

MPLS 라우터 설계와 구현에서 네트워크 프로세서 사용의 경험적 고찰

김은아*, 전우직**

An Empirical Study on a Network Processor for a MPLS Router's Design and Implementation

Eunah Kim*, Woojik Chun**

요 약

인터넷이 급성장하면서 망 사용자들은 대역폭의 증가라는 양적인 측면 외에도 서비스 품질의 개선과 보장이라는 질적인 측면까지 요구하게 되었다. 점점 더 망의 기능성이 강조됨에 따라 망 장비의 성능과 기능도 중요하게 여겨지고 있는데, 특히 에지 라우터는 고성능, 다기능, 유연성이 모두 요구되는 시스템이다. 이러한 특성의 시스템 설계를 위해 최근 고려되고 있는 방법이 네트워크 프로세서를 사용하는 것이다. 본 논문에서는 ASIC의 고성능과 소프트웨어 기반의 유연성을 모두 제공하는 네트워크 프로세서를 사용하여 에지 급의 MPLS 라우터를 개발하고 그 결과를 기반으로 네트워크 프로세서의 향후 적용가능성을 살펴본다.

키워드: MPLS, 라우터, 네트워크 프로세서 (Network Processor)

ABSTRACT

The demands of network users emphasize the improvement and guarantee of service quality as well as the increment of bandwidth. As a result, high performance and additional new functions are important features to build network equipments, especially an edge router. For this structure, network processors with high performance and flexibility are considered as a main part of a packet forwarding module. In this paper, we design an edge MPLS router with a network processor, which supports high performance and multi-functionalities and examine its advantages and limitation

I. 서 론

인터넷의 폭발적인 증가와 보급은 대역폭(bandwidth)의 보장이라는 서비스의 양적인 측면뿐만 아니라 서비스 품질(Quality of Service: QoS)의 개선과 보장이라는 질적 측면의 중요성도 강조하게 되었다. 망 사용자들은 이미 T1/E1급의 속도를 넘어 T3/E3급으로의 선로 증설을 요구하고 있으며, 서비스 제공자들 역시 대역폭 요구의 증가를 예측하여 OC-N 형태의 공급 장치로 가입자들을 SONET/D SH의 매트릭스 링(metro ring)^[1]에 직접 연결하고 있다. 이러한 속도 및 대역폭의 증가와 더불어 부가

가치 IP 서비스의 부각은 요구되는 망의 특성이 기존의 최선형(best effort) 망에서 지능형(intelligent) 망으로 바뀌고 있음을 의미한다. 최선형 망에서는 망 장비들의 목적이 주어진 환경에서 최선의 방법으로 빠르고 신뢰성 있게 데이터를 전송하는 것에 중점을 두고 있는 반면, 지능형 망에서는 다양한 서비스 제공을 위해 데이터 트래픽(traffic)의 제어와 관리, QoS 보장을 포함하는 다기능에 중점을 두고 있다. 특히 현재 인터넷이 IP 망에서 구축되어 있는 상황을 고려한다면, 망 장비 중에서 라우터는 지능형 망으로의 이전에서 중요하게 고려되어야 할 장비이다. 그러나 기존의 전형적인 IP 라우터는 그

* (주)라오넷 (eakim@raonet.com), ** (주)라오넷 (chun@raonet.com)

논문번호 : 98244-0608, 접수일자 : 1998년 6월 8일

구현 방법과 라우팅 방법에서 이러한 요구 사항들을 만족시켜주기 어려운 근본적인 제약들이 있다.

초기 라우터의 구조인 GPP(General Purpose Processor)를 사용하는 소프트웨어 기반의 구조나 그 대안으로 제시된 ASIC(Application Specific Integrated Circuit) 기반 구조는 각각 유연성(flexibility)과 고성능이라는 장점을 제공해주지만, 두 특성의 통합된 기능을 제공해주는 데는 한계가 있다¹⁾. 또한 전형적인 IP 라우터에서 사용하고 있는 홉 단위(hop-by-hop)와 패킷 단위(packet-by-packet)로 LPM(Longest Prefix Matching) 메커니즘(mechanism)을 적용한 데이터 전송 방법은, 빠른 전송과 서비스 품질의 차별적인 제공이라는 측면에서는 제약사항을 가지고 있다. 이러한 한계를 극복하고자 기가 급 라우터를 위한 빠른 룩업(lookup) 메커니즘²⁾들이 연구되고 있으나 룩업 메커니즘의 보완만으로 IP 망의 근본적인 문제를 해결하기는 어렵다. 마지막 문제점은 기존 인터넷 망에서는 서비스 제공 정책으로 최선형 제공 방식을 사용한다는 것이다. IP 망이 비연결형(connectionless)으로 동작하고 있고, 잘 정의된 패킷별 차등처리 메커니즘이 없기 때문에 음성, 비디오와 같이 높은 대역폭과 실시간 전송을 요구하는 서비스들을 만족할만한 품질로 차별화된 서비스를 제공하는 것은 어렵다. 인터넷 표준화 기구인 IETF(Internet Engineering Task Force)에서는 IP 망에서 QoS를 제공하기 위한 방안으로 IntServ(Integrated Services)³⁾와 DiffServ(Differentiated Services)⁴⁾를 제안하였다. 그러나 IntServ는 플로우 단위 서비스를 제공하고 멀티캐스팅에 적합하다는 장점과 더불어 모든 라우터가 IntServ를 지원하는 망에서만 사용할 수 있다는 단점과 확장성(scalability)의 문제가 있고, DiffServ는 플로우(flow)들을 합병(agggregation)함으로써 확장성 문제를 해결했지만 플로우 양 종단간의 서비스 보장이 불가능하고 아직 완전한 모델로 정립되어 있지 않다는 문제점이 있다.

본 논문에서는 이러한 문제점들의 해결 방안으로 네트워크 프로세서(Network Processor: NP)⁵⁾를 사용한 MPLS 라우터의 설계와 구현에 대해서 다루었다. 첫 번째 문제의 성능과 유연성에 대해서는 네트워크 프로세서를 사용함으로써 ASIC 기반의 고성능과 소프트웨어 기반의 유연성을 모두 제공해주며, IP 망의 데이터 전송 문제의 대안으로는 MPLS (Multi Protocol Label Switching)⁶⁾의 데이터 전송 방식을 적용하였다. 세 번째 문제는 NP

LS의 FEC (Forwarding Equivalence Class)에 대응되는 레이블의 제공과 시작지와 목적지 사이의 LSP(Label Switched Path) 설정을 통한 자원의 할당과 유지, 비연결형 망에 대한 연결성 특성의 제공으로 해결하였다. 그 결과 앞서 제시했던 제약 사항들을 모두 개선시켜, 성능과 기능면에서 지능형 망에 적용이 가능한 10Gbps 급의 에지 급 라우터를 구현할 수 있었다. 이러한 시도는 네트워크 프로세서 적용의 시작 단계인 국내의 상황에서, 앞으로의 망 장비 개발에 네트워크 프로세서의 사용 가능성을 가늠할 수 있는 중요한 의미를 갖는다.

본 논문의 제 2절에서는 네트워크 프로세서와 MPLS에 대해 알아보고, 제 3절에서 라우터의 설계 기준과 구조에 대해 기술한다. 이를 바탕으로 제 4절에서 네트워크 프로세서를 사용한 라우터의 구현과 그 성능을 분석하고, 제 5절에서 결론 및 향후 연구 과제로 맺는다.

II. 관련 연구

1. 네트워크 프로세서

라우터를 설계하는데 있어서 중요한 것은 패킷을 처리하는 포워딩 엔진(forwarding engine)의 설계이다. 초기 네트워크 장비들이 GPP를 이용한 소프트웨어 기반으로 개발된 이후로 ASIP(Application Specific Instruction Processor), Co-processor, FPGA (Field Programmable Gate Array), ASIC 기반의 구현까지 다양한 구현 방법들이 제시되었다⁷⁾. 1990년대 후반까지 대부분의 망 라우터들은 GPP를 이용한 개인 컴퓨터와 비슷한 구조로 만들어졌는데, 이 구조에서는 CPU가 테이블 룩업, 접근 제어 필터링(filtering), 라우팅 정보 수정 등의 작업을 담당하였다. 또한 모든 명령어들이 소프트웨어로 저장되어 있어서 PC처럼 시스템 소프트웨어의 추가와 수정만으로 새로운 기능들의 간단한 추가가 가능하고, V.35나 HSSI(High Speed Serial Interface)를 지원하는 추가 인터페이스들의 설계도 기본 프로세서의 구조 수정 없이 쉽게 구현 가능하였다. 이와 같이 범용 CPU를 이용하여 설계하는 방식을 소프트웨어 기반 설계라고 하는데, 이 방식은 짧은 TTM (Time-to-Market)을 제공하고 새로운 프로토콜 표준의 추가와 망에서 요구되는 부가 기능들의 적용이 쉬워 유연성을 제공한다는 장점이 있다. 비록 이 방법이 작업의 양이 크고 복잡하며 상

대적으로 느리게 진행된다는 문제점이 있지만, 적은 규모의 이더넷(Ethernet), 토큰 링(Token Ring), 56Kbps 정도 속도의 WAN으로 이루어진 초기 망에서는 망의 요구사항을 만족시켜 줄 수 있었다^[2]. 그러나 망이 성장함에 따라 구조의 유지관리 비용이 높아지고 성능 면에서도 망 성능의 저해요소가 되어 그 한계를 나타내었다. 전형적인 소프트웨어 기반 라우터들의 패킷 처리 속도가 기가비트(Gigabit) 이더넷과 POS (Packet over SONET)와 같은 새로운 고속 기술들을 지원하기에는 충분하지 않았으며, 대부분의 소프트웨어 기반 라우터들은, 복잡한 수동 설정을 요구하여 망 관리가 점점 더 복잡해졌기 때문이다.

이의 대안으로 ASIC 기반의 방식이 소개되었다. ASIC 설계 방식은 하드웨어로 데이터 전송 작업을 처리하기 때문에 고속 라우터와 고속 스위치의 개발이 가능하지만, 새로운 기능들을 추가하기 위해서는 ASIC을 교체해야 하는 문제점이 있어 TTM의 지연이 증가하게 되었다. 또한 복잡한 QoS 라우팅, 상위계층 플로우의 인식, 보고, 정보 수집, 접근 제어 필터링과 같은 기능들을 처리하기에는 많은 한계를 가지고 있어서, ASIC 기반의 구현 방법은 보다 간단하고 고정된 기능을 요구하는 장치들을 만드는데 사용되었으며 ATM 스위칭이나 기본적인 이더넷/IP 스위칭과 같은 좁은 범위 내에서 응용들의 기능들에 적용되었다^[2].

그러나 망에 대한 요구사항이 서비스의 다양성과 질적, 양적인 면으로 확대되어 가는 현 시점에서는 양쪽의 어느 한 가지 방법만으로 적합한 설계 방법

표 1. 라우터 구현 방법의 특성 비교

	S/W 기반 라우터	ASIC 기반 라우터	NP 기반 라우터
Density	✓	✓	✓
성능/효율		✓	✓
Intelligence/QoS	✓		✓
새로운 서비스에 대한 적용력	✓		✓

것으로, 패킷 처리를 위한 특수 회로와 구조적인 특성을 갖추고 있으면서도 소프트웨어적으로 프로그래밍이 가능한 장치이다. 네트워크 프로세서는 그림 1과 같이 성능 면에서는 ASIC 수준의 와이어 급의 성능을 제공함과 동시에 기존의 GPP 기반 구조가 제공하는 유연성과 빠른 TTM을 제공하며, 전력소비량도 적어서 ASIC과 소프트웨어 기반 설계 방법의 경계에 위치하고 있다고 할 수 있다. 이런 네트워크 프로세서의 특성은 통신 시스템 디자인에 있어서 획기적인 접근방법을 제공하게 되었다. 표 1은 소프트웨어 기반 라우터, ASIC 기반 라우터, 네트워크 프로세서 기반 라우터의 특성 비교표이다^[8].

현재 대표적인 네트워크 프로세서들로는 Motorola의 C-5 DCP^[9], EZchip의 NP-1^[10], IBM의 Power NP^[11], Intel의 IXP1200^[12], AMCC의 nP^[13], Agere와 Lucent의 PayloadPlus(FPP/RSP/ASI)^[14] 등이 있다. 표 2는 이들 네트워크 프로세서의 특성을 비교한 것이다^[7]. 국내의 경우 LG와 삼성은 각각 Agere와 Intel의 네트워크 프로세서를 사용하여 장비를 개발 중이며, 외국의 Cisco, Juniper와 같은 대형 장비 업체들은 자체적으로 네트워크 프로세서 제품을 개발해서 사용하고 있다. 그 외의 상품화된 네트워크 프로세서를 사용한 업체들에 관한 정보가 잘 알려져 있지는 않지만, 다수의 업체들이 시도를 하고 있는 추세로 판단되며, 그 사용은 점점 더 증가할 것이다.

2. MPLS

MPLS는 레이블이라는 짧고 고정된 길이의 식별자를 사용하여 패킷 포워딩을 하는 L3 스위칭 기술이다^[6]. 레이블 값은 ATM 스위치에서의 VPI/VCI와 같이 해당 노드에서만 국부적인 의미를 갖는 값으로, 레이블 스위칭에 사용되는 라우터인 LSR (Label Switching Router)은 데이터 전송을 위해 단대단 경로로 LSP를 설정하고 레이블 스와핑(swapping)을 통해 패킷을 전송한다. 레이블을 동적으

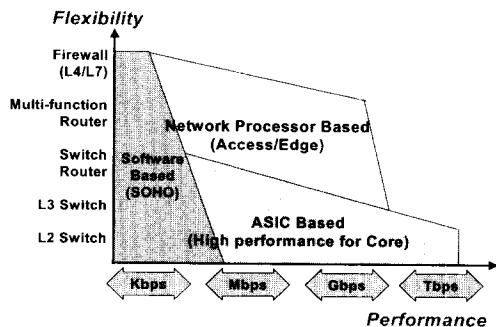


그림 1. 네트워크 프로세서의 특성을 제공할 수 없으므로, 기능과 성능을 모두 고려할 수 있는 방법이 모색되어야 한다. 이 요구에 부합되는 것이 최근 급성장하고 있는 네트워크 프로세서이다. 네트워크 프로세서는 ASIP 그룹에 속하는

표 2. 네트워크 프로세서들의 특성 비교

	Motorola (C-5 DCP)	EZchip (NP-1)	IBM (PowerNP)	Intel (IXP1200)	AMCC (nP)	Agere (PayloadPlus)
Capacity	50Gb/s	10Gb/s	8Gb/s	2.6Gb/s	5Gb/s	5Gb/s
OSI layers	2 ~ 7	2 ~ 7	2 ~ 5	2 ~ 7	2 ~ 7	2 ~ 4
Motul-PE	16 channel processors	64 task oriented processors	16 programmable protocol processor	6 programmable micro engines	No	3-FPP, RSP, ASI
Compilers	C/C++	C, Assembler	Assembler only	C, Assembler	C/C++	Yes, for FPP, RSP, and ASI
Libraries	CPI	None	None	None	Network software reference library	Application Code library with basic wire-speed

로 할당하고 관리하기 위한 신호(signaling) 프로토콜로는 IETF의 MPLS 워킹 그룹^[15]에서 위상 기반 라우팅 정보를 이용하는 LDP(Label Distribution Protocol)^[16]과 조건 기반 라우팅과 트래픽 관리에 기반을 둔 CR LDP(Constraint-based Routing LDP)^[17], RSVP-TE^[18]를 표준화하고 있다.

MPLS는 처음에는 빠른 포워딩을 제공하기 위한 방법으로 제시되었으나 근래에는 트래픽 엔지니어링(Traffic Engineering: TE)과 QoS 제공 등의 응용 쪽으로 연구가 진행되고 있다. MPLS의 레이블 스위칭 능력과 VC(Virtual Circuit)와 유사한 스위칭 기능, IP 경로 상에서 지속적으로 발생하는 룩업의 제거는 IP 기반의 망에 고속의 연결성과 IP 패킷 포워딩의 유연성과 더불어 레이블을 이용한 FEC 별 패킷 처리로 미세한 QoS의 차별을 제공할 수 있게 되었다. 그러므로 고성능과 다기능이 모두 요구되는 에지(edge) 급 라우터에서 MPLS를 이용하는 것은 필수적이며, 제 2계층과 제 3계층의 다양한 프로토콜에 적용 가능한 MPLS의 특성은 그 적용범위의 폭이 넓어 MPLS 응용의 제공은 부가가치 창출을 위한 응용서비스 생성의 첫 출발이라고 할 수 있다.

III. 시스템 설계

1. 설계 목표

다양한 망의 연결 서비스와 향상된 대역폭의 제공을 비롯하여 다양한 서비스들의 요구가 증가함에 따라 서비스 제공 망은 더욱 기능과 성능 면에서 복잡해지고 있으며, 특히 합병(aggregation) 기능과 지능형 기능들이 요구되는 에지에서는 기능과

성능이 모두 중요요건이 되고 있다. 또한 에지 망에서는 망의 가용성(availability), 유연성, 보안성(security), 빠른 서비스 적용력, 저렴한 비용 등의 사용자 요구사항^[19]을 수용할 수 있어야 한다. 에지 라우터는 망 사용자들의 서비스 질에 대한 요구사항과 망 서비스 제공자들의 트래픽 제어에 관한 요구사항이 맞물리는 지점이기 때문에, 대용량의 고속 라우팅을 주 관건으로 삼는 코어(core) 망 장비와는 달리 성능과 다기능 면에서 그 중요성이 점점 더 강조되고 있다. 표 3은 코어 라우터와 에지 라우터의 특성을 비교한 것이다^[20].

본 논문에서는 이와 같은 다기능 에지 급의 차세대 라우터의 설계와 구현을 위해 다음과 같은 설계 기준을 정의하였다.

표 3. 에지 라우터와 코어 라우터의 비교

항목	에지 라우터	코어 라우터
Data Rate (packets)	2~4백만	30+백만
물리적인 크기	Small size	Room size
Environment	Heterogeneous	Heterogeneous
프로토콜	Ethernet, ATM, T1	SONE, ATM
구현	Network processor	Dedicated Switch
요구되는 기능	합병, 트래픽별 관리 및 제어	대용량 전송 빠른 전송

1) 기가급 와이어 속도의 패킷 처리 기능

망 사용자의 데이터가 텍스트 데이터 중심에서 음성과 동영상 같은 다양한 멀티미디어 데이터로 변화하고 있고, LAN과 WAN의 속도가 지속적으로 향상됨에 따라 충분한 대역폭의 확보와 함께 패

킷의 빠른 처리 능력은 라우터의 주요 기능이 되었다. 특히 가입자들의 트래픽을 모아 코어 망으로 전달하는 에지 라우터는 빠른 패킷 처리 능력이 필요하다.

2) QoS 제공 기능

다양한 서비스들이 새로 정의되고, 사용자들이 서비스에 따른 자원들의 할당과 보장을 요구함에 따라 QoS는 사용자의 중요한 서비스 이슈가 되고 있다. QoS 보장을 위해서는 사용자, 트래픽, 링크, 응용의 종류 등에 따른 패킷의 구별과 제어, 자원의 할당과 유지 등의 기능들이 요구된다.

3) MPLS 및 그 응용의 수용

MPLS는 최근 들어서 고속 라우팅과 트래픽 엔지니어링을 위한 방안으로 연구되고 있는 방법 중 하나로, 기존 IP 망에 연결형 설정을 제공해주고, 레이블을 이용하여 각 데이터 패킷들에 대한 차별적인 서비스를 가능하게 해준다. 그러므로 라우터의 성능과 기능 면을 고려할 때에 MPLS의 수용은 필요하며, MPLS를 이용한 다양한 응용서비스를 수용할 수 있는 기능도 부가적으로 포함시켜야 할 것이다. MPLS와 그 응용들의 수용을 위해서는 MPLS 신호 프로토콜과 응용에서 사용할 수 있는 MPLS API의 개발도 필수적이다.

4) 서비스 적응형

서비스 적응형(adaptive)이란, 망 사용자들의 다양한 요구사항에 따른 다양한 기능들의 제공과 더불어 이러한 서비스들을 망에 쉽게 적용시킬 수 있는 유연성을 의미한다. 새로운 서비스들과 기능들을 기존의 라우터에 짧은 기간 내에 적용시키는 것은 쉽지 않으며, 변화되는 조건과 상황에 따라 새로운 시스템을 개발하는 것도 유연성의 문제점을 갖는다. 그러므로 단기간 내에 간단한 방법으로 변경된 서비스나 새로 정의된 서비스를 라우터에 쉽게 적용시킬 수 있는 방법과 서비스 적응력이 중요하다.

2. 시스템 구조

그림 2는 네트워크 프로세서를 이용하여 설계한 시스템을 간략하게 그린 것이다. 관리 평면(management plane)에서는 시스템을 관리하고 환경을 설정할 수 있는 기능을 제공하며, 제어 평면(control plane)에서는 라우팅 프로토콜들과 MPLS 신호 프로토콜을 통해 FIB(Forwarding Information Ba

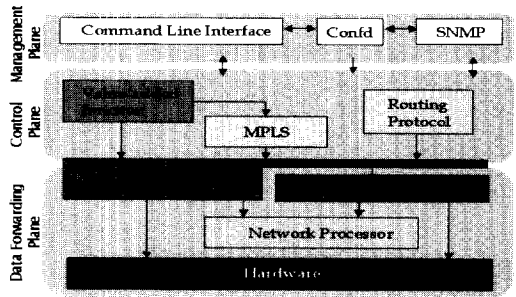


그림 2. 네트워크 프로세서를 사용한 MPLS 라우터 시스템의 구조

se)와 LIB(Label Information Base) 및 데이터 전송과 시스템 제어에 관련된 정보들을 생성한다. 데이터 포워딩 평면(data forwarding plane)에서는 FIB와 LIB 정보를 참조하여 IP 패킷과 MPLS 패킷을 네트워크 프로세서에서 처리한다.

3. 소프트웨어 구조

소프트웨어 모듈은 그림 3과 같이 관리 모듈, 라우팅 모듈, 커널(kernel) 모듈, 마이크로코드(microcode) 모듈로 이루어져 있다.

1) 관리 모듈

관리 모듈은 CRM(Control Routing Module)에서 실행되며, clish, confd, channeld의 세 부분으로 나뉜다. Clish(Command Line Interface Shell)은 사용자의 입력 명령어를 받아들여서 번역하고 직접 필요한 동작을 수행하거나 다른 모듈들이 연산할 수 있도록 알려주며, Confid(Configuration Daemon)는 시스템의 환경 설정을 저장하고 각 응용

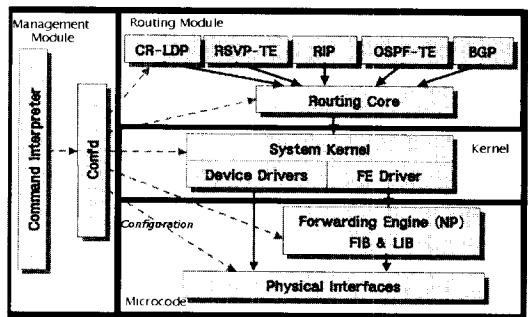


그림 3. 시스템의 소프트웨어 모듈의 블록 다이어그램

들에게 설정된 상태를 제공해준다. 본 라우터는 시스템의 안정화를 위해 두 개의 CRM(활성 CRM과

대기 CRM)을 사용하는데, Channeld(Channel Daemon)는 시스템의 이중화(redundancy)를 위한 것으로 활성 CRM과 대기 CRM 모두에서 기본으로 동작한다. 또한 커널 모듈이 제공하는 채널 드라이버를 사용하여 CRM의 상태를 감시하고, 활성 CRM에서 대기 CRM으로의 데이터 백업 동작과 활성 CRM의 고장이 발견되었을 때 Fail over 동작을 취하는 기능도 수행한다. 이 외에도 시스템이 구동될 때 마이크로코드를 다운로드해주는 유틸리티 등과 시스템 로그를 관리하는 rlogd와 Syslogd(System Logging daemon)도 관리 모듈에 속한다.

2) 라우팅 모듈

라우팅 모듈은 라우팅 프로토콜 데몬, MPLS 신호 프로토콜 데몬, Rcore(Routing core) 데몬으로 이루어져 있다. Rcore는 그림 4와 같이 라우팅 프로토콜들과 MPLS 신호 프로토콜들이 제공하는 포워딩 정보와 레이블 정보들의 처리 기능, 그 결과를 커널과 포워딩 엔진에 적용하는 기능과 시스템에서 발생하는 링크의 변화를 커널로부터 전달받아 관련 루트에 반영시키는 작업을 한다.

라우팅과 레이블 정보는 커널과 포워딩 엔진에서 각각 FIB와 LIB로 관리된다. 본 라우터에서 포워딩 엔진의 FIB는 64K의 엔트리를 유지하도록 설계되었으며, LIB에서 관리되는 레이블의 범위는 1~65,535로 레이블 범위는 플랫폼 기반^[6]으로 관리된다. 포워딩 엔진에서는 그 외에도 LIB와 FIB에서 참고로 사용할 NIB(Neighbor Information Base)를 관리하며, ATM PIM(Physical Interface Module)에는 ATM에서 사용되는 VPI/VCI 값의 PVC 테이블도 유지한다.

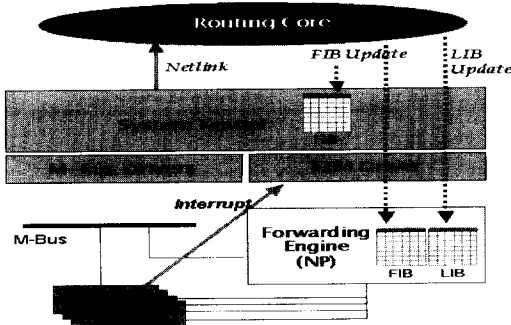


그림 4. 라우팅 모듈의 소프트웨어 구성도

3) 커널 모듈

커널 모듈에는 시스템 동작에 필요한 기본 기능,

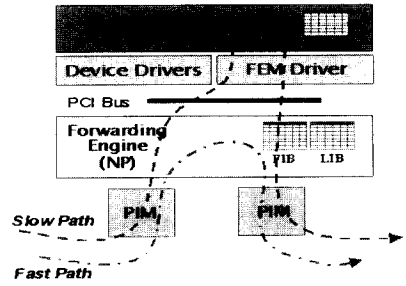


그림 5. 커널 모듈 및 I/O의 제어 구조

CRM 내의 포워딩 테이블을 유지, FEM(Forwarding Engine Module)과 함께 'slow path'를 구현하는데 필요한 디바이스 드라이버가 포함된다. 이 외에도 시스템의 각 보드 제어 및 상태 보고를 위해 사용되는 M-Bus 드라이버, 이중화 지원을 위한 채널 드라이버 등이 포함된다. 각 인터페이스에 도달한 패킷이 단순 IP 패킷이나 MPLS 패킷인 경우에는 그림 5와 같이 네트워크 프로세서에서 유지하고 있는 FIB와 LIB 정보를 이용하여 FEM에서 포워딩 기능을 수행하는 fast path로 전송된다. 그러나 패킷 분열(fragmentation)이 필요한 경우, IP 패킷에 옵션이 사용된 경우, 전송 경로를 모르는 경우에는 FEM 내부에서 패킷의 실시간 처리가 어려우므로 slow path를 통하여 CRM으로 전달되고, CRM의 커널 모듈이 포워딩 기능을 수행한다.

4) 마이크로코드

마이크로코드는 FEM 보드 내의 네트워크 프로세서에서 수행되며, 시스템 시작 시에 관리 모듈에 의해 FEM 보드로 다운로드 된다. 네트워크 프로세서의 초기화, CRM과 slow path를 담당하는 XP (Executive Processor) 코드, 채널 프로세서(Channel processor: CP)의 코어용 코드와 채널 프로세서 내의 Bit/Byte Rx/Tx 동작을 수행하는 SDP (Serial Data Processor) 코드가 있다.

4. 포워딩 엔진의 설계

IP 패킷과 MPLS 패킷을 모두 처리하는 포워딩 엔진은 그림 6과 같이 설계되었다. 채널 프로세서는 대역폭의 크기에 따라 하나 이상이 모여서 하나의 물리적 인터페이스도 동작하기도 하고, 하나의 채널 프로세서가 여러 인터페이스 포트를 지원할 수도 있다. TLU(Table Lookup Unit)는 포워딩에 관련된 테이블을 유지 관리하는 기능을 제공하며, 패킷을 포워딩할 때 테이블 룩업 기능을 제공한다.

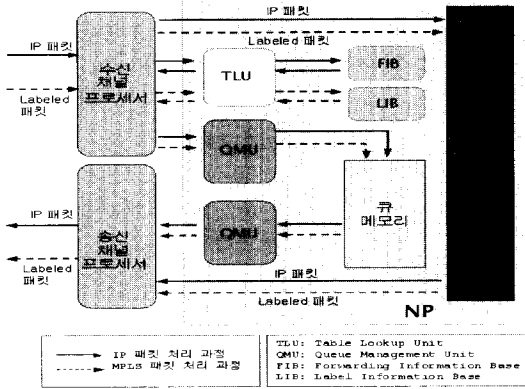


그림 6.포워딩 엔진에서 IP 패킷과 MPLS 패킷의 처리 과정

QMU (Queue Management Unit)는 응용에서 정의한 포워딩 서술자(descriptor)를 채널 프로세서와 주고받는 기능을 제공한다.

패킷이 전송되는 경우, 수신 채널 프로세서는 패킷의 'Type' 필드를 검사하여 IP/MPLS 패킷을 구별하고, IP 패킷인 경우에는 TLU가 목적지 IP로 FIB를 룩업하여 포워딩 정보를 받아 오고, 그 동안 패킷의 페이로드(payload)는 버퍼 메모리에 저장된다. 만약 포워딩 정보를 발견하지 못하면, ICMP "Destination Unreachable" 메시지를 패킷의 시작지로 보낸다. 만약, 목적지가 라우터 자신인 경우에는 패킷은 라우터 시스템의 프로토콜 스택으로 전달되며, 그렇지 않은 경우에는 FIB로부터 얻은 포워딩 관련 정보를 패킷 서술자를 담아서 출력 인터페이스 큐로 전달한다. 송신 채널 프로세서는 큐로부터 포워딩 정보를 가져오고, 이 정보를 이용해서 버퍼 메모리에 있는 패킷의 수정 과정을 거친 후 전송하게 된다. 이 수정 과정의 예로는 IP TTL의 변경, 목적지 MAC과 시작지 MAC의 변경 등이 있다.

MPLS 패킷인 경우에는 IP 주소 대신 레이블 값으로 LIB를 룩업한다. 그러나 Ingress LER(Label Edge Router)에서는 레이블 룩업 대신 IP 룩업을 통해서 출력 인터페이스 정보와 다음 홉으로 전달될 레이블 정보를 얻어오며, 송신 채널 프로세서를 통해 MPLS 패킷이 전송된다. 그와 달리 Egress LER에서는 레이블로 룩업을 시도하고, 그 결과에 따라 IP 패킷이나 MPLS 패킷(LSP의 계층적 구조인 경우)이 전송된다.

실제 송신 채널 프로세서나 수신 채널 프로세서는 모두 물리적인 인터페이스에 위치하며 모든 인

터페이스에서 처리 과정들이 발생한다. 포워딩 엔진에서 룩업 방법은 성능 결정의 주요 요소이므로 하나의 패킷을 처리하는데 필요한 룩업 횟수를 최대한으로 줄이는 것이 중요하다. 본 설계에서는 IP 패킷 처리나 LER의 MPLS 패킷 처리에서 발생할 수 있는 회귀적 룩업(recursive lookup)을 한 번의 룩업으로 처리하기 위해 Rcore에서 최적화된 경로를 FIB와 LIB에 넣어주며, FIB에는 Ingress LER에서 사용될 레이블 정보를 같이 넣어주었다.

IP 패킷이 MPLS 패킷으로 수정되어 전송될 때 Shim 헤더^[21]가 추가되는데, 본 라우터에서는 레이블 스택^[21]의 크기(depth)를 2로 설계하였다. 레이블 스택을 사용하면 계층적인 MPLS 망의 설정이 가능하다는 장점도 가지고 있으나 MPLS 라우터의 처리 면에서 보면 망의 에지에서 룩업 발생 횟수와 관련되어 성능에도 영향을 주므로 사용할 응용에 따라서 크기를 미리 정의하여 룩업을 단순화시켰다.

입력 패킷이 ATM 셀인 경우에는 레이블은 VPI와 VCI 필드의 조합^[22]으로 표시하고, LC-ATM^[2]을 통해서 송수신되는 패킷들은 MPLS 패킷으로 인식하도록 하는 방법을 사용한다. ATM 셀은 그림 7과 같이 AAL5 헤더에 shim 헤더를 추가한 후 이것을 셀로 만들어서 전송한다. 이 경우 shim 헤더의 최상위 레이블 필드는 0으로 설정되고, 실제 레이블 값은 VPI/VCI 필드에 표시된다.



그림 7.ATM 셀의 레이블 인코딩

IV. 시스템 구현

1. 구현 환경

라우터에 새로운 기능들을 구현하기 위해서는 운영체제 커널의 수정은 필수적이다. 본 라우터의 운영체제로는 안정성 있게 잘 구현되어 있고 소스 코드가 공개되어 있는 Linux의 여러 버전 중 Hardhat^[23] 커널 버전 2.4.2를 선택하여 인터페이스의 상태와 라우팅 정보 상태 관리를 위한 Netlink, 응용 프로그램들과의 인터페이스, 각종 드라이버 등을 라우터의 설계에 기반하여 수정과 추가하였다. 시스

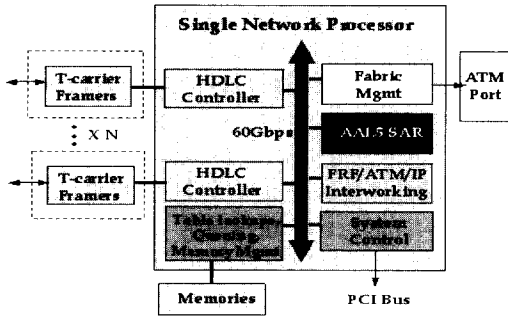


그림 8. 네트워크 프로세서를 사용한 인터워킹 설계

템 전반에 걸친 프로그래밍은 C언어와 GNU C 컴파일러인 gcc를 사용하였다. 포워딩 엔진은 Motorola사의 C-5 DCP^[9]를 사용하여 구현하였다. DCP는 제 2계층에서 제 7계층에 이르기까지 모든 계층의 프로토콜과 다양한 인터페이스를 통한 패킷, 셀, 데이터 스트림 등의 적용이 가능하고, 와이어 속도와 다양한 기능들을 제공하므로 에지급 라우터의 포워딩 엔진 구현에 사용하기 적합한 네트워크 프로세서이다. 그림 8은 C-5를 이용한 ATM 인터워킹 설계이다. GPP를 사용하는 전형적인 라우터에서는 HDLC controller, system control, AAL5 SAR, PCI 버스가 모두 분리되어 설계^[2]되지만, C-5는 고수준의 네트워크 기능 통합 특성을 가지고 있으므로 시스템의 설계가 간단해지고 각 요소가 분리되어 구현되는 전형적인 라우터에서 발생할 수 있는 병목현상도 회피할 수 있게 된다. 또한 개발자들에게 친근한 C와 C++ 언어를 이용한 프로그램이 가능하므로 개발이 쉽고, 코드의 재활용성이 좋다. DCP에서 SDP관련 부분은 마이크로코드로 작성해야 하지만, Motorola에서는 마이크로코드도 gcc에서 컴파일 가능하도록 수정된 gcc를 제공하고 있어

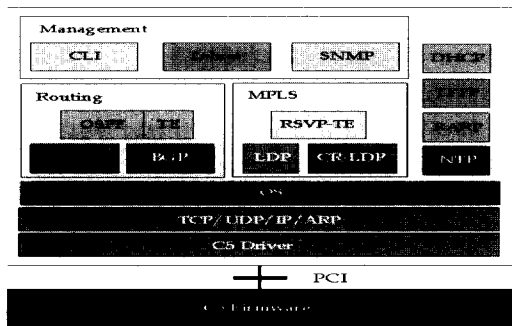


그림 9. 개발된 라우터의 프로토콜 스택서 개발이 편리하다.

라우팅 프로토콜은 공개된 라우팅 프로그램 중에서 Zebra^[24]를 선택하여 Rcore 데몬, Confd, CLI, SNMP 등과의 인터페이스 및 OSPF의 Opaque^[25], NSSA^[26]를 포함한 새로운 기능들을 추가하였다. Zebra는 C언어로 작성되어 있어서 코드의 이해와 수정, 추가가 용이하며, 국내에서 가장 많이 사용되고 있는 Cisco의 라우터와 CLI 체계가 유사하게 작성되어 있어서 사용자에게 친밀감을 주는 장점이 가지고 있다. 라우팅 프로토콜의 운용 버전은 RIP 버전 2^[27], OSPF 버전 2^[28], BGP 버전 4^[29]이다. MPLS 신호 프로토콜로는 LDP^[16]와 CR-LDP^[17], RSVP-TE^[18]를 C언어로 구현하였으며, CR LDP 및 RSVP-TE와 라우팅 프로토콜간의 연동을 위해 OSPF-TE^[30]를 구현하였다. 그 외로 라우터의 관리 및 운용을 위해서 CLI, Telnet 세션, 웹 기반의 관리 인터페이스, SNMP 등을 개발하였으며, 트래픽 제어, 응용관리, 시스템 상태관리를 위해 로깅(logging) 기능도 추가하였다.

2. 시스템의 성능 분석

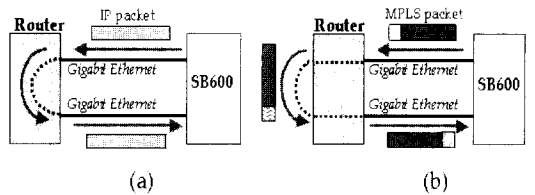


그림 10. IP 패킷(a)과 MPLS 패킷(b) 전송 성능 시험 환경

그림 10은 개발된 라우터와 기가비트 이더넷으로 연결된 SmartBit 600 장비에서 각각 IP 패킷과 MPLS 패킷의 트래픽을 발생시켜, 라우터의 처리 성능을 시험한 것이다. 기가비트 인터페이스는 Rx와 Tx 모두 1G의 대역폭을 갖는다. 라우터의 패킷 처리 성능은 입력 패킷 수에 대한 출력 패킷 수의 비율로 계산되었으며, (a)와 (b)는 각각 IP 패킷과 MPLS 패킷의 처리 성능(레이블 스위칭 처리)을 시

표 4. IP 패킷과 MPLS 패킷의 처리율(%) 비교표 (출력 패킷 수 / 입력 패킷 수)

입력패킷크기 (bytes)	64	78	84	86	92	94	1518
IP 패킷	81.45	94.40	100.0	100.0	100.0	100.0	100.0
MPLS 패킷	75.32	91.30	96.75	98.57	100.0	100.0	100.0

험한 것이다. 시험은 이더넷의 최소 프레임 크기인

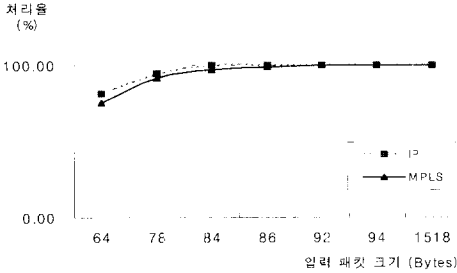


그림 11. IP 패킷과 MPLS 패킷 처리 효율 비교 그래프

64바이트부터 최대 크기인 1,518바이트까지 시험하였으며, 패킷의 수는 SmartBits 600에서 발생시킨 패킷 수와 라우터로부터 처리 후 출력된 패킷 수를 SmartBits 600의 프로그램에서 계산하였다.

표 4에 의하면, IP 패킷은 패킷 프레임의 크기가 84 바이트가 되면 100% 처리율에 도달하고, MPLS 패킷의 경우에는 92 바이트에서 100% 처리율에 도달한다. 두 패킷에 대해 처리율이 다른 이유는 IP 패킷의 경우 시스템에서 인터페이스별 통계정보만을 계산하는데 반해, MPLS 패킷은 인터페이스별 통계와 더불어 LSP 별 통계정보를 처리하므로 그만큼의 처리부담이 있기 때문이다. MPLS 패킷을 처리할 때, LSP 별 통계정보 처리 과정을 제거하면, 성능은 IP 패킷과 같은 곡선을 그리게 된다. 이것은 IP 패킷 처리 성능이 MPLS 패킷의 처리 성능과 같다는 것인데, 이것이 바로 네트워크 프로세서 사용의 성과이다. 소프트웨어 기반 라우터에서는 MPLS 패킷 처리보다 IP 패킷 처리의 성능이 떨어

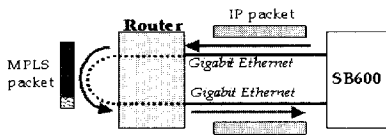


그림 12. 라우터가 LER로 동작할 때의 패킷 처리 효율 시험 환경

진다.

그림 12는 SmartBits 600에서 IP 패킷을 전송시키고, 라우터에서 레이블의 추가와 제거가 발생하는 LER의 경우에 대한 패킷 처리 효율 시험이다.

표 5를 보면, 라우터가 LER로 사용될 때, 입력 패킷에 대한 출력 패킷의 처리효율이 떨어지는 것을 볼 수 있다. 그러나 이 경우에 한 가지 고려해야 할 사항은 라우터가 IP 패킷을 받은 후, 레이블을

표 5. LER로 동작할 때의 MPLS 패킷 처리 효율 표 (P_o: 출력 패킷 수, P_i: 입력 패킷 수)

입력패킷크기 (bytes)	64	84	86	90	1514
P _o /P _i (%)	75.18	91.20	94.80	96.50	99.73
MPLS 패킷의 처리 성능(%)	82.34	95.00	98.50	100.00	100.00

추가하므로 전체적으로 패킷의 크기가 4바이트만큼 늘어난다는 것이다. 그러므로 LER의 성능을 볼 때에는 IP 패킷이 MPLS 패킷으로 바뀌어서 처리된 결과(P_o/P_i)뿐 아니라 내부에서 MPLS 패킷으로 변경된 패킷에 대한 처리효율을 보는 것도 중요하다. 이를 위해 수식(1)처럼 IP 패킷에서 MPLS 패킷으로 변경되어 사용될 패킷의 수를 구한 다음 이 값과 출력 패킷의 수를 비교하여 'MPLS 패킷의 처리 성능'을 계산했다.

$$Number\ of\ MPLS\ packet = \frac{band\ width}{input\ packet\ size + shim\ header\ size + gap} \quad (1)$$

'input packet size'는 SmartBits 장비에서 생성한 패킷의 크기이고, 'shim header size' 값은 IETF 표준 문서인 RFC 3032^[21]에서 정의하고 있는 4바이트이며, 'gap'은 MAC Preamble 8바이트와 Inter Frame Gap 12바이트를 합인 20바이트이다. 이 계산식에 의한 MPLS 패킷 수와 SmartBits에서 확인된 출력 패킷 수를 계산한 결과 입력 패킷 크기가 90 바이트인 경우 100%의 성능에 도달했으며, 입력된 IP 패킷이 MPLS 패킷으로 바뀌어서 처리되는 패킷의 수도 수식(1)로 구한 계산적인 패킷의 변경 수와 일치함을 알 수 있었다.

구현된 시스템의 성능은 표 6에 정리되어 있다. 본 라우터의 가장 중요한 특성은 네트워크 프로세서의 사용으로 인한 고성능과 유연성의 획득이다.

표 6. 구현된 라우터의 성능

항목	값
병합(aggregation) 대역폭	10 Gbps
IP 패킷 전송율	10 Mbps
전송 엔진 성능	6000 MIPS
lookup 엔진 성능	266M lookup/sec
라우팅 엔트리수	64K
레이블 범위	1 ~ 65,535

두 번째 특성은 MPLS의 제공으로, MPLS를 사용하면으로써 패킷의 출발지와 목적지 사이의 경로에 위치하는 라우터들은 LSP를 설정할 때에만 IP 룩업을 수행하면 되므로 패킷과 홉 별로 룩업이 발생하는 IP 패킷에 비해 빠른 패킷 전송 처리가 가능하다. 또한 레이블로 패킷들을 구분하고 처리할 수 있으므로 패킷의 차별적인 처리가 가능하다. 신호 프로토콜을 이용하여 조건에 따른 LSP를 설정하면, 동적인 트래픽 제어와 QoS 제공의 기반을 마련할 수 있다. 또한 레이블 스택을 사용하므로 VPN (Virtual Private Network)^[31] 서비스의 응용이 가능하다. 본 라우터는 IP 응용과 MPLS 응용들을 모두 적용 가능하며, 이를 위해서 MPLS API를 개발하였다. MPLS API는 LSP 설정과 MPLS 응용을 위한 레이블 정보의 제어 및 관리에 필요한 인터페이스를 제공한다.

표 7은 개발된 라우터와 기존의 전형적인 라우터를 비교한 것으로, 처리성능은 ASIC 기반 구현의 수준을 나타내고 유연성 면에서는 소프트웨어 기반의 특성을 보여주고 있다. 이러한 유연성은 라우터의 개발 기간을 단축시킴으로써, 사용자에게 다기능의 저렴한 가격의 라우터를 제공할 수 있게 해준다.

본 시스템의 물리적 인터페이스로는 Fast Ethernet, Gigabit Ethernet, E1 ATM, OC-3 ATM을 개발하여 다양한 환경의 망 가입자들을 포함시킬 수 있는 기능도 제공하고 있으며, 논리적 인터페이스로는 룬백 인터페이스와 터널(tunnel) 인터페이스를 구현하였다. 룬백 인터페이스는 라우팅 프로토콜을 설정할 때 가상의 망을 대표하도록 사용할 수 있고, 터널 인터페이스들은 물리적 인터페이스와 연결되며 대역폭과 QoS 파라미터들을 정의해줄 수 있어서, VPN 설정 인터페이스나 트래픽 제어용 인터페이스로 사용하기에 적당하다. 그 외의 주요 기능으로는 동적인 환경 설정, hot-swap 가능한 모듈의 제공, 이중화 설정, 부하 분산, Virtual LAN 등이 있다.

이러한 특성으로 인해 본 라우터는 터널링, CoS (Class of Service), QoS, 로드 분할(load sharing), 차선 경로 선택(alternative path selection), 경로 복구(path restoration), 빠른 복구, 트래픽 엔지니어링, VPN 등에 응용이 가능하며, 서비스 망에서 서비스 제공자 쪽의 통합 라우터나 가입자의 에지 라우터로 모두 사용할 수 있다. 통합 라우터로 사용될 때에는 다양한 종류의 인터페이스와 MPLS의 지원으로 인해 다양한 가입자들과 트래픽

표 7. 네트워크 프로세서를 사용하여 개발된 라우터와 전형적인 라우터의 비교

	NP 기반 라우터	ASIC 기반 라우터	S/W 기반 라우터
유연성	높음	낮음	높음
성능	좋음	좋음	상대적으로 낮음
처리속도	10Gbps	Gbps~Tbps	Kbps~Mbps
MPLS 기능	가능	가능	가능
패킷의 차별적인 처리	보장	보장	보장
가격	저가	고가	저가

들을 처리할 수 있는 기능을 제공하며, 에지 라우터로 동작할 때에는 고속 인터페이스, 통합 기능, 정책 라우팅 등을 제공한다.

V. 결론 및 향후 과제

인터넷의 폭발적인 증가와 다양한 서비스의 출현은 지금까지의 최선형 망을 지능형 망으로 변화시키고 있으며, 그 결과 사용자 단말기로부터 전달된 데이터를 목적지로 전송하는 역할이 주된 기능이었던 라우터는 성능 뿐 아니라 정교한 QoS의 제공과 다양한 기능을 필요하게 되었다. 본 논문에서는 라우터 중에서 성능과 기능 면이 모두 강조되는 에지 라우터의 설계와 구현에 네트워크 프로세서를 사용하여, 네트워크 프로세서 사용의 가능성을 살펴보았다. 본 라우터는 기가 급의 와이어 속도 패킷 처리 능력과 다기능 및 용이한 서비스 적용력, MPLS 및 응용의 수용이라는 설계 기준을 목표로 하여 네트워크 프로세서와 MPLS 기술을 적용함으로써 개발되었다. 구현된 라우터는 통합 대역폭이 10Gbps인 에지 급 라우터로, 네트워크 프로세서의 특성인 ASIC 기반 라우터와 소프트웨어 기반 라우터가 제공하는 고성능과 유연성을 모두 제공하게 되었다. 그 결과 다양한 서비스의 수용이 용이하게 되었으며 개발 기간의 단축과 짧은 TTM을 제공할 수 있게 되었다. 또한 MPLS 기능의 사용으로 기존의 IP 망에서도 차별적인 데이터 처리를 통한 QoS 보장 및 트래픽 엔지니어링의 기반을 제공할 수 있는 서비스 적용성을 제공함으로써 사용자들은 필요에 따른 차등화된 서비스의 사용이 가능하게 되어, 망 서비스 제공자의 입장에서도 다양한 서비스의 지원

과 다양한 환경의 망 가입자들을 영입할 수 있는 장점이 있다.

네트워크 프로세서는 앞서 살펴본 바와 같이 필요한 기능을 프로그램 함으로써 소프트웨어 기반의 유연성을 제공하고, 네트워크 계층의 기능을 ASIC으로 구현함으로써 ASIC 기반의 성능을 제공해줄 수 있다는 것이 가장 큰 장점이다. 그러나 네트워크 프로세서는 그 제품마다 다양한 특성과 성능을 가지고 있으므로 망 장비의 설계를 고려할 때에는 적합한 네트워크 프로세서를 선택하는 것이 장비의 성능과 특성을 결정짓는 중요한 요인이 된다. 네트워크 프로세서 중에는 C나 C++와 같이 일반화된 언어로 프로그래밍할 수 있는 제품들도 있지만, 여전히 시스템의 성능과 관련된 부분들은 어셈블리어로 작성해야 하는 단점을 가지고 있다. 개발자들을 위한 API들도 제공되고 있지만, 그 범위가 제한적이고 그 외의 기능이나 추가되어야 할 것들은 개발자가 개발해야 하는 몫이기 때문이다. 네트워크 프로세서의 가격이 비교적 고가라는 것도 시스템의 개발비용을 높게 만드는 문제를 가지고 있다. 네트워크 프로세서는 성능과 유연성 사이에 trade-off를 가지고 있으므로, 이들의 최적화가 가장 중요한 문제가 된다. 현재는 네트워크 프로세서 개발의 초기 단계라 고려되어야 할 사항들과 해결해야 할 문제들이 많지만, 차세대 네트워크 장비의 구현에 있어서 네트워크 프로세서의 사용은 불가피하며, 그 적용은 점점 더 증가할 것이다. 그러므로 네트워크 프로세서를 사용한 시스템의 설계와 적합한 선택의 기준을 위해 각 네트워크 장비들의 특성에 따라 고려되어야 할 네트워크 프로세서의 성능과 기능의 기준 안을 마련하는 것도 향후에 연구과제로 중요하다. MPLS 라우터에서는 어떤 성능과 기능들을 제공해줄 수 있는가 뿐 아니라 어떤 서비스들을 사용할 수 있는가의 여부도 중요하다. 그러므로 에지 라우터에 대한 향후 연구과제로 효율적인 MPLS TE^[32] 방법의 연구도 중요하다. 이를 위해 OSPF-TE 외에 IS-IS를 이용한 TE^[33] 연동방안, 대역폭의 할당 매커니즘, 트래픽 제어 및 관리 매커니즘의 연구가 진행되어야 한다. 또한 VPN과 VoMPLS^[34] 등을 포함하여 다양한 MPLS 응용분야에 대한 연구도 요구된다.

참 고 문 헌

- [1] M. Veeraraghavan, H. Lee, J. Anderson, and K.Y. Eng, "A network through comparison of optical metro ring architectures," Optical Society of America, 2001.
- [2] D. Husak, "Network Processors: A Definition and Comparison," White Paper, C-PORT.
- [3] N-F. Huang, S-M. Zhao, J-Y. Pan, and C-A. Su, "A Fast IP Routing Lookup Scheme for Gigabit Switching Routers," *Proc. IEEE INFOCOM*, pp.1429-1436, March 1999.
- [4] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," IETF RFC 2210, September 1997.
- [5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.
- [6] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF RFC 3031, January 2001.
- [7] N. Shah, "Understanding Network Processors," September 2001.
- [8] "Network Processors: The Optimal Building Block for Next Generation IP Routers," Technical Paper, Alcatel, November 2000.
- [9] "C-5 Digital Communications Processor," White paper, C-Port Corporation.
- [10] "Network Processor Designs for Next-Generation Networking Equipment," White paper, EZchip Technologies.
- [11] J. Allen *et al*, "PowerNP Network Processor Hardware, Software and Applications," White Paper, IBM.
- [12] "Intel IXP1200 Network Processor," Datasheet, Intel.
- [13] "nP family Network Processor," Data sheet, Applied Micro Circuits Corp.
- [14] "The Challenge for Next Generation Network Processors," White Paper, Agere, Inc.
- [15] IETF MPLS Charter, <http://www.ietf.org/html.charters/mpls-charter.html>.
- [16] L. Andersson, P. Doolan, N. Feldman, A.

Fredette, and B. Thomas, "LDP Specification," IETF RFC 3036, January 2001.

[17] B. Jamoussi *et al*, "Constraint-Based LSP Setup using LDP," IETF RFC 3212, January 2002.

[18] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF RFC 3209, December 2001.

[19] "Building The Optimal Edge Network," White Paper, TeleChoice, February 2002.

[20] M. Attia and I. Verbaughede, "Flexible Forwarding Engines for Multi Gigabit Switching," Project Report, SwitchCore Corp., 2000.

[21] E. Rosen *et al*, "MPLS Label Stack Encoding," IETF RFC 3032, January 2001.

[22] B. Davie *et al*, "MPLS using LDP and ATM VC Switching," IETF RFC 3035, January 2001.

[23] Hardhat, Montavista software, <http://www.mvista.com/>

[24] Zebra, IP Infusion Inc., <http://www.zebra.org>.

[25] R. Coltun, "The OSPF Opaque LSA Option," IETF RFC 2370, July 1998.

[26] R. Coltun and V. Fuller, "The OSPF NSSA Option," IETF RFC 1587, March 1994.

[27] G. Malkin, "RIP Version 2," IETF RFC 2453, November 1998.

[28] J. Moy, "OSPF Version 2," IETF RFC 2328, April 1998.

[29] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," IETF RFC 1771, March 1995.

[30] P. Srisuresh and P. Joseph, "OSPF-xTE: An experimental extension to OSPF for Traffic Engineering," IETF internet-draft, February 2003.

[31] P. Tomsu and G. Wieser, MPLS-Based VPNs, Prentice Hall PTR, 2002.

[32] D. Awduche, J. Malcolm, J. Agobua, M. O'Dell, J. McManus, "Requirements for Traffic Engineering Over MPLS," IETF RFC 2702, September 1999.

[33] K. Kompella and Y. Rekhter, "IS-IS Extensions in Support of Generalized MPLS," IETF Internet-draft, December 2002.

[34] D. Minoli, Voice over MPLS: Planning and Designing Networks, McGraw-Hill, 2002.

김 은 아(Eun-Ah Kim)

준회원



1993년 2월 : 성신여자대학교
전산학과 졸업

1995년 2월 : 성신여자대학교
전산학과 석사

1997년 3월~현재 : 충남대학교
컴퓨터공학과 박사과정

1995년 3월~1997년 3월:
성신여자대학교 전산원 직원

2000년 6월~현재: (주)라오넷 선임연구원

<주관심분야> 초고속정보통신, 고품질 네트워크, MPLS 망

전 우 직(Woo-Jik Chun)

정회원



1982년 2월 : 서울대학교
컴퓨터공학과 졸업

1984년 2월 : 서울대학교
컴퓨터공학과 석사

1989년: University of
Delaware 전산학과 석사

1992년: University of
Delaware 전산학과 박사

1984년~1987년 : 한국전자통신연구소 연구원

1992년~1993년: 한국전자통신연구소 연구원

1993년~2002년: 충남대학교 컴퓨터공학과 교수

1997년~1998년: 미국 National Institute of
Standards and Technology 객원연구원

2000년 4월 ~ 현재: (주)라오넷 대표이사

<주관심분야> 초고속정보통신, 프로토콜공학