

가상 환경에서의 다중 그룹통신을 위한 효율적인 그룹관리 설계 및 구현

김산^{0*} 윤미연^{*} 신용태^{*} 곽재창^{**}
*송실대학교 컴퓨터학과
**서경대학교 컴퓨터학과
sanmt@cherry.ssu.ac.kr

Design and Implementation of effective group management for multi-group communication in a virtual environments

San Kim^{0*} Miyouon Yoon^{*} Yongtae Shin^{*} JaeChang Kwak^{**}
*Dept. of Computer Science, Soongsil University
**Dept. of Computer Science, Seokyeong University

요 약

가상 환경에서의 다중 그룹통신이란 인터넷, 회사의 인트라넷과 같은 네트워크와 컴퓨터를 이용한 가상 환경에서 어떤 공통의 목적을 가진 멤버들을 그룹화하여 특정서비스를 제공할 수 있도록 하는 것이다. 그러나 일반적인 유니캐스트나 브로드캐스트 기술을 그룹관리에 적용하기엔 여러 문제점이 있다. 본 논문에서는 멀티캐스트 개념을 적용한 체계적인 그룹관리의 설계를 통해 멤버 및 그룹간의 관계를 정의하고, 여러 가상환경에 적합하고 효율적인 그룹관리 방안을 제시하며, 또한 API 설계를 통해 멤버 및 그룹관리 서비스를 구현한다.

1. 서 론

최근 기본적인 Best-effort 서비스를 비롯해 회사의 웹서비스뿐만 아니라 나아가 Internet telephony와 Video conference에 이르기까지 여러 서비스를 제공하기 위한 많은 연구들이 진행 중에 있다. 여러 서비스를 제공하고자 할 때, 특정 서비스를 원하는 사용자들은 한 그룹으로써 관리되어야만 한다. 또한 회사나 군대와 같은 어떤 계층적인 구조가 필요한 단체에서는 다자간 통신의 기반 모델이 되는 그룹통신의 그룹멤버십 서비스가 지원되어야 한다. 따라서 본 논문에서는 멀티캐스트 개념을 적용하여, 멤버들과 멤버들로 구성된 그룹간의 관계를 정의하고 API 설계를 통해 멤버관리, 그룹관리를 지원하는 인프라에서의 어플리케이션 멀티캐스트 서비스를 제공하고자 한다. 2장에서는 본 연구와 구현의 기반이 되는 멀티캐스트와 IGMP에 대해 살펴본다. 3장에서는 구체적인 그룹관리 방안과 그룹관리 API를 제시한다. 4장에서는 구현된 서비스 결과를 보여주고, 5장에서는 향후 연구 방향을 제시한다.

2. 관련연구

2.1 그룹통신의 개요

그룹통신은 크게 전송 서비스와 멤버십 서비스로 나눌 수 있다. 전송 서비스는 객체간의 통신 시 신뢰적인 전

송 등을 보장하도록 제공하는 서비스이며, 멤버십 서비스는 그룹에 참여하는 객체의 동적인 가입 및 탈퇴를 보장하고 네트워크의 오류 및 객체의 오류로 인한 서비스 중단 및 복구를 가능하도록 제공하는 서비스로써 동적인 그룹관리를 보장하는 것이다. 요약하면, 그룹통신은 그룹간의 통신을 네트워크 상으로 지원하는 것이다.

2.2 전송방식

전송방식에는 크게 유니캐스트, 브로드캐스트, 멀티캐스트[1]가 있으며, 최근에 새롭게 정의된 애니캐스트도 있다. 멀티캐스트는 송신자가 해당 데이터를 받기를 원하는 수신자들(멀티캐스트 그룹)에게만 데이터를 전달하는 방식이다. 이는 일-대-다 또는 다-대-다 전송을 지원함으로써 유니캐스트와 브로드캐스트의 단점을 보완한 것이다. 송신지 측은 하나의 데이터 패킷을 보냄으로써 네트워크의 효율성을 높이고 네트워크의 혼잡상황을 줄일 수 있고, 수신자 측은 원하는 호스트에게만 데이터를 복사해줌으로써 불필요한 데이터 수신을 방지할 수 있다.

인터넷 멀티캐스트 기술은 멀티캐스트 응용 서비스 자체와 TCP와 상응하는 수송계층에서의 멀티캐스트 신뢰성 제공기술, 그리고 데이터 패킷 전송경로를 설정하는 메커니즘인 네트워크 계층에서의 멀티캐스트 라우팅 기

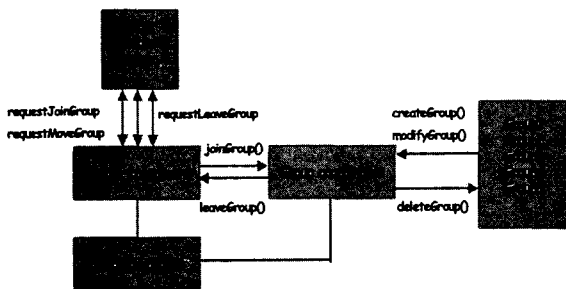
술로 분류할 수 있다. 멀티캐스트 기술은 아직 만족할만한 그리고 대규모 인터넷에 쉽게 적용할 만한 해법이 제시되지 못하고 있다. 수송계층에서의 신뢰성 제공기술인 RMT(Reliable Multicast Transport)등 여러 멀티캐스트 기술들이 계속 연구 중에 있다[2].

2.3 IGMP

멀티캐스트는 IGMP(Internet Group Management Protocol)을 이용하여 그룹관리를 한다[3][4]. IGMP 프로토콜은 그룹에 대한 가입/탈퇴 처리를 담당한다. IGMP는 멀티캐스트 라우터가 해당 서브넷에 존재하는 호스트 그룹 멤버의 존재를 인식하는데 사용된다. 기본적으로 query와 report메시지를 사용한다. IGMP가 해당 서브넷에 query메시지를 보내면 해당 멀티캐스트 그룹에 가입을 원하는 호스트만이 report메시지를 보냄으로써 그룹에 대한 가입할 수 있다. 그룹의 탈퇴는 일정시간 동안 query메시지에 대한 응답을 하지 않으면 이루어진다. 이는 기본적인 IGMP[4] 프로토콜로서 문제점을 보완하기 위해 IGMP 버전3[5]까지 나왔다.

3. 그룹관리의 설계

그룹관리의 전체적인 상태는 [그림 1]과 같다. 멤버는 멤버 코디네이터에 의해 관리되며, 그룹 또한 그룹 코디네이터에 의해 관리된다. 클라이언트의 요청에 따라 서버가 동작하는 방식을 채택하고 있다.



[그림 1] 그룹관리의 상태 다이어그램

3.1 그룹관계의 정의

그룹관리를 위한 DB를 두어 멤버와 그룹의 정보를 관리하도록 하고 멤버의 정보를 담고 있는 'Member' 테이블과 'Group' 테이블을 구성하도록 한다. 또 구체적인 API를 제공하는 'CMemberCtrl' 클래스와 'CGroupCtrl' 클래스가 존재한다.

그룹관계 정의를 위하여 'parent' 필드와 'ex_flag', 'en_flag'를 둔다. 이는 수직관계와 수평관계, 외부관계와

적대관계를 정의한다. 수직관계는 그룹의 상하관계를 표시하며 수평관계는 자신과 평등한 관계를 보여준다. 수직관계와 수평관계를 나타내기 위하여 DB에 'parent'라는 부모를 나타내는 필드를 정의해준다. 'ex_flag'는 외부관계가 개방형 그룹인지, 폐쇄형 그룹인지를 나타낸다. 개방형 그룹은 어떤 그룹의 멤버가 아닐지라도 그 그룹에게 메시지를 보낼 수 있으며, 폐쇄형 그룹은 이와는 반대로, 그룹의 멤버만이 그룹에 메시지를 보낼 수 있다. 'en_flag'는 적대관계를 나타내며, 이는 필요시 사용한다. 수직관계와 수평관계 외부관계와 적대관계를 나타내기 위하여 트리 형태로 전체적인 구조를 표현한다.

동적인 그룹관리 위해 코디네이터에게 그룹의 생성, 삭제, 수정과 멤버의 생성, 삭제, 수정이 가능하도록 API를 제공하고, 사용자에게 자신의 정보를 수정하고 원하는 그룹에 가입과 탈퇴가 가능하도록 API를 제공한다.

3.2 그룹트리의 구성

효율적인 그룹관리를 위해 그룹들간의 관계정의를 계층적인 구조를 나타낼 수 있는 트리구조 형태로 구성한다. 이를 위해 모든 그룹들은 하나의 그룹리스트로 서버의 DB에 저장되며, 이에 속한 모든 멤버들 또한 하나의 멤버리스트로 저장된다. 그룹리스트의 각 그룹정보에는 그룹들간의 관계를 정의하기 위해서 'Parent Group' 필드가 존재해야 하고, 멤버리스트의 각 멤버정보에는 멤버와 그룹간의 관계를 정의하기 위해서 'EngagedGroup' 필드가 존재해야 한다. 그룹트리는 여러 환경에 맞게 최상위그룹이 여러 개가 설정될 수 있으며, 한 그룹이 다른 하위그룹이 포함하면 같은 계층에서도 먼저 구성된다. 코디네이터는 멤버들과 그룹들을 생성, 삭제, 수정하고, 사용자는 형성된 그룹트리 이미지에서 그룹을 선택하여 그 그룹에 특정서비스를 사용할 수 있게 된다.

GroupTree Algorithm (Coordinator or Client)

```

=====
Step 1 : Receiver  $C_i$  or  $M_j$  request GROUPLIST and MEMBERLIST to Server DB
Step 2 : Receiver  $C_i$  or  $M_j$  receive GROUPLIST and MEMBERLIST from Server DB
Step 3 :
GroupFree()
{
    Create treeImage
    while(Search the highest ranking Group){
        Insert the group in treeImage }
}
    
```

```

for(GroupList(i) size){
    Group = GroupList(i) info
    if(the highest ranking group == Group ID)
        Insert the group in treeImage
        SearchChildGroup(Group) }
}
SearchChildGroup()
{
    for(GroupList(i) size){
        subGroup = GroupList(i) info
        if(Group ID == subGroup ID)
            Insert the group in the treeImage
            SearchChildGroup(subGroup)
            Go to the former trunk of treeImage }
}
=====

```

3.3 그룹관리 API

그룹관리 API는 [표 1]과 같이 제공된다.

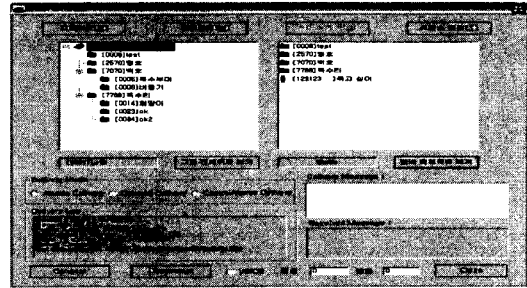
[표 1] 그룹관리 API

API	
멤버 관리	bool searchMember(CString memberID, int mode)
	void createMember(CMember* newMember)
	void modifyMember(CMember* modifyMember)
	void deleteMember(CString mID, CString mUser)
그룹 관리	bool searchGroup(CString groupID, int mode)
	void createGroup(CGroup *newGroup)
	void modifyGroup(CGroup *modifyGroup)
	void deleteGroup(CString groupID)
	void joinGroup(CMember *joinMember, CGroup *joinGroup)
	void leaveGroup(CMember *leaveMember, CGroup *leaveGroup)

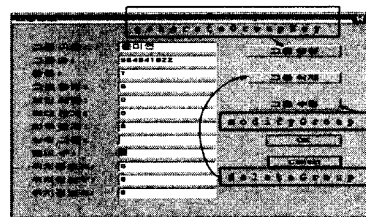
코디네이터 프로그램은 그룹관리 API와 멤버관리 API 모두를 사용하여 멤버와 그룹을 관리하게 한다. 클라이언트 프로그램은 멤버 API의 modifyMember()을 사용하여 개인정보를 수정하고 그룹관리 API의 joinGroup(), leaveGroup()을 사용하여 그룹에 가입, 탈퇴를 한다.

4. 구현결과

[그림 2]는 코디네이터 전용프로그램이고 멤버관리와 그룹관리 서비스를 제공한다. 좌측상단에 그룹트리를 구성되며, 우측 상단에 선택한 그룹에 속한 서브그룹들과 멤버들이 표시된다. [그림 3]은 그룹관리를 선택하였을 때 나타나는 창이며, 그룹관리 API를 사용할 수 있다.



[그림 2] 코디네이터 프로그램



[그림 3] 그룹관리

5. 향후 연구 방향

본 논문에서는 멀티캐스트를 기반으로 그룹관계를 정의하고 효율적인 그룹관리 API를 제공함으로써 여러 가지의 배경환경에 적용 가능하도록 구현하였다.

향후에는 적용된 그룹관리에 좀 더 많은 멀티미디어 전송 등과 같은 여러 특정서비스들이 지원 가능하도록 개발되어야 할 것이다. 또한 멀티캐스트 정보보호에 있어서 사용자 인증과 그룹 인증을 위한 보안이 요구된다.

참고문헌

[1] R. Talpade, M. Ammar, "An architecture for providing a reliable Multicast transport service" *In 15th IEEE international Conference on Distributed computing Systems*, June 1995.

[2] J. C. Lin, and S. Paul, "RMTP : A Reliable Multicast Transport Protocol", *IEEE INFOCOM '96*, March 1996.

[3] K. McCloghrie, D. Farinacci and D. Thaler, "Internet Group Management Protocol MIB", *IETF RFC 2933*, October 2000.

[4] W. Fenner "Internet Group Management Protocol, Version 2", *IETF RFC 2236*, November 1997.

[5] Brad Cain, Bill Fenner and Isidor Kouvelas, "Internet Group Management Protocol, Version 3", *IETF draft-ietf-idmr-igmp-v3-11.txt*, May 2002.