

플로우 레이블을 지원하는 IPv6멀티캐스트 라우팅 데몬의 구현

이주철^o, 안중석
동국대학교 컴퓨터공학과

The Extension of IPv6 Multicast Routing Daemon For Using the Flow Label

Joo-Chul Lee^o, Jong-Suk Ahn
Dept. of Computer Engineering, Dongguk University

요 약

멀티캐스트 라우팅 데몬은, 패킷 포워딩이 가능한 호스트 상에서 작동하여 호스트를 멀티캐스트 라우팅이 가능한 라우터로 만들어주는 역할을 하는 응용 프로그램이다. 현재의 리눅스 IPv6 멀티캐스트 환경을 살펴보면 로컬 네트워크 상에서의 멀티캐스트 통신만을 지원하도록 되어있다. 즉 서로 다른 서브넷 상에 존재하는 호스트들 사이에서는 멀티캐스트 통신을 할 수 없다. 따라서, 본 논문에서는 리눅스 IPv6 환경에서 멀티캐스트 라우팅이 가능하도록 IPv6용 멀티캐스트 라우팅 데몬을 구현하였다. 멀티캐스트 라우팅이 가능하기 위해서는 두가지 문제가 해결되어야 하는데 첫째는 멀티캐스트 라우팅 정보를 주고받는 데몬 프로그램이고, 둘째는 데몬이 주고받은 라우팅 정보를 이용하여 멀티캐스트 패킷을 포워딩하는 커널 포워딩 루틴이다. 이 두가지가 본 논문에서 중심으로 다룰 내용이다.

1. 서론

인터넷이 처음 생긴 이래로 최근 몇 년간, 인터넷의 사용인구는 폭발적으로 증가해 왔다. 그에 따른 부수적 현상으로 처음엔 충분하리라 여겨졌던 인터넷(IPv4기반)의 여러 한계점들이 드러나기 시작했다. 최근에 인터넷에 접속되어 있는 호스트의 증가속도를 감안하면 현재의 인터넷이 수용할 수 있는 최대 주소의 개수인 2^{32} 를 수년 내에 초과할 것으로 예상되며, 컴퓨터 이외의 지능형 가전기와 인터넷 활용 통신 기기들도 인터넷에 접속되는 추세를 감안하면 현재의 인터넷은 빠른 속도로 포화상태에 이를 것이다. 이러한 문제를 해결하기 위한 방안이 몇 가지 제시되고 있는데, 부족한 IPv4의 주소 영역을 효율적으로 사용하기 위하여 무계층 주소(CIDR)[1] 개념의 도입, 인트라넷 등 내부의 모든 호스트가 외부와 통신할 필요가 없는 네트워크에서는 임시적 IP주소를 고유의 IP주소로 변환해 주는

NAT(Network Address Translation)[2] 등이 그것이다. 하지만 이러한 방법들은 모두 임시 방편적인 것들로 근본적인 해결책은 되지 못한다. 따라서 현재의 인터넷에 존재하는 한계점들을 극복하기 위해서는 궁극적으로 차세대 IP인 IPv6[3] 를 사용해야 할 것으로 생각된다. IPv4와 비교해 볼 때 IPv6는 몇 가지 특징을 갖는데, 그 중 하나로 플로우 레이블(flow label)이 헤더에 포함된 점을 들 수 있다. 플로우 레이블은 패킷이 지니고 온 레이블 필드의 값에 따라 차별적인 서비스를 제공해 주기 위한 용도를 예상하고 고안되었다. 즉 IPv6라우터로 하여금 같은 플로우 레이블 값을 가진 패킷들에 대하여 특별한 처리를 해주도록 하는 것이다. 본 논문에서는 이러한 레이블의 용도에 추가하여 레이블의 값을 보고 포워딩 하는 방법을 제안하고, 이 포워딩 엔진을 기반으로 하여 멀티캐스트 라우팅 데몬을 구현하였다.

현재 리눅스 커널(2.2.X)에서는 부분적으로 IPv6코드를 지원하고 있는데, 같은 로컬 네트워크 상에서의 멀티캐스트 통신은 지원하지만 아직 멀티캐스트 라우팅은

지원하지 않고 있다. IPv4에서 작동하는 멀티캐스트 라우팅 데몬(mrouted)[4][6]을 기준으로 살펴보면, 멀티캐스트 라우팅은 실제 라우팅과 관련된 메시지를 주고 받는 데몬과, 데몬이 유지하는 라우팅 정보에 기반해서 실제 패킷을 포워딩 하는 커널내부의 지원 모듈로 구성이 되어있다. 이 중에서 커널내부의 지원모듈을 앞에서 소개한 플로우레이블 포워딩 방법을 이용해서 구현하고, 라우팅 데몬은 mrouted의 소스를 참고하여 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 구현에 관련된 기반 메커니즘에 대해 기술하고, 3장에서 구현을 테스트한 결과를 설명한다. 그리고 마지막으로 4장에서 결론과 향후계획을 기술한다.

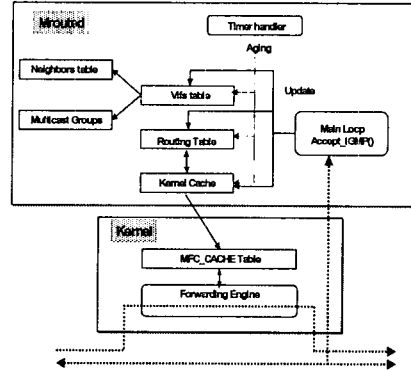
2. 관련연구

2.1. mrouted 개관

mrouted는 유닉스 환경에서 데몬(daemon)형태로 구동되는 프로그램으로 패킷 포워딩 기능을 가진 호스트 상에서 구동되어 호스트가 멀티캐스트 라우터 역할을 하도록 해준다. mrouted가 취급하는 정보는 크게 멀티캐스트 라우팅 도메인(domain)에서 일반 호스트들과 직접 연결되어있는 가장자리 라우터(edge router)가 무선적으로 접하게 되는 그룹멤버의 가입과 탈퇴에 대한 정보와, 도메인 내의 라우터들끼리 서로 주고받는 라우팅 테이블에 대한 정보, 두 가지가 있다. 이 정보들을 이용해서 특정 근원지주소와 그룹주소를 가진 패킷이 왔을 때, 라우터의 입장에서 어느 인터페이스로 패킷을 포워딩 할 것인가를 결정하게 된다. 만약 하나 이상의 인터페이스로 패킷이 포워딩 되어야 한다면, 커널에서 인터페이스의 개수만큼 패킷이 복사되어 각각 전송되어지게 된다. 멀티캐스트 트래픽의 전송경로 상에 위치한 각각의 라우터가 가지고 있는 이러한 정보들을 종합해보면 근원지와 수신자들 사이에 트리 구조가 형성되는데, 이 트리를 멀티캐스트 트리라 한다.

이 정보들은 대부분 mrouted내에서 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. 이러한 자료구조의 에이징(aging)은 mrouted의 타이머 핸들러가 맡고 있다. 처음 mrouted가 구동이 되면, 타이머 시그널에 대한 핸들러가 등록되고 매 초마다 타이머 핸들러가 구동이 되어 라우팅 테이블의 유효기간 초과와, 그룹멤버의 존재

유무 등을 지속적으로 조사하게 된다. [그림 1]은 이러한 동작구조를 설명하고 있다.



[그림 1] mrouted 블록도

2.2. 라우팅 메시지 처리

이 장에서는 주요한 라우팅메시지의 처리에 관하여 설명한다. ()앞의 명칭은 리눅스 헤더파일에 정의되어 있는 메시지에 대한 상수 이름이고, ()안은 MLD(Multicast Listener Discovery[5])에서의 명칭이다.

① ICMP6_MEMBERSHIP_QUERY(Multicast Listener Query)

이 메시지를 라우터가 받았다면, 같은 서브넷 상에 가장자리 라우터가 하나 이상 있다는 의미이다. 만약 질의를 보낸 라우터의 주소가 질의를 받은 라우터의 주소보다 낮다면, 메시지를 수신한 라우터는 질의자의 자격을 메시지 송신자에게 양보한다.

② ICMP6_MEMBERSHIP_REPORT(Multicast Listener Report)

이 메시지에 담겨있는 그룹주소가 메시지를 받은 인터페이스가 가입하고있는 멀티캐스트 그룹테이블에 존재한다면, 그룹주소의 만료시간(expiration time)을 업데이트 한다. 만약 존재하지 않으면 새로운 그룹주소 엔트리를 추가하고, 새로 추가된 인터페이스로 데이터를 보낼 수 있도록 MFC를 업데이트 한다.

③ ICMP6_MEMBERSHIP_REDUCTION(Multicast Listener Done)

이 메시지를 받으면 본 메시지에 포함된 그룹주소로 ICMP6_MEMBERSHIP_QUERY를 보낸 후 타이머를 세트한다. 타이머가 만료될 때까지 ICMP6_MEMBERSHIP_QUERY에 대

한 응답이 오지 않는다면 더 이상 멤버가 없는 것으로 간주하고, 인터페이스의 그룹 테이블에서 그룹주소를 제거한다. 만약 그 그룹주소에 가입한 인터페이스가 더 이상 없다면 상위 라우터로 prune 메시지를 보낸다.

④ DVMRP_PROBE

이 메시지를 보낸 라우터의 주소를 자신의 이웃 (neighbor) 라우터 목록에서 검색해서 만약 없으면 자신의 목록에 추가한 후, 원래 probe 메시지의 이웃 목록에도 추가하여 재 전송한다.

⑤ DVMRP_REPORT

이 메시지에 들어있는 라우팅 엔트리들을 하나의 테이블로 작성한 후에 각 엔트리들을 mrouter가 유지하고 있는 라우팅 테이블의 각 항목과 비교하여 비용이 적게 드는 쪽으로 업데이트 한다. 라우팅 엔트리에 업데이트가 발생하게 되면 라우팅 엔트리에 의존하고 있는 MFC도 업데이트 해준다.

⑥ DVMRP_PRUNE

이 메시지의 대상 그룹주소가 해당 MFC의 prune 테이블에 존재한다면 이전에 prune 메시지를 받았던 적이 있는 경우이므로 타이머를 업데이트 해주고, 만약 새로 받은 prune 메시지라면 MFC의 prune 테이블에 추가해주고 메시지를 수신한 인터페이스를 포워딩 목록에서 제거한다. 커널 MFC도 업데이트 해준다.

⑦ DVMRP_GRAFT

메시지를 수신한 인터페이스를 포워딩 목록에 추가시켜주고, graft에 대한 ack를 보내준다.

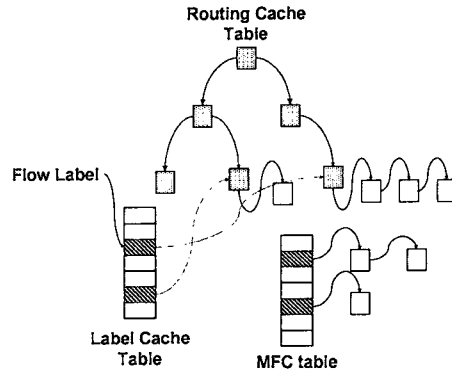
2.3. 커널 mrouter 지원 모듈

이 장에서는 mrouter가 유지 관리하는 라우팅정보를 이용하여 실제 패킷 포워딩을 수행하는 커널 지원 모듈에 관하여 설명한다.

2.3.1. 커널 캐쉬 관리 모듈

mrouter가 가지고있는 라우팅 정보를 이용해 포워딩을 하려면 커널 내에 포워딩 시 참조할 캐쉬(MFC)와, 이 캐쉬를 이용해 실제 포워딩을 해주는 포워딩 엔진이 필요하다. 커널 MFC구조는 mrouter가 라우팅 정보에 의거하여 구성한 포워딩 정보를 담고있는 자료구조이다. 처음 mrouter가 구동되면 RPM(Reverse Path Multicast)[8]방법으로 멀티캐스트 트리를 만들기 위해 하나의 패킷을 모든 인터페이스로 멀티캐스팅하게 되는데, 이 패킷을 받은 라우터는 패킷의 플로우 레이블 값으로 레이블 테이블에 해당 엔트리가 있는지 찾는다. ([그림

2]참고).



[그림 2] 커널 레이블 포워딩 엔진

만약 발견되지 않으면 패킷을 담고있는 소켓버퍼의 정보를 이용하여 새로운 라우팅 캐쉬 엔트리를 할당한다. 그런 후에 받은 패킷을 포워딩 할 인터페이스를 결정하기 위해 MFC 테이블을 찾는다. 만약 이 (근원지주소, 목적지그룹주소)의 MFC엔트리가 테이블에 존재하지 않는다면 소켓버퍼의 정보를 이용하여 새로운 엔트리를 추가한다. 패킷이 가진 정보만을 이용해서 MFC의 모든 정보를 채울 수는 없으므로, 이 사실을 mrouter에게 알리기 위해서 수신한 패킷을 복제한 후, 사본 IPv6헤더의 다음헤더(next header)필드를 0으로 만들어서 mrouter가 메시지 수신용으로 생성한 소켓에 쓴다. 원본 패킷은 차후에 MFC의 정보가 보충된 후에 포워딩 될 수 있도록 MFC의 mfc6_cache.mfc_unresolved리스트에 큐잉 시킨다. 이렇게 생성된 MFC는 mrouter가 MFC를 업데이트할 때마다, mrouter의 MFC 제어구조에 의해 유지 관리되어진다.

2.3.2. 커널 포워딩 모듈

플로우 레이블 포워딩을 사용하는 멀티캐스트 라우팅 도메인 내에서는 진입/진출(Ingress/Egress) 라우터와 중간(Intermediate) 라우터의 포워딩 구조가 약간 다르다. 진입/진출 라우터에 레이블 포워딩 도메인으로 진입하는 패킷이 도착하면 커널은 패킷의 IP주소에 기반하여 라우팅 테이블을 검색하고, 검색된 라우팅 캐쉬 엔트리(rt6_info)에 저장되어있는 레이블 값(outl)을 다시 패킷의 플로우 레이블 필드에 저장한다. 그리고 패킷의 (근원지주소, 목적지그룹주소)에 해당하는 MFC의 정보를 이용하여 패킷을 포워딩 한다.

진입/진출 라우터에 레이블 포워딩 도메인에서 빠져나가는 패킷이 도착하는 경우의 처리는 다음과 같다.

먼저 패킷의 플로우 레이블 필드에서 레이블 값을 얻어 내어 레이블 테이블의 엔트리를 찾는다. 이렇게 찾아진 엔트리는 라우팅 캐쉬의 적합한 엔트리를 가리키는 포인터를 가지고 있으므로 패킷의 플로우 레이블 필드를 0으로 만들고 (근원지주소, 목적지그룹주소) MFC를 찾아서 해당하는 인터페이스로 포워딩 해준다. ([그림 2] 참고) 중간 라우터에서는 진입/진출 라우터에 패킷이 진출방향으로 도착했을 때와 유사한 작동을 한다. 커널은 단지 플로우 레이블필드의 값을 꺼내어 레이블 캐쉬의 엔트리를 찾는다. 레이블 캐쉬의 엔트리는 라우팅 캐쉬에 대한 포인터를 가지고 있으므로, 라우팅 캐쉬의 정보와 (근원지 주소, 목적지 그룹주소)에 해당하는 MFC가 가지고 있는 정보를 이용하여 패킷을 포워딩 한다.

2.3.3. 레이블의 할당

mouted는 처음 시작되면 멀티캐스트 트리를 만들기 위해서 모든 인터페이스로 패킷을 포워딩 한다. 이 패킷을 수신한 중간 라우터들도 절단 메시지를 받기 전까지는 RPM에 의하여 패킷을 포워딩 한다. 이러한 포워딩 작업에는 포워딩 캐쉬가 필요한데, 이 캐쉬를 만드는 과정에서 레이블이 할당된다. 진입/진출 라우터나 중간 라우터에 레이블 패킷이 도착하면 2.3.1에서와 같은 과정으로 커널 캐쉬가 만들어지게 된다. 이 과정에서 mouted가 커널의 요청을 받아서 MFC를 업데이트 할 때 커널은 커널내부의 레이블 마스크를 검사하여 할당되지 않은 레이블 중 가장 작은 값을 라우팅캐쉬의 해당필드(rt6_info.oul)에 할당한다. 이렇게 할당된 레이블 값은 패킷이 전송되어질 때 IPv6의 플로우 레이블 필드에 넣어지게 되며, 패킷이 가지고 들어온 레이블 값은 해쉬 인덱스 값으로 사용되어 패킷이 들어왔을 때 레이블 테이블에서 해당하는 엔트리를 찾을 수 있게 한다. 차후에 단말 서버넷으로부터 절단 메시지가 수신되면 포워딩 캐쉬와 라우팅 캐쉬, 레이블 테이블의 해당 엔트리는 삭제되고, 레이블 값 역시 레이블 마스크를 조정하여 차후에 재사용 될 수 있게된다.

3. 실험

본 논문에서 사용한 실험 망은, 본래 IPv6의 플로우 레이블을 이용한 QoS 알고리즘을 개발하기 위해 구축한 테스트베드에서 이루어졌다. 본 망은 KAIST와 256kbps의 전용선을 통해 연결되어 있다. 실험을 위한 각 호스트와 mouted가 운영되고 있는 호스트에는 모두 IPv6가 작동되도록 컴파일 된 리눅스가 설치되어있다. 서버넷

은 동국대 2개, KAIST 1개로 모두 3개의 서버넷으로 구성되어있으며 동국대와 KAIST는 256kbps의 전용선으로 중간(intermediate)라우터를 연결하고 있다. 실험은 간단한 멀티캐스트 프로그램과 IPv6용으로 수정된 화상회의 프로그램인 Vic을 이용하여 가입/탈퇴를 반복하며 수행하였다. 작동과정에 대한 검증은 Tcpdump를과 mouted의 디버그모드를 이용하여 멀티캐스트 트래픽이 올바르게 전송/중지 되는지 확인하였다. 실험망 설정은 [표1]과 같다.

서버넷	Prefix	IP address	가입한 그룹주소
동국대1	3ffe:2e01:8:2::/64	3ffe:2e01:8:2:0:01:5a76:7bdc/64	ff45::50/16
		3ffe:2e01:8:2:0:60:97b3:769b/64	ff45::55/16
동국대2	3ffe:2e01:8:5::/64	3ffe:2e01:8:5:0:10:5a71:7759/64	ff45::50/16
		3ffe:2e01:8:5:0:01:0292:5271/64	ff45::55/16
KAIST	3ffe:2e01:8:4::/64	3ffe:2e01:8:4:0:a0:24bd:3147/64	ff45::50/16
		3ffe:2e01:8:4:0:80:3e7d:69e6/64	ff45::55/16

[표1] 실험망 설정

[그림 3]과 [그림 4]는 디버그 모드에서 작동하고 있는 mouted의 출력 결과를 캡처한 것으로, [그림 3]은 중간 라우터의 각 인터페이스에 이웃하고 있는 이웃(peer)에 대한 정보와 라우팅 테이블을 보여주고 있다. 이웃에 대한 정보는 멀티캐스트 트리를 유지할 때 절단 메시지나 접목메시지를 직접 보낼 목적지 주소로써 쓰이게 된다. 라우팅 테이블은 현재의 라우터로 멀티캐스트 트래픽을 보낼 수 있는 호스트가 속한 서버넷들과, 특정 서버넷으로부터 트래픽이 올 때 현재 라우터의 바로 이전에 거쳐야 할 라우터(from-gateway)에 대한 정보를 보여준다. [그림 4]는 그룹에 조인한 호스트와 직접 접해있는 가장자리 라우터의 정보를 표시한 그림이다. 중간 라우터와 마찬가지로 이웃에 대한 정보를 가지고 있으며, 특정 인터페이스에 조인한 그룹들의 목록을 추가로 유지하고 있다. 특정 그룹에 속한 멤버가 탈퇴메시지를 가장자리 라우터에게 보내면, 가장자리 라우터는 그 그룹을 대상으로 그룹질의를 보내어 일정시간동안 응답이 없으면 조인그룹의 목록에서 삭제한다.

[그림 5]는 간단한 멀티캐스트 송신/수신 프로그램을 수행시키면서 가장자리 라우터에서 MLD메시지를 주고받는 과정을 Tcpdump로 캡처한 그림이다. 그룹 질의 메시지는 ipv6-allnodes(ff02::01)로 전송되며, v6jindo0가 ff45::50, ff45::90, ff46::45, ff50::50등의 그룹에 조인해 있음을 알수 있다. icmpv6 type#140이라고 표시된 메시지는 Probe 메시지이다. 그 위쪽의 캡처된 udp 패킷들은 멀티캐스트 주소 ff50::50과 ff46::45로 전송되고 있는 것들이다.

```

Virtual Interface Table
Vif Name Local-Address M Thr Rate Flags
0 eth0 3ffe:2e01:8:1:0 10:5a80:d842:3ffe:2e01:8:1 1 1 0 leaf
  peers: 3ffe:2e01:8:1:0:10:5a76:3487 (Ox)
  pkts in 0
  pkts out 0
1 eth1 3ffe:2e01:8:3:0 50 da88:2721:3ffe:2e01:8:3 1 1 0 leaf
  peers: 3ffe:2e01:8:3:0:10:5a80:d874 (Ox)
  pkts in 0
  pkts out 0
2 eth2 3ffe:2e01:8:6:0 10:5a76:3656:3ffe:2e01:8:6 1 1 0 querier leaf
  pkts in 0
  pkts out 0

Multicast Routes Table (5 entries)
Orig:Subnet From-Gateway Metric Thr In-Vif Out-Vifs
3ffe:2e01:8:6 1 56 2 0 1
3ffe:2e01:8:5 3ffe:2e01:8:1:0:10:5a76:3487 2 40 0 1 2*
3ffe:2e01:8:3 1 56 1 0 2*
3ffe:2e01:8:2 2 0 1 0 2*
3ffe:2e01:8:1 1 56 0 1 2*
    
```

[그림 3] 디버그 모드: 중간 라우터의 인터페이스상태와 이웃의 정보

```

Virtual Interface Table
Vif Name Local-Address M Thr Rate Flags
0 eth0 3ffe:2e01:8:1:0 10:5a76:3487:3ffe:2e01:8:1 1 1 0 querier
  peers: 3ffe:2e01:8:1:0:10:5a80:d842 (Ox)
  pkts in 0
  pkts out 5
1 eth1 3ffe:2e01:8:5:0 10:5a76:3650:3ffe:2e01:8:5 1 1 0 querier leaf
  groups: 3ffe:2e01:8:5:0:10:5a76:3650
  pkts in 10
  pkts out 0

Multicast Routes Table (5 entries)
Orig:Subnet From-Gateway Metric Thr In-Vif Out-Vifs
3ffe:2e01:8:6 2 40 0 1*
3ffe:2e01:8:5 1 160 1 0
3ffe:2e01:8:3 2 40 0 1*
3ffe:2e01:8:2 3 40 0 1*
3ffe:2e01:8:1 1 160 0 1*
    
```

[그림 4] 디버그 모드: 가장자리 라우터에 조인한 그룹과 이웃의 정보

```

20:01:01.658318 v6jindo0 > ff50::50 v6jindo0.40000 > ff50::50.40000: udp 1
20:01:01.857761 3ffe:2e01:8:5:0:1:292:5271 > ff46::45 3ffe:2e01:8:5:0:1:292:5271: Membership
ff46::45:30000: udp 1
20:01:02.668321 v6jindo0 > ff50::50 v6jindo0.40000 > ff50::50.40000: Query
20:01:02.867751 3ffe:2e01:8:5:0:1:292:5271 > ff46::45 3ffe:2e01:8:5:0:1:292:5271: Membership
ff46::45:30000: udp 1
20:01:03.877738 3ffe:2e01:8:5:0:1:292:5271 > ff50::50 v6jindo0.40000 > ff50::50.40000: udp 1
20:01:03.877738 3ffe:2e01:8:5:0:1:292:5271 > ff46::45 3ffe:2e01:8:5:0:1:292:5271.1025 >
ff46::45:30000: udp 1
20:01:04.097368 fe80::210:5aff:fe80:d874::1:ipv6-allnodes icmpv6: query group
memberhip
20:01:04.097607 v6soyang1 > ff02::4:icmpv6: query group
20:01:04.098327 v6jindo0 > ff50::50 v6jindo0.40000 > ff50::50.40000: Membership
20:01:04.098407 v6jindo0 > ff45::90 icmpv6: join mcast group
20:01:04.098488 v6jindo0 > ff46::45 icmpv6: join mcast group
20:01:04.098570 v6jindo0 > ff50::50 icmpv6: join mcast group
    
```

[그림 5] 그룹의 가입과 탈퇴

4. 결론 및 향후연구계획

IPv6는 현재 인터넷이 당면하고 있는 문제점을 근본적으로 극복할 수 있는 유일한 해결책이다. 이에 대비하여 이미 많은 업체와 연구소등에서 IPv6 솔루션개발에 참여하고 있는 실정으로, 특히 응용프로그램의 기반이 되는 운영체제레벨에서의 IPv6지원은 중요한 문제라고 생각된다. 현재 데스크톱 운영체제의 90%이상을 점

유하고 있는 마이크로소프트사에서 자사의 윈도우 운영체제에서 작동할 수 있는 IPv6솔루션을 발표한 상태이며, 최근 주목받고있는 리눅스 운영체제에서도 실험적이거나 코드가 구현되어있는 상태이다. 하지만 아직 멀티캐스트에 대한 부분은 개발이 미약한 상태로, 특히 멀티캐스트 라우팅 쪽은 극히 미약한 상태이다. 멀티캐스팅은 최근 급증하고 있는 멀티미디어 스트림 서비스등을 제공할 때 네트워크의 부하를 줄일 수 있는 솔루션으로써 가까운 장래에 반드시 범용화 되어질 것으로 보이는 중요한 기술이다. 이러한 추세에 맞추어 본 논문에서는 IPv6테스트 베드를 구축하고 IPv6의 플로우레이블을 기반으로 하는 포워딩 엔진을 구현하였으며, 본 엔진과 IPv4환경의 멀티캐스트 라우팅 데몬을 기반으로 하여 IPv6환경에서 수행되는 라우팅데몬을 개발하였다.

본 연구를 통하여 망 구축경험과, 차세대 IP인 IPv6에 대한 지식을 습득 할 수 있었고, 리눅스 커널소스를 읽고 수정하는 것을 통해 운영체제에 대한 지식도 얻을 수 있는 계기가 되었다. 향후 연구 계획으로는 RSVP를 수정하여 플로우 레이블의 값에 따라 자원을 예약할 수 있도록 하는 것과, 유니캐스트 환경에서 플로우 레이블을 자동으로 설정하는 모듈을 구현하는 것이다.

[참고문헌]

- [1] V. Fuller, T.Li, J. Yu and K. Varadhan, "Classless Inter-Domain Routing (CIDR):an Address Assignment and Aggregation Strategy", RFC-1519, September 1993
- [2] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)", RFC-1631, May 1994
- [3] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC-2460, December 1998
- [4] Christian Huitema, "Routing in the Internet", PH PTR press 2000
- [5] S. Deering, W. Fenner, B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC2710, October 1999
- [6] T. Pusateri, "Distance Vector Multicast Routing Protocol", Internet Draft, August 2000
- [7] W. Fenner, "Internet Group Management Protocol, Version 2", RFC-2236, November 1997
- [8] Deering, S., "Multicast Routing in Internetworks and Extended LANs", SIGCOMM Summer 1988 Proceedings, August 1988.