

# 동적인 초고속 통신망에서 가상위상을 이용한 신뢰성있는 그룹통신 플랫폼 개발에 관한 연구

신용태<sup>†</sup> · 조민수<sup>††</sup>

## 요 약

본 논문은 동적인 초고속 통신망에서 신뢰성있는 그룹통신 플랫폼을 위한 가상위상 *Multiph*를 제안하고 메시지 전송 프리미티브와 동적인 그룹관리 프리미티브를 정의하였다. *Multiph*는 그래프와 같이 정점과 간선으로 구성된다. *Multiph*의 정점은 프로세스 그룹을 의미한다. 간선은 각 정점들간의 연결로 프로세스 그룹간의 메시지 통로를 의미한다. 메시지를 송수신하는 프로세스가 여러개의 멀티캐스트 그룹에 중복된 경우에는 중복된 프로세스들을 중복이 없는 논리적 프로세스 그룹으로 구성한다. 이것을 메타그룹(*meta-group*)이라 한다. 메타그룹으로 형성된 멀티캐스트 그룹을 *Multiph*로 구성한다. *Multiph*는 하이퍼트리(*hypertree*)(2) 형태의 그래프 구조이다. *Multiph*는 동일한 그룹에 도착하는 메시지들에 대한 순서성(*ordering*)을 보장한다. 또한, *Multiph*는 다른 논리적 구조에서 메시지 전송의 효율을 저하시키는 추가 노드를 가지고 있지 않다. 그리고 프로세스의 가입과 탈퇴 등 그룹의 동적인 변화에 대하여도 처음 구성된 플랫폼의 구조를 새로 재구성하지 않는다.

## Reliable Group Communication Platform using Virtual Topologies in a Dynamic High-Speed Network

Yong Tae Shin<sup>†</sup> · Min Soo Cho<sup>††</sup>

## ABSTRACT

We propose *Multiph* which is a virtual topology for a reliable group communication platform on dynamic high speed network. We also define group communication primitives for transmission of messages and group management. *Multiph* consists of a set of nodes which are process groups and a set of edges which are the connection between nodes providing the route of messages. Processes can send and receive messages overlapped in a several multicast groups. Overlapped process organizes non-overlapped logical group, which called *meta-group*. *Multiph* guarantees ordering of messages that received at the destination group. It has no extra node that decreases system performance. Although the group has dynamic group change to leave and add process, it does not reconstruct the existing platform structure.

### 1. 서 론

컴퓨터 및 통신 기술의 발달로 사용자들간에 많은

정보의 교환이 일어나게 되었다. 기존의 전송자가 하나 이고, 수신자도 하나인 일-대-일 정보 전달 방식은 정확하고 빠른 정보의 수신을 보장하지만 같은 정보를 수신자가 하나 이상인 수신자에게 전송하고자 할 때는 정보전달을 위해 처리하여야 하는 반복적인 작업이 너무 많아 신속한 정보 전달이 어렵다. 이러한 일-대-일 정보전달 방식의 단점을 보완하고자 일-대-전부의 통신

\* 본 연구는 학술진흥재단 1996년 신진교수과제(과제번호: 04E0328)의 지원을 받아 수행되었음.

†정희원: 숭실대학교 컴퓨터학부 조교수

††정희원: 경인여자대학 사무자동화과 전임강사

논문접수: 1998년 2월 19일, 심사완료: 1998년 4월 23일



용하여 생성한다. 메타그룹은 통신망에 산재되어있는 프로세스들을 수신하는 메시지에 따라 모아놓은 객체(object)이다. *Multiph*는 정점과 간선으로 형성된 그래프이고, 각 정점은 메타그룹을 의미하며, 간선은 물리적으로 연결된 각 메타그룹간의 통신경로를 의미한다. *Multiph*는 전송되는 메시지에 대한 인과적(causal)순서성을 보장하는 구조이고, 메시지에 따라서 필요하지 않는 노드나 프로세스들의 간선을 제거하고 필요한 간선을 추가하는 다른 가상위상과는 달리 가상위상의 구조를 전체적으로 재구성하지 않는다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존에 개발된 다중전송 프로토콜에 대하여 조사하였으며, 3장에서는 그룹통신 메카니즘이 가져야 하는 기본적인 성질에 대하여 기술하였으며, 4장에서는 본 논문에서 제안하는 신뢰성있는 그룹통신 플랫폼을 위한 가상위상 *Multiph*의 구조와 생성 알고리즘을 기술하였고, 5장에서는 구성된 플랫폼에서 사용되는 그룹통신 프리미티브와 그룹관리 프리미티브를 기술하였고, 6장에서는 제안한 *Multiph*에 대한 모의 실험과 결과를 기술하였다. 그리고, 마지막으로 7장에서는 본 논문의 결과와 향후 연구방향에 대하여 기술하였다.

2. 관련 연구

많은 통신시스템이 개발되고 사용되었지만, 분산시스템으로 구성된 여러 대의 컴퓨터에서 사용하는 소프트웨어와 하드웨어의 특성에 따라 분산처리에 근간을 이루는 통신시스템은 컴퓨팅 성능의 향상보다는 저하를

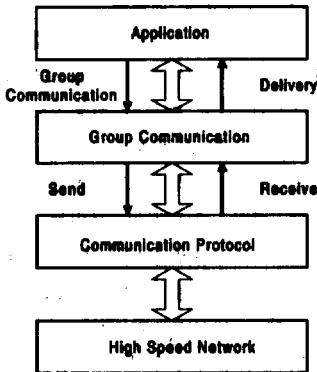
가져오는 경우가 많다. 본 절에서는 분산환경에서의 기존의 멀티캐스트 프로토콜에 관하여 기술한다.

(그림 1)은 그룹통신을 위한 일반적인 시스템 모델이다. 가장 아래 계층은 메시지의 실질적인 전송을 담당하는 계층이다. ATM 네트워크나 초고속통신망을 사용할 수 있다. 초고속통신망 계층의 상위 계층인 커뮤니케이션 프로토콜 계층은 네트워크상에 메시지를 전송하는 계층으로 현재 많이 사용되고 있는 TCP/IP를 이용해 구현할 수 있다. 다음 계층인 그룹통신 계층은 다중전송을 지원하는 계층으로 프로세스들을 그룹으로 형성하거나 형성된 그룹을 관리하는 계층으로 신뢰성있는 그룹통신을 위한 본 논문에서 핵심적인 계층이다. 최상위 계층인 응용 계층은 사용자가 직접적으로 시스템과 접하는 부분으로 그룹통신을 위한 응용 프로그램이 존재하는 계층이다.

Amoeba 시스템(9)은 분산 처리 시스템을 위하여 개발된 운영체제 중 하나이다. 메시지를 처음으로 전송 받은 컴퓨터나 프로세서를 서버라 하고, 메시지를 전송한 컴퓨터나 프로세서를 클라이언트라 한다. 서버로 메시지를 전송한 클라이언트는 블럭상태에 들어가고, 서버는 수신한 메시지를 클라이언트인 다른 프로세서나 컴퓨터에 전송하고, 처음으로 메시지를 서버로 전송한 클라이언트에 언블럭 메시지를 전송한다. 서버로부터 언블럭 메시지를 받을 때까지 서버에 메시지를 전송한 클라이언트는 아무런 작업도 수행하지 않는다. 또한 각 그룹내에는 모든 프로세서나 컴퓨터에 메시지 전송 책임을 갖는 시퀀서(sequencer)를 갖는다.

Transis(14)는 히브리 대학에서 개발한 통신 서비스 시스템이다. 이 시스템은 통신을 지원하는 여러 계층의 프로토콜 중 하위의 계층인 트랜스포트 계층에 신뢰성있는 다중전송을 지원하도록 통신 계층을 구성한 시스템으로, 브로드캐스트 도메인과 커뮤니케이트 도메인이라는 두 개의 모델로 구성된 시스템이다. 브로드캐스트 도메인에서 Transis의 트랜스포트 계층 서비스를 제공하는 가장 작은 단위를 Lansis라 하고, Lansis들을 연결시켜 하나의 커뮤니케이트 도메인을 구성하였다.

ISIS시스템(5)은 코넬 대학에서 개발한 다중전송을 위한 통신 시스템으로 가장 널리 알려진 다중전송 시스템으로, 분산 컴퓨팅 환경과 유닉스의 이질적인 네트워크에서 분산 응용 프로그램을 지원해 응용 프로그램의 성능과 가용성(availability)을 증대시켰다. ISIS시스템에서는 "virtual synchrony"라는 통신 프로그래밍



(그림 1) 시스템 구성도 (Fig. 1) System Structure

스타일을 제공하는데, 이것은 시스템안에서 일어나는 사건(event)들이 모든 프로세스들에게 일관적인 순서로 전달되며, 사건이 마치 동기적인 것처럼 프로세스들에게 동일한 변화를 유발하는 성질이다. ISIS 시스템에서는 "virtual synchrony" 기법을 이용해 전달되는 메시지의 전송서정(total ordering)을 보장한다.

이 밖에도 프로토콜에만 변화를 주어 메시지의 다중 전송 방식을 이룩하는 방법들도 있다. 메시지를 수신하는 한 그룹내의 프로세스와 가입과 탈퇴에 일관성을 유지시키는 Trans 프로토콜(6)과 전송되는 메시지의 부분 순서(partial order)를 유지시키기 위해 메시지에 대한 구문 그래프(context graph)를 이용하는 Paync 프로토콜(7), 그리고 메시지의 전송을 초대 단계, 알람 단계 그리고 위임 단계로 구분 메시지의 다중전송을 이룩하는 Luan & Glior 프로토콜(11)등이 있다.

앞에서 기술한 프로토콜들은 메시지를 수신하는 그룹이 중복된 경우에 대해서는 언급이 없다. 프로세스 그룹이 중복된 경우에도 메시지의 순서성과 효율적인 전달을 수행하기 위한 방법으로는 프로세스 그룹을 논리적인 구조로 재구성하여 메시지를 전송하는 방법이 주로 사용되었다. 프로세스 그룹의 재구성 방법에는 중복된 프로세스 그룹을 전파 트리(propagation tree)(13)로 구성하는 방법이 있다. 전파 트리는 트리 구조의 특징을 가지고 있기 때문에, 메시지 전송을 위한 하나의 경로를 제공하는 구조이다.

### 3. 그룹통신

그룹은 하나 또는 그 이상의 프로세스로 구성되고, 이들 프로세스들은 일부 서비스에 대해서는 서로 달리 실행하고, 일부 서비스에 대해서는 협력하거나 응용 프로그램을 같이 실행할 수 있다. 이러한 프로세스간에 발생하는 메시지의 송수신에는 메시지에 대한 원자성과 순서성 그리고 영속성이 만족되어야 한다. 그룹통신은 다음 세 가지의 요소로 구성되며, 본 절에서는 그 구조를 기술한다(1, 16)...

#### 3.1 원자성

신뢰성이란 통신망을 통해 전송되어지는 메시지가 송신자에서 수신자로 정확하게 전송되어짐을 보장하는 것이다. 통신 프로토콜을 구성하는 물리 계층은 송신측에서 수신측이 존재하는 곳에 통신 선로를 통해 정확한

메시지의 전송만을 보장한다. 그러므로 그룹통신에서 통신 선로상의 신뢰성은 통신망에서 발생할 수 있는 임의의 결함을 허용하면서 메시지를 전송하는 기능으로 신뢰성을 갖는 브로드캐스트 네트워크에서 자연히 이루어진다.

신뢰성을 보장하는 브로드캐스트 네트워크를 통과한 메시지는 그룹으로 형성된 목적지까지는 정확한 메시지 전송을 보장받는다. 그러므로 그룹통신에서 신뢰성은 그룹 내부에 포함되어있는 프로세스들에 전송되어지는 모든 메시지가 정확하게 전달되어짐을 보장하는 기능이 다. 정확한 메시지의 전달을 보장하기 위해서 그룹통신에서는 메시지의 수신(receive) 단계와 메시지의 인도(delivery) 단계로 나눈다. 메시지가 그룹의 한 프로세스에 도착하는 단계를 수신이라고 하고, 그룹 프로세스의 모든 프로세스가 메시지를 받는 단계를 인도라 한다. 그룹내 과반수 이상의 프로세스가 메시지를 수신하였다하더라도 수신하지 못한 프로세스가 있다면 지금까지 메시지를 수신한 모든 프로세스에서도 메시지 수신을 취소하는 방식으로, 이러한 성질을 원자성(atomicity or all-or-nothing)이라 한다.

#### 3.2 순서성

앞 절에서 기술한 그룹통신에서의 원자성은 메시지를 수신하여야 하는 프로세스를 포함하는 한 그룹에 메시지가 도착하고 이 메시지를 그룹에 속해있는 모든 프로세스들이 메시지를 받도록 보장하는 것으로 하나의 메시지만을 생각하였다. 그러나 그룹에는 하나의 메시지만 전송되어지는 것이 아니다. 여러 개의 메시지가 동일한 그룹에 수신될 수도 있다. 이러한 경우, 그룹에 포함되는 모든 프로세스들은 그룹에 도착하는 여러 개의 메시지를 동일한 순서로 수신하도록 보장하는 성질이 순서성이다.

메시지의 순서성을 보장하는 방법에는 두 가지 대표적인 방법이 존재한다. 송신측에서 발생하는 모든 메시지에 대하여 일정한 순서를 부여함으로써 수신측의 그룹들이 그 순서에 맞춰 메시지를 수신하도록 하는 방법으로, 이것은 메시지에 시간도장을 발급하는 방법이다. 이 방식은 전송되는 모든 메시지에 일관된 순서를 부여할 수 있다는 장점이 있지만, 메시지를 전송하는 프로세스의 수가 많을 경우, 일관성있는 시간 도장 발급에 문제점이 있다.

또다른 방식은 발생되어지는 메시지들간의 인과적인

관계로 순서를 결정하는 방식이다. 이 방식은 모든 메시지에 순서를 부여하는 것이 아니라, 메시지들 사이에 인과적인 관계가 존재하는 메시지들에만 순서를 정의하는 방식으로 전체적인 순서를 부여하는 방식보다는 효율적인 메시지 전송을 보장한다. 그러나, 이 방식은 메시지의 순서를 부여하는 방식과 순서를 부여하는 기준 선정이 어려워, 효율적인 메시지 전송에 장애를 줄 수도 있다.

3.3 영속성

영속성은 물리적이거나 논리적인 결합으로 그룹간의 메시지 전송이 어려울 때에도 그룹간에 원자성과 순서성을 만족하면서 끊임없는 메시지의 전송을 보장하는 성질이다. 그룹통신의 영속성을 지원하는 방법에는 물리적 결합을 극복하는 방법과 논리적 결합을 극복하는 방법 모두를 사용한다.

하드웨어적인 결합은 물리적 결합을 의미하는 것으로, 그룹들을 연결하는 물리적인 링크가 끊어지거나, 손상으로 인해 지속적인 메시지 전송이 어려운 상황을 의미한다. 이 결합은 손상입은 링크를 대신할 수 있는 물리적인 여분의 링크를 갖게함으로써 극복할 수 있는 방법과 알고리즘에 의해서 결합이 발생한 두 그룹을 연결하는 논리적인 우회 경로를 배정함으로써 해결할 수 있다. 반면에 소프트웨어적인 결합은 논리적 결합을 의미한다. 소프트웨어적인 결합에는 많은 종류가 있다. 이것을 해결하기 위한 방법으로는 다음 세 개의 단계로 해결할 수 있다. 먼저, 시스템에서 허용할 수 있는 결합에 대한 정의가 이루어져야 한다. 두 번째로, 결합이 발생한 프로세스가 오동작이나 다른 정상적인 프로세스에 영향을 미치지 못하도록 고립시킨다. 세 번째로, 결합이 발생한 프로세스의 역할을 정상적인 프로세스로 이동하거나 아니면 종료한다. 이와 같은 방법으로 소프트웨어적인 결합을 해결한다.

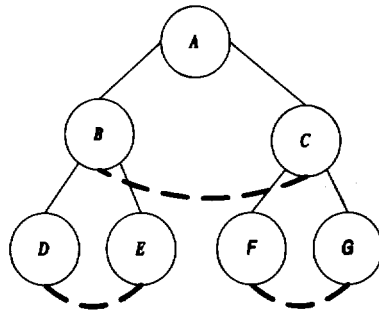
4. Multiph를 구성하는 알고리즘

본 절에서는 본 논문에서 제안하는 다중전송을 위한 Multiph에 대하여 기술한다. 먼저, 구조를 기술하기 전에 구조의 설명을 위해 사용된 정의에 대하여 기술하고, 이러한 정의를 바탕으로 프로세스 그룹을 트리 구조와 유사한 하이퍼 트리(hyper-tree) [2] 구조로 구성하는 알고리즘에 대하여 기술한다. Multiph는 그래프

프로 다음과 같이 정의할 수 있다.

(정의 1) Multiph :  $M = (V, E)$  이라면,  $V$ 는 정점들의 집합을 의미하고,  $E \subseteq V \times V$  이다.

트리 그래프는 그래프를 구성하는 정점과 간선 사이에 사이클이 존재하지 않는 그래프를 말한다. 트리 구조의 단점은 동일한 깊이에 존재하는 다른 서브 트리에 존재하는 두 정점사이의 경로는 뿌리 정점에서 정점이 존재하는 깊이에 두배가 된다는 것이다. 그리고 모든 경로 배정시 뿌리 정점으로 모든 메시지가 집중된다는 것이었다. 이와 같은 단점을 극복하기 위해 동일한 깊이에 존재하는 정점사이에 하나의 간선을 추가하여 트리의 단점을 보완한 그래프가 하이퍼 트리이다. 이 때, 추가된 간선을 하이퍼 간선(hyper-edge)이라 한다. 다음 (그림 2)는 7개의 정점으로 구성된 이진 트리구조에서 동일한 깊이에 위치하고있는 정점 B와 정점 C는 정점 A에 대한 자식정점으로써 형제정점라 부르고, 이 두 정점을 연결한 점선이 하이퍼 간선이다.



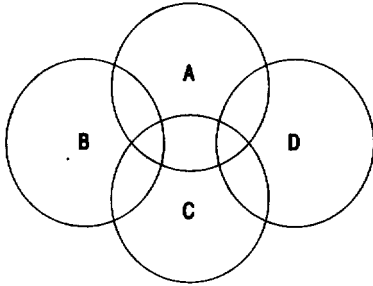
(그림 2) 하이퍼 간선의 예 (Fig. 2) Example of Hyper edge

4.1 정의

(정의 2) 다중전송 그룹(multicast group) : 프로세스의 그룹으로 메시지를 수신하거나 송신할 수 있으며, 대문자로 표현된다.

다중전송 그룹은 (그림 3)에서와 같이 A, B, C, D 와 같이 대문자로 표현한다. 메시지를 수신하는 프로세스들이 여러개의 다중전송 그룹에 포함되는 경우에는

다음의 (그림 3)에서와 같이 다중전송 그룹들간에 중복이 존재할 수 있다.



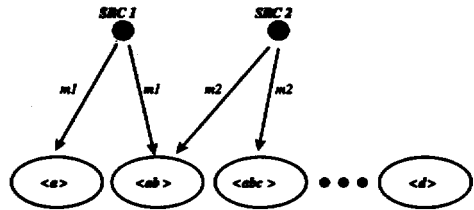
(그림 3) 중복된 다중전송 그룹  
(Fig. 3) Overlapped Multicast Group

두 개 이상의 다중전송 그룹에 포함된 프로세스들은 중복하는 다중전송 그룹의 모든 메시지를 전송받아야 하고, 또한 중복된 프로세스들은 수신하여야 하는 모든 메시지에 대하여 같은 순서로 메시지를 수신하여야 한다. 그렇지만, 두 개 이상의 다중전송 그룹에 포함되는 프로세스들은 앞에서와 같이 두 그룹은 서로 다른 전송자로부터 메시지를 수신하고, 서로 다른 대표자 프로세스를 가지고 있어 중복된 그룹에 있는 프로세스에 대한 메시지의 순서를 정의하기가 어렵다. 그러므로 두 개 이상의 다중전송 그룹이 중복된 그룹을 중복이 하나도 없는 하나의 다중전송 그룹으로 다시 구성한다. 중복이 없는 다중전송 그룹은 메타그룹(meta group)이라 부르며, 각 메타그룹에는 메시지의 송수신을 담당하는 대표자 프로세스가 존재한다. 이 프로세스는 그룹에 포함되는 모든 프로세스들에게 동일한 순서로 메시지를 전송한다.

[정의 3] 메타그룹(meta group, MP) : 두 개 이상의 다중전송 그룹에서 중복되지 않는 그룹이다. 메타그룹은 소문자로 나타내며, < >로 표현된다.

둘 이상의 다중전송 그룹이 중복된 부분은 메타그룹이 된다. 그리고 이렇게 형성된 메타그룹의 이름은 메타그룹의 형성을 위해 참여한 모든 다중전송 그룹의 이름을 사용한다. 위의 (그림 3)을 메타그룹으로 형성하면 다음과 같은 여러 개의 메타 그룹을 얻는다. <a>, <b>, <c>, <d> 메타그룹은 메타그룹을 형성하기 위해

참여한 다중전송 그룹이 하나인 경우이다. 두 개 이상의 다중전송 그룹에 의해서 형성된 메타그룹은 다음과 같이 하나의 메타그룹을 형성하기 위해 참여한 모든 다중전송 그룹들의 이름을 나열한 것이다. <ac>, <cd>, <ab>, <ad>, <bd>, <acd>, <abd>, 이것은 (그림 3)의 중복된 다중전송 그룹을 다중전송 그룹간에 프로세스의 중복이 하나도 없이 형성된 메타그룹들이다.



(그림 4) (그림 3)에 대한 멀티캐스트 표현  
(Fig. 4) Presentation of Multicast for fig. 3

위의 (그림 4)는 메타그룹으로 중복된 다중전송 그룹을 형성함으로써 동일한 그룹에 전송되는 메시지들은 동일한 순서로 그룹원에 전송할 수 있음을 보여준다.

#### 4.2 Multiph 구성

다중전송 그룹을 본 논문에서 제안하는 Multiph로 구성하기 위해서는 서로 중복되어져 있는 다중전송 그룹을 먼저 중복이 없는 메타그룹으로 변환한다. 메타그룹의 이름은 메타그룹을 형성하기 위해 참여한 다중전송 그룹의 이름들을 순서적으로 기술한 것으로 하며, 그 이름의 갯수를 다음과 같이 정의한다.

[정의 4] 농도(cardinality) : 메타그룹을 형성하기 위해 참여한 다중전송 그룹의 개수를 의미하며, 표기법은 다음과 같이 "I"로 한다.

예를 들어 메타그룹 <abc>의 농도는 다음과 같이 표기하고,  $I_{\langle abc \rangle} = 3$ , 값은 정수값을 갖는다. 농도가 '1'인 메타그룹은 처음에 주어진 다중전송 그룹에서 다른 다중전송 그룹과 겹치는 부분이 없음을 의미하고, 본 논문에서 구성된 Multiph에 따라 전송되는 메시지의 출발점이 된다.

[정의 5] 배제(exclusion) : 두 개의 메타그룹이 배제 관계에 있다면, 두 메타그룹은 서

로 포함 관계가 상향하지 않는다. 다시 말해서 '서로 소'인 관계를 말한다. 표현은 다음과 같이 ~"로 한다.

예를 들어, (그림 3)의 다중전송 그룹에 의해 생성된 메타그룹에서 메타그룹 <abc>와 <acd>인 두 메타그룹의 관계를 말한다. 배제 관계의 메타그룹은 농도가 서로 같거나, 한 메타그룹이 다른 메타그룹을 포함하지 못하고, 그 역도 성립한다. 이러한 메타그룹은 여러개가 존재할 수도 있다. (그림 3)에서 생성된 메타그룹 <abc>의 배제는 다음과 같이 표현한다. ~<abc> = {<acd>, <cd>, <ad>, <bd>, <d>} 인 메타그룹의 집합이 된다.

제안하는 *Multiph*는 먼저, 주어진 다중전송 그룹에서 메타그룹을 형성하고, 형성된 메타그룹을 *Multiph*로 구성하는 단계와 구성된 그래프를 바탕으로 메시지의 순서를 유지하면서 각 그룹의 프로세스에게 메시지를 전송하는 메시지 전달 단계로 구성된다. 다음의 알고리즘은 다중전송 그룹에서 메타그룹을 형성한 후 다중전송을 위한 *Multiph*를 구성하는 알고리즘이다.

```

main ()
{
    mG = max ( |MPi| ), 1 ≤ i ≤ k
    label(mG) = 1;
    construct_graph (mG);
}

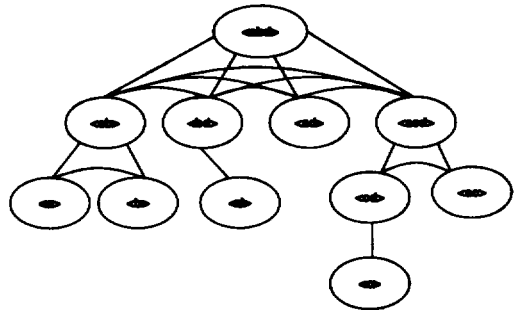
construct_graph (mG)
{
    for label(MPi) = 0 do
    {
        if ((|mG| = |MPi| - 1) && (mG ⊂ MPi)) then
            label(MPi) = label(mG) + 1
        else if (mG ⊄ MPi) then
            ~mG = ~mG ∪ MPi;
    }
    for (exist ~mG) do
    {
        Add_MP = MPi : min ( |MPi| ), MPi ∈ ~mG
        label(Add_MP) = label(mG) + 1;
    }
    link to selected MPi;           /*Hyper-link*/
    for (label (MPi) = 0) do, 1 ≤ i ≤ k-i
    {
        mG = min ( |MPi| ); label(MPi) > 0 && not mG
        /*Select new mG*/
        construct_graph (mG);      /*Recursive call*/
    }
}

```

위의 알고리즘에서 사용하는 label은 각 메타그룹이

*Multiph*를 형성하는데 선택되었음을 나타내는 표시 필드이다. 모든 메타그룹들은 알고리즘이 실행되기 이전에 label 필드는 '0'으로 초기화 된다. mG 그룹을 선택하고 다시 *construct\_graph()*를 호출하기 전에 label 값이 '0'이 아닌 메타그룹들 중 하나의 메타그룹을 선택하게 된다. *Multiph*를 형성하는 동안 각 메타그룹의 label값은 모두 '0'보다는 큰 값을 갖게 된다. label값이 '0'인 메타그룹이 존재하면, 아직도 *Multiph*로 형성되지 못한 메타그룹이 있음을 의미한다.

다음 (그림 5)는 위와 같은 알고리즘에 (그림 3)에서 생성된 메타그룹을 입력으로 넣었을 때, 형성되는 *Multiph*로 트리형태를 가지고 있는데, 트리에서 형제 정점간에 간선을 추가한 형태의 그래프로 하이퍼 트리의 형태를 가지고 있는 그래프이다. 아래의 그래프는 트리의 속성을 그대로 유지하면서 트리 구조의 단점인 깊이가 커졌을 때, 비례적으로 증가하는 메시지 전달의 지연 시간을 줄일 수 있다.



(그림 5) *Multiph* 형성  
(Fig. 5) Construction of *Multiph*

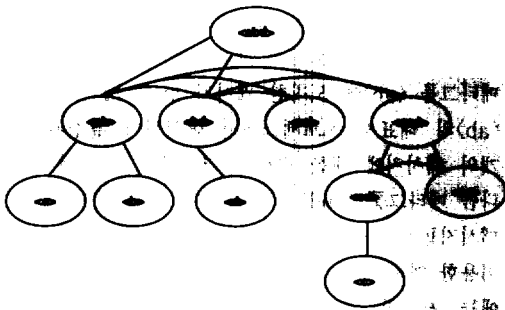
*Multiph*는 임의의 메타그룹에서 구성한다. *Multiph*가 구성되면, 이 그래프의 정보는 모든 메타그룹에 전송한다. 다중전송 그룹에서 메타그룹을 형성하기 위해서는 다중전송 그룹에 포함된 모든 프로세스를 구성된 배열을 사용함으로써 형성할 수 있다. 이때, 소요되는 시간은  $O(n \log n)$ 의 시간이 소요 된다. ( $n$ 은 모든 다중전송 그룹에 포함된 프로세스의 총 수) 메타그룹이 형성된 다음에는 위에서 기술한 알고리즘에 의해서 *Multiph*를 형성는데, 이 때, 소요되는 시간은  $O(m_s)$ 이다. ( $m_s$ 는 메타그룹의 총수) 따라서 주어진 다중전송 그룹에서 다중전송을 위한 논리적 위상인 *Multiph*를 형성하는데 소요되는 시간은  $O(n \log n + m_s)$ 의 시간

복잡도를 갖는다.

4.3 Multiph의 최적화

앞장에서 기술한 Multiph는 동일한 부모 정점을 갖는 형제 정점들간에 간선을 추가하였다. 그러므로 형성된 Multiph에는 메시지 전송에 실질적으로 필요하지 않는 간선의 수가 많이 있다는 단점을 가지고 있다. 예를 들어, (그림 5)에서 메타그룹 <ad>에서 <acd>로의 간선은 전혀 사용되지 않는 간선이고, 또한 메타그룹 <ad>에서 <abd>로의 간선도 전혀 사용되지 않는 간선이다. 그리고, 농도가 '1'인 메타그룹 <a>에서 <b>로의 간선도 메시지 전송이 전혀 일어나지 않는 간선이다. 이러한 간선들은 제거되어도 형성된 Multiph의 성능에서는 전혀 지장을 주지 않으며, 더욱 간결하게 가상위상을 만들어 준다.

이러한 간선을 찾아내는 방법은 위의 알고리즘으로 형성된 Multiph로 메시지를 전송하기 전에 메시지의 출발점인 Multiph의 일정점에서 최적화 메시지를 전송함으로써 찾아 낼수 있다. (그림 5)의 Multiph의 노드 <a>, <b>, <c>, <d>에서 출발한 최적화 메시지는 Multiph 간선으로 먼저 보내지고 다음 하이퍼 간선으로 보내진다. 이때, 메시지 수신이 중복되거나 수신하지 않아도 되는 메시지를 수신하도록 하는 간선을 제거함으로써 (그림 6)과 같은 Multiph를 얻을 수 있다.



(그림 6) 최적화된 Multiph (Fig. 6) Optimized Multiph

5. 그룹통신을 위한 프리미티브

이 장에서는 앞장에서 구성한 가상위상 Multiph상에서 메시지를 전송하는데 필요한 프리미티브와 형성된 가상위상에 참여하거나 탈퇴를 원하는 프로세스에 의해

서 발생하는 메타그룹의 생성과 삭제에 관련된 그룹관리 프리미티브에 대하여 기술하고자 한다.

5.1 통신 프리미티브

가상위상 Multiph상에서 메시지의 전송 시작은 가상위상의 말단에 위치한 농도가 '1'인 메타그룹에서부터 시작 된다. (그림 5)의 Multiph에서 (그림 6)의 Multiph로 이루어진 가상위상의 행태를 가상위상에 참여한 모든 메타그룹의 대표자 프로세스에 이웃하는 메타그룹이 어느것인가를 알린다. 이때, 트리간선과 하이퍼 간선으로 연결된 이웃하는 메타그룹을 알리는데, 이것이 경로 배정표가 된다. 가상위상상에서 메시지의 전송을 수행하는 프리미티브로는 다음과 같은 프리미티브에 의해서 전송이 이루어진다.

메시지의 전송을 처음 시작하는 메타그룹은 주어진 가상위상 Multiph에서 말단에 존재하는 농도가 '1'인 그룹에서부터 메시지의 전송은 시작되고, 전송되어진 메시지가 가상위상에서 가장 높은 곳에 위치한 농도가 가장 큰 메타그룹에 도달하게 되면, 모든 메시지의 전송은 이루어진 것으로 생각한다. Multiph의 메타그룹에서 메시지의 수신은 각 메타그룹의 대표자 프로세스가 수신한다. 수신한 메타그룹의 농도가 가장 큰 메타그룹이 아니면 수신된 메시지를 다시 인출하는 메타그룹으로 전송하는 책임도 갖는다. 그러므로 대표자 프로세스는 그룹간의 메시지 송수신에 대한 책임과 함께 그룹내에서 메시지들의 순서성과 원자성을 보장하는 책임도 갖는다.

그룹의 대표자 프로세스에 도착한 메시지들은 그룹에 포함된 모든 프로세스에게 그룹통신의 신뢰성을 만족시키기 위해 원자성과 순서성 그리고 영속성이 만족되어 지도록 전송한다. 신뢰성을 만족하는 메시지 전송에 대한 책임도 또한 대표자 프로세스에서 갖고 있어 신뢰성이 만족되지 않는 메시지 전송이 이루어진다면, 대표자 프로세스에서 신뢰성이 만족될 때까지 메시지 전송을 책임진다. 즉 그룹에 포함된 모든 프로세스들에 대한 식별자로 대표자 프로세스는 알아야 한다.

다음의 프리미티브들은 Multiph상에서 그룹간(inter-group) 또는 그룹내(intra-group) 메시지 전송을 위해 필요한 프리미티브를 나타낸 것으로, 전송되는 메시지들의 순서성과 원자성 그리고 영속성을 만족해 전체적인 그룹통신의 신뢰성을 보장하는 프리미티브이다. 각 프리미티브들은 몇 개의 인자를 가지고 있으며, 각



인자들은 다음과 같다.

**Multicast (msg, MG) :**

가상위상의 간선을 따라 목적지 그룹 MG로 메시지를 전송하는 프리미티브로 msg는 실질적인 메시지를 의미하고, MG는 목적지 그룹을 나타낸다.

**Receive (msg, message\_buf) :**

메시지를 수신하는 그룹에서 전송된 메시지를 수신하는 프리미티브이다. 여기서 msg는 수신된 메시지를 의미하고, message\_buf는 메시지를 저장하는 버퍼를 의미하는 것으로, 메시지에 대한 순서성과 원자성을 보장하기 위해 사용한다.

**Send (msg, member\_id) :**

각 그룹의 대표자 프로세스에서 그룹에 포함된 각각의 프로세스들에게 메시지를 전송하는 프리미티브이다. msg는 실질적인 메시지이고, member\_id는 메시지의 수신하여야 하는 그룹내 프로세스들의 식별자를 의미한다.

**Receipt (msg, message\_buf) :**

각 그룹의 모든 프로세스들이 메시지를 수신하는 프리미티브이다. 수신된 msg를 message\_buf에 저장한다.

**Error (message\_buf\_num, member\_id) :**

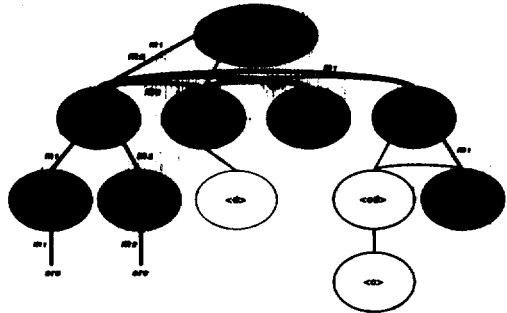
각 프로세스에서 올바른 메시지의 수신이 이루어지지 않았을 때, 사용하는 프리미티브이다. message\_buf\_num은 올바르게 수신하지 못한 메시지의 번호를 나타내고, member\_id는 멤버 프로세스의 식별자이다.

**Resend (msg, member\_id) :**

각 그룹의 대표자 프로세스에 Error 프리미티브가 수신되면, 대표자 프로세스에서 메시지 재전송을 위한 프리미티브이다.

이와 같이 위에서 기술한 프리미티브들을 가지고 제 1안된 가상위상에서 메시지를 전송한다. 먼저, 그룹간 메시지전송이 발생하는데, 이때 사용되는 프리미티브들은 Multicast, Receive 프리미티브를 이용한다. 메시

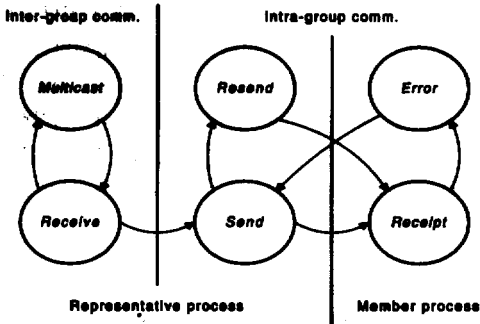
지의 시작은 가상위상에서 농도가 가장 작은 그룹에서부터 출발하고 메시지의 종착점은 메시지를 수신하여야 하는 그룹의 농도가 가장 큰 그룹에서 삭제 된다. 이와 같은 방법으로 메시지를 전송하면 다음과 같은 그림으로 메시지가 전송된다.



(그림 7) Multiph상에서 메시지 전송의 예  
(Fig. 7) Example of Message Transmission on Multiph

위의 (그림 7)은 (그림 6)에서 형성된 가상위상에서 다중전송 그룹 그룹 A에 포함되는 프로세스에게 메시지  $m_1$ 을 전송하고, 다중전송 그룹 B에 포함되는 프로세스에게는 메시지  $m_2$ 를 전송하려 할 때, 각 메타그룹의 변화를 나타낸 것이다. 전송자에서 출발한 메시지는 먼저 농도가 '1'인 메타 그룹 (a), (b)에 메시지가 도착한다. 메타그룹 (a)에 도착한 메시지  $m_1$ 은 다음 전송되어야 하는 메타그룹은 (ab)로 전송된다. 그리고  $m_2$ 도 메타그룹 (b)에서 메타그룹 (ab)로 전송된다. 메타그룹 (ab)에 도착한 메시지  $m_1$ 과  $m_2$ 는 메타그룹 (ab)의 대표자 프로세스에서 순서성이 결정된다. 두 개의 메시지에 대한 순서가 결정되면, 결정된 순서로 다음 메타그룹 (abd)로 전송된다. 이와 같이 전송되는 메시지들은 가상위상 Multiph의 간선중 트리 간선을 이용한 메시지 전송을 보인 것이다. 가상위상 Multiph에는 하이퍼 간선도 가지고 있는 구조로 메타그룹 (ab)에서 메타그룹 (ad)와 (acd)로는 메시지  $m_1$ 을 전송하고, 메타그룹 (bd)로는 메시지  $m_2$ 를 하이퍼 간선을 통해 전송한다. 메타그룹 (acd)에 도착한 메시지  $m_1$ 은 농도가 하나 작은 메타그룹 (ac)로 트리 간선을 이용해 메시지를 전송한다. (그림 7)에서 메시지가 소멸되는 메타그룹은 농도가 가장 큰 (abd)에서 소멸된다. 그리고, (ad)와 (ac)에서 메시지가 소멸된다. 이때 사용하는 프리미티브들은 Send, Receipt, Error,

Resend 프리미티브를 사용하게 되고, 이러한 프리미티브들에 의해서 메시지에 대한 순서성과 원자성이 보장되게 된다. 다음의 (그림 8)은 가상위상 *Multiph*의 메타그룹에 포함되는 프로세스의 프리미티브들에 따라 변화하는 프로세스의 상태를 표현한다.



(그림 8) 프로세스의 상태 전이도  
(Fig. 8) Transition Diagram of Process

5.2 그룹관리 프리미티브

동적인 통신망은 통신망에 연결된 프로세스들의 가입과 탈퇴가 자유롭게 이루어지는 통신망이다. 동적인 초고속통신망상에서 이루어지는 그룹통신이므로, 프로세스의 그룹으로 형성된 각 그룹들은 프로세스의 가입과 탈퇴가 자유롭게 이루어진다. 프로세스의 동적인 변화는 프로세스의 이동성과 보다 효율적인 작업 요구에 의해 발생한다. 이 때, 이미 형성된 가상위상의 형태를 많이 변형하지 않는 범위에서 그룹원의 가입과 탈퇴 그리고 새로운 그룹의 생성이나 삭제등이 가능하여야 한다. 이 절에서는 그룹의 동적인 관리를 지원하는 프리미티브들을 기술한다.

*Multiph*는 새로운 그룹이 생성되거나, 이미 형성된 그룹이 삭제가 발생하여도 이미 형성된 가상위상 환경 조에는 전체적인 변화는 없다. 그 이유는 *Multiph*에 참여하는 그룹들이 하나의 독립된 그룹으로 다른 그룹들과의 포함 관계를 갖지 않는 메타그룹이기 때문이다. 다음의 프리미티브들은 그룹에 참여를 원하는 프로세스나 그룹에서 탈퇴를 원하는 프로세스가 발생하였을 때, 이것을 처리하는 프리미티브를 나타낸다.

- Add (MG) : 그룹 MG에 가입을 원하는 프리미티브
- Create (New\_MG) : 새로운 그룹 New\_MG를 생성

하고자 하는 프리미티브

Delete (MG) : 그룹 MG에서 탈퇴를 원하는 프리미티브

가입을 원하는 프로세스는 가입하고자하는 그룹 (MG)의 대표자 프로세스에 가입을 요청한다. 새로운 그룹, *New\_MG*을 생성하고자 하는 프로세스는 수신하는 메시지의 수에 의해서 그룹의 이름이 결정되고, 그룹의 이름은 그룹의 농도를 의미한다. 그룹의 농도가 결정되면 농도가 하나 큰 그룹들 중 포함 관계에 있는 그룹을 선택한다. 선택한 그룹의 대표자 프로세스에게 새로운 그룹이 생성되었음을 알리고, 메시지를 주고 받을 수 있는 간선을 연결한다. 그리고, 형제 그룹들과의 하이퍼 간선을 연결한다. 만약, 포함관계가 아닌 배제 (exclusion) 관계에 있다면, 같은 농도를 갖는 가상위상의 임의의 그룹을 선정하고 그 그룹의 대표자 프로세스에게 새로운 그룹이 생성되었음을 알리고 간선을 연결한다. 그리고, 위에서와 같이 하이퍼 간선도 연결한다.

탈퇴를 원하는 프로세스는 탈퇴하고자하는 그룹 (MG)의 대표자 프로세스에 탈퇴를 요청한다. 탈퇴를 요청받은 그룹의 대표자 프로세스는 탈퇴를 원하는 프로세스가 삭제된 후에도 다른 멤버 프로세스를 보유하고 있다면 탈퇴를 원하는 프로세스만을 탈퇴시킨다. 하지만, 탈퇴를 원하는 프로세스를 탈퇴한 후 그 그룹의 프로세스가 존재하지 않을 경우에는 그 그룹의 자원을 가상위상상에서 제거하여야 한다. 삭제되는 그룹이 만약 농도가 '1'인 그룹이라면, 그 그룹은 가상위상에서 일정점에 해당하는 그룹이며, 메시지의 출발점이 되는 그룹이다. 이 경우에는 그룹을 완전히 그룹의 상위에 존재하는 그룹으로 삭제할 목적으로 연결된 간선만을 제거함으로써 그룹을 탈퇴시킨다.

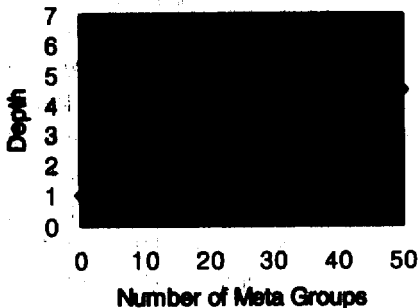
삭제를 위해서는 그룹의 농도가 '1'인 경우에는 그룹을 완전히 그룹의 상위에 존재하는 그룹의 상위에 존재하는 그룹으로 삭제할 목적으로 연결된 간선만을 제거하여야 하는 그룹의 대표자 프로세스에서 메시지를 수신하는 그룹들 중 하나의 그룹을 선정한다. 선정된 그룹의 이름을 상위 존재하는 그룹의 대표자 프로세스의 알림으로써 간선을 수정한다. 이 때, 그룹과 연결되었던 하이퍼 간선은 제거만이 있고, 더 이상의 새로운 하이퍼 간선을 추가하지 않는다.

6. 성능 평가

본 논문에서 제안하는 *Multiph*를 사용하는 메시지의 다중전송 그룹 전송에는 메시지 전송을 위해 사용하는 가상위상에서 농도가 '1'인 메타그룹에서 농도가 가장 큰 메타그룹까지의 거리를 나타내는  $d$ 에 의해서 메시지의 수와 네트워크 지연시간의 변화에 큰 영향을 미친다. 따라서 본 논문에서는 생성되는 *Multiph*의 일정점에서 뿌리 정점까지의 거리를  $d$ 라 할 때,  $d$ 와 *Multiph* 생성에서 다음 변수와의 관계를 주로 평가하였다.

먼저, 가상위상의 형성에 참여하는 메타그룹의 수 ( $M$ )가 증가함에 따라 생성되는 *Multiph*의 깊이( $d$ )를 측정하는 하였다. 메타그룹의 수가 10에서부터 50개로 그룹으로 증가할 때, 본 논문에서 제안하는 *Multiph* 생성 알고리즘에 의해서 생성되는 *Multiph*의 깊이를 실험하였다. 각 메타그룹은 50개의 프로세서들이 산재되어 있는 다중전송 그룹에서 생성되었고, 다중전송 그룹들의 겹쳐진 정도는 10으로 제한하였다. 이것은 주어진 다중전송 그룹에서 생성되는 메타그룹의 농도를 최대 10으로 제한하는 것과 같은 의미이다. 위와 같은 조건을 100번씩 반복하여서 얻은 결과를 토대로 다음(그림 9)과 같은 결과를 얻을 수 있었다.

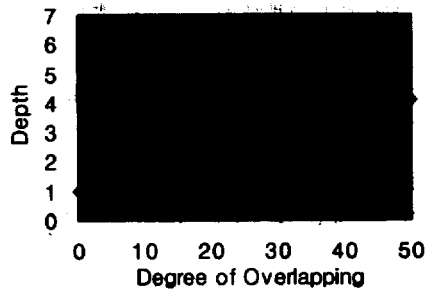
실험의 결과에서 보는 바와 같이 메타 그룹의 수가 증가하여도 형성되는 *Multiph*의 깊이에는 많은 변화가 발생하지 않으며 거의 같은 깊이로 형성됨을 알 수 있었다. 따라서 본 논문에서 제안하는 메시지 다중 전송을 위한 가상위상인 *Multiph*를 형성에는 깊이의 변화가 크지 않고 일정한 깊이를 갖는다는 것을 알 수 있었다.



(그림 9) 메타그룹 수와 깊이와의 관계  
(Fig. 9) Relation on Number of Meta-group vs. Depth

두 번째, 실험은 생성된 메타그룹의 중복도( $\lambda$ )가 변할 때, 생성되는 *Multiph*의 깊이( $d$ )와의 관계를 실험하였다.

실험을 위해 사용된 다중전송 그룹의 수는 50개의 멀티케스트 그룹을 사용하였다. 각 다중전송 그룹에 포함된 프로세스의 수는 50개로 고정하였다. 이때, 중복도를 10에서부터 50까지 변화하면서, 각 중복도마다 100번의 반복실험을 하였을 때, 다음의(그림 10)과 같은 결과를 얻을 수 있다. 아래의 실험 결과에서도 앞에서의 실험 결과와 비슷한 결과를 보였다. 메타그룹의 중복도의 증가는 메타 그룹에 포함되어있는 프로세스들이 포함된 메타 그룹의 수가 많다는 것을 의미하는 것으로 복잡한 메타 그룹을 의미한다. 이러한 상황에서도 생성되는 가상위상 *Multiph*의 수는 중복도의 증가와는 상관없이 일정한 깊이의 가상위상이 생성됨을 알 수 있었다.



(그림 10) 중복도와 깊이와의 관계  
(Fig. 10) Relation on Overlapping Degree vs. Depth

## 7. 결 론

본 논문에서는 동적인 초고속 통신망에서 가상위상을 이용한 그룹통신 플랫폼을 제안하였다. 특히 가상위상 *Multiph*를 제안하고, 제안된 가상위상에서의 통신과 그룹관리 프리미티브도 정의하였다. 가상위상 *Multiph*는 서로 중복된 프로세스 그룹들에 메시지를 전송하고자 할 때, 중복된 다중전송 그룹들을 서로 중복되지 않은 메타그룹으로 형성하고, 다중전송을 위해 메타그룹을 *Multiph* 구조로 형성하였다. 형성된 *Multiph*는 트리 구조와 유사한 구조를 가지고 있지만, 동일한 깊이에 존재하는 정점들간에 하이퍼 간선을 가지고 있는 구조이다.

가상위상에 대한 첫 번째 실험에서, *Multiph*를 형성하기 위해 참여하는 메타그룹의 수가 증가함에 따라

산술적인 비례가 아닌 로그리증적인  $Multiph$ 를 알 수 있었는데, 이것은 메타그룹의 수가 많아도 얻을 수 있는  $Multiph$ 이 깊어 변화는 매우 작음을 보여주는 것이다. 그리고 두 번째 실험에서는 멀티캐스트 그룹의 중복도를 증가하여도 생성되는 가상위상  $Multiph$ 의 깊이는 거의 변화가 없음을 보여주었는데, 이것도 또한 그룹의 중복도와 가상위상의 깊이와는 관계가 없음을 보여주는 것으로, 가상위상  $Multiph$ 는 메타그룹의 수에 의해서나 중복도에 의해서 깊이에 대한 변화가 없는 안정적인 가상위상을 생성함을 의미한다. 그리고, 프로세스의 가입과 탈퇴가 일어나는 동적인 그룹의 변화에 대하여도 프리미티브를 이용해 가입하고자 하거나 탈퇴하고자 하는 그룹의 대표자 프로세스와의 협의를 통해 전체적인 가상위상의 변화가 없이 지속적인 메시지 전송을 할 수 있었다.

앞으로의 연구 방향은 제안된  $Multiph$ 를 초고속통신망에서 구현하는 것이고, 구현된 가상위상위에서 본문에서 정의한 그룹통신 프리미티브를 이용한 실질적인 메시지 전송을 적용하는 것이다. 이와 함께 다중전송 시스템에서 그룹통신 계층없이 커뮤니케이션 프로토콜 계층에서 그룹통신을 지원할 수 있는 프로토콜에 대한 연구가 앞으로의 연구 방향이다.

### 참 고 문 헌

- [1] A. Schiper and M. Raynal, "Distributed systems," Communications of The ACM, vol. 39, no. 4, April. 1996.
- [2] Cambini, Riccardo Gallo, Giorgio Scutella', Macia Grazia, "Flows on Hypergraphs", Technical Report: TR-4-96, Stanford Univ.
- [3] D. R. Cheriton and W. Zwaenepoel, "Distributed process groups in the V kernel," ACM Trans. on Computer Systems, vol. 3, no. 2, pp. 77-107, May. 1985.
- [4] Hector Garcia-Molina, Annemarie Spitzer, "Ordered and Reliable Multicast Communication", ACM Trans. on Computer System, vol 9, no 3, pp 242-271, Aug. 1991
- [5] K. Birman, "The process group approach to reliable distributed computing," Commun. ACM, vol. 36, no. 12, pp. 37-53, Dec. 1993.
- [6] L. E. Moser, P. M. Melliar-Smith and V. Agrawala, "Membership algorithms for asynchronous distributed systems," The 11th Intn'l Confer. on Distributed Computing Systems, pp 480-488, May. 1991.
- [7] L. L. Peterson, N. C. Buchholz and R. D. Schlichting, "Preserving and using context information in interprocess communication," ACM Trans. on Computer Systems, vol. 7, no. 3, pp. 217-246, Aug. 1989.
- [8] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," Commun. ACM, vol 21, no 7, pp. 558-565, July. 1978.
- [9] M. F. Kaashoek and A. S. Tanenbaum, "Group communication in the AMOEBA distributed operating system," The 11th Intn'l Confer. on Distributed Computing Systems, pp. 222-230, May. 1991.
- [10] M. K. Reiter, "Distributing trust with the Rampport Toolkit," Communications of The ACM, vol. 39, no. 4, pp. 71-74, April. 1996.
- [11] S-W. Luan and V. D. Gligor, "A fault-tolerant protocol for atomic broadcast," IEEE Trans. Parallel Distributed Syst., vol 1, pp. 271-285, July. 1990.
- [12] T. P. Ng, "Ordered broadcast for large applications," in Proc. 10th Sysmp. Reliable Distrib. Syst., pp. 188-197, 1991.
- [13] Xiaohua Jia, "A total ordering mulitcast protocol using propagation trees," IEEE Trans. on Parallel and Distributed Syst., vol. 6, no 6, pp. 617-627, June. 1995.
- [14] Y. Amir, D. Dolve, S. Kramer and D. Malki, "Transis : A communication sub-system for high availability," Technical Report: ftcs-22, ps.gz, Hebrew Univ.
- [15] Luping Liang, Samuel T. Chanson, and Gerald W. Neufeld, "Process Group and Group Communication : Classification and Requirements," IEEE Computer, pp. 56-66, Feb. 1990.
- [16] 신용태, 이정훈, "대용량 통신 처리 시스템에서의 다자간회의를 위한 멀티캐스팅에 관한 연구", 한국

전자통신연구원 최종 연구 보고서, Dec. 1996.

[17] 조민수, 신용태, 이철희, "다중전송을 위한 멀티캐스트 그래프", 97년도 한국통신학회 추계학술발표회, pp.383-386, Nov. 1997.

### 신용태

1985년 한양대학교 산업공학과 졸업 (학사)

1990년 Univ. of Iowa, 전산학과 졸업 (석사)

1994년 Univ. of Iowa, 전산학과 졸업 (박사)

1994년 4~8 Univ. of Iowa 전산학과 객원교수

1994년 8~1995. 2 Michigan State University, 전산학과 전임강사

1995년~현재 숭실대학교 정보과학대학 컴퓨터학부 조교수

관심분야 : 컴퓨터 네트워크, 그룹통신, 분산 컴퓨팅, 인터넷 프로토콜, 초고속 통신망, 전자상거래 기술

### 조민수

1991년 숭실대학교 수학과 졸업 (학사)

1994년 숭실대학교 대학원 전자계산학과 졸업 (학사)

1997년 숭실대학교 대학원 전자계산학과 박사 수료

1997년~현재 경인여자대학 사무자동화과 전임강사

관심분야 : 컴퓨터네트워크, 그룹통신, 인터넷 프로토콜.