

# ***Utilizzare una connessione ADSL con OpenBSD***

Questo documento ha lo scopo di illustrare l'utilizzo di una connessione ADSL, con un modem Alcatel (ma va bene anche un modem dotato di un'interfaccia ethernet) e, naturalmente, un sistema operativo OpenBSD.

Per chi accede ad Internet con l'ausilio del protocollo pptp, puo' continuare a leggere di seguito, altrimenti coloro i quali utilizzano il protocollo pppoe, possono saltare in fondo al documento ignorando questa prima parte.

Se siete sicuri di accedere alla rete con il protocollo pptp vi occorre un file chiamato pptp-obsd.tar.gz, PPTP sta per Point to Point Tunnelling Protocol, lo trovate nei ports date un'occhiata in

*ports/net/pptp,*

oppure lo potete scaricare alle seguenti url:

<http://www.f00bar.net/dl/pptp-obsd.tar.gz>

<http://www.packetstorm.net/html/projects/howto/pptp-obsd.tar.gz>

<http://www.bsdf.org/doc/pptp-open-1.0.2.tar.gz>

una volta scaricati i sorgenti, scompattateli ed installateli:

```
% make  
% make install
```

Se qualcosa non dovesse andare per il verso giusto copiate a mano i file: *pptp*, e *pptp\_callmgr* nella directory */usr/bin*

```
% cp pptp /usr/bin  
% cp pptp_callmgr /usr/bin
```

Create il file */etc/ppp/options*, e inseritegli le seguenti righe:

```
name "NOMELOGIN"  
noauth  
noipdefault  
defaultroute  
debug
```

Naturalmente il campo "NOMELOGIN" deve essere sostituito con la login che vi ha assegnato il vostro provider. Create ora il file chiamato */etc/ppp/pap-secrets*, e inserite al suo interno le seguenti righe:

```
NOMELOGIN 10.0.0.138 PASSWORD
```

Come nel file precedente, il campo NOMELOGIN va sostituito con lo stesso campo usato precedentemente, il campo contenente l'indirizzo va sostituito con l'IP a cui risponde il vostro modem/router (nel mio caso 10.0.0.138), in fine il campo PASSWORD va sostituito con la password che vi ha assegnato il vostro provider.

Vi ricordo che l'indirizzo 10.0.0.138, è l'indirizzo di default che indica fisicamente il mio modem, quindi una volta configurata l'interfaccia alla quale è connesso (vi spiego dopo come fare), potete fare un semplice ping -c1 10.0.0.138, per controllare se è raggiungibile.

L'unico utente che può leggere questo file è l'utente root, quindi settate pure i permessi di lettura e scrittura in modo corretto con il seguente comando:

```
% chmod 600 /etc/ppp/pap-secrets
```

Giunti a questo punto dobbiamo inserire i nomi dei server DNS, del vostro provider nel file */etc/resolv.conf* (se non esiste createlo):

```
nameserver 212.216.112.112
nameserver 212.216.172.62
```

Ho inserito gli ip dei server DNS del mio provider (non dovrei fare pubblicità), voi inserite i vostri.

Come vi dicevo prima, ora dovete configurare la vostra interfaccia di rete alla quale è connesso il modem (Alcatel nel mio caso).

Assegnate all'interfaccia un indirizzo ip che faccia parte della rete a cui è appartiene l'IP del modem. Nel mio caso la mia rete è 10.0.0.0/24, e assegno alla mia interfaccia l'IP 10.0.0.1:

```
% ifconfig INTERFACCIA inet 10.0.0.1 netmask 255.255.255.0 broadcast
10.0.0.255
```

Sostituite il campo INTERFACCIA con il nome del dispositivo, nel mio caso il modem si trova connesso ad una scheda ethernet Realtek, quindi rl0.

Per rendere queste modifiche sull'interfaccia in modo permanente, bisogna modificare il file */etc/hostname.INTERFACCIA* (nel mio caso */etc/hostname.rl0*) nel modo seguente:

```
inet 10.0.0.1 255.255.255.0 NONE media autoselect
```

Bon, ora siete pronti per connettervi,

```
% ptp 10.0.0.138&
```

e per chiudere la connessione potete killare i processi:

```
% kill -9 [numero del processo del pppd]
% kill -9 [numero del processo del ptp]
```

Se avete dei problemi con la connessione fate riferimento al file di log */var/log/messages* potete monitorare il tutto con il seguente comando:

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

Se la vostra connessione alla rete resta in piedi solamente qualche manciata di secondi, molto probabilmente vi trovate di fronte ad un problema con lo pseudo device GRE, che se attivo nel kernel cattura tutti i pacchetti GRE e il PPTP non riesce a vederli !!!! causandone un inesorabile time out.

Se fate riferimento al file di log sopracitato (*/var/log/messages*), vi troverete una dicitura simile a questa:

```
LCP:Timeout sending Config-Request
```

Potete provare a disabilitare lo pseudo device GRE usando il comando "*sysctl*"

```
% sysctl -w net.inet.gre.allow=0
```

Se non dovesse funzionare, cosa molto probabile (nella versione 2.9, ad esempio non funziona), allora la soluzione è la ricompilazione del kernel !!! Non temete non è una cosa difficile.

Naturalmente, non serve che ve lo dica io, dovete avere i sorgenti del kernel, se non li avete, scaricateli.

Questa modifica del kernel serve solo a disabilitare lo pseudo device GRE: Aprite il file */usr/src/sys/conf/GENERIC* Commentate la riga:

```
option pseudo-device gre
```

Per commentare una riga intendo l'inserimento del simbolo # Ora entrate nella dir che corrisponde alla vostra macchina (mi riferisco all'architettura) es: */i386* se avete una macchina con architettura intel

```
% cd /usr/src/sys/arch/i386/conf
```

bene, compiliamo il kernel con la modifica che abbiamo appena apportato:

```
% config ./GENERIC  
% cd ../compile/GENERIC  
% make depmod  
% make
```

Bene, una volta compilato il kernel bisogna copiarlo nella directory / con il nome "bsd.2"

```
% cp bsd /bsd.2
```

Rebootate... non è una parolaccia... dovete riavviare la macchina, e durante il prompt del loader, digitiamo "bsd.2" boot:bsd.2 Provate ora a connettervi, dovrebbe essere

andato tutto per il verso giusto, e se vi da fastidio oppure vi reca impedimenti immettere `bsd.2` ad ogni prompt di boot, potete salvare il vostro vecchio "kernel", con il nome di `bsd.old` (ad esempio), e rinominare `bsd.2` con il nome `bsd` in questo modo:

```
% cp /bsd /bsd.old
% mv /bsd.2 /bsd
```

Così facendo il boot caricherà direttamente il kernel da voi compilato. In aggiunta a questo articolo, potrete scaricare uno script che in modo automatico si riconnette alla rete, quando si verificano dei problemi, tramite il crontab: è stato scritto da Peter Chiocchetti.

## Connessione tramite PPPoE

Se per accedere ad Internet utilizzate il protocollo PPPoE, come faccio io con Alice di Telecom, sappiate che non dovete imbattervi nella ricompilazione del kernel ne tanto meno nell'istallazione di applicativi come sopra. Il supporto per il PPPoE e' incluso in user space, sin dalla release 2.9, quindi dobbiamo solamente dare uno sguardo alla configurazione del file `/etc/ppp/ppp.conf` e ad un programmino chiamato `/usr/sbin/pppoe`. Il file `/etc/ppp/ppp.conf` deve contenere le seguenti righe:

default:

```
set log Phase Chat LCP IPCP CCP tun command
set redial 15 0
set reconnect 15 10000
```

pppoe:

```
set device "r10"
disable acfcomp protocomp
deny acfcomp
set mtu 1492
set speed sync
enable lqr
set lqrperiod 5
set cd 5
set dial
set login
set timeout 0
set authname *****
set authkey *****
add default HISADDR
enable DNS
enable mssfixup
```

Tramite il parametro "set device" locato nella linea 6, il ppp, incarica il programma */usr/sbin/pppoe*, che dopo chiamerò per facilità solo pppoe, di gestire l'input e l'output, anziché usare una porta seriale oppure un'altra connessione per la rete.

Daremo uno sguardo più avanti al programma pppoe, ma per ora continuiamo a dare un'occhiata alla linea 9, nella quale troviamo il valore dell'mtu (Maximum Transmit Unit), settato a 1492, ma potrebbe variare a seconda delle esigenze, dato che questo valore, insieme a quello dell'"mru", serve solamente a dare la possibilità ai pacchetti ethernet in uscita, lo spazio per gli headers del pppoe.

Infatti il Maximum Ethernet frame è di 1518 bytes, 14 bytes sono usati dall'header e 4 bytes per il frame-check sequence, restano quindi 1500 bytes di payload.... per questa ragione il Maximum Transmission Unit di un'interfaccia ethernet è solitamente 1500 bytes e questo è il più grande datagramma IP che può essere trasmesso su un'interfaccia ethernet senza frammentazioni. Il PPPoE aggiunge altri sei bytes di overhead, e il protocollo PPP altri due.... il conteggio è presto fatto... restano 1492 bytes per il datagramma IP.

La linea con la dicitura "set speed sync" informa il ppp di usare una decodifica in modo sincrono per i pacchetti inviati tra esso ed il pppoe.

Vi sconsiglio di non abilitare il "protocol compression", ed il "protocomp" (date un'occhiata all' RFC 2516 per ulteriori delucidazioni). Non c'è bisogno di ricordarvi di inserire la userid e password (linea 17 e 18), che il vostro provider vi ha assegnato.

La configurazione è finita ! facile no ?

Ah, naturalmente dovete configurare il file */etc/resolv.conf* (vedi sopra) e lanciare la connessione in questo modo:

```
%ppp -ddial pppoe &
```

lasciate stare per ora i messaggi che vi butta fuori il demone. Il ppp non fa altro che aprire un'interfaccia chiamata "tun" (ad esempio "tun0", oppure "tun1" se la tun0 è già stata attivata), e successivamente va in background.

Il collegamento non sarà in piedi fino a quando i dati uscenti, non saranno trovati sull'interfaccia "tun" e a quel punto ppp, prova a metter su il collegamento in tutto e per tutto.

Nell'arco di tempo in cui il ppp inizia a creare il collegamento (oppure il "link"), i pacchetti ricevuti (compreso il primo), rimarranno in coda per un periodo di circa 2 minuti.

Potete settare questo valore di timeout con l'opzione "set choked [timeout]", dove timeout è un numero che indica in secondi, il lasso di tempo, in cui i pacchetti vengono tenuti in attesa e dopo di che scartati da ppp.

Ho usato -ddial (simile a "-auto") per far sì che il ppp ristabilisca automaticamente il collegamento ogni qual volta esso cessa, per un qualunque motivo :)

Come detto prima il ppp usa il programma */usr/sbin/pppoe* per gestire il traffico sulla scheda di rete, se volete potete usare il comando */usr/sbin/pppoe*, anche per metter su un server PPPoE la sintassi è la seguente:

*pppoe [-s [-p system]] [-v] [-i interface] [-n service]*

Anche qui il parametro "-i interface" è lo stesso, e cioè indica l'interfaccia sulla quale è in ascolto in modem ( nel mio caso la rl0), in caso in cui non venga specificato questo parametro, il programma pppoe userà la prima interfaccia ethernet che trova configurata.

Una cosa da sapere (mi riferisco agli utenti alle prime armi), è che con OpenBSD il nome di un interfaccia non fa riferimento al tipo di connessione, ma al nome del tipo di scheda di rete.

Potete vedere il tipo di scheda di rete durante il processo di boot, oppure successivamente con il comando "dmesg (8).

Ad esempio, io ho due schede Realtek 8139, quindi mi troverò una rl0 e una rl1, se avessi avuto una Intel EtherExpress 16 ed una Novell NE2000 mi sarei ritrovato con una "ie0" ed una "ne1".

Il parametro "-n service", vi permette di usare un nome per il servizio, ad esempio -n Alice, quando negozierete con il vostro server/provider. Di default, il client vi chiederà il nome del servizio, e non lo farà naturalmente il server che risponderà dall'altra parte della connessione.

Se volete specificare i dati di configurazione in modo da usarli per una connessione PPP Over Ethernet, potete usare il parametro "-p server", ma di solito questi dati si maneggiano solo se si vuol metter su un server PPPoE, in altri casi vi consiglio di non toccar nulla.

Per far partire il pppoe come server passate pure il parametro "-s" al comando pppoe, altrimenti se non è menzionato, il pppoe girerà in modalità client, e vi collegerà al vostro gestore.

In fine l'immane flag "-v", che vi aumenta la prolissità (si può dire???) in fase di esecuzione (vi consiglio di includere questo flag per tutti gli usi di pppoe).

Per la configurazione in modalità client del programma dobbiamo modificare solamente il file di configurazione del ppp, cioè quello che abbiamo visto in precedenza.

Vi ricordo che il software gira completamente in user mode, e se cercate altri programmi per l'implementazione del pppoe siete dei kamikaze ! comunque se siete votati all'autolesionismo e dovete smanettare su tutto ciò che che vi capita date un'occhiata al programma rp-pppoe scritto però per NetBSD. Lo potete scaricare qui:

<http://www.roaringpenguin.com/pppoe/#download>

Non so se il pacchetto per NetBSD giri correttamente su Open, comunque io vi ho avvertito...

Una volta scaricato date:

```
%tar xvfz rp-pppoe-3.4.tar.gz
%cd rp-pppoe-3.4/src
%./configure --mandir=/usr/share/man
%make
```

*#make -i install*  
*#adsl-setup*

(Da quello che dicono i coder di Roarin Penguin, c'è un piccolo bug nel codice, che vi conviene ignorare in fase di compilazione "usando l'opzione -i"!!!! dicono che non abbia conseguenze durante l'utilizzo del programma.)

Con questo è tutto, se avete domande, suggerimenti, quesiti, critiche fatelo mandandomi una e-mail a:

Steno Francesco aka "oVERDRIVE"

[overdrive@OpenGeeks.it](mailto:overdrive@OpenGeeks.it) oppure a [overdrive@Linux.it](mailto:overdrive@Linux.it)

Un ringraziamento va fatto a: \* Hopfer Alois from [www.inode.at](http://www.inode.at) for pointing me to the FreeBSD howto \* Lukas Ertl for the FreeBSD howto that can be found at <http://mailbox.univie.ac.at/~le/freebsd+adsl-howto.html> \* the openbsd mailing list people who gave me a lot of hints \* Caff caff@OpenBEER.it