Linux PPP HOWTO

Robert Hart, hartr@interweft.com.au v3.0, 31 Marzo 1997 — Traduzione v1.1, 4 Giugno 1998

Questo documento illustra come connettere sotto Linux il proprio PC a un server PPP, come usare PPP per collegare assieme due LAN e fornisce un metodo per configurare come server PPP il proprio computer con Linux. Il documento inoltre fornisce dell'aiuto per il debug di connessioni PPP non funzionanti. Traduzione di Giovanni Bortolozzo, *borto@pluto.linux.it*.

Indice

1 Introduzione **5**

1.1 Client e server . . . 6

1.2 Differenze tra le distribuzioni di Linux . . . 6

1.3 Programmi di configurazione del PPP specifici delle distribuzioni . . . 7

2 Numeri IP **7**

3 Scopi di questo documento **8**

3.1 Configurazione di un client PPP . . . 8

3.2 Collegare due LAN o una LAN a Internet usando PPP . . . 9

3.3 Configurazione di un server PPP . . . 9

3.4 L'uso di PPP su una connessione diretta null modem . . . 9

3.5 Questo documento al momento non tratta... . . . 9

4 Versioni del software trattate **10**

5 Altri documenti utili/importanti **10**

5.1 Utili mailing list su Linux . . . 11

6 Panoramica di quello che va fatto per usare come client il PPP **11**

6.1 Trovare/installare il software . . . 11

6.2 Compilare il supporto per il PPP nel kernel . . . 12

6.3 Informazioni da chiedere al proprio ISP . . . 12

6.4 Configurare il proprio modem e la porta seriale . . . 12

6.5 Configurare la risoluzione da nome a indirizzo (DNS) . . . 13

6.6 PPP e Privilegi di root . . . 13

6.7 Verificare i File del PPP della propria distribuzione e impostare le Opzioni del PPP . . . 13

6.8 Se il proprio server PPP usa PAP (Password Authentication Protocol) . . . 14

6.9 Connettersi a mano a un server PPP . . . 14

6.10 Automatizzare la propria connessione PPP . . . 14

6.11 Chiudere la connessione . . . 14
6.12 Se ci sono problemi . . . 14
6.13 Dopo che la connessione è stabilita . . . 15
6.14 Problemi con i servizi IP standard su una connessione PPP a numero IP dinamico . . . 15

7 Configurare il proprio kernel di Linux 15
7.1 Installare i sorgenti del kernel di Linux . . . 15
7.2 Riconoscere il proprio hardware . . . 16
7.3 Compilazione del kernel – il kernel di Linux 1.2.13 . . . 16
7.4 Compilazione del kernel – i kernel di Linux 1.3.x e 2.0.x . . . 17
7.5 Nota su PPP-2.2 e `/proc/net/dev` . . . 17
7.6 Considerazioni generali sulla configurazione del kernel per il PPP . . . 18

8 Quali informazioni si debbono conoscere sul server PPP 18

9 Configurare il proprio modem e la porta seriale 19
9.1 Una nota sulle porte seriali e le loro velocità . . . 20
9.2 Nomi delle Porte Seriali . . . 21
9.3 Configurare il proprio modem . . . 21
9.4 Note sul controllo del flusso seriale . . . 22
9.5 Verificare il proprio modem per la connessione . . . 22

10 Configurare la risoluzione da nome a indirizzo (DNS) 24
10.1 Il file `/etc/resolv.conf` . . . 24
10.2 Il file `/etc/host.conf` . . . 25

11 L'uso di PPP e i privilegi di root 25

12 Impostare i file di connessione PPP 26
12.1 Il file di opzioni tipo options.tpl . . . 27
12.2 Quali opzioni dovrei usare? (No PAP/CHAP) . . . 33

13 Se il proprio server usa PAP (Password Authentication Protocol) 34
13.1 Usare MSCHAP . . . 35
13.2 Il file secrets di PAP/CHAP . . . 35
13.3 Il file secrets di PAP . . . 36
13.4 Il file secrets di CHAP . . . 36
13.5 Gestione di connessioni multiple autentificate con PAP . . . 37

14 Instaurare manualmente una connessione PPP 37

15 Automatizzare la propria connessione e creazione degli script 40

15.1 Script di connessione per l'autentificazione con Nome utente/Password . . . 41

15.2 Lo script ppp-on . . . 41

15.3 Modificare gli script forniti per l'avvio del PPP . . . 43

15.3.1 Lo script ppp-on . . . 43

15.3.2 Lo script ppp-on-dialer . . . 43

15.4 Il significato di uno script di conversazione (chat)... . . . 44

15.4.1 L'avvio di PPP dal lato server . . . 45

15.5 Uno script di conversazione per le connessioni autentificate con PAP/CHAP . . . 46

15.6 Le opzioni "`debug`" e "`file file_opzioni`" di pppd . . . 46

16 Verificare i propri script di connessione 47

17 Chiudere la connessione PPP 49

18 Risoluzione dei problemi 49

18.1 Ho compilato il supporto per il PPP nel kernel, ma... . . . 49

18.1.1 Si è fatto il boot con il kernel giusto? . . . 49

18.1.2 Si è compilato il supporto per il ppp come modulo? . . . 49

18.1.3 Si sta usando la versione corretta di PPP per il proprio kernel? . . . 50

18.1.4 Si sta eseguendo pppd come root? . . . 50

18.2 Il mio modem si connette ma il ppp non parte mai . . . 50

18.3 Il log di sistema dice "`serial line is not 8 bit clean`" . . . 50

18.3.1 Non ci si è loggati correttamente nel server . . . 50

18.3.2 Non si è avviato il PPP nel server . . . 51

18.3.3 Il processo PPP remoto è lento a partire . . . 51

18.4 Instradamento predefinito non impostato (`Default route not set`) . . . 51

18.5 Altri problemi . . . 52

19 Ottenere aiuto quando ormai non si sa più cosa provare 52

20 Problemi comuni una volta che il collegamento funziona 52

20.1 Non riesco a vedere niente oltre al server PPP al quale mi connetto . . . 52

20.2 Posso spedire email, ma non posso riceverla . . . 53

20.3 Perché la gente non può fare finger, WWW, gopher, talk nella mia macchina? . . . 53

21 Usare i servizi Internet con numeri IP dinamici 53

21.1 Configurazione dell'email . . . 54

21.2 Configurare un name server locale . . . 54

22 Connettere assieme due reti usando PPP **55**

22.1 Impostare i numeri IP 55

22.2 Impostare l'instradamento 55

22.3 Sicurezza di rete 56

23 Dopo che la connessione è attiva: lo script `/etc/ppp/ip-up` **56**

23.1 Instradamenti speciali 57

23.2 Gestione delle code di posta 57

23.3 Un esempio di script `/etc/ppp/ip-up` 57

23.4 Gestione delle email 58

24 Usare `/etc/ppp/ip-down` **59**

25 Problematiche di instradamento su una LAN **59**

25.1 Note sulla sicurezza 60

26 Configurare un server PPP **61**

26.1 Compilazione del kernel 61

26.2 Panoramica del sistema server 61

26.3 Procurarsi il software che serve 62

26.4 Configurare un accesso standard (accesso shell) 62

26.5 Impostare i file delle opzioni di PPP 62

26.6 Impostare pppd per permettere agli utenti di eseguirlo (con successo) 63

26.7 Impostare un alias globale per pppd 63

27 Usare PPP su una una connessione null modem (seriale diretta) **64**

Copyright

Questo documento è distribuito sotto i termini della GPL (GNU Public License).

Distribuzione

Questo documento sarà postato su comp.os.linux.answers appena ne sarà prodotta una nuova versione. È disponibile anche in formato HTML a:

- *Linux Howto Index* <http://sunsite.unc.edu/mdw/linux.html##howto>
- *PPP-HOWTO* <http://www.interweft.com.au/other/ppp-howto/ppp-howto.html>

Altri formati (SGML, ASCII, PostScript, DVI) sono disponibili da *Howtos - other formats* <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats>.

Poiché sunsite.unc.edu supporta un carico già molto elevato, si prega di usare il mirror più vicino.

Ringraziamenti

Un crescente numero di persone mi ha fornito assistenza nella preparazione di questo documento. Un ringraziamento particolare va a Al Longyear per la guida sul PPP stesso (se qui ci sono errori sono miei

e non suoi), a Greg Hankins (il manutentore del sistema di HOWTO di Linux) e a Debi Tackett (della MaximumAccess.com) per molti suggerimenti utili sullo stile, contenuto, ordine, logica e chiarezza delle spiegazioni.

Infine, a quanti mi hanno contattato via email offrendomi commenti, vanno i miei ringraziamenti. Come per tutti gli autori di HOWTO, la soddisfazione di aiutare è tutto il compenso che riceviamo ed è abbastanza. Scrivendo questo HOWTO sto ripagando in minima parte il debito che io - e tutti gli altri utenti di Linux - abbiamo verso quanti scrivono e mantengono il SO da noi scelto.

Note sulla traduzione

La traduzione di questo documento è opera mia e di un momento di pazzia: non mi ero reso conto che fosse così lungo :-).

L'ultima versione, insomma quella che ha la pretesa di essere la più aggiornata e corretta, di questa traduzione la troverete sempre (spero) nei vari siti che fanno da mirror al progetto ILDP del Pluto (`<http://www.pluto.linux.it/ildp/>`).

Data la lunghezza di questo documento dentro ci saranno sicuramente errori, frasi incomprensibili, e tutto lo schifo che la mia mente, sfruttando gli ultimi neuroni buoni, può produrre quando è stanca. Quindi tutte le porcherie che troverete sono colpa mia e vi invito (anzi se potessi vi *obbligherei*) a farmele notare, a spedirmi correzioni, diff, consigli ecc.

Ringrazio in anticipo chiunque mi segnalerà tali errori, ed un ringraziamento particolare va a **Paolo Farinelli**, che gentilmente si è preso l'onere di rivedere tutta la mia traduzione e senza del quale vi trovereste fra le mani una inutile accozzaglia di cavolate ed errori. *Grazie Paolo! Avanzi una birra!* :)

A tutti, buona (lunga) lettura.

1 Introduzione

Il PPP (Point to Point Protocol) è un meccanismo per creare ed usare IP (Internet Protocol) ed altri protocolli di rete su di una linea seriale - sia essa una connessione seriale diretta (usando un cavo null-modem) su di una connessione stabilita con telnet, oppure una connessione fatta tramite modem e linee telefoniche (e usando naturalmente anche linee digitali come ISDN).

Usando PPP, si può connettere il proprio PC Linux ad un server PPP ed accedere alle risorse della rete alla quale è connesso il server (quasi) come si fosse connessi direttamente a quella rete.

Inoltre si può configurare il proprio PC Linux come server PPP, cosicché gli altri computer possano collegarsi, effettuando una richiesta di connessione (dial in), al vostro computer ed accedere alle risorse di quest'ultimo e/o della rete a cui è connesso.

Poiché PPP è un sistema peer-to-peer ("da pari a pari"), si può usare PPP su due PC Linux per connettere assieme due reti (o una rete locale con Internet), creando una Wide Area Network (WAN).

Una delle differenze sostanziali tra PPP e una connessione Ethernet è ovviamente la velocità - una connessione Ethernet standard opera teoricamente con una velocità di trasferimento dati (throughput) massima di 10 Mbs (Mega - milioni di bit al secondo), mentre un modem analogico opera a velocità sino a 56 kbps (kilo - migliaia di bit al secondo).

Inoltre, a seconda del tipo di connessione PPP, ci possono essere alcune limitazioni nell'uso di alcune applicazioni e servizi.

1.1 Client e server

PPP è in sostanza un protocollo **peer to peer**; non c'è (teoricamente) alcuna differenza tra la macchina che fa la chiamata (dial) e la macchina alla quale questa è fatta. Comunque, per amor di chiarezza, è utile pensare in termini di **server** e **client**.

Quando si fa la chiamata in un sito per stabilire la connessione PPP, si è un **client**. La macchina a cui ci si connette è il **server**.

Quando si configura una macchina Linux per ricevere e gestire connessioni PPP in ingresso, si sta configurando un **server** PPP.

Qualsiasi PC Linux può essere server e client PPP, anche contemporaneamente se si ha più di una porta seriale (e modem se necessario). Perciò, per come è stato pensato il PPP, non c'è differenza tra client e server una volta effettuato il collegamento tra le due macchine.

Questo documento fa riferimento alla macchina che inizia la chiamata (che fa il dial in) come il **CLIENT**, mentre la macchina che risponde al telefono, verifica l'autenticità della richiesta di dial in (usando nome utente, password e probabilmente altri meccanismi) sarà indicata come il **SERVER**.

L'uso del PPP come un client per collegare una o più macchine in Internet è, probabilmente, quello a cui molti sono interessati.

La procedura descritta in questo documento permetterà di stabilire e automatizzare la propria connessione a Internet.

Questo documento guiderà anche nella configurazione di un PC Linux come `server` PPP e nel collegamento di due LAN (con instradamento – routing – completo) usando PPP (cosa che viene frequentemente caratterizzata come costituzione di un collegamento WAN – wide area network).

1.2 Differenze tra le distribuzioni di Linux

Ci sono molte distribuzioni di Linux e tutte hanno le proprie idiosincrasie e metodi per fare le cose.

In particolare, esistono due metodi diversi seguendo uno dei quali una macchina Linux (e Unix) viene praticamente avviata, configura le proprie interfacce e così via.

Esistono sostanzialmente il **sistema di inizializzazione BSD** e il **sistema di inizializzazione System V**. Se si gira per qualche newsgroup di Unix, si troveranno spesso guerre di religione tra i sostenitori dei due sistemi. Se questo tipo di cose vi solleticano, divertitevi a sprecare banda e partecipate!

In pratica le distribuzioni usate maggiormente sono

- Slackware
  che usa il sistema di inizializzazione BSD
- Red Hat (e la sua ex associata Caldera)
  che usa il sistema di inizializzazione SysV (sebbene in una forma leggermente modificata)
- Debian
  che usa il sistema di inizializzazione SysV

La inizializzazione in stile BSD tipicamente mantiene i propri file di inizializzazione in `/etc/...` e questi sono:

```
/etc/rc
/etc/rc.local
```

```
/etc/rc.serial
        (e possibilmente qualche altro file)
```

Di recente, qualche sistema di inizializzazione BSD usa una directory `/etc/rc.d...` per conservare i file si inizializzazione invece di mettere tutto in `/etc`.

L'inizializzazione System V mantiene i propri file di inizializzazione nelle directory sotto `/etc/...` o `/etc/rc.d/...` e in diverse sotto directory dentro a queste:

```
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 init.d
-rwxr-xr-x   1 root     root         1776 Feb  9 05:01 rc
-rwxr-xr-x   1 root     root          820 Jan  2  1996 rc.local
-rwxr-xr-x   1 root     root         2567 Jul  5 20:30 rc.sysinit
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc0.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc1.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc2.d
drwxr-xr-x   2 root     root         1024 Jul 18 18:07 rc3.d
drwxr-xr-x   2 root     root         1024 May 27  1995 rc4.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc5.d
drwxr-xr-x   2 root     root         1024 Jul  6 15:12 rc6.d
```

Se si deve provare a scoprire dove vengono effettivamente configurati la propria interfaccia Ethernet e l'instradamento di rete associato, si dovranno esaminare questi file per trovare i comandi che lo fanno.

1.3 Programmi di configurazione del PPP specifici delle distribuzioni

In alcune installazioni (per esempio Red Hat e Caldera), c'è un sistema di configurazione di PPP sotto X Window. Questo HOWTO non copre tali strumenti specifici delle distribuzioni. Se si hanno problemi con questi si contatti direttamente il distributore!

Per gli utenti di Red Hat 4.x, ora c'è un *Red Hat PPP-TIP* <http://www.interweft.com.au> nell'area delle risorse per Linux e anche a *Red Hat Software* <http://www.redhat.com> nell'area del supporto.

2 Numeri IP

Qualsiasi dispositivo che è connesso ad Internet deve avere il proprio ed unico numero IP. Questi sono assegnati centralmente da un'autorità designata in ogni paese.

Se si connette una rete locale (LAN) ad Internet, si **DEVE** usare per ognuno dei computer e dei dispositivi presenti nella propria LAN, un numero IP compreso nel range assegnato. **NON SI DEVONO** prendere numeri a caso e usarli per connettersi ad un'altra LAN (let alone the Internet). Alla meno peggio semplicemente non funzionerà affatto e si può causare un disastro totale in quanto i numeri IP 'rubati' interferiranno con le comunicazione di un'altro computer che sta già usando i numeri IP che ci si è inventati.

Si noti che i numeri IP usati in questo documento (con alcune eccezioni) sono fra la serie dei 'numeri di reti non connesse' che sono riservati per l'uso in reti che non sono (né mai saranno) connesse a Internet.

Questi numeri IP sono dedicati in particolare a LAN che non si connettono a Internet. Queste sono le sequenze di numeri IP:

- Un Indirizzo di Rete di Classe A
  10.0.0.0 (netmask 255.0.0.0)

- 16 Indirizzi di Rete di Classe B
  172.16.0.0 - 172.31.0.0 (netmask 255.255.0.0)
- 256 Indirizzi di Rete di Classe C Class
  192.168.0.0 - 192.168.255.0 (netmask 255.255.255.0)

Se si ha una LAN per la quale l'autorità responsabile del proprio paese **non** ha allocato numeri IP, per le proprie macchine si dovrebbe usare uno dei numeri di rete nelle sequenze suddette.

Questi numeri non dovrebbero **mai** essere usati su Internet.

Comunque, possono essere usati per la Ethernet locale su una macchina che è connessa a Internet. Questo perché i numeri IP in realtà sono allocati su un'interfaccia di rete, non ad un computer. Così, mentre la propria interfaccia Ethernet può usare 10.0.0.1 (per esempio), quando ci si aggancia ad Internet usando PPP, alla propria interfaccia PPP sarà assegnato un altro (e valido) numero IP dal server. Il proprio PC sarà connesso ad Internet mentre non lo saranno gli altri computer nella propria LAN.

Comunque, usando Linux e le funzionalità dell'IP Masquerade (noto anche come NAT – Network Address Translation) di Linux e il software ipfwadm, si può connettere la propria LAN ad Internet (con alcune restrizioni di servizi), anche se non si hanno numeri IP validi per le macchine nella propria Ethernet.

Per maggiori informazioni su come far ciò si veda l'IP Masquerade mini-HOWTO a *Linux IP Masquerade mini HOWTO* `<http://sunsite.unc.edu/mdw/HOWTO/mini/IP-Masquerade>`

Alla maggior parte degli utenti, che sono connessi con una macchina ad un fornitore di servizi Internet tramite PPP, non sarà necessario ottenere un numero IP (o meglio un numero di rete).

Se si vuole connettere una piccola rete LAN ad Internet, molti Internet Service Provider (ISP) possono fornire una sottorete dedicata (una sequenza specifica di numeri IP) presa dal loro spazio di indirizzi IP (ovvero dalla serie di indirizzi IP a loro assegnati o da loro richiesti). In alternativa si usi l'IP Masquerading.

Per gli utenti, che si connettono con un PC ad Internet tramite un ISP, molti provider usano l'assegnamento **dinamico** dei numeri IP. Cioè, come parte del processo di connessione, il servizio PPP che si contatta dirà alla propria macchina quale numero IP usare per l'interfaccia PPP durante la sessione corrente. Questo numero non sarà lo stesso ogni volta che ci si connette al proprio ISP.

Con i numeri IP dinamici, **non** si ottiene lo stesso numero IP ogni volta che ci si connette. Ciò ha conseguenze per le applicazioni di tipo server sulla propria macchina Linux come sendmail, ftpd, http e così via. Questi servizi si basano sul presupposto che il computer che offre il servizio sia accessibile tutto il tempo allo stesso numero IP (o almeno allo stesso fully qualified domain name - FQDN – e che sia disponibile la risoluzione DNS del nome in indirizzo IP).

Le limitazioni del servizio dovute all'assegnamento dinamico del numero IP (e modi per aggirarle, dove possibile) sono discusse successivamente nel documento.

3 Scopi di questo documento

3.1 Configurazione di un client PPP

Questo documento fornisce una guida per chi intenda usare Linux e il PPP per la connessione ad un server PPP e la creazione di una connessione IP usando PPP. Assume che il PPP sia stato compilato ed installato nella propria macchina Linux (ma brevemente ripassa la riconfigurazione/ricompilazione del kernel per includerne il supporto).

Sebbene DIP (il metodo standard per creare una connessione SLIP) possa essere usato per impostare una

connessione PPP, gli script DIP sono generalmente piuttosto complessi. Per questa ragione questo documento NON tratta l'uso di DIP per impostare una connessione PPP.

Piuttosto questo documento descrive il software PPP standard di Linux (chat/pppd).

3.2 Collegare due LAN o una LAN a Internet usando PPP

Questo documento fornisce le informazioni (di base) per collegare due LAN o una LAN ad Internet usando PPP.

3.3 Configurazione di un server PPP

Questo documento fornisce una guida su come configurare il proprio PC Linux come server PPP (permettendo ad altri di connettervisi e stabilire una connessione).

Si dovrebbe notare che c'è una miriade di modi per configurare Linux come un server PPP. Questo documento illustra un metodo usato dall'autore per impostare diversi piccoli server PPP (ognuno di 16 modem).

Questo metodo funziona, comunque non è necessariamente il migliore.

3.4 L'uso di PPP su una connessione diretta null modem

Questo documento fornisce una breve panoramica sull'uso di PPP per collegare due PC tramite un cavo null modem. In questo modo è anche possibile connettere altri SO a Linux. Per farlo si deve consultare la documentazione per il sistema operativo che interessa.

3.5 Questo documento al momento non tratta...

- La compilazione del demone PPP
  Si veda la documentazione distribuita assieme con la versione di pppd che si usa.
- La connessione e la configurazione di un modem con Linux (in dettaglio)
  Si veda il Serial-HOWTO per l'inizializzazione specifica del modem, e si veda *Modem Setup Information* `<http://www.in.net/info/modems/index.html>` per informazioni che possono essere d'aiuto nella configurazione del proprio modem.
- L'uso di DIP per stabilire connessioni PPP
  Si usi chat invece...
- L'uso di sock o di IP Masquerade
  Ci sono già documenti molto buoni che coprono questi due pacchetti.
- L'uso di `diald` per instaurare automaticamente la connessione
  Si veda la documentazione di `diald` per maggiori informazioni.
- L'uso di EQL per usare due modem assieme in uno stesso collegamento PPP.
- Metodi di connessione PPP specifici delle distribuzioni (come il network configuration tool di Red Hat 4.x).
  Si veda la distribuzione per la documentazione sul metodo usato.
- Il crescente numero di programmi disponibili per automatizzare la configurazione di PPP
  Si veda la documentazione appropriata.

4 Versioni del software trattate

Questo HOWTO assume che si usi un kernel di Linux 1.2.x con il software di PPP 2.1.2 oppure Linux 1.3.X/2.0.x e PPP 2.2.

Al momento della stesura di questo HOWTO, l'ultima versione ufficiale disponibile di PPP per Linux era ppp-2.2f. La nuova versione (ppp-2.3) è ancora in beta.

È possibile usare PPP 2.2.0 con kernel 1.2.13. Ciò richiede delle patch per il kernel. Si raccomanda agli utenti del kernel versione 1.2.13 di installare il ppp-2.2 in quanto corregge diversi bug.

Si dovrebbe notare, in particolare, che non è possibile usare il software PPP 2.1.2 con i kernel di Linux versione 2.0.x

Si noti che questo documento **NON** discute i problemi causati dall'uso dei moduli caricabili del kernel 2.0.x. Si veda il kerneld mini-HOWTO e la documentazione kernel/module del 2.0.x (nei sorgenti di Linux 2.0.x in `/usr/src/linux/Documentation/...`).

Poiché questo documento è pensato per assistere i nuovi utenti, si raccomanda fortemente di usare la versione del kernel di Linux e la giusta versione di PPP, che siano notoriamente stabili assieme.

5 Altri documenti utili/importanti

Gli utenti sono invitati a leggere:

- la documentazione distribuita con il pacchetto PPP;
- le pagine di manuale di chat e pppd;
  (si usi `man chat` e `man pppd` per vederle)
- la Linux Network Administration Guide (NAG);
  si veda *The Network Administrators' Guide* <http://sunsite.unc.edu/mdw/LDP-books/nag-1.0/nag.html>
- il Net-2/3 HOWTO;
  si veda *Linux NET-2/3-HOWTO* <http://sunsite.unc.edu/mdw/HOWTO/NET-2-HOWTO.html>
- La documentazione del kernel di Linux installata in `/usr/src/linux/Documentation` quando si installa il codice sorgente di Linux;
- La modem setup information page - si veda *Modem Setup Information* <http://www.in.net/info/modems/index.html>
- Gli eccellenti libri su Unix/Linux pubblicati da O'Reilly and Associates. Si veda *O'Reilly and Associates On-Line Catalogue* <http://www.ora.com/>. Se si è nuovi a Unix/Linux, si **corra** (non si cammini) al più vicino negozio di libri di computer e se ne comprerino un po'!
- La PPP-FAQ mantenuta da Al Longyear, disponibile da *Linux PPP-FAQ* <ftp://sunsite.unc.edu/pub/Linux/docs/faqs>.
  Questa contiene un sacco di informazioni utili nel formato domanda/risposta molto utili quando si vuole capire perché il PPP non funziona correttamente.
- Il crescente numero di libri su Linux di diverse case editrici ed autori;
  Si suggerisce di controllare l'attualità di tali libri. Lo sviluppo di Linux e le distribuzioni tendono ad evolvere abbastanza rapidamente, mentre la revisione di libri si muove (solitamente) molto più

lentamente! Comperare un libro eccellente (e ce ne sono tanti) che ora sia obsoleto causerà considerevole confusione e frustrazione ai nuovi utenti.

In generale il miglior punto di partenza per la documentazione su Linux è *The Linux Documentation Project Home Page* <http://sunsite.unc.edu/mdw/>. La tendenza è di revisionare abbastanza regolarmente gli HOWTO.

Sebbene si possa usare questo documento per creare la propria connessione PPP senza leggere nessuno dei suddetti documenti, se li si legge si capirà molto di più di quello che si andrà a fare! Si sarà inoltre in grado di risolvere da soli i problemi (o almeno si porranno domande un pelino più intelligenti nei newsgroup comp.os.linux.* e nelle mailing list di Linux).

Questi documenti (oltre ad altri, tra cui gli RFC sull'argomento) forniscono spiegazioni aggiuntive e più dettagliate di quanto possa offrire questo HOWTO.

Se si connette una LAN ad Internet usando PPP, si avrà bisogno di una ragionevole conoscenza sul networking TCP/IP. In aggiunta ai suddetti documenti, un notevole beneficio lo si potrà trarre dai libri della O'Reilly "TCP/IP Network Administration" e "Building Internet Firewalls"!

5.1 Utili mailing list su Linux

Ci sono molte mailing list su Linux che operano come mezzo di comunicazione tra gli utenti con diversi livelli di abilità. In ogni modo si sottoscrivano quelle che più interessano e si collabori fornendo la propria esperienza e visione della cosa.

Un paio di regolette: alcune liste sono specificatamente destinate agli utenti di più alto livello ("high powered") e/o a particolari problematiche/argomenti. Mentre nessuno si preoccupa di chi assiste passivamente alle discussioni (ovvero fa il 'lurk', seguendo le discussioni ma non postando mai messaggi), è facile invece ricevere commenti alquanto sgarbati (se non una serie completa di flame) nel caso in cui si postasse un messaggio da principiante ("newbie") liste non appropriate.

Ciò non è perché gli utenti guru odino i nuovi utenti, ma perché queste liste esistono per discutere argomenti particolari a particolari livelli di difficoltà.

Ad ogni modo si sottoscrivano le liste aperte, ma si mantengono i propri commenti su argomenti inerenti alla lista!

Un buon punto di partenza per le mailing list su Linux è *Linux Mailing List Directory* <http://summer.snu.ac.kr/~djshin/linux/mail-list/index.shtml>

6 Panoramica di quello che va fatto per usare come client il PPP

Questo documento contiene un sacco di informazioni che aumentano ad ogni versione dello stesso!

Di conseguenza, questo paragrafo intende fornire una concisa panoramica delle azioni da fare per far sì che il proprio sistema Linux possa connettersi come client ad un server PPP.

6.1 Trovare/installare il software

Se la propria distribuzione Linux non contiene il software per il PPP, lo si può scaricare da *il demone PPP di Linux* <ftp://sunsite.unc.edu/pub/Linux/system/Network/serial/ppp/ppp-2.2.0f.tar.gz>.

Al momento questa è l'ultima versione ufficiale. Comunque, si scelga l'ultima versione disponibile da questo sito (ppp-2.3 era in versione beta quando stavo scrivendo questo documento e dovrebbe essere rilasciato tra poco).

Il pacchetto PPP contiene le istruzioni su come compilare ed installare il software, **quindi questo HOWTO non lo spiegherà**!

6.2 Compilare il supporto per il PPP nel kernel

Il PPP in Linux si compone di due parti

- il demone PPP menzionato prima
- il supporto per il PPP nel kernel

Molte distribuzioni forniscono di default il supporto a livello kernel per il PPP nei loro kernel di installazione, ma altre non lo fanno.

Se al boot il proprio kernel mostra un messaggio simile a

```
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.
```

il proprio kernel ha il supporto per il PPP compilato al suo interno.

Detto questo, probabilmente si vorrà comunque compilare il proprio kernel qualsiasi sia la propria distribuzione, in modo da ottenerne uno che usi in modo più efficiente le risorse del sistema data la propria configurazione hardware. Non è male ricordare che il kernel deve risiedere completamente in memoria principale (non può mai essere swappato fuori da essa) e quindi mantenere il kernel più piccolo possibile può essere solo un vantaggio nelle macchine con poca memoria.

Questo documento fornisce le istruzioni essenziali per ricompilare il kernel nel paragrafo 7 (Configurare il proprio Kernel di Linux).

Per maggiori dettagli, si veda il Kernel-HOWTO a *The Linux Kernel HOWTO* <http://sunsite.unc.edu/mdw/HOWTO/Kernel-HOWTO.html>

6.3 Informazioni da chiedere al proprio ISP

Il server PPP può essere configurato in un'infinità di modi. Per potersi connettere al proprio ISP (o al server PPP della propria organizzazione per accedere alla propria intranet), si debbono ottenere un po' di informazioni sul funzionamento del server PPP.

Poiché si sta usando Linux, si possono incontrare alcune difficoltà con il supporto utenti di alcuni ISP (e del server PPP per la intranet al lavoro) che sono a conoscenza solo di client MS Windows.

Comunque, un numero sempre crescente di ISP usano Linux per fornire il loro servizio - e Linux sta penetrando pure nell'ambiente aziendale, perciò, se si è fortunati non si incotreranno resistenze.

Il paragrafo 8 (Quali informazioni si debbono conoscere sul server PPP) dirà quello che si deve sapere sul server PPP al quale ci si vuole connettere e come trovare le informazioni che si debbono conoscere.

6.4 Configurare il proprio modem e la porta seriale

Per potersi connettere ad un server PPP e ottenere la velocità di trasferimento migliore possibile, si deve configurare correttamente il proprio modem.

Similmente, si devono configurare correttamente le porte seriali sui propri modem e sul computer.

Il paragrafo 9 (Configurare il proprio modem e la porta seriale) fornisce informazioni in proposito.

6.5 Configurare la risoluzione da nome a indirizzo (DNS)

Oltre ai file che lanciano PPP ed effettuano il login automatico nel server PPP, c'è un certo numero di file di configurazione che devono essere impostati per permettere al proprio computer di risolvere ("convertire") nomi come `www.interweft.com.au` negli indirizzi IP in effetti usati per contattare quel computer. Questi sono:

- `/etc/resolv.conf`
- `/etc/host.conf`

Si veda il paragrafo 10 (Configurare la Risuluzione da Nome ad Indirizzo) per i dettagli.

In particolare, **NON** è necessario che si faccia girare un name server nel proprio PC Linux per potersi connettere ad Internet (sebbene lo si possa comunque fare). Tutto quel che si deve conoscere è l'indirizzo IP di almeno un name server che si possa usare (preferibilmente quello del proprio ISP).

6.6 PPP e Privilegi di root

Lo stabilire un collegamento PPP tra la propria macchina Linux ed un altro server PPP richiede la manipolazione dei dispositivi di rete ("network device") (l'interfaccia PPP è un'interfaccia di rete) e della tabella di instradamento del kernel ("kernel routing table"), e quindi pppd necessita dei privilegi di root.

Per i dettagli si veda la sezione 11 (L'uso di PPP e i privililegi di root).

6.7 Verificare i File del PPP della propria distribuzione e impostare le Opzioni del PPP

Ci sono alcuni file di configurazione che debbono essere impostati per rendere operativo il PPP. Nella distribuzione di PPP ci sono alcuni esempi e questo paragrafo mostra quali file si dovrebbe avere:

```
/etc/ppp/options
/etc/ppp/scripts/ppp-on
/etc/ppp/scripts/ppp-on-dialer
/etc/ppp/options.tpl
```

Può essere necessario creare alcuni altri file a seconda di cosa esattamente si ha intenzione di fare con PPP:

```
/etc/ppp/options.ttyXX
/etc/ppp/ip-up
/etc/ppp/pap-secrets
/etc/ppp/chap-secrets
```

Inoltre, il demone PPP può usare un numero enorme di opzioni in riga di comando ed è importante usare quelle giuste; questa sezione quindi fa una panoramica sulle opzioni standard del PPP e aiuta nella scelta delle opzioni che andrebbero usate.

Per i dettagli si veda 12 (Impostare i file di connessione del PPP).

6.8 Se il proprio server PPP usa PAP (Password Authentication Protocol)

Molti server PPP di ISP ed aziende usano PAP. Se il proprio server **non** richiede che si usi PAP (se si può fare manualmente login e si riceve la richiesta normale di nome e password in modo testo, il server non usa PAP) si può tranquillamente saltare questa sezione.

Invece di fare il login nel server usando il nome utente e la password quando richieste per entrare nel server, un server PPP che usa PAP non richiede un login in modo testo.

Le informazioni per l'autentificazione dell'utente vengono piuttosto scambiante come parte del protocollo di controllo di connessione ("link control protocol" - LCP) che costituisce la prima parte dell'instaurazione del collegamento PPP.

Il paragrafo 13 (Se il proprio server PPP usa PAP (Password Authentication Protocol)) fornisce informazioni sui file da impostare per stabilire una connessione PPP usando PAP.

6.9 Connettersi a mano a un server PPP

Dopo aver impostato i file di base, è una buona idea verificarli connettendosi a mano (usando minicom o seyon) e lanciando pppd sulla propria macchina Linux.

Si veda la sezione 14 (Instaurare manualmente una connessione PPP) per tutti i dettagli.

6.10 Automatizzare la propria connessione PPP

Una volta in grado di connettersi a mano, ci si può concentrare sulla creazione di un insieme di script che automatizzeranno l'instaurazione della connessione.

La sezione 15 (Automatizzare la propria connessione - Creazione degli script di connessione) copre la configurazione degli script necessari, concentrandosi particolarmente su `chat` e sullo script di login sul server PPP.

Questa sezione discute gli script per l'autentificazione con nome utente e password così come gli script per i server con autentificazione PAP/CHAP.

6.11 Chiudere la connessione

Una volta che la connessione è stabilita e funzionante, si deve essere in grado di deattivarla.

Ciò è spiegato nel paragrafo 17 (Chiudere la connessione PPP).

6.12 Se ci sono problemi

Molti hanno problemi a far funzionare direttamente il PPP. Le variazioni nei server PPP e in come questi richiedono sia configurata la connessione sono enormi. Similmente, ci sono molte opzioni per PPP e addirittura alcune combinazioni di queste non funzionano assieme.

Oltre ai problemi di login e avvio del servizio PPP, ci sono problemi sia con i modem che con le linee telefoniche!

Il paragrafo 18 (Risoluzione dei problemi) fornisce alcune informazioni basilari sugli errori comuni, e su come isolarli e correggerli.

Lo scopo di tale paragrafo altro **NON** è se non quello di fornire le informazioni basilari. Al Longyear mantiene la PPP-FAQ che contiene molte più informazioni su questo argormento!

6.13 Dopo che la connessione è stabilita

Una volta che la connessione PPP è operativa (precisamente, almeno il livello IP è operativo), il PPP di Linux può eseguire automaticamente uno script (come root) per effettuare **qualsiasi** funzione che si può scrivere in uno script.

Il paragrafo 23 (Dopo che la connessione è attiva) fornisce informazioni sullo script `/etc/ppp/ip-up`, i parametri che riceve da PPP e su come usarlo per fare operazioni come scaricare la posta dal proprio provider, spedire qualsiasi email nella coda di attesa di trasmissione nella propria macchina e così via.

6.14 Problemi con i servizi IP standard su una connessione PPP a numero IP dinamico

Come notato nell'introduzione, i numeri IP dinamici hanno conseguenze sulla capacità del proprio PC Linux di funzionare come server in Internet.

Il paragrafo 21 (Problemi con i servizi IP standard su una connessione PPP a numero IP dinamico) fornisce informazioni sui (principali) servizi che ne risentono e su cosa si può fare (se qualcosa è possibile) per ovviare a questi problemi.

7 Configurare il proprio kernel di Linux

Per poter usare PPP, il proprio kernel di Linux dev'essere compilato includendone il supporto. Ci si procuri il codice sorgente del kernel di Linux se già non lo si ha - dovrebbe essere il `/usr/src/linux` in un file system standard di Linux.

Si dia un'occhiata a questa directory - molte distribuzioni di Linux installano l'albero dei sorgenti (i file e le sottodirectory) durante il processo di installazione.

Al boot, il proprio kernel di Linux mostra svariate informazioni. Tra queste ci sono anche quelle sul PPP, se il kernel ne include il supporto. Per vedere queste informazioni, si veda il file di log del proprio sistema o si usi `dmesg |less` per visualizzare le informazioni sullo schermo. Se il proprio kernel include il supporto per il PPP, si dovrebbe vedere una cosa di questo genere:

```
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.
```

(questa è per la serie 2.0.x di kernel di Linux).

I sorgenti del kernel di Linux possono essere reperiti via ftp da `sunsite.unc.edu` o da uno dei suoi (molti) mirror.

7.1 Installare i sorgenti del kernel di Linux

Quanto segue sono le istruzioni in breve per procurarsi ed installare i sorgenti del kernel di Linux. Informazioni più complete le si può trovare nel *Linux Kernel HOWTO* <`http://sunsite.unc.edu/mdw/HOWTO/Kernel-HOWTO.html`>.

Per poter installare e compilare il kernel di Linux, si deve essere loggati come root.

1. Passare alla directory `/usr/src`
   `cd /usr/src`

2. Controllare in `/usr/src/linux` se già ci sono installati i sorgenti.

3. Se non si hanno i sorgenti, li si prenda dalla *directory dei sorgenti del kernel di Linux* `<ftp://sunsite.unc.edu/pub/Linux/kernel/v2.0>` o nel mirror più vicino.
   Se si cerca una delle prime versione del kernel (come la 1.2.X), queste sono conservate nella *directory dei sorgenti dei vecchi kernel di Linux* `<ftp://sunsite.unc.edu/pub/Linux/kernel/old>`.

4. Si scelga il kernel appropriato, solitamente il più recente fra quelli disponibili nel sito. Lo si scarichi e si metta il file tar dei sorgenti in `/usr/src`.
   Nota: un file 'tar' è un archivio – probabilmente compresso (come lo sono i file tar dei sorgenti del kernel di Linux) – che contiene molti file in diverse directory. È l'equivalente Linux di un file zip multidirectory di DOS.

5. Se già si hanno i sorgenti di Linux installati si debbono rimuovere i vecchi sorgenti. Si usi il comando
   `rm -rf /usr/src/linux`

6. Ora si decomprimano e si estraggano i sorgenti usando il comando
   `tar xzf linux-2.0.XX.tar.gz`

7. Adesso, si faccia `cd /usr/src/linux` e si legga il file README. Questi contiene un'eccellente spiegazione su come configurare e compilare un nuovo kernel. Lo si legga (può essere una buona idea stamparlo per averne una copia sotto mano durante la compilazione, finché non la si è fatta abbastanza volte da conoscerne ogni minimo dettaglio).

7.2 Riconoscere il proprio hardware

Se si ha intenzione di ricompilare il kernel si **DEVE** sapere quali schede/dispositivi ci sono installati nel proprio PC!!! Per alcuni dispositivi (come le schede audio) ci sarà bisono di conoscere anche le diverse impostazioni (come IRQ, indirizzi di I/O e così via).

7.3 Compilazione del kernel – il kernel di Linux 1.2.13

Per iniziare il processo di configurazione, si seguano le istruzione nel file README installato con i sorgenti. La configurazione viene avviata con

```
make config
```

Per poter usare PPP, si deve configurare il kernel per includere il supporto per il PPP (PPP ha bisogno sia di pppd sia del supporto nel kernel).

```
PPP (point-to-point) support (CONFIG_PPP) [n] y
```

Si risponda alle altre domande del make config a seconda dell'hardware nel proprio PC e delle caratteristiche che si vogliono dal sistema operativo Linux. Si continui poi nella lettura del README per compilare ed installare il nuovo kernel.

Il kernel 1.2.13 crea solo 4 device PPP. Nel caso si possieda una scheda seriale multiporta, ci sarà bisogno di modificare i sorgenti del PPP nel kernel per avere più porte (si veda il file README.linux distribuito assieme al PPP-2.1.2 per maggiori dettagli sulle semplici modifiche da fare).

Nota: il processo di configurazione dell'1.2.13 NON permettere di tornare indietro – se si fa un'errore rispondendo ad una domanda durante il dialogo con `make config`, si esca premendo CTRL C e si ricominci.

7.4 Compilazione del kernel – i kernel di Linux 1.3.x e 2.0.x

Per Linux 1.3.x e 2.0.x, si può usare un processo simile a quello di Linux 1.2.13. Ancora, si seguano le istruzioni del file README installato con i sorgenti. Il processo di configurazione si avvia con

```
make config
```

Comunque, c'è anche la possibilità di usare

```
make menuconfig
```

Ciò fornisce un sistema di configurazione a menu con un help in linea che permette di spostarsi avanti e indietro nel processo di configurazione.

Esiste inoltre un'interfaccia di configurazione altamente consigliata che gira in X Window

```
make xconfig
```

Il supporto per il PPP può essere compilato direttamente dentro al kernel oppure come modulo.

Se quando la propria macchina Linux è operativa, si usa PPP solo per brevi periodi, allora si raccomanda di compilare il supporto per il PPP come modulo. Usando 'kerneld' il proprio kernel caricherà automaticamente i moduli richiesti per fornire il supporto per il PPP quando si lancia il processo di connessione con PPP. Ciò fa risparmiare porzioni importanti di memoria: nessuna parte del kernel può essere messa in swap, ma i moduli caricabili sono automaticamente rimossi se non sono in uso.

Per far ciò si deve abilitare il supporto per i moduli caricabili:

```
Enable loadable module support (CONFIG_MODULES) [Y/n/?] y
```

Per aggiungere il supporto nel kernel per il PPP, si risponda alla seguente domanda:

```
PPP (point-to-point) support (CONFIG_PPP) [M/n/y/?]
```

Per il modulo caricabile del PPP, si risponda **M**, altrimenti per compilare il PPP dentro al kernel rispondere **Y**.

Diversamente dal kernel 1.2.13, il kernel 2.0.x crea device PPP al volo quando sono necessari e non c'è più bisogno di modificare i sorgenti per incrementare il numero di dispositivi PPP disponibili.

7.5 Nota su PPP-2.2 e `/proc/net/dev`

Se si usa PPP-2.2, si scoprirà che un effetto collaterale nella creazione al volo dei dispositivi PPP è che nessun dispositivo viene mostrato nel file system `/proc/net` finché il dispositivo non è creato lanciando pppd:

```
[hartr@archenland hartr]$ cat /proc/net/dev
Inter-|   Receive                  |  Transmit
 face |packets errs drop fifo frame|packets errs drop fifo colls carrier
    lo:  92792    0    0    0    0    92792    0    0    0     0    0
  eth0: 621737   13   13    0   23   501621    0    0    0  1309    0
```

Una volta avviati uno (o più) servizi ppp, si vedranno delle voci così (da un server ppp):

```
[root@kepler /root]# cat /proc/net/dev
Inter-|   Receive                   |  Transmit
 face |packets errs drop fifo frame|packets errs drop fifo colls carrier
    lo: 428021    0    0    0    0   428021    0    0    0     0    0
  eth0:4788257  648  648  319  650  1423836    0    0    0  4623    5
  ppp0:   2103    3    3    0    0     2017    0    0    0     0    0
  ppp1:  10008    0    0    0    0     8782    0    0    0     0    0
  ppp2:    305    0    0    0    0      297    0    0    0     0    0
  ppp3:   6720    7    7    0    0     7498    0    0    0     0    0
  ppp4: 118231  725  725    0    0   117791    0    0    0     0    0
  ppp5:  38915    5    5    0    0    28309    0    0    0     0    0
```

7.6 Considerazioni generali sulla configurazione del kernel per il PPP

Se si sta mettendo su un PC Linux per far da server PPP, si deve compilare il supporto per l'IP forwarding. Questo è pure necessario se si vuole usare Linux per collegare alcune LAN assieme o per connettere la propria LAN ad Internet.

Se si connette una LAN ad Internet (o si connettono assieme due LAN), si dovrebbe essere ben coscienti delle problematiche sulla sicurezza. L'aggiunta del supporto per i firewall IP nel kernel probabilmente è un DOVERE!

Questo sarà necessario anche se si vuole usare IP masquerade per connettere una LAN che usa uno dei suddetti numeri di rete IP 'non connessi'.

Per abilitare l'IP Masquerade e l'IP fire walling, si **DEVE** rispondere affermativamente alla prima domanda nel processo `make config`:

```
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL)?
```

Ai nuovi utenti questo potrebbe sembrare un po' fuori luogo, ma molti utenti Linux usano attivamente l'IP Masquerade e l'IP Firewalling dei kernel Linux 2.0.XX senza alcun problema.

Una volta installato il nuovo kernel e riavviata la macchina, si può cominciare a configurare e a testare il(i) collegamento(i) PPP.

8 Quali informazioni si debbono conoscere sul server PPP

Prima di poter stabilire una connessione PPP con un server, è necessario procurarsi le seguenti informazioni (ci si rivolga al supporto clienti o agli amministratori del proprio server PPP):

- Il o i numeri telefonici da comporre per il servizio
  Se si è dietro ad un PABX, è necessario conoscere anche il numero PABX che abilita le chiamate esterne - solitamente è lo zero (0) o il nove (9).
- Il server usa numeri IP DINAMICI o STATICI?
  Se il server usa numeri IP STATICI, allora è necessario conoscere quale numero IP usare per la propria estremità della connessione IP. Se il proprio ISP vi fornisce una sottorete di numeri IP validi, bisogna conoscere anche quali numeri IP è possibile usare e qual'è la maschera di rete (netmask).

La maggior parte degli Internet Service Provider usa numeri IP DINAMICI. Come detto in precedenza, ciò ha qualche implicazione in termini di servizi che si possono usare.

Comunque, anche se si stanno usando numeri IP STATICI, molti server PPP non permetteranno mai (per ragioni di sicurezza) ad un client di specificare un numero IP, in quanto ciò è un rischioso. Si **deve** comunque avere questa informazione!

- Quali sono i numeri IP dei Domain Name Server dell'ISP?
  Dovrebbero essere almeno due sebbene ne serva solo uno.

  Ci potrebbe essere un problema a questo punto. La configurazione del PPP di MS Windows 95 permette che l'indirizzo del DNS sia passato al client come parte del processo di connessione. Quindi il proprio ISP potrebbe benissimo rispondere che non c'è la necessità di conoscere gli indirizzi dei server DNS.

  Per Linux, si **DEVE** conoscere l'indirizzo di almeno un DNS. L'implementazione del PPP in Linux non permette di impostare dinamicamente il numero IP del DNS durante la connessione, e probabilmente non lo farà mai nemmeno in futuro.

  Nota: mentre Linux (come client PPP) non può accettare l'indirizzo del DNS da un server, può, quando funziona come server, passare l'informazione ai client usando l'opzione `dns-addr` di pppd.

- Il server richiede l'uso di PAP/CHAP?
  Se questo è il caso di deve conoscere "id" e "secret" da usare durante la connessione (probabilmente queste sono il proprio nome utente e la propria password nel proprio ISP).

- Il server avvia automaticamente il PPP o si deve fornire un qualche comando per far partire il PPP una volta fatto il login?
  Se si deve dare un comando per avviare il PPP, qual è?

- Il server è un sistema Microsoft Windows NT e, se così è, usa il sistema MS PAP/CHAP?
  Molte LAN di aziende usano MS Windows NT in questo modo per aumentare la sicurezza.

Ci si annoti con cura queste informazioni, che tra un po' andremo ad usarle!

9 Configurare il proprio modem e la porta seriale

Ci si dovrebbe assicurare che il proprio modem sia impostato correttamente e conoscere a quale porta seriale è connesso.

Si ricorda:

- DOS com1: = Linux /dev/cua0 (e /dev/ttyS0)
- DOS com2: = Linux /dev/cua1 (e /dev/ttyS1)
  eccetera

Non è male ricordare che se si hanno 4 porte seriali, la configurazione predefinita dei PC è di avere la com1 e la com3 che condividono IRQ4 e la com2 e la com4 che condividono IRQ3.

Se si hanno dispositivi su una porta seriale standard che condivide un IRQ con il proprio modem, probabilmente si avranno dei problemi. Ci si deve assicurare che la porta seriale a cui è connesso il modem abbia un

IRQ di sua proprietà e non condiviso. Molte schede seriali moderne (e le porte seriali sulle schede madri di buona qualità) permettono di spostare l'IRQ delle porte seriali.

Se si usa un kernel versione 2 di Linux, si possono vedere quali IRQ sono in uso con `cat /proc/interrupts`, che produrrà un output simile a:

```
 0:    6766283   timer
 1:      91545   keyboard
 2:          0   cascade
 4:     156944 + serial
 7:     101764   WD8013
10:     134365 + BusLogic BT-958
13:          1   math error
15:    3671702 + serial
```

Qui è mostrata una porta seriale sull'IRQ4 (un mouse) ed una porta seriale sull'IRQ15 (la connessione PPP permanente via modem ad Internet). Ci sono anche le porte seriali com2 su IRQ3, e la com4 su IRQ14, ma non sono utilizzate e perciò non vengono mostrate.

Attenzione: bisogna sapere quello che si fa quando si gioca con gli IRQ! Non solo si deve aprire il proprio computer, tirare fuori le schede e giocare con i jumper, ma si deve pure sapere cosa c'è su un determinato IRQ. Nel mio caso ho un PC totalmente basato su SCSI, e così ho disabilitato l'interfaccia IDE sulla scheda madre che normalmente usano gli IRQ14 e 15!

Si dovrebbe tener presente che se il proprio PC si avvia anche con altri sistemi operativi, lo spostamento degli IRQ può far sì che tali sistemi operativi non si avviino correttamente, o che non si avviino affatto.

Se si spostano le porte seriali su IRQ non standard, allora bisogna dire a Linux qual è IRQ che ogni porta usa. Ciò è fatto usando `setserial` ed è meglio farlo durante la sequenza di avvio del sistema in `rc.local` o `rc.serial` che viene chiamato da `rc.local` oppure come parte dell'inizializzazione SysV. Per la macchina descritta in precendenza, i comandi usati sono:

```
/bin/setserial -b /dev/ttyS2 IRQ 11
/bin/setserial -b /dev/ttyS3 IRQ 15
```

Comunque, se si usa il modulo per le seriali e lo si fa caricare dinamicamente quando necessario da `kerneld`, non si possono impostare gli IRQ ed il resto e poi dimenticarsene una volta avviato il sistema, in quando se il modulo viene scaricato, Linux si dimentica delle impostazioni speciali.

Quindi, se si fa caricare su richiesta il modulo per le seriali, bisogna riconfigurare gli IRQ ed il resto ogni volta che viene caricato il modulo.

9.1 Una nota sulle porte seriali e le loro velocità

Se si usa un modem (esterno) ad alta velocità (14,400 Baud o superiore), la propria porta seriale dev'essere in grado di gestire il flusso di dati che un tale modem è in grado di produrre, in particolare quando il modem comprime i dati.

Ciò richiede che la propria porta seriale usi un UART (Universal Asynchronous Receiver Transmitter - Ricevitore/Trasmettitore Asincrono Universale) moderno come un 16550(A). Se si usa una vecchia macchina (o una vecchia scheda seriale), è abbastanza probabile che la propria porta seriale abbia solamente un UART 8250, che può causere considerevoli problemi quando utilizzata con un modem ad alta velocità.

Si usi il comando

```
setserial -a /dev/ttySx
```

per far sì che Linux dica il tipo di UART in proprio possesso. Se non si ha una UART del tipo 16550A, si investa un po' di soldi in una nuova scheda seriale (disponibile a meno di $50). Una volta comperata, ci si assicuri di poterne spostare gli IRQ!

Nota: la prima versione del chip UART 16550 ha un errore. Questo è stato scoperto quasi subito ed è stata rilasciata una revisione del chip: l'UART 16550A. Comunque circolano un numero relativamente basso di chip fallati. Non è facile incappare in uno di questi, ma comunque, almeno per le schede abbastanza pregiate, si dovrebbe ottenere una risposta al comando precedente che dice 16550A.

9.2 Nomi delle Porte Seriali

Storicamente, Linux usava i device `cuaX` per le chiamate in uscita (dial out) e i device `ttySx` per quelle in ingresso (dial in).

La porzione del kernel che richiedeva ciò è stata cambiata nella versione 2.0.x del kernel e si dovrebbero usare i `ttySx` sia per il dial in che per il dial out. Da quel che ho capito i device `cuaX` potrebbero benissimo scomparire in una versione futura del kernel.

9.3 Configurare il proprio modem

Per usare PPP bisogna configurare correttamente il proprio modem. Per far ciò **SI LEGGA IL MANULE DEL PROPRIO MODEM**! La maggior parte dei modem hanno delle **impostazioni predefinite dal costruttore** che selezionano le opzioni richieste per il PPP. La configurazione minima richiede:

- Controllo di flusso hardware (RTS/CTS) (`&K3` in molti modem Hayes)

Altre impostazioni (in comandi standard Hayes) che si dovrebbero controllare sono:

- E1 Echo ON (richiesto per il funzionamento di chat)
- Q0 Report result codes (richiesto per il funzionamento di chat)
- S0=0 Auto Answer OFF (a meno che non si voglia che il proprio modem risponda al telefono)
- `&C1` Carrier Detect ON solo dopo la connessione
- `&S0` Data Set Ready (DSR) sempre ON
- (forse) Data Terminal Ready

Esiste un sito che offre le impostazioni per un modem per una varietà in continua crescita di costruttori e modelli di modem a *Modem setup information* `<http://www.in.net/info/modems/index.html>` che può tornare utile in questa operazione.

Non è male neppure investigare sul funzionamento dell'interfaccia seriale del modem usata per connetterlo al proprio computer. La maggior parte dei modem permette di far funzionare tale interfaccia a velocità FISSA, mentre lasciano all'interfaccia con la linea telefonica la possibilità di variare la sua velocità in modo da usare quella più alta che sia lei che la corrispondente nel modem remoto riescono a gestire.

Ciò è noto come funzionamento "split speed". Se il proprio modem lo supporta, si blocchi l'interfaccia seriale del modem alla velocità più alta disponibile (generalmente 115200 baud ma più probabilmente 38400 baud per i modem a 14400).

Si usi il proprio software di comunicazione (p.es. minicom o seyon) per scovare la configurazione del proprio modem e per impostare quanto richiesto per il PPP. Molti modem ritornano le loro impostazioni correnti in risposta a AT&V, ma si consulti comunque il manuale del proprio modem.

Se si fa un casotto con le impostazioni, si può ritornare alle impostazioni sane (solitamente) dando il comando AT&F che riporta il modem alle impostazioni del costruttore (comunque per la maggior parte dei modem che ho incontrato, le impostazioni di fabbrica includono tutto quello che serve per il PPP - ma si dovrebbe comunque controllare)

Una volta scoperta la stringa di configurazione richiesta dal modem, la si annoti da qualche parte. Ora ci sono due possibilità: si possono immagazzinare queste impostazioni nella memoria non volatile del modem in modo da poter essere richiamate con un apposito comando AT, oppure si può passare le impostazioni corrette al modem come parte del processo di composizione del numero.

Se si usa il proprio modem solamente da Linux per chiamare il proprio ISP o server aziendale, la cosa più semplice sarà salvare la propria configurazione del modem nella RAM non volatile.

Se d'altra parte, il proprio modem è usato da altre applicazioni e sistemi operativi, è più sicuro passare le informazioni al modem ogni volta che viene fatta la chiamata cosicché è garantito che il modem è sempre nello stato corretto per la chiamata (ciò ha pure il vantaggio aggiuntivo di registrare la stringa di configurazione del modem nel caso il modem perda il contenuto della NV-RAM, che può sempre succedere).

9.4 Note sul controllo del flusso seriale

Quando i dati viaggiano su linee di comunicazione seriale, può capitare che i dati arrivino più velocemente di quanto il computer possa gestire (il computer potrebbe essere occupato a fare qualcos'altro – si ricordi che Linux è un sistema operativo multi-utente e multi-processo). Per poter assicurare che i dati non vengano persi (ovvero che i dati non superino la capacità del buffer d'ingresso e quindi vengano persi), sono necessari alcuni metodi per controllare il flusso di dati.

Nelle linee seriali ci sono due modi per farlo:

- Usando segnali hardware (Clear To Send/Request to Send – CTS/RTS)
- Usando segnali software (control S e control Q, noti anche come XON/XOFF).

Mentre l'ultimo può andare bene per la connessione di un terminale (testo), i dati in un collegamento PPP usano tutti gli 8 bit ed è abbastanza probabile che da qualche parte nei dati ci siano dei byte di dati che corrispondono a control S e control Q. Quindi, se il modem è impostato per il controllo di flusso software, le cose possono rapidamente andare in sfacelo!

Per le connessioni ad alta velocità che usano PPP (che usa dati a 8 bit) il controllo di flusso hardware è vitale ed è per questa ragione che si deve usare questo tipo di controllo di flusso.

9.5 Verificare il proprio modem per la connessione

Ora che si è configurato il proprio modem e la porta seriale, è una buona idea assicurarsi che queste impostazioni facciano realmente il loro lavoro, chiamando il proprio ISP e verificando di potersi connettere.

Usando un pacchetto di comunicazione (come minicom), si imposti la inizializzazione richiesta per il PPP e per chiamare il server PPP al quale ci si vuole connettere in una sessione PPP.

(Nota: a questo punto **NON** stiamo provando a fare una connessione PPP, ma solo verificando di avere il numero di telefono giusto e di scoprire **esattamente** cosa il server ci risponde per poter poi fare il login ed avviare il PPP).

Durante questo processo, si catturi (registrandolo su un file) l'intero processo di login oppure si annoti accuratamente (*molto accuratamente*) **esattamente** quello che il server risponde per far capire che è il momento di inserire il proprio nome utente e la propria password (e qualsiasi altro comando necessario per stabilire una connessione PPP).

Se il proprio server usa PAP, non si dovrebbe vedere una richiesta di login, ma sul proprio schermo si dovrebbe invece vedere la rappresentazione in modo testo del protocollo di controllo della connessione (che assomiglia tanto a della spazzatura).

Qualche avvertenza:

- alcuni server sono abbastanza intelligenti: si può fare il login usando nome/password in modo testo oppure usando PAP. Quindi se il proprio ISP o server aziendale usa PAP ma non si vede la spazzatura partire immediatamente, ciò non vuol dire che si è fatto qualcosa in modo errato;
- alcuni server richiedono che inizialmente si immetta un po' di testo e *poi* avviano una sequanza PAP standard;
- alcuni server PPP sono passivi, ovvero stanno lì semplicemente non spedendo niente finché il client che sta facendo il dialing non gli spedisce un pacchetto lcp valido. Se il server PPP a cui ci si sta connettendo opera in modo passivo, non si vedrà mai la spazzatura!
- alcuni server non avviano il PPP finché non si preme ENTER, quindi non è una cattiva idea provare a farlo se ci si è connessi correttamente ma non si vede la spazzatura!

Non è male connettersi almeno un paio di volte – alcuni server cambiano i loro prompt (p.es. con il passare del tempo!) ogni volta che ci si connette. I due prompt critici che la propria macchina Linux ha necessità di identificare ogni volta che si fa la connessione sono:

- il prompt che richiede di inserire il proprio nome utente;
- il prompt che richiede di inserire la propria password.

Se si deve lanciare un comando per avviare il PPP nel server, è necessario trovare anche il prompt che il server dà una volta che ci è loggati per dire che ora è possibile inserire il comando per lanciare il PPP.

Se il proprio server, una volta loggati, lancia automaticamente il PPP si inizierà a vedere un po' di spazzatura sullo schermo - queste sono le informazioni che il server sta spedendo per avviare e configurare la connessione PPP.

Dovrebbero essere qualcosa di questo tipo:

```
~y}#.!}!}!} }8}!}$}%U}"}&} } } } }%}& ...}'}"}(}"} .~~y}
```

(NdT: è questa la "spazzatura" a cui si faceva e si farà riferimento.)

Su alcuni sistemi il PPP deve essere esplicitamente avviato sul server. Questo di solito è perché il server è stato configurato per permettere sia il login in PPP che il login con una shell usando la stessa coppia nome utente/password. Se questo è il proprio caso, si inserisca il comando una volta effettuato il login. Anche in questo caso, si vedrà della spazzatura non appena viene avviato il lato server della connessione PPP.

Se non si vede la spazzatura immediatamente dopo la connessione (e nemmeno dopo il login e l'avvio manuale del server PPP, se richiesti) si prema **Enter** per vedere se almeno questo avvia il server PPP...

A questo punto, si può "riattaccare" (N.d.T.: nel senso della cornetta del telefono) il proprio modem (solitamente, battendo velocemente +++ e poi dando il comando ATHO una volta che il modem ha risposto con OK).

Se non si riesce a far andare il proprio modem, si legga il suo manuale, le pagine man del software di comunicazione e il Serial HOWTO! Una volta fatto questo, si rifaccia (per bene) quanto spiegato in precedenza.

10 Configurare la risoluzione da nome a indirizzo (DNS)

Mentre agli umani piace dare nomi alle cose, i computer preferiscono di molto i numeri. In una rete TCP/IP (che è quello che Internet è), chiamiamo le macchine con un nome particolare per ogni macchina che vive in un particolare "dominio". Per esempio, la mia workstation Linux si chiama **archenland** e risiede nel dominio **interweft.com.au**. Il suo indirizzo "umano" è quindi archenland.interweft.com.au (che è noto come il FQDN - fully qualified domain name).

Comunque, quando i computer comunicano attraverso Internet, affinché questa macchina possa essere trovata dagli altri computer in Internet, è in realtà nota con il suo numero IP.

La traduzione (risoluzione) del nome (e del dominio) della macchina nei numeri realmente usati su Internet è compito di macchine che offrono il servizio di Domain Name (Domain Name Service).

Ciò che succede è questo:

- la propria macchina ha bisogno di sapere l'indirizzo IP di un particolare computer. L'applicazione che richiede questa informazione chiede al 'risolutore' (resolver) nel proprio PC Linux di fornire questa informazione;
- il risolutore interroga il file locale degli host (`/etc/hosts`) e i domain name server di cui è a conoscenza (l'esatto comportamento del resolver è determinato da `/etc/host.conf`);
- se la risposta è trovata nel file degli host, ne viene ritornato il valore in risposta;
- se è specificato un domain name server, il proprio PC interroga questa macchina;
- se la macchina DNS sa già il numero IP per il nome richiesto, lo ritorna in risposta. Se così non è, interroga altri name server in Internet per trovare l'informazione. Il name server poi passa questa informazione al risolutore e questi all'applicazione che ne aveva fatto richiesta.

Quando si fa una connessione PPP, si deve dire alla propria macchina Linux dove andare a prendere le informazioni sulla conversione da nome ad indirizzo IP (risoluzione dell'indirizzo - address resolution) in modo che **noi** si possa usare i nomi delle macchine ma il nostro **computer** possa tradurli in numeri IP di cui a bisogno per fare il proprio lavoro.

Un modo è di inserire ogni host con il quale si vuole dialogare nel file `/etc/hosts` (il che è realisticamente totalmente impossibile se si è connessi ad Internet); un altro è di usare i numeri IP della macchine invece dei loro nomi (un esercizio di memoria impossibile tranne che per le LAN più piccole).

Il modo migliore è configurare Linux in modo che sappia dove andare a prendere automaticamente queste informazioni nome/numero. Questo servizio è fornito dai sistemi Domain Name Server (DNS). Tutto ciò che serve è di inserire il numero (o i numeri) IP dei domain name server nel proprio file /etc/resolv.conf.

10.1 Il file `/etc/resolv.conf`

L'amministatore del proprio server PPP o il servizio assistenza clienti dovrebbe aver fornito due numeri IP di DNS (ne è necessario solo uno, ma due danno un po' di ridondanza in caso di problemi in uno dei due).

Come menzionato in precedenza, Linux non può impostare il proprio name server nel modo che usa MS Windows 95. Quindi si **deve** insistere (mantenendo la calma) in modo che il proprio ISP fonisca questa informazione!

Il proprio `/etc/resolv.conf` potrebbe essere qualcosa di simile a:

```
domain nome.di.dominio.del.proprio.isp
nameserver 10.25.0.1
nameserver 10.25.1.2
```

Si modifichi questo file (creandolo se necessario) per metterci le informazioni fornite dal proprio ISP. Dovrebbe avere la proprietà e i permessi che seguono:

```
-rw-r--r--   1 root     root           73 Feb 19 01:46 /etc/resolv.conf
```

Se si è già configurato un `/etc/resolv.conf` perché si è in una LAN, semplicemente si aggiungano i numeri IP dei server DNS per il PPP al file che già esiste.

10.2 Il file `/etc/host.conf`

Si dovrebbe pure controllare che il proprio file `/etc/host.conf` sia configurato correttamente. Dovrebbe essere una cosa così:

```
order hosts,bind
multi on
```

Ciò dice al risolutore di usare le informazioni nel file degli host prima di interrogare un DNS per la risoluzione.

11 L'uso di PPP e i privilegi di root

Poiché PPP ha bisogno di impostare alcuni dispositivi di rete, modificare la tabella di instradamento del kernel ed altro, richiede i privilegi di root per farlo.

Se altri utenti oltre a root devono poter avviare una connessione PPP, il programma pppd dev'essere setuid root:

```
-rwsr-xr-x   1 root     root        95225 Jul 11 00:27 /usr/sbin/pppd
```

Se /usr/sbin/pppd non è impostato in questo modo, allora **come root** si digiti il comando:

```
chmod u+s /usr/sbin/pppd
```

Quello che fa è rendere pppd eseguibile con i privilegi di root **anche se** l'eseguibile è lanciato da un utente ordinario. Ciò permette agli utenti normali di lanciare pppd con i privilegi necessari per impostare le interfacce di rete e la tabella di instradamento del kernel.

I programmi che sono eseguiti come 'set uid root' potenzialmente sono dei buchi di sicurezza e si dovrebbe essere estremamente cauti nel fare programmi 'suid root'. Un certo numero di programmi (incluso pppd) sono stati scritti con tutte le attenzioni per minimizzare i pericoli conseguenti alla loro esecuzione in suid root, quindi dovrebbe essere sicuro farlo (ma non ci sono garanzie).

A seconda di come si vuole che il proprio sistema funzioni - specificatamente se si vuole che QUALSIASI utente nel proprio sistema sia in grado si avviare una connessione PPP, si dovrebbe rendere i propri script ppp-on/off leggibili ed eseguibili al mondo intero (probabilmente questo va bene solo se il proprio PC è usato SOLO da se stessi).

Comunque, se NON si vuole che qualsiasi utente sia in grado di avviare una connessione PPP (per esempio, i propri figli hanno un account sulla macchina Linux, ma non vuoi che vadano in giro per Internet senza la una supervisione), bisogna creare un gruppo per il PPP (come root, si modifichi il file /etc/group) e:

- si renda pppd suid root, di proprietà dell'utente root e del gruppo PPP, con gli altri permessi su questo file vuoti. Insomma dovrebbe avere questi permessi (o simili):

```
-rwsr-x---   1 root     PPP         95225 Jul 11 00:27 /usr/sbin/pppd
```

- si rendano gli script ppp-on/off di proprietà dell'utente root e del gruppo PPP;
- si rendano gli script ppp-on/off leggibili/eseguibili al gruppo PPP

```
-rwxr-x---   1 root     PPP           587 Mar 14  1995 /usr/sbin/ppp-on
-rwxr-x---   1 root     PPP           631 Mar 14  1995 /usr/sbin/ppp-off
```

- si rendano vuoti gli altri diritti di accesso a ppp-on/off;
- si aggiungano gli utenti che si vuole possano avviare il PPP al gruppo PPP in /etc/group.

Anche se si fa questo, gli utenti ordinari non saranno ANCORA in grado di buttare giù la connessione via software! L'esecuzione dello script `ppp-off` richiede i privilegi di root. Comunque, qualsiasi utente può semplicemente spegnere il modem (o disconnettere la linea telefonica da un modem interno).

Un alternativa (e metodo migliore) a questa impostazione è di usare il programma `sudo`. Questo offre maggiore sicurezza e permetterà di impostare le cose in modo tale che qualsiasi utente (autorizzato) possa attivare/disattivare la connessione usando gli script. Usando `sudo` si permetterà ad un utente autorizzato di attivare/disattivare la connnessione PPP in modo pulito e sicuro.

12 Impostare i file di connessione PPP

Adesso è necessario essere loggati come **root** per creare le directory e modificare i file necessari per impostare il PPP, anche se poi si vuole che il PPP sia accessibile a tutti gli utenti e non solo a root.

Il PPP usa un certo numero di file per connettersi e impostare la connessione PPP. Questi differiscono per nome e locazione in PPP 2.1.2 e 2.2.

Per il PPP 2.1.2 i file sono:

```
/usr/sbin/pppd          # l'eseguibile PPP
/usr/sbin/ppp-on        # lo script di composizione del numero e connessione
/usr/sbin/ppp-off       # lo script di disconnessione
/etc/ppp/options        # le opzioni che pppd usa per tutte le connessioni
/etc/ppp/options.ttyXX  # le opzioni specifiche per una connessione su
                        # questa porta
```

Per il PPP 2.2 i file sono:

```
/usr/sbin/pppd                  # l'eseguibile PPP
/etc/ppp/scripts/ppp-on # lo script di composizione del numero e connessione
/etc/ppp/scripts/ppp-on-dialer  # parte 1 dello script di composizione
                                # del numero
/etc/ppp/scripts/ppp-off        # il vero script di chat
/etc/ppp/options            # le opzioni che pppd usa per tutte le connessioni
/etc/ppp/options.ttyXX      # le opzioni specifiche per una connessione su
                            # questa porta
```

Gli utenti di Red Hat Linux dovrebbero notare che l'installazione predefinita di Red Hat 4.X mette questi script in `/usr/doc/ppp-2.2.0f-2/scripts`.

Nella propria directory /etc ci dovrebbe essere una directory ppp:

```
drwxrwxr-x   2 root     root         1024 Oct  9 11:01 ppp
```

Se non esiste, la si crei con questo proprietario e permessi.

Se la directory già esiste, dovrebbe contenere un file di opzioni tipo, chiamato **options.tpl**. Questo file è incluso qui sotto nel caso che non ci sia.

Lo si stampi in quanto contiene una spiegazione di praticamente tutte le opzioni per PPP (utile da leggere assieme alle pagine man di pppd). Sebbene si possa usare questo file come punto di partenza del proprio file `/etc/ppp/options` file, è probabilmente meglio creare un proprio file di opzioni che non contenga tutti i commenti, il quale sarà così molto più breve e più facile da leggere e mantenere.

Se si posseggono più linee seriali o modem (caso tipico di un server PPP), si crei un file `/etc/ppp/options` generale contenente le opzioni comuni a tutte le porte seriali sulle quali si vuole supportare il dial in/out e poi si crei un file di opzioni individuali per ognuna delle line seriali sulle quali si voglia poter stabilire una connessione PPP con le impostazioni specifiche richieste per ogni porta.

Questi file specifici per le porte si chiamano `options.ttyx1`, `options.ttyx2` e così via (dove x è la lettera appropriata per le proprie porte seriali).

Comunque, per una connessione PPP singola, si può semplicemente usare il file `/etc/ppp/options`. In alternativa si possono mettere tutte le opzioni come argomenti per il comando pppd stesso.

È più semplice mantenere una configurazione che usa i file `/etc/ppp/options.ttySx`. Se si usa PPP per connettersi ad un certo numero di siti diversi, si possono creare file di opzioni per ogni sito in `/etc/ppp/options.site` e poi specificare il file di opzioni come un parametro del comando PPP quando ci si connette (usando l'opzione `file file_opzioni` nella riga di comando di pppd).

12.1 Il file di opzioni tipo options.tpl

Alcune distribuzioni sembrano aver perso il file options.tpl, e quindi eccone qui una versione completa. Suggerisco di NON modificare questo file per creare il proprio (o i propri) file `/etc/ppp/options`. Piuttosto, lo si copi su un nuovo file e poi si modifichi quello. Se si fa casino con le modifiche è sempre possibile tornare a quello originale e ricominciare.

```
# /etc/ppp/options -*- sh -*- opzioni generali per pppd
# created 13-Jul-1995 jmk
# autodate: 01-Aug-1995
# autotime: 19:45
```

```
# Usa l'eseguibile o il comando shell specificato per impostare la
# linea seriale. Questo script solitamente userebbe il programma
# "chat" per dialogare con il modem ed avviare la sessione ppp
# remota.
#connect "echo Bisogna installare un comando connect."

# Avvia l'eseguibile o il comando shell specificato dopo che pppd ha
# concluso la connessione, Questo script potrebbe, ad esempio, inviare
# comandi al modem per fargli riagganciare se non sono disponibili i
# segnali di controllo hardware del modem.
#disconnect "chat -- \d+++\d\c OK ath0 OK"

# mappa di caratteri asincrona -- esadecimale a 32-bit; ogni bit è un
# carattere del quale è necessario fare l'escape quando pppd lo
# riceve. 0x00000001 rappresenta '\x01', e 0x80000000 rappresenta
# '\x1f'.
#asyncmap 0

# Richiede che il peer si autentifichi da solo prima di permettere la
# trasmissione o la ricezione di pacchetti di rete.
#auth

# Usa il controllo di flusso hardware (i.e. RTS/CTS) per controllare
# il flusso di dati sulla porta seriale.
#crtscts

# Usa il controllo di flusso software (i.e. XON/XOFF) per controllare
# il flusso di dati sulla porta seriale.
#xonxoff

# Aggiunge, quando la negoziazione IPCP è completata con successo, un
# instradamento predefinito nella tabella di instradamento di sistema,
# usando il peer come gateway. Questa voce viene automaticamente
# rimossa quando la connessione PPP è interrotta.
#defaultroute

# Specifica che di certi caratteri dovrebbe essere fatto l'escape in
# trasmissione (indipendentemente dal fatto il peer richieda tale
# operazione con la sua mappa di caratteri di controllo
# asincrona). Questi caratteri di cui dev'essere fatto l'escape sono
# specificati come una lista di numeri esadecimali separati da
# virgole. Si noti che può essere specificato praticamente qualsiasi
# carattere, mentre l'opzione asyncmap permette di specificare solo
# caratteri di controllo. I caratteri dei quali non è possibile fare
# l'escape sono quelli con valore esadecimale 0x20 - 0x3f o 0x5e.
#escape 11,13,ff

# Non usa le linee di controllo del modem.
#local
```

```
# Specifica che pppd dovrebbe usare un lock in stile UUCP sul
# dispositivo seriale per assicurare l'accesso esclusivo al
# dispositivo.
#lock

# Usa le linee di controllo del modem.  In Ultrix, questa opzione
# implica il controllo hardware del flusso come l'opzione
# crtscts. (Questa opzione non è completamente implementata.)
#modem

# Imposta il valore di MRU [Maximum Receive Unit - Unità Massima di
# Ricezione] a <n> per la negoziazione. pppd chiederà al peer di
# inviare pacchetti di non più di <n> byte. Il minimo valore di MRU è
# 128. Il valore predefinito è 1500. Un valore di 296 è raccomandato
# per connessioni lente (40 byte per l'header TCP/IP + 256 byte di
# dati).
#mru 542

# Imposta la netmask per l'interfaccia a <n>, una netmask a 32 bit in
# notazione "decimal dot" (p.es. 255.255.255.0).
#netmask 255.255.255.0

# Disabilita il comportamento predefinito quando non è specificato
# nessun indirizzo IP locale, che è di determinare (se possibile)
# l'indirizzo IP locale dal nome dell'host.  Con questa opzione, il
# peer deve fornire un indirizzo IP locale durante le negoziazione
# IPCP (a meno che non sia specificato esplicitamente nella riga di
# comando o nel file delle opzioni).
#noipdefault

# Abilita le opzioni "passive" nell'LCP. Con questa opzione, pppd
# proverà ad iniziare la connessione; se non riceve risposta dal peer,
# allora pppd attenderà passivamente un pacchetto LCP valido da parte
# del peer (invece di uscire, come farebbe senza questa opzione).
#passive

# Con questa opzione, pppd non trasmetterà pacchetti LCP per iniziare
# una connessione finché non riceve un pacchetto LCP valido dal peer
# (come per l'opzione "passive" per le vecchie versioni di pppd).
#silent

# Non richiede o permette la negoziazione di qualsiasi opzione per LCP
# o IPCP (usa i valori predefiniti).
#-all

# Disabilita la negoziazione di compressione Indirizzo/Controllo (usa
# quanto predefinito, ovvero campi address/control disabilitati).
#-ac

# Disabilita la negoziazione asyncmap (usa la asyncmap predefinita,
```

```
# i.e. fa l'escape di tutti i caratteri di controllo).
#-am

# Non fa il fork per diventare un processo in background (altrimenti
# pppd lo farebbe se è specificato un dispositivo seriale).
#-detach

# Disabilita la negoziazione dell'indirizzo IP (con questa opzione
# l'indirizzo IP remoto deve essere specificato con un'opzione nella
# riga di comando o nel file delle opzioni).
#-ip

# Disabilita la negoziazione magic number. Con questa opzione pppd non
# può rilevare una linea cortocircuitata (looped-back).
#-mn

# Disabilita la negoziazione MRU [Maximum Receive Unit] (usa il valore
# predefinito, i.e. 1500).
#-mru

# Disabilita la negoziazione della compressione del campo protocollo
# ("protocol field compression") (usa quanto predefinito,
# i.e. compressione del campo protocollo disabilitata).
#-pc

# Richiede che il peer si autentifichi da solo usando il PAP. Questo
# richiede un'autentificazione a DUE VIE - NON usare per una
# connessione standard con autentificazione PAP ad un ISP, in quanto
# richiede che anche la macchina dell'ISP si autentifichi da sola
# nella propria macchina (e non sarà in grado di farlo).
#+pap

# Non accetta di autentificare usando PAP.
#-pap

# Richiede che il peer di autentifichi da solo usando
# l'autentificazione CHAP [Cryptographic Handshake Authentication
# Protocol]. Questo richiede un'autentificazione a DUE VIE - NON
# usare per una connessione standard con autentificazione CHAP ad un
# ISP, in quanto richiede che anche la macchina dell'ISP si
# autentifichi da sola nella propria macchina (e non sarà in grado di
# farlo).
#+chap

# Non accetta di autentificare usando CHAP.
#-chap

# Disabilita la negoziazione della compressione dell'header IP in
# stile Van Jacobson (usa quanto predefinito, i.e. nessuna compressione).
#-vj
```

```
# Incrementa il livello di debug (analoga -d).  Se è data questa
# opzione, pppd registrerà il contenuto di tutti i pacchetti spediti e
# ricevuti in un formato leggibile.  I pacchetti sono registrati
# usando syslog con funzionalità daemon e debug. Queste informazioni
# possono essere redirette ad un file impostando opportunamente il
# file /etc/syslog.conf (vedere syslog.conf(5)) (se pppd è compilato
# abilitando l'extra debugging, registrerà i messaggi usando le
# funzionalità local2 invece di daemon).
#debug

# Aggiunge il nome del dominio <d> al nome dell'host locali per scopi
# di autentificazione.  Per esempio, se gethostname() ritorna il nome
# porsche, ma il FQDN (Fully Qualified Domain Name) è
# porsche.Quotron.COM, si dovrebbe usare l'opzione domain per
# impostare il nome dominio a Quotron.COM.
#domain <d>

# Abilita il codice di debug nel driver del PPP a livello kernel.
# L'argomento n è un numero che è la somma dei seguenti valori: 1 per
# abilitare i messaggi di debug generici, 2 per richiedere la
# visualizzazione del contenuto dei pacchetti ricevuti, e 4 per
# visualizzare quello dei pacchetti trasmessi.
#kdebug n

# Imposta il valore di MTU [Maximum Transmit Unit - Unità Massima di
# Trasferimento] a <n>. A meno che il peer non richieda un valore
# minore tramite la negoziazione MRU, pppd chiederà al codice di
# gestione della rete del kernel di inviare attraverso l'interfaccia
# di rete del PPP pacchetti di dati di non più di n byte.
#mtu <n>

# Imposta il nome del sistema locale ad <n> per scopi di
# autentificazione. Se si sta usando PAP/CHAP probabilmente dev'essere
# impostato al nome utente del proprio ISP.
#name <n>

# Imposta a <n> il nome utente da usare per autentificare questa
# macchina con il peer usando il PAP.
# NON usare questa opzione se si sta usando l'opzione 'name'
# suddetta.
#user <u>

# Obbliga l'uso del nome dell'host come nome del sistema locale per
# scopi di autentificazione (ha priorità sull'opzione name).
#usehostname

# Imposta a <n> il nome assunto del sistema remoto per scopi di
# autentificazione.
#remotename <n>
```

```
# Aggiunge una voce nella tabella ARP [Address Resolution Protocol -
# Protocollo di Risoluzione Indirizzi] di sistema con l'indirizzo IP
# del peer e l'indirizzo Ethernet di questo sistema.
#proxyarp

# Usa la base di dati di password di sistema per autentificare con il
# peer usando PAP.
#login

# Se è specificata questa opzione, pppd invierà un frame echo-request
# LCP al peer ogni n secondi.  Sotto Linux, echo-request è inviata
# quando per n secondi non si sono ricevuti pacchetti dal
# peer. Normalmente il peer dovrebbe rispondere a echo-request
# inviando un echo-reply. Questa opzione può essere usata con
# l'opzione lcp-echo-failure per rilevare che il peer non è più
# connesso.
#lcp-echo-interval <n>

# Se è specificata questa opzione, pppd presumerà che il peer è morto
# se sono inviati n echo-request LCP senza ricevere un echo-reply LCP
# valido. Se questo accade, pppd terminerà la connessione. L'uso di
# questa opzione richiede un valore non nullo come parametro a
# lcp-echo-interval. Questa opzione può essere usata per abilitare pppd
# a terminare dopo l'interruzione della connessione fisica (p.es. il
# modem ha messo giù la linea) nelle situazioni nella quali non sia
# disponibile il controllo hardware delle linee da parte del modem.
#lcp-echo-failure <n>

# Imposta l'intervallo di riavvio LCP (timeout nella ritrasmissione) a
# <n> secondi (valore predefinito 3).
#lcp-restart <n>

# Imposta il massimo numero di trasmissioni di terminate-request LCP a
# <n> (valore predefinito 3).
#lcp-max-terminate <n>

# Imposta il numero massimo di trasmissioni di configure-request LCP a
# <n> (valore predefinito 10).
# Alcuni server PPP non lenti nell'avviarsi. Si può dover incrementare
# questo valore se si ottengono errori tipo 'serial line looped back'
# e si è SICURI di aver fatto correttamente il login e che il il PPP
# dovrebbe essere già avviato sul server.
#lcp-max-configure <n>

# Imposta il numero massimo di configure-NAK LCP ricevuti prima di
# inviare configure-Rejects <n> (valore predefinito 10).
#lcp-max-failure <n>

# Imposta l'intervallo di riavvio di IPCP (timeout in ritrasmissione)
```

```
# a <n> secondi (valore predefinito 3).
#ipcp-restart <n>

# Imposta il numero massimo di trasmissioni di terminate-request IPCP
# a <n> (valore predefinito 3).
#ipcp-max-terminate <n>

# Imposta il numero massimo di trasmissioni di configure-request IPCP
# a <n> (valore predefinito 10).
#ipcp-max-configure <n>

# Imposta il numero massimo di configure-NAK IPCP ricevuti prima di
# inviare configure-Rejects <n> (valore predefinito 10).
#ipcp-max-failure <n>

# Imposta l'intervallo di riavvio PAP (timeout in ritrasmissione)
# a <n> secondi (valore predefinito 3).
#pap-restart <n>

# Imposta il numero massimo di trasmissioni di authenticate-request
# PAP a <n> (valore predefinito 10).
# <n> (default 10).
#pap-max-authreq <n>

# Imposta l'intervallo di riavvio CHAP (timeout in ritrasmissione per
# challenge) a <n> secondi (valore predefinito 3).
#chap-restart <n>

# Imposta il numero massimo di trasmissioni challenge CHAP a <n>
# (valore predefinito 10).
#chap-max-challenge

# Con questa opzione, pppd reinterrogherà il peer ogni <n> secondi.
#chap-interval <n>

# Con questa opzione, pppd accetterà il suggerimento del peer per il
# proprio indirizzo IP locale, anche se l'indirizzo IP locale è stato
# specificato in un'opzione.
#ipcp-accept-local

# Con questa opzione, pppd accetterà il suggerimento del peer per il
# suo indirizzo IP (remoto), anche l'indirizzo IP remoto è stato
# specificato in un'opzione.
#ipcp-accept-remote
```

12.2 Quali opzioni dovrei usare? (No PAP/CHAP)

Beh, come in tutte le cose... dipende. Le opzioni qui specificate dovrebbero funzionare con la maggior parte dei server.

Comunque, se NON funzionassero, SI LEGGA IL FILE DI OPZIONI TIPO (`/etc/ppp/options.tpl`) **e** la pagina man di pppd **e** si parli con l'amministratore o il supporto clienti del server a cui ci si connette.

Si dovrebbe inoltre notare che gli script di connessione qui presentati usano alcune opzioni in riga di comando di pppd per rendere il tutto un poco più semplice da modificare.

```
# /etc/ppp/options (NIENTE PAP/CHAP)
#
# Previene pppd da fare il fork e passare in background
-detach
#
# usa le linee di controllo del modem
modem
# usa il lock in stile uucp per assicurare l'accesso esclusivo al
# dispositivo seriale.
lock
# usa il controllo di flusso hardware
crtscts
# crea un instradamento predefinito per questa connessione nella
# tabella di instradamento
defaultroute
# NON imposta l'"escape" di nessuna sequenza di controllo
asyncmap 0
# usa la dimensione massima dei pacchetti in trasmissione di 552 byte
mtu 552
# usa la dimensione massima dei pacchetti in ricezione di 552 byte
mru 552
#
#-------FILE DEL FILE ESEMPIO /etc/ppp/options (niente PAP/CHAP)
```

13 Se il proprio server usa PAP (Password Authentication Protocol)

Se il server al quale ci connette richiede l'autentificazione PAP o CHAP c'è un po' più di lavoro da fare.

Al file delle opzioni precedente, si aggiungano le seguenti righe:

```
#
# forza pppd a usare il proprio nome utente nell'ISP come 'nome host'
# durante il processo di autentificazione
name <proprio nome utente nell'ISP>     # si deve modificare questa riga
#
# Se si fa girare un *server* PPP e si ha bisogno di forzare il PAP o
# il CHAP si tolga il commento alla riga appropriata fra le due che
# seguono. NON usarle se si è un client che si connette ad un server
# PPP (anche se questi usa PAP o CHAP) in quanto dicono al SERVER di
# autentificare se stesso nella propria macchina (e quasi certamente
# non potrà farlo e la connessione fallirà).
#+chap
#+pap
```

```
#
# Se si usano secret CRIPTATI nel file /etc/ppp/pap-secrets, allora
# togliere il commento alla riga che segue.
# Nota: questo NON è la stessa cosa che usare le password MS criptate
# come si può fare con MS RAS in Windows NT.
#+papcrypt
```

13.1 Usare MSCHAP

Il RAS di Microsoft Windows NT può essere configurato per usare una variante di CHAP (Challenge/Handshake Authentication Protocol). Nella distribuzione sorgente di PPP, è possibile trovare un file chiamato README.MSCHAP80 che lo discute.

Abilitando il debug di pppd è possibile determinare se il server richiede l'autentificazione usando questo protocollo. Infatti se il server la richiede, si vedrà una riga di questo tipo:

```
rcvd [LCP ConfReq id=0x2 <asyncmap 0x0> <auth chap 80> <magic 0x46a3>]
```

L'informazione critica è **auth chap 80**.

Per poter usare MS CHAP, sarà necessario ricompilare pppd per supportarlo. Si vedano le istruzioni nel file README.MSCHAP80 nei sorgenti di PPP per le istruzioni su come compilare e usare questa variante.

Si dovrebbe notare che al momento il codice supporta solo i client PPP Linux che si connettono ad un server MS Windows NT. **NON** supporta la configurazione di un server PPP Linux che usi MSCHAP80 come autentificazione dei client.

13.2 Il file secrets di PAP/CHAP

Se si usa l'autentificazione pap o chap, allora si deve creare anche il file secrets. Questi sono:

```
/etc/ppp/pap-secrets
/etc/ppp/chap-secrets
```

Il proprietario dev'essere l'utente root, il gruppo root e per ragioni di sicurezza i permessi devono essere impostati a 740.

La prima cosa da notare su PAP e CHAP è che sono stati progettati per autenticare **computer** non **utenti**.

"Huh? Qual è la differenza?" Vi chiederete.

Beh, una volta che la propria macchina ha fatto la sua connessione PPP al server, **QUALSIASI** utente nel proprio sistema può usare quella connessione: non solamente voi. Questo spiega perché è possibile creare una connessione WAN (wide area network) che unisca assieme due LAN (local area networks) usando PPP.

PAP può (e CHAP **LO FA**) richiedere una autentificazione **bidirezionale**, ovvero ognuno dei due sistemi richiede all'altro un nome valido e un secret. Comunque, questo **NON** è il modo nel quale funzionano le connessioni autenticate con PAP per la maggioranza dei server PPP.

Il vostro ISP probabilmente vi avrà assegnato un nome utente e una password per permettervi la connessione al loro sistema e quindi ad Internet. Il provider non è affatto interessato al nome del vostro computer, quindi probabilmente si dovrà usare il proprio nome utente nel proprio ISP come nome per il proprio computer.

Ciò è fatto usando usando l'opzione `name nome_utente` di pppd. Quindi, se si usa il nome utente assegnato dal proprio ISP, si aggiunga la riga

```
name nome_utente_nel_proprio_ISP
```

al proprio file `/etc/ppp/options`.

Tecnicamente, si dovrebbe in realtà usare **user nome_utente_nel_proprio_ISP** per PAP, ma pppd è suffientemente intelligente da interpretare `name` come `user` se è richiesto l'uso di PAP. Il vantaggio di usare l'opzione `name` è che è valida anche per CHAP.

Poiché PAP è per l'autentificazione di **computer**, tecnicamente si dovrebbe specificare anche un nome remoto per il computer. Comunque, poiché molti hanno solo un ISP, si può usare una wildcard (*) per il nome del computer remoto nel file secrets.

Non è male notare anche che molti ISP usano banchi di modem multipli connessi a differenti server terminali, ognuno con un nome diverso, ma ai quali si ACCEDE (a rotazione) da un unico numero. Può essere quindi un po' difficile in alcuni casi sapere qual'è il nome del computer remoto in quanto dipende a quale server terminale ci si connette!

13.3 Il file secrets di PAP

Il file `/etc/ppp/pap-secrets` è qualcosa di simile a

```
# Secrets per l'autentificazione usando PAP
# client        server       secret     indirizzi_IP_locali_accettabili
```

I quattro campi sono delimitati da spazi bianchi e l'ultimo può essere vuoto (il che è probabilmente quello che si vuole per l'allocazione di indirizzi IP dinamica e probabilmente anche per quella statica da parte del proprio ISP).

Supponiamo che il proprio ISP assegni il nome utente `fred` e la password `flintstone`, si dovrà aggiungere l'opzione `name fred` in `/etc/ppp/options[.ttySx]` e modificare il file `/etc/ppp/pap-secrets` come segue

```
# Secrets per l'autentificazione usando PAP
# client        server  secret          indirizzi_IP_locali_accettabili
fred              *    flintstone
```

Questo file dice per il nome di macchina locale `fred` (che si è detto di usare a pppd anche se non è il nome locale della propria macchina) e per **QUALSIASI** server, bisogna usare la password (secret) `flintstone`.

Si noti che non c'è bisogno di specificare un indirizzo IP locale, a meno che non si voglia FORZARE un indirizzo IP statico particolare. Anche se si prova, è abbastanza difficile che funzioni in quanto la maggior parte dei server PPP (per ragioni di sicurezza) non permettono al sistema remoto di impostare il numero IP che gli stanno per dare.

13.4 Il file secrets di CHAP

Ciò richiede di avere metodi di autentificazione mutua, ovvero si deve permettere alla propria di macchina di autentificare il server remoto **E** al server remoto di autentificare la propria macchina.

Quindi, se la propria macchina è `fred` ed il sistema remoto è `barney`, la propria macchina dovrebbe usare `name fred remotename barney` e la macchina remota dovrebbe usare `name barney remotename fred` nei rispettivi file `/etc/ppp/options.ttySx`.

Il file `/etc/chap-secrets` per fred dovrebbe assomigliare a

```
# Secrets per l'autentificazione usando CHAP
# client        server  secret            indirizzo IP locale accettabile
fred            barney  flintstone
barney          fred    wilma
```

e per barney

```
# Secrets per l'autentificazione usando CHAP
# client        server  secret            indirizzo IP locale accettabile
barney          fred    flintstone
fred            barney  wilma
```

Si noti in particolare che entrambe le macchine devono avere voci per l'autentificazione bidirezionale. Ciò permette alla macchina locale di autentificarsi presso il server **E** a quella remota di autentificarsi in quella locale.

13.5 Gestione di connessioni multiple autentificate con PAP

Alcuni possono connettersi a più di un server che usa PAP. Finché il proprio nome utente è diverso su ogni macchina alla quale ci si vuole connettere, questo non è un problema.

Comunque, molti utenti hanno lo stesso nome utente su due (o più o anche su tutti) sistemi sui quali si connettono. Ciò causa un problema nella corretta selezione della riga appropriata da `/etc/ppp/pap-secrets`.

Come ci si potrebbe aspettare, PPP fornisce un meccanismo per risolverlo. PPP permette di impostare un 'nome assunto' per la terminazione remota della connessione (il server) usando l'opzione **remotename** di pppd.

Si supponga di connettersi a due server PPP usando il nome utente fred. Allora si imposterà il proprio `/etc/ppp/pap-secrets` a qualcosa così

```
fred    pppserver1      barney
fred    pppserver2      wilma
```

Ora, per impostare la connessione a pppserver1 si dovrebbe usare `name fred remotename pppserver1` nel proprio ppp-options mentre per pppserver2 `name fred remotename pppserver2`.

Poiché è possibile selezionare il file delle opzioni del ppp da usare con pppd usando l'opzione `file nomefile`, si può creare uno script per connettersi ad ognuno dei propri server PPP, prendendo correttamente il file delle opzioni da usare e quindi selezionando l'opzione `remotename` giusta.

14 Instaurare manualmente una connessione PPP

Una volta creati i propri file `/etc/ppp/options` e `/etc/resolv.conf` (e, se necessario, il file `/etc/ppp/pap|chap-secrets`), è possibile verificare le impostazioni stabilendo manualmente una connessione PPP (una volta che la connessione funziona manualmente, automatizzeremo il processo).

Per far ciò, il proprio software di comunicazione dev'essere in grado di uscire SENZA reinizializzare il modem. Minicom lo può fare con ALT Q (o nelle versioni più vecchie con CTRL ALT Q).

Ci si assicuri di essere loggati come root.

Si lanci il proprio software di comunicazione (come minicom), si componga il numero del server e si faccia normalmente il login. Se è necessario dare un comando per avviare il PPP nel server, lo si faccia. Ora si vedranno le porcherie di cui si è parlato in precedenza.

Se si usa pap o chap, allora semplicemente connettendosi al sistema remoto si dovrebbe far partire il ppp e si vedranno le porcherie senza fare il login (sebbene ciò possa anche non accadere per alcuni server - si provi a premere **Enter** e si veda se iniziano le porcherie).

Ora si esca dal software di comunicazione *senza reinizializzare il modem* (ALT Q o CTL A Q in minicom) e al prompt di Linux (come root) si digiti

```
pppd -d -detach /dev/ttySx 38400 &
```

L'opzione -d abilita il debug - la conversazione di avvio di connessione ppp sarà registrata nei file di log del proprio sistema - il che torna utile in caso di problemi.

Le lucette del proprio modem dovrebbero accendersi e spegnersi finché la connessione PPP non è stabilita.

A questo punto si può dare un'occhiata all'interfaccia PPP, usando il comando

```
ifconfig
```

Oltre ai vari dispositivi Ethernet e loopback che si possiedono, si dovrebbe vedere qualcosa di simile a

```
ppp0      Link encap:Point-Point Protocol
          inet addr:10.144.153.104  P-t-P:10.144.153.51 Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:552  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

dove

- inet addr:10.144.153.10 è il numero IP del proprio capo della connessione.
- P-t-P:10.144.153.5 è il numero IP del SERVER.

(naturalmente, ifconfig non riporterà questi numeri IP ma quelli usati dal proprio server PPP).

Nota: ifconfig dice anche che la connessione è SU e FUNZIONANTE!

Se non viene riportato nessun dispositivo ppp o qualcosa di simile a

```
ppp0      Link encap:Point-Point Protocol
          inet addr:0.0.0.0  P-t-P:0.0.0.0  Mask:0.0.0.0
          POINTOPOINT  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

la propria connessione PPP non è stata fatta...si veda oltre per il debug!

Si dovrebbe essere in grado di vedere anche un instradamento all'host remoto (ed oltre). Per far ciò, si lanci il comando

```
route -n
```

Si dovrebbe vedere qualcosa di simile a

```
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
10.144.153.3    *               255.255.255.255 UH    1500   0        1 ppp0
127.0.0.0       *               255.0.0.0       U     3584   0       11 lo
10.0.0.0        *               255.0.0.0       U     1500   0       35 eth0
default         10.144.153.3    *               UG    1500   0        5 ppp0
```

Di particolare importanza, si noti che ci sono DUE voci che puntano alla propria interfaccia ppp.

La prima è un instradamento di tipo HOST (indicato dal flag H) e che permette di vedere l'host al quale si è connessi, ma niente altro.

La seconda è l'instradamento predefinito (default), stabilito fornendo a pppd l'opzione `defaultroute`. Questo è l'instradamento che dice al proprio PC Linux di inviare qualsiasi pacchetto NON destinato alle Ethernet locali - per le quali ci sono degli instradamenti specifici - al server PPP stesso. I server PPP è quindi responsabile dell'instradamento dei nostri pacchetti verso Internet e dell'instradamento dei pacchetti di ritorno verso di noi.

Se non si vede una tabella di instradamento con queste due voci, qualcosa è andato storto. In particolare se i log di sistema mostrano un messaggio che dice che pppd non ha rimpiazzato un instradamento predefinito, allora si ha l'instradamento predefinito che punta alla propria interfaccia Ethernet - il quale **DEVE** essere rimpiazzato da un instradamento di rete specifico: **SI PUÒ AVERE SOLO UN INSTRADAMENTO PREDEFINITO!!!**

Si dovranno allora esplorare i file di inizializzazione del sistema per trovare dove viene impostato questo instradamento predefinito (probabilmente verrà usato un comando tipo `route add default...`). Si cambi questo comando in qualcosa di simile a `route add net...`.

Ora si verifichi la connessione facendo 'ping' al server usando il numero IP riportato dall'output di ifconfig, i.e.

```
ping 10.144.153.51
```

Si dovrebbe ricevere un output simile a

```
PING 10.144.153.51 (10.144.153.51): 56 data bytes
64 bytes from 10.144.153.51: icmp_seq=0 ttl=255 time=328.3 ms
64 bytes from 10.144.153.51: icmp_seq=1 ttl=255 time=190.5 ms
64 bytes from 10.144.153.51: icmp_seq=2 ttl=255 time=187.5 ms
64 bytes from 10.144.153.51: icmp_seq=3 ttl=255 time=170.7 ms
```

Questo elenco andrà avanti per sempre - per fermarlo si usi CTRL C, e a quel punto si riceveranno un altro po' di informazioni:

```
--- 10.144.153.51 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 170.7/219.2/328.3 ms
```

Fin qua tutto bene.

Ora si provi a fare il ping ad un host usando il suo nome (non il nome del server PPP stesso ma un host in un altro sito che si SA essere probabilmente attivo...). Per esempio

```
ping sunsite.unc.edu
```

A questo punto ci sarà una piccola pausa mentre Linux ottiene dal DNS specificato in `/etc/resolv.conf` il numero IP per il nome host "fully qualified" al quale si è fatto il ping - quindi non ci si preoccupi (si vedranno comunque le lucette del modem lampeggiare). In breve si otterrà un output simile a

```
 PING sunsite.unc.edu (152.2.254.81): 56 data bytes
64 bytes from 152.2.254.81: icmp_seq=0 ttl=254 time=190.1 ms
64 bytes from 152.2.254.81: icmp_seq=1 ttl=254 time=180.6 ms
64 bytes from 152.2.254.81: icmp_seq=2 ttl=254 time=169.8 ms
64 bytes from 152.2.254.81: icmp_seq=3 ttl=254 time=170.6 ms
64 bytes from 152.2.254.81: icmp_seq=4 ttl=254 time=170.6 ms
```

Ancora, lo si fermi con CTRL C per ottenere le statistiche...

```
--- sunsite.unc.edu ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 169.8/176.3/190.1 ms
```

Se non si riceve nessuna risposta, si provi a fare il ping all'indirizzo IP del server DNS del proprio ISP. Se si ottiene risposta, allora sembra proprio che ci sia qualche problema nel proprio `/etc/resolv.conf`.

Se non funziona, invece si ha un problema di instradamento oppure il proprio ISP ha qualche problema ad instradare i pacchetti di ritorno. Si verifichi che la tabella di instradamento sia come quella vista prima, e se è a posto si contatti il proprio ISP. Un buon test per ISP è di usare un altro sistema operativo per connettersi. Se si può andare oltre il proprio ISP con quello, allora il problema dipende dalla propria terminazione del collegamento.

Se tutto funziona, si può buttare giù la connessione digitando

```
ppp-off
```

Dopo una breve pausa, il modem dovrebbe riattaccare da solo.

Se questo non funziona, si spenga il proprio modem oppure si lanci il proprio software di comunicazione e si interrompa il modem con +++ e poi si riagganci con ATH0 quando si riceve il prompt OK del modem.

Si deve inoltre cancellare il file di lock creato da pppd

```
rm -f /var/lock/LCK..ttySx
```

15 Automatizzare la propria connessione e creazione degli script

Sebbene si possa continuare a connettersi a mano come appena mostrato, è molto più elegante creare alcuni script che lo facciano automaticamente per noi.

Un insieme di script che automatizzano il login e l'avvio del PPP cosicché tutto quello che si deve fare (come root o come membro del gruppo PPP) è dare un unico comando che avvii la connessione.

15.1 Script di connessione per l'autentificazione con Nome utente/Password

Se il proprio ISP NON richiede l'uso di PAP/CHAP, questi sono gli script per te!

Se il pacchetto ppp è installato correttamente, si dovrebbero avere due file di esempio. Per PPP 2.1.2 sono in `/usr/sbin` e per PPP 2.2 sono in `/etc/ppp/scripts`. Si chiamano

per PPP-2.1.2

```
ppp-on
ppp-off
```

e per PPP-2.2

```
ppp-off
ppp-on
ppp-on-dialer
```

Ora, se si sta usando PPP 2.1.2, suggerisco caldamente di cancellare i file d'esempio. Ci sono potenziali problemi con questi. E non venitemi a dire che funzionano bene, li ho usati per anni anch'io (e li raccomandavo nella prima versione di questo HOWTO)!

A beneficio degli utilizzatori di PPP 2.1.2, qui sono riportate delle versioni d'esempio MIGLIORI prese dalla distribuzione di PPP 2.2. Suggerisco di copiarsi ed usare questi script **invece** dei vecchi script di PPP 2.1.2.

15.2 Lo script ppp-on

Questo è il primo di una COPPIA di script che lanciano la connessione.

```
#!/bin/sh
#
# Script per inizializzare una connessione PPP. Questo è il primo di
# una coppia di script. Questa coppia di script non è sicura in quanto
# il codice è visibile usando il comando 'ps'. Comunque, è semplice.
#
# Questi sono i parametri. Li si modifichi a quel che necessita.
TELEPHONE=555-1212      # Il numero telefonico per la connessione
ACCOUNT=george          # Il nome di account per il login (come in
                        # 'George Burns')
PASSWORD=gracie         # La password per questo account (e 'Gracie Allen')
LOCAL_IP=0.0.0.0        # Indirizzo IP locale, se noto. Dinamico = 0.0.0.0
REMOTE_IP=0.0.0.0       # Indirizzo IP remoto, se desiderato.
                        # Dinamico = 0.0.0.0
NETMASK=255.255.255.0   # La netmask appropriata, se necessaria
#
# Gli esporta rendendoli globali così saranno disponibili anche a
# 'ppp-on-dialer'
export TELEPHONE ACCOUNT PASSWORD
#
# Questa è la posizione dello script che compone il numero e fa il
# login. Si invita ad usare il nome file assoluto in quanto la
# variabile $PATH non è usata nelle opzioni di connessione (e farlo in
```

```
# un account come 'root' sarebbe un buco di sicurezza, quindi non lo
# si richieda).
#
DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
#
# Inizializza la connessione
#
#
exec /usr/sbin/pppd debug /dev/ttySx 38400 \
        $LOCAL_IP:$REMOTE_IP \
        connect $DIALER_SCRIPT
```

Ecco qui lo script ppp-on-dialer:

```
#!/bin/sh
#
# Questa è la seconda parte dello script ppp-on. Effettua il
# protocollo di connessione per la connessione desiderata.
#
/usr/sbin/chat -v                                                    \
        TIMEOUT         3                               \
        ABORT           '\nBUSY\r'                      \
        ABORT           '\nNO ANSWER\r'                 \
        ABORT           '\nRINGING\r\n\r\nRINGING\r'    \
        ''              \rAT                            \
        'OK-+++\c-OK'   ATH0                            \
        TIMEOUT         30                              \
        OK              ATDT$TELEPHONE                  \
        CONNECT         ''                              \
        ogin:--ogin:    $ACCOUNT                        \
        assword:        $PASSWORD
```

Per PPP-2.2, lo script `ppp-off` è qualcosa di simile a:

```
#!/bin/sh
########################################################################
#
# Determina il dispositivo da chiudere
#
if [ "$1" = "" ]; then
        DEVICE=ppp0
else
        DEVICE=$1
fi

########################################################################
#
# Se esiste il pid file ppp0 allora il programma è in esecuzione. Lo
# ferma.
if [ -r /var/run/$DEVICE.pid ]; then
```

```
        kill -INT `cat /var/run/$DEVICE.pid`
#
# Se il kill non funziona allora non c'è alcun processo in esecuzione
# con questo pid. Significa inoltre che il file di lock è stato
# lasciato lì. Allo stesso tempo si può quindi voler cancellare il
# file di lock.
        if [ ! "$?" = "0" ]; then
                rm -f /var/run/$DEVICE.pid
                echo "ERRORE: Rimozione del vecchio file pid"
                exit 1
        fi
#
# Successo. Si lascia che pppd pulisca da solo la sua roba vecchia.
        echo "Connessione PPP a $DEVICE terminata."
        exit 0
fi
#
# Il processo ppp non è in esecuzione per ppp0
echo "ERRORE: connessione PPP non attiva su $DEVICE"
exit 1
```

15.3 Modificare gli script forniti per l'avvio del PPP

Poiché gli script sono due, andremo a modificarne uno alla volta.

15.3.1 Lo script ppp-on

Si deve modificare questo script per riflettere il PROPRIO nome utente nell'ISP, la PROPRIA password per l'ISP, e il numero di telefono di quest'ultimo.

Ognuna delle righe simili a `TELEPHONE=` in realtà impostano una variabile shell che contiene l'informazione a destra di '=' (ad esclusione dei commenti naturalmente). Quindi si modifichi ognuna di queste righe in modo che vadano bene per la propria connessione e il proprio ISP.

Inoltre, se si imposta il numero IP (se ce n'è bisogno), nel file `/etc/ppp/options` si CANCELLI la riga che dice

```
$LOCAL_IP:$REMOTE_IP \
```

Inoltre, ci si assicuri che la variabile shell DIALER_SCRIPT punti al percorso completo e al nome dello script che compone il numero che si intende usare. Quindi, se si è spostato o rinominato lo script, ci si assicuri di modificare in modo adeguato lo script `ppp-on`.

15.3.2 Lo script ppp-on-dialer

Questo è il secondo degli script che mettono in funzione la propria connessione PPP.

Nota: uno script di conversazione (chat) è solitamente tutto in una riga. I backslash ('\') sono usati per permettere la continuazione di una riga in più righe fisiche (per comodità di lettura) e non fanno parte dello script stesso.

Comunque, è molto utile analizzarlo in dettaglio così da capire cosa realmente fa (o dovrebbe fare)!

15.4 Il significato di uno script di conversazione (chat)...

Un script di conversazione è una sequenza di coppie "stringa attesa" "stringa inviata". In particolare si noti che ci si aspetta **SEMPRE qualcosa** prima di inviare qualcos'altro.

Se si invia qualcosa **SENZA** prima ricevere niente, si deve usare una stringa attesa vuota (indicata da "") e similmente per aspettare qualcosa senza mandare niente! Inoltre, se una stringa consiste di diverse parole (p.es. NO CARRIER), bisogna racchiuderla tra apici in modo che chat la veda come una singola entità.

La riga di conversazione nel nostro esempio è:

```
exec /usr/sbin/chat -v
```

Invocando chat, il -v dice a chat di copiare TUTTO il suo I/O nei log di sistema (solitamente /var/log/messages). Una volta che si è contenti di come funziona lo script, si modifichi questa riga per rimuovere il -v per preservare i log di sistema da confusione non necessaria.

```
TIMEOUT         3
```

Questo imposta il timeout per la ricezione di un input atteso a tre secondi. Ci può essere bisogno di incrementarlo diciamo a 5 o 10 secondi se si usa un modem veramente lento!

```
ABORT           '\nBUSY\r'
```

Se chat riceve la stringa BUSY, termina l'operazione.

```
ABORT           '\nNO ANSWER\r'
```

Se chat riceve la stringa NO ANSWER, termina l'operazione.

```
ABORT           '\nRINGING\r\n\r\nRINGING\r'
```

Se chat riceve (ripetutamente) la stringa RINGING, termina l'operazione. Significa che qualcuno vi sta chiamando al telefono!

```
"               \rAT
```

Attende che il modem non ritorni niente prima di inviare la stringa AT.

```
OK-+++\c-OK   ATH0
```

Questa è un po' più complicata in quanto sfrutta qualcuna delle capacità di recupero dagli errori di chat.

Quel che dice è... Attendi OK, se NON è ricevuto (perché in modem non è in modo comandi) allora invia +++ (la stringa predefinita compatibile Hayes che riporta il modem in modo comandi) e attendi OK. Poi invia ATH0 (la stringa di "riattacco" del modem). Ciò permette al nostro script di far fronte alle situazioni nella quali il proprio modem è "bloccato in linea"!

```
TIMEOUT         30
```

Imposta il timeout a 30 secondi per il resto dello script. Se si ha esperienza di problemi con lo script di chat che termina a causa di timeout, lo si incrementi a 45 secondi o più.

```
OK              ATDT$TELEPHONE
```

Attende OK (la risposta del modem al comando ATH0) e compone il numero che si vuole chiamare.

```
CONNECT         ''
```

Attende CONNECT (che il modem invia quando il modem remoto risponde) e non invia niente in risposta.

```
ogin:--ogin:    $ACCOUNT
```

Ancora, abbiamo anche qui un po' di recupero dagli errori. Aspetta il prompt per il login (...ogin:) ma se non lo riceve entro il timeout, invia un return e poi attende ancora il prompt per il login. Quando ha ricevuto il prompt, invia il nome utente (salvato nella variabile $ACCOUNT).

```
assword:        $PASSWORD
```

Attende il prompt per la password e invia la password (ancora salvata in una variabile shell).

Questo script ha delle ragionevoli capacità per il recupero dagli errori. chat ha considerevolmente molte più possibilità di quelle qui dimostrate. Per maggiori informazioni si consulti la pagina di manuale di chat (`man 8 chat`).

15.4.1 L'avvio di PPP dal lato server

Sebbene lo script ppp-on-dialer vada bene per server che avviano automaticamente il pppd da loro lato della connessione una volta effettuato il login, alcuni server richiedono che si avvii esplicitamente il PPP nel server.

Se si deve dare un comando per avviare il PPP sul server, allora si DEVE modificare lo script ppp-on-dialer.

Alla FINE dello script (dopo la riga della password) si aggiunga un'altra coppia **attesa inviata** - questa cercherà il proprio prompt della shell (si faccia attenzione ai caratteri che hanno un significato particolare nelle Bourne shell, come $ e [o] (parentesi quadre aperte e chiuse).

Una volta che chat ha trovato il prompt della shell, deve dare il comando di avvio del PPP richiesto dal proprio ISP.

Nel mio caso, il mio server PPP usa il prompt della Linux Bash predefinito

```
[hartr@kepler hartr]$
```

e richiede che io digiti

```
ppp
```

per avviare il PPP sul server.

È una buona idea permettere un po' di recupero dagli errori anche qui, quindi nel mio caso uso

```
        hartr--hartr    ppp
```

Questo dice che se non si riceve il prompt entro il timeout, di inviare un ritorno carrello (carriage return) e di cercare un'altra volta il prompt.

Una volta ricevuto il prompt, allora invia la stringa `ppp`.

Nota: non si dimentichi di aggiungere un \ alla fine della riga precedente cosicché chat pensi ancora che l'intero script di conversazione sia in una riga!

Sfortunatamente, alcuni server producono un insieme di prompt molto variabili. Può essere necessario effettuare più volte il login usando minicom per capire l'andamento e scegliere una stringa "attesa" stabile.

15.5 Uno script di conversazione per le connessioni autentificate con PAP/CHAP

Se il proprio ISP usa PAP/CHAP, allora lo script di conversazione è molto più semplice. Tutto ciò che lo script deve fare è di comporre il numero telefonico, attendere la connessione e poi lasciare che pppd gestisca il login!

```
#!/bin/sh
#
# Questa è la parte 2 dello script ppp-on. Effettuerà il protocollo di
# connessione per la connessione desiderata.
#
exec /usr/sbin/chat -v                                  \
        TIMEOUT         3                               \
        ABORT           '\nBUSY\r'                      \
        ABORT           '\nNO ANSWER\r'                 \
        ABORT           '\nRINGING\r\n\r\nRINGING\r'    \
        ''              \rAT                            \
        'OK-+++\c-OK'   ATH0                            \
        TIMEOUT         30                              \
        OK              ATDT$TELEPHONE                  \
        CONNECT         ''                              \
```

15.6 Le opzioni "`debug`" e "`file file_opzioni`" di pppd

Come si è già visto, si può abilitare la registrazione delle informazioni di debug usando l'opzione -d di pppd. L'opzione 'debug' è equivalente.

Poiché stiamo stabilendo una connessione con un nuovo script, per il momento si lasci l'opzione per il debug (Attenzione: se lo spazio del disco è stringente, la registrazione degli scambi pppd può far aumentare rapidamente la dimensione del file syslog, causando problemi - questo succede solo nel caso in cui si fallisce la connessione e si continua a provare per qualche minuto).

Una volta contenti di come tutto funziona, si può rimuovere questa opzione.

Se si è chiamato il file delle opzioni di ppp con qualcosa di diverso da `/etc/ppp/options` o `/etc/ppp/options.ttySx`, si specifichi il nome del file con l'opzione `file` di pppd - p.es.

```
exec /usr/sbin/pppd debug file options.myserver /dev/ttyS0 38400 \
```

16 Verificare i propri script di connessione

Si apra un nuovo Xterm per root (se si è in X) oppure si apra una nuova console virtuale e si faccia il login come root.

In questa nuova sessione, si lanci il comando

```
tail -f /var/log/messages
```

(o di qualsiasi altro sia il file di registrazione di sistema).

Nella prima finestra (o console virtuale) si lanci il comando

```
ppp-on &
```

(o qualsiasi altro nome con il quale si è chiamato la versione modificata di /usr/sbin/ppp-on). Se non si fa andare lo script in background specificando & alla fine del comando, non si riotterrà il prompt del terminale finché ppp non esce (quando termina il collegamento).

Ora si torni alla finestra che sta mostrando i log di sistema.

Si vedrà qualcosa di simile a quanto segue (a patto che si sia specificata l'opzione -v per chat e -d per pppd).... ciò che segue è quanto lo script di conversazione registra nei log di sistema seguito dalle informazioni di avvio per pppd:

```
Oct 21 16:09:58 hwin chat[19868]: abort on (NO CARRIER)
Oct 21 16:09:59 hwin chat[19868]: abort on (BUSY)
Oct 21 16:09:59 hwin chat[19868]: send (ATZ^M)
Oct 21 16:09:59 hwin chat[19868]: expect (OK)
Oct 21 16:10:00 hwin chat[19868]: ATZ^M^M
Oct 21 16:10:00 hwin chat[19868]: OK -- got it
Oct 21 16:10:00 hwin chat[19868]: send (ATDT722298^M)
Oct 21 16:10:00 hwin chat[19868]: expect (CONNECT)
Oct 21 16:10:00 hwin chat[19868]: ^M
Oct 21 16:10:22 hwin chat[19868]: ATDT722298^M^M
Oct 21 16:10:22 hwin chat[19868]: CONNECT -- got it
Oct 21 16:10:22 hwin chat[19868]: send (^M)
Oct 21 16:10:22 hwin chat[19868]: expect (ogin:)
Oct 21 16:10:23 hwin chat[19868]: kepler login: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (hartr^M)
Oct 21 16:10:23 hwin chat[19868]: expect (ssword:)
Oct 21 16:10:23 hwin chat[19868]:  hartr^M
Oct 21 16:10:23 hwin chat[19868]: Password: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (??????^M)
Oct 21 16:10:23 hwin chat[19868]: expect (hartr)
Oct 21 16:10:24 hwin chat[19868]: [hartr -- got it
Oct 21 16:10:24 hwin chat[19868]: send (ppp^M)
Oct 21 16:10:27 hwin pppd[19872]: pppd 2.1.2 started by root, uid 0
Oct 21 16:10:27 hwin pppd[19873]: Using interface ppp0
Oct 21 16:10:27 hwin pppd[19873]: Connect: ppp0 <--> /dev/cua1
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: LCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(LCP): Rcvd id 1.
```

```
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MRU
Oct 21 16:10:27 hwin pppd[19873]: (1500)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ASYNCMAP
Oct 21 16:10:27 hwin pppd[19873]: (0)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MAGICNUMBER
Oct 21 16:10:27 hwin pppd[19873]: (a098b898)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd PCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ACCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: returning CONFACK.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 2, id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfack(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:27 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:27 hwin pppd[19873]: (45)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:28 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 1.
Oct 21 16:10:30 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:30 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:30 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 255.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:31 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:31 hwin pppd[19873]:  (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:31 hwin pppd[19873]: (45)
Oct 21 16:10:31 hwin pppd[19873]:  (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:31 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 255.
Oct 21 16:10:31 hwin pppd[19873]: fsm_rconfack(IPCP): Rcvd id 1.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: up
Oct 21 16:10:31 hwin pppd[19873]: local  IP address 10.144.153.104
Oct 21 16:10:31 hwin pppd[19873]: remote IP address 10.144.153.51
```

Si noti che uso numeri IP STATICI, quindi la mia macchina li spedisce al server PPP, e quindi non si vedrà questa parte se si usano numeri IP DINAMICI. Inoltre questo server richiede un comando specifico per avviare il ppp dalla sua parte.

Se tutto sembra OK, lo si verifichi ulteriormente come prima con ping a numeri IP e a nomi di host.

Si lanci il proprio web browser o quant'altro e si cominci a navigare, si è finalmente connessi!

17 Chiudere la connessione PPP

Quando si è finito di usare la connessione PPP, si usi il comando standard ppp-off per buttarla giù (si ricorda che bisogna essere o root oppure un membro del gruppo PPP!).

Nei file di log di sistema di vedrà qualcosa di simile a:

```
Oct 21 16:10:45 hwin pppd[19873]: Interrupt received: terminating link
Oct 21 16:10:45 hwin pppd[19873]: ipcp: down
Oct 21 16:10:45 hwin pppd[19873]: default route ioctl(SIOCDELRT): Bad address
Oct 21 16:10:45 hwin pppd[19873]: fsm_sdata(LCP): Sent code 5, id 2.
Oct 21 16:10:46 hwin pppd[19873]: fsm_rtermack(LCP).
Oct 21 16:10:46 hwin pppd[19873]: Connection terminated.
Oct 21 16:10:46 hwin pppd[19873]: Exit.
```

Non ci si preoccupi del `SIOCDELRT`, significa solo che pppd si è accorto che sta terminando e non è niente di preoccupante.

18 Risoluzione dei problemi

Ci possono essere diversi motivi per i quali la propria connessione non funziona: chat non è arrivato correttamente alla fine della sua esecuzione, oppure si ha una linea molto disturbata, ecc. Quindi si controlli nei log di sistema per indicazioni.

18.1 Ho compilato il supporto per il PPP nel kernel, ma...

Un problema molto comune è che la gente compila il supporto per il PPP nel kernel, ed ancora quando provano a lanciare pppd, il kernel afferma che non ha il supporto per il ppp! Ci sono diverse ragioni per le quali ciò può accadere.

18.1.1 Si è fatto il boot con il kernel giusto?

Sebbene si sia **ricompilato** il supporto per il ppp nel proprio kernel, non si è fatto il boot con il nuovo kernel. Ciò può succedere se non si è aggiornato `/etc/lilo.conf` e rilanciato lilo.

Una buona verifica sul kernel la si può ottenere usando il comando `uname -a`, che dovrebbe produrre una riga simile a

```
Linux archenland 2.0.28 #2 Thu Feb 13 12:31:37 EST 1997 i586
```

Questa mostra la versione del kernel e la data della sua compilazione, che dovrebbero dare un'idea abbastanza buona di cosa sta funzionando.

18.1.2 Si è compilato il supporto per il ppp come modulo?

Se si è compilato il supporto per il ppp a livello kernel come modulo, ma non si sono creati ed installati i moduli, allora si può ottenere questo tipo di errore. Si veda il Kernel-HOWTO e il file README in `/usr/src/linux`!

Un'altra possibilità connessa ai moduli è che ci si aspetta che il modulo sia caricato automaticamente, ma non è in esecuzione il demone `kerneld` (che automaticamente carica e scarica i moduli quando necessario). Si veda il kerneld mini-HOWTO per informazioni su come configurare kerneld.

18.1.3 Si sta usando la versione corretta di PPP per il proprio kernel?

Si **deve** usare ppp-2.2 con kernel versione 2.0.x. Si può usare ppp-2.2 con kernel versione 1.2.x (se si applica una patch al kernel) altrimenti si deve usare ppp-2.1.2.

18.1.4 Si sta eseguendo pppd come root?

Se non si esegue pppd come utente root (e pppd non è suid a root), si può ricevere questo messaggio.

18.2 Il mio modem si connette ma il ppp non parte mai

Ci sono innumerevoli variazioni sul tema (si dia un'occhiata a comp.os.linux...).

Un errore **MOLTO** comune è che si è scritto male qualcosa nei propri script. La sola cosa da fare in questo caso è assicurarsi di registrare nei log di sistema (/var/log/messages) la conversazione di chat tra il proprio PC Linux e il server e poi analizzarla *riga per riga*. Può essere necessario connettersi manualmente al server PPP e controllare ancora tutto.

Nella registrazione si devono controllare molto attentamente i prompt reali, tenendo bene in mente che noi umani abbiamo la tendenza a leggere quello che PENSIAMO di aver scritto, e non quello che c'è realmente scritto!

18.3 Il log di sistema dice "`serial line is not 8 bit clean`"

Ci sono anche qui delle varianti, come `serial line looped back`, ecc., e le cause possono essere molteplici (anche più di una contemporaneamente).

Per capire cosa succede in questi casi, è necessario comprendere un po' di quel che succede dietro le quinte durante l'esecuzione di pppd.

Quando pppd viene avviato, invia pacchetti LCP (Link Control Protocol - Protocollo di Controllo del Collegamento) alla macchina remota. Se riceve una risposta valida allora passa allo stadio successivo (usando pacchetti IPCP - IP Control Protocol - Protocollo di Controllo IP) e solo quando questa negoziazione è completa viene avviato lo strato IP in modo che si possa usare la connessione PPP.

Se non c'è un server ppp funzionante all'altro capo quando il proprio PC invia pacchetti LCP, questi vengono riflessi dal processo di login della terminazione remota della connessione. Poiché questi pacchetti usano 8 bit, la loro riflessione causa la soppressione dell'ottavo bit (si ricorda che ASCII è un codice a 7 bit). Il PPP vede questo e si comporta di conseguenza.

Ci sono diverse ragioni per cui può avvenire questa riflessione.

18.3.1 Non ci si è loggati correttamente nel server

Quando il proprio script di conversazione termina, pppd viene avviato nel proprio PC. Comunque, se non si è completato il processo di login nel server (incluso l'invio di un qualsiasi comando necessario per avviare il PPP nel server), PPP non verrà avviato.

Quindi, i pacchetti LCP sono riflessi e si riceve questo errore.

Si deve controllare attentamente e correggere (se necessario) il proprio script di conversazione (si veda più indietro).

18.3.2 Non si è avviato il PPP nel server

Alcuni server PPP richiedono l'inserimento di un comando e/o di un RETURN dopo la completazione del processo di login prima di avviare il ppp dal loro lato della connessione.

Si verifichi il proprio script di conversazione (si veda più indietro).

Se si fa il login manualmente e si scopre che è necessario inviare un RETURN per avviare il PPP, si aggiunga semplicemente una coppia di stringhe attesa/inviata vuote alla fine del proprio script di conversazione (una stringa vuota in realtà invia un RETURN).

18.3.3 Il processo PPP remoto è lento a partire

Questa è un po' una birichinata!

Di default, il proprio pppd è compilato per inviare un massimo di 10 richieste di configurazione lcp. Se il server è un po' lento a partire, tutte le 10 richieste possono essere spedite prima che il server sia pronto a riceverle.

Nella propria macchina, pppd si vede tutte le 10 richieste che gli tornano indietro (senza l'ottavo bit) ed esce.

Ci sono due modi per aggirare questo problema:

Aggiungere un `lcp-max-configure 30` alle proprie opzioni del ppp. Ciò incrementa il numero massimo di pacchetti di configurazione LCP che pppd invia prima di uscire. Per server veramente lenti, può essere necessario incrementare ancora di più tale numero.

In alternativa, si può essere un po' furbini. Si dovrebbe aver notato che quando si fa il login a mano nel server PPP e il PPP viene avviato, il **primo** carattere ad apparire fra le porcherie prodotte dal server ppp è sempre il carattere tilde (~).

Usando questa conoscenza a priori, si può aggiungere una nuova coppia di stringhe `attesa/inviata` alla fine dello script di conversazione che aspetta una tilde e non invia niente. Cioè qualcosa di simile a:

```
\~      ''
```

Nota: poiché il carattere tilde ha un significato particolare nella shell, ne deve essere fatto l'escape (e quindi metterci prima un backslash).

18.4 Instradamento predefinito non impostato (`Default route not set`)

Se pppd si rifiuta di impostare un instradamento predefinito, è perché (abbastanza giustamente) si rifiuta di rimuovere/rimpiazzare un instradamento predefinito preesistente.

La ragione classica per cui capita questo errore è che alcune distribuzioni impostano l'instradamento predefinito verso la propria scheda Ethernet invece di impostare uno specifico instradamento di rete.

Si veda la Linux NAG ed il Net2/3 HOWTO per informazioni su come impostare correttamente la propria scheda Ethernet e gli instradamenti associati.

Un'altra ragione potrebbe essere che la propria LAN usi già un gateway/router e che la propria tabella di instradamento sia già impostata per puntare, tramite l'instradamento predefinito, a tale gateway.

Sistemare questa situazione può richiedere un po' di buone nozioni di IP networking e va oltre lo scopo di questo HOWTO. Si consiglia di trovare qualche aiuto esperto (tramite newsgroup o da qualcuno a cui si possa chiedere la soluzione e che sia raggiungibile in maniera diretta).

18.5 Altri problemi

Ci sono molte ragioni oltre a queste che possono causare il fallimento della connessione ppp e/o non farla funzionare correttamente.

Si vedano le PPP FAQ (che sono veramente una serie di domande e risposte). È un documento omnicomprensivo e le risposte SONO lì! Per la mia (brutta) esperienza, se le risposte al proprio problema non solo lì, il problema NON è un errore del ppp! Nel mio caso usavo un kernel ELF e non avevo aggiornato i moduli correttamente. Ho perso solamente due giorni (e buona parte di una notte) a stanare quello che si è rivelato essere un perfetto server PPP prima delle luci dell'alba!

19 Ottenere aiuto quando ormai non si sa più cosa provare

Se non si riesce a far funzionare la propria connessione PPP, si riveda tutto questo documento e si verifichi tutto, in congiunzione con l'output creato da "chat -v..." e "pppd -d" nel log di sistema.

Si consulti anche la documentazione di PPP e le FAQ, oltre agli altri documenti qui citati!

Se ancora non se ne esce, si provi nei newsgroup comp.os.linux.misc e comp.os.linux.networking che sono regolarmente seguiti da persone che possono aiutare con il PPP, così come nel newsgroup comp.protocols.ppp

Si può pure provare a mandare a me una email personale, ma non ho una giornata lavorativa (e di vita) regolare e quindi non garantisco di rispondere rapidamente, in quanto tutto dipende dal carico di lavoro attuale e dallo stato della mia vita privata!

In particolare, **NON SI POSTINO RISME DI OUTPUT DI DEBUG NEI GRUPPI DI DISCUSSIONE E NEMMENO LI SI SPEDISCA A ME VIA EMAIL**, nel primo caso si sprecherà un sacco di banda e nel secondo verranno consegnati a /dev/null (a meno che io non li abbia espressamente richiesti).

20 Problemi comuni una volta che il collegamento funziona

Un problema che si troverà è che i provider supportano solamente il pacchetto software di connessione che loro stessi distribuiscono ai nuovi abbonati. Questo è (tipicamente) per Microsoft Windows :-(e molti servizi di supporto clienti sembrano non saperne niente di Unix (o Linux). Così, ci si prepari ad un'assistenza molto limitata da quella parte.

Si potrebbe comunque fare a quegli individui un favore ed educarli su Linux (qualsiasi persona che lavora nel supporto clienti di un ISP dovrebbe essere abbastanza addentro alla terminologia di Internet e dovrebbe quindi avere a casa una macchina Linux!)

20.1 Non riesco a vedere niente oltre al server PPP al quale mi connetto

OK, la propria connessione PPP è su e funzionante e si può fare ping al server PPP con il suo numero IP (il secondo o "remoto" numero IP mostrato da `ifconfig ppp0`), ma non si riesce a raggiungere nient'altro.

Prima di tutto, si provi a fare ping ai numeri IP specificati nel proprio /etc/resolv.conf come name server. Se questi funzionano, si **può** vedere oltre il proprio server PPP (a meno che questi non siano gli stessi numeri IP del numero IP "remoto" della propria connessione). Quindi si provi a fare ping al nome completo del proprio provider, per esempio

```
ping my.provider.net.au
```

Se questo NON funziona, si ha un problema con la risoluzione dei nomi. Ciò probabilmente dipende da un errore di battitura nel proprio file /etc/resolv.conf. Lo si controlli accuratamente con le informazioni ottenute chiamando il proprio provider. Se tutto sembra a posto, si chiami il proprio provider e si verifichi di aver scritto correttamente i numeri IP.

Se ANCORA non funziona (e il proprio provider conferma che i suoi name server sono attivi e funzionanti), c'è un problema da qualche altra parte, e suggerisco di controllare attentamente la propria installazione di Linux (in particolare si controllino i permessi dei file).

Se ANCORA non si riesce a fare ping ai name server del proprio provider tramite i loro numeri IP, allora questi non sono attivi (si chiami il provider al telefono e si controlli) oppure c'è qualche problema di instradamento da parte del proprio provider. Ancora, lo si chiami e si verifichi.

Una possibilità è che la "terminazione remota" sia un server PPP Linux nel quale non sia stata specificata l'opzione di IP forwarding nel kernel!

Una buona verifica generale è di provare a connettersi al proprio provider usando il software che molti di questi forniscono per (gulp) Microsoft Windows. Se tutto funziona da un altro sistema operativo con esattamente lo stesso account, allora il problema è nel proprio sistema Linux e NON nel proprio provider.

20.2 Posso spedire email, ma non posso riceverla

Se si stanno usando numeri IP dinamici, è perfettamente normale. Si veda 21 (Usare i servizi Internet con numeri IP Dinamici) più avanti.

20.3 Perché la gente non può fare finger, WWW, gopher, talk nella mia macchina?

Ancora, se si stanno usando numeri IP dinamici, è perfettamente normale. Si veda 21 (Usare i servizi Internet con numeri IP Dinamici) più avanti.

21 Usare i servizi Internet con numeri IP dinamici

Se si stanno usando numeri IP dinamici (e molti provider danno solo numeri IP dinamici a meno che non si paghi significativamente di più per la propria connessione), allora si debbono accettare le limitazioni che questo impone.

Prima di tutto, le richieste di servizi esterni funzioneranno bene. Ovvero, si possono spedire email usando sendmail (a patto di averlo configurato correttamente), scaricare file in ftp da siti remoti, fare finger ad utenti su altre macchine, navigare nel web, ecc.

In particolare, si può rispondere mentre si è off line (ovvero non si è connessi) alle email che si sono scaricate nella propria macchine. La posta resterà semplicemente in attesa nella propria coda di posta (mail queue) finché non ci si connette al proprio ISP.

Comunque, la propria macchina NON è connessa ad Internet 24 ore al giorno, né ha lo stesso numero IP ogni volta che viene connessa. È quindi impossibile ricevere email direttamente nella propria macchina e veramente molto difficile configurare un server web o ftp a cui possano accedere i propri amici! Per come Internet è progettata la propria macchina non esiste come macchina unica e permanentemente contattabile quanto non ha un unico numero IP (si ricordi che altre macchine useranno il numero IP a voi assegnato in una connessione, quando gli viega allocato per la loro connessione).

Se si configura un server WWW (o qualsiasi altro server), tale server rimane totalmente sconosciuto a qualsiasi utente in Internet FINCHÉ questi non sappiano che la macchina è connessa E quale sia il suo reale (e momentaneo) numero IP. Ci sono diversi modi per far ottenere ad altri utenti questa informazione: si può chiamarli o mandargli un email per dirglielo oppure usare in modo astuto i file ".plan" su un account shell nel proprio provider (assunto che questi permetta accessi shell e finger).

Ora, per la maggior parte degli utenti questo non è un problema: tutto ciò che la maggior parte della gente vuole è inviare e ricevere email (usando il proprio account nel provider) e fare connessioni in uscita in WWW, ftp e ad altri servizi su Internet. Se veramente si DEBBONO avere connessioni entranti nella propria macchina, si dovrebbe realmente ottenere un numero IP statico. In alternativa si possono esplorare i metodi suggeriti in precedenza...

21.1 Configurazione dell'email

Anche per numeri IP dinamici, si può certamente configurare sendmail sulla propria macchina per spedire fuori qualsiasi email composta localmente. La configurazione di sendmail può essere oscura e difficoltosa, quindi questo documento non prova nemmeno a dire come farla. Comunque, si dovrebbe probabilmente configurare sendmail cosicché il proprio ISP sia designato come proprio "smart relay" host (l'opzione **DS** di `sendmail.cf`) (per maggiori informazioni sulla configurazione di sendmail, si veda la documentazione di sendmail, e si dia un'occhiata alle configurazioni m4 distribuite con sendmail. Ce n'è certamente una che va bene per il proprio caso).

Ci sono anche eccellenti libri su Sendmail (da citare la 'bibbia' della O'Reilly and Associates), ma questi esulano quasi certamente dalle necessità della maggior parte degli utenti!

Una volta configurato sendmail, probabilmente si vorrà fare in modo che, non appena la connessione PPP è attiva, sendmail spedisca qualsiasi messaggio in attesa nella coda di email in uscita. Per far ciò, si aggiunga il comando

```
sendmail -q &
```

al proprio script /etc/ppp/ip-up (si veda più avanti).

Le email in ingresso sono un problema nel caso di numeri IP dinamici. Il modo per gestirle è di:

- configurare il proprio mail user agent (elm, pine, ecc.) in modo che tutta la posta venga spedita spedita fuori con un header "reply" che riporti il proprio indirizzo di email nel proprio ISP.
  Se possibile, si dovrebbe anche impostare il proprio indirizzo FROM al proprio indirizzo email nel proprio provider.
- usare i programmi popclient o fetchmail per scaricare la propria posta dal proprio provider. In alternativa, se il proprio ISP usa IMAP, usare un MUA che supporta IMAP (come pine).

Si può automatizzare questo processo al momento dell'attivazione delle connessione mettendo i comandi appropriati nello script `/etc/ppp/ip-up` (si veda più avanti).

21.2 Configurare un name server locale

Sebbene si possano tranquillamente usare i DNS localizzati nel proprio ISP, si può anche impostare un name server (secondario) di sola cache locale che venga attivato dallo script ip-up. Il vantaggio di usare un name server locale (di sola cache) è che fa risparmiare tempo (e banda) se si contattano frequentemente gli stessi siti durante una lunga sessione on-line.

La configurazione di un DNS di sola cache (che usi una riga "forwarders" che punta al DNS dell'ISP nel file named.boot) è relativamente semplice. Il libro della O'Reilly (DNS and Bind) spiega tutto quello che serve sapere sull'argomento.

È disponibile anche un DNS-HOWTO.

Se si fa funzionare una piccola LAN che può accedere Internet attraverso il proprio PC Linux (usando IP Masquerade per esempio), probabilmente è una buona idea far girare un name server locale (con una direttiva forwarders) quando la connessione è attiva, in quanto ciò minimizzerà la banda e il ritardo associato con la risoluzione di nomi.

Una nota di Netiquette (galateo della rete): si chieda il permesso al proprio ISP prima di iniziare ad usare un name server secondario di sola cache nel dominio del proprio ISP. Propriamente configurato, il proprio DNS non causerà nessun problema a proprio provider, ma se si sbaglia qualcuno ne può causare.

22 Connettere assieme due reti usando PPP

In linea di principio non c'è differenza tra il collegare una macchina Linux ad un server PPP e il collegare due LAN usando PPP su una macchina di ognuna delle LAN. Si ricordi che il PPP è un protocollo **peer to peer** (da pari a pari).

Comunque, si deve **DEFINITIVAMENTE** comprendere come avviene l'instradamento. Si legga il NET-3 HOWTO e la Linux Network Administrator Guide (NAG). Di insostituibile aiuto può essere anche "TCP/IP Network Administration" (pubblicato dalla O'Reilly and Assoc - ISBN 0-937175-82-X).

Se si ha intenzione di usare il sub networking in un numero IP di rete su entrambi i lati della connessione, si troverà utile il Linux sub networking mini-howto (bozza). È disponibile a *Linux Sub networking mini-HOWTO* `<http://www.interweft.com.au/other/>`.

Per poter connettere due LAN, si **devono** usare numeri IP di rete diversi (o sottoreti dello stesso numero di rete) e si dovranno usare numeri IP statici oppure usare IP masquerade. Se si vuole usare IP masquerade, si veda l'IP masquerade mini-howto per le istruzionei sulla sua configurazione.

22.1 Impostare i numeri IP

Ci si metta d'accordo con l'amministratore di rete dell'altra LAN su i numeri IP che saranno usati per ognuno degli estremi dell'interfaccia PPP. Se si stanno usando numeri IP statici, ciò probabilmente richiede che ci si connetta ad uno specifico numero telefonico.

Si modifichi il file `/etc/ppp/options[.ttyXX]` appropriato - è una buona idea avere un modem e porta specifici dal proprio lato per questa connessione. Può essere pure necessario modificare il proprio file `/etc/ppp/options`, e creare appositi file options.ttyXX per le altre connessioni!

Si specifichino i numeri IP per il proprio lato del collegamento PPP nel file delle opzioni appropriato esattamente come mostrato in precedenza per i numeri IP statici.

22.2 Impostare l'instradamento

Ci si deve organizzare in modo che i pacchetti nella propria LAN locale siano instradati attraverso l'interfaccia che la connessione PPP stabilisce. Questo è un processo in due stadi.

Prima di tutto, si deve stabilire un instradamento dalla macchina che esegue la connessione PPP alla(e) rete(i) all'altro estremo del collegamento. Se il collegamento è verso Internet, ciò può essere gestito dal-

l'instradamento predefinito stabilito da pppd stesso nel proprio estremo della connessione, usando l'opzione 'defaultroute'.

Se comunque, la connessione è solo un collegamento tra due LAN, allora deve essere aggiunto un instradamento di rete specifico per ognuna delle reti accessibili attraverso la connessione. Ciò è fatto usando un comando 'route' per ognuna delle reti nello script /etc/ppp/ip-up (si veda 23 (Dopo che la connessione è attiva...) per le istruzioni su come farlo).

La seconda cosa che bisogna fare è dire agli altri computer sulla propria LAN che la propria macchina Linux è in realtà un 'gateway' (passaggio) per la(e) rete(i) all'altro estremo della connessione PPP.

Naturalmente, l'amministratore di rete dell'altro estremo della connessione ppp deve fare anche lui le stesse cose! Comunque, poiché egli instraderà i pacchetti per la vostra rete specifica, sarà richiesto un'**instradamento di rete specifico**, non un'instradamento predefinito (a meno che le LAN all'altro estremo della connessione non siano connesse a voi per accedere ad Internet attraverso la vostra connessione).

22.3 Sicurezza di rete

Se si connette la propria LAN ad Internet usando PPP, anche solamente ad una LAN "straniera", si deve essere consci delle questioni sulla sicurezza. Suggerisco caldamente di considerare la configurazione di un firewall!

Si dovrebbe parlare con l'amministratore LAN del proprio sito **PRIMA** di cominciare a connettere LAN straniere o Internet in questo modo. Gli errori che si potrebbero fare potrebbero risultare innocui come provocare problemi veramente seri!

23 Dopo che la connessione è attiva: lo script `/etc/ppp/ip-up`

Una volta che la connessione PPP è attiva, pppd cerca il file `/etc/ppp/ip-up`. Se questo script esiste ed è eseguibile, il demone PPP lo esegue. Ciò permette di automatizzare qualsiasi comando di instradamento che possa risultare utile, e qualsiasi altra azione che si vuole avvenga ogni volta che la connessione PPP viene attivata.

Questo è semplicemente uno script shell e può fare qualsiasi cosa che uno script shell può fare (ovvero virtualmente qualsiasi cosa si vuole).

Per esempio, si può fare in modo che sendmail invii qualsiasi messaggio in attesa d'uscita nella coda di posta.

Similmente, si può inserire in ip-up il comando per collezionare (usando pop) qualsiasi email in attesa nel proprio ISP.

Ci sono delle restrizioni su `/etc/ppp/ip-up`:

- È eseguito in un ambiente deliberatamente ristretto per aumentare la sicurezza. Ciò significa che si devono specificare i percorsi completi per gli eseguibili, ecc.
- Tecnicamente, `/etc/ppp/ip-up` è un *programma* non uno script. Ciò significa che può essere eseguito direttamemente, e quindi richiede il file magic standard (`#!/bin/bash`) all'inizio della prima riga e dev'essere leggibile ed eseguibile da root.

23.1 Instradamenti speciali

Se si stanno connettendo due LAN, si dovranno impostare instradamenti specifici per le LAN 'straniere'. Questo è facile da fare usando lo script `/etc/ppp/ip-up`. La sola difficoltà nasce se la propria macchina gestisce più connessioni PPP.

Questo perché `/etc/ppp/ip-up` è eseguito per OGNUNA delle connessioni PPP che si fanno, e quindi è necessario far eseguire i comandi di instradamento corretti per la connessione particolare che si fa in quel momento, e solo in quel caso e non quando si fa un qualsiasi altro collegamento!

23.2 Gestione delle code di posta

Quando la connessione tra due LAN è attiva, ci si vorrà assicurare che le email in coda ad entrambi gli estremi siano *scaricate* (flushed), ovvero inviate a destinazione. Ciò è fatto aggiungendo la chiamata appostita a `sendmail`.

Si può fare usando il costrutto 'case' della bash su un parametro appropriato che pppd passa allo script. Ad esempio, questo è lo script `/etc/ppp/ip-up` che uso per gestire le nostre connessioni WAN e le connessioni alla mia Ethernet di casa (gestita sempre dallo stesso server PPP).

23.3 Un esempio di script /etc/ppp/ip-up

L'esempio che segue gestisce una varietà di usi tipici.

```
#!/bin/bash
#
# Script che gestisce le questioni di instradamento necessarie per
# pppd. Solo la connessione a Newman richiede questa gestione.
#
# Quando la connessione ppp è attiva, questo script viene chiamato con
# i seguenti parametri
#       $1      il nome dell'interfaccia usata da pppd (p.es. ppp3)
#       $2      il nome del dispositivo tty
#       $3      la velocità del dispositivo tty
#       $4      l'indirizzo IP locale per l'interfaccia
#       $5      l'indirizzo IP remoto
#       $6      il parametro specificato dall'opzione 'ipparam' di pppd
#
case "$5" in
# Gestisce gli instradamenti al server del Campus Newman
        202.12.126.1)
                /sbin/route add -net 202.12.126.0 gw 202.12.126.1
# e scarica la mail queue per spedire là le loro email il prima possibile!
                /usr/sbin/sendmail -q &
                ;;
        139.130.177.2)
# La nostra connessione ad Internet
# Quando la connessione è attiva, avvia il time server (se non è già
# attivo) e si sincronizza con il mondo
                if [ ! -f /var/lock/subsys/xntpd ]; then
                        /etc/rc.d/init.d/xntpd.init start &
```

```
                fi
# Avvia il news server (se non è già attivo)
                if [ ! -f /var/lock/subsys/news ]; then
                        /etc/rc.d/init.d/news start &
                fi
                ;;
        203.18.8.104)
# Spedisce la mail in attesa nella mia macchina non appena il link è
# attivo.
# Non è richiesto nessun instradamento in quanto la mia Ethernet di
# casa è gestita da un instradamento di IP masquerade e proxyarp.
                /usr/sbin/sendmail -q &
                ;;
        *)
esac
exit 0
```

Come risultato della connessione ppp al nostro Campus Newman e di questo script, si ottengono le seguenti voci nella tabella di routing (questa macchina è pure il nostro server PPP e gestisce le nostre connessioni a Internet – ho aggiunto dei commenti nell'output per spiegare a cosa corrisponde ognuna delle voci):

```
[root@kepler /root]# route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
# l'instradamento HOST al nostro gateway Internet remoto
139.130.177.2   *               255.255.255.255 UH    1500   0      134 ppp4
# l'instradamento HOST al server del nostro campus Newman
202.12.126.1    *               255.255.255.255 UH    1500   0       82 ppp5
# l'instradamento HOST alla mia ethernet casalinga
203.18.8.104    *               255.255.255.255 UH    1500   0       74 ppp3
# due delle nostre linee PPP di dial up
203.18.8.64     *               255.255.255.255 UH    552    0        0 ppp2
203.18.8.62     *               255.255.255.255 UH    552    0        1 ppp1
# l'instradamento di rete specifico per la LAN del campus Newman
202.12.126.0    202.12.126.1    255.255.255.0   UG    1500   0        0 ppp5
# l'instradamento alla nostra Ethernet locale (una "super rete"
# formata da due reti di classe C adiacenti)
203.18.8.0      *               255.255.254.0   U     1500   0     1683 eth0
# l'instradamento al dispositivo di loop back
127.0.0.0       *               255.0.0.0       U     3584   0      483 lo
# l'instradamento predefinito a Internet
default         139.130.177.2   *               UG    1500   0     3633 ppp4
```

23.4 Gestione delle email

Il paragrafo precedente ha mostrato come gestire la posta in uscita, semplicemente svuotando la code delle mail una volta che la connessione è attiva.

Se si sta usando una connessione WAN, ci si può accordare con l'amministratore della LAN remota per fare esattamente le stesse cose. Per esempio, all'estremo del Campus Newman della nostra connessione WAN, lo script `/etc/ppp/ip-up` assomiglia a:

```
#!/bin/bash
#
# Script che gestisce le problematiche di instradamento come richiesto
# per dal pppd.
# Solo la connessione a Hedland necessita questa gestione.
#
# Quando la connessione ppp è attiva, questo script viene chiamato con
# i seguenti parametri
#       $1      il nome dell'interfaccia usata da pppd (p.es. ppp3)
#       $2      il nome del dispositivo tty
#       $3      la velocità del dispositivo tty
#       $4      l'indirizzo IP locale per l'interfaccia
#       $5      l'indirizzo IP remoto
#       $6      il parametro specificato dall'opzione 'ipparam' di pppd
#
case "$5" in
        203.18.8.4)
                /usr/sbin/sendmail -q
                ;;
        *)
esac
exit 0
```

Comunque se si il proprio ISP fornisce solamente un aconnessione PPP ad IP dinamico, è necessario scaricarsi la propria email dal proprio account nella macchina del provider. Ciò solitamente viene fatto usando il POP (Post Office Protocol). Questo processo può essere gestito usando il programma 'popclient', oltre allo script ip-up utile per automatizzare anche questo processo!

Si crei semplicemente uno script `/etc/ppp/ip-up` che contenga la chiamata appropriata di popclient. Per il mio portatile su cui è installata la Red Hat (e che mi porto dietro nei miei viaggi), uso:

```
popclient -3 -c -u hartr -p <password> kepler.hedland.edu.au |formail -s procmail
```

Si può usare slurp o quant'altro per fare la stessa cosa per le news, e così via. Si ricordi che lo script ip-up è uno script bash standard e quindi può essere usato per automatizzare QUALSIASI cosa che dev'essere svolta ogni volta che una certa connessione PPP viene fatta.

24 Usare /etc/ppp/ip-down

Si può creare uno script che verrà eseguito ogni volta che la connessione termina. Questo è `/etc/ppp/ip-down`. Può essere usato per annullare qualsiasi cosa speciale fatta nel corrispondente script /etc/ppp/ip-up.

25 Problematiche di instradamento su una LAN

Se si è connessi ad una LAN ma si vuole comunque usare PPP sulla propria macchina Linux personale, è necessario configurare le cose in modo che i pacchetti vengano instradati, a seconda dei casi, dalla propria macchina verso la LAN (attraverso la propria interfaccia Ethernet) o verso il server PPP remoto e oltre.

Questa sezione NON ha intenzione di insegnare niente sull'instradamento, tratta semplicemente un semplice caso speciale di instradamento (statico)!

Suggerisco caldamente di leggere la Linux Network Administrator Guide (NAG) se NON si ha familiarità con l'instradamento. Anche il libro della O'Reilly "TCP/IP Network Administration" tratta questo argomento in una forma molto comprensibile.

La regola base di un instradamento statico è che l'instradamento PREDEFINITO dovrebbe essere quello che punta al MAGGIOR numero di indirizzi di rete. Per le altre reti, si aggiungano degli instradamenti specifici nella tabella di instradamento.

La SOLA situazione che ho intenzione di trattare qui è quella in cui la propria macchina Linux è in una LAN che non è connessa ad Internet, e ci si vuole connettere ad Internet per uso personale mantenendo la connessione alla LAN.

Prima di tutto, ci si assicuri che il proprio instradamento Ethernet sia impostato agli indirizzi di rete specifici disponibili nella propria LAN, e NON impostato come instradamento predefinito!

Lo si verifichi usando il comando route, che dovrebbe mostrare qualcosa di simile a quanto segue:

```
[root@hwin /root]# route -n
Kernel routing table
Destination   Gateway  Genmask         Flags MSS    Window Use Iface
loopback      *        255.255.255.0   U     1936   0       50 lo
10.0.0.0      *        255.255.255.0   U     1436   0      565 eth0
```

Se la propria interfaccia Ethernet (eth0) punta all'instradamento predefinito (la prima colonna mostrerebbe "default" nella riga di eth0), si deve modificare il proprio script di inizializzazione della Ethernet per farla puntare a dei numeri di rete specifici piuttosto che all'instradamento predefinito (si consulti il Net3 HOWTO e la NAG).

Ciò permetterà a pppd di impostare il proprio instradamento predefinito come mostrato qui sotto:

```
[root@hwin /root]# route -n
Kernel routing table

Destination   Gateway       Genmask         Flags MSS    Window Use Iface
10.144.153.51 *             255.255.255.255 UH    488    0        0 ppp0
127.0.0.0     *             255.255.255.0   U     1936   0       50 lo
10.1.0.0      *             255.255.255.0   U     1436   0      569 eth0
default       10.144.153.51 *               UG    488    0        3 ppp0
```

Come si può vedere, c'è un instradamento di tipo host verso il server PPP (10.144.153.51) tramite ppp0 ed anche un instradamento predefinito che usa il server PPP come gateway.

Se si necessita un impostazione più complessa di questa, si leggano i documenti sull'instradamento menzionati in precedenza e si consulti un esperto nel proprio sito!

Se la propria LAN ha già degli instradamenti, allora probabilmente saranno già stati impostati dei gateway alle altre reti disponibili nel proprio sito. In questo caso, si dovrebbe COMUNQUE far puntare il proprio instradamento predefinito all'interfaccia PPP, e rendere gli altri instradamenti specifici solamente per le reti che servono.

25.1 Note sulla sicurezza

Quando si configura una macchina Linux su una LAN già esistente per connettersi ad Internet, potenzialmente si sta aprendo l'intera LAN ad Internet, e agli hacker che ci sono. Prima di far ciò, suggerisco di

consultare il proprio ammistratore di rete e le politiche di sicurezza del proprio sito. Se la propria connessione PPP ad Internet è usata per attaccare con successo il proprio sito, per quanto poco si subiranno le ire dei propri colleghi e degli amministratori di sistema e di rete. Inoltre si potrebbe andare in contro a problemi assai più seri.

Prima di connettere una LAN ad Internet, si dovrebbero considerare le implicazioni di sicurezza dettate dalla connessione DINAMICA, e quindi far riferimento al già citato libro della O'Reilly "Building Internet Firewalls"!

26 Configurare un server PPP

Come già detto, ci sono molti modi per farlo. Quel che presento è il modo nel quale io lo faccio (usando una scheda seriale multiporta della Cyclades) e un insieme di linee telefoniche a selezione rotativa.

Se non vi piace il metodo qui presentato, sentitevi liberi di farlo a modo vostro. Avrei comunque piacere di includere altri metodi nelle versioni future di questo HOWTO. Quindi, vi invito ad inviarmi i vostri commenti e metodi!

Si noti, che questa sezione tratta solamente la configurazione di Linux come server PPP. Non ho intenzione di includere (mai) informazioni sulla configurazione di terminal servers speciali o altro.

Inoltre, devo ancora fare esperimenti con le shadow password (ma lo farò prima o poi). Le informazioni qui presentate quindi NON comprendono tutti gli ammenicoli che sono richiesti dalla shadow suite.

26.1 Compilazione del kernel

Si applicano ancora tutte le considerazioni precedenti sulla compilazione del kernel e sulla relazione tra la versione del kernel e quella di pppd. Questa sezione assume che si siano già lette le sezioni precedenti di questo documento!

Per un server PPP, si **DEVE** includere l'IP forwarding nel proprio kernel. Si possono voler includere anche altre caratteristiche (come IP firewall, accounting ecc ecc).

Se si usa una scheda seriale multiporta, ovviamente si devono anche includere nel kernel i driver necessari!

26.2 Panoramica del sistema server

Noi offriamo account in PPP (e SLIP) e account shell usando la stessa coppia nome utente/password. Ciò ha il vantaggio (per noi) che gli utenti necessitano solo di un account e lo possono usare per tutti i tipi di connettività.

Poiché siamo un'organizzazione educativa, non facciamo pagare il nostro staff e gli studenti per l'accesso, e quindi non dobbiamo preoccuparci di problematiche di accredito e messa in conto (accounting e charging).

Abbiamo un firewall tra il nostro sito ed Internet, e ciò restringe alcuni accessi agli utenti in quanto le linee di dial up sono dentro il nostro firewall (di Internet) (per ragioni abbastanza ovvie, i dettagli degli altri firewall interni non sono qui presentati, e sono comunque irrilevanti).

Il processo che un utente deve seguire per stabilire una connessione PPP nel nostro sito (una volta che sia in possesso di un account valido naturalmente) è:

- chiamare il dialer rotativo (ovvero un unico numero telefonico connesso ad un banco di modem, dei quali verrà usato il primo libero);
- fare il login usando un nome utente e password validi;

- al prompt della shell, lanciare il comando `ppp` per avviare il PPP sul server;
- avviare il PPP sul proprio PC (il fatto che questi funzioni in Windows, DOS, Linux MAC OS o quant'altro è un suo problema).

Il server usa un file `/etc/ppp/options.ttyXX` per ognuna delle porte d'ingresso che imposta il numero IP remoto per l'allocazione degli IP dinamici. Il server usa instradamenti proxyarp per i client remoti (impostati tramite l'apposita opzione di pppd). Ciò ovvia la necessità di far girare routed o gated.

Quando gli utenti riappendono dalla loro parte, il pppd se ne accorge e dice al modem di riattaccare, portando giù, nello stesso tempo, la connessione PPP.

26.3 Procurarsi il software che serve

Sarà necessario il software seguente:

- Linux, propriamente compilato per includere le opzioni necessarie;
- la giusta versione di pppd per il proprio kernel;
- un programma 'getty' che gestisca in modo intelligente le comunicazioni via modem.
  Noi usiamo getty_ps2.0.7h, ma anche mgetty è altamente considerato. Da quel che ho capito mgetty riesce a riconoscere una chiamata che sta usando pap/chap (pap è lo standard per Windows95) e a chiamare automaticamente pppd, ma non ho ancora avuto il tempo di provarlo;
- un domain name server (DNS) funzionante ed accessibile agli utenti in dial up.
  Si dovrebbe veramente far girare un proprio DNS se possibile...

26.4 Configurare un accesso standard (accesso shell)

Prima di poter configurare il proprio server PPP, la propria macchina Linux dev'essere in grado di gestire accessi standard di dial up.

Questo HOWTO NON tratta la configurazione di questo. Si veda la documentazione del getty scelto e il Serial HOWTO per informazioni in proposito.

26.5 Impostare i file delle opzioni di PPP

Bisognerà impostare un `/etc/ppp/options` generico con le opzioni comuni per tutte le porte di dial up. Le opzioni da noi usate sono:

```
asyncmap 0
netmask 255.255.254.0
proxyarp
lock
crtscts
modem
```

Nota: noi NON usiamo nessun instradamento (ovvio) ed in particolare non c'è l'opzione defaultroute. La ragione di ciò è che tutto quello che è richiesto di fare (come server PPP) è di instradare i pacchetti **dal** client PPP verso la propria LAN/Internet e di instradare i pacchetti dalla propria LAN ed oltre **verso** il client.

Tutto ciò che serve per far questo è un instradamento di tipo host verso la macchina client e l'uso dell'opzione 'proxyarp' di pppd.

L'opzione 'proxyarp' imposta (sorpresa) una voce di proxy ARP nelle tabella ARP del server PPP che in pratica dice 'spedisci a me tutti i pacchetti destinati al client PPP'. Questo è il modo più semplice per impostare un instradamento ad un unico client PPP, ma non la si può usare se si fa l'instradamento tra due LAN: si devono usare gli appositi instradamenti di rete i quali non possono usare proxy ARP.

Quasi certamente si vorrà fornire un'allocazione dinamica di numeri IP agli utenti in dial up. Ciò lo si può fare allocando un numero IP per ognuna delle porte di dial up. Ora si crei un `/etc/ppp/options.ttyXX` per ognuna delle porte.

In questo si metta semplicemente il numero IP locale (server) ed il numero IP che viene usato per quella porta. Per esempio:

```
kepler:slip01
```

In particolare, si noti che si possono usare nomi host validi in questo file (ho scoperto che ricordo solamente i numeri IP delle macchine e dei dispositivi critici nella mia rete, i nomi sono molto più significativi!).

26.6 Impostare pppd per permettere agli utenti di eseguirlo (con successo)

Poiché l'avvio di una connessione PPP implica la configurazione di un dispositivo del kernel (un'interfaccia di rete) e la manipolazione della tabella di instradamento del kernel, sono richiesti privilegi particolari, in pratica i pieni privilegi di root.

Fortunatamente, pppd è stato progettato per essere eseguito in modo sicuro in setuid a root. Quindi bisognerà fare

```
chmod u+s /usr/sbin/pppd
```

Se si vanno a vedere i permessi del file, dovrebbero apparire come

```
-rwsr-xr-x   1 root     root        74224 Apr 28 07:17 /usr/sbin/pppd
```

Se non lo si fa, gli utenti non saranno in grado di attivare la loro connessione ppp.

26.7 Impostare un alias globale per pppd

Per poter semplificare le cose ai nostri utenti PPP, abbiamo creato un alias globale (in /etc/bashrc) cosicché con un semplice comando possano lanciare ppp sul server una volta fatto il login.

Questi può essere una cosa del tipo

```
alias ppp="exec /usr/sbin/pppd -detach"
```

Quello che fa è:

- exec : questo implica il rimpiazzo del programma in esecuzione (in questo caso la shell) con il programma che è eseguito;
- pppd -detach : avvia pppd e NON fa il fork in background. Ciò assicura che quando il pppd esce non ci sono ancora processi in giro.

Quando un utente fa il login in questo modo, apparirà nell'output del comando 'w' come

```
  6:24pm  up 3 days,  7:00,  4 users,  load average: 0.05, 0.03, 0.00
User     tty       login@  idle   JCPU   PCPU  what
hartr    ttyC0     3:05am  9:14                -
```

E questo è quanto...vi avevo detto che questo era un semplice sistema server PPP!

27 Usare PPP su una una connessione null modem (seriale diretta)

È molto semplice, non c'è un modem in mezzo e quindi le cose sono molto più semplici.

Prima di tutto, si scelga una macchina come 'server', si configuri getty su una porta seriale in modo da poter verificare che c'è la connettività usando minicom per accedere alla porta seriale nel 'client'.

Una volta che questo funziona, si può rimuovere getty A MENO CHE non ci si voglia assicurare che la connessione venga convalidata usando la coppia nome utente/password come per la connessione in dial up. Poiché si ha il 'controllo fisico' di entrambe le macchine, presumo che NON lo si voglia fare.

Ora, sul server, si rimuova getty e ci si assicuri che le porte seriali su entrambe le macchine siano state configurate correttamente usando 'setserial'.

Tutto quel che bisogna fare adesso è di far partire pppd su entrambi i sistemi. Assumerò che la connessione usi /dev/ttyS3 su entrambe le macchine. Quindi su entrambe si esegua il comando:

```
pppd -detach crtscts lock <IP locale>:<IP remoto> /dev/ttyS3 38400 &
```

Ciò attiverà il collegamento, ma non si è ancora specificato un instradamento. Si può verificare la connessione facendo ping da una macchina verso l'altra e viceversa. Se funziona, si disattivi il collegamento killando uno dei due processi pppd.

L'instradamento di cui si avrà bisogno naturalmente dipende esattamente da cosa si vuole provare a fare. Generalmente, una delle macchine sarà connessa ad una Ethernet (e ad altro) e quindi l'instradamento necessario sarà esattamente quello richiesto per un server e un client PPP.

Quindi nella macchina equipaggiata con la Ethernet, il comando pppd potrebbe essere:

```
pppd -detach crtscts lock proxyarp <IP locale>:<IP remoto> /dev/ttyS3 38400 &
```

e sull'altra macchina:

```
pppd -detach crtscts lock defaultroute <IP locale>:<IP remoto> /dev/ttyS3 38400 &
```

Se si stanno connettendo due reti (usando una connessione seriale!) o si hanno necessità di instradamento più complesse, si può usare /etc/ppp/ip-up in un modo analogo a quello mostrato in precedenza.

Robert Hart
Port Hedland, Western Australia
Melbourne, Victoria, Australia Agostost/Ottobre 1996 Gennaio/Marzo 1997

Traduzione di **Giovanni Bortolozzo**
Febbraio/Maggio 1998