

Multiband Atheros Driver for Wireless Fidelity(MadWiFi)

Ad Hoc & AP 架設

李宏杰

Date : 2010.11.23

實驗目標(1/3)

- 在一般情況下，電腦使用無線網卡（wireless NIC，NIC: Network Interface Card）上網，是視此電腦為網路工作站（station）。其實，無線上網包括兩種架構：
 - Infrastructure（BSS: Basic Service Set）
 - Ad Hoc（也稱為Infrastructureless, Independent BSS）。

實驗目標(2/3)

- 在這兩種架構下，有包含**工作站**、**存取點** (Access Point, AP) 等各種不同的角色，來完成無線網路 (wireless network) 中的各種功能。使用 Atheros chipset 製作出的無線網卡，能在 Linux 的環境下實作出其中許多角色，在 "MadWifi project" 中，用以下數種模式稱之：
 - sta : typical WLAN client station
 - ap : Access point
 - adhoc : IBSS mode
 - ahdemo : Ad-hoc Demo
 - monitor : This device can be used to "sniff" raw 802.11 frames
 - wds : Wireless Distribution System

實驗目標(3/3)

- 本實驗分為三個部分，基本架構模擬、monitor模擬、以及WDS模擬，藉此了解各種不同架構下，無線網路中各個角色的功能及運作。

實驗環境(1/2)

- 硬體：

- NB – Acer 4720 * 2
- D-Link DWL-G650 無線網卡 * 2

- 軟體(作業系統)：

- Ubuntu (個人比較推薦，目前最紅的Linux Distribution，蠻人性化的)
- CentOS
- Fedora
- FreeBSD
- ... 等等

以上Linux作業系統都可至義守大學檔案伺服器取得

<http://ftp.isu.edu.tw/>

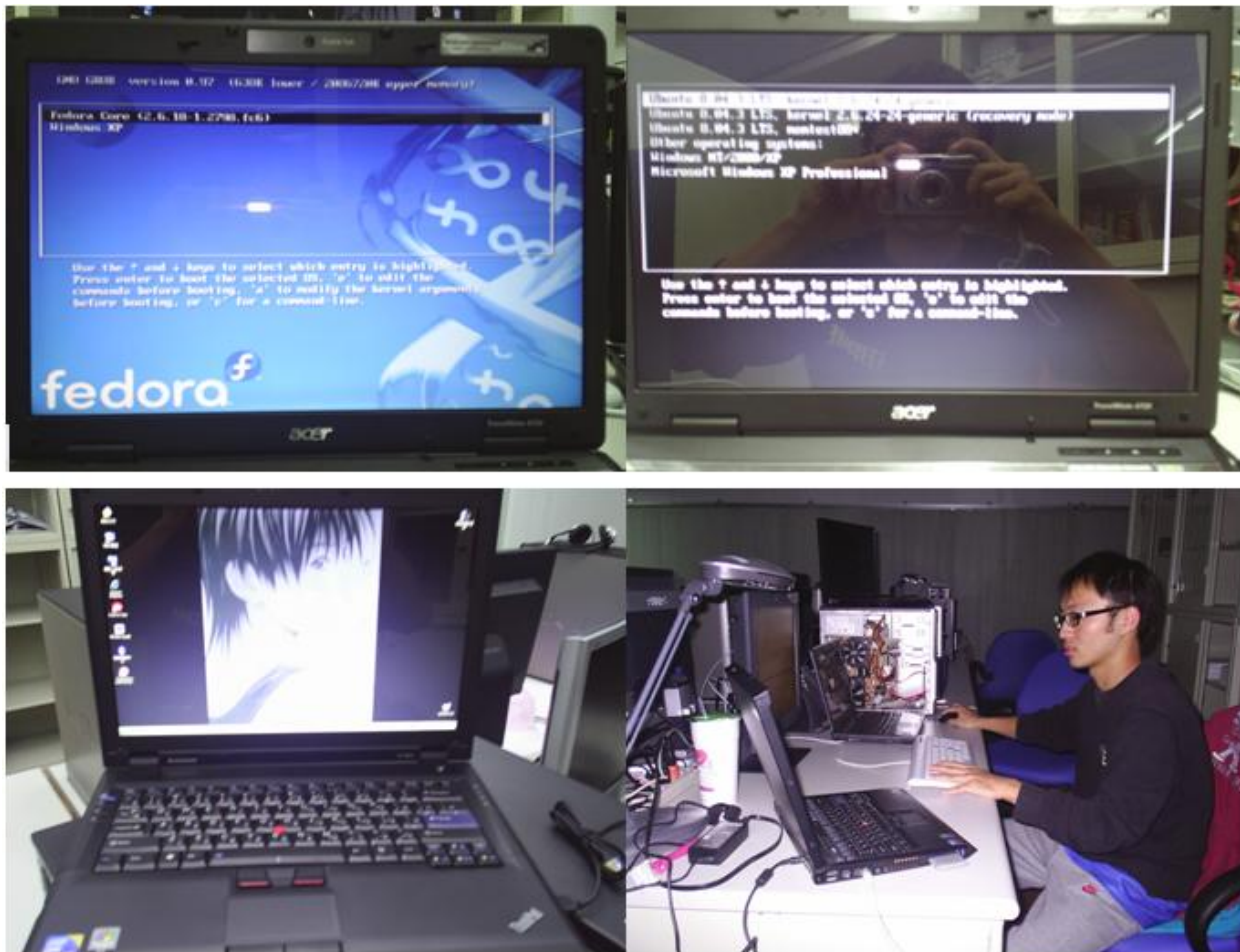
實驗環境(2/2)

- 軟體(作業系統)-MadWiFi目前支援的Ubuntu版本
 - Ubuntu Server Edition
 - Ubuntu 5.10 (Breezy)
 - Ubuntu 6.06 (Dapper)
 - Ubuntu 6.10 (Edgy)
 - Ubuntu 7.04 (Feisty)
 - Ubuntu 7.10 (Gutsy)
 - Ubuntu 8.04 (Hardy)
 - Ubuntu 8.10 (Intrepid)

實作照片(1/2)



實作照片(2/2)



安裝網卡驅動(1/3)-方法一

- 複製學姐給的檔案madwifi-0.9.4.tar.gz(目前也是這個版本)

下載的地方

<http://sourceforge.net/projects/madwifi/>

#tar -zxvf madwifi-0.9.4.tar.gz//因為此檔為.gz，
所以用 tar -zxvf解壓縮，如果為.bz，就用 -jxvf，
要對照相對的指令解壓縮

#cd madwifi-0.9.4 //切到剛解壓縮完成的目錄

#make //compile

#make install //安裝驅動

重開機，輸入指令

#ifconfig ath0 up

//如果網卡燈Link &
Act 這兩個燈同步穩定
閃爍代表驅動成功

安裝網卡驅動(2/3)-方法二

- `#apt-get update` //先更新套件伺服器清單
- `#apt-get install subversion g++ make`
//跟Fedora比較不一樣，Fedora是yum install，安裝subversion & g++ & make
- `# svn checkout http://svn.madwifi-project.org/madwifi/trunk madwifi-ng`
//利用subversion下載madwifi-ng安裝需要的檔才可compile

若上述指令不work，dhcp3-server 檔案在<https://launchpad.net/ubuntu/gutsy/i386/dhcp3-server>能夠手動下載到

安裝網卡驅動(2/3)

- `#cd madwifi-ng`
//切到剛下載完成的檔案目錄底下
- `#make` //compile
- `#make install` //安裝驅動
- `#modprobe ath_pci` //載入網卡
- `#init 6` **or** `#reboot` //重開機，linux有run level，印象中有6個，init 0為關機，init 3為單人模式。重開機進入後網卡燈Link & Act 這兩個燈同步穩定閃爍代表驅動成功

實作步驟 Ad hoc-Ubuntu(1/2)

- 建立一個shell script

```
#!/bin/sh
```

```
wlanconfig ath0 destroy //將ath0之前的設定洗掉
```

```
echo "destroy ath0" //echo為印出動作，此為表示已完成上述動作將ath0之前的設定洗掉
```

```
wlanconfig ath0 create wlandev wifi0 wlanmode adhoc //啟動wifi裝置為ad-hoc功能
```

```
echo "create ath0 as ad hoc" //在螢幕上顯示已啟動ad-hoc功能
```

```
iwconfig ath0 essid "as" //設定 essid(共同使用無線網路的電腦群組名稱)
```

```
echo "iwconfig ath0 essid as as" //顯示 essid 為 as
```

```
iwconfig ath0 channel 10 //設定 channel(IEEE 802.11有11個channel可供使用)
```

```
echo "set ath0 channel as 10" //顯示使用 channel 10
```

```
ifconfig ath0 192.168.1.15 netmask 255.255.255.0 //設定IP和子網路遮罩
```

```
echo "set ath0 ip as 192.168.1.15, netmask as 255.255.255.0" //顯示設定的IP和子網路遮罩
```

- 之後存檔 :wq離開

```
#chmod 755 adhoc.sh //更改檔案權限，才可執行，一般剛建好的檔為644  
rwx (r=4, w=2, x=1)
```

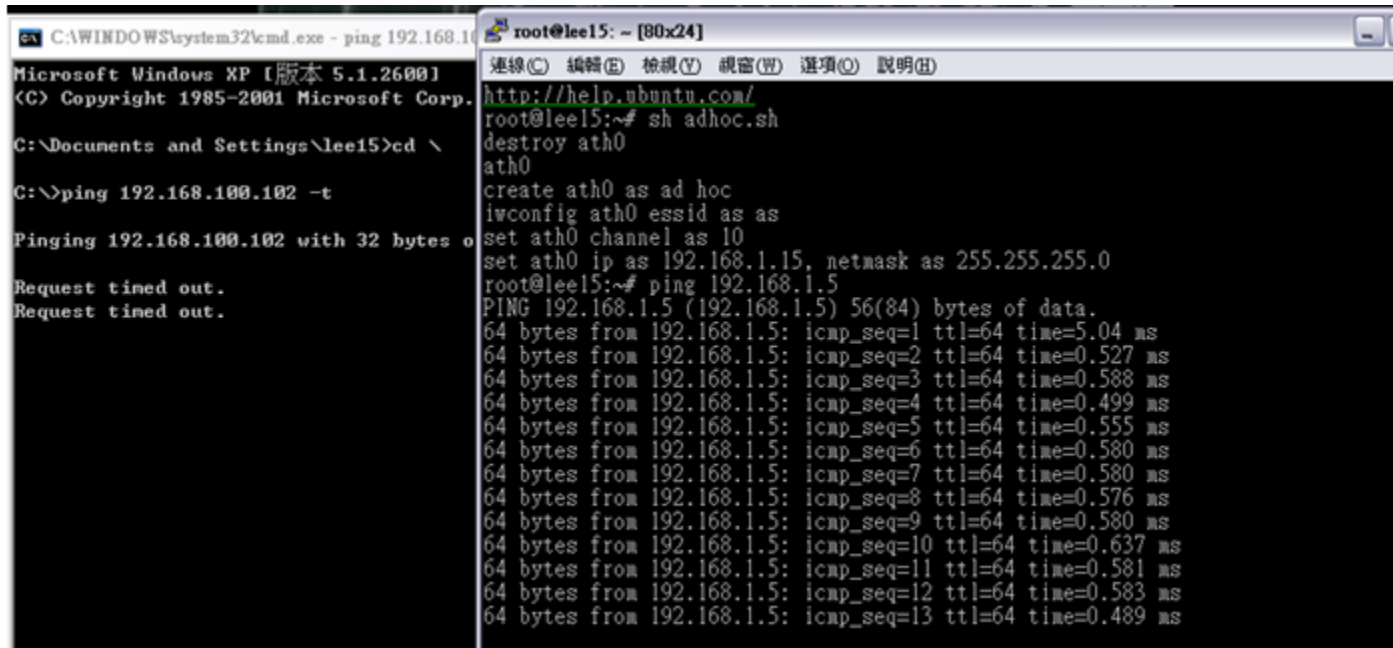
實作步驟 Ad hoc-Ubuntu(2/2)

- 執行

```
#sh adhoc.sh
```

- 另一台NB just repeat Step1 to 3，記得更改IP Address(192.168.1.5)即可
- 測試-2台NB互ping，可ping到表示成功，
Congratulations!

Ad hoc 測試結果(1/2)

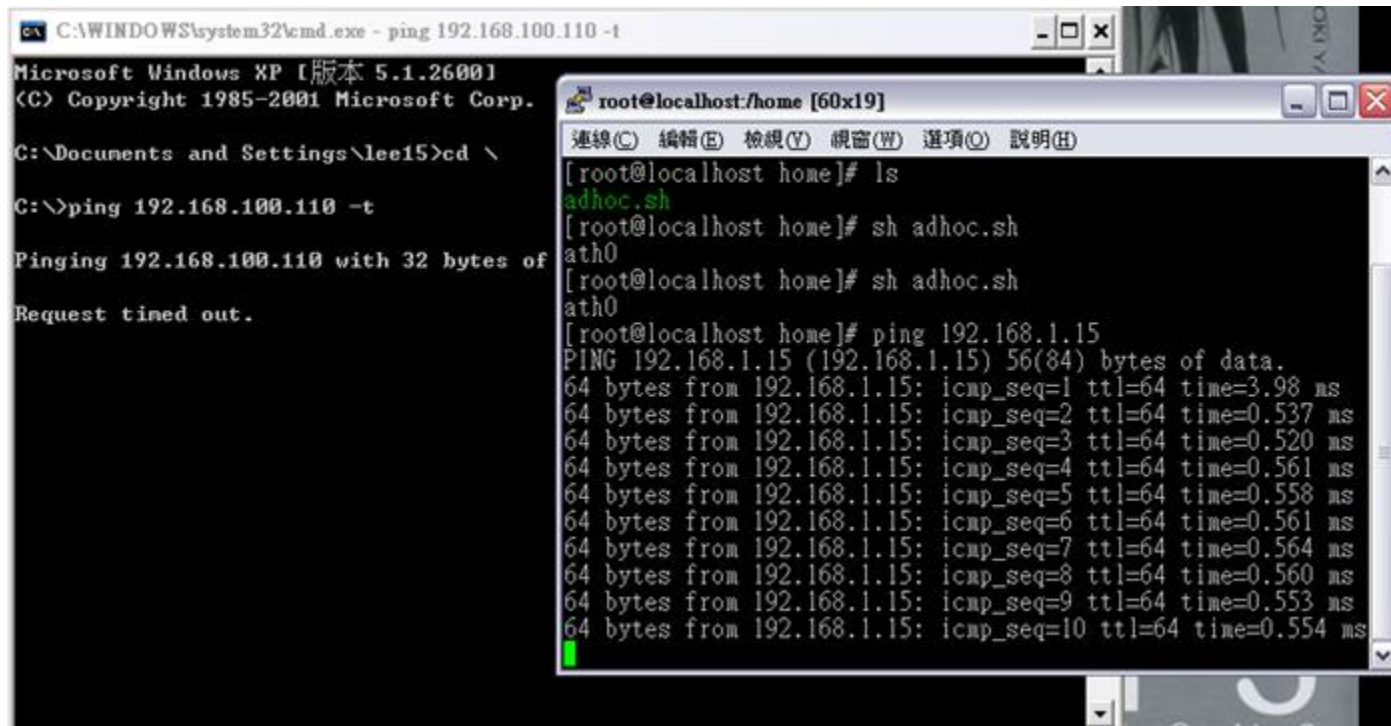


The image shows two terminal windows side-by-side. The left window is a Windows XP command prompt running a ping command. The right window is a Linux terminal running a script to configure an ad hoc network and then performing a ping test.

```
C:\WINDOWS\system32\cmd.exe - ping 192.168.1.102
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\lee15>cd \
C:\>ping 192.168.100.102 -t
Pinging 192.168.100.102 with 32 bytes of data:
Request timed out.
Request timed out.
```

```
root@lee15: ~ [80x24]
http://help.ubuntu.com/
root@lee15:~# sh adhoc.sh
destroy ath0
ath0
create ath0 as ad hoc
iwconfig ath0 essid as as
set ath0 channel as 10
set ath0 ip as 192.168.1.15, netmask as 255.255.255.0
root@lee15:~# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=64 time=5.04 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=64 time=0.527 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=64 time=0.588 ms
64 bytes from 192.168.1.5: icmp_seq=4 ttl=64 time=0.499 ms
64 bytes from 192.168.1.5: icmp_seq=5 ttl=64 time=0.555 ms
64 bytes from 192.168.1.5: icmp_seq=6 ttl=64 time=0.580 ms
64 bytes from 192.168.1.5: icmp_seq=7 ttl=64 time=0.580 ms
64 bytes from 192.168.1.5: icmp_seq=8 ttl=64 time=0.576 ms
64 bytes from 192.168.1.5: icmp_seq=9 ttl=64 time=0.580 ms
64 bytes from 192.168.1.5: icmp_seq=10 ttl=64 time=0.637 ms
64 bytes from 192.168.1.5: icmp_seq=11 ttl=64 time=0.581 ms
64 bytes from 192.168.1.5: icmp_seq=12 ttl=64 time=0.583 ms
64 bytes from 192.168.1.5: icmp_seq=13 ttl=64 time=0.489 ms
```

Ad hoc 測試結果(2/2)



The image shows two overlapping terminal windows. The background window is a Windows XP command prompt titled 'C:\WINDOWS\system32\cmd.exe - ping 192.168.100.110 -t'. It shows the user 'lee15' in the 'C:\' directory running a continuous ping to 192.168.100.110, which results in 'Request timed out.'.

The foreground window is a Linux terminal titled 'root@localhost:/home [60x19]'. It shows a root user performing several commands: 'ls' (listing 'adhoc.sh'), running 'sh adhoc.sh' twice (both resulting in 'ath0'), and running 'ping 192.168.1.15'. The ping command shows 10 successful packets with varying times (3.98 ms to 0.554 ms).

```
C:\WINDOWS\system32\cmd.exe - ping 192.168.100.110 -t
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\lee15>cd \

C:\>ping 192.168.100.110 -t

Pinging 192.168.100.110 with 32 bytes of data:
Request timed out.

root@localhost:/home [60x19]
[root@localhost home]# ls
adhoc.sh
[root@localhost home]# sh adhoc.sh
ath0
[root@localhost home]# sh adhoc.sh
ath0
[root@localhost home]# ping 192.168.1.15
PING 192.168.1.15 (192.168.1.15) 56(84) bytes of data:
64 bytes from 192.168.1.15: icmp_seq=1 ttl=64 time=3.98 ms
64 bytes from 192.168.1.15: icmp_seq=2 ttl=64 time=0.537 ms
64 bytes from 192.168.1.15: icmp_seq=3 ttl=64 time=0.520 ms
64 bytes from 192.168.1.15: icmp_seq=4 ttl=64 time=0.561 ms
64 bytes from 192.168.1.15: icmp_seq=5 ttl=64 time=0.558 ms
64 bytes from 192.168.1.15: icmp_seq=6 ttl=64 time=0.561 ms
64 bytes from 192.168.1.15: icmp_seq=7 ttl=64 time=0.564 ms
64 bytes from 192.168.1.15: icmp_seq=8 ttl=64 time=0.560 ms
64 bytes from 192.168.1.15: icmp_seq=9 ttl=64 time=0.553 ms
64 bytes from 192.168.1.15: icmp_seq=10 ttl=64 time=0.554 ms
```

實作步驟 AP-Ubuntu(1/6)

- 先裝 dhcp3 server

```
#apt-get update          //更新伺服器清單
```

```
#apt-get install dhcp3-server //安裝  
dhcp3-server套件
```

- 再來編寫dhcpd.conf

```
#vim /etc/dhcp3/dhcpd.conf
```

```
ddns-update-style none;
```

```
option domain-name "genius.lee";
```

```
option domain-name-servers 120.107.179.10,  
168.95.1.1;    //設定DNS做正解反解功能，後面那個  
為中華電信
```

實作步驟 AP-Ubuntu(2/6)

```
default-lease-time 600;
max-lease-time 7200;
log-facility local7;
//DHCP-SERVER 自動分配的IP位址，private IP有三個 class，class A
10.0.0.0~10.255.255.255，class B 172.16.0.0~172.3.255.255，class C
192.168.0.0~192.168.255.255，皆可分配，我分配192.168.1.0這個區段
subnet 192.168.1.0 netmask 255.255.255.0{
range dynamic-bootp 192.168.1.60 192.168.1.70;           //我設為動態分
                                                            配IP，範圍
                                                            是.60~.70

option domain-name-servers 120.107.179.10, 168.95.1.1;
option routers 192.168.1.15;
option broadcast-address 192.168.1.255;
}
```

- 之後 :wq 存檔後離開

實作步驟 AP-Ubuntu(3/6)

- DNS也要檢查一下設定

```
#vim /etc/resolv.conf
```

```
nameserver 120.107.179.10
```

```
nameserver 120.107.179.50
```

```
nameserver 168.95.1.1
```

//此為中

華電信，我自己加的

實作步驟 AP-Ubuntu(4/6)

- 接著再來寫一個shell script

```
#vim pri_ap.sh          //檔名可自取，我有試在DHCP IP底下 & public IP  
底下，此為DHCP的private IP兩者皆成功  
#!/bin/bash  
wlanconfig ath0 destroy          //洗掉先前ath0的設定  
echo "destroy ath0"              //印出動作，可有可無，方便自己知道前面  
動作是否做了  
wlanconfig ath0 create wlandev wifi0 wlanmode ap          //啟動  
wifi裝置為AP功能  
echo "create ath0 as access point"  
iwconfig ath0 essid "unique15lee"          //設定essid  
echo "iwconfig ath0 essid as unique15lee"  
iwconfig ath0 channel 9          //設定channel  
echo "set ath0 channel as 9"  
ifconfig ath0 192.168.1.15 netmask 255.255.255.0 up          //內部  
IP設定
```

實作步驟 AP-Ubuntu(5/6)

```
echo "set ath0 ip as 192.168.1.15, netmask as 255.255.255.0"
ifconfig eth0 192.168.100.110 netmask 255.255.255.192 up
    //對外的IP設定
echo "set eth0 ip 192.168.100.110, netmask as 255.255.255.192"
route add default gw 192.168.100.126          //gateway IP
echo "set gateway as 192.168.100.126"
modprobe iptable_nat          //載入NAT模組
echo "load iptable_nat"
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE          //設定
IP偽裝規則
echo "set net rule"
echo 1 > /proc/sys/net/ipv4/ip_forward          //啟動封包轉送，
非常重要
/etc/init.d/dhcp3-server start          //啟動 dhcp3 server
echo "start DHCP"
#chmod 755 pri_ap.sh          //更改權限，才可執行
```

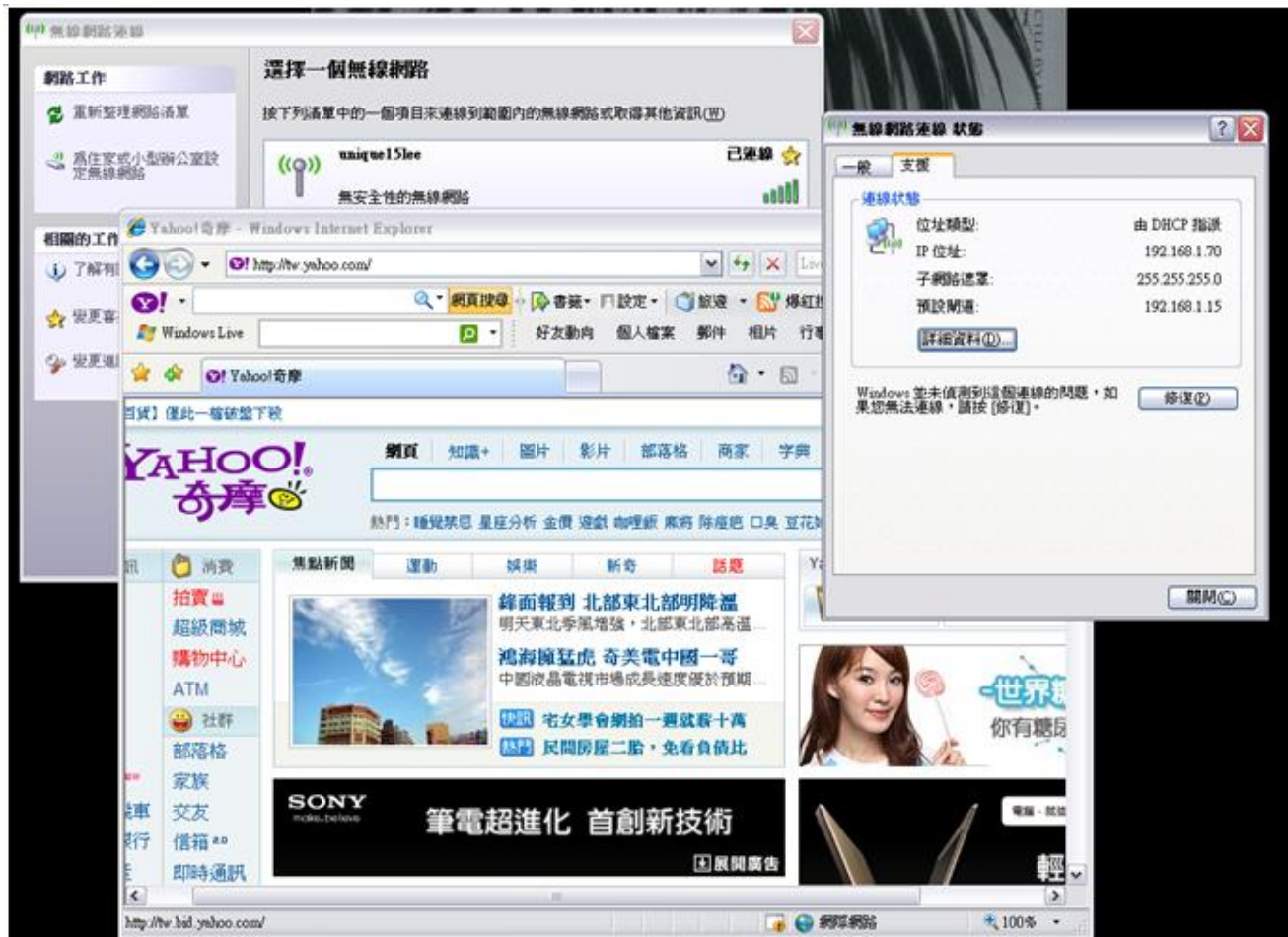
實作步驟 AP-Ubuntu(6/6)

- 執行
#sh pri_ap.sh
- 測試-利用PC or NB的無線網卡可搜尋到AP，
連上網即可。
- SSID **unique15lee**、**channel 9**

AP-Ubuntu 測試結果(1/2)



AP-Ubuntu 測試結果(2/2)



實作步驟 AP-FreeBSD(1/8)

- 灌好FreeBSD之後，先設passwd，剛灌好系統預設沒有密碼
- #adduser
//新增使用者，把這使用者加入wheel群組，才可用su- 切換至root權限方便用pietty遠登，因為FreeBSD不可用root遠登，要先用使用者登入再切換

實作步驟 AP-FreeBSD(2/8)

- 重編kernel，驅動D-Link DWL-G650無線網卡，順便也啟動PF防火牆

```
#cd /usr/src/sys/i386/conf
```

```
#vi GENERIC //kernel的設定檔要加入以下的設定
```

```
# Wireless support
```

```
device ath //Atheros IEEE 802.11 wireless network driver
```

```
device ath_hal //Atheros Hardware Access Layer
```

```
device ath_rate_sample //John Bicket's SampleRate control algorithm.
```

```
device wi
```

```
device wlan //802.11 support (Required)
```

```
device wlan_wep //WEP crypto support for 802.11 devices
```

```
device wlan_ccmp //AES-CCMP crypto support for 802.11 devices
```

```
device wlan_tkip //TKIP and Michael crypto support for 802.11 devices
```

```
device wlan_xauth //External authenticator support for 802.11 devices
```

```
device wlan_acl //MAC-based ACL support for 802.11 devices
```

實作步驟 AP-FreeBSD(3/8)

```
# Packet filter firewall support
device pf
device pflog
device pfsync
options ALTQ
options ALTQ_CBQ
```

```
#config GENERIC                                //開始編譯kernel
#cd ../compile/GENERIC
#make cleandepend; make depend all install
```

```
#vi /boot/loader.conf                            //開機就自動載入無線網路的function
wlan_wep_load="YES"
wlan_tkip_load="YES"
wlan_ccmp_load="YES"
wlan_xauth_load="YES"
wlan_acl_load="YES"
```

實作步驟 AP-FreeBSD(4/8)

```
#vi /etc/sysctl.conf          //開啟NAT功能讓封包可轉出去
net.inet.ip.forwarding=1
net.inet.ip.fastforwarding=1
```

```
#vi /etc/inetd.conf          //開啟 ftp 代理，這是 PF 比較特殊的一點，一定要開
                             啟這個 Intranet 的 ftp client 才能出得去
ftp-proxy      stream tcp    nowait root    /usr/libexec/ftp-proxy ftp-
proxy
```

```
#vi /etc/pf.conf             //加入 PF 防火牆之規則，測試用所以我防火牆規則全開
ext_if="bge0"
int_if="ath0"
nat on $ext_if from $int_if:network to any -> ($ext_if)
rdr on $int_if proto tcp from any to any port 21 -> 127.0.0.1 port 8021
pass in all
pass out all
```

實作步驟 AP-FreeBSD(5/8)

- 接著架設DHCP Server

```
#cd /usr/ports/net/isc-dhcp3-server
```

```
#make install clean
```

```
#cd /usr/local/etc
```

```
#cp dhcpd.conf.sample dhcpd.conf
```

例檔

//裝好DHCP Server之後會預設一個範
給你參考，再複製成dhcpd.conf

```
#true > dhcpd.conf //清掉裡面設定檔，我想自行寫入
```

```
#vi dhcpd.conf
```

```
default-lease-time 6000;
```

```
max-lease-time 7200;
```

```
option subnet-mask 255.255.255.0;
```

```
option domain-name-servers 120.107.179.10, 168.95.1.1; //DNS
```

```
option routers 192.168.1.254;
```

```
option broadcast-address 192.168.1.255;
```

```
option interface-mtu 1500;
```

```
option perform-mask-discovery on;
```

```
option mask-supplier on;
```

```
ddns-update-style none;
```

實作步驟 AP-FreeBSD(6/8)

Wireless LAN 自動分配IP

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
option routers 192.168.1.254;  
option broadcast-address 192.168.1.255;  
range 192.168.1.100 192.168.1.120;  
}
```

```
subnet 120.107.164.0 netmask 255.255.255.0 {  
    //用不到的介面也要定義介面卡資料  
}
```

```
#touch /var/db/dhcpd.leases
```

實作步驟 AP-FreeBSD(7/8)

```
#vi /etc/hosts          //DHCP由255.255.255.0做廣播
255.255.255.255 For-DHCP
#route add -host DHCP -interface ath0          //指定 Wireless LAN ath0
                                                提供 DHCP 服務
#/usr/local/etc/rc.d/isc-dhcpd.sh status        //查看 dhcp 的 pid
dhcpd is running as pid 520.
#vi /etc/rc.conf        //增加開機自動啟動的服務項目
ifconfig_bge0="inet 120.107.164.246 netmask 255.255.255.0"
defaultrouter="120.107.164.254"
usbd_enable="YES"
sshd_enable="YES"        //遠登要啟動
ifconfig_ath0="inet 192.168.1.254 netmask 255.255.255.0"
inetd_enable="YES"
pf_enable="YES"
pflog_enable="YES"
dhcpd_enable="YES"
#reboot or #init 6      //重開機讓改有的服務都跑起來
```

實作步驟 AP-FreeBSD(8/8)

- FreeBSD 的加密功能

```
#ifconfig ath0 ssid AS wepmode on wepkey 66666 mode 11g  
mediaopt hostap
```

採用 WEP 一般 ASCII 64 bit 加密模式 只要輸入 5 碼數字的密碼即可

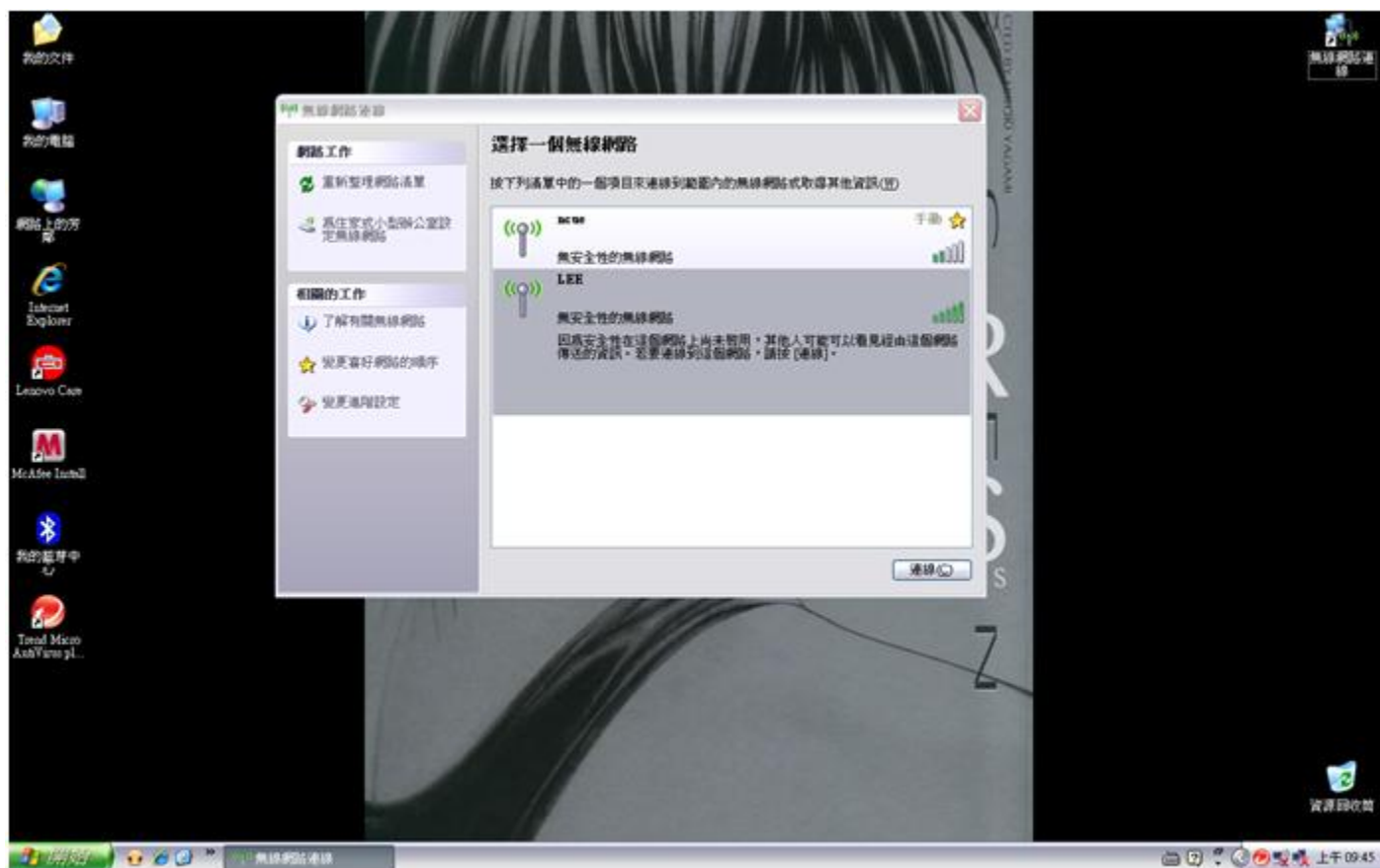
```
#ifconfig ath0 ssid AS wepmode on wepkey 0x0919634163  
mode 11g mediaopt hostap
```

採用 WEP 16 進位 64 bit 加密模式則要輸入 0x 再加 10 碼數字

mode 11g 也可改成 mode 11b(速度會不一樣，g是54Mbps，b是11Mbps)

AP-FreeBSD 測試結果(1/2)

- SSID **LEE channel 11**



AP-FreeBSD 測試結果(2/2)



參考文獻

- [1] Madwifi 官網 <http://madwifi.org/>
- [2] Ubuntu 官網 <http://www.ubuntu-tw.org/>
- [3] 鳥哥的Linux私房
<http://linux.vbird.org/>
- [4] 酷學園 <http://phorum.study-area.org/>
- [5] FreeBSD Handbook
http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html