

## IBM PowerPC 405GP를 이용한 Wireless LAN Access Point 개발에 관한 연구

김도규\*

### 요 약

본 논문에서는 IBM의 고성능 임베디드(embedded) 프로세서인 405GP와 Linux 2.4.21를 이용하여 5.2GHz 대역에서 최대 54Mbps의 대역폭을 제공할 수 있는 IEEE 802.11a 기반의 AP (Access point) 개발에 관하여 연구하였다. AP의 하드웨어는 PowerPC기반의 IBM 405GP 프로세서를 기반으로 설계하였고 AP 소프트웨어는 405GP를 위한 PowerPC 부트로더, open source인 최근 Linux 커널 2.4.21을 사용하여 구현하였다. 또한 시스템 패키징은 최신 Linux 커널이 지원하는 MTD 기술과 JFFS2 플래쉬 파일시스템을 이용하여 최적으로 구현하였다.

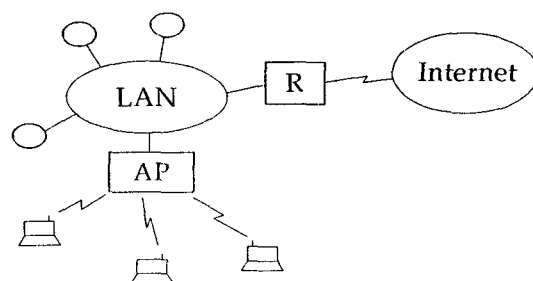
### 1. 서론

최근 노트북 컴퓨터, PDA 등 무선 디바이스의 보급이 증가함에 따라 IEEE 802.11b WLAN (wireless local area network)에 대한 수요가 점차 확대되고 있는 실정이다. 특히 WLAN은 케이블의 설치가 필요 없고 무선 단말의 이동이 용이하기 때문에, 기존 스위칭 허브, 라우터 등으로 구성된 유선 기반의 LAN을 보완하여 무선 인터넷 서비스의 활성화를 앞당기고 있다.

2.4GHz 대역에서 최대 11Mbps를 지원하는 IEEE 802.11b WLAN을 유선 기반의 LAN에서 일반적으로 사용되는 있는 100Mbps의 Fast Ethernet과 비교해 볼 때, 현재의 WLAN은 멀티미디어 서비스를 이용하는 데 있어서 속도의 장애가 있다. 그러나 최근 IEEE 802.11b를 개선한 5.2GHz 대역에서 최대 54Mbps의 대역폭을 제공하는 IEEE 802.11a 기반의 WLAN 제품의

개발이 증가하고 있는 실정이다.

WLAN 서비스를 위한 핵심 장치는 AP(access point)로서 무선 단말에서 보내는 패킷을 수용하여 유선 LAN을 경유하여 인터넷으로 전달하는 장치이다 (그림 1). 본 논문에서는 IBM의 고성능 임베디드(embedded) 프로세서인 405GP, PPC-Boot, open source인 최근 Linux 커널 2.4.21을 이용한 IEEE 802.11a 기반의 AP 개발에 관하여 연구하였다. 본 논문의 구성은 서론에 이어, II장에서는 PowerPC 기반의 IBM 405GP를 이용한 AP의 하드웨어 구조에 대하여 기술하고, III장에



(그림 1) Access point

\* 성결대학교 정보통신공학부



### 2.3. I/O

#### 2.3.1. PCI-to-CARDBus controller

AP 보드의 무선 인터페이스 카드는 Telecis 사의 CARDBus를 이용하였다. PCI-to- CARDBus 브릿지인 PCI1410A는 405GP와 CARDBus 소켓 간의 인터페이스를 하는 부분으로 405GP내부의 PCI 인터페이스 로직과 CARDBus 소켓을 직접 연결한다. 무선 인터페이스 카드가 삽입되면, 405GP 내부에 인터럽트가 발생하고 리눅스의 카드서비스가 동작된다.

#### 2.3.2. 10/100Mbps Ethernet Interface부

AP 보드에는 10/100Mbps Ethernet transceiver 가 있고, Ethernet transceiver는 405GP와 MII 인터페이스를 통해 연결된다. Ethernet transceiver는 10Mbps와 100Mbps 동작 모드를 auto-negotiation 할 수 있는 기능이 있으며, 동작 가능한 모드에는 10Mbps Half duplex와 100Mbps Half duplex, 100Mbps Full duplex 모드가 있다.

#### 2.3.3. RS-232 Interface부

AP 보드에는 하나의 RS-232 connector가 있으며 이 connector는 monitor 및 debug port로 사용된다.

#### 2.3.4. 전원 발생부

AP 보드 내부의 사용전원은 2.5V, 3.3V, 5V 인데 regulator를 사용하여 각각의 전원을 만들어준다. Regulator의 입력전원은 DC Jack으로부터 입력되는 전원 및 Banana Jack을 통한 DC Power supply로부터 오는 전원 등 2가지 전원 공급원이 있는데 사용 환경에 따라 각각의 입력 전원을 선택하게 되어있다.

#### 2.3.5. 클럭 발생부

AP 내부에서 사용되는 clock은 오실레이터로부터 발생되는 33.33MHz의 기준 clock을 405GP에서 multiplication하여 200MHz로 동작하고 AP 내의 SDRAM으로 공급된다. 또한 10/100Mbps Ethernet transceiver칩에는 crystal로부터 25MHz clock이 공급된다.

<Table 1> memory map

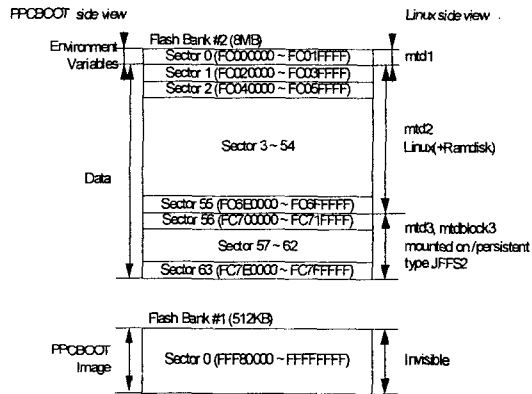
Function	Start Address	End Address	Size	Comment
SDRAM Bank 0	0x0000_0000	0x03FF_FFFF	64MB	SDRAM Bank 0
SDRAM Bank 1	0x0400_0000	0x07FF_FFFF	64MB	SDRAM Bank 1
PCI Bridge	0x8000_0000	0xEF5F_FFFF		PCI in the PPC405GP
UART0	0xEF60_0300	0xEF60_0307		PPC405G Internal Register Map - Fixed in the Chip
UART1	0xEF60_0400	0xEF60_0407		
IC	0xEF60_0500	0xEF60_0510		IC registers in the PPC405GP
GPB Arbiter	0xEF60_0600	0xEF60_0601		GPB Arbiter
GPIO	0xEF60_0700	0xEF60_077F		GPIO registers in the PPC405GP
EMAC Register	0xEF60_0800	0xEF60_086F		EMAC registers in the PPC405GP
Flash Dis (E28F128J3)	0xFC00_0000	0xFCFF_FFFF	16MB	Peripheral Bank 2 16MB Width
Boot Flash (AM28LV04)	0xFFFF_0000	0xFFFF_FFFF	512KB	Peripheral Bank 0 (Boot Flash) 512KB Width

## III. IEEE 802.11a WLAN AP의 소프트웨어 구조

Embedded 시스템을 위한 소프트웨어의 개발은 일반적으로 x86기반의 시스템에서 이루어진다. 그리고 target이 되는 embedded 시스템은 x86기반이 아닌 다른 CPU를 사용하는 경우가 일반적이다. 따라서 먼저 소프트웨어 개발 환경의 준비가 선행되어야 한다. AP의 경우 IBM의 PowerPC 405GP를 사용하므로 PPC(PowerPC)용 개발환경이 필요하다. 이를 위하여 montavista의 HHL(hard hat linux) 2.0을 이용하여 CDK(Cross Development Kit) 환경을 구성하였다.

AP의 소프트웨어는 크게 boot loader, Linux 커널, root 파일시스템으로 구성되어 있으며, 특히 용량이 많은 root FS은 플래쉬 메모리에 저

장될 수 있도록 최적화 하였다. AP 보드는 boot loader와 zjsjf 및 파일시스템을 위하여 2개의 flash memory bank를 가지고 있다. Bank#1은 512K의 크기로 PPCBOOT 이미지를 저장하고 있어서 power-on의 경우에 boot loader로서 동작한다. Bank#2는 8M 크기로 Intel의 StrataFlash E28F640J3A120로서 PPCBOOT의 환경변수, Linux 커널, Root FS이 저장되어 있다. AP 보드의 flash memory map은 (그림 3)과 같다 (Table 2).



(그림 3) 405GP AP 보드의 flash memory map

405GP 보드를 초기화 한 후 (ii) 리눅스 커널을 메모리에 로드하고 압축을 해제한 후 제어를 커널에 넘겨준다. AP보드의 부트로더는 PPCBoot-1.2.1을 이용하였다. PPCBoot가 제공하는 기능을 정리하면 아래와 같다.

- (1) PowerPC 405 CP CPU의 초기화
- (2) 보드정보 관리
- (3) flash 메모리를 포함한 메모리 관리
- (4) 환경변수의 관리
- (5) 개발호스트에서 프로그램을 downloading 하여 동작

부트로더는 일반적으로 flash 메모리에 저장되어 있고 power-on이 되면 동작하게 된다. PPCBoot-1.2.1의 경우 cpu/ppc4xx/start.S 파일이 entry point이고 board/core405/ppcboot.lds에서 그 정보를 찾을 수 있다.

AP를 위한 PPCBoot-1.2.1 부트로더의 소프트웨어 디렉토리 구조를 (그림 4)에 보여주고 있다. 중요한 디렉토리에 대한 설명은 아래와 같다.

- 1) README : PPCBoot에 대한 전체적인 설명을 하고 있다. 필독이 요구된다.

<Table 2> 405GP AP 보드의 flash memory에 대한 operations

PPCBOOT side view			Linux side view		
Bank: sector	Usage	Operations	Device	Usage	Operations
1:0	ppcboot image	R: md	invisible	-	-
2:0	environment variables	R: md, printenv W: saveenv E: erase	mtd1	reserved	R/W/E
2:1-2:55	User data	R: md E: erase	mtd2	Linux image	R/W/E
2:56-2:63			mtd3 mtdblock3	mount on /config JFFS2	R/W/E

### 3.1. PPCBoot Bootloader

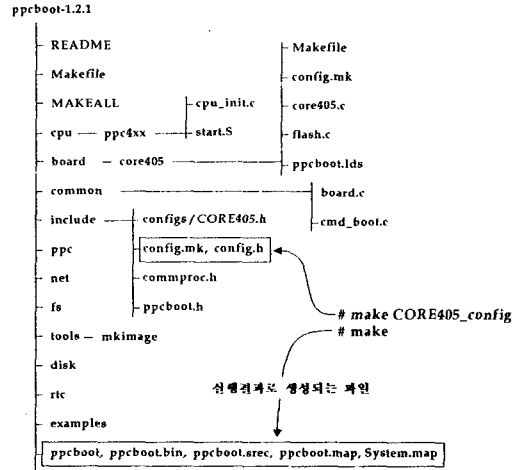
부트로더(boot loader)의 기본적인 기능은 (i)

- 2) include/configs: target보드를 위한 configuration 파일 및 헤더파일을 포함하고 있다.
- 3) common/ : 보드의 초기화 관련 파일과

PPCBoot 부트로더 명령에 대한 파일을 포함하고 있다.

- 4) board/core405 : target 보드에 적용하기 위한 파일을 포함하고 있다.(예: flash, SDRAM 제어를 위한 코드가 있다)
- 5) cpu/ : PPC4xx, MPC8xx, MPC8260 등의 PowerPC에 대한 초기화를 위한 파일을 포함하고 있다.
- 6) tools/ : mkimage 등과 같이 PPCBoot 이 미지 생성을 하는데 필요한 파일이 존재한다

AP보드의 power가 on 되면 405GP의 CS0에 연결되어 있는 ATMEL 플래쉬의 PPCBoot-1.2.1 코드가 실행된다. (PPCBoot의 최초 시작은 cpu/ppc4xx/start.S이고 PPC 어셈블리 코드로 구성되어 있다). cpu/ppc4xx/start.S에서 405GP에 대한 초기화를 수행한 후 common/board.c의 board\_init\_f를 호출하여 target보드 관련하여 초기화(특히 SDRAM의 초기화)를 한다. 이후 ATMEL 플래쉬에 있는 PPCBoot 부트로더는 SDRAM 영역으로 자신을 relocation하고 (이때까지는 플래쉬에 있는 코드가 동작한다), 다시 common/board.c의 board\_init\_r를 호출하여 target보드에 대한 나머지 초기화를 수행한 후 PPCBoot의 command prompt 상태로 간다 (relocation 한 후 PPCBoot는 SDRAM영역에 있는 코드가 실행된다). PPCBoot command prompt 상태에서는 사용자가 입력하는 PPCBoot 명령을 처리한다. 마지막으로 PPCBoot의 bootcmd 및 bootm 명령을 이용하여 INTEL flash에 저장된 Linux-2.4.21-pre5 커널이미지를 메모리에 load한 후 제어를 리눅스 커널에 넘겨준다.



(그림 4) AP를 위한 PPCBoot-1.2.1 부트로더의 구조

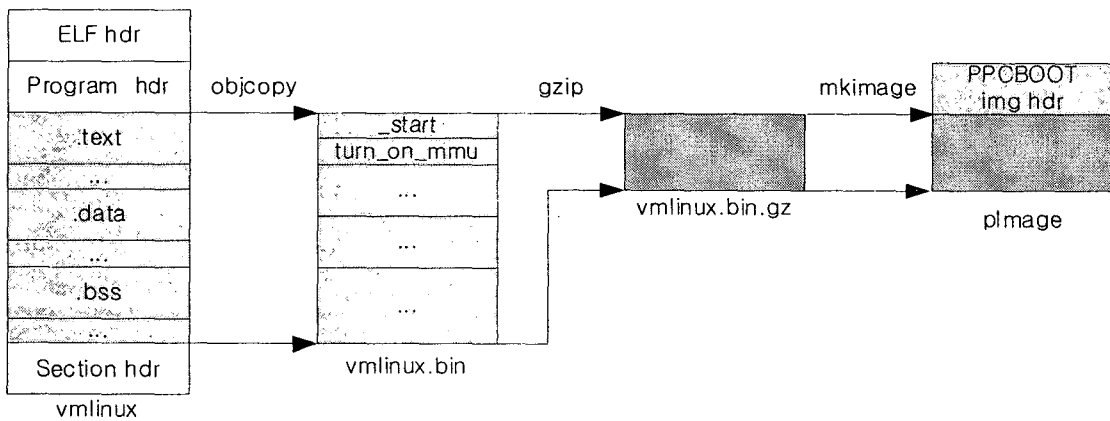
### 3.2. Linux 커널과 루트(root) 파일시스템

리눅스 커널이 다양한 CPU에 porting이 가능하도록 arch/ 디렉토리를 제공하고 있지만 이는 일반적인 desktop의 경우에 해당된다. 즉, PowerPC 계열인 경우 보통 PowerMAC과 같은 시스템을 의미하므로 IBM의 PowerPC 405GP를 위한 코드는 arch/ppc 디렉토리에 있는 커널 코드를 수정해야만 한다. 405GP 프로세서를 지원하기 위한 커널의 수정은 아래와 같다.

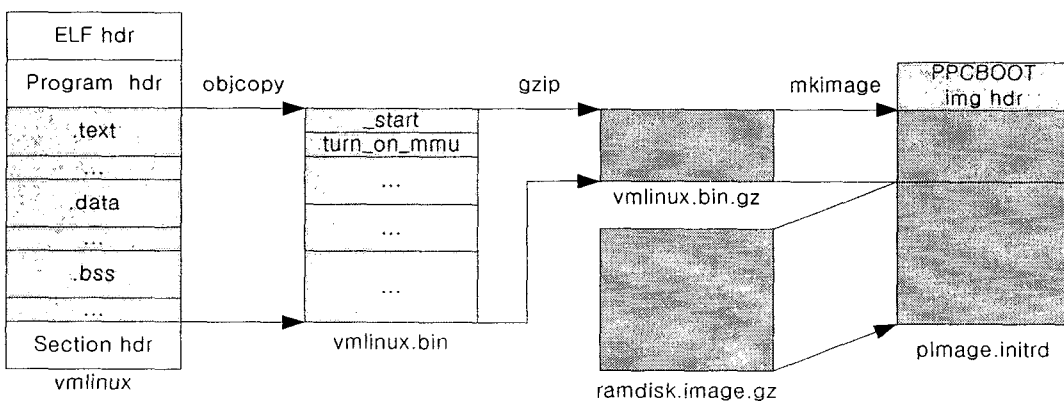
- (1) Makefile : cross compile을 위한 설정과 커널 build를 위한 target 생성
- (2) arch/ppc/config.in : 커널 configuration을 위한 menu에서 platform 관련 부분을 지정
- (3) include/asm-ppc/405.h : 선택한 보드에 따라서 포함시켜야 하는 header 파일을 지정
- (4) include/asm-ppc/processor.h : 보드에 ID를 부여하는 부분
- (5) arch/ppc/platforms/405board.h : 405GP

- 보드에 대한 특징을 초함하고 있는 파일
- (6) boot loader에서 정의된 bd\_t 구조체를 arch/ppc/platforms/405board.h에 추가
- (7) drivers/char/serial.c : AP가 동작을 하던 도중에 console이 동작하지 않는 경우 또는 시스템의 동작을 확인할 수 없는 경우에 커널 내부의 프로세스 상태 등을 출력하여 시스템의 상태를 진단할 수 있는 기능을 포함하고 있다.
- (8) drivers/mtd/maps/physmap.c : MTD 기능을 지원하기 위한 부분

root FS은 커널이 부팅되면서 마운트가 이루어진다. 현재 boot loader 환경변수, 커널과 root FS은 8M 용량의 application 플래쉬에 저장된다. 따라서 응용프로그램, 유틸리티, 라이브러리 등을 포함하고 있는 root FS의 크기를 최적화 하는 것이 중요하다. 이를 위하여 시스템 운영에 필수적인 유틸리티를 하나의 파일로 제공하는 busybox를 이용하여 root FS을 구성하였다.



(a)



(b)

(그림 5) Embedded Linux에서 target image의 생성

### 3.3. AP target image의 구성 방법과 부팅 과정

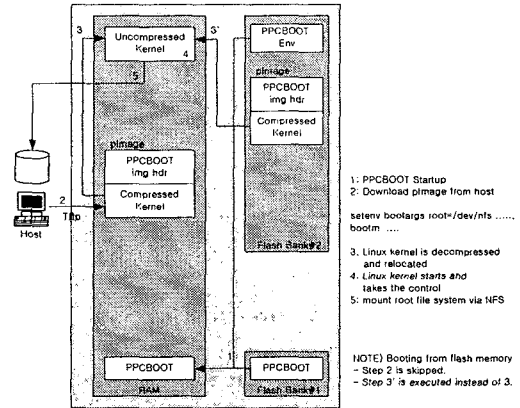
Embedded Linux의 software packaging 요소는 Root FS과 Linux kernel로 구분된다. 이때 Root FS과 kernel을 별도의 image로 구성하여 packaging할 수도 있고 하나의 image로 packaging할 수도 있다. (그림 5)(a)는 kernel을 하나의 image로 구성하는 과정을 보여주고 있다. 이 경우에는 root FS 역시 하나의 image로 만든 후 kernel이 booting되면서 root FS을 호출하는 방식이다. (그림 5)(b)는 kernel과 root FS을 image로 만든 후 kernel이 booting되면서 root FS을 호출하는 방식이다.

Target image의 생성이 완료된 후 이들 image를 packaging을 하고 system을 동작시키는 방법은 크게 2가지로 구분할 수 있다. (그림 6)(a)는 kernel image가 flash에 존재하고 root FS은 개발호스트에 존재하는 경우로 주로 embedded system의 개발단계에서 사용되는 방법이다. 반면에 (그림 6)(b)는 kernel image와 root FS image가 모두 flash에 존재하는 상태로 system의 모든 동작이 embedded board 자체에서 이루어진다. Embedded system의 개발이 완료된 다음의 동작모습이다.

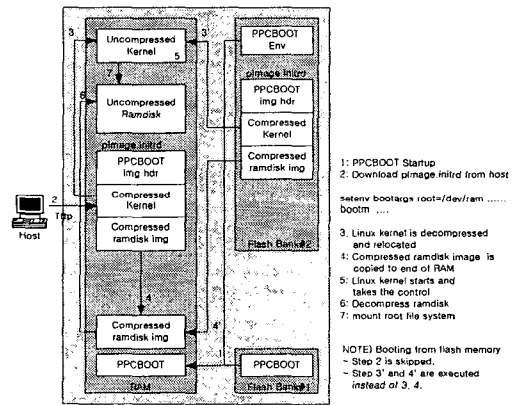
NFS를 이용하는 경우의 booting sequence는 다음과 같다.

- (1) Flash에 있는 monitor program인 boot loader가 수행되면서 CPU, memory, interrupt 등의 초기화를 수행한다.
- (2) bootloader에서 호스트에 있는 kernel image를 download 한다. 또는 flash에 있는 kernel이 동작하도록 할 수도 있다.
- (3) kernel image가 RAM에서 압축이 풀리고 호스트에 있는 root FS에 NFS로 접속하

여 Linux booting이 이루어진다.



(그림 6) (a) NFS booting



(그림 6) (b) Ramdisk booting

개발이 완료된 후 모든 image가 flash에 저장된 상태에서의 booting은 다음과 같은 절차로 이루어진다.

- (1) flash에 있는 bootloader가 동작하면서 system을 초기화 한다.
- (2) Linux kernel image가 RAM에서 압축이 해제되고 root FS을 mount 한다.
- (3) Root FS이 무엇인지에 따라 RAM이나

flash에 FS이 구성되고 Linux booting이 완료된다.

#### IV. 결론

본 논문에서는 IEEE 802.11a 기반의 WLAN AP의 개발에 관하여 연구하였다. AP의 하드웨어는 IBM PowerPC 405 GP 프로세서를 사용하여 200MHz로 동작하도록 설계되었고 부트 플래쉬와 응용 플래쉬 메모리를 분리되어 있다. 또한 무선을 위한 인터페이스는 32 비트의 CARDBus 규격으로 설계되었다. AP의 소프트웨어는 부트로더, 커널, 파일시스템으로 구성되어 있다. 부트로더는 PPCBoot-1.2.1을 사용하고 있고 부트 플래쉬에 저장되어 있으며, 커널은 Linux-2.4.21을 사용하고 있고 응용 플래쉬에 저장되어 있다. 그리고 파일시스템은 응용 플래쉬 메모리에 저장될 수 있도록 busybox 등을 이용하여 구현하였다. 특히 시스템 패키징을 위하여 Linux 커널이 지원하는 MTD 기술과 JFFS2 플래쉬 파일시스템을 이용하여 최적으로 구현하였다.

현재 개발된 WLAN AP는 시험용이기 때문에 다소 불안정한 동작을 하는 경우가 나타나고 있으므로 이에 대한 안정화 작업이 추가로 요구되고 있다. 또한 최근 리눅스 커널이 2.6.x의 버전이 공개되었고, 따라서 AP가 linux-2.6.x 커널 기반으로 동작할 수 있도록 소프트웨어 upgrade 작업이 필요하다.

#### 참고문헌

- [1] Motorola, *Programming environment manual*
- [2] IBM, *IBM PowerPC 405GP User's Manual*
- [3] Dokyu Kim, *Technical report for the development of AP software, 2002*
- [4] <http://www.gnu.org/>
- [5] <http://ppcboot.sourceforge.net/>
- [6] <http://www.linux-mtd.infradead.org/>
- [7] <http://sources.redhat.com/jffs2/jffs2-html/>
- [8] <http://www.denx.de/>
- [9] <http://www.kesl.org/>
- [10] <http://www.denx.de/>



## Development of WLAN AP based on IBM 405GP

Do-Gyu Kim\*

### Abstract

The evaluation AP embedded Linux board is implemented. The board is made of IBM 405 GP processor, PPCBoot-1.2.1 boot loader, Linux-2.4.21 kernel and root file system. The evaluation board has two flash memories, boot flash and application flash of size 512Kbyte and 16Mbyte, respectively. And it supports IEEE 802.11a which provide the maximum throughput of 54Mbps in the 5.2GHz frequency band.

MTD(Memory Technology Device) and JFFS2(Journalling Flash File System version 2) technologies are adopted to optimally package the system software, boot loader, kernel and root file system. And in order to optimize root file system, busybox package and tiny login are used. Linux kernel and root file system is combined together with mkimage utility.

---

\* Professor, Dept. of Information Communication Engineering, Sungkyul University