

UNI 4.0 기반 ATM 망에서의 IP 멀티캐스트 지원 방안을 위한 서비스 구조

(A Service Architecture to support IP Multicast Service over UNI 4.0 based ATM Networks)

이 미 정[†] 정 선^{**} 김 예 경^{**}

(MeeJeong Lee) (Sun Jung) (YeKyung Kim)

요약 인터넷에서의 대부분의 중요한 실시간 멀티미디어 응용들은 멀티캐스트 서비스를 요구하므로 이들 응용을 ATM 기반의 인터넷에서 지원하기 위해서는 IP 멀티캐스트의 효율적인 처리가 필수적이다. IETF에서는 UNI 3.0/3.1 기반 ATM 망에서 연결형 일대다 ATM VC로 비연결형의 IP 멀티캐스트를 지원하기 위해 MARS(Multicast Address Resolution Server) 모델을 제안하였다. UNI 3.0/3.1 시그널링으로는 동적으로 멀티캐스트 그룹에 참여하려는 수신원을 VC 연결의 주체인 송신원에게 알릴 수 없기 때문에, MARS는 동적으로 멀티캐스트 그룹에 참여하려는 멤버들의 요청을 받아 이를 송신원에게 알려주는 오버레이 서비스를 제공한다. 그런데 ATM Forum이 UNI 3.0/3.1 시그널링을 수정, 보완하여 표준화한 UNI 4.0 시그널링에서는 일대다 ATM VC에 참여하기 위해 수신원이 직접 송신원으로 신호할 수 있는 LIJ(Leaf Initiated Join) 기능이 제안되었다. 본 논문에서는 UNI 4.0 기반의 ATM 망에서 LIJ 기능을 이용하여 IP 멀티캐스트 서비스를 제공하는 UNI4MARS 모델을 제안한다. MARS 서버와는 대조적으로, 제안하는 UNI4MARS 서버는 수신원이 LIJ 요청을 할 때 필요로 하는 송신원 정보를 알려주기 위하여 멀티캐스트 그룹에 대한 송신원의 등록/탈퇴 정보를 유지한다. 시뮬레이션을 통하여 비교해 본 결과, 제안하는 UNI4MARS 모델은 동적인 IP 멀티캐스트를 지원하는 기능적인 면에서는 기존의 MARS 모델과 동일하지만 송신원에 비하여 수신원의 수가 많은 비대칭적 다중점 대 다중점 응용의 경우 구현에 필요한 프로세싱 오버헤드 및 메모리와 대역폭 요구량 측면에서 MARS 모델보다 유리함을 알 수 있었다.

Abstract Most of the important real time multimedia applications require multipoint transmissions. To support these applications in ATM based Internet environments, it is important to provide efficient IP multicast transportations over ATM networks. IETF proposed MARS(Multicast Address Resolution Server) as the service architecture to transport connectionless IP multicast flows over connection oriented ATM VCs. MARS assumes UNI3.0/3.1 signalling. Since UNI3.0/3.1 does not provide any means for receivers to request a join for a multicast ATM VC, MARS provides overlay service to relay join request from IP multicast group members to the sources of the multicast group. Later on, ATM Forum standardized UNI4.0 signalling, which is provisioned with a new signalling mechanism called LIJ(Leaf Initiated Join). LIJ enables receivers to directly signal the source of an ATM VC to join. In this paper, we propose a new service architecture providing IP multicast flow transportation over ATM networks deploying UNI4.0 signalling. The proposed architecture is named UNI4MARS. It comprises service components same as those of the MARS. The main functionality provided by the UNI4MARS is to provide source information to the receivers so that the receivers may exploit LIJ to join multicast ATM VCs dynamically. The implementation overhead of UNI4MARS and that of MARS are compared by a course of simulations. The simulation results show that the UNI4MARS supports the dynamic IP multicast group changes more efficiently with respect to processing, memory and bandwidth overhead.

· 이 논문은 1998년 정보통신연구진흥원의 대학기초 연구지원사업 공모과제에 의하여 연구되었음.

† 정 회 원 : 이화여자대학교 컴퓨터학과 교수
lmj@mm.ewha.ac.kr

** 비 회 원 : 이화여자대학교 컴퓨터학과
982COG33@mm.ewha.ac.kr
982COG30@mm.ewha.ac.kr

논문접수 : 1999년 9월 14일
심사완료 : 2000년 5월 6일

1. 서론

인터넷에서의 중요한 실시간 멀티미디어 응용들은 대부분 멀티캐스트 서비스를 요구한다. ATM 망을 기반으로 하는 인터넷에서 이러한 응용들을 효과적으로 지원하기 위해서는 IP(Internet Protocol) 멀티캐스트의 효율적인 처리가 필수적이다. IETF에서는 ATM 일대다 VC를 사용하여 비연결형 IP 멀티캐스트를 지원하기 위한 서비스 구조로서 MARS(Multicast Address Resolution Server)를 제안하였다[1]. MARS 모델은 ATM의 UNI 3.0/3.1 시그널링을 기반으로 하며, IP 멀티캐스트 주소와 ATM 주소간의 매핑 및 IP/ATM 단말들의 동적인 멀티캐스트 그룹 참여/탈퇴를 지원하고 있다. IP 멀티캐스트는 추상적인 그룹 주소에 수신원들이 동적으로 참여함으로써 이루어지게 되는 반면, UNI 3.0/3.1 시그널링으로는 동적으로 멀티캐스트 그룹에 참여하려는 수신원을 VC 연결의 주체인 송신원에게 알릴 수 없기 때문에, MARS는 동적으로 멀티캐스트 그룹에 참여하려는 멤버들의 요청을 받아 이를 송신원에 알려주는 오버레이 서비스를 제공한다.

ATM Forum은 UNI 3.0/3.1 시그널링을 수정, 보완하여 여러 가지 기능들을 추가한 UNI 4.0 시그널링을 1996년에 표준화하였다. UNI 4.0 시그널링에서는 멀티캐스트 그룹의 수신원이 직접 송신원에게 참여를 신청하는 시그널링을 할 수 있도록 해주는 LIJ(Leaf Initiated Join) 기능이 새롭게 제공되었다[2]. 이러한 UNI 4.0의 LIJ 기능을 활용한다면 MARS의 기본적인 기능인 수신자를 송신자에게 알려주는 기능은 불필요하게 된다. 반면, IP 멀티캐스트 그룹의 송신원도 동적으로 변하고 한 개 이상일 수 있으므로 이러한 경우 수신원들이 송신원으로 LIJ 요청 시그널링을 보낼 수 있기 위해서는 멀티캐스트 그룹의 송신원을 그 그룹에 참여하려는 수신원에게 알려주는 서비스가 필요하다[3].

[4]에서는 멀티캐스트 그룹의 수신원이 LIJ 기능을 이용하여 송신원으로 연결을 시도할 때, LIJ 시그널링에 필요한 송신원의 ATM 주소를 제공해주는 서비스 구조인 UNI4MARS가 제안된 바 있다. UNI4MARS는 MARS와 마찬가지로 클러스터 단위에 적용되며 멀티캐스트 주소 해석을 위한 서버인 UNI4MARS를 사용한다. 단, UNI4MARS는 MARS와는 반대로 수신원들에게 멀티캐스트 그룹의 송신원들의 주소를 알려주는 역할을 한다. 그러나 UNI4MARS도 MARS와 마찬가지로 멀티캐스트 그룹 멤버의 동적인 참여/탈퇴를 지원하며 VC 메쉬구조와 MCS구조 모두 적용 가능하다. 본 논문에서는 [4]에서 제안된 UNI4MARS의 기본적인 구조와 기능을

UNI4MARS구조의 구성 엔티티들간 프리미티브 교환과 이에 관련된 프로세싱 및 유지하는 데이터 구조와 VC 관리 측면에서 자세히 정의하고, 그 성능과 특성을 보여 주기 위한 시뮬레이션 결과를 분석하였다.

[5]에서도 인터넷 멀티캐스트 데이터 전송을 지원해주는 새로운 구조로서 EMARS(Enhanced MARS)라는 서비스 구조가 제안된 바 있다. EMARS와 UNI4MARS는 멀티캐스트 그룹의 멤버가 해당 ATM VC에 연결되기 위해 LIJ 시그널링을 사용한다는 면에서는 동일하다. 그러나, EMARS와 UNI4MARS는 각각 해당 주소해석 서버인 MARS와 UNI4MARS에서 멀티캐스트 그룹의 수신원들을 관리하는 측면에서는 차이점이 있다.

EMARS의 경우에는 멀티캐스트 그룹에 참여하려는 모든 호스트는 MARS로부터 그룹의 다른 멤버들의 정보를 얻기 위해서 클러스터 멤버로써 등록을 하게 된다. 반면에, UNI4MARS에서는 UNI4MARS에게 멀티캐스트 그룹의 송신원이 누구인지를 묻는 질의만으로도 클러스터 멤버가 되도록 하고 있다. 즉, UNI4MARS는 질의한 멤버에 관해서는 주소 정보를 유지하지는 않지만 이를 자신의 클러스터 제어 VC에 연결시켜두어 추후 동적인 멤버쉽 변화를 통보 받을 수 있도록 한다. 이는 LIJ 특성상 수신원 정보는 다른 멤버들이 알 필요가 없기 때문에 송신원 호스트의 정보만을 유지하기 위함이다. 결과적으로, EMARS에서는 모든 멤버들로부터 다른 모든 멤버들의 일대다 VC가 형성되는 반면 UNI4MARS에서는 그룹 멤버중 실질적으로 송신원의 역할을 하는 멤버만이 등록해도 클러스터 멤버관리가 가능하도록 하므로써 멤버중 송신원으로부터만 일대다 VC가 형성될 수 있다.

이러한 UNI4MARS의 특성은 순전히 수신원으로만 작동하는 멤버가 많은 비대칭적인 형태의 다중점 대 다중점 IP 멀티캐스트 응용의 경우 매우 유리할 수 있다. 즉, 제안하는 UNI4MARS 모델은 동적인 IP 멀티캐스트를 지원하는 기능적인 면에서는 기존의 MARS 모델이나 EMARS와 동일하지만 멀티캐스트 그룹의 규모가 크면서 송수신원의 비율이 비대칭적인 응용의 경우, 프로세싱 오버헤드, 메모리 및 대역폭 요구량 측면에서 매우 유리한 특성을 가지고 있다. 본 논문에서는 시뮬레이션을 통하여 이러한 UNI4MARS의 성능 특성을 보이고, 그 결과를 MARS와 비교하였다.

본 논문의 구성은 다음과 같다. 2장에서는 UNI 3.0/3.1 기반 ATM 망에서의 IP 멀티캐스트 지원 서비스 구조인 MARS와 UNI 4.0 시그널링 서비스에서 새롭게 지원되는 LIJ 기능을 설명한다. 3장에서는 UNI 4.0의 LIJ 기능을 이용하여 효율적으로 IP 멀티캐스트를 지원하는 UNI4

MARS 모델을 설명하고 4장에서는 MARS와 UNI4MARS의 구현 오버헤드를 비교하기 위한 시뮬레이션 및 그 결과를 설명한다. 끝으로 5장에서 결론을 맺는다.

2. 관련연구

2.1 UNI 3.0 기반 ATM 망에서의 IP 멀티캐스트 지원 구조: MARS 모델

MARS는 연결형의 ATM 망에 비연결형 IP 멀티캐스트 서비스를 지원하기 위한 프로토콜로서 UNI 3.0/3.1 기반의 ATM 망을 대상으로 고안되었다[1]. MARS는 클러스터라는 단위로 3계층 멀티캐스트 그룹에 대한 ATM 단말들의 등록을 담당한다. 클러스터란 MARS 서버를 중심으로 형성되는 멀티캐스트 멤버들을 일컫는다. MARS 클라이언트라 불리는 ATM 단말들은 MARS에게 MARS_JOIN이라는 등록 메시지를 보내어 자신의 ATM 주소와 참여하기 원하는 멀티캐스트 그룹 주소를 등록하고, 탈퇴하기 위해서는 MARS_LEAVE 메시지를 보낸다. MARS는 MARS 클라이언트들이 각 멀티캐스트 그룹 주소에 등록/탈퇴한 정보를 유지하여 어떤 멀티캐스트 그룹의 송신원인 클라이언트의 요구가 있을 때 멀티캐스트 그룹에 대한 그룹 멤버들의 ATM 인터페이스를 알려줌으로써 송신원이 수신원들로 향하는 일대다 VC를 설정할 수 있도록 해준다.

UNI 3.0/3.1 환경에서 일대다 VC를 설정하기 위해서는 일대다 VC의 자식에 해당하는 수신원들을 송신원이 알아야 하는데 UNI 3.0/3.1 신호로는 동적으로 참여하는 IP 멀티캐스트 그룹의 수신원들이 송신원에게 자신을 알릴 수 있는 신호 방법이 없다. 구체적으로 UNI 3.0/3.1 시그널링을 이용하여 일대다 VC가 설정되는 과정을 살펴보면, 송신원은 먼저 SETUP 메시지를 사용하여 자식 중 하나에게 일대다 VC를 설정하고, ADD_PARTY/DROP_PARTY 메시지를 사용하여 이 후에 발생하는 자식들의 멀티캐스트 그룹 추가/탈퇴를 위한 일대다 VC 변경을 신호한다. 따라서 송신원이 MARS를 통해 멀티캐스트 그룹에 해당하는 수신원들의 ATM 인터페이스를 알아내야만 IP 멀티캐스트를 위한 일대다 VC 설정 및 변경 작업이 가능한 것이다. MARS 클라이언트는 멀티캐스트 그룹 멤버들에 대한 질의를 보내고 응답을 받기 위해 MARS로 일시적인 양방향 일대일 VC를 설정하며, MARS가 멀티캐스트 그룹의 변화를 클라이언트들에게 알리기 위하여 MARS와 클라이언트들간에 반영구적인 일대다 VC를 유지한다.

MARS 구조에는, 각 송신원이 각 멀티캐스트 그룹에 대하여 독자적인 일대다 VC를 설정하여 데이터를 전송

하는 VC 매쉬 구조와 여러 송신원이 하나의 멀티캐스트 그룹에 대하여 하나의 일대다 VC를 공유하는 MCS(Multicast Server) 구조가 있다. MCS 구조에서는 ATM 계층의 서버인 MCS를 사용하여 한 그룹에 해당하는 모든 송신원들이 그 그룹을 담당하는 MCS로 일단 멀티캐스트 데이터를 전송하고 MCS가 이를 다시 MCS로부터의 일대다 VC를 통해 전송하도록 함으로써 여러 송신원이 동일한 그룹에 대한 일대다 VC를 공유하도록 한다. VC 매쉬 구조는 MCS 구조에 비해 처리량 및 점대점 대기 시간 측면에서 더 좋은 성능을 나타내지만 같은 멀티캐스트 그룹에 대하여 송신자마다 독자적인 VC를 설정해야 하므로 VC 소모가 크고 관리 오버헤드가 크다. 반면 MCS 구조는 MCS에서의 병목 현상이 발생할 수 있고 성능면에서 VC 매쉬보다 떨어지지만 한 그룹에 대하여 하나의 일대다 VC만을 사용하고 동적으로 멀티캐스트 그룹이 변하는 경우, MCS로부터 그룹의 수신원들로 연결된 하나의 일대다 VC만을 바꾸면 되기 때문에 VC 매쉬를 이용하는 방법보다 VC를 효율적으로 사용하고 관리할 수 있다.

2.2 UNI4.0의 LIJ 시그널링

ATM의 UNI3.0/3.1 시그널링과는 달리 UNI 4.0 시그널링에서는 수신원에 해당하는 ATM 단말이 스스로 기존의 일대다 VC에 참여하기 위하여 송신원으로 신호 메시지를 발생할 수 있는 LIJ 메카니즘이 제공된다. UNI 4.0 시그널링에서는 LIJ를 지원하기 위하여 표 1과 같은 두 개의 신호 메시지를 추가하였다.

표 1 UNI 4.0 시그널링에서 LIJ를 지원하기 위하여 추가된 메시지

LEAF SETUP REQUEST	자식이 LIJ 절차를 시작하기 위해 전송하는 메시지
LEAF SETUP FAILURE	이전에 자식이 보냈던 LEAF SETUP REQUEST 요청이 일대다 VC의 루트나 네트워크 사정에 의하여 실패되었음을 알리는 메시지

멀티캐스트 그룹에 속하고자 하는 수신원은 일대다 VC의 루트로 LEAF SETUP 요청을 보냄으로써 일대다 VC의 자식으로 참여하는데 여기에 포함된 정보는 leaf sequence number, 루트의 ATM 주소, LIJ call ID이다. Leaf sequence number는 특정 LEAF SETUP 요청에 대한 응답으로 일대다 VC의 루트에 의해 발생하는 SETUP, ADD PARTY, LEAF SETUP FAILURE 메시지를 대응시켜주는 역할을 한다. 따라서 LEAF SETUP 요

청에 의하여 발생하는 SETUP, ADD PARTY 메시지는 표 2와 같이 LJJ 요청을 지원하기 위한 정보를 추가로 유지한다.

표 2 UNI 4.0 시그널링에서 LJJ를 지원하기 위해 SETUP과 ADD PARTY 메시지에 각각 추가 되는 정보

SETUP	LJ call identifier
	LJ parameters
	Leaf sequence number
ADD PARTY	Leaf sequence number

네트워크에서는 루트의 ATM 주소와 LJ call ID를 사용하여 해당 호를 전역적으로 구분할 수 있다. 즉, ATM 주소에 해당하는 단말내에서 여러 멀티캐스트 그룹에 대하여 일대다 VC가 있을 경우 LJ call ID를 사용하여 해당 멀티캐스트 그룹을 담당하고 있는 일대다 VC를 찾아내는 것이다. 자식이 LJ call ID를 알아내는 방법은 IP 그룹 주소를 LJ call ID로 대응시켜주는 잘 알려진 알고리즘을 사용하거나 특정 디렉토리 서비스를 이용한 외부 메커니즘을 통해서 가능하다.

3. UNI4MARS: UNI 4.0 기반 IP 멀티캐스트 지원 모델

본 장에서는 2.2절에서 살펴본 UNI 4.0의 LJJ를 활용하여 ATM 망에서 IP 멀티캐스팅을 보다 효율적으로 지원하기 위해 기존 MARS 서비스를 변형한 UNI4MARS 모델을 제안한다. UNI 3.0/3.1에서는 일대다 ATM VC에 참여하기를 원하는 수신원이 ATM 계층에서 이를 송신원에게 알리기 위해 신호를 보낼 수 없기 때문에 MARS라는 오버레이 서비스를 사용하였다. 즉, ATM 계층과 IP 계층 사이에 MARS 계층을 두어 IP 멀티캐스트에 참여하려는 ATM 단말의 경우 MARS 계층 제어를 위한 ATM VC를 통하여 MARS 서버에게 이를 알리고 MARS가 다시 이를 MARS 계층 제어 VC를 통해 송신원에게 알리도록 함으로써 IP 멀티캐스트의 동적인 그룹 변화를 지원하였다. 그러나, UNI 4.0의 LJJ 시그널링을 사용하면, IP 멀티캐스트 그룹에 참여하려는 수신원이 기존의 일대다 ATM VC의 자식으로 참여하기 위하여 직접 송신원으로 신호 메시지를 발생할 수 있다. 따라서, UNI 3.0/3.1 기반에서는 IP 멀티캐스트의 지원을 위해 수신원들의 동적인 멀티캐스트 그룹 참여/탈퇴 변화에 대한 정보를 모

아서 송신원에게 제공해 주는 서비스 구조가 필요했지만, UNI 4.0 시그널링을 사용하는 ATM 망에서는 수신원이 직접 송신원으로 일대다 VC에 참여시켜 주기를 요구하는 신호 메시지를 발생할 수 있도록 송신원들의 ATM 주소를 모아 수신원들에게 제공해주는 메커니즘이 필요하다.

구체적으로 살펴보면, UNI 4.0의 LJJ 기능을 사용하여 IP 멀티캐스트 그룹으로 참여하기를 원하는 ATM 단말은 송신원의 ATM 주소와 32-bit LJ call ID를 알아내어 송신원에게 LJJ 요청을 보내야 한다. 따라서 UNI 4.0 기반의 ATM 망에서 IP 멀티캐스트를 지원하기 위해서는 이 두가지 정보를 클러스터 멤버들에게 알려주어야 하는데, 본 논문에서는 이들 정보를 제공해주는 서버로서 UNI4MARS를 제안하였다. 제안하는 UNI4MARS 메커니즘에서는 LJ call ID를 알아내는 방식으로 외부 디렉토리 서비스를 사용하지 않고 멀티캐스트 그룹 주소와 LJ call ID가 동일하게 32bits라는 점에 착안하여, 멀티캐스트 그룹 주소를 LJ call ID로 결정한다. 표 3과 같이 D class의 IP 멀티캐스트 주소는 28 bits를 사용하여 특정 멀티캐스트 그룹을 나타내므로 멀티캐스트 그룹 주소를 LJ call ID로 결정하였을 경우 4bits의 잉여 필드가 생긴다. 이후에 QoS를 지원하는 멀티캐스트 서비스로 확장하는 모델에서, 한 그룹에 대한 일대다 VC가 QoS별로 여러개 생길 경우에는 4bits의 잉여 필드를 사용하여 QoS에 따른 VC를 명시할 수도 있다. 즉 외부 디렉토리 서비스를 사용하지 않아도 그룹 주소를 통한 call ID 대응방식이 이후에 QoS를 지원하는 확장 모델에서 사용될 수 있음을 의미한다.

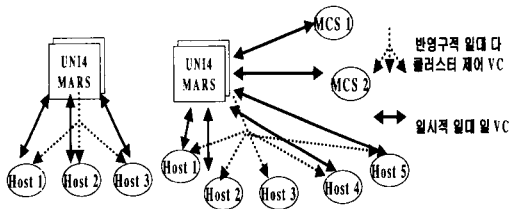
표 3 D class에 해당하는 IP 멀티캐스트 그룹 주소

1	1	1	0	Group Identification
---	---	---	---	----------------------

2.1절의 UNI 3.0/3.1 기반 MARS 모델에서 살펴본 바와 같이 VC 매쉬 구조와 MCS 구조는 각각 서로 대조가 되는 장단점이 있으므로 UNI4MARS 모델에서도 이 두 구조를 모두 수용하여 VC 매쉬 구조와 MCS 구조를 위한 방법을 각각 제안하였다. UNI4MARS 모델의 VC 매쉬 구조에서는 수신원들이 멀티캐스트 그룹에 대한 각 송신원으로서의 독자적인 일대다 VC에 참여하기 위해 송신원으로 LJJ 요청을 보내고, MCS 구조에서는 수신원들이 멀티캐스트 그룹에 대한 MCS로의 공동의 일대다 VC에 참여하기 위하여 MCS로 LJJ 요청을 보낸다. UNI4MARS는 수신원이 LJJ 요청을 할 수 있도록, 두가지 구조중 어

는 구조인가에 따라 멀티캐스트 그룹 멤버들에게 송신원의 ATM 주소를 알려주거나 MCS의 ATM 주소를 알려준다.

UNI4MARS 모델은 MARS와 마찬가지로 UNI4MARS 서버를 중심으로 형성되는 클러스터 단위로 ATM 단말들에 IP 멀티캐스트 서비스를 지원한다. 또한, VC 사용에 있어서도 MARS(2.2절 참고)와 유사하게 UNI4MARS와 클러스터 멤버 혹은 MCS간에 일시적인 일대일 VC와 반영구적인 일대다 VC(클러스터 제어 VC)를 설정하여 통신한다. 그림1의 (a)와 (b)에서 보듯이, UNI4MARS 서비스를 통해 IP 멀티캐스트 그룹에 참여하고자 하는 호스트들은 질의 혹은 등록/탈퇴를 위해 UNI4MARS와 일시적 일대일 VC를 설정한다. 또한, UNI4MARS는 자신이 담당하고 있는 클러스터의 모든 멤버들에게 새로운 송/수신원의 등록/탈퇴와 같은 그룹 멤버십에 대한 갱신 정보를 알리기 위하여 클러스터의 모든 멤버들과 반영구적인 일대다 VC인 클러스터 제어 VC를 유지한다.



(a) VC 매쉬 (b) MCS

그림 1 UNI4MARS에서 유지되는 제어 VC

송/수신원이 클러스터 멤버로 등록하여 UNI4MARS를 서비스 받기 위해서는 각각 다음의 과정이 필요하다. 먼저, 송/수신원은 MARS구조에서 MARS의 주소를 얻는 방법과 동일하게 ILMI(Interim Local Management Information) MIB(Management Information Base)로부터 해당 LIS의 UNI4MARS ATM주소를 획득한다[6] [7]. UNI4MARS의 ATM 주소를 얻은 송신원은 UNI4MARS에게 일시적 VC를 통하여 UNI4MARS_SRCJOIN 메시지를 보낸다. 이것을 받은 UNI4MARS는 이 송신원을 클러스터 제어 VC의 리프 노드로 추가하며 CMI(Cluster Member Identifier)를 부여한다. 만일 송신원이 탈퇴를 하려면 UNI4MARS_SRCLEAVE 메시지를 보낸다. 이 결과 UNI4MARS는 클러스터 제어VC에서 이 송신원을 제거하며, 이와 관련된 모든 매핑정보를 지우고 부여된 CMI도 회수한다. 수신원의 경우에는 멀티캐스트 그룹의 송신원

들 주소를 알기 위해 UNI4MARS_GRPQUERY를 요청하면 UNI4MARS는 UNI4MARS_GRPREPLY로 응답함과 동시에, 송신원의 경우와 마찬가지로 클러스터 제어VC에 수신원을 추가하며 CMI를 부여한다. 즉, 수신원은 그룹에 등록과 관계없이 질의만으로 클러스터 멤버가 된다. 그러나, UNI4MARS는 수신원에 관한 정보는 유지하지 않는다. 클러스터 멤버에서 탈퇴를 하기 위해서는 수신원도 탈퇴 메시지(CLUSTER_LEAVE)를 보내야 하고, CLUSTER_LEAVE를 받은 UNI4MARS는 해당 리프 노드를 일대다 클라이언트 제어VC에서 제거하고 CMI도 무효화시킨다.

이제 3.1절과 3.2절에서는 각각 VC 매쉬 구조와 MCS 구조에서의 UNI4MARS 메카니즘에 대하여 프리미티브와 자세한 프로세싱을 설명한다.

3.1 VC 매쉬 구조

VC 매쉬 구조에서는 멀티캐스트 그룹 멤버에 해당하는 ATM 단말들이 각 송신원으로 LIJ 요청을 보낸다. LIJ 요청시에 필요한 정보는 송신원의 ATM 주소와 LIJ call ID인데 송신원의 ATM 주소는 UNI4MARS 서버를 통하여 알게되고 LIJ call ID는 앞에서 설명된 바와 같이 IP 멀티캐스트 그룹 주소 값으로 결정된다. UNI4MARS 서버는 표 4와 같이 멀티캐스트 그룹에 대하여 트래픽을 보내는 송신원의 ATM 주소를 유지하여, 새롭게 멀티캐스트 그룹에 참여하는 ATM 단말의 요구에 응답한다.

표 4 VC 매쉬 구조에서 UNI4MARS가 유지하는 자료구조

그룹 주소	Src ATM 주소1, ..., Src ATM 주소 n
-------	--------------------------------

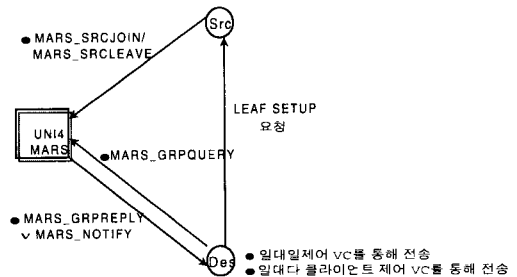


그림 2 VC 매쉬 구조에서 사용하는 UNI4MARS 프리미티브

그림 2는 VC 매쉬 구조에서 사용하는 프리미티브들을 보여주고 있다. 새롭게 멀티캐스트 그룹에 참여하고자 하는 수신원은 UNI4MARS 서버에게 IP 멀티캐스트 그룹

주소를 담은 MARS_GRPQUERY를 보낸다. 이를 받은 UNI4MARS 서버는 해당 멀티캐스트 그룹에 대한 송신원의 ATM 주소를 담은 MARS_GRPQUERY로서 수신원에 응답한다. 그리고 동시에 그 수신원을 일대다 클러스터 제어 VC(ClusterControlVC)에 추가하고 CMI를 부여한다. 송신원의 등록/탈퇴시와는 달리 수신원의 등록/탈퇴 정보에 대해서는 클러스터 제어 VC와 CMI 사용 정보만 변화될 뿐 클러스터 멤버에게 이러한 정보를 알릴 필요는 없다. MARS_GRPQUERY로부터 송신원의 ATM 주소를 얻은 수신원은 그룹 주소와 동일한 LJ call ID를 사용하여 송신원들에게 LEAF SETUP 요청을 보냄으로써 해당 멀티캐스트 그룹에 대응되는 일대다 ATM VC에 참여하게 된다. 멀티캐스트 그룹에서 탈퇴하기를 원하는 수신원은 UNI4MARS 서버에게 CLUSTER_LEAVE를 보냄으로써 탈퇴하려는 CMI를 알린다. 이 메시지를 받은 UNI4MARS는 해당 수신원을 클러스터 제어VC에서 제거하고 CMI를 회수한다. 종합적으로, UNI4MARS에서는 수신원에 대한 정보는 그룹 멤버십 정보로 저장되지도 않고 클러스터 멤버에게 알려지지도 않으며 수신원은 단지 클러스터 멤버로만 관리되어 UNI4MARS로부터 필요한 정보를 공급 받을 뿐이다.

새롭게 멀티캐스트 그룹에 참여하는 송신원은 UNI4MARS 서버에게 MARS_SRCJOIN을 보냄으로써 전송하고자 하는 IP 멀티캐스트 그룹 주소와 자신의 ATM 주소를 등록한다. 송신원으로부터 MARS_SRCJOIN을 받은 UNI4MARS는 클러스터 제어VC에 그 송신원 노드를 추가하고 새로운 CMI를 할당한다. 한편, MARS_SRCJOIN이나 MARS_SRCLEAVE를 받아 그룹 멤버십이 변경되었다면, UNI4MARS 서버는 그룹에 대한 송신원의 ATM 주소를 담은 MARS_NOTIFY를 클러스터 제어VC를 통해 클러스터 멤버 전체에게 보냄으로써 해당 멀티캐스트 그룹 멤버들이 새롭게 멀티캐스트 그룹에 참여하는 송신원의 ATM 주소를 알아내어 LEAF SETUP 요청을 보낼 수 있도록 한다. 멀티캐스트 그룹에서 탈퇴하기를 원하

는 송신원은 UNI4MARS 서버에게 MARS_SRCLEAVE를 보냄으로써 탈퇴하고자 하는 IP 멀티캐스트 그룹 주소와 자신의 ATM 주소를 알린다. MARS_SRCLEAVE를 받은 UNI4MARS 서버는 그룹에 대한 해당 송신원의 ATM 주소를 표 4와 같이 자신이 유지하는 자료구조에서 삭제한다. 표 5는 이상의 각 프리미티브가 포함하는 주요 정보를 보여준다.

3.2 MCS 구조

MCS 구조에서는 멀티캐스트 그룹 멤버에 해당하는 ATM 단말들이 해당 그룹을 담당하고 있는 MCS로 LJ 요청을 보낸다. LJ 요청시에 필요한 정보는 MCS의 ATM 주소와 LJ call ID인데 MCS의 ATM 주소는 UNI4MARS 서버를 통하여 알게되고 LJ call ID는 IP 멀티캐스트 그룹 주소 값이 된다. 또한 송신원은 UNI4MARS에 대한 질의를 통하여 자신이 전송하고자 하는 멀티캐스트 그룹에 대한 MCS의 ATM 주소를 얻어 해당 MCS로 VC를 설정한다. MARS와는 달리 UNI4MARS 서버는 호스트 맵을 둘 필요가 없고 서버 맵(표 6)만을 유지하며, 서버맵에는 멀티캐스트 그룹에 대한 MCS의 ATM 주소를 유지한다.

표 6 MCS 구조에서 UNI4MARS가 유지하는 서버 맵

그룹 주소	MCS ATM 주소1, MCS ATM 주소2, ..., MCS ATM 주소k.
-------	---

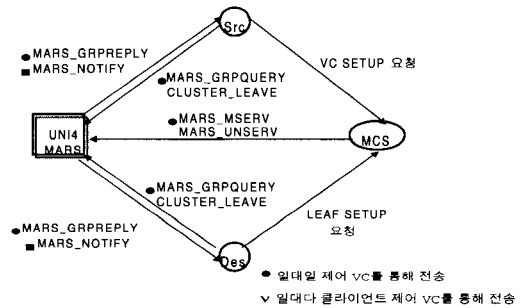


그림 3 MCS 구조에서 사용하는 UNI4MARS 프리미티브

표 5 VC 매쉬 구조에서 프리미티브들이 전송하는 주요 정보

MARS_SRCJOIN MARS_SRCLEAVE	그룹 주소, Src ATM 주소
MARS_GRPQUERY CLUSTER_LEAVE	그룹 주소
MARS_GRPQUERY MARS_NOTIFY	그룹 주소, Src ATM 주소 1, ..., Src ATM 주소 n

그림 3은 MCS 구조에서 사용하는 프리미티브들을 보여주고 있다. 새롭게 멀티캐스트 그룹에 참여하고자 하는 수신원이 UNI4MARS 서버에게 그룹 주소를 담은 MARS_GRPQUERY를 보내면 UNI4MARS 서버는 해당 멀티캐스트 그룹에 대한 MCS의 ATM 주소를 담은 MARS_GRPQUERY로서 수신원에 응답한다. 그리고 동시에 그 수신원을 일대다 클러스터 제어 VC(Cluster

ControlVC)에 추가하고 CMI를 부여한다. 즉, MARS와는 달리 수신원은 참여신청이 아닌 MARS_GRPQUERY에 의하여 클러스터 멤버가 되고 제어 VC에 추가 된다. 그러나, 수신원의 참여/탈퇴는 멀티캐스트 그룹 정보에는 변화를 주지 않으며, 특정 CMI의 사용 유/무 정보만이 갱신된다. 한편, 해당 수신원은 MARS_GRPQUERY로 얻은 MCS의 ATM 주소 및 그룹 주소와 동일한 LJJ call ID를 사용하여 MCS로 LEAF SETUP 요청을 보낸다. 멀티캐스트 그룹에서 탈퇴하기를 원하는 수신원은 UNI4MARS 서버에게 CLUSTER_LEAVE를 보냄으로써 탈퇴하려는 CMI를 알린다. 이 메시지를 받은 UNI4MARS는 해당 수신원을 클러스터 제어VC에서 제거하고 CMI를 회수한다.

새롭게 멀티캐스트 그룹에 참여하고자 하는 송신원은 UNI4MARS 서버에게 그룹 주소를 담은 MARS_GRPQUERY를 보내며, UNI4MARS 서버는 해당 멀티캐스트 그룹에 대한 MCS의 ATM 주소를 담은 MARS_GRPQUERY로서 송신원에 응답한다. 그리고 동시에 그 송신원을 일대다 클러스터 제어 VC(Cluster ControlVC)에 추가하고 CMI를 부여한다. 즉, MCS 모델에서는 송신원도 수신원의 경우와 마찬가지로 MARS_GRPQUERY에 의하여 클러스터 멤버가 되고 제어 VC에 추가 된다. 또한, 송신원의 참여/탈퇴도 UNI4MARS가 유지하는 멀티캐스트 그룹 정보에 변화를 UNI4MARS를 통하여 MCS의 ATM 주소를 알아낸 송신원들은 MCS에게로 일대일 양방향 VC 설정을 하여 데이터를 전달한다. 멀티캐스트 그룹에서 탈퇴하기를 원하는 송신원은 UNI4MARS 서버에게 CLUSTER_LEAVE를 보냄으로써 탈퇴하려는 CMI를 알린다. 이 메시지를 받은 UNI4MARS는 해당 수신원을 클러스터 제어VC에서 제거하고 CMI를 회수한다. 종합적으로, MCS 모델의 경우에는 송신원과 UNI4MARS 사이에 전개되는 작업이 수신원과 UNI4MARS 사이에 전개되는 작업과 동일하다. 단, 수신원은 UNI4MARS를 통한 주소해석으로 MCS를 발견하여 MCS로 LEAF SETUP 요청을 하는 반면, 송신원은 MCS로 VC 셋업 요청을 하게 된다.

새롭게 멀티캐스트 그룹에 참여하는 MCS는 UNI4MARS 서버에게 MARS_MSERV를 보냄으로써 전송하고자 하는 IP 멀티캐스트 그룹 주소와 자신의 ATM 주소를 등록한다. MARS_MSERV를 받은 UNI4MARS 서버는 새로운 MCS의 ATM 주소와 해당 멀티캐스트 그룹 주소를 담은 MARS_NOTIFY를 클러스터 제어 VC를 통해 클러스터 멤버들에게 보낸다. 새롭게 멀티캐스트 그룹에 참여하는 MCS의 ATM 주소를 받은 해당 멀티캐

스트 그룹의 수신원들은 MCS에게 LEAF SETUP 요청을 보내고, 송신원은 새로운 MCS에게로 VC SETUP 요청을 보낸다. 멀티캐스트 그룹에서 탈퇴하기를 원하는 MCS는 UNI4MARS서버에게 MARS_UNSERV를 보냄으로써 탈퇴하고자 하는 IP 멀티캐스트 그룹 주소와 자신의 ATM 주소를 알린다. MARS_UNSERV를 받은 UNI4MARS 서버는 그룹에 대한 해당 MCS의 ATM 주소를 표 7과 같이 자신이 유지하는 자료구조에서 삭제한다. 그리고 클러스터 제어VC를 통해 이를 클러스터 멤버들에게 알린다. 표 7은 이상의 각 프리미티브가 포함하는 주요 정보를 보여준다.

표 7 MCS 구조에서 프리미티브들이 전송하는 주요 정보

MARS_MSERV MARS_UNSERV	그룹 주소, MCS ATM 주소
MARS_GRPQUERY CLUSTER_LEAVE	그룹 주소
MARS_GRPQUERY MARS_NOTIFY	그룹 주소, MCS ATM 주소

4. 시뮬레이션 모델 및 결과 분석

본 장에서는 시뮬레이션을 통하여 VC mesh 구조에 대하여 기존의 MARS 모델과 제안하는 UNI4MARS 모델의 구현 비용을 프로세싱 오버헤드, 대역폭 요구량, 메모리 소요량 측면에서 비교해 보고자 한다. 먼저 성능을 평가하기 위한 시뮬레이션 모델을 설명하고 그 결과를 분석한다. 먼저, 호스트는 그 특성에 따라 다음과 같이 분류하였다. 송신원은 멀티캐스트 데이터의 근원지 호스트, 수신원은 멀티캐스트 데이터의 목적지 호스트, 송수신원은 송신원이면서 동시에 수신원인 호스트를 말한다. 또한, 받기만 하는 수신원은 R_수신원(Receive only_수신원)으로 명명하였다. 이밖에, 송/수신원은 송신원과 수신원인 호스트 모두를 가리키기 위해 사용하였다. 시뮬레이션 네트워크 모델은 MARS/UNI4MARS 서버와 그 서비스를 사용하는 클러스터 멤버로 이루어진다. 이때, 클러스터 멤버는 IP 멀티캐스트 그룹의 멤버이면서 메시지를 보내기도 하고 받기도 하는 호스트(송신원과 송수신원)와 IP 멀티캐스트 메시지를 받기만 하는 호스트(R_수신원)들로 이루어진다.

본 시뮬레이션은 Sun Solaris 워크스테이션에서 C 언어로 구현하였으며, 멀티캐스트 세션의 R_수신원의 수, 그룹 멤버의 동적인 정도 그리고 세션의 평균 지속 기간

등을 변화시켜보면서 시뮬레이션을 수행하였다. 4.1절에서는 성능 평가를 위한 시뮬레이션 모델을 설명하고, 4.2절에서는 실험 결과 및 연관성을 토대로 두 방식의 성능 평가 결과를 제시한다.

4.1 시뮬레이션 모델

본 시뮬레이션에서는 한 호스트는 한 시점에서 하나의 세션에만 가입되어 있으며, 동적인 정도에 있어서 송/수신원들간에 호스트 개별적인 차이는 없다고 가정한다. 즉, 메시지를 보내기도 하고 받기도 하는 호스트의 그룹 참가/탈퇴의 동적인 정도와 R_수신원의 그룹 참가/탈퇴 정도는 같다고 가정한다. 그리고 특별히 본 시뮬레이션에서는 R_수신원 수가 많은 응용을 대상으로 두 시스템의 성능을 비교하고자 각 세션당 송수신을 모두 담당하는 호스트 수를 평균 6으로 고정하고 R_수신원수를 평균 0에서 180까지 증가시켜 보면서 시뮬레이션을 실행하였다. 그리고, 하나의 LIS 안에 하나의 MARS(UNI4MARS) 서버가 서비스를 담당한다고 가정하였다.

한 멀티캐스트 세션이 생성되어 종료되기까지의 기간은 지수 분포를 따르며, 하나의 세션이 생성되면, 그 세션에 해당하는 IP 멀티캐스트 그룹의 멤버 수와 R_수신원의 수는 해당 평균아래에서 지수 분포로 발생시켰다. 또한 MARS의 경우 송신원의 질의/송신원의 탈퇴 메시지, 클러스터 멤버의 그룹 참가/탈퇴 메시지, 그리고 UNI4MARS의 경우 클러스터 멤버의 질의/클러스터 멤버의 탈퇴 메시지, 송신원의 그룹 참가/탈퇴 메시지는 해당 세션에 대해 결정된 멤버 수와 R_수신원 수에 따라 세션의 생성으로부터 종료 시점까지 균일 분포로 발생하게 된다. 메시지 크기는 MARS 권고 안을 참조하여 설정하였으며, 멀티캐스트 그룹 멤버의 동적인 정도에 따른 성능 변화를 살펴보기 위해서 한 멀티캐스트 세션에 참여하는 호스트들의 메시지 발생 빈도를 변화시켜 조절하였다.

이러한 시뮬레이션 모델과 가정 하에서 서버의 프로세싱 오버헤드, 서버가 필요로 하는 메모리량, 서버가 질의에 응답할 때 혹은 등록/탈퇴를 알리기 위해 사용하는 대역폭의 양을 비교해 보았다.

먼저, 프로세싱 오버헤드는 서버에게 보내진 각 패킷에 대하여 들어온 시간과 처리되어 서비스가 끝난 시간 사이의 차이를 구하고 그 합을 총 메시지수로 나누어 서버의 평균 메시지 처리시간으로 보았다. 처리하는 패킷의 양이 많고 복잡도가 높은 처리가 많을수록 처리 대기열에 많은 패킷이 쌓이게 되어 처리시간이 길어진다. 따라서, 평균처리시간으로 처리오버헤드를 간접적으로 측정할 수 있다. MARS의 경우 송신원 질의에 대한 MARS

프로세싱 시간(서버 테이블 검색 시간)을 t 로 볼 때, 송신원 탈퇴(서버 테이블 검색 불필요)는 $t/2$ 로, 클러스터 멤버의 그룹 참가/탈퇴(서버 테이블 검색 시간 + 삽입/삭제 시간)로 인한 프로세싱 시간을 $2.5t$ 로 가정하였다. UNI4MARS의 경우도 클러스터 멤버의 질의에 대한 UNI4MARS 프로세싱 시간(서버 테이블 검색 시간)을 t 로, 클러스터 멤버의 탈퇴 서비스 시간(서버 테이블 검색 불필요)을 $t/2$ 로, 송신원의 그룹 참가/탈퇴로 인한 프로세싱 시간(서버 테이블 검색 시간 + 삽입/삭제 시간)을 $2.5t$ 로 가정하였다.

서버가 유지하는 테이블은 IP 멀티캐스트 주소와 해당 그룹 멤버들의 ATM 어드레스로 구성된다. 따라서, 서버가 필요로 하는 메모리량을 구하기 위해서, 먼저 각 멀티캐스트 그룹 세션마다 멀티캐스트 그룹의 멤버 수를 계산한다. 그룹의 멤버 수는 MARS의 경우 클러스터 멤버가 그룹 참여/탈퇴할 때마다 그 값을 증감하고 UNI4MARS의 경우 송신원이 그룹 참여/탈퇴할 때마다 그 값을 증감함으로써 구한다. 이처럼, 한 세션동안 멀티캐스트 그룹의 멤버 수는 증감을 거듭하게 되는데, 임의의 시간 슬롯에서의 한 세션의 멀티캐스트 그룹의 평균 멤버 수 k 는 다음과 같은 식으로 표시된다.

$$k = \frac{\sum_{t=1}^s n(t)}{s}, \quad n(t) : \text{슬롯 } t \text{에서의 그룹 멤버 수,}$$

$$s : \text{세션 지속 기간}$$

결과적으로, 세션 i 가 임의의 시간 슬롯에 필요로 하는 평균 메모리량 m_i 는 다음과 같다.

$$m_i = 4\text{bytes}(\text{IP 멀티캐스트 그룹 주소}) + k * 20 \text{ bytes}$$

그러므로 요구되는 서버의 총 메모리 양은 다음과 같이 표시될 수 있다.

$$M = \sum_{i=1}^{\text{발생된총멀티캐스트그룹수}} m_i$$

MARS와 UNI4MARS 두 경우 모두 서버로부터 전송되는 메시지에는 두 가지 종류가 있다. 먼저, MARS_MULTI는 송신원의 질의시 서버가 현재 가지고 있는 클러스터 멤버의 ATM주소를 질의한 호스트에게 알리기 위해 사용한다. 이에 반해, UNI4MARS_MULTI는 클러스터 멤버의 질의시 서버가 현재 가지고 있는 송신원의 ATM주소를 질의한 호스트에게 알리기 위해 사용한다. 시뮬레이션에서는 MARS 권고 안에서 제시한 MARS_MULTI 메시지의 형식에 맞추어 MARS_MULTI와 UNI4MARS_MULTI 메시지 크기를 결정하였다. 하나

의 MARS_MULTI 혹은 UNI4MARS_MULTI 메시지는 MARS 권고 안에 따라 N 을 등록되어 주소 정보를 저장하고 있는 해당 멀티캐스트 멤버수라고 할 때 $(60\text{bytes} + 20 * N)$ 이며, 멤버 수에 따라 메시지의 크기가 결정된다. MARS_NOTIFY는 송신원의 그룹 참여/탈퇴시 변경되는 서버정보를 모든 클러스터 멤버에게 전달하기 위해 사용된다. 이 메시지는 MARS_JOIN과 MARS_LEAVE가 클러스터 멤버에게 전달되는 것이므로 MARS 권고 안에서 제시하는 MARS_JOIN/LEAVE 메시지의 크기를 따른다. 즉, MARS_NOTIFY의 크기는 $(32\text{bytes} + 20\text{bytes}(\text{소스/목적지의 ATM주소}) + 4\text{bytes}(\text{IP 멀티캐스트 주소}))$ 에 해당하는 56bytes가 된다. UNI4MARS_NOTIFY는 클러스터 멤버의 그룹 참여/탈퇴시 변경되는 서버 정보를 모든 클러스터 멤버에게 전달하기 위해 사용된다. 이는 UNI4MARS_JOIN과 UNI4MARS_LEAVE가 클러스터 멤버에게 전달되는 것이므로 MARS_NOTIFY와 마찬가지로 MARS_JOIN/LEAVE 메시지의 크기를 따른다.

4.2 시뮬레이션 수행과 결과 분석

본 절에서는 제안한 MARS와 UNI4MARS 서버 성능을 평가하기 위하여 앞에서 설명한 모델과 파라미터를 기반으로 하여 수행된 시뮬레이션의 결과를 분석한다. 시뮬레이션의 결과로써 측정되는 값은 다음과 같다. 우선 MARS와 UNI4MARS의 프로세싱 오버헤드를 비교하기 위해서는 각 모델에서 메시지를 처리하는 평균 시간을 측정하였다. 그리고 각 서버가 필요로 하는 메모리량은 시뮬레이션 기간에 걸쳐 소요된 총 메모리의 양으로써 비교하였다. 마지막으로 소요되는 대역폭을 비교하기 위해 MARS가 전송하는 MARS_MULTI 전송에 사용된 대역폭 총량과 MARS_NOTIFY 전송에 사용된 대역폭 총량의 합, 그리고 UNI4MARS가 전송하는 UNI4MARS_MULTI 전송에 사용된 대역폭 총량과 UNI4MARS_NOTIFY 전송에 사용된 대역폭 총량의 합을 구하여 각각 대응치를 비교하였다.

[그림 4,5,6]은 모두 클러스터 규모의 변화에 따른

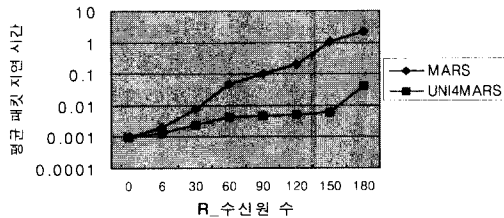


그림 4 R_수신원수 변화시 평균 패킷 지연 시간의 변화

MARS와 UNI4MARS 서버 성능을 보여주고 있다. [그림 4]는 $t=0.002$ 초로 가정하였을 때 MARS와 UNI4MARS 서버의 평균 메시지 처리 시간을 보여준다. 모든 그룹 멤버가 송신원이자 수신원인 경우인 R_수신원 수=0 에서는 두 모델이 동일한 평균 패킷 지연 시간을 보인다. 그러나 R_수신원 수가 증가함에 따라 MARS의 경우 평균 패킷 처리 시간이 급격한 증가를 보임에 반해, UNI4MARS는 비교적 완만한 증가를 보인다. MARS는 클러스터 멤버의 그룹 참여/탈퇴로 인한 프로세싱 시간이, 그리고 UNI4MARS의 경우는 송신원의 그룹 참여/탈퇴로 인한 프로세싱 시간이 가장 비중이 큰 처리 시간을 요구한다. 따라서 MARS는 R_수신원의 수가 증가하면 그룹 멤버 참여/탈퇴 메시지와 이를 처리하는 시간이 늘어남으로써 패킷 처리 시간이 급격히 증가한다. 반면에, UNI4MARS는 R_수신원의 수가 증가하면 각각 비교적 프로세싱 오버헤드가 적은 클러스터 멤버들의 질의에 대한 프로세싱 시간과 클러스터 멤버의 탈퇴 서비스 시간만이 증가하므로 완만한 증가를 보인다.

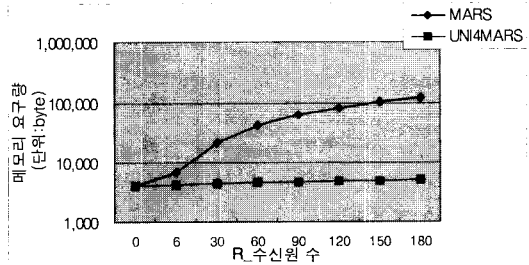


그림 5 R_수신원 수 변화시 요구되는 총 메모리량의 변화

[그림5]는 R_수신원 수를 변화시켰을 때 MARS와 UNI4MARS 서버에서 시뮬레이션 기간에 걸쳐 소요된 총 메모리량의 변화를 보여주고 있다. MARS 모델의 경우 각 멀티캐스트 그룹에 대하여 클러스터 멤버 전체의 정보를 유지해야 하는 반면 UNI4MARS 모델의 경우에는 각 멀티캐스트 그룹에 대하여 송신원 정보만을 유지한다. 따라서 R_수신원의 수가 증가할 수록 즉, 클러스터의 규모가 커질수록 MARS의 메모리 요구량은 급격히 증가하는 반면, UNI4MARS는 R_수신원에 대한 정보는 유지하지 않으므로 그 값이 변하지 않는다. 따라서, R_수신원의 수가 송신원 수에 비하여 훨씬 많은 응용의 경우, 서버에게 요구되어 지는 메모리량의 면에서, UNI4MARS가 유리하다.

[그림6]은 R_수신원 수를 변화시켰을 때 단위 시간당

안 MARS와 UNI4MARS 서버로부터 전송되는 메시지량을 비교한 것이다. 이 메시지량은 MARS의 경우 MARS_MULTI와 MARS_NOTIFY의 전송을 위한 대역폭의 합, UNI4MARS의 경우 UNI4MARS_MULTI와 UNI4MARS_NOTIFY의 전송을 위한 대역폭의 합으로 계산하였다. 송신원 멤버의 질의시 서버로부터 전송되는 MARS_MULTI메시지와 클러스터 멤버의 질의시 서버로부터 전송되는 UNI4MARS_MULTI 메시지를 살펴보면, MARS 모델의 경우에는 질의된 멀티캐스트 그룹에 속하는 모든 클러스터 멤버(수신원)들을 알려주어야 하는 반면, UNI4MARS 모델의 경우에는 질의된 멀티캐스트 그룹에 속하는 모든 송신원들을 알려주어야 한다. 따라서 R_수신원의 수가 증가하면 서버가 전송하는 메시지의 크기는 MARS가 UNI4MARS보다 더 크다. 그러나, UNI4MARS의 경우는 R_수신원 수가 증가함에 따라 질의하는 주체의 수가 증가하므로, 종합적으로는 UNI4MARS의 전송 메시지량이 조금 많은 것으로 나타났다. 반면에, 그룹 멤버에 변화가 생겼을 경우 클러스터 멤버들에게 통보하는 MARS_NOTIFY/UNI4MARS_NOTIFY 메시지 전송량은 MARS의 경우가 훨씬 많다. MARS는 그룹에 대한 클러스터 멤버의 그룹 참여/탈퇴에 따라 이 메시지를 전송하는 반면, UNI4MARS는 송신원의 그룹 참여/탈퇴에 따라 이 메시지를 전송하기 때문이다. 종합적으로는 NOTIFY 메시지로 인한 대역폭 소비가 MULTI 메시지로 인한 것보다 훨씬 크기 때문에 결과적으로 R_수신원이 증가할 수록 총 메시지량은 MARS의 경우가 급격한 증가를 보이면서 UNI4MARS와의 차이가 더욱 커지게 된다.

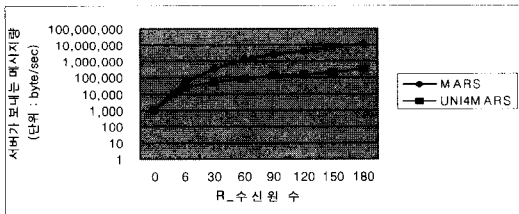


그림 6 R_수신원 수 변화시 전송되는 총 메시지량의 변화

[그림7]은 송/수신원 역할을 모두 담당하는 멤버의 수가 평균 6이고, R_수신원 수가 평균 180인 경우, 멀티캐스트 그룹 멤버들의 그룹에 대한 참여/탈퇴의 동적인 정도를 증가시키면서, 평균 패킷 처리 시간을 비교한 결과를 보인 것이다. 그룹의 동적인 정도는 세션의 평균 멤버

수에 대한 중간에 참여/탈퇴하는 멤버수의 비율을 의미한다. 따라서, 그룹의 동적인 정도가 커지게 되면 세션 진행중 멤버들의 그룹 참여/탈퇴횟수가 증가하게 되고, 이로인해 MARS나 UNI4MARS서버가 처리해야 할 제어 메시지 수가 증가하게 된다. R_수신원이 멤버의 주를 이루는 경우 MARS는 그룹 멤버의 그룹 참여/탈퇴 패킷 수가 증가함으로써 패킷 처리 시간이 급격히 증가하는 반면에, UNI4MARS는 비교적 프로세싱 오버헤드가 적은 그룹 멤버의 질의에 대한 프로세싱 시간과 그룹 멤버의 탈퇴 서비스 시간만이 증가하므로 완만한 증가를 보인다.

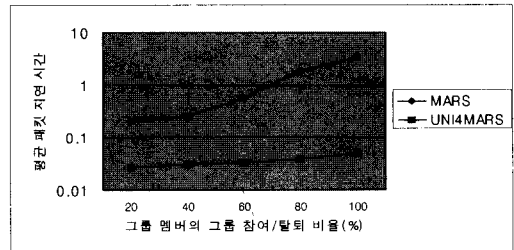


그림 7 멀티캐스트 그룹 멤버의 동적인 정도에 따른 평균 패킷 지연 시간 변화

[그림8]은 역시 송신원 수를 그룹 멤버의 동적인 정도에 따라 MARS와 UNI4MARS의 총 메모리 요구량 변화 정도를 보여주는 그래프이다. 멀티캐스트 그룹 멤버의 동적인 정도에 따른 메모리 요구량 변화는 두 구조가 큰 차이가 없는 것으로 나타났다. 이는 멤버의 참여/탈퇴가 빈번하여도 평균적으로 정보가 유지되는 멤버수는 두 구조 모두 거의 일정하기 때문이다.

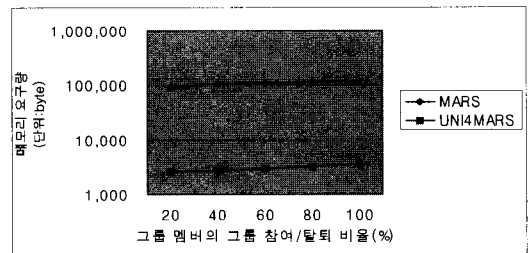


그림 8 멀티캐스트 그룹 멤버의 동적인 정도에 따른 총 메모리 요구량 변화

[그림9]는 멀티캐스트 그룹 멤버의 동적인 정도에 따라 MARS와 UNI4MARS 서버가 전송해야 하는 메시지량

의 변화를 보여주는 그래프이다. 그룹 멤버의 동적인 정도를 크게 해도 UNI4MARS는 완만한 증가를 보이는 반면, MARS는 급격한 증가를 보인다. 서버가 전송해야 하는 메시지량의 경우, MARS는 MULTI 메시지량에는 변화가 거의 없는 반면, 그룹 멤버의 멀티캐스트 그룹의 등록/탈퇴 패킷이 증가함에 따라 NOTIFY 메시지량이 크게 증가한다. 반면, UNI4MARS의 경우, NOTIFY 메시지량에는 거의 변화가 없지만 그룹 멤버의 멀티캐스트 그룹에 대한 질의/그룹 멤버의 탈퇴 패킷이 늘어남으로써 MULTI 메시지량이 증가하게 된다. 결국, NOTIFY 메시지로 인한 대역폭 소비가 MULTI 메시지로 인한 것보다 훨씬 크기 때문에 결과적으로 그룹 멤버의 동적인 정도가 커지면 총 메시지량은 MARS의 경우가 급격한 증가를 보이면서 UNI4MARS와의 차이가 더욱 커지게 된다.

MCS 구조의 경우에도 프로세싱 오버헤드 메모리 소모량, 메시지 전송량등의 측면에서 모두 본 절에서 보여준 VC Mesh 구조 경우와 매우 유사한 결과를 보이게 되며 그 원인도 유사하므로 이에 대한 결과는 생략하였다.

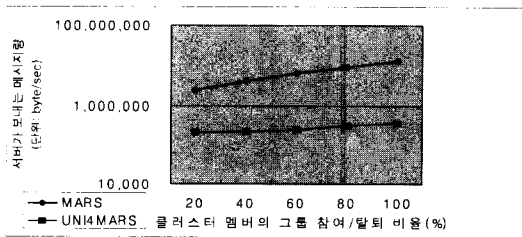


그림 9 멀티캐스트 그룹 멤버의 동적인 정도에 따른 전송 메시지량 변화

7. 결론

본 논문에서는 UNI 4.0 기반의 ATM 망에서 IP 멀티캐스트 서비스를 지원하는 UNI4MARS 모델을 제안하였다. 제안하는 모델은 UNI 4.0 시그널링의 LJJ 기능을 이용함으로써 ATM 망에서 수신원 제어적인 IP 멀티캐스트 서비스를 지원할 수 있다. 또한, 프로세싱 오버헤드, 메모리 및 대역폭 요구량 등의 측면에서 구현 비용을 비교해 볼 때 R_수신원의 수가 많은 응용일수록 그리고, 멀티캐스트 그룹 멤버들의 그룹에 대한 참여/탈퇴의 동적인 정도가 큰 응용일 수록 UNI4MARS가 MARS에 비하여 더 효율적임을 볼 수 있었다. 제안하는 모델의 경우, LJJ 기능의 활용으로 인해 IP 멀티캐스트에서의 동적인 수신원 가입/탈퇴를 지원하는 측면에서의 새로운 기

능적 가능성을 제공하는 것은 아니지만, LJJ 시그널링을 활용함으로써 R_수신원이 많은 멀티캐스트 응용에 대하여 ATM 망에서의 IP 멀티캐스트 지원이 MARS 모델보다 효율적으로 이루어질 수 있음을 보여주었다.

참고 문헌

- [1] G. Armitage, "Support for Multicast over UNI 3.0/3.1 based ATM Networks," RFC 2022, Nov. 1996.
- [2] ATM Forum, "ATM User Network Interface (UNI) Signalling Specification Version 4.0," af-sig-0061.000, Jul. 1997.
- [3] G. Armitage, "IP Multicasting over ATM Networks," Journal on Selected Areas in Communications VOL. 15 No. 3, Apr. 1997.
- [4] 박 미룡, "UNI 4.0 기반의 ATM 망에서 인터넷 멀티캐스트", 한국통신학회 발표집, 1999년 7월.
- [5] 최 정현, 이미정, "UNI 4.0 기반 ATM 망에서의 IP 멀티캐스트 지원을 위한 서비스 구조", 통신 정보 합동 학술 대회, 논문집 2권, 1999년 4월.
- [6] ATM Forum "Integrated Local Management Interface (ILMI) Specification Version 4.0," af-ilmi-0065.000, Sep. 1996.
- [7] Mike Davison, "ILMI-Based Server Discovery for MARS," Internet-Draft, Nov. 1998.
- [8] R. Talpade, "Multicast Server Architecture for MARS based ATM multicasting," RFC 2149, May. 1997.
- [9] J. Heinanen, "Multiprotocol Encapsulation over ATM Adaptation Layer 5," RFC 1483, Telecom Finland, Jul. 1993.
- [10] M. Laubach, "Classical IP and ARP over ATM," RFC 1577, Jan. 1994.
- [11] S. Deering, "Host Extensions for IP Multicasting," RFC 1112, Aug. 1989.
- [12] S. Deering et al, "Distance Vector Multicast Routing Protocol," RFC 1075, Nov. 1988.
- [13] Torsten Braun et al, "Comparison of Concepts for IP Multicast over ATM," IEEE ATM '96 Vol. 1, Aug. 1996.
- [14] Rajesh R. Talpade et al, "Experience with Architectures for Supporting IP Multicast over ATM," IEEE ATM '96 Vol. 1, Aug. 1996.

안 MARS와 UNI4MARS 서버로부터 전송되는 메시지량을 비교한 것이다. 이 메시지량은 MARS의 경우 MARS_MULTI와 MARS_NOTIFY의 전송을 위한 대역폭의 합, UNI4MARS의 경우 UNI4MARS_MULTI와 UNI4MARS_NOTIFY의 전송을 위한 대역폭의 합으로 계산하였다. 송신원 멤버의 질의시 서버로부터 전송되는 MARS_MULTI메시지와 클러스터 멤버의 질의시 서버로부터 전송되는 UNI4MARS_MULTI 메시지를 살펴보면, MARS 모델의 경우에는 질의된 멀티캐스트 그룹에 속하는 모든 클러스터 멤버(수신원)들을 알려주어야 하는 반면, UNI4MARS 모델의 경우에는 질의된 멀티캐스트 그룹에 속하는 모든 송신원들을 알려주어야 한다. 따라서 R_수신원의 수가 증가하면 서버가 전송하는 메시지의 크기는 MARS가 UNI4MARS보다 더 크다. 그러나, UNI4MARS의 경우는 R_수신원 수가 증가함에 따라 질의하는 주체의 수가 증가하므로, 종합적으로는 UNI4MARS의 전송 메시지량이 조금 많은 것으로 나타났다. 반면에, 그룹 멤버에 변화가 생겼을 경우 클러스터 멤버들에게 통보하는 MARS_NOTIFY/UNI4MARS_NOTIFY 메시지 전송량은 MARS의 경우가 훨씬 많다. MARS는 그룹에 대한 클러스터 멤버의 그룹 참여/탈퇴에 따라 이 메시지를 전송하는 반면, UNI4MARS는 송신원의 그룹 참여/탈퇴에 따라 이 메시지를 전송하기 때문이다. 종합적으로는 NOTIFY 메시지로 인한 대역폭 소비가 MULTI 메시지로 인한 것보다 훨씬 크기 때문에 결과적으로 R_수신원이 증가할 수록 총 메시지량은 MARS의 경우가 급격한 증가를 보이면서 UNI4MARS와의 차이가 더욱 커지게 된다.

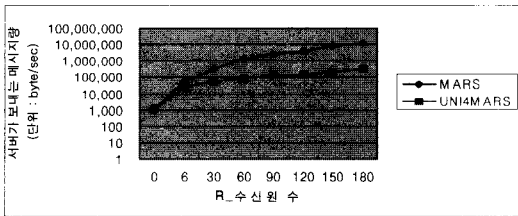


그림 6 R_수신원 수 변화시 전송되는 총 메시지량의 변화

[그림7]은 송/수신원 역할을 모두 담당하는 멤버의 수가 평균 6이고, R_수신원 수가 평균 180인 경우, 멀티캐스트 그룹 멤버들의 그룹에 대한 참여/탈퇴의 동적인 정도를 증가시키면서, 평균 패킷 처리 시간을 비교한 결과를 보인 것이다. 그룹의 동적인 정도는 세션의 평균 멤버

수에 대한 중간에 참여/탈퇴하는 멤버수의 비율을 의미한다. 따라서, 그룹의 동적인 정도가 커지게 되면 세션 진행중 멤버들의 그룹 참여/탈퇴횟수가 증가하게 되고, 이로인해 MARS나 UNI4MARS서버가 처리해야 할 제어 메시지 수가 증가하게 된다. R_수신원이 멤버의 주를 이루는 경우 MARS는 그룹 멤버의 그룹 참여/탈퇴 패킷 수가 증가함으로써 패킷 처리 시간이 급격히 증가하는 반면에, UNI4MARS는 비교적 프로세싱 오버헤드가 적은 그룹 멤버의 질의에 대한 프로세싱 시간과 그룹 멤버의 탈퇴 서비스 시간만이 증가하므로 완만한 증가를 보인다.

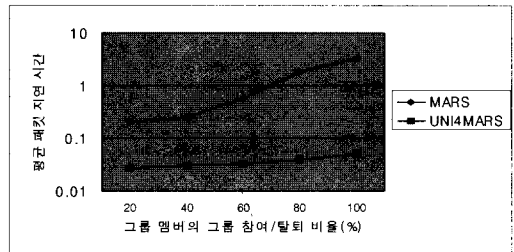


그림 7 멀티캐스트 그룹 멤버의 동적인 정도에 따른 평균 패킷 지연 시간 변화

[그림8]은 역시 송신원 수를 그룹 멤버의 동적인 정도에 따라 MARS와 UNI4MARS의 총 메모리 요구량 변화 정도를 보여주는 그래프이다. 멀티캐스트 그룹 멤버의 동적인 정도에 따른 메모리 요구량 변화는 두 구조가 큰 차이가 없는 것으로 나타났다. 이는 멤버의 참여/탈퇴가 빈번하여도 평균적으로 정보가 유지되는 멤버수는 두 구조 모두 거의 일정하기 때문이다.

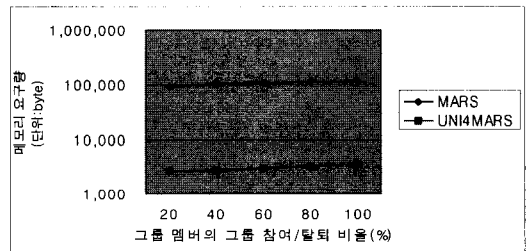


그림 8 멀티캐스트 그룹 멤버의 동적인 정도에 따른 총 메모리 요구량 변화

[그림9]는 멀티캐스트 그룹 멤버의 동적인 정도에 따라 MARS와 UNI4MARS 서버가 전송해야 하는 메시지량

의 변화를 보여주는 그래프이다. 그룹 멤버의 동적인 정도를 크게 해도 UNI4MARS는 완만한 증가를 보이는 반면, MARS는 급격한 증가를 보인다. 서버가 전송해야 하는 메시지량의 경우, MARS는 MULTI 메시지량에는 변화가 거의 없는 반면, 그룹 멤버의 멀티캐스트 그룹의 등록/탈퇴 패킷이 증가함에 따라 NOTIFY 메시지량이 크게 증가한다. 반면, UNI4MARS의 경우, NOTIFY 메시지량에는 거의 변화가 없지만 그룹 멤버의 멀티캐스트 그룹에 대한 질의/그룹 멤버의 탈퇴 패킷이 늘어남으로써 MULTI 메시지량이 증가하게 된다. 결국, NOTIFY 메시지로 인한 대역폭 소비가 MULTI 메시지로 인한 것보다 훨씬 크기 때문에 결과적으로 그룹 멤버의 동적인 정도가 커지면 총 메시지량은 MARS의 경우가 급격한 증가를 보이면서 UNI4MARS와의 차이가 더욱 커지게 된다.

MCS 구조의 경우에도 프로세싱 오버헤드 메모리 소요량, 메시지 전송량등의 측면에서 모두 본 절에서 보여준 VC Mesh 구조 경우와 매우 유사한 결과를 보이게 되며 그 원인도 유사하므로 이에 대한 결과는 생략하였다.

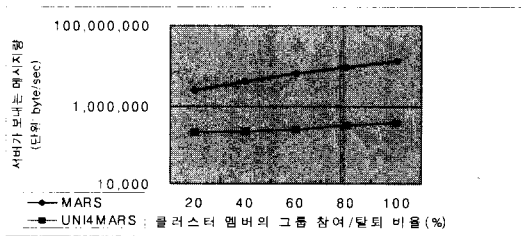


그림 9 멀티캐스트 그룹 멤버의 동적인 정도에 따른 전송 메시지량 변화

7. 결론

본 논문에서는 UNI 4.0 기반의 ATM 망에서 IP 멀티캐스트 서비스를 지원하는 UNI4MARS 모델을 제안하였다. 제안하는 모델은 UNI 4.0 시그널링의 LIJ 기능을 이용함으로써 ATM 망에서 수신원 제어적인 IP 멀티캐스트 서비스를 지원할 수 있다. 또한, 프로세싱 오버헤드, 메모리 및 대역폭 요구량 등의 측면에서 구현 비용을 비교해 볼 때 R 수신원의 수가 많은 응용일수록 그리고, 멀티캐스트 그룹 멤버들의 그룹에 대한 참여/탈퇴의 동적인 정도가 큰 응용일 수록 UNI4MARS가 MARS에 비하여 더 효율적임을 볼 수 있었다. 제안하는 모델의 경우, LIJ 기능의 활용으로 인해 IP 멀티캐스트에서의 동적인 수신원 가입/탈퇴를 지원하는 측면에서의 새로운 기

능적 가능성을 제공하는 것은 아니지만, LIJ 시그널링을 활용함으로써 R 수신원이 많은 멀티캐스트 응용에 대하여 ATM 망에서의 IP 멀티캐스트 지원이 MARS 모델보다 효율적으로 이루어질 수 있음을 보여주었다.

참고 문헌

- [1] G. Armitage, "Support for Multicast over UNI 3.0/3.1 based ATM Networks," RFC 2022, Nov. 1996.
- [2] ATM Forum, "ATM User Network Interface (UNI) Signalling Specification Version 4.0," af-sig-0061.000, Jul. 1997.
- [3] G. Armitage, "IP Multicasting over ATM Networks," Journal on Selected Areas in Communications VOL. 15 No. 3, Apr. 1997.
- [4] 박 미룡, "UNI 4.0 기반의 ATM 망에서 인터넷 멀티캐스트", 한국통신학회 발표집, 1999년 7월.
- [5] 최 정현, 이미정, "UNI 4.0 기반 ATM 망에서의 IP 멀티캐스트 지원을 위한 서비스 구조", 통신 정보 합동 학술 대회, 논문집 2권, 1999년 4월.
- [6] ATM Forum "Integrated Local Management Interface (ILMI) Specification Version 4.0," af-ilm-0065.000, Sep. 1996.
- [7] Mike Davison, "ILMI-Based Server Discovery for MARS," Internet-Draft, Nov. 1998.
- [8] R. Talpade, "Multicast Server Architecture for MARS based ATM multicasting," RFC 2149, May. 1997.
- [9] J. Heinanen, "Multiprotocol Encapsulation over ATM Adaptation Layer 5," RFC 1483, Telecom Finland, Jul. 1993.
- [10] M. Laubach, "Classical IP and ARP over ATM," RFC 1577, Jan. 1994.
- [11] S. Deering, "Host Extensions for IP Multicasting," RFC 1112, Aug. 1989.
- [12] S. Deering et al, "Distance Vector Multicast Routing Protocol," RFC 1075, Nov. 1988.
- [13] Torsten Braun et al, "Comparison of Concepts for IP Multicast over ATM," IEEE ATM '96 Vol. 1, Aug. 1996.
- [14] Rajesh R. Talpade et al, "Experience with Architectures for Supporting IP Multicast over ATM," IEEE ATM '96 Vol. 1, Aug. 1996.



이 미 정

1983년 ~ 1987년 이화여자대학교 전자계산학 학사. 1987년 ~ 1989년 University of North Carolina at Chapel Hill 컴퓨터학 석사. 1990년 ~ 1994년 North Carolina State University 컴퓨터공학 박사. 1994년 ~ 현재 이화여자대학교 공과대학 컴퓨터학과 부교수. 관심분야는 고속 통신 프로토콜 설계 및 성능 분석, ATM 연동 기술, 비디오 멀티캐스트, QoS 라우팅트래픽 엔지니어링



정 선

1993년 ~ 1997년 이화여자대학교 전산학과 학사. 1998년 ~ 현재 이화여자대학교 컴퓨터학과 석사과정 재학 중. 관심분야는 멀티캐스팅, QoS 라우팅



김 예 경

1992년 ~ 1996년 이화여자대학교 사회생활과 학사. 1995년 ~ 1997년 (주)두산정보통신 근무. 1998년 ~ 현재 이화여자대학교 컴퓨터학과 석사과정 재학 중. 관심분야는 멀티캐스트 라우팅, 이동 데이터 통신