How to Install Wireless 802.11a/b/g Devices on Linux When There Isn't A Driver *

Preston Lee preston.lee@typeglob.org

May 26th 2004. Last Revision May 28th, 2004.

1 Introduction

Many common wireless network cards on the market simply lack functioning Linux drivers, often times leading to a confusing and frustrating experience when they don't work out-of-the-box. While in the authors opinion this is largely the vendors fault for not releasing ample information to driver writers, the bottom line is that it's easy to get a bad taste for Linux when your brand new, common-off-the-shelf (COTS) CardBus/PCMCIA or PCI card doesn't work when it's plug in. The good news is that the ndiswrapper project provides a workaround: allowing you to run the Windows driver instead.

I'll be installing a Linksys WPC54G CardBus card on an HP Pavillion zv5000 series laptop running Fedora Core 2 and a custom 2.6.6 kernel, using the Windows driver and ndiswrapper. None of the native Linux drivers (such as prism54) seemed to work with this combination. I'll try to stay distribution neutral and will point out distro-specific tidbits as they come.

2 Identifying Your Cards Controller

While a wide variety of cards exist, many COTS cards like the WPC54G use the same controllers internally. These are purchased and relabeled by the popular consumer network hardware vendors (Linksys, D-Link, Netgear etc), and is why the driver for your card may be branded with the controller manufacturers name instead of the one printed on the card.

Use **lspci -v** to identify the controller. My Linksys uses a Broadcom BCM94306 controller.

```
[preston@laptop Desktop]$ /sbin/lspci -v

(Other devices omitted for brevity.)

03:00.0 Network controller: Broadcom Corporation BCM94306 802.11g (rev 02)
        Subsystem: Linksys WPC54G
        Flags: bus master, fast devsel, latency 64, IRQ 10
        Memory at 1c800000 (32-bit, non-prefetchable)
        Capabilities: <available only to root>

[preston@laptop Desktop]$
```

*In part extrapolated from the INSTALL file of ndiswrapper.

3 Installing The Software

3.1 The Windows Device Driver

Obtain the Windows driver for your card (preferable an up to date one from the vendors web site) and extract the two .inf and .sys files with the same name. If the only thing you have is a self-installer made to run on Windows, you may have to fire up a Windows box, install the driver, find the .inf and .sys files (in my case it's bcmwl5.inf and bcmwl5.sys—yours may be different), and copy them to your Linux box. Hopefully you can find a zipped version using Google so you don't have to deal with Windows. You may be able to use Wine for self-installers if you know how.

3.2 Installing ndiswrapper

Download the ndiswrapper source, uncompress it, change to the new directory and **make install** it. Distro-specific packages are available.

Register the Windows driver with ndiswrapper. For example,

```
ndiswrapper -i ~preston/windows_crap/bcmwl5.inf
```

Use **ndiswrapper -l** to verify the driver has been installed and to see if it has detected your card and thinks it can control it. My system reports..

```
[root@laptop ndiswrapper-0.7]# ndiswrapper -l
Installed ndis drivers:
bcmwl5  present
[root@laptop ndiswrapper-0.7]#
```

If all is well, you should now be able to load the ndiswrapper module into the kernel without warnings using **modprobe ndiswrapper** and then see your card listed by **iwconfig**. My shows up as the following. [1]

```
wlan1 IEEE 802.11g ESSID:"ap0" Nickname:"laptop.lee.home"
 Mode:Managed Frequency:2.437GHz Access Point: 00:04:5A:EC:F1:11
 Bit Rate=11Mb/s Tx-Power:8 dBm
 RTS thr=2347 B Fragment thr=2346 B
 Encryption key:off
 Power Management:off
 Link Quality:97/100 Signal level:-71 dBm Noise level:-256 dBm
 Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
 Tx excessive retries:3110 Invalid misc:6779 Missed beacon:0
```

If you've made it this far your card will probably work. All that's left is to configure it for your particular network. If you're comfortable doing this on your own, it's safe to stop reading and do your own configuration with your tool of choice. (Once everything is working you can optionally run **ndiswrapper -m** to have ndiswrapper automatically load at startup, but don't do this yet if you need help configuring.)

4 Configuring The Device

This is dependent on how your specific network is configured. Fedora Core 2 has **system-config-network**[2] which is pretty slick. Graphical network configuration tools and the config files they use vary by distribution, so I can't really help you in that department. If you'd like to stick with the command line, you'll need to use **iwconfig** and **ifconfig**. See their respective manpages for deeper examplations of the following steps.

[1] Notice it's wlan1 instead of wlan0, because this machine has an additional onboard controller already bound to wlan0. Yours should show up as wlan0 if it is the only such device in the system.

[2] This is the same as redhat-config-network in Fedora Core 1 and previous Red Hat releases

4.1 Setting your ESSID

If your ESSID—also known as an SSID or network name—is 'asu', for example, you would set it like this..

```
[root@laptop Documentation]# iwconfig wlan1 essid ap0
```

4.2 Set The Operating Mode

Unless you know otherwise, this should probably be set to 'Managed'. Do it like this..

```
[root@laptop Documentation]# iwconfig wlan1 mode Managed
```

4.3 Optionally set the WEP key

This is only neccesary if you're using encryption and is highly recommended if you are able. See **man iwconfig** for good examples and explanations of supported capabilities. Trust me, it's easy.

4.4 Bring Up The Device

The wlan0 device can now be activated like this..

```
[root@laptop Documentation]# ifconfig wlan0 up
```

Issue **ifconfig** (without any arguments) and the device should be shown!

4.5 Get An IP Address

If you're using DHCP, use a client such as dhclient or pump to get an IP address. On recent Fedora Core and Red Hat systems, like this..

```
[root@laptop Documentation]# dhclient wlan0
Internet Software Consortium DHCP Client V3.0.1rc12
Copyright 1995-2002 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP

sit0: unknown hardware address type 776
sit0: unknown hardware address type 776
Listening on LPF/wlan0/00:06:25:3d:fe:d4
Sending on   LPF/wlan0/00:06:25:3d:fe:d4
Sending on   Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 7
DHCPOFFER from 192.168.1.10
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 7
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 7
DHCPOFFER from 192.168.1.10
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.10
bound to 192.168.1.221 -- renewal in 39701 seconds.
[root@laptop Documentation]#
```

The device is now bound to address 192.168.1.221 and can be verified in the future (when you inevitably forget your assigned address) by running **ifconfig** (without arguments).

4.6 Optionally Automate Everything

4.6.1 Load ndiswrapper At Startup

Use **ndiswrapper -m** to create an entry in /etc/modules.conf and have the driver wrapper and driver loaded at startup.

4.6.2 Configure The Device At Startup

Modify your startup scripts and/or configuration files to your liking. This is /etc/sysconfig/network-scripts/ifcfg-wlan0 in Fedora Core 2 and friends.

Note: If you have a CardBus card and are using Fedora Core or Red Hat, you may want to futz with the order of the startup scripts to make sure **pcmcia** is started before **network**. Stock installs tend to have **network** execute before **pcmcia**. You can manually change the ordering by changing the symlinks listed in /etc/rc.d/rc5.d/ if you boot into a graphical login screen or /etc/rc.d/rc5.d/ if your boot into multiuser mode without X. On my system, I simply deleted the existing pcmcia symlink and created a new right before network. Now I have..

```
S09pcmcia -> ../init.d/pcmcia
S10network -> ../init.d/network
```

..which will run network immediately after pcmcia on startup.

If all has gone well, your wireless device is functioning on the network and will automatically be started the next time you reboot. Happy hacking!

Preston Lee