MadWifi 'First Time User' HOWTO

Welcome to the MadWifi 'first time user' howto. This document is intended to be a complete set of instructions on how to get, install and use the latest MadWifi driver. No previous experience of wireless networking under Linux is assumed.

This howto describes the manual build process for MadWifi drivers. However your Linux distribution may already distribute pre-built (but old) MadWifi drivers. Distributions may also have their own way of building kernel modules for integration in the package management system. Have a look at UserDocs/Distro.

Note: This guide only shows you the steps to take for managed mode operation (aka. station). Refer to the UserDocs for more information about other modes.

Requirements

For Debian OS like root : # apt-get install build-essential

Make sure you have Linux headers installed: sudo apt-get install linux-headers-$(uname -r)

(If you don't you will get an error: /lib/modules/2.6.24-19-server/build is missing, please set KERNELPATH.)

Please check Requirements before proceeding. This includes having an Atheros chipset physically installed.

Getting MADWiFi Sources

Download a MadWifi release from sourceforge.net and unpack it. Open a shell terminal in the MadWifi source directory.

Removing old modules

For this step you must be logged on as root.

First, set all your MadWifi devices down:

```
ifconfig ath0 down
ifconfig wifi0 down
#Repeat these 2 ifconfig lines for every MadWifi device you have (ath1, etc)
```

Assuming that you're inside the MadWifi directory, execute the following scripts to remove the current modules from your system and its memory:

```
cd scripts
./madwifi-unload
./find-madwifi-modules.sh $(uname -r)
cd ..
```

You should then be asked if you are sure that you want to remove the old modules.

Building MadWifi

Now that you have the MadWifi code, it's time to compile it into the actual driver. Thankfully, this is easy.

Assuming that you've met all of the requirements above, and you're inside the MadWifi directory, you can just type:

```
make
```

Which will start the build process. Watch for any questions you might be prompted to answer - when it finishes, quickly scan through for any errors. If everything went according to plan, you can proceed to the next step. Make sure you have all the Requirements or the build process may fail.

Installing MadWifi

This step will take the built MadWifi, and install it on your system. Once again, `make` does all of the work for you.

This step needs to be done as *root*, so either type `su` and enter *root's* password, or if you have it set up (e.g. Ubuntu), prefix the following command with `sudo`.

To install the driver, type:

```
make install
```

This will copy all of the modules, tools and man pages to the correct directories on your system. You've now completed the basic install.

Loading the MadWifi Module

This step will load the MadWifi driver module into your running system. This essentially lets all other software know how to talk to your MadWifi hardware.

This step needs to be done as *root*, so either type `su` and enter *root's* password, or if you have it set up (e.g. Ubuntu), prefix the following command with `sudo`.

To load the driver module, type:

```
modprobe ath_pci
```

If you have any problems with building the MadWifi driver, please refer to UserDocs/BuildProblems.

Creating an Interface

MADWiFi supports *virtual access points*, which means you can create more than one wireless device per wireless card. By default, a *sta* mode VAP is created, which is MadWifi talk for a 'managed mode wireless interface'.

If your svn snapshot is more recent than the 23rd January 2006, (r1407) than you can skip the following step:

If not, then follow these instructions to make a normal *station* mode interface. Type (as root):

```
wlanconfig ath0 create wlandev wifi0 wlanmode sta
```

If wlanconfig doesn't work, you retry it after executing 'wlanconfig ath0 destroy'.

Now, if you type `iwconfig`, you should see a list like the following:

```
eth0      no wireless extensions.


lo        no wireless extensions.

wifi0     no wireless extensions.

ath0      IEEE 802.11g  ESSID:""
          Mode:Managed      Frequency:2.457   GHz      Access   Point:
00:00:00:00:00:00
          Bit Rate:0 kb/s   Tx-Power:20 dBm   Sensitivity=0/3
          Retry:off   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality=0/94  Signal level=-95 dBm  Noise level=-95 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Then we need to bring up the wireless interface. This is done by typing (as root):

```
ifconfig ath0 up
```

There is more information on the creating of interfaces in UserDocs.

Scanning for Access Points

If you know that there are some APs around, having a quick scan can be an excellent way of getting some instant gratification, and knowledge that everything's working OK.

The first step is to insert the scanning module. Type (as root):

```
modprobe wlan_scan_sta
```

Next, you can do the actual scan, which can be done in two ways.

The first way is specific to MadWifi. and gives you a nice, succinct results table.

This is done by issuing the command (again, as root):

```
wlanconfig ath0 list scan
```

This should give you a list that looks something like this:

```
SSID            BSSID              CHAN RATE  S:N   INT CAPS
eddie           00:06:25:e8:3a:05    6   54M 36:0   100 EPs
```

The second way is the more usual way of scanning (and works with other wireless cards), it also gives you somewhat more information, and is therefore a bit less easy to read. The command is:

```
iwlist ath0 scan
```

Which gives an output which looks like this:

```
ath0      Scan completed :
          Cell 01 - Address: 00:06:25:E8:3A:05
                    ESSID:"eddie"
                    Mode:Master
                    Frequency:2.437 GHz (Channel 6)
                    Quality=37/94  Signal level=-58 dBm  Noise level=-
95 dBm
                    Encryption key:on
                    Bit Rate:1 Mb/s
                    Bit Rate:2 Mb/s
                    Bit Rate:5.5 Mb/s
                    Bit Rate:11 Mb/s
                    Bit Rate:18 Mb/s
                    Bit Rate:24 Mb/s
                    Bit Rate:36 Mb/s
                    Bit Rate:54 Mb/s
                    Bit Rate:6 Mb/s
                    Bit Rate:9 Mb/s
```

```
                    Bit Rate:12 Mb/s
                    Bit Rate:48 Mb/s
                    Extra:bcn_int=100
```

Especially useful is the line reading `Encryption key:on`, which indicates that this AP is running some kind of WEP.

If you get a message such as:

```
ath0      Failed to read scan data : Resource temporarily unavailable
```

instead of actual scan results, and you are in an environment that requires a shared encryption key, try running:

```
iwconfig ath0 key <yourkey>
iwpriv ath0 authmode 2
```

This will tell the card that it is operating in a restricted, shared-key environment, and thus it needs to use the key you supply with iwconfig. To use an open system key (which is often considered more secure) use iwpriv authmode 1:

```
iwconfig ath0 key <yourkey>
iwpriv ath0 authmode 1
```

Once this is done, re-run the scan, and you may see proper results.

Connecting to an open AP

If the scan you did above says "Encryption key:off", then you may not actually need to do anything to make the driver associate with the AP, since the driver will automatically connect to the one with the strongest signal. It's still a good idea to know how to tell it which AP/Network to connect to however.

Typing (as root):

```
iwconfig ath0 essid "eddie"
```

Will connect you to the AP with the ESSID (network name) eddie.

You can also specify which AP you want to connect to by using its MAC address (in topmost scan output, this is the field marked BSSID, and in the bottom one, it's the field called Address).

To specify the AP using its MAC/BSSID, type:

```
iwconfig ath0 ap 00:06:25:E8:3A:05
```

If you then decide you want to let the driver decide automatically which AP to associate with, you can type:

```
iwconfig ath0 ap any
```

Or:

```
iwconfig ath0 ap auto
```

Just being connected to an AP is like having an ethernet cable plugged into your machine - you're now 'on the network', However without getting an IP address you can't really do anything. For this reason the next step is to get an IP address, and again, this is fairly easy. The tools you use to get an IP with a wireless interface, are exactly the same as they are for a wired one.

First of all, you need to know if the network you're connecting to has a DHCP server (this is a server which gives you a network address automatically, and tells you how to do things like access the internet and perform DNS lookups).

There are several methods of finding out whether or not the network has DHCP:

- If you've used the wireless card in Windows (on this network), and you let it 'Obtain an IP address automatically', then the chances are that the network *does*.
- If you've used the wireless card in Windows (on this network), and you had to put in an IP address yourself, then it *doesn't*, and if you can, you should find the network details you used previously.
- If you've connected to some kind of combined wireless access point/router, then it almost certainly *will*.
- If you're in an internet Cafe, then it probably *does*.
- If there's someone around who knows about the network you're connected to, then ask them if it has DHCP.
- If you can't find out whether or not the network has DHCP, then try the method below anyway, it might work, and you'll have answered the question yourself.

Connecting with DHCP

There are various different DHCP clients available, and which one(s) are available is largely dependent on which GNU/Linux distribution you're using. The most common one is `dhclient`, which is what will be used here.

To get an IP address from a DHCP enabled network, type (as root):

```
dhclient ath0
```

You should then see something like:

```
Internet Software Consortium DHCP Client 2.0pl5
```

```
Copyright 1995, 1996, 1997, 1998, 1999 The Internet Software
Consortium.
All rights reserved.

Please contribute if you find this software useful.
For info, please visit http://www.isc.org/dhcp-contrib.html

eth1: unknown hardware address type 24
eth1: unknown hardware address type 24
Listening on LPF/ath0/00:02:6f:20:14:81
Sending on   LPF/ath0/00:02:6f:20:14:81
Sending on   Socket/fallback/fallback-net
DHCPDISCOVER on ath0 to 255.255.255.255 port 67 interval 7
DHCPOFFER from 192.168.0.254
DHCPREQUEST on ath0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.254
bound to 192.168.0.152 -- renewal in 7200 seconds.
```

The bottom line shows that the DHCP server allocated us the address 192.168.0.152. It will also have told the machine where to find a DNS server and gateway, if they're available.

Alternately, your system may have `dhcpcd` installed instead of `dhclient`:

```
dhcpcd ath0
```

If the network you've connected to is connected to the internet, then you should be able to type:

```
ping bbc.co.uk
```

To see if everything is working. If it is, you should get an output like this:

```
PING bbc.co.uk (212.58.224.131) 56(84) bytes of data.
64 bytes from rdirwww-vip.thdo.bbc.co.uk (212.58.224.131): icmp_seq=1
ttl=119 time=15.4 ms
64 bytes from rdirwww-vip.thdo.bbc.co.uk (212.58.224.131): icmp_seq=2
ttl=119 time=14.3 ms
64 bytes from rdirwww-vip.thdo.bbc.co.uk (212.58.224.131): icmp_seq=3
ttl=119 time=15.1 ms
```

Connecting without DHCP

Connecting to a network without DHCP makes life a bit more complicated - you'll need to know a few details about the network, including:

- An IP address which you're allowed to use. This might be something like 192.168.0.10, but could be anything.
- The netmask of the network. This is usually 255.255.255.0, or 255.255.0.0, but again, it could be just about anything. The netmask describes the address range of the local network.

- The address of a local nameserver (if there is one). A nameserver is used to get the IP address from a hostname like **madwifi.org**, and vice versa.
- The address of the network's gateway (if there is one). A gateway allows you to connect to networks outside of the local subnet - usually the internet.
- The local domain name of the network (optional). This is the name which is prefixed to all hostnames in the local network. E.g.: suppose two machines on the local network are called `ns1.localnet.com` and `ns2.localnet.com`. In this case, the local domain name would be `localnet.com`.

Getting any one of the above pieces of information wrong could mean that your network connection doesn't work properly, but won't do any permanent damage.

To assign the IP address and netmask to the interface made above (`ath0`), type the following (as root):

```
ifconfig ath0 <IP address> netmask <netmask> up
```

To use the nameserver address, open `/etc/resolv.conf` in a text editor. Again, you need to do this as root:

```
gedit /etc/resolv.conf
```

If you prefer another editor, replace gedit with the command that runs it.

Once the file is opened, add a '#' to any lines which start with the word '`nameserver`', and add your own lines like the following:

```
nameserver <nameserver IP>
```

If you have multiple nameserver addresses, you can add more than one 'nameserver' line. If you know the network's local domain, add a line like this:

```
search <local domain>
```

Then save the file, and close the editor.

The final step is to tell your machine to use the gateway (if there is one). To do this, you must use the `route` command, which tells the kernel how to accesses different subnets. As root, you should type:

```
route add default gw <gateway hostname>
```

Or:

```
route add default gw <gateway address>
```

You should then be able to 'see' out of the network and access the Internet. To check if everything worked, type:

```
ping bbc.co.uk
```

You should get an output like this:

```
PING bbc.co.uk (212.58.224.131) 56(84) bytes of data.
64 bytes from rdirwww-vip.thdo.bbc.co.uk (212.58.224.131): icmp_seq=1
ttl=119 time=15.4 ms
64 bytes from rdirwww-vip.thdo.bbc.co.uk (212.58.224.131): icmp_seq=2
ttl=119 time=14.3 ms

64 bytes from rdirwww-vip.thdo.bbc.co.uk (212.58.224.131): icmp_seq=3
ttl=119 time=15.1 ms
```

Connecting to an AP with WEP

Connecting to an AP using WEP encryption is very similar to connecting to an open AP. The main difference is that you need to specify the WEP key using **iwconfig**. You may also need to specify whether your WEP key is "open" or "shared" (this is a function of how the AP's WEP encryption is set up).

First, connect to the access point using "iwconfig" as described above. It is a good idea, when using WEP, to specify the AP you want to connect to (rather than just using "ap auto" or "ap any"). This is done by specifying either the AP's MAC address or its ESSID, as shown above.

Next, you may need to specify whether your WEP key is "open" or "shared." If you are using a shared WEP key, use iwpriv authmode 2 by entering this command (as root):

```
iwpriv ath0 authmode 2
```

If you are using an open WEP key (which is often considered more secure) use iwpriv authmode 1 by entering this command (as root):

```
iwpriv ath0 authmode 1
```

Once you have specified whether you are using a shared or open WEP key, enter the key by issuing this command (as root):

```
iwconfig ath0 key <wep key (in hex)>
```

Or, if you are using an ASCII key rather than a hexadecimal key, issue this command (as root):

```
iwconfig ath0 key <s:ASCII string of key>
```

This should establish a connection between your wireless card and your AP. The next step is to set up your internet connection by either using DHCP or by specifying the details of your internet connection, as explained above.

Connecting to an AP with WPA

WPA currently offers the best security scheme currently available. This extra security takes a little bit more time and effort to setup, but it is well worth it. The following links are to pages that describe MadWifi's support for WPA, and how to take advantage of it:

* UserDocs/802.11i

* UserDocs/WPA_PSK_on_Both_Ends

Removing MadWifi

Removing MadWifi is easy, but needs to be done as root. First, change to the directory with the sources in (or, if you no longer have the sources, download them again). Then type:

```
make uninstall
```

This will run the scripts which find and remove any modules and tools which have been installed.

Troubleshooting

If something goes wrong, then the first thing you should do is check back through what you've done, and make sure you did it right. It sounds simple, but when you aren't familiar with something, it's easy to make small mistakes.

After that, if something still isn't working, you should have a look at UserDocs/Troubleshooting, a page of FAQs about Madwifi driver problems. There is also quite a lot of documentation on the Madwifi wiki, under UserDocs, having a look around the wiki might prove quite handy. If there's nothing on there that matches the problems you're having, then it might be a good idea to have a look at Support. This page explains how to go about looking for, and getting help on Madwifi. In most cases, the solution to your problem will be available in either the mailing list archives, or by asking on the mailing list or IRC. Details of the both of these can be found on the Support page.

More Documentation

If you want to view some of the documentation that comes with MADWiFi, you can find it all in the `docs` directory of the source tree. Once you are in the right directory, you can compile the file users-guide.tex to a PDF file by typing:

```
make
```

You will need a latex distribution installed for this to work.

You can also compile the users-guide as HTML, by installing latex2html, and then using the command:

```
latex2html users-guide.tex
```

Which will create a directory called `users-guide` with its output inside.