

ATM 환경에서 MPLS 라우터를 위한 포워딩 엔진의 설계와 구현

최승환*, 김은아, 손명희, 전우직
충남대학교 컴퓨터공학과

The Design and Implementation of Forwarding Engine for MPLS router on the ATM environment

Seunghan Choi*, Eunah Kim, Myunghee Son, Woojik Chun
Dept. of Computer Engineering, Chungnam National University

요 약

인터넷 사용자의 증가와 이에 따른 인터넷 트래픽 증가는 현재의 인터넷 서비스의 질을 현저히 떨어뜨리고 있다. 이러한 배경에서 인터넷 망 사업자는 사용자의 증가에 따라 망을 확장해야 하는 과제가 시급히 요구되었는데 이런 요구 사항에 의해 제시된 것이 MPLS 라는 L3 스위칭 기법이다. 본 논문에서는 MPLS 기술을 기반으로 하여 ATM 환경에서의 개발중인 MPLS 라우터의 일부가 되는 포워딩 엔진의 설계와 구현에 대해 기술하고자 한다.

1. 서론

인터넷 사용자의 증가와 이에 따른 인터넷 트래픽 증가는 현재의 인터넷 서비스의 질을 현저히 떨어뜨리고 있으며 이러한 배경에서 인터넷 망 사업자는 사용자의 증가에 따라 망을 확장해야 하는 과제가 시급히 요구되고 있다. 하지만 단순히 대역폭을 확장하는 것 뿐 아니라 라우터 수의 증가와 라우터가 관리하는 라우터 테이블 수의 증가를 포함하는 망의 확장성을 해결할 수 있는 방안을 요구하고 있다. 이런 요구 사항에 의해 제시된 것이 MPLS(Multiprotocol Label Switching)라는 L3 스위칭 기법이다.

본 논문에서는 현재 ATM 환경에서 개발중인 MPLS 라우터의 일부가 되는 포워딩 엔진에 대해 다루고자 한다. 2 장에서는 MPLS 의 개념과 MPLS 라우터에서 포워딩 엔진의 역할을 설명하고, 3 장에서는 포워딩 엔진의 설계와 구현에 대해 설명하고자 한다. 마지막으로 4 장에서는 결론 및 향후과제를 기술한다.

2. 관련 연구

2.1 MPLS 개념

MPLS[1,2]는 레이블(label)이라는 짧고 고정된 길이의 식별자를 사용하여 패킷 포워딩을 하는 L3 스위칭 방법이다. 레이블 값은 해당 노드에서만 의미를 갖는 값으로, 레이블 스위칭에 사용되는 디바이스인 LSR(Label Switched Router)이 레이블에 대한 스와핑(Swapping)을 통해 패킷 포워딩을 진행한다.

MPLS 를 사용하게 되면 기존의 데이터그램 라우팅 네트워크에 비해 간단한 포워딩, 효율적인 Explicit 라우팅과 QoS 라우팅의 제공, Traffic Engineering 의 지원, IP 패킷의 FEC(Forwarding Equivalence Class) 대응의 단순화 등의 장점을 취할 수 있다. 또한 ATM 과 Frame Relay 에 대해서는 향상된 라우팅 확장성, 패킷과 셀 미디어에 대해 공통적인 연산 적용, 쉬운 관리, 규모가 큰 네트워크에서의 라우팅의 문제점 제거 등의 장점이 있다.

2.2 MPLS 라우터에서 LDP 와 포워딩 엔진

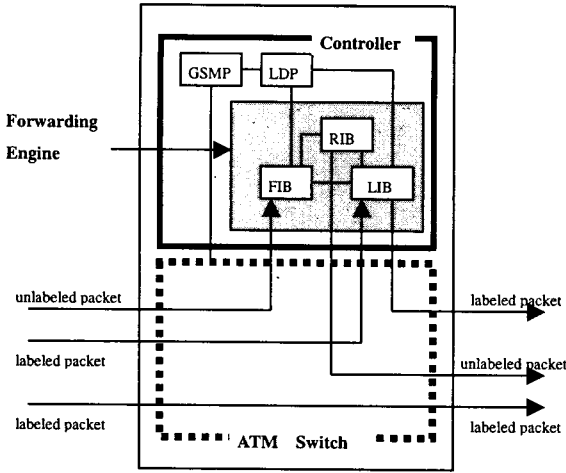
레이블을 사용하려면 레이블의 부여와 분배 과정을 거쳐야 하는데, 이 기능을 담당하는 프로토콜이 LDP(Label Distribution Protocol)[3]이다. LDP 는 3 계층 라우팅 정보를 2 계층 경로로 직접 매핑시킨 망에서 LSR 이 LSP(Label Switched Path)들을 설정하기 위해 필요한 메시지와 절차들로 구성된 프로토콜이다. 위의 과정과 절차에 의해 LDP 는 입력 레이블과 출력 레이블과의 매핑 정보를 LIB(Label Information Base) 테이블로 유지한다. 포워딩 엔진은 라우팅 테이블을 기반으로 FIB(Forwarding Information Base)를 생성하며, FIB 와 LDP 에 의해 생성된 LIB 테이블을 기반으로 패킷을 포워딩한다.

3. 포워딩 엔진의 설계와 구현

3.1 ATM 환경에서 MPLS 라우터 모델

ATM 환경에서 MPLS 도메인이 다른 MPLS 도메인과 계층적 구조를 가지지 않는다고 가정할 때, MPLS 도메인 내에 있는 LSR 들은 다음 세 가지 경우에 대해 패킷을 포워딩할 수 있는

능력을 가진다.



[그림 1] ATM 환경에서 MPLS 라우터 모델

- 1) 레이블이 붙은 패킷(Labeled packet)을 포워딩할 수 있다.
 - 2) MPLS 도메인으로 들어오는 트래픽(L3 패킷)의 FEC를 분류한 후, 레이블을 트래픽에 추가시켜 포워딩할 수 있다.
 - 3) MPLS 도메인을 떠나는 트래픽에 대해 포함된 레이블정보를 제거하여 L3 패킷을 만든 후 포워딩할 수 있다.
- 이런 능력을 가진 LSR 라우터의 모델은 [그림 1]과 같다.

LSR은 크게 컨트롤러와 ATM 스위치로 구분된다. 컨트롤러는 LDP와 포워딩엔진, GSMP로 구성되어 있으며 ATM 환경에서 레이블은 ATM 셀 헤더의 VPI/VCI 값으로 의미를 부여한다. 라우팅 프로토콜에 의해 주소 prefix, next hop, FEC (Forwarding Equivalence Class)가 결정되고, 레이블 스와핑에 필요한 입력/출력 레이블은 LDP의 기능 중 레이블과 FEC 바인딩 절차에 의해 결정된다.

3.2 GSMP를 이용한 ATM 스위치 제어

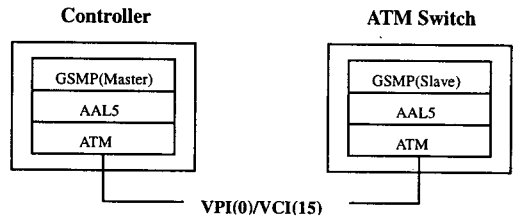
GSMP(General Switch Management Protocol)[7]는 스위치 외부의 컨트롤러가 ATM 스위치를 제어할 수 있도록 하기 위한 프로토콜이다. 기존의 ATM 스위치에서는 가상 채널의 설정과 해제를 위해 ATM 시그널링 프로토콜(Q.2931)을 이용하여 설정하였다. 하지만 MPLS에서는 LDP를 이용하여 ATM 환경에서의 레이블인 VPI/VCI를 부여하고 분배하게 되며 이렇게 할당된 VPI/VCI를 이용하여 ATM 스위치내의 GSMP를 이용해서 가상 채널을 설정하고 해제한다. 현재 GSMP 버전 1.1은 VP 스위칭을 지원하지 않기 때문에 VCI 값만을 레이블로 사용하였다. GSMP는 가상 채널의 설정과 해제 뿐만 아니라 스위치에 대한 여러 가지 관리 기능을 수행한다.

GSMP를 통해 제공받을 수 있는 기능은 다음과 같다.

- ◆ 스위치에 가상채널을 설정하고 해제할 수 있다.
- ◆ 일 대 다중 형태(Point to Multipoint)의 연결을 지원한다.
- ◆ 스위치에 대한 관리(루프백 테스트 등)를 할 수 있다.
- ◆ 스위치의 구성정보(VPI/VCI의 범위 등)를 알 수 있다.

- ◆ 가상채널을 지나는 트래픽에 대한 정보를 알 수 있다.

GSMP 프로토콜은 제어기가 마스터(Master)로, 스위치가 슬레이브(Slave)로 동작하는 비대칭형 프로토콜이다. 그래서, 제어기는 항상 스위치에 요구 메시지를 보내게 된다. 본 연구에서 제어기는 리눅스(커널버전 2.2.9)를 운영체제로 사용하는 PC에서 구현하였으며, PC에 장착되어 있는 ATM 카드는 Fore사의 PCA-200E 모델을 사용하였다(그림 2). ATM 스위치는 Virata사의 155Mbps 두 개 포트와 25Mbps 열 두개 포트를 가진 VM1000 스위치를 이용하였다.

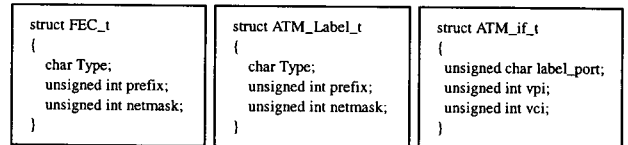


[그림 2] GSMP를 통한 스위치제어 구현 모델

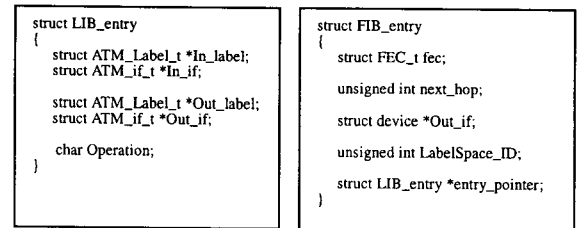
3.3 포워딩 엔진의 기능

포워딩 엔진 구현 환경으로 컨트롤러는 PC 상에 리눅스(커널 버전 2.2.9) 운영체제 사용하였으며 ATM 스위치는 Virata사의 VM1000, ATM 카드는 Fore사의 PCA-200E를 사용하였다.

포워딩 엔진에서 패킷을 처리하기 위해서는 FIB와 LIB 테이블이 필요하다. 현재 구현된 FIB와 LIB의 엔트리 내용과 필요한 정보는 [그림 3]과 같다.



(a) FEC 구조체 (b) ATM Label 구조체 (c) 인터페이스 구조체



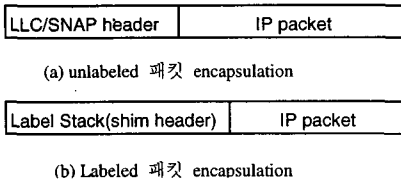
(d) FIB 엔트리 (e) LIB 엔트리

[그림 3] FIB와 LIB 엔트리의 내용

3.3.1 Labeled 패킷과 Unlabeled 패킷의 구별

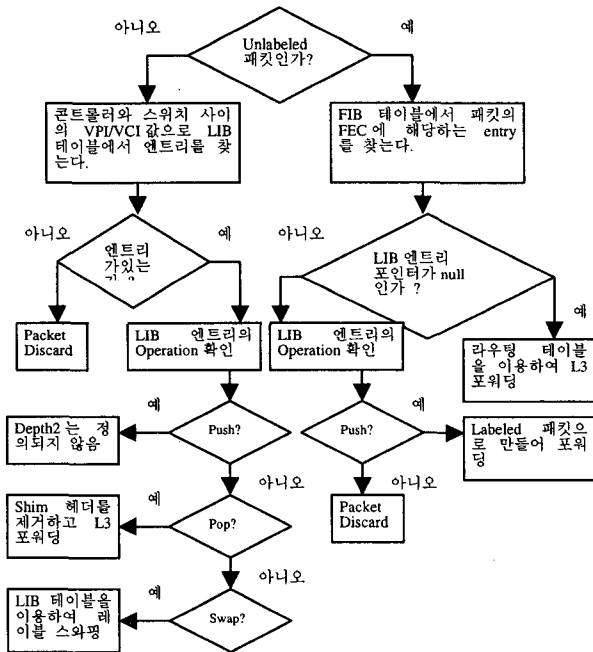
스위치로부터 컨트롤러로 패킷이 전달되면 labeled 패킷인지 unlabeled 패킷인지 구별해야 하며, 또한 labeled 패킷과 unlabeled 패킷에 맞는 encapsulation 방식을 이용하여 패킷을 처리해야 된다. Labeled 패킷에 대해서는 "Null encapsulation"의 AAL5 데이터 패킷으로 전송하고, OSPF(Open Shortest Path First) 패킷등의 unlabeled 패킷과 LDP 피어(Peer)사이의 LDP 제어 패킷에는 LLC(Logical Link Control) encapsulation[5]의 AAL5 데이터

패킷으로 전송한다(그림 4).



[그림 4] Unlabeled 패킷과 Labeled 패킷의 Encapsulation 방식

포워딩 엔진에서 패킷에 대한 동작은 레이블을 패킷에 붙이는 푸쉬(Push), 레이블을 패킷에서 제거하는 팝(Pop), 입력 레이블을 출력 레이블로 바꾸는 스왑(Swap) 등 세 가지가 있다.



[그림 5] 포워딩 엔진 플로우 차트

3.3.2 Unlabeled 패킷의 처리

Unlabeled 패킷일 경우는 IP 패킷에 대한 처리를 해야 한다. IP 패킷으로 FIB 테이블을 검색하여 FEC가 같은 FIB 엔트리를 찾아서 LIB 엔트리 포인터를 조사한다(그림 5). NULL 이면 FEC에 해당되는 LIB 엔트리가 존재하지 않는다는 의미이다. 즉, 출력 레이블(Outgoing Label)이 아직 할당되지 않은 상태이므로 LDP에 할당을 요청해야 한다. 만약 NULL이 아니라면 LIB 엔트리가 존재하는 것이므로 이 엔트리의 동작(Operation)을 확인한다. Unlabeled 패킷에 대한 동작은 출력 레이블을 붙이는 푸쉬(Push) 동작밖에 없으므로 그 외는 오류처리 한다. 푸쉬일 경우 shim 헤더[6]를 만들어 unlabeled 패킷에 추가하여 labeled 패킷을 만든 후 출력 인터페이스로 포워딩한다.

3.3.3 Labeled 패킷의 처리

Labeled 패킷일 경우 컨트롤러에 전달된 labeled 패킷의 가상 채널의 VPI/VCI를 가지고 LIB 엔트리의 입력 인터페이스를 검색하여 같은 엔트리가 있는지 확인한다(그림 5). ATM 환경에서는 컨트롤러와 스위치가 분리되어 있는 상태이므로 입력 레이블로 사용되는 VPI/VCI가 무엇인지 컨트롤러에서는 확인할 수 없다. 그러므로 해당 엔트리를 발견했을 경우 이에 대한 동작을 확인한다. Labeled 패킷에 대해 푸쉬 동작을 할 경우는 MPLS 도메인이 2 이상의 계층구조를 가지고 있는 경우에 해당되는데, 본 논문에서는 이에 대한 처리는 다루지 않고 있다. 팝(Pop)일 경우 도메인이 계층구조가 없는 상황에서 레이블 스택 엔트리가 한 개이고 이 엔트리를 제거하여 unlabeled 패킷(IP 패킷)으로 만든 후, 라우팅 테이블을 참조하여 L3 포워딩을 한다. 스왑(Swap)일 경우는 컨트롤러까지 labeled 패킷이 전달되지 않고 ATM 스위치에 의해서 레이블 스와핑이 이루어진다.

4. 결론 및 향후과제

지금까지 ATM 환경에서 MPLS 라우터의 포워딩 엔진의 기능으로 unlabeled 패킷과 labeled 패킷에 따라 처리하는 절차를 설명하였다.

ATM 기반의 LSR은 컨트롤러와 스위치가 분리되어 동작하기 때문에 이더넷 기반의 포워딩과는 차이가 있다. 즉, 이더넷 기반에서는 LSR의 LIB 테이블을 lookup하고 스와핑이 이루어지지만 ATM 기반에서는 주로 ATM 스위치에 의해 레이블 스와핑이 이루어진다. 또한, 컨트롤러와 스위치가 분리되어 있기 때문에 이에 대한 링크도 관리를 해주어야 한다.

네트워크 도메인들은 일반적으로 계층구조를 갖을 수 있다. 그러므로 본 연구의 포워딩 엔진은 계층구조를 지원할 수 있도록 확장하는 것도 고려해야 한다. 이 외에도 Shim 헤더의 TTL(Time To Live)에 대한 조정도 필요하다.

[참고문헌]

- [1] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow, A. Viswanathan, "A Framework for Multiprotocol Label Switching", Internet Draft <draft-ietf-mpls-framework-04.txt>, July, 1997.
- [2] Eric C. Rosen, Arun Viswanathan, Ross Callon, "Multiprotocol Label Switching Architecture", Internet Draft <draft-ietf-mpls-arch-05.txt>, April, 1999.
- [3] Loa Andersson, Paul Doolan, Nancy Feldman, Andre Fredette, Bob Thomas, "LDP Specification", Internet Draft <draft-ietf-mpls-ldp-05.txt>, June, 1999.
- [4] Bruce Davie, Jeremy Lawrence, Keith McCloghrie, Yakov Rekhter, Eric Rosen, George Swallow, Paul Doolan, "MPLS using LDP and ATM VC Switching", Internet Draft <draft-ietf-mpls-atm-02.txt>, April, 1999.
- [5] Heinanen, J., "Multiprotocol Encapsulation over ATM Adaptation Layer 5", RFC 1483, July 1993.
- [6] Eric C. Rosen, Yakov Rekhter, Daniel Tappan, Dino Farinacci, Guy Fedorkow, Tony Li, Alex Conta, "MPLS Label Switching Encoding", Internet Draft <draft-ietf-mpls-label-encaps-03.txt>, Sep., 1998.
- [7] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, G. Minshall, "Ipsilon's General Switch Management Protocol Specification version 1.1", RFC1987, Agu. 1996.