Linux Ethernet-HOWTO

di Paul Gortmaker v2.9, 25 Agosto 2003

Questo è l'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`, che sono coperte in svariati altri materiali di documentazione Linux. Traduzione fino al 1999 a cura di Lorenza Romano (*titti@dei.unipd.it*) e Giovanni Bortolozzo (*borto@pluto.it*), dopo il 1999 a cura di Federico Lucifredi (*lucifred@fas.harvard.edu*); revisione a cura di Claudio Cattazzo *claudio@pluto.it* .

Indice

1 Introduzione

L'Ethernet-Howto include dettagliate informazioni sul corrente livello di supporto per la maggior parte delle schede Ethernet più comuni. Sono trattati i comuni problemi di configurazione, problemi associati con la scelta del driver giusto, ed il caricare e rendere funzionale detto driver. Non vengono qui trattati i problemi che l'utente deve affrontare negli stadi seguenti del processo di configurazione (come la scelta di un indirizzo IP, il routing e così via). Tali informazioni possono essere facilmente reperite in altre parti della documentazione di Linux.

Ai tempi dell'infanzia di Linux, le vecchie schede di espansione basate sul bus ISA erano la regola. Il bus ISA non dispone di una ragionevole ed affidabile maniera di determinare quali schede siano installate, o che configurazione vada usata con tali schede. Ciò risultava in un più grande coinvolgimento dell'utente nel fornire queste informazioni a Linux, e gli utenti facevano riferimento a questo documento come una guida che li assistesse in questo compito.

Fortunatamente, il nuovo bus PCI si trova in praticamente tutti i computer di oggi, e al bus ISA non resta che raccogliere polvere nei vecchi 386 e 486 dei tempi andati. I progettisti del bus PCI conoscevano la debolezza del bus ISA e così hanno aggiunto funzionalità che permettessero alla scheda di comunicare produttore, modello e settaggi di configurazione da usare al sistema.

Il tramonto del bus ISA ha drasticamente ridotto il coinvolgimento dell'utente nella configurazione delle schede di rete. Di conseguenza, il tipico utente Linux odierno non ha bisogno di fare riferimento a questa guida per assistenza. Vi sono tuttavia sempre delle eccezioni in cui le cose non funzionano come dovrebbero, o dei problemi inaspettati che richiedono risoluzione. E, ovviamente, esistono ancora parecchi vecchi computer ad architettura ISA che continuano a lavorar duro macinando compiti ingrati nascosti nel fondo degli armadi più bui.

Questa revisione tratta i driver Ethernet inclusi coi kernel stabili fino alla versione 2.4.21 compresa. Alcune caratteristiche del futuro kernel 2.6 vengono comunque menzionate.

Ethernet-Howto è scritto da:

Paul Gortmaker, `p_gortmaker @ yahoo.com`

La fonte principale di informazioni per la versione iniziale dell' Ethernet-Howto, originariamente disponibile esclusivamente in formato ASCII:

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

A cui dobbiamo anche la nostra gratitudine per aver scritto la grande maggioranza dei driver attualmente disponibili su Linux per schede Ethernet. Grazie Donald!

Questo documento è Copyright (c) 1993-2003 di Paul Gortmaker. Si, sono oramai dieci anni che io mantengo questo documento! Si vedano la liberatoria e le informazioni sulla copia alla fine di questo documento (8.5 (copyright)) per informazioni circa la ridistribuzione e le solite liberatorie legali come non siamo responsabili per ciò che riuscirete 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 l'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 1.4 (AIUTO - Non funziona!) è rivolta ai nuovi utenti di Linux e vi 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 trovi risposta nella sezione FAQ di questo documento (2 (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 (2.2 (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 (4 (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 (7 (Informazioni tecniche)).

1.3 Cosa devo fare per far funzionare una scheda Ethernet?

Nel modo più conciso possibile, possiamo indicare che dovrete: 1) avere una scheda di espansione Ethernet o una scheda madre con supporto Ethernet integrato, 2) determinare il produttore e modello della scheda o del chip Ethernet integrato, 3) determinare se esiste un driver Linux per questo modello di scheda o di chipset, 4) individuare e caricare detto driver, 5) controllare l'output di questo driver per verificare che abbia individuato questa scheda e 6) configurare i settaggi della vostra nuova interfaccia di rete.

1.4 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 dell'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 ad usare l'utilità `lspci` per ottenere informazioni sui dispositivi installati sul bus PCI del vostro computer. Usare `cat /proc/pci` produce informazioni simili ma non altrettanto complete. Per schede di tipo ISA, si veda la sezione di aiuto sulle schede misteriose (cfr. 4.52 (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 (4 (Informazioni specifiche su...)) che elenca in ordine alfabetico i produttori di schede, i numeri identificativi dei modelli e se esiste o meno un driver per Linux. Se la vostra scheda è catalogata come Non supportata ci si può praticamente arrendere. 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 (8.3 (Usare un driver Ethernet come modulo)).

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 vi aiuterà nel 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 invece nulla 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 e una volta fatto il login, si digiti `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 (2 (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 risulterà inizializzato correttamente e la scheda continuerà a non essere utilizzabile. La maggior parte dei 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 maniere 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 come 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.

Vi 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. Ovviamente lo stesso criterio va seguito quando si invia notizia di un problema - fornite sempre quante più informazioni sia possibile.

1.5 Tipi di cavo che la propria scheda dovrebbe supportare

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 thinnet o cablaggio Ethernet sottile (cavo coassiale RG-58) con connettori BNC (di metallo, da spingere e girare per chiudere) è chiamata tecnicamente 10Base2.

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.

Stanno diventando comuni anche moltissime schede Ethernet in una versione mista (combo) che tipicamente costa solo $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.

Installazioni aziendali estese useranno probabilmente 10BaseT piuttosto che 10Base2. 10Base2 non offre alcuna possibilità di aggiornamento a una 100Base-qualsiasi. 10base2 7egrave; una scelta accettabile per hobbisti che vogliano costruire una rete locale quando l'aquisto di un hub non è desiderabile per qualsiasivoglia motivo.

Si veda 5 (Cavi, coassiali,...) per altre informazioni riguardanti i diversi tipi di cavi Ethernet.

2 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` .

2.1 Come spiego a Linux che driver usare?

Nella maggior parte delle distribuzioni Linux, i driver esistono come moduli a caricamento dinamico, piccoli file binari che vengono caricati dinamicamente dal sistema operativo a runtime. Un modulo fornisce al sistema (più appropriatamente, al kernel) le informazioni necessarie sul come controllare una particolare scheda Ethernet. Il nome del modulo da impiegare è elencato nella intestazione della sezione dedicata alla vostra scheda in questo documento. Una volta che si è a conoscenza del nome del modulo, lo si deve aggiungere al file `/etc/modules.conf` in modo che il kernel Linux sappia che modulo caricare per la vostra scheda. La sintassi è solitamente come segue:

```
alias eth0 nome_del_modulo
options nome_del_modulo optione1=valore1 optione2=valore2 ...
```

Solitamente la riga di opzioni è necessaria solo per hardware antiquato che si appoggia al bus ISA. In sistemi dotati di molteplici schede Ethernet, linee addizionali per `eth1`, `eth2`, e così via sono spesso necessarie.

I moduli sono solitamente collocati nella directory `/lib/modules/`, che è solitamente ulteriormente suddivisa secondo le diverse versioni del kernel (per determinare la versione del kernel che state eseguendo usate `uname -r`) e il sottosistema di cui fanno parte (in questo caso si tratta di `net`). I moduli sono posti qui dal curatore della distribuzione o dal singolo utente quando viene eseguito il comando `make modules_install` dopo la compilazione di un kernel personalizzato e relativi moduli (si veda il kernel HOWTO per ulteriori dettagli sul come compilare un kernel personalizzato secondo i vostri gusti).

Se decidete di ricompilare il vostro kernel, avete tra le vostre opzioni la anche la possibilità di compilare tutti i moduli direttamente nel kernel piuttosto che crearli come moduli esterni. Quando si sceglie di far questo, i driver riconosceranno le componenti hardware di loro competenza al momento del boot. Opzioni possono essere passate ai moduli precompilati nel kernel (si veda il BootPrompt Howto per ulteriori dettagli in merito) dalla riga di comando. L'utente sceglie espressamente quali moduli incorporare nel kernel durante l'esecuzione di del comando `make config` nel processo di ricompilare il kernel (di nuovo, si faccia riferimento al kernel HOWTO per i dettagli).

2.2 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 essa 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. Inoltre, alcune schede sono ora in grado di eseguire parte delle operazioni di checksum direttamente in hardware, liberando la CPU da un ulteriore overhead. Per maggiori dettagli si faccia riferimento alla pagina seguente:

Hardware Checksum/Zerocopy Page `http://www.uow.edu.au/~andrewm/zerocopy.html`

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.

2.3 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 in fatto di driver può essere trovato sul sito Web di Donald: `www.scyld.com`. Qui le cose cambiano abbastanza frequentemente perciò si cerchi attentamente. In alternativa, per localizzare il driver che si sta cercando, può essere più facile usare un browser WWW all'indirizzo:

Don's Linux Network Home Page `http://www.scyld.com/network/`

(ci si guardi da browser che danneggiano i file trasferiti silenzionsamente sostituendo i punti di tabulazione nel sorgente con spazi – se si è incerti si ricorra ad ftp o almeno ad 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 (offri codice o driver in fase di sviluppo o incompleti). Si dovrà rispondere Y (sì) per richiedere l'inclusione di un qualche driver alpha/sperimentale.

2.4 Come usare più di una scheda Ethernet per macchina

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

La risposta a questo quesito dipende se si sta usando il 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 il sistema di un particolare utente vengono caricati una volta che l'avvio del sistema è arrivato al punto tale da poter accedere ai file dei moduli dei driver (contenuti di solito in `/lib/modules/`).

Nel caso di schede PCI, i moduli/driver dovrebbero individuare automaticamente tutte le schede supportate presenti nel sistema, vale a dire che l'utente non è costretto a fornire nessuna configurazione (come l'indirizzo I/O di base ed il numero di IRQ) tranne che in casi eccezzionali, come ad esempio quando si cerca di supportare hardware non conforme agli standard.

Alcuni dei primi kernel avevano un limite massimo di 16 schede Ethernet che potevano venire rilevate al momento del boot, ed alcuni driver modulari per schede ISA avevano un limite di quattro schede supportate per ogni copia del modulo caricata. Si può sempre caricare un altra copia dello stesso modulo sotto un nome diverso per poter utilizzare altre quattro schede se questo rappresenta un limite, oppure ricompilare il modulo con supporto per tante schede quante si desideri.

2.4.1 Con il Driver Come Modulo

Il rilevamento di una scheda non è una operazione affidabile sul bus 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/modules.conf`.

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

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

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 viene 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 poter gestire più schede, la maggior parte dei driver modulari ISA sono in grado di accettare diversi valori di I/O separati da virgole, come in questo esempio. Tuttavia, alcuni driver (forse più vecchi), come il modulo 3c501.o, al momento sono in grado di gestire solo una scheda per modulo caricato. In tal caso si può caricare il modulo due volte per far sì che entrambe le schede siano rilevate. Il file `/etc/modules.conf` dovrebbe in questo caso 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. Viene 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 ciò non sarebbe realmente necessario).

Come esempio finale, si consideri un utente con una scheda 3c503 all'indirizzo `0x350` e una SMC Elite16 (wd8013) a `0x280`. Si avrà la seguente configurazione:

```
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'indirizzo I/O di 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` risponde con la versione del kernel (es. 2.0.34). Si controlli detta directory per vedere quale modulo corrisponda alla propria scheda. Una volta che si hanno le impostazioni corrette nel proprio file `modules.conf`, si può collaudare il tutto con:

```
modprobe ethN
dmesg | tail
```

dove 'N' è il numero dell'interfaccia Ethernet che si sta collaudando. Si noti che il nome dell'interfaccia (`ethX`) asssegnato al driver dal kernel è indipendente dal nome usato sulla riga di alias. Per ulteriori dettagli si faccia riferimento alla sezione 8.3 (Usare un driver Ethernet come modulo).

2.4.2 Con il Driver Compilato nel Kernel

Dato che il rilevamento automatico (probing, N.d.T.) di una scheda ISA può mandare in crash la macchina, i kernel fino al 2.4 incluso, effettuano per default la ricerca automatica di *una sola* scheda Ethernet su bus ISA. Dato che non vi sono più distribuzioni con dei kernel contenenti un gran numero di driver modulari ISA, tale restrizione non viene più imposta a partire dal kernel 2.6.

Nei kernel della serie 2.2 (e più recenti), i processi di rilievo automatico al boot sono stati separati in sicuri ed insicuri, in modo che tutti quelli sicuri (ad esempio PCI e EISA) trovino automaticamente tutte le loro schede. In sistemi con più di una scheda Ethernet di cui almeno una su bus ISA, è ancora necessario fare una delle cose seguenti.

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 dei parametri al kernel, solitamente usando LILO. Il rilevamento della seconda scheda può essere ottenuto con un semplice parametro di boot come `ether=0,0,eth1`. In questo caso `eth0` e `eth1` saranno assegnate nell'ordine in cui vengono rilevate le schede durante il boot. Nel caso in cui, per esempio, si voglia che la scheda a `0x300` sia `eth0` e quella a `0x280` sia `eth1`, allora si può usare:

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

Il parametro `ether=` è in grado di accettare più della terna composta da IRQ, indirizzo I/O di base e nome appena illustrata. Si veda la sezione 8.2 (Passare argomenti Ethernet...) per la sintassi completa, i parametri specifici delle schede ed alcune dritte su 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 effettuare la ricerca automatica per quel dispositivo – rimpiazzandola con uno zero si abiliterà l'autorilevamento di quel dispositivo.

2.5 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 driver in forma modulare, perciò il comando `ether=` viene usato di rado (parti della documentazione sono ancora state aggiornate per riportare questo cambiamento). Se si vogliono usare delle 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 installare il file di configurazione aggiornato.

2.6 Problemi con schede NE1000/NE2000 (e cloni)

Problema: Una scheda PCI compatibile con NE2000 non viene rilevata all'avvio del sistema usando una versione del kernel 2.0.x.

Causa: Il driver `ne.c`, fino alla versione 2.0.30 del kernel, riconosce solo il numero identificativo PCI delle schede compatibili basate su RealTek 8029. Da allora, molti altri produttori hanno distribuito schede PCI compatibili con la NE2000 corredate di altri numeri identificativi PCI che il driver riconosce.

Soluzione: La soluzione più semplice consiste nell'aggiornare il kernel di Linux alla versione 2.0.31 (o più recente). Questa versione del kernel è al corrente dei numeri identificativi di circa cinque chip diversi PCI/NE2000 e li rileva automaticamente all'avvio o nella fase di caricamento del modulo. Inoltre, se si aggiorna il kernel 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 PCI compatibile con NE2000 viene identificata come una ne1000 (a 8 bit!) all'avvio o quando si carica il modulo ne.o per la versione 2.0.x del kernel, e di conseguenza 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 esse siano schede NE1000.

Soluzione: È necessario aggiornare il kernel alla versione 2.0.31 (o più recente) come descritto in precedenza. La nuova versione del driver è stata aggiornata per tener conto di questo problema hardware.

Problema: Una scheda PCI compatibile NE2000 presenta prestazioni orribili, anche se si riduce la dimensione della window come descritto nella sezione 3 (Suggerimenti per le prestazioni).

Causa: Le specifiche per il chip 8390 originale, progettato e commercializzato più di dieci anni fa, indicavano che per ottenere la massima affidabilità, era necessaria una lettura fittizia dal chip prima di ogni operazione di scrittura. Il driver è in grado di far questo ma tale funzionalità è stata disabilitata per default sin dai tempi dei kernel 1.2. Un utente ha riferito che riabilitare questa 'mis-feature' ha migliorato le prestazioni ottenute con un clone economico NE2000 su bus PCI.

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` riporta messaggi di errore del tipo `timeout waiting for Tx RDC` con una scheda compatibile NE2000 su PCI e non funziona correttamente.

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 da 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 siano 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 viene rilevata.

Causa: Le specifiche originarie NE2000 (e perciò il driver per Linux NE2000 un tempo incluso con il kernel) non supportano il Plug and Play.

Soluzione: Si installi la versione 2.4 del kernel (o successive), che include un driver NE2000 con supporto PnP, oppure si usi il disco di configurazione DOS fornito con la scheda stessa per disabilitare PnP e per assegnare la scheda ad uno specifico indirizzo I/O e IRQ. Si aggiunga una riga a `/etc/modules.conf` 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.

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

Causa: Ciò è collegato alla modifica appena menzionata. Dopo la verifica iniziale che un 8390 è all'indirizzo di I/O rilevato, si effettua la re inizializzazione (reset) della scheda. Quando la scheda ha completato tale operazione, si suppone che essa confermi che il reset è stato completato. La vostra scheda non si comporta così è il driver di conseguenza 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`. 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 essere configurata nel modo seguente:

```
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 usare l'opzione `bad=0xbad` nella stessa maniera in cui si indica 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 a quella corrente. Sembra confinato a poche schede clone configurabili in software. Apparentemente queste schede si 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 presente) a `0x370` o il controller secondario IDE a `0x376-0x377`. Se la/le porta/e sono già assegnate ad un altro driver, il kernel non consente al driver di tentare il rilevamento.

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

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

Causa: Il problema è lo stesso appena esaminato, ma su di un kernel più vecchio che non verifica la sovrapposizione delle regioni di I/O. Si usi la stessa soluzione vista prima e ancor meglio si installi un nuovo kernel.

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

Causa: Prima di tutto, c'è 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 signature della scheda, sempre che si fornisca anche un valore per l'indirizzo I/O base diverso da zero. Questa soluzione non richiede la ricompilazione 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: "Galleria della Vergogna") intorno alla riga 42. Questo elenco è usato per rilevare i 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 invece 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 come `0x340`. In alternativa si può usare il parametro 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 tentando l'autorilevamento nel suo spazio di I/O. 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 in precedenza, 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 che si è appena visto.

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. Il dimenticarsi di eseguire lilo, ecc. può portare all'avviamento

del vecchio kernel (si guardi attentamente l'ora e la data di compilazione riportati all'avvio). È un errore banale, ma lo abbiamo compiuto tutti in passato. Ci si assicuri che il driver sia effettivamente incluso nel nuovo kernel cercando nel file `System.map` una voce come `ne_probe`.

2) Si controllino attentamente i messaggi di avvio. Davvero non si accenna mai al fatto che si sta facendo un rilevamento ne2k, ad esempio 'NE*000 probe at 0xNNN: not found (bla bla bla)' o fallisce proprio silenziosamente? C'è una grossa differenza tra i due casi. 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 quello del 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) In maniera simile a quanto appena menzionato, si controlli `/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 dal fallimento silenzioso del driver, allora lo si modifichi aggiungendo alcune chiamate a printk() alla procedura di rilevamento. Per esempio nel driver ne2k si potrebbero aggiungere/rimuovere righe (contrassegnate rispettivamente con un '+' o '-') in `linux/drivers/net/ne.c` come dal seguente esempio:

```
        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;
+        }
```

La procedura di rilevamento produrrà ora messaggi di output per ogni indirizzo di porta che esamina e si potrà capire se l'indirizzo della propria scheda è stato rilevato o meno.

6) Ci si può anche procurare il diagnostico ne2k nel sito ftp di Don (già 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 in 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 simile al seguente:

```
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` compaiono poiché quello è il valore restituito quando si legge una porta di I/O libera. Se qualche altro tipo di hardware si trova nella regione rilevata, si possono vedere dei valori diversi da `0xff`.

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 fornito con la scheda. Può essere che eseguano 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.

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

Problema: Si ricevono messaggi come 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 incluso con il kernel a partire dalla versione 2.0 supporta anche la modalità ad I/O programmato. Si aggiorni alla versione 2.0 (o più recente) del kernel.

Problema: Le vecchie wd8003 e/o le wd8013 configurabili con jumpers hanno 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 sballati 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.

2.8 Problemi con le schede 3Com

Problema: La 3c503 si sceglie l'IRQ N, ma questo serve ad un qualche altro dispositivo che ha bisogno dello stesso IRQ (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 svariate cose, incluso il valore dell'IRQ.

Ciò che segue imposta IRQ 9, indirizzo I/O di 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 fa parte del precedente insieme, si otterrà il messaggio di errore suindicato. 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 al momento del caricamento 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 se invece si carica il driver come un modulo è sufficiente aggiungere `xcvr=1` alla riga delle opzioni del modulo insieme con i propri valori di I/O e IRQ.

2.9 FAQ non specifiche ad una particolare scheda

2.9.1 Linux e schede Ethernet ISA di tipo Plug and Play

Per ottenere i migliori risultati (e la minor scocciatura)) si raccomanda di usare il programma (di solito DOS) fornito con la propria scheda per disabilitare il meccanismo ISA-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/modules.conf`. Può darsi che si debba anche entrare nel BIOS/CMOS setup e contrassegnare l'IRQ come Legacy-ISA al posto di ISA-PnP (se il proprio computer ha questa opzione).

Si noti che non è necessario avere il 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 pussono anche scaricare gratuitamente OpenDOS e FreeDOS.

Se si necessita per compatibilità con un altro sistema operativo che sia abilitato il meccanismo ISA-PnP, allora dovrete prendere provvedimenti diversi a seconda della versione del kernel che state usando. Con i kernel 2.2.x e precedenti si dovrà usare il pacchetto isapnptools che provvederà a 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=`.

Per i kernel 2.4.x e seguenti, supporto per il meccanismo ISA-PnP può essere compilato direttamente nel kernel e se il vostro driver fa uso di questi meccanismi, la vostra scheda sarà configurata con un indirizzo I/O di base ed una linea IRQ disponibili senza bisogno di nessun intervento da parte vostra. Si *eviti* di usare le applicazioni del pacchetto isapnptools ed il supporto per ISA-PnP contenuto nel kernel allo stesso tempo - sarebbe una pessima idea!

Alcuni sistemi possiedono una opzione BIOS 'enable PnP OS' (or similare), ma questa opzione non ha nulla a che vedere con l'hardware ISA-PnP. Si legga di seguito se si desidera avere maggiori dettagli in merito.

2.9.2 un sistema PCI rileva la scheda ma il driver non riesce ad autoconfigurarsi (PnP OS)

Alcuni BIOS PCI potrebbero non abilitare tutte le schede PCI all'accensione, specialmente se è attivata l'opzione PNP OS del BIOS. Questa cosiddetta opzione è stata creata per supportare la versione corrente di Windows che utilizza ancora alcuni driver in real mode. Si disabiliti questa opzione oppure si provi ad aggiornare il sistema con un driver più recente che abbia la capacità di abilitare una scheda disabilitata.

Si noti che i kernel della serie 2.4 supportano meglio l'uso di questa opzione - nello specifico, voi dovreste poter usare questa opzione ed il kernel o i driver dovrebbero essere in grado di abilitare le schede da soli.

2.9.3 In un sistema PCI, tutte le schede vengono rilevate ma due non funzionano

La prima versione della specifica PCI permetteva ad alcuni slot di essere designati come bus master mentre altri (non bus master) venivano designati come slave. Per evitare i problemi causati da utenti che mettevano schede bus master in slot slave, la seconda versione della specifica indicava che tutti gli slot devono essere in grado di operare come bus master. Ciononostante, la maggior parte dei chipset PCI possiedono solo quattro pin BM, per cui se voi avete una scheda madre con 5 slot, e più che probabile che due slot condividano lo stesso pin BM! Questo permette alla scheda madre di soddisfare i requisiti della seconda specifica (ma certo non l'intento dei suoi autori). Per cui, se avete parecchie schede e due di loro si rifiutano di funzionare, è possibile che stiano condividendo lo stesso pin BM.

2.9.4 Il mio sistema ha `/etc/conf.modules` e non `/etc/modules.conf`.

Distribuzioni più stagionate avranno ancora `conf.modules` e non `modules.conf`, che è un nome più sensato. Le più recenti utilità si aspettano il nuovo nome, un fatto che dovete tener presente quando aggiornate un vecchio sistema.

2.9.5 La scheda Ethernet non viene rilevata all'avvio

Di solito la causa di questo è che si sta utilizzando un kernel che non include il supporto specifico per la propria scheda. Per un kernel modulare ciò tipicamente significa che non si è richiesto il caricamento del modulo di cui si ha bisogno.

Se si sta usando un kernel modulare come quelli installati dalla maggior parte delle distribuzioni Linux, allora si provi ad usare l'utilità di configurazione della distribuzione per selezionare il modulo/driver della propria scheda. Per le schede ISA è buona norma determinare l'indirizzo I/O della scheda e aggiungerlo come opzione (ad esempio `io=0x340`) se l'utilità di configurazione permette di indicare delle opzioni. Se la vostra distribuzione non include alcuna utilità di configurazione allora si dovrà aggiungere il nome corretto del modulo (e le sue opzioni) al file `/etc/modules.conf` – si veda `man modprobe` per maggiori dettagli.

Un'altra causa comune è la presenza di un altro dispositivo che utilizza 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 richiederà la regione `0x300-0x31f`. Se un qualsiasi altro dispositivo ha registrato anche solo una porta all'interno di quell'intervallo, il rilevamento non avrà luogo a quell'indirizzo e il driver continuerà silenziosamente con il prossimo indirizzo fra quelli da testare. Quindi, a boot completato, si faccia un `cat /proc/ioports` per verificare che l'intero spazio d'I/O che la scheda richiede sia libero.

Un altro tipo di problema consiste nell'avere la propria scheda impostata dai ponticelli a un indirizzo di I/O che non viene controllato per default. L'elenco degli indirizzi controllati da ogni driver è facilmente reperibile giusto dopo i commenti iniziali nel sorgente del driver stesso. Anche se l'impostazione I/O della propria scheda non è nell'elenco degli indirizzi controllati, la si può fornire al kernel durante il boot (per i driver compilati nel kernel) con il parametro `ether=` come descritto in 8.2 (Passare argomenti Ethernet...). I driver modulari possono fare uso dell'opzione `io=` in `/etc/modules.conf` per specificare un indirizzo che non è testato per default.

2.9.6 Il driver dichiara `unresolved symbol ei_open` e non viene caricato

State utilizzando una delle molte schede basate sul chip 8390 (o un suo clone). Il driver per tali schede è composto da due parti - la parte che voi avete cercato di caricare senza successo (come ad esempio `ne2k-pci.o, ne.o, wd.o, smc-ultra.o`), e la parte dedicata al chip 8390. Questi driver vengono marcati con *(+8390)* accanto al nome del modulo nella lista di informazioni specifiche per ogni produttore (4 (Informazioni specifiche su...)).

Si deve fare in modo che il modulo `8390.o` venga caricato *prima* del caricamento della seconda meta del driver, che in questo modo avrà a sua disposizione tutte le funzioni necessarie.

Cause possibili includono: (1)aver dimenticato di eseguire depmod dopo aver installato un nuovo kernel e relativi moduli cosicché le varie relazioni di dipendenza tra moduli come questa vengano gestite automaticamente; (2)l'uso di `insmod` al posto di `modprobe`, dato che insmod non controlla se esistono dei vincoli di ordine nel caricamento dei moduli; (3) il fatto che il modulo `8390.o` non sia nella sua directory accanto all'altra metà del driver come dovrebbe.

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

No, non lo fa, lo si sta solamente interpretando in modo sbagliato. Questo *non* è un bug e i numeri riportati sono corretti. Si da il caso che in alcune schede basate su 8390 (wd80x3, smc-ultra, ecc.) il chip 8390 in realtà risieda ad 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 usi `cat /proc/ioports` che risponderà con i numeri che ci si aspettava.

2.9.8 Le schede ISA a memoria condivisa non funzionano in un sistema PCI (`0xffff`)

Questo problema solitamente si presenta come una lunga serie di letture di valori `0xffff`. Nessun tipo di scheda a memoria condivisa potrà mai funzionare in un sistema PCI se il BIOS ed i valori nella CMOS non sono stati prima settati correttamente. Il BIOS deve essere impostato per permettere l'accesso in memoria condivisa dal bus ISA alla regione di memoria che la propria scheda sta cercando di usare. Se non si sa quali impostazioni siano appropriate allora si chieda al proprio fornitore o al locale esperto di computer. Nei i BIOS AMI c'è solitamente una sezione Plug and Play dove si trovano impostazioni come ISA Shared Memory Size e ISA Shared Memory Base. Per schede come la wd8013 e la SMC Ultra, si modifichi la dimensione predefinita da Disabled a 16kB e si indichi l'indirizzo della memoria condivisa della propria scheda come indirizzo di base.

2.9.9 Sembra che la scheda invii dati ma non riceve niente

Si faccia un `cat /proc/interrupts`, il totale aggiornato del numero di eventi di interrupt generati dalla propria scheda sarà incluso nell'output. Se tale numero 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 (indipendentemente dal fatto che l'altro dispositivo abbia o meno un driver installato/disponibile). Si cambi l'IRQ di uno dei due dispositivi con un IRQ ancora libero.

2.9.10 Supporto per Asynchronous Transfer Mode (ATM)

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

Werner dice 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/`

2.9.11 Supporto per Gigabit Ethernet

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

Si, al momento c'è ne sono diversi. Uno dei principali sviluppatori di Linux in ambito networking ha recentemente valutato le prestazioni delle schede dotate di driver Linux come segue: 1) Intel E1000, 2) Tigon/Acenic, 3) SysKonnect sk-98xx, 4) Tigon3/bcm57xx. Questo era lo stato delle cose nel marzo del 2002 ed è ovviamente una situazione in evoluzione.

2.9.12 Supporto FDDI

C'è supporto per FDDI sotto Linux?

Sì. Larry Stefani ha scritto un driver per il kernel 2.0 usando le schede DEFEA (FDDI EISA) e DEFPA (FDDI PCI) della Digital che è stato incluso nel kernel 2.0.24. Al momento attuale non sono supportate altre schede.

2.9.13 Supporto Full Duplex

Il Full Duplex mi darà 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, esso può mantenere la connessione occupata in entrambe le direzioni. Non esistono cose come 10BASE-2 o 10BASE-5 full duplex. Il full duplex funziona disabilitando il rilevamento delle collisione nell'adattatore, e questo è il motivo per cui non lo si può fare con un cavo coassiale; la LAN in questo modo non funzionerebbe. Lo standard 10BASE-T (interfaccia RJ45) usa cavi separati per la trasmissione e la ricezione ed è così possibile utilizzare 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 non ci si deve aspettare un incremento delle prestazioni di un fattore 2. Che il full duplex sia o meno supportato, la cosa dipende dalla scheda e forse anche dal driver. Alcune schede possono fare 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 utenti che facciano serio uso della rete noteranno la differenza tra le due modalità.

2.9.14 Schede Ethernet per Linux su macchine SMP

Se si è fatta la spesa per un computer multi processore (MP) allora si compri anche una buona scheda Ethernet. Per i kernel 2.0 questo non è veramente necessario, ma lo è sicuramente per quelli 2.2. La maggior parte delle schede più vecchie non intelligenti (vale a dire disegnate per bus ISA in PIO e memoria condivisa) non furono mai pensate considerando in alcun modo l'uso con macchina multiprocessore. L'executive summary è di comperare una scheda intelligente di progettazione moderna e assicurarsi che il driver sia stato scritto (o aggiornato) per gestire il funzionamento in contesto MP (le parole chiave qui 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 operare in contesto MP. Si legga di seguito per i dettagli completi sul perché di dovrebbe acquistare una buona scheda per l'uso in ambiente MP (e cosa accade se non lo si fa).

Nei kernel 2.0, solo un processore alla volta poteva entrare 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 attorno a tutto il kernel che permetteva l'accesso ad un solo processore alla volta. In questo modo si sa che non si avranno mai due processori che provano a cambiare la stessa cosa allo stesso tempo!).

Lo svantaggio nel permettere ad un solo processore alla volta di entrare in modalità kernel è che si ottengono delle prestazioni degne di una macchina MP solamente se si eseguono programmi indipendenti e che fanno gran uso di operazioni di calcolo. Se i programmi fanno un sacco di input/output (I/O) come ad esempio il leggere e scrivere 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 degradano rapidamente verso quelle di una macchina a singolo processore.

Poiché questa situazione è chiaramente inferiore al caso ideale (specialmente per server di file/WWW, router, e così via) i kernel 2.2 hanno un lock più granulare. Ciò significa che più di un processore alla volta può operare 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 del disco.

Con tutto questo bene in mente, ecco le insidie: il locking più granulare 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 statistiche di funzionamento 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 sono ricevuti i dati al posto delle statistiche. La scheda è stata un po' confusa dalla richiesta di fare due (o più!) cose allo stesso tempo ed è anche probabile che mandi in crash la macchina mentre ci prova.

Quindi il driver che funzionava benissimo su di una macchina con CPU singola non è più abbastanza buono e necessita di essere aggiornato con dei lock che controllino l'accesso alla scheda cosicché i tre processi di ricezione, trasmissione e manipolazione dei dati di configurazione siano serializzati sufficentemente da poter permettere alla scheda di funzionare stabilmente. La faccenda preoccupante è che un driver non ancora aggiornato con lock per operazioni stabili in contesto MP sembrerà probabilmente funzionare su macchine MP in presenza di un leggero carico di rete, ma provocherà il crash della macchina o come minimo esibirà uno strano comportamento quando due (o più!) processori proveranno a fare più di una di queste tre operazioni nello stesso momento.

Un driver Ethernet aggiornato per uso su di macchine MP richiederà (come minimo) un lock attorno al driver che limiti l'accesso ai punti d'ingresso dal kernel nel driver alla maniera di uno alla volta, prego. Con tale modifica, le operazioni saranno serializzate cosicché l'hardware sottostante dovrebbe venire trattato come se fosse usato in una macchina UP, e quindi dovrebbe essere stabile. Lo svantaggio di questa soluzione 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 (anche se 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 ottenuti in due modi. Si può provare a minimizzare il tempo che intercorre tra quando si richiede il lock e quando lo si rilascia, e/o si può implementare un locking 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).

tuttavia, per le più vecchie schede non intelligenti che non sono mai state progettate pensando all'uso in ambito MP, potrebbe non essere possibile realizzare nessuno di questi miglioramenti.

Ancora più è il fatto che le schede non intelligenti tipicamente richiedano che il processore sposti dati tra la scheda e la memoria del computer, per cui nel caso peggiore il lock sarà mantenuto per tutto il tempo necessario a spostare ciascun pacchetto da 1.5kB sul bus ISA.

Le più moderne schede intelligenti tipicamente spostano i dati direttamente da e per la memoria del computer senza nessun aiuto da parte del un processore. Questo è un grande vantaggio, 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 a richiedere meno spesso un unico grosso lock attorno all'intero driver.

2.9.15 Schede Ethernet per Linux su piattaforma Alpha/AXP e bus PCI

Dalla versione 2.0 del kernel, i soli 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 elencati nella pagina Web di Donald poiché sono stati scritti con la portabilità tra diverse architetture in mente.

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 tutte le dereferenziazioni di puntatori nella memoria di I/O (da 640k a 1MB) con le appropriate chiamate readb() writeb() readl() writel(), come mostrato nel seguente 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` incluso con kernel recenti.

2.9.16 Ethernet per Linux su hardware SUN/Sparc

Per le informazioni più aggiornate sull'hardware Sparc, si visiti il seguente sito:

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

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

I problemi riguardati il porting dei driver PCI su piattaforma Sparc sono simili a quelli citati in precedenza per la piattaforma AXP. Inoltre ci possono essere un po' di questione relative all'ordine dei byte, poiché gli Sparc sono big endian mentre gli AXP e gli ix86 sono little endian.

2.9.17 Ethernet per Linux su altro hardware

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

2.9.18 Connettere 10 o 100 BaseT senza un hub

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

Senza altri dispositivi o marchingegni si possono collegare 2 macchine (ma non di più) usando un cavo crossover. Tuttavia, alcune delle recenti schede autonegozianti con più fronzoli potrebbero non riuscire a raccapezzarsi in un tale ambiente. E *no*, non si può metter su un hub semplicemente incrociando un po' di fili assieme. È praticamente impossibile produrre bene il segnale di collisione senza creare un hub a tutti gli effetti.

2.9.19 SIOCSIFxxx: No such device

Ricevo un sacco di messaggi 'SIOCSIFxxx: No such device' durante il 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 al caricamento del modulo e quando vengono eseguiti `ifconfig` e `route` essi non trovano nessun dispositivo con cui operare. Si usi `dmesg | more` per rivedere i messaggi di boot e vedere se c'è un qualche messaggio riguardante il rilevamento della scheda Ethernet.

2.9.20 SIOCSFFLAGS: Try again

Ottengo il messaggio di errore 'SIOCSFFLAGS: Try again' quando eseguo 'ifconfig'. Eh?

Qualche altro dispositivo si è preso l'IRQ che la propria scheda Ethernet vuole 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 finito. Esempi di questo sono alcuni schede audio, porte seriali, driver per floppy disk, ecc. Si può usare il comando `cat /proc/interrupts` per vedere quali interrupt sono al momento *in uso*. La maggior parte dei driver per schede Ethernet su 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.

2.9.21 Usando 'ifconfig' e Link di tipo UNSPEC con 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 succede 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 quindi non ha importanza pratica 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.

2.9.22 Enorme numero di errori in RX e TX

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

Si guardi di nuovo. Dice `RX packets` *grande numero* **PAUSE** `errors 0` **PAUSE** `dropped 0` **PAUSE** `overrun 0`. Ed è più o meno 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 si è ancora confusi da questa cosa, si provi invece ad usare `cat /proc/net/dev`.

2.9.23 Voci in `/dev/` per le schede Ethernet

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

Contrariamente a ciò che probabilmente vi è stato detto, i file in /dev/* non vengono utilizzati. Si può cancellare qualsiasi `/dev/wd0, /dev/ne0` e simili voci.

2.9.24 Accesso a basso livello al dispositivo Ethernet

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

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

Questo codice vi consente di avere un socket che riceve qualsiasi tipo di protocollo. Si facciano chiamate a `recvfrom()` su 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 dati grezzi attraverso chiamate a `sendto()`. Naturalmente per fare entrambe le cose si deve avere l'accesso come root.

3 Suggerimenti per migliorare le prestazioni

Ecco qua un po' di trucchi da usare se si soffre di problemi di basso throughput Ethernet o per guadagnare un pochino 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 della macchina in trasmissione (si faccia il 'get' almeno due volte, poiché la prima volta si caricherà la cache del buffer nella macchina in 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 al disco.

3.1 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 dove mettere i nuovi pacchetti.

In questo caso o si perdono (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 ritrasmissione e degradando seriamente le prestazioni fino ad un fattore di 5!

Le schede con a bordo più memoria sono in grado di bufferizzare più pacchetti e quindi gestire grossi picchi di traffico (burst) back-to-back senza perdere nessun pacchetto. D'altra canto questo significa che la scheda non necessita di una così bassa latenza nell'estrazione dei pacchetti dal buffer da parte del computer 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 per Linux riserva 3kB del buffer per due buffer di trasmissione, quindi nel caso di una scheda a 8 bit rimangono solo 5kB di spazio per la ricezione. Questo è spazio sufficiente solo per tre pacchetti Ethernet a dimensione massima (1500 byte).

3.2 Schede ISA e velocità del bus ISA

Come menzionato in precedenza, se i pacchetti vengono rimossi abbastanza velocemente dalla scheda allora la condizione di drop/overrun non si verificherà anche se la dimensione del buffer per i pacchetti in arrivo è ridotta. Il fattore che determina la frequenza con la quale i pacchetti vengono rimossi dalla scheda e messi nella memoria del computer corrisponde alla velocità del percorso dati che collega le due, ovverosia la velocità del bus ISA (ma 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 setup della CMOS, 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 vincolati dalle 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 bloccata 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.

3.3 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 dei 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 o gateway sono abbastanza 'bufferizzati' da non avere 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 possono verificare le condizioni operative con `cat /proc/net/dev`, che elencherà il numero di drop/overrun verificatisi.

3.4 Incrementare le prestazioni NFS

Alcuni utenti hanno osservato che l'uso di una scheda a 8 bit con client NFS causa prestazioni peggiori di quanto previsto quando vengono usati pacchetti di 8kB (la dimensione nativa Sun).

Una possibile causa di questo potrebbe essere la differente dimensione del buffer a bordo tra schede a 8 e 16 bit. La dimensione massima di un pacchetto Ethernet è di approssimativamente 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. Il fatto che le schede a 8 bit necessitano di un ulteriore ciclo di bus ISA per ogni trasferimento non 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 svuotato più velocemente. Ho scoperto che una vecchia scheda WD8003E a 8MHz (senza altro carico di sistema) può sostenere un traffico notevolei 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 un 1kB per i pacchetti NFS e si usa come minimo una scheda ISA a 16 bit, si riscontrerà un incremento significativo nel passare a pacchetti NFS di 4kB (o addirittura 8kB).

4 Informazioni specifiche su produttori e modelli

Di seguito sono elencate molte schede in ordine alfabetico secondo il nome del produttore e il nome/sigla identificativa del prodotto. Accanto all'identificativo di ciascuna scheda, si vedrà indicato Supportata, Semi supportata, Obsoleta, Rimossa oppure Non supportata.

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

Semi supportata significa che esiste un driver, ma si verifica almeno una delle condizioni seguenti: (1) Il driver e/o l'hardware ha dei bug che possono provocare prestazioni scadenti, perdita di connessione e persino crash del sistema; (2) Il driver è nuovo e la scheda è poco comune e quindi il driver non è stato collaudato molto e il suo autore ha quindi ricevuto poco feedback sul suo funzionamento. Ovviamente la condizione (2) è preferibile alla (1) e la descrizione di ciascuna scheda/driver dovrebbe chiarire quale delle due condizioni sia applicabile. 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` per poter poi utilizzare questi driver.

Obsoleta significa che un driver esiste e che la scheda era probabilmente considerata una volta come semi-supportata. Tuttavia, a causa di mancanza di interesse, utenti e supporto, il driver è risaputo essere non più funzionante. Il driver stesso viene ancora incluso nel kernel, ma è disabilitato nel menu delle opzioni di configurazione. In generale il piano è che se un driver marcato come obsoleto non viene aggiornato durante il seguente ciclo di sviluppo del kernel, esso verrà rimosso completamente. Solitamente un driver indicato come obsoleto richiede solamente una rinfrescata per adattarsi a cambiamenti introdotti nell'interfaccia per driver del kernel o ad altri cambiamenti simili nelle API del kernel.

Rimossa significa che un driver che era ad un tempo obsoleto (vedi sopra) è stato rimosso dal sorgente per il kernel corrente a causa della mancanza generale di interesse nell'aggiornarlo. Nulla impedisce a qualcuno di recuperare il sorgente per il driver da una versione precedente del kernel e, dopo aver provveduto ai necessari cambiamenti, usarlo.

Non supportata significa invece che attualmente non è disponibile alcun driver per quella scheda. Ciò può essere dovuto alla mancanza di interesse in hardware che è 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. Quindi ci si consideri cautelati perché può succedere che si trovi una scheda segnata come semi-supportata che nel caso in questione funziona perfettamente (il che è ottima cosa) esattamente come si può incappare in una scheda indicata come supportata che in realtà da origine ad una serie infinita di problemi (il che non è proprio un bene) sulla vostra macchina.

Dopo lo stato, è elencato il nome che il driver ha nel kernel di Linux. Questo è anche il nome del modulo del driver che andrebbe eventualmente utilizzato nella voce `alias eth0 nome_driver` nel file di configurazione dei moduli `/etc/modules.conf`.

4.1 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 dall'assembly number della scheda. La 3Com ha un documento 'Identifying 3Com Adapters By Assembly Number' (ref 24500002) che molto probabilmente farà al vostro caso per chiarire le cose. Si controlli anche i siti Web ed FTP `www.3Com.com` di 3com per informazioni e strumenti che possano risultavi utili (ivi inclusa la documentazione tecnica sulle schede rilasciata in formato PDF).

4.1.1 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 utilizzata, 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, come non può ricevere un pacchetto mentre sta caricando un pacchetto da trasmettere. Questo non era un problema in reti composte da due 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, il DMA non viene usato, l'autoprobe ricerca solo agli indirizzi `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, dato che ci si inchioderà mentre si ascoltano *tutti* i pacchetti multicast. Si vedano i commenti in cima al codice sorgente per ulteriori dettagli.

4.1.2 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 su disable (disabilitato) 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 ad I/O programmato a meno che non se ne abbia bisogno per compatibilità con un altro sistema operativo installato sullo stesso computer.

La linea IRQ della 3c503 è impostata dal software, con nessun aiuto da parte della EEPROM. Diversamente dal driver MS-DOS, quello per Linux ha funzionalità di autoIRQ e adotta la prima linea IRQ libera nell'insieme {5,2/9,3,4}, selezionandola ogni volta che la scheda è configurata con ifconfig. 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 nella sezione 2.8 (Problemi con...).

Se si intende usare questo driver come modulo caricabile probabilmente si dovrebbe fare riferimento alla sezione 8.3 (Usare un driver Ethernet come modulo) per informazioni specifiche sui moduli.

4.1.3 Etherlink Plus 3c505

Stato: Semi supportata, Nome del driver: 3c505

Queste schede usano il chip i82586, ma non ci sono poi tante schede di questo tipo in giro. Il driver viene incluso nel kernel standard, ma è dichiarato come driver alpha. Si veda la sezione 2.3 (Driver alpha) per importanti informazioni sull'uso con Linux di driver Ethernet in alpha testing.

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

4.1.4 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 alpha. Si veda la sezione 2.3 (Driver alpha) per importanti informazioni sull'uso con Linux di driver Ethernet in alpha testing.

4.1.5 Etherlink III, 3c509 / 3c509B

Stato: Supportata, Nome del driver: 3c509

Questa scheda era piuttosto economica e aveva buone prestazioni per essere una ISA senza bus-master. Gli svantaggi erano che la 3c509 originale richiedeva 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 trasferimenti in PIO mode, analogamente a una scheda ne2000 e quindi, in confronto, una scheda a memoria condivisa come una wd8013 sarà più efficiente.

La 3c509 originale aveva un buffer per pacchetti troppo ridotto (4kB totali, 2kB in ricezione, 2k in trasmissione), cosicché qualche volta il driver perdeva dei pacchetti se gli interrupt venivano mascherati troppo a lungo. Per minimizzare questo problema, si può provare a non mascherare gli interrupt durante i trasferimenti dai 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 di memoria e il buffer può essere diviso in 4/4, 5/3 o 6/2 per ricezione e trasmissione. Questa impostazione è modificabile con l'utilità di configurazione DOS ed è salvata nella EEPROM. Questo dovrebbe alleviare il problema della 3c509 originale.

Gli utilizzatori della 3c509B dovrebbero usare l'utilità DOS fornita con la scheda 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 l'autodetect dell'uscita utilizzata, quindi si *deve* selezionare 10Base-T, 10Base-2 oppure AUI. Si noti che se si disabilita completamente il PnP, si dovrebbe uscire dall'utilità di configurazione e far seguire un hard reset della macchina per assicurarsi che i nuovi settaggi abbiano avuto effetto.

Alcuni chiedono delucidazioni sulle impostazioni Server or Workstation e Highest Modem Speed presenti nella utilità di configurazione sotto DOS. Questi settaggi in effetti non cambiano nulla in hardware e sono solo dei consigli per il driver DOS. Il driver di Linux non ha bisogno ne fa uso questi parametri. Infine, non si abiliti la modalita EISA su questa scheda ISA a meno che non si disponga effettivamente di una macchina EISA, o si potrebbe finire a cercare di recuperare una macchina EISA solo per poter riportare la scheda in modalita ISA!

La scheda con il più basso indirizzo Ethernet hardware alla fin fine sarà *sempre* la `eth0` in una configurazione con più 3c509. Questa cosa non dovrebbe preoccupare nessuno, tranne coloro che vogliano assegnare un indirizzo hardware da 6 byte a una particolare interfaccia.

Se quest'ultima cosa vi preoccupa veramente, si dia un'occhiata all'ultimo driver di Donald, in quanto si può essere in grado di usare il valore `0x3c509` nei campi di indirizzo di memoria non utilizzati per ordinare il rilevamento ed adattarlo alle proprie particolari necessità.

4.1.6 3c515

Stato: Supportata, Nome del driver: 3c515

Questa scheda, nome in codice CorkScrew, è la scheda a 100Mbps ISA offerta dalla 3COM, ma si noti che non è possibile raggiungere la piena velocità di 100Mbps su di un bus ISA.

4.1.7 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.

4.1.8 3c527 Etherlink MC/32

Stato: Semi supportata, Nome del driver: 3c527

Sì, un'altra scheda MCA basata sul chip i82586. No, non riscuote molto interesse. Se si è impelagati con l'MCA si avrà probabilmente maggior fortuna con la 3c529, dato che usa il collaudato core 3c509.

4.1.9 3c529

Stato: Supportata, Nome del driver: 3c509

Questa scheda in realtà usa lo stesso chipset della 3c509. Degli utenti hanno effettivamente utilizzato questa scheda su macchine MCA.

4.1.10 3c556

Stato: Supportata, Nome del driver: 3c59x

Una interfaccia mini PCI usata in parecchi laptop IBM e HP, altresì conosciuta come un 'laptop tornado'.

4.1.11 3c562

Stato: Supportata, Nome del driver: 3c589_cs

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 Linux condividano lo stesso interrupt. C'è il supporto per alcuni nuovi registri e un po' per la condivisione degli interrupt hardware.

Grazie ancora a Cameron per aver fornito a David Hinds un campione di questo prodotto e la relativa documentazione.

4.1.12 3c575

Stato: Supportata, Nome del driver: 3c59x

Si noti che per supportare questo dispositivo cardbus in vecchi kernel 2.2 si doveva usare 3c575_cb.c, che poteva essere recuperato nel package pcmcia_cs.

4.1.13 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 proprio impressionante. 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.

4.1.14 3c589 / 3c589B

Stato: Semi-supportata, Nome del driver: 3c589_cs

Molti utenti stanno usando questa scheda PCMCIA da un po' di tempo. La B nel nome significa la stessa cosa che nel caso della 3c509.

4.1.15 3c590 / 3c595

Stato: Supportate, Nome del driver: 3c59x

Queste schede Vortex sono per macchine a bus PCI: la 590 era il prodotto di 3Com a 10Mbps mentre la 595 era il prodotto per il segmento di mercato a 100Mbps. Si noti inoltre che si può usare la 595 come una 590 (cioè in modalità a 10Mbps). La linea 3c59x è stata rimpiazzata dalla 3c9xx da parecchio tempo, e di conseguenza queste schede vengono considerate piuttosto obsolete.

Si noti che in giro si sono due diverse schede 3c590: i primi modelli avevano 32kB di memoria sulla scheda mentre gli ultimi modelli hanno solo 8kB di memoria. La 3c595 ha 64kB, in quanto non si può andare molto lontano con soli 8kB a 100Mbps!

4.1.16 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.

4.1.17 3c900 / 3c905 / 3c905B / 3c905C / 3c905CX

Status: Supportate, Nome del driver Name: 3c59x

Queste schede (altrimenti note come 'Boomerang' o EtherLink III XL) sono state rilasciate per sostituire le schede 3c590/3c595, e supporto aggiuntivo è stato inserito nel driver vortex/3c59x. Il driver incluso in versioni del kernel più vecchie potrebbe non supportare le ultime revisioni di queste schede, in qual caso si potrebbe aver bisogno di driver aggiornati.

Si noti che la 3c905C ha funzionalità di checksum a bordo (in hardware), cosa che riduce il carico di lavoro della CPU.

4.1.18 3c985 (Gigabit acenic, Tigon2)

Stato: Supportata, Nome del driver: acenic

Questo driver supporta diverse altre schede Gigabit oltre al modello 3Com.

4.1.19 3c996 (Gigabit broadcom, Tigon3)

Stato: Supportata, Nome del driver: tg3, bcm5700(vecchio driver)

Questo driver supporta svariate altre schede Gigabit oltre al modello di 3com. Il driver `tg3` è una riscrittura completa a cura di un gruppo di diversi sviluppatori Linux fatta nel tentativo di produrre un driver migliore di quellodel produttore `bcm5700`.

4.2 Accton

4.2.1 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.

4.2.2 Accton EN1203, EN1207, EtherDuo-PCI

Stato: Supportata, Nome del driver: de4x5, tulip o 8139too

Sembra che ci siano state diverse revisioni della EN1207 (da A a D), di cui la A, B e C basate sul chipset tulip e la D basata sul chip Realtek 8139, che richiede quindi un driver diverso.

Come per qualsiasi acquisto, la si dovrebbe provare assicurandosi di poterla restituire nel caso non la possiate usare sul vostro sistema.

4.2.3 Accton EN2209 Parallel Port Adaptor (EtherPocket)

Stato: Semi supportata, Nome del driver: ?

Era disponibile un driver per questi adattatori su porta parallela per i kernel 2.0 e 2.1. La sua ultima locazione conosciuta era:

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

4.2.4 Accton EN2212 PCMCIA Card

Stato: Supportata, Nome del driver: pcnet_cs

4.3 Adaptec

Si noti che le più vecchie schede Adaptec a 32 bit usavano un clone del chipset tulip.

4.3.1 Adaptec DuraLAN/Starfire, 64bit ANA-6922

Stato: Supportata, Nome del driver: starfire

4.4 Allied Telesyn/Telesis

4.4.1 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. Queste 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 nella sezione 4.5.1 (AMD LANCE).

4.4.2 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? durante la compilazione. Ciò è dovuto semplicemente allo poco feedback avuto sulla stabilità del driver a causa della relativa rarità della scheda. Se si hanno problemi con il driver distribuito assieme al kernel, un driver alternativo è 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.

Il chip Fujitsu utilizzato nella AT1700 ha un difetto di progettazione: può essere reinizializzato completamente solo togliendo alimentazione alla macchina. Premendo solamente il pulsante di reset non si inizializza l'interfaccia del bus. Questo non sarebbe un grosso problema, se non fosse per il fatto che la scheda può essere rilevata con certezza solo quando è stata reinizializzata. La soluzione è di spegnere la macchina se il kernel ha problemi a rilevare l'AT1700.

4.4.3 AT2400

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

Ancora un altra scheda compatibile con la NE2000 PCI. Questa è basata sul chip RealTek 8029.

4.4.4 AT2450

Stato: Supportata, Nome del driver: pcnet32

Questa è la versione PCI della AT1500 e non soffre dei problemi che ha la scheda 79c970 PCI della Boca. Informazioni sulla selezione del DMA e la numerazione del chip possono essere trovate nella sezione 4.5.1 (AMD LANCE).

4.4.5 AT2500

Stato: Supportata, Nome del driver: 8139too, rtl8139(vecchio driver)

Questa scheda usa il chip 8139 della RealTek. Si veda la sezione 4.36.6 (RealTek 8139).

4.4.6 AT2540FX

Stato: Semi supportata, Nome del driver: eepro100

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

4.5 AMD / Advanced Micro Devices

Carl Ching dell'AMD è stato talmente gentile da fornire una descrizione molto dettagliata di tutti i prodotti Ethernet dell'AMD rilevanti per questo documento, il che mi hanno aiutato a chiarire un po' questa sezione.

4.5.1 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 sulla 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 4.4.1 (AT1500)) e la NE1500/2100 (si veda la sezione 4.31.4 (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 necessario, il 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 piena 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 apparizione del messaggio 'busmaster arbitration failure'. Questo accade quando il driver LANCE non può accedere al bus dopo che è passato un ragionevole intervallo di tempo (50us). Questo solitamente indica che l'implementazione del bus-master della scheda madre ha dei problemi o che qualche altro dispositivo sta intasando il bus, oppure che c'è un conflitto di canale DMA. Se il proprio BIOS include 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 la scheda solo in questi indirizzi: `0x300, 0x320, 0x340, 0x360` e che qualsiasi indirizzo fornito con l'argomento di boot `ether=` viene ignorato silenziosamente (questo problema verrà corretto). 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, vengono usati dei 'bounce-buffer' in memoria bassa (vale a dire che qualsiasi dato sopra il 16 MB è copiato dentro un buffer sotto i 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 8.2.1 (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 l'eccezione: su questa scheda è facile leggere i valori impostati nella EEPROM per IRQ e DMA.

4.5.2 AMD 79C901 (Home PNA PHY)

Stato: Supportata, Nome del driver: sis900

Il file `sis900.txt` incluso con i kernel 2.4 afferma che la AM79C901 HomePNA PHY non è stato testato per esteso e potrebbero esserci dei driver nel cambio al volo del transceiver, per cui si potrebbe voler controllare questo file se si usa un kernel più recente.

4.5.3 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 associate con il bus ISA.

4.5.4 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 il kernel abilitando il supporto PCI. La 970A aggiunge al progetto originale il supporto full duplex ed 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.

4.5.5 AMD 79C971 (PCnet-FAST)

Stato: Supportata, Nome del driver: pcnet32

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

4.5.6 AMD 79C972 (PCnet-FAST+)

Stato: Supportata, Nome del driver: pcnet32

Si è ricevuto conferma che questa scheda funziona proprio come la 971.

4.5.7 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 quanto sia supportata la parte SCSI del chip: questo è l'*Ethernet-HowTo*, non lo SCSI-HowTo.

4.6 Ansel Communications

4.6.1 AC3200 EISA

Stato: Semi-supportata, Nome del driver: ac3200

Questa scheda EISA è basata sul comune chip 8390 utilizzato anche nelle schede ne200 e nella wd80x3. Si noti che per accedere a questo driver durante il `make config` si deve rispondere 'Y' alla domanda iniziale Prompt for development and/or incomplete code/drivers? durante la compilazione del kernel. Questo è semplicemente dovuto al poco feedback ricevuto dagli utenti sulla stabilità del driver a causa della relativa rarità della scheda. Non si è ricevuto molto riscontro su questo driver nonostante esso sia stato incluso nel kernel a partire dalla versione 1.1.25.

4.7 Apricot

4.7.1 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 collocata solamente all'indirizzo di I/O `0x300` e guardando nei sorgenti del driver, sembra anche che l'IRQ sia fissato in 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.

4.8 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 dotati di schede Arcnet.

Un vantaggio di Arcnet è che tutte le schede hanno la medesima interfaccia cosicché un unico driver funziona per tutte. Queste schede hanno gestione degli errori integrata e quindi si può supporre che non perdano mai pacchetti (ottimo per il traffico UDP!). Si noti che il driver arcnet usa arc0 come suo nome invece del consueto eth0 dei dispositivi Ethernet.

Nel kernel standard ci sono file d'informazione sull'impostazione dei ponticelli, suggerimenti generali e indicazione di dove inviare bug reports.

Sembra che il driver funzioni anche con le schede ARCnet a 100Mbs!

4.9 Boca Research

Sì, non producono solo schede seriali multiporta. :-)

4.9.1 Boca BEN400

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

Apparently this is a NE2000 clone, using a VIA VT86C916 chip.

4.9.2 Boca BEN (ISA, VLB, PCI)

Stato: Supportata, Nome del driver: lance, pcnet32

Questa schede sono basate sui chip PCnet dell'AMD. Molti utenti hanno avuto un sacco di problemi con queste schede VLB/PCI. Sembra che il problema sia dovuto alla mancata installazione di alcuni condensatori nel progetto di Boca che AMD aveva raccomandato (i precedenti modelli ISA non sono affetti da questo problema).

La Boca offriva una 'warranty repair' (ripazione in garanzia) per i possessori di tali schede che consisteva nell'aggiunta di un condensatore, ma sembra che la cosa non abbia funzionato al 100% per la maggior parte delle persone, sebbene abbia aiutato alcuni. Queste schede sono ora così vecchie che non vale la pena di disturbarsi a provare.

Per ulteriori informazioni generali sui chip della AMD si veda la sezione 4.5.1 (AMD LANCE).

4.10 Broadcom

4.10.1 Broadcom Tigon2

Stato: Supportata, Nome del driver: acenic

4.10.2 Broadcom Tigon3

Status: Supportata, Nome del driver: tg3

4.11 Cabletron

Mancato rilascio di informazioni necessarie alla programmazione delle loro schede da parte di Cabletron quando i driver furono sviluppati per queste schede è risultata in un livello di supporto per queste schede inferiore a quello che si sarebbe potuto altrimenti ottenere.

Apparentemente la Cabletron ha cambiato la sua politica riguardo le informazioni per la programmazione (come la Xircom). Comunque, a questo punto, c'è una richiesta minima per driver modificati o aggiornati per le vecchie schede E20xx e E21xx.

4.11.1 E10**, E10**-x, E20**, E20**-x

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

Queste sono praticamente dei cloni del3a serie NEx000 che dovrebbero funzionare con i driver standard NEx000, grazie a una verifica specifica per dispositivi Cabletron durante il rilevamento.

4.11.2 E2100

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

La E2100 è un orrido progetto. Ogniqualvolta mappa la sua memoria condivisa durante un trasferimento di pacchetti, la mappa in *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. La scheda funzionerà senza problemi per maggior parte del tempo, ma una volta ogni tanto il problema vi pungolerà (questo problema può essere evitato disabilitando gli interrupt mentre si trasferiscono i pacchetti, ma quasi certamente si perderanno dei battiti del clock). Inoltre, se si commette un errore programmando la scheda o si ferma la macchina proprio al momento sbagliato, nemmeno il pulsante di reset potrà salvarvi. Si *deve* spegnere la macchina e *lasciarla* spenta per almeno 30 secondi.

La selezione del tipo di cavo è automatica, ma volendo la si può forzare con i bit meno significativi del parametro dev->mem_end. Se veda 8.2.1 (PARAM_2). Gli utilizzatori di driver modulari possono specificare un valore `xcvr=N` come `options` nel file `/etc/modules.conf`.

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

Se si ha intenzione di usare questo driver come modulo caricabile probabilmente si dovrebbe leggere la sezione 8.3 (Usare un driver Ethernet come modulo) per informazioni specifiche sull'uso di driver modulari.

4.11.3 E22**

Stato: Semi supportata, Nome del driver: lance

Secondo le informazioni contenute in un bollettino tecnico della Cabletron, queste schede usano il chip PC-Net dell'AMD (si veda 4.5.1 (AMD PC-Net)) e dovrebbero quindi funzionare con il driver lance generico.

4.12 Cogent

4.12.1 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 deve ancora essere verificata.

4.12.2 Cogent eMASTER+, EM100-PCI, EM400, EM960, EM964

Stato: Supportate, Nome del driver: de4x5, tulip

Queste schede sono un'altra implementazione del DEC 21040 e quindi si spera che funzionino 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 4.18.4 (DEC 21040) per maggiori informazioni su queste schede e la situazione di supporto attuale.

4.13 Compaq

La Compaq non è veramente un costruttore di schede Ethernet, ma un sacco di loro sistemi hanno controller Ethernet integrati nella scheda madre.

4.13.1 Compaq Deskpro / Compaq XL (Embedded AMD Chip)

Stato: Supportati, Nome del driver: pcnet32

Le macchine 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 piazzi 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 viene mai rilevata e non è nemmeno usabile (altro effetto: non funziona nemmeno il mouse). Il modo per aggirare questo problema (come descritto approfonditamente in http://www-c724.uibk.ac.at/XL/) è di avviare il sistema in 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 funzionante che conosco. Il driver semplicemente si limita a spostare la directory PCI nel posto dov'è di solito (e dove Linux la può trovare).»

L'utilità DOS `movepci.exe` dovrebbe essere reperibile nel package di supporto `SP1599.EXE` se si avesse bisgno di una copia.

Altre informazioni a carattere generico sui chip AMD possono essere reperite nella sezione 4.5.1 (AMD LANCE).

4.13.2 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 nella sezione 4.44.1 (ThunderLAN).

4.13.3 Compaq PCI card

Stato: Supportata, nome del driver: eepro100

Si controlli la scheda: se ha numero di parte 323551-821 e/o un chip Intel 82558 allora è un altra scheda basata sull'Intel EEPro100.

4.14 Danpex

4.14.1 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 la sezione 4.18.4 (DEC 21040) per maggiori informazioni su queste schede e la corrente situazione di supporto del driver.

4.15 Davicom

4.15.1 Davicom DM9102

Stato: Supportata, Nome del Driver: tulip, dmfe

Questo è praticamente un clone del chip tulip e di conseguenza è possibile utilizzare per questa scheda il driver tulip o il driver dmfe del produttore. L'approccio comune è di provare prima con il driver tulip, e solo in seguito di provare con il driver dmfe, che sembra essere la scelta migliore solo per schede molto vecchie.

4.16 D-Link

4.16.1 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 sequenza di identificazione `0x57` nella PROM, ma il driver ne2000 ne è al corrente. Per quanto riguarda le schede configurabili via software, si può scaricare il programma di setup dal sito `www.dlink.com`. Si noti che esistono anche schede della Digital (DEC) chiamate DE100 e DE200, ma che la somiglianza finisce qui.

4.16.2 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 4.5.1 (AMD LANCE).

4.16.3 DE-528

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

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

4.16.4 DE-530

Stato: Supportata, Nome del driver: de4x5, tulip

Questa è un'implementazione generica del chip DEC 21040 e ci è stato riferito che funziona con il driver generico tulip per il 21040. Si noti che questa scheda NON èla DFE530.

Si veda la sezione 4.18.4 (DEC 21040) per maggiori informazioni su queste schede e sullo stato attuale del driver.

4.16.5 DE-600

Stato: Supportata, Nome del driver: de600

La DE600 è una vecchia interfaccia Ethernet per porta parallela destinata tipicamente all'uso di utenti laptop.

Ci si aspetti una velocità di trasferimento di circa 180kb/s da questo dispositivo. Si dovrebbe leggere il file README.DLINK nei 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.

4.16.6 DE-620

Stato: Supportata, Nome del driver: de620

Analoga alla DE-600, ma con due formati d'uscita. Si vedano le informazioni precedenti sulla DE-600.

4.16.7 DE-650

Stato: Semi supportata, Nome del driver: pcnet

Alcuni utenti hanno usato questa scheda PCMCIA da qualche tempo nei loro computer portatili. In pratica è un design basato sul chip 8390, come la NE2000. Si suppone anche che la scheda PCMCIA della LinkSys e la IC-Card Ethernet siano dei cloni della DE-650.

4.16.8 DFE-530TX

Stato: Supportata, Nome del driver: via-rhine

Un altra scheda costruita sul chipset VIA Rhine. Schede di costruzione più utilizzano il RhineII (si veda la sezione 4.46.2 (VIA Rhine)). Non si confonda questa scheda con la DE-530 che invece basata sul chipset tulip o con la DFE-530+ che invece è una 8139.

4.16.9 DFE-530TX+, DFE-538TX

Stato: Supportata, Nome del driver: 8139too, rtl8139 (vecchio driver)

Questa scheda usa il chipset RealTek 8139, si faccia riferimento alla sezione 4.36.6 (RealTek 8139)

4.16.10 DFE-550TX

Stato: Supportata, Nome del driver: sundance

4.16.11 DFE-570TX

Stato: Supportata, Nome del driver: tulip

Questa è una scheda tulip (DS1143) a quattro porte.

4.16.12 DFE-580TX

Stato Supportata, Nome del driver: sundance

4.16.13 DGE-500T

Stato: Supportata, Nome del driver: ns83820

4.16.14 DGE-550T

Stato: Supportata, Nome del driver: dl2k

4.17 DFI

4.17.1 DFINET-300 e DFINET-400

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

Queste due sono altri terribili cloni della scheda NE2000, che usano 'DFI' come identificativo nei primi 3 byte della PROM, invece di usare `0x57` nei byte 14 e 15, che è quello che tutte le schede NE1000 e NE2000 dovrebbero fare (la 300 è un pseudo-clone della NE1000 a 8 bit mentre la 400 è una cattiva imitazione della NE2000).

4.18 Digital / DEC

4.18.1 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 e che queste schede sono tutte basate sul chip LANCE dell'AMD. Si faccia riferimento alla sezione 4.5.1 (AMD LANCE) per maggiori informazioni. Si possono usate sino ad un massimo di due schede ISA sullo stesso sistema, poiché esse possono essere installare solamente agli 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 su macchine basate su CPU Alpha e ci sono diverse ioctl() con le quali l'utente può smanettare.

4.18.2 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 in schede precedenti come la DE200. Queste schede supportano sia la memoria condivisa che l'I/O programmato, sebbene si subisca un 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 nel file `README.ewrk3`. Ambedue i file sono nella distribuzione standard del kernel. Questo driver include supporto per le CPU Alpha, giusto 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 così via. Gli smanettoni vadano a 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`) ed altri strumenti, che 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`).

4.18.3 DE425 EISA, DE434, DE435, DE500

Stato: Supportate, Nome del driver: de4x5, tulip

Queste schede si basano sul chip 21040 menzionato di seguito, in particolare la DE500 usa il chip 21140 per fornire connessioni Ethernet a 10/100Mbs. Si deve legga la sezione seguente sul chip 21040 per ulteriori informazioni. Esistono alcune 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 tipo di cavo (tranne, temporaneamente, la DE500 a causa di un brevetto).

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

4.18.4 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 bus PCI ma sembra che questi chip non vengano più prodotti da quando Intel ha acquisito la divisione semiconduttori di Digital e ha deciso di preferire la propria linea di chip Ethernet.

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 particolare caso i driver potrebbero non funzionare*. David C. Davies ci scrive:

«Non c'è alcuna garanzia che il 'tulip.c' o il 'de4x5.c' funzionino con una qualsiasi scheda basata sulla famiglia 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 di schede e tutti lo fanno in maniera diversa e (2) nei DC21142/3 c'è ora un registro di controllo SIA (alla maniera della DC21041).

L'unico barlume di speranza è che si riesca a decodificare la SROM per aiutare nell'impostazione del driver. Comunque, questa non è una soluzione garantita in quanto alcuni produttori (per esempio nel caso della scheda SMC 9332) non seguono le raccomandazioni di Digital Semiconductor sul formato di programmazione della SROM.»

In termini non tecnici, ciò significa che se non si è sicuri che una scheda sconosciuta con un chip DC2114x funzionerà con i driver per Linux, e quindi ci si assicuri di poterla 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 supporta lo standard 100Base-T 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.

Se si hanno problemi con il tulip driver, si può provarne la versione più recente dal sito ftp/WWW di Donald.

Tulip Driver `http://www.scyld.com/network`

All'URL precedente c'è anche un elenco (non esaustivo) delle diverse schede e produttori che usano il chip 21040.

4.19 Farallon

La Farallon vende le schede e i transceiver EtherWave. Questi dispositivi permettono di concatenare diversi dispositivi 10baseT.

4.19.1 Farallon Etherwave

Stato: Supportato, Nome del driver: 3c509

Ci è stato riportato che questo dispositivo è un clone del 3c509 che include il transceiver EtherWave. La gente l'ha usato con successo sotto Linux con l'attuale driver 3c509. Queste schede sono troppo costose per l'uso comune, ma sono una utilissima opzione in casi speciali. I prezzi degli hublet partono da $125, e l'Etherwave aggiunge $75-$100 al prezzo della scheda

4.19.2 Farallon PCI 593

Stato: Supportata, Nome del driver: de4x5, tulip

Ci è stato riferito che questa scheda è stata rilevata dal driver `de4x5`.

4.20 Fujitsu

Diversamente da molti sviluppatori di chip Ethernet, la Fujitsu ha prodotto e venduto anche alcune schede di rete basate sui loro chip.

4.20.1 Fujitsu FMV-181/182/183/184

Stato: Supportato, Nome del driver: at1700, fmv18x(vecchio driver)

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.

I kernel precedenti usavano usavano il driver `fmv18x` ma il supporto per queste schede è stato aggiunto al driver `at1700` e così il vecchio driver è stato progressivamente ritirato.

Older kernels used the driver `fmv18x` but support for these cards was added to the `at1700` driver and so the former has been phased out.

4.21 Hewlett Packard

4.21.1 HP Night Director+ 10/100

Stato: Supportata, Nome del driver: pcnet32

Apparentemente queste schede usano il chip AMD 79C972.

4.21.2 27245A

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

Una 10BaseT a 8 bit basata sul 8390, non è raccomandata per tutte quelle ragioni caratteristiche delle schede a 8 bit.

4.21.3 HP EtherTwist, PC Lan+ (27247, 27248, 27252A, 27269B)

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

La HP PC Lan+ è diversa dalla scheda HP PC Lan standard e può essere fatta funzionare in modalità PIO come una ne2000, o in modalità a memoria condivisa come una wd8013.

4.21.4 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 socket per il boot PROM. 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 stato incluso grazie alle informazioni fornite da Glenn Talbott dell'HP.

4.21.5 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 4.5.1 (AMD LANCE).

4.21.6 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 sia schede ISA che EISA e PCI, quando si esegue `make config` sui sorgenti del kernel lo si trova nella sezione relativa alle schede ISA.

4.21.7 HP NetServer 10/100TX PCI (D5013A)

Stato: Supportata, Nome del driver: eepro100

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

4.22 IBM / International Business Machines

4.22.1 IBM Thinkpad 300

Stato: Obsoleta, Nome del driver: znet

Questa scheda è basata sul chip Intel 82593 ed il relativo driver è stato dichiarato obsoleto nei kernel della serie 2.4.

4.22.2 IBM Credit Card Adaptor for Ethernet

Stato: Semi-supportato, Nome del driver: pcnet_cs

4.22.3 IBM 10/100 EtherJet PCI

Stato: Supportata, Nome del driver: eepro100

Questa scheda è stata indicata come compatibile a livello di driver con la Intel EtherExpress Pro 100.

4.22.4 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 source routing per token ring. Ed è proprio la scrittura di queste ultime che richiederebbe il più grande impegno.

L'iniziale sviluppo dei driver è stato fatto con schede IBM Token Ring per bus ISA e MCA ed è stato testato con una scheda MCA 16/4 Megabit Token Ring, ma dovrebbe funzionare con altre schede basate sul chipset Tropic.

4.23 Schede Ethernet ICL

4.23.1 ICL EtherTeam 16i/32

Stato: Supportata, Nome del driver: eth16i

Questo Driver Supporta sia la versione ISA (16i) che la versione EISA (32) della scheda ed usa il chip MB86965 della Fujitsu, usato anche nelle schede at1700.

4.24 Schede Ethernet Intel

Si noti che la nomenclatura delle diverse schede Intel è ambigua e quanto meno tende a confondere. Se si hanno dubbi, si veda il numero `i8xxxx` sul chip principale della scheda oppure, per le schede PCI, si usino le informazioni PCI contenute nella directory `/proc` e poi le si confronti con i numeri qui elencati. Infine, una pagina nell'area dedicata alle reti sul sito http://support.intel.com risulta spesso d'aiuto se non si riesce ad identificare la scheda di cui si è in possesso.

4.24.1 Ether Express

Stato: Supportata, Nome del driver: eexpress

Questa scheda usa il chip i82586 dell'Intel. Le prime versioni di questo driver (nei kernel 1.2) erano classificate come sperimentali e non funzionavano bene per la maggior parte degli utenti. Il driver nei kernel 2.0 sembra funzionare molto meglio per coloro che lo hanno provato, sebbene il driver sia ancora indicato come 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 rallentamento di rimpiazzare nel driver tutti gli `outb` con `outb_p` sembra aver funzionato per almeno un utente. Si controlli inoltre che la dimensione del buffer in RAM indicata dal driver corrisponda con quanto riportato dall'utilità di Intel.

4.24.2 Ether Express PRO/10 (PRO/10+)

Stato: Supportata, Nome del driver: eepro

Bao Chau Ha ha scritto un driver per queste schede che era incluso nei primi kernel 1.3.x e può funzionare anche con alcuni sistemi Ethernet integrati della Compaq basati sul chip i82595. Potrebbe essere necessario usare l'utilità fornita con la scheda per disabilitare la funzionalità plug and play qualora questo fosse necessario.

4.24.3 Ether Express PRO/10 PCI (EISA)

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

Esiste un driver per la versione PCI di queste schede che viene distribuito separatamente dai kernel di default. Queste schede usano il chip d'interfaccia PCI PLX9036 ed un LAN controller chip i82596 della Intel. Se la propria scheda usa invece il chip i82557, allora `non` si possiede questa scheda ma piuttosto la versione discussa di seguito e quindi si deve invece usare il driver EEPro100.

Si può ottenere il driver sperimentale per la scheda PCI PRO/10 assieme alle relative istruzioni 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 rilevamento usati nei due casi.

4.24.4 Ether Express PRO 10/100B

Stato: Supportata, Nome del driver: e100 o eepro100

Il driver e100 è stato rilasciato da Intel, mentre il driver eepro100 è stato scritto da Donald. Si noti che il driver eepro100 *non* funzionerà con le vecchie schede 100A. I numeri dei chip elencati nel driver sono i82557, i82558, i82559, i82801 e circa 25 altri ID PCI. Per aggiornamenti del driver e/o supporto sul driver, si veda

EEPro-100B Page `http://www.scyld.com/network`

4.24.5 E1000 Gigabit

Stato: Supportata, Nome del driver: e1000

4.25 Kingston

La Kingston produce diverse schede, tra cui schede basate su NE2000+, AMD PCnet e DEC tulip. La maggior parte di queste schede dovrebbe funzionare bene con i rispetti driver. Si faccia riferimento alla *pagina web della Kingston* `http://www.kingston.com`

4.26 LinkSys

La LinkSys ha prodotto un numero di diversi cloni NE2000, alcune delle quali sono schede ISA semplici, altre ISA plug-and-play e altre ancora sono cloni PCI della ne2000 basati su uno dei chip PCI ne2000 supportati. Ci sono semplicemente troppi modelli per elencarli tutti qui. Il loro sito è `www.linksys.com`

4.26.1 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 mentre la terza revisione usava un chip Linksys 82c19 e la quarta usa il chip Comet di AMDtek. Il supporto per le ultime tre varianti è stato aggiunto al driver tulip standard e potrebbe essere necessario aggiornare i vostri driver a seconda di quanto vecchi essi siano.

Ulteriori informazioni sul PNIC sono disponibili sul sito

`http://www.scyld.com/network`

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

4.26.2 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 il corrispondente driver. Si veda la sezione 4.16.6 (DE-620) per maggiori informazioni.

4.26.3 LinkSys PCMCIA Adaptor

Stato: Supportato, Nome del driver: pcnet_cs

Questa è una versione rimarchiata del DE-650.

4.27 Microdyne (Eagle)

Eagle Technology (un tempo anche conosciuta sotto il nome di Novell cards) è stata acquisita da Microdyne. Se non potete trovare qui la vostra scheda, controllate la sezione di questo documento dedicata a Novell. Mentre Microdyne non è più sul mercato con schede di rete, c'è ancora un po' di materiale relativo ai loro prodotti di rete sul loro sito `ftp.microdyne.com`.

4.27.1 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.

4.28 Mylex

La Mylex può essere raggiunta ai seguenti numeri, nel caso qualcuno voglia chieder loro 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`

4.28.1 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.

4.28.2 Mylex LNP101

Stato: Supportata, Nome del driver: de4x5, tulip

Questa è una scheda PCI basata sul chip 21040 della DEC. Può essere selezionata 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 (4.18.4 (DEC 21040)) per maggiori informazioni.

4.28.3 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).

4.29 Myson

4.29.1 Myson MTD-8xx 10/100 PCI

Stato: Supportata, Nome del driver: fealnx

Sembra che le schede vendute con il marchio Surecom EP-320X-S includano anche loro questo chip.

4.30 National Semiconductor

National Semiconductor è un produttore di microchip, non di schede di rete. Dei terzi acquistano il loro chipset, li saldano su un pezzo di fibra di vetro con altre schifezze, scrivono il loro nome sul tutto e ve lo rivendono.

4.30.1 NS8390, DP8390, DP83905 etc.

Stato: Supportata, Nome del driver: 8390

L'infame chip 8390, reperibile in uno zilione di schede ISA, e clonato da diversi altri produttori di chip. Si noti che il file `8390.o` non contiene un driver completo in se, ma va usato in congiunzione con un altro driver che conosce la modalità di interfacciamento tra l'8390 ed il bus del computer. Esempi di questa seconda metà del driver sono `wd.o`, `3c503.o`, `smc-ultra.o`, `ne2k-pci.o` e così via.

4.30.2 DP83800 with DP83840

Stato: Not Supportata.

Si veda la sezione per la NE 10/100 di seguito.

4.30.3 DP83815/83816

Stato: Supportata, Nome del driver: natsemi

`http://www.scyld.com/network/natsemi.html`

Questo driver è reperibile con i kernel 2.4 e seguenti.

4.30.4 NS83820, DP83820

Stato: Supportato, Nome del driver: ns83820

L'83820 è una scheda a 64 bit da 10/100/1000 Mbps per il bus PCI, e la 83821 è la corrispondente versione a 32 bit (ma sembra che le due siano in effetti identiche e che la EEPROM sia incaricata di definire la larghezza dell'uscita dati. Esattamente come nel caso del chip 8390, solitamente non si incappa in questo numero se non si guardano i chip sulla scheda.

4.31 Novell Ethernet, NExxxx e cloni associati

Il prefisso 'NE' sta per Novell Ethernet. La Novell ha seguito il progetto più economico del databook della National Semiconductor e ha venduto i diritti di produzione a Eagle (che ha anche lanciato?), solo per immettere sul mercato schede Ethernet a prezzi ragionevoli (la ormai onnipresente scheda NE2000).

4.31.1 NE1000, NE2000

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

ne2000 è oramai divenuto il nome generico per un design essenziale basato sul chip 8390 della National Semiconductor. Queste schede usano I/O programmato piuttosto che memoria condivisa, il che comporta maggiore facilità di installazione ma prestazioni un po' più basse ed alcuni problemi. Alcuni dei problemi più comuni con le schede NE2000 sono elencati nella sezione 2.6 (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 durante il boot. Alcuni cloni NE2000 scadenti non la prendono bene ad essere pungolati in aree sbagliate e risponderanno bloccando la macchina. Inoltre anche `0x320` non va bene perché i driver SCSI rilevano all'indirizzo `0x330`.

Donald ha scritto un progamma di diagnostica per NE2000 (ne2k.c) che va bene per tutte le schede ne2000. Si veda la sezione 6.2 (Programmi diagnostici) per maggiori informazioni.

Se si intende usare questo driver come modulo caricabile nel kernel probabilmente si dovrebbe fare riferimento alla sezione 8.3 (Usare un driver Ethernet come modulo) per informazioni specifiche sui moduli.

4.31.2 NE2000-PCI (RealTek/Winbond/Compex)

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

Sì, che ci si creda o no, c'e' gente che sta facendo schede PCI basate sul progetto dell'interfaccia 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 2.0 ha il supporto per rilevare automaticamente tutte queste schede ed usarle (se si sta usando un kernel 2.0.34 o più vecchio, lo si dovrebbe aggiornare per assicurarsi che la propria scheda venga rilevata). Ci sono ora 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 lontanamente 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 vengono spostati da e per la scheda in blocchi più grandi, e senza alcuna pausa tra un'operazione e l'altra come le vecchie NE2000 ISA invece richiedono per operare in maniera affidabile. Il risultato è un driver più efficiente, ma non ci si esalti troppo in quanto le differenze non saranno ovvie sotto carichi di traffico normale (se si desidera veramente la massima efficenza unitamente ad un basso utilizzo della CPU, allora una NE2000 PCI è semplicemente una pessima scelta). Aggiornamenti del driver e ulteriori informazioni possono essere trovate nel sito:

`http://www.scyld.com/network`

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

Si noti inoltre che diversi prodottori di schede sono noti per l'aver etichettato come 'NE2000 Compatible' dei prodotti che sono in effetti completamente differenti (e.g. PCNet-PCI o RealTek 8139). Se si è in dubbio si confronti il numero del chip principale della scheda con quelli elencati in questo documento.

4.31.3 NE-10/100

Stato: Non supportata.

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

4.31.4 NE1500, NE2100

Stato: Supportate, Nome del driver: lance

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

Alcune delle prime versioni del driver lance avevano problemi nel determinare la linea IRQ attraverso autoIRQ nelle schede Novell/Eagle 7990. Si spera che questo problema sia ora stato corretto, ma se non lo fosse, allora si specifichi l'IRQ usando LILO e ci si informi del fatto che il problema ancora sussiste.

Informazioni sulla selezione del DMA e la numerazione del chip possono essere trovate nella sezione 4.5.1 (AMD LANCE).

4.31.5 NE/2 MCA

Stato: Semi supportata, Nome del driver: ne2

Ci sono state alcune schede NE2000 microchannel prodotte da diverse compagnie. Questo driver, disponibile nei kernel 2.2, rileverà le seguenti schede MCA: Novell Ethernet Adapter NE/2, Compex ENET-16 MC/P e Arco Ethernet Adapter AE/2.

4.31.6 NE3200

Stato: Non supportata

Anche se non c'è un driver per questa scheda nel kernel corrente (2.4), Rask Ingemann Lambertsen ha sperimentato con una vecchia macchina EISA ed aveva rilasciato un prototipo di driver sul seguente sito:

`http://vip.cybercity.dk/~ccc94453/linux/ne3200/`

4.31.7 NE3210

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

Questa scheda EISA è completamente diversa dalla NE3200, essendo basata sul chip 8390 della National Semiconductor. Si può reperire il driver tra i sorgenti del kernel 2.2. Ci si assicuri di impostare un indirizzo della memoria condivisa al di sotto del primo MB o al di sopra dell'indirizzo più alto della RAM fisica installata nella macchina.

4.31.8 NE4100

Stato: Supportata, Nome del Driver: pcnet_cs

4.31.9 NE5500

Stato: Supportata, Nome del driver: pcnet32

Queste sono delle schede basate sul chip PCnet-PCI ('970A) dell'AMD. Maggiori informazioni sulle schede basate su LANCE/PCnet possono essere trovate nella sezione 4.5.1 (AMD LANCE).

4.32 Netgear

4.32.1 Netgear FA-311

Status: Supportata, Nome del Driver: natsemi

4.32.2 Netgear GA-620

Status: Supportata, Nome del Driver: acenic

4.32.3 Netgear GA-621

Status: Supportata, Nome del Driver: ns83820

4.33 Proteon

4.33.1 Proteon P1370-EA

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

Sembra che questa sia un clone NE2000, e funziona bene con Linux.

4.33.2 Proteon P1670-EA

Stato: Supportata, Nome del driver: de4x5, tulip

Questa è un'altra scheda PCI basata sul chip Tulip della DEC, e si dice che funzioni bene con Linux.

Si veda la sezione sul chip 21040 (4.18.4 (DEC 21040)) per maggiori informazioni sul driver.

4.34 Pure Data

4.34.1 PDUC8028, PDI8023

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

Le serie di schede PDUC8028 e PDI8023 della Pure Data sono 'quasi cloni' delle schede wd80x3 e c'è del codice scritto specificamente per verificare la loro eventuale presenza in `wd.c`.

4.35 Racal-Interlan

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

4.35.1 ES3210

Stato: Semi supportata, Nome del driver: es3210

Questa è una scheda EISA a memoria condivisa basata sul chip 8390. Con il kernel 2.2 è distribuito un driver sperimentale che si dice funzioni bene, ma il rilevamento in EISA dell'IRQ e dell'indirizzo della memoria condivisa non sembra funzionare bene con (almeno) le prime revisioni della scheda (comunque questo problema non è ristretto al solo mondo Linux...). In questo scenario, si devono fornire manualmente le informazioni necessarie al driver. Per esempio, per una scheda all'IRQ 5 e memoria condivisa all'indirizzo `0xd0000` che usa un driver modulare, si aggiunga `options es3210 irq=5 mem=0xd0000` a `/etc/modules.conf`. Oppure, con il driver compilato nel kernel, all'avvio si passino i parametri `ether=5,0,0xd0000,eth0`. L'indirizzo I/O di base viene rilevato automaticamente e quindi si dovrebbe usare il valore 0.

4.35.2 NI5010

Stato: Semi supportata, Nome del driver: ni5010

Un tempo ci si doveva procurare separatamente il driver per queste vecchie schede a 8 bit della MiCom-Interlan, ma ora viene distribuito con i kernel 2.2 come driver sperimentale.

4.35.3 NI5210

Stato: Semi supportata, Nome del driver: ni52

Anche questa scheda usa uno dei chip della Intel. Michael Hipp ha scritto un driver che viene ora incluso nel kernel standard come driver 'alpha'. Michael vorrebbe ricevere commenti dagli utenti che posseggono questa scheda. Si veda la sezione 2.3 (Driver sperimentali) per importanti informazioni sull'uso dei driver Ethernet sperimentali.

4.35.4 NI6510 (non EB)

Stato: Semi supportata, Nome del driver: ni65

Esiste anche un driver per la NI6510 (basata su chip LANCE) ed è stato anche questo scritto da Michael Hipp. Come l'altro, anche questo è un driver sperimentale. Per qualche ragione questa scheda non è compatibile con il driver LANCE generico. Si veda la sezione 2.3 (Driver sperimentali) per importanti informazioni sull'uso dei driver Ethernet sperimentali.

4.35.5 EtherBlaster (aka NI6510EB)

Stato: Supportata, Nome del driver: lance

A partire dal kernel 1.2.23, al driver LANCE generico è stato aggiunto un controllo per il valore (signature) `0x52, 0x44` specifica della NI6510EB. Alcuni hanno riferito che questa firma non è la stessa per tutte le schede NI6510EB, il che fa sì che il driver LANCE non rilevi sempre questa scheda. Se questo succede, si può sostituire il controllo di detto valore (circa alla riga 322 in lance.c) a printk() cosicché vengano stampati i valori per la propria scheda, valori che poi si potrà usare come default invece di `0x52, 0x44`.

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

4.36 RealTek

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

Stato: Supportato, Nome del driver: atp

Questo è un adattatore pocket generico a basso costo, venduto dalla AT-Lan-Tec e (probabilmente) da diversi altri produttori. Nel kernel standard è incluso un driver che lo supporta. 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`.

4.36.2 RealTek 8008

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

È stato riferito che questo chip si comporta in maniera similare all' AT/LANTIC in quanto può essere configurato per operare sia in modalità ne/PIO che wd/MMIO impiegando il software fornito dal produttore (SET8008R).

4.36.3 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 Web della RealTek (`http://www.realtek.com.tw`) o via FTP dallo stesso sito può essere scaricato il programma `rset8009.exe`.

4.36.4 RealTek 8019

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

Questa è la versione Plug and Pray (innesta e prega, N.d.T.) 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/modules.conf`). Dal sito WWW della RealTek (`http://www.realtek.com.tw`) o via FTP dallo stesso sito può essere scaricato il programma `rset8009.exe`.

4.36.5 RealTek 8029

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

Questa è una implementazione PCI a singolo chip di un clone NE2000. Diversi produttori vendono schede basate su questo chip. Si veda la sezione 4.31.2 (NE2000-PCI) per informazioni sull'uso di una qualsiasi di queste schede. Si noti che questo è un design di più di 10 anni fa banalmente adattato al bus PCI. Le prestazioni non saranno molto migliori rispetto a quelle dell'equivalente modello ISA.

4.36.6 RealTek 8129/8139

Stato: Supportate, Nome del driver: 8139too, rtl8139 (vecchio driver)

Una soluzione Ethernet PCI a chip singolo della RealTek. Un driver per le schede basate su questo chip è stato incluso nella release 2.0.34 del kernel Linux. In kernel recenti, il driver principale per questa scheda va sotto il nome `8139too`. Nei kernel precedenti, il driver per queste schede si chiamava `rtl8139` e solitamente era necessario rispondere 'Y' quando veniva richiesto se si desiderava accesso ai driver sperimentali.

4.37 Sager

4.37.1 Sager NP943

Stato: Semi supportata, Nome del driver: 3c501

Questa è semplicemente un clone della 3c501, con un diverso prefisso S.A. in PROM. Suppongo che sia talmente demente quanto la 3c501 originale. Il driver cerca l'ID della NP943 e poi la tratta semplicemente come un 3c501. Si veda la sezione 4.1.1 (3Com 3c501) per tutte le ragioni per le quali non si deve proprio usare una di queste schede.

4.38 Schneider & Koch

4.38.1 SK G16

Stato: Obsoleta, Nome del driver: sk_g16

Questo driver era già stato incluso nei kernel 1.1 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 (la 7990). Ancora una volta, sembra che pure questa scheda non funzioni con il driver LANCE generico.

Questo driver è diventato obsoleto a partire dai kernel 2.4.

4.39 SEEQ

4.39.1 SEEQ 8005

Stato: Supportata, Nome del driver: seeq8005

Nel driver sono incluse pochissime informazioni su questa scheda e quindi ci sono pure poche informazioni da includere qui. Se si hanno domande, probabilmente la cosa migliore è di scrivere all'autore del driver come elencato nel sorgente del driver.

Questo driver è diventato obsoleto a partire dai kernel 2.4.

4.40 SiS (Silicon Integrated Systems)

SiS produceva chipset per schede madri già nell'era del processore 386 (come il mitico 'el cheapo', N.d.T.), ed adesso producono anche alcuni chip Ethernet, anch'essi alquanto popolari.

4.40.1 SiS 900 (7016, 630E, 962)

Stato: supportata, Nome del driver: sis900

Questo dispositivo può essere incontrato sia come scheda di espansione a se stante che come dispositivo integrato nella scheda madre. Il driver è stato disponibile a partire dagli ultimi kernel 2.2.

4.41 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 Web: *SMC* `http://www.smc.com` .

4.41.1 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 il risparmio solo se destinato ad un uso leggero in termini di traffico di rete. 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 questo fallisce, il driver assegna l'IRQ 5 senza dir niente. Dal sito FTP della SMC si possono scaricare i dischetti di configurazione/driver. Si noti che alcune delle versioni più recenti del programma 'SuperDisk' della SMC falliranno nel rilevare le schede veramente vecchie senza EEPROM. Il file `SMCDSK46.EXE` sembra essere in generale una buona scelta. Inoltre le impostazioni dei ponticelli per tutte le schede SMC si possono reperire in un file testo ASCII nel summenzionato archivio. 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 dovrebbero consultare le sezioni successive per maggiori informazioni.

4.41.2 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 Semiconductor presente nelle prime schede. Il progetto a memoria condivisa rende queste schede un po' più veloci delle schede PIO, specialmente con pacchetti di dimensioni notevoli. Più importante, dal punto di 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 incappa nel il registro dati dell'I/O programmato che pianta la macchina durante la scansione al boot.

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

Le schede con a bordo dimensioni di memoria non standard le possono specificare all'avvio (o come opzione in `/etc/modules.conf` 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. Ad 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 (con I/O=0x280 e IRQ 9) al seguente parametro di avvio:

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

Si veda anche la sezione 2.7 (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 consultare la sezione 8.3 (Usare un driver Ethernet come modulo) per informazioni specifiche sui moduli.

4.41.3 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 all'83c690. Sebbene possieda una modalità simile alle schede SMC più vecchie, la scheda non è completamente compatibile con i vecchi driver WD80*3. Tuttavia, 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, si suppone che il controllo per la Ultra la individui prima che il controllo per la wd8013 abbia la possibilità di identificarla 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 in dischi SCSI utilizzati attraverso un adattatore host aha-154X. Questo problema colpisce 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 driver. Il driver Ultra attuale si protegge da questo problema abilitando la memoria condivisa solo durante lo scambio di dati con la scheda. Ci si assicuri che la propria versione del kernel sia almeno la 1.1.84 e che la versione riportata dal driver al boot sia almeno `smc-ultra.c:v1.12`, altrimenti si è vulnerabili a questo problema.

Se si intende usare questo driver come modulo caricabile probabilmente si dovrebbe consultare la sezione 8.3 (Usare un driver Ethernet come modulo) per informazioni specifiche sui moduli.

4.41.4 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 della serie 2.0 che della serie 2.2. Grazie a Leonard Zubkoff per aver acquistato alcune di queste schede dimodoché sia stato possibile aggiungerne il supporto a Linux.

4.41.5 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 Web/FTP) per disabilitare il PnP e attivare la modalità a memoria condivisa. Si vedano le informazioni precedenti per alcune note sul driver Ultra.

Per i kernel 1.2, la scheda deve essere configurata per l'uso a memoria condivisa. Comunque i kernel 2.0 possono usare la scheda sia in modalità a memoria condivisa che a I/O programmato. La modalità a memoria condivisa è leggermente più veloce e usa anche meno risorse di CPU.

4.41.6 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 nei sorgenti del kernel corrente.

Si veda la sezione 4.18.4 (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.

4.41.7 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 2.0 e 2.2 per supportare queste schede. Per ulteriori dettagli si veda:

`http://www.scyld.com/network`

4.41.8 SMC 1211TX 10/100

Stato: Semi supportata, Nome del driver: 8139too, rtl8139 (vecchio driver)

Sembra che SMC non sia più la stessa compagnia che ha rilasciato schede come la Ultra e la EPIC. La divisione che si occupa del design di chip si chiama ora SMSC e d'ora in poi vedremo il nome SMC attaccato su schede destinate al mercato economico come questa - una Realtek 8139 con una EEPROM modificata.

4.41.9 SMC 3008

Stato: Non supportata.

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

4.41.10 SMC 3016

Stato: Non Supportata.

Queste sono schede 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. Queste schede non sono molto comuni.

4.41.11 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 piuttosto poco comune.

4.41.12 SMC 91c100

Stato: Semi supportata, Nome del driver: smc9194

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

4.41.13 SMC 9452TX/9462TX

Stato: Supportata, Nome del driver: ns83820

4.42 Sundance

4.42.1 Sundance ST201, Alta

Stato: Supportata, Nome del driver: sundance

Il chip Sundance Alta viene utilizzato su schede OEM (Original Equipment Manufacturer, N.d.T.), utilizza trasferimenti in bus-master, può trasmettere e ricevere in buffer allineati in modo arbitrario ed ha un hash di multicasting a 4 elementi. Tutte le versioni di questo chip hanno controllo di flusso in hardware e stati ACPI di alimentazione.

4.43 SysKonnect

4.43.1 SysKonnect sk-98xx Gigabit Ethernet

Stato: Supportata, Nome del driver: sk98

I primi report indicano che questo chipset ha un problema con i checksum di trasmissione, il che riduce un poco le prestazioni.

4.44 Texas Instruments

4.44.1 ThunderLAN

Stato: Supportata, Nome del driver: tlan

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

4.45 Thomas Conrad

4.45.1 Thomas Conrad TC-5048

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

Si veda la sezione sul chip 21040 (4.18.4 (DEC 21040)) per maggiori informazioni.

4.46 VIA

Probabilmente non si avrà a che fare con una scheda di rete VIA, in quanto la VIA costruisce diversi chip usati da altri per costruire le loro schede Ethernet. Il loro sito Web è il seguente:

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

4.46.1 VIA 86C926 Amazon

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

Questo chip è la soluzione VIA compatibile PCI-NE2000. Si può scegliere tra il driver `ne.c` per ISA e PCI o il driver solo PCI `ne2k-pci.c`. Si veda la sezione sulle schede PCI-NE2000 per ulteriori informazioni.

4.46.2 VIA 86C100A Rhine II (and 3043 Rhine I)

Stato: supportato, Nome del driver: via-rhine

Questo driver relativamente nuovo è incluso 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 a 32 bit del buffer limitano i benefici conseguenti. Per maggiori dettagli e driver aggiornati si veda:

`http://www.scyld.com/network`

4.47 Western Digital

Si veda la sezione 4.41 (SMC) per informazioni sulle schede SMC (la SMC ha comprato la divisione schede di rete della Western Digital molti anni fa).

4.48 Winbond

La Winbond in realtà non fabbrica schede complete da vendere al pubblico, piuttosto costruisce soluzioni Ethernet su un singolo chip che altre compagnie possono acquistare ed usare nelle schede PCI poi vendute sotto i loro rispettivi marchi. Alcuni programmi di setup e supporto tecnico sono disponibili sul sito:

`http://www.winbond.com.tw`

4.48.1 Winbond 89c840

Stato: Supportato, Nome del driver: winbond-840

Questo chip è stato descritto come il 'figlio mutante di una NE2000 ed un clone Tulip' – si vedano le note del driver per ulteriori dettagli. Questo driver supporta anche il chip TX9882 presente nella Compex RL100-ATX.

4.48.2 Winbond 89c904, 89c905, 89c906

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

Questi sono i chip Winbond compatibili con NE2000 per bus ISA a 10 Mbps. I programmi di setup sono disponibili sul sito della Winbond.

4.48.3 Winbond 89c940

Stato: Supportata, 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 produttori. Si noti che questo è sempre un progetto vecchio più di dieci anni adattato per il bus PCI. Le prestazioni non saranno tanto migliori rispetto a quelle dell'equivalente modello ISA.

4.49 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 un accordo di confidenzialità rigidissimo per averle. Apparentemente abbastanza utenti Linux li hanno subissati di richieste per un driver (dichiarano di supportate tutti i più popolari sistemi operativi di rete...) da far alterare la loro politica in materia e permettere il rilascio di documentazione senza dover firmare un accordo di confidenzialità. Ad alcuni è stato detto che sarebbe stato rilasciato il codice sorgente del driver SCO, mentre ad altri hanno detto che non vengono più fornite informazioni su prodotti 'obsoleti' come i primi modelli PE. Se si è interessati e si vuole verificare personalmente come stanno le cose, si può contattare la Xircom ai numeri 1-800-874-7875, 1-800-438-4526 o +1-818-878-7600.

4.49.1 Xircom PE1, PE2, PE3-10B*

Stato: Non supportate.

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

4.49.2 Xircom CE, CEM, CE2, CE3

Stato: Supportata, Nome del driver: xirc2ps_cs

Secondo il driver, sono supportate le schede CE2, CE IIps, RE-10, CEM28, CEM33, CE33, CEM56, CE3-100, CE3B, RE-100, REM10BT e la REM56G-100.

4.49.3 Xircom CBE-100

Stato: Supportata, Nome del driver: xircom_tulip_cb

Una implementazione simile al design tulip su CardBus.

4.50 Zenith

4.50.1 Z-Note

Stato: Obsoleta, Nome del driver: znet

L'adattatore di rete built-in Z-Note è basato su un i82593 dell'Intel ed usa *due* canali DMA. Si noti che il ThinkPad 300 dell'IBM è compatibile con Z-Note.

4.51 Znyx

4.51.1 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 Donald.

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 4.18.4 (DEC 21040) per maggiori informazioni su queste schede e la situazione corrente del driver.

4.52 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 di modello che potrebbe costituire un indizio. Qualsiasi numero di modello contenente il numero 2000 sarà probabilmente 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 queste.

4.52.1 Identificare il Network Interface Controller

Si cerchi il chip più grosso sulla scheda. Questo sarà il network controller (NIC) e la maggior parte può essere identificata dal numero di parte. Se si sa quale NIC c'è sulla scheda, quanto segue può aiutare a scoprire quale sia la scheda.

Probabilmente il NIC ISA 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 probabile che sia un clone della ne1000 o della ne2000. Il secondo tipo di schede più comuni basate su 8390 sono le schede wd80x3 e i loro cloni. Le schede con un DP83905 possono essere configurate per funzionare 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 Ultra. Tutte le schede 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 ne2k-pci dovrebbe automaticamente rilevarle al boot.

4.52.2 Identificare l'indirizzo Ethernet

Ogni scheda Ethernet ha un suo unico indirizzo a sei byte. I primi tre byte di quell'indirizzo sono gli stessi per ogni scheda fatta da un particolare produttore. Per esempio tutte le schede SMC hanno indirizzi che iniziano con il prefisso con `00:00:c0`. Gli ultimi tre byte dell'indirizzo vengono invece assegnati dal produttore in modo che ogni scheda venduta abbia un indirizzo diverso da tutte le altre.

Se la propria scheda ha un etichetta che indica tutti i 6 byte del suo indirizzo, si può risalire al costruttore dai primi tre. Tuttavia, è più comune il vedere solo gli ultimi tre byte stampati su un'etichetta attaccata sulla PROM, il che non ci dice niente.

Si può determinare a quale produttore è stato assegnato un determinato indirizzo dall'RFC 1340. Apparentemente in giro ci sono anche elenchi più aggiornati. Si provi a fare una ricerca sul Web o in FTP per `EtherNet-codes` o `Ethernet-codes` e qualcosa salterà fuori.

4.52.3 Identificare la scheda a partire dal numero di FCC ID

Come parte del processo di certificazione che una scheda deve tipicamente passare prima di essere vendibile all'utente finale, essa deve venire testata dalla FCC, e durante questo processo essa riceve un identificativo FCC che dovrebbe essere stampato da qualche parte sulla scheda. Per esempio, una scheda con sovrastampato il valore `FCC ID: J158013EWC` risulta poi essere una SMC/WD8013-EWC. Alcuni siti Web come `www.driverguide.com` e `drdriver.com` fanno uso degli identificativi FCC per fornire aiuto agli utenti nell'identificazione delle schede più oscure. La FCC stessa mette a disposizione uno strumento di ricerca che può risultare utile all'indirizzo Web seguente:

FCC IDs `http://www.fcc.gov/oet/fccid`

4.52.4 Suggerimenti per provare ad usare una scheda sconosciuta

Se non si è ancora sicuri di che scheda si tratti 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 sia configurata ad uno degli indirizzi che i driver controllano quando cercano una scheda di quel tipo. Se questo è il caso, si può provare a scaricare `scanport.tar.gz` da un archivio ftp dedicato a Linux e vedere se può scoprire a che indirizzo la scheda è piazzata. Questo strumento esamina lo spazio di I/O del bus ISA a partire da `0x100` e fino a `0x3ff` cercando dispositivi che non siano registrati in `/proc/ioports`. Se trova un dispositivo sconosciuto a qualche indirizzo specifico, si può esplicitamente indirizzare il rilevamento a quell'indirizzo con il parametro `ether=` al boot.

Se si è riusciti a far rilevare la scheda, allora solitamente si può scoprire a cosa servano i ponticelli spostandone 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 staffa 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 sembrano non servire a 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/`

4.53 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 il routing, ma che realmente non trasmette pacchetti.

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

`ibmtr.c` – IBM Token Ring, che non è proprio Ethernet. Broken-Ring richiede source routing ed altre brutte cose.

`loopback.c` – Dispositivo loopback al quale vanno tutti i pacchetti destinati alla vostra macchina partenti da questa stessa. Essenzialmente si limita a spostare 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 due porte parallele connesse in modalità point-to-point.

`ppp.c` – Point-to-Point Protocol (RFC1331, 1548, 1661), per la trasmissione di datagrammi multiprotocollo su una connessione Point-to-Point (solitamente modem).

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

`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 ad esempio la Intel EtherExpress.

5 Cavi, coassiali, doppini intrecciati

Se si sta mettendo su una nuova rete, si utilizzerà probabilmente cablaggio di tipo Cat5 per 10/100BaseT (cavi tipo telco a doppini intrecciati con connettori 'telefonici' RJ-45 a otto vie). Se vi capita di incappare in un po di vecchi cavi 10Base2 thin Ethernet (coassiale RG58 con connettori BNC), si può decidere di riciclarli in una piccola rete casalinga. La vecchia thick Ethernet, con cavi RG-5 a N connettori, è veramente obsoleta e si vede ormai poco in giro.

Si veda la sezione 1.5 (Tipi di cavi...) per una panoramica introduttiva sui cavi. Si noti inoltre che la FAQ di *comp.dcom.lans.ethernet* contiene un sacco di informazioni utili sui cavi e materiali collegati. Ci si connetta in FTP a rtfm.mit.edu e si cerchi nella directory `/pub/usenet-by-hierarchy/` la FAQ di quel newsgroup.

5.1 Thin Ethernet (thinnet)

Thinnet è decisamente obsoleta ormai, ma va ancora bene per qualcuno che sta smanettando con una piccola rete o che vuole costruire una rete per casa. Ci sono due svantaggi principali ad usare thinnet. Il primo è che si è limitati 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 giunzione a T . E 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! Inoltre e necessario che il vostro cavo non abbia 'appendici': il connettore a T deve essere collegato direttamente alle schede.

5.2 Doppino intrecciato (twisted pair)

Le reti a doppini intrecciati necessitano di hub (concentratori) attivi, che costano attorno ai $50. Si può tranquillamente ignorare quelli che dicono che si può usare il cablaggio telefonico già esistente visto che è una rara circostanza.

D'altra parte, tutte le specifiche Ethernet a 100Mb/sec usano doppini intrecciati e sono usati pure nella maggior parte delle nuove installazioni in uffici. I cavi devono essere di categoria 5. Qualsiasi installazione con cavi di specifica inferiore al Cat 5 è inutile.

Se si sta solamente connettendo due macchine, è possibile evitare l'uso di un hub, acquistando o fabbricando uno speciale cavo cross-over (detto anche null cable), ma si tenga presente che alcune schede cercano di rilevare le funzionalità di autonegotiation e di conseguenza si aspettano di star parlando con un hub e non con un altra scheda e quindi potrebbero non funzionare in questa configurazione.

6 Configurazione del software e diagnotici

Per le più vecchie (o le più economiche) schede ISA, i settaggi della scheda erano controllati da piccoli ponticelli (jumpers) di contatto neri piazzati su file di pin. Con l'aumento delle capacità delle schede, questi settaggi sono stati spostati in ambito elettronico piuttosto che fisico e l'utente è ora in grado di di immagazzinare la configurazione prescelta in nella memoria non volatile integrata nella scheda. Un programma fornito dal produttore può essere impiegato dall'utente per alterare questa configurazione, eliminando il bisogno di aprire il computer solo per riconfigurare una scheda.

Nella maggior parte dei casi, se la configurazione viene fatta in software e salvata poi in una EEPROM, si dovrà avviare il DOS e usare il programma per DOS fornito dal rivenditore per impostare IRQ, I/O, indirizzo di memoria e quant'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 cercare nel sito Web 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 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

Fantasiose interfacce a menu. Per poterli usare, nella maggioranza dei casi sarà necessario leggerene 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 ogni caso, si dovrà usare la maggior parte di questi programmi come root (per permettere l'accesso alle porte di I/O) e prima di farlo è consigliabile disattivare la scheda Ethernet usando il comando `ifconfig eth0 down`.

6.1 Programmi di configurazione per le schede Ethernet

6.1.1 Schede WD80x3

Per quanti hanno schede wd80x3, c'è il programma `wdsetup` che può essere reperito nell'archivio `wdsetup-0.6a.tar.gz` nei siti ftp dedicati a Linux. Non è mantenuto attivamente e non è aggiornato da un bel po' di tempo. Se nel vostro caso funziona, allora bene. Altrimenti, 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. Naturalmente, si *deve* possedere una scheda con la EEPROM per poter usare questo programma. Le vecchie, ma proprio *vecchie*, schede wd8003 e alcuni vecchi cloni wd8013 usano dei ponticelli per configurare la scheda.

6.1.2 Schede Digital/DEC

La scheda Digital EtherWorks 3 può essere configurata in maniera simile con il programma DOS `NICSETUP.EXE`. David C. Davies ha scritto questo programma e altri strumenti per la EtherWorks 3 oltre al driver stesso. Si cerchi il file `ewrk3tools-X.XX.tar.gz` nella directory `/pub/linux/system/Network/management` sul sito FTP dedicato a Linux a cui fate riferimento.

6.1.3 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 di setup `atlantic.c` dal server ftp di Donald, `www.scyld.com`. Inoltre, con tutte queste schede sembra funzionino anche i programmi per le schede DP83905 della Kingston, poiché non controllano che l'indirizzo hardware sia specifico produttore all'avvio. Si vada all'URL seguente: *Kingston* `http://www.kingston.com/download/etherx/etherx.htm` e si scarichino `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 accidentale 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 ci si può togliere d'impiccio nel modo seguente: si rimuova la scheda dalla macchina, si riavvii e si entri nel menu di configurazione del BIOS. Si modifichi la voce 'Display Adapter' in 'Not Installed' e si imposti 'A:' (il vostro floppy drive) come disco di avvio di default. Si modifichi 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 di sistema DOS ('format a: /s /u') e ci si copi dentro il floppy il programma `default.exe` dell'archivio `20XX12.EXE` suddetto. Quindi 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 il sistema 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. Una volta riacceso il computer un ultima volta, si dovrebbe avere di nuovo un display che funziona, permettendo così di risistemare le impostazioni del BIOS e modificare i valori della EEPROM della scheda come si desideri.

Si noti che se non si ha DOS sotto mano, si può eseguire quanto sopra descritto 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`.

6.1.4 Schede 3Com

La famiglia di schede Etherlink III della 3Com (p.es. 3c5x9) può essere configurata usando un'altra utilità di configurazione di Donald. Si può scaricare il file `3c5x9setup.c` dal server ftp di Donald, `www.scyld.com` to (si noti che il programma DOS di configurazione 3c5x9B può avere più opzioni pertinenti alla nuova serie B della famiglia Etherlink III).

6.2 Programmi diagnostici per schede Ethernet

Tutti i programmi diagnostici scritti da Donald possono essere ottenuti visitando il suo sito web:

Ethercard Diagnostics `http://www.scyld.com/network`

Allied Telesis AT1700 – `at1700.c`

Cabletron E21XX – `e21.c`

HP PCLAN+ – `hp+.c`

Intel EtherExpress – `eexpress.c`

PCI NE2000 cards – `ne2k-pci-diag.c`

ISA NE2000 cards – `ne2k.c`

RealTek (ATP) Pocket adaptor `atp-diag.c`

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

7 Informazioni tecniche

Queste informazioni saranno utili a quanto vogliono capire un po' meglio come funziona la loro scheda, oppure vogliono smanettare con il driver. Se non si rientra in queste categorie, allora si può anche considerare di saltare questa sezione.

7.1 I/O programmato, memoria condivisa e DMA a confronto

Se già si è in grado di ricevere e inviare pacchetti back-to-back, allora non si possono immettere più bit di così 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.

7.1.1 Programmed I/O (I/O Programmato) (es. NE2000, 3c509)

Pro: Non usa nessuna risorsa di sistema limitata, solo alcuni registri di I/O e non ha un limite a 16M.

Contro: Solitamente più bassa è la velocità di trasferimento, più la CPU attende senza far niente, e l'accesso ai pacchetti in modalità interleaved è solitamente difficile se non impossibile.

7.1.2 Shared memory (Memoria Condivisa) (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 vengono copiati dalla scheda, il che risulta in un'ulteriore riduzione dell'uso della CPU rispetto ad una equivalente scheda PIO.

Contro: Usa più memoria (un grosso contro per utenti DOS, ma sostanzialmente non un problema sotto Linux) e blocca comunque la CPU.

7.1.3 DMA (Accesso diretto alla memoria) in bus mastering (es. LANCE, DEC 21040)

Pro: Libera la CPU durante il trasferimento dei dati, può collegare assieme buffer separati, col bus ISA richiede pochissimo tempo della CPU (praticamente nulla). La maggior parte dei driver per Linux in bus-mastering usano lo schema 'copybreak' nel quale i grossi pacchetti vengono messi direttamente dalla scheda nel buffer di rete del kernel, e i pacchetti piccoli vengono invece copiati dalla CPU che li carica in cache per le elaborazioni successive.

Contro:(Applicabile solamente alle schede ISA) Per la scheda sono necessari buffer in memoria bassa e un canale di 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 per scheda madre mal progettati hanno problemi con il bus-mastering di schede ISA.

7.2 Implicazioni della Larghezza di Bus per le Prestazioni

Il bus ISA può raggiungere i 5.3MB/sec (42Mb/sec), il che sembra più che adeguato per una Ethernet a 10Mbps. Nel caso di schede a 100Mbps, chiaramente è necessario un bus più veloce per sfruttare la larghezza di banda della rete.

7.2.1 Schede ISA a 8 e 16 Bit

Vi risulterà probabilmente difficile riuscire a comprare una scheda Ethernet per bus ISA di questi tempi, ma potreste anche riuscire a recuperare una scheda obsoleta o un fondo di magazzino per scopi di networking casalingo. Se proprio volete cedere al fascino del passato, potete persino utilizzare una vecchia scheda a ISA a mezzo slot (8 bit), ma siete avvertiti che la maggior parte di queste schede sono per 10Base-2.

Alcune schede a 8 bit che sono in grado di darvi prestazioni adeguate per traffici leggeri o medi sono la wd8003, la 3c503 e la ne1000. La 3c501 ha delle prestazioni disastrose, e questi poveri reperti dei tempi dell'XT, vecchi di 15 anni, dovrebbero essere evitati - Inviateli invece a chi li colleziona.

La banda passante del bus a 8 bit non limita le prestazioni più di tanto, tanto che potete aspettarvi di ottenere dai 500 agli 800kB/s in un download FTP con una scheda wd8003 installata su di un bus ISA veloce e su di una macchina (relativamente) veloce. Se il vostro traffico è diretto altrove, allora il collo di bottiglia della connessione sarà da un altra parte e l'unica differenza di velocità che voi noterete riguarderà il vostro traffico locale.

7.2.2 Schede Ethernet a 32 Bit per bus PCI (e VLB/EISA)

Ovviamente una interfaccia a 32 bit verso il computer è un requisito per reti a 100Mbps e oltre. Se state pensando di adottare GigE, allora i 133 megabyte al secondo del bus PCI costituiranno ancora il vostro limite.

Ma un più vecchio network a 10Mbps non ha veramente bisogno di una interfaccia a 32 bit. Si veda la sezione 7.1 (I/O programmato, memoria condivisa...) sul perché avere una scheda Ethernet da 10Mbps montata su di un bus ISA a 8Mhz non costituisca in realtà un limite. Anche se l'avere una scheda lenta su di un bus veloce non sarà all'origine di trasferimenti di dati più veloci, solitamente essa causerà meno carico sulla CPU, il che è sempre un bene su sistemi multiutente.

7.3 Impatto sulle prestazioni di Zero Copy

Come i dati vengono ricevuti o inviati, voi potete facilmente immaginarveli mentre essi vengono copiati dall'applicazione nella memoria del kernel e da quest'ultima nella memoria della scheda. Tutto questo movimento di dati richiede tempo e risorse. Come accennato in precedenza nella sezione sul Bus Mastering DMA, una scheda ben disegnata può ridurre tutto questo copiar di dati, e il caso ideale sarebbe ovviamente Zero Copy. Su alcune schede PCI moderne, Zero Copy diventa possibile semplicemente indicando alla scheda dove si trovano i dati ed essenzialmente dicendogli prenditeli da te. Se si desiderano massime prestazioni in condizioni di poco traffico, allora si controlli che sia l'hardware che il driver supportino funzionamento in Zero Copy.

7.4 Impatto sulle Prestazioni dei Checksum in Hardware

Non c'è garanzia che i vostri dati viaggeranno dal computer A al computer B senza venire corrotti. Per essere sicuri che i dati siano inalterati, il mittente somma tutti i numeri che costituiscono i dati da inviare, e allega a questi il checksum calcolato. Il destinatario ricalcola questo checksum e controlla che coincida con quello ricevuto insieme ai dati. Se i due non corrispondono, il destinatario sa che i dati sono stati corrotti e scarterà il pacchetto ricevuto e relativi dati alterati.

Calcolare queste somme richiede tempo e costituisce un ulteriore carico sulla CPU del computer. Alcune delle migliori schede hanno circuiti in hardware per calcolare i checksum di ricezione e di trasmissione senza disturbare il processore principale.

Quelle schede che richiedono copia dei dati non traggono gran vantaggio dai checksum in hardware, dato che l'operazione di somma può essere combinata con l'operazione di copia con una minima differenza di lavoro per il processore. Di conseguenza, checksum di trasmissione in hardware vengono usati solo in situazioni di zero copy (come ad esempio un applicazione che chiama `sendfile()`), e checksum hardware di ricezione sono al momento attuale ben più utili.

Si noti che un computer di prestazioni ragionevoli può saturare una connessione 100BaseT anche se deve calcolare i checksum in CPU e compiere operazioni di copia, per cui zero copy e checksum hardware risultano comunque solamente in un ridotto uso della CPU, si dovrebbe esaminare uno scenario GigE per vedere un aumento di prestazioni in rete.

7.5 Impatto sulle Prestazioni del NAPI (Rx interrupt mitigation)

Quando una scheda riceve un pacchetto dalla rete, solitamente la scheda richiama l'attenzione del processore con un interrupt. La CPU determina che dispositivo ha causato l'interrupt e quindi esegue l'handler d'interrupt incluso nel driver della scheda, che a sua volta rileva lo stato di interrupt della scheda per determinare che cosa la scheda volesse e, in questo caso, usa la funzione di ricezione dati inclusa nel driver e (finalmente), termina.

Adesso immaginatevi di stare continuamente ricevendo un sacco di dati, diciamo diecimila pacchetti al secondo, su un qualche server. Potete immaginarvi che la rincorsa di richieste di servizio per interrupt analoghe a quanto appena descritto causi un certo overhead, ed infatti un sacco di tempo del processore può essere facilmente risparmiato semplicemente sospendendo la generazione di interrupt e rimanendo nella funzione di ricezione del driver, dato che più o meno sa di dover servire un flusso di lavoro costante. Quest'idea sostanzialmente costituisce NAPI.

A partire dai kernel 2.6, alcuni driver hanno un opzione di configurazione per abilitare NAPI. Un poco di documentazione al riguardo è inclusa nella directory `Documentation/networking` del kernel.

8 Miscellanea

Tutta quella roba che non stava bene da nessun'altra parte è stata cacciata qui. Potrebbe non essere rilevante e potrebbe non essere d'interesse generale, ma è comunque qui.

8.1 Buffer FIFO di Trasmissione ed Errori di Underrun

Donald ha scritto una bella descrizione di che cosa sia un buffer FIFO di trasmissione e di cosa succede quando si verifica un errore. Eccola a voi:

Nei casi in cui l'hardware lo supporta, i miei driver includono codice per ottimizzare il comportamento della FIFO di trasmissione. Un tipico chip Ethernet ha una FIFO di trasmissione che contiene i dati arrivati dal bus prima che questi vengano trasmessi sul cavo. La maniera in cui questa FIFO viene gestita è importante per le prestazioni.

Idealmente, si dovrebbe voler iniziare a trasmettere non appena il primo pacchetto in trasmissione arriva sul chip. Il Tx FIFO threshold è un parametro che specifica di iniziare a trasmettere quando N byte sono arrivati al NIC. Inizialmente, questo parametro è settato per una configurazione tipica, ma se una scheda video o un controller SCSI stanno causando dei prolungati picchi di traffico PCI, il chip NIC esaurirà i dati nel suo buffer di trasmissione prima di poter accedere di nuovo al bus, e questo causa una FIFO underrun.

Il driver risponde ad una FIFO underrun incrementando il Tx FIFO threshold, e se questo succede un numero sufficiente di volte il chip finirà per operare in modalità store-and-forward, vale a dire che la trasmissione di un pacchetto non avrà inizio sino a quando esso non sia stato completamente trasferito al NIC.

Alcuni progetti, come l'Adaptec Starfire, compiono un passo ulteriore e forniscono un segnale che la FIFO ha quasi esaurito i dati. Questo permette al driver di configurare questo settaggio senza rischiare un errore di trasmissione.

Dovrebbe essere piuttosto raro l'incontrare più di uno o due FIFO underrun: o il chip ha un settaggio del Tx FIFO threshold che permette poche scelte oppure il driver aumenta questo valore in grossi passi per mantenere i picchi di traffico PCI contenuti dai loro limiti naturali.

8.2 Passare al kernel argomenti Ethernet

Ci sono due comandi generici che possono essere passati al kernel al momento del boot: `ether` e `reserve`. Si può far questo con LILO, loadlin o qualsiasi altra utilità di boot che accetti 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`

Questi parametri di boot possono essere resi permanenti cosicché non si debba reinserirli ogni volta. Solitamente questo richiede solamente l'inserimento di una riga come `append=blah=123,456,789` all'inizio del vostro `/etc/lilo.conf`. Si veda la documentazione di LILO per ulteriori dettagli.

Per maggiori informazioni sui parametri di boot (e un elenco completo) si veda il

BootPrompt-HOWTO `http://metalab.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html` .

8.2.1 Il parametro `ether`

Il parametro `ether=` viene usato in congiunzione con i driver direttamente compilati nel kernel. *Non ha assolutamente alcun effetto* su driver modulari. Nella sua forma più generale, può somigliare a qualcosa come quel che segue:

```
ether=IRQ,IND_BASE,PARAM_1,PARAM_2,NOME
```

Tutti gli argomenti sono opzionali. Il primo argomento non numerico è interpretato come NOME.

IRQ: ovvio. Un valore di '0' (solitamente il valore predefinito) indica di usare autoIRQ. È accidentale che l'impostazione dell'IRQ sia prima di quella dell'ind_base: questo verrà corretto non appena cambia anche qualcos'altro.

IND_BASE: altrettanto ovvio. Il valore '0' (solitamente quello predefinito) indica di rilevare una scheda nell'elenco di indirizzi specifici 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 riferimento 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 per LILO. Il kernel 1.0 trattava le schede Ethernet basate su LANCE in modo speciale. Gli argomenti di LILO venivano sempre 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).

8.2.2 Il comando `reserve`

Questo comando di lilo viene usato proprio come il precedente 'ether=', ovvero viene 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 controllare la presenza di dispositivi (autoprobing) in regioni specifiche. Ciò può essere dovuto ad hardware mal progettato che causa il *blocco* del boot (come alcune schede Ethernet), ad hardware che viene mal identificato, ad hardware il cui stato è stato modificato da un rilevamento precedente, o semplicemente ad hardware che non vuole essere inizializzato dal kernel.

L'argomento di boot `reserve` indirizza questo problema specificando una regione di porte di I/O che non deve 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 a meno che non si sia riscontrato un problema (o di avere a che fare con un caso particolare).

Le porte I/O nella regione specificata vengono protette dai controlli per la ricerca dei dispositivi. Questo si è reso necessario per gestire i casi in cui alcuni driver si piantavano su di una NE2000 o erroneamente identificavano altri dispositivi come propri. Un driver di dispositivo correttamente scritto non dovrebbe controllare una regione riservata a meno che qualche altro argomento di boot non specifici esplicitamente di far questo. 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 se un dispositivo è presente nella regione 0x300-0x31f.

Come al solito per i comandi di boot, c'è un limite di 11 parametri, quindi si possono specificare solo 5 regioni riservate per ciascun comando `reserve`. Possono essere specificati più `reserve` se si hanno richieste insolitamente complicate da descrivere.

8.3 Usare un driver Ethernet come modulo

La maggior parte delle distribuzioni di Linux utilizzano ora dei kernel con pochissimi driver al loro interno. I driver vengono ora 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 kernel 2.0) o 'kmod' (nei kernel 2.1) che a loro volta chiamano `modprobe`.

La vostra distribuzione può offrire un comodo strumento grafico per la configurazione dei moduli Ethernet. Se possibile prima si dovrebbe provare a usare quello. La descrizione che segue descrive che cosa sta dietro a qualsiasi programma di configurazione e indica cosa quei programmi vadano a modificare.

Le informazioni che indicano quali moduli vengono usati e quali opzioni vengono passate a ciascun modulo vengono solitamente salvate nel file `/etc/modules.conf`. 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 informazioni sui moduli.

I moduli stessi sono tipicamente collocati 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 li per vedere quale modulo corrisponda alla propria scheda.

La prima cosa che ci deve essere nel file `modules.conf` è una riga che indichi a `modprobe` quale driver usare per l'interfaccia di rete `eth0` (e per `eth1` e per...). Per fare questo si dovrà usare il comando `alias`. Per esempio, se si ha una scheda ISA EtherEZ della SMC che usa il driver modulare `smc-ultra.o`, si deve creare un `alias` per questo driver che corrisponda a `eth0`, aggiungendo la riga:

```
alias eth0 smc-ultra
```

Nota Importante: l'alias summenzionato viene usato solo dalle utilità per i moduli per convertire un nome di device generico (come ad esempio `eth0`) nel nome specifico del modulo del driver. Quando il driver viene caricato non vede mai questo alias: invece, esso si limita a scegliere il primo nome libero (`ethN` per N=0,1,2,...). Quindi, se più di un modulo Ethernet viene caricato, l'`ethN` assegnato al driver dal kernel può non corrispondere con quello assegnatogli dall'alias, a seconda dell'ordine in cui i moduli vengono caricati. Se dovete assicurarvi che una specifica scheda venga assegnata un indirizzo IP specifico, allora si legga l'indirizzo della hardware della scheda e si assegni l'indirizzo IP a seconda di quello. Se state scrivendo un vostro shell script per far questo, potete limitarvi a fare il parsing dell'output di ifconfig, altrimenti in C potete usare la chiamata

`ioctl(ethfd, SIOCGIFHWADDR, &ifreq)`.

L'altra cosa di cui si può aver bisogno è una riga `options` che indichi 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 vi avviserà (si veda `dmesg`) che l'auto rilevamento di una scheda ISA non è una buona idea. Per eliminare questo problema, si aggiunga un'altra riga che indica al modulo a quale indirizzo I/O base è collocata la scheda, ad esempio perl'indirizzo esadecimale `0x280` si usi

```
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 reperita nel file:

`/usr/src/linux/Documentation/networking/net-modules.txt`

Si raccomanda la sua lettura per scoprire quali opzioni si possano 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 unica copia del 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 transceiver 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 usare `ifconfig eth0 down` (per 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.

8.4 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) all'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`

8.5 Liberatoria e copyright (in originale inglese)

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.

8.6 Chiusura

Agli albori dell'era Linux (circa dieci anni fa), non esistevano molti driver e nemmeno molti utenti, e io avevo il tempo di seguire lo sviluppo dei singoli driver, leggere dei problemi comuni nelle newsgroup, e rispondere alle domande e alle e-mail. Ma le cose sono molto diverse ora: ci sono moltissimi driver, ed un numero ancora più grande di utenti,e non esiste maniera in cui io possa restare al corrente di tutti i nuovi sviluppi! Qui è dove ho bisogno del vostro aiuto: se avete trovato un nuovo driver che non viene menzionato qui, se avete trovato qualche errore madornale o informazione non aggiornata, inviatemi una e-mail. È un grande documento ed è facile non accorgersi di qualcosa. Se avete inviato una mail per una modifica e questa non è stata inclusa nella versione successiva, non si esiti a scrivere ancora, in quanto potrebbe essersi persa tra la marea di SPAM e spazzatura che ricevo nella posta elettronica.

Grazie!

Paul Gortmaker, `p_gortmaker@yahoo.com`