

이동 통신망 환경에서 멀티캐스트를 제공하기 위한 계층적 Xcast++ 기법[☆]

A hierarchical Xcast++ mechanism for multicast services in mobile communication environment

김 태 수*
Tae-Soo Kim*

이 광 휘**
Kwang-Hui Lee**

요 약

이동 통신망 환경에서 이동 호스트에게 멀티캐스트 서비스를 제공하기 위한 방안으로 본 논문에서는 기존의 Xcast++를 계층적 구조를 갖도록 확장한 HXcast++를 제안하였다. 이동 호스트들을 대신하여 DR(Designated Router)이 그룹에 가입하도록 함으로써 이동 호스트의 위치와 상관없이 최적화된 경로로 서비스를 받을 수 있도록 하였으며, 계층적 개념을 도입하여 빈번한 핸드오프로 인한 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 제안하였고, 핸드오프 시 발생하는 다량의 패킷 손실을 줄이기 위해 Layer 2 Mobile Trigger를 이용하는 fast handoff 기법을 적용하였다. 그룹 가입 지연 동안 발생할 수 있는 패킷 손실은 buffering&forwarding 기법을 이용하여 손실을 줄였다.

Abstract

In order to provide mobile hosts with multicast service in mobile communication environment, we proposed a multicast mechanism named HXcast++ which is an extended version of the existing Xcast++ with hierarchical architecture.

We assured that mobile hosts could get multicast service through an optimal path regardless of their location by making DR(Designated Router) join a group on behalf of the mobile hosts. In this present research we introduced hierarchical architecture in order to reduce the maintenance cost resulting from frequent handoff. We also proposed a GMA (Group Management Agent) based group management mechanism which enables the mobile hosts to join the group without waiting for a new IGMP Membership Query.

A fast handoff method with L2 Mobile Trigger was, in this work, employed in order to reduce the amount of the packet loss which occurs as a result of the handoff. We also managed to curtail the packet loss caused by the latency of the group join by using a buffering and forward mechanism.

† Keyword : 이동 IP, 멀티캐스트, Xcast, HXcast++.

1. 서 론

이동 통신의 발달로 이동 환경에서 사용자들의 요구사항은 다양화되고 있으며, 고정망에서와 같은

수준의 서비스를 요구하고 있다. 통신망을 이용한 원격 교육, 인터넷 방송, VOD, 온라인 게임 등 다양한 응용들이 늘어나고 있다. 이들 서비스들은 대부분 멀티캐스트 기술을 필요로 하고 있다. 따라서 이동 통신망 환경에서 멀티캐스트 서비스의 제공은 이동 통신 사용자들의 다양한 요구사항을 충족시키기 위한 기본 서비스가 되고 있다.

기존 멀티캐스트 라우팅 프로토콜의 대부분이 고정 호스트를 가정하여 만들어졌기에 이동성을 고려하지 않으므로 이동 망 환경에 그대로 적용하

* 정 회 원 : 창원대학교 컴퓨터공학과 IT 초빙 교수
tskim@changwon.ac.kr(제 1저자)

** 정 회 원 : 창원대학교 컴퓨터공학과 교수
khlee@changwon.ac.kr(공동저자)

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

[2004년/05/14 투고-2004/05/31 심사-2004/10/21 심사완료]

기 어렵다. 이러한 문제를 해결하기 위해 여러 가지 방안들이 여러 연구자와 연구기관들에 의하여 수행되었다.

이동 망 환경에서 멀티캐스트를 지원하기 위한 방법으로 IETF(Internet Engineering Task Force)에서는 Mobile IP에서 사용 가능한 HA(Home Agent)기반 방식과 FA(Foreign Agent)기반 방식을 제안하고 있다[1,2]. FA기반 방식에서는 이동한 네트워크의 FA를 통해 멀티캐스트 트리에 재가입 한다. 이는 이동 호스트가 어떤 곳으로 이동하더라도 이동 호스트의 위치에 맞게 멀티캐스트 경로를 최적화 할 수 있는 장점이 있다. 하지만 Co-located CoA를 사용해야 하며 호스트가 이동할 때마다 멀티캐스트 트리를 재구성해야 하는 오버헤드를 가진다. 이와는 다르게 HA기반 방식에서는 이동 호스트가 어떤 곳으로 이동하더라도 HA로부터 터널링을 통해 멀티캐스트 서비스를 받는다. 이 방법은 이동 호스트의 움직임에 대한 투명성을 보장할 수는 있으나, 멀티캐스트 서비스경로가 비최적화 되고 이동 호스트 당 터널을 가져야 하기 때문에 터널에 대한 비용이 증가된다. 뿐만 아니라 그룹 관리를 위해 IGMP를 사용할 경우, 주기적인 IGMP membership query 메시지로 인해서 긴 가입 지연 및 탈퇴 지연이 발생하게 된다[3,4].

이동 사용자가 증가하고 지역적으로 밀집되는 현상으로 인해 셀의 크기가 작아져 핸드오프의 빈도가 증가하고 있다[5]. 빈번한 핸드오프로 인하여 멀티캐스트 트리의 갱신이 유발된다. 뿐만 아니라, Mobile IP의 경우 L3(Layer 3)와 L2(Layer 2)를 분리하는 체계를 가지고 있으므로 확장성은 좋지만 빠른 이동성 감지가 어렵다[6-8]. 그러므로 Mobile IP에서는 느린 이동 감지 및 위치 등록으로 인한 긴 지연이 발생하게 되며 핸드오프 지연 기간에 비례하여 패킷 손실이 발생한다.

따라서 이동 통신망 환경에서 이동 호스트에게 멀티캐스트 서비스를 제공하기 위해서는 다음과 같은 조건을 만족하는 멀티캐스트 서비스의 제공이 필요하다.

첫째, 중복되지 않고 최적화된 경로로 멀티캐스트 데이터그램이 전송되게 해야 한다. 즉, 멀티캐스트 전송의 효율성을 위해 이동 호스트의 위치에 상관없이 최적화된 전송 경로를 따라 멀티캐스트 데이터그램이 전송 되도록 하고 중복 전송을 피해야 한다. 이것은 멀티캐스트 자체의 효율성은 물론 QoS(Quality of Service)제공에 많은 영향을 미치기도 한다.

둘째, 빈번한 핸드오프가 발생하더라도 멀티캐스트 서비스를 유지하기 위한 비용이 최소화 되어야 한다. 기존 멀티캐스트 프로토콜의 경우, 멀티캐스트 트리를 유지해야 하는 비용이 크다. 즉, 주기적인 브로드캐스팅 방식이나 멤버의 변화에 따른 멀티캐스트 트리 갱신 비용은 멀티캐스트의 효율성을 떨어뜨릴 뿐만 아니라 확장성을 저해하는 요인이 된다. 따라서 빈번한 핸드오프로 인한 멤버십의 변화에도 멀티캐스트 서비스를 유지하기 위한 비용을 최소화할 수 있어야 한다.

셋째, 핸드오프로 인한 패킷 손실을 최소화해야 한다. 특히 느린 이동성 감지로 인한 핸드오프 지연과 멀티캐스트 가입 지연 등의 복합적인 지연에도 불구하고 서비스 단절 현상이나 서비스의 품질을 저하시키지 않는 방안이 필요하다.

본 논문에서는 첫 번째와 두 번째 제약 조건을 만족할 수 있도록 최근 소규모 그룹 통신을 위해 제안되고 있는 Explicit Multicast(Xcast) 기법[9-11]을 이용한다. 그중 few-to-few Xcast+(이하 Xcast++)[11]는 멀티캐스트 트리를 구성하고 관리하는 오버헤드가 없으며, 멀티캐스트 데이터그램을 중복 없이 최적화된 경로로 전송함을 보장한다. 그러나 중소규모의 그룹을 지원하도록 설계하였으며, 그룹 멤버가 널리 퍼져있는 경우 패킷 헤더를 처리하는 비용이 증가하는 단점을 가지고 있다. 따라서 기존 Xcast++에 계층적 개념을 도입하여 중대규모 그룹을 지원하도록 확장하는 방안을 제안하고자 한다.

이와 더불어 세 번째 제약 조건을 극복하는 방안으로 핸드오프 시 IGMP Membership Query를

기다림 없이 즉각 그룹에 가입할 수 있는 GMA (Group Management Agent) 기반 그룹 관리 메커니즘을 제안한다. 또한 기존 L2 Trigger를 이용하여 빠르게 이동성을 감지하고 이를 이용하여 핸드오프 시 발생할 수 있는 패킷 손실을 줄이기 위해 이전 에이전트에서 버퍼링하였다가 포워딩하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 살펴보고, 3장에서는 기존 Xcast++에 계층적 개념을 도입한 HXcast++의 구성 요소와 동작 과정과 GMA 기반 그룹 관리 메커니즘에 대해 설명한다. 4장에서는 기존 시스템과 제안 시스템의 성능을 비교 분석하고, 마지막으로 5장에서는 결론과 향후 연구 과제를 언급한다.

2. 관련 연구

2.1 Explicit Multicast 기법

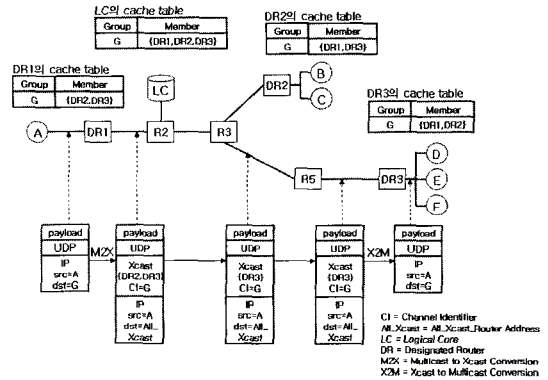
Xcast+[9]는 소규모 그룹 통신에 초점을 맞추어 유니캐스트 전송을 기반으로 멀티캐스트 서비스를 제공하는 Xcast(Explicit Multicast)[10] 기법에 호스트 그룹 관리 제어어를 추가하여 중규모 그룹 통신을 지원할 수 있도록 확장한 방법이다. 그러나 Xcast+는 하나의 송신자만을 허용하는 소스 기반 멀티캐스트 응용을 전제하고 있다.

Xcast+ few-to-few(Xcast++)[11]는 기존 Xcast+의 제어 플레인를 확장하여 few-to-few 멀티캐스트를 가능하게 한 방식으로 DR(Designated Router)이 하위의 가입자에 대해서는 로컬 멀티캐스트를 이용하여 서비스하며 DR이 대신하여 Xcast 그룹에 가입한다. LC(Logical Core)개념을 도입하여 동적으로 그룹 멤버십 관리를 수행한다.

그림 1은 Xcast++의 그룹 관리 메커니즘 및 데이터 전달 메커니즘을 보여주고 있다. 호스트(A~F)는 IGMP/MLD를 이용하여 그룹에 가입하며, DR은 호스트를 대신하여 xcast 그룹에 가입한다. DR들은 LC와 Registration Request/ Registration Reply

메시지를 교환하여 그룹 멤버 정보를 cache table에 저장해 둔다.

송신호스트 A가 멀티캐스트 메시지를 전송하면, DR1은 자신의 cache table을 참조하여 Xcast 헤더에 그룹 멤버 DR들의 주소 목록을 엔코딩 한 후, Xcast 지원 가능한 라우터에게 목적지의 다음 홉을 기준으로 전송한다(M2X). 교차 지점인 R3에서는 유니캐스트 라우팅 정보를 참조하여 패킷을 분할하여 전송하고, 수신 측 DR이 메시지를 받았을 때는 표준 멀티캐스트 패킷으로 그룹에 가입한 수신자들에게 전송한다(X2M).



〈그림 1〉 Xcast++ 개요

Xcast++는 DR간 제어 메커니즘을 추가하여 동적인 그룹 관리 및 제어를 통한 다중 송수신자들 간의 멀티캐스트 통신을 가능하도록 하였다. 이 기법은 멀티캐스트 트리를 구성하고 관리하는 오버헤드가 없으며 멀티캐스트 데이터그램의 최적화된 전송을 보장한다. 그러나 멀티캐스트 그룹의 멤버가 널리 퍼져있는 경우 패킷 헤더를 처리하는 부하가 증가하므로 제한적이다.

2.2 그룹 관리 기법

IP 멀티캐스트는 그룹 관리와 멀티캐스트 라우팅으로 구성된다. 그룹 관리 프로토콜은 지역 멤버십 정보를 동적으로 유지 관리하고 멀티캐스트

라우팅 프로토콜(DVMRP, CBT, MOSPF, PIM-DM, PIM-SM 등)은 그룹 정보를 이용하여 멀티캐스트 데이터그램을 전송한다[12].

그룹 관리 프로토콜로 가장 일반적으로 사용되고 있는 것이 IGMP(Internet Group Management Protocol)이다. 멀티캐스트 라우터는 IGMP를 이용하여 LAN(local area network)상에 어떤 그룹의 멤버가 있는지를 알기위해 주기적으로 IGMP membership query 메시지를 전송하고, 이를 받은 호스트는 IGMP report 메시지로 자신이 가입한 멀티캐스트 그룹 정보를 보고한다. Query에 대한 응답이 없는 경우 그룹 테이블에서 그룹 정보를 삭제한다. 또한 즉각적인 그룹 탈퇴를 위해 IGMP leave 메시지를 이용하고 있다[13,14].

FA 기반 멀티캐스트 서비스를 가정할 경우, 이동 호스트는 이동한 지역 멀티캐스트 라우터를 통해 멀티캐스트 그룹에 가입하여야 한다. 그러나 이동 호스트가 핸드오프 하였을 경우 IGMP 응용이나 IGMP 메커니즘 자체는 핸드오프를 감지할 수 없으므로 주기적으로 방송되는 IGMP query 메시지를 기다려야만 한다. 뿐만 아니라, IGMP report 메시지의 폭주를 회피하기위해 보고 지연 타이머(backoff timer)를 두고 있으며, 멀티캐스트 트리에 연결되는 지연이 추가적으로 발생한다. 따라서 핸드오프 이후 상당히 긴 시간의 가입 지연이 발생하여 패킷 손실 및 서비스 단절을 유발시키며, 또한 IGMP query에 대한 응답이 없을 경우 탈퇴하는 방식을 채택하고 있으므로 탈퇴 지연으로 인한 네트워크 자원의 낭비를 초래하게 된다. 이러한 문제를 해결하기 위해 MSA (Mobility Support Agent)를 이용한 사전 가입 기법[3] 및 별도의 그룹 관리를 수행하는 MHA(Multicast Handoff Agent)기법[4] 등이 제안되고 있다.

2.3 빠른 핸드오프 기법

기존의 Mobile IP는 네트워크 계층만으로 이동을 감지하여 핸드오프를 처리하고 있다. 즉, 에이전트는

주기적으로 에이전트 광고(Agent Advertisement) 메시지를 방송하고 이를 수신한 MH가 이동을 감지하게 된다. 그러나 에이전트 광고 주기가 긴 경우 MH는 기존 에이전트의 영역을 벗어나더라도 새로운 에이전트 광고 메시지가 수신되지 않으면 이동 사실을 알 수 없다. 따라서 다량의 패킷 손실이 발생할 수 있으며, 이로 인한 서비스의 단절이 발생할 수도 있다. 이를 해결하기 위해 주기를 짧게 할 경우 MH의 전력 소비가 많아지게 되는 문제가 생긴다. 그래서 이러한 문제를 해결하기 위해 Low latency handoff[6], FA Assisted handoff[7], FMIPv6[8] 등은 L2 트리거 기법을 이용하여 기존 Mobile IP의 단점을 보완할 수 있도록 제안하고 있으며, 교차 라우터나 old FA에서 핸드오프가 완료될 때까지 버퍼링하였다가 핸드오프가 완료된 후 포워딩하는 방법[15-17] 등이 제안되고 있다.

3. 제안 시스템(HXcast++)

본 논문에서는 이동 호스트들에게 효율적인 멀티캐스트 서비스를 제공함과 동시에 확장성을 부여하기 위한 방안으로 기존 Xcast++에 계층적 개념을 도입하여 확장하였다.

3.1 시스템 구성 요소

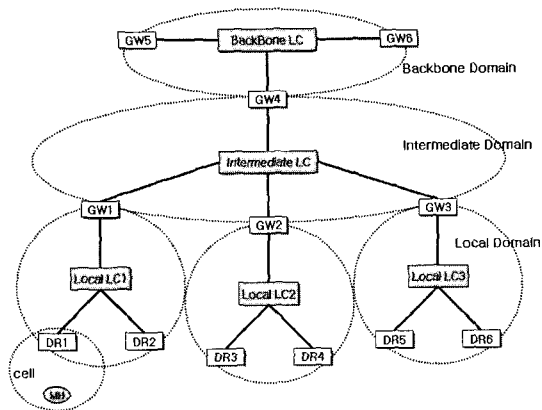
본 논문에서는 백본 도메인, 중계 도메인, 지역 도메인으로 계층화 시키고, 각 도메인에 LC들을 두었다(그림2). 그리고 GW(Gateway)들은 망간 상호연결을 담당하고, DR은 Mobile IP 서비스(Foreign Agent)와 함께 Xcast++의 DR역할을 담당하는 라우터들이다.

계층적 개념을 도입하기 위해 필요한 시스템 구성요소는 다음과 같다.

3.1.1 Designated Router(DR)

DR은 Xcast++의 DR과 마찬가지로 다수의 이동 호스트(MH)들을 대신하여 Xcast그룹에 가입

과 탈퇴를 수행하며, X2M과 M2X를 수행한다. DR은 Local LC의 주소를 알고 있는 것으로 가정하였다. DR은 핸드오프 시에 생길 수 있는 패킷 손실을 막기 위해 buffering과 forwarding 기능을 담당한다.



〈그림 2〉 계층적 구조

3.1.2 Group Management Agent(GMA)

본 논문에서는 DR과 MH사이에서는 그룹 멤버쉽 관리를 위해 GMA를 제안한다. 2.2절에서 언급한 IGMP 사용 시 발생하는 가입 지연 및 탈퇴 지연을 최소화하기 위해서이다. DR(FA)는 에이전트 광고를 통해 GMA의 사용을 알려야 한다.

3.1.3 Logical Core(LC)

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

3.1.4 게이트웨이(GW)

게이트웨이는 본 논문에서 추가된 요소이며, Xcast++ 제어 메시지와 데이터그램을 내·외부 도메인으로 전달을 담당하는 노드이다. 즉, 그림 3에서 GW1은 지역 도메인의 DR 역할과 중계 도메인의 DR

역할을 겸하며 cache table 2개를 가진다. GW는 내·외부 도메인의 LC의 주소를 알고 있는 것으로 가정하였다.

3.1.5 추가적인 메시지 정의

계층적 구조를 수용하기 위해 본 논문에서는 기존 Xcast++의 제어 메시지에 아래의 4가지 메시지를 추가로 정의하였다. 각 메시지들의 사용에 대하여는 다음 절에서 설명한다.

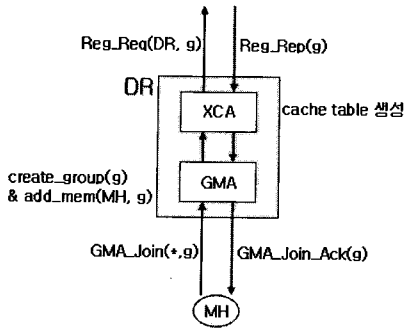
- Solicitation Registration Request(Sol_Reg_Req) : Local LC나 Intermediate LC가 상위의 GW에게 도메인을 대표하여 Xcast 그룹에 가입하기를 요구하는 메시지
- Solicitation Registration Reply(Sol_Reg_Rep) : Solicitation Registration Request 메시지에 대한 응답으로 GW가 그룹에 가입되었음을 알리는 메시지
- Holding Request(Holding_Req) : GW가 도메인 내부의 DR(또는 GW)들에게 도메인 외부에 그룹 멤버가 없으므로 GW쪽으로는 데이터그램 전송을 중지하라는 holding 상태를 요구하는 메시지
- Holding Release Request(Holding_Release_Req) : GW가 도메인 내부의 DR(또는 GW)들에게 도메인 외부에 그룹 멤버가 있으므로 holding 상태를 해제하고 데이터그램을 전송하도록 요구하는 메시지

3.2 HXcast++의 동작 절차

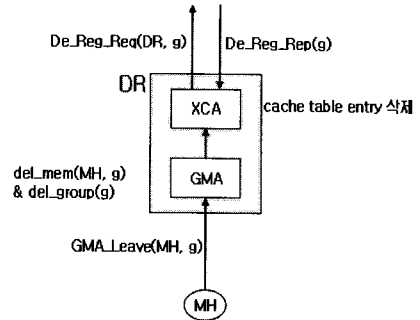
본 절에서는 GMA 메커니즘, 그룹 가입/탈퇴, 멀티캐스트 데이터의 전송 및 핸드오프 절차를 설명한다.

3.2.1 GMA 메커니즘

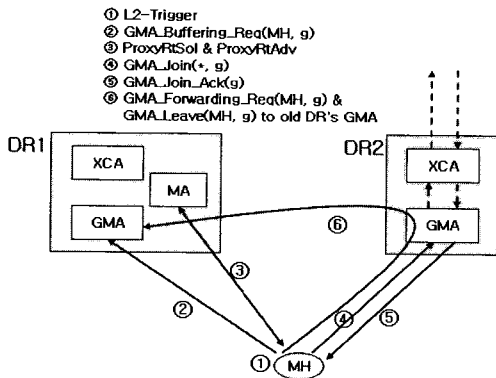
그림 3(a)는 GMA Join 절차를 보여주고 있다. MH가 그룹에 가입하고자 할 경우, 멀티캐스트 응용은 GMA_Join(*,g)메시지를 DR에게 전송한다. GMA에서 사용하는 메시지 포맷은 그림 4와 같다.



(a) GMA_Join 절차

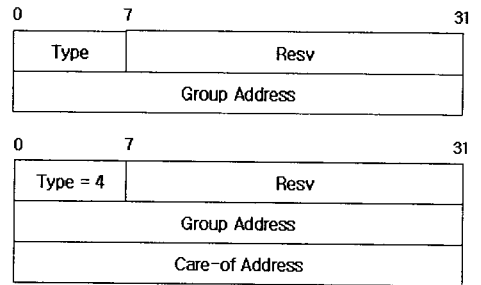


(b) GMA_Leave 절차



(c) GMA handoff 절차

〈그림 3〉 GMA 메커니즘



- Type :
- 0 = GMA_Join
 - 1 = GMA_Join_Ack
 - 2 = GMA_Leave
 - 3 = GMA_Buffering_Request
 - 4 = GMA_Forwarding_Request

〈그림 4〉 GMA 메시지 포맷

MH에 있는 GMA 클라이언트는 DR주소를 목적지로 하여 메시지를 전달한다. 메시지 전송에 IGMP와 달리 유니캐스트 주소를 사용하는 것은 핸드오프하여 다른 네트워크로 이동하더라도 주기적인 query를 기다리지 않고 가입하며, 이전 FA로부터 자유롭게 탈퇴하기 위해서이다.

GMA_Join 메시지를 받은 GMA는 Xcast Control Agent(XCA)에게 Xcast 그룹 가입을 요구하고 표 1과 같이 멤버십 테이블을 유지/관리한다. Group과 Interface 항목은 기존 IGMP에서 관리하던 정보이며 Member List와 Status는 본 논문에서 추가한 항목이다. Status에 설정된 Active는 멀티캐스트 포워딩 상태를 말하며, Buffering은 멀티캐스트 데이터그램을 포워딩하지 않고 호스트별 버퍼에 저장중인 상태를 나타낸다. XCA가

Xcast 그룹에 가입되었다는 응답을 받으면 이를 GMA에게 알리고, GMA는 GMA_Join_Ack를 MH에게 전송한다.

〈표 1〉 GMA 그룹 멤버십 테이블

Group	Interface	Member List	Status
G1	if2	MH1_ID	Active
	if3	MH2_ID	Buffering

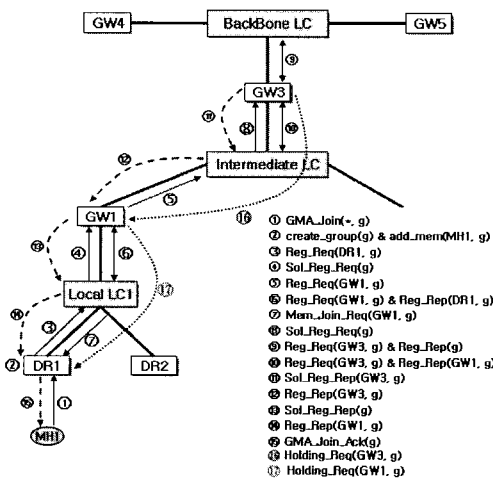
MH가 그룹으로부터 탈퇴하는 절차는 그림 3(b)에서 보여준다. MH가 GMA_Leave 메시지를 보내면 GMA는 그룹 멤버십 테이블에서 그 엔트리를 삭제한다. 마지막 멤버이면 이를 XCA에게 알리고 XCA는 Xcast 그룹으로부터 탈퇴한다.

MH가 DR1영역에서 DR2로 핸드오프하는 과

정을 그림 3(c)에서 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 Buffering 상태로 설정하고, DR이 X2M을 수행할 때 참조할 수 있게 한다. 그런 다음 MH는 빠른 핸드오프를 위해 CoA를 요구하고, 이 과정에서 MH는 옮겨갈 DR2의 GMA주소를 함께 획득한다(③). MH는 GMA_Join 메시지를 DR2에게 보내어 가입한다(④). DR2의 GMA는 기존 멤버가 있을 경우, XCA에게 보고하지 않고 GMA_Join_Ack로 응답한다(⑤). MH는 이전 DR2에게 GMA_Forwarding_Request 메시지와 GMA_Leave 메시지를 보내어 Buffering된 데이터그램을 즉각 전송하고 Xcast 그룹으로부터 탈퇴하도록 요구한다(⑥).

3.2.2 그룹 가입

이동 호스트(MH1)가 멀티캐스트 그룹에 가입하는 절차를 그림 5에 나타내었다.



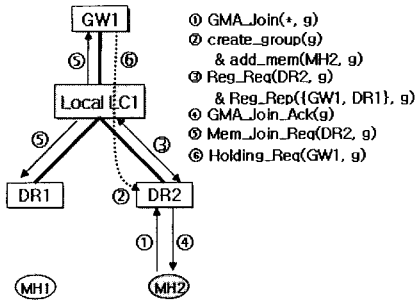
<그림 5> 이동 호스트의 그룹 가입 절차(1)

MH1이 Join메시지를 보내면(①), GMA는 이를 수신하여 그룹 엔트리를 생성하고(②), DR1의 XCA에게 알려 MH를 대신하여 그룹에 가입한다(③). Reg_Req메시지를 수신한 Local LC1은 그룹 G에 관한 아무런 정보가 없기 때문에 Xcast++와는 달리 즉각 Reg_Rep로 응답하지 않고 GW1에게 Sol_Reg_Req메시지를 보내어 외부로의 가입을 요청한다(④). 즉, GW1에게 지역 도메인 내에 있는 DR들을 대신하여 Xcast 그룹에 가입하도록 요청하는 것이다(⑤). 이때 GW1은 외부로부터 수신된 xcast 데이터를 DR들에게 전달하고, DR들로부터 수신된 xcast 데이터를 외부에 있는 수신노드들에게 전달하기 위해 Local LC1에도 가입해야 한다(⑥). 따라서 GW1의 그룹 가입 사실이 DR1에게 Mem_Join_Req 메시지를 통해 알려진다(⑦).

⑤를 수신한 Intermediate LC도 상위 네트워크와 연결된 GW3에게 Sol_Reg_Req 메시지를 전송한다(⑧). 이를 수신한 GW3은 GW1과 마찬가지로 Intermediate Domain에 속한 GW들을 대신하여 상위 네트워크에 가입한다(⑨)과 동시에 자신의 내부 네트워크에도 가입한다(⑩). 이때 BackBone LC는 즉각 Reg_Rep 메시지로 응답한다(⑪).

GW3은 그룹에 가입되었다는 상태를 기록해두고, ⑧에 대한 응답으로 Sol_Reg_Rep메시지로 응답한다(⑫). Intermediate LC 또한 가입상태를 기록해두고 ⑤에 대한 응답으로 Reg_Rep를 전송한다(⑬). ⑬,⑭메시지를 통해 Local LC1과 DR1도 응답을 받아 그룹에 가입 상태가 된다. 이때 DR1이 그룹에 가입되면, GMA는 Join_Ack로 응답한다(⑮).

만약, MH1이 송신 호스트라고 한다면, MH1에 그룹에 참여한 멤버가 없음에도 불구하고, 이를 알 수 없는 MH1은 멀티캐스트 데이터그램을 송신할 것이다. 따라서 불필요한 트래픽이 전체 네트워크로 전파되는 문제가 발생한다. 이러한 문제를 해결하기 위하여 Holding_Req메시지를 추가 정의하였다. 즉, GW3은 다른 멤버(DR 또는



〈그림 6〉 이동 호스트의 그룹 가입 절차(2)

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 전파 한다(⑩). 즉, 그룹에 가입은 하였지만, 아직 xcast 데이터는 전송하지 말라는 의미이다. GW1의 외부 DR이 holding 메시지를 받게 되면 이를 내부 DR에게 알리고, 내부 DR은 Holding_Req 메시지를 DR1에게 전달한다(⑪).

그림 6에서 보는 바와 같이, 동일한 지역 도메인내의 다른 DR에 MH2가 추가로 가입하는 경우, DR2는 그룹에 MH2를 대신하여 Reg_Req 메시지를 송신한다(③). 이때 Local LC1은 이미 가입한 GW1과 DR1의 리스트를 Reg_Rep 메시지에 포함시켜 응답한다. 이를 수신한 DR2는 가입이 이루어졌으므로 Join_Ack 메시지를 보낸다

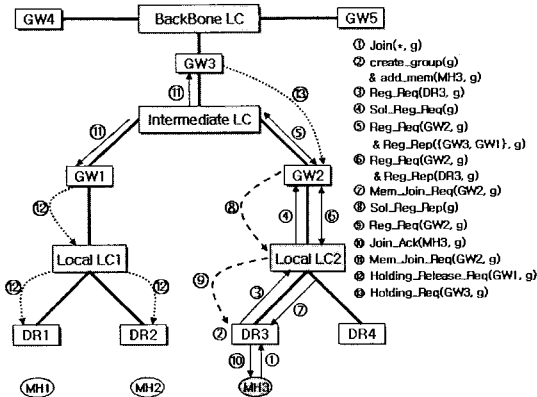
(④). 그리고 GW1은 Holding_Req 메시지를 새롭게 가입한 DR2에게만 전송한다(⑥).

다른 도메인에 있는 MH3이 가입하게 되면(그림 7), 앞에서 설명한 절차를 따라 그룹에 가입한다. GW3이 Mem_Join_Req 메시지(⑪)를 받게 되면 Holding_Req 메시지를 전송하여 xcast 데이터그램을 보내지 않도록 한다(⑬). 그러나 GW1이 Mem_Join_Req 메시지(⑪)를 받게 되면 다른 도메인에도 동일한 그룹에 참여한 멤버가 있다는 사실을 알게 되고, xcast 데이터의 전달을 위해 Holding_Release_Req 메시지(⑫)를 도메인 내 가입된 모든 DR들에게 전달한다.

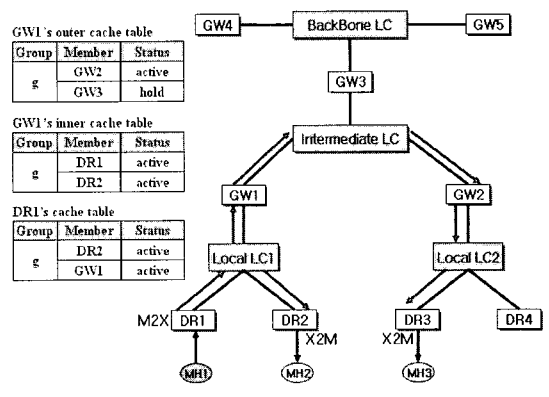
3.3.3 데이터 전송

그림 8은 MH1이 전송한 멀티캐스트 데이터그램이 그룹 멤버들에게 전달되는 과정을 보여준다.

MH1이 멀티캐스트 데이터그램을 전송하면, DR1 (즉, DR)이 이를 받아서 처리한다. GMA 그룹 멤버십 테이블을 참조하여 그룹에 속한 다른 MH들이 있다면 로컬 멀티캐스트를 이용하여 멀티캐스트 데이터그램을 전송하고, Xcast cache table을 참조하여 xcast 패킷을 전송한다. GW1이 도메인 내부로부터 xcast 데이터그램을 수신하면, 외부 DR의 cache table을 참조하여 xcast 데이터그램을 전달한다. 이때, GW3은 holding 상태에 있으므로 패킷을 생성할 때 포함시키지 않는다. 그



〈그림 7〉 이동 호스트의 그룹 가입 절차(3)



〈그림 8〉 멀티캐스트 데이터 전송 절차

러므로 GW2에게만 데이터가 전송되고, 이를 받은 GW2는 도메인 외부로부터 수신된 xcast 데이터그램이므로 내부 DR의 cache table을 참조하여 xcast 데이터그램을 도메인 내부로 전달한다.

3.3.4 도메인내 핸드오프

그림 9는 MH3이 DR3에서 DR4로 핸드오프하는 절차를 나타내고 있다. MH3은 DR4의 CoA를 획득하고 난 후, 그림 3(c)에서 설명한 핸드오프 절차와 동일하게 수행한다.

3.3.5 도메인 간 핸드오프

그림 10은 DR2의 마지막 멤버인 MH2가 동일한 멀티캐스트 그룹의 멤버인 MH3이 있는 DR3으로 도메인 간 핸드오프하는 절차를 보여주고 있다. MH2는 핸드오프 이벤트를 받아 buffering을 요구하고(②), 새로운 CoA를 획득한 후, Join 메시지를 보낸다(③). 이를 수신한 DR3은 이미 그룹에 가입이 된 상태이므로 즉시 Join_Ack 메시지를 송신한다(④).

⑥~⑪까지 앞에서 설명한 절차를 따라 DR2가 그룹 탈퇴를 수행한다. 이때 GW2는 GW1의 탈퇴로 인해 holding 상태에 있는 GW3만을 그룹 cache table에 갖게 된다. 따라서 GW2의 외부 DR은 내부 DR에게 도메인내의 그룹 멤버 DR들에게 Holding_Req 메시지를 전파시키도록 한다(⑫). 즉, GW2 바깥쪽엔 그룹에 참여한 멤버가

없으므로 GW2에게 데이터그램을 보내지 않도록 알려주는 것이다.

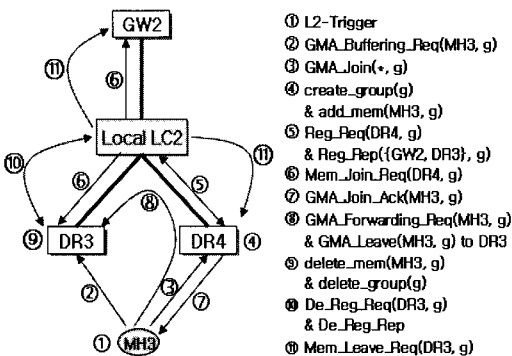
4. 성능 평가

본 논문에서 제안하는 계층적 Xcast++(HXcast++) 기법의 성능을 평가하기 위해 네트워크 시뮬레이션 툴인 NS-2[18]를 이용하였다.

4.1 실험 환경

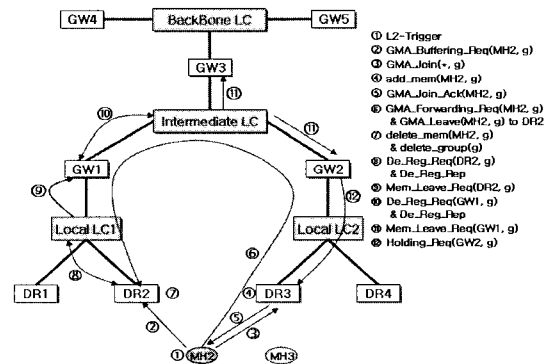
제안 시스템(HXcast++)의 패킷 헤더 부하와 패킷 처리 부하에 대하여 Xcast, Xcast++, DVMRP와 비교하였으며, 그룹 관리 부하에 대하여 Xcast++와 HXcast++를 비교하였다.

그림 11은 성능평가를 위해 GT-ITM 토폴로지 생성기[19]를 사용하여 생성한 네트워크이다. 100개의 노드와 156개의 링크로 구성되어 있으며, 100개의 노드 모두 라우터라고 가정하고 95번 노드에 송신 호스트가 있는 것으로 가정하였다. 그리고 Xcast++와 HXcast++를 위하여 Backbone LC를 3번 노드에 두었으며, 68개의 노드를 DR로 지정하였다. HXcast++를 위하여 14개의 GW 그리고 14개의 Local LC를 두어 2계층으로 구성하는 것으로 가정하였다. 그리고 그룹 멤버는 68개의 DR중 무작위로 선택하여 멀티캐스트 그룹에 가입하는 것으로 가정하였다.



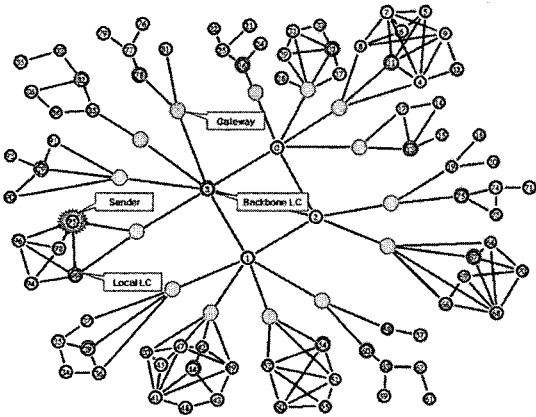
- ① L2-Trigger
- ② GMA_Buffering_Req(MH3, g)
- ③ GMA_Join(+, g)
- ④ create_group(g) & add_mem(MH3, g)
- ⑤ Reg_Req(DR4, g) & Reg_Rep((GW2, DR3), g)
- ⑥ Mem_Join_Req(DR4, g)
- ⑦ GMA_Join_Ack(MH3, g)
- ⑧ GMA_Forwarding_Req(MH3, g) & GMA_Leave(MH3, g) to DR3
- ⑨ delete_mem(MH3, g) & delete_group(g)
- ⑩ De_Reg_Req(DR3, g) & De_Reg_Rep
- ⑪ Mem_Leave_Req(DR3, g)

<그림 9> 도메인 내 핸드오프 절차



- ① L2-Trigger
- ② GMA_Buffering_Req(MH2, g)
- ③ GMA_Join(+, g)
- ④ add_mem(MH2, g)
- ⑤ GMA_Join_Ack(MH2, g)
- ⑥ GMA_Forwarding_Req(MH2, g) & GMA_Leave(MH2, g) to DR2
- ⑦ delete_mem(MH2, g) & delete_group(g)
- ⑧ De_Reg_Req(DR2, g) & De_Reg_Rep
- ⑨ Mem_Leave_Req(DR2, g)
- ⑩ De_Reg_Req(GW1, g) & De_Reg_Rep
- ⑪ Mem_Leave_Req(GW1, g)
- ⑫ Holding_Req(GW2, g)

<그림 10> 도메인 간 핸드오프 절차

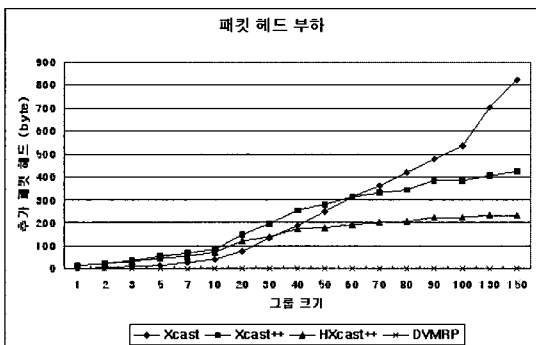


〈그림 11〉 실험 네트워크 구성

4.2 패킷 헤더 부하

패킷 헤더 부하는 그룹 크기에 따라 추가적인 패킷 헤더가 얼마나 필요한지를 그림 12에서 보여주고 있다. DVMRP의 경우, 송수신 주소 외에 추가적인 헤더는 필요 없으므로 그래프에서 가장 아래쪽에 나타나고 있다.

Xcast는 수신 노드의 수만큼의 추가적인 헤더가 필요하므로 수신 노드수가 증가할수록 패킷 헤더 부하가 증가함을 알 수 있다. 그러나 그룹의 규모가 작은 경우의 패킷 헤더 부하는 Xcast++나 HXcast++보다 낮았다. 즉, Xcast++나 HXcast++는 멤버의 주소뿐 아니라 그룹 주소를 헤더에 포함하고 있기 때문이다.



〈그림 12〉 패킷 헤더 부하

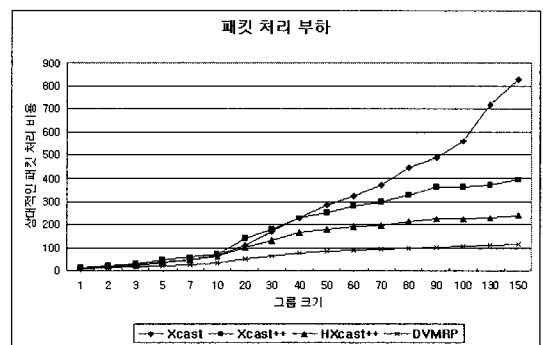
Xcast++의 경우, DR이 자신의 하위 네트워크에 있는 멤버들을 대신하여 그룹에 가입하므로 DR 영역에 수신 노드의 밀집도가 높아지기 시작하는 시점(60~70)부터 패킷 헤더 부하의 증가 추세가 감소하였다.

HXcast++인 경우, DR이 하위 네트워크 멤버를 대신해서 가입하고, GW가 DR들을 대신하여 외부 네트워크에 가입하는 개념이므로, GW 영역의 밀집도가 높아지기 시작하는 시점(40~45)부터 패킷 헤더 부하의 증가 추세가 감소함을 볼 수 있다. 이러한 현상은 계층적 구조로 인한 효과라고 볼 수 있다.

4.3 패킷 처리 부하

패킷 처리 부하를 계산하기 위해 [20,21]에서 사용한 방법과 같이, 유니캐스트 패킷을 처리하는 비용을 1이라고 가정하고, 1중 라우팅 테이블을 참조하는 비용을 0.8, 나머지 0.2를 패킷을 포워딩하는 비용이라고 가정하였다.

Xcast류의 경우 패킷 처리는 패킷 헤더에 포함되는 수신 노드의 수에 비례하여 증가하므로, Xcast++와 HXcast++는 각 DR 영역 및 GW 영역내의 수신 노드 밀집도가 증가하는 시점부터 증가 추세가 둔화되는 것을 볼 수 있다. 그리고 HXcast++의 경우, Xcast++보다 효율적임을 알 수 있다 (그림 13).



〈그림 13〉 패킷 처리 부하

4.4 그룹 관리 메시지 부하

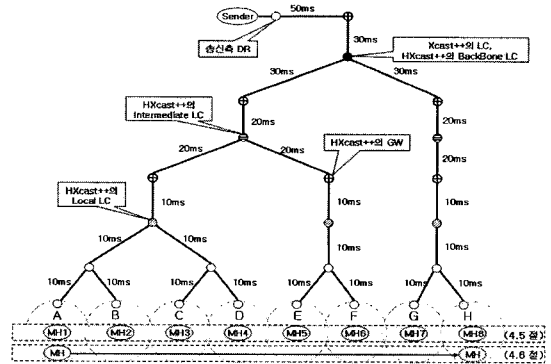
멀티캐스트 그룹 멤버십 관리를 위한 제어 패킷의 양을 측정하였다. 그림 14는 그룹에 참여하는 DR의 수를 증가시키면서 필요한 제어 패킷의 양을 그래프로 나타낸 것이다. 즉, DR이 그룹에 가입하면 LC를 통해 기존 멤버의 리스트를 응답 받게 되고, 가입한 DR의 정보가 기존 모든 멤버들에게 알려져야 한다.

HXcast++의 경우, DR의 수가 증가함에 따라 Xcast++보다 느린 증가 추세를 보이고 있다. 이는 계층적 구조로 인하여 지역 도메인의 그룹 멤버십 정보가 첫 번째 가입 시 외에는 GW를 넘어 가지 않기 때문이다. 따라서 계층적 구조를 다계층으로 나눌수록 효과가 있을 것으로 사료된다.

4.5 그룹 가입 지연

멀티캐스트 그룹 가입 지연과 핸드오프 시 패킷 손실을 측정하기 위해 그림 15의 토폴로지를 가정하였다. 하나의 멀티캐스트 송신 호스트가 있고, 각 노드 간 지연은 임의대로 설정하였다.

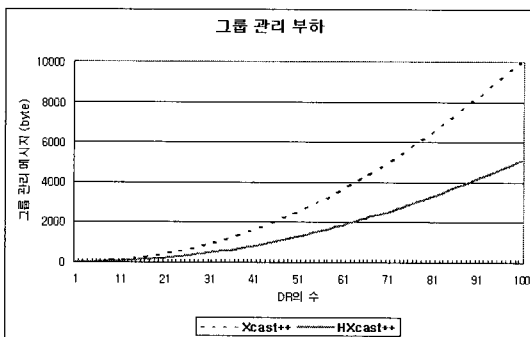
멀티캐스트 그룹 가입 지연을 측정하기 위해 각 이동 호스트(MH1~MH8)가 A cell로부터 H cell에 까지 순차적으로 가입하는 것을 가정하였으며, 이동 호스트가 Join 메시지를 송신한 시간으로부터 멀티캐스트 데이터그램을 수신하는데 까지 걸리는



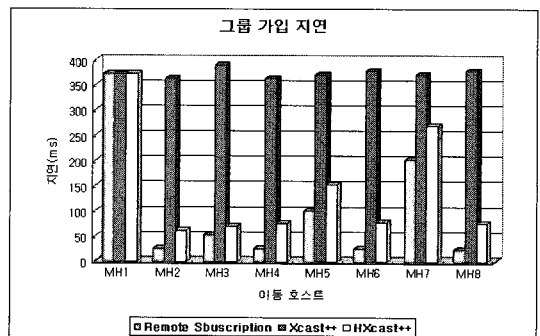
〈그림 15〉 실험 네트워크 구성

시간을 각각 측정하여 각 방식(RS, Xcast++, HXcast++)을 상호 비교하였다(그림 16).

A cell에서 MH1이 가입하였을 경우에 3가지 방법 모두 동일하게 나타났다. 이것은 송신자에게 수신 노드들의 가입 사실이 알려지고 난 후 데이터그램이 송신되기 때문이다. 그러나 B cell에서 MH2가 가입하는 경우, A cell에서 멀티캐스트 데이터그램을 수신하고 있는 호스트가 있기 때문에, RS의 경우엔 멀티캐스트 데이터그램이 전송되고 있는 교차(crossover) 지점까지 graft 메시지가 전달되면 곧바로 멀티캐스트 데이터그램이 전송되므로 상대적으로 짧은 가입 지연이 발생하는 것을 알 수 있다. 그러나 이를 위해서는 주기적인 prune 메시지가 전송되어야 하므로 트리를 유지하기 위한 부담이 크다.



〈그림 14〉 그룹 관리 부하



〈그림 16〉 그룹 가입 지연

MH7이 G cell에서 가입하는 경우를 보면, RS 방식에서는 멀티캐스트가 전송되고 있는 교차지점(그림에서 BackBone LC)까지 graft 메시지가 전송되어야 데이터를 전달 받을 수 있으며, HXcast++에서는 송신 측 게이트웨이에게 알려져야 하므로 RS보다는 가입 지연이 길게 나타났다. 그러나 Xcast++의 경우엔 항상 송신 측 DR까지 수신 측 DR의 가입 사실이 알려져야 멀티캐스트 데이터를 받을 수 있으므로 가장 긴 지연이 생기는 것을 알 수 있다.

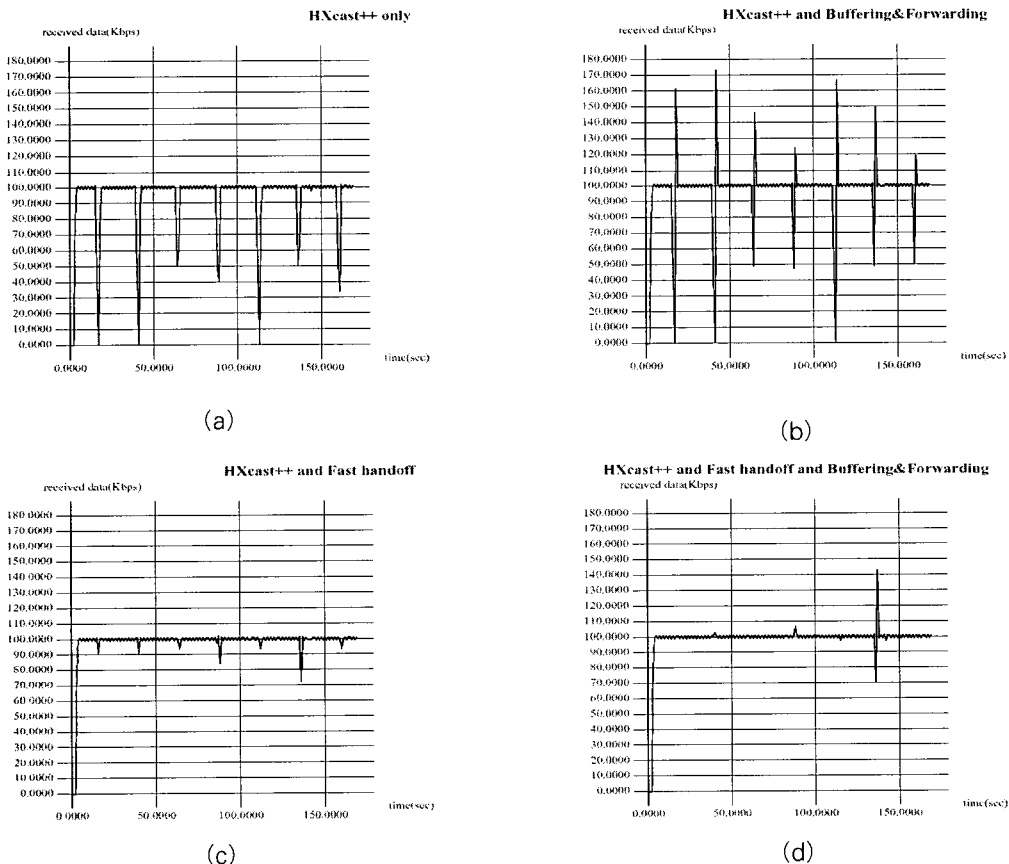
4.6 수신 호스트의 핸드오프에 따른 패킷 손실

수신 호스트의 핸드오프 시 패킷 손실을 측정하기 위해 그림 15의 토폴로지를 가정하였다. DR들의

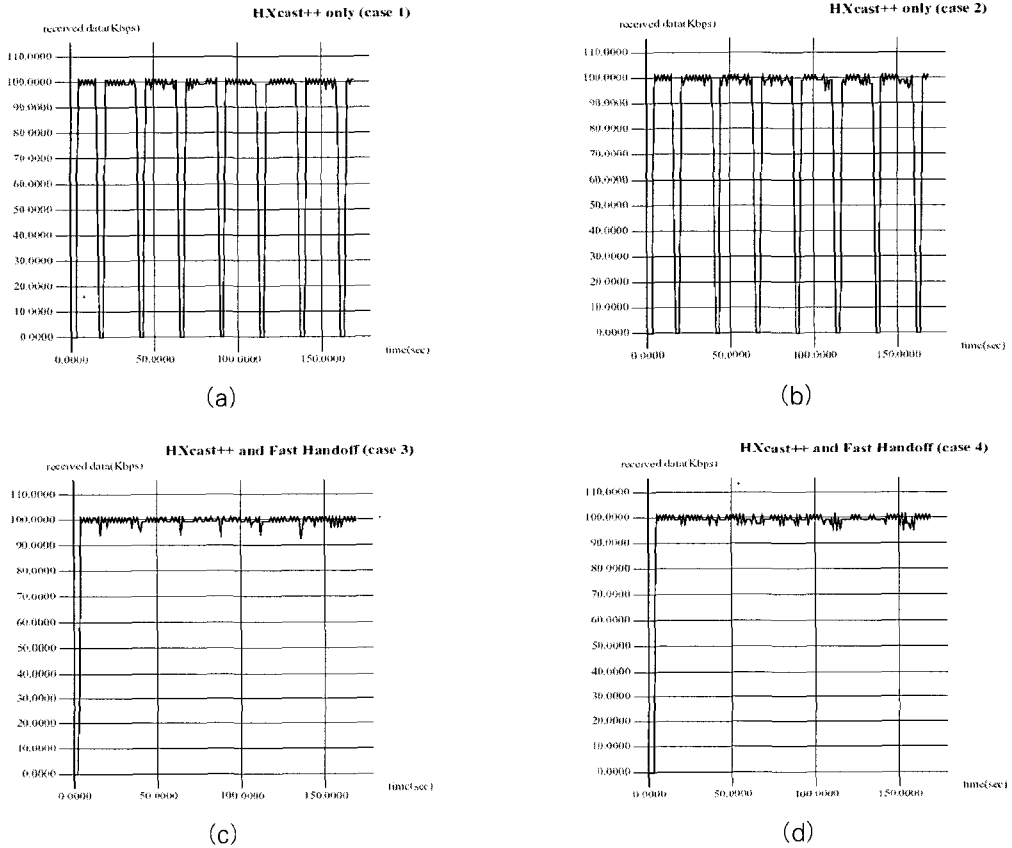
에이전트 광고주기는 3초로 하였으며, 이동 호스트의 이동 속도는 20m/sec로 하였다. 그리고 송신 노드는 100Kbps의 CBR(Constant Bit Rate) 트래픽을 전송하는 것으로 하였으며 백그라운드 트래픽은 발생시키지 않았다. HXcast++ 기법을 이용하여 하나의 이동 호스트(MH)가 cell A에서 시작하여 cell H까지 이동하면서 수신한 패킷의 양을 측정하였다.

HXcast++만 사용한 경우와 패킷 손실을 줄이기 위한 각 방식(fast handoff, buffering& forwarding)을 적용한 결과를 그림 17에 나타내었다.

그림 17(a)는 HXcast++만 사용한 경우이다. 핸드오프 구간마다 패킷 손실이 발생하는 것을 볼 수 있다. 에이전트 광고 주기가 길면 길수록 패킷 손실은 많이 발생하게 된다. 그림 17(b)는 L2



〈그림 17〉 HXcast++를 사용하는 경우 핸드오프 방법에 따른 수신 호스트의 패킷 손실



〈그림 18〉 송신 호스트의 핸드오프에 따른 패킷 손실

Mobile Trigger를 이용하여 핸드오프 직전에 DR에게 버퍼링을 요구하고, 에이전트 광고를 받아 정상적으로 Join_Ack를 수신한 다음, 포워딩을 요구한 경우의 그래프이다. 이 그래프에서는 손실되는 패킷보다 적은 양의 데이터를 포워딩 받는 것을 알 수 있다. 이것은 광고 주기가 길어서 많은 패킷을 버퍼링하여 포워딩하였지만 수신 노드의 버퍼 크기의 제한으로 수신 도중에 손실된 것이다. 따라서 buffering&forwarding 방법만으로는 이를 해결할 수 없음을 알 수 있다.

그림 17(c)는 L2 Mobile Trigger를 이용하여 fast handoff[6]를 수행한 결과이다. 빠른 핸드오프로 인하여 패킷 손실이 상대적으로 적기는 하나, 핸드오프 이후 멀티캐스트 트리가 생성되는 동안은 데이터를 수신할 수 없음을 알 수 있다.

그림 17(d)는 fast handoff와 buffering&forwarding을 함께 사용한 경우이다. 도메인 간 핸드오프 시 다소 많은 패킷의 buffering과 forwarding이 생겼지만, 패킷의 손실은 전혀 발생하지 않았다. 그러나 적은 양의 중복 패킷이 발생하였다.

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

4.7 송신 호스트의 핸드오프에 따른 패킷 손실

송신 호스트의 핸드오프로 인한 패킷 손실을 측정하기 위해 4.6절(그림 15)에서의 가정과 동일

하게 하여 아래의 4가지 경우에 대하여 측정하였다. 송신 호스트의 이동에 대해서는 버퍼링과 포워딩을 적용하기 곤란하므로 고려하지 않았다.

case 1 : HXcast++만 사용 함. 송신 호스트는 cell A로부터 cell H까지 멀티캐스트 데이터그램을 전송하면서 이동, 수신 호스트는 그림 15의 sender 위치에 있는 것으로 가정.

case 2 : case 1과 동일한 가정, 그리고 각 cell(A~H)에 송신 호스트 외에 같은 그룹의 수신 호스트들이 이미 서비스를 받고 있는 것으로 가정.

case 3 : case 1과 동일한 가정, 그리고 Fast Handoff 메커니즘을 적용.

case 4 : case 2와 동일한 가정, 그리고 Fast Handoff 메커니즘을 적용.

case 1(그림 18(a))의 경우, 송신 호스트는 이동한 DR로부터 새로운 CoA를 획득해야만 그룹에 가입할 수 있고, 멀티캐스트 데이터그램을 전송할 수 있다. 따라서 그에 상응하는 패킷 손실이 발생함을 알 수 있다.

case 2(그림 18(b))의 경우는 송신 호스트가 이동한 DR에 이미 같은 그룹의 멤버가 그룹에 가입하여 서비스를 받고 있지만, case 1과 마찬가지로 새로운 CoA를 획득하는데 걸리는 시간만큼의 패킷 손실은 줄일 수 없다.

case 3(그림 18(c))의 경우, L2 Mobile Trigger를 이용하여 빠른 핸드오프를 하므로 CoA를 획득하는데 걸리는 시간을 획기적으로 줄일 수 있다. 그러나 핸드오프 이후 이동한 DR에서 다시 가입하므로 가입 지연에 해당하는 기간만큼 패킷 손실이 발생함을 알 수 있다.

case 4(그림 18(d))의 경우는, 기존의 그룹 멤버가 서비스를 받고 있었으므로 핸드오프 이후 이동한 DR로 하여금 그룹에 재 가입하므로 인한

가입 지연을 줄일 수 있다. 따라서 소량의 패킷 손실만 발생하는 것을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 이동 통신망 환경에서 이동 호스트들에게 멀티캐스트 서비스를 제공하기 위한 방안으로 기존의 Xcast++를 계층적 구조로 확장한 HXcast++를 제안하였다.

이동 호스트들을 대신하여 DR이 그룹에 가입하도록 함으로써 이동 호스트의 위치와 상관없이 최적화된 경로로 서비스를 받을 수 있도록 하였으며, 계층적 개념을 도입하여 빈번한 핸드오프로 인한 유지비용을 줄일 수 있도록 하였다. 또한 이동망 환경에 적합한 그룹 관리를 위하여 GMA 메커니즘을 제안하여 그룹 가입 지연을 줄이고, 탈퇴 지연으로 인한 자원의 낭비를 제거하였다.

핸드오프 시 발생하는 다량의 패킷 손실을 줄이기 위하여 본 논문에서 L2 Mobile Trigger를 이용하는 fast handoff 기법을 사용하였으며, 멀티캐스트 트리가 생성되는 동안 발생할 수 있는 패킷 손실은 buffering&forwarding 기법을 이용하여 줄였다.

성능평가를 통해 HXcast++는 기존 Xcast++보다 그룹 관리 및 멀티캐스트 패킷 전달을 위한 추가적인 부하가 모든 면에서 개선된 것을 알 수 있었다. 또한 그룹 멤버의 밀집도가 높을수록 효율적이었다. 그러나 기존 멀티캐스트와 비교하여 멀티캐스트 트리 재구성 부담이 없는 반면, HXcast++역시 패킷 헤더 부하나 패킷 처리 측면에서 여전히 부하가 높게 나타남을 알 수 있었다.

향후 연구로는, HXcast++에 QoS를 제공하기 위한 방안에 관한 연구가 필요할 것이다. 이동 통신망 환경에서 QoS를 제공하기 위해서는 QoS 재설정 지연이 발생한다. 따라서 QoS를 보장하면서도 세션 재설정 지연을 줄이기 위한 방안에 대한 연구가 필요하다.

참고 문헌

- [1] C. Perkins, "IP Mobility Support for IPv4," RFC 3344, August 2002.
- [2] D. Johnson, et al., "Mobility Support in IPv6," IETF Internet-Draft, draft-ietf-mobileip-ipv6-21.txt, February 2003.
- [3] Jiang Wu, "An Mobility Support Agent Architecture for Seamless IP Handover," Licentiate Thesis, Royal Institute of Technology (KTH), Sweden, June 2000.
- [4] Byung-Soon Kim, Ki-Jun Han, "A Multicast handoff agent scheme for micro-mobility in all-IP wireless network," IEE Electronics Letters, vol 38. No. 6, pp.596-597, 2002.
- [5] P. Reinbold and O. Bonaventure, "A survey of IP micro-mobility protocols", Infonet Technical Report 2002-06, March 2002.
- [6] K. Malki, et al., "Low latency Handoffs in Mobile IPv4," IETF Internet-Draft, draft-ietf-mobileip-lowlatency-handoffs-v4-04.txt, June 2002.
- [7] P. R. Calhoun, et al., "Foreign Agent Assisted Hand-off," IETF Internet-Draft, draft-calhoun-mobileip-proactive-fa-03.txt, November 2000.
- [8] Rajeev Koodli, "Fast Handovers for Mobile IPv6," IETF Internet-Draft, draft-ietf-mobileip-fast-mip6-08.txt, October 2003.
- [9] Myung-Ki Shin, et al., "Explicit Multicast Extension (Xcast+) Supporting Receiver Initiated Join," IETF Internet-Draft, draft-shin-xcast-receiver-join-01.txt, March 2002.
- [10] R. Boivie, et al., "Explicit Multicast (Xcast) Basic Specification," IETF Internet-Draft, draft-ooms-xcast-basic-spec-04.txt, January, 2003
- [11] Ki-II Kim, et al., "Xcast+ Extension for Few-to-Few Multicast Communication," IETF Internet-Draft, draft-kim-xcast+-few-2-few-00.txt, October 2002.
- [12] R. Wittmann, M. Zitterbart, "Multicast Communication," Morgan Kaufmann Publishers, 2001.
- [13] W. Fenner, "Internet Group Management Protocol, Version 2," RFC2236, November 1997.
- [14] B. Cain, et al., "Internet Group Management Protocol, Version 3," RFC3376, October 2002.
- [15] R. Caceres, W.N. Padmanabhan, "Fast and Scalable Handoffs for Wireless Internetworks," Proc. ACM Mobicom, 1996.
- [16] A. Valko, et al., "Cellular IP," IETF Internet-Draft, draft-valko-cellularip-00.txt, November 1998.
- [17] R. Hsieh, et al., "S-MIP:A Seamless Handoff Architecture for Mobile IP," In Proceedings of INFOCOM, San Francisco, USA, 2003.
- [18] NS Simulator, Version 2.26, available from <http://www.isi.edu/nsnam/ns/>
- [19] Calvert, K., Zegura, E., "GT-ITM: Georgia Tech Internetwork Topology Models," available from <http://www.cc.gatec.edu/projects/gtitm/>
- [20] Y. Chu, et al., "A Case for End System Multicast," IEEE Journal on Selected Areas in Communication(JSAC), Special Issue on Networking Support for Multicast, 2002.
- [21] Ki-II Kim, et al., "Native Multicast Datagram Delivery using Xcast over IPv6 Networks," APCC 2003, September 2003.

● 저 자 소개 ●



김 태 수

1995년 경상대학교 임산공학과 졸업 (이학사)
1998년 창원대학교 대학원 컴퓨터공학과 졸업 (이학석사)
2004년 창원대학교 대학원 컴퓨터공학과 박사 (공학박사)
2004년~현재 창원대학교 컴퓨터공학과 IT 초빙 교수
관심분야 : 멀티캐스트 프로토콜, 이동 통신망, QoS 관리, 센서 네트워크
E-mail : tskim@changwon.ac.kr



이 광 휘

1983년 고려대학교 전자공학과 졸업 (공학사)
1985년 고려대학교 대학원 전자공학과 졸업 (공학석사)
1989년 고려대학교 대학원 전자공학과 졸업 (공학박사)
1988년~현재 창원대학교 컴퓨터공학과 교수
1991년~1992년 영국 Walse대학(Swansea) 및 Newbridge Networks사 연구원
1994년~1995년 영국 런던대학(UCL) 연구원
1997년~1999년 영국 Reading대학 연구원
관심분야 : 망관리시스템, 멀티캐스트 프로토콜, 이동 통신망, QoS 관리, 분산시스템,
QoS 라우팅, 센서 네트워크
E-mail : khlee@changwon.ac.kr