

멀티캐스트 서비스를 지원하기 위한 그룹관리 시스템의 설계 및 구현

정회원 박 판 우* 정회원 조 국 현**

Design and Implementation of Group Management System for Multicast Services

Pan Woo Park*, Kuk Hyun Cho** *Regular Members*

要 約

근래의 컴퓨터 통신망 환경에서 메시지를 다중 목적지로 전송하는 유형의 멀티캐스트 서비스가 급격히 요구되고 있다. 멀티캐스트 서비스는 미리 정의된 하나 이상의 프로세스 집단으로 구성된 그룹에게 정보를 전송하는 서비스이다. 멀티캐스트 프로세스 그룹은 전송 메시지의 목적지가 되는 일련의 프로세스 집단이며, 이러한 프로세스들은 하나 혹은 그 이상의 호스트에 분산될 수 있다. 따라서, 멀티캐스트 서비스를 효율적으로 지원하기 위해서는 무엇보다도 멀티캐스트 그룹을 구성하는 구성원 요소들의 관리가 체계적으로 이루어져야 한다. 본 논문에서는 멀티캐스트 서비스를 지원하기 위한 그룹 관리 시스템을 설계, 구현한다. 이를 위하여 프로세스 그룹 관리 시스템(PGMS), 호스트 그룹 관리 시스템(HGMS)으로 나누어 설계하고 각 각의 기능을 수행하는 프리미티브를 구현한다. 또한 그룹들의 구성요소와 상호 관계를 나타내는 멤버십 트리(Membership tree) 관리 알고리즘을 설계 구현하여 멀티캐스트 메시지 송수신시에 라우팅 정보로 이용할 수 있도록 하였다.

ABSTRACT

Recently, some applications require the multicast service that transmit messages to multiple destinations in computer communication network enviroment. Multicast service is to send messages to the group which consists of a number of processes.

A multicast group is a collection of processes which are destinations of the transmitted messages and these processes may run on one or more hosts. Therefore, it is important to manage each member of process groups in order to provide efficient multicast services.

In this paper, we design and implement group management system to support multicast services. Group Management System was designed with Process Group Management System(PGMS) and Host Group Management System(HGMS). We have implemented basic primitives of PGMS, HGMS. Also, membership tree management algorithm is designed and implemented. Membership tree provides the relation of the members of multicast groups and routing informations.

* 大邱教育大學校

** 光云大學校 電子計算學科

Dept. of Computer Science, KwangWoon Univ.

論文番號 : 93 - 111

I. 서 론

근래에 고속망 등의 출현으로 컴퓨터 통신 환경이 빠르게 변화되고 있으며, 네트워크 환경에서의 많은 응용 프로그램이 개발되어 실용화되고 있다.

예전에는 한 목적지와의 메세지 통신 서비스인 Unicast 프로토콜이 일반적이었으나, 이제는 여러 목적지에 메세지를 동시에 전송하는 유형의 서비스가 급격히 요구되고 있다^{(1),(9),(12)}.

그 중에서 미리 정의된 프로세스 그룹들 간에 메세지를 상호 주고받는 프로세스 그룹 통신에 관한 연구가 활발히 진행되고 있다. 프로세스 그룹은 분산되어 있는 통신 주체들인 프로세스 집단이며, 이러한 프로세스 그룹들 간의 통신을 특별히 멀티캐스트 서비스라고 한다. 메세지 전송프로토콜은 송수신자 사이의 데이터를 빠르고 신뢰성 있게 전송할 수 있어야 한다^{(8),(11)}. 멀티캐스트 프로토콜은 분산 시스템 환경에서 프로세스 그룹들 간의 메세지 전송을 효율적으로 실현하기 위한 프로토콜이다^{(2),(4),(6),(7)}.

멀티캐스트 프로세스 그룹은 메세지 전송의 목적지가 되는 일련의 프로세스 집단으로서, 하나의 프로세스 그룹은 네트워크 상의 한 호스트에 존재할 수도 있고 여러 호스트 상으로 분산될 수도 있다. 그리고, 특정프로세스가 동시에 한 개 이상의 프로세스 그룹의 구성원이 될 수도 있다.

따라서 멀티캐스트 서비스를 효율적으로 지원하기 위해서는 멀티캐스트 그룹을 구성하는 구성원 요소들의 관리가 체계적으로 이루어져야 한다^{(3),(10),(14)}.

즉, 프로세스 그룹을 구성하는 프로세스들 상호간의 메세지 배달, 관리, 장애 감지, 새로운 프로세스의 가입, 탈퇴, 그룹의 삭제 등이 효과적으로 지원되어야 한다. 또한 프로세스 그룹 구성 요소들이 실존하는 호스트들의 집합으로 이루어진 호스트 그룹이 잘 관리되어야 한다.

본 논문에서는 이러한 기능을 갖는 그룹 관리 시스템(GMS : Group Management System)을 PGMS (Process Group Management System)와 HGMS (Host Group Management System)를 각각 나누어 구성하였다.

한편, 복잡하게 연결된 컴퓨터 망의 효율적 관리를 위하여 ISO(International Standard Organization)에서는 OSI 관리(Open System Intereconnection Management) 표준화 작업을 진행하고 있다. OSI

관리는 5부분으로 나누어 표준화 작업이 진행되는데, 망의 장애 처리 및 회복 기능을 수행하는 장애 관리, 망의 성능을 측정 감시하는 성능 관리, 통신자원 사용에 관한 요금 계산 기능을 갖는 계정 관리, 망 구성 자원들의 속성과 특성 값을 관리하는 구성 관리, 망 접근에 관한 보안 관리 등으로 구성되어 있다⁽¹⁵⁾.

응용층의 OSI 관리 서비스를 위한 특정 관리 기능 영역 중, 객체 관리(Object Management)등의분야에서 넓은 의미의 모든 망 자원들의 변화, 속성 관리 등에 관하여 표준화 작업을 진행 중이지만, 본 논문에서의 그룹 관리는 멀티캐스트 서비스를 지원하기 위한 하위 계층의 그룹 관리로 한정한다.

본 논문의 구성은 제2장에서 멀티캐스트 통신 모델에 대하여 설명하고, 제3장에서는 그룹 관리 시스템(GMS)의 설계에 관하여 기술하며, 제4장에서 GMS의 구현에 대하여 설명하고 제5장에서 결론을 맺는다.

II. 멀티캐스트 통신 모델

컴퓨터 네트워크 환경에서의 기본적인 그룹 통신 모델은 그림 2-1과 같다⁽¹³⁾. 그림 2-1의 모델에서 사용자는 인간 또는 그룹 통신을 이용하는 응용 서비스이며 GCUA(Group Communication User Agent)는 사용자와의 인터페이스를 제공하면서 그룹 구성 요소들의 관리, 검색 등의 기능을 제공한다.

한편 GCSA(Group Communication System Agent)는 실제적인 그룹 통신을 담당하는 서비스들로 구성된 모델이다.

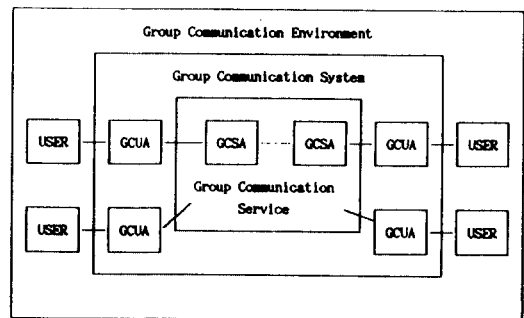


그림 2-1. 그룹 통신 모델

Fig 2-1. A model of group communication

본 논문에서는 그림 2-1과 같은 기본적인 그룹 통신 모델 위에서 그룹화된 특정 프로세스, 호스트 집단으로 패킷을 전송하는 멀티캐스트 통신 모델을 설계하고 이러한 멀티캐스트 통신 모델의 서비스를 위하여 필요한 프로세스 그룹 및 호스트 그룹 관리 시스템에 관하여 연구한다.

멀티캐스트 통신 서비스는 한 호스트에서 네트워크 상에 존재하는 하나 이상의 여러 호스트들로 구성된 특정 그룹에게 하나의 메시지를 전송할 수 있다. 이때 호스트 그룹의 임의의 한 호스트에 전송된 패킷 정보는 그 호스트 상에 존재할 수 있는 여러 목적 프로세스에게 각 각 분배된다. 즉, 한 호스트 상에서 여러 개의 목적 프로세스가 존재할 때 프로세스 레벨의 멀티캐스트 전송이 가능한 것이다.

Unicast가 한 목적지와의 통신 서비스를 제공하고 브로드캐스트(broadcast)가 모든 목적지와의 통신 서비스를 제공한다면 멀티캐스트 통신 서비스는 그룹화된 일정 목적지에게 정보를 전송하는 유용한 서비스이다.

그림 2-2에 본 논문에서 설정하는 멀티캐스트 통신 모델을 OSI 참조 모델과 비교하여 나타내었다. 그림 2-2의 멀티캐스트 모델은 호스트 그룹 계층(Host Group Layer), 전송 프로토콜 계층(Transport Protocol Layer), 프로세스 그룹 계층(Process Group Layer) 등의 3계층으로 구성된다.

그림 2-2와 같이 본 논문에서의 멀티캐스트 통신 모델은 현존하는 패킷망의 하위층 구조위에서 설계하였다. 한편, 오늘 날의 많은 통신망 형태는 중계 장치를 통한 다양한 LAN의 연결 형태로 되어 있다. 본

논문의 실험적 구현은 10 Mbps Ethernet 상에서 행하였으나 일반적 패킷 통신망인 WAN 구조에 적용하기 위하여, 통신망을 상호 연결하는 브리지; 라우터 등의 중계장치가 멀티캐스트 패킷을 인지하고 전송할 수 있는 기능을 갖고 있으며, 별도의 멀티캐스트 라우팅 프로토콜이 실행되는 것으로 가정한다.

2.1 호스트 그룹 계층

호스트 그룹 계층(Host Group Layer)은 통신 모델의 최하위의 계층으로서 분산 환경에서 하나 이상의 호스트들의 집합으로 구성된다. 가장 단순한 형태의 호스트 그룹은 그 분산 환경 네트워크 시스템의 모든 호스트를 포함하는 단 하나의 그룹만을 갖는 경우이다.

호스트 그룹 계층의 단순한 형태는 패킷 전송을 위한 브로드캐스트(broadcast) 기능을 제공하기 위한 것이지만 OSI-7 계층의 하위 3계층에 해당되는 여러 가지 기능들을 갖도록 확장할 수 있다. 호스트 그룹 계층은 네트워크 환경의 패킷 전송, 호스트 그룹 관리, 라우팅 정책 관리 등을 담당한다. 본 논문의 호스트 그룹 계층은 현존하는 LAN 형태의 하위 계층위에서 설계된다.

2.2 프로세스 그룹 계층

프로세스 그룹이란 하나 혹은 그 이상의 프로세스들로 구성된 집단이며 구성원은 서로 다른 호스트 상에 존재할 수도 있다. 프로세스 그룹은 멀티캐스트 통신 서비스시에 하나의 논리적 엔티티로서 취급되어 정보가 전송된다.

호스트 그룹 계층이 호스트 레벨의 멀티캐스트 서비스를 제공한다면 프로세스 그룹 계층은 프로세스 레벨에서의 멀티캐스트 전송을 담당한다. 따라서 프로세스 그룹 계층은 여러 개의 프로세스들로 구성된 프로세스 그룹의 생성, 삭제, 수정 등의 관리 기능을 담당하면서 프로세스간 배달, 통신 서비스를 제공한다.

2.3 전송 프로토콜 계층

전송 프로토콜 계층은 프로세스 그룹 계층에서 요구되는 다양한 멀티캐스트 전송 서비스를 담당해 주는 계층이다⁽¹²⁾. 즉, 전송 프로토콜 계층은 호스트 그룹 계층의 여러 가지 정보를 이용하여 신뢰성 있고 효율적인 엔드-투-엔드 멀티캐스트 전송 서비스를 제공해 준다.

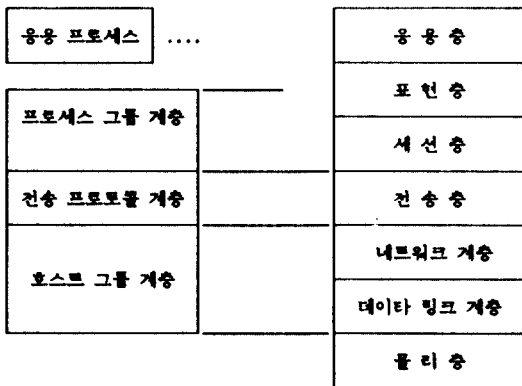


그림 2-2. 멀티캐스트 통신 모델
Fig 2-2. A model of multicast communication

Ⅲ. 그룹 관리 시스템(GMS)

제3장에서는 제2장의 멀티캐스트 통신 모델을 효율적으로 지원하기 위한 그룹 관리 시스템(GMS: Group Management System)에 관하여 기술한다.

그룹 관리 시스템은 크게 프로세스 그룹 관리 시스템(PGMS)와 호스트 그룹 관리 시스템(HGMS)로 나누어 구성한다.

3.1 PGMS(Process Group Management System) 구조

프로세스 그룹이란 하나 혹은 분산된 호스트 상에서 존재하는 하나 이상의 프로세스로 구성된 집단을 말한다. 따라서 한 프로세스 그룹은 하나의 논리적 엔티티로서 정의될 수 있으며, 이 엔티티상에서 구성된 상호간의 그룹 통신이 가능하다. 프로세스 그룹 계층은 프로세스 레벨의 통신을 담당하는 계층으로서 프로세스 상호간의 통신, 장애 감지, 변경 등을 효율적으로 처리할 수 있는 기능이 있어야 한다.

본 장에서는 프로세스 상호간에 발생하는 다양한 오퍼레이션을 수행, 관리할 수 있는 프로세스 관리 시스템(PGMS: Process Group Management System)을 설계하고 각 기능을 담당하는 프리미티브를 구현한다.

그림 3.1은 PGMS의 구조를 나타낸다.

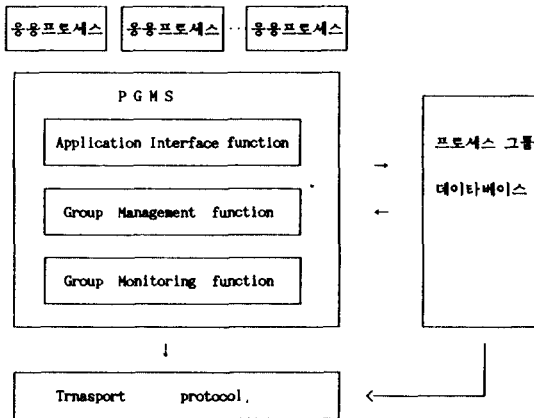


그림 3-1. PGMS 구조
Fig 3-2. PGMS structure

PGMS의 중요한 기능은 응용 프로세스와의 인터페이스를 담당하는 기능, 프로세스 그룹의 상태 변화를 관리하는 그룹 관리 기능, 프로세스 장애 발생 등을 감독하는 감독 기능 등으로 구분할 수 있다.

(1) 응용 프로세스 인터페이스

멀티캐스트 환경에서 응용 프로그램이 호출하는 프로세스 그룹 관리 오퍼레이션은 다음과 같다.

·Group_Id = Pg_create(gname)

char *gname

⇒Pg_create()는 새로운 프로세스 그룹의 생성을 실행한다.

gname은 생성하는 프로세스 그룹 이름이며 Group_Id는 프로세스 그룹의 식별자(Identifier)이다. 그룹 식별자는 프로세스 그룹 시스템에서 유일한 값을 갖는다.

·Group_Id = Pg_lookup(gname)

⇒Pg_lookup()은 그룹 이름 gname의 주소 값을 결정해 주는 프리미티브이다.

·Ret_Code = Pg_Join(Group_Id)

⇒Pg_Join() 프리미티브는 식별자 Group_Id로 주어지는 프로세스 그룹으로의 가입을 실행한다. Ret_Code 값은 프리미티브의 성공여부를 나타내며 성공시에는 프로세스 그룹의 주소 값을 갖는다.

·Ret_Code = Pg_leave(Group_Id)

⇒Pg_leave()는 특정 프로세스를 Group_Id식별자를 갖는 프로세스 그룹으로부터 탈퇴시키는 프리미티브이다.

그룹의 구성원인 임의의 한 프로세스가 정상적으로 실행을 끝마쳤거나 혹은, 장애 발생 등으로 실행 종료되었을 때, Pg_leave() 프리미티브가 호출되는데 이는 종료된 프로세스가 구성원인 모든 프로세스 그룹에서 일어난다.

·Ret_Code = Pg_delete(Group_Id)

⇒Pg_delete()는 지정된 그룹의 제거를 실행한다.
Pg_delete()프리미티브가 호출되고 나면 해당되는 Group_Id의 모든 자료는 소멸된다. 후에 다시 생성되더라도 이제 다른 그룹 식별자를 갖게된다.

·Address = Pg_getview(Group_Id)

⇒pg_getview() 프리미티브는 지정된 그룹에 관한 뷰 구조를 제공하는 함수이다. 프로세스 그룹 뷰는 그룹의 여러 가지 정보를 갖는 구조로서 뷰 번호, 그룹 이름, 구성원 수, 구성원 리스트 등을 제공한다.

한편, 한 프로세스는 동시에 여러 프로세스 그룹의 구성원이 될 수 있다. 그리고 어느 임의의 한 프로세스는 그 프로세스가 소속된 그룹으로 보내진 데이터를 모두 수신한다. 프로세스 그룹들 사이의 데이터 전송을 담당하는 송수신 오퍼레이션은 다음과 같은 Send(), receive() 프리미티브가 있다.

·Send(Group_Id, messagepoint)
·Receive(Group_Id, messagepoint)

⇒위의 두 프리미티브에서 Group_Id는 그룹 식별자 혹은 단일 프로세스 식별자가 될 수 있으며, 이 값으로 전송층에서 그룹 통신을 지원하거나 단일 통신(unicast)을 지원한다. messagepoint는 송수신 되는 메세지이다.

(2) 프로세스 그룹 관리 기능

그룹 관리 기능은 프로세스 그룹 구성원들의 가입, 탈퇴 등에 따른 관계변화를 관리하여 그룹의 프로세스 상태를 저장하고 구성원에게 통지한다. 프로세스 그룹 관리 기능 모델을 그림 3-2에 나타내었다.

컴퓨터 네트워크 환경의 각 호스트상에 존재하는 프로세스 그룹은 수시로 동적으로 변화하기 때문에 특별한 프로세스 그룹 관리 기능을 수행할 필요가 있다.

프로세스 그룹 관리 기능은 그룹의 상태나 구성원 관계를 변화시키는 모든 오퍼레이션의 영향을 받는다.

프로세스 구성원 관계, 상태 등을 변화시키는 오퍼레이션은 새로운 그룹의 생성, 구성원 가입, 탈퇴, 그

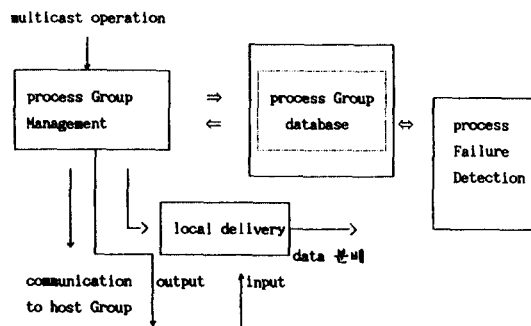


그림 3-2. 프로세스 그룹 관리 기능 모델
Fig 3-2. A functional model of process group management

룹 삭제, 장애 감지 등이 있다. 따라서 이러한 오퍼레이션 요구가 발생할 때 마다 그룹 관리 기능은 프로세스 그룹 데이터 베이스를 수정해야 한다.

프로세스 그룹 데이터 베이스를 구성하는 각 멀티캐스트 프로세스 그룹은 네트워크에서 유일한 값을 갖는 단일 식별자(Identifier)로 구분된다.

임의의 한 프로세스가 pg_join() 오퍼레이션을 실행할 때 마다 프로세스 그룹 데이터 베이스가 탐색되어 해당되는 프로세스 그룹의 구성원으로서 추가된다. 만일 해당되는 그룹 엔트리가 존재하지 않으면 다시 pg_create() 오퍼레이션이 실행되고 그 프로세스는 생성된 그룹의 최초 구성원이 된다. 같은 방법으로 pg_leave() 오퍼레이션이 실행되면 프로세스 그룹 데이터 베이스의 해당되는 프로세스가 제거되고, 만일 pg_leave() 실행으로 제거되는 프로세스가 해당 그룹의 마지막 프로세스일 때는 pg_delete() 오퍼레이션이 호출되어 그룹 엔트리가 삭제된다.

그리고, 프로세스 그룹 관리 기능은 프로세스 그룹을 호스트 그룹별로 재구성하여 호스트 그룹 데이터 베이스가 효율적으로 유지될 수 있도록 지원한다. 즉, 프로세스 그룹들이 단일 식별자들로 구분되어 프로세스 그룹 데이터 베이스에 저장되는 반면, 각 프로세스들이 실제로 존재하는 네트워크상의 호스트 그룹을 별도로 구성하여 호스트 그룹 계층으로 전송한다. 그리고 프로세스 그룹 관리 기능은 특정 호스트로 전송되어온 멀티캐스트 메세지를 프로세스 그룹 데이터 베이스를 이용하여 그 호스트 상에 존재하는 여러 프로세스에게 분배할 수 있도록 한다.

3.2 HGMS 구조

HGMS(Host Group Management System)은 PGMS(Process Group Management System)로부터 인계받은 호스트 그룹들을 재구성하고 관리하여 효율적인 멀티캐스트 메시지의 전송 및 수신을 지원한다. 그림 3-3은 HGMS의 기능 모델을 나타낸다.

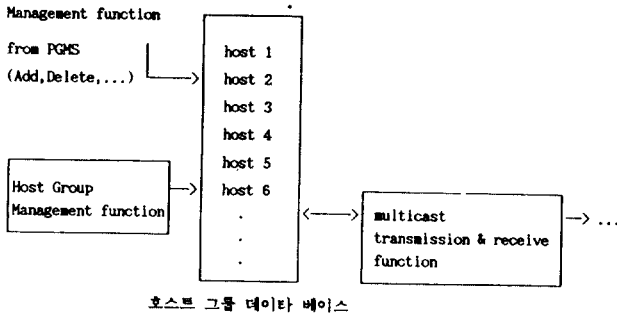


그림 3-3. HGMS의 기능 모델
Fig 3-3. A functional model of HGMS

HGMS의 기능은 모든 호스트 그룹을 포함하는 호스트 그룹 데이터 베이스를 유지한다. HGMS의 다른 기능은 호스트 그룹 데이터 베이스를 이용하여 라우팅(Routing) 정보를 제공하는 것이다. 그리고 장애가 발생한 호스트를 인식하여 호스트 그룹 데이터 베이스를 수정한다. 따라서 HGMS는 특정 호스트 그룹 엔트리의 삭제, 생성 등을 위하여 항상 상위층의 PGMS와 협력한다. 이것은 PGMS가 특정 그룹의 생성이나 삭제 등을 요구하는 메시지를 HGMS로 보내는 것으로 가능하다.

본 논문에서 제공하는 HGMS는 호스트 그룹 데이터 베이스를 트리(tree)구조로 형성하여 호스트 상호관계를 나타낼 수 있도록 하였다.

멀티캐스트 전송시 메시지를 해당되는 서브 트리의 루트 호스트(root host)에게 전송하면 루트 호스트가 다시 목적지의 해당 서브 트리로 전송할 수 있도록 하였다. 즉, 멀티캐스트 메시지를 수신한 호스트는 메시지의 목적지를 검사하여 필요하면 호스트 그룹 데이터 베이스 상의 서브 호스트로 전송하고 최종 목적 프로세스가 호스트 자신에 포함되어 있을 때는 단순히 배달 기능을 수행한다.

예를 들어 네트워크 시스템 상에 h1, h2, h3, h4, h5, h6, h7, h8, h9 등의 9개의 host가 존재하고, 이들 호스트로 구성된 멀티캐스트 전송을 위한 8개의

호스트 그룹이 다음과 같이 존재한다고 가정하자.

- hg1 = (h1, h3, h4), hg2 = (h2, h3, h4), hg3 = (h3, h4, h5)
- hg4 = (h1, h4, h5), hg5 = (h5, h6, h7), hg6 = (h2, h3, h8)
- hg7 = (h6, h9), hg8 = (h4, h5, h6)

이 때 호스트 그룹 데이터 베이스에는 8개의 호스트 그룹을 중복해서 구성하지 않고 다음 그림 3-4와 같은 멤버십(membership) 트리 구조를 구성한다.

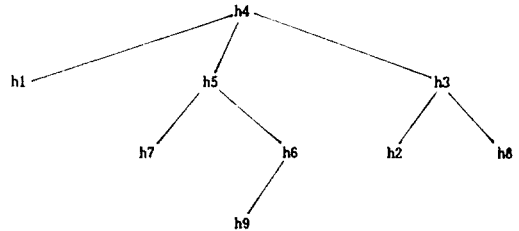


그림 3-4. 호스트 그룹 멤버십 트리
Fig 3-4. Membership tree of host group

제4장에서 멤버십 트리 구조 알고리즘을 기술하였다. 그림 3-4와 같은 트리 구조로 호스트 그룹 데이터 베이스를 구성하면, 호스트 그룹 hg1으로의 멀티캐스트 전송시에는 호스트 그룹 데이터 베이스에서 h4로 메시지를 전송하면 트리 구조에서 h3로의 전송은 바로 이루어진다. 한편 호스트 그룹 h3로 멀티캐스트할 때는 hg3를 구성하는 서브 트리 구조의 루트인 h4로 메시지가 전송되고 다음에 h3, h5로 전송되고 다음에 h3가 h2로 메시지를 전송한다.

IV. GMS(Group Management System) 구현

제4장에서는 멀티캐스트 서비스를 지원하기 위한 프로세스 그룹 관리, 그 프로세스들이 존재하는 네트워크 상의 호스트 그룹 관리 등을 담당하는 GMS(Group Management System)의 기본적 프리미티브와 관리 함수의 구현에 관하여 기술한다.

4.1 구현 환경

GMS의 구현은 10 Mbit/s Ethernet에 연결되어 있는 SunOS 4.1.1의 SUN 워크스테이션에서 실시하

였다. 그림 4-1은 단일 호스트 상에서의 프로세스 그룹 및 호스트 그룹 관리 시스템의 구현 환경을 나타낸다.

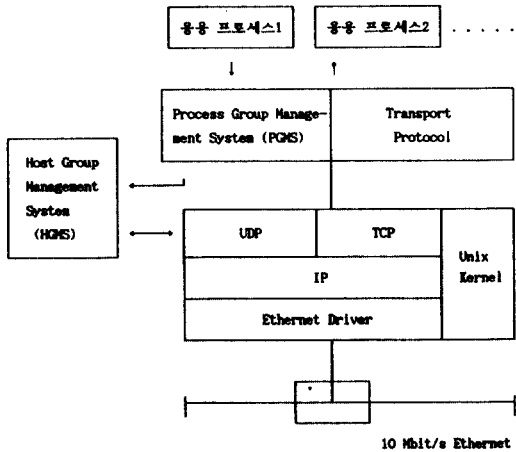


그림 4-1. 구현 환경
Fig 4-1. Implementation configuration

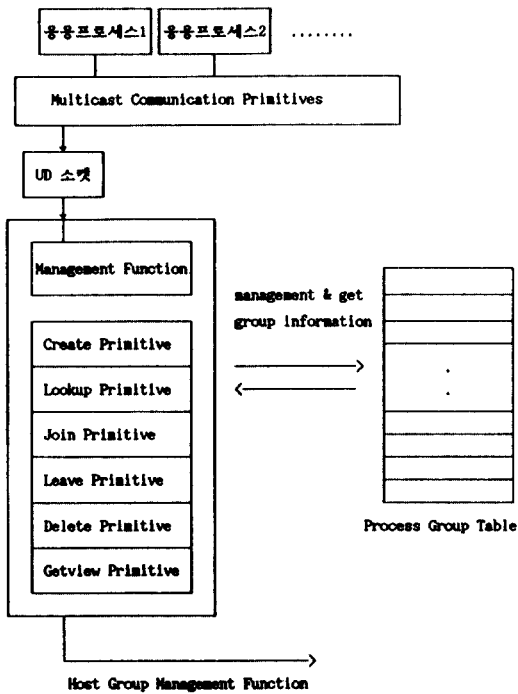


그림 4-2. PGMS 구현 구조
Fig 4-2. Structure of PGMS implementation

4.2 PGMS(Process Group Management System)

PGMS는 프로세스 레벨의 그룹들을 생성, 유지, 관리하는 시스템으로 프로세스 그룹 테이블과 그룹 관리하는 관리 함수들로 구성된다.

PGMS의 구현 구조는 그림 4-2와 같다.

프로세스 그룹 관리는 그림 4-2의 PGMS 구조에서 socket() 시스템 콜을 이용하여 생성된 unix domain(UD) 소켓을 이용하여 이루어진다. 임의의 한 프로세스가 특정 프로세스 그룹에 가입하고자 하면, 관리 시스템과의 통신을 위하여 UD 소켓이 생성된다. 한편, unix 소켓 이름은 프로세스 그룹 테이블 엔트리에 저장된다.

프로세스 그룹 관리를 위하여 구현한 기본적 관리 프리미티브를 표 4-1에 나타내었고 각 프리미티브의 기능은 3.1절에 기술하였다.

관리 시스템의 프로세스 그룹을 구분하는 식별자(Process Group Identifier, Group_Id)는 다음과 같은 구조체(identifier)로 정의 하였다.

struct identifier

```

{
short address_process; /* UNIX process identifier */
short address_port; /* UDP port number */
short address_host; /* host identifier */
u_char address_entry; /* entry point to invoke */
u_char address_res; /* reserved */
}
    
```

위의 Identifier 구조체에서 host Identifier 값은 Unix커맨드 hostid() 함수를 이용하여 결정하고, UNIX Process Identifier 값은 getpid() 함수를 각각 이용한다.

표 4-1. 프로세스 그룹 관리 프리미티브

Table 4-1. Primitives of process group management

프리미티브 타입	프리미티브
그룹 생성	Group_Id = pg_create(gname)
그룹 탐색	Group_Id = pg_lookup(gname)
정보 추출	Address = pg_getview(Group_Id)
프로세스 가입	Ret_code = pg_Join(Group_Id)
그룹 탈퇴	Ret_code = pg_leave(Group_Id)
그룹 제거	Ret_code = pg_delete(Group_Id)
데이터 전송	send(Group_Id, messagepoint)
데이터 수신	receive(Group_Id, messagepoint)

한편, 프로세스 그룹 테이블을 구성하는 엔트리는 다음의 groupview 구조와 같고, 특정 프로세스 그룹에 관한 정보는 프로세스 그룹 테이블의 groupview를 호출하여 알 수 있다.

```

/* groupview */
#define pg_    len 64
typedef struct
{
int    gv_viewid; /* group view number */
identifier gv_gaddress /* group address */
char    gv_gname[pg_len] /* group name */
short   gv_nmem /* number of group members */
identifier gv_members[pg_len] /* list of group members */
int    gv_reference /* group reference count */
} groupview ;
    
```

위에서 정의된 프로세스 그룹 테이블의 groupview 엔트리 정보, 관리 프리미티브 등을 이용하여 관리 함수는 외부로부터 관리 데이터를 입력 받아서 여러 가지 동작을 행하게 된다. 관리 시스템에게 특정 오퍼레이션 요구가 발생할 때 마다 해당 관리 프리미티브에 의하여 그룹 테이블의 엔트리가 변경 수정된다. 프로세스 그룹 관리 시스템이 행하는 관리 절차는 다음과 같다.

〈관리 알고리즘〉

```

while(request_event) /* 관리 요구 데이터 수신 */
{
switch(type of request_event)
{case join : pg_join() ; /* 특정 프로세스 그룹으로의 가입 */
if (pg_join() = -1) /* 지정 그룹이 존재하지 않으면 */
pg_create() ; /* 새로운 그룹 생성 */
case leave : pg_leave() ; /* 호출 프로세스를 그룹으로부터 제거 */
if (nmem = 0) /* 구성원이 더 이상 존재하지 않으면 */
pg_delete() ; /* 해당 그룹을 테이블에서 삭제 */
case lookup : pg_lookup() ; /* 특정 프로세스 그룹 어드레스 탐색 */
case getview : pg_getview() ; /* 지정된 프로세스 그룹 정보 추출 */
}
}
    
```

그리고 메시지 전송 프로토콜 계층에서는 멀티캐스트 메시지의 송수신을 실행하는데, 이때 전송 패킷 형태는 그림 4-3와 같이 구성한다. 본 논문에서는 멀티

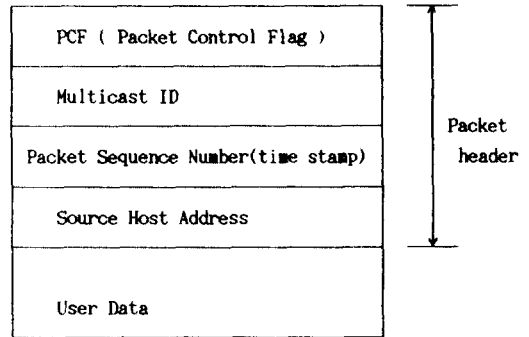


그림 4-3. 패킷 형태
Fig 4-3. Packet format

캐스트 전송 프로토콜에 관해서는 기술하지 않는다.

4.3 HGMS(Host Group Management System)

HGMS(Host Group Management System)의 주된 기능은 Unix 소켓을 통한 상위 PGMS와 요구측 호스트의 요청에 따라서 시스템의 모든 호스트 그룹을 포함하는 호스트 테이블을 관리, 유지하는 것이다. 호스트 그룹 테이블에 저장되어 있는 자료는 임의 호스트가 특정 멀티캐스트 메시지의 수신여부를 결정하는데 이용되며, 멀티캐스트 전송 메시지의 라우팅 정보를 제공한다.

그림 4-4는 HGMS의 구현 모델을 나타낸다.

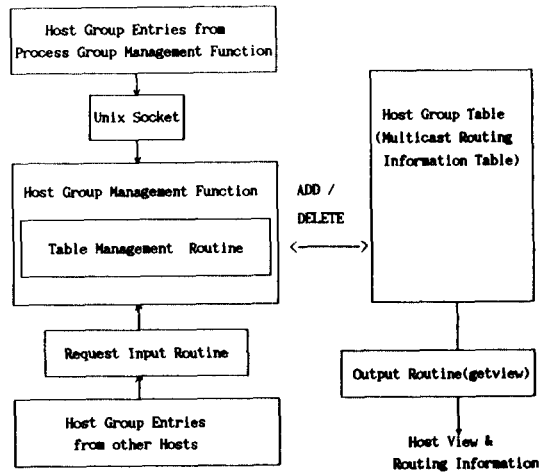


그림 4-4. HGMS 구현 모델
Fig 4-4. Model of HGMS implementation

그림 4-4에서 테이블 관리 루틴(Table Management Routine)은 호스트그룹 테이블을 생성하고 관리한다. 그리고 출력 루틴(Output Routine)의 host_getview() 함수는 호스트 그룹 구조와 라우팅 정보를 제공한다.

호스트 그룹 테이블을 구성하는 엔트리 구조는 다음과 같다.

```
/* format of host group element */
struct host_ele {
    short address_host ; /* host identifier */
    char host_name[ ] ; /* host name */
}
/* format of host group view */
typedef struct
{
    int hg_viewid ; /* host group view number */
    int hg_nmem ; /* number of host member */
    host_ele hg_list[64] /* list of host */
}host_view
```

한편, 테이블 관리 루틴은 프로세스 그룹 데이터 베이스로 부터 프로세스가 존재하는 호스트 리스트를 넘겨 받아 호스트 그룹을 구성된 상호 관계를 이용한 멤버십 트리 형태로 구성한다.

예를 들어 3개의 프로세스 그룹이 다음과 같이 존재한다고 가정하자.

PG1 = (a1, a2, b1, b2, c1)

PG2 = (a1, c1, d1)

PG3 = (a1, a2, a3, c1, e1)

여기서 a1, a2, a3는 모두 같은 a 호스트에, b1, b2는 b 호스트에, c1, c2는 c 호스트에 각각 존재하는 프로세스일 때 호스트 그룹으로 전송되는 호스트 식별자는 a, b, c, d, e의 5개 구성원이다. 따라서 HGMS는 a, b, c, d, e의 5개 호스트로써 멤버십 트리를 구성하여, 호스트 그룹 데이터 베이스를 구축한다.

호스트 그룹이 멤버십 트리 구조로 재구성되면 3.2절의 예제에서 알 수 있듯이, 멀티캐스트 메시지 송수신시에는 해당 그룹을 구성하는 서브 트리의 루트 호스트와 송수신하면 된다. 그 다음 루트 호스트는

다시 해당되는 서브 트리로 메시지를 진행시킨다. 여기서 한 호스트에 여러 개의 목적 프로세스가 존재할 때는 그 호스트 내에서 단순히 IPC(InterProcess Communication) 형태로 메시지 배달을 행한다.

다음은 멤버십 트리 구조를 위한 구현 알고리즘이다. 멤버십 트리 구조 알고리즘은 J.Chang⁽⁵⁾의 프로토콜을 기본으로 하여 설계하였다.

멤버십 트리 관리 알고리즘

```
begin
Group_List <- set of groups
repeat
Select a site S in Group_List
/* Group_List에 가장 많이 중복되어 있는 S를 선택한다. */
```

```
membershiptree(S)
/* S를 루트로 하여 트리를 형성한다. */
until(Group_List is empty)
end
```

```
membershiptree(S)
{
Crossed_Sites <- the sites set of group with S
Group_List <- Group_List - ∇g, group g contains a site S
divide Group_List into sg1, sg2, ..., sgn which are not crossed.
/* subgroup의 어느 site도 교차되지 않는 형태로 Group_List를 n개로 분할한다. */
```

```
Leaf_Sites <- Crossed_Sites - Gropu_List
while(∇s ∈ Leaf_Sites)
{S → s} /* s is linked to Root S */
for i = 1 to n
{
Select a site, Schild
/* Schild는 sgi ∩ Crossed_Sites의 원소 중에서 가장 많이 중복되어 있는 site이다. */
S → Schild
membershiptree(Schild)
}
}
```

다음의 예는 그룹 테이블에 저장되어 있는 정보를

추출하여 보인 것이다. 멤버십 트리 관리 알고리즘으로 형성된 그룹 테이블의 내용을 HGMS output routine의 host_getview() 프리미티브를 호출하여 호스트 이름을 출력한 것이다. 아래에서 호스트 이름 앞의 같은 번호는 트리 구조의 같은 자손을 의미한다.

```
.....
id:1
number of member:9
0:h4
1:h1
1:h5
1:h3
2:null
3:h7
3:h6
4:null
5:h9
6:unll
7:h2
7:h8
8:unll
9:null
10:*
.....
```

V. 결 론

본 논문에서는 멀티캐스트 통신 모델을 프로세스 그룹 계층, 전송 계층, 호스트 그룹 계층으로 나누어 구성하였다.

제안된 통신 모델에서 프로세스 그룹으로의 멀티캐스트 서비스를 지원하기 위한 그룹 관리 시스템을 설계, 구현하였다.

프로세스 그룹 계층에서는 프로세스 그룹의 생성, 제거, 한 프로세스의 가입, 탈퇴 등의 오퍼레이션을 수행하면서, 프로세스 그룹 관리를 담당하는 PGMS (Process Group Management System)를 설계하고, 기본 프리미티브를 구현하였다.

한편, 분산된 프로세스들이 존재하는 호스트 집합을 정돈된 호스트 그룹 형태로 관리하여 전송 프로토콜에게 제공하기 위한 HGMS (Host Group Management System)를 설계하였다.

HGMS는 호스트 그룹의 구성원들을 멤버십 트리 구조로 구성하여 관리하고, 전송 프로토콜에게 멀티캐스트 서비스의 라우팅 정보를 제공한다.

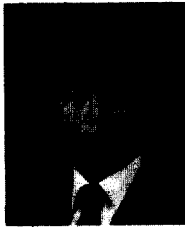
복잡한 분산 처리 환경하에서의 프로세스 및 호스트 그룹 관리 기능은 전송 프로토콜의 효율성을 위하여 필수적이다.

앞으로의 연구 과제로 효율적이고 신뢰성있는 전송 계층 프로토콜을 설계, 구현하고자 한다. 전송 프로토콜 설계시, 본 논문에서 제공하는 그룹관리 시스템(GMS)을 활용할 수 있다.

REFERENCE

1. Alete Ricciardi and P.Birman, "Using process groups to implement failure detection in asynchronous environments," Technical Report, Cornell University Computer Science department, February 1991.
2. David Cheriton and Carey Williamson, "VMTP as the transport layer for high-performance distributed systems," IEEE communication Magazine, pages 37-44, June, 1989.
3. D.R. Cheriton and W. Zwaenepoel, "Distributed Process Groups in the V Kernel," ACM Trans. Computer Systems, Vol.3, No.2, May 1985.
4. Frank Schmuck, "The use of Efficient Broadcast Primitives in Asynchronous Distributed Systems," Ph.D. dissertation, Cornell University, 1988.
5. J. Chang and N. Maxemchuk, "Reliable broadcast protocols," ACM Transactions on Computer Systems, 2(3) : 251-273, August 1984.
6. Keith Marzullo, "Maintaining the Time in a Distributed System," Ph.D. dissertation, Stanford University, Department of Electrical Engineering, June 1984.
7. Kenneth P. Birman and Thomas A. Joseph, "Exploiting virtual synchrony in distributed systems," In Proceedings of the Eleventh ACM Symposium on Operating Systems Principles, pages 123-138, Austin, Texas, November 1987.

8. Kenneth P. Birman and Thomas A. Joseph, "Reliable communication in the presence of failures," *ACM Transactions on Computer Systems*, 5(1) : 47-76, February 1987.
9. L.H Ngoh, "Multicast support for group communication," *Computer Networks and ISDN Systems*, 22, p.165~178, 1991.
10. L. L. Peterson, R. D. Schlichting, "Preserving and Using Context Information in Interprocess Communication," *ACM Transactions on Computer Systems*, Vol.7, No.3, Augst 1989.
11. M. Frans Kaashoek, Andrew S. Tanenbaum, Susan Flynn Hummel, and Henri E. Bal, "An efficient reliable broadcast protocol," *Operating Systems Review*, 23(4) : 5-19, October 1989.
12. T. P. Hopkins, "Transport Protocol requirements for Distributed Multimedia Information Systems," *Computer J.* 32(3), P.252-261, 1989.
13. W.prinz, R. Speth, "Group Communication and Related Aspects in office Automation," *proc. of IFIP WG 6.5*, 1987.
14. 박현제, 전길남, "부산 시스템에서의 그룹 상호작용," 한국과학기술원, 박사 학위 논문, 1990.
15. ISO/IEC Information Processing System, Open System Interconnection, Management Information Services-Structure of Management Information-Part 1 : Management Information Model, 1989.



박 판 우 (Pan-Woo Park) 정회원

1961년 8월 1일생

1984년 2월 : 경북대학교 전자공학과 전산 전공 (공학사)

1986년 6월 : 광운대학교 대학원 전자계산학과 (이학석사)

1988년 8월 ~ 1992년 8월 : 광운대학교 대학원 박사과정 수료

1987년 3월 ~ 1991년 2월 : 기전여자전문대학 전자계산과 조교수

1991년 3월 ~ 현재 : 대구교육대학교 조교수

※주관심분야 : 네트워크 프로토콜, 분산처리, CAI

조 국 현 (Kuk-Hyun Cho)

정회원

1953년 12월 22일생

1970년 3월 ~ 1977년 2월 : 한양대학교 전자공학과 졸업 (공학사)

1979년 4월 ~ 1981년 3월 : 일본 동북대학 대학원 전자통신공학과 졸업 (공학석사)

1981년 4월 ~ 1984년 3월 : 동 대학 대학원 동 학과 (공학박사)

1984년 3월 ~ 현재 : 광운대학교 전자계산학과 조교수, 부교수

1987년 4월 ~ 1991년 2월 : 한국정보과학회 정보통신분과위원회 총무, 운영위원

1991년 3월 ~ 1993년 9월 : 동 분과위원장

※주관심분야 : 컴퓨터 네트워크, 프로토콜 성능평가