

Drivers for Wireless LAN Cards

Driver Safari



It would seem that the major Linux distributors have never heard of wireless networks. Nobody is currently offering configuration or setup programs, not to mention actually recognizing wireless LAN cards during the installation procedure.

Driver support is in a sorry state, too with SuSE restricted to the PCMCIA Card Services ([1]) and the PCMCIA and PC cards included in the 2.4.18 kernel. Unfortunately, the drivers do not support common chipsets such as Prism 2.5 or 3, and even 3Com cards normally require updated drivers. Enter the driver safari across your system and back, if you want to use a current card.

Drivers for the 3Com X-Jack

The Poldhu chipset, which is used for the 3Com X-Jack card (amongst others) requires a separate driver, which is available from [2], although it only runs on the new 2.4 version of the Linux kernel. As is the case for Next-

The distributors still have not discovered wireless LAN – most distribution CDs include tried and trusted PCMCIA Card Services. So that means a lot of manual configuration work to get modern wireless LAN cards up and running.

BY DANIEL COOPER

Generation modules, you will need the sources for the current kernel, although you can do without the sources for the PCMCIA Card Services.

After expanding the package, launch the `./Configure` program (with a capital C for a change). You are then prompted for the kernel and module directories, as well as the configuration directory for the PCMCIA Card Services.

You can use the defaults options in most cases. You can then type `make all` and `make install` to compile the sources. Finally, store the finished modules at the designated locations and copy the newly created configuration file `poldhu.conf` to `/etc/pcmcia`.

3Com Configuration

The `/etc/pcmcia/wireless.opts` file is used to configure the cards – as it is for PCMCIA Card Services drivers. The file assigns values to variables, as in `variable = val` or `variable = "val"` and is loaded when the card is inserted.

If you want to use multiple cards, you can restrict a configuration block by referring to the hardware address (MAC address) of a wireless LAN card or series (or part of it). The hardware address (which you can display in hex or other formats by typing `ifconfig`) is an ID with a length of 6 bytes. The first three or four bytes are usually sufficient to specify the manufacturer and series.

Let's take a look at the hardware address 00:04:DB:A5:72:E0 as an example, where the first three bytes (i.e. 00:04:DB) provide sufficient ID. The configuration block for a card in `/etc/pcmcia/wireless.opts` starts with the entry `"*,*,*,*)"`, where the end of the line is terminated by a double semicolon `;;`:

You can copy this block to a position directly below the line with the double semicolon and then edit the `upper` block: The four comma-separated asterisks match any hardware address of any card, so this block will be executed for any card – unless another block has been processed previously. So let's change the beginning of the upper block to `"*,*,*,00:04:DB*)"`, taking care to use capital letters for all of the hexadecimal characters where necessary.

The upper block is now processed for any card whose hardware address starts with 00:04:DB. As a maximum of one block can be processed, the second block will only be used for hardware addresses outside of this range, as a kind of standard configuration. We can now set variables within the blocks.

`ESSID` contains the name of your wireless LAN – put some careful thought into choosing this name as wireless devices with the same (E)SSID automatically belong to the same network. You might want to use your own phone number or part of your name.

Use `MODE` to select the operating mode: "Ad hoc" refers to a network without access points that provides a direct connection, "Managed" means that the nodes in the wireless LAN connect to a access point and use the AP as a kind of wireless hub (or switch).

You can use the `FREQ` variable to set the channel frequency, although – as this is somewhat complex – you will probably prefer to use the `CHANNEL` variable and simply choose a channel between 1 and 14.

The `KEY` variable contains the encryption key for WEP as a hexadecimal sequence, such as "0102-0304-0506-0708-090a-0b0c-0d". If you are implementing 128 bit encryption, you will need to enter 13 bytes of this type – the three "missing" bytes are used for the "Initialization Vector" (IV), which is then used for every packet transmitted,

thus providing an actual key length of only 104 bytes.

Next Generation Wireless

You will normally need the "Next Generation" driver package (`linux-wlan-ng`, [3]) from [4] to operate more recent wireless LAN cards. As most major distributions do not automatically include this package at present, you will probably need to perform a time-consuming manual installation.

We used the current stable version 0.1.14 ([5]) for our test. You will also require the sources for your kernel which will at least need to be set up using `make xconfig` or `make menuconfig`.

In SuSE's case the kernel sources are not the same as the distribution kernel, so that will mean recompiling the kernel in most cases – but don't expect that to work if you use `/boot/vmlinuz.config`, the configuration file for the default kernel.

For SuSE 8.0 you will need to enable support for WAN devices (Wide Area Network is located under "Network device support", "Wan interfaces"), and disable the emulation of other processor architectures (under "Binary emulation of other systems").

If you have the time and skill, you might like to take this opportunity to modify the kernel to suit your requirements. Type the following to start re-compiling the kernel:

```
make dep modules modules_?
install bzlilo
```

The finished kernel is then automatically installed and launches LILO.

In the case of PC Card (PCMCIA) wireless cards you will definitely require the sources for the PCMCIA Card Services from [6] or the source code package provided by your distributor. Type `make config` to configure the Card Services and preferably `make all` to recompile, followed by `make install` to perform the installation.

After completing these preliminary steps, you can setup the `linux-wlan-ng` package using `make config`. You can select drivers for the PCMCIA Card Services, PCI adapters, PCI cards, and USB adapters. The cards discussed in this article require either Card Services

drivers or USB adapter drivers. You can accept the defaults for the remaining prompts in the configuration dialog.

Now launch the compiler by typing `make all`. Unfortunately, there is no single solution if something happens to go wrong. We performed tests with the 2.2.20, 2.4.18 and 2.4.19 kernels, and version 3.1.33 of the Card Services, with PCMCIA support disabled at kernel level. Providing no errors occur on compilation, you can install the modules and configuration files by typing `make install`. You will then want to call `depmod -a`.

Card Configuration

Card Services drivers are configured via the `/etc/pcmcia/wlan-ng.opts` file, any other drivers via `/etc/wlan.conf`. The format of both files is similar, and both contain variable assignments, just like `/etc/pcmcia/wireless.opts`. To assign various blocks for different hardware addresses, follow the procedure detailed for the 3Com card. The files are adequately documented, so let's concentrate on other points.

The configuration files are loaded immediately on inserting or attaching a wireless LAN card. Ensure that the variable `WLAN_ENABLE` is set to "y" to allow the card to be set up.

The "WEP" section contains the encryption settings. To enable encryption, you will need to set `dot11Privacy? Invoked=true`, and `dot11Exclude? Unencrypted=true`, which ensures that your WLAN card will always use encryption.

To enable the 128 bit encryption, and this is strongly recommended even if just in SOHO environments, you need to set `PRIV_KEY128=true` and enter the key in hexadecimal notation for the `dot11WEP? DefaultKey0` variable. The key must comprise exactly 13 bytes in hexadecimal notation, as in 01:02:03:04:05:06:07:08:09:0a:0b:0c:0d, for example.

The last three sections allow you to choose the type of network. If you set `IS_ADHOC=n`, you will need an access point (AP) to connect the wireless computers. To use an access point you will need to enter the name of your SOHO network as the variable `DesiredSSID` in the "Infrastructure Station Start" section.

Ad hoc mode (*IS_ADHOC=y*) is normally the cheaper variant and allows you to connect up to 16 computers directly without an access point. The disadvantage is that all of these computers need to “talk” to each other, i.e. the two computers furthest apart will still need a direct wireless connection. The network name (SSID) is also used to identify the network in ad hoc mode. You can also use the *CHANNEL* variable to select one of the 14 available channels. Since the upper channels may be partially occupied by Bluetooth devices, you will want to select channel 7 or lower.

You may need to reduce the data transfer rate (from the maximum 11 mbit/s) for computers working in ad hoc mode that are some distance apart – the lower the data transfer rate, the greater the maximum transmission distance. You can use the *OPRATES* to define the various transfer rates in units of 500 kbits/s – i.e. 22 means 11 mbit/s, whereas 2 means a rate of 1 mbit/s.

Troubleshooting Driver Conflict Issues

The driver packages store a list of all supported PC Card and Wireless chipsets in a separate configuration file under */etc/pcmcia/*.conf*. Unfortunately, some contradictory entries may lead to obsolete drivers being loaded on occasions. You can type

```
grep -e manfid -e version \
/etc/ppp/config /etc/ppp/*. \
conf | sort +2 -3 -d | less
```

to sort the entries in the configuration file by manufacturer ID, displaying the filename at the start of each line. If a card does not work properly with the loaded driver, despite it having been correctly recognized and despite apparent support, you should look for doubles. Just comment out the offending line to quickly find the right entry, but note that the PCMCIA Card Services will need to be relaunched after every change by typing */etc/init.d/pcmcia restart*.

Network Configuration

So far, we have concentrated on configuring WLAN cards themselves, and not looked at the network settings, which are defined for all network PC

Cards in the centrally held */etc/ppp/network.opts* file. As we have seen in the context of */etc/pcmcia/wireless.opts*, you can also create blocks for individual cards or models in *network.opts* by assigning values to variables – follow the same scheme here.

You can leave the *INFO* variable blank, although a sensible value will not do any harm. If there is a DHCP server on your network, you can set *DHCP=y* to have the card set the required network environment automatically – depending on your server, you may need to set *DHCP_HOSTNAME* to the domain name for your network.

In case of static IPs, set *DHCP=n*, type an IP address for the card for the *IPADDR* variable (“192.168.2.2”, for example), then enter the subnet mask *NETMASK*, and the base address of the network *NETWORK* (“255.255.255.0” and “192.168.2.0” in our case), finally enter the broadcast address (“192.168.2.255”), and, if required, the IP address of your DSL router or Internet gateway as *GATEWAY*.

You may not have access to the same name server on your wireless LAN as on a wired Ethernet; in this case you can use the variables *DNS_1* through *DNS_3* to specify the addresses of the name servers responsible for your wireless network. These variables are set immediately on inserting the wireless LAN card. Any other variables are not of interest for normal network operations.

USB Adapters – a Special Case

In the case of USB adapters, such as the Actiontec Wireless 802UI3 for example, the configuration file */etc/wlan.conf* is used for basic device configuration, however, the applicable script, */etc/init.d/wlan*, is not automatically launched when you attach or enable the wireless device.

There is an easy workaround for this issue, if you change the order of the boot scripts in */etc/init.d*. For SuSE 7.3 this means renaming the symbolic links *S06hotplug* to *S05hotplug* and *K17hotplug* to *K18hotplug*, and also renaming *S05network* to *S06network* and *K18network* to *K17network* – all of which can be found in the */etc/init.d/rc3.d* and */etc/init.d/rc5.d* directories. You will also need to create two new links by typing *ln*

-s ../wlan S05wlan and *ln -s ../wlan K17wlan* in both directories. This ensures that the network configuration is launched after loading the USB Hotplug Manager and the WLAN setup – i.e. the wireless adapter is configured just like any other network device via the distribution tools, the only difference being that the wireless adapter is called “wlan0” instead of “eth0”. The order is reversed when you shutdown the system; first switch off the network, and then unload USB. You also need to add the line *alias wlan0 prism2_usb* to the file */etc/modules.conf*. We would also recommend adding the *sleep 1* command to line 2 of the */etc/init.d/wlan* file.

The USB Hotplug Manager had not completely initialized on our test system, and so this led to the driver setup for the WLAN failing. We found that this issue was successfully resolved by adding the *sleep* command.

Future

Setting up wireless LAN is a task that involves a lot of manual steps at present, as none of the major distributors provides the required modules or even configuration tools for the job in hand – and that can be a big issue for newbies. We can only hope that future distributions will be better equipped, and not simply continue to ignore wireless LANs. ■

INFO

- [1] PCMCIA Card Services Homepage: <http://pcmcia-cs.sourceforge.net>
- [2] Poldhu Driver 0.1.12 for 3Com X-Jack: <http://www.xs4all.net/~bvermeul/swallow/poldhu-0.2.12.tar.gz>
- [3] Linux WLAN Project: <http://www.linux-wlan.org>
- [4] “Next Generation” Driver Packages: <ftp://ftp.linux-wlan.org/pub/linux-wlan-ng>
- [5] linux-wlan-ng-0.1.14 Driver Package: <ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/linux-wlan-ng-0.1.14.tar.gz>
- [6] PCMCIA Card Services 3.1.33: <http://pcmcia-cs.sourceforge.net/ftp/pcmcia-cs-3.1.33.tar.gz>
- [7] Homepage for Swallow/Poldhu Drivers: <http://www.xs4all.net/~bvermeul/swallow/>
- [8] Howtos and Drivers: http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux