

## 다중 접근 네트워크에서의 새로운 멀티캐스트 기법

正會員 鄭 敏 奎\* 正會員 金 龍 珉\* 正會員 金 鍾 權\*

### A New Multicasting Method in a Multiple Access Network

Min Kyu Jeong\*, Yong Min Kim\*, Chong Kwon Kim\* *Regular Members*

#### 要 約

화상 회의시스템이나 컴퓨터 지원 협동 작업 시스템과 같은 고도 정보 통신 서비스는 멀티-파티 통신 기법이 필요하다. 따라서 이러한 서비스들을 제공하기 위해서 미래의 네트워크는 유연하고 효율적인 멀티-파티 통신을 지원할 수 있어야 한다. 본 논문에서는 물리적 망 계층과 논리적인 네트워크 계층에서 패킷을 여과하는 새로운 개념의 멀티캐스트 기법을 제안하였다. 이 새로운 방법은 멀티캐스트 통신의 주요 오버헤드인 패킷 전송 오버헤드와 패킷 처리 오버헤드를 줄일 수 있다는 가능성을 가지고 있다. 새로운 방법을 실제 패킷 방송 방식의 네트워크에 응용하기 위해, 여러 개의 목적지 호스트를 포함할 수 있는 적당한 그룹 주소집합을 찾는 그룹 매칭 알고리즘을 개발하였고, 두 개의 예를 들어 새로운 멀티캐스트 기법과 알고리즘을 설명하였다.

주요어 : 멀티캐스트, 방송, 그룹 주소, 그룹 매칭 알고리즘

#### ABSTRACT

Broadband ISDN will support advanced communication services such as multimedia conference, VOD and electronic news service. Because many of these advanced services require multimedia data delivery, future communication networks must have flexible and efficient multiparty communications capabilities. In this paper we propose a new multicasting method which uses packet filtering capabilities both in a physical network and in a logical network layers. This new scheme has potential to reduce the transmission and packet processing overheads of multicast communications. For the new multicasting method, we develop a group matching algorithm which finds a suitable set of groups that covers multiple target hosts. We show the application of the multicasting method and the group matching algorithm with two simple examples.

Keywords : Multicast, Group, Group Matching Algorithm

\*서울대학교 계산통계학과  
Dept. of Computer Science Statistics, Seoul National University  
論文番號 : 94165  
接受日字 : 1994年 6月 18日

## I. 서 론

VLSI와 광섬유 기술이 발전됨에 따라 고속 통신망이 가까운 장래에 실현될 전망이다. 고속 통신망은 화상 회의(desktop video conference)[12], 컴퓨터 지원 협동 작업(CSCW : computer supported cooperative work)[19], 고도화된 전자 우편(enhanced electronic mail), 분산된 데이터 베이스의 갱신과 같은 다양한 서비스를 지원할 것이다. 현재 널리 이용되고 있는 저속의 집대집 통신 서비스와는 달리, 새로운 형태의 통신 서비스는 동일한 정보를 한 곳에서 여러 곳으로 동시에 전송할 수 있는 멀티캐스트 통신 기능을 필요로 한다. 예를 들어, 화상 회의 서비스에서는 여러 사용자들이 동일한 비디오 화면을 볼 수 있어야 하므로 화상 데이터가 생성된 곳에서 회의에 참가하는 모든 사용자에게 화상 데이터를 전송해야 한다. 고도 정보 통신 서비스를 성공적으로 제공하기 위해서 미래의 통신 네트워크는 유연하고 효율적인 멀티캐스트 통신 기능을 가지고 있어야 한다.

멀티캐스트 기법에 관한 연구는 원거리 통신망(WAN : wide area network)에서의 라우팅(routing)[3, 15], 순서화(ordering)[1, 9], 그룹 멤버십(group membership)[2, 6], 그룹 주소 할당(group address allocation)[6]을 중심으로 활발히 이루어지고 있다. 그러나 근거리 통신망(LAN : local area network) 환경에서의 멀티캐스트 기법에 관한 연구는 미미한 실정이다. 본 논문에서는 근거리 통신망 환경에서의 멀티캐스트 기법에 관한 연구로서 다중 접근 네트워크에서 사용될 수 있는 새로운 멀티캐스트 기법을 제안하려 한다.

현재 사용되고 있는 대부분의 근거리 통신망은 이더넷(ethernet)[8, 16], 토큰 링(token ring)[8, 16], FDDI[8, 16], DQBD[10] 등과 같은 다중 접근 네트워크이다. 다중 접근 네트워크의 모든 호스트는 하나의 매체(media)를 공유하여 공유된 매체를 통해 전달되는 모든 프레임(frame)을 수신할 수 있다. 다중 접근 네트워크에서 전송되는 프레임은 물리적 매체를 통해 모든 호스트에서 전달되며 프레임을 수신한 호스트는 프레임의 헤더(header)에 있는 물리적 주소를 보고 자신이 받을 메시지의 여부를 결정한다. 결국 다중 접근 네트워크의 호스트는 모든 트래픽을 물리적 매체의 방송을 통해 전송한다.

본 논문에서는 현재 사용되는 여러 멀티캐스트 기법의 장단점을 살펴보고 다중 접근 네트워크에서 사용할 수 있는 새로운 개념의 멀티캐스트 기법을 제안한다. 본 논문에서 제안하는 멀티캐스트 기법은 물리적 망 계층의 그룹 통신 기능을 이용할 뿐만 아니라, 네트워크 계층에서의 패킷 여과 기능을 이용하여 효율적으로 멀티캐스트 기능을 지원한다. 새로운 방식은 다수의 집대집 통신 대신 다수의 호스트에게 한번에 전송할 수 있는 물리적 멀티캐스트 통신을 이용한다. 이 때 하나의 물리적 멀티캐스트 통신으로 그룹 내의 모든 호스트에게 전송할 수 없으면 여러 번의 물리적 멀티캐스트를 수행하여야 한다. 즉, 하나의 논리적 멀티캐스트를 여러 개의 물리적 멀티캐스트로 대응시킴으로써 멀티캐스트 기능을 제공한다. 만약 물리적 계층의 그룹이 적당하게 설정되어 있다면, 여러 개의 집대집 통신에서의 패킷 전송 오비헤드나 방송과 여과 방법을 사용할 때의 패킷 처리 오비헤드를 상당히 줄일 수 있다.

새롭게 제안된 멀티캐스트 기법을 위하여 본 논문에서는 그룹 주소 대응기(group address resolver)와 그룹 매칭 알고리즘(group matching algorithm)을 제안하였다. 그룹 주소 대응기는 특별한 기능을 하는 서버 호스트로서 그룹 매칭 알고리즘을 수행하여 하나의 논리적 그룹에 속하는 모든 호스트를 포함할 수 있는 물리적 그룹의 집합을 구하는 역할을 한다. 그룹 매칭 알고리즘은 행렬 축소 단계와 그래프 탐색 단계를 거쳐 최소 비용을 가지는 물리적 그룹 주소 집합을 구한다. 행렬 축소 단계에서는 물리적 그룹 주소에 대한 정보를 나타내는 행렬로부터 의미 없는 정보를 없애으로써 문제를 간단하게 만든다. 그래프 탐색 알고리즘은 축소된 행렬을 이용하여 최적의 물리적 그룹 주소 집합을 구한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 새로운 멀티캐스트 기법의 가정과 기본적인 개념 및 작동 과정을 소개했다. 3장에서는 행렬 축소 단계와 그래프 탐색 단계의 두 단계로 구성되어 있는 그룹 매칭 알고리즘을 소개하였다. 4장에서는 두 가지 예를 들어 본 논문에서 제안한 알고리즘을 설명하였으며, 5장에서는 결론을 제시하였다.

## II. 새로운 멀티캐스트 기법

### 1. 멀티캐스트 방법

멀티캐스트를 지원하는 방법은 멀티캐스트 패킷을

전송하는 방식에 따라 두가지로 나눌 수 있다. 먼저 점대점 통신을 여러번 수행하여 멀티캐스트 기능을 제공하는 방법이 있다. 점대점 통신을 이용하면 동일한 패킷을 여러 번 전송해야 하므로 트래픽을 증가시켜서 네트워크의 부하를 높인다. 멀티캐스트 패킷을 전송하는 호스트는 멀티캐스트 패킷을 여러 개의 동일한 패킷으로 복사해서 전송해야 하므로 패킷 처리 오버헤드 또한 매우 높다. 따라서 멀티캐스트 통신에 참여하고 있는 모든 호스트들이 패킷을 받을 때까지는 많은 지연 시간이 걸리게 된다.

다른 하나의 방법은 물리적 방송 기능을 이용하는 것이다.[5] 패킷을 멀티캐스트하고자 하는 호스트는 우선 네트워크에 패킷을 방송하며 네트워크에 있는 모든 호스트는 이 패킷을 수신한다. 멀티캐스트 패킷을 받은 호스트는, 목적지 주소가 자신을 가리킬 경우에만 이 패킷을 받아들인다. 이러한 여러 기능을 보통 네트워크 계층이나 상위 계층에서 소프트웨어적으로 처리된다. 물리적 방송 기능을 이용하면 네트워크에 존재하는 모든 호스트가 멀티캐스트 패킷을 여과해야 하는 오버헤드를 가지게 된다. 즉, 멀티캐스트 통신에 참여하지 않는 호스트일지라도 멀티캐스트 트래픽에 의해 큰 영향을 받게 된다. 멀티캐스트가 자주 일어나지 않는 환경에서는 이러한 오버헤드를 무시할 수 있지만 그룹 통신 트래픽이 많은 경우에는 상당한 성능 저하를 일으키게 된다.

2. 통신프로토콜과 멀티캐스트 통신

통신 프로토콜은 여러 복잡한 기능들을 가지고 있기 때문에 다양한 기능들을 계층 구조(layered architecture)로 나누어 설계한다. 현재 사용되고 있는 통신 프로토콜을 살펴 보더라도 대부분 계층적 구조로 구성되어 있으며, 프로토콜에 따라서 네 개 내지 일곱 개의 계층을 가지고 있다. OSI 참조 모델 [14]은 응용, 표현, 세션, 전송, 네트워크, 데이터 링크, 물리 계층의 일곱 계층으로 구성되어 있으며, DoD 모델[17]은 응용, 전송, 인터넷, 네트워크 집합 계층의 네 계층으로 구성되어 있다.

본 논문에서는 설명의 명료성을 위해서 통신 프로토콜의 구조를 상위 계층(upper layer), 네트워크 계층(network layer), 네트워크 인터페이스 계층(network interface layer)의 세 계층으로 나누었다(그림 1). 상위 계층은 터미널 모드 사이에서 수행되는 단대단(end-to-end) 기능을 수행하며, OSI 참조 모델의 응용, 표현, 세션, 전송 계층을 포함한다. OSI 참

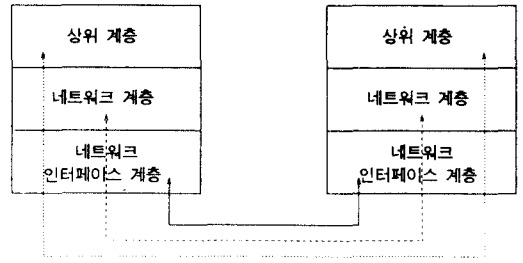


그림 1. 통신 프로토콜의 계층 구조  
Figure 1. layered architecture of communication protocol

조 모델과 동일한 네트워크 계층은 각 호스트에 유일한 논리적 주소(logical address)를 부여하여 라우팅 기능을 수행한다. 네트워크 인터페이스 계층은 각 호스트에 유일한 물리적 주소(physical address)를 부여하고 물리적 네트워크와 관련된 기능을 수행하는 계층이며, OSI 참조 모델의 데이터 링크 계층 및 물리적 계층이 이에 속한다.

상위 계층에서 멀티캐스트를 지원하는 방법에서는, 그룹을 표시하는 논리적 그룹 주소는 상위 계층이 관리하도록 하고 네트워크 계층 및 네트워크 인터페이스 계층의 방송 기능을 이용한다. 상위 계층의 멀티캐스트는 네트워크 계층 및 네트워크 인터페이스 계층의 방송으로 대응되며, 다중 점을 네트워크에 연결된 모든 호스트는 방송된 패킷을 수신하여 상위 계층으로 전달한다. 방송된 패킷을 전달받은 상위 계층은 자신의 그룹 주소와 패킷의 목적지 주소를 비교하여 패킷을 받아들이거나 제거한다. 이와 같이 상위 계층에서 멀티캐스트 기능을 수행하면, 모든 멀티캐스트 패킷이 호스트의 하위 계층(네트워크 인터페이스 계층, 네트워크 계층)을 통과하여야 하므로 커다란 오버헤드를 지불해야 한다. 특히, 네트워크 계층과 상위 계층은 보통 소프트웨어로 구현되므로 패킷 처리 오버헤드는 더욱 심각하다. 따라서 상위 계층보다는 네트워크 인터페이스 계층이나 네트워크 계층에서 멀티캐스트 기능을 지원하는 방법이 많이 쓰인다.

네트워크 계층에서 멀티캐스트를 지원한다면, 각 호스트는 자신을 표시할 수 있는 고유한 논리적 주소와 함께 자신이 속한 멀티캐스트 그룹을 표시하는 복수 개의 논리적 그룹 주소를 네트워크 계층에 가지고 있다. 이러한 논리적 그룹 주소는 필요에 따라 동적

으로 할당되어서 유연하게 관리한다. 어떤 호스트가 특정한 그룹으로 메시지를 전달하면, 먼저 논리적 목적지 그룹 주소를 포함하는 패킷을 만들어 네트워크 인터페이스 계층에 전송을 요청하고 네트워크 인터페이스 계층에서는 이 패킷을 네트워크에 방송한다. 모든 호스트의 네트워크 인터페이스 계층에서는 이 패킷을 수신하여 네트워크 계층에 전달한다. 패킷을 받은 네트워크 계층에서는 패킷의 목적지 주소가 가리키는 논리적 그룹에 자신이 포함될 경우에만 이 패킷을 상위 계층으로 전달한다.

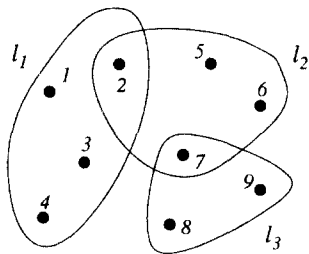


그림 2. 그룹 통신의 예  
Figure 2. example of group communications

네트워크 인터페이스 계층의 방송 기능을 이용하여 네트워크 계층에서 멀티캐스트 기능을 수행하는 방법을 예를 들어 설명하기로 하자. 그림 2에서 호스트 집합 {1, 2, 3, 4}, {2, 5, 6, 7}, {7, 8, 9}가 각각 논리적 그룹이며, 논리적 그룹 주소  $l_1, l_2, l_3$ 가 호스트 집합에 각각 할당되어 있다고 하자. 이 때, 호스트 1이 그룹 {1, 2, 3, 4}에게 패킷을 전송하려 한다면, 먼저 논리적 주소가  $l_1$ 인 패킷을 만든 후 이 패킷을 물리적으로 방송한다. 모든 호스트가 물리적으로 방송된 패킷을 받지만, 호스트 2, 3, 4만이 이 패킷을 네트워크 계층에서 여과하여 상위 계층으로 전달한다.

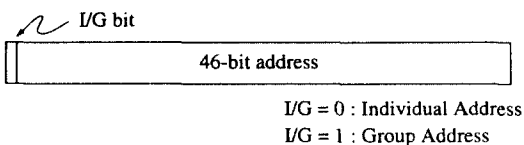


그림 3. IEEE의 멀티캐스트 주소  
Figure 3. multicast address format of IEEE

네트워크 인터페이스 계층에서 멀티캐스트 통신을 지원하는 경우도 있다. 실제로, IEEE에서 제정한 근거리 통신망은 멀티캐스트 통신을 지원한다. 네트워크 인터페이스 계층에서 사용하는 그룹의 주소를 물리적 주소라고 하는데, IEEE에서 지원하는 물리적 멀티캐스트 주소의 형식[16]은 그림 3과 같다. 네트워크 인터페이스 계층에서의 멀티캐스트는 네트워크 계층이나 상위 계층의 소프트웨어적 여과 과정을 거치지 않고 네트워크 인터페이스 계층의 하드웨어를 통해 여과 과정을 수행하므로 수행 속도를 빠르게 할 수 있다. 만약 네트워크 인터페이스 계층에서 사용하는 그룹이 목적지 호스트의 논리적 그룹과 정확하게 일치한다면, 단 한 번의 물리적 전송으로 논리적 그룹에 속하는 호스트들에게 메시지를 전달할 수 있다. 그러나, 네트워크 인터페이스 계층에서 사용하는 물리적 그룹 주소는 보통 ROM에 저장되어 있어서 동적으로 바꿀 수 없기 때문에, 가능한 모든 그룹에 대해 물리적 그룹 주소를 미리 만들어 네트워크 인터페이스 계층에 저장해 놓아야 한다. 그러므로 멀티캐스트 통신의 참여자가 자주 바뀌는 상황에서는 물리적 그룹 주소만을 사용하여 효율적이며 유연한 멀티캐스트 통신을 지원할 수 없다.

위에서 제시한 방법들의 단점을 보완하기 위해 본 논문에서는 네트워크 계층과 네트워크 인터페이스 계층에서 동시에 그룹 주소를 사용하는 새로운 방식의 멀티캐스트 기법을 제안한다. 네트워크 계층에서는 그룹 주소를 동적으로 변화시켜 유연하게 그룹을 관리할 수 있으며, 네트워크 인터페이스 계층에서 사용하는 그룹 주소는 패킷을 하드웨어적으로 빠르게 여과할 수 있어서 네트워크 계층의 오버헤드를 줄일 수 있다.

본 논문에서 네트워크 인터페이스 계층은 하나의 고유한 물리적 주소와 물리적 방송 주소, 몇 개의 그룹 주소로 이루어진 제한된 숫자의 물리적 주소를 ROM에 저장하고 있어서 동적으로 변화시킬 수 없다고 가정하였다. 호스트의 고유한 물리적 주소는 하나의 호스트만을 포함하는 물리적 그룹 주소로 볼 수 있으며, 물리적 방송 주소는 네트워크에 있는 모든 호스트를 포함하는 물리적 그룹 주소라고 할 수 있다. 반영구적으로 고정된 물리적 주소와는 달리, 소프트웨어로 구현되어 관리되는 네트워크 계층은 동적으로 할당되어 변화시킬 수 있는 논리적 그룹 주소를 제한없이 사용할 수 있다고 가정한다.

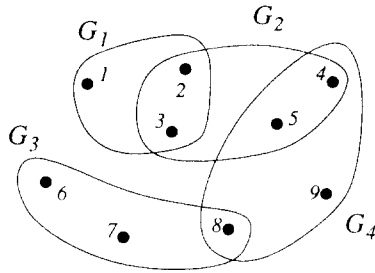


그림 4. 호스트와 물리적 그룹  
Figure 4. physical group addresses

예를 들어 네트워크에 아홉 개의 호스트가 있고 그림 4와 같이 물리적 그룹이 형성되어 있다고 하자. 호스트 그룹 1, 2, 3이 멀티캐스트 통신에 참여하고 있다고 하자. 호스트 1, 2, 3은 공통된 물리적 그룹 주소인  $G_1$ 을 가지고 있으므로 목적지의 물리적 주소를  $G_1$ 으로 만들어 패킷을 한 번만 전송하면 된다. 그리면 다중접속 네트워크의 특성상 모든 호스트는 패킷을 받게 되고 네트워크 인터페이스 계층에서 하드웨어적으로 여과되어 멀티캐스트 통신에 참여하고 있는 호스트에게만 패킷이 전달된다.

위의 예와 같이 멀티캐스트 통신에 참여한 호스트 그룹이 이미 형성된 물리적 멀티캐스트 그룹과 정확하게 일치하는 경우에는 단 한 번의 물리적 멀티캐스트를 수행하면 된다. 그러나, 호스트의 물리적 그룹 주소는 동적으로 바뀔 수 없고, 한정된 숫자의 물리적 그룹 주소만을 가지고 있기 때문에, 물리적 그룹 주소에 정확히 대응되는 물리적 그룹 주소가 존재한다는 보장이 없다. 그러므로 여러 개의 물리적 그룹 주소를 이용하여 적당히 논리적 그룹 주소와 대응시켜서 멀티캐스트 기능을 지원하는 방안을 본 논문에서는 제안하였다. 이 새로운 멀티캐스트 방법은 기존의 멀티캐스트 방법에 비해 전송 오버헤드와 패킷 처리 오버헤드를 상당히 줄일 수 있다는 장점을 가지고 있다.

그림 4와 같이 물리적 그룹이 설정되어 있고 호스트 집합 1, 2, 3, 4가 서로 통신을 하고자 하는 논리적 그룹이라면, 이 논리적인 그룹에 대해서 물리적 그룹  $G_1=1, 2, 3$ 과  $G_2=2, 3, 4, 5$ 를 대응시킬 수 있다. 논리적 그룹에 대해 두 개의 물리적 그룹을 대응시킴으로써, 호스트는 동일한 패킷을 서로 다른 물리적 그룹 주소를 사용하여 두 번만 전송하면 된다.

그러나 호스트 5는 자신이 속하지 않은 논리적 그룹의 패킷을 받게 되어, 이 패킷을 네트워크 계층에서 여과해서 제거해야 한다. 또한, 호스트 2와 3은 동일한 패킷을 두 번 받게 되는데, 두 번째 받은 패킷을 제거해야 한다. 이와 같이 자신과 관계없는 패킷이나 여러 번 전달된 동일한 패킷을 제거하는 오버헤드를 줄이기 위해서, 주어진 논리적 그룹에 대해 최소의 오버헤드를 갖는 물리적 그룹 주소집합을 찾아야 한다.

### 3. 구조(Architecture)

논리적 그룹에 대응하는 물리적 그룹 주소 집합을 찾기 위해서는 우선 네트워크에 어떤 물리적 그룹 주소가 있으며 각 물리적 그룹 주소는 어떤 호스트들을 포함하고 있는지를 알아야 한다. 호스트들과 물리적 그룹 주소간의 포화 관계를 이용하여 최소의 오버헤드를 갖는 물리적 그룹 주소를 찾는 것은 상당히 복잡하다. 또한 모든 호스트가 그룹 주소에 대한 정보를 유지하여 멀티캐스트 패킷을 전송하려 할 때마다 논리적 그룹 주소와 대응되는 물리적 그룹 주소 집합을 찾는 것은 부러가 있다. 호스트들이 직접 물리적 그룹 주소를 찾는 오버헤드를 줄이기 위해, 본 논문에서는 하나의 목적지 호스트 집합이 주어졌을 때, 여기에 대응되는 물리적 그룹 주소들을 찾을 수 있는 서버 호스트를 제안하였다. 이러한 서버 호스트를 그룹 주소 대응기(group address resolver)라고 부르며 논리적 그룹에 대응되는 물리적 그룹 주소의 검색을 그룹 주소 대응기에 요구하는 노드를 클라이언트(client)라 한다. 각 클라이언트는 그룹 주소 대응기의 주소를 알고 있으며 논리적인 멀티캐스트 그룹에 대응되는 물리적 그룹 주소가 필요한 경우에는 그룹 주소 대응기에 요구한다. 요구를 받은 그룹 주소 대응기는 대응되는 물리적 그룹 주소를 찾아 클라이언트에게 알려준다. 클라이언트는 대응되는 그룹

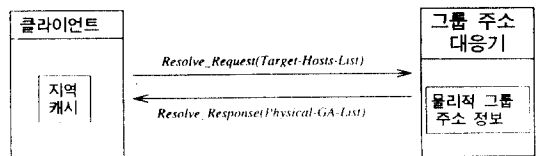


그림 5. 클라이언트와 그룹 주소 대응기  
Figure 5. client and group address resolver

주소를 패킷의 목적지 주소 필드에 적고 패킷을 전송한다. 그룹 주소 대응기에서 논리적인 그룹에 대응되는 물리적 그룹 주소 집합을 찾는 그룹 매칭 알고리즘은 3장에 제시되어 있다.

클라이언트가 그룹 통신을 하고자 하는 호스트 집합에 대응하는 물리적 그룹 주소 집합을 알고 싶은 경우에는 다음과 같은 메시지를 그룹 주소 대응기에 보낸다.(그림 5 참조)

*Resolve-Request(Target-Hosts-List)*

Target-Hosts-List는 그룹 통신을 하고자 하는 호스트들의 주소 집합이다. 그룹 주소 대응기는 Target-Hosts-List에 대응하는 물리적 그룹 주소 집합(Physical-GA-List)을 찾아 요구한 클라이언트에게 다음과 같은 메시지를 이용하여 알려준다.

*Resolve-Response(Physical-GA-List)*

위의 클라이언트와 그룹 주소 대응기간의 상호 동작은 TCP/IP 프로토콜의 ARP(Address Resolution Protocol)와 비슷하다. ARP에서 캐시를 사용하듯이 클라이언트는 그룹 주소 대응기로부터 받은 논리적 그룹과 물리적 그룹 주소에 대한 정보를 자신의 지역 캐시(local cache)에 유지한다. 동일한 논리적 그룹으로 보낼 패킷은 자신의 지역 캐시에 저장된 정보를 이용하여 전송하고, 지역 캐시에 없는 정보에 대해서만 그룹 주소 대응기에서 요청한다. ARP에서는 한 개의 IP 주소와 한 개의 물리적 주소를 대응시키는 반면 그룹 주소 대응기는 하나의 논리적 그룹 주소를 여러 개의 물리적 그룹 주소로 대응시킨다. 또한 ARP에서는 모든 호스트가 서버 역할을 하나 새로운 멀티캐스트 기법에서 서버 역할을 하는 그룹 주소 대응기는 다중 접근 네트워크에 하나만 존재하여 모든 요구를 처리한다.

Ⅲ. 그룹 매칭 알고리즘(Group Matching Algorithm)

멀티캐스트를 위해서 논리적 그룹을 물리적 그룹 주소로 대응시키는 문제는 최소 비용을 갖는 행렬의 열을 구하는 0/1 정수 프로그래밍(0/1 integer programming)의 집합 커버링 문제(set covering problem([11, 18])와 유사하다. 본 논문에서는 논리적 그

룹을 이루는 호스트를 포함하는 물리적 그룹 주소의 집합을 구하기 위해 행렬 축소 알고리즘[18]과 그래프 탐색 알고리즘의 두 단계로 구성되어 있는 그룹 매칭 알고리즘(Group Matching Algorithm)을 제안하였다. 이 장에서는 여러가지 가정과 용어에 대해 정의하며 이 알고리즘에 대한 설명과 행렬 축소 알고리즘에서 사용할 축소 규칙의 타당성을 증명하였다.

그룹 매칭 알고리즘은 행렬 축소 알고리즘과 그래프 탐색 알고리즘의 두 개의 부 알고리즘으로 구성되어 있다. 행렬 축소 알고리즘은 물리적 그룹에 대한 정보를 나타내는 행렬을 축소하면서 일부의 호스트를 포함하는 물리적 그룹 주소 집합을 구한다. 제한된 문제에서는 논리적 그룹의 모든 호스트를 포함하는 최적의 물리적 그룹 주소 집합을 구할 수도 있다. 모든 호스트를 포함하지 못할 경우에는 축소된 행렬을 이용하여 그래프 탐색 알고리즘을 수행한다. 행렬 축소 알고리즘과 그래프 탐색 알고리즘이 각각 구한 두 개의 물리적 그룹 주소 집합을 합하면 논리적 그룹의 모든 호스트를 포함하는 최소 비용의 물리적 그룹 주소 집합을 구할 수 있다.

1. 그룹매칭문제(Group Matching Problem)

본 논문에서 제안하는 멀티캐스트 기법은 여러 개의 공유 접근 네트워크 세그먼트가 브릿지(bridge)에 의해 연결된 인터넷 환경에서도 동작이 가능하나 설명의 명료성을 위해 하나의 세그먼트만이 존재한다고 가정하여 설명하도록 하자. 먼저 네트워크에  $n$ 개의 물리적 그룹 주소가 있으며 이를  $G_1, G_2, \dots, G_n$ 라고 하자. 이 물리적 그룹 주소는 각 호스트에 분배되어 ROM에 저장된다. 2.2절에서 설명한 바와 같이 물리적 그룹 주소는 물리적 호스트 주소와 물리적 방송 주소를 포함한다. 네트워크에  $m$ 개의 호스트  $\{1, 2, 3, \dots, m\}$ 가 있고, 각 호스트는 네트워크 계층 주소  $h_1, h_2, \dots, h_m$ 과 물리적 계층 주소  $p_1, p_2, \dots, p_m$ 을 각각 하나씩 가지고 있다. 고유한 물리적 계층 주소와 더불어, 호스트  $i$ 는  $g_{i1}, g_{i2}, \dots, g_{iq(i)}$ 라는  $q(i)$ 개의 물리적 그룹 주소를 네트워크 인터페이스 계층에 유지하며 각 호스트의 네트워크 인터페이스 계층은 멀티캐스트된 프레임의 물리적 그룹 주소와 자신의 물리적 그룹 주소중의 하나가 일치하면 그 프레임을 네트워크 계층으로 전달한다.

논리적 그룹 주소를 여러개의 물리적 그룹 주소와 대응시키는 문제를 명확하게 나타내기 위해 다음과 같은 기호를 사용한다.

- $a_{ij}$  : 호스트와 물리적 그룹 주소사이의 관계를 나타낸다. 물리적 그룹 주소  $G_i$ 가 호스트  $i$ 를 포함할 경우에  $a_{ij}$ 는 1이며, 그렇지 않을 경우에는  $a_{ij}=0$ 이다.
- $A_{i\cdot}=(a_{i1}, a_{i2}, \dots, a_{im})^T$  : 물리적 그룹 주소  $G_i$ 가 어떤 호스트들을 포함하는지를 나타내는 벡터이다.
- $A_{\cdot i}=(a_{1i}, a_{2i}, \dots, a_{mi})$  : 호스트  $i$ 가 어떤 그룹 주소에 포함되는지를 나타내는 벡터이다.
- $c=(c_1, c_2, \dots, c_n)^T$  : 물리적 그룹 주소에 대한 비용을 나타내는 비용 벡터이다. 벡터의 원소  $c_i$ 는 물리적 그룹 주소  $G_i$ 를 사용하여 패킷을 전송할 경우의 비용을 나타낸다. 물리적 그룹 주소  $G_i$ 를 사용할 때마다 네트워크에서 한번의 전송이 이루어지게 되며  $\|A_{i\cdot}\|$ 개의 호스트에서 여파가 일어나게 된다. 그러므로,  $\alpha$ 와  $\beta$ 를 각각 전송 비용과 여파 비용이라고 할 때,  $c_i=\alpha+\beta\|A_{i\cdot}\|$ 가 된다.
- $t=(t_1, t_2, \dots, t_m)^T$  : 논리적 그룹이 어떤 호스트들을 포함하고 있는지를 나타내는 목적 호스트 집합이다. 논리적 그룹에 호스트  $i$ 가 포함된다면  $t_i=1$ 이며 그렇지 않을 경우에는  $t_i=0$ 이다.
- $x=(x_1, x_2, \dots, x_n)^T$  : 그룹 매칭 문제에서 구하려고 하는 해를 나타낸다. 논리적 그룹을 나타내는 목적 호스트 집합과 대응되는 물리적 그룹 주소 집합에 물리적 그룹 주소  $G_i$ 가 포함될 경우에는  $x_i=1$ 이며 그렇지 않을 경우에는  $x_i=0$ 이다.
- 벡터  $u=(u_1, u_2, \dots, u_k)$ 와  $v=(v_1, v_2, \dots, v_k)$ 가 있을 때, 모든  $i$ 에 대하여  $u_i \geq v_i$ 일 경우에  $u \geq v$ 로 나타낸다.

물리적 그룹 주소  $G_i$ 는  $m$ 개의 원소를 갖는 벡터  $A_{i\cdot}$ 로 표시되므로  $n$ 개의 그룹 주소 벡터를 모아  $m \times n$  행렬  $A=(A_{1\cdot}, A_{2\cdot}, \dots, A_{n\cdot})$ 를 구성할 수 있으며, 행렬  $A$ 의  $(i, j)$  원소는  $a_{ij}$ 이다. 이와 같은 가정하에서  $Ax \geq t$ 인 벡터  $x$ 를 구하면, 벡터  $x$ 가 나타내는 그룹 주소들은 벡터  $t$ 가 나타내는 목적 호스트들을 포함하는 해이다. 그룹 매칭 문제에서는 여러 개의 해가 존재할 수 있는데, 만약 벡터  $x$ 의 총비용  $c^T x$ 가 다른 어떤 해의 총비용보다 크지 않다면 벡터  $x$ 는 그룹 매칭 알고리즘에서 구하는 최적해가 된다. 따라서 그룹 매칭 알고리즘에서는  $Ax \geq t$ 이면서 총비용  $c^T x$ 가 최소인 벡터  $x$ 를 구한다.

## 2. 행렬 축소 규칙

그룹 주소 대응기는 요구한 목적지 호스트들을 최소의 비용으로 포함할 수 있는 물리적 그룹 주소의 집합을 구해야 한다. 만약 네트워크에 많은 수의 호스트와 물리적 그룹 주소가 존재한다면 행렬  $A$ 가 매우 크기 때문에 행렬  $A$ 로부터 필요없는 행과 열을 제거함으로써 문제의 크기를 축소시킬 수 있다. 만약 호스트  $i$ 가 목적지 호스트에 포함되지 않는다면 행렬  $A$ 의 열  $A_{i\cdot}$ 를 제거할 수 있다(축소 규칙 1). 뿐만 아니라 만약 어떤 그룹 주소가 원하는 호스트를 하나도 커버하지 못하는 경우에는 이 그룹 주소를 제외시킬 수 있다. 즉,  $A_{i\cdot} \cdot t=0$ 이면  $A_{i\cdot}$ 를  $A$ 에서 제외시킬 수 있다(축소 규칙 2). 이와 같은 두 가지의 간단한 규칙과 아래에서 제시될 세가지의 규칙에 의해서 행렬을 축소시킬 수 있다. 이렇게 행렬 축소 작업은 문제의 크기를 줄여서 다음에 적용될 그래프 탐색 단계를 간단하게 한다.

(규칙 1) 호스트  $i$ 가 목적지 호스트가 아니라면, 행렬  $A$ 로부터  $A_{i\cdot}$ 을 제거하고 벡터  $t$ 로부터  $t_i$ 를 제거한다.

(규칙 2)  $A_{i\cdot}$ 가 0벡터라면, 행렬  $A$ 로부터  $A_{i\cdot}$ 를 제거하고 벡터  $c$ 로부터  $c_i$ 를 제거한다.

(규칙 3)  $A_{i\cdot}=e^s$ 이라면 ( $e^s$ 는  $s$ 번째 원소만이 1이고 나머지 원소는 모두 0인  $n$ 개의 원소를 갖는 벡터이다), 다음과 같은 작업을 수행한다.

1. 물리적 그룹  $G_i$ 를 구하는 해에 포함시킨다( $x_i \leftarrow 1$ ).
2.  $a_{ij}=1$ 인 모든  $j$ 에 대해, 행렬  $A$ 에서  $A_{j\cdot}$ 를 제거하고 벡터  $t$ 에서  $t_j$ 를 제거한다.
3. 행렬  $A$ 에서  $A_{i\cdot}$ 를 제거하고 벡터  $c$ 에서  $c_i$ 를 제거한다.

(규칙 4)  $A_{i\cdot} \geq A_{k\cdot}$ 이라면, 행렬  $A$ 에서  $A_{i\cdot}$ 를 제거하고 벡터  $t$ 에서  $t_i$ 를 제거한다.

(규칙 5)  $A_{i\cdot} \leq A_{k\cdot}$ 이고  $c_i \geq c_k$ 이라면, 행렬  $A$ 에서  $A_{i\cdot}$ 를 제거하고 벡터  $c$ 에서  $c_i$ 를 제거한다.

규칙 3은 어떤 목적지 호스트를 포함하는 그룹이 하나뿐이라면, 그 그룹을 무조건 해집합에 포함시켜야 함을 뜻한다. 하나의 그룹을 선택하고 나면, 선택된 그룹이 포함하고 있는 다른 호스트들도 제거할 수도 있다. 규칙 4는 만약  $A_{i\cdot}$ 가  $A_{k\cdot}$ 보다 크거나 같다면,  $A_{i\cdot}$ 를 제거할 수 있다. 이것은 호스트  $k$ 를 포함하는 물리적 그룹 주소는 호스트  $i$ 도 당연히 포함한다는 의미이다. 규칙 5는 더 적은 수의 목적지 호스트를 포함하면서 비용이 더 큰 물리적 그룹 주소는 선택할

필요가 없다는 의미이다.

축소 규칙을 행렬 A에 적용하여 행  $A_k$ 과 열  $A_i$ 를 제거했다면, 축소된 결과는

$$\begin{aligned} A'_{i \cdot} &= (a_{i1}, \dots, a_{i(l-1)}, a_{i(l+1)}, \dots, a_{in}) \\ A'_{\cdot j} &= (a_{1j}, \dots, a_{(k-1)j}, a_{(k+1)j}, \dots, a_{mj})^T \\ c' &= (c_1, c_{l-1}, c_{l+1}, \dots, c_n)^T \\ t' &= (t_1, t_{k-1}, t_{k+1}, \dots, t_m)^T \\ A' &= (A'_{\cdot 1}, A'_{\cdot 2}, \dots, A'_{\cdot (l-1)}, A'_{\cdot (l+1)}, \dots, A'_{\cdot n}) \end{aligned}$$

이 된다.

### 3. 행렬 축소 규칙의 타당성

규칙 1, 2와 5는 명백하기 때문에 규칙 3과 4의 증명만을 제시하고자 한다. 축소 규칙들의 타당성은 축소된 행렬을 사용해도 원래 행렬의 최적해를 구할 수 있다는 것을 보이면 된다. 정리 1은 축소 규칙 3을 행렬 A에 적용하여 얻은 축소 행렬 A'의 최적해에 그룹 주소 G를 포함시키면 원래 행렬의 최적해가 된다는 것을 증명한다. 정리 2는 축소 규칙 4를 행렬 A에 적용하여 축소된 행렬 A'을 얻었다면, A'에서 얻은 최적해는 원래의 행렬 A의 최적해가 된다는 것을 증명한다.

(정리1) 행렬 A에 규칙 3을 적용하여 축소된 행렬이 A'이라 하고,  $x=(x_1, x_2, \dots, x_n)^T$ 와  $y=(y_1, y_2, \dots, y_{n-1})^T$ 가 각각 행렬 A와 A'의 최적해라고 하자. 이 때, 벡터 y'을 아래와 같이 정의하면 y'은 행렬 A의 최적해가 된다.

$$\begin{aligned} y' &= (y'_1=y_1, y'_2=y_2, \dots, y'_{s-1}=y_{s-1}, y'_s=1, \\ & y'_{s+1}=y_s, \dots, y'_n=y_{n-1})^T \end{aligned}$$

(증명)

벡터 y'이 행렬 A의 해가 됨은 명백하며 이 때, x와 y의 관계는 다음의 두가지 경우이다.

1.  $x=y'$ 라면,  $c^T x = c^T y'$ 임은 명백하다.
2.  $x \neq y'$ 라면,
  - (a)  $c^T x > c^T y'$ 라면, x는 행렬 A의 최적해가 아니므로 x가 A의 최적해라는 가정에 위배된다.
  - (b)  $c^T x < c^T y'$ 일 경우.

벡터 x'을 ( $x'_1=x_1, x'_2=x_2, \dots, x'_{s-1}=x_{s-1}, x'_s=x'_s, \dots, x'_{n-1}=x_n$ )<sup>T</sup>라고 하면,  $c^T x' = c^T y'$ 임은 명백하다. 그리고  $x \neq y'$ 이라는 가정에 의해서  $x' \neq y'$ 이며,  $A'x' \geq t'$ 이므로 x'은 행렬

A'의 해가 된다. 그러면, y보다 총비용이 더 작은 해 x'이 존재하므로 y는 행렬 A'의 최적해가 아니다. 이것은 y가 행렬 A'이 최적해라는 가정에 위배된다.

y'이 행렬 A의 해가 되며  $c^T x = c^T y'$ 이므로 y'은 축소 규칙 3을 적용하기 전의 행렬 A의 최적해이다.

(정리 2) 행렬 A에 규칙 4를 적용하여 축소된 행렬이 A'이라 하면, 행렬 A'의 최적해에도 또한 행렬 A의 최적해이다.

(증명)

행렬 A와 A'의 최적해를 각각 x와 y라고 하면  $Ax \geq t$ 이며 따라서  $A'x \geq t'$ 이다. 또한  $A'y \geq t'$ 이고  $A'y \geq A_k \cdot y$ 이므로  $Ay \geq t$ 가 된다. 따라서 벡터 y는 행렬 A의 해가 된다. 이제 A와 A'에 대한 두가지 문제의 최적해가 동일하다( $c^T x = c^T y$ )는 것을 증명하면 된다. x와 y의 관계는 다음의 두가지 경우로 나누어 생각할 수 있다.

1.  $x=y$ 라면,  $c^T x = c^T y$ 이다.
2.  $x \neq y$ 일 경우.

- (a)  $c^T x < c^T y$ 라면, y는 행렬 A'의 최적해가 아니다. 이것은 가정에 위배된다.
- (b)  $c^T x > c^T y$ 라면, x는 행렬 A의 최적해가 아니다. 이것은 가정에 위배된다.

벡터 y는 행렬 A의 해이며  $c^T x = c^T y$ 이므로 y는 행렬 A의 최적해가 된다.

### 4. 축소 알고리즘

그룹 주소 대응기가 그룹통신을 하고자 하는 호스트로부터 목적지 호스트에 대한 정보를 받으면 최소비용의 물리적 그룹 주소 집합을 찾기 위해 다음과 같은 축소 알고리즘을 적용한다.

t, A, c가 주어졌을 때,

단계 0 :  $x \leftarrow 0$

단계 1 : 각 행에 규칙 1을 적용한다.

단계 2 : 각 열에 규칙 2를 적용한다.

단계 3 : 규칙 3이 적용될 수 있다면,

수행한 후, 단계 1로 간다.

단계 4 : 모든 행에 대해 규칙 4를 적용한다.

단계 5 : 모든 열에 대해 규칙 5를 적용한다.

단계 6 :  $A' \leftarrow$  축소된 행렬

$A' = \emptyset$ 이거나  $A' = A$ 이면, 종료.

단계 7 :  $A \leftarrow A', c \leftarrow c', t \leftarrow t'$

단계 1로 간다.

위에서 제시한 축소 알고리즘의 수행 시간은 호스



트의 수  $m$ , 물리적 그룹 주소의 갯수  $n$ 에 의존한다. 파라미터인  $m$ 과  $n$ 은 모두 1보다 큰 정수이다. 축소 알고리즘의 단계별로 수행 시간을 계산해 보면, 먼저 단계 1에서는  $O(m)$ 이라는 시간이 걸린다. 단계 2~5까지는 각각  $O(mn)$ ,  $O(mn)$ ,  $O(m'n)$ ,  $O(mn^2)$ 이 걸리며 단계 6에서는  $O(mn)$ 이라는 시간이 걸린다. 축소 알고리즘은 단계 1에서 7까지 부표를 돌게 되는데 루프의 횟수는 최악의 경우에  $O(m+n)$ 가 된다. 따라서 이것을 모두 계산해 보면, 축소 알고리즘은 최악의 경우  $O((m'n+mn^2)(m+n)) \leq O(m'n^2)$ 이라는 시간이 걸리게 된다.

### 5. 그래프 탐색 알고리즘

제한된 문제에서는 3.4절에서 설명한 행렬 축소 알고리즘만을 사용하여 최적해를 구할 수 있으나( $A' = \emptyset$ 의 조건으로 축소 알고리즘이 종료된 경우), 많은 경우에 있어서 축소 알고리즘을 해를 구하지 못한다( $A' = A$ 의 조건으로 축소 알고리즘이 종료된 경우). 따라서 목적지 호스트 그룹을 완전히 포함하는 물리적 그룹 주소 집합을 구하기 위해서는 축소 알고리즘에 의해 축소된 행렬  $A'$ 의 최적해도 구해야 한다. 그래프 탐색 알고리즘을 축소 알고리즘의 결과인  $A'$ ,  $c'$ ,  $t'$ 에 적용하여  $A'$ 의 최적해를 구한다. 그래프 탐색 알고리즘은 인공 지능 분야의 생산 시스템(production systems)을 풀기 위한 휴리스틱(heuristic) 알고리즘인 A\* 알고리즘과 비슷하다. A\* 알고리즘은 노드(node)와 에지(edge)로 구성된 그래프에서 목표 노드(goal node)를 찾는 그래프 탐색 알고리즘이다[4, 7, 13].

A\* 알고리즘은 그래프의 특정 노드  $u$ 에 대한 평가 함수인  $f(u) = g(u) + h(u)$ 를 계산하여 가장 작은 평가 함수값을 갖는 노드를 선택한다. 여기서  $g(u)$ 는 노드  $u$ 의 비용을 나타내며  $h(u)$ 는 노드  $u$ 에서 목표 노드까지 남은 예상 비용을 나타낸다. 만약, 예상 비용  $h(u)$ 를 실제 노드  $u$ 에서 목표 노드까지의 남은 비용보다 작게 예상할 수 있다면 A\* 알고리즘은 항상 최적해를 찾을 수 있다. A\* 알고리즘의 탐색 시간은 평가 함수가 얼마나 정확하기에 달려 있는데, 평가 함수가 실제 비용을 보다 정확히 예상할 수 있다면 탐색 시간은 줄어든다.

본 논문의 그래프 탐색 알고리즘은 탐색 과정에서 여러 개의 노드와 에지로 구성된 그래프를 생성한다. 각 노드는 A의 물리적 그룹에 대한 특정한 조합을 나타내는 벡터이고 최적해에 대한 그래프 탐색 알고

리즘의 추정값이다. 만약  $n$ 개의 물리적 그룹이 있다면  $2^n$ 개의 조합이 가능하므로 그래프는 최대  $2^n$ 개의 노드를 가질 수 있다. 하지만 A\* 알고리즘과 같이 평가 함수를 사용하면 일부의 노드만을 생성하여 최적해를 찾을 수 있다. 탐색을 시작하는 초기 노드는 0 벡터이며, 초기 노드의 자식(children) 노드인  $n$ 개의 노드는 절대값이 1인 벡터이다. 일반적으로 노드  $q$ 가 노드  $p$ 의 자식 노드라면,  $q > p$  이고  $|q| = |p| + 1$ 이다.

본 논문에서 찾으려고 하는 목표 노드는 최소 비용으로 모든 목적 호스트를 포함할 수 있는 물리적 그룹 주소를 나타내는 벡터이다. 평가 함수의 두 요소인  $g(u)$ 와  $h(u)$ 는 다음과 같이 결정된다. 함수  $g(u)$ 는 노드  $u$ 의 비용을 나타내므로  $c^T u$ 이다. 예상 비용  $h(u)$ 는  $u$ 가 모든 목적 호스트를 포함하면  $h(u) = 0$ 이고, 그렇지 않으면  $h(u) = \alpha + \beta |t' - u|$ 이다.

집합  $X$ 를 그래프 탐색 알고리즘에서 아직 탐색하지 않은 노드들의 집합이라고 하면, 초기에  $X$ 는 0 벡터만을 가진 집합이 된다. 그래프 탐색 알고리즘은 아래와 같은 단계를 거쳐 수행된다.

단계 0 :  $X \leftarrow \{0\}$ ,  $n \leftarrow 0$

단계 1 : 노드  $n$ 을 확장한다.

단계 2 :  $X \leftarrow (X \cup \{n\}) \cap$  확장된 노드들

단계 3 :  $n \leftarrow$  집합  $\{u | u \in X, f(u) \leq f(v) \text{ for all } v \in X\}$ 의 임의의 원소

단계 4 : 만약  $n$ 이 모든 행을 커버할 수 있으면,

종료하고 그렇지 않으면, 단계 1로 간다.

위의 알고리즘에서 노드를 확장한다는 의미는 자신의 노드에서 하위 노드를 생성하는 것을 의미하고 이미 생성된 노드와 동일한 노드는 생성하지 않는다. 이 그래프 탐색 알고리즘은, 구하는 목표 노드의 실제 비용보다 큰 비용을 갖는 노드는 확장하지 않는다.

위에서 제시한 그래프 탐색 알고리즘의 수행 시간은 노드의 확장 횟수로서 결정된다. 축소 알고리즘의 결과로  $n$ 개의 물리적 그룹이 남아있다면  $2^n$ 개의 조합이 가능하므로 확장될 수 있는 노드는 모두  $2^n$ 개이다. 따라서, 그래프 탐색 알고리즘의 수행 시간은 최악의 경우  $O(2^n)$ 이다.

### IV. 새로운 멀티캐스트 방법의 예

이 장에서는, 두 가지 간단한 예제를 이용하여 그룹 배정 알고리즘을 설명하려 한다. 첫번째 예에서는

그래프 탐색 알고리즘을 적용하지 않고 행렬 축소 알고리즘만을 적용하여 최적해를 구할 수 있는 경우이다. 두번째 예는 행렬 축소 알고리즘을 적용하여 최적해를 구하지 못하기 때문에, 행렬 축소 알고리즘의 결과를 이용하여 그래프 탐색 알고리즘을 적용하는 경우이다.

1. 예 1

이 예에서는 그림 6과 같이 다섯 개의 호스트와 여섯 개의 그룹 주소가 있는 경우이다. 전송 비용과 여과 비용이  $\alpha=3, \beta=1$ 이라면, 비용 벡터는  $c=(6, 5, 5, 5, 5, 6)^T$  이 된다. 목적지 호스트 벡터가  $t=(0, 1, 1, 1, 0)^T$ 로 주어졌다고 하면, 이러한 정보를 이용해서 그룹 주소 대응기는 먼저 축소 알고리즘을 수행한다.

$$x = (0, 0, 0, 0, 0, 0)^T, c = (6, 5, 5, 5, 5, 6)^T$$

	A <sub>•1</sub>	A <sub>•2</sub>	A <sub>•3</sub>	A <sub>•4</sub>	A <sub>•5</sub>	A <sub>•6</sub>
A <sub>1•</sub>	1	0	1	0	0	1
A <sub>2•</sub>	0	1	0	1	0	0
A <sub>3•</sub>	1	0	1	1	1	0
A <sub>4•</sub>	0	1	0	0	0	1
A <sub>5•</sub>	1	0	0	0	1	1

그림 6. 예 1 : 초기 행렬  
Figure 6. example 1 : initial matrix

$$x = (0, 0, 0, 0, 0, 0)^T, c = (6, 5, 5, 5, 5, 6)^T$$

	A <sub>•1</sub>	A <sub>•2</sub>	A <sub>•3</sub>	A <sub>•4</sub>	A <sub>•5</sub>	A <sub>•6</sub>
A <sub>2•</sub>	0	1	0	1	0	0
A <sub>3•</sub>	1	0	1	1	1	0
A <sub>4•</sub>	0	1	0	0	0	1

그림 7. 예 1 : 첫번째 루프 단계 1의 결과 행렬  
Figure 7. example 1 : resulting matrix after the step 1 of the first loop

$$x = (0, 0, 0, 0, 0, 0)^T, c = (5, 5)^T$$

	A <sub>•2</sub>	A <sub>•4</sub>
A <sub>2•</sub>	1	1
A <sub>3•</sub>	0	1
A <sub>4•</sub>	1	0

그림 8. 예 1 : 첫번째 루프의 결과 행렬  
Figure 8. example 1 : resulting matrix after the first loop

먼저 단계 1에서는, 호스트 1과 5가 t에 포함되지 않았기 때문에 A<sub>1•</sub>, A<sub>5•</sub>를 제거할 수 있으며, 그 결과가 그림 7의 행렬이다. 다음의 단계 2, 3, 4에서는 적용되는 규칙이 없다. 단계 5에서, A<sub>•3</sub> ≤ A<sub>•4</sub>이고 c<sub>3</sub> ≥ c<sub>4</sub>이므로 규칙 5를 적용하여 A<sub>•1</sub>를 제거할 수 있다. 계속해서, A<sub>•3</sub> ≤ A<sub>•4</sub>, c<sub>3</sub> ≥ c<sub>4</sub>이고, A<sub>•5</sub> ≤ A<sub>•4</sub>, c<sub>5</sub> ≥ c<sub>4</sub>이고, A<sub>•6</sub> ≤ A<sub>•4</sub>, c<sub>6</sub> ≥ c<sub>4</sub> 이므로 규칙 5를 적용함으로써 A<sub>•3</sub>, A<sub>•5</sub>, A<sub>•6</sub>를 제거할 수 있다. 이로써 그림 8의 행렬을 얻을 수 있다. 단계 6에서 A' ≠ ∅, A' ≠ A이므로, 단계 7에서 A ← A', c ← c', t ← t'의 연산을 수행하고 단계 1로 간다.

$$x = (0, 0, 0, 1, 0, 0)^T, c = (5)^T$$

	A <sub>•2</sub>
A <sub>4•</sub>	1

그림 9. 예 1 : 두번째 루프의 결과 행렬  
Figure 9. example 1 : resulting matrix after the second loop

두번째 루프의 단계 1과 2에서는 적용되는 규칙이 없으며 단계 3에서, A<sub>4•</sub>가 1인 원소가 오직 한 개뿐이므로 규칙 3을 적용할 수 있고, 그림 9의 행렬을 얻게 된다. 이제,  $x=(0, 0, 0, 1, 0, 0)^T$ 이며 단계 1로 다시간다. 세번째 루프의 단계 1과 2에서는 적용될 수 있는 규칙이 없다. 단계 3에서는 A<sub>1•</sub>에 규칙 3이 적용될 수 있으며, 그 결과로  $x=(0, 1, 0, 1, 0, 0)^T$ 를

연계 단계 1로 간다. 네번째 루프의 단계 1~5에서는 적용되는 규칙이 없으며, 단계 6에서  $A' = \emptyset$ 라는 조건으로 축소 알고리즘이 종료된다.

행렬 축소 알고리즘이  $A' = \emptyset$ 라는 조건으로 종료되었기 때문에  $x = (0, 1, 0, 1, 0, 0)^T$ 가 구하는 최적해가 되며, 호스트 2, 3, 4를 포함하는 최소 비용이 그룹 주소 집합은  $\{G_2, G_4\}$ 가 된다.

2. 예 2

그림 10처럼 다섯 개의 호스트와 여섯 개의 그룹 주소가 있다고 하자. 목적 호스트는  $t = (0, 1, 1, 1, 0, 0)^T$ 이고,  $\alpha = 3, \beta = 1$ 이라 하자. 이때, 그룹 주소의 비용은  $c = (6, 5, 5, 5, 5, 7)^T$ 이다.

$$x = (0, 0, 0, 0, 0, 0)^T, c = (6, 5, 5, 5, 5, 7)^T$$

	$A_{\bullet 1}$	$A_{\bullet 2}$	$A_{\bullet 3}$	$A_{\bullet 4}$	$A_{\bullet 5}$	$A_{\bullet 6}$
$A_{1\bullet}$	1	0	1	0	0	1
$A_{2\bullet}$	0	1	0	1	0	0
$A_{3\bullet}$	1	0	1	1	1	1
$A_{4\bullet}$	0	1	0	0	0	1
$A_{5\bullet}$	1	0	0	0	1	1

그림 10. 예 2 : 초기 행렬  
Figure 10. example 2 : initial matrix

$$x = (0, 0, 0, 0, 0, 0)^T, c = (6, 5, 5, 5, 5, 7)^T$$

	$A_{\bullet 1}$	$A_{\bullet 2}$	$A_{\bullet 3}$	$A_{\bullet 4}$	$A_{\bullet 5}$	$A_{\bullet 6}$
$A_{2\bullet}$	0	1	0	1	0	0
$A_{3\bullet}$	1	0	1	1	1	1
$A_{4\bullet}$	0	1	0	0	0	1

그림 11. 예 2 : 첫번째 루프 단계 1의 결과 행렬  
Figure 11. example 2 : resulting matrix after the step 1 of the first loop

$$x = (0, 0, 0, 0, 0, 0)^T, c = (5, 5, 5, 7)^T$$

	$A_{\bullet 2}$	$A_{\bullet 4}$	$A_{\bullet 6}$
$A_{2\bullet}$	1	1	0
$A_{3\bullet}$	0	1	1
$A_{4\bullet}$	1	0	1

그림 12. 예 2 : 축소 알고리즘의 결과 행렬  
Figure 12. example 2 : resulting matrix of the reduction algorithm

먼저, 축소 알고리즘의 단계 1에서 호스트 1과 5가 목적 호스트 t에 속하지 않으므로  $A_{1\bullet}$ 과  $A_{5\bullet}$ 을 제거할 수 있다(그림 11). 단계 2, 3, 4에서는 적용되는 규칙이 없으며, 단계 5에서는  $A_{\bullet 1} \leq A_{\bullet 3}$ 이고  $c_1 \geq c_3$ 이므로  $A_{\bullet 1}$ 을 제거할 수 있다. 또한,  $A_{\bullet 3} \leq A_{\bullet 4}$ 이고  $c_3 \geq c_4$ 이고,  $A_{\bullet 2} \leq A_{\bullet 4}$ 이고  $c_2 \geq c_4$ 이므로,  $A_{\bullet 2}$ 과  $A_{\bullet 5}$ 를 제거할 수 있다. 그림 12에 그 결과 행렬이 있다. 단계 6에서는  $A' = \emptyset$ 이고  $A' = A$ 이므로 단계 1로 간다. 두번째 루프의 단계 1~5에서는 적용되는 규칙이 없고, 단계 6에서  $A' = A$ 의 조건으로 축소 알고리즘이 종료된다. 벡터 x가 0벡터이므로 행렬 축소 알고리즘에서 구한 그룹 주소 집합은  $\{G_2\}$ 이 된다.  $A' = A$ 이라는 조건은 행렬 축소 알고리즘에서 구한 x가 최적해가 아니라는 것을 뜻하기 때문에 축소된 행렬을 가지고 그래프 탐색 알고리즘을 적용한다.

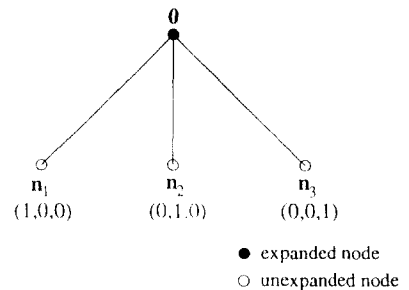


그림 13. 첫번째 루프의 결과 그래프  
Figure 13. resulting graph of the first loop

표 1. 평가 함수값의 계산

Table 1. calculation of the evaluation function

u	g(u)	h(u)	f(u)
n <sub>1</sub>	5	4	9
n <sub>2</sub>	5	4	9
n <sub>3</sub>	7	4	11

그래프 탐색 알고리즘의 단계 0에서,  $X=\{0\}$ 이고  $n=0$ 이다. 단계 1과 2에서 노드 0을 확장하여 생성된 노드  $n_1=(1, 0, 0)$ ,  $n_2=(0, 1, 0)$ ,  $n_3=(0, 0, 1)$ 는  $X$ 에 첨가되고 노드 0는  $X$ 에서 제거된다. 이 결과를 나타내는 그래프가 그림 13에 나타나 있다. 단계 3에서, 최소 평가 함수값을 갖는 노드를 결정하기 위해  $X$ 에 있는 노드들의 평가 함수값을 표 1과 같이 계산한다.

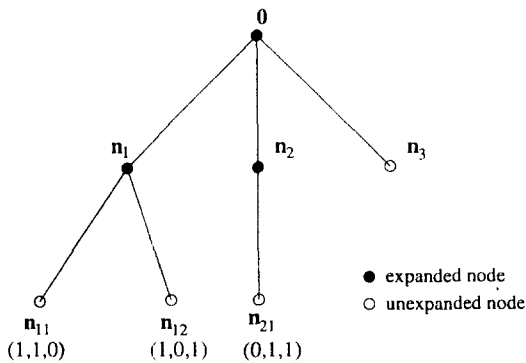


그림 14. 세번째 루프의 결과 그래프  
Figure 14. resulting graph of the second loop

표 1에서 최소의 평가 함수값을 갖는 노드는  $n_1$ ,  $n_2$  되며 같은 값의 평가 함수값을 가지면 임의로 선택한다. 만약  $n_1$ 을 선택하였다고 하면, 단계 4에서 모든 목적 호스트를 커버하지 못하기 때문에 단계 1로 간다. 두번째 루프의 단계 1에서 노드  $n_1$ 이 확장되며 두 개의 자식 노드  $n_{11}=(1, 1, 0)$ ,  $n_{12}=(1, 0, 1)$ 가 생성된다. 단계 2에서  $X$ 는  $\{n_2, n_3, n_{11}, n_{12}\}$ 가 된다. 단계 3에서 평가 함수값을 계산하여  $f(n_{11})=10$ ,  $f(n_{12})=12$ 를 얻으며, 노드  $n_2$ 가 최소값을 가지므로  $n_2$ 를 선택한다. 이러한 선택과 확장 과정을 계속하면, 그림 14와 같은 그래프를 얻게 된다. 이제  $n_{11}$ 이 최소의 평가 함수값을 가지므로  $n_{11}$ 을 선택하고  $n_{11}$ 이 행렬의

모든 행을 커버하므로 그래프 탐색 알고리즘은 종료된다.

그래프 탐색 알고리즘이 구한  $(1, 1, 0)$ 이라는 해는 물리적 그룹 주소 집합  $\{G_2, G_4\}$ 를 의미한다. 행렬 축소 알고리즘을 통해 이미 구한 그룹 주소 집합은  $\{ \}$ 이므로 두 집합을 합한  $\{G_2, G_4\}$ 가 호스트  $\{2, 3, 4\}$ 를 포함하는 최소 비용의 그룹 주소 집합이 된다.

### V. 결 론

고속 통신망은 화상 회의, 컴퓨터 지원 협동 작업, 고도화된 전자메일, 분산된 데이터 베이스의 갱신과 같은 서비스를 제공한다. 고속의 통신 기술을 이용하여 제공되는 고도 서비스는 한 장소에서 여러 장소로 정보를 전송하는 멀티캐스트 통신 기능을 포함하고 있다. 멀티캐스트 기법에 관한 연구는 원거리 통신망에서의 라우팅, 순서화, 가상 동기화, 그룹 멤버십, 그룹 주소 할당을 중심으로 이루어지고 있으나 근거리 통신망 환경에서의 멀티캐스트 기법에 관한 연구는 미미한 실정이다. 본 논문에서는 근거리 통신망 환경에서의 멀티캐스트 기법에 관한 연구로서 다중 접근 네트워크에서 사용될 새로운 멀티캐스트 기법을 제안하였다.

본 논문에서는 동적으로 변하는 그룹에 대한 멀티캐스트 통신을 효율적으로 지원하기 위해 네트워크 인터페이스 계층에서 빠른 처리 속도의 물리적 멀티캐스트 기능을 사용하고 네트워크 계층에서는 동적인 변화가 가능한 논리적 그룹 주소를 이용하는 새로운 멀티캐스트 기법을 제안하였다. 네트워크의 모든 호스트는 각각 유일한 물리적 주소, 물리적 방송 주소, 반영구적으로 할당된 여러 개의 물리적 그룹 주소를 가지고 있다. 호스트의 네트워크 인터페이스 계층에서는 패킷의 물리적 목적지 주소가 자신이 있는 물리적 주소중의 하나와 일치할 경우에만 패킷을 네트워크 계층으로 전달한다.

여러 목적지로 메시지를 전달하기 위해서는 패킷 전송 오버헤드와 패킷 처리 오버헤드를 최소화할 수 있는 물리적 그룹 주소를 선택해야 한다. 최소 비용을 갖는 물리적 그룹 주소를 찾으려면 네트워크에 어떤 호스트와 물리적 그룹 주소가 있으며 각 물리적 그룹 주소는 어떤 호스트들을 포함하는지를 알아야 한다. 모든 호스트가 물리적 그룹 주소에 대한 정보를 유지하기는 현실적으로 어렵기 때문에, 물리적 그룹 주소에 대한 정보를 유지하는 특수한 호스트인 그

룹 주소 대응기를 두었다. 그룹 주소 대응기는 물리적 그룹 주소에 대한 정보를 이용하여 주어질 논리적 그룹에 대응하는 최소 비용의 물리적 그룹 주소 집합을 찾는 그룹 매칭 알고리즘을 수행한다.

그룹 매칭 알고리즘은 행렬 축소 알고리즘, 그래프 탐색 알고리즘의 두 단계로 구성되어 있다. 많은 수의 호스트와 물리적 그룹 주소를 가지고 있는 환경에서는 문제의 크기가 매우 크기 때문에, 필요없는 요소를 없애으로써 문제의 크기를 줄이는 것이 중요하다. 본 논문에서는 5가지의 간단한 축소 규칙을 제시했으며 이 축소 규칙들이 타당함을 보였다. 많은 경우에 있어서, 이런 축소 규칙들을 반복적으로 적용하여 최적해(최소 비용해)를 구할 수 있으나, 어떤 경우에는 축소 단계에서 문제를 해결할 수 없는 경우가 있다. 축소 단계에서 최적해를 구하지 못하면 그래프 탐색 알고리즘을 사용한다. 그래프 탐색 알고리즘에서는 탐색 시간을 줄이기 위해 평가 함수를 제시하였으며 최소의 평가 함수값을 갖는 노드를 먼저 탐색함으로써 빠르게 최적해를 찾을 수 있다. 본 논문에서 제안한 그룹 매칭 알고리즘의 이해를 돕기 위해서, 축소 알고리즘이 최적해를 찾는 경우와 행렬 축소 알고리즘과 함께 그래프 탐색 알고리즘을 사용하는 두 가지의 예를 보였다.

본 논문에서 제안한 멀티캐스트 기법은 물리적 그룹 주소를 네트워크 인터페이스 계층이 사용하여 빠른 속도로 멀티캐스트 패킷을 처리할 수 있도록 하였다. 또 참여자가 자주 바뀌는 멀티캐스트 통신을 위해 동적으로 변화시킬 수 있는 논리적 그룹 주소를 네트워크 계층에서 사용하였다. 논리적 그룹 주소와 물리적 그룹 주소의 상충을 결합함으로써 유연하고 효율적인 멀티캐스트 기능을 사용자에게 제공할 수 있다.

### 참 고 문 헌

1. R. Aiello, E. Pagani, and G.P. Rossio. Causal Ordering in Reliable Group Communications. In *ACM SIGCOMM*, pages 106-115, September 1993.
2. Yair Amir, Danny Dolev, and Dalia Maliki Shlomo Kramer. Transis: A communication subsystem for high availability. Computer Science Department, The Hebrew University of Jerusalem, Jerusalem, Israel, 1992.
3. Tony Ballardie, Paul Francis, and Jon Crowcroft. Core Based Trees(CBT), An Architecture for Scalable Inter-domain Multicast Routing. In *ACM SIGCOMM*, pages 85-95, 1993.
4. Avron Barr and Edward A. Feigenbaum. *The Handbook of Artificial Intelligence*, volume I. William Kaufmann Inc., 1981.
5. J. Crowcroft K. Paliwoda. A Multicast Transport Protocol. In *ACM SIGCOMM*, pages 247-256, August 1988.
6. Alexandros Eleftheriadis, Sassan Pejhan, and Dimitris Anastassion. Multicast Group Address Management and Connection Control for Multiparty Applications. Department of Electrical Engineering and Center for Telecommunication Research, Columbia University, New York, April 1993.
7. Alberto Martelli. Under Complexity of Admissible Search Algorithm. *Artificial Intelligence : An International Journal*, 8, October 1976.
8. James Martin. *Local Area Networks, Architectures and Implementations*, chapter 6, 8, 9, pages 81-96, 113-140. Prentice Hall International, Inc., 1989.
9. Erwin Mayer. An Evaluation Framework for Multicast Ordering Protocols. In *ACM SIGCOMM*, pages 177-187, August 1992.
10. B. Mukherjee and S. Banerjee. Alternative Strategies for Improving the Fairness in and an Analytical Model of DQDB Networks. In *Proceedings of IEEE INFOCOM*, pages 879-888, 1991.
11. G.L. Nemhauser and A.H.G. Rinnooy Kan. *Handbooks in Operations Research and Management Science*, volume I, chapter 6. Elsevier Science Publishing B.V., 1989.
12. T. Ohmori, K. Maeno, S. Sakata, H. Fukuoka, and K. Watabe. Cooperative Control for Sharing Applications Based on Distributed Multiparty Desktop Conferencing System : MERMAID. In *International Conference on Communications*, pages 1069-1075, 1992.
13. Judea Pearl. *Heuristics : Intelligent Search Strategies for Computer Problem Solving*. Addison Wesley

ley, 1984.

14. Marshall T. Rose. *The Open Book, A Practical Perspective on OSI*. Prentice-Hall International, Inc., 1990.

15. Nachum Shacham. Multicast Routing of Hierarchical Data. In *International Conference of Communications*, pages 1217-1221, 1992.

16. William Stallings. *Handbook of Computer Communications Standards*, volume 1, chapter 2, pages 30-55. Prentice-Hall International, Inc., 2nd edition, 1990.

17. William Stallings. *Data and Computer Communications*. Maxwell Macmillan International Editions, 3rd edition, 1991.

18. Maciej M. Syslo, Narsingh Deo, and Janusz S. Kowalik, *Discrete Optimization Algorithm with Pascal Programs*, chapter 1. Prentice-Hall Inc., 1st edition, 1983.

19. Paul Wilson. Computer Supported Cooperative Work(CSCW) : Origins, Concepts and Research Initiatives. *Computer Networks and ISDN Systems*, pages 91-95. 1991.



鄭 敏 奎 (Min Kyu Jeong) 정회원  
1970년 2월 20일 생  
1993년 : 서울대학교 계산통계학과 졸업(학사)  
1993년 ~ 현재 : 서울대학교 계산통계학과 석사과정 재학중



金 龍 珉 (Yong Min Kim) 정회원  
1964년 7월 7일 생  
1989년 : 전남대학교 전산통계학과 졸업(학사)  
1991년 : 전남대학교 전산통계학과 졸업(석사)  
1993년 ~ 현재 : 서울대학교 계산통계학과 박사과정 재학중



金 鍾 權 (Chong Kwon Kim) 정회원  
1958년 2월 2일 생  
1981년 : 서울대학교 산업공학과 졸업(학사)  
1982년 : 미국 조지아 공대 졸업(O. R. 석사)  
1987년 : 미국 일리노이 대학 전산과학과 졸업(박사)

1984년 ~ 1985년 : IBM 샌 호세 연구소 연구조원  
1987년 ~ 1991년 : 미국 벨 통신 연구소 연구원  
1991년 ~ 현재 : 서울대학교 계산통계학과 조교수