

# 안정된 양방향 멀티캐스트 전송 프로토콜의 설계

김 은 기<sup>†</sup> · 나 근 식<sup>††</sup>

## 요 약

본 논문은 안정된 멀티캐스트 데이터 전송 프로토콜의 설계에 관한 연구이다. 그룹에 가입한 모든 노드는 하나의 멀티캐스트 링크를 통하여 데이터를 송신 및 수신할 수 있다. 전체 그룹은 몇 개의 부그룹으로 구성되며, 각 부그룹에는 자신의 멤버들에게 안정된 데이터 전송을 지원하는 DN 노드가 존재한다. 사용자는 기존의 그룹 멤버에 영향을 미치지 않고 간단한 과정을 통하여 그룹에 가입 및 탈퇴를 할 수 있다.

본 논문에서 연구된 MM-RMTP 프로토콜은 멀티캐스트 그룹 멤버들을 연결하는 링크의 재 구성이 간단하며, 멀티캐스트 기능을 지원하는 라우터를 필요로 하지 않는다. 또한, 많은 송신 노드의 존재로 인하여 발생할 수 있는 ACK 급격 증가 문제가 발생하지 않는 장점을 갖는다.

## Design of Reliable Bi-directional Multicast Transfer Protocol

Eun Gi Kim<sup>†</sup> · Guen Sik Na<sup>††</sup>

### ABSTRACT

In this study, a reliable multicast transfer protocols are designed. Any node in a group can transmit and receive data packet through the single multicast link. A multicast group is split into a few sub-groups, and each group has a DN node which support reliable transfer of data to its sub-group members. Join and leave operation to the group is simple, and this operation does not affect to the existing members.

Reconfiguration operation of multicasting tree in the designed MM-RMTP protocol is simple, and multicast capable router does not needed. ACK explosion problem that can be occurred according to the increase of sending node is avoided.

### 1. 서 론

데이터 전송 방식 중 멀티캐스트(multicast) 방식은 동일한 데이터를 여러 호스트들에게 전송하는 방식이다. 멀티캐스트 전송 프로토콜의 사용자는 한 번의 데이터 전송 요청을 수행하여 여러 수신자들에게 효과적으로 데이터를 전송할 수 있는 장점을 갖는다. 따라서 멀티캐스트 전송 프로토콜은 원격 강의와 CSCW 등과

같이 여러 사용자를 대상으로 하는 통신 응용에서 많이 사용될 것이다.

멀티캐스트 데이터 전송에 관한 관심은 오래 전에 시작되어 인터넷 IP 프로토콜에서는 클래스 D의 주소들을 멀티캐스트 용으로 할당하였다.<sup>(8, 11)</sup> 인터넷에서는 멀티캐스트 기능을 지원하는 라우터와<sup>(8)</sup> IGMP, RIP 등을 이용하는 'IP 멀티캐스트'라는<sup>(8)</sup> 방식이 사용되고 있다. IP 멀티캐스트 방식에서 사용되는 라우팅 프로토콜들은 DVMRP<sup>(12)</sup>, MOSPF<sup>(10)</sup> 등이 있으며, 각 호스트는 일부의 소프트웨어를 추가하거나 수정하는 방식으로 멀티캐스트 서비스를 지원 받을 수 있다.<sup>(8)</sup>

† 정 회 원 : 대전산업대학교 정보통신공학과 교수

†† 정 회 원 : 한신대학교 전자계산학과 교수

논문접수 : 1997년 11월 24일, 심사완료 : 1998년 6월 10일

조장기에 개발된 멀티캐스트 프로토콜들은 송신측에서 데이터를 전송하고 수신측으로부터 모든 패킷에 대한 ACK를 수신하는 방식을 사용함으로써 송신측의 로드를 증가시키거나 ACK 급격 증가(ACK explosion) 현상 등의 문제점이 발견되었다.<sup>[13]</sup> 이후에 개발된 프로토콜들은, 수신측에서 데이터가 올바르게 수신되지 못한 경우에만 NAK를 전송하는 방식을 사용하여 ACK 이상 증가 현상을 줄일 수 있도록 하였다.<sup>[13]</sup> 그러나, 이러한 경우에도 많은 수신 노드에서 에러가 발생한 경우에는 NAK 급격 증가 문제가 발생할 수 있으며, 또한, 송신측은 재전송에 대비하여 버퍼에 저장중인 데이터 패킷을 버퍼에서 제거할 정확한 시간을 알지 못하게 되는 단점을 갖는다. 이러한 문제점을 해결하기 위하여 수신측에서 주기적으로 ACK 패킷을 전송하는 방식을 사용하는 프로토콜들이 개발되었다[5]. 이후에는 전체 그룹이 여러개의 부그룹들로 구성되고 부그룹들이 트리 형태로 연결된 구조를 갖는 프로토콜이 개발되었으며, 이들 중 RMTP는[1] 여러가지 장점을 갖는 프로토콜로 좋은 평가를 받고 있다.

그러나, RMTP를 포함한 지금까지 개발된 멀티캐스트 프로토콜들은 하나의 송신 노드를 가정하여 개발되어졌다. 즉, 하나의 송신 노드를 중심으로 멀티캐스트 트리가 구성되고, 사용자 데이터는 이 트리 경로를 따라서 전송되는 방식이다. 둘 이상의 노드에서 데이터를 전송하는 경우에는 각 송신 노드마다 독립적인 멀티캐스트 트리가 구성되어야 한다. 이러한 방식은 트리를 구성하는데 필요한 과정을 고려해 볼 때 효율적인 방법이라고 할 수 없으며, 또한 그룹 멤버의 가입 및 탈퇴에 따라서 구성된 트리가 동적으로 재구성될 때도 많은 오버헤드를 갖는다.

본 연구에서 개발된 MM-RMTP(Multipoint-to-multipoint Reliable Multicast Transfer Protocol)에서는 그룹에 가입한 모든 노드들이 데이터를 송신 및 수신할 수 있으며, 데이터는 하나의 멀티캐스트 링크를 통하여 송수신된다. 멀티캐스트 링크는 가입한 노드의 주소 값에 따라서 구성된다. 본 프로토콜이 인터넷에 사용될 경우, 노드의 주소는 IP 주소를 사용할 수 있다.

2장에서 MM-RMTP의 개요에 관하여 설명하고, 3장에서는 데이터 송수신 방식에 관하여 기술한다. 4장에서는 그룹에의 가입 및 탈퇴에 필요한 과정을 기술하고, 5장에서 부그룹 관리 방식에 관하여 기술한다. 마

지막으로 6장에서 결론 및 문제점에 관하여 기술한다.

## 2. MM-RMTP 개요

MM-RMTP 는 신뢰성 있는 멀티캐스트 프로토콜로 다음과 같은 특성을 가진다.

- 멀티캐스트 그룹내의 어떤 노드도 데이터를 자유롭게 송수신 할 수 있다.
- 동적인 환경을 지원하기 위해 멀티캐스트 트리 구성을 위한 오버헤드를 최소화 한다.
- 멀티캐스트 그룹에 가입 및 탈퇴를 자유롭게 하며, 가입 및 탈퇴로 인한 오버헤드를 최소화 한다.
- 성능 최적화를 위해 그룹 병합 및 분리를 동적으로 할 수 있게 하고, 이로 인한 오버헤드를 최소화 한다.

MM-RMTP 에서 멀티캐스트 그룹은 노드나 부그룹들로 구성되고, 부그룹은 다시 노드나 하위 부그룹들로 구성된다. 하나의 부그룹에는 하나의 DN (Designated Node) 이 있다. DN 은 부그룹에 속하는 모든 노드를 관리하며, 한 부그룹과 다른 부그룹과의 통신을 담당한다. 이를 위해 DN 은 그 부그룹에 속하는 모든 멤버 노드들의 주소와 다른 DN 들의 주소를 가지고 있다.

멀티캐스트 그룹에 속하는 모든 노드는 유일한 네트워크 주소를 가지고 있고, 네트워크 주소는 순서있는 값이다. 그리고 부그룹들 사이에도 순서 관계가 있다. 이때 부그룹과 DN, 부그룹내의 멤버 노드들의 주소에는 다음과 같은 관계가 있다.

조건 1 : 모든  $i, j$  에 대해  $\text{Addr}(\text{DN}_i) > \text{Addr}(N_{ij})$  이다.

조건 2 :  $\text{Addr}(\text{Pred}(\text{DN}_i)) < \text{Addr}(\text{DN}_i) < \text{Addr}(\text{Succ}(\text{DN}_i))$  이다.

$\text{DN}_i$  : 부그룹  $i$  의 DN

$N_{ij}$  : 부그룹  $i$  의  $j$  노드

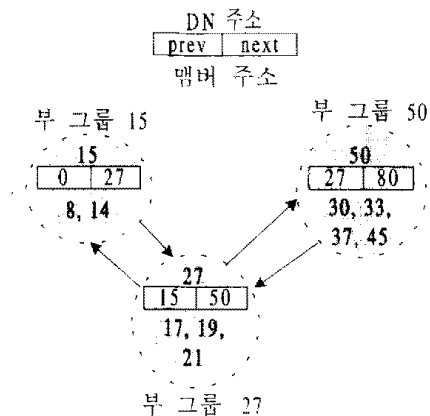
$\text{Pred}(\text{DN}_i)$  : 부그룹  $i$  의 앞 부그룹 DN

$\text{Succ}(\text{DN}_i)$  : 부그룹  $i$  의 뒤 부그룹 DN

$\text{Addr}(N)$  : 노드  $N$  의 네트워크 주소

조건 1은 한 부그룹의 DN 은 그 부그룹 멤버 노드

중에서 가장 큰 주소값을 가진다. 이것을 의미한다. 이 조건을 한 부그룹내에서 DN 설정을 쉽게 해 준다. 조건2는 부그룹간에 선후 관계가 있다는 것을 의미한다. 이러한 부그룹들간의 선후 관계는 트리 구성을 간편하게 한다. 그림1은 부그룹들이 연결된 멀티캐스팅 링크의 구조를 보여 준다.



(그림 1) 멀티캐스트 링크의 예  
(Fig. 1) Example of multicast link

어떤 노드가 멀티캐스트 그룹에 가입하거나 탈퇴하더라도 조건1 과 조건2 는 만족되어야 한다. 따라서 한 노드가 멀티캐스트 그룹에 가입하거나 탈퇴할 때 그 노드의 주소값에 따라 멀티캐스트 트리 구조는 동적으로 변할 수 있지만 그 변화는 하나의 부그룹내에 국한되어 처리되거나, 최대 두개의 부그룹만이 영향을 받는다. 즉, 가입하는 노드가 부그룹의 일반 멤버가 되는 경우에는 해당 그룹내에서 모든 처리가 이루어지지만, 가입하는 노드가 어떤 부그룹의 DN 이 되거나, 부그룹의 DN 이 부그룹에서 탈퇴하는 경우에 인접 부그룹의 정보가 갱신된다.

하나의 부그룹에 너무 많은 멤버가 가입해서 DN의 부하가 증가되면 하나의 부그룹은 여러개의 부그룹으로 분리될 수 있다. 반대로 몇 개의 부그룹들에 속하는 멤버수가 작을 때는 이 부그룹들을 하나의 부그룹으로 병합하므로써 종단간 데이터 전송 지연을 줄일 수 있다. 그러나 부그룹의 분리 및 병합시에도 다른 부그룹들은 영향을 받지 않는다.

DN은 자신의 이전과 이후에 있는 DN에게 자신의 부그룹에 있는 멤버의 수에 관한 정보를 주기적으로 전송한다.

### 3. 데이터 송신 및 수신

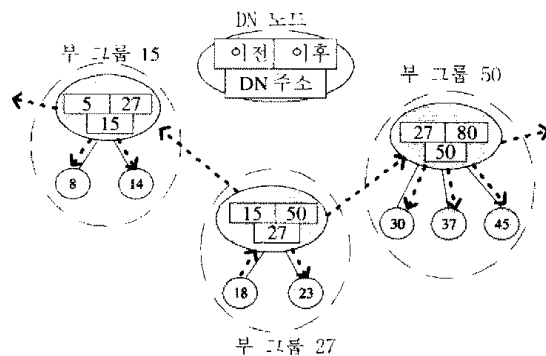
데이터를 전송하려는 노드  $N_i$ (또는  $DN_i$ )는 자신의 부그룹을 담당하는  $DN_i$ 에게 데이터 패킷을 전송하며,  $DN_i$ 과 부그룹 멤버들은 주기적으로 ACK 패킷을 교환하여 ACK 압력 증가 현상을 방지한다.

데이터 패킷을 수신한  $DN_i$ 의 동작은 다음과 같다.

- 데이터 패킷을 자신의 버퍼에 저장하며, 송신측이 DN인 경우에는 즉시 ACK를 전송한다. 송신측이 부그룹 멤버인 경우에는 주기적인 다음 ACK를 보낼 때 까지 ACK의 전송을 지연한다. ACK 패킷은 올바르게 수신된 패킷들 중에서 가장 큰 패킷의 번호를 갖는다.
- $\{Pred(DN_i), \text{부그룹 멤버들}, Succ(DN_i)\}$  중 패킷이 송신된 곳을 제외한 다른 곳에 선택적 재전송 방식으로 데이터 패킷을 전송한다.
- 모든 ACK를 수신한 후 패킷을 버퍼에서 제거한다.

하나의 부그룹이 계층적으로 구성된 경우에도, 각 DN은 자신이 직접 관리하는 멤버들에게 안정된 데이터 전송 기능을 지원한다.

그림 2는 18번 노드가 전송한 데이터가 멀티캐스트 링크를 따라 그룹 멤버들에게 전송되는 예를 보여준다.



(그림 2) 데이터 전송 예  
(Fig. 2) Example of data transfer

### 4. 그룹에의 가입 및 탈퇴

#### 4.1 그룹에의 가입

그룹에 가입하려는 노드  $N_j$ 는  $JoinRequest(Addr(N_j))$ 를 방송(broadcasting) 방식으로 전송하

고,  $DN_i$ 로부터  $Addr(DN_i)$  받으면 그룹 가입이 완료 된다. 이때, JoinRequest를 수신한 DN들 중에서 조건 3을 만족하는 DN이  $DN_i$ 가 되어  $N_j$ 를 자신의 부그룹 멤버에 포함시킨다.

조건 3 :  $Addr(Pred(DN)) \langle Addr(N_j) \langle Addr(DN)$  이다.

JoinRequest를 수신한 DN 중에서 조건 3을 만족하지 않는 DN들은 JointRequest가 해당 DN에 올바르게 도착할 수 있도록 하기 위하여 JoinRequest를 중계할 수 있다. 그러나, 이러한 경우에도 중계되는 JoinRequest의 수가 많아지면 불필요한 트래픽이 증가하게 된다. JoinRequest를 수신한 DN 중에서 조건 3을 만족하지 않는 DN들은 그룹 멤버들의 분포에 따라서 다음과 같은 동작이 가능하다.

- JoinRequest를 무시(discard): 좁은 지역의 밀집된 분포인 경우
- JoinRequest를 해당 DN 쪽으로 중계: 넓은 지역의 드문드문한 분포인 경우

$DN_i$ 가  $N_j$ 를 자신의 부그룹 멤버에 포함하는데 필요한 동작은 다음과 같다.

- $DN_i$  부그룹이 단일 부그룹인 경우:  $DN_i$ 의 멤버 리스트에  $Addr(N_j)$ 를 추가하고,  $Addr(DN_i)$ 를  $N_j$ 에 전송한다.
- $DN_i$  부그룹이 계층적인 부그룹들로 이루어진 경우:  $Addr(N_j)$ 에 따라서  $N_j$ 가 추가되어야 할 부

그룹이 결정되며, 다음과 같은 두 가지 상황이 발생할 수 있다.

- " $Addr(하위\ 부그룹\ DN) \langle Addr(N_j) \langle Addr(DN_i)$ "를 만족할 경우

$N_j$ 를  $DN_i$  자신의 부그룹에 추가하며, 이를 위하여  $DN_i$ 의 멤버 리스트에  $Addr(N_j)$ 를 추가하고,  $Addr(DN_i)$ 를  $N_j$ 에 전송한다.

- " $Addr(Pred(DN_i)) \langle Addr(N_j) \langle Addr(하위\ 부그룹\ DN)$ "를 만족하는 경우

$N_j$ 가 자신의 하위 부그룹에 추가 되어야 하는 경우로, 하위 부그룹의 DN에게 JoinRequest ( $Addr(N_j)$ )를 중계한다.

- 하나의 부그룹이 여러 계층의 부그룹들로 구성된 경우에는 위의 과정이 동일하게 반복된다.

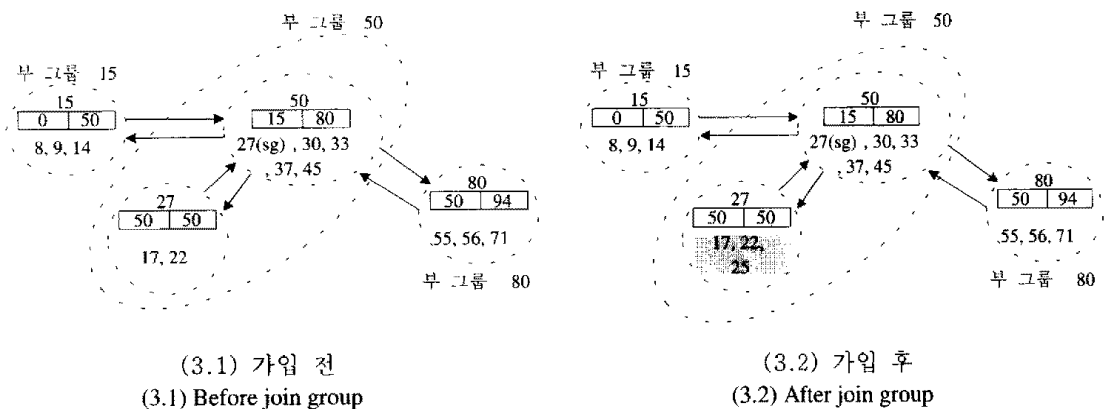
그림 3은 주소가 25인 노드가 그룹에 가입하는 경우를 보여주고 있다. 그림에서 25번 노드는 조건 3에 따라서 50번 부그룹의 하위 그룹에 가입되었다.

#### 4.2 그룹에서의 탈퇴

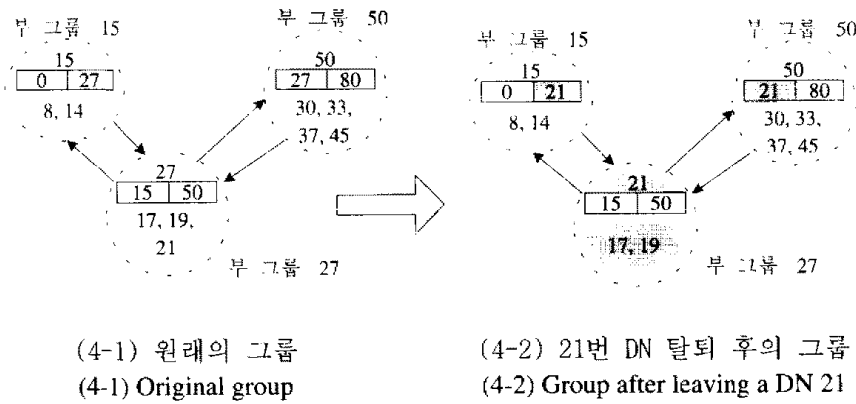
그룹에서 탈퇴하는 노드는 일반 노드 또는 DN이 될 수 있다.

그룹을 탈퇴하려는 일반 노드  $N_{ij}$ 는 LeaveRequest ( $Addr(N_{ij})$ )를  $DN_i$ 에게 전송한다. LeaveRequest를 수신한  $DN_i$ 는 자신의 부그룹 멤버 리스트에서  $Addr(N_{ij})$ 를 제거한다.

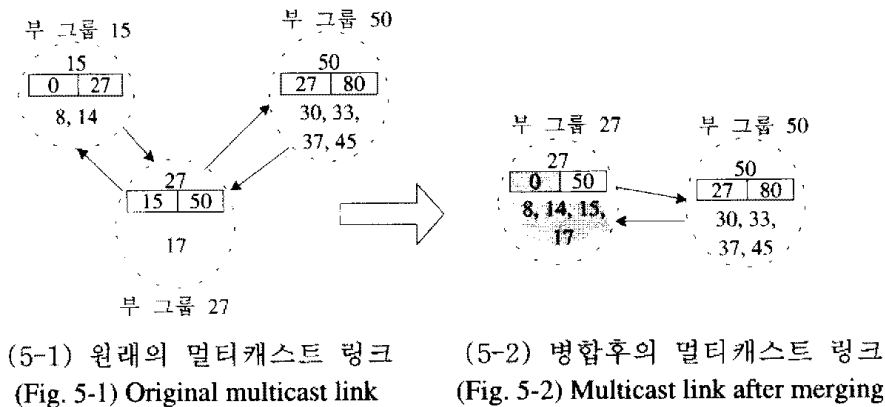
그룹에서 탈퇴하려는  $DN_i$ 는 자신의 부그룹 멤버들 중에서 가장 큰 주소를 갖는 노드를 새로운 DN으로 결정하고, 이 노드에게 NewDNRequest ( $Addr(Pred$



(그림 3) 그룹에의 가입  
(Fig. 3) Join to the group



(그림 4) DN의 탈퇴에 의한 멀티캐스트 링크의 재구성  
(Fig. 4) Reconfiguration of multicast link after leaving a DN



(그림 5) 부 그룹의 단일병합  
(Fig. 5) Flat merge of sub-groups

( $DN_i$ ),  $Addr(Succ(DN_i))$ , 멤버 리스트)를 전송한다. 새롭게 결정된 DN으로부터 NewDNRequest에 대한 NewDNRequest-ACK를 수신한  $DN_i$ 는  $Pred(DN_i)$ ,  $Succ(DN_i)$ , 그리고 새로운 부그룹 멤버들에게 ChangeDNRequest ( $Addr(DN_i)$ ,  $Addr(\text{새로운 } DN)$ )를 전송한다.

그림 4는 DN27이 그룹을 떠나서 멀티캐스트 링크가 다시 구성되는 것을 보여주고 있다.

## 5. 부그룹 관리

### 5.1 부그룹 병합

#### (1) 단일 병합(Flat Merge)

단일 병합은 두 개의 부그룹이 병합되어 새로운 단일의 부그룹을 구성하는 동작이다.  $DN_i$ 는  $Succ(DN_i)$ 가 관리하는 멤버 수와 자신이 관리하는 멤버 수의 합이 일정한 수 이하가 되는 경우 단일 병합 동작을 수행한다. 이때, 병합 동작은  $DN_i$ 가  $Succ(DN_i)$ 에 포함되는 방향으로만 가능하다.

단일 병합을 수행하려는  $DN_i$ 는 FlatMergeRequest ( $Addr(Pred(DN_i))$ , 멤버 리스트)를  $Succ(DN_i)$ 에게 전송하고, 자신의 부그룹에 속하는 멤버들에게 ChangeDNRequest ( $Addr(DN_i)$ ,  $Addr(Succ(DN_i))$ )를 전송한다.

그림 5.는 15번 부그룹이 27번 부그룹으로 단일 병합되는 것을 보여주고 있다.

(2) 계층적 병합(Hierarchical Merge)

계층적 병합 동작을 수행하면, 두 부그룹이 병합되어 계층적 구조를 갖는 새로운 부그룹이 구성된다. 계층적 병합은 종단까지의 데이터 전송 지연 시간을 줄일 수 있는 장점을 갖는다. 계층적 병합도 단일 병합과 동일하게  $DN_i$ 가  $Succ(DN_i)$ 의 부계층으로 포함되는 방향으로만 가능하다.

계층적 병합을 수행하려는  $DN_i$ 는  $Succ(DN_i)$ 에게  $HierMergeRequest (Addr(Pred(DN_i)))$ 를 전송하고,  $Pred(DN_i)$ 에게  $ChangeDNRequest (Addr(DN_i), Addr(Succ(DN_i)))$ 를 전송한다.

$HierMergeRequest$ 를 수신한  $Succ(DN_i)$ 는  $DN_i$ 를 자신의 멤버 리스트에 하위 그룹 DN으로 추가하고, 자신의 앞 DN 주소를  $Addr(Pred(DN_i))$ 로 수정한다.

그림 6은 계층적 병합으로 인하여 하나의 부그룹 내에 몇 개의 계층이 구성되는 것을 보여주고 있다.

5.2 부그룹 분리

DN은 자신이 관리하는 멤버 수가 일정한 수 이상이

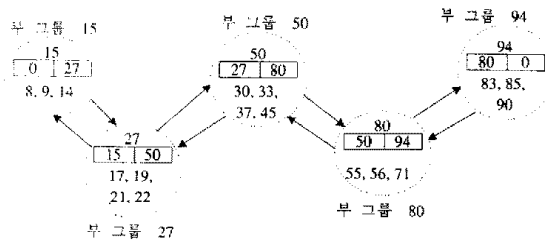
되는 경우 분리 동작을 수행한다. 이때, DN의 부그룹이 계층적으로 구성된 경우 하위 계층은 하나의 멤버로 계산된다.

부그룹을 분리하는 동작은 분리하려는 부그룹이 하나의 단일 그룹인 경우와 계층적 구조를 갖는 부그룹인 경우로 분리할 수 있다.

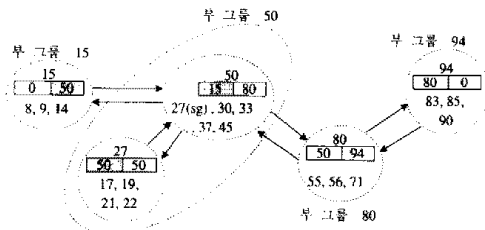
(1) 단일 부그룹의 분리

DN은 자신이 관리하는 멤버 수를 둘로 나누어, 하위 주소를 갖는 노드들을 새로운 단일 부그룹으로 구성한다.

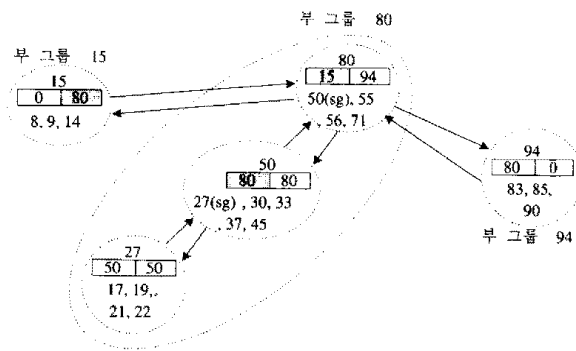
멤버의 분리를 결정한  $DN_i$ 는 새로운 부그룹으로 구성될 멤버들 중 가장 큰 번호를 갖는 노드를 새로운 DN으로 결정하고, 이 노드에게  $NewDNRequest (Addr(Pred(DN_i)), Addr(DN_i), \text{새로운 멤버 리스트})$ 를 전송한다. 그리고,  $Pred(DN_i)$ 에게  $ChangeDNRequest (Addr(DN_i), Addr(\text{새로운 DN}))$ 를 전송하고 자신의 이전 DN 주소를 새로운 DN의 주소로 변경한다.



(a) 원래의 멀티캐스트 링크  
(a) Original multicast link



(b) 부 그룹 27의 계층적 병합 후  
(b) Multicast link after hierarchical merging of sub-group 27



(c) 부 그룹 50의 계층적 병합 후  
(c) Multicast link after hierarchical merging of sub-group 50

(그림 6) 계층적 병합에 따른 멀티캐스트 링크  
(Fig. 6) Multicast link according to the hierarchical merging

NewDNRequest를 수신한 새로운  $DN_h$ 는 자신의 멤버들에게 ChangeDNRequest ( $Addr(Pred(DN_i)), Addr(DN_h)$ )를 전송한다.

### (2) 계층적 부그룹의 분리

DN은 자신의 하위 계층의 부그룹을 하나의 독립적인 부그룹으로 구성할 수 있다.

이를 위하여  $DN_i$ 는 하위 계층의 부그룹을 관리하는  $DN_h$ 에게 NewDNRequest ( $Addr(Pred(DN_i)), Addr(DN_i), 0$ )를 전송하고,  $Pred(DN_i)$  DN에게 ChangeDNRequest ( $Addr(DN_i), Addr(DN_h)$ )를 전송한다. 그리고, 자신의 이전 DN 주소를 새로운 값으로 수정한다.

### 5.3 부그룹 병합 및 분리 동작의 검토

병합 방식에 있어서 계층적 병합 방식은 병합되는 부그룹의 멤버들에게 ChangeDNRequest를 전송하지 않기 때문에 단일 병합 방식에 비하여 우수하다. 또한, 계층적으로 구성된 부그룹이 있는 경우 하위 계층의 부그룹 멤버들에게 안정된 데이터 전송을 지원하는 것은 하위 계층 DN 책임이므로, 상위 계층 DN은 병합의 결과 자신이 처리하여야 하는 일의 양이 이전과 거의 동일하다는 장점을 갖는다. 분리 방식에 있어서도 분리되는 부그룹의 멤버들에게 ChangeDNRequest를 전송하지 않기 때문에 단일 부그룹 분리 방식보다, 계층적 부그룹 분리 방식이 더 우수하다.

그러나, 계층적 병합/분리 동작만을 지원할 경우에는 적은 수의 멤버들을 갖는 DN들이 여러 개 존재함으로써 인하여 오버헤드를 가져올 수 있다.

## 6. 결 론

본 논문에서는 양방향의 안정된 데이터 전송을 지원하는 멀티캐스트 전송 프로토콜에 관하여 기술하였다.

이전에 개발된 멀티캐스트 전송 프로토콜들과는 달리, 본 논문에서 제안된 MM-RMTP는 그룹에 가입한 모든 노드가 데이터를 전송할 수 있으며 그룹 내에서 하나의 멀티캐스트 링크를 유지한다는 장점을 갖는다. 또한, 그룹 멤버의 가입 및 탈퇴에 따라서 멀티캐스트 링크가 동적으로 쉽게 변화할 수 있으며, 이에 필요한 오버헤드가 아주 적다는 장점을 제공한다. 본 프로토콜은 하위 계층에 단순히 데이터 전송 서비스만을 요구

하며, IP 주소를 갖는 인터넷 환경에서 멀티캐스트 라우터 없이 쉽게 구현될 수 있다.

MM-RMTP가 IP 주소를 사용하는 인터넷 환경에서 동작하는 경우 "통과(passing-through)" 문제가 발생할 수 있다. 이 문제는 데이터 패킷이 근거리의 서브넷(subnet)에 있는 그룹 멤버에게 직접 전달되지 못하고 원거리를 경유하여 도착하는 문제이다. 이러한 상황은 비슷한 IP 주소 값을 갖는 노드가 원거리에 존재하고, IP 주소 값에 큰 차이를 보이는 노드들이 근거리 존재할 경우에 발생할 수 있다.

현재는 MM-RMTP 프로토콜을 구현하고 있으며, 이 구현이 완료되면 실제의 인터넷 환경에서 시뮬레이션을 통하여 프로토콜의 성능을 다른 프로토콜들과 비교, 분석하는 연구를 수행할 예정이다. 그리고, MM-RMTP의 가장 큰 취약점으로 생각될 수 있는 통과 문제를 시뮬레이션으로 분석하는 연구를 수행할 예정이다.

### 부록: MM-RMTP 함수와 파라미터

JoinRequest (가입 노드 주소);

LeaveRequest (탈퇴 노드 주소);

NewDNRequest (앞 DN 주소, 뒤 DN 주소, 멤버 리스트);

ChangeDNRequest (이전의 DN 주소, 새로운 DN 주소);

FlatMergeRequest (앞 DN 주소, 멤버 리스트);

HierNergeRequest (앞 DN 주소);

## 참 고 문 헌

- [1] Sanjoy Paul, Krishan K. Sabnani, "Reliable Multicast Transport Protocol(RMTP)", IEEE JSAC, Vol.15, No.3, pp.407~421, April 1997.
- [2] Sally Floyd, Van Jacobson, "A Reliable multicast framework for light-weight sessions and application level framing", ACM SIGCOMM95, ACM Computer Comm. Review, pp.342~356, Aug. 1995.
- [3] Tony Ballardie, Paul Francis, Jon Crowcroft, "Core based trees(CBT): An architecture for scaleable inter-domain multicast routing", ACM SIGCOMM'94, pp.85~95, 1994.

[4] Stephen Deering, Deborah Estrin, "An architecture for wide-area multicast routing", ACM SIGCOMM'94, pp.126~135, 1994.

[5] S. Ramakrishnan, B.N. Jain, "A negative acknowledgment with periodic polling protocol for multicast over LAN", IEEE Infocom, pp.502-511, Mar. 1987.

[6] Rajendra Yavatlar, James Griffioen, "A Reliable dissemination protocol for interactive collaborative applications", ACM Multimedia, pp.333~344, 1995.

[7] Christophe Diot, Walid Dabbous, Jon Croft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms", IEEE JSAC, Vol.15, No.3, pp.277~290, April, 1997.

[8] "Host extensions for IP multicasting", Internet RFC 1112, Aug. 1989.

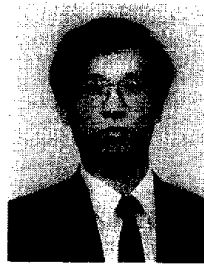
[9] "Distance vector multicast routing protocol", Internet RFC 1075, Nov. 1988.

[10] "Multicast extensions to OSPF", Internet RFC 1584.

[11] "Internet Protocol", Internet RFC 760, 1980.

[12] "DVMRP", Internet RFC 1075.

[13] Sridhar Pingali, Don Towsley, "A Comparison of Sender-initiated and receiver-initiated reliable multicast protocols", Performance Review, Vol.22, pp.221~230, May, 1994.



**김 은 기**

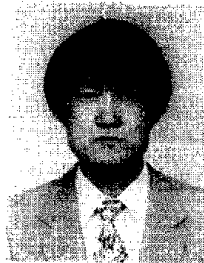
1987년 2월 고려대학교 전자공학과(학사)

1989년 2월 고려대학교 전자공학과 대학원(석사)

1994년 2월 고려대학교 전자공학과 대학원(박사)

1995년 2월~현재 대전산업대학교 정보통신공학과 조교수

관심분야 : 컴퓨터 네트워크



**나 근 식**

1986년 2월 고려대학교 전자공학과(학사)

1988년 2월 고려대학교 전자공학과 대학원(석사)

1992년 8월 고려대학교 전자공학과 대학원(박사)

1992년 9월~현재 한신대학교 전자계산학과 조교수

관심분야 : 컴퓨터 네트워크, 이동통신