

Linux Wireless LAN Howto

Jean Tourrilhes

8 September 99

*Linux & Wireless LANs : Un*x, with no string attached...*

1 Introduction

This document will explore the magical world of **Wireless LANs** and **Linux**. Wireless LAN is not a very widespread and well known technology, even in the Linux world, so we will try to gather here most of the available information. Despite the fact that it is very similar to common networking technologies, it is significantly different to justify this specific document covering the subject.

1.1 What is a Wireless LAN ?

It's a **networking** technology allowing the connection of computers without any wires and cables (apart from the mains), mostly using **radio** technology (and sometime **infrared**). It's called LAN (Local Area Network) because the range targeted is small (within an office, a building, a store, a small campus, a house...). This technology is slowly growing (I should say maturing), and despite a general lack of interest, Linux is able to take advantage of some of the wireless networks available.

1.2 Content of this document

My first task is to talk a bit about the different Wireless LANs options under Linux. What the **products** on the market are, their compatibility with Linux and where to find the necessary bits and pieces to make them work. This should help you to make your mind on the product of your dreams.

Once you've picked a Wireless LAN, you will have to live with it. The next chapter go through the main **differences** of Wireless LAN compared to other networking technologies. This includes the main steps of the installation and usage considerations.

Then, we will have a nice overview of the **Wireless Extensions**. The Wireless Extensions is a new standard interface to configure Wireless LAN devices and get wireless specific statistics from them. Of course, this is a Linux exclusivity !

At this point, you will find a long and dense section, talking mostly of the different **technologies** used in Wireless LANs and other boring related stuff. It is quite safe to skip that one.

1.3 Target and Assumptions

The main goal of this document is to reduce the traffic of unanswered questions related to wireless in the Linux newsgroups and mailing lists (and in my

mailbox). After that, you should have no more arguments for asking foolish questions around.

I hope that this document will help people to make the most of their Wireless LAN under a competent operating system and understand what is in the box. If I could convince people to give it a try, it would make me happy.

This document act mostly as a complement to the exhaustive documentation existing for Linux. Because of that, I might not explain every details of everything and target already quite knowledgeable people. Don't worry, there is a section on how to improve your culture at the beginning of the *section 3*.

1.4 Legal stuff

Strange world where everybody has to protect himself from sharks, lawyers and crazy people :

Any information in this document is purely fictitious and any resemblance to real hardware, software or driver is purely coincidental..

I mean, if because you read this document your hardware burn, you get fired from your job or anything else bad happen, I'm not responsible, it can't be my fault, so please use your own brain. Writing this kind of documents is not part of my job at HP, so I don't expect them to claim any responsibility for its content.

Any brand mentioned in this document is trademark of its respective owner. For example Linux is a trademark of Linus Torvalds.

Then, this is my document, written by me (Jean Tourrilhes), therefore I own its copyright. So don't remove my name (and copyright notice) and pretend that you wrote it yourself. In matter of copy, distribution and modification, you should ask me politely and use common sense.

Having said that, this document is also licensed under the terms of the **Linux Documentation Project Copying License**.

1.5 This document

This document is only available in the format that are convenient to me (acrobat/pdf, html). It might be updated in the future (if I feel like it and if I have some time). I guess that it is pretty safe to assume that it will still be available for the time to come at these **web** addresses :

http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html

http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wavelan.html

I may be reached at the following **e-mail** address :

jt@hpl.hp.com

Constructive comments and interesting information are welcomed. I hope that you will help me to keep this document up to date and improve its content.

Comments about my english and my style will be answered in french. Flames and spam will be processed through a Rayleig Fading channel with a -120 dB attenuation in order to reduce the noise :-)

2 The devices, the drivers

This section describes the most common Wireless LAN products available on the market and their **compatibility** with Linux. I will make a short description of each product and will mainly focus on the drivers.

Except in a few case, you need a **driver** to interface you wireless network device to the Linux kernel. The availability of a driver is as usual your main concern, especially with wireless devices because few people are using such hardware, so few of them are willing to develop, debug and maintain such a piece of code.

For each driver, I will list its *status* (stable, buggy...), the *maintainer*, the *version*, how to *get it* and the main *features*. If you hear about something new or if you have developed yourself a driver, please notify me.

2.1 Lucent Wavelan & DEC RoamAbout DS

Driver status : stable
Version : v19 (20/4/99) or v20 (29/7/99)
Where : isa : Linux kernel (2.0.37, 2.2.11 & 2.3.15)
 pcmcia : Pcmcia package (3.0.11)
Maintainers : Jean Tourrilhes <jt@hpl.hp.com>
Web page : http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wavelan.html
Documentation : man pages, headers
Configuration : Wireless Extensions
Statistics : Wireless Extensions
Multi-devices : isa : up to 4
 pcmcia : yes
Other features : module, hardware multicast, Wireless Extensions
Non implemented : roaming
Bugs : see release notes on web page :-(
License : GPL & OpenSource
Vendor web pages : <http://www.wavelan.com/>
 <http://www.networks.digital.com/dr/wireless/>
 <http://www.cabletron.com/dnpg/dr/npg/lanfm-mn.html>

2.1.1 The device

The Wavelan has been around for quite a while now. The Wavelan is a radio LAN, using the 900 MHz or 2.4 GHz ISM band (Direct Sequence). It is built by Lucent, formerly AT&T, formerly NCR, and there is a few EOM version (for example the DEC RoamAbout DS). The Wavelan comes in two flavours, an ISA card and a PCMCIA card (plus the access point).

The Wavelan appears to the PC as a standard network card and interfaces naturally with the networking stack. The configuration includes setting the

frequency (10 different channels), Network ID (16 bits). Hardware encryption is optional (DES or AES - 64 bits key).

This product is built around a standard Ethernet controller (that may be found in some 3Com and Intel Ethernet cards), and the Ethernet physical layer is replaced by a radio modem. The ISA and Pcmcia cards share the same basic architecture, have the same modem, but have different Ethernet Controllers and bus interfaces (the pcmcia has only one transmit buffer). Because the Wavelan doesn't use a specific radio MAC (no MAC level retransmissions for example), it uses very efficiently the bandwidth, but is more sensitive to packet loss and collisions.

There is two versions of the modem, a 900 MHz and a 2.4 GHz version. Revision 2 of the 2.4 GHz modem allows the user to set the frequency (from a set of predefined channels - the availability of each channel depend on the regulation). The Wavelan is Direct Sequence Spread Spectrum (11 chips encoding), using a 2 Mb/s signalling rate (using effectively 22 MHz of bandwidth) and diversity antennas.

2.1.2 The driver

The ISA driver has also been around for quite a while now in the kernel and is pretty stable. The last set of modifications were to solve a few remaining small problems and add Wireless Extensions and some other features, so the driver is fairly complete now. The only things remaining to do is the implementation of the roaming protocol (but it might come, if I'm not too lazy...).

The Pcmcia driver has caught up with the ISA one to offer the same level of functionality and reliability. The only difference are the pcmcia specific functions (auto loading, auto unloading, power saving).

The drivers use the card EEprom to save the configuration changes for subsequent reboots. Wireless Extensions let you configure the NWID, the frequency, the sensitivity and the encryption key (optional). Statistics include the signal quality, signal level, noise level and the count of packet received with an invalid NWID (see Wavelan documentation). Private Wireless Extensions include the setting of the quality threshold.

2.2 Lucent Wavelan IEEE, Apple Airport & Cabletron RoamAbout 802

Driver status : fairly stable
Version : v1.0.1
Where : <http://www.fasta.fh-dortmund.de/users/andy/wvlan/>
Maintainer : Andreas Neuhaus <andy@fasta.fh-dortmund.de>
Documentation : readme, headers
Configuration : Wireless Extensions & module parameters
Statistics : Wireless Extensions
Multi-devices : yes, but the ISA to Pcmcia bridge must be reconfigured
Other features : MTU selection, Wireless Extensions, private extensions

Non implemented : some IEEE 802.11 support (power saving, authentication).

Bugs : none reported...

License : GPL

Vendor web page : <http://www.wavelan.com/>

<http://www.apple.com/airport/>

<http://www.cabletron.com/wireless/>

2.2.1 The device

Even if it uses the same name, the Wavelan IEEE product is completely different from the old Wavelan, and totally incompatible in term of protocol and hardware interface. It is still built by Lucent, and it still operate in the 2.4 GHz ISM band (Direct Sequence), but the new hardware fully support the IEEE 802.11 protocol and is no longer based on a Ethernet chip. There is only a Pcmcia version (the ISA version uses a ISA to Pcmcia bridge) and the different access points.

The recently announced Apple AirPort is in fact the Lucent Wavelan IEEE repackaged (the second generation, 802.11 HR compliant). The AirPort card for the iBook is the EOM version of the Wavelan, with the antennas being pre-integrated in the laptop. Note that the interface doesn't seem to be Pcmcia compatible even is the connector is the same, so this card can't be used in the normal PC-Card slot of other laptops. Cabletron is also selling the Wavelan IEEE as RoamAbout 802.11.

The Wavelan appears to the PC as a standard network card and interfaces naturally with the networking stack. The configuration includes only setting the network name (ESSID), the rest is automatic (finding the equivalent BSSID and channel). As usual for Lucent, the documentation and website are rich.

As with all IEEE 802.11 products, the Wavelan offer a fully featured MAC protocol, including MAC level acknowledgement (good news for all of us having dealt with the old Wavelan card), optional RTS/CTS, fragmentation, automatic rate selection, roaming. This seems exhaustive, but is mandatory for IEEE 802.11 compliance. Different version of the card include different level of security (bronze is basic, silver is with WEP (RC4-40 bits) and gold is with proprietary 128 bit encryption.

The 2.4 GHz modem is an enhanced version of the previous generation, Direct Sequence Spread Spectrum (11 chips encoding), using both 1 and 2 Mb/s signalling rate (using effectively 22 MHz of bandwidth), diversity antennas and with 13 different frequencies (depending on the regulations). The modem sensitivity seems especially good (so, I predict a good range).

Lucent also sells a "turbo" version of the card, which in addition to the standard IEEE 802.11 bit rate (1 and 2 Mb/s) adds 5 and 10 Mb/s bit-rates for shorter range using Lucent proprietary modulations (so, not compatible with 802.11 HR and other 11 Mb/s products).

Finally, Lucent offers as well the second generation of the Wavelan IEEE, which is much cheaper and fully compliant with the new 802.11 HR standard, supporting 5.5 and 11 Mb/s bit-rate (compatible with other 11 Mb/s products). This is the version used in the Apple Airport.

2.2.2 The driver

Andreas Neuhaus is busy working on the driver to get it out of beta stage. The driver is based on Lucent source code, but doesn't contain any of the code necessary for the higher level 802.11 functionality (authentication, encryption and power saving), however the driver fully interoperable with other Wavelan nodes and APs.

Andreas has done a very good job into providing features like Wireless Extensions (I did help him a bit ;-)) and many configuration parameters (station name, channel, mtu size) and plan to add more and consolidate the source of the driver. Power management and encryption are not yet implemented.

Note that Lucent has also released a binary library driver (see below) which is more complete than the driver of *Andreas* (except for Wireless Extensions).

2.3 Lucent Wavelan IEEE (binary library driver)

Driver status : stable
Version : v4.00
Where : ftp://ftp.wavelan.com/pub/SOFTWARE/IEEE/PC_CARD/LINUX/
ftp://sourceforge.org/pcmcia/contrib/
Contact : Lucent support <usasupport@wavelan.com>
Maintainers : Richard van Leeuwen <rleeuwen@lucent.com>
Dean W. Gehmert <deang@tpi.com>
Documentation : Extensive readme
Configuration : Module parameters, Wireless Extensions
Statistics : Wireless Extensions
Multi-devices : yes, but the ISA to Pcmcia bridge must be reconfigured
Other features : WEP encryption, power management and turbo mode
Non implemented : ?
Bugs : ?
License : Binary only for the core from Lucent + OpenSource Linux wrapper
Vendor web page : <http://www.wavelan.com/>

2.3.1 The device

This is the same device as the previous entry (*section 2.2*).

2.3.2 The driver

Lucent has decided to not put all its eggs in the same basket and developed a bold strategy for the support of the Wavelan IEEE under Linux. Not only they have released some source code to allow the source driver mentioned above, but they have as well contracted *Dean* to release a driver based on a binary library. This gives Linux users the choice, a GPL full source driver to hack with and a stable full featured binary driver (the official term from Lucent is "Linux Driver Source/Library").

Dean has written the code interfacing between Linux and the library, and had put together a nice package easy to install and with documentation. As expected, the binary driver is more stable and complete than the full source driver mentioned above (for example, encryption and power management are not available in the full source driver), and offers all the features of Lucent Window drivers, plus a nice integration with Linux and the Wireless Extensions. This driver supports both the basic version of the card and the “turbo”.

But, as with any binary driver, you should check if your architecture and your version of the kernel and Pcmcia package are supported.

2.4 Netwave AirSurfer & Xircom Netwave

Driver status : fairly stable
Version : v 0.4.1
Where : Pcmcia package (2.9.12)
Maintainers : John Markus Bjørndalen <johnm@cs.uit.no>
Dag Brattli <dagb@cs.uit.no>
Web pages : <http://www.cs.uit.no/~johnm/>
<http://www.cs.uit.no/~dagb/>
Documentation : man page
Configuration : Module parameters & Wireless Extensions
Statistics : Wireless Extensions
Multi-devices : yes (except for module parameters setting)
Other features : -
Non implemented : hardware multicast, multiple transmit buffers
Bugs : -
License : OpenSource
Vendor web page : <http://www.netwave-wireless.com/>

2.4.1 The device

The Netwave is also a quite common product. This is a radio LAN operating in the 2.4 GHz ISM band. It is now build by Netwave Technologies, formerly part of Xircom. The Netwave is Pcmcia only, and comes in a small form factor (everything is included on the Pcmcia card !).

The Netwave use a specific MAC protocol designed for radio (a pre 802.11 protocol, with fancy stuff such as RTS/CTS, virtual carrier sense and fragmentation). It uses a 9 bits domain (Network ID), the highest bit of it used for the type of network (set for access point operation and unset for ad-hoc operation). The Netwave uses also a 16 bits scrambling key (encryption). The Modem offers a 1 Mb/s signalling rate and frequency hopping (100 ms hop period). On the bad side, the Netwave has no antenna diversity and a high overhead.

Note that the Netwave AirSurfer plus is a very different beast (see below).

2.4.2 The driver

The original author of the driver (*John*) has made a very good job for debugging it, and his good friend (*Dag*) has joined the project, and is fixing the remaining bugs and adding new features. The driver is quite simple and don't implement yet the full Wireless Extensions. The driver uses only one transmit buffer, which lower slightly the performance. The device configuration includes the domain and the scrambling key which can be set through Wireless Extensions or as module parameters (need to be set in `/etc/pcmcia/config.opts` - don't forget to restart `cardmgr` after a change).

It seems that the Netwave is quite picky with some pcmcia sockets and you might need to choose carefully the interrupt (try different ones) and set the memory speed correctly. In some cases, under high load (big ftp), the transmission sometime get stuck (I guess that some interrupt are lost) and the driver has to reset the card (you won't notice it, it just decreases the performance).

2.5 Netwave AirSurfer plus

Driver status : fairly stable
Version : 1.0.2
Where : <http://ipoint.vlsi.uiuc.edu/wireless/asplus.html>
<ftp://sourceforge.org/pcmcia/contrib/>
Maintainer : Jay Moorman <jrmoorma@uiuc.edu>
Documentation : Readme, man page
Configuration : Module parameters, Wireless Extensions
Statistics : Wireless Extensions
Multi-devices : yes (except for module parameters setting)
Other features : -
Non implemented : 802.11 mode, hardware multicast, multiple transmit buffers
Bugs : -
License : OpenSource
Vendor web page : <http://www.netwave-wireless.com/>

2.5.1 The device

The Netwave AirSurfer plus is the second generation of Netwave card, and still operate in the 2.4 GHz ISM band and is as well a small Pcmcia card. Netwave Technologies has now been acquired by BayNetwork, now a part of Nortel. In theory, the BayStack 650 is the same beast.

The AirSurfer plus has two modes of operation, compatible with the old generation of Netwave, or 802.11 compliant. The hardware is based on an AMD core, and a 1 Mb/s frequency hopping modem.

2.5.2 The driver

Jay took the code of the original Netwave driver and modified it to support the new AirSurfer plus, keeping most of the features with it. So, you still have the Wireless Extensions, and modules parameters (in `/etc/pcmcia/config.opts`).

The current driver support the AirSurfer plus only in Netwave compatible mode, and doesn't support the AirSurfer plus with the 802.11 firmware.

2.6 BayStack 660, ZoomAir, YDI and other Harris Prism based cards...

Driver status : stable
Version : 0.2.7, 0.2.7a, 0.3.1.1 (beta version) and 0.3.4 (beta version)
Where : <http://www.absoval.com/linux-wlan/>
<http://www.astro.umd.edu/~teuben/linux/wireless.html>
<http://www.cs.berkeley.edu/~jhill/linuxwlan/>
Maintainer : Mark S. Mathews <mark@absoval.com>
Peter Teuben <teuben@astro.umd.edu>
Jason Hill <jhill@cs.berkeley.edu>
Documentation : Readme, man page
FAQ : <http://linux.grmbl.be/wlan/>
Configuration : Module parameters & configuration tool
Statistics : Statistic tool
Multi-devices : yes
Other features : Quite exhaustive 802.11 support
Non implemented : WEP
Bugs : -
License : MPL
Vendor web pages : <http://www.netwave-wireless.com/>
<http://www.zoomtel.com/zoomair/>
<http://www.ydi.com>
<http://www.intersil.com/prism/>
<http://www.amd.com/products/npd/npd.html>
<http://www.intalk.com/>
<http://www.dbtel.com.tw/>
<http://www.gemtek.com.tw/>
<http://www.sem.samsung.co.kr/>

2.6.1 The device

The Harris Prism chipset and the AMD AM930 controller are some highly integrated parts designed to ease the process of building 802.11 products. Harris has done a Pcmcia reference design based on their chipset and the AMD core,

which explain the high number of vendors building variants of this card (the Harris website has a longer list than mine ;-). A special mention for YDI (Young Design Inc) which openly support Linux (see below).

The AMD core integrates a generic microcontroller and the hardware baseband (ASIC) to do the time critical functions of 802.11. AMD has developed the 802.11 firmware with all the usual basic 802.11 features (MAC level ACK, RTS/CTS, Fragmentation...). The Prism chipset is a 2.4 GHz Direct Sequence modem offering 1 and 2 Mb/s. The Prism chipset can also be extended to supports the new 802.11 HR standard, with 5.5 and 11 Mb/s bit-rate.

The Pcmcia cards are mostly similar from vendor to vendor. Some vendors offer ISA cards, and the Access Points are where vendors are making their difference (ZoomTelephonics uses a software AP on a PC, others have hardware AP). Each vendor also has to provide the high level 802.11 in their drivers (authentication, WEP, Roaming...), so those might be different (not that it does matter much under Linux).

The BayStack 650 and Netwave AirSurfer plus use the same AMD controller, but a different physical layer (Frequency Hopping), so are not compatible with this driver.

Harris has just become Intersil and released the Prism II chipset, this time including the MAC controller, so they won't use anymore the AMD part in their reference design. The Prism II chipset offers lower cost, better performance and full compatibility to the 802.11 HR standard (5.5 and 11 Mb/s). I expect that current product using the Prism chipset will migrate to the Prism II chipset sooner or later, the bad news is that the hardware interface is likely to be not totally compatible because of the new MAC controller.

2.6.2 The driver

Mark and the people at AVS have developed a full 802.11 driver for the Prism reference design card (AMD controller + Prism chipset), and this driver work for the many other implementations as well (see web page).

The driver is well written and very complete : it's currently the only driver where most of the higher layer 802.11 functionality is implemented. There is also many initialisation parameters and a tool to configure the card. Because the 802.11 standard is very complex, not everything is totally finished and a few features like WEP (RC4 40 bits encryption) are missing.

There is currently two branches maintained by *Mark*, 0.2.X which is stable and 0.3.X which is experimental.

Peter (with help from YDI) has created a alternative version of *Mark's* package to add ISA support, fix a few bugs and with explicit support of cards from YDI. In the long run, those changes should find their way in *Mark's* package...

Jason has created a version of the 0.3.1 beta driver with support for the BayStack 660, by porting bits from 0.2.6 (this allow support for both the BayStack 660 and infrastructure).

I believe that this driver doesn't support the BayStack 650 and Netwave AirSurfer plus cards (which don't use the Prism chipset but Frequency Hopping), but the changes for that might not be that hard to implement.

2.7 Proxim RangeLan2, Proxim Symphony, DEC RoamAbout FH and AMP Wireless

Driver status : stable
Version : 1.5.1
Where : <http://www.komacke.com/distribution.html>
Creator : Paul Chinn <loomer@1000klub.com>
Maintainer : Dave Koberstein <davek@komacke.com>
Documentation : Readme file
Configuration : specific tool, partial implementation of Wireless Extensions
Statistics : none
Multi-devices : no ("insmod -o" multiple modules)
Other features : Uses Proxim source code
Non implemented : -
Bugs : -
License : Binary only for the core from Proxim + OpenSource Linux wrapper
Vendor web pages : <http://www.proxim.com>
<http://www.wlif.com>
<http://www.networks.digital.com/dr/wireless/>

2.7.1 The device

The RangeLan2 is a classical product using the 2.4 GHz band, made by Proxim, a small californian company. The products are certified and sold in approximately 50 countries. The RangeLan2 is based on Proxim proprietary protocol, OpenAir, that Proxim is trying to push as an alternative to 802.11. Of course, you will find many EOM version (like the DEC and AMP versions). It comes as ISA cards, Pcmcia cards, design-in modules, and access point.

The RangeLan2 implements a specific MAC protocol designed for radio (OpenAir, another pre 802.11) implemented on a generic microcontroller. It uses a 4 bits domain, 4 bits channel and 4 bits subchannel, and also a station type (primary master, secondary, slave - this is used for network synchronisation). There is no encryption, instead it uses a technique called Security ID (which is a simple password used to derive the network ID). The OpenAir protocol is heavily based on RTS/CTS, offer a good robustness but some overhead. It offers as well a modulable contention window size, contention free access for the master, packet fragmentation and power saving.

The Modem uses frequency hopping, and 2 levels of modulations (2FSK/4FSK) : it runs a 1.6 Mb/s signalling rate for good channel condition (short to medium distances) and falls back to 0.8 Mb/s otherwise.

Note that the Symphony line of product (home networking) offered by Proxim use the MAC protocol of the RangeLan2 (OpenAir) with a lower cost radio, and the main difference is the software bundle and the price. On the other hand, the Proxim RangeLan802 line is very different from OpenAir products, using the 802.11 protocol and a different interface, so the Linux driver won't work.

2.7.2 The driver

Dave uses the Proxim driver source code to build a library (distributed as object only), so we should expect a good quality code. *Paul* wrote the part to interface with the Linux kernel and *Dave* maintains it. He has written as well a small utility to set the configuration in the driver (through `ioctl`). The driver supports the Proxim Rangelan2, the Proxim Symphony, the DEC RoamAbout FH and the AMP Wireless products. The driver support both ISA PnP and Pcmcia cards.

The current driver doesn't support the RangeLan802 line, but you may contact *Dave* if you would like to see a driver for RangeLan802.

2.8 Symbol Spectrum24

Driver status : experimental (Pcmcia only)
Version : alpha 2
Where : <ftp://sourceforge.org/pcmcia/contrib/>
Maintainer : Lee John Keyser-Allen <lkeyser@wpi.edu>
Documentation : Readme file
Configuration : module parameters, partial support of Wireless Extensions
Statistics : none
Multi-devices : -
Other features : Support of micro-AP
Non implemented : Multicast, statistics...
Bugs : -
License : GPL
Vendor web page : <http://www.symbol.com/products/wireless/wireless.html>

2.8.1 The device

Symbol is one of the other major player for Frequency Hopping devices in the 2.4 GHz band and has been selling its Spectrum24 line of products for ages. Symbol sells mostly to vertical market (in their bar-code readers, in warehouses, in supermarket), so their products are not usually found in retailers. The Spectrum24 family include an Access Point, a ISA card, a Pcmcia card and a Pcmcia card with micro-AP functionality.

The Spectrum24 products were designed from the start to be compliant with the 802.11 standard, way before the standard was eventually adopted. Their products are compatible and interoperable with other 802.11 products, but not officially compliant (almost compliant).

Symbol is also very active in developing Voice over IP solutions for their wireless LANs, and that's why they are also selling some Spectrum24 phones. They are using the H.323 codec, compression and call setup (raw 64 kb/s, compressed 10 times) and a 30 ms packet rate (but I fail to see what they have done to overcome overhead and latency issues at the MAC level).

The MAC has all the usual features of the 802.11 standard, like MAC level retransmission, RTS/CTS, fragmentation, auto bit-rate selection, power saving and roaming. A nice feature of the MAC is the support of the micro-AP functionality, which allows to turn a PC into an Access Point (I would like more vendors to start doing that). However, their products don't seem to support ad-hoc mode (maybe the reason for non-compliance).

The physical layer is Frequency Hopping supporting 1 and 2 Mb/s, with 100 mW output power and 100 ms dwell size.

2.8.2 The driver

Lee has written the driver as a student project for Symbol, so with active help from Symbol. He plans to continue supporting it, and Symbol may get more active in distributing the driver.

The driver supports only the Pcmcia card (LA2400 and micro-AP version), and despite being alpha, is stable and supports most features of the card (like micro-AP, shared memory access...).

2.9 Aironet ARLAN

Driver status : stable (ISA only)
Version : 2.0 & 2.1b
Where : Linux kernel (2.3.10 & 2.2.7-acX), web-page for 2.0.X version
Maintainers : Elmer Joandi <Elmer.Joandi@ut.ee>
Cullen Jennings <c.jennings@ieee.org>
Web pages : <http://www.ylenurme.ee/~elmer/655/>
<http://www.cs.ubc.ca/spider/jennings/>
Documentation : README file + web page
Configuration : /proc interface (2.1.X kernels and up only)
Statistics : ?
Multi-devices : ?
Other features : -
Non implemented : Multicast (driver is point to point ?)
Bugs : -
License : GPL

Vendor web page : <http://www.aironet.com/products/2200fam/2200fams.html>

2.9.1 The device

The Arlan is a radio LAN, built by Aironet, using the 900MHz or 2.4GHz ISM band (Direct Sequence). The Arlan comes in 3 flavour, an ISA (655), an MCA (670) and a pcmcia (690) card (plus the access point). They have now renamed the ISA card IC2200 and the Pcmcia card PC2200.

The configuration include setting the frequency and Network ID (24 bits ?). The MAC protocol is implemented on a generic microcontroler. There is two versions of the modem, a 900 MHz and a 2.4 GHz version. Both use Direct Sequence Spread Spectrum. The 900 MHz modem allow signalling rate up to 860 kb/s (fall back to 215 kb/s) and 12 channels. The 2.4 GHz version allow signalling rate up to 2 Mb/s (fall back to 1 Mb/s) and 5 channels.

2.9.2 The driver

Russell Nelson told me a while ago that he was trying to convince Aironet to release the specifications of the Arlan to develop a Linux driver. *Cullen Jennings* started the development of a point to point driver, *Elmer Joandi* rewrote some parts and added a lot of features to be compatible with the Access Point, released the whole under GPL, and here is the result.

The driver support only the ISA version of the card (655 or IC 2200). The driver have been fully tested and optimised by *Elmer Joandi*, includes a complete /proc interface and should be soon included in the kernel.

2.10 Aironet ARLAN 802.11

Driver status : fairly stable
Version : 16/6/99 ?
Where : <http://www.cse.ucsc.edu/~breed/airo.html>
<ftp://sourceforge.org/pcmcia/contrib/>
Maintainers : Benjamin Reed <breed@almaden.ibm.com>
Documentation : README file
Configuration : /proc interface
Statistics : none
Multi-devices : N/A
Other features : Support ad-hoc and managed mode
Non implemented : WEP (encryption)
Bugs : Doesn't work with SMP
License : MPL
Vendor web page : <http://www.aironet.com/products/>

2.10.1 The device

I admit that I'm a bit lost with all the different Aironet Arlan product names. The 4500 and 4800 series are their latest offering, based on the 802.11 DS

standard, whereas their 2200 series is their previous generation (see above). They sell also the 3500 series, which is 802.11 Frequency Hopping.

The Arlan 4500 and 4800 series are 802.11 compliant wireless LANs in the 2.4GHz ISM band (Direct Sequence). Both series are offered in ISA, PCI, Pcmcia, serial, Ethernet and multi-Ethernet version, plus the Access Point.

These cards are based on the Harris prism chipset, like many other cards (see *section 2.6*). The 4500 offer standard 1 and 2 Mb/s bit rate, whereas the 4800 series adds 5.5 and 11 Mb/s for shorter range, compatible with the 802.11 HR standard. The MAC includes all the standard 802.11 features, with Power Saving, Ad-Hoc mode and roaming.

2.10.2 The driver

This is a totally new driver and I'm trying to get information about it. It supports the ISA, PCI and Pcmcia cards (both 4500 & 4800 versions), it looks fairly complete and debugged, with a nice /proc interface. *Ben* is now working on implementing WEP (encryption in the driver), the experimental driver can be grabbed on his web page.

Ben also told me that the driver was able to recognise the PC3500 cards, but more work would be needed there to get it fully working.

Elmer Joandi, who maintains the other ARLAN driver, told me that he was working as well on a ARLAN 802.11 driver (4500/4800, isa/pci/pcmcia), probably to be released in September.

2.11 Raytheon Raylink and WebGear Aviator2.4

Driver status : stable
Version : 1.58 (stable) and 1.65 (experimental)
Where : <http://world.std.com/~corey/raylink.html>
Maintainer : Corey Thomas <corey@world.std.com>
Documentation : README file + headers
Configuration : modules parameters
Statistics : /proc interface
Multi-devices : yes (except for module parameters setting)
Other features : hardware multicast, MTU selection
Non implemented : Some high level 802.11 functionalities.
Bugs : -
License : GPL
Vendor web page : <http://www.raylink.com/micro/raylink/>
<http://www.webgear.com/>

2.11.1 The device

The Raylink is a IEEE 802.11 FH device build by Raytheon for the 2.4 GHz ISM band. Raytheon build only a Pcmcia card and an Access Point. I've been told

that some version of the BreezeCom BreezeNet Pro Pcmcia card was an EOM version of the Raylink.

You are more likely to buy the Raylink as a WebGear product, either as Aviator2.4 or Aviator2.4 pro (which have nothing in common with their old Aviator 900 MHz line). The Aviator2.4 and Aviator2.4 pro are in fact the same product as the Raylink, the Aviator2.4 line comes pre-configured in ad-hoc mode and offer only the Pcmcia card, whereas the Aviator2.4 pro comes preconfigured in managed mode and offer both the Pcmcia card and the Access Point. Of course, it is possible to change the mode in the driver and all these products are fully interoperable. WebGear also offers a ISA to Pcmcia bridge to install the Pcmcia card in desktops.

The Raylink delivers all the features expected from a 802.11 compliant device, with ad-hoc networking, access point operation, authentication and roaming. The MAC protocol is as defined in 802.11 : CSMA/CA with MAC level retransmissions. Configuration includes mostly the ESSID (network name).

The modem is 2.4 GHz Frequency Hopping, with 1 Mb/s and 2 Mb/s bit rate, and includes antenna diversity.

2.11.2 The driver

Corey has implemented a very complete driver supporting most of the feature of the hardware and some 802.11 functionality (it should be able to talk to some 802.11 nodes). There is an exhaustive list of configuration parameters, a /proc interface for more parameters, and a tool to dump 802.11 frames. Good work !

The driver has been developed for the Raytheon Raylink and has also been successfully tested with the WebGear Aviator2.4.

The experimental driver adds authentication, which is needed to communicate with the Windows driver. The Raylink driver has also been added to the 2.3 kernel series, but this is quite experimental...

2.12 No Wires Needed

Driver status : stable
Version : 0.2.0
Where : <http://www.xs4all.nl/~bvermeul/sw550/>
Maintainer : Bas Vermeulen <bvermeul@blackstar.xs4all.nl>
Documentation : README file
Configuration : Module parameters
Statistics : no
Multi-devices : unknown
Other features : -
Non implemented : Multicast, security features, statistics
Bugs : -
License : GPL
Vendor web page : <http://www.nwn.com/>

2.12.1 The device

No Wires Needed is a small company in the Netherlands building a range of 802.11 DS devices, including a Pcmcia card (Swallow), an Access Point and a Hub. They also offer a ISA version using a ISA to Pcmcia bridge.

The Swallow delivers all the features expected from a 802.11 compliant device, with ad-hoc networking, authentication and roaming. The main difference with other 802.11 devices is that NWN offers some strong link layer encryption and a key management and distribution system.

The modem is the famous Prism chipset used in many other cards (see *section 2.6*), which is 2.4 GHz Direct Sequence, with 1 Mb/s, 2 Mb/s, 5.5 Mb/s and 11 Mb/s bit rate. No Wires Needed use their own MAC design on an embedded RISC processor.

2.12.2 The driver

Bas has implemented a simple driver for the Swallow 550 card (Pcmcia) and working hard to extend it. Support for the Swallow 550's security features and the Wireless extensions are in the works.

2.13 Z-Com LANEscape

Driver status : ?
Version : ?
Where : http://www.boerde.de/~matthias/airnet/zcom_v12/
<http://www.boerde.de/~matthias/airnet/zcom/>
Maintainer : ?
Documentation : README file
Configuration : no
Statistics : no
Multi-devices : unknown
Other features : -
Non implemented : -
Bugs : Doesn't work with the new firmware revision.
License : Binary only, no license info
Vendor web page : <http://www.zcom.com.tw/>

2.13.1 The device

Z-Com is based in Taiwan, and the WL2400 family is based on the classic AMP+Prism design. The family includes the usual ISA and Pcmcia cards, the Access Point, and also a PC104 version (that's interesting)...

Z-Com claims that the WL2400 is firmware upgradable to 802.11, but I've been told that some old cards have an hardware bug preventing it. Anyway, the card has all the usual 802.11 features, and the modem is classical Direct Sequence at 2.4 GHz, supporting 1 and 2 Mb/s.

Z-Com also announce on the web site the WL2300, which is Frequency Hopping, and the WL2500 family which support 5.5. and 11 Mb/s (probably using a Prism II chipset).

2.13.2 The driver

The driver has been written by the manufacturer, and *Matthias* put it on its web site. The driver only contains the object files (no source) and seem to have been designed for kernel 1.3.X and working in 2.0.X kernels (but, as the driver interfaces in the kernel have changed since, this driver might not work in 2.2.X). The driver only work with old firmware revisions, and doesn't work with the 802.11 compliant firmware.

Matthias seems to now have access to the driver source code and is investigating compatibility with 2.2.X and new firmware revisions.

2.14 BreezeCom BreezeNet

Driver status : not needed (for Pcmcia, see below)

Version : -

Where : -

Maintainer : none

Documentation : none

Configuration : none

Statistics : none

Multi-devices : yes

Other features : -

Non implemented : configuration & statistics

Vendor web page : <http://www.breezecom.com/>

2.14.1 The device

The BreezeNet is a Radio LAN using the 2.4 GHz ISM band (Frequency Hopping). It is built by BreezeCom, a small company from Israel, and some EOM version might be available. The BreezeNet doesn't connect to any of the usual PC bus but instead uses an Ethernet network card to interface to the host computer, and so require no driver to work (they have also some real access points). A Pcmcia card is available as well, the old one was probably an EOM version of the **Netwave**, I've been told that some version may be an EOM version of the **Raylink**, but their latest pro.11 Pcmcia card is 100 % BreezeCom.

There is two versions of the BreezeNet, the old one, Netwave compatible, and the new pro.11 version, which is 802.11 compliant. In both case the modem includes Frequency Hopping Spread Spectrum (20 ms hop period), a 3 Mb/s signalling rate (fall back to 2 and 1 Mb/s) and antenna diversity.

2.14.2 The driver

No driver is needed, this product use an Ethernet connection. You need to have an dedicated Ethernet 10baseT card configured under Linux to plug it into.

For the device configuration and statistics, unless someone write the necessary tools for Linux, I guess that you must return to DOS/Windows.

For the old Pcmcia card (not pro), you may use the `newave_cs` driver (in all good Pcmcia packages). BreezeCom told me that a driver for the new Pcmcia card is coming soon (watch this space...).

2.15 Not supported

Netwave AirSurfer plus (in 802.11 mode), BayStack 650 : Now that a driver for the BayStack 660 is available, it should be quite easy to make a driver for those cards, by reusing the physical layer parts in the AirSurfer plus driver. FreeBSD seems to have a driver for this device...

RadioLan has a 10 Mb/s at 5 GHz product, rather very short range and no Linux drivers.

Diamond Multimedia HomeFree Wireless : this is a cheap home networking product at 2.4 GHz. So cheap that there is no MAC controller and the full MAC protocol is implemented in the driver, like for IrDA, WinModem and WinPrinters. A driver may come for it soon (stay tuned)...

WebGear Aviator 900 MHz : connect to the parallel port and offer cable replacement solution. No functional Linux driver yet.

The IBM Wireless LAN Entry is a discontinued product that may be sometime found for a very very low price. Unfortunately, there is no working driver for those and information on the device is impossible to find.

2.16 A note on driver licenses

Donald Becker's web page alerted me on the **license and copyright** issues for networking drivers (see <http://cesdis.gsfc.nasa.gov/linux/misc/modules.html#legal> for details). If you just plan to use the driver in your Linux PC, there should be no problem, but if you plan other use of the drivers you should pay attention to the exact license the driver come in.

Most drivers are **GPL**, which prevent their use with non-GPL kernels (so commercial operating systems can't reuse the code) and prevent to use portions of the source in non-GPL drivers, except with the explicit authorisation from the author.

Some other drivers come with a **binary library**, which restrict its potential use (the driver can't be ported to other architectures).

This may be tough, but those people have spend long nights and week ends convincing the hardware manufacturer to release information, writing and debugging the code, so please respect their copyrights and decisions.

2.17 More information on the devices, other Wireless LANs

You will notice that I don't give too much information on the different devices. The web page of each **vendors** usually contain the full specification of the products they sell.

They are many more products available than the ones that I've listed (which are the most common). If your favourite wireless LAN is not listed above, either there is no driver under Linux that I know of, or it is an **EOM** version of one of these (same hardware under a new brand).

To have a good picture of all the devices available and their characteristics, you should redirect your favourite browser to :

<http://hydra.carleton.ca/info/wlan.html>

2.18 Other Wireless technologies

2.18.1 Wireless bridges

Wireless bridges allow to connect different networks via radio, their goal is to replace a dedicated leased line (T1, for example). They usually offer longer distance through **directional antennas**, and are peer to peer.

These devices are a totally independent box (like other bridges, routers or gateways) and not a card to plug in your PC, so have no interactions with Linux.

2.18.2 Radio Amateur and AX25

These devices are quite specific and are described in their own *howto*.

2.18.3 Infrared

Standard **Infrared** (IrDa) usually appears to the system as a serial port, so you might be able to use that (with *SLIP* or *PPP*). There is some development effort going on under Linux to build an **IrDa stack** (this is quite a work...) which would allow to use products such as NetBeamer and printers :

<http://www.cs.uit.no/linux-irda/>

There is also some real Wireless LANs using **diffuse infrared** (no more peer to peer), but I don't have any information on these.

2.18.4 Digital mobile phones and other radio WAN

Again, this is quite different from Wireless LANs. I don't know anything about those devices.

Digital mobile phones (*GSM*, *TDMA*, *CDMA*) often allow data connections, but I guess that they use a *modem* (serial) interface. **Wireless WANs** such as the *Metricom* and *ARDIS* should use modem interface as well.

3 Wireless LANs in use

Installing and using a Wireless LAN is not such a big deal, and is not much different from other kind of networks. In this chapter, I will give you a few tricks on how to install those beast and will mostly redirect you to a lot of literature explaining the things much better than I would do.

Then, I will explain some of the difference of Wireless LANs compared to wired technology from the user point of view and why it reacts sometime differently. For more curious people, see the *section 5*.

3.1 Choice and selection of a Wireless LAN

There is far too many people buying a Wireless LAN and discovering only after that it is not supported under Linux. So, please, check that a driver is **available** for the hardware you plan to use.

Most Wireless LANs are designed to work well in most configurations, but my experience tells that some Wireless LANs or some environment may be capricious. Of course, the vendor won't advertise this, so it's your responsibility to check that the Wireless LAN is working with your particular setup. If you intend to cover a large range, **test** as many physical locations and combinations as possible to avoid surprises. Know the limits of your hardware.

The performance of different Wireless LANs may vary widely, depending on many factors. The throughput of two Wireless LANs advertising the same bit rate may vary by a factor 5 (I won't give the names). So, be warned and **benchmark** your Wireless LAN...

If you are not happy with your choice of Wireless LAN, don't hesitate and **return** it to where you bought it for a refund.

3.2 Installation

The reader should be familiar with some of the documents listed in the *Useful readings chapter* below, because the information here mainly acts as a complement to them. A good knowledge of your Wireless LAN is also a prerequisite before switching to Linux.

Most Wireless LAN vendors have tried to make things easy and offer product with an interface as similar as possible as **Ethernet**, and which work mostly the same way. So, a bit of background on Ethernet and the general Linux networking is welcomed (see below).

The operating system need a piece of software to interface to the hardware. That is the role of the **driver**. Basically, when Linux gives to the driver a packet to send, the driver have to copy the packet to the hardware and toggle the correct bits in the correct register on the card to send it. It is the same when the card generates an interrupt, the driver go and read the packet and give it to Linux. Of course, the driver needs to know about the specific hardware details and the specific operating system ways.

In conclusion, you must check first if the driver for your Wireless LAN exists (see *previous section*), because in many case it proves to be quite useful..

With Linux, you have to **compile** the driver source code (some Linux distribution may offer precompiled modules). There is usually two compilation options : drivers compiled statically in the **kernel** and as a **module**. If the driver is already in the kernel sources, the compilation is quite simple (you have to enable it in the kernel configuration, static or module). If it is in the **Pcmcia package**, you just need to install the package. Otherwise, see the installation instructions coming with the driver.

Once you've got the driver compiled, you must tell your system about it. For pcmcia drivers, the package has its own **configuration scripts** (see pcmcia

documentation). For other drivers, you will have to edit the system configuration scripts. You will likely to have to add a *ifconfig* line in some script (*/etc/init.d/network* for the *Debian*). There is many other networks configuration files in */etc*. For modules, you need as well to change some stuff in the module configuration files. See the list of readings below for more information.

3.3 Useful Linux related readings

- [Ethernet HowTo](#) - How to install and configure most of the network drivers
- [Net2 HowTo](#) - The network stack story
- [Module HowTo](#) - To compile you driver as a module
- [Pcmcia HowTo](#) - An excellent medicine for pcmcia drivers
- [AX25 HowTo](#) - AX25 and Radio Amateur users should enjoy this one
- [The Linux Network Administration guide](#) - A lot of background and tips about networking and Linux
- Your distribution documentation or FAQ (if any)

3.4 Driver parameters

A lot of users are confused when it comes to set **driver parameters**. Those parameters usually allow to specify the *base address* and *interrupt* of the hardware to avoid scanning, but also might be used for multi-device configuration or wireless specific parameters (see below).

As it is explained nowhere correctly, I disgress a bit and give you a few hints...

For driver compiled **statically in the kernel**, the parameters are passed on the kernel command line. The syntax is "ether=*irq,base,name*" where *base* is the base address, *irq* the interrupt and *name* the device logical name (ex : *eth0*). The kernel command line is passed by *lilo* (or *loadlin*) itself, so in fact it means that you add in */etc/lilo.conf* a line which look like this :

```
append="reboot=warm ether=0,0,eth1 ether=10,0x3E0,eth2 ether=11,0x390,eth3"
```

For drivers compiled as **modules**, the parameter interface is much more flexible and each driver may be different, so you must look in the documentation. Basically, the driver define a set of parameters by their name and you may set for each keyword an array (one value for each instance of the hardware). The module configuration is usually done in */etc/conf.modules* like this :

```
alias eth1 hp100
alias eth2 wavelan
options wavelan io=0x3E0,0x390 name=eth2,eth3 irq=10,11
```

For **pcmcia modules**, the configuration is done in the pcmcia scripts.

3.5 More parameters to configure

The most obvious difference with Ethernet is that there is **more parameters** to configure. In order to communicate, all nodes of the network must have those

parameters configured the same. Some examples are : *frequency* or *hopping pattern*, *network id* or *domain*, *encryption key* (for security)...

Under *Windows*, the installation program usually opens a nice window and asks the user to enter these parameters, or sets them to a default value. Some drivers set those parameters in a permanent storage in the device (*EEProm*), so the Linux driver is able to reuse them. But, the current tendency is to scrap the *EEProm* and to use the *Windows 95 registry* to save those parameters instead. Of course, the Linux driver can't retrieve the parameters in those conditions.

The **Wireless Extensions** (see *next section*) has been designed to simplify the process of setting those parameters under Linux by providing an unified interface across drivers, but not all drivers support (yet) the **Wireless Extensions**...

In conclusion, you must read your documentation to know what parameters need to be set, what they are used for, and look the Linux driver documentation to know how to set them under Linux. See below for a suggested list of information sources.

It is usually quite a good idea to install the Wireless Lan first under some mainstream operating system with the official vendors driver and tools, to have a feeling of how the beast does work. You might also compare the performance before and after :-)

Once you've got all those new parameters set, your Wireless LAN should be up and running.

3.6 Where to get information about your Wireless LAN

- The official documentation that come with your product.
- Manufacturer web page and support.
- Linux driver source code, documentation (headers, man pages), maintainer.

3.7 Wireless LAN deployment

From the network administrator point of view, the main problem with Wireless LANs is that the **medium is shared**. If on a cable you know who is there, anybody and anything can use the radio band.

To try to separate everyone out there, most products define some *network identifier* (Network ID, Network Name or Domain, in 802.11 it is also called *ESSID*). This is a number or character string which is used to identify all the people wanting to be one the same **logical network**. Networks using different *network identifiers* still share the bandwidth, but are logically separate and don't interfere with each other.

This situation is not totally ideal, so that's why usually you have some **distinct channels** (or *frequencies*, or *hopping patterns*). People on distinct channels use different part of the bandwidth, so don't interfere at all. If you want to install multiple independent networks in the same area, this is the way to go.

The Wireless LAN has only a limited range, so you may reach only device within that range. This is usually why you should define some cells where everybody is in range. If you want those cells to communicate or a node to move

across cells, you should install an **access point** in each of those and configure those with the same *network identifier* (and add an Ethernet segment between the access points).

On the other hand, some time you just want to quickly set up a network between a group of nodes and don't want to build an infrastructure. Most Wireless LANs offer **ad-hoc networking**, allowing you to just do that (apart from TCP configuration).

Some network administrators are also a bit scared by security problem over the medium. The only solution is to use **encryption**.

3.8 Point to point links (connecting different LANs by wireless)

Most Wireless LANs are designed to be used as a local area network, where all the nodes can see each other or can see the access point, and they are attached to other networks through a single access point (or not at all in ad-hoc mode).

Some people have asked me question on how to use Wireless LANs to connect different LANs together using wireless technology, usually those LANs are in **distant places** (across the street). Most of the time, you can't use a Wireless LAN because you don't have a fully connected topology (some node can't see each other, it's more a set of point to point links) and you may need to use directional antennas to overcome the distance.

I've never personally tried this, but I see 2 ways to achieve this.

The first solution is to use **Wireless Bridges**. Each Wireless Bridge is connected to one of the LAN section and redirect the traffic over the air to the correct destination. There is many products on the market, they are a bit expensive but very flexible, transparent and optimised for the task.

The second is to use normal Wireless LAN cards, and to plug them in a *router* (for example a Linux PC). I recommend to use a Wireless LAN supporting *RTS/CTS* if you have more than one link, and to set them in ad-hoc mode (no access point). Each LAN segment must have a different *IP subnet*, and the wireless link must have it's own subnet (it can be a private subnet). After much configuration of the routing tables of your network, you should be able to get it working.

Some people using the Aironet Arlan cards for this kind of application have made a very nice **Arlan Wireless Routing Howto**, and I believe it can apply to most other Wireless LANs as well :

http://www.rage.net/wireless/wireless_howto.html

3.9 Performance

If the Wireless LANs give enough performance for most users and applications, usually the experienced user is a bit disappointed when doing some real benchmarks. This is because the Wireless LANs are slower to start with and on top of that uses less efficiently the available bandwidth.

Most Wireless LANs have a **signalling rate** around 1 or 2 Mb/s. The signalling rate is the speed the bits are send over the air (Ethernet is 10 Mb/s), but doesn't account of all the overhead of the protocols.

The Wireless LAN protocols have usually a higher overhead than their wired counterpart (like Ethernet) because of some technological limitations and to improve the reliability and the coverage of the Wireless LAN (optimisation trade-offs). On the other hand, Wireless LANs protocols are also usually less sensitive to high load (the throughput doesn't drop when you overload the network - which could happen more often).

Some protocols also adapt the signalling rate depending on the quality of the link (for example a 2 level modulation 2FSK/4FSK). When the link is clear and reception is strong, it will use the fastest rate, but when there start to be noise or the device is further away, it goes down to the more robust rate. The throughput that you will get will depend on that as well (for example the high speed might be only usable in line of sight).

3.10 Reliability

Most Wireless LANs protocols include mechanisms to improve the **reliability** of the packet transmissions to be at the same level or even better than Ethernet (*MAC level retransmissions* for example). Anyway, if you are using a protocol such as TCP (the default under Linux), you will be fully protected again any loss or corruption of data over the air. In other word, when you copy a file across the radio, it can't be corrupted (but it might fail).

3.11 Coverage

The **propagation** of radio transmissions is influenced by many factors. Walls and floors tend to decrease and reflect the signal, and background noise make it more difficult to extract. The channel quality vary quite a lot over the time (*fading*).

Depending on the quality of reception, the error rate will change (forcing packet retransmissions), or the system may switch to a more robust (and slower) mode (fragmentation or modulation), so the actual throughput will vary from good to nothing.

Because of the way radio transmission are affected by the environment, it is quite difficult to predict the comportment of the system and to define a range. You will have some good, fair and bad area/period, the closer you are the more likely you are to be in a good one.

3.12 Mobility

One of the main advantage of Wireless LANs is that they offer **mobility**. It mean that even when moving around, you retain your connection to the network.

Of course, this mobility is limited by the range of the Wireless LAN. To extend the range, you must cover the area with **access points**, which very often include **roaming** : you switch transparently to the closer access point which provide you a connection to the rest of the world and nodes out of range.

If you want to move across IP subnets, this is time to try **Mobile IP** :-)

3.13 Security and Privacy

Because they use radio waves, wireless LANs are usually perceived as a security problem. For most users, the *network identifier* will be enough protection

against casual users (other people can't accidentally join your network unless they guess the correct network identifier).

Some people are more concerned about those issues or may want to increase the security of their system. Some Wireless LANs offer **MAC level encryption**, sometimes as an option, which is designed for those concerned users and target security equivalent to a having an Ethernet cable, and a few systems offer stronger encryption. This encryption is totally transparent to the higher layer and the user just need to set the same encryption key in the access point and all nodes of the network.

3.14 Benchmarks

For all the reasons described above, I think it is quite tricky to **benchmark** Wireless LANs, and measuring coverage or throughput in isolation is not fair. This is why I don't give any numbers. Some computer magazine publish from time to time some extensive review of all those products.

If you want to test the throughput of your device, you should use a tool called *Netperf*. You might want to submit your results in the database...

<http://www.netperf.org/netperf/NetperfPage.html>

4 Wireless Extensions for Linux

See the document about Wireless Extensions.

5 Wireless LAN technology overview

See the document about the technology overview.