1 Why Use Wireless?

Because you have to. Warehouses, hospitals, robots, anywhere you cannot connect with wire. Disadvantages: For data, distortion is not tolerated at all: any errors have to be corrected (complicating the data transfer) or restransmitted (slowing the data-rate).

2 Where are We in the Network?

The **IEEE 802.11a** standard is located in the 5.0 GHz Unlicenced National Information Infrastructure frequency band, utilising Orthogonal-Frequency-Division-Multiplexing modulation with Quadrature-Phase-Shift-Keying. This band competes in part with some Cordless Phones but is relatively free from interference and is fast but short-range. **IEEE 802.11b** is located in the 2.4 GHz Industrial, Scientific and Medical frequency band, utilising Direct-Sequence-Spread-Spectrum modulation. This band also contains these sources of interference: Bluetooth, Microwave ovens, some Cordless Phones and Amateur Radio, and is slower but of greater range.

3 How to Manage Network Connections

In a *wired network* each computer has an ethernet NIC (Network Interface Card) which is configured, then brought up (made available to the router) and given an IP address on the same subnet, and connected to a hub. Each computer can now communicate with any other using TCP/IP commands. In a *mobile network* each computer has a wireless NIC (Network Interface Card) which is configured, then brought up (made available to the router) and given an IP address on the same subnet, and using the airwaves each computer can communicate with any other using TCP/IP commands.

Here we shall try to describe what is intended, so you can work out for yourself what to do and how to debug, in a logical sequence, depending on real device operation.

4 Essential Steps for Debugging and Connecting

My experience is with two types of wireless network interface cards: the first is embedded in the computer **(the Atheros chipset in the EeePC-701)** with Debian Lenny 5.0 installed and running the Linux kernel 2.6.26; the second is a PCMCIA card **(the RaLink chipset in the D-Link AirPlus DWL-G630)** on Debian Etch 4.0 running the Linux kernel 2.6.18, and also on Debian Lenny 5.0 running the Linux kernel 2.6.26. Here I shall go through the things that I do to get wireless working on each.

5 Installation of Necessary Software to run NICs

Here are some things I discovered were needed to get the chipsets working.

1. D-Link DWL-G630 under Debian 4.0 (Etch) running kernel 2.6.18

```
# echo "deb http://www.backports.org/debian/ etch-backports main contrib non-free"
      >> /etc/apt/sources.list
# apt-get install firmware-ralink
# apt-get install build-essential linux-headers-$(uname -r)
$ wget http://rt2x00.serialmonkey.com/rt61-cvs-daily.tar.gz
$ tar zxvf rt61-cvs-daily.tar.gz
$ cd rt61-cvs-2[TAB]/M[TAB]                 [TAB] means just press the TAB key
$ make
# make install
# modprobe rt61
# echo "rt61">>/etc/modules
```

2. D-Link DWL-G630 under Debian 5.0 (Lenny) running kernel 2.6.26

```
# apt-get install firmware-ralink
```

3. Atheros AR242x under Debian 5.0 (Lenny) running kernel 2.6.26

```
# apt-get install madwifi-source madwifi-tools module-assistant
# module-assistant update
# module-assistant prepare
# module-assistant auto-install madwifi   [for putting Atheros into ad-hoc mode]
# modprobe ath_pci
```

6 Experience Setting up an Ad-Hoc Network

I set up three computers – call them GREEN, BLUE and YELLOW - that intend to form an ad-hoc network called `lsgnet`. (Refer to the internet for help on installing the drivers for your actual chipsets, or the procedures above if you have the cards I am using for this demonstration.) In what follows, `$` is the prompt for a normal user, `#` is the root prompt; and sometimes not all of the command output is shown, just the interesting parts.

6.1 GREEN: Configure and Activate the Wireless NIC

The GREEN computer runs Debian 4.0 (Etch) kernel 2.6.18-6-686. using a PCMCIA AirPlus G D-Link DWL-G630 wireless card.

1. If it is installed, stop network-manager interfering with your efforts.

```
$ sudo /etc/rc2.d/S??network-manager stop
$ sudo /etc/rc2.d/S??network-manager-dispatcher stop
```

2. Insert D-Link DWL-G630 (H/W Ver. E1 F/W Ver 5.00) card into PCMCIA slot.

3. Check that the card was seen being inserted by the kernel.

```
$ dmesg
pccard: CardBus card inserted into slot 0
rt61: RT61: RfIcType= 3
```

4. Check that the card has been recognised by reading its identity from the hardware.

```
$ lspci|grep RT
06:00.0 Network controller: RaLink RT2561/RT61 rev B 802.11g
```

5. Check that the necessary modules were subsequently loaded by kernel.

```
$ lsmod|grep rt61
rt61                  178948    0
firmware_class          9600    2   rt61,pcmcia
```

6. Find out what the interface has been named and use it in the script

```
$ /sbin/ifconfig -a
wlan0    Link encap:Ethernet   HWaddr  00:15:E9:B9:A6:9E
```

7. Bring the interface up to see what networks are around.

```
$ sudo ifconfig wlan0 up
$ sudo iwlist wlan0 scanning
wlan0    No scan results
```

8. Ensure that the interface is down whilst being re-configured.

```
$ sudo ifconfig wlan0 down
```

9. Examine the default properties of the wireless network interface card.

```
$ /sbin/iwconfig wlan0
wlan0     RT61 Wireless  ESSID:""
          Mode:Managed  Frequency:2.412 GHz  Bit Rate=54 Mb/s
          ...
```

10. Put NIC into ad-hoc mode and define the network name (`lsgnet`).

```
$ sudo iwconfig wlan0 mode ad-hoc essid lsgnet
$ /sbin/iwconfig wlan0
wlan0     RT61 Wireless  ESSID:"lsgnet"
          Mode:Ad-Hoc  Frequency:2.412 GHz  Bit Rate=54 Mb/s
          Encryption key:off     ...
```

11. Bring the interface up and note that it has created a unique Basic Service Set ID (*viz.:* `EE:5F:B1:1D:76:AD`) for the network (`lsgnet`).

```
$ sudo ifconfig wlan0 up
wlan0     RT61 Wireless  ESSID:"lsgnet"
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: EE:5F:B1:1D:76:AD
          Bit Rate=11 Mb/s      ...
```

6.2 BLUE: Configure and Activate the wireless NIC.

The BLUE computer runs Debian 5.0 (Lenny) kernel 2.6.26-1-686, using a PCMCIA AirPlus G D-Link DWL-G630 wireless card.

1. If it is installed, stop network-manager interfering with your efforts.

   ```
   $ sudo /etc/rc2.d/S??network-manager stop
   $ sudo /etc/rc2.d/S??network-manager-dispatcher stop
   ```

2. Insert D-Link DWL-G630 (H/W Ver. E2 F/W Ver. 5.00) card into PCMCIA slot.
3. Check that the card was seen being inserted by the kernel.

   ```
   $ dmesg
   [  120.108245] pccard: CardBus card inserted into slot 0
   [  120.108573] rt61 0000:03:00.0: enabling device (0000 -> 0002)
   [  120.108843] firmware: requesting rt2561.bin
   [  120.326944] rt61: RT61: RfIcType= 3
   ```

4. Check that the card has been recognised by reading its identity from the PCI bus.

   ```
   $ lspci|grep RT
   03:00.0 Network controller: RaLink RT2561/RT61 rev B 802.11g
   ```

5. Check that the necessary modules were subsequently loaded by the kernel.

   ```
   $ lsmod|grep rt61
   rt61pci                20960  0
   crc_itu_t               2080  1 rt61pci
   rt2x00pci               7648  1 rt61pci
   rt2x00lib              22432  2 rt61pci,rt2x00pci
   eeprom_93cx6            2144  1 rt61pci
   rt61                  170148  0
   firmware_class          6816  3 rt2x00lib,rt61,pcmcia
   ```

6. Find out what the interface has been named and use it in the script.

   ```
   $ /sbin/ifconfig -a
   wlan0    Link encap:Ethernet    HWaddr: 00:1b:11:ca:03:a9
   ```

7. Bring the interface up and examine the default properties of the NIC.

   ```
   $ sudo ifconfig wlan0 up
   $ /sbin/iwconfig wlan0
   wlan0     RT61 Wireless  ESSID:""  Nickname:""
             Mode:Managed  Frequency:2.412 GHz  Bit Rate=54 Mb/s      ...
   ```

8. Scan for networks – this finds the one we just activated on GREEN.

```
$ sudo iwlist wlan0 scanning
wlan0   Scan completed :
        Cell 01 - Address: EE:5F:B1:1D:76:AD
              ESSID:"lsgnet"
              Mode: Ad-Hoc
              Channel: 1
              Encryption key: off
              Bit Rates:0 kb/s
              Quality:0/100   Signal level:-37 dBm   Noise level:0 dBm
```

9. Ensure that the interface is down whilst we re-configure it.

```
$ sudo ifconfig wlan0 down
```

10. Put NIC into ad-hoc mode and prepare to associate with our network (`lsgnet`).

```
$ sudo iwconfig wlan0 mode ad-hoc essid lsgnet
$ /sbin/iwconfig wlan0
wlan0     RT61 Wireless  ESSID:"lsgnet"  Nickname:""
          Mode:Ad-Hoc  Frequency:2.412 GHz  Bit Rate=54 Mb/s      ...
```

11. Bring the interface up and wait until it is associated with network `lsgnet`.

```
$ sudo ifconfig wlan0 up
$ time (while [ "$(/sbin/iwconfig wlan0|grep 'Cell'|wc -l)" -ne 0 ]; do :; done)
real   0m3.368s
user   0m1.160s
sys    0m2.116s
$ /sbin/iwconfig wlan0
wlan0     RT61 Wireless  ESSID:"lsgnet"  Nickname:""
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: EE:5F:B1:1D:76:AD
          Bit Rate=11 Mb/s     ...
```

12. Perform a scan of the wireless networks in the area.

```
$ sudo iwlist wlan0 scan
wlan0   Scan Completed :
        Cell 01 - Address: EE:5F:B1:1D:76:AD
              ESSID:"lsgnet"
              Mode: Ad-Hoc
              Encryption key: off
              Channel: 1
              Bit Rates:0 kb/s
              Quality:87/100   Signal level:-41 dBm   Noise level:0 dBm
```

6.3 YELLOW: Configure and Activate the wireless NIC.

The YELLOW computer runs Debian 5.0 (Lenny) kernel 2.6.26-6-686, using the embedded Atheros AR242x chipset in the EeePC-701.

1. If it is installed, stop network-manager interfering with your efforts.

   ```
   $ sudo /etc/rc2.d/S??network-manager stop
   $ sudo /etc/rc2.d/S??network-manager-dispatcher stop
   ```

2. Check that the chipset was seen by the kernel.

   ```
   $ dmesg|grep Ath
   [    8.812824] Atheros(R) L2 Ethernet Driver - version 2.0.5
   $ dmesg|grep ath
   [ 9.068937] ath_hal: module license 'Proprietary' taints kernel.
   [10.205777] MadWifi: ath_attach: Switching rfkill capability off.
   [15.236895] wifi0: Atheros AR2425 chip found (MAC 14.2, PHY SChip 7.0, Radio 10.2)
   [15.455210] ath_pci: wifi0: Atheros 5424/2424: mem=0xfbef0000, irq=18
   ```

3. Check that the card has been recognised by reading its identity from hardware.

   ```
   $ lspci|grep Ath
   01:00.0 Ethernet controller: Atheros Communications Inc.
                        AR242x 802.11abg Wireless PCI Express Adapter (rev 01)
   ```

4. Check that necessary modules were subsequently loaded by kernel.

   ```
   $ lsmod|grep ath
   ath_rate_sample        11104  1
   ath_pci               202712  0
   wlan                  194000  4 wlan_scan_sta,ath_rate_sample,ath_pci
   ath_hal               300768  3 ath_rate_sample,ath_pci
   ```

5. Find out what the interface has been named and use it in the script.

   ```
   $ /sbin/ifconfig -a
   ath0    Link encap:Ethernet    HWaddr 00:15:af:6b:3b:0b
   ```

6. Bring the interface up to do some scanning – finds the NICs on GREEN and BLUE. Note the common BSSID (`EE:5F:B1:1D:76:AD`) now.

   ```
   $ sudo ifconfig ath0 up
   $ sudo iwlist ath0 scanning
   ath0   Scan Completed :
           Cell 01 - Address: EE:5F:B1:1D:76:AD
                 ESSID:"lsgnet"
                 Mode: Ad-Hoc
   ```

```
                    Frequency: 2.412 GHz (Channel 1)
                    Quality:82/100   Signal level:-41 dBm   Noise level:0 dBm
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2Mb/s; 5.5Mb/s; 11Mb/s
                    Extra:bcn_int=100
            Cell 02 - Address: EE:5F:B1:1D:76:AD
                    ESSID:"lsgnet"
                    Mode: Ad-Hoc
                    Frequency: 2.412 GHz (Channel 1)
                    Quality:68/100   Signal level:-41 dBm   Noise level:0 dBm
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2Mb/s; 5.5Mb/s; 11Mb/s
                    Extra:bcn_int=100
```

7. Examine the properties of the wireless network interface card.

```
$ /sbin/iwconfig ath0
ath0      IEEE 802.11g  ESSID:""  Nickname:""
          Mode:Managed  Frequency: 2.437 GHz  Access Point: Not-Associated
          Bit Rate:0 kb/s   Tx-Power:17 dBm   Sensitivity=1/1
          Encryption key:off       ....
```

8. Ensure that the interface is down so that we can re-configure it.

```
$ sudo ifconfig ath0 down
```

9. Put NIC into ad-hoc mode

```
$ sudo wlanconfig ath0 destroy
ath0    No such interface
$ sudo wlanconfig ath0 create wlandev wifi0 wlanmode adhoc
ath0
```

10. Set the common network name equal to the essid string – note Cell is `Not-Associated`.

```
$ sudo iwconfig ath0 essid lsgnet
$ /sbin/iwconfig ath0
ath0      IEEE 802.11g   ESSID:""   Nickname:""
          Mode:Ad-Hoc  Channel:0   Cell: Not-Associated
          ...
```

11. Bring the interface up and wait until it is associated with network `lsgnet`.

```
$ sudo ifconfig ath0 up
$ while [ "$(/sbin/iwconfig ath0|grep 'Not-Associated'|wc -l)" -ne 0 ]; do :; done
real   0m2.777s
user   0m0.992s
```

```
sys    0m1.484s
$ /sbin/iwconfig ath0
ath0      IEEE 802.11g  ESSID:"lsgnet"  Nickname:""
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: EE:5F:B1:1D:76:AD
          Bit Rate:0 kb/s   Tx-Power:17 dBm   Sensitivity=1/1
          Link Quality=52/70  Signal level=-96 dBm  Noise level=-96 dBm    ...
```

12. Scan for networks.

```
$ sudo iwlist ath0 scanning
ath0    Scan completed :
        Cell 01 - Address: EE:5F:B1:1D:76:AD
                ESSID:"lsgnet"
                Mode: Ad-Hoc
                Frequency:2.412 GHz (Channel: 1)
                Quality:49/70   Signal level:-37 dBm   Noise level:0 dBm
                Encryption key: off
                Bit Rates:1 Mb/s; 2Mb/s; 5.5Mb/s; 11Mb/s
                Extra:bcn_int=100
        Cell 01 - Address: EE:5F:B1:1D:76:AD
                ESSID:"lsgnet"
                Mode: Ad-Hoc
                Frequency:2.412 GHz (Channel: 1)
                Quality:68/70   Signal level:-37 dBm   Noise level:0 dBm
                Encryption key: off
                Bit Rates:1 Mb/s; 2Mb/s; 5.5Mb/s; 11Mb/s
                Extra:bcn_int=100
```

6.4 Assign each NIC a unique IP address

The IP addresses need to be on the same subnet, e.g., the private network 10.0.0.

```
user@GREEN $ sudo ifconfig wlan0 10.0.0.11
$ /sbin/ifconfig wlan0
wlan0    Link encap: Ethernet   HWaddr 00:15:E9:B9:A6:9E
         inet addr: 10.0.0.11   Bcast: 10.255.255.255  Mask: 255.0.0.0
         UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
         RX packets: 3135   errors: 0   dropped: 0  overruns: 0  frame: 0
         TX packets: 173    errors: 14  dropped: 0  overruns: 0
         Collisions: 11   txqueuelen: 1000
         RX bytes:180633 (176.3 kiB)   TX bytes:2586 (2.5 kiB)
         Interrupt 11

user@BLUE $ sudo ifconfig wlan0 10.0.0.22
wlan0    Link encap: Ethernet   HWaddr 00:1b:11:ca:03:a9
         inet addr: 10.0.0.22   Bcast: 10.255.255.255  Mask: 255.0.0.0
         UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
         RX packets: 5186   errors: 0   dropped: 0  overruns: 0  frame: 0
         TX packets: 899    errors: 1   dropped: 1  overruns: 0
         Collisions: 0   txqueuelen: 1000
         RX bytes:299773 (292.7 kiB)   TX bytes:5272 (5.1 kiB)
         Interrupt 20

user@YELLOW $ sudo ifconfig ath0 10.0.0.33
ath0     Link encap: Ethernet   HWaddr 00:15:af:6b:3b:0b
         inet addr: 10.0.0.33   Bcast: 10.255.255.255  Mask: 255.0.0.0
         UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
         RX packets: 0   errors: 0   dropped: 0  overruns: 0  frame: 0
         TX packets: 6   errors: 0   dropped: 0  overruns: 0
         Collisions: 0   txqueuelen: 0
         RX bytes:0 (0.0 kiB)   TX bytes:460 (460.0 B)
```

6.5 Check Wireless Connectivity

1. For example, try to ping BLUE and YELLOW from GREEN.

```
user@GREEN $ ping 10.0.0.22
.....4.52ms .....0.96ms .....1.02ms .....0.94ms  (etc)
user@GREEN $ ping 10.0.0.33
.....6.73ms .....8.03ms .....50.7ms .....5.71ms  (etc)
```

2. You may now connect as you wish and do what you want. For example, use TCP/IP commands to connect from GREEN (10.0.0.11) to BLUE (10.0.0.22) and copy files from BLUE to YELLOW (10.0.0.33).

```
user@GREEN $ ssh user@10.0.0.22
user@10.0.0.22's passwd: ******
user@BLUE:~$ scp file user@10.0.0.33:/home/user/file
file                              100%  1587  1.6KB/s  00.00
```

7 Useful Reference Books

"Residential Networks" Les Baxter (ISBN 1-4018-6267-5; Thomson 2006).

"802.11 Wireless Networks: the definitive guide (2nd Ed.)" Matthew Gast (ISBN 0-596-10052-3; O'Reilly 2004).

"Wireless Networks: First-Step" Jim Geier (ISBN 1-58720-111-9; Cisco Press 2005).

"Linux Unwired" Weeks, Dumbill & Jepson (ISBN 0-596-00583-0; O'Reilly 2004).

"Wireless Hacks" Flickenger & Weeks (ISBN 0-596-10144-9; O'Reilly 2006).

"IEEE Std 802.11-2007 – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" IEEE Computer Society 12 June 2007).

"A Technical Tutorial on the IEEE802.11 Protocol" Pablo Brenner, BreezeCOM Wireless Communications, 1997).