Come funziona Linux: La connessione PPP

Dodicesima parte

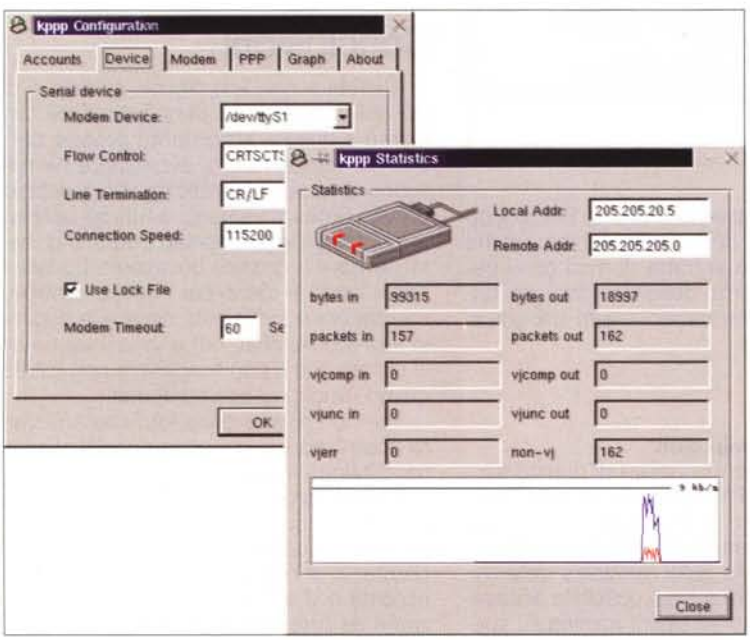
di Giuseppe Zanetti

Il protocollo di rete che interessa maggiormente l'utente medio è quasi sicuramente il PPP, in quanto fra le altre cose consente il collegamento via modem ad un provider Internet. In questa puntata impareremo come funziona in Linux.

PPP, acronimo di Point to Point Protocol, permette di fare un collegamento TCP/IP (oppure con altri protocolli come IPX) usando una connessione punto-a-punto fra due macchine.

Il metodo più comune consiste nell'utilizzare una linea telefonica ed un modem, ma è possibile anche usare PPP per collegare fra loro due macchine mediante un cavo seriale "null modem" oppure un "tunnel" realizzato appoggiandosi sopra un telnet o un collegamento criptato tipo ssh.

Mediante PPP è possibile sia collegare ad Internet (o ad una rete privata) un singolo computer, che fare il routing di pacchetti IP fra reti diverse.



Installazione ed uso di Kppp, uno strumento di configurazione del PPP per l'ambiente KDE

Client e server PPP

Anche se si è soliti riferirsi alla macchina che instaura la chiamata PPP col termine di "client" e a quella che la riceve come "server", trattandosi di un collegamento peer-to-peer, non vi è differenza nel protocollo utilizzato. In Linux si utilizza pertanto lo stesso software, ovviamente configurato in modo diverso, sia per "chiamare" che per gestire i collegamenti PPP in entrata.

Funzionamento di PPP

Il funzionamento del PPP in Linux è demandato in parte al kernel e in parte al "demone" pppd, fornito di serie in tutte le distribuzioni di Linux o prelevabile dal sito ftp://ftp.linuxcare.com.au/pub/ppp/, il quale si occupa di fare la chiamata e fornisce la parte del protocollo che gestisce la linea telefonica.

PPP è composto da tre strati: un metodo per incapsulare pacchetti di dati su una linea seriale,

un protocollo che gestisce il link (LCP, Link Control Protocol) ed una famiglia di protocolli che si occupa di far girare sopra PPP i differenti protocolli di rete (NCP, Network Control Protocols).

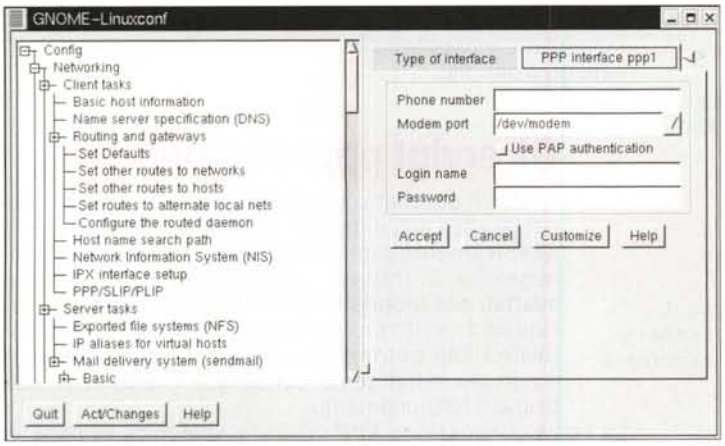
In Linux l'incapsulamento dei dati viene gestito da un driver interno al kernel, mentre il demone esterno pppd, funzionante come un programma utente, si occupa di gestire LCP, e la parte di NCP che permette di far funzionare il protocollo IP (IPCP, IP Control Protocol).

Supporto del PPP nel kernel

Per prima cosa è perciò necessario controllare di avere abilitato il supporto per il PPP nel kernel, cosa che può avvenire i fase di compilazione rispondendo "Yes" alla relativa richiesta, oppure caricando un modulo a run-time:

```
# insmod ppp.o
```

La presenza del supporto per il PPP è confermata dal seguente messaggio del kernel:



Configurazione del PPP usando Linuxconf

```
PPP: version 2.3.7 (demand dialling)
PPP line discipline registered.
```

Il demone pppd

La versione pppd da usare dipende da quella del kernel installato, poichè nelle diverse versioni di Linux cambiano alcune strutture interne. Con un kernel della serie 2.0 è necessario utilizzare almeno la versione 2.2 di pppd.

Poichè il PPP ha bisogno di attivare le opportune interfacce di rete e di creare le tabelle di routing, è necessario avviarlo con i permessi di root. Per non doversi collegare come root ogni volta che si deve usare il PPP, è possibile rendere il programma pppd setuid di root:

```
# chmod u+s /usr/sbin/pppd
# ls -l /usr/sbin/pppd
drwsr-xr-x  1 root  root  95225 Jul 11 00:27 /usr/sbin/pppd
```

In questo modo quando un qualunque utente eseguirà il programma, esso girer à automaticamente con i privilegi di root. Ovviamente è necessario porre una certa attenzione prima di rendere un programma setuid, in quanto si potrebbero creare dei problemi di sicurezza. In particolare è bene compiere tale operazione solo su programmi testati e sicuri, che non possano permettere all'utente che li ha lanciati ulteriori operazioni come ad esempio lanciare comandi esterni.

Volendo che solo alcuni utenti possano avviare il PPP, è possibile creare un gruppo "ppp" ed inserirvi la lista degli username desiderati. Si può far ciò sia modificando a mano il file /etc/group che utilizzando gli strumenti messi a disposizione da Linux:

```
# groupadd ppp
# gpasswd -a beppe ppp
Adding user beppe to group ppp
# gpasswd -a pippo ppp
Adding user pippo to group ppp
```

La linea corrispondente nel file /etc/group risultante sarà simile a

```
ppp:x:1060:beppe,pippo
```

Si devono poi assegnare il gruppo ppp e gli opportuni permessi al demone pppd:

```
# chgrp ppp /usr/sbin/pppd
# chmod 750 /usr/sbin/pppd
# chown u+s /usr/sbin/pppd
```

Il file risultante avrà i seguenti permessi:

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

Sarebbe bene assegnare permessi opportuni anche agli script che controllano la connessione (/usr/sbin/ppp-on e /usr/sbin/ppp-off).

```
-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
```

Configurazione della porta seriale

Per evitare colli di bottiglia nel trasferimento di dati fra computer e modem, è necessario tenere la velocità della porta seriale (collegamento fra DTE e DCE) la più elevata possibile.

Si noti che questa non corrisponde alla velocità con cui i dati attraversano la linea telefonica, ma dovrebbe essere maggiore, in quanto, sfruttando la compressione, i modem possono trasferire una quantità di dati anche molto maggiore rispetto alla loro velocità nominale.

La maggior parte dei modem recenti supporta come velocità standard nella porta seriale i 115.200 bps, ma essa è utilizzabile solamente se il computer dispone di una UART di tipo 16550A. Essa è presente su tutti i computer attuali ma tale limitazione deve essere tenuta in considerazione se si sta installando Linux su un computer vecchio (496 o primi Pentium). In questo caso per verificare il tipo di seriale in uso si può ricorrere al comando "setserial -a /dev/ttySx", dove la lettera x corrisponde alla porta da testare.

Le versioni di Linux antecedenti la 2.0 differenziavano le porte seriali se usate in ingresso (/dev/ttySx) o in uscita (/dev/cuax). Dalla 2.0 si utilizza /dev/ttySx in entrambi i casi. Le porte sono numerate a partire da 0, cosicchè /dev/ttyS0 corrisponde alla prima porta presente nella macchina, che in Windows si chiama COM1:.

In alcune versioni del kernel di Linux la velocità di 115.200 bbs non è accessibile direttamente, ma è necessario prima configurare in modo opportuno la porta seriale mediante il programma "setserial":

```
# setserial /dev/cua1 spd_hi
```

Esso permette di configurare anche altre caratteristiche come l'IRQ o l'indirizzo di I/O, cosa particolarmente utile nel caso si utilizzino porte "multi-seriali"

```
# setserial /dev/cua28 uart 16550A port 0x160 irq 12
```

Per testare il funzionamento corretto della porta seriale e del modem prima di continuare l'installazione del PPP si può usare un programma di emulazione di terminale come minicom.

Collegare due reti usando PPP

Quando si avvia il demone pppd, viene creata una interfaccia di rete ppp0 (oppure ppp1, ppp2, ... nel caso si abbiano più connessioni contemporanee).

```
# ifconfig ppp0
ppp0 Link encap:Point-to-Point Protocol
     inet addr:213.213.48.118  P-t-P:213.255.6.250
       Mask:255.255.255.255
     UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
     RX packets:21 errors:0 dropped:0 overruns:0 frame:0
     TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:30
```

Essa può essere gestita come una normale interfaccia di rete, ad esempio per fare il routing di traffico IP verso reti diverse:

```
route add -net 193.43.99.0 netmask 255.255.255.0 dev ppp0
```

Collegare il proprio computer ad un provider Internet

Si tratta dell'utilizzo più frequente: all'interfaccia associata al PPP viene assegnata una route di default (vedremo essa viene creata automaticamente dal demone pppd se si specifica l'opzione "defaultroute").

Nel caso di collegamento ad un provider è necessario per prima cosa ottenere dal proprio fornitore i dati necessari, che sono di norma i seguenti:

- numero di telefono del POP del provider
- tipo di autenticazione utilizzato (PAP, CHAP o script send-expect)
- username e password
- indirizzo dei DNS

Gli script ppp-on e ppp-off

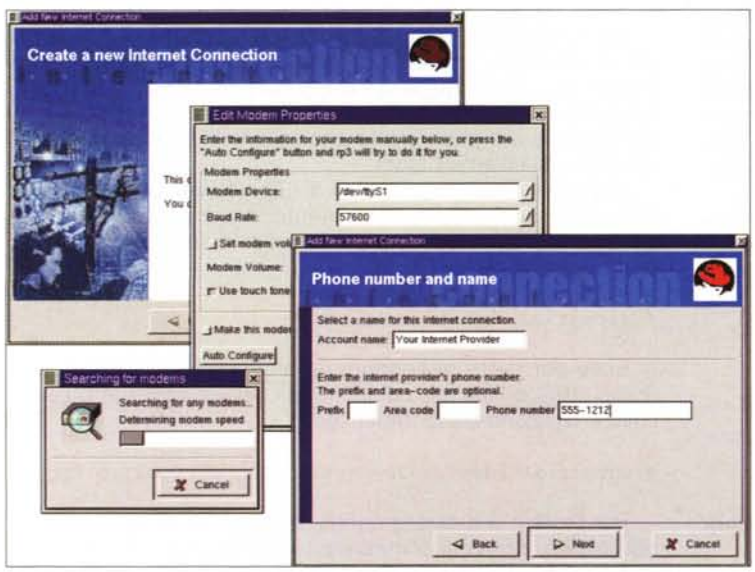
Con il pacchetto del software pppd vengono forniti gli shell script, ppp-on e ppp-off, che si usano rispettivamente per iniziare e per abbattere la connessione. Si tratta di esempi che devono essere adattati alle proprie esigenze. Come è spiegato nel riquadro, esistono dei programmi più complessi (dialer) che permettono di fare le stesse cose in modo più semplice. E' comunque interessante studiarne il funzionamento.

La tipica connessione PPP via linea telefonica avviene in due fasi: per prima cosa viene resettato ed inizializzato il modem. In seguito è necessario fargli comporre il numero di telefono del provider. Una volta che il collegamento è instaurato, si lancia il pppd, il quale si occupa di fare l'autenticazione CHAP/PAP e sovraintende al resto della connessione.

Analiziamo lo script ppp-on presente nella versione 2.3.10 di ppd. Si tratta in realtà solamente di un po'di collante, in quanto la maggior parte del lavoro viene svolta dal programma chat e dal pppd.

Per prima cosa vengono definiti i diversi parametri del collegamento e posti in altrettante variabili d'ambiente, cosa non strettamente necessaria ma che permette di tenerli tutti in un unico posto e di accedervi con maggiore facilità:

```
#!/bin/sh

TELEPHONE=049123456      # Numero di telefono del provider
ACCOUNT=beppe            # Username fornito dal provider
PASSWORD=nonteladico     # Password fornita dal provider
```



RP3 è di serie sulla distribuzione Red Hat

Nel caso ci si colleghi usando un indirizzo IP fisso, è necessario specificarlo, assieme all'indirizzo della macchina (di solito un router o un access server, oppure un'altra macchina Linux) a cui ci si collega e all'eventuale netmask.

Nel caso si utilizzi invece un "IP dinamico", cosa frequente nel caso di collegamento ad un Internet Provider, si può lasciare l'indirizzo fittizio 0.0.0.0 e specificare nella linea di comando del pppd l'opzione "noipdefault".

```
LOCAL_IP=0.0.0.0
REMOTE_IP=0.0.0.0
NETMASK=255.255.255.0
```

Alcune delle variabili appena definite vengono esportate in modo da renderle disponibili anche ai programmi lanciati dal pppd:

```
export TELEPHONE ACCOUNT PASSWORD
```

L'inizializzazione del modem e la composizione del numero di telefono del provider vengono effettuati parlando direttamente col modem mediante i tipici co-

mandi AT. Tale operazione non viene compiuta direttamente da pppd, ma viene demandata ad un programma o script esterno. Tale script viene lanciato direttamente da pppd ed è associato alla stessa porta seriale. Inoltre eredita da esso alcune variabili d'ambiente (oltre a quelle prima definite).

```
DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
```

Come ultima operazione viene avviato il demone vero e proprio. Poichè esso provvede anche all'inizializzazione ed alla gestione della porta seriale, è necessario specificare il nome della stessa (/dev/ttyS0) e il baud rate da utilizzare (nell'esempio 38400).

```
exec /usr/sbin/pppd debug lock modem crtscts /dev/ttyS0 38400 \
        asyncmap 20A0000 escape FF kdebug 0 $LOCAL_IP:$REMOTE_IP \
        noipdefault netmask $NETMASK defaultroute \
        connect $DIALER_SCRIPT
```

Alcune delle opzioni che nell'esempio sono specificate direttamente nella linea di comando di pppd, possono essere in alternativa inserite nel file /etc/ppp/options, una per riga:

```
debug
  lock
  modem
  crtscts
  asyncmap 20A0000
  escape FF
  kdebug 0
  noipdefault
  netmask 255.255.255.0
  defaultroute
  connect /etc/ppp/ppp-on-dialer

In questo modo la linea che lancia il pppd può diventare semplicemente:

  /usr/sbin/pppd /dev/ttyS0 38400 $LOCAL_IP:$REMOTE_IP
```

Nel caso si volesse configurare in modo diverso il PPP a seconda della interfaccia seriale utilizzata, si possono inserire i parametri nel file /etc/ppp/options.ttySn corrispondente.

Nella tabella 1 è riportato il significato dei parametri più importanti di PPP:

Tabella 1

debug	genera, mediante syslog, un log della parte di protocollo gestita dal demone pppd.
kdebug n	indica il livello di debugging per quanto riguarda la parte di protocollo PPP gestita internamente al kernel . Per non generare alcun output si usa "kdebug 0".
lock	crea un file di lock, in modo da evitare conflitti con altri programmi che tentino di accedere alla porta seriale.
modem	tiene in considerazione i segnali generati dal modem (es: carrier detect).
crtscts	indica di utilizzare il controllo di flusso RTS-CTS.
asyncmap chars escape chars	permettono di descrivere quali caratteri non possono essere spediti e rice vuti correttamente dalla linea. Tali caratteri vengono inviati come una se quenza di 2 byte facendoli precedere da un carattere di escape. In questo modo è possibile ad esempio far funzionare il PPP anche attraverso connessioni che filtrano certi caratteri. E' utile ad esempio per appoggiare una connessione PPP sopra un telnet o rlogin, che fanno passare solamente i 7 bit meno significativi.
local:remote	indica l'indirizzo IP locale e quello del router remoto remoto nel caso di collegamento con IP fisso. E' possibile specificare solamente uno dei due oppure usare ":" nel caso venga assegnato un IP dinamico (vedere anche "noipdefault").
noipdefault	indica di non utilizzare indirizzi IP di default ma di accettare quelli inviati dal computer remoto (IP dinamico).
netmask mask	indica l'eventuale netmask da assegnare all'interfaccia di rete ppp0 creata (default 255.255.255.255 che corrisponde ad una connessione punto-a-punto).
defaultroute	crea una route di default verso l'interfaccia PPP. E' utilizzato nel caso ad esempio di collegamento ad un provider Internet.
user username name username	specificano lo username da utilizzare nel l'autenticazione CHAP o PAP (la password si

PPP su ISDN

Esistono due tipi di Terminal Adapter ISDN: quelli con interfaccia seriale, che possono essere usati con il pppd standard come un qualunque modem e quelli su scheda, che necessitano di un opportuno driver e del supporto ISDN4Linux. Rispetto al collegamento via modem, che usa una porta seriale asincrona, è necessario utilizzare la versione "sincrona" del protocollo PPP. Il relativo demone, ipppd. Esso è derivato dal pppd e ultimamente i due programmi sono stati riuniti in un unico software in grado di gestire entrambi i protocolli.

La configurazione dell'ipppd, è abbastanza simile a quanto abbiamo visto per il pppd, salvo alcune differenze che sono ampiamente spiegate nella documentazione inclusa nel pacchetto. La parte più difficile è invece la configurazione della scheda ISDN. Si tratta in realtà di caricare e configurare il driver della scheda ed il supporto per isdn4linux. Abbiamo già trattato l'argomento in un numero passato ed eventualmente lo riprenderemo in mano in una prossima puntata. Ulteriori informazioni su http://www.isdn4linux.de/

	trova in /etc/ppp/chap-secret o /etc/ppp/pap-secret).
mtu n	specificano la massima dimensione di un
mru n	frame di dati che può essere trasmesso o ricevuto attraverso la conessione PPP. Generalmente questo dato viene concorda to automaticamente fra le due macchine.
demand	specificando questo parametro, pppd apre una connessione in modo automatico in presenza di traffico verso l'interfaccia ppp0.
idle n	abbatte automaticamente la connessione dopo un determinato periodo di non utilizzo.

Ulteriori spiegazioni sui parametri di pppd si possono trovare nel manuale in linea (man pppd).

Il programma chat

Per inizializzare il modem e fare la telefonata si utilizza il programma chat, il cui funzionamento è configurabile usando una serie di stringhe di tipo "expect-send": viene spedita (send) una stringa al modem e si attende (expect) da esso una determinata risposta.

Ad esempio, per far comporre al modem un numero di telefono, si deve spedire una stringa del tipo "ATD0491234567" ed attendere il risultato, che può essere "CONNECT", se tutto ha funzionato per il verso giusto, oppure un messaggio di errore.

Se analizziamo lo script /etc/ppp/ppp-on-dialer introdotto nell'esempio precedente, vediamo che le diverse stringhe vanno specificate di seguito sulla linea di comando di chat. Per una migliore leggibilità, è possibile "indentarle" usando degli spazi e spezzare una linea di comando lunga su più linee successive usando il carattere \ prima di un a capo, cosicchè la seguente linea

```
#!/bin/sh

exec 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
```

è del tutto equivalente a:

```
      exec 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
```

Il seguente esempio mostra come resettare il modem (ATZ), inviargli una semplice stringa di inizializzazione (ATM1L3X3) e fargli comporre un numero di telefono (ATD):

```
chat -v "" AT OK ATZ OK ATM1L3X3 OK
ATDT0492345678 CONNECT
```

Il funzionamento è il seguente: chat senza attendere prima nulla dal modem, poichè si richiede una stringa vuota "", gli spedisce la stringa "AT" (viene automaticamente aggiunto un a capo in coda alla stringa) e attende "OK come risposta. Lo stesso viene ripetuto per gli altri comandi AT, con la differenza che nel caso di ATD la stringa attesa è del tipo "CONNECT 33600", che indica che il modem si è connesso a 33600 bps. In realtà il dato sulla velocità nel nostro caso è ininfluente, in quanto chat termina nonappena riceve il connect.

Uno script come quello appena visto non è molto robusto. Esso infatti non è in grado di riconoscere eventuali errori, se non quello dovuto al timeout (ovvero il modem non invia la stringa attesa entro un certo tempo).

La gestione dei messaggi di errore emessi dal modem non è banale, in quanto a parità di comandi inviati possono capitare di volta in volta errori di tipo diverso. Ad esempio se il modem remoto è occupato viene generata la stringa "BUSY", mentre se esso risponde ma fallisce l'handshacking si ottiene il messaggio "NO CARRIER". E' possibile insegnare a chat queste diverse tipologie di errore utilizzando il parametro "ABORT stringa":

```
ABORT BUSY ABORT 'NO CARRIER' '' ATZ OK
ATDT0491234567 CONNECT
```

Così facendo, qualora il modem durante lo script inviasse la stringa "BUSY" o "NO CARRIER", il comando chat terminerebbe, passando a chi l'ha lanciato (in questo caso a pppd) un codice di uscita diverso da zero.

Chat come programma a sè stante

In alternativa a far lanciare chat da pppd, si possono eseguire i due programmi uno in successione all'altro. In questo caso bisogna però aver cura di ridirigere adeguatamente lo standard input e lo standard output di chat verso la porta seriale.

Eventuali messaggi per l'utente possono essere ridiretti verso lo standard error usando 1>&2. Inoltre bisogna prima settare opportunamente il baud rate della porta e resettarla mediante il comando stty:

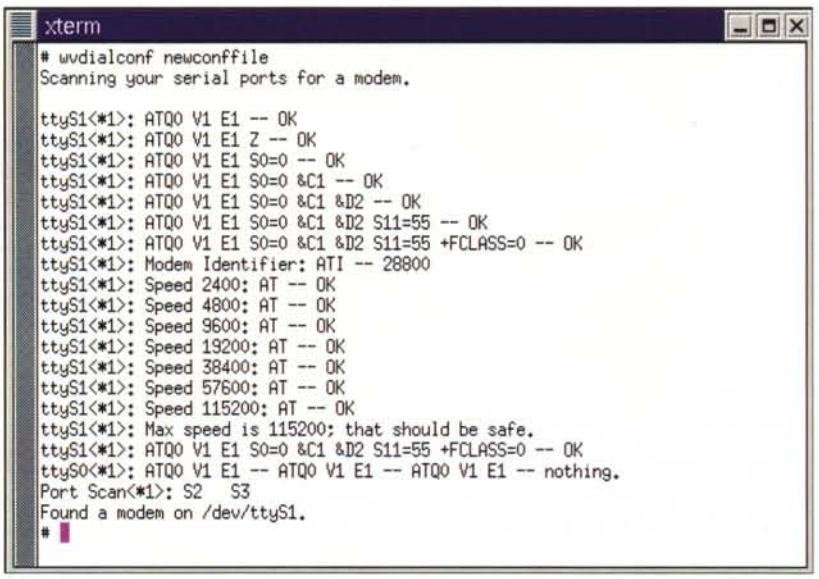
```
(
        echo -n "setting tty ... " 1>&2
        stty 0
        stty 38400 -tostop

        echo "connecting ... " 1>&2

        if chat -v "" AT OK ATZ OK ATD0491234567 CON-
NECT
        then
                echo "avvio PPP" 1>&2
                pppd ....
        else
                echo "chat fallita" 1>&2
        fi
) </dev/ttyS0 >/dev/ttyS0
```

Tale utilizzo di chat è molto comodo anche al di fuori dell'uso in abbinamento al protocollo PPP, in quanto è possibile ad esempio far comporre automaticamente al modem un numero di telefono a partire da un database.

E' anche possibile utilizzare chat su un input diverso dalla porta seriale per automatizzare alcune procedure, tuttavia in questo caso può essere interessante utilizzare il programma expect, in quanto è molto più potente e flessibile (si tratta di un vero e proprio linguaggio di programmazione).

Autenticazione con uno script expect-send

Chat viene usato oltre che per comporre il numero, anche per autenticare l'utente nel caso il computer o il router remoto non avviino direttamente il protocollo PPP ma prevedano prima il login in modo testo, come nel seguente esempio (in grassetto i comandi inviati dall'utente):



Wvdialconf mentre tenta di riconoscere il modem

Strumenti di configurazione facilitata diPPP

Le diverse distribuzioni di Linux mettono a disposizione strumenti utili per semplificare la configurazione del PPP. I più utilizzati sono kppp, WvDial e RP3 (RedHat PPP). In alternativa è possibile utilizzare il "solito" Linuxconf, compatibile con quasi tutte le distribuzioni di Linux.

WvDial

Si tratta di una utility a linea di comando, che perciò è utilizzabile anche senza disporre di un ambiente grafico. Essa è composta da due programmi distinti: wvdialconf e wvdial. Il primo si occupa di riconoscere automaticamente la porta seriale ed il tipo di modem collegato e di generare in base ai dati raccolti un file /etc/wvdial.conf.

Il file così generato verrà poi utilizzato dal dialer wvdial per gestire il collegamento vero e proprio, non prima però di aver aggiunto a mano il numero di telefono del provider. lo username e la password.

```
Phone = 555-1212
Username = my_login_name
Password = my_login_password
```

Per collegarsi è sufficiente lanciare da una sessione di lavoro in modo testo il comando wvdial. Per terminare il collegamento basta invece un semplice CTRL-C.

Volendo una interfaccia grafica, è possibile utilizzare Kwdial (http://www.cnss.ca/~ppatters/KWvDial.html), scritta da Patrick Patterson, adatta all'ambiente KDE, oppure RP3, oggetto del prossimo paragrafo.

La homepage del programma è http://www.worldvisions.ca/wvdial/.

RP3

E' lo strumento di configurazione fornito di serie con le versioni di Red Hat successive alla 6.2. Si trova nel menù "Internet I Dialup Configuration Tool". Esso è basato su wvdial e fornisce un "wizard" che aiuta ad installare in modo semplice il collegamento. Una volta configurato, se lo si lancia dal menù "Internet I RH PPP Dialer", esso funge anche da dialer e da contascatti.

Ulteriori informazioni su RP3 possono essere reperite sul sito della Red Hat http://www.redhat.com/support/manuals/RHL-6.2-Manual/getting-started-guide/ch-ppp.html

Kppp

Viene fornito di serie con l'ambiente di desktop KDE nel menù "Internet". Ha la caratteristica di fornire un ottimo sistema di help in linea che fa apparire dei messaggi "pop up" ogniqualvolta l'utente si posiziona su uno dei bottoni dell'applicazione. Kppp può essere utilizzato sia per configurare la connessione che come "dialer" per attivarla, oltre che per tenere sotto controllo i costi di connessione.

Per ulteriori informazioni si consultino il sito del programma, http://devel-home.kde.org/~kppp/index.html e http://www.redhat.com/support/manuals/RHL-6.2-Manual/getting-started-guide/s1-ppp-kppp.html

Linuxconf

La configurazione del PPP usando Linuxconf è molto semplice: il relativo pannello si trova sotto la voce "Config I Networking I Client Tasks" e permette di configurare anche i protocolli SLIP (Serial Line IP) e PLIP (collegamento tramite porta parallela). Lo stesso pannello può essere utilizzato per attivare o abbattere il collegamento o in alternativa è possibile usare i comandi "netconf –connect" e "netconf --disconnect". Purtroppo mancano alcune funzioni importanti, come la possibilità di autenticarsi usando il protocollo CHAP o uno script di tipo send-expect.

Diald

Permette l'automatizzazione del collegamento PPP in presenza di traffico. Nonostante tale funzione sia supportata di serie nelle versioni recenti di pppd, questo software aggiunge parecchie cose interessanti, come la possibilità di filtrare i pacchetti che mandano on-line la connessione.

```
ATD049123456
CONNECT 14400/ARQ/V32/LAPM/V42BIS

<a capo>

**************************
Welcome to XYZ Corporation
**************************

User Access Verification

Username: beppe
Password: <password>
> ppp
```

Un semplice script per automatizzare tale procedura è il seguente:

```
chat -v "" ATZ OK ATD049123456 CONNECT "\r" erna-
me: beppe ssword: lamiapassword ">" ppp
```

Si noti che il carattere ">" è stato quotato per evitare che venisse interpretato dalla shell come un operatore di ridirezione. Lo stesso per il carattere "\r", che serve per inviare un "a capo".

Per rendere più flessibile gli script è possibile inserire all'interno di una sequenza send-expect delle sottosequenze, separate dal carattere "-":

ername:-\r-ername: beppe ssword: lamiapassword

Questa linea indica che il programma chat deve attendere la stringa "ername:". Se esso non la riceve nel tempo prestabilito viene spedito al modem remoto un "a capo" e si riattende la stessa stringa.
Se invece viene ricevuta la stringa al primo tentativo, lo script continua normalmente.
Quelle appena viste solo solamente alcune delle opzioni usabili nel comando chat. Per ulteriori informazioni si faccia riferimento al manuale in linea (man chat).

Metodi di autenticazione CHAP e PAP

Attualmente la quasi totalità degli Internet Provider per evitare complicazioni agli utenti non usa una autenticazione di tipo "expect-send", ma preferisce ricorrere alle autenticazioni CHAP o PAP, che vengono gestite direttamente dal protocollo PPP. In questo caso l'utente deve comunicare a pppd il proprio username usando le opzioni

```
user beppe
name beppe
```

La password deve essere invece inserita in uno dei file /etc/ppp/chap-secrets oppure /etc/ppp/pap-secrets, a seconda del metodo di autenticazione utilizzato. Entrambi i file possono contenere più linee nel seguente formato:

```
client      server  secret       IP addresses
```

La password che serve viene selezionata dal pppd in base al nome specificato nella opzione "name" o usando l'indirizzo che viene assegnato alla macchina dal router remoto o dall'indirizzo all'altro capo della connessione (o mediante combinazioni degli stessi).
Nel caso di collegamento ad un Provider con IP dinamico, è possibile usare la wildcard * al posto dei dati che non si conoscono:

```
beppe * lamiapassword *
```

Autenticazione MSCHAP

E' possibile usare come metodo di autenticazione una variazione di CHAP chiamata MSCHAP. In questo caso è necessario avere una versione abbastanza recente del demone pppd e si deve obbligatoriamente specificare nella linea di comando o in /etc/options il nome del server a cui ci si sta connettendo:

```
remotename ntserver
```

Lo stesso nome deve essere inserito in /etc/ppp/chap-secret per selezionare la password corretta:

```
beppe ntserver password *
```

Gli script /etc/ppp/ip-up e /etc/ppp/ip-down

Subito dopo che è stata instaurata la connessione, viene eseguito lo script /etc/ppp/ip-up. In questo modo è possibile ad esempio tenere un log delle connessioni o prelevare automaticamente la posta dal proprio provider. Lo script viene chiamato da pppd passandogli sulla linea di comando i seguenti valori: interface-name, tty-device, speed, local-IP-address, remote-IP-address. Essi possono essere utilizzati per selezionare operazioni diverse a seconda del router a cui si è collegati:

```
#!/bin/sh

echo "`date` INIZIO" >>/var/log/isdn.log

case $5 in
       193.43.99.*)
              # sono collegato al mio ufficio
              mount -t smbfs //ntserver/beppe
/home/beppe &
              ;;
       *)
              # sono collegato al Provider
Internet
              /usr/local/bin/PRENDI_EMAIL &
              /usr/local/bin/PRENDI_NEWS &
```

```
                /usr/sbin/sendmail -q &
        ;;
esac
```

Analogamente, lo script /etc/ppp/ip-down viene eseguito al termine della connessione o quando cade la linea:

```
#!/bin/sh

echo "`date` FINE" >>/var/log/isdn.log
```

Debugging e risoluzione di eventuali problemi

L'opzione -v fa in modo che chat generi, mediante syslog, una traccia delle operazioni in /var/log/messages. Ciò facilita enormemente il debugging nel caso qualcosa non funzionasse per il verso giusto.

```
Jan 24 15:55:41 freddy chat[6176]: send (AT^M)
Jan 24 15:55:41 freddy chat[6176]: expect (OK)
Jan 24 15:55:41 freddy chat[6176]: AT
Jan 24 15:55:41 freddy chat[6176]: OK
Jan 24 15:55:41 freddy chat[6176]:  -- got it
Jan 24 15:55:41 freddy chat[6176]: send (ATZ^M)
Jan 24 15:55:41 freddy chat[6176]: expect (OK)
Jan 24 15:55:41 freddy chat[6176]: ATZ
Jan 24 15:55:42 freddy chat[6176]: OK
Jan 24 15:55:42 freddy chat[6176]:  -- got it
Jan 24 15:56:03 freddy chat[6202]: send (atdt049123456^M)
Jan 24 15:56:04 freddy chat[6202]: expect (CONNECT)
Jan 24 15:56:04 freddy chat[6202]: atdt049123456
Jan 24 15:56:27 freddy chat[6202]: CONNECT
Jan 24 15:56:27 freddy chat[6202]:  -- got it
```

Con l'opzione "debug" di pppd si ottiene invece il log del protocollo PPP:

```
Jan 24 15:56:28 freddy kernel: registered device ppp0
Jan 24 15:56:28 freddy pppd[6204]: pppd 2.3.10 started by root, uid 0
Jan 24 15:56:28 freddy pppd[6204]: Using interface ppp0
Jan 24 15:56:28 freddy pppd[6204]: Connect: ppp0 <--> /dev/modem
Jan 24 15:56:30 freddy kernel: PPP BSD Compression module registered
Jan 24 15:56:31 freddy kernel: PPP Deflate Compression module registered
Jan 24 15:56:31 freddy pppd[6204]: Unsupported protocol (0x803f) received
Jan 24 15:56:31 freddy pppd[6204]: local  IP address 193.43.99.222
Jan 24 15:56:31 freddy pppd[6204]: remote IP address 193.43.99.1
...
Jan 24 16:27:49 freddy pppd[6204]: Terminating on signal 2.
Jan 24 16:27:49 freddy pppd[6204]: Connection terminated.
Jan 24 16:27:49 freddy pppd[6204]: Connect time 31.4 minutes.
Jan 24 16:27:49 freddy pppd[6204]: Sent 3370 bytes, received 4663 bytes.
Jan 24 16:27:49 freddy pppd[6204]: Exit.
```

Per un log più accurato si deve abilitare l'opzione "kdebug". I messaggi generati possono essere diretti in un file di log specifico aggiungendo la seguente linea in /etc/syslog.conf:

```
*.=debug                /var/log/debug
```

Risoluzione dei nomi

Affinchè funzioni la risoluzione dei nomi, è necessario aver configurato opportunamente il file /etc/resolv.conf come abbiamo visto la volta scorsa. In alternativa, se il router a cui ci colleghiamo fornisce questo servizio, è possibile ottenere automaticamente l'indirizzo del o dei nameserver, specificando fra le opzioni di pppd "usepeerdns".

Gli indirizzi ottenuti dal provider vengono passati allo script /etc/ppp/ip-up, come variabili d'ambiente DNS1 e DNS2. Essi possono essere utilizzati per aggiornare in modo automatico il file /etc/resolv.conf usando delle linee simili alle seguenti in /etc/ppp/ip-up:

```
echo DNS1 >/etc/resolv.conf
echo DNS2 >>/etc/resolv.conf
```

PPP come server

Abbiamo già accennato al fatto che è possibile utilizzare il PPP anche come server per far connettere altri alla nostra rete. tale operazione è leggermente più complessa, in quanto richiede che si installi un programma che risponda alle chiamate (getty) e che lanci pppd.

Inoltre è necessario gestire in modo opportuno l'autenticazione degli utenti.

In questo caso infatti sarà il nostro PPP a chiedere all'utente remoto di autenticarsi. Per far ciò si deve usare l'opzione "auth" di pppd.

E' possibile assegnare una password diversa ad ogni utente usando il campo client di /etc/ppp/chap-secret, tuttavia esistono diverse possibilità alternative, come usare le stesse password di Linux o un server di autenticazione (Tacacs o radius).

Conclusioni

Abbiamo visto solo pochi esempi di come configurare ed usare il PPP. In realtà si tratta di uno strumento che può essere utilizzato in molti modi differenti.

Per ulteriori informazioni si consulti il manuale del pppd oppure l'apposito "Linux PPP HOWTO", scritto da Corwin Light-Williams e Joshua Drake. Esso è disponibile su http://www.linuxdoc.org/HOWTO/PPP-HOWTO/index.html

MC