

## 0.1 Pilotes wifi : Linux-wlan

Les pilotes linux-wlan sont basés sur le standard IEEE 802.11 (a/b/g)) et supportent pour l'instant les périphériques à base de chipsets Intersil Prism2/2.5/3. Une liste du matériel pris en charge peut-être consulté [ICI](#)<sup>1</sup>. Dans l'exemple suivant nous allons mettre en place un réseau wifi point à point (Ad-Hoc) entre un PC/Slackware et un Portable/Slackware grâce à des dongles USB NETGEAR MA411. **NOTA BENE** Ces pilotes ne fonctionnent de manière stable qu'à partir de la version 0.2.1-pre17 avec la branche 2.6.\*.

### 0.1.1 1. Prerequis

Support USB dans le noyau 2.6 :

```
CONFIG_USB_DEVICEFS=y
CONFIG_USB=y
CONFIG_USB_EHCI_HCD=m
CONFIG_USB_UHCI_HCD=m
CONFIG_USB_OHCI_HCD=m
CONFIG_USB_HID=m
CONFIG_USB_HIDINPUT=m
```

Sinon, support USB dans le noyau 2.4 :

```
CONFIG_USB=y
CONFIG_USB_DEVICEFS=y
CONFIG_USB_EHCI_HCD=m
CONFIG_USB_UHCI=m
CONFIG_USB_UHCI_ALT=m
CONFIG_USB_OHCI=m
CONFIG_USB_HID=m
CONFIG_USB_HIDINPUT=m
```

Et bien sûr les sources de votre noyau indispensables à la compilation de nouveaux modules. Le rc.hotplug actif (ça reste un mystère mais sans lui la configuration ne se fait pas)

### 0.1.2 2. Compilation et installation

Téléchargez l'archive la plus à jour [ICI](#)<sup>2</sup>, pour moi ce sera le fichier linux-wlan-ng-0.2.1pre21.tar.bz2. Vous avez la possibilité de choisir les modules à compiler, à savoir :

- prism2\_cs Modules pour Prism2.x & Prism3 cartes PCMCIA.
- prism2\_pci Modules pour Prism2.5 Cartes PCI basé sur ISL3874.
- prism2\_plx Modules pour Prism2.x cartes PCMCIA utilisées avec un adaptateur PLX9052 PCI/PCMCIA.
- prism2\_usb Modules pour Prism2.x Connectique USB.

L'installation se déroule ainsi :

---

<sup>1</sup> [http://www.linux-wlan.org/docs/wlan\\_adapters.html.gz](http://www.linux-wlan.org/docs/wlan_adapters.html.gz)

<sup>2</sup> <ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/>

```

$ tar xjvf linux-wlan-ng-0.2.1pre21.tar.bz2
$ cd linux-wlan-ng-0.2.1pre21
$ make config
----- Linux WLAN Configuration Script -----
The default responses are correct for most users.
Build Prism2.x PCMCIA Card Services (_cs) driver ? (y/n) [y] : n
Build Prism2 PLX9052 based PCI (_plx) adapter driver ? (y/n) [n] : n
Build Prism2.5 native PCI (_pci) driver ? (y/n) [n] : n
Build Prism2.5 USB (_usb) driver ? (y/n) [y] :
Linux source directory [/usr/src/linux] :
The kernel source tree is version 2.4.27.
The current kernel build date is mar août 17 13 :18 :30 Local time zone must be
set--see zic manu.
Alternate target install root directory on host [] :
Module install directory [/lib/modules/2.4.27] :
It looks like you have a BSD-ish init file setup.
You'll need to edit /etc/rc.d/rc.S to invoke /etc/rc.d/rc.wlan (for ISA/PCMCIA cards)
so that wlan cards will be started at boot time.
Prefix for build host compiler ? (rarely needed) [] :
Build for debugging (see doc/config.debug) (y/n) [n] :
Configuration successful.

```

Comme vous l'avez deviné, j'ai activé le support pour le module "Prism2.x Connectique USB" et mon noyau est le 2.4.27. Les sources de mon noyau se trouvent dans /usr/src/linux et les modules dans /lib/modules/2.4.27. Terminez l'installation par :

```

$ make all
$ su
# make install

```

### 0.1.3 3. Configuration

Deux fichiers sont à éditer pour la configuration, notre reseau wifi se nommera pour l'exemple MyWlan. Le premier est /etc/wlan/wlan.conf, qui doit ressembler à ceci :

```

WLAN_DEVICES="wlan0"
SSID_wlan0="MyWlan"
ENABLE_wlan0=y

```

Ensuite créer un fichier /etc/wlan/wlancfg-MyWlan à partir d'un exemple par :

```
cp /etc/wlan/wlancfg-DEFAULT /etc/wlan/wlancfg-MyWlan
```

Editez-le. Dans un premier temps ne nous occupons pas du cryptage WEP. Nous allons travailler en mode point à point :

```

IS_ADHOC=y
AuthType="opensystem"
BASICRATES="22"
OPRATES="2 4 11 22"

```

La ligne **BASICRATES="22"** c'est pour avoir du 11Mbps et **OPRATES** indique les débits supportés. Pour vérifier et mettre en place les paramètres, branchez votre Dongle et tapez en root :

```
# modprobe prism2_usb prism2_doreset=1
# wlanctl-ng wlan0 lnxreq_ifstate ifstate=enable
message=lnxreq_ifstate
ifstate=enable
resultcode=success
# wlanctl-ng wlan0 lnxreq_autojoin ssid=MyWlan authtype=opensystem
message=lnxreq_autojoin
ssid='MyWlan'
authtype=opensystem
resultcode=success
```

Vous devez obtenir un success a chacune de ces commandes comme dans l'exemple ci-dessus. A partir de là, un iwconfig devrait vous faire apparaître votre wlan0, à noter que les commandes du package wireless-tools sont inutilisables (à part le iwconfig sans options). Les programmeurs ont développé leur propre jeu d'instructions listables par la commande wlanctl-ng commands. Concrètement, on attribue au PC fixe une adresse IP :

```
ifconfig wlan0 192.168.10.1
```

et au PC portable également :

```
ifconfig wlan0 192.168.10.2
```

Les LEDs de la carte ne clignote plus le reseau est donc actif, je peux me pinguer sans fil ... Le cryptage WEP a pour but de chiffrer les données transmises à l'aide de reseaux sans fils, bien que ce ne soit pas un mode de sécurité inflexible, il serait encore plus dangereux de ne pas l'activer. Chacques personnes voulant se connecter au reseau doivent avoir la clé WEP. Sur chaque PC, éditez le fichier /etc/wlan/wlancfg-MyWlan et ajoutez ce qui suit en fin de fichier :

```
lnxreq_hostWEPDecrypt=true
lnxreq_hostWEPDecrypt=true
dot11PrivacyInvoked=true
dot11WEPDefaultKey0="01 :23 :45 :67 :89"
dot11WEPDefaultKey1="01 :23 :45 :67 :89"
dot11WEPDefaultKey2="01 :23 :45 :67 :89"
dot11WEPDefaultKey3="01 :23 :45 :67 :89"
```

Pour les dot11WEPDefaultKey\* n'hésitez pas à rajouter des caractères non imprimables. Voilà vous pouvez surfer sur la terrasse, sous la couette ;-), pour le partage de connection ca se passe par ICI<sup>3</sup>

---

<sup>3</sup> [http://www.trustonme.net/didactels/112.html#firewall\\_nat](http://www.trustonme.net/didactels/112.html#firewall_nat)