

플로우 레이블을 지원하는 IPv6라우팅 데몬의 구현

김형준^o 오승현 안종석
동국대학교 컴퓨터공학과
{stragon, shoh, jahn}@dongguk.edu

The Extension of IPv6 Routing Daemon For Using The Flow Label

Hyung-Jun Kim^o, Seung-Hyun Oh, Jong-Suk Ahn
Dept. of Computer Engineering, Dongguk University

요약

본 논문은 IPv6 헤더의 플로우 레이블 필드를 이용한 레이블 포워딩 방법을 제안하고 이 방법을 기반으로 리눅스 운영체제에서 멀티캐스팅 라우팅 데몬과 유니캐스팅 데몬을 구현하여 레이블 포워딩에 의한 유니캐스트 및 멀티캐스트 전송을 구현하였다. 레이블을 이용한 포워딩은 라우팅 엔트리를 결정하기 위해 IP주소 중 가장 많은 부분이 일치되는 엔트리를 찾는 방법(longest prefix match)을 기본으로 하는 IP주소검색 방법에 비해, 짧은 레이블 값 전체가 일치하는 엔트리를 찾는 방법(short label exact match)을 원칙으로 하고있어 상대적으로 빠른 속도로 라우팅 테이블을 검색할 수 있으며, 쉽게 QoS를 제공할 수 있는 구조를 제공한다. 실험을 통해 구현된 레이블 포워딩을 이용한 유니캐스트 및 멀티캐스트 전송이 잘 동작함을 확인하였고, 성능비교 실험을 통해 레이블 포워딩이 일반 IP 검색 포워딩 방법보다 더 좋은 성능을 발휘함을 확인할 수 있다.

1. 서론

새로운 Internet Protocol로서의 IPv6[1][9]의 제안은 먼저 IPv4가 32bits 기준으로 addressing을 구현하고 있으므로 네트워크의 이름을 효율적인 구조로 관리하기 어려우며, 나아가 가까운 장래에 주소공간이 포화 될 위험에 처하리라는 인터넷의 환경 변화에 기인하여 제안되어졌다. IPv4와 비교해 볼 때 IPv6는 몇 가지 특징을 갖는데, 그 중 하나로 플로우 레이블(flow label) 필드가 헤더에 포함된 점을 들 수 있다. 플로우 레이블은 원래 QoS를 지원하기 위해 제안되었는데 아직 구체적인 활용방안에 대한 표준안은 나오지 않고 있다. 다만 integrated service와 관련하여 flow별 QoS를 지원하기 위해 플로우 레이블이 사용될 수 있다고 생각된다.

본 논문에서는 플로우 레이블에 의한 라우팅(routing)이 가능하도록 레이블의 값에 따라 포워딩을 수행하는 방법을 제안하고, 이 포워딩 엔진을 기반으로 유니캐스트 라우팅 데몬을 구현하였으며, 나아가 멀티캐스트 라우팅 데몬까지 확장하여 구현하였다. 현재 리눅스커널(2.2.X)에서는 부분적으로 IPv6코드를 지원하고 있고 유니캐스트 라우팅도 지원하고 있는데 이는 일반 IP검색방법을 이용하고 있다.

본 논문에서는 일반 IP검색방법 대신 플로우 레이블에 의한 포워딩을 구현하여 QoS 제공이 손쉬운 네트워크 구조와 빠른 라우팅 테이블 검색을 할 수 있도록 하였다. 또한 플로우 레이블에 의한 포워딩 방법을 이용하여 멀티캐스트부분까지 확장하여 멀티캐스트 라우팅 데몬을 구현하였다. 현재의 리눅스 IPv6 멀티캐스트 환경을 살펴보면 로컬 네트워크 상에서의 멀티캐스

트 통신만을 지원하도록 되어있다. 즉, 서로 다른 서브넷 상에 존재하는 호스트들 사이에서는 멀티캐스트 라우팅 데몬의 부재로 멀티캐스트 통신을 할 수 없다. 따라서, 본 연구에서는 리눅스 IPv6환경에서 멀티캐스트 라우팅이 가능하도록 IPv6용 멀티캐스트 라우팅 데몬을 구현하였다. 멀티캐스트 라우팅이 가능하게 하기 위해서는 두 가지 문제가 해결이 되어야 하는데 첫째는 멀티캐스트 라우팅 정보를 주고받는 데몬 프로그램이고, 둘째는 데몬이 주고받는 라우팅 정보를 이용하여 멀티캐스트 패킷을 포워딩 하는 커널 포워딩 루틴이다. 이에 IPv6환경에서 멀티캐스트 라우팅정보를 주고받는 데몬 프로그램을 구현하였고 패킷을 포워딩 하는 커널 포워딩 루틴을 플로우 레이블에 의해 포워딩 하도록 하였다. 유니캐스트 라우팅 데몬은 현재 zebra[7], gated등의 라우팅 데몬이 지원된 상태이다. 본 연구에서는 zebra에 플로우 레이블 포워딩 엔진을 추가하여 레이블에 의해 포워딩을 하도록 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 레이블에 의한 포워딩 동작과정을, 4장에서는 실험결과를 설명한다. 그리고 마지막으로 5장에서 결론을 기술한다.

2. 관련 연구

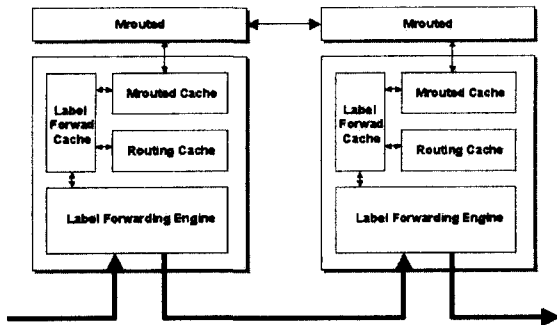
2.1. mrouted

mrouted[2][4]는 유닉스 환경에서 데몬(daemon)형태로 구동되는 프로그램으로 포워딩 기능을 가진 호스트 상에서 구동되어 호스트가 멀티캐스트 라우터 역할을 하도록 해준다. 현재 mrouted는 IPv4만을 지원하고 있는데 본 연구에서는 이를 확장하여 IPv6 주소를 지원하도록 설계하였다. mrouted가 취급

© 이 연구는 '2000 대학기초연구사업의 지원을 받아 이루어졌습니다.

하는 정보는 크게 멀티캐스트 라우팅 도메인(domain)에서 일반 호스트들과 직접 연결되어있는 가장자리 라우터(edge router)가 우선적으로 접하게 되는 그룹멤버의 가입과 탈퇴에 대한 정보와, 도메인 내의 라우터들끼리 서로 주고받는 라우팅 테이블에 대한 정보, 두 가지가 있다. 이 두 가지 정보들은 대부분 mrouter내에서 유지·관리가 되지만, 실제 포워딩과 관련된 정보들은 커널내부의 자료구조와 동기화가 되어야 한다. 이 자료구조를 멀티캐스트 포워딩 캐쉬(MFC :Multicast Forwarding Cache)라고 한다. 이 자료구조는 그룹이 생성·삭제되고, 멀티캐스트 트래픽을 발생시키는 송신자가 생겨나고 없어짐에 따라 지속적으로 수정되어진다. mrouter가 가지고 있는 라우팅 정보를 이용해 포워딩을 하려면 커널 내에 포워딩 시 참조할 캐쉬(MFC)와 이 캐쉬를 이용해 실제 포워딩을 해주는 포워딩 엔진이 필요하다.

기존 IPv4용 mrouter는 패킷을 포워딩을 할 때 라우팅정보와 MFC를 이용해 포워딩을 하는데 비해 본 연구에서의 플로우 레이블에 의한 포워딩은 레이블 포워드캐쉬를 이용하는데 이는 mrouter가 처음 시작되면서 멀티캐스트 트리를 만들기 위해서 모든 인터페이스로 패킷을 포워딩 한다. 이 패킷을 수신한 중간 라우터들도 절단 메시지를 받기 전까지는 RPM (Reverse Path Multicasting)[5]에 의하여 패킷을 포워딩 한다. 이러한 동작을 하면서 레이블 포워드캐쉬가 만들어지고, 이 캐쉬를 만드는 과정에서 레이블이 할당된다. 본 논문에서의 레이블 포워딩 엔진에서는 이 레이블 포워드캐쉬를 이용하여 패킷을 포워딩을 한다. <그림1>은 본 논문에서 구현한 IPv6상에서의 레이블 포워딩을 지원하는 멀티캐스트 라우팅 데몬의 구성도로 레이블 포워딩 엔진에 의해 패킷을 포워딩하는 구조를 보여준다.

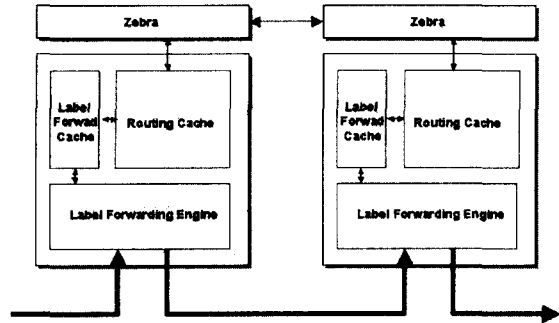


<그림 1> 멀티캐스트 라우팅 데몬

2.2. zebra

zebra는 유닉스 환경에서 데몬(daemon)형태로 구성되는 프로그램으로 동적 라우팅 프로토콜을 이용하여 호스트가 유니캐스트 라우터 역할을 하도록 해준다. zebra는 다수의 라우팅 프로토콜을 지원하는데 특히 IPv6를 지원하는 RIPng[8] 프로토콜을 포함하고 있다. 라우팅 도메인 내부의 라우터간에 라우팅 정보를 교환하기 위해 RIPng packet을 이용하여 라우팅 정보를 교환하고, 이 정보를 이용하여 zebra는 커널의 라우팅 테이블을 갱신한다. 기존에는 이 라우팅테이블정보를 이용해 패킷을 포워딩을 하는데 비해 본 연구에서는 플로우 레이블을 이용하여 포워딩을 하므로 레이블 포워드캐쉬가 필요한데 라우터에 레이블 패킷이 도착하면서 동적으로 레이블 포워드캐쉬가 만들어지는데 이 과정에서 커널내부의 레이블 마스크를 검사하여 레이블을 할당한다. 이 레이블 포워드캐쉬를 이용해 레이블 포

워딩 엔진에서 패킷을 포워딩 한다. <그림 2>는 본 논문에서 구현한 플로우 레이블 포워딩을 지원하는 유니캐스트 라우팅 데몬의 구성도로 레이블 포워딩 엔진에 의해 패킷을 포워딩하는 구조를 보여준다.



<그림 2> 유니캐스트 라우팅 데몬

3. IPv6 플로우 레이블 스위칭(Switching)

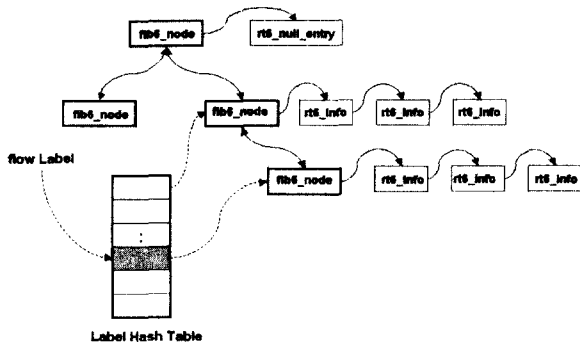
이 장에서는 라우팅 데몬이 유지 관리하는 라우팅 정보를 이용하여 실제 레이블에 의한 포워딩을 수행하는 레이블 포워딩 모듈에 관하여 설명한다.

3.1. 커널 포워딩 모듈

플로우 레이블 포워딩을 사용하는 라우팅 도메인 내에서는 진입/진출(Ingress/Egress) 라우터와 중간(Intermediate) 라우터의 포워딩 구조가 약간 다르다. 진입/진출 라우터에서는 레이블 도메인 내에 들어오는 패킷에 레이블을 할당하는 일과 나가는 패킷의 레이블을 0으로 만들어주는 역할을 한다. 진입 패킷일 때는 커널은 패킷의 IP주소를 이용하여 라우팅 테이블을 검색하고, rt6_info에 저장되어 있는 레이블 정보(outl)를 가져와 플로우 레이블 필드에 저장하며, 진출 패킷일 때는 레이블 값으로 인덱싱되는 rt6_info의 라우팅 정보를 사용해서 라우팅 정보를 가져온다. 그리고 라우팅 캐쉬 정보를 이용하여 패킷을 포워딩하고, 멀티캐스트 전송인 경우 패킷의 {근원지주소, 목적지그룹주소}에 해당하는 MFC의 정보를 이용하여 패킷을 포워딩 한다. 중간 라우터에서는 진입하는 패킷이 도착하면 커널은 단지 플로우 레이블 필드의 값을 꺼내어 레이블 포워드캐쉬의 엔트리를 찾는다. 레이블 포워드캐쉬의 엔트리는 라우팅 캐쉬에 대한 포인터를 가지고 있으므로, 라우팅 캐쉬의 정보와 {근원지 주소, 목적지 그룹주소}에 해당하는 MFC가 가지고 있는 정보를 이용하여 패킷을 포워딩 한다.

3.2 레이블 포워딩 테이블

레이블 포워딩 테이블은 <그림 3>과 같은 구조로 되어있다. 들어오는 패킷의 플로우 레이블을 해서 테이블의 엔트리로 하여 레이블에 해당하는 rt6_info의 포인터를 가지게 하여 레이블을 보고 라우팅 정보를 바로 가져올 수 있게 하였다. 즉 들어오는 패킷의 레이블만 가지고 스위칭을 행한다. 들어온 패킷의 레이블을 보고 해서 테이블의 엔트리를 찾은 후 매치 되는 rt6_info를 찾아 rt6_info에 저장되어 있는 외부로 나가는 레이블(Outgoing Label)을 다시 플로우 레이블 필드에 저장하여 전송한다.



<그림 3> 레이블 포워딩 테이블

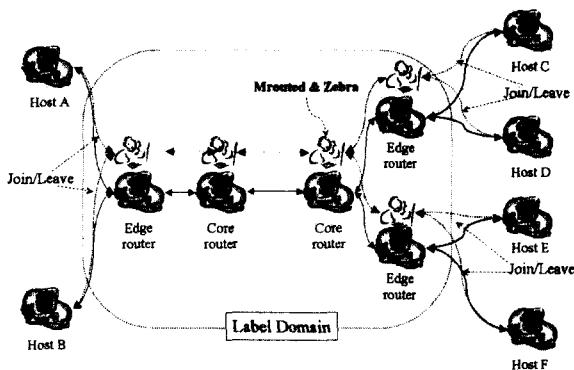
3.3 레이블 동적 할당

멀티캐스트 트래픽을 위한 레이블할당은 mroute가 처음 시작되면서 멀티캐스트 트리를 만들기 위해서 모든 인터페이스로 패킷을 포워딩 한다. 이 패킷을 수신한 중간 라우터들도 절단 메시지를 받기 전까지는 RPM(Reverse Path Multicasting)에 의하여 패킷을 포워딩 한다. 이러한 포워딩 작업에는 포워드 캐쉬가 필요한데, 이 캐쉬를 만드는 과정에서 레이블이 할당된다. 이 과정에서 mroute가 커널의 요청을 받아서 MFC를 업데이트 할 때 커널은 커널내부의 레이블 마스크를 검사하여 할당되지 않은 레이블 중 가장 작은 값을 라우팅캐쉬의 해당필드(rt6_info.ou)에 할당한다. 차후에 단말 서브넷으로부터 절단 메시지가 수신되면 멀티캐스트 포워딩 캐쉬(MFC)와 라우팅 캐쉬, 레이블 테이블의 해당 엔트리는 삭제되고, 레이블 값 역시 레이블 마스크를 조정하여 차후에 재사용 될 수 있게된다.

유니캐스트 트래픽을 위한 레이블 할당도 멀티캐스트와 유사하게 동작하는데 진입/진출 라우터나 중간 라우터에 레이블 패킷이 도착하면 레이블 포워드캐쉬가 만들어지는데 이 과정에서 레이블 포워드캐쉬를 만들 때 커널내부의 레이블 마스크를 검사하여 할당되지 않은 레이블 중 가장 작은 값을 라우팅캐쉬의 해당필드(rt6_info.ou)에 할당한다. 차후 zebra에 의해 해당 루트(route)가 삭제 시 라우팅 캐쉬, 레이블 테이블의 해당 엔트리는 삭제되고, 레이블 값 역시 레이블 마스크를 조정하여 차후에 재사용 될 수 있게 된다.

4. 실험

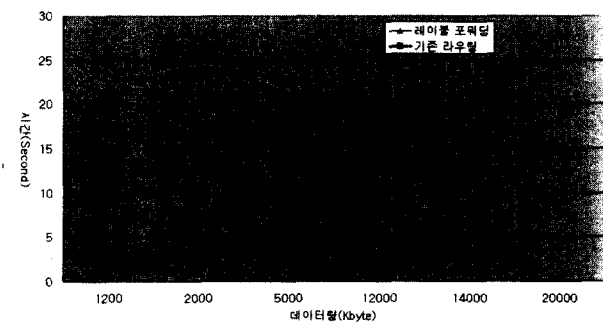
본 논문에서 사용한 실험 망은 <그림 4>와 같으며, 망의 각 호스트에는 IPv6 지원 리눅스(커널 2.2.X)가 설치되어있다. 특



<그림 4> 실험망 토폴로지

히 라우터 호스트에는 레이블 포워딩을 지원하는 수정된 mroute와 zebra가 설치되어있다. 망은 모두 3개의 서브넷으로 구성되어있다. 실험은 멀티캐스트 동작 및 유니캐스트 동작을 위해 간단한 멀티캐스트 송신/수신 프로그램으로 가입/탈퇴를 반복하며 수행하였고, IPv6용으로 수정된 화상회의 프로그램인 Vic을 이용하여 수정된 mroute와 zebra의 동작을 실험하였고, 레이블 포워딩과 일반 IP검색방법간의 성능비교 실험을 하였다.

<그림 5>는 기존의 엔트리를 찾는 방법(longest prefix match)을 이용한 IP주소검색방법과 레이블 값 전체가 일치하는 엔트리를 찾는 방법(short label exact match)을 이용한 레이블을 이용한 포워딩을 비교 실험한 결과로 실험 망에서 호스트에서 호스트로 FTP를 이용하여 데이터를 전송하여 나타낸 결과이다. 기존 라우팅 방식보다 레이블에 의해 전송을 했을 때 최대 13%정도의 성능이 향상됐음을 알 수 있다.



<그림 5> 레이블 포워딩 과 기존 라우팅의 비교

5. 결 론

현재 IPv6의 플로우 레이블 필드의 활용방안에 대해 많은 방안들이 나오고 있는 실정인데, 본 논문에서는 플로우 레이블을 이용해 라우팅이 가능하도록 레이블의 값에 따라 포워딩을 수행하는 방법을 제안하였다. 또한 제안된 플로우 레이블 포워딩 엔진을 이용하여 유니캐스트 라우팅을 구현하였고, IPv6환경에서 지원되지 않았던 멀티캐스트 라우팅까지 확장하여 IPv6환경에서 수행되는 멀티캐스팅 라우팅 데몬을 개발하였다. 또한 실험결과를 통해 IPv6의 플로우 레이블 포워딩 엔진이 일반 IP 검색방법보다 약 13% 더 좋은 성능을 발휘함을 확인하였다.

참고 문헌

- [1] S. Deering and R. Hinden, "Internet Protocol, Version 6(IPv6) Specification", RFC-2460, December 1998.
- [2] Christian Huitema, "Routing in the Internet", PH PTR press, 2000.
- [3] S. Deering, W. Fenner, B. Haberman, "Multicast Listener Discovery(MLD) for IPv6", RFC2710, October 1999.
- [4] T. Pusateri, "Distance Vector Multicast Routing Protocol", Internet Draft, August 2000.
- [5] Deering, S., "Multicast Routing in Internetworks and Extended LANs", SIGCOMM Summer 1998 Proceedings, August 1998.
- [6] Uyless Black, "MPLS and Label Switching Networks", PH PTR press, 2001.
- [7] GNU Zebra, <http://www.zebra.org>.
- [8] G. Malkin, Xylogics, R. Minnear, "RIPng for IPv6", RFC-2080, January 1997.
- [9] Mark A. Miller, P.E. , "Implementing IPv6", M&T Books, 1999.