Implementing 802.11b Wireless using FreeBSD on VIA mini-ITX Motherboards.

Dr Paul A. Van den Bergen
Centre for Advanced Internet Architectures. Technical Report 030403A
Swinburne University of Technology
Melbourne, Australia
pvandenbergen@swin.edu.au

Abstract-

The MAGIC project at CAIA required a wireless network testbed upon which to conduct Mobile IP experiments. This paper documents the development of wireless network nodes using VIA EPIA mini-ITX motherboards running FreeBSD.

Keywords- *802.11b, CAIA, FreeBSD, MAGIC, VIA EPIA mini-ITX, Wireless Network.*

I. Introduction.

The MAGIC (Mobile Applications and Global Internet Communications) project required a flexible wireless testbed to enable its planned research program. The following requirements for the testbed have been identified.

- **Flexible and reconfigurable.**
  Able to support a variety of activities related to Mobile IP and to change configuration with little administrative overhead.
- **Upgradeable.**
  Use of generic hardware and non-proprietary software.
- **Robust.**
  Stable, Secure, Fault tolerant, Reliable.
- **Compatible.**
  Not conflict with existing Swinburne Wireless Infrastructure.

Figure 1 shows the conceptual layout of the network for the MAGIC project. The intention is to achieve a low cost of ownership network testbed using generic hardware for the network infrastructure. The open source operating system, FreeBSD[1], comes with a very large number of ported applications and can be configured to perform any required network functions. For example, FreeBSD utilities exist to allow the machine to act as a firewall, router, DNS, and/or access point. Applications are available that allow service provision such as game servers, web servers, mail and news servers, ftp servers, authentication servers and radius servers. In addition, a large fraction of the applications developed for Linux will also run on FreeBSD with little or no reconfiguration.

A considerable range of hardware components that fulfill the requirements for a testbed computer are available commercially. The choice of particular products should not be taken as an explicit recommendation of that product. The hardware for the system was chosen on the basis of a combination of careful consideration of the system requirements (the desire to test the EIPA motherboards as wireless platforms under FreeBSD) and the use of components that “happened to be lying around” (i.e. the remainder of the components).

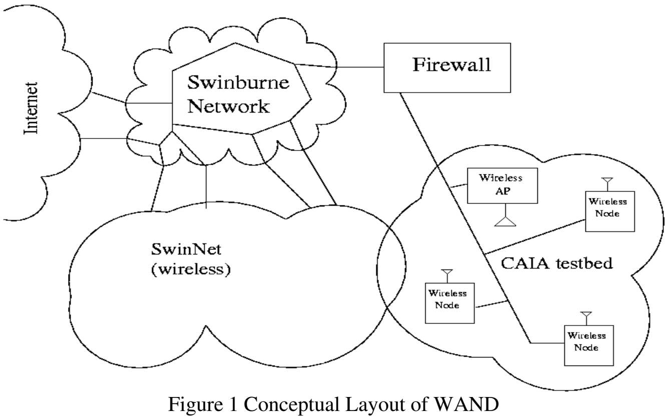

Figure 1 Conceptual Layout of WAND

A large amount of information about the peculiarities of hardware, troubleshooting and methods of setting up wireless on FreeBSD and Unix systems was found on a number of User Forums in the Unix and BSD[2][3], hardware[4][5][6] and wireless networking[7][8][9] domains, and especially the various wireless community groups in Australia and around the world[10][11][12][13][14][15][16].

II. Hardware.

The system consist of;

- VIA EPIA 533MHz mini-ITX motherboard[17] fitted with one 256MB SDRAM DIMM.
- 20 GB HDD IBM TravelStar (IC25N020ATCS04) Laptop HDD[18].
- Carry PCD-RP-101CE[19], a Ricoh R5C476 II based PCI-PCMCIA adapter.
- PCMCIA Cabletron CSIBD-AB 802.11 DS high rate wireless card (WiFi, 802.11b), which is re-badged as Enterasys RoamAbout, then re badged as SkyNetGlobal. This card is based on the Hermes chipset.

- Generic PSU.

A. *Hardware Specifics and Peculiarities.*

The VIA EPIA motherboard is an ATA PSU compliant motherboard based on the Embedded VIA Eden ESP 5000 processor. It uses the VIA VT8231 South bridge chip and the VIA Apollo PLE133 North bridge chip. It has a number of the standard all-in-one on board integrated peripherals: Trident Blade 3D graphics, 10/100 ethernet (VT6103), sound (VT1612A AC97 Codec), dual USB, serial, parallel, PS2 mouse and keyboard, a single PCI slot, 2 IDE Ports (ATA100) and 2 PC66/100/133 SDRAM DIMM slots (up to 1GB). In addition, it has S-video and RCA or S/P DIF TV/video out ports(VT1621, supporting 640x480, 800x600, PAL, NTSC). It's low power consumption (47W max, 60W peak) means it requires no fan for cooling, making it very quiet.

The TravelStar (IC25N020ATCS04) is a 20 GB, 2.5 inch laptop hard disk drive supporting ATA 6 at 4200 rpm with a 12 ms seek time. This was connected to the 3.5 inch IDE ribbon on the EPIA motherboard via a 2.5 inch to 3.5 inch adapter.

The PCMCIA cradle from Carry (PCD-RP-101CE) utilities the Ricoh R5C476 II PCMCIA – PCI bridge chipset. It supports all kinds of 16 bit cards compliant with PCMCIA 2.1/JEIDA 4.1 (SRAM card, Linear Flash card, ATA card, Modem card and LAN card) and 32 bit (CardBus) PC Cards, at voltages 3V/5V/12V. It supports IRQ steering, and is Intel 82365SL register compatible, ExCA compatible, hot swapping, and ATA disk interface support.

The Cabletron CSIBD-AB 802.11 DS high rate wireless card card has had a varied history. They are based on the same core as the Lucent Orinoco Silver, were re badged as Enterasys RoamAbout cards[20] and became available as SkyNetGlobal Cards. These became available as as secondhand cards[21]. They have an internal diversity antenna and an MCCardConnector external antenna jack. They are based on the Hermes chipset with a nominal equivalent isotropic radiated power (EIRP) of 30 W. The card supports data transfer rates of 11Mbps (receive sensitivity is -84dBm), 5.5Mbps (-87dBm), 2Mbps (-90dBm) and 1Mbps (93dBm).

Since the hardware for the Orinoco and Wavelan cards are identical and the firmware is flash upgradeable, drivers for the Orinoco cards can be used on the Enterasys cards. The card hardware is capable of 40 bit WEP (equivalent to Orinoco Bronze), 60 bit WEP (Orinoco Silver) and 128 bit WEP (Orinoco Gold). The Cabletron card was used as received without modification or upgrading. It supported 40 bit WEP. However, due to the inherent insecurity of WEP it shall not be used in this case. Methods are available to upgrade the WEP capability to 128 bit[22].

The Hermes chipset supports both IBSS and BSS mode. However, it does not support the internal bridging required to allow the card to act as a hardware Access Point (server), only as a client to a wireless Access Point[23].

The use of wireless cards based on the Prism (Intersil) chipset[24] in future is recommended as these support BSS (Infrastructure or Access Point) modes as both client and server, as well as IBSS (ad hoc) mode[23].

B. *Setting Up the Wireless Card.*

This section outlines the installation and configuration of the hardware under FreeBSD on the EPIA motherboard.

A number of distinct steps are required to correctly install the wireless card. The kernel modules need to be loaded. The wireless card access needs to be activated and the card configured so the Operating system correctly identifies it. Finally, the card must be configured for the type of access required and the wireless network interface correctly configured.

Root access is assumed.

Wireless Card Activation.

Next, edit **/etc/rc.conf** and add the line;

```
pccard_enable="YES"
```

Reboot the machine and the appropriate pccard should be detected when you push it in the slot. Of course, at this point it is not configured to any network settings, but the existence of a card and the type of card in the slot should be recognised. For the hardware configuration chosen for the test bed, the card type was not recognised and the following error message was displayed.

```
pccard: card inserted, slot 0

pccardd[59]: No card in database for
"(null)"("(null)")
```

After some exploration it was discovered that this was due to a choice of memory range for the wireless card. The following syntax can be used to explore the wireless card memory options. The details can be found in the man page for **pccardc** and **pccardd**.

From the command line enter;

```
# pccardc pccardmem dN000
```

where 'N' is 0, 4, 8 or c.

To permanently set this value on restart, add the following line to **/etc/rc.conf**;

```
pccard_mem="0xdN000"
```

In the specific case used here the correct value for the string is "OxD8000" (i.e. 'N' = 8).

More settings and detail related to pccard and wireless card settings may be found in the file **/etc/defaults/pccard.conf**.

If the above configuration still does not produce a working card or the error message is different to above, then there are other things that can be attempted, such as changing the IRQ default values. For example, one article[25] lists the following;

> If you're using a PCI adapter and you run into trouble (such as FreeBSD hanging on boot, not recognizing your wireless card, or the card simply

not working), try adding the following lines to **/boot/loader.conf**.

```
hw.pcic.intr_path="1"
```

```
hw.pcic.irq="0"
```

It is not mentioned in the article what the effect of this behavior is, but a look at the man page for **pcic** reveals that the first line sets the PCI **pcic** devices to routing interrupts to the ISA subsystem. The second line, equivalent to the default setting, is required to explicitly prevent PCI polling.

Setting Up the Wireless Interface.

To configure the wireless card from the command line, use the following command syntax.

To set the cards channel, execute the following command;

```
# wicontrol -i wi0 -f N
```

where N is the channel number (1 – 14 (2.412 to 2.484 GHz) in Australia). Note: two wireless cards **must** be on the same channel in order to communicate. Channels adjacent to a selected channel will experience interference up to 3 channels away. In the range 1-14, channels 1, 5, 9, and 13 (or 2, 6, 10 and 14) can be selected simultaneously in the same geographic area without significant interference.

To enable WEP, execute the following command;

```
# wicontrol -i wi0 -e 1
```

```
# wicontrol -i wi0 -k "NNN"
```

where NNN is the key value.

Infrastructure Mode.

To set the card in BSS (infrastructure) mode, execute the following command;

```
# wicontrol -i wi0 -p 1
```

To set the cards ssid (network name) , execute the following command;

```
# ifconfig wi0 ssid "network name"
```

or

```
# wicontrol -i wi0 -n "network name"
```

In this context the network name needs to be the same as the AP you wish to connect to. An empty string should allow connection to the first AP encountered. The string "ANY" should also work.

Adhoc Mode.

To set the card in IBSS (ad hoc) mode, execute the following command;

```
# wicontrol -i wi0 -p 3
```

To set the cards ssid (network name) , execute the following command;

```
# ifconfig wi0 ssid "network name"
```

or

```
# wicontrol -i wi0 -n "network name"
```

In this context, network name is the ad hoc network ' group' you wish to join.

Finally, configure the interface;

```
# ifconfig wi0 inet a.b.c.d
                        netmask w.x.y.z
```

```
# ifconfig wi0 up
```

To have the interface come up every time you start the machine or (re)insert the wireless card, add the following lines to **/etc/rc.conf**.

```
pccard_ifconfig="inet a.b.c.d
                       netmask w.x.y.z"
```

III. Testing the Wireless Node

Once the wireless functionality was setup according to the above instructions, the connectivity and correct function of the wireless networking was tested for the BSS (access point client) mode. The functionality of the IBSS (ad hoc) mode could not be tested as a second ad hoc wireless node was not available. The connectivity and functionality of the wireless configuration was tested using two different methods.

Firstly, **dstumbler** (part of the **bsdairtools**[26] package) was installed. This program is a wireless packet sniffer and is used to discover adjacent wireless installations. Once installed, the Swinburne wireless infrastructure could be seen, as could a local Linksys WAP11 Access Point[27].

Secondly, the Linksys AP and the wireless node were configured for connection (same network name, channel and IP network). Connection from the PC connected to the AP to the wireless enabled EPIA machine using **ssh** and a file upload confirmed the connectivity of the two machines via the wireless infrastructure.

IV.Non-wireless Hardware Issues

Problems were encountered with the TravelStar laptop HDD. On startup the lowest available transfer mode PIO2 was selected instead of the highest available mode UDMA33. The workaround for this was to use a script file to correctly set the speed, using **atacontrol**, at the conclusion of the boot sequence.

Limitations with the VIA ethernet port (vr0) were encountered.

Dramatic reductions in data rate occasionally occurred when autonegotiation of the connection speed to our 100Mbit/sec Ethernet switch was enabled. This was corrected by explicitly setting the interface to 100MBits/sec.

```
# ifconfig vr0 media 100baseTX
                   mediaopt full-duplex
```

802.1Q vlan tagged packets require an additional 4 bytes, which the VIA ethernet card does not seem to be able to handle correctly. A workaround for this has not been attempted. Where vlan functionality was required an alternative ethernet card in the (single) PCI slot was used.

A patch for the vr driver was applied to correct freezing of the interface after multiple **ifconfig**s[28].

V.CONCLUSIONS

FreeBSD 4.7 was successfully installed on a VIA EPIA mini-ITX motherboard with wireless peripherals installed. The procedure for correctly setting up the wireless hardware was established.

The wireless hardware consisted of a Carry PCMCIA to PCI bridge based on the Ricoh chipset and a Cabletron CSIBD-AB 802.11 DS high rate wireless card based on the Hermes chipset.

The Ricoh-based PCMCIA to PCI cradle required the following systems configuration modifications to enable it. The following lines were added to **`/etc/rc.conf`**;

```
pccard_enable=”YES”

pccard_mem=”0xd8000”
```

The Hermes based wireless cards are limited in their hard-wired functionality and do not support BSS (Access Point) mode as a server, only as a client. In this regard, the use of Intersil (Prism) based wireless cards is recommended for future hardware installations.

[1] http://www.freebsd.org/

[2] http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/wireless.html

[3] http://www.live.com/wireless/unix-base-station.html

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

[5] http://www.tomshardware.com/

[6] http://www.onlamp.com/lpt/a/778

[7] http://www.linuxsecurity.com/articles/network_security_article-4781.html

[8] http://darkminds.net/wlan/freebsd/freebsd.php

[9] http://www.epinions.com/content_2571411588

[10]http://wireless.org.au/

[11]http://www.australianwireless.org/

[12]http://www.sydneywireless.com/links.php?op=viewlink&cid=3

[13]http://www.freenetworks.org/

[14]http://nz.wireless.org.au/

[15]http://seattlewireless.net/

[16]http://www.bawug.org/

[17]http://www.viavpsd.com/product/epia_mini_itx_spec.jsp?motherboardId=21

[18]http://www.hgst.com/hdd/support/table.htm

[19]http://www.carry.com.tw/products/PCI/101cs.htm

[20]http://www.enterasys.com/products/wireless/

[21]http://melbourne.wireless.org.au/wiki/?EnterasysCards

[22]http://www.paris-sansfil.net/index.php/UpgradeSilverToGoldWindowsEn

[23]http://www.freebsd.org/releases/4.7R/hardware-i386.html

[24]http://www.intersil.com/design/prism/index.asp

[25]http://www.samag.com/documents/s=7121/sam0205a/sam0205a.htm

[26]http://www.dachb0den.com/projects/bsd-airtools.html

[27]http://www.linksys.com/products/product.asp?grid=33&scid=35&prid=501

[28]Pers Comm., Mike Silbersack, silby@silby.com, Thu., 14 Nov., 2002.