IEEE 802.11b WLAN AP를 위한 리눅스의 이식

박현문, 장영민, 정옥조**
국민대학교 전자정보통신공학부*, 한국전자 통신 연구원**
e-mail: kimagu@empas.com, yjang@mail.kookmin.ac.kr, okjo@*etri.re.kr*

Porting linux for IEEE802.11b WLAN AP

Hyun-Moon Park, Yeong-Min Jung ,Okjo Jeong**

Abstract

최근 IEEE에서 WLAN을 표준으로 하는 802.11g가 확정되면서 이를 지원하는 AP가 출시되었다. AP에서 사용되는 OS는 POSIX를 기반으로 된 Linux 같은 OS를 통하여 구현되고 있다. 이 논문에서는 가장 범용으로 사용되고 있는 Rad Hat을 기반으로 한 embeded linux를 만들어본다. 이와 함께 embeded linux를 가지고서 Network기능을 추가하고, AP 하드웨어 드라이버 및 AP 모듈을 올려서 구연한다. 여기에 추가적으로 DHCP나 Web UI기능 등의 부가 기능을 추가하여 AP를 구현하는데 목적이 있다.

Ⅰ.서 론

최근에 무선 네트워크가 시장에서 핫이슈가 되면서 여러 가지의 무선네트워크 관련한 제품이 쏟아져 나오고 있다. 더욱이 서비스 영역도 근거리, 원거리 및 전역까지 확장되고 있다. 이러한 무선네트워크 중에서도 최근 각광을 받는 것은 Wireless 802.11X을 표준으로 하는 Wireless-LAN이다. 현재 PDA이나 GPRS(General Packet Radio System)모바일 폰에서의 서비스가 이루어지고 있고, 최근에 WLAN Hot-zone들이 생기면서, Notebook에서 이러한 서비스가 이루어지고 있다. 이러한 서비스를 하기위해서 유·무선의 결합된 네트워크 망을 구축해야 한다. 유선에서 Router나 Hub와 연결이 되면서, 무선으로써의 Router나 Hub의 역할을 해주는 필요한데. 이것이 AP(Access-Point)이다.

AP는 네트워크 및 인터넷이 가능한 유선망을 가지고서 무선 서비스를 하는 하나의 Hub의 개념을 가지고 있으면서도, DHCP나 Routing 같은 기능을 지니고 있기도 하다.

AP는 OSI 계층구조의 물리층, 링크 계층, 그리고 네트워크 계층상에서 동작을 하는데, 패킷의 전송경로를 설정하고 해제하는 라우팅 기능을 한다. 이러한 AP의 기능은 기업적인 용도에서는 LAN망에서의 서비스 뿐 만 아니라 기업외의 일반 공간에서의 무선 서비스의 연동 및 사용자간의 이동성 보장(IAPP) 및 사용자 보안(WPA) 등의 기능과 함께, 일반 가정에서 사용되는 가정용 기기를 인터넷과 부합되어 원격으로 감시하고 제어 또한 가능한 영역으로 확장되어 이야기 되고 있다. 홈 네트워크에서 사용되는 AP는 저 전력에 다양한 기능을 요구하지만, 사용되는 마이크로프로세서의 성능은 제한됨으로 OS의 경량화를 필히 요구가 되어 진다. 이러한 것을 구현하기 위해서는 데스크탑 보드가 아닌 embeded system kit이나 uclinux 같은 embeded linux로 AP를 개발한다. 이러한 이유는 kernel이 대부분 선점형 구조를 가지고 있고, OS자체가 가벼우며, 실시간성을 목적으로 개발되었기 때문에 구조적으로 간단하다.

일반 리눅스를 사용할 경우에는 kernel 및 OS의 크기가 크고, kernel이 비선점형 구조로 RTOS(Real Time Operate system)의 부적한 성격을 가지고 있으며 구조가 복잡하기 때문에 처음 사용하는 사람의 경우에는 구성자체에서도 시간이 많이 요구된다.

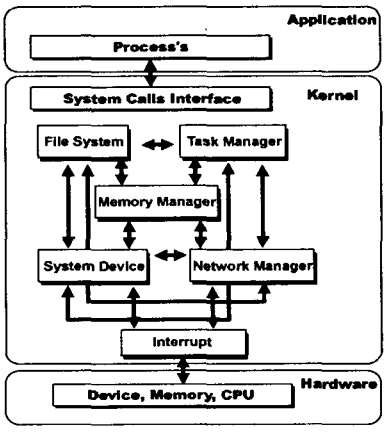
그러나 논문에서는 일반 Rad Hat기반의 linux로 경량화된 OS 개발함으로써, 차후 기술구현의 용의함에 대한 몇 가지 장점을 가지게 된다. 가장 많은 시스템 디바이스 드라이버를 가지고 있으며, Open OS와 Open sauce 라는 강점을 지니고 있다. 더욱이 디바이스 드라이버끼리의 호환성도 지니고 있어서, 완전히 새로운 하드웨어가 아니라면, 디바이스 드라이버의 이식도 가능하다. 마지막으로 자유로운 환경에 대한 구축으로 약간만 변경을 하면, 차후 어떤 하드웨어에도 구축이 가능하다는 장점을 지니고 있다.

이번 논문에서는 PC의 구축환경에서 교차컴파일러를 구축하고, Taget AP(데스크탑 PC)에 이식할 리눅스(RadHat)의 크기를 줄여 최소화 및 kernel을 최적화 시킨다. 또한 이를 통하여, 하드웨어에 최적화된 embeded 형태의 Linux로 AP를 구축 동작의 편의성을 제공하는 것이 목적이다. 더 나가서는 이러한 AP가 완전히 구현이 되었을 경우 css, c, html의 프로그램을 통해서 간단한 application 구현을 목적으로 하고 있다.

Ⅱ. Red Hat Linux Porting

실시간 운영체제(RTOS)의 대표적인 예로는 pSOS, VxWorks, RT-Linux, uclinux, Linu@, eCOS, Tynx, Blue Cat 등이 존재 한다. 이러한 임베디드 리눅스들 중에 대부분이 아래에서 설명하는 절차에 kernel에 대한 재구성 및 X-windows 서비스, 컴파일러의 환경구축, application을 강화를 통해서 개발에 대한 편의를 제공을 하고 있다.

AP의 하드웨어를 시스템에 이식하기 위해서는 하드웨어의 정확한 정보의 습득과 드라이버의 명확한 지식이 있어야 한다. 사용할 linux가 지원이 가능한 부분과 불가능한 부분에 대한 지식과 함께 하드웨어에 대한 지식 및 사전적 kerenl에 지원에 대한 지식도 분명해야 하드웨어의 porting이 쉽게 가능해진다.

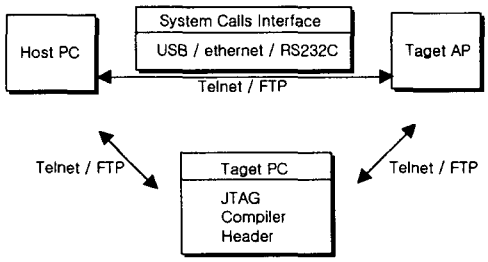


[그림 1] Kernel 구조

교차 컴파일러의 환경을 구축하기 위해서는 위와 같이 PC를 2~3대를 사용하고 Taget Hardware를 구성한다. 또한 완전히 새로운 임베디드 하드웨어의 경우는 JTAG이 별도로 분리되어 구성된다.

그러나 여기서는 개발자 환경이 아닌 PC의 구현 환경이기 때문에, 개발 PC, Taget PC, Taget AP로만 간단하게 구현이 된 것이다. 먼저 이루어지는 embeded 리눅스를 구축하기 위해서는 kernel의 환경설정을 통하여 모듈로 설정을 한다.

환경 설정은 3가지 종류인 make config, make *menuconfig*, make *xconfig*로 할 수 있지만, 가장 안전한 kernel의 환경 설정은 make config 임으로, kernel의 환경설정을 한다.[1][2]



[그림 2] 교차 컴파일러 환경

이러한 것을 차례로 나열을 한다면,

① make config
② make dep
③ make clean
④ make bzlmage or make zlmage and test bzdisk
⑤ make modules
⑥ /usr/scr/linux/kerel_/boot/
⑦ make install
⑧ make modules_install
⑨ depmode-a 2.4.20
⑩ /dev 필요없는 부분 및 /etc 작업 연동성 조사
⑪ Taget AP에 이식 RAMDISK mount
⑫ kernel 복사
⑬ umount
⑭ booting
⑮ network 설정

여기서 가장 중요한 것은 PCMCIA의 모듈을 올려야 한다. 칩은 802.11b을 대부분 호환되는 것으로는 RICOH R5C475II라는 칩이 있으며, pcmcia-cs-3.1 드라이버를 올리고, 무선랜 카드의 모듈을 올려주면, 제품을 인식하게 된다. 이러한 작업을 통해서, linux의 시스템에 자원을 최적화 한다. 작업이 끝나는 순간 두 대의 PC는 동일한 AP가 구성이 되게 된다. 단지 두 대의 차이는 하나는 compiler 및 시

스템 환경이 완전히 구축하여 개발 환경이 가능한 것이고 Taget AP는 Taget PC에서 내부 구조상 문제가 없을 것이라는 검증된 필요한 영역을 구현되는 것을 이동 시켜서, 구현을 한다.

⑪의 경우에는 RAMDISK 즉 flash Memory나 ROM을 통해서 구성을 할 경우에는 사용하며, 이때는 블록의 장치로 sf0, sf1, sf2로 나누어져서 사용되어 진다. [8]

그리고 일반적인 장치로서는 /dev/consolee 및 /dev/tty 장치파일을 접근하여 해결한다. 하드디스크에 구축을 할 때는 디바이스 드라이버를 직접 지정해주는 식으로 설정을 해주는 것이 좋다. 여기서는 데스크탑 환경에서 구현을 하기 때문에 일반적인 embeded 환경에서 구축하는 방식을 변형된 형태로 구현되어있다.

Ⅲ. AP 의 구현

기본적으로 시스템을 구성하기 위해서는 Taget AP에 embeded를 만들면서 설정을 하는 경우가 일반적인 경우지만, 특수한 경우로 인하여, 새로운 하드웨어로 교체를 하거나 업그레이드를 하기위해서 시스템이 변경이 되었을 경우에, 모듈별로 설치하여 새롭게 컴파일을 할 수도 있다. 이 경우에는 이것을 실험할 Taget PC에 구성을 하고, 이것을 컴파일하여, ethernet을 통해서 Taget board 이식을 해야 한다. 이유는 embeded system의 환경에서 system의 power가 문제가 되기 때문에, 일반적인 구성에서는 Taget PC에서 일을 구성하기가 어려운 문제가 있지만 여기서는 충분한 System Power가 제공되기 때문에, Taget PC에 직접 컴파일 사용도 가능하다는 장점을 지니고 있다.

AP의 모드의 설정은 Ad-Hoc과 Infrastructure 두가지가 존재 한다. Ad-Hoc의 경우 AP가 없이 각 PC끼리 서로 무선랜 카드를 통해 네트워크에 연결하고자 할 경우에 사용하며, 1:1방식이며 Peer to Peer 방식이다. Infrastructure방식은 서버를 운영하면서 대규모의 PC 환경에 맞는 방식으로 KT의 네스팟(NESPOT)서비스를 비롯한 공중망서비스 역시도 동일한 방식이며 거의 모든 무선랜 환경이 이와 유사한 방식이다. Infrastructure network 구성은 적어도 2개 이상의 AP 연결은 무선 클라이언트들의 무선 유효범위에서 자유롭게 움직이는 동안에도 계속하여 이용 할 수 있도록 무선 사용자들의 로밍을 제공하는 역할도 이루어진다. 이런 확장된 서비스를 가능하도록 하기 위해서는 무선 PC Card, AP 등이 동일한 Network name(SSID)으로 설정되어진다.[5] 장치드라이버를 위에서 인식하면 좋겠지만, 그렇지 못하는 경우가 대부분이다. 결국은 크로스 컴파일러 환경에 구축되고, 여기서는 그것을 응용한 단계이기 때문에 그렇게까지 번거롭게 적용을 하지 않아도 되고, 직접 모듈을 올려서 컴파일을 해도 문제가 되지 않는다. 우선적으로는 무선 랜에 관련된 하드웨어를 이식하기 위해서는 드라이버를 설치한 후에 /etc/pcmcia/hostap.conf에서의 추가나 변경이 이루어질 수 있다. [7][9]

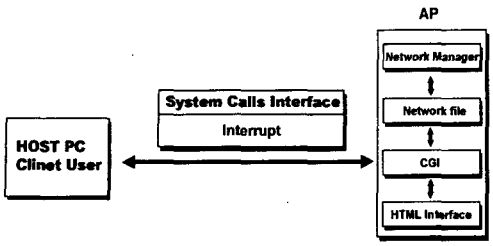
```
[root@wireless /]# lsmod
Module                  Size  Used by    Not tainted
parport_pc             17508   1  (autoclean)
lp                      8588   0  (autoclean)
parport                33952   1  (autoclean) [parport_pc lp]
autofs                 12148   0  (autoclean) (unused)
orinoco_cs              5488   1
orinoco                33688   0  [orinoco_cs]
hermes                  7492   0  [orinoco_cs orinoco]
ds                      8136   1  [orinoco_cs]
yenta_socket           12864   1
pcmcia_core            53152   0  [orinoco_cs ds yenta_socket]
8139too                17088   1
mii                     3728   0  [8139too]
e100                   56644   1
ipt_REJECT              3672   6  (autoclean)
iptable_filter          2316   1  (autoclean)
ip_tables              14488   2  [ipt_REJECT iptable_filter]
keybdev                 2720   0  (unused)
mousedev                5284   1
hid                    28772   0  (unused)
input                   5632   0  [keybdev mousedev hid]
usb-uhci               24652   0  (unused)
usbcore                73088   1  [hid usb-uhci]
ext3                   64704   2
jbd                    47828   2  [ext3]
```

[그림 3] AP에 사용된 하드웨어 모듈

일반적으로의 환경설정에 hostap 구성을 하는 경우 무선랜의 장치 이름은 wlan0으로 설정이 된다. eth0이나 ifcfg-eth0 처음에 잡히는 경우가 대부분인데 이것을 wlan0으로 설정을 해주는 것이 가장 우선되어야 한다. /etc/sysconfig/networking/devices/있는 설정파일 수정을 하고 마지막으로 /etc/modules.conf에 대한 수정을 한다. AP의 Routing 프로젝트 GPL중에 하나인 hostap-driver-0.1.1.tar.gz을 시스템에 맞게 컴파일을 한다. host에 대한 루프백 인터페이스를 설정하고, Rout에 대한 활성화와 함께 Routing ip-masquerade을 지정을 하여 Routing 서비스를 하도록 한다. 마지막으로는 /etc/rc.d/rc.local을 수정하여 제시작시에도 서비스를 제공하기 위해서 설정을 한다.[1][2][3][4]

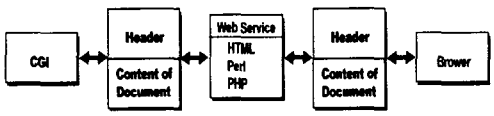
Ⅳ. AP의 인터페이스 구현

AP에 대한 보안된 인터페이스 구현은 크로스 컴파일러 환경이 완전히 구축이 된다는 전제에서 이루어질 수가 있다. 단순하게 C를 통해서 환경을 구축할 수도 있지만, 이러한 경우에는 사용자가 접속해서 하나하나의 네트워크 설정을 보기에는 어려움이 있다. 이러한 이유로 보다 쉬우면서도 사용하기 편한 CGI를 통한 html이 우세하다. 여기서는 C와 CGI, PHP를 통해서 간단한 Interactive 웹 페이지를 구축 하도록 한다.



[그림 4] 시스템 구성

일반적인 form의 요소를 처리를 하고, 이에 대한 php을 통한 IP의 확인 및 변경, 각 연동된 shell에 대한 access 관리를 만들어 준다. 이러한 것을 만들게 되면, 인터페이스 처리과정에서 툴을 만들어야 하는 경우가 있는데, 대부분이 Network Manager를 수정하는 것보다는 system shell을 추가하거나 변경을 하여서, 간단히 처리를 하는 것이 용의하다.



[그림 5] UI인터페이스 동작 과정

V. 결론 및 추후 연구본론

본 논문에서는 일반 데스크탑 PC를 가지고 교차 컴파일러 환경을 구축하여, 일반 리눅스를 최적화된 embeded linux를 porting하는 방법을 보였다. 또한 리눅스를 탑재한 AP의 실제 리눅스 환경에서 설정이나 구현에 대해 구현하였다. 이러한 리눅스를 porting은 상용이나 GPL 임베디드 리눅스 비해서는 부족하지만 앞으로 각광을 받은 폭넓은 범위에서의 응용과 함께, AP에서 Ad-Hoc 같은 작은 Home Network 보다는 Infrastructure방식에서의 LAN 그룹을 테스트를 해서 기본적인 AP의 기능 이해하고, 기능을 점차 추가 시키는 것을 기본으로 한다. 여기서의 AP 구현은 802.11b의 기존의 문제점을 다시 확인하고, 이를 계선하기 위한 초석이 된다고 생각한다. 앞으로의 연구에서는 AP간의 이동성에 관련된 연구(IAPP: Inter-Access Point Protocol) 와 함께 서비스질의 향상 (QoS: Quality of Service)의 연구와 구현에 중점을 두고, 좀더 세분화된 연구를 통하여 앞으로 네트워크 이슈에 대해 다루도록 하겠다.

참고문헌

[1] Alessandro Rubini "Linux Device Driver", O'REILLY,1998

[2] Remy Card, Eric Dumas, "the Linux Kernel Book", Wiley 1998

[3] Andrew Sun, "Using & Manasing PPP" O'REILLY, 1999

[4] IEEE Std 802.11b. "Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifiation". 2000

[5] Medu Cast "802.11 Wireless Networks" O'REILLY, 2001

[6] http://prism2.unixguru.raleigh.nc.us/v15/ v15.html

[7] Korean Linux Documentation Project, Home Page, http://kldp.org/

[8] Linux Routing Documentation Project, http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html

[9] Red Hat, Inc, Home page, http://www.redhat.com/