

## Wireless LAN con Linux



Wang Xu  
<wangxu[at]linuxfocus.org>

### *L'autore:*

Wang Xu e' uno studente in PhD alla Beijing University of Posts and Telecommunications, Cina, ed e' coinvolto in comunicazioni wireless. E' diventato un amante di Linux dal 1999 quando era uno studente di college. Oltre a Linux, gli piacciono anche TeX, C/C++, Perl, ecc.



### *Premessa:*

In questo articolo, l'autore tratta dei driver per alcuni comuni adattatori Wireless LAN e di altri argomenti correlati come l'autenticazione basata su 802.1x.

---

## Introduzione

La WLAN (IEEE 802.11b/a/g) diventa vieppiu' popolare mano a mano che i dispositivi WLAN diventano piu' a buon mercato e diversi organismi consentono al loro staff od al pubblico un accesso WLAN. Quasi ogni nuovo computer laptop ha una scheda WLAN incorporata e quelli vecchi potrebbero usare una scheda PCMCIA WLAN; anche i desktop potrebbero avere delle schede USB WLAN e addirittura delle schede WLAN on-board. D'altro canto i punti di accesso WLAN (AP) si possono trovare nei campus universitari, uffici, hotel, residence e cosi' via. La WLAN facilita lo sviluppo delle reti locali e consente l'uso del computer in modo mobile, non piu' stanziale, il che rappresenta un'altra rivoluzione nel nostro modo di lavorare e nella vita di tutti i giorni.

Quindi, nel mondo Linux, e' ovviamente essenziale supportare l'accesso WLAN. Il resto di questo articolo e' organizzato come segue: prima esaminiamo come configurare le schede; poi proviamo ad accedere alle reti tramite autenticazione; quindi daremo una breve introduzione agli strumenti necessari per la configurazione dell'interfaccia WLAN; infine trarremo delle conclusioni.

# Configurare le schede WLAN

Una volta che una scheda Wireless LAN e' installata nel vostro computer, il primo passo da compiere e' quello di installare il driver e farlo funzionare. Una scheda WLAN implementa le funzioni del physical layer (PW) – lo strato fisico – ed il media access control sublayer (MAC) – il sottostrato per il controllo di accesso – come specificato in almeno una delle serie di protocolli IEEE 802.11, mentre il driver controlla la scheda, fornisce una interfaccia di rete identica a quella in una interfaccia ethernet, e provvede alla gestione di altre interfacce specifiche WLAN.

Non esiste un metodo comune per installare i driver dato lo svariato numero di fornitori e di schede. Comunque, la maggior parte di essi puo' essere installato attraverso tre metodi:

- Usare il supporto nativo del kernel di Linux® per la scheda,
- Compilare ed installare un modulo di driver per la scheda specifica,
- Usare un NDIS wrapper [1] per configurare le schede usando i driver per MS Windows®.

Nelle seguenti sezioni, l'autore illustrera' questi metodi attraverso esempi.

Nota: anche se usate uno degli ultimi due metodi, dovete assicurarvi che il supporto per wireless LAN sia stato impostato nella configurazione del kernel:

```
gnawux@APTITUDE:~$ grep CONFIG_NET_RADIO /boot/config-`uname -r`  
CONFIG_NET_RADIO=y
```

In caso negativo, dovrete riconfigurare il kernel, abilitando l'opzione "Wireless LAN (non-hamradio) Drivers and Wireless Extensions".

## Supporto del kernel per WLAN

I driver che sono sufficientemente maturi e che non hanno problemi di licenza sono stati incorporati nel kernel di Linux. Quindi l'elenco delle schede WLAN supportate dipende dalla versione del kernel, ed e' buona cosa verificare se un nuovo kernel fornisce un migliore supporto per la vostra scheda prima di installare il driver.

In questa sezione l'autore illustrera' come usare il driver per le schede WLAN basate sui chipset Intersil Prism (ISL38xx). L'elenco delle schede supportate si puo' trovare in <http://prism54.org> [2].

Per usare una scheda basata su Prism, dovete avere l'ultimo kernel 2.6, ed abilitare l'opzione "Intersil Prism GT/Duette/Indigo PCI/Cardbus" nella sezione driver Wireless LAN nella configurazione del kernel, quindi ricompilarlo.

Se leggete attentamente il testo di aiuto del modulo quando configurate il kernel, dovrete notare che avete bisogno di recuperare il firmware dal [sito web prism54.org project](http://prism54.org) [2]. Questo perche' la scheda non ha una EPROM dove memorizzare il proprio firmware. Di conseguenza, occorre scaricare il firmware quando il driver sta inizializzando la scheda. Il firmware non puo' fare parte del kernel per problemi di licenza. Dopo avere ottenuto il firmware ed averlo salvato in "/usr/lib/hotplug/firmware/", riavviate il computer, e scoprirete di avere un'interfaccia ethernet addizionale fornita dalla scheda WLAN.

## Moduli indipendenti per schede WLAN specifiche

Molte nuove schede, proprio come qualsiasi altro nuovo hardware, non hanno driver GPL-compatibili forniti dai produttori, oppure i driver sviluppati dalla comunita' open source non sono ancora abbastanza maturi da essere compresi nel kernel. Quindi questi driver sono forniti come moduli, ed alcuni di essi potranno essere aggiunti successivamente ai kernel futuri quando saranno completati.

Uno dei driver piu' conosciuti e' l' [ipw2100](#) [3] per la scheda Intel Pro/Wireless 2100, che fa parte della tecnologia Intel Centrino®, installata in molti computer laptop. In questa sezione, l'autore introdurrà l'installazione del driver ipw2100.

Per prima cosa occorre scaricare il pacchetto sorgente del driver assieme al firmware dal sito web del progetto, <http://ipw2100.sourceforge.net>. Dopo essersi assicurati che il proprio kernel sia abbastanza recente e che sia stato compilato con il supporto per i moduli, per l'hotplug del firmware e per la wireless LAN come detto piu' sopra, decomprimete il pacchetto sorgente:

```
APTITUDE:/usr/src# tar -zxvf ipw2100-1.0.1.tgz
```

Quindi portatevi nella directory del sorgente per compilare ed installare:

```
APTITUDE:/usr/src/ipw2100-1.0.1# make
APTITUDE:/usr/src/ipw2100-1.0.1# make install
```

Installando il modulo, esso vi indicherà che dovete installare il firmware:

```
Don't forget to copy firmware to /usr/lib/hotplug/firmware/ and have the
hotplug tools in place.
```

Proprio come dice il messaggio: decomprimete il firmware nella directory di hotplug, e la procedura di installazione e' completata. Ora potete attivare il modulo ipw2100 digitando:

```
APTITUDE:/usr/src/ipw2100-1.0.1# modprobe ipw2100
```

Potreste dovere aggiungere alcuni parametri qui per configurazioni diverse. Ad esempio il parametro ifname puo' specificare il nome dell'interfaccia:

```
APTITUDE:/usr/src/ipw2100-1.0.1# modprobe ipw2100 ifname=wlan0
```

Quindi l'interfaccia verra' chiamata wlan0. Per altri parametri, potete leggere la documentazione nel pacchetto sorgente del driver per ipw2100.

## Configurare altre schede

Sfortunatamente alcune schede non hanno alcun driver per Linux, oppure, se esiste, non funziona per una qualsiasi ragione. Comunque, non vuol dire che non le possiate usare sotto Linux. Infatti abbiamo [NDIS wrapper](#) [1].

La maggior parte delle schede WLAN per i desktop od i portatili supportano Windows 2000/XP, che gestiscono il supporto WLAN tramite una interfaccia standard chiamata NDIS. Quindi i driver per dette schede in genere supportano NDIS. Allora possiamo "impacchettare" quel driver per farlo lavorare sotto Linux come se si trattasse di Windows 2000/XP, e questo ci porta al progetto ndiswrapper.

In questa sezione, l'autore installera' il ndiswrapper per una scheda Netgear 121 WLAN come esempio. Per prima cosa occorre scaricare il ndiswrapper dal sito web del progetto <http://ndiswrapper.sourceforge.net> e preparare i driver NDIS per Windows. Il ndiswrapper comprende un modulo kernel ed un set di strumenti. Dovreste compilare ed installarlo:

```
APTITUDE:/usr/src/ndiswrapper-0.11# make install
```

Poi caricare il driver windows nel wrapper

```
APTITUDE:/usr/src/ndiswrapper-0.11# ndiswrapper -i ../wg121/WG121V200/ndis5/netwg121.inf
```

dove il file `.inf` e' il driver NDIS per Windows. Dopo l'installazione dovreste vedere

```
APTITUDE:/usr/src/ndiswrapper-0.11# ndiswrapper -l
Installed ndis drivers:
netwg121          driver present
```

Bene! La procedura di installazione e' completata.

## Autenticazione

Se accedete ad una WLAN da qualche ambiente pubblico, la WLAN potrebbe richiedere qualche metodo di autenticazione per ragioni di sicurezza. La maggior parte dei metodi di autenticazione disponibili per le WLAN sono basati sui metodi IEEE 802.1x (EAP) e IEEE 802.11i, i metodi basati su EAP sono attualmente i piu' diffusi.

Ci sono molti metodi di autenticazione basati su EAP, tipo EAP-MD5, EAP-TLS, EAP-TTLS, EAP-SIM, LEAP, ecc.. Gli utenti Linux possono usare *xsupplicant* fornito dal [progetto Open1x](#) [4] per accedere ad una rete che richiede una autenticazione basata su 802.1x. In questa sezione, l'autore fara' uso del protocollo LEAP, proposto da Cisco corp., per illustrare. Nota: sebbene il protocollo possa essere supportato, e' dipendente dalla scheda e dal driver, quindi anche se il vostro *xsupplicant* e' installato e configurato correttamente, potreste comunque non essere in grado di accedere alla rete perche' la scheda od il driver non lo supportano.

Dovreste scaricare *xsupplicant* dal sito del progetto, <http://open1x.sourceforge.net>, ed installarlo. Poi modificare il file di configurazione, `/etc/xsupplicant/xsupplicant.conf`. Ecco un esempio per LEAP.

```
#example of /etc/xsupplicant/xsupplicant.conf
#for LEAP protocol

network_list = all
#the list of networks to access

default_netname = default
#the default access network

first_auth_command = <BEGIN_COMMAND>dhclient %i<END_COMMAND>
#The command before authentication, which is usually used to get some info from
#the network

logfile = /var/log/xsupplicant.log
#log file

mysssid #here is your network id, may be listed in the network list
```

```

{
  type = wireless
  ssid = <BEGIN_SSID>myssid<END_SSID>
  allow_types = all
  identity = <BEGIN_ID>aptitude<END_ID>
  eap-leap {
    username = <BEGIN_UNAME>aptitude<END_UNAME>
    password = <BEGIN_PASS>passwd<END_PASS>
  }#setup for leap
}

```

LEAP e' un semplice metodo di autenticazione, ci sono molte altre impostazioni per altri metodi, quindi fate riferimento alla documentazione ed agli esempi di xsupplicant.

## Utilita' per WLAN

Come sapete, WLAN fornisce una interfaccia di rete identica ad ethernet, e la potete usare proprio come un'altra interfaccia ethernet. D'altro canto una scheda WLAN ha molte piu' possibilita' di una scheda ethernet per via della trasmissione dati senza fili. Quindi ci sono una serie di strumenti per configurare la WLAN e per ottenere informazioni circa il suo status. Potete trovare gli strumenti wireless in [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html) [5], gestiti da Jean Tourrilhes.

Lo strumento piu' utile e' *iwconfig*, che puo' essere usato in modo simile ad *ifconfig*. Il comando *iwconfig* digitato senza parametri oltre al nome dell'interfaccia stampera' lo status della scheda in funzione:

```

gnawux@APTITUDE:~$ /sbin/iwconfig wlan0
wlan0      unassociated ESSID:off/any Nickname:"ipw2100"
           Mode:Managed Channel=0 Access Point: 00:00:00:00:00:00
           Bit Rate=0 kb/s Tx-Power:off
           Retry:on RTS thr:off Fragment thr:off
           Power Management:off
           Link Quality:0 Signal level:0 Noise level:0
           Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
           Tx excessive retries:0 Invalid misc:0 Missed beacon:0

```

Con il parametro "mode", potete cambiare il modo di lavoro della scheda WLAN:

```

APTITUDE:/home/gnawux# iwconfig wlan0 mode 1
APTITUDE:/home/gnawux# iwconfig wlan0
wlan0      unassociated ESSID:off/any Nickname:"ipw2100"
           Mode:Ad-Hoc Channel=0 Cell: 00:00:00:00:00:00
           Bit Rate=0 kb/s Tx-Power:off
           Retry:on RTS thr:off Fragment thr:off
           Encryption key:off
           Power Management:off
           Link Quality:0 Signal level:0 Noise level:0
           Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
           Tx excessive retries:0 Invalid misc:0 Missed beacon:0

```

In questo caso attiviamo nella scheda WLAN la modalita' "Ad Hoc". Possiamo anche cambiare il nome di rete con il parametro "essid":

```

APTITUDE:/home/gnawux# iwconfig wlan0 essid gnawux
APTITUDE:/home/gnawux# iwconfig wlan0
wlan0      IEEE 802.11b ESSID:"gnawux" Nickname:"ipw2100"
           Mode:Ad-Hoc Frequency:2.412 GHz Cell: 02:0C:F1:0F:11:2A

```

```
Bit Rate=0 kb/s Tx-Power:off
Retry:on RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=60/100 Signal level=-83 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Forse avete notato che ora abbiamo delle informazioni sullo stato piu' esaustive di prima perche' ora abbiamo un valido ESSID. Per altri parametri di iwconfig, leggete iwconfig(8).

Un'altra potente utility e' *iwlist*, con la quale potete ottenere l'elenco delle risorse disponibili. Con il parametro "scanning", potete ottenere un'elenco dei punti di accesso disponibili:

```
gnawux@APTITUDE:~$ /sbin/iwlist wlan0 scanning
wlan0 Scan completed :
      Cell 01 - Address: 00:0D:BD:6F:B4:48
                ESSID:"<hidden>"
                Protocol:IEEE 802.11b
                Mode:Master
                Channel:6
                Encryption key:on
                Bit Rate:11 Mb/s
                Extra: Rates (Mb/s): 1 2 5.5 11
                Extra: Signal: -70 dBm
                Extra: Last beacon: 59ms ago
      Cell 02 - Address: 86:CF:C1:34:12:06
                ESSID:"gnawux"
                Protocol:IEEE 802.11b
                Mode:Ad-Hoc
                Channel:11
                Encryption key:off
                Bit Rate:11 Mb/s
                Extra: Rates (Mb/s): 1 2 5.5 11
                Extra: Signal: -37 dBm
                Extra: Last beacon: 2ms ago
```

</hidden>

E con il parametro "frequency" (*freq*), possiamo ottenere l'elenco delle frequenze:

```
gnawux@APTITUDE:~$ /sbin/iwlist wlan0 freq
wlan0 14 channels in total; available frequencies :
      Channel 01 : 2.412 GHz
      Channel 02 : 2.417 GHz
      Channel 03 : 2.422 GHz
      Channel 04 : 2.427 GHz
      Channel 05 : 2.432 GHz
      Channel 06 : 2.437 GHz
      Channel 07 : 2.442 GHz
      Channel 08 : 2.447 GHz
      Channel 09 : 2.452 GHz
      Channel 10 : 2.457 GHz
      Channel 11 : 2.462 GHz
      Channel 12 : 2.467 GHz
      Channel 13 : 2.472 GHz
      Channel 14 : 2.484 GHz
      Current Channel=1
```

Potete far riferimento a iwlist(8) per altri parametri.

A parte le due utility di cui sopra ce ne sono altre, ad esempio iwevent, iwgetid, iwpriv, iwspy, dalle quali potete ottenere e gestire lo status operativo della vostra scheda WLAN

## Conclusione

L'autore ha illustrato l'installazione dei driver per varie schede WLAN, ed ha mostrato come eseguire l'autenticazione. In questo articolo sono state anche introdotte delle potenti utilita' per la configurazione della WLAN.

Un grazie per il contributo alla comunita' open source, grazie alla quale possiamo accedere alle Wireless LAN con Linux.

## Riferimenti

1. Il progetto NDIS wrapper, <http://ndiswrapper.sourceforge.net>;
2. Il progetto Prism54, <http://prism54.org>;
3. Il progetto IPW2100, <http://ipw2100.sourceforge.net>;
4. Il progetto Open1x, <http://open1x.sourceforge.net>;
5. Jean Tourrilhes, wireless tools, [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html).

---

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Wang Xu "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Translation information: cn --&gt; -- : Wang Xu &lt;wangxu[at]linuxfocus.org&gt; cn --&gt; en: Wang Xu &lt;wangxu[at]linuxfocus.org&gt; en --&gt; ita: Roberto Pauletto &lt;neverquit/at/cwazy.co.uk&gt;</p>
---	---