong significa usare almeno due buffer di pacchetti per i pacchetti da trasmettere. Uno è caricato mentre la scheda sta trasmettendo l'altro. Il suo contenuto viene poi trasmesso non appena è conclusa la trasmissione dell'altro e così via. In questo modo, la maggior parte delle schede sono in grado di inviare continuamente pacchetti back-to-back nel cavo.)

OK. Quindi si è deciso che si vuole scrivere un driver per la scheda Ethernet CippaLippa, poiché si possiedono le informazioni di programmazione e nessuno l'ha ancora fatto (... queste sono le due condizioni principali ;-) Si dovrebbe partire dallo scheletro per un driver di rete fornito assieme ai sorgenti di Linux. Può essere trovato nel file /usr/src/linux/drivers/net/skeleton.c in tutti i kernel recenti. Si dia un'occhiata anche alla Kernel Hackers Guide, a questo indirizzo: *KHG* `<http://www.redhat.com:8080/HyperNews/get/khg.html>`

## 8.3 Interfaccia dei driver verso il kernel

Ecco alcune note sulle funzioni che si dovranno scrivere se si crea un nuovo driver. Si leggano queste cose assieme allo scheletro del driver citato prima per aiutarsi a chiarire un po' le cose.

### 8.3.1 Probe (rilevamento)

È chiamata all'avvio per controllare l'esistenza della scheda. Meglio se si può farlo non intrusivamente leggendo dalla memoria, ecc. Può anche leggere dalle porte I/O. La scrittura iniziale nelle porte di I/O durante il probe *non è una buona cosa* in quanto può uccidere un altro dispositivo. Una parte dell'inizializzazione del dispositivo è fatta qui (allocando lo spazio I/O, IRQ, riempendo i campi di dev->???, ecc). Si deve sapere a quali porte di I/O o zone di memoria può essere configurata la scheda, come abilitare la memoria condivisa (se usata), come selezionare/abilitare la generazione degli interrupt, ecc.

### 8.3.2 Interrupt handler (gestore dell'interrupt)

È chiamato dal kernel quando la scheda genera un interrupt. Ha il compito di determinare perché la scheda lo ha generato e comportarsi di conseguenza. Le usuali condizioni di interrupt sono la ricezione di dati, il completamento della trasmissione, la segnalazione di condizioni d'errore. Si deve conoscere ogni bit di stato dell'interrupt di una qualche importanza in modo da portersi comportare di conseguenza.

### 8.3.3 Funzione transmit

È collegata a dev->hard_start_xmit() ed è chiamata dal kernel quando ci sono alcuni dati che il kernel vuole depositare nel dispositivo. Mette i dati nella scheda e avvia la trasmissione. Si deve sapere come raccogliere i dati, come metterli dentro la scheda (copia della memoria condivisa, trasferimento PIO, DMA?) e il posto giusto dove metterli. Poi si deve sapere come dire alla scheda di inviare i dati nel cavo e (possibilmente) sollevare un interrupt quando ha fatto. Quando l'hardware non può accettare ulteriori pacchetti dovrebbe impostare il flag dev->tbusy. Quando è disponibile dello spazio, solitamente durante un interrupt per il completamento della trasmissione, dev->tbusy dovrebbe essere liberato e i livelli più elevati informati con `mark_bh(INET_BH)`.

### 8.3.4 Funzione receive

È chiamata dall'interrupt handler del kernel quando ci sono dei dati nella scheda. Toglie i dati dalla scheda, li impacchetta in un sk_buff e informa il kernel che i dati sono lì, dimodoché questo possa fare una netif_rx(sk_buff). Si deve sapere come abilitare la generazione di un interrupt sulla ricezione di dati, come controllare qualsiasi bit di stato della ricezione e come togliere i dati dalla scheda (ancora memoria condivisa, PIO, DMA, ecc.).

### 8.3.5 Funzione open

È collegata a dev->open ed è chiamata dagli strati di rete quando qualcuno fa `ifconfig eth0 up` per attivare il dispositivo ed abilitarlo per la Rx/Tx di dati. Qualsiasi inizializzazione speciale che non viene fatta nella sequenza di probe (abilitare la generazione degli IRQ, ecc.) dovrebbe andare qui.

### 8.3.6 Funzione close (opzionale)

Questa mette la scheda in uno stato decente quando qualcuno fa `ifconfig eth0 down`. Dovrebbe liberare gli IRQ e i canali DMA se l'hardware lo permette e disabilitare qualsiasi cosa che possa far risparmiare corrente (come il transceiver).

### 8.3.7 Funzioni varie

Cose come una funzione di reinizializzazione, cosicché se le cose vanno male, il driver può provare, come ultima risorsa, a ripristinare la scheda. Solitamente è fatto quando una trasmissione va in time out o cose simili. Può essere utile anche una funzione per leggere i registri di statistica delle scheda (se ce li ha).

## 8.4 Informazioni tecniche dalla 3Com

Se si è interessati a lavorare su un driver per una scheda 3Com, si possono ottenere informazioni tecniche direttamente dalla 3Com. Cameron è stato talmente gentile da spiegarci come averle:

«Gli adattatori Ethernet della 3Com sono documentati per chi vuole scrivere un driver nei nostri 'Technical Reference' (TR). Questi manuali descrivono le interfacce di programmazione delle schede ma non parlano della diagnostica, programmi d'installazione, ecc. tutte quelle cose che interessano all'utilizzatore finale.

I TR sono distribuiti dal dipartimento di marketing della Network Adapter Division. Per far funzionare le cose in maniera efficiente, la distribuzione è centralizzata in una cosa detta 'CardFacts'. CardFacts è un sistema telefonico automatizzato. Se lo si chiama con un telefono a toni può inviare fax dove si richiede. Per ottenere un TR, si chiami CardFacts al 408-727-7021. Gli si chieda il "Developer's Order Form" (modulo d'ordine per sviluppatori), documento numero 9070. Si tenga sotto mano il proprio numero di fax quando si chiama. Si compili il modulo d'ordine e lo si invii via fax al 408-764-5004. I manuali sono spediti con il servizio "2nd Day" della Federal Express.

Alcuni pensano che diamo via troppo facilmente i manuali e cercano prove per dire il sistema è troppo costoso e che occupa troppo tempo e sforzi. Da tempo, i nostri clienti sono sempre stati contenti di questa cosa e non ci sono mai stati problemi nonostante le quantità di richieste che riceviamo. Abbiamo bisogno della loro continua cooperazione e riserbo per mantenere le cose così.»

## 8.5 Note sulle schede basate su PCnet / LANCE di AMD

Il chip LANCE (Local Area Network Controller for Ethernet) era l'offerta originale dell'AMD e da allora è stato rimpiazzato dal chip 'PCnet-ISA', noto anche come 79C960. Si noti che il nome 'LANCE' è stuck e alcuni ancora fanno riferimento ai nuovi chip con il vecchio nome. Dave Roberts della Network Products Division di AMD è stato talmente gentile da fornire le seguenti informazioni riguardo questo chip:

«Funzionalmente, è equivalente a una NE1500. L'insieme dei registri è indentico a quello del vecchi LANCE con le aggiunte dell'architettura 1500/2100. I vecchi driver per 1500/2100 funzioneranno anche con il PCnet-ISA. L'architettura degli NE1500 e degli NE2100 in pratica è la stessa. Inizialmente la Novell l'ha chiamata 2100, ma hanno poi provato a distinguere tra le schede per il coassiale e le 10Base-T. Qualsiasi cosa che era solamente 10Base-T è stata numerata nell'intervallo 1500. Questa è la sola differenza.

Molte compagnie offrono prodotti basati su PCnet-ISA, incluse HP, Racal-Datacom, Allied Telesis, Boca Research, Kingston Technology, ecc. Le schede in pratica sono le stesse se non per il fatto che alcuni costruttori hanno aggiunte le caratteristiche 'jumperless' (senza ponticelli) che permettono la configurazione via software. La maggior parte non l'ha fatto. L'AMD offre un progetto standard per una scheda che usa il chip PCnet-ISA e molti costruttori usano il nostro progetto senza modifiche. Questo significa che chiunque voglia scrivere dei driver per la maggior parte delle schede basate su PCnet-ISA può semplicemente prendere il data-sheet dall'AMD. Si chiami il nostro centro di distribuzione della documentazione al (800)222-9323 e si chieda del data sheet per il Am79C960, il chip PCnet-ISA. È gratis.

Un modo veloce per capire se la scheda è una di quelle basate sul progetto standard, è quello di osservarla. Se è una standard, dovrebbe avere un grosso chip, un cristallo, una piccola PROM d'indirizzo IEEE, probabilmente un socket per una ROM di boot e un connettore (1, 2 o 3 a seconda dei mezzi supportati). Si noti che se è una scheda per cavo coassiale, avrà anche un po' di roba per il trasceiver, ma dovrebbe essere nei pressi del connettore e ben distante dal PCnet-ISA.»

Una nota per gli aspiranti hacker è che differenti implementazioni del LANCE effettuano il 'restart' (riavvio) in modi diversi. Alcune riprendono da dove sono state interrotte, mentre altre ripartono dall'inizio, come se fossero state inizializzate.

## 8.6 Multicast e modalità promiscua

Un'altra delle cose a cui ha lavorato Donald è l'implementazione degli "agganci" (hook) per il multicast e la modalità promiscua (promiscuous mode). Tutti i driver ISA *rilasciati* (cioè **non** ALPHA) ora supportano

la modalità promiscua.

Donald scrive: «Inizierei con il parlare della modalità promiscua, che concettualmente è facile da implementare. Per la maggior parte dell'hardware semplicemente si deve impostare un bit di registro e da allora si riceveranno tutti i pacchetti che passano su quel cavo. Beh, non è proprio così facile: per alcune schede si deve disabilitare la scheda (potenzialmente perdendo alcuni pacchetti), riconfigurarla e poi riabilitarla. OK, visto che questo è facile, adesso passo a qualcosa che non è proprio così ovvia: il multicast. Può essere fatto in due modi:

1. Usando la modalità promiscua e un filtro di pacchetti come il Berkeley packet filter (BPF). Il BPF è un linguaggio stack di pattern matching (ricerca delle corrispondenze), dove si scrive un programma che raccoglie gli indirizzi ai quali si è interessati. Il suo vantaggio è che è molto generale e programmabile. Lo svantaggio è che non c'è un metodo generale per il kernel per evitare di attivare la modalità promiscua e far passare ogni pacchetto che attraversa il cavo su ognuno dei filtri di pacchetti registrati. Si veda 8.7 (Il Berkeley Packet Filter) per maggior informazioni.
2. Usare il filtro multicast su scheda che hanno la maggior parte delle schede.

Immagino che dovrei elencare quel poco che mettono a disposizione le schede o i chip:

| Chip/scheda | Promiscua | Filtro multicast |
|---|---|---|
| Seeq8001/3c501 | Sì | Filtro binario (1) |
| 3Com/3c509 | Sì | Filtro binario (1) |
| 8390 | Sì | Hash a sei bit con Autodin II (2) (3) |
| LANCE | Sì | Hash a sei bit con Autodin II (2) (3) |
| i82586 | Sì | Hash a sei bit con Hidden Autodin II (2) (4) |

1. Queste schede affermano di avere un filtro, ma è un semplice sì/no: «accetta tutti i pacchetti multicast» o «non accettare i pacchetti multicast».
2. AUTODIN II è il polimonio di CRC (codice ciclico a ridonadanza) standard di Ethernet. In questo schema, degli indirizzi multicast viene calcolata l'hash e vengono poi ricercati in una tabella hash. Se è abilitato il bit corrispondente del pacchetto, allora questo è accettato. I pacchetti Ethernet sono progettati in modo che l'hardware per fare questa cosa sia banale: semplicemente si memorizzano i sesti bit del circuito CRC (comunque necessario per il controllo degli errori) dopo i primi sei otteti (l'indirizzo di destinazione) e li si usa per indicizzare nella tabella hash (sei bit: una tabella a 64 bit).
3. Questi chip usano la hash a sei bit e la tabella dev'essere calcolata e caricata dall'host. Questo significa che il kernel deve contenere il codice per il CRC.
4. Il 82586 usa la hash a sei bit internamente, ma calcola da solo la tabella di hast dall'elenco degli indirizzi multicast da accettare.

Si noti che nessuno di questi chip fa un filtraggio perfetto e c'è ancora bisogno di un modulo a livello intermedio per fare il filtraggio finale. Si noti pure che in tutti i casi si deve mantenere un'elenco completo degli indirizzi multicast accettati per ricalcolare la tabella di hash quando cambia.

## 8.7 Berkeley Packet Filter (BPF)

L'idea diffusa fra gli sviluppatori è che le funzionalità BPF non dovrebbero essere fornite dal kernel, ma dovrebbero essere in una libreria di compatibilità (si spera poco usata).

Per quelli che non lo conoscono, il BPF (Berkeley Packet Filter – filtro dei pacchetti di Berkeley) è un meccanismo per specificare agli strati di rete del kernel i pacchetti ai quali si è interessati. È implementato come un interprete specializzato di un linguaggio stack costruito dentro il codice di rete a basso livello. Un'applicazione passa un programma scritto in questo linguaggio al kernel ed il kernel esegue il programma su ciascun pacchetto in ingresso. Se il kernel ha più applicazioni BPF, ogni programma è eseguito su ogni pacchetto.

Il problema è che è difficile dedurre dal programma di filtraggio dei pacchetti a quale tipo di pacchetti l'applicazione è veramente interessata, e quindi la soluzione generale è di eseguire sempre il filtro. Si immagini un programma che registra un programma BPF per raccogliere il flusso a bassa velocità inviato ad un indirizzo multicast. La maggior parte delle schede hanno un filtro hardware per indirizzi multicast implementato come una tabella di hash a 64 voci che ignora i pacchetti multicast maggiormente non voluti. Quindi c'è la possibilità di rendere questa operazione poco costosa. Ma con il BPF il kernel deve passare l'interfaccia in modalità promiscua, ricevere _tutti_ i pacchetti e farli passare attraverso il filtro. Questa cosa funziona, ma quel che è difficile è darne conto al processo che ha richiesto i pacchetti.

# 9 Networking con computer portatili

Ci sono diversi modi per mettere il proprio portatile in una rete. Si può usare il codice di SLIP (e andare alla velocità delle porte seriali); si può prendere un portatile con già al suo interno uno slot PCMCIA; si può prendere un portatile con una docking station e attaccarci una scheda Ethernet ISA; oppure si può usare un adattatore Ethernet su porta parallela.

## 9.1 Usare SLIP

Questa è la soluzione più economica, ma considerevoltemente la più difficoltosa. Inoltre non permetterà velocità di trasmissione molto elevate. Poiché lo SLIP non centra con le schede Ethernet non sarà discusso ulteriormente qui. Si veda il NET-2 HOWTO.

## 9.2 Supporto PCMCIA

Si provi a determinare esattamente l'hardware che si possiede (cioè costruttore della scheda, costruttore del chip del controller PCMCIA) e poi si chieda nel canale LAPTOPS. Non ci si aspetti che le cose sia poi così semplici. Si dovranno fare un po' di cose a mano, applicare patch al kernel, ecc. Forse un giorno si sarà in grado di scrivere semplicemente 'make config' 8-)

Al momento, i due chipset PCMCIA supportati sono il Databook TCIC/2 e l'Intel i82365.

A tsx-11.mit.edu ci sono diversi programmi nella directory /pub/linux/packages/laptops/ che potrebbero tornare utili. Si va da driver per schede Ethernet PCMCIA a programmi che comunicano con il chip del controller PCMCIA. Si noti che questi driver sono solitamente legati ad uno specifico chip PCMCIA (l'Intel 82365 e il TCIC/2)

Per le schede compatibili con NE2000, alcuni hanno avuto successo semplicemente configurando la scheda sotto DOS e poi avviando Linux dalla riga di comando del DOS con loadlin.

La situazione sembra buona per quelli che vogliono il supporto PCMCIA, in quanto sono stati fatti dei progressi sostanziali. Il pioniere in questi sforzi è David Hinds. Il suo ultimo pacchetto per il supporto PCMCIA può essere ottenuto da:

*PCMCIA Package* `<ftp://cb-iris.stanford.edu/pub/pcmcia>`

Si cerchi un file tipo `pcmcia-cs-X.Y.Z.tgz` dove X.Y.Z sarà il numero dell'ultima versione. Questo sarà probabilmente depositato anche nel sito FTP `tsx-11.mit.edu`.

Si noti che l'abilitatore PCMCIA di Donald funziona come processo a livello utente mentre quella di David Hinds è una soluzione a livello kernel. Si sarà meglio serviti dal pacchetto di David in quanto è maggiormente usato e in continuo sviluppo.

### 9.3 Schede Ethernet ISA nella docking station

Le docking station per i portatili costano circa 250 dollari e forniscono due slot ISA a piena grandezza, due porte seriali e una porta parallela. La maggior parte delle docking station sono alimentate direttamente dalle batterie del portatile e solo alcune permettono l'aggiunta di batterie addizionali se si usano schede ISA corte. Si può così aggiungere una scheda Ethernet poco costosa e gioire delle prestazioni di una Ethernet a piena velocità.

### 9.4 Adattatori pocket (su porta parallela)

Anche gli adattori Ethernet 'pocket' possono soddisfare le proprie necessità. Si noti che la velocità di trasferimento non sarà affatto elevata (forse picchi di 200kB/s?) a causa delle limitazioni dell'interfaccia su porta parallela.

Inoltre ci si deve collegare alla presa di alimentazione sulla parete. Qualche volta si può evitare di attaccare l'adattatore alla presa, comprando o facendosi un cavo per prendere l'alimentazione dalla porta per la tastiera del portatile (si veda 5.31.1 (alimentazione dalla tastiera)).

Si veda 5.14.5 (DE-600 / DE-620) e 5.31.1 (RealTek) per due adattatori pocket supportati.

## 10 Miscellanea

Tutta quella roba che non ci stava bene da nessun'altra parte è stata messa qui. Potrebbe non essere rilevante e potrebbe non essere d'interesse generale, ma è comunque qui.

### 10.1 Passare al kernel argomenti Ethernet

Ci sono due comandi generici che possono essere passati al kernel al momento del boot: `ether` e `reserve`. Questa cosa può essere fatta con LILO, loadlin o qualsiasi altra utilità di boot che accetta argomenti opzionali.

Per esempio, se il comando fosse 'blah' e si aspettasse 3 argomenti (diciamo 123, 456 e 789) allora, con LILO, si potrebbe usare:

```
LILO: linux blah=123,456,789
```

Per maggiori informazioni sugli argomenti di boot (e un elenco completo) si veda il *BootPrompt-HOWTO* `<http://metalab.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html>`.

#### 10.1.1 Il comando `ether`

L'argomento `ether=` è usato in congiunzione a driver direttamente compilati dentro al kernel. *Non ha assolutamente alcun effetto* su driver compilati come moduli. Nella sua forma più generale, può essere qualcosa di simile:

```
ether=IRQ,IND_BASE,PARAM_1,PARAM_2,NOME
```

Tutti gli argomenti sono opzionali. Il primo argomento non numerico è intrepretato come NOME.

**IRQ:** ovvio. Un valore di IRQ di '0' (solitamente il valore predefinito) significa autoIRQ. È accidentale che l'impostazione dell'IRQ sia prima di quella dell'ind_base: questa cosa verrà corretta non appena cambierà qualcos'altro.

**IND_BASE:** altrettanto ovvio. Il valore '0' (solitamente quello predefinito) indica di rilevare una scheda nell'elenco di indirizzi spefici per quella scheda.

**PARAM_1:** Originariamente usato come valore per ridefinire l'indirizzo iniziale della memoria per una scheda Ethernet a memoria condivisa, come la WD80*3. Alcuni driver usano i 4 bit meno significativi di questo valore per impostare il livello dei messaggi di debug. 0: predefinito, 1-7: livello 1..7 (con 7 si ottiene il massimo della verbosità), 8: livello 0 (nessun messaggio). Inoltre, il driver LANCE usa i 4 bit meno significativi di questo valore per selezionare il canale DMA. Altrimenti usa auto-DMA.

**PARAM_2:** Il driver 3c503 usa questo valore per selezionare tra il transceiver interno ed esterno. 0: predefinito/interno, 1: AUI esterno. La scheda E21XX della Cabletron usa i 4 bit meno significativi di PARAM_2 per selezionare il mezzo d'uscita. Altrimenti lo rileva automaticamente.

**NOME:** Seleziona il dispositivo di rete a cui fa rifemento il valore. Il kernel standard usa i nomi 'eth0', 'eth1', 'eth2' e 'eth3' per le schede Ethernet attaccate sul bus e 'atp0' per gli adattatori Ethernet 'pocket' su porta parallela. Il driver arcnet come nome usa 'arc0'. L'impostazione predefinita è per il rilevamento di un'unica scheda Ethernet, la 'eth0'. L'abilitazione di più schede è possibile solamente impostando esplicitamente il loro indirizzo base usando questi parametri di LILO. Il kernel 1.0 trattava le schede Ethernet basate su LANCE in modo speciale. Gli argomenti di LILO erano ignorati e alle schede LANCE erano sempre assegnati i nomi 'eth<n>' partendo da 'eth0'. Ulteriori schede Ethernet non LANCE dovevano essere esplicitamente assegnate a 'eth<n+1>', e il rilevamento standard di 'eth0' doveva essere disabilitato con qualcosa di simile a 'ether=0,-1,eth0' (sì, questo è un bug).

### 10.1.2 Il comando `reserve`

Questo comando di lilo è usato proprio come il precedente 'ether=', ovvero è accodato al nome di avvio selezionato in lilo.conf.

```
reserve=IO-base,estensione{,IO-base,estensione...}
```

In alcune macchine può essere necessario impedire ai driver di dispositivo di controllare la presenza di dispositivi (auto probing) in regioni specifiche. Ciò può essere dovuto ad hardware mal progettato che causa il *congelamento* del boot (come alcune schede Ethernet), ad hardware che viene mal identificato, ad hardware il cui stato è cambiato da un rilevamento precedente, o semplicemente ad hardware che non vuole essere inizializzato dal kernel.

L'argomento di boot `reserve` indirizza questo problema specificando la regione di una porta I/O che non dovrebbe essere controllata. Tale regione viene riservata nella tabella di registrazione delle porte del kernel come se vi si fosse già rilevato un dispositivo. Si noti che questo meccanismo non dovrebbe essere necessario nella maggior parte della macchine. Sarà necessario usare questo comando solo quando c'è un problema o un caso particolare.

Le porte I/O nella regione specificata sono protette dai controlli dei dispositivi. Questa cosa si è resa necessaria quando alcuni driver si piantavano su una NE2000 o mal identificavano altri dispositivi come propri. Un driver di dispositivo corretto non dovrebbe controllare una regione riservata a meno che qualche altro argomento di boot non specifici esplicitamente di farlo. La maggior parte dei driver ignora la tabella di registrazione delle porte se gli viene passato un indirizzo specifico.

Per esempio, la riga di boot

```
LILO: linux reserve=0x300,32 ether=0,0x300,eth0
```

impedisce a tutti i device driver, tranne a quelli della scheda Ethernet, di controllare la regione 0x300-0x31f.

Come al solito, c'è un limite di 11 parametri come in tutti i comandi di boot, quindi si possono specificare solo 5 regioni riservate per ciascun comando `keyword`. Possono essere specificati più `reserve` se si hanno richieste insolitamente complicate.

## 10.2 Usare un driver Ethernet come modulo

La maggior parte delle distribuzioni di Linux utilizzano ora dei kernel con pochissi driver al loro interno. I driver sono piuttosto forniti come gruppo di moduli indipendenti caricabili dinamicamente. Questi driver modulari sono tipicamente caricati dall'amministratore con il comando `modprobe(8)` o in alcuni casi sono automaticamente caricati dal kernel attraverso 'kerneld' (nei 2.0) o 'kmod' (nei 2.1) che a loro volta chiamano `modprobe`.

La propria distribuzione può offrire uno strumento grafico carino per l'impostazione dei moduli Ethernet. Se possibile prima si dovrebbe provare a usare quello. La descrizione che segue fornisce le informazioni che stanno sotto a qualsiasi programma di configurazione e indica cosa quei programmi vanno a modificare.

Le informazioni che controllano quali moduli sono usati e quali opzioni sono passate a ciascun modulo sono solitamente salvate nel file `/etc/conf.modules`. Le due opzioni principali di interesse per le schede Ethernet che saranno usate in questo file sono `alias` e `options`. Il comando `modprobe` consulta questo file per le informazioni sui moduli.

I moduli stessi sono tipicamente salvati in una directory chiamata `/lib/modules/‘uname -r‘/net` dove il comando `uname -r` restituisce la versione del kernel (e.g. 2.0.34). Si può andare là per vedere quale modulo corrisponde alla propria scheda.

La prima cosa che ci deve essere nel proprio file `conf.modules` è qualcosa che dice a `modprobe` quale driver usare per l'interfaccia di rete `eth0` (e per `eth1` e per...). Per questa cosa si dovrà usare il comando `alias`. Per esempio, se si ha una scheda ISA EtherEZ della SMC che usa il modulo del driver `smc-ultra.o`, si deve creare `alias` per questo driver che corrisponda a `eth0`, aggiungendo la riga:

```
alias eth0 smc-ultra
```

L'altra cosa di cui si può aver bisogno è una riga `options` che indica quali opzioni devono essere usate con un particolare modulo (o alias di modulo). Continuando con l'esempio di prima, se si usa solamente la riga `alias` con nessuna riga `options`, il kernel avviserà (si veda `dmesg`) che l'auto rilevamento di una scheda ISA non è una buona idea. Per venir a capo di questo problema, si aggiunga un'altra riga che dice al modulo a quale indirizzo I/O base è configurata la scheda, ad esempio l'indirizzo esadecimale `0x280`.

```
options smc-ultra io=0x280
```

La maggior parte dei moduli ISA accettano parametri come `io=0x340` e `irq=12` sulla riga di comando di `insmod`. Fornire questi parametri è *RICHIESTO* o almeno *CALDAMENTE CONSIGLIATO* per evitare il rilevamento della scheda. Diversamente dai dispositivi PCI e EISA, non c'è alcun modo sicuro per fare l'auto rilevamento della maggior parte dei dispositivi ISA e quindi lo si dovrebbe evitare quando si usa un driver come modulo.

Un elenco di tutte le opzioni che ciascun modulo accetta può essere trovata nel file:

`/usr/src/linux/Documentation/networking/net-modules.txt`

Si raccomanda la sua lettura per trovare quali opzioni si possono usare con la propria scheda. Si noti che alcuni moduli supportano un elenco di valori separati da virgola. Solitamente questo è il caso di moduli che sono in grado di gestire più dispositivi con un unico modulo, come tutti i driver basati su 8390 e il driver PLIP. Per esempio:

```
options 3c503 io=0x280,0x300,0x330,0x350 xcvr=0,1,0,1
```

Con la riga precedente si avrà un unico modulo che controlla quattro schede 3c503, con la scheda 2 e 4 che usano il tranceiver esterno. Non si aggiungano spazi attorno a '=' o alle virgole.

Si noti inoltre che un modulo *occupato* non può essere rimosso. Ciò significa che si dovrà fare `ifconfig eth0 down` (disattivare la scheda Ethernet) prima di rimuovere il modulo.

Il comando `lsmod` mostrerà quali moduli sono caricati e se sono in uso, mentre il comando `rmmod` può essere usato per rimuoverli.

## 10.3 Documentazione correlata

La maggior parte dei queste informazioni provengono da messaggi che ho salvato dai gruppi comp.os.linux, che si sono dimostrati una insostituibile fonte di informazioni. Altre informazioni utili provengono da un gruppo di piccoli file di Donald stesso. Naturalmente, se si sta impostando una scheda Ethernet, allora sarà bene leggere il NET-2 Howto in modo che si possa realmente configurare il software che la userà. Inoltre, se ci si diverte a fare un po' l'hacker, si possono sempre trovare alcune informazioni addizionali nei file sorgente dei driver. Solitamente prima che cominci il codice c'è sempre un paragrafo o due che descrive i punti importanti.

A quanti cercano informazioni che non sono in alcun modo specifiche su Linux (per esempio, cos'è 10BaseT, cos'è AUI, cosa fa un hub, ecc.) suggerisco caldamente di rivolgersi a newsgroup come *comp.dcom.lans.ethernet* e/o *comp.sys.ibm.pc.hardware.networking*. Gli archivi dei newsgroup come quelli a `dejanews.com` possono essere una insostituibile fonte di informazioni. Si possono prendere le FAQ da RTFM (che conserva tutte le FAQ dei newsgroup) al l'URL seguente:

*Usenet FAQs* `<ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/>`

Si può anche dare un'occhiata alla cosiddetta 'Ethernet-HomePage', che è all'URL seguente:

*Ethernet-HomePage* `<http://wwwhost.ots.utexas.edu/ethernet/ethernet-home.html>`

## 10.4 Liberatoria e copyright (in inglese)

This document is *not* gospel. However, it is probably the most up to date info that you will be able to find. Nobody is responsible for what happens to your hardware but yourself. If your ethercard or any other hardware goes up in smoke (...nearly impossible!) we take no responsibility. ie. THE AUTHORS ARE NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION INCLUDED IN THIS DOCUMENT.

This document is Copyright (c) 1993-1997 by Paul Gortmaker. Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that this copyright notice is included exactly as in the original, and that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions.

A hint to people considering doing a translation. First, translate the SGML source (available via FTP from the HowTo main site) so that you can then generate other output formats. Be sure to keep a copy of the original English SGML source that you translated from! When an updated HowTo is released, get the new SGML source for that version, and then a simple `diff -u old.sgml new.sgml` will show you exactly what has changed so that you can easily incorporate those changes into your translated SMGL source without having to re-read or re-translate everything.

If you are intending to incorporate this document into a published work, please make contact (via e-mail) so that you can be supplied with the most up to date information available. In the past, out of date versions of the Linux HowTo documents have been published, which caused the developers undue grief from being plagued with questions that were already answered in the up to date versions.

## 10.5 Chiusura

Se in questo documento si è trovato un qualsiasi errore madornale o informazione non aggiornata, mi si mandi una e-mail. È grande ed è facile non accorgersi di qualcosa. Se si è inviata una mail per una modifica e non è stata inclusa nella versione successiva, non si esiti a scrivere ancora, in quanto potrebbe essersi persa tra la marea di SPAM e junk mail che ricevo.

Grazie!

Paul Gortmaker, `p_gortmaker@yahoo.com`