

# 차별화 된 서비스를 제공하는 MPLS 기반의 VPN 구조 설계 및 구현

Design and Implementation of MPLS based VPN with DiffServ Capabilities

오홍석, 오영선, 이현태  
목원대학교

OH Hong-Seok, OH Young-Seon, Lee Hyeun-Tae  
Mokwon Univ.

## 요약

본 논문은 MPLS 기반의 VPN 서비스를 제공하고 각 VPN 사이트가 요구하는 QoS를 제공할 수 있는 에지 라우터를 설계하고 구현한다. 에지 라우터 설계를 위하여 라우팅과 시그널링 프로토콜을 기반으로 하는 제어 구조와 패킷 포워딩을 위한 데이터 구조를 설계하고 이를 이용하여 각각의 기능적 블록간의 인터페이스를 설계한다. 또한 에지 라우터의 설계를 기반으로 차별화 된 VPN 서비스를 제공하는 MPLS 기반의 VPN을 구현하고, 시뮬레이션을 통하여 설계된 에지 라우터 간 차별화 된 경로를 설정해 줌으로써 VPN 사이트 간에 요구되는 대역폭과 지연 등 트래픽 특성을 보장해 줄 수 있음을 확인한다.

## Abstract

In paper, we present design and implementation of MPLS based VPN with QoS capabilities. We propose a design architecture of an edge router for MPLS based VPN with QoS capabilities in backbone network. We design functional blocks, interface specifications, detail routing and forwarding information based structure, and packet forwarding engine for the purpose of implementation of an edge router. We extend bgp-4 software for implementation differentiated service packet forwarding schemes in MPLS based VPN.

## I. 서론

최근 기업간 인프라 구축이 빠르게 확산됨에 따라 공중망을 이용하는 가상 사설망(VPN) 서비스의 수요가 크게 증가하고 있다. 이를 기반으로 많은 요구 사항을 만족시키기 위한 다양한 VPN(Virtual Private Network) 서비스 기술에 관한 연구가 활발히 진행되고 있다. 특히 IPSec 프로토콜을 이용한 IP 터널링 기술은 인터넷 망 내에서 정보보안이 가능한 터널을 제공함으로써 VPN 서비스를 제공해왔다. 그러나 IP 터널링 VPN 서비스는 품질 보장이 제한적이며, 가입자 사이트 간 풀 메쉬 IP 터널을 구성해 주어야 하는 구조적 단점을 갖고 있다. 통신 사업자들은 공중망 서비스 수용을 위해 ATM(Asynchronous Transfer Mode) 교환 망을 이용한 Overlay 구조로 망을 개선하거나 네트워크 용량을 필요 이상으로 확장함으로써 인터넷 트래픽 증

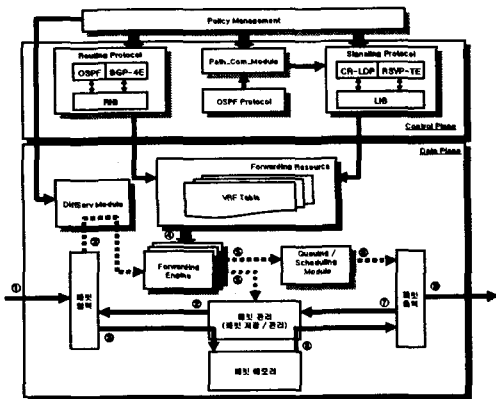
가에 대응하고 있다[1]. 그러나 이러한 대응에도 불구하고 인터넷 트래픽을 수용하기에는 경제적/기술적 한계에 직면하고 있다. 또한 가입자 측면에서는 보다 좋은 품질의 서비스를 안정적으로 제공받기를 희망하고 있다. 이에 따라 통신 사업자들은 한정된 망 자원을 이용하여 동적으로 바뀌는 트래픽을 효과적으로 처리할 수 있고 QoS(Quality of Service)를 차별적으로 지원함으로써 인터넷 망을 효과적으로 운용/관리하고, 네트워크 장애 발생시 신속한 복구가 가능한 네트워크 구조로의 진행을 추진하고 있다[2]. 이에 대한 기술적 대안으로 최근 관심을 끌고 있는 기술이 MPLS(Multi-Protocol Label Switching) 기반의 VPN 기술이다. MPLS 기반의 VPN 기술은 MPLS LSP(Label Switched Path)를 이용하여 VPN 사이트 간 경로를 설정해 줌으로써 VPN 서비스를 제공한다. 또한 가입자 특성에 따라 서

비스 품질의 차별적 지원이 가능한 Traffic Engineered LSP를 통해 VPN 트래픽을 전달한다[3][4].

본 논문은 이러한 기술을 바탕으로 QoS 특성을 고려한 MPLS 백본 망에서 VPN 패킷의 트래픽 처리에 대한 제어와 이를 통해 QoS를 제공하는 MPLS 기반의 VPN의 에지 라우터를 설계하고 이를 기반으로 차별화된 VPN 패킷 포워딩 절차를 서술한다. 이렇게 설계한 제어 구조를 바탕으로 리눅스 환경에서 각각의 데몬을 구성하고 실행시켜 QoS를 제공하는 MPLS 기반의 VPN을 구현하였다.

## II. 차별화된 서비스를 제공하는 에지 라우터 설계

차별화된 서비스를 제공하기 위한 에지 라우터는 그림 1과 같이 제어 구조(Control Plane)와 데이터 구조(Data Plane)로 이루어진다.



▶▶ 그림 1. 에지 라우터 설계

제어 구조(Control Plane)에서는 라우팅 프로토콜(Routing Protocol)인 확장된 BGP-4를 통해 VPN 패킷에 대한 도달성 정보와 멤버십 정보가 교환되고 가입자는 망 사업자와 서비스 사용에 대한 SLA 계약을 맺는다. 계약 내용에는 서비스의 종류, 서비스별 요구 품질을 포함한다. 서비스 계약이 이루어지면 SLA와 망사업자의 Path\_Comp\_Module에 의해 자원 사용 정책을 고려하여, 계약 내용을 만족시킬 수 있는 경로(Constraint Routed Path)를 찾는다. 이 경로를 기반으

로 CR-LDP나 RSVP-TE와 같은 시그널링 프로토콜(Signaling Protocol)을 이용하여 ER-LSP를 설정한다. MPLS 망으로 입력되는 트래픽에 대해 트래픽 특성과 SLA를 고려하여 VPN 패킷을 분류한 다음 적합한 LSP 혹은 ER-LSP로 분배한다. QoS를 제공하는 TE 경로에 대해서는 지속적으로 계약된 성능을 만족시키는지를 감시하고, 필요시에는 Re-routing 대역폭 증감과 같은 Re-optimization을 지속적으로 수행한다. 정책관리자(Policy Management)는 SLA 프로파일, QoS 정책, LSP별 품질 정보, QoS 메트릭과 같은 TE DB를 관리하고 이를 이용하여 경로 계산, 부하 분산, Re-routing등의 TE 제어 기능을 네트워크 전반에 대해 수행한다.

데이터 구조(Data Plane)는 패킷 포워딩을 처리하기 위한 모듈로 이루어져 있다.

DiffServ 모듈(DiffServ Module)은 MPLS Shim 헤더에 차별화된 서비스를 위해 EXP에 대한 DSCP 코드를 설정해 줌으로써 가입자의 계약 범위 안에서 트래픽 특성에 따라 PHB를 결정한다[5].

포워딩 리소스(Forwarding Resource)는 Routing Protocol에 의해 생성된 RIB 및 FIB와 시그널링 프로토콜에 의해 생성된 LIB의 정보를 구성하고 있다. 이를 VRF 테이블이라 하는데 VRF 테이블은 에지 라우터에 연결된 각각의 VPN 사이트마다 존재하게 된다.

포워딩 엔진(Forwarding Engine)은 정책 관리자에 의해 차별화된 VPN 패킷을 포워딩하기 위하여 포워딩 리소스를 룩업하여 큐잉/스케줄링 모듈로 전달한다.

큐잉/스케줄링 모듈(Queueing/Scheduling Module)은 QoS 관리자는 MPLS Shim 헤더의 EXP의 DSCP 코드를 룩업하고 정책 관리자의 정책에 따라 패킷을 구분하는 차별화된 메모리를 설정한다. 차별화된 패킷은 스케줄러에 의해 스케줄링되어 다음 홉으로 포워딩하게 된다.

에지 라우터 제어 구조를 기반으로 에지 라우터의 VRF 테이블이 생성되면 입력된 VPN 패킷은 VRF 테이블을 룩업하여 패킷을 포워딩한다. 그림 1과 같이 VPN 패킷에 대해 차별화된 서비스를 지원하는 인터넷에서 입력되는 VPN 패킷이 에지 라우터에서 포워딩

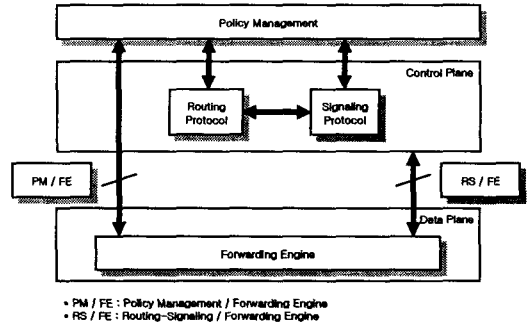
되는 절차는 다음과 같다.

- ① 패킷이 입력되면
- ② 저장할 패킷의 포인터를 패킷 관리기능으로부터 할당 받는다.
- ③
  - 할당 받은 패킷 포인터에 따라 입력 패킷을 패킷 메모리에 저장한다.
  - 입력된 패킷이 차별화된 패킷 포워딩을 위해 정책관리자의 제어를 받아 DiffServ Module에서 처리된 패킷의 헤더 정보와 I/F를 포함하는 패킷정보를 포워딩 엔진으로 전달한다.
- ④
  - VPN을 구분하는 정보(I/F 혹은 VPN ID)에 따라 해당 VRF 테이블을 룩업한다.
  - 룩업 결과에 따라 출력할 I/F와 레이블 정보를 할당 받는다.
- ⑤
  - 룩업한 결과에 따라 해당 출력 스케줄러로 패킷을 출력하기 위한 정보를 전달한다.
  - 출력할 패킷에 부착할 레이블 작업을 수행한다.
- ⑥ 스케줄러는 트래픽에 따른 출력 버퍼링 메커니즘을 적용하여 패킷 출력 기능을 동작 시킨다.
- ⑦ 패킷 출력 기능은 스케줄러의 명령에 따라 해당 패킷을 읽어 출력한다.

에지 라우터의 구현에서 기존에 있는 소프트웨어 모듈이나 하드웨어 모듈을 사용하는 것이 일반적이다. 특히 포워딩 기능의 설계는 완전히 소프트웨어로 수행하기에는 성능을 만족할 수 없고, 특별히 설계한 ASCI(Application Specific Integrated Circuit)으로 설계하는 것도 기능의 유연성이나 확장성 측면에서 문제가 있기 때문에 최근에는 고 성능의 NP(Network Processor)를 이용하여 설계하는 추세이다. 반면 Control Plane은 일반적으로 하드웨어가 아닌 소프트웨어로 구현하는 것이 일반적이다.

결과적으로 그림 2와 같이 라우팅 프로토콜, 시그널링 프로토콜 기능으로 구성된 Control Plane과 포워딩 기능을 수행하는 Data Plane 기능간의 인터페이스 기능이 필요하다. 따라서 본 절에서는 Data Plane과 Control Plane간의 인터페이스를 설계하고 VPN

Management와 Forwarding Engine간의 인터페이스(VM/FR I/F)과 Routing & Signaling 기능과 Forwarding Engine간의 인터페이스(RS/FE I/F)를 설계하였다.



▶▶ 그림 2. Control Plane과 Data Plane간 인터페이스 설계

RS/FE 인터페이스는 라우팅 프로토콜에 의해 생성된 포워딩 정보와 시그널링 프로토콜에 의해 설정된 레이블 정보를 포워딩 기능의 LFIB에 매핑하기 위한 인터페이스이다.

PM/FE 인터페이스는 VPN을 위해 확장된 BGP-4와 CR-LDP, RSVP-TE와 같은 시그널링 프로토콜에 의해 생성된 정보와 VPN의 추가 및 제거 설정을 포워딩 기능의 VRF에 매핑하기 위한 인터페이스이다[6][7].

### III. 구현

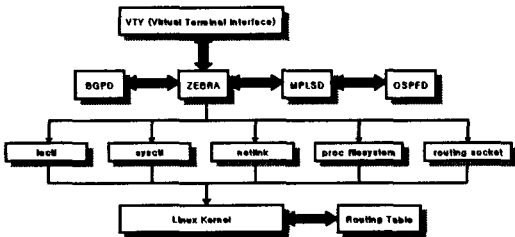
QoS를 제공하는 MPLS 기반의 VPN 구현은 IETF에서 표준화된 문서를 기반으로 GNU Zebra Software[8]를 이용하여 구현하였다. 또한 차별화된 VPN 패킷 포워딩을 위한 제약 기반 라우팅 설정은 요구된 대역폭을 만족하는 링크에 대해서 최단 경로를 구성하도록 하였고 패킷 입력 요청을 받았을 때 경로를 계산하여 비교적 최근의 링크 메트릭을 기준으로 더 좋은 경로를 계산하도록 하였다.

GNU Zebra Software는 TCP/IP 기반의 라우팅 프로토콜을 지원하는 소프트웨어로써 표 1은 Zebra에서 제공하는 라우팅 데몬과 각 데몬이 운영되는 라우팅 프로토콜을 나타내고 있다.

[표 1] Zebra에서 제공하는 라우팅 데몬과 프로토콜

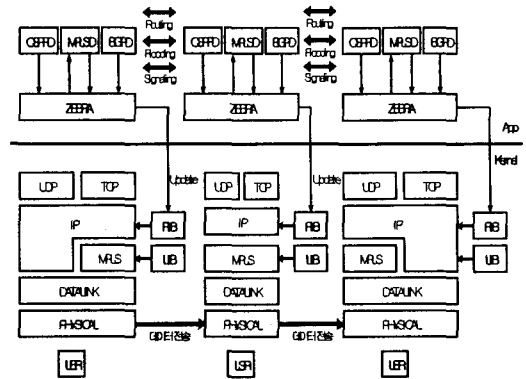
라우팅 데몬	라우팅 프로토콜
bgpd	BGP_4, BGP_4+
ripd	RIPv1, RIPv2
ripngd	RIPng
ospfd	OSPFv2
ospf6d	OSPFv3
zebra	커널 라우팅 테이블의 갱신과 위의 라우팅 프로세스와의 라우팅 정보 교환

Zebra에서 제공하는 라우팅 데몬을 기반으로 본 논문에서는 MPLS 데몬을 추가하여 QoS를 제공하는 MPLS 기반의 VPN 서비스를 제공하도록 구현하였다. 그림 3은 본 논문에서 구현된 ZEBRA 데몬의 구성도이다. ZEBRA 데몬은 다른 프로토콜 데몬(BGP 데몬, MPLS 데몬, OSPF 데몬 등)을 포함하고 있다. 각각의 프로토콜 데몬은 직접적으로 리눅스 커널과는 통신을 하지 않는다. 대신에 VTY(Virtual Terminal Interface)에서 ZEBRA 데몬을 핸들링 함으로써 다른 프로토콜 데몬과 통신을 하고 ZEBRA 데몬은 리눅스 커널과 통신을 한다. 또한 각각의 프로토콜 데몬은 라우팅 테이블을 가지고 있어 ZEBRA 데몬에 의해 라우팅 테이블이 갱신되고 커널의 라우팅 테이블을 갱신하게 된다.



▶▶ 그림 3. ZEBRA 데몬 구성도

Zebra Software를 이용하여 ZEBRA 데몬은 라우팅 프로토콜과 시그널링 프로토콜을 연동하기 위한 각각의 데몬을 제어하고 결과적으로 VRF 테이블이 구성됨으로써 QoS를 제공하는 MPLS 기반의 VPN을 구현하였다. 그림 4는 구현에 관한 전체적인 프로토콜 구조이다.

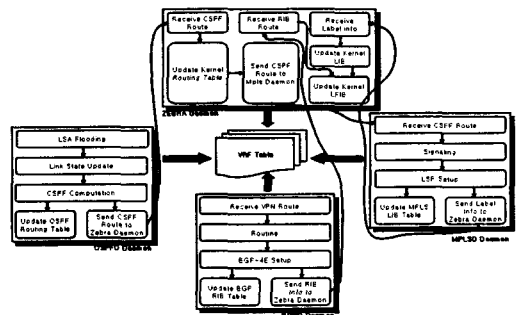


▶▶ 그림 4. 프로토콜 구조

에지 라우터로 VPN 패키지가 입력되면 OSPF 데몬은 헤더 정보를 록업하여 목적지 VPN 사이트로 VPN 패킷을 포워딩하기 위해 최대 가용대역폭을 만족하는 경로에 대해 최단 거리를 설정하기 위한 CSPF 알고리즘을 실행하게 된다. 생성된 라우팅 정보는 ZEBRA 데몬을 통해 MPLS 데몬을 호출하게 된다. MPLS 데몬은 OSPF 데몬에서 설정된 라우팅 정보를 기반으로 LDP를 이용하여 시그널링 프로토콜을 동작하고 레이블 정보를 ZEBRA 데몬을 통해 갱신한다. 또한 BGP 데몬에 의해 VPN-IPv4 형태의 패킷을 포워딩하기 위해 라우팅 프로토콜을 동작하여 VPN-IPv4 패킷의 정보를 교환하게 된다. 교환된 정보는 ZEBRA 데몬에 의해 갱신된다.

결과적으로 에지 라우터에 패킷이 입력되면 Zebra 환경에 의해 갱신된 VRF 테이블을 록업하고 목적지 VPN 사이트로 포워딩하게 된다.

각각의 데몬에 대한 상세 구조는 그림 5와 같다.



▶▶ 그림 5. 구현 상세 구조



비스를 위한 PE2 라우터의 ZEBRA 데몬과 BGP 데몬, MPLS 데몬의 구성결과를 나타내고 있다.

```

pe_router# sh run
Current Configuration:
! --- zebra 설정결과 시작 ---
!
! Host name
hostname pe2_router
!
! 목적지 VPN 라우터에 접근
! 목적지 VPN ID 할당 설정
ip vrf vrf1 rd 100 101
ip vrf vrf2 rd 100 102
!
! PE1으로부터 PE2으로 통행되는
! 인터넷 액세스 및 라우트
! 인터페이스 eth0
ip address 203.230.150.205/24
!
! PE2에서 VPN A 사이트로 통행되는
! 인터넷 액세스, 라우트 및 목적지 VPN ID
interface eth1
ip address 192.168.21.1/24
ip vrf forwarding vrf1
!
! 인터페이스 eth2
ip address 192.168.22.1/24
ip vrf forwarding vrf2
! --- zebra 설정결과 끝 ---
!

bgpd# sh run
Current Configuration:
! --- bgp 설정결과 시작 ---
!
! PE 라우터간 확장된 BGP-4 설정 및 이웃 포워딩
router bgp 100
 redistribute connected
 af ipv4 neighbor 203.230.150.214 remote-as 210 activate
 af ipv4 vrf neighbor 192.168.21.2 remote-as 210 activate
 af ipv4 vrf neighbor 192.168.22.2 remote-as 210 activate
! --- bgp 설정결과 끝 ---
!

mpls# sh run
Current Configuration:
! --- mpls 설정결과 시작 ---
!
! mpls 라우터 설정
mpls ldp router-id eth0
!
! 인터페이스 eth0
interface eth0
 ip address 203.230.150.205/24
 mpls ldp forwarding ldp_label
!
! 인터페이스 eth1
interface eth1
 ip address 192.168.21.1/24
 mpls ldp forwarding ldp_label
!
! 인터페이스 eth2
interface eth2
 ip address 192.168.22.1/24
 mpls ldp forwarding ldp_label
! --- mpls 설정결과 끝 ---
!
    
```

▶▶ 그림 8. 각 데몬의 구성 결과

또한 Zebra Software를 이용하여 VPN을 구현함에 있어 표 2와 같은 구성을 요하게 된다.

[표 2] VPN 구성 요소

속 성	설 명
ip vrf VRF_NAME rd ASN VPN_ID	vrf 구성을 위해 사용되며 ZEBRA 데몬에서 구성한다.
ip vrf forwarding VRF_NAME	VPN 사이트에 접근하기 위해 인터페이스를 구성하는데 사용된다. 이는 VPN-IPv4를 전송하기 위한 모든 인터페이스 정보를 구성한다.
af vrfv4 neighbor A.B.C.D remote-as ASN activate	VPN 사이트에 연결된 PE 라우터 간 정보 교환 목적으로 구성된다.

그림 9는 PE2 라우터에서 VRF 테이블 결과를 나타내고 있다. 결과 값을 살펴보면 PE 라우터 간 확장된 BGP-4를 이용하여 VPN ID 정보 및 VPN-IPv4 정보를 교환하게 된다. 즉, 해당 목적지 VPN 사이트로 패킷을 전송하기 위해서는 입력 인터페이스에 VPN ID 및 RD값을 설정하여 자신의 VPN 사이트가 어느 위치에 존재하는지 그리고 어떠한 경로로 전송될 것인지에 대해 경로 루틴을 VRF 테이블 삽입시킨다. 또한 PE 라우터 간 시그널링 정보를 교환하기 위하여 LDP를 이용하여 레이블 값을 교환한다. 레이블 값을 교환함으로써 PE 라우터까지 패킷을 포워딩하기 위한 Outer 레이블

과 PE 라우터에서 VPN 사이트까지 포워딩하기 위한 Inner 레이블 값을 확장된 BGP-4를 이용하여 교환하게 된다.

```

pe2router# sh ip router
Codes : K - Kernel route, C - connected, S - static, R - RIP,
        O - OSPF, B - BGP, s - FIB route.

K* 0.0.0.0/0 0 0 eth0 (2) 203.230.150.255
C* 127.0.0.0/8 0 0 lo (1) direct
C* 203.230.150.0/24 eth0 (2) direct
C* 192.168.21.0/24 100 101 eth1 (6) direct
C* 192.168.22.0/24 100 102 eth2 (4) direct

bgpd# sh bgp interface

eth0 0 0 0
eth1 0 100 101 vrf1
lo 0 0 0
eth2 0 100 102 vrf2

bgpd# sh bgp routes

prefix/prefixlen RD
192.168.21.0/24 100 : 101
192.168.22.0/24 100 : 102
inet routes
203.230.150.0/24 0 : 0
    
```

▶▶ 그림 9. PE2 라우터의 VRF 테이블 결과 값

그림 10은 구현 환경에서 IP-VPN과 MPLS 기반의 VPN에 대해 PE 라우터 간 경로에 따른 패킷 포워딩의 소요시간을 비교한 것이다.

결과를 살펴보면 기존의 IP-VPN인 경우 목적지 노드까지 포워딩을 위한 최대 소요시간은 1.636ms이고 최소 소요시간은 0.226, 평균 소요시간은 0.275ms가 나왔다. 반면 LDP를 이용하여 LSP를 설정해 줌으로써 목적지 노드까지 포워딩을 위한 최대 소요시간은 1.956ms이고 최소 소요시간은 0.211ms, 평균 소요시간은 0.247ms의 시간이 소요되는 것을 확인하였다. 결과 값을 비교해보면 최대 소요시간인 경우 MPLS 기반의 VPN은 라우팅 정보를 기반으로 레이블 정보를 교환하기 때문에 기존의 IP-VPN보다 시간이 오래 걸리지만 패킷 포워딩을 위한 LSP가 설정되면 최소 소요시간 및 평균 소요시간에서는 좀더 빠른 패킷 포워딩이 이루어지는 것을 확인하였다.

구현에 있어 환경적인 요소의 부족으로 IP-VPN보다 MPLS 기반의 VPN 서비스에 대한 고속의 패킷 포워딩의 차이가 많지는 않지만 이에 대한 검증은 시뮬레이션을 통하여 확인하였다[9][10].

```
// LDP 동작 전 패킷 포워딩 소요시간
[root@pe2 /root]# traceroute 203.230.150.214
traceroute to 203.230.150.214(203.230.150.214), 30 hops max, 40 byte packets
 1 203.230.150.214 (203.230.150.214) 1.636ms 0.275ms 0.226ms

// LDP 동작 후 패킷 포워딩 소요시간
[root@pe2 /root]# traceroute 203.230.150.214
traceroute to 203.230.150.214(203.230.150.214), 30 hops max, 40 byte packets
 1 203.230.150.214 (203.230.150.214) 1.956ms 0.247ms 0.211ms
```

▶▶ 그림 10. 경로에 따른 패킷 포워딩 소요시간 비교

#### IV. 결론

MPLS(Multi-Protocol Label Switching) 기반의 VPN(Virtual Private Network)은 다양한 액세스 환경의 가입자에게 품질 보장형의 서비스를 경제적인 비용으로 지원하고 망 사업자가 서비스 운용관리를 포함하여 다양한 부가 서비스를 효과적으로 제공할 수 있는 차세대 인터넷의 핵심 기술이다.

본 논문은 QoS를 지원하는 MPLS 기반의 VPN을 설계하고 구현하였다. 본 논문에서는 RFC 2547-bis로 표준화된 BGP/MPLS VPN을 기반으로 QoS를 제공하기 위한 에지 라우터를 설계하고 구현하였다.

에지 라우터 설계를 위해 라우팅과 시그널링 프로토콜을 기반으로 한 제어 구조와 패킷 포워딩을 위한 데이터 구조를 설계하였고 각각의 기능적 블록간의 인터페이스를 설계하였다.

또한, 설계한 에지 라우터를 기반으로 QoS를 제공하는 MPLS 기반의 VPN을 구현하기 위하여 GNU Zebra Software를 기반으로 차별화 서비스를 제공하는 MPLS 기반의 VPN을 구현하였다. 구현 결과, 에지 라우터간 LSP가 설정되면 패킷 포워딩을 위해 최소 소요시간 및 평균 소요시간이 기존의 IP-VPN보다 고속의 패킷 포워딩이 이루어지는 것을 확인하였다.

향후 연구 과제로는 VPN 패킷의 CoS(Class of Service) 등급을 구분하는 패킷 구별 기술과 MPLS VPN 망 관리 기술에 대한 모니터링, 최적화 기술 등에 대한 많은 연구가 이루어져야 할 것이다.

#### ■ 참고문헌 ■

- [1] Tony Li, "MPLS and the Evolving Internet Architecture," IEEE Communications Magazine, Vol. 37, No. 12 Dec. 1999.
- [2] Daniel O. Awduche, "MPLS and Traffic Engineering in IP Networks," IEEE Communications Magazine, Vol. 37, No. 12, Dec. 1999.
- [3] "Multi Protocol Label Switching," White paper, Future Software Private Limited, India,
- [4] Layer3 Switching Using MPLS," White paper, Harrison & Jeffries.
- [5] L. Wu and B. Davie, S. Davari, P. Cheval et al., "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," *RFC3270*, May. 2002.
- [6] J. Ash, Y. Lee, and P. Ashwood-smith, B. Jamoussi, "LSP Modification Using CR-LDP," *RFC3214*, Jan. 2002.
- [7] D. Awduche and L. Berger, D. Gan et al., "RSVP-TE : Extensions to RSVP for LSP Tunnels," *RFC3209*, Dec. 2001.
- [8] URL : <http://www.zebra.org>
- [9] URL : <http://www.raonet.com>
- [10] URL : <http://isi.edu/nsnam>