HowTo setup a Lucent WLAN

Olivier DOLE

07/17/02

Contents

1 iPAQ configuration **2**
- 1.1 Required drivers . . . 2
- 1.2 Drivers configuration . . . 2
- 1.3 WL110 parameters . . . 3
- 1.4 Network parameters . . . 3
- 1.5 Useful files and tools . . . 4
  - 1.5.1 Useful files . . . 4
  - 1.5.2 Useful tools . . . 5

2 Access Point **6**
- 2.1 How to reach the access point. . . . 6
  - 2.1.1 Wired LAN administrator station . . . 6
  - 2.1.2 Wireless LAN administrator station . . . 7
- 2.2 General settings . . . 8
  - 2.2.1 About IP addresses and subnet . . . 8
  - 2.2.2 Configuration with Linux . . . 9

3 Applications **10**
- 3.1 How to connect iPAQ to the internet . . . 10
  - 3.1.1 Using WLAN . . . 10
  - 3.1.2 Using the workstation . . . 10
  - 3.1.3 Using PPP link . . . 11
- 3.2 Use your host as a http server for iPAQ . . . 12
- 3.3 Using dillo, wget... . . . 12
  - 3.3.1 Using dillo . . . 12
  - 3.3.2 Using wget . . . 13

1 iPAQ configuration

1.1 Required drivers

As hardware, an iPAQ H3600 with a PCMCIA interface and a Compaq WL110 card are used.
To work correctly, PCMCIA interface requires drivers :

- pcmcia-modules which contains misceallaneous PCMCIA drivers like the pcmcia_core driver or the ds driver.
- pcmcia-cs which is the PCMCIA Card Services for Linux, includes a set of client drivers for specific cards and the card manager daemon.

Compaq wireless card requires a driver too :

- wvlan-modules which is Lucent 802.11b drivers

Another useful package can be downloaded :

- wireless-tools which contains the wireless tools to manipulate the Linux Wireless Extensions. The Wireless Extension is an interface allowing to set WLAN specific parameters and get the specific stats.

All these packages can be downloaded at `http://ipkgfind.handhelds.org/`

1.2 Drivers configuration

At the boot time, PCMCIA drivers loading is automatically done. So, only drivers configuration has to be achieved. But before, it could be clever to check if the drivers are really loaded thanks to ***lsmod***:

```
# lsmod
Module                  Size   Used by
wvlan_cs                33024  1
ds                      7920   2 [wvlan_cs]
h3600_generic_sleeve    4528   0 (unused)
sa1100_cs               9680   2 [h3600_generic_sleeve]
pcmcia_core             39328  0 [wvlan_cs ds sa1100_cs]
```

Nonetheless, a change in /etc/pcmcia/rc.pcmcia is necessary: PCIC variable which is initially set to ***i82365*** will be set to ***sa1100_cs***. This modification will take effect as soon as the PCMCIA interface will be restarted: ***rc.pcmcia restart***.
For futher informations about PCMCIA interface configuration, see the PCMCIA HowTo: `http://pcmcia-cs.sourceforge.net/`.

1.3 WL110 parameters

The inserted card will be identified thanks to /etc/pcmcia/config file which find (the kernel) out about the type of device (network, memory, scsi...) and the module to use (wvlan_cs, wavelan_cs, memory_cs...).
The parameters to configure WL110 card and WLAN essid are located in /etc/pcmcia/config.opts file. For example:

```
# Options for WL110 card
module "wvlan_cs" opts "port_type=1 station_name=ipaq network_name=ipaq_WLAN eth=0"
```

useful options:

port_type=[1-3]: set port_type to 1 for a Basic Service Set, 2 for a Wireless Distribution System and 3 for Pseudo-IBSS (peer-to-peer).

channel=[0-14]: the channel (frequency) option is useless if using an access point. The valid range depends on local restriction as an example France is limited to channel 11 to 14.

ap_density=[1-3]: the access point density can also be set: 1 means a low density of access points and 3 a high one. This setting will affect roaming thresholds.

eth=[0-1]: this option is used to change the network device naming. For example set this to 0 to have devices named wvlan# and to 1 to have devices named eth#.

mtu=[256-2296]: Maximum transfert unit value is 1500 by default.

Nonetheless you can modify most of these options or simply look how the card is configured using ***iwconfig***. To understand how iwconfig and his options run, see man pages.
Another file to configure wireless network card is /etc/pcmcia/wireless.opts. It allows the user to configure the card with few knowledges using iwconfig in a transparent way. Thus, the user will be able to define encryption keys, network ESSID, frequency, rate...

1.4 Network parameters

Network parameters are edited in /etc/pcmcia/network.opts. This file is really interesting because of his flexibility. Indeed, the same iPAQ can be used in different places where the network configuration is not the same. For example, at home it is up to your ISP to allocate you an IP address, what is done via a DHCP server or something like that. Whereas at office you may have a static IP address. Such situations are easily solved thanks to the scheme which allow you to have differents network configurations according where the iPAQ is used. In the same way, different configurations can be used according to the MAC

address or the interface.
The /etc/pcmcia/network.opts file allows you to configure network parameters:

IP address: If it is not a static one, you can use either DHCP or BOOTP.

Netmask address and network one: The network address is not the same as the IP address, it is the IP address masked by the netmask.

Gateway address: This address is used for static routing.

DNS: The configuration of DNS parameters for the wireless interface is also possible. In a such case, the edited informations will be added to /etc/resolv.conf.

1.5 Useful files and tools

1.5.1 Useful files

The /proc directory is very interesting because it contains a lot of different informations about the wireless card, its interrupts or its packet state.

/proc/interrupts: This file contains a first column with the IRQ, a second one with the number o interrupts for each IRQ since last reboot and a last one with the associated module.

/proc/kmsg: It displays kernel messages.

/proc/ksyms: This file is identical to the command ***ksyms -a***.

/proc/net/dev: This file contains statistics about packets received and transmited for each network interface.

/proc/net/wireless: This file contains statistics about 802.11 packets and 802.11 link quality.

Another helpful directory is /var which contains log files.

/var/run/stab: It contains the socket table.

/var/log/ksymoops/*: This directory contains 3 types of files.
yyyymmdd.log which is a classical log file.
yyyymmddhhmmss.ksyms which is equivalent to: at hhmmss do ksyms > yyyymmddhhmmss.ksyms.
yyyymmddhhmmss.modules which is equivalent to: at hhmmss do lsmod > yyyymmddhhmmss.modules.

1.5.2 Useful tools

ifconfig: In order to see the network interface configuration or to modify it, the use of ***ifconfig*** and ***iwconfig*** is necessary. The iwconfig command is close to ifconfig one except that it is specific to wireless, i.e. iwconfig options are much more specific to wireless field.

cardctl: This command is generally used to get some informations about the wireless card status. It also allows the "root" user to restart, suspend or eject the card if necessary. In fact, ***cardctl*** monitors, controls the state of PCMCIA sockets and selects between different PCMCIA configuration schemes.

iwpriv: This tool is used to configure optionnals private parameters of a wireless network interface. ***Iwpriv*** is the companion tool to ***iwconfig***. It deals with parameters and settings specific to each driver, as opposed to ***iwconfig*** which deals with generic ones, and a few commands that doesn't fit well in iwconfig like listing the available frequencies.

iwspy: Get wireless statistics from specific nodes. These statistics are the same as the one available in /proc/net/wireless.

tcpdump: this tool prints out the headers of packets on a network interface that match an expression. That is very useful to analyze a network traffic and filter it if necessary.

2 Access Point

First of all, read the whole documentation concerning the access point. Personnally, I have used an Orinoco AP1000.

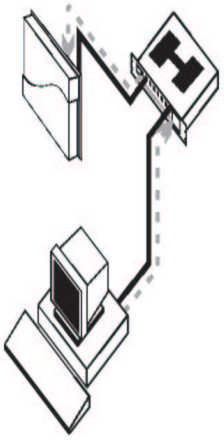
2.1 How to reach the access point.

First, it is important to notice that for an Orinoco AP1000, the default IP address is 153.69.254.254 with a class B network i.e. a subnet mask equals to 255.255.0.0. This means that to reach the access point, provided that the access points are on the same subnet as the administrator station, either the station has a similar network address (153.69.254.xxx) or the access point is identified by means of his ethernet MAC address. Of course, the second solution is much more convenient because it allows the administrator to set up IP address at a correct value for his own network.
Then, the configuration software has to be installed on a computer which can reach the access point. there are four differents manners to reach the access point which depends on the administrator station. Indeed, this one can access the access point via wired LAN or wireless LAN:

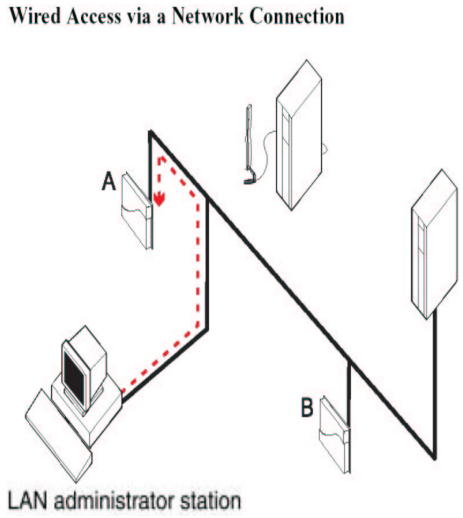
2.1.1 Wired LAN administrator station

Wired access via a direct cable connection: This way to proceed is quite easy to understand. A hub needs to be placed between the administrator station and the access point and that's all. It is also important to notice that the access point network address has to be the same as the administrator one.

Wired Access via a Direct Cable Connection



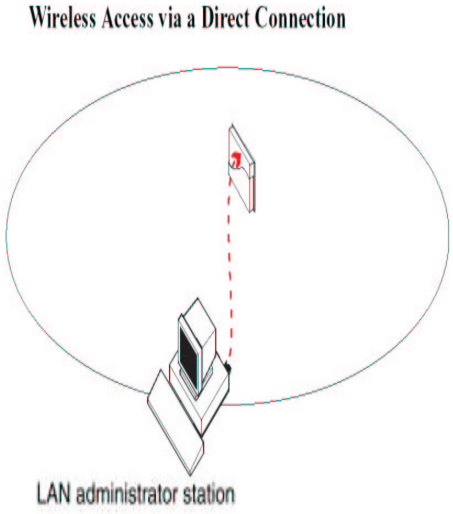
Wired access via a network connection: To proceed as follow, the wireless LAN administrator station has to be on the same network as the access point. The main advantage is that the administrator can configure access points A and B without changing anything in his network topology.



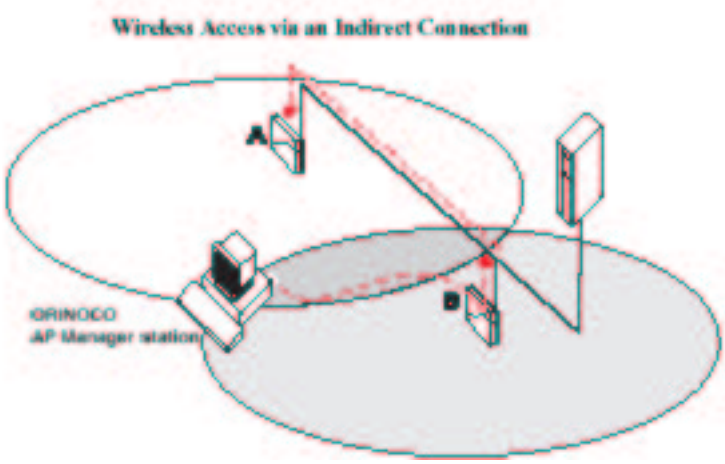
2.1.2 Wireless LAN administrator station

To configure an access point thank to a wireless administrator station, it is obvious that the network interface of the LAN administrator is already configured to match the settings of the access point.

Wireless access via a direct connection: The idea is to establish a wireless point-to-point connection as shown below. This method is interesting when configuring access points sequentially.



Wireless access via an indirect connection: The idea is to establish a wireless point-to-point connection with another access point that provides access to the "target" access point via a network backbone as pictured below.



2.2 General settings

2.2.1 About IP addresses and subnet

In large networks, the architecture may include differents subnets typically separated by routers or gateways. It is important to remind that in a such network architecture all access points and wireless stations must be installed on the same

subnet, i.e. on the same side of the router or gateway. Indeed some functionnalities like roaming do not work over routers, or when access points are connected to differents subnets, a mobile station may loose its network connection when it physically enters an area where the access points are connected to a different subnet.
To ensure that the access points at specific locations keep the same IP address, it is advised to assign them "static" IP address.

2.2.2 Configuration with Linux

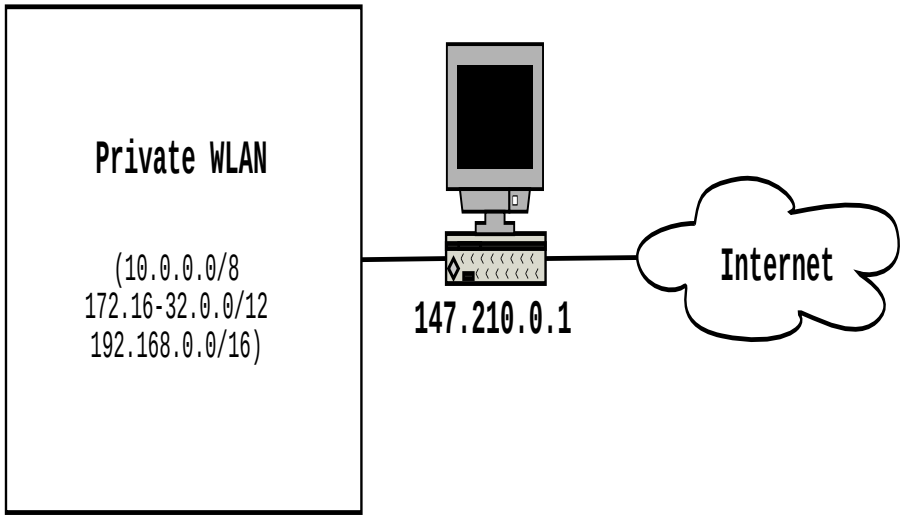
The software to configure Orinoco AP1000 via Linux is available at: `ftp://ftp.orinocowireless.com/pub/software/ORiNOCO/CLI_Lin/R7.5spring2002/clili117.tgz`.
This software is a command line one and has not yet graphical mode. That's why, I would advice you to configure as fully as possible your access point via Windows and use Linux to make minor changes.

3 Applications

3.1 How to connect iPAQ to the internet

3.1.1 Using WLAN

Connecting WLAN to the internet is quite easy if your access point has a "static" IP address and is already linked to a network connected to the internet. As a matter of fact, the only thing to do is to edit the /etc/pcmcia/network.opts file and correctly set up the gateway IP address.
Nonetheless, your WLAN can have a specific architecture and thus makes this setting harder. As an example, what to do if the network topology looks like that:



In a such case, the workstation has to be configured to forward the packets from the WLAN to the internet.

3.1.2 Using the workstation

The preceding situation sets two problems:

- how to forward packets between the 2 networks via the workstation.
- how to reach an ipaq with a private address.

One very useful tool allow us to solve this 2 problems: iptables. The following explains how to use iptables for a specific case, for further informations, see the excellent Howto: `http://iptables-tutorial.haringstad.com/`.
To enable iptables configuration for the workstation, only 2 commands lines are sufficient:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -s ipaq -o eth1 -j SNAT --to-source 147.210.0.1
```

The first line enables packets forwarding. The second one tells the workstation to forward packets from ipaq on port output_interface. Note that iptables appends a new header with the to source address because ipaq on WLAN have private IP address which don't allow their packets to be routed on the internet. To make this change becomes permanent, add these 2 lines in the /etc/rc.d/rc.local file. The client has to be configured too to use iptables. To do that, two files have to be edited:
Sample /etc/sysconfig/network file:

```
NETWORKING=yes
FORWARD_IPV4=no
HOSTNAME=hostname.domain.net
DOMAINNAME=domain.net
GATEWAY=147.210.0.1
GATEWAYDEV=eth0
```

Sample /etc/resolv.conf file:

```
search domain.com
nameserver 207.217.120.92
```

3.1.3 Using PPP link

A WLAN is not necessary to connect the ipaq to the internet, a ppp link between the workstation and the ipaq can be sufficient. The configuration becomes:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -A FORWARD -s ipaq -i ppp0 -o eth0 -j ACCEPT
```

The /etc/sysconfig/network file becomes:

```
NETWORKING=yes
FORWARD_IPV4=no
HOSTNAME=hostname.domain.net
DOMAINNAME=domain.net
GATEWAY=147.210.0.1
GATEWAYDEV=ppp0
```

What is interesting with iptables is that the rules can be combined. Thus a complicated situation can be cut in many simple situations.

3.2 Use your host as a http server for iPAQ

It is nice to download ipkg packages or useful files to your workstation and load them from there using the http protocol. Moreover in a such case, you don't have to pay the internet connection or to make a search on the internet to find back a file.
First, your host has to run a http server, apache one for example. To restart it, execute:

```
/usr/local/apache2/bin/apachectl restart
```

With your web browser go to the page: `http://localhost/.`
The configuration file to edit is /usr/local/apache2/conf/httpd.conf where many parameters can be set as shown in the following partial example:

```
ServerName 172.18.0.2:80
UseCanonicalName On
DocumentRoot "/home/user/ipaq"
<Directory "/home/user/ipaq">
```

The DocumentRoot is the directory out of which your documents will be served. By default, all requests are taken from this directory. Make sure , that the complete path has execution rights for user, group and others. Otherwise, you will have an error 403 (permission denied) when accessing the download dir.
The httpd.conf file is fully commented and so well documented. To have further information, let's see the official site: `http://httpd.apache.org/docs-2.0/`.
To test if everything is ok, try the following command with your ipaq:

```
cd /tmp
wget http://localhost/anyfile
```

The wget transfer printout should be seen and a file named anyfile should be present in /tmp.

3.3 Using dillo, wget...

3.3.1 Using dillo

Now, there are many ways to proceed dillo installation:

- Internet: you only have to run ***ipkg [options] install dillo***. Note that it only works if the ipaq is connected to the internet. The pros of such a method is that you don't have to worry about depencies that are not installed, they are automatically installed.
- Your own http server: you have to run ***ipkg [options] install dillo_ version_ arm.ipk***. Nonetheless, the /etc/ipkg.conf file has to be modified: the line begining with `src familiar http://familiar.handhelds.org/...` must be replaced with `src familiar http://localhost/where/dillo/is/`. The cons of such a method, and the following, is that you have to install dependencies that are not installed.

- NFS: you have to run ***cd /mnt/nfs/where/dillo/is/*** and then ***ipkg [options] install dillo_version_arm.ipk***.
- wget: run ***cd /tmp*** then ***wget http://localhost/where/dillo/is/dillo_version_arm.ipk*** and eventually i***pkg [options] install dillo_version_arm.ipk***.

Once dillo is installed, his configuration is quite easy. The configuration file is located in $HOME/.dillo/ and is called dillorc. This file is quite simple to understand and there should not be problems with it.
Dillo is quite simple and easy to use, but his main drawback is that it does not support some langages like javascript or many protocols like https, ftp...

3.3.2 Using wget

wget is a network utility to retrieve files from the web using http and ftp protocols. It's a non-interactive utility which means that it can work in the background whereas no user is present.
The more useful options of wget are the following:

-o logfile: Log all messages to logfile instead of to the standard error.

-i file: Read URLs from file, in which case no URLs need to be on command line.

-O file: All the downloaded files will be concatenated together and written to file.

-c: Continue getting a partially-downloaded file.

–proxy=on/off: Turn proxy support on or off. The proxy is on by default if the appropriate environmental variable is defined in the .wgetrc file.

-r: Turn on recursive rettrieving.

-l depth: Specify recursion maximum depth level.

-k: After the download is complete, convert the links in the document to make them suitable for local viewing.

-p: This option causes Wget to download all the files that are necessary to properly display a given HTML page.

References

[1] http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/

[2] http://handhelds.org/

[3] http://www.tldp.org/

[4] http://www.orinocowireless.com/

[5] http://www.netfilter.org/documentation/index.html#HOWTO

[6] http://www.uv-ac.de/