

# End-to-End QoS를 지원하기 위한 이동 멀티캐스트 기법

정회원 김 태 수\*, 이 광 휘\*

## A Mobile Multicast Mechanism for End-to-End QoS Delivery

Tae-Soo Kim\*, Kwang-Hui Lee\* *Regular Members*

### 요 약

본 논문은 이동 통신망 환경에서 사용자들의 다양한 요구사항을 충족시킬 수 있도록 End-to-End QoS를 제공하기 위한 이동 멀티캐스트 기법을 제안하였다. 투명한 이동성 보장을 위해 빠른 핸드오프 기법을 적용하였다. L2 Mobile Trigger를 이용하여 핸드오프 시 발생하는 지연으로 인한 다량의 패킷 손실을 줄일 수 있도록 하였다. Xcast++에 계층적 개념을 도입하여 확장한 HXcast++를 제안하여 멀티캐스트 전송 경로의 최적화를 제공하고, 빈번한 핸드오프로 인한 멀티캐스트 서비스 유지비용을 줄일 수 있도록 하였다. 핸드오프 시 IGMP Membership Query를 기다림 없이 즉각 그룹에 가입할 수 있는 GMA(Group Management Agent) 기반 그룹 관리 메커니즘을 제안하여 그룹 가입 지연을 줄이고, 탈퇴 지연으로 인한 자원의 낭비를 제거하였다. 그리고 멀티캐스트 트리가 생성되는 동안 발생할 수 있는 패킷 손실은 buffering&forwarding 기법을 이용하여 줄였다. End-to-End QoS 제공을 위해 지역 네트워크에서는 IntServ/RSVP를 사용하고 광역 네트워크에서는 DiffServ를 사용하도록 구성하였다. 그리고 핸드오프로 인한 RSVP의 세션 재 설립 지연을 줄이기 위해 HXcast++ 제어 메시지를 확장하여 PATH 메시지를 요구하도록 하였다. 본 논문에서 제안하는 HXcast++는 이동 통신망 환경에서 End-to-End QoS를 제공하는 멀티캐스트 기법으로서 사용자들의 다양한 요구사항을 충족시킬 수 있는 한 가지 방안이 될 것이다.

Key Words : Mobile IP, Mobility, Multicast, QoS, Xcast

### ABSTRACT

This paper proposed a mobile multicast technique to satisfy end-to-end QoS for various user requirements in mobile network environment. In order to provide seamless mobility, fast handoff technique was applied. By using L2 mobile trigger, it was possible to minimize remarkable amount of packet loss by delay occurred during handoff. To provide efficient multicast, concept of hierarchy was introduced to Xcast++, which results in a creation of HXcast++. HXcast++ optimized transfer path of multicast and reduced expensive multicast maintenance costs caused by frequent handoff. Suggestion of GMA (Group Management Agent) mechanism allows joining to group immediately without waiting IGMP Membership query during handoff. GMA mechanism will minimize the delay for group registration process and the resource usage due to delay of withdrawal process. And also use of buffering & forwarding technique minimized packet loss during generation of multicast tree. IntServ/RSVP was used to provide End-to-End QoS in local domain and DiffServ was used in global domain. To minimize reestablishment of RSVP session delay, extended HXcast++ control messages were

\* 창원대학교 컴퓨터공학과 통신시스템 연구실({tskim, khlee}@changwon.ac.kr)

논문번호 : KICS2004-10-228, 접수일자 : 2004년 10월 12일

※ 본 논문은 정보통신부의 정보통신 기초기술연구지원사업(2003-1-01076)으로 수행된 결과의 일부임.

designed to require PATH message. HXcast++ proposed in this thesis is defined as multicast technique to provide end-to-end QoS and also to satisfy various user requirements in mobile network environment.

## I. 서론

최근 통신망을 이용한 원격 교육, 인터넷 방송, VoD, VoIP, 온라인 게임 등 다양한 응용들이 늘어나고 있다. 이들 서비스들은 대부분 멀티캐스트 기술을 필요로 하고 있으며 응용에 따라 안정적인 서비스를 위해 QoS(Quality of Service)의 보장도 필요로 한다[1,2]. 뿐만 아니라, 이동 통신의 발달로 이동 환경에서 사용자들의 요구사항은 다양화되고 있으며, 고정 망에서와 같은 수준의 서비스를 요구하고 있다. 특히 투명한 이동성 보장, 원활한 멀티캐스트 서비스 그리고 QoS의 제공 등이다. 따라서 이동 통신망 환경에서 QoS를 지원하는 멀티캐스트 서비스의 제공은 이동 통신 사용자들의 다양한 요구사항을 충족시키기 위한 기본 서비스가 되고 있다.

그러나 이동 통신망 환경에서 QoS를 지원하는 멀티캐스트 서비스를 제공하기 위해서는 해결해야 할 많은 문제점들이 있다. 특히 빈번한 핸드오프로 인한 멀티캐스트 트리의 변화는 멀티캐스트 자체의 효율성을 저하시키고 3가지 서비스가 결합 되었을 경우 각각에서 발생된 지연들의 합에 해당하는 지연(핸드오프 지연, 그룹 가입 지연, QoS 협상 지연)이 발생하게 된다. 이러한 지연은 패킷 손실을 유발시키고, 서비스의 단절 현상 및 QoS 제공을 어렵게 하는 원인이 된다.

따라서 본 논문에서는 이동 통신망 환경에서 이동 호스트에게 종단 간 QoS를 지원하는 멀티캐스트 서비스를 제공하기 위해 다음과 같은 해결 방안을 제시한다.

첫째, 멀티캐스트의 최적화된 전송 경로를 보장하고 핸드오프로 인한 멀티캐스트 서비스 유지비용을 최소화 시킬 수 있도록 최근 소규모 그룹 통신을 위해 제안되고 있는 Explicit Multicast 기법을 이용한다. 그 중 Xcast++[3]는 멀티캐스트 트리를 구성하지 않으므로 관리하는 오버헤드가 없으며, 멀티캐스트 데이터그램을 중복 없이 최적화된 경로로 전송함을 보장한다. 그러나 Xcast++는 중·소규모의 그룹을 지원하도록 설계하였으며, 그룹 멤버가 널리 퍼져있는 경우 패킷 헤더를 처리하는 비용이 증가하는 단점을 가지고 있다. 따라서 기존 Xcast++에 계층적 개념을 도입하여 중·대 규모 그룹을 지원하도록 확장하는 방안을 제안한다.

둘째, 핸드오프로 인한 지연 및 패킷 손실을 최소화 할 수 있도록 핸드오프 시 IGMP membership query 메시지를 기다림 없이 즉각 그룹에 가입하고 탈퇴할 수 있는 메커니즘이 필요하다[4]. 따라서 본 논문에서는 이러한 문제점을 해결하기 위해 GMA(Group Management Agent) 기반 그룹 관리 메커니즘을 제안한다. 이와 더불어 핸드오프 시 발생할 수 있는 패킷 손실을 최소화하기 위해 L2 트리거 기법[5,6]을 이용하여 빠르게 이동성을 감지하고, 이를 이용하여 이동 호스트는 필요한 경우 이전 에이전트에게 버퍼링을 요구하고 핸드오프 이후 포워딩 받는 방법을 적용한다.

셋째, 확장성과 이동 환경에 적합한 종단 간 QoS의 제공을 위해 지역 네트워크에서는 IntServ 방식을, 광역 네트워크에서는 DiffServ 방식을 적용하여 QoS 보장과 확장성 문제를 절충하도록 한다. 그리고 핸드오프 시 발생할 수 있는 QoS 세션 재 설정 지연을 줄이기 위해 가입 시 PATH 메시지를 요구할 수 있도록 확장하는 방안을 제안한다.

논문의 구성은 다음과 같다. 서론에 이어 2장에서는 관련 연구를 살펴보고, 3장에서는 기존 Xcast++에 계층적 개념을 도입한 HXcast++의 구성 요소를 설명한다. 그리고 4장에서는 HXcast++의 동작 과정을 설명한다. 5장에서는 제안 시스템의 성능을 분석하고, 마지막으로 6장에서는 결론을 언급한다.

## II. 관련 연구

표 1은 이동 멀티캐스트를 제공하는 기존 시스템들의 특성을 상호 비교 분석한 표이다. IETF 접근 방식인 BT(Bidirectional Tunneling)와 RS(Remote Subscription)[7], Xcast를 Mobile IP에 적용하도록 제안한 XMIP(Explicit Multicast over Mobile IP)[8], 그리고 Xcast+[9], Xcast++를 상호 비교하였다.

고정 망에서와는 달리 이동 망에서는 호스트의 이동으로 인해 멀티캐스트 그룹 멤버십이 변경되고 이는 멀티캐스트 트리의 재구성(Tree Reconstruction)을 유발한다. 그러므로 빈번한 핸드오프는 멀티캐스트 트리 재구성 비용을 증가시키는 직접적인 원인이 된다. RS의 경우, 이동한 네트워크에서 다시 가입해야 하므로 멀티캐스트 트리 재구성 비용이 크다. 그러나 BT의 경우 HA(Home Agent)를 경유

표 1. 기존 방법의 비용 분석[9,10]

	BT	RS	XMP	Xcast+	Xcast++	HXcast++
Tree Reconstruction	L	H	L	L	L	L
Optimized Routing	L	H	L	H	H	H
Control Overhead	H	H	NA	M	M	M.L
Extr Header Processing Overhead	L	L	H	H/M	H/M	M
Multicast Application	n:n	n:n	l:m	l:m	m:m	n:n

H : High, M : Medium, L : Low or none, NA : Not applicable, n : many, m : small or few

하여 서비스를 받으므로 멀티캐스트 트리는 재구성되지 않아도 되므로 그 비용이 낮다. Xcast 류의 경우는 멀티캐스트 트리를 구성하지 않으므로 비용이 없다.

멀티캐스트 전송 경로의 최적화 측면(Optimized Routing)에서는 BT와 XMIP의 경우 HA를 경유하여 전달하므로 비 최적화되고, 나머지는 FA(Foreign Agent)를 통해 가입하고 멀티캐스트 데이터를 전달받으므로 전송 경로가 최적화된다.

멀티캐스트 전송 트리를 유지하기 위한 제어 부하(Control Overhead)의 경우, 전통적인 멀티캐스트 기법을 사용하는 BT와 RS는 주기적인 플러딩(flooding) 기법을 이용하므로 부하가 크다. 그러나 Xcast+와 Xcast++의 경우 DR(Designated Router) 간에 주기적으로 멤버십 정보를 교환하는 방법을 사용하므로 부하가 적다.

그룹 멤버의 수가 증가함에 따른 추가적인 패킷 헤더 처리 비용(Extra Header Processing Overhead)은 Xcast+와 Xcast++의 경우, 동일한 서버넷에 그룹 멤버가 밀집될 경우 효과적으로 줄일 수 있다. 그러나 전통적인 멀티캐스트 기법을 사용하는 BT나 RS에 비해 패킷 헤더 처리 비용은 여전히 높으며, 이로 인해 그룹 멤버의 수에 제한을 받을 수밖에 없다.

그림 1은 Mobile IP 환경에서 일반적인 멀티캐스트 서비스와 IntServ 방식의 QoS를 제공 받는 이동 호스트가 기존 멤버가 없는 네트워크로의 핸드오프 절차를 나타낸 그림이다.

전체적인 핸드오프 지연은 이동한 네트워크로부터 새로운 CoA를 받아 서비스를 개시할 수 있을 때까지 걸리는 지연( $T_{move}$ )과 멀티캐스트 데이터를 받기까지 소요되는 지연( $T_{join}$ ) 그리고 QoS 재협상에 필요한 지연( $T_{qos}$ )으로 구성된다.

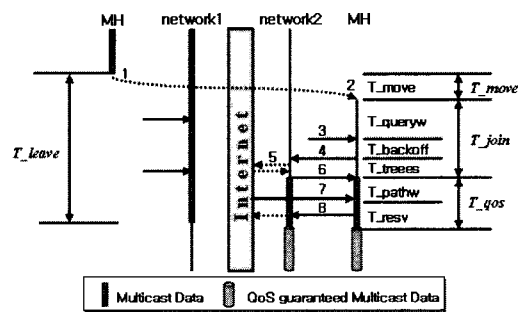


그림 1. 핸드오프 지연[4]

그림의 각 절차에 대한 설명은 아래와 같다.

1. MH가 network1의 영역을 벗어남.
2. MH가 network2의 영역에서 새로운 CoA를 획득하고 홈 등록을 수행.
3. MH가 Querier Router로부터 첫 번째 IGMP Query 메시지를 수신.
4. MH가 Backoff timer 종료 후 IGMP report 메시지를 전송.
5. 멀티캐스트 라우터가 멀티캐스트 트리 설정 절차를 수행.
6. MH가 network2로부터 첫 번째 멀티캐스트 데이터를 수신.
7. MH가 송신 호스트로부터 PATH 메시지를 수신.
8. MH가 RESV 메시지를 전송.

$T_{move}$ 의 원인은 이동 호스트의 느린 이동 감지와 홈 등록으로 인한 지연이다. Mobile IP에서 이동 호스트의 이동 감지는 에이전트 광고에 따라 이루어지며 기본 주기는 약 460~600초에 달한다. 뿐만 아니라 이동 호스트가 홈 네트워크로부터 먼 거리에 있을 때 등록으로 인한 지연이 추가적으로 적용된다.

*T<sub>join</sub>*의 원인은 이동 호스트가 network2로 이동했을 때, 주기적으로 방송되는 *IGMP query*를 기다려야 하기 때문이다. 뿐만 아니라 이전 네트워크의 영역을 벗어났으나 *leave* 메시지를 보낼 수 없기 때문에 *T<sub>leave</sub>*의 지연이 발생하게 되고 이에 따른 자원의 낭비를 초래한다.

*T<sub>qos</sub>*의 원인 또한 network2로 이동했을 때 *PATH*를 요구할 수 없으므로 주기적인 *PATH* 갱신 메시지를 기다려야 하기 때문이다. 따라서 각각의 지연들을 효과적으로 처리할 수 있는 메커니즘이 필요하다.

### III. Hierarchical Xcast++의 구성

본 논문에서는 이동 통신망 환경에서 효율적인 멀티캐스트 서비스를 제공하기 위해 기존 Xcast++에 계층적 개념을 도입하여 확장하는 방안(이하 HXcast++)을 제안한다. 그리고 종단 간 QoS 전송을 위해 최근의 연구경향인 IntServ/DiffServ 혼합 방식[11]을 적용한다.

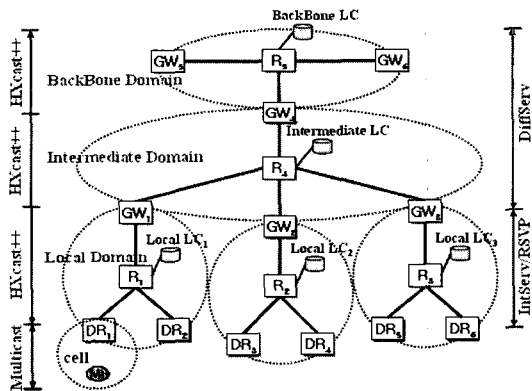


그림 2. 제안 시스템 구성도

멀티캐스트 트리를 구성하지 않으면서 최적화된 경로로 전송하는 Xcast++는 전통적인 멀티캐스트 기법을 사용하는 BT나 RS보다 이동 환경에 적합하여 이를 채택하였다. 그러나 추가적인 패킷 처리 부하로 인해 대규모 멀티캐스트 응용에 사용하기에 부적합하므로 이러한 단점을 극복하기 위해 Xcast++에 계층적 개념을 도입하였다(그림 2).

네트워크를 특성과 관리 영역에 따라 지역 도메인(Local Domain), 중계 도메인(Intermediate Domain) 그리고 백본 도메인(Backbone Domain)으로 나누었

다. 지역 도메인은 학교나 기업 등의 소규모 종단 네트워크를 말하며, 중계 도메인은 ISP 네트워크를 말한다. 그리고 백본 도메인은 ISP 네트워크가 상호 연결된 네트워크를 말한다.

멀티캐스트 전송을 위하여 각 도메인에서는 Xcast++를 확장한 HXcast++를 사용한다. 그리고 Xcast++와 마찬가지로 로컬 멀티캐스트를 이용하여 이동 호스트들에게 멀티캐스트 데이터그램을 전송한다. QoS 제공을 위하여 중계 도메인과 백본 도메인에서는 확장성이 뛰어난 DiffServ 방식을 사용하고, 지역 도메인에서는 명시적 QoS 보장이 가능한 IntServ/RSVP를 사용한다. 본 논문에서는 이동성, 멀티캐스트 그리고 QoS가 결합되면서 생기는 대기 시간의 합으로 나타나는 지연을 줄이는 방안에 대하여 역점을 둔다. 따라서 이동 호스트의 핸드오프로 인한 지연을 최소화하여 패킷 손실을 줄이고, 이 방법으로 해결할 수 없는 경우는 MH가 이전 DR에게 버퍼링을 요구하고 핸드오프 이후 포워딩을 요구하여 전달 받는 방법을 사용한다.

계층적 개념을 도입하기 위해 필요한 시스템 구성 요소는 다음과 같다.

#### 3.1 DR(Designated Router)

DR은 Xcast++의 DR과 마찬가지로 다수의 MH들을 대신하여 Xcast 그룹에 가입과 탈퇴를 수행하며, X2M과 M2X를 수행한다. DR은 핸드오프 시에 생길 수 있는 패킷 손실을 막기 위해 버퍼링과 포워딩 기능을 담당한다. 그리고 DR은 지역 LC의 주소를 알고 있는 것으로 가정한다.

#### 3.2 GMA(Group Management Agent)

본 논문에서는 DR과 MH 사이에서의 그룹 멤버십 관리를 위해 GMA 메커니즘을 제안한다. IGMP 사용 시 발생하는 가입 지연 및 탈퇴 지연을 최소화하기 위해서이다. GMA는 네트워크 계층 구성 요소가 아니라 응용 계층 구성 요소이다. 그리고 멀티캐스트 주소를 사용하지 않고 유니캐스트 주소를 사용하도록 한다. 그러므로 DR은 에이전트 광고를 통해 GMA의 사용을 알려야 한다.

#### 3.3 LC(Logical Core)

LC는 Xcast++의 LC와 마찬가지로 그룹에 참여한 DR들의 그룹 정보 관리를 담당하며, 계층적 기능을 수행하기 위해 지역, 중계, 백본 LC별로 추가적인 기능을 가진다.

### 3.4 GW(Gateway)

GW는 HXcast++ 제어 메시지와 데이터를 내외부 도메인으로 전달하는 역할을 담당하는 노드이다. 그림 3은 GW 노드의 내부 구조와 LC를 중심으로 한 연결 관계를 보여준다.

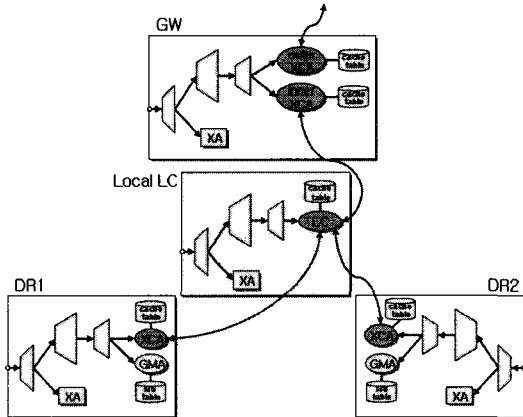


그림 3. 제어 노드들의 연결 관계

계층적 기능을 수행하기 위해 GW는 2개의 XCA (Xcast Control Agent)를 가지고, 각 XCA는 별도의 캐시 테이블을 유지 관리 한다. DR의 XCA는 LC와 연결되어 있고, GW의 inner XCA는 내부 도메인의 LC와 연결되고, GW의 outer XCA는 외부 도메인의 LC와 연결된다. GW 노드의 XA는 외부 도메인으로부터 전달되어온 데이터를 inner XCA의 캐시 테이블을 참조하여 내부 도메인의 각 DR들에게 전파시킨다. 반대로 내부 도메인의 DR들로부터 수신된 데이터는 outer XCA의 캐시 테이블을 참조하여 전달한다.

### 3.5 추가 메시지 정의

계층적 구조를 수용하고 QoS 제공을 위해 기존 Xcast++의 제어 메시지를 확장하였다. 계층적 구조를 수용하기 위한 메시지로 기존 Xcast++ 제어 메시지와 더불어 아래의 메시지를 추가로 정의하였다.

#### 1) Solicitation\_Registration\_Request

기존 Xcast++에는 LC가 네트워크 전체에 하나만 존재한다. 그러나 HXcast++에는 계층적 개념과 멤버십 정보를 분산 관리하기 위해 각 도메인마다 LC들이 존재한다. 그래서 GW는 도메인 내부의 멤버들을 대신하여 도메인 외부로 가입해야 하고 도메인 외부에서 전송되어 오는 메시지를 도메인 내부

로 전달하기 위해 도메인 내부로도 가입해야 한다. 따라서 지역 LC나 중계 LC가 지정된 GW에게 도메인을 대표하여 외부 Xcast 그룹에 가입하고 도메인 내부로도 가입을 요구하는 메시지가 필요하다. 이 메시지는 도메인 내 첫 번째 멤버가 가입하였을 때만 LC에 의해 GW로 전달된다.

#### 2) Holding\_Request

DR은 도메인 외부에 그룹 멤버가 있는지 없는지 알 수 없기 때문에 도메인 외부에 그룹 멤버가 없는 경우에도 송신 호스트를 가진 DR은 GW 쪽으로 데이터를 전송하게 된다. 따라서 도메인 외부에 그룹 멤버가 없는 경우엔 GW 쪽으로 데이터를 송신하지 않도록 알려줄 필요가 있다. GW의 outer XCA는 캐시 테이블 참조를 통해 도메인 외부에 그룹 멤버가 있는지 없는지를 알 수 있다. 따라서 outer XCA는 inner XCA에게 이벤트를 통해 알리고, inner XCA는 Holding\_Request 메시지를 도메인 내의 멤버들에게 전송한다. 이를 받은 DR은 GW를 holding 상태로 유지하고 데이터 전송 시 Xcast 헤더에 GW의 주소를 인코딩하지 않는다. 따라서 도메인 외부에 그룹 멤버가 없는 경우 GW 쪽으로의 트래픽을 발생시키지 않게 된다.

#### 3) Holding\_Release\_Request

만약 GW가 도메인 내부의 DR들에게 holding 상태로 되어있다면, 도메인 외부에 새로운 멤버가 추가되었을 때 holding 상태 해제를 요구할 수 있어야 한다. 따라서 GW는 도메인 내부의 DR들에게 Holding\_Release\_Req 메시지를 전송하여 holding 상태를 해제하고 데이터를 전송하도록 요구한다.

#### 4) QoS Parameter Extension

본 논문에서는 지역 도메인에서 IntServ/RSVP를 이용하고 중계 도메인과 백본 도메인에서는 DiffServ 기법을 이용하여 QoS를 제공하려 한다. 이를 위해 지역 도메인의 DR은 PATH 요구 메시지를 보낼 수 있어야 한다. 따라서 HXcast++의 Registration\_Regeust 메시지를 전송할 때 QoS Parameter Extension에 QoS 요구사항을 함께 전송한다. RSVP QoS Information 메시지에는 송신자가 발생할 트래픽에 대한 특성을 나타내는 RSVP Sender\_TSpec의 내용을 포함하고 있다. 이는 수신 노드가 이미 그 트래픽 특성을 알고 있다고 가정한 것이다. 이를 수신한 송신 노드를 포함한 DR이나 GW는 PATH 메

시지를 전송하도록 한다.

#### IV. Hierarchical Xcast++의 동작

본 장에서는 HXcast++의 동작 과정을 설명한다.

##### 4.1 GMA 메커니즘

Xcast++와는 다르게 본 논문에서 제안하는 HXcast++에서는 DR과 MH 사이의 그룹 관리를 위해 GMA 기반 멀티캐스트 그룹 멤버십 관리 메커니즘을 사용한다.

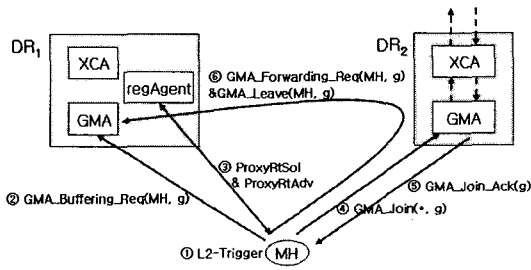


그림 4. GMA 핸드오프 절차

GMA는 그룹 관리를 위한 *GMA\_Join/GMA\_Leave* 외에 *QoS Parameter Extension*, *GMA\_Buffering\_Request* 그리고 *GMA\_Forwarding\_Request* 메시지를 포함하고 있다. 이동 호스트는 핸드오프 이후, 이전 네트워크의 GMA 유니캐스트 주소로 *GMA\_Forwarding\_Request* 메시지를 전송한다. 따라서 이동 호스트는 에이전트 광고를 통해 알려지는 GMA 관련 정보를 보관하고 있어야 한다.

MH가 DR<sub>1</sub> 영역에서 DR<sub>2</sub>로의 핸드오프 과정을 그림 4에서 보여주고 있다.

본 논문에서는 MH의 핸드오프를 L2 Mobile Trigger를 이용하여 알 수 있도록 하였다. 핸드오프 이벤트가 발생하면 멀티캐스트 응용으로 보고 되고 ①, 멀티캐스트 응용은 핸드오프 중 발생할 수 있는 패킷 손실을 줄이기 위해 *GMA\_Buffering\_Request* 메시지를 전송하여 버퍼링을 요구한다(②). GMA는 그룹 멤버십 테이블의 *Status* 필드를 *b(buffering)* 상태로 설정하고, DR이 X2M을 수행할 때 참조하도록 한다. 그런 다음 MH는 빠른 핸드오프를 위해 CoA를 요구하여 획득하고(③), GMA는 멀티캐스트 응용으로 하여금 *GMA\_Join* 메시지를 DR<sub>2</sub>에게 보내도록 한다(④). DR<sub>2</sub>의 GMA는 기존 멤버가 있을 경우, XCA에게 보고하지 않고

*GMA\_Join\_Ack* 메시지로 응답한다(⑤). MH는 *GMA\_Forwarding\_Request* 메시지를 이전 DR<sub>2</sub>에게 보내어 버퍼링된 데이터를 즉각 전송하도록 요구하며, *GMA\_Leave* 메시지를 전송하여 탈퇴하도록 한다(⑥).

##### 4.2 그룹 가입

이동 호스트(MH<sub>1</sub>)가 그림 2의 토폴로지에서 멀티캐스트 그룹에 가입하는 절차를 그림 5에 나타내었다.

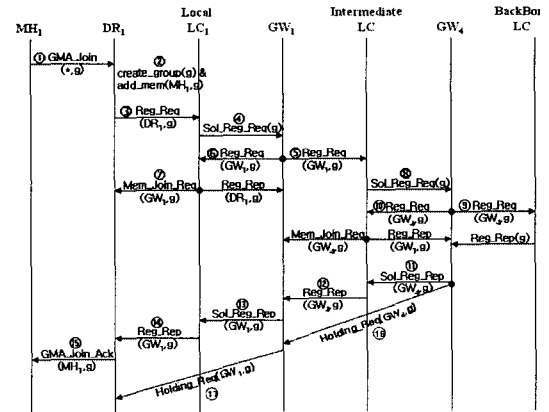


그림 5. 이동 호스트의 그룹 가입 절차

MH<sub>1</sub>이 *GMA\_Join* 메시지를 보내면(①), GMA는 이를 수신하여 그룹 엔트리를 생성하고(②), DR<sub>1</sub>의 XCA에게 알려 MH를 대신하여 그룹에 가입한다(③). *Reg\_Req* 메시지를 수신한 지역 LC<sub>1</sub>은 그룹 *g*에 관한 아무런 정보가 없기 때문에 Xcast++와는 달리 즉각 *Reg\_Rep*로 응답하지 않고 GW<sub>1</sub>에게 *Sol\_Reg\_Req* 메시지를 보내어 외부로의 가입을 요청한다(④). 즉, GW<sub>1</sub>에게 지역 도메인 내에 있는 DR들을 대신하여 Xcast 그룹에 가입하도록 요청하는 것이다(⑤). 이때 GW<sub>1</sub>은 외부로부터 수신된 Xcast 데이터를 DR들에게 전달하고, DR들로부터 수신된 Xcast 데이터를 외부에 있는 수신 노드들에게 전달하기 위해 지역 LC<sub>1</sub>에도 가입해야 한다(⑥). 따라서 GW<sub>1</sub>의 그룹 가입 사실이 DR<sub>1</sub>에게 *Mem\_Join\_Req* 메시지를 통해 알려진다(⑦). ⑤를 수신한 중계 LC도 상위 네트워크와 연결된 GW<sub>3</sub>에게 *Sol\_Reg\_Req* 메시지를 전송한다(⑧). 이를 수신한 GW<sub>3</sub>은 GW<sub>1</sub>과 마찬가지로 중계 도메인에 속한 GW들을 대신하여 상위 네트워크에 가입함(⑨)과 동시에 자신의 내부 네트워크에도 가입 한다(⑩).

이때 백본 LC는 즉각 *Reg\_Rep* 메시지로 응답 한다(⑨).  $GW_3$ 은 ⑧에 대한 응답으로 *Sol\_Reg\_Rep* 메시지로 응답한다(⑩). 중계 LC는 ⑤에 대한 응답으로 *Reg\_Rep*를 전송한다(⑫). ⑬, ⑭메시지를 통해 지역  $LC_1$ 과  $DR_1$ 도 응답을 받아 그룹에 가입 상태가 된다. 이때  $DR_1$ 이 그룹에 가입되면  $GMA$ 를 통해 *Join\_Ack*로 응답 하게한다(⑮).

$GW_3$ 은 다른 멤버( $DR$  또는  $GW$ )들의 리스트가 포함되지 않은 *Reg\_Rep* 메시지(⑨)를 응답 받았을 때, 자신이 첫 번째 가입했다는 사실을 알게 된다. 따라서 *Holding\_Req* 메시지를 도메인 내에 있는 멤버들에게 전파한다(⑯). 즉, 그룹에 가입은 하였지만 아직 Xcast 데이터는 전송하지 말라는 의미이다.  $GW_1$ 의 외부  $DR$ 이 holding 메시지를 받게 되면, 내부  $DR$ 에게 알려 *Holding\_Req* 메시지를  $DR_1$ 에게 전달 하도록한다(⑰).

### 4.3 자원 예약 및 해제

본 논문에서는 지역 도메인의 QoS 제공을 위해 IntServ/RSVP를 사용하고 이동 호스트는 수신 호스트로만 가정한다.

Xcast를 제공하는 네트워크에서 RSVP 메커니즘을 적용하기 위해서는 고려할 사항이 있다. RSVP 메시지를 Xcast 그룹 멤버들에게 보내는 패킷 헤더의 구조를 보면, IP 헤더는 다음 프로토콜을 Proto\_Xcast로 지정하고, Xcast 헤더의 다음 프로토콜을 RSVP(46)로 지정하고 있다. 그리고 Xcast는 모든 패킷의 IP 헤더에 목적지 주소를 All\_Xcast\_Router Address로 하여 전송한다. 즉, Xcast 헤더는 CI(Channel Identifier)에 해당 그룹 주소를 기록하고, IP 헤더에는 로컬 멀티캐스트 주소가 사용되기 때문이다. 따라서 본 논문에서는 Xcast 메시지를 처리할 수 있도록 RSVP의 Classifier를 수정한다. 즉 RSVP 제어 메시지를 비롯한 모든 패킷이 지나갈 때 IP 헤더의 목적지 주소만을 인식하지 않고, Xcast 헤더가 포함된 경우 IP 목적지 주소 대신에 CI를 목적지 주소로 인식할 수 있도록 RSVP Classifier를 수정한다.

그림 6은 이동 호스트의 자원 예약과 해제 절차를 나타내고 있다. QoS를 요구하는  $MH_1$ 은 그룹에 가입할 때 그룹에 서비스될 것이라고 알려진 QoS parameter를 *GMA\_Join* 메시지 전송 시 *QoS Parameter Extension* 헤더에 포함하여 전송한다(①). 이를 받은  $DR_1$ 의  $GMA$ 는 그룹 엔트리를 생성하고(②),  $XCA$ 에게 알려  $MH$ 를 대신하여 그룹에 가입

한다(③). 이때 *Reg\_Req* 메시지에도 *QoS Parameter Extension* 헤더를 포함시킨다. *Reg\_Req* 메시지를 수신한 지역  $LC_1$ 은  $GW_1$ 에게 *QoS Parameter Extension* 헤더를 포함한 *Sol\_Reg\_Req* 메시지를 보내어 가입을 요청하고(④) 4.1절에서 설명한 이후의 그룹 가입 절차를 수행한다.

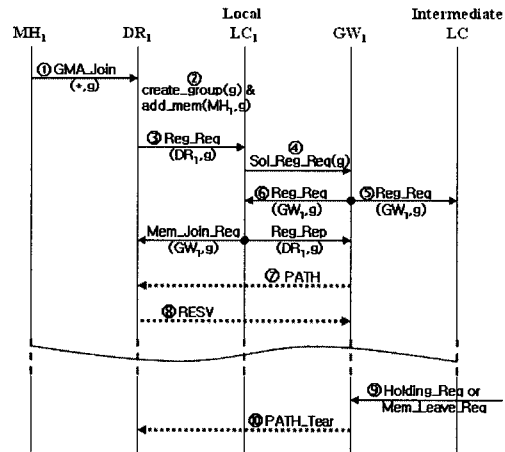


그림 6. 자원 예약 및 해제 절차

$GW_1$ 은 외부로부터 전송되어오는 멀티캐스트 데이터를 지역 도메인으로 전달하는 역할을 하므로 가상의 송신 노드가 된다. 따라서 지역 도메인 내부의  $DR$ 들로부터 *QoS Parameter Extension* 헤더가 포함된 *Sol\_Reg\_Req*(혹은 *Reg\_Req*) 메시지를 받으면 즉각 *PATH* 메시지를 전송한다(⑦). *PATH*를 받은  $DR_1$ 은 *RESV* 메시지를 전송하여 자원 예약을 수행한다(⑧).  $GW_1$ 이 외부 도메인으로부터 *Holding\_Req*나 *Mem\_Leave\_Req* 메시지를 받아 active 상태에 있는 멤버가 없는 경우엔(⑨) 지역 도메인 내 모든 멤버  $DR$ 들에게 *PATH\_Tear* 메시지를 보내어 자원을 해제한다(⑩).

### 4.4 데이터 전송

그림 7은  $MH_1$ 이 전송한 멀티캐스트 데이터그램이 그룹 멤버들에게 전달되는 과정을 보여준다.

$MH_1$ 이 멀티캐스트 데이터그램을 전송하면,  $DR_1$ (즉,  $DR$ )이 이를 받아서 처리한다.  $GMA$  그룹 멤버십 테이블을 참조하여 그룹에 속한 다른  $MH$ 들이 있다면 로컬 멀티캐스트를 이용하여 멀티캐스트 데이터를 전송하고, 캐시 테이블을 참조하여 Xcast 패킷을 전송한다.  $GW_1$ 이 도메인 내부로부터 데이터를 수신하면, 외부  $DR$ 의 캐시 테이블을 참조하여 데이

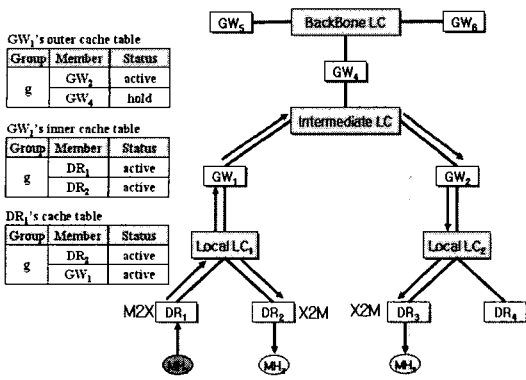


그림 7. 멀티캐스트 데이터의 전송

터 전달한다. 이때, GW<sub>1</sub>은 holding 상태에 있으므로 Xcast 패킷을 생성할 때 포함시키지 않는다. 그러므로 GW<sub>1</sub>에게만 데이터가 전송되고, 이를 받은 GW<sub>1</sub>는 도메인 외부로부터 수신된 데이터이므로 내부 DR의 캐시 테이블을 참조하여 도메인 내부로 전달한다.

4.5 핸드오프

그림 8은 QoS 서비스를 받던 DR<sub>2</sub>의 마지막 멤버인 MH<sub>2</sub>가 동일한 멀티캐스트 그룹의 멤버가 서비스 받고 있는 DR<sub>3</sub>으로 도메인 간 핸드오프 절차를 보여주고 있다.

MH<sub>2</sub>는 핸드오프 이벤트를 받아 버퍼링을 요구하고(①), 새로운 CoA를 획득한 후, GMA\_Join 메시지를 보낸다(②). 이를 수신한 DR<sub>3</sub>은 이미 그룹에 가입이 된 상태이므로 즉시 GMA\_Join\_Ack 메시지를 송신한다(⑤). GMA\_Join\_Ack를 받은 MH<sub>2</sub>는

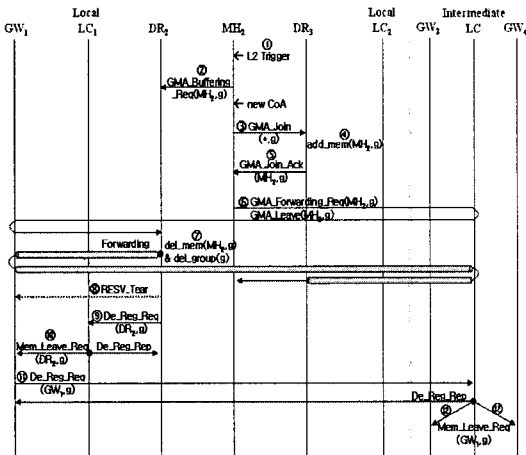


그림 8. 도메인 간 핸드오프 절차

GMA\_Forwarding\_Request를 DR<sub>2</sub>에게 보내어(⑥) 핸드오프 기간 중 받지 못한 메시지를 터널링으로 포워딩 받는다(⑥). 그리고 GMA\_Leave를 보내어 DR<sub>2</sub>가 예약된 자원을 해제할 수 있도록 한다(⑧). 그런 다음, 이후의 탈퇴 절차를 수행한다.

V. 성능 평가

본 논문에서 제안하는 HXcast++ 기법의 성능을 평가하기 위해 네트워크 시뮬레이션 툴인 NS(Network Simulator) 버전 2.26[12]을 이용하고 [13]의 Xcast++ 모듈을 수정 확장하였다. 그리고 [14]의 RSVP 모듈을 수정하여 사용하였고, [15]의 DiffServ 모듈을 수정 없이 각각 사용하였다.

핸드오프 시 패킷 손실을 측정하기 위해 그림 9의 토폴로지를 가정하였다. 하나의 멀티캐스트 송신 호스트가 있고, 각 노드 간 지연은 임의대로 설정한 것이다.

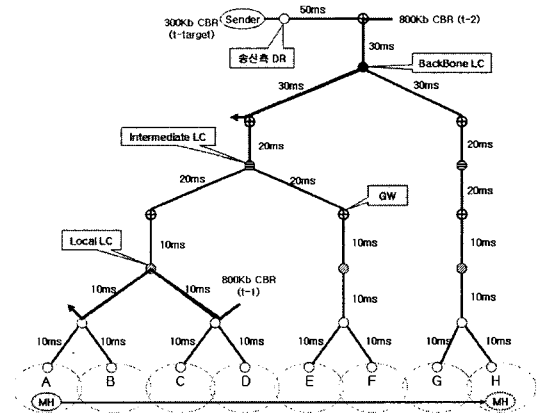


그림 9. 실험 네트워크 구성

5.1 수신 호스트의 핸드오프에 따른 패킷 손실  
수신 호스트의 핸드오프 시 패킷 손실을 측정하기 위해 DR들의 에이전트 광고주기는 3초로 하였으며, 이동 호스트의 이동속도는 20m/sec로 하였다. 그리고 각 링크 대역폭은 1Mbps로 하였으며 링크 지연은 그림 9에 표시한 것과 같다. 그리고 MH를 제외한 다른 이동 호스트는 없으며, 배경 트래픽도 없는 것으로 가정하였다. 송신 트래픽을 100Kbps의 CBR로 하고, 에이전트 광고주기에 관한 seed를 다르게 하여 10회 실험하였다.

HXcast++ 기법을 이용하여 하나의 이동 호스트 (MH)가 A cell에서 시작하여 H cell까지 이동하면



서 수신한 패킷의 양과 패킷 손실 그리고 중복 수신된 패킷을 측정하였다. HXcast++만 사용한 경우와 패킷 손실을 줄이기 위한 각 방식(빠른 핸드오프, buffering&forwarding)을 적용한 결과를 그림 10에 나타내었다.

그림 10(a)은 HXcast++만 사용한 경우이다. 핸드오프 구간마다 패킷 손실이 발생하는 것을 볼 수 있다. 에이전트 광고 주기가 길면 길수록 패킷 손실은 많이 발생하게 된다.

그림 10(b)은 L2 Mobile Trigger를 이용하여 핸드오프 직전에 DR에게 버퍼링을 요구하고, 에이전트 광고를 받아 정상적으로 Join\_Ack를 수신한 다음, 포워딩을 요구한 경우의 그래프이다. 이 그래프에서는 손실되는 패킷이 없었으며, 적은 양의 중복 패킷이 발생하였다. 그러나 에이전트 광고 주기가 긴 경우 버퍼링되는 패킷이 많아져 버퍼를 위한 많은 메모리가 필요한 문제가 있을 수 있다.

그림 10(c)은 L2 Mobile Trigger를 이용하여 빠른 핸드오프를 수행한 결과이다. 빠른 핸드오프로 인하여 패킷 손실이 상대적으로 적기는 하나, 핸드오프 이후 멀티캐스트 트리가 생성되는 동안은 데이터를 수신할 수 없음을 알 수 있다.

그림 10(d)은 빠른 핸드오프와 buffering & forwarding을 함께 사용한 경우이다. 도메인 간 핸드오프 시 buffering&forwarding 기법만 사용하였을 경우보다 상대적으로 소량의 buffer를 소모하였고 패킷의 손실은 전혀 발생하지 않았다. 그러나 적은 양의 중복 패킷이 발생하였다.

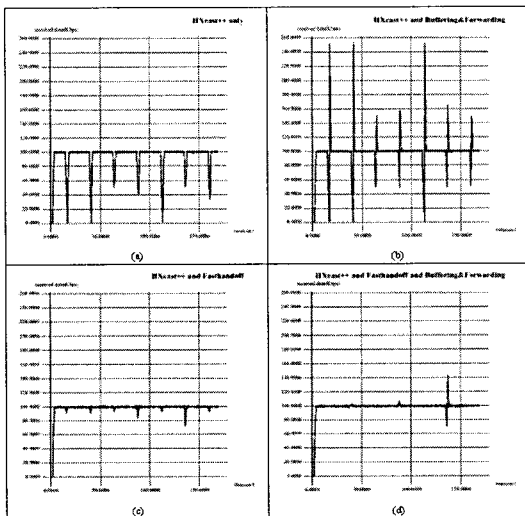


그림 10. 핸드오프 방법에 따른 패킷 손실

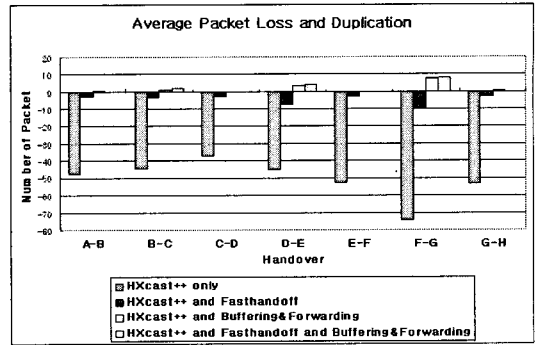


그림 11. 평균 패킷 손실 및 중복 수신

그림 11은 각 방식에 대한 평균 패킷 손실 및 중복 수신 패킷을 비교한 그래프이다. HXcast++만 사용한 경우는 에이전트 광고주기에 영향을 받아 다량의 패킷 손실이 불규칙하게 발생하였다. 그리고 HXcast++와 빠른 핸드오프 기법을 함께 사용한 경우엔 에이전트 광고주기에 영향을 받지 않고 DR의 Xcast 그룹 가입 지연에 해당하는 시간만큼의 패킷 손실이 발생하였다. 그리고 HXcast++와 buffering & forwarding 기법을 함께 사용한 경우엔 패킷 손실이 전혀 발생하지 않았으며 오히려 소량의 중복 패킷이 수신됨을 알 수 있었다. 중복 패킷의 발생 원인은 수신 호스트가 포워딩을 요구하는 시점에 이미 멀티캐스트 데이터를 new DR로부터 수신하기 때문이다. 즉, 수신 호스트가 GMA\_Forwarding\_Request 메시지를 보낸 후 old DR에 도착할 때까지 버퍼링된 패킷들은 중복된다.

따라서 수신 호스트의 핸드오프 시 발생할 수 있는 패킷 손실을 줄이기 위해서는 즉각적인 가입이 가능한 GMA와 같은 메커니즘이 필요하며, 또한 빠른 핸드오프와 buffering&forwarding 기법을 함께 사용해야 함을 알 수 있다.

### 5.2 QoS 제공 방안에 관한 성능 평가

수신 호스트의 이동에 대하여 종단 간 QoS 제공이 잘 이루어지는지를 확인하기 위해 그림 9의 토폴로지를 가정하였다.

DR들의 에이전트 광고주기는 3초로 하였으며, 이동 호스트의 이동속도는 20m/sec로 하였다. 그리고 각 링크 대역폭은 1Mbps로 하였으며 링크 지연은 그림 9와 같다. HXcast++ 기법을 이용하여 하나의 이동 호스트(MH)가 A cell에서 시작하여 H cell까지 이동하면서 수신한 패킷의 양을 측정하였다. MH를 제외한 다른 이동 호스트는 없으며, 빠른 핸드오

프 기법과 buffering&forwarding 기법을 사용한다.

배경 트래픽의 영향을 확인하기 위해 아래 4가지 경우를 가정하여 측정하였다.

- case 1 : 송신 호스트가 t-target(300 Kbps, CBR traffic)을 발생시키고, 배경 트래픽은 발생하지 않는다.
- case 2 : 송신 호스트가 t-target(300 Kbps, CBR traffic)을 발생시키고, 배경 트래픽으로 IntServ 구간에 t-1(800 Kbps, CBR traffic)을 best effort로 발생시킨다.
- case 3 : 송신 호스트가 t-target(300 Kbps, CBR traffic)을 발생시키고, 배경 트래픽으로 DiffServ 구간에 t-2(800 Kbps, CBR traffic)를 송신 호스트의 요구와 동일한 클래스로 발생시킨다.
- case 4 : 송신 호스트가 t-target(300 Kbps, CBR traffic)을 발생시키고, 배경 트래픽으로 IntServ 구간에 t-1 트래픽best effort로 발생시키고, DiffServ 구간에 t-2 트래픽을 송신 호스트의 요구와 동일한 클래스로 발생시킨다.

case 1(그림 12(a))의 경우는 배경 트래픽이 없으므로 패킷 손실도 거의 발생하지 않았다.

case 2(그림 12(b))의 경우는 IntServ 구간에 best effort 트래픽이 배경 트래픽으로 주어졌다. 지역 네

트워크에서는 IntServ 방식으로 보장형 서비스를 받고 있으므로 배경 트래픽이 손실되는 것을 볼 수 있다. 그러나 보장형 서비스를 받고 있긴 하지만, 버퍼링 후 포워딩을 하였을 때, 포워딩 되는 패킷에 대해서는 보장하지 않으므로 이를 수신하지 못하는 문제를 확인할 수 있다.

case 3(그림 12(c))의 경우는 DiffServ 구간에 송신 호스트의 서비스 클래스와 동일한 클래스로 배경 트래픽을 발생시켰다. 따라서 두 트래픽은 경쟁적으로 전송되므로 두 트래픽 모두 패킷 손실이 생기게 된다. 이것은 DiffServ가 명시적 보장을 하지 않기 때문에 생기는 현상이다.

case 4(그림 12(d))의 경우는 case 3의 경우와 동일하게 나타났다. 즉, IntServ 구간에서는 명시적 보장이 되므로 어떠한 트래픽으로부터도 영향을 받지 않으나, DiffServ 구간에서는 경쟁적으로 패킷 손실이 발생할 수 있음을 알 수 있다.

## VI. 결론

본 논문에서는 이동 통신망 환경에서 사용자들의 다양한 요구사항을 충족시킬 수 있도록 End-to-End QoS를 제공하기 위한 이동 멀티캐스트 기법을 제안하였다. 이를 위해 이동성, 멀티캐스트 그리고 QoS가 결합되면서 발생하는 문제점들을 정의하고, 이를 해결하기 위한 종합적인 방안을 제시하였다.

투명한 이동성 보장을 위해 빠른 핸드오프 기법을 적용하였다. L2 Mobile Trigger를 이용하므로 핸드오프 시 발생하는 지연으로 인한 다량의 패킷 손실을 줄일 수 있도록 하였다. 그리고 효율적인 멀티캐스트를 제공하기 위해 기존의 Xcast++에 계층적 개념을 도입하여 확장한 HXcast++를 제안하여 빈번한 핸드오프로 인한 멀티캐스트 서비스 유지비용을 줄일 수 있도록 하였다. 이동 호스트들을 대신하여 DR이 그룹에 가입하도록 함으로써 이동 호스트의 위치와 상관없이 최적화된 경로로 서비스를 받을 수 있도록 하였으며, 이동 통신망 환경에 적합한 그룹 멤버십 관리를 위하여 GMA 메커니즘을 제안하여 그룹 가입 지연을 줄이고, 탈퇴 지연으로 인한 자원의 낭비를 제거하였다. buffering & forwarding 기법을 이용하여 멀티캐스트 트리가 생성되는 동안 발생할 수 있는 패킷 손실은 줄였다. 또한 QoS 제공을 위하여 최근의 연구 동향에 맞춰 지역 네트워크에서는 IntServ/RSVP를 사용하고 광역 네트워크에서는 DiffServ를 사용하도록 구성하였

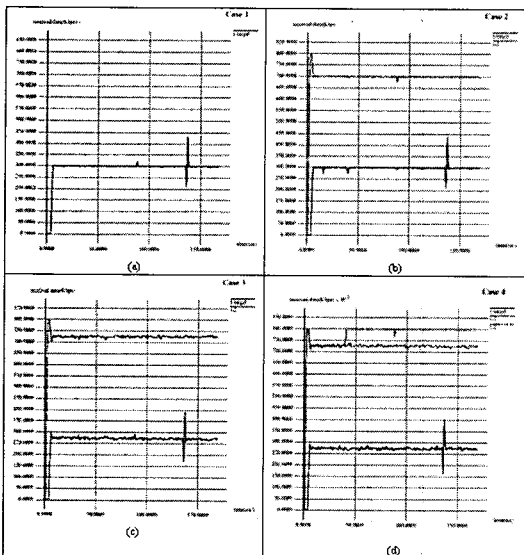


그림 12. 배경 트래픽의 영향

다. 그리고 핸드오프로 인한 RSVP의 세션 재 설립 지연을 줄이기 위해 HXcast++ 제어 메시지를 확장하여 PATH 메시지를 요구하도록 하였다.

본 논문에서 제안하는 HXcast++ 기법은 이동 통신망 환경에서 End-to-End QoS를 제공하는 멀티캐스트 기법으로서 다양한 요구사항을 충족시킬 수 있는 한 가지 방안이 될 것이다.

참 고 문 헌

[1] C-H. Wu and J. D. Irwin, *Emerging Multimedia Computer Communication Technologies*, Prentice Hall, 1998.

[2] A. Striegel and G. Manimaran, "A Survey of QoS Multicasting Issues," *IEEE Communications*, vol. 40, no. 6, pp. 82-87, June 2002.

[3] K-I Kim, et al., "Xcast+ Extension for Few-to-Few Multicast Communication," *IETF Internet-Draft*, draft-kim-xcast+-few-2-few-00.txt, October 2002.

[4] J. Wu, "A Mobility Support Agent Architecture for Seamless IP Handover," *Licentiate Thesis, Royal Institute of Technology (KTH)*, Sweden, June 2000.

[5] K. Malki, et al., "Low latency Handoffs in Mobile IPv4," *IETF Internet-Draft*, draft-ietf-mobileip-lowlatency-handoffs-v4-04.txt, June 2002.

[6] R. Koodli, "Fast Handovers for Mobile IPv6," *IETF Internet-Draft*, draft-ietf-mobileip-fast-mipv6-06.txt, March 2003.

[7] C. Perkins, "IP Mobility Support for IPv4," *IETF RFC 3344*, August 2002.

[8] J-W Lee, "Explicit Multicast over Mobile IP," *IETF Internet-Draft*, draft-lee-xcast-mobileip-01.txt, November 2001.

[9] M-K Shin, Y-J Kim and S-H Kim, "Explicit Multicast Extension(Xcast+) Supporting Receiver Initiated Join," *IETF Internet-Draft*, draft-shin-xcast-receiver-join-01.txt, March 2002.

[10] 김재현, 김기택, 김동근, 이재용, "이동환경에

서 다중 송수신자를 지원하는 효율적인 멀티캐스트 기법", *JCCI 2003*, 2003. 4.

[11] Y. Berenet, et al., "A Framework for End-to-End QoS Combining RSVP/IntServ and Differentiated Services," *IETF Internet-Draft*, draft-berenet-intdiff-00.txt, March 1998.

[12] NS Simulator Ver 2.26, <http://www.isi.edu/nsman/ns/>

[13] Xcast++ module, <http://lionking.cnu.ac.kr/~multicast/>

[14] Marc Greis' RSVP module, <http://www.ncc.up.pt/~rprior/ns/index-en.html>

[15] Notel Networks DiffServ Model, <http://www7.nortel.com:8080/CTL/>

김 태 수 (Tae-Soo Kim)

정회원



1995년 2월 경상대학교 입산 공학과 졸업  
 1999년 2월 창원대학교 컴퓨터 공학과 석사  
 2004년 8월 창원대학교 컴퓨터 공학과 박사  
 2004년~현재 창원대학교 컴퓨터 공학과 IT 초빙교수

<관심분야> 멀티캐스트 프로토콜, 이동통신망, QoS 관리, 센서 네트워크

이 광 휘 (Kwang-Hui Lee)

정회원



1983년 2월 고려대학교 전자공학과 졸업  
 1985년 2월 고려대학교 전자공학과 석사  
 1989년 2월 고려대학교 전자공학과 박사  
 1988년~현재 창원대학교 컴퓨터 공학과 교수

1991년~1992년 영국 Walse대학(Swansea) 및 Newbridge Networks사 연구원  
 1994년~1995년 영국 런던대학(UCL)연구원  
 1997년~1999년 영국 Reading 대학 연구원  
 <관심분야> 멀티캐스트 프로토콜, 이동 통신망, QoS 관리, QoS 라우팅, 분산시스템, 센서 네트워크