Serial Laplink HOWTO

Willem J. Atsma

Come creare una connessione seriale tra due computer? Questo documento spiega come impostare una connessione “laplink” tra due computer. Avendo avuto qualche problema nel trovare informazioni per far funzionare la connessione, questo documento dovrebbe rendervi il lavoro più facile. Traduzione di Alessio Rolleri (aroller(at)libero.it) e revisione di Luca Bruno (luca(at)unofree.it) e pctips (pctips(at)hardwaretips.com).

Sommario

1. Introduzione

Questo breve documento descrive come impostare una connessione seriale “laplink” tra due macchine Linux o una macchina Linux ed una Windows. Il collegamento seriale consiste in un cavo “null-modem” e in un protocollo di comunicazione: PPP. Ho usato questa impostazione per inviare file da e verso il mio laptop; può essere usata abbastanza bene per installare nuovo software da CDROM quando il laptop non lo possiede. Ho avuto più problemi di quanti ne avrei dovuto incontrare durante l’installazione della connessione seriale. Spero che questo documento aiuti a risolvere i problemi al primo tentativo.

Entrambe le parti client e server della configurazione che descriverò potranno essere effettuate su una macchina Linux o su un altro sistema operativo, purché supporti il protocollo PPP su un cavo null-modem. L’implementazione di MS Windows è risultata leggermente differente dallo standard, richiedendo attenzioni particolari. Questo documento ha due sezioni principali. La prima descrive come implementare la parte server. Questa è la postazione principale, con la porta seriale dedicata ad una connessione PPP. Alternativamente la connessione può essere avviata quando necessario da linea di comando. La seconda parte descrive la configurazione del client che si connette al server per caricare o scaricare file.

Questo documento è stato preparato per la versione 2.4.0 di pppd, sulla versione 2.4.2 del kernel.

Per favore, fatemi sapere se avete particolari difficoltà e/o se avete suggerimenti per migliorare questo documento; il mio indirizzo email è elencato in fondo.

2. Parte Server

2.1. Configurare pppd

La comunicazione passerà attraverso la porta seriale. Ci collegheremo al server attraverso il dispositivo “/dev/ttyS1” (COM2 su DOS/Windows). Sul proprio computer potrebbe essere differente e si dovranno adattare gli esempi seguenti. Io impongo l’uso dell’autenticazione perché uso questa configurazione per fornire accesso ai file su un file server per un gruppo di persone. Dopo aver effettuato il logon, essi possono usare la condivisione file di Samba per copiare i file da e verso il laptop. Se non si è preoccupati per la sicurezza si possono commentare le opzioni relative.

Il protocollo PPP (Point to Point Protocol) fornisce una comunicazione TCP/IP attraverso una connessione seriale. In altre parole: quando si vuole effettuare una navigazione su Internet attraverso un modem, la si sta usando. PPP è implementato su Linux dal demone PPP “pppd”. La sua configurazione viene effettuata attraverso i file della directory “/etc/ppp/”. Useremo i seguenti file:

/etc/ppp/options contiene tutte le opzioni generali per le connessioni PPP

/etc/ppp/options.ttyS1

: contiene le opzioni di PPP specifiche per le connessioni attraverso “/dev/ttyS1”

/etc/ppp/pap-secrets

: contiene le informazioni sull’autenticazione

/etc/inittab

: avvia i servizi di sistema iniziali

Il file “/etc/ppp/options” dovrebbe apparire più o meno così:

```
#/etc/ppp/options
lock
#auth  forza l'autorizzazione da una macchina peer
#login fa fare l'autenticazione dal file di password di sistema
#NOTA: il mio file pap-secret permette l'accesso a chiunque, così;
#      se non viene specificato chiunque può connettersi! Se la propria macchina
#      si connette all'esterno, commentate auth e login e spostateli nel file
#      /etc/options.ttySn
auth
login
```

Le linee che iniziano con "#" sono commenti. Il parametro "lock" indica che verrà creato un file di lock per assicurare un accesso esclusivo al dispositivo seriale, il parametro "auth" indica che il client deve autenticarsi, e "login" indica a pppd di usare i nomi utenti e password di sistema per l'autenticazione.

Notare che pppd controllerà il file "/etc/ppp/pap-secrets" per le informazioni relative a nome utente e password. Una singola linea speciale fa in modo che pppd usi solo le informazioni di sistema. Il file "/etc/ppp/pap-secrets" apparirà così:

```
#/etc/ppp/pap-secrets
# informazioni riservate per l'autenticazione usando PAP
# client                 server                  secret        IP addresses
*                        *                       ""            ppp_laplink_client
```

Questo permette ad ogni macchina con indirizzo IP "ppp_laplink_client" di connettersi al server, senza usare una password (si può vedere perché è comodo avere le voci di ppp_laplink_client nel proprio file "/etc/hosts"). Comunque, il parametro "login" nel file "/etc/ppp/options" fa in modo che il nome utente e la password forniti dal client per l'autorizzazione devono corrispondere *sia* al file "/etc/ppp/pap-secrets" *che* ai nomi utente e password di sistema, in modo che la connessione avrà successo solo se saranno forniti un nome utente e una password validi.

Le opzioni specifiche della linea seriale a cui ci si connette sono poste nel file "/etc/ppp/options.ttySn", dove n è il numero del dispositivo seriale. Il mio server usa "/dev/ttyS1", quindi le opzioni vanno nel file... "/etc/ppp/options.ttyS1".

```
#/etc/ppp/options.ttyS1
asyncmap 0
crtscts
#local indica che il modem non viene usato
local
#silent fa attendere pppd fino a che la connessione è stabilita dall'altra parte
silent
#queste sono le voci che esistono nel file /etc/hosts
#il collegamento non funzionerà se questo è alla fine del file - l'ordine conta!
ppp_laplink_server:ppp_laplink_client
#auth  forza l'autorizzazione dalla macchina peer
#login fa fare l'autenticazione dal file di password di sistema
#NOTE: il mio file pap-secrets permette l'accesso a chiunque, cos&igrave; se non specificato
#         chiunque pu connettersi! Se questo è un server che non user&agrave; mai PPP
#         con connessioni esterne, si dovrebbe spostare auth e login nel file /etc/ppp/options
#auth
#login
```

```
#usare PAP e non CHAP per l'autenticazione
require-pap
115200
```

Tutte queste opzioni sono ben documentate nella pagina man di pppd; alcune opzioni chiave sono spiegate nei commenti del file. Il parametro “crtscts” indica a pppd di usare il controllo di flusso hardware. È raccomandato perché è più veloce. In alternativa si può specificare “xonxoff” per usare il controllo di flusso software - si vorrà specificare quest'ultimo se il proprio cavo null modem non si connette alle linee RTS/CTS (improbabile se comprate il cavo in un negozio). “115200” specifica la velocità di trasmissione dei dati - se avete problemi nel connettervi potreste tentare ad una velocità inferiore. Nella pagina man di termios si possono trovare le impostazioni valide per la velocità. La scelta di un parametro non valido genererà un messaggio di errore del demone pppd che chiaramente lo riporterà ad un qualche valore di default.

Notare che le opzioni “auth” e “login” sono commentate, perché sono specificate nel file generale di opzioni. Se si usa il computer anche per connettersi ad un ISP, si vorranno specificare qui anziché nel file “/etc/ppp/options”, altrimenti quando vi connettete all'ISP gli verrà chiesto di autenticarsi e questo probabilmente non succederà. Esse non sono specificate in questo file perché se si hanno altre connessioni PPP in entrata o le si avranno nel futuro, si vuole essere certi che verranno sempre autenticate. Ricordarsi che il file “pap-secrets” mostrato qui non fornisce alcuna protezione.

Infine, la voce “ppp_laplink_server:ppp_laplink_client” specifica gli indirizzi IP locali e remoti dopo che la connessione è stabilita. Si possono usare indirizzi IP effettivi (es. 192.168.0.1:192.168.1.1) o voci dal file “/etc/hosts”, come ho fatto io. Il vantaggio di quest'ultima soluzione consiste nell'usare in seguito i nomi per riferirsi ad essi. Questi indirizzi sono riservati per le reti locali e non esistono su Internet, così si evitano possibili conflitti. Dopo che la connessione è stabilita, il client può riferirsi al server attraverso l'indirizzo ppp_laplink_server (192.168.0.1), e il server può riferirsi al client con l'indirizzo ppp_laplink_client (192.168.1.1).

Si possono usare diverse voci nel file “/etc/ppp/pap-secrets” per permettere l'accesso solo a utenti selezionati. Io uso il protocollo PAP per l'autenticazione; si può usare il protocollo CHAP se si desidera - la configurazione è la stessa, usando il file “chap-secrets”. Per queste ed altre opzioni si possono consultare le pagine man e la documentazione indicata alla fine del documento.

2.2. Una installazione "alla getty" di pppd

Si può far partire il demone pppd all'avvio del sistema e fare monitorare una linea seriale a propria scelta. Un modo elegante per fare questo consiste nel modificare il file “/etc/inittab”. Questo file contiene informazioni per l'inizializzazione del sistema. Aggiungere le linee seguenti al file:

```
# Avvio di pppd per la connessione seriale laplink.
pd:2345:respawn:/usr/sbin/pppd /dev/ttyS1 nodetach
```

Questo si legge come segue: per i runlevel 2, 3, 4 e 5 avviare “/usr/sbin/pppd /dev/ttyS1 nodetach” e se termina (alla fine della connessione) effettuare un respawn (farne partire uno nuovo). L'opzione “nodetach” fa in modo che pppd rimanga connesso al terminale che lo ha avviato, invece che effettuare un fork e terminare. Questa opzione è necessaria perché altrimenti il processo “init” ne genererà un altro nuovo immediatamente. Le altre voci del file inittab specificano i processi getty da avviare sui terminali seriali (tty); la loro inizializzazione assomiglia molto a questa.

Per attivare la nuova configurazione scrivere:

```
[root@griis /root]# /sbin/init q
```

2.3. Avviare il server quando necessario

Se ci si vuole connettere al server solo occasionalmente, si potrebbe preferire di avviare la connessione manualmente. Tutte le impostazioni rimangono le stesse; si può far partire il server scrivendo semplicemente:

```
/usr/sbin/pppd /dev/ttyS1 nodetach
```

dalla linea di comando. L'opzione "nodetach" non è davvero necessaria, ma rende più semplice uccidere il processo premendo "ctrl-c".

2.4. Servire client Windows

Sfortunatamente l'implementazione di MS Windows non è del tutto standard. Prima di iniziare la connessione PPP essa richiede lo scambio delle stringhe di testo "CLIENT" (dal client) e "CLIENTSERVER" (dal server). La linea seguente deve essere aggiunta al file "/etc/ppp/options.ttyS1" per conformarsi ad un client Windows:

```
connect 'chat -v -f /etc/ppp/scripts/winclient.chat'
```

Poi si crea la directory scripts e il file chat "/etc/ppp/scripts/winclient.chat":

```
TIMEOUT 3600
CLIENT CLIENTSERVER\c
```

L'opzione di connect permette di specificare un programma per trattare lo scambio delle stringhe prima della connessione. Normalmente viene usato il programma "chat"; controllare il manuale per maggiori dettagli. Lo script si occupa del problema della connessione Windows. Esso non è necessario quando ci si connette con una Linux box.

3. Parte Client

Una volta configurato il server, la configurazione del client è semplice. Il file "/etc/ppp/options" contiene una singola voce:

```
#/etc/ppp/options
lock
```

La porta seriale del mio client è "/dev/ttyS0" (COM1), così ho dovuto anche creare il file "/etc/ppp/options.ttyS0":

```
#/etc/ppp/options.ttyS0
115200
crtscts
local
```

```
user zaphod
noauth
```

Ci sono solo alcune nuove opzioni in questo file. Notare che la velocità del server deve essere la stessa del client (qui è 115200). Viene specificato il nome utente del client con “user zaphod”. Questo nome utente deve corrispondere ad una voce del file “/etc/ppp/pap-secrets” del client e deve essere un nome utente valido del server. L’opzione “noauth” specifica che la macchina peer (server) non deve autenticarsi al client.

L’ultimo pezzo: impostare il client name (nome utente) e il suo secret (password) nel file “/etc/ppp/pap-secrets”:

```
#/etc/ppp/pap-secrets
# Informazioni segrete per l'autenticazione tramite PAP
# client                server                  secret               IP addresses
zaphod                  *                       gargleBlaster
```

In questo modo l’utente “zaphod” usa la password “gargleBlaster”, che lo farà accedere al sistema. Notare che il file contiene informazioni sensibili, quindi assicuratevi di impostare i permessi correttamente.

3.1. Avviare il client

Da linea di comando, scrivere:

```
/usr/sbin/pppd /dev/ttyS0 nodetach
```

3.2. Connettersi a un server MS Windows

Cosa fare per effettuare una connessione ad un server Windows? Ancora una volta, è necessaria l’opzione “connect” e uno script “chat”. Aggiungere la seguenteriga al proprio file “/etc/ppp/options.ttyS0”:

```
connect chat -v -f /etc/ppp/scripts/winserver.chat
```

Creare anche lo script “/etc/ppp/script/winserver.chat”:

```
TIMEOUT 10
" CLIENT\c
```

Questo fa in modo che il client invii la stringa “CLIENT” prima di tentare di avviare la connessione PPP.

3.3. Configurare un client MS Windows (95)

Per connettersi ad un server usando Windows, bisogna usare il programma "Connessione Diretta Via Cavo" e configurarlo come “guest”. Per connettere il client Windows alla massima velocità, si deve cambiare la velocità di comunicazione della connessione diretta via cavo. Per fare questo, andare su Pannello Di Controllo->Sistema->Gestione Periferiche->Modem e selezionare cavo seriale. Procedere con l’opzione ’modem’ e selezionare la velocità massima di 115200 o qualunque altra sia impostata nel computer Linux. A questo punto è importante riavviare, o NON FUNZIONERÀ (almeno con Windows 95 versione A, non sono sicuro per le altre versioni).

3.4. Stranamente...

Stranamente ho creato un server Linux che doveva connettersi sia con computer Linux che Windows, così mi sono trovato nella strana situazione di dover impostare entrambi i sistemi Linux in modo che debbano connettersi a server e client Windows!

4. Connettersi...

Il momento della verità: connettete i computer con il cavo null-modem; prima avviate la parte server (se avete seguito l'approccio "/etc/inittab" dovrebbe essere già avviata), e poi la parte client. Dovreste vedere qualcosa simile a questo sul server:

```
[root@griis /root]# pppd /dev/ttyS1 nodetach
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS1
user zaphod logged in
Deflate (15) compression enabled
local  IP address 192.168.0.1
remote IP address 192.168.1.1
```

Notare che se avviate pppd dal file "/etc/inittab" questi messaggi appariranno nel file "/var/log/messages". Dalla parte client si dovrebbe vedere qualcosa di simile:

```
[root@wylg /root]# pppd /dev/ttyS0 nodetach
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
Remote message: Success
Deflate (15) compression enabled
local  IP address 192.168.1.1
remote IP address 192.168.0.1
```

Da entrambi le parti si può effettuare un ping verso l'altro computer per vedere se la connessione è attiva:

```
[zaphod@wylg zaphod]$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) from 192.168.1.1 : 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=0 ttl=255 time=31.7 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=20.3 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=19.2 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=20.3 ms

--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 19.2/22.8/31.7 ms
```

Adesso si può accedere al server attraverso qualunque servizio TCP/IP che potrebbe essere disponibile, come FTP, HTTP, TELNET e SAMBA.

5. Internet attraverso la porta seriale

Si suppone di avere un computer laptop e di voler accedere occasionalmente a Internet. Si suppone inoltre di avere una macchina che ha una connessione ethernet e una porta seriale. Si può impostare il laptop come client, l'altra macchina come server e usare il *masquerading* per connettersi a Internet attraverso il cavo null-modem. Consultate il IP-Masquerading-Howto per un aiuto su come farlo.

6. Altri documenti utili

- Pagine man di pppd e chat
- PPP-HOWTO
- How to hook up PPP (http://axion.physics.ubc.ca/ppp-linux.html)
- Modem-HOWTO
- Serial-HOWTO
- IP-Masquerade-HOWTO

7. Licenza

Copyright (c) 2000, 2001 by Willem J. Atsma

Questo documento è coperto dalla Gnu Free Documentation License (http://www.gnu.org/copyleft/fdl.html). Si prega di copiare e distribuire liberamente (vendendo o cedendo) questo documento in qualunque formato. È richiesto che le correzioni e/o i commenti siano inoltrati al mantenitore del documento. Si può creare un lavoro che deriva dal documento originale e distribuirlo se:

- Si invia il documento derivato (nel formato più appropriato come sgml) a LDP (Linux Documentation Project) o simile per la pubblicazione su Internet.
- Si rilascia il lavoro derivato sotto questa stessa licenza o la licenza GPL. Includere una nota di copyright e almeno un collegamento alla licenza usata.
- Si fornisce il dovuto riconoscimento agli autori precedenti e a chi ha contribuito maggiormente.

Se si sta considerando di creare un lavoro derivato diverso da una traduzione, è richiesto che si discuta il progetto con il mantenitore attuale del documento.

LICENZA ORIGINALE: This document is covered by the Gnu Free Documentation License (http://www.gnu.org/copyleft/fdl.html). Please freely copy and distribute (sell or give away) this document in any format. It's requested that corrections and/or comments be fowarded to the document maintainer. You may create a derivative work and distribute it provided that you:

- Send your derivative work (in the most suitable format such as sgml) to the LDP (Linux Documentation Project) or the like for posting on the Internet.
- License the derivative work with this same license or use GPL. Include a copyright notice and at least a pointer to the license used.
- Give due credit to previous authors and major contributors.

If you're considering making a derived work other than a translation, it's requested that you discuss your plans with the current maintainer.

8. Crediti

Ho ricevuto una gran quantità di feedback utili, specificatamente da J.M. e B.S., e una buona parte di essi è stata incorporata in questa nuova versione. Grazie!

Autore: W.J. Atsma Email: watsma@mech.ubc.ca Revisione: 29 Luglio 2001