

주 제

ATCA 기반 라우터 시스템의 Embedded control channel 구조 설계

LG-Nortel 이현철, 이학묵

차 례

- I. 서 론
- II. ATCA 기반 라우터 구조
- III. Embedded control channel의 구조 및 구현
- IV. 결 론

본 기고문은 ATCA를 기반으로 한 라우터 시스템 내에서 Embedded control channel의 구현 및 적용 사례를 설명하는데 중점을 두었다. 시스템 내에서 신뢰성 있는 제어를 위한 프로세서 간의 통신 경로를 구현함에 있어서, 구조가 단순한 이더넷 방식을 실시간 운영 체제인 FT-Linux상에 적용하여, 시스템의 용량 확장 및 유지가 용이하도록 설계하였다. 또한 In-band Control channel의 단점인 트래픽 처리 용량의 손실이 없는 구조를 지향한다.

I. 서 론

일반적으로 임베디드 시스템이란 규격화, 표준화된 시스템을 설계자가 의도한 특정한 기능만을 수행하는 장치, 미리 정해진 특정한 기능을 수행하도록 프로그램이 내장되어 있는 시스템, 시스템의 일부이거나 독립된 시스템, 특별한 업무를 수행하거나 사용자

가 임의로 정한 업무를 수행, 하드웨어와 소프트웨어를 포함하는 특정한 응용 시스템, 하드웨어와 소프트웨어의 변경이 매우 어려운 시스템, 개인 휴대 정보 단말, 지리 정보 시스템, 의료 정보 단말, 정보 가전, 게임기기, 자동차, 항공기 및 우주선, 의료 및 산업 원격 조종 장비 등의 시스템을 총칭하는 것으로서, 임베디드 시스템의 구성은 임베디드 H/W인 프로세서/컨트롤러, 메모리, I/O 장치, 네트워크 장치, 센서, 구동기 등과 임베디드 S/W인 운영체제, 시스템 S/W, 응용 S/W등으로 나눌 수 있다.

임베디드 시스템의 설계 시 핵심이 되는 사항으로는 하드웨어와 소프트웨어가 개발할 기능의 분할이며, 이를 위해 적절한 프로세서의 선택, 적절한 운영체제의 선택 및 적절한 디버깅/개발 툴의 선택이 뒤따라야 한다.

이러한 임베디드 시스템의 개발은 통신 장비와 접목되어 고속 고품질의 IP기반 멀티미디어 서비스로 전환되는데 중요한 역할을 하고 있다.

정보통신 시스템의 발전 방향에 대하여 살펴보면, 정보통신 서비스는 광대역화 및 이동성을 지향하며 발전하고 있으며, 향후에는 정보통신 네트워크 서비스 및 기기의 융합이 가속화 되는 것으로 전망하고 있다.

광대역 통합망에서는 데이터 서비스뿐만 아니라 음성과 영상 그리고 방송을 포함한 다양한 서비스들이 단일화된 패킷 전달망을 통하여 각각의 특성과 요구 사항에 맞게 처리 될 수 있어야 하며 이를 위해서는 시스템이 다기능 및 고성능을 제공하여야 한다. 또한 폭발적으로 늘어나는 정보통신 단말과 홈네트워크로 대변되는 정보 가전의 발전으로 인하여 통신에 참여하는 개체수가 기하급수적으로 증가함에 따라 이들을 수용할 수 있는 새로운 통신 장치가 필수적이라고 할 수 있다.

그리고 망이 통합되면서 통신 사업자들은 기존의 다양한 액세스망에 품질 보장형 망을 추가로 구축하여 효율적으로 운영하기를 원하고 있으며 이를 위해서는 표준화된 개방형 시스템 플랫폼[1]의 사용이 요구되고 있다

본 기고문은 광대역 통합망의 전달망 요소인 에지 라우터에 요구되는 기능 및 성능항목인 개방형 시스템 플랫폼을 기본적으로 적용한 상태에서 Micro processor 및 FT-Linux를 이용한 Embedded control channel의 구조 및 구현에 대하여 설명할 것이다.

ATCA 기반 라우터 시스템의 Embedded control channel 구조 설계에 대한 본 기고문의 구성은 다음과 같다. II장에서는 개방형 플랫폼인 ATCA의 설명과 더불어 ATCA에 기반을 두고 설계된 라우터의 논리적, 물리적 구조에 대해서 기술한다. III장에서는 Embedded control channel의 H/W구성 요소로서 프로세서와 L2 스위치에 대한 설명을 하고, S/W구성 요소로서 실시간 운영 체제 및 S/W 드라이버에

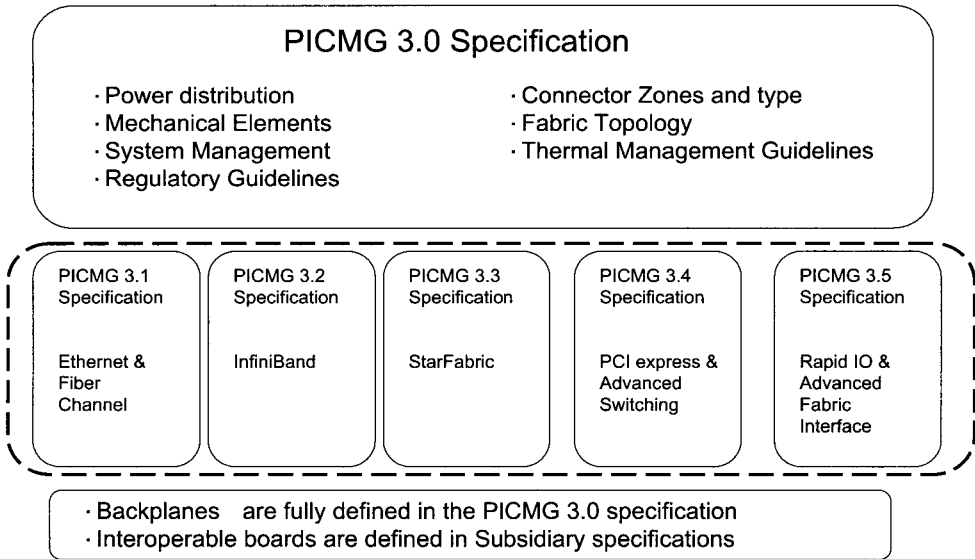
대한 설명을 한 후, Embedded control channel의 구조 및 구현 방법에 대해서 기술하며, IV장에서 결론을 맺는다.

II. ATCA 기반 router 구조

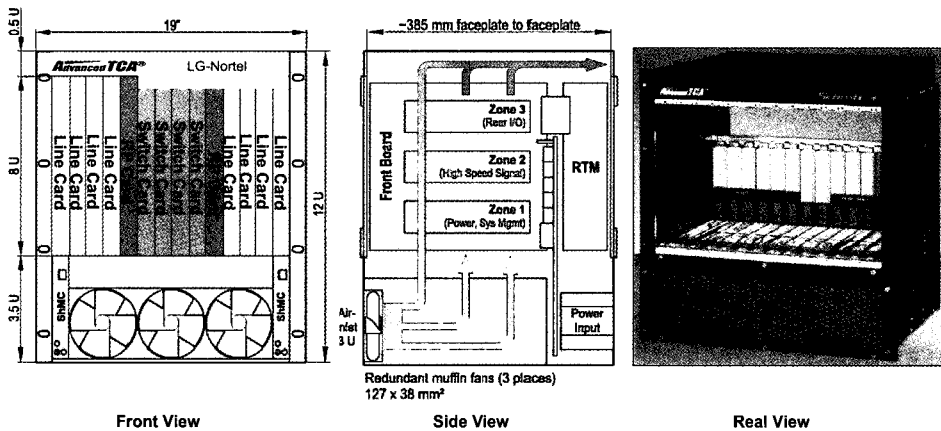
1. Advanced TCA

AdvancedTCA[1]은 PICMG (PCI Industrial Computer Manufacturers Group)에서 제안된 표준으로서 통신 시스템 (에지 및 코어 전달망 시스템, 무선, 광네트워크)에 적용 가능한 표준화된 플랫폼이다. 시스템을 구성하는 물리적인 규격이 우선적으로 통일되어야 각기 별도로 개발된 보드들이 하나로 통합될 수 있다.

PICMG에서는 라우터를 포함하여 다양한 통신 플랫폼에 적용될 수 있는 물리적인 규격과 함께 플랫폼 및 컴포넌트 관리에 대한 표준화를 진행하여 왔으며, 그 결과 PICMG 3.x 규격으로 발표된 표준이 ATCA (Advanced Telecom Computer Architecture) 구조이다. Intel의 분석에 의하면 ATCA 구조를 채택할 경우, 시스템 개발 비용 면에서 40% 정도의 절감 효과가 있으며, 새로운 에지 시스템을 시장에 공급하고자 하는 경우 12개월에서 18개월 정도 기간 단축을 기대할 수 있다고 보고하고 있다. 현재 개발된 ATCA 규격은 (그림 1)과 같다. (그림 1)은 Back plane표준을 근간으로 하는 PICMG3.0을 중심으로 가입자 블록이나 스위치 정합을 위한 PICMG3.x 표준들로 구성되어 있는 개방형 플랫폼 구성을 보여주고 있다. 핵심 규격인 PICMG 3.0에서는 통신 시스템 아키텍처 전반에 대한 내용을 다루고 있으며, 나머지 규격에서는 Backplane에서의 각 인터커넥션 방식에 대하여 다루고 있다. 인터커넥션 방식이 여러 가지인



(그림 1) PICMG3.0 Specification



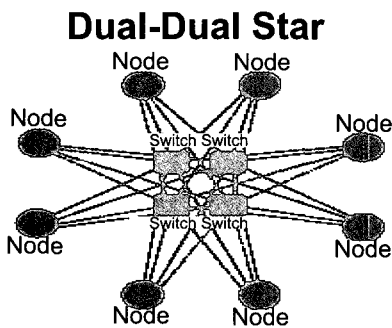
(그림 2) ATCA system configuration

까닭에 각 보드들의 완전한 호환성이 있는 것은 아니지만 AS (Advanced Switching) 규격을 통해서 이러한 문제를 완화하고 있다. 통신 시스템의 경우 업체들이 각기 독자적인 시스템 구조를 채택하고 있어, 모든 컴포넌트를 개발할 수 있는 자금력과 기술을 보유

하고 있는 몇몇 메이저급 업체들이 시장의 대부분을 차지하고 있다.

AdvancedTCA의 적용은 시스템 제조사로서는 시스템의 모든 기능 모듈을 독자 개발하지 않아도 되기 때문에 개발 기간 및 총 비용을 줄일 수 있는 효과

가 있으며, 개방형 구조 및 모듈화된 구성품으로서 특정 기능 모듈에 개발 역량을 집중함으로써 시스템의 안정성 및 신뢰성을 확보할 수 있다. 뿐만 아니라 표준화된 단일 플랫폼을 사용할 수 있어 지속적인 시스템의 업그레이드 및 운용의 효율성을 취할 수 있고 적기에 시장 진입을 할 수 있는 장점이 있다.



(그림 3) Dual-Dual star topology

ATCA의 특징으로서 최대 2.5 Tbps의 스위칭 용량 확장이 가능하고, 패브릭 정합시 고속 직렬 링크를 적용하고 있으며, 스위치 패브릭 기술로는 (그림 3)과 같은 이중-이중성형(Dual-Dual star), 이중성형(Dual-Star) 및 전연결(Full-Mesh) 데이터 전송로를 구성하고 있다. 각기 제어 경로로 사용할 수 있는 듀얼 스타 구조의 기가 비트급 이더넷 인터페이스와 데이터 경로로 사용되는 패브릭 인터페이스가 있으며, 서로 상대방 보드가 제대로 운용되고 있는지의 상태를 체크할 수 있는 업데이트 포트, 전원, 온도 등 플랫폼의 상태 데이터가 지나다니는 IPMI(Intelligent Platform Management Interface) 등이 있다. 규격화되어 있는 Zone 1, Zone 2 이외에 시스템 설계에서 독자적으로 사용할 수 있는 Zone 3 영역이 정의되어 있으며, 이를 통해 RTM 보드와 전면 보드가 연결된다.

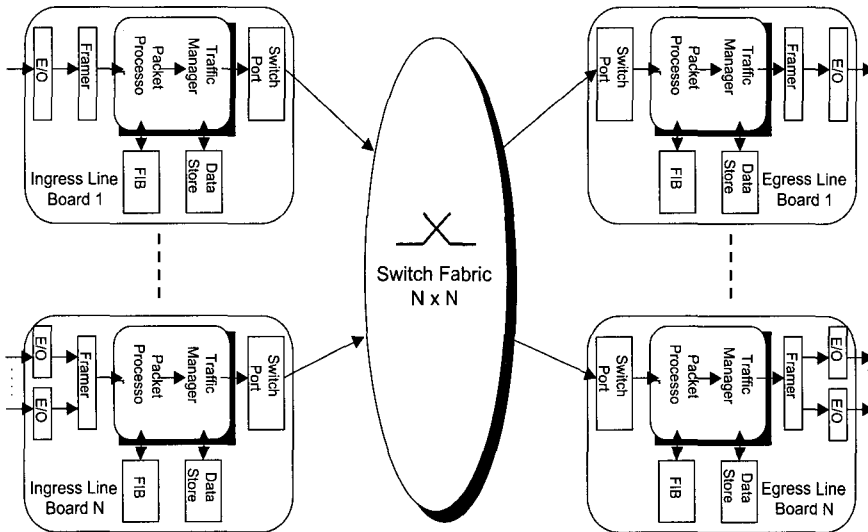
2. ATCA 기반 라우터 구조

ATCA 기반 에지 라우터는 시스템 플랫폼에 대한 광대역 통합망[2]의 요구사항을 수용하여 차세대 개방형 표준 플랫폼 구조를 가질 수 있도록 설계 한다. ATCA 기반 라우터는 단일화된 패킷망인 광대역 통합망 환경에서 기존의 다양한 가입자 네트워크와 IP/MPLS로 대변되는 코어망 사이에 위치하여 가입자로부터의 트래픽 또는 역방향의 트래픽을 수용하고 정해진 트래픽의 특성에 맞게 처리하여 전달망의 코어 또는 가입자 블록으로 보내는 역할을 한다.

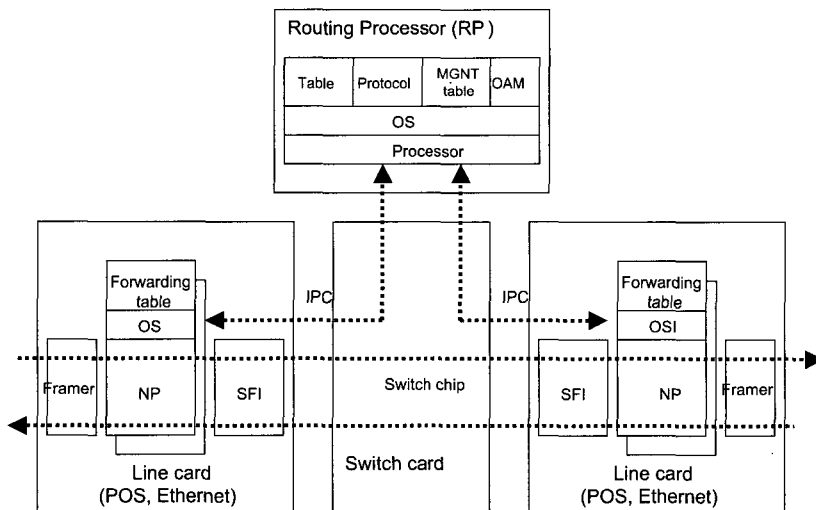
ATCA 기반 라우터의 경우, (그림 4)와 같이 다수의 인터페이스 라인카드와 라우팅 프로토콜이 수행되는 프로세서, 그리고 각각의 라인카드와 프로세서를 연결해 주는 스위치로 구성된다.[3]

이와 함께 시스템의 신뢰성을 높이기 위해서 프로세서와 스위치와 같은 주요 컴포넌트들은 이중화되어, 동작하던 보드에 이상이 발생하더라도 대기하던 백업용 보드가 이후의 처리를 이어받아 중단 없는 운용이 이루어지도록 하고 있다. 패킷 프로세서의 역할은 패킷을 분류하여 정보를 분석하고, 각 패킷의 전달을 위한 적절한 수정 및 필요한 실행(pass, drop, mark)을 수행한다.

대용량 라우터에서는 패킷 전달을 고속으로 분산 처리하기 위해서, 라우팅 프로토콜에 의해 생성된 RIB(Routing Information Base) 정보가 각 라인카드에 FIB(Forwarding Information Base) 형태로 내려지며, 라인카드로 들어오는 패킷들은 이 FIB를 룩업하여 전달되어야 할 라인카드 및 인터페이스를 결정한 후 데이터를 전달하게 된다. 라우터에서의 패킷은 자신을 시발점이나 종착점으로 하는 컨트롤 패킷과 이웃한 라우터로 단순히 전달되는 데이터 패킷으로 구분되는 데, 이들 패킷이 전달되는 경로를 각기 제어 경로와 데이터 경로라고 한다. (그림 5)는



(그림 4) ATCA 기반 광대역 라우터의 물리 구성도



(그림 5) ATCA 기반 광대역 라우터의 논리 구성도

ATCA 기반 광대역 라우터 시스템의 전체적인 구성을 논리적으로 그린 것이다.

라우터는 기능상으로 프로토콜 및 통신 응용들이 수행되는 제어 요소 (Control Element)와 데이터를 고속으로 전달하는 전달 요소 (Forwarding

Element)로 구성된다.

ATCA 기반 광대역 라우터는 일반적인 라우터 시스템과 마찬가지로 크게 제어부, 스위칭부, 그리고 라인카드의 세 부분의 기능 블록으로 구성되어 있다. 제어부는 CPU와 운영체제를 비롯하여 각종 제어모듈

과 메모리가 주된 구성요소이며 시스템 전체의 동작과 기능 관리 및 라우팅 프로토콜과 시그널링 프로토콜의 처리를 수행한다.

라인카드는 네트워크 프로세서를 이용하여 물리 인터페이스로부터 들어오거나 나가는 각종 패킷의 프로토콜 처리 기능을 담당하게 된다. 그리고 스위치 패브릭 칩으로 구성된 스위칭부는 입력포트로 들어온 데이터에 대해서 올바른 목적지 인터페이스로 보내기 위한 스위칭 기능을 수행한다. 시스템의 이러한 유기적이고 원활한 동작을 위해 각 기능 블록간에 정보 및 데이터 교환이 필요할 경우 Control channel (IPC: Inter processor communication)를 통하여 통신하게 된다.

최근의 라우터 구조를 보면 모듈화, 확장형 라우터 구조로서 분산형 구조를 채택하고 있다. 분산된 패킷 전달, 내부 트래픽 처리를 함으로 인해서 모듈화가 가능해 진다.

ATCA 기반 광대역 라우터는 시스템의 기본형상으로서 통신 산업계에 새로운 하드웨어 표준으로 부상하고 있는 차세대 개방형 플랫폼 구조를 적용했다. 차세대 개방형 플랫폼 구조는 새로운 폼팩터 (Form factor)와 스위치 패브릭 구조에 기반한 차세대 통신 장비의 위한 표준화된 개방형 플랫폼의 한 형태이다.

차세대 개방형 플랫폼 구조에서 스위치 카드와 라인카드를 포함한 각 기능 카드간의 연결 토폴로지에는 10 Gb/s 라인속도 지원에 대한 성능문제와 스위치와 프로세서가 각각 독립적으로 이중화 운영될 수 있는 구조인 듀얼-듀얼 성형 토폴로지를 사용한다.

ATCA 기반 광대역 에지라우터는 시스템의 라인카드는 POS, Ethernet, ATM 인터페이스를 동시에 수용하면서 10 Gb/s 라인속도를 제공하며 크게 세 부분의 기능 블록으로 나눌 수 있다. 네트워크 프로세서를 포함하는 NP 모듈이 각각 입구 (Ingress)와 출구 (Egress) 트래픽을 처리하고 스위치 카드의 스위

치 패브릭 칩과의 연결을 위한 스위치 패브릭 인터페이스 및 트래픽 매니저 부분, 그리고 실제적인 물리 케이블 및 포트가 접속되는 물리카드인 PMC (PCI mezzaine card)가 있다.

PMC 는 믹스앤매치 (Mix and match) 기술을 적용함으로써 라인카드의 물리 인터페이스의 제작 종수를 감소시킬 수 있고 통신사업자는 하나의 라인카드에서 다양한 종류의 인터페이스 및 속도를 제공할 수 있기 때문에 망의 환경에 따라 최적의 시스템 운용을 할 수 있는 장점을 가진다.

III. Embedded control channel의 구조 및 구현

1. Embedded H/W 구성

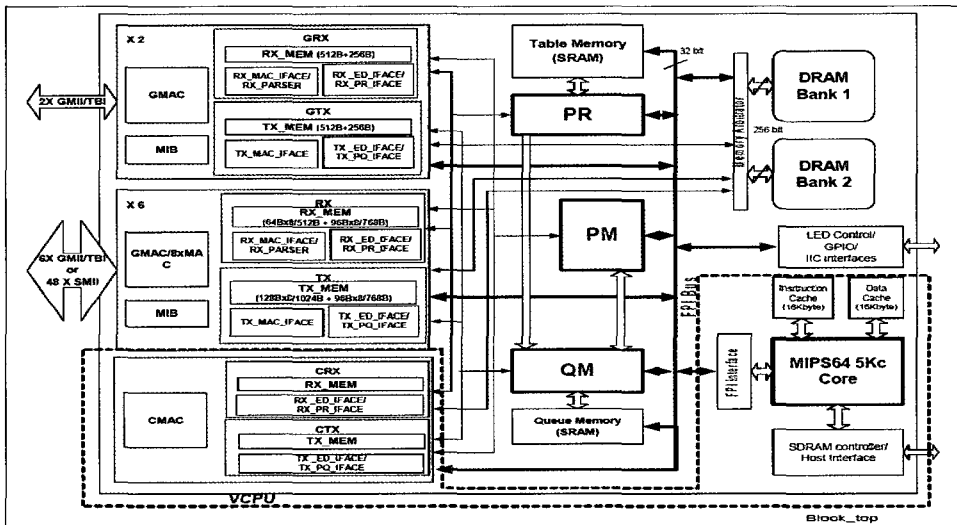
Embedded control channel의 H/W구성은 Ethernet 스위치[4], 스위치 내의 MIPS계열의 프로세서 및 Control channel 경로로 구성되어 있다. Ethernet 스위치는 라우터내의 각종 프로세서간의 packet을 L2 스위칭하는 기능을 수행하며, MIPS계열의 프로세서는 운영체제를 비롯한 각종 S/W 드라이버를 포팅하게 된다. 그리고 프로세서간의 경로는 PCB 패턴 및 물리적 케이블을 이용하여 구현하도록 설계되었다.

Ethernet 스위치는 최대 8포트의 GbE 포트가 있으며, Embedded packet resolution 과 Forwarding engine은 Full-duplex GbE 포트 스위칭을 수행함에 있어서 라인 속도를 지원하도록 설계되었다.

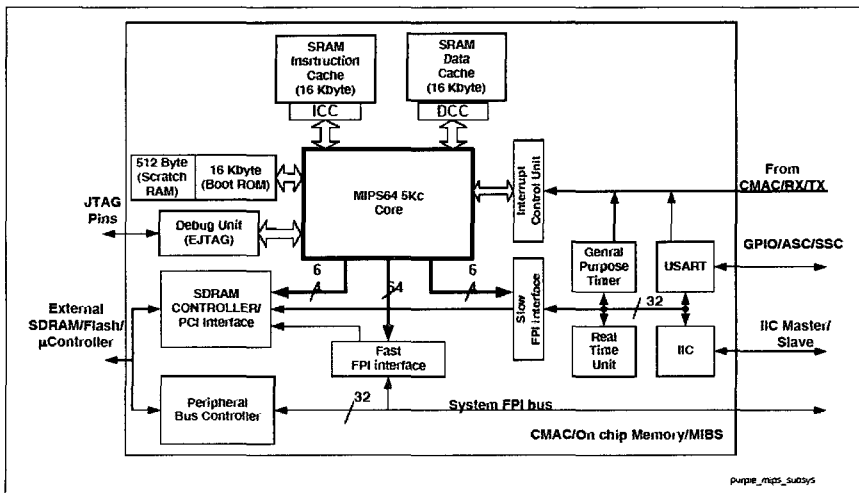
On-chip 프로세서는 (그림 7)에 도시된 바와 같이 64 bit RISC CPU (MIPS64 5Kc)으로서 Embedded DRAM 과 내부 memory를 이용하여 8 포트 GbE 스위치를 제어할 수 있도록 설계 되었다.

동작 주파수는 125 MHz로서 16 Kbytes의 명령 및 데이터 캐쉬를 각각 수행할 수 있다. FPI (Flexible Peripheral Interconnect) 지원으로 32bit bus를 이용한 주요 블록 및 메모리와 연결되고, PC-133 SDRAM과도 호환된다. 또한 On-chip UART와 타이머, GPIO를 지원하여 효과적으로 자원을 제어 할

수 있다. Ethernet 스위치의 8개 기가 포트중에 6개의 포트는 FE 포트로 구현 가능하여 48FE와 2GbE와 같이 구현 가능하다. 패킷 버퍼는 Embedded DRAM으로서 256 bit의 두 뱅크로 구성되어 있으며, 동작 주파수는 125 MHz로서 최대 21 Gbit/s의 메모리 대역까지 구현 가능하다.



(그림 6) Ethernet 스위치 및 프로세서의 블록도

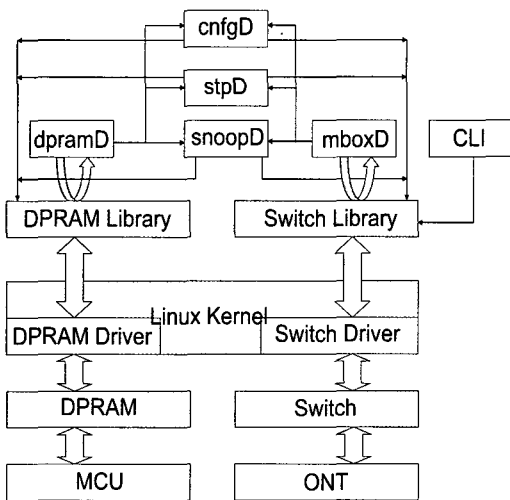


(그림 7) 프로세서의 구조

스위칭 속도는 초당 1200만 패킷이며, 4 단계 우선순위를 큐를 운영하여 512개의 플로우를 처리한다. SMII는 10/100 Mbit/s 정합을 지원하고 GMI/TBI/RGMII/MII는 1000 Mbit/s 정합을 지원한다.

2. Embedded S/W 구성

Embedded control channel의 S/W 구성은 Boot, 실시간 운영체제, 구동 드라이버로 나눌 수 있다. Boot는 운영체제와의 호환성을 위하여 Redboot을 사용하였으며, 실시간 운영체제는 FT-Linux를[5], 구동 드라이버는 Ethernet 스위치용 driver를 적용하였다. (그림 8)은 FT-Linux와 Ethernet switch driver와의 I/F를 나타낸다.

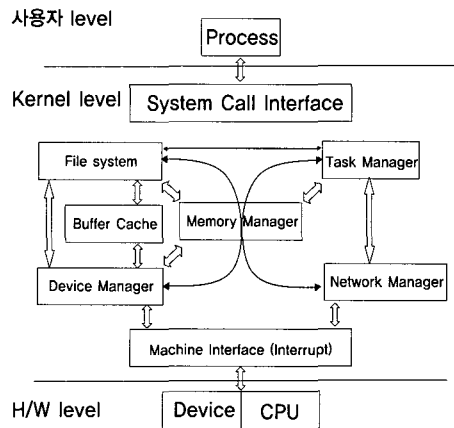


(그림 8) Linux Kernel 의 동작 방식

운영체제인 리눅스는[6] 리누스 토발즈가 미닉스에서 개발했다. 처음부터 멀티태스킹이 가능한 유닉스 호환을 지향했고 현재는 POSIX에서 제안한 유닉스 표준을 지키고 있다. 초기에는 리누스가 모든 개발을 진행했지만 인터넷에 커널 소스를 개방시킨 결과

수많은 개발자들이 기여를 하여 기능 추가, 버그 개선 등이 빠른 속도로 진행되었다. 이 과정에서 개발자들이 사용자가 되고 사용자가 개발자가 되는 상승 효과로 리눅스 사용자 또한 빠르게 증가했다. 여기에 더해 리눅스 상용 패키지 회사까지 경쟁적으로 나타나 유닉스에 경험이 전혀 없는 초보자도 몇 시간 안에 리눅스를 설치하고 설정까지 최적화시킬 수 있을 정도로 발전했다.

리눅스 커널[7] 소스의 디렉토리 구조는 /usr/src/linux 에 소스 트리가 생성되며, 각 트리 별로 arch/ 디렉토리에는 지원하고 있는 CPU에 의존적인 코드가 있는데, 부팅 과정, 메모리 관리 방법, 프로세스 처리 부분들이 모두 다르므로 가장 하드웨어와 밀접한 코드가 모여 있다. fs/ 디렉토리는 논리적인 파일 시스템을 처리하는 부분이다. 리눅스에서는 가상 파일 시스템을 사용하여 동시에 여러 가지 파일 시스템을 지원할 수 있다.



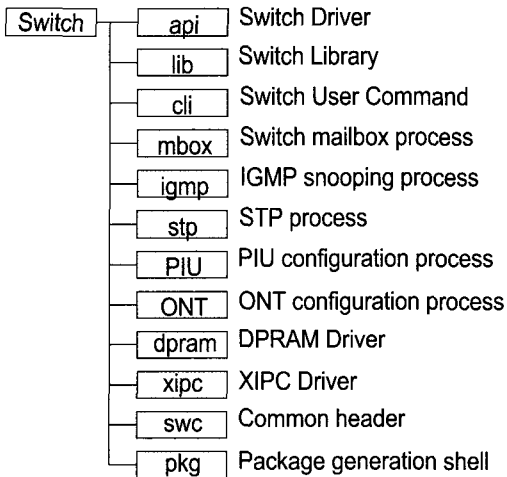
(그림 9) 커널 내부 구조

mm/ 디렉토리에서는 메모리 처리 관련 프로그램이, net/ 에는 네트워크 관련 코드가 있다. 커널 디바이스 드라이버를 만들기 위한 drivers/ 디렉토리에는

커널용 디바이스 드라이버가 모두 모여 있다. 그 외 여러 디렉토리가 있다.

리눅스를 운영체제로 선정하게 된 가장 주된 이유는 개발사가 지원하지 않아도 사용자가 필요한 하드웨어에 대한 명세를 구하여 스스로 만들 수 있다는 것이다. 리눅스는 커널까지 완전히 개방되어 있기 때문에 개발자는 전세계의 모든 리눅스 사용자들이며, 새로운 하드웨어를 위한 디바이스 드라이버는 이미 누군가 먼저 개발해 놓고 누구나 사용할 수 있도록 인터넷에 올라와 있는 경우가 대부분이다.

(그림 10)은 S/W 구조로서 Ethernet switch의 구동 드라이버를 비롯하여 XIPC와 각종 제어 데이터의 처리 및 Switching을 수행하는 프로세스로 구성되어 있다.



(그림 10) S/W architecture

마지막으로 리눅스 개발 툴은 Monta Vista 리눅스 프로세서널 판 3.0으로서 임베디드 운영체제와 상호 개발 환경을 완벽하게 제공한다. 툴의 설치는 호스트의 /opt 디렉토리에 한다.

3. Control channel 구조

기본 shelf와 확장 shelf 스위치 카드내 Ethernet L2 switch간 IPC 연결은 full cross 이중화로 구성되어 각 shelf에 control backplane hub에 위치하는 스위치 카드가 한 개씩 불량 상태가 되더라도 시스템 내 IPC 경로는 계속 정상적으로 유지할 수 있도록 설계되었다.

L2 Ethernet switch는 Unmanaged mode에서 Fast aging 형태로 운용되며 개별 shelf내 유지되는 두 개의 L2 Ethernet switch 가운데 한 개는 비 동작 상태로 되어 shelf내 라인 카드와 RP에서는, 1개의 L2 switch만 보이도록 운용한다.

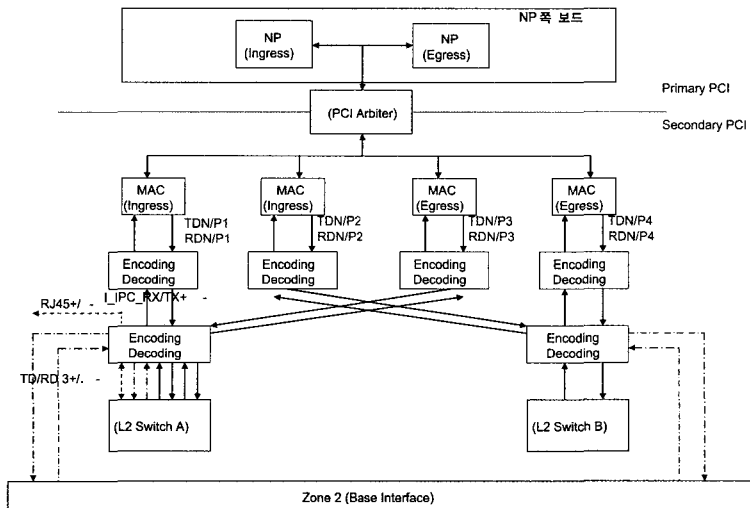
시스템 클럭은 Control backplane hub 의 62.5MHz의 클럭 신호를 사용하며, 시스템 backplane 클럭 정합을 통해 시스템 내 라인카드와 RP에 공급한다.

Extension shelf관련 모든 데이터 경로 및 제어 경로에 대한 연결은 스위치 카드 전면을 통한 물리적 라인을 통해 구현한다.

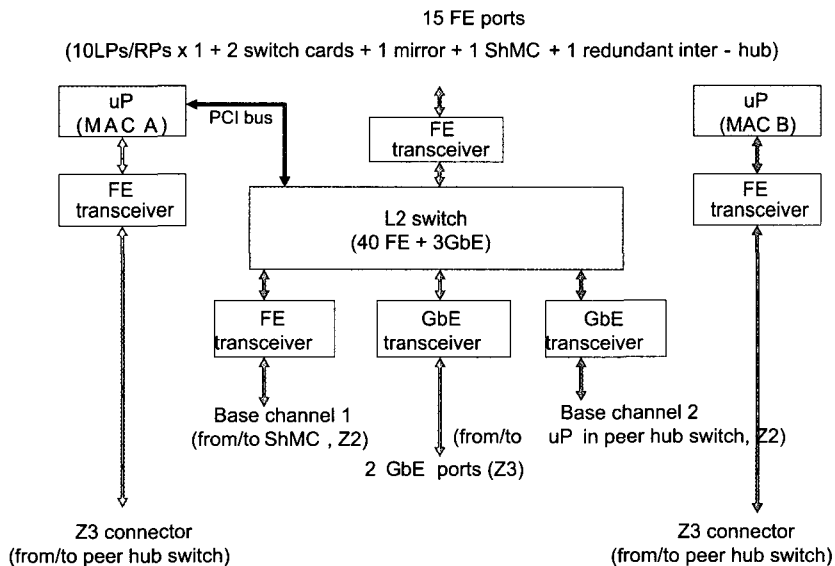
Control channel 정합은 Dual star 구조의 Z2 base interface를 통한 hub-slot간 GbE interface가 존재한다. 이것은 shelf내 가입자에 Dual star 구조를 통해 두 개의 L2 스위치 연결을 갖는 Control channel 구조에서 한 개의 L2 스위치와 연결된 100Base-T control channel 링크에 불량이 발생한 경우에 다른 가입자 Control channel 경로에는 영향을 주지 않고, 불량 발생 링크만 상대방 L2 스위치의 해당 링크로 전환할 수 있는 Control channel의 포트 별 이중화를 지원하기 위한 연결이다. Base interface Hub는 보드 내 processor MAC정합을 통해 상대편 Hub의 L2 스위치에 100Base-T로 연결된다. Shelf내 non-hub는 가입자와 마찬가지로 Dual star 구조의 Z2 base interface backplane을

통해 이중화된 100Base-T 링크로 연결된다. 단, Z2 base interface의 Non-hub의 100Base-T Control channel 경로는 Z3 trace를 통해 연결한다. Hub내 L2 switch가 모두 active로 동작할 경우,

동일 Source/destination MAC에 대해 다중 경로를 형성하며, loop를 형성한다. 이러한 loop를 방지하기 위해 Control channel switching 용 L2 switch에는 RSTP가 운용되어 각 Source/destination MAC에



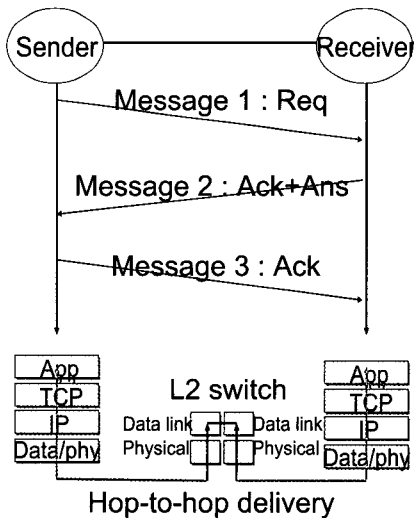
(그림 11) Source/Destination의 Control channel 경로 구조



(그림 12) 동일 shelf내 Control channel의 구현

대해 단일 Active 경로만을 갖도록 설정하며, Hub 불량 시 우회 Control channel 경로를 제공한다.

각 프로세서간의 Control channel을 이용한 통신 프로토콜은 (그림 13)과 같이 이더넷 프레임에 이용한 L2 스위칭 이기에 데이터 링크 레벨까지 처리가 되는 단순한 구조이다. 프로세서간의 메시지는 3단계로 이루어지는데, 송신단에서는 수신단의 목적지 address를 설정하여 요구 메시지를 보내고, 수신단에서는 응답과 함께 수신 확인 메시지를 송신하며, 송신단에서는 응답을 정상적으로 수신하였다는 수신 확인 메시지를 송부함으로써 통신이 이루어질 수 있다. 여러 개의 프로세서간의 스위칭은 L2 스위치가 전담하기에 블록킹은 발생하지 않는다.



(그림 13) Peer-to-peer flow

소로 인식되어 왔다. ATCA와 같은 개방형 플랫폼 또한 개발 기간의 단축 및 시스템의 호환 측면에서 라우터 설계에 있어서 중요한 요소이다. 따라서 ATCA 구조의 라우터 시스템에서 Control channel의 임베디드 시스템화 함으로서 저 비용으로서 고 신뢰성의 제어 경로를 구현할 수 있고, 광대역망에서 고품질의 라우터로서 역할을 수행할 수 있는 시스템을 설계할 수 있다.

본 기고문에서는 광대역망의 전달망 요소인 라우터에서 임베디드 Control channel을 구현함에 있어서 효율적이고 최적화된 제어 경로를 보장하기 위해 MIPS64 계열의 마이크로 프로세서와 L2 스위치를 사용하였으며, RedBoot 및 실시간 운영체제인 FT-Linux를 사용하여 L2 스위치 구동 드라이버를 포팅하였다.

Control channel의 경로는 이중 성형 구조를 채택하여 Active channel의 불량 발생 시 여유 channel로 전환되어 서비스 될 수 있는 구조이며, Hub와 Non-Hub간의 역할 분담을 통한 최적화된 제어 경로를 제공한다.

본 기고문에서 설명한 ATCA 기반 광대역 라우터의 Embedded control channel 구조 설계는 라우터 설계 시 기본적인 각 블록의 요구 사항 만족은 물론 고 신뢰성의 프로세서간 제어 통신 경로를 임베디드화 하여 설계 함으로써 시스템 운영 시 발생할 수 있는 손실을 방지할 수 있도록 하였다.

[참고 문헌]

IV. 결 론

광대역망의 전달망 요소인 라우터에서 각각의 중요 프로세서간의 Control channel구현은 중요한 요

[1] PICMG 3.0 revision 1.0, "AdvancedTCA Base Specification," Dec. 30, 2002.
 [2] 정보통신 진흥 연구원, "IT 차세대 성장 동력 기획 보고서 (BcN)," 정보통신부, Dec. 2003.

- [3] Hak-Mook Lee, “ 광대역 IPv6 라우터의 10G 패킷 처리,” LG전자 Telecommuni- cation journal 제2권 제2호, June. 2005.
- [4] Jiun-Ji Liu, Chen, W.-S.E “High speed IP switching over the EtherSwitch,” 1999 International Workshop, pp. 548-553, Sept. 1999.
- [5] AN2579 Rev. 0.1, “Porting Linux to the MIPS,” Motorola semiconductor, Inc., Jan. 2004.
- [6] Preliminary data sheet, DS1.8, “Purple : Configurable Ethernet switch-on-a- chip,” Infineon Technologies AG, Oct. 2003.
- [7] Jenq-Nuh Hsu, “Design of an IPv6 SOHO Router Based on Embedded Linux System,” 19th AINA '05. IEEE, 2005.



이현철

1983년 광운대학교 전자공학 학사
2005년 광운대학교 공학 석사
1985년 ~ 현재 LG-노텔 시스템 H/W 개발실 책임 연구원 (팀장)



이학묵

1995년 충북대학교 정보통신 공학 학사
2005년 고려대학교 공학 석사
1995년 ~ 현재 LG-노텔시스템 H/W개발실 책임 연구원
