# 차별 제어와 키 관리 기능을 통한 트리 기반의 신뢰성 있는 멀티캐스트 프로토콜의 설계 및 구현

김 영 재[†] · 박 은 용[††] · 안 상 준[††] · 현 호 재[†††] · 한 선 영[††††]


## 요 약

IP 멀티캐스트는 하나의 연결을 통해 데이터를 동시에 여러 컴퓨터로 보낼 수 있게 함으로서 비디오 스트림과 같은 대용량의 데이터 전송에 효율적으로 사용될 수 있다. 하지만, IP 멀티캐스트는 UDP에 기반 한 best-effort 서비스로서, TCP에서 제공하는 신뢰 전송, 혼잡 제어 및 흐름 제어를 제공하지 못하는 단점이 있다. 또한 멀티캐스트 그룹 참가자는 어떠한 인증 과정이나 절차 없이 멀티캐스트 그룹에 참가/탈퇴가 가능하고, 하나의 서브넷 내에서는 브로드캐스트를 이용하기 때문에, 데이터의 기밀을 유지하기가 어렵다는 단점이 있다. 본 논문에서는, 일-대-다 멀티캐스트 응용에 적합한 흐름 제어, 혼잡 제어, 키 관리 기법을 제공하는 신뢰전송 프로토콜인 TRDMP를 제안한다.


# Design and Implementation of Tree-based Reliable Dissemination Multicast Protocol With Differential Control and Key Management

Young-jae Kim[†] · Eun-yong Park[††] · Sang-joon Ahn[††] · Ho-jae Hyun[†††] · Sun-young Han[††††]


## ABSTRACT

While the Internet is suffering from the massive data such as video stream, IP multicast can ease the load of the Internet by enabling one copy of digital information to be received by multiple computers simultaneously. But IP multicast is based on UDP, packets are delivered using a best-effort policy without any reliability, congestion control or flow control. Multicast group members can join or leave a multicast group at will, and multicast uses broadcast mechanism, it's very hard to keep security from unauthorized members. In this paper, we introduce a new reliable multicast protocol TRDMP proper for one-to-many multicast model with reliability, flow control, congestion control and key management.


**키워드 : TRDMP, Reliable, Protocol, Multicast**

## 1. 서 론

IP 멀티캐스트는 IP D class 주소를 이용하여 인터넷 환경 하에서 효율적으로 다-대-다 통신을 제공한다[1-3]. 이 IP 멀티캐스트는 유니캐스트의 단점, 즉 하나의 노드가 다른 여러 노드에게 데이터를 전송하고자 할 때 각각에 대한 연결을 만들어 주어야 하기 때문에 망에 대한 부하가 높아지는 단점과 분산 환경 하에서 서버 다운에 대한 장애 극복이 어렵다는 단점을 해결할 수 있으며, 수 많은 사용자에게 동시에 데이터를 효율적으로 전송할 수 있는 장점을 가진다. IP 멀티캐스트는 일대다(one-to-many) 또는 다-대-다(many-to-many) 형태로 응용될 수 있다.

※ 본 논문은 2001년도 정보통신부에서 지원하는 대학기초연구지원사업(2001 062 3)에 의한 결과임.
† 정 회 원 : 건국대학교 대학원 컴퓨터 · 정보통신공학과
†† 준 회 원 : 건국대학교 대학원 컴퓨터 · 정보통신공학과
††† 준 회 원 : 건국대학교 대학원 컴퓨터 · 정보통신공학과
†††† 정 회 원 : 건국대학교 컴퓨터 · 정보통신공학과 교수
논문접수 : 2001년 9월 10일, 심사완료 : 2002년 2월 8일

신뢰성 있는 전송 서비스는 공유 작업 환경과 분산 파일 전송 등과 같은 응용에서 요구되어 진다. 이 서비스는 각 응용의 특징(일대다 통신과 다-대-다 통신)에 따라 각기 다른 형태의 서비스를 요구하며, 분산 파일 전송과 같은 응용들은 차별화 된 전송량을 제공 받기를 원한다. 또한 많은 신뢰성 있는 프로토콜들은 호스트 처리 능력이나 네트워크 자원에 의한 잠재적인 병목 현상을 피하기 위해 다양한 흐름 제어와 정체 제어를 연구하고 있다[4].

신뢰성 있는 전송 프로토콜은 흐름 제어와 오류 제어를 수행하는 주체에 따라 송신자 주도형과 수신자 주도형 프로토콜로 나누어 진다[5, 6]. 송신자 주도형 프로토콜들은 각 송신자가 모든 수신자들에 대한 정보를 유지하며, 각 수신자는 모든 패킷에 대해 ACK을 송신자에 전송한다. 이러한 송신자 주도형 프로토콜의 문제점은 수신자의 수가 증가함에 따라 ACK 폭주가 발생하고, 데이터 전송 지연이 증가된다. 수신자 주도형 프로토콜은 송신자 주도형 프로토

콜과는 반대로 수신자가 오류 검출을 수행하며 패킷을 손실을 감지할 경우 NACK을 전송한다. 수신자 주도형 프로토콜의 문제점은 송신자의 버퍼 넘침이 발생하며, 패킷 손실이 발생할 경우 오류 복구에 대한 양단간의 지연이 커진다. 또한 이 방법을 멀티캐스트 응용에 적용할 경우, 재전송을 전체 그룹에 멀티캐스트하기 때문에 네트워크 트래픽이 불필요하게 증가된다[7].

본 논문에서는 신뢰성 있는 프로토콜은 일대다 응용에 적합한 트리 기반의 신뢰성 있는 프로토콜(TRDMP)을 설계 및 구현한다. TRDMP(Tree-based Reliable Dissemination Multicast Protocol)는 데이터 전송에 관련된 TRTP(Tree-based Reliable Transfer protocol)와 제어 트리 구성을 위한 DCTCP(Dynamic Control-Tree Configuration Protocol)로 이루어진다.

TRDMP는 송신자 기반 프로토콜과 수신자 기반 프로토콜들의 단점을 해결하기 위해 지역 그룹을 기반으로 한 계층적 제어 트리를 이용한다. 각 지역 그룹은 오류 제어를 위해 동적으로 선정되는 그룹 관리자가 존재한다. TRDMP는 그룹 관리자를 통해 오류에 대한 회복 시간과 ACK 메시지의 폭주를 최소화 한다. 그리고 잠재적인 병목 현상을 피하기 위해 흐름 제어와 정체 제어를 제공한다. 흐름 제어는 전송률 기반(rate-based) 기법과 윈도우 기반(windows-based) 기법을 사용하며 정체 제어는 slow-start 기법과 정체 회피(congestion avoidance) 기법을 사용한다[8].

TRDMP는 전송 지연과 오류 복구 지연 시간을 최소화하며, 그룹 관리자의 결함 허용을 빠르게 처리한다. 또한 각각 차별화 된 흐름 제어와 정체 제어 서비스를 제공한다. 차별화 된 제어 서비스의 장점은 분산 파일 전송이나 웹 캐싱 등과 같은 응용에서 모든 수신자들이 하나의 흐름 제어와 정체 제어를 통해 데이터를 전송 받을 때, 모든 수신자들의 데이터 수신 시간이 유사한 것에 비해, 차별화된 서비스를 통해 각 수신자들의 데이터 수신 시간에 차이가 발생한다.

본 논문의 2장에서는 기존의 신뢰성 있는 멀티캐스트 전송 프로토콜과 그 특징들을 조사 함으로 신뢰성 있는 멀티캐스트 프로토콜이 가져야 하는 특징을 살펴 본다. 3장에서는 TRDMP의 설계와 구현을 다루었다. 4장에서는 구현 결과를 알아보고, 5장에서는 성능 평가를 기술하였다.

## 2. 신뢰성 있는 멀티캐스트 전송 프로토콜

현재, 많은 신뢰성 있는 멀티캐스트 전송 프로토콜이 연구 되고 있으며, 그 중 대표적인 프로토콜에 대해 그 특징을 살펴 보았다. 다음 <표 2.1>은 신뢰성 있는 멀티캐스트 프로토콜들의 특징을 나타낸다. 비교 기준은 프로토콜의 특징을 나타내는 프로토콜 구조, 데이터 전송 방식, 오류 재전송, 데이터 흐름 제어, 계층 구조를 가지는 프로토콜에서 도메인 관리자의 장애 제어 등으로 선택한다. <표 2.1>에서 프로토콜 구조에서는 "계층적 구조"는 오류 제어 과정에서 도메인 단위로 계층 구조를 가지는 프로토콜을 나타내며, "Flat"는 그렇지 않은 경우를 나타낸다. 전송 방식에서는 데이터의 전송에 멀티캐스트와 유니캐스트이 혼용 여부를 알 수 있으며, 재전송은 패킷 손실에 대한 재전송에 어떤 방법을 사용하는지를 알 수 있다. 흐름 제어에서 "전송률"은 전송률 기반 흐름 제어의 형태를 나타내며, "윈도우"는 윈도우 기반 흐름 제어를 의미하고, "전송률/윈도우"는 두 가지 방법의 혼용을 의미한다. 마지막으로 장애 제어는 프로토콜 구조가 계층적인 경우, 각 계층의 관리자에 대한 장애 제어를 지원하는가를 알 수 있다.

## 3. 트리 기반의 신뢰성 있는 프로토콜 설계 및 구현

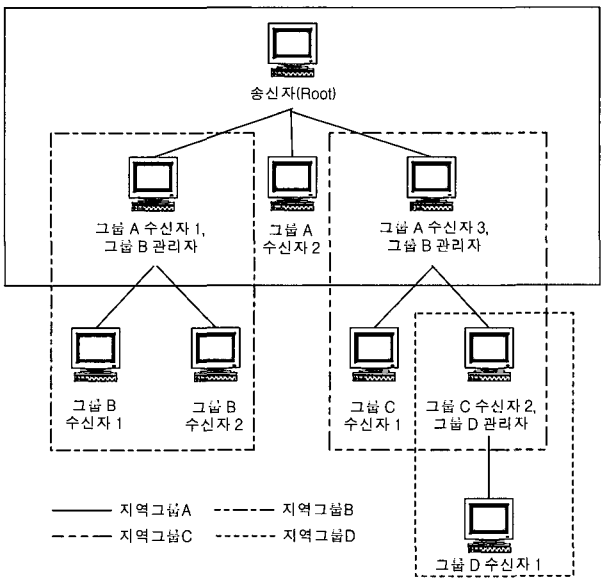
### 3.1 TRDMP 구조

TRDMP 프로토콜은 데이터 전송을 위한 TRTP와 제어 트리 구성을 위한 DCTCP로 구성되어 있다. 모든 참가자는 제한된 TTL값으로 지역 그룹을 구성하고, 지역 그룹은 오류 제어를 위해 각 그룹간의 계층적인 제어 트리 구조를 이룬다. 각 지역 그룹은 하나의 그룹 관리자를 가지며, 지역 관리자는 해당 그룹의 패킷 손실에 대한 오류 제어를 책임진다. 또한 그룹 관리자는 멀티캐스트 데이터를 암호화하고 복호화 할 수 있는 비밀키를 생성, 배포 및 관리한다. 각 그룹의 그룹 관리자는 상위 그룹 관리자와 하위 그룹

〈표 2.1〉 신뢰성 있는 프로토콜들의 특징 비교

| 프 로 토 콜 | 구 조 | 전 송 방 식 | 재 전 송 | 흐름 제어 | 장애 제어 |
|---|---|---|---|---|---|
| TMTP[9] (Tree-based Multicast Transport Protocol) | 계층구조 | 멀티캐스트/유니캐스트 | 멀티캐스트 | 전송률/윈도우 | 없 음 |
| LGMP[10] (Local Group based Multicast Protocol) | 계층구조 | 멀티캐스트/유니캐스트 | 멀티캐스트/유니캐스트 | 전 송 률 | 제 공 |
| SRM[11] (Scalable Reliable Multicast) | 평면구조 | 멀티캐스트 | 멀티캐스트 | 전 송 률 | 해당 없음 |
| RAMP[12] (Reliable Adaptive Multicast Protocol) | 평면구조 | 멀티캐스트/유니캐스트 | 유니캐스트 | 전 송 률 | 해당 없음 |
| RMTP[13] (Reliable Multicast Transport Protocol) | 계층구조 | 멀티캐스트/유니캐스트 | 멀티캐스트/유니캐스트 | 전송률/윈도우 | 제 공 |

관리자에 대한 관계를 형성하여 하나의 계층적인 구조를 이룬다. 이러한 계층적 제어 트리의 형태는 (그림 3.1)과 같다.




(그림 3.1) TRDMP의 계층적인 제어 트리

지역 그룹에서 패킷 손실에 대한 재전송을 TTL값으로 전송 범위를 제한 할 경우, 각 그룹의 경계에 위치한 타 그룹의 수신자들에게 불필요한 데이터를 전달하게 되고, 트래픽을 낭비하게 된다. 이를 해결하기 위해 지역 그룹은 새로운 멀티캐스트 주소를 사용하여 지역 그룹에 데이터를 전송한다. 즉, 그룹 관리자는 그룹 생성시 새로운 멀티캐스트를 생성하고, 그룹 참가자들에게 그룹용 멀티캐스트 주소를 알려준다. 그룹 관리자는 상위 그룹으로부터 받은 데이터를 그룹용 멀티캐스트 주소를 사용하여 그룹 내의 참가자들에게 전달한다. 또한, 데이터를 일정 기간 동안 버퍼에 유지하고 있다가, 참가자의 재전송 요구를 처리한다.

지리적으로 넓게 분포되어 있는 그룹 참가자들이 계층적 트리를 구성 할 경우, 각 그룹간의 네트워크 상황이 다를 수 있다. 이러한 조건에서, 단일의 흐름 제어, 정체 제어를 실시할 경우, 계층적 트리 전체의 전송률을 트리에서 가장 느린 그룹의 전송률에 맞춰 낮아질 수 밖에 없다. 일-대-다 라고 하는 특성상 이에 적합한 차별화 된 흐름 제어, 정체 제어가 필요하다. TRDMP는 전체 계층적 트리를 RTT 값에 기반 하여 세 개의 독립적인 그룹으로 형성하고, 각 그룹의 조건에 맞는 차별화된 제어를 실시한다.

TRDMP 프로토콜의 구성 요소들에 대한 설명은 다음과 같다.

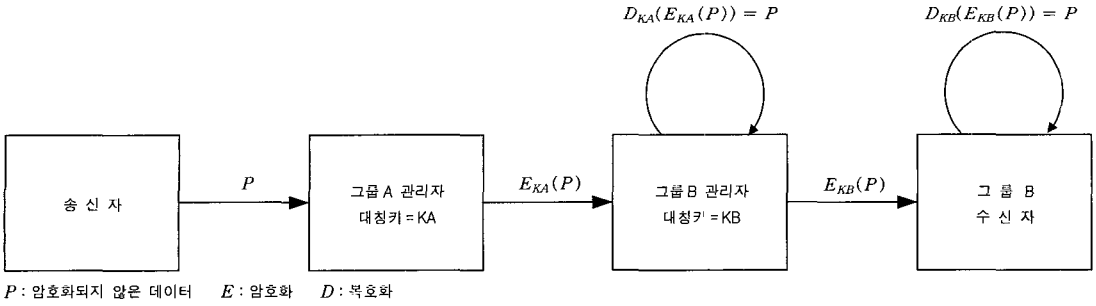
- 송신자
  - 데이터를 전송하는 주체이다.
- 참가자(수신자)
  - 송신자로부터 전송되는 모든 데이터를 수신한다.
- 지역 그룹
  - 다양한 제어 서비스 및 신뢰 전송을 수행하기 위해 이루어진 논리적 그룹이며, 제한된 TTL에 의해 그 그룹의 범위가 결정된다.
- 지역 그룹 관리자
  - 그룹 참가자들에 대한 다양한 제어 서비스를 제공하며, 상위 그룹과 하위 그룹을 연결시켜주는 매개체 역할을 한다. 동적인 방법에 의해 선정된다.
    - 상위 그룹 관리자
      그룹 간의 계층적 제어 구조에서 하위에 그룹 관리자를 관리하는 관리자이다.
    - 중간 그룹 관리자
      그룹 간의 계층적 제어 구조에서 상위와 하위 그룹 관리자 사이에 존재하는 관리자이다.
    - 종단 그룹 관리자
      그룹 간의 계층적 제어구조에서 가장 하위에 존재하는 그룹 관리자이다.

### 3.2 DCTCP를 이용한 제어 트리 구성

DCTCP는 계층적인 제어 트리를 구성하기 위해 확장 링 검색 방법을 이용한다. 확장 링 검색 방법은 제한된 TTL을 이용하여 지역 그룹을 형성한다. 지역 그룹 관리자는 상위 그룹 관리자를 검색하기 위해 지역 그룹을 형성하는 TTL 값 보다 큰 TTL값을 이용한다. 송신자는 계층적인 제어 트리의 루트가 되며 송신자는 자신의 지역 그룹 관리자이다.

#### 3.2.1 DCTCP의 키 관리

DCTCP에서 각 지역 그룹은 각각의 멀티캐스트 주소와 함께 대칭형 암호화 키를 가지며, 지역 그룹 관리자가 지역 그룹의 키를 생성하고 각 지역 그룹의 참가자들에게 배포하는 키 관리자(key management center)역할을 한다. 지역 그룹 관리자는 상위 지역 그룹을 통해 받은 멀티캐스트 데이터를 복호화 한 후, 다시 지역 그룹의 암호화 키로 암호화 하여 그룹에 멀티캐스트 한다. 이는 전체 참가자 중 하나의 참가자라도 그룹을 탈퇴할 경우, 탈퇴한 참가자가 더 이상 멀티캐스트 데이터를 수신하지 못하도록 다른 모든 참가자들에게 암호화 키를 재분배(re-keying)해야 한다는 단점을 극복하기 위한 것이다. 즉, DCTCP에서는 참가자가 탈퇴를 할 경우, 해당 지역 그룹 내에서만 키의 재분배(re-keying)가 발생함으로 전체 그룹에 걸친 빈번한 키의 재분배를 방지할 수 있다. (그림 3.2)는 송신자로부터 송신된 데이터를 각 그룹 관리자가 그룹의 암호화 키로 암호화 하고, 하위 그룹 관리자가 이를 복호화 한 후, 다시 하위 그룹의 암호화 키로 암호화 하여 종단의 참가자에게 전달하

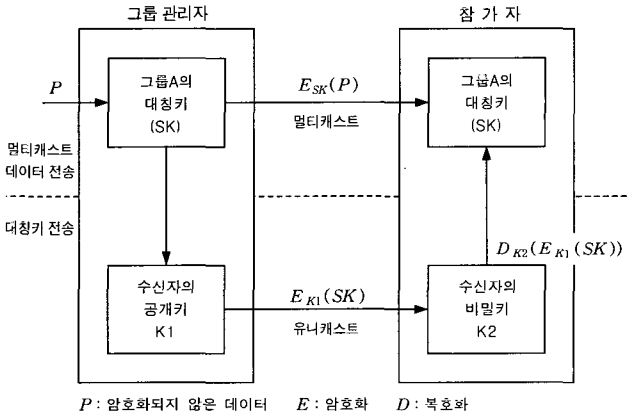



(그림 3.2) 대칭키 방식을 사용한 멀티캐스트 데이터의 전송

는 과정을 보여준다.

DCTCP는 멀티캐스트 데이터의 암호화를 위해 두 가지 방식의 암호화 기법을 사용한다. 멀티캐스트 데이터를 전송하기 위해서는 키의 분배가 용이하고 암호/복호화 속도가 빠른 대칭키(symmetric key) 암호화 기법을 사용하고, 대칭키를 지역 그룹에 배포하기 위해 비대칭 키(asymmetric key) 암호화 기법을 사용한다. 즉, 지역 그룹 관리자는 지역 그룹내의 모든 참가자의 공개키를 보관하고 있으며, 그룹내의 참가자가 탈퇴를 할 경우, 지역 그룹 관리자는 새로운 지역 그룹용 대칭키를 생성하여 참가자들의 공개키를 사용하여 암호화 한 후 참가자들에게 유니캐스트로 전송한다.

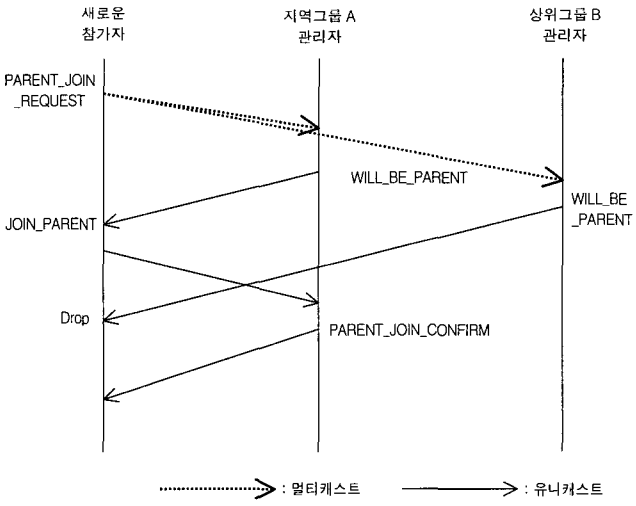
(그림 3.3)은 비대칭 암호화 기법을 사용하여 대칭키를 배포하는 구조를 보여준다. 그룹 관리자는 수신자의 공개키를 유지하고 있으며, 그룹용 대칭키를 수신자의 공개키를 통해 암호화 한 후 수신자에게 전달한다. 수신자는 자신의 비밀키로 관리자가 보내준 그룹용 대칭키를 복호화 해 낼 수 있다. 그 후, 그룹 관리자는 그룹용 대칭키로 암호화한 멀티캐스트 데이터를 수신자에게 전송하고, 수신자는 그룹용 대칭키로 데이터를 복호화 할 수 있다.



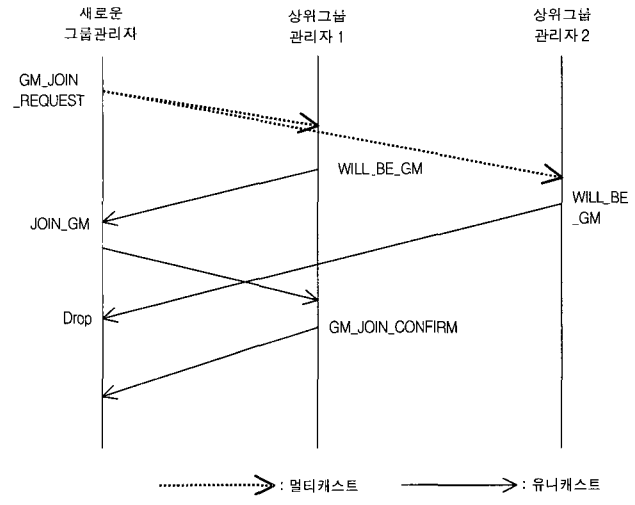

(그림 3.3) 비대칭 암호화 기법을 사용한 대칭키 배포

3.2.2 계층적 트리 구성

지역 그룹 관리자를 찾기 위한 메시지 흐름은 (그림 3.4)과 같고, 지역 그룹 관리자가 상위 그룹 관리자를 찾기 위한 메시지의 흐름은 (그림 3.5)과 같다.




(그림 3.4) 지역 그룹 관리자 선정시 메시지 흐름




(그림 3.5) 상위 그룹 관리자 선정시 메시지 흐름

DCTCP는 그룹 관리자가 모든 참가자에 대한 정보를 유지하게 되는데, 이는 참가/탈퇴를 통해 각 참가자의 멤버쉽 정보를 관리한다.

DCTCP에서 그룹 참가 방법은 다음과 같다.

● 그룹 관리자가 존재하지 않는 경우

1) 그룹 관리자 존재 여부 판별
- 그룹 관리자의 존재 여부를 판별하기 위해 참가자는 제한된 TTL 값으로 제어 메시지를 멀티캐스팅 한다.
- 제한된 시간 내에 응답이 도착하지 않을 경우, 새로운 그룹을 생성하고 그룹 관리자 역할을 수행한다.

2) 새로운 그룹 생성 및 그룹 초기화
- 지역 그룹용 멀티캐스트 그룹 주소 및 암호화 키 생성

3) 상위 그룹 검색 및 참가
- 지역 그룹 구성 TTL 보다 큰 값으로 제어 메시지를 전송하여 상위 그룹 검색.
- 상위 그룹을 찾지 못할 경우, 보다 큰 TTL 값으로 제어 메시지 전송.
- 상위 그룹에 참가 후, 그룹의 정보(지역 그룹용 멀티캐스트 그룹 주소)와 자신의 공개키를 상위 그룹 관리자에게 전송.
- 상위 그룹 관리자는 그룹 상태정보에 하위 그룹의 정보를 추가하여 그룹의 모든 참가자들에게 전달.

● 그룹 관리자가 존재하는 경우

1) 그룹 관리자 존재 여부 판별
- 지역 그룹 구성 TTL 보다 큰 값으로 제어 메시지를 전송하여 상위 그룹 검색.
- 그룹 관리자로부터 응답을 받음.

2) 검색된 그룹에 참가
- 검색된 그룹에 참가 메시지를 보내어 참가함.
- 자신의 공개키를 그룹 관리자에게 전송.

3) 그룹 정보 회신 받음
- 그룹 관리자는 그룹 정보에 새로운 참가자 정보를 추가.
- 그룹 정보와 갱신된 그룹 정보(그룹용 멀티캐스트 주소, 암호화 키, 그룹 참가자 리스트, 상위 그룹 관리자 주소, 하위 그룹 관리자들 주소)를 각 참가자의 공개키로 암호화 하여 전송.

DCTCP에서 참가자가 그룹을 탈퇴하고자 할 경우 그룹 관리자에게 탈퇴 메시지를 보낸 후 그 참가자는 탈퇴를 하게 된다. 탈퇴하고자 하는 참가자가 그룹 관리자인 경우는 새로운 그룹 관리자를 선정한 후 탈퇴하고, 반대로 일반 참가자일 경우는 단지 탈퇴 메시지만을 보낸다.

지역 그룹 구성원의 탈퇴 절차는 다음과 같다.

● 일반 참가자가 탈퇴할 경우

1) 그룹 관리자에게 탈퇴 메시지 전달
- 참가자는 탈퇴 메시지를 그룹 관리자에게 전달하고, 그룹 관리자는 참가자를 탈퇴 처리 함.

2) 그룹 상태 정보 갱신 및 새로운 그룹용 암호화 키 분배
- 그룹 관리자는 그룹 상태정보에서 탈퇴한 참가자의 정보를 삭제함.
- 새로운 그룹용 암호화 키를 생성.
- 각 참가자의 공개키로 그룹 상태정보와 그룹용 새 암호화 키를 암호화 하여 그룹내의 모든 참가자들에게 전달.

● 그룹 관리자가 탈퇴할 경우

1) 후임 그룹 관리자 선정
- 그룹 관리자가 탈퇴하는 경우, 우선 후임 관리자를 선정 함.
- 후임 관리자는 그룹 참가자 리스트의 첫번째 참가자로 선정함.
- 후임 관리자 및 상위 그룹 관리자, 지역 그룹 모든 참가자들에게 후임 관리자 선정을 알림.

2) 지역 그룹 키 재분배 및 그룹 정보 갱신
- 후임 관리자는 유지하고 있던 그룹 참가자 리스트에서 기존 관리자를 제거.
- 지역 그룹을 위한 암호화 키를 새로 생성하고, 각 참가자들의 공개키로 전송.

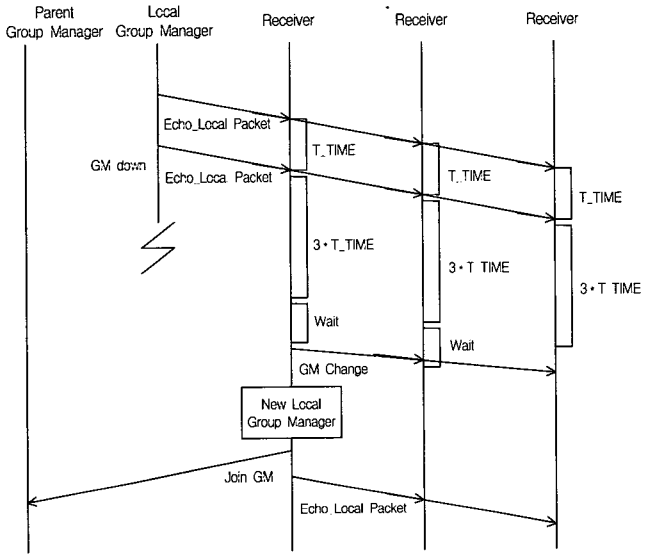
3) 상위 그룹 키 재분배 및 그룹 정보 갱신
- 상위 그룹 관리자는 그룹 관리자를 탈퇴 처리함.
- 후임 그룹 관리자를 상위 그룹 참가자 리스트에 추가한다.
- 상위 그룹 관리자는 상위 그룹용 암호화 키를 새로 생성하고, 각 참가자의 공개키로 암호화 하여 전송한다.

3.2.3 그룹 관리자에 대한 결함 허용

각 그룹 관리자는 주기적으로 자신의 상태를 각 그룹 참가자에게 알린다. 상태 정보에는 관리자가 상위로부터 데이터를 수신할 때 사용한 윈도우 정보와, 마지막으로 받은 패킷의 일련번호가 포함되어 있다. 그룹 관리자의 결함이 발생하여 후임 관리자가 선정 될 경우에 즉시 이 윈도우 정보를 활용할 수 있다. 그룹 참가자들은 새로운 메시지를 받을 때마다 다운 타이머를 동작시킨다. 만약 그룹 관리자로부터 데이터를 일정 시간 이상동안 받지 못하는 경우, 그룹 관리자가 다운되었다고 간주하고 새로운 그룹 관리자를 선정한다. 다운 타이머의 계산은 참가자의 재전송 타이머에서 T_TIME을 구하는 방법을 사용하며, 다운 타이머의 값은 3*T_TIME이다.

그룹 관리자의 다운을 발견한 참가자는 일정 시간을 기다린 후 그룹 참가자가 다운 되었다는 정보와 아울러 새로운 그룹 관리자의 정보를 멀티캐스팅 한다. 일정한 시간을

기다리는 이유는 다수의 참가자들이 그룹 참가자의 다운을 알았을 경우 제어 메시지의 폭주를 막기 위함이다. 이때 그룹 참가자의 재선정은 각 참가자들이 상위 그룹 관리자, 하위 그룹 관리자와 각 참가자에 대한 정보를 알고 있기 때문에 참가자 리스트의 두 번째에 있는 참가자가 새로운 그룹 관리자로 선정이 된다. 새로운 참가자가 그룹 관리자가 되면 자신이 그룹 관리자라는 것을 상위 그룹 관리자와 하위 그룹 관리자에게 알린다. 변경된 정보를 받은 그룹 관리자들은 그 그룹의 참가자들에 변경된 정보를 알린다. (그림 3.6)은 지역 그룹 관리자의 다운 되었을 경우 새로운 지역 그룹 선정하는 과정을 나타낸다. 이 방법은 그룹 관리자가 다운된 그룹의 참가자들이 상위 그룹에 속하는 RMTP에 비해 효율적이다.




(그림 3.6) 그룹 관리자의 다운 처리

### 3.3 TRTP를 이용한 데이터 전송

TRDMP에서 응용 데이터 전송은 TRTP를 사용한다. 송신자는 응용데이타를 보내기 전에 그룹을 생성한다. 그런 후 일정 시간이 지난 후 멀티캐스트 그룹으로 데이터를 전송한다. 각 참가자들은 해당 그룹에 참가한 후 송신자로부터 데이터를 받게 된다. 이때 자신이 그룹 관리자인지 일반 참가자인지가 동적으로 결정된다.

송신자는 데이터를 보낼 때 마다 순서번호를 1씩 증가시켜 보내고, 각 참가자와 그룹 관리자는 오류 재전송을 위해 일정량의 패킷을 버퍼링한다. 일반 참가자가 버퍼링을 하는 이유는 지역 그룹 관리자가 다운 되었을 경우 그 역할을 수행하기 위해서 이다. 또한 각 그룹 관리자의 버퍼 넘침(overflow)을 방지하기 위해 그룹 관리자는 주기적을 상태 메시지를 보내고, 이 상태 메시지를 통해 각 참가자와 그룹 관리자의 일정량의 버퍼의 크기를 유지하도록 한다.

각 그룹 관리자는 패킷을 받을 때마다 상위 그룹 관리자나 송신자에게 ACK을 보낸다. 이러한 ACK을 통해 그룹 관리자의 패킷 손실 여부를 검사한다. 일반 참가자는 오류 패킷이 발생하였을 때만 NAK 패킷을 통해 재전송을 요청한다.

#### 3.3.1 오류 제어 및 재전송

TRTP에서의 오류 제어는 ACK과 NACK 메시지를 사용한다. 각 그룹 관리자는 새로운 데이터를 받을 때마다 ACK 메시지를 상위 그룹 관리자에게 보내며, 지역 그룹의 참가자는 오류가 발생하였을 경우에만 NACK 메시지를 보낸다. 이는 지역 그룹의 모든 참가자가 ACK을 보내게 되면 전송 지연 시간이 증가하게 되고 또한 ACK 메시지가 폭주하게 된다.

TRTP에서는 오류 제어를 위해 3개의 타이머를 사용한다. 하나는 송신자와 그룹 관리자가 하위 그룹 관리자의 패킷 손실을 검사하기 위해 재전송 타이머 사용하고, 또 하나는 일반 참가자가 손실된 패킷을 검사하기 위해 재전송 타이머를 사용한다. 그리고 나머지 하나는 각 참가자들이 그룹 관리자의 다운 여부를 검사하기 위한 타이머이다. 마지막 타이머에 대한 설명은 그룹 관리자의 결함허용 처리부분에 기술되어 있다.

송신자는 데이터를 보낼 때 마다 재전송 타이머를 작동하며, 그룹 관리자는 새로운 데이터를 받을 때 마다 재전송 타이머를 작동한다. 송신자와 각 그룹 관리자에서 송신자와 그룹 관리자간 또는 상위 그룹 관리자와 하위 그룹 관리자의 RTT를 이용하여 재전송 타이머를 계산한다. 재전송 타이머 값을 구하는 수식은 다음과 같다.

$$
\begin{aligned}
Err &= M - A \\
A &= A + g*Err \\
D &= D + h(|Err| - D) \\
RTO &= A + 4D
\end{aligned}
$$

- M은 각 ack에 대한 RTT 값
- A는 RTT에 대한 평균값이다.
- Err은 각 RTT값과 평균 RTT값의 차이다.
- g와 h은 $0 < g, h < 1$ 사이의 임의 값이다.
  $g = 0.125$, $h = 0.25$
- A와 D의 초기값은 0이다.

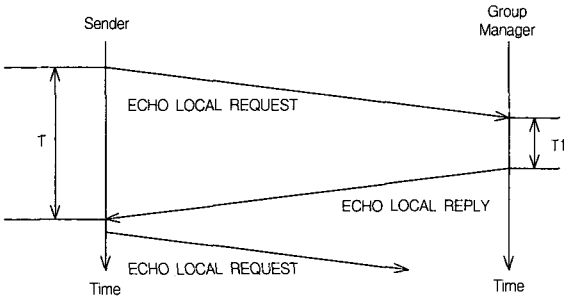
이 수식에서 RTT 값은 송신자에서 사용할 경우 데이터 전송 시작 시간과 그 전송된 데이터에 대한 ACK 응답에 의해서 계산이 된다. 그리고 그룹 관리자에서 사용할 경우 ECHO_LOCAL 패킷을 지역 그룹 참가자들에게 주기적으로 보내고 그에 대한 응답 메시지를 통해 RTT = T - T1를 계산한다(그림 3.7).

일반 참가자가 사용하는 재전송 타이머는 송신자로부터

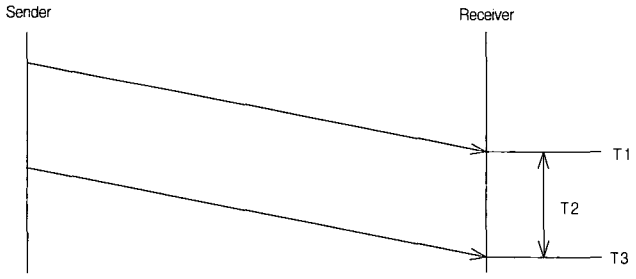
전송되는 메시지를 통해 재전송 타이머 값을 계산한다. 이 타이머는 각 데이터의 도착 시간의 차이를 구한 값에 2를 곱한다. 이는 데이터가 무질서하게 올 경우 일정 타임을 기다리기 위함이다. 참가자의 재전송 타이머 값을 계산하기 위한 수식은 다음과 같다.

NEW_T_TIME = T3 - T1
T_TIME = (T_TIME + NEW_T_TIME)/2
RCV_RTO = 2 * T_TIME

이 수식에서 NEW_T_TIME의 계산 방법은 (그림 3.8)과 같다.




(그림 3.7) RTT(Round Trip Time) 값




(그림 3.8) 참가자의 재전송 타임아웃

일반 참가자는 데이터를 받을 때마다 참가자 윈도우의 정보를 통해 오류 가능성을 검사한다. 이러한 참가자 윈도우의 변수는 (그림 3.9)과 같다.

일반 참가자는 새로운 데이터를 받을 때마다 다음과 같은 조건을 검사하여 만족하면 재전송 타이머의 시작 여부를 결정한다.

rcv_ack < 받은 패킷 순서 번호

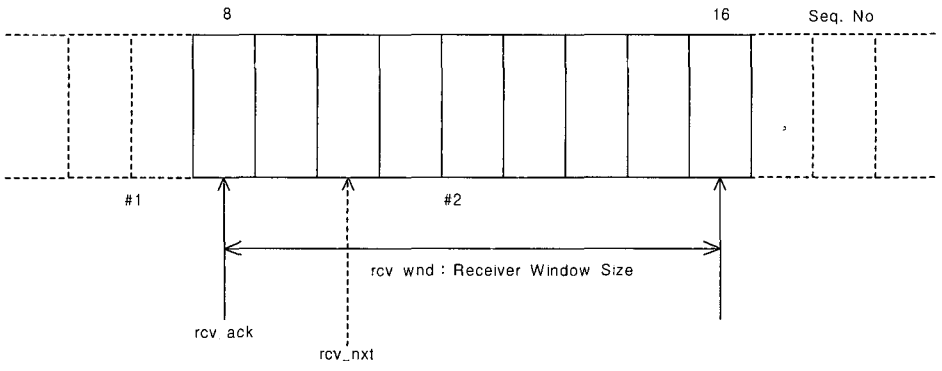
TRTP의 재전송은 송신자나 그룹관리자의 재전송 타이머가 종료하거나, 일반 참가자로부터 NACK 메시지를 받았을 때, 그리고 송신자나 그룹 관리자가 하나의 참가자로부터 중복된 ACK 번호를 3번 이상 받았을 때 수행한다.

재전송 타이머가 종료되거나 NAK 메시지를 받았을 경우 일정 시간을 기다린 후 데이터를 재전송한다. 이는 같은 데이터에 대한 오류 복구 요청이 들어오는 것을 체크하기 위해서이다. 데이터 재전송 시 같은 데이터에 대한 오류 복구 요청의 수를 계산하여 멀티캐스트로 전송할 지 유니캐스트로 전송할지를 선택한다. 본 논문 구현에서 특정 패킷에 대해 하나의 참가자만이 데이터를 손실하였을 경우 유니캐스트로 재전송하고, 그렇지 않을 경우는 멀티캐스트로 재전송한다.

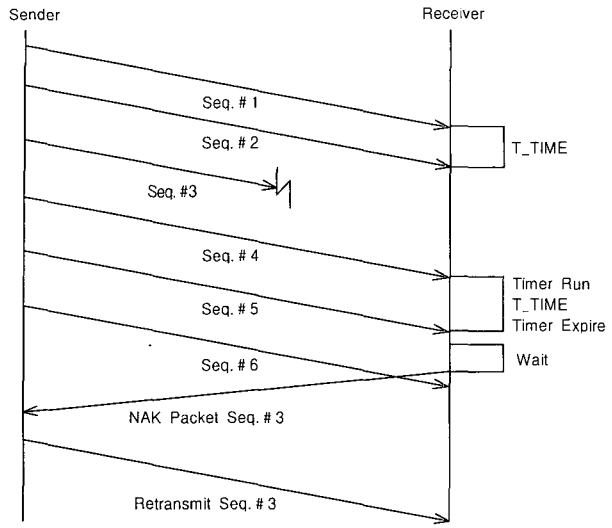
만약 하나의 참가자로 중복된 ACK 번호를 여러 번 받았을 경우 해당 패킷에 대해 즉시 재전송을 수행한다. 이 방법은 재전송 타이머가 종료된 후 재전송하는 보다 더 빠른 오류 복구를 수행한다.

송신자나 그룹 관리자는 재전송된 데이터에 대해 지수적인 백 오프를 통해 재전송 타이머 값을 증가시킨다. 만약 같은 데이터에 대해서 3번 이상 재전송이 실패하면 해당 참가자를 TRTP세션에서 제거한다.

참가자로부터의 재전송 요청은 참가자의 NAK에 의해 수행된다. 참가자는 참가자 타이머 시작 조건을 검사하여 타이머를 동작시킨다. 만약 참가자 타이머가 종료되면 NAK 메시지의 폭주를 막기 위해 다시 일정한 시간만큼 기다린 후 NAK을 보낸다. 손실된 패킷을 발견한 후 NAK메시지를 전송하는 과정은 (그림 3.10)과 같다.




(그림 3.9) 참가자 윈도우 정보




(그림 3.10) 참가자의 재전송 처리 과정

3.3.2 흐름 제어

TRTP는 흐름 제어를 위해 송신자와 참가자의 상태를 고려한 window-based와 rate-based 흐름 제어 기법을 같이 사용한다. 멀티캐스트 전송에 있어서 흐름 제어는 양단간의 공정성(fairness)을 고려해야 한다. 만약 송신자 측만 고려한다면 참가자의 버퍼 넘침(buffer overflow) 현상이 발생한다. 반대로 참가자의 상태만 고려한다면 적절한 처리량을 제공하지 못한다. window-based 흐름 제어를 위해 송신자의 congestion window(cwnd)와 참가자의 advertised window (rwnd)을 사용한다. 송신자는 항상 min(cwnd, rwnd) 값까지 데이터를 전송함으로써 흐름 제어를 수행한다. rwnd 값은 각 그룹 관리자들이 ACK 메시지에 자신의 버퍼 크기를 포함하여 상위 그룹 관리자에게 보낸다. 이 메시지를 받은 상위 그룹 관리자는 자신의 버퍼 크기와 하위 그룹 관리자로부터 받은 버퍼 크기들간의 최소값을 계산한 후 그 값을 또 다른 상위 그룹 관리자에게 전송한다. 이러한 과정을 반복적으로 수행하여 송신자에게 전송하게 된다. 송신자는 그룹 관리자들의 버퍼 크기 중 가장 적은 값을 유지하고 이 값을 이용하여 송신자의 윈도우 크기를 결정한다.

rate-based 흐름 제어는 전송 지연 시간을 설정하여 사용한다. 송신자는 자신이 원하는 지연 시간을 설정하거나 아니면 rate-based 흐름 제어를 설정하지 않을 수도 있다. 전송 지연 시간의 설정은 정적인 방법으로 사용자의 옵션에 의해서 지정한다.

3.3.3 정체 제어

TRTP의 정체 제어는 congestion avoidance와 slow-start 기법을 사용한다. TRDMP는 정체를 제어를 수행하기 위해 congestion window(cwnd), 참가자의 advertised window (rwnd)와 slow-start threshold(ssthresh)값을 유지한다. 초기에 데이터를 전송할 때에 congestion window의 크기는 1이 되며, 각 데이터에 전송시 모든 지역 그룹 참가자로부터 응답이 도착 할 경우 cwnd의 크기를 1씩 증가시킨다. 송신자는 항상 cwnd와 rwnd 두 값의 최소값 크기만큼 데이터를 전송시킨다. 만약 데이터 전송 중 정체가 발생하면 cwnd의 값과 ssthresh값을 감소시켜 정체를 제어한다. cwnd와 ssthresh 값을 비교하여 만약 cwnd 값이 크면 slow-start 방법을 이용하고 그렇지 않다면 congestion avoidance를 이용하여 정체 제어를 수행한다. (그림 3.11)은 TRTP의 정체 제어를 수행하기 위한 알고리즘을 나타낸다. 여기서 twnd은 전송 window의 사이즈이다.

```
cwnd = 1kbyte , ssthresh = 65kbytes ;
twnd = min(cwnd, rwnd)
while (up to twnd) {
        transfer_data ;
        if (response of all receivers == ACK) {
                if (cwnd <= ssthresh)  /* slow start 수행 */
                        cwnd = cwnd + 1 ;
                else  /* congestion avoidance 수행 */
                        cwnd = 1/cwnd
        }
        if (TIMEOUT) {
                cwnd = 1 ;
                ssthresh = twnd /2
                retransmit_data
        }
        if (response of receiver == duplicate ACK)
                        ssthresh = twnd/2
                        retransmit_data  /* fast retransmit */
                        cwnd = ssthresh + 3 ; /* fast recovery */
        }

        twnd = min(cwnd, rwnd)
}
```

(그림 3.11) TRTP의 정체 제어 알고리즘

3.3.4 흐름 제어와 정체 제어를 위한 차별화 된 그룹 구성

일반적으로 거의 모든 신뢰성 있는 프로토콜들은 송신자에 의해 흐름 제어 및 정체 제어가 이루어지고 각 참가자들에 대해서는 고려하지 않는다. 두 개의 참가자가 있을 경우 하나의 참가자는 다른 참가자에 비해 상대적으로 처리 능력이 낮을 수 있다. 이럴 경우 각 참가자의 상황을 고려하여 흐름 제어와 정체 제어를 수행 한다면 전체적인 서비스의 품질이 향상 될 수 있다.

본 논문에서 각 참가자의 상황을 고려한 차별화 된 제어 서비스를 제안한다. 이러한 차별화 된 제어 서비스는 송신자의 데이터 전송에 대한 부담을 최소화하고 각 참가자의 처리량을 최대화하기 위해 최대 3개의 전송 그룹으로 나누어질 수 있다. 각각의 전송 그룹은 각기 다른 흐름 제어와 정체 제어를 수행한다.

차별화 된 흐름 제어와 정체 제어를 위해 송신자는 주기적으로 ECHO_GLOBAL 메시지를 전송하며, 이 메시지는 종단 그룹 관리자가 속한 그룹의 참가자로부터 보낸 응답 메시지를 이용하여 지연시간을 계산한다. 이렇게 계산된 지연 시간을 토대로 3개의 등급으로 그룹을 구분한다. 3개의

등급은 각 그룹의 지연시간에 대한 평균을 구하고 평균 지연시간 를 이용하여 하위, 중간, 상위 등급을 계산 한다. 이는 각 등급에 따라 각기 다른 흐름 제어와 정체 제어에 필요한 파라미터를 재 정의하기 위함이다.

차별화 된 그룹을 구성하기 위한 절차는 다음과 같다.

1) 전체 그룹의 평균 지연 시간 계산

- 송신자는 주기적으로 ECHO_GLOBAL 메시지를 보낸다.
- 메시지를 받은 참가자들은 자신이 종단 그룹의 참가자이면 그룹 관리자에게 응답을 보낸다.
- 그룹 관리자는 자신이 ECHO_LOCAL에 의해 계산된 각 참가자들의 RTT 값들 중 가장 큰 값을 다시 상위 그룹 관리자에게 보낸다. 그리고,
- 상위 그룹 관리자들은 다시 하위 그룹 관리자로부터 받은 값 중 가장 큰 값을 상위 그룹 관리자에게 보낸다. 이러한 값들은 송신자에게까지 전달된다.
- 송신자는 자신의 하위 그룹 관리자들로부터 받은 RTT 값을 이용하여 평균 지연 시간(AVR_RTT)을 계산한다.
- 평균 지연 시간을 계산하는 수식은 다음과 같다(그림 3.12 참고).

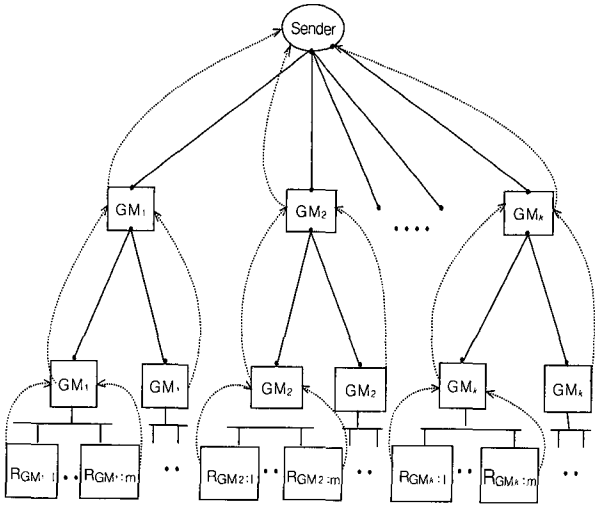
2) 계산된 평균 시간을 토대로 임의의 지연 시간 $\alpha$를 설정한다.

- 하위 그룹 = 각 그룹의 지연 시간 < (AVR_RTT − $\alpha$)
- 중간 그룹 = (AVR_RTT − $\alpha$) < 각 그룹의 지연 시간 < (AVR_RTT + $\alpha$_)
- 상위 그룹 = 각 그룹의 지연 시간 > (AVR_RTT + $\alpha$)

3) 송신자가 위의 그룹을 구분한 후 새로운 전송 그룹 주소를 부여한다.

$$AVR\_RTT = (\sum_{i=1}^{M} \max_{k=1\sim M} f(GM_i, R_{GM_i:k}))/M$$

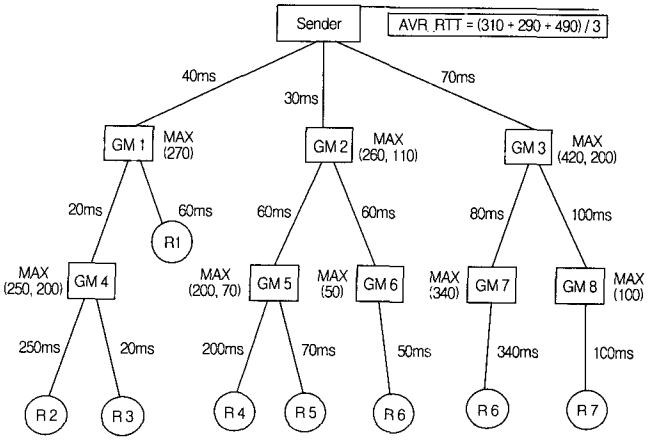
$$f(GM_i, R_{GM_i:k}) = RTT(GM_i, R_{GM_i:k})/2$$



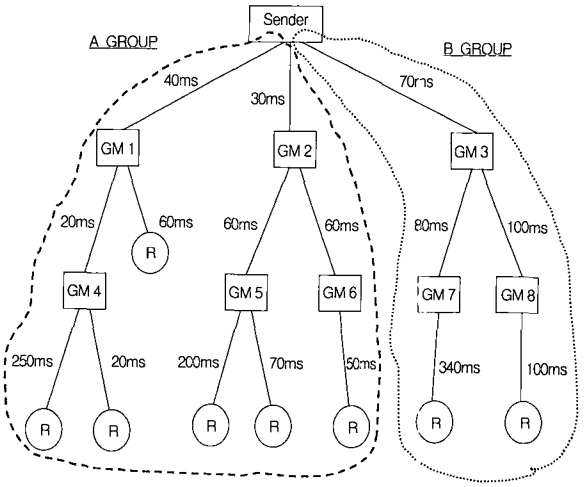

(그림 3.12) RTT 계산

이러한 과정이 거쳐서 TRTP의 프로토콜의 전송 그룹은 최대 3개의 그룹으로 나누어 진다. 이 후, 송신자는 새로운 데이터를 전송할 때 같은 데이터를 3개의 전송 그룹으로 멀티캐스팅 한다.

(그림 3.13)과 같은 제어 트리를 이용하여 차별화 된 서비스 그룹을 구성한 형태는 (그림 3.14)와 같다. 여기서 $\alpha$의 값은 100ms으로 설정하였다. 그림에서 GM1은 단지 GM4의 값만을 이용하였다. 이는 ECHO GLOBAL 메시지에 대한 응답은 단지 종단 그룹 관리자의 참가자들만이 각 그룹 관리자들에게 응답하기 때문이다. 즉 R1은 중간 그룹 관리자에 속해있는 참가자이기 때문에 ECHO GLOBAL에 대한 응답 메시지를 보내지 않았다.




(그림 3.13) TRDMP의 계층적인 제어 트리

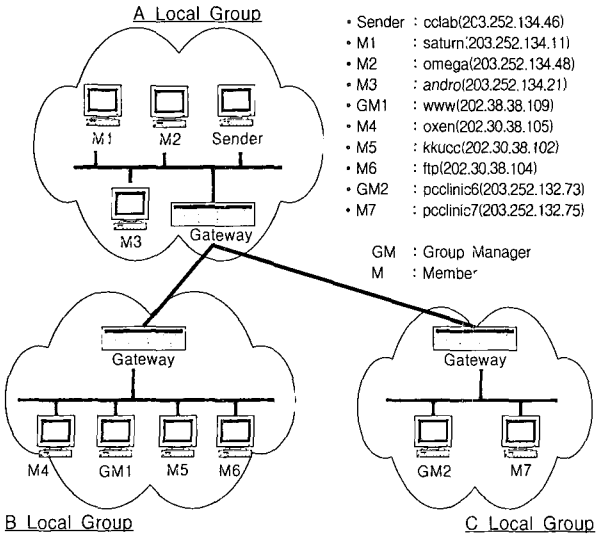



(그림 3.14) TRDMP의 차별화 된 계층적인 제어트리

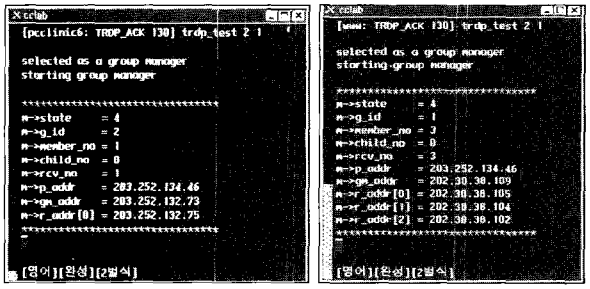
## 4. 구현 결과

TRDMP 구현 결과를 실험하기 위해 (그림 4.1)과 같은

테스트 베드를 구축하였다. 설정으로는, 지역 그룹을 생성하기 위한 TTL은 16, 데이터 전송을 위한 TTL은 32로 설정하였다. 테스트 결과 세 개의 지역 그룹으로 구성이 되었고, (그림 4.2)은 DCTCP 프로토콜을 통해 구성된 각 지역 그룹의 그룹 관리자들이 유지하고 있는 그룹 정보를 보여주고 있으며, 각 그룹 관리자들에 등록된 지역 그룹 참가자들의 정보를 확인 할 수 있다.



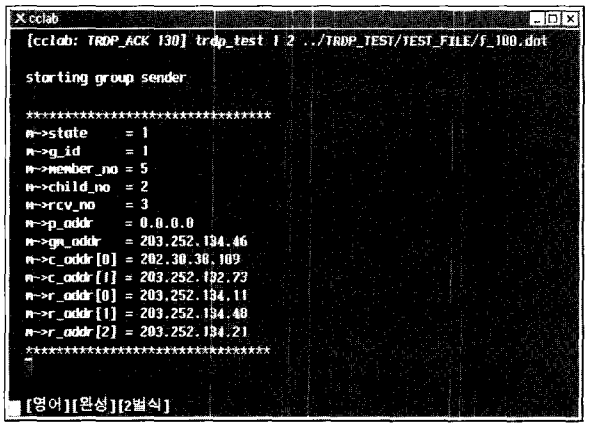

(그림 4.1) TRDMP 실험 환경



a. 그룹관리자 cclinic 6이 유지하고 있는 그룹 정보



b. 그룹 관리자 www가 유지하고 있는 그룹 정보




c. 그룹 관리자 cclab이 유지하고 있는 그룹 정보

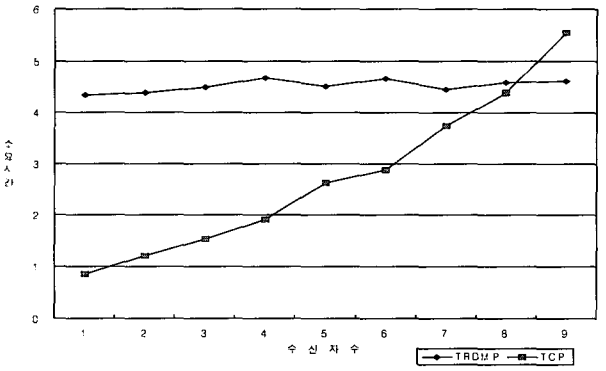
(그림 4.2) DCTCP를 통해 구성된 지역 그룹들의 그룹 정보

## 5. TRDMP에 대한 성능 평가

성능 평가는 (그림 4.1)의 실험 환경에서 실시 되었다. 테스트 항목으로는 TRDMP와 TCP를 사용하여 동일한 양의 데이터를 전송할 때 걸리는 시간을 측정하였으며, 수신자 수가 증가함에 따라 양 프로토콜간의 소요 시간 차이를 측정하였다. TRDMP를 TCP와 비교를 한 이유는, 기타 신뢰성 있는 멀티캐스트 전송 프로토콜의 구현을 구하기 어려웠기 때문이다. TRDMP와 TCP의 전송시간 비교를 위해 TRDMP의 전송률은0.02msec/MSS로 MSS의 크기는 1400bytes로 설정하여 총 200Kbytes를 전송하여 실험하였다.

〈표 5.1〉 TRDMP와 TCP 성능평가 결과

| 수신자 수 | 소 요 시 간 (초) | |
|---|---|---|
| | TRDMP | TCP |
| 1 | 4.347618 | 0.857359 |
| 2 | 4.391827 | 1.204128 |
| 3 | 4.500992 | 1.534029 |
| 4 | 4.678454 | 1.915801 |
| 5 | 4.505827 | 2.630507 |
| 6 | 4.656629 | 2.876575 |
| 7 | 4.458696 | 3.731406 |
| 8 | 4.584246 | 4.388510 |
| 9 | 4.618820 | 5.543099 |

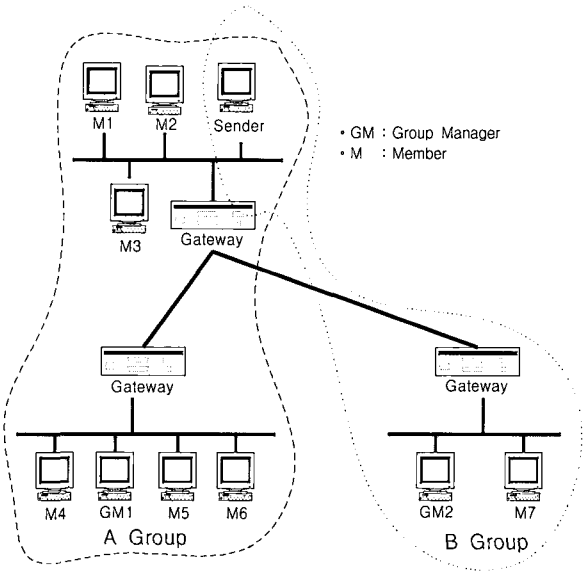



(그림 5.1) TRDMP와 TCP 비교

<표 5.1>은 사용자 수의 증가에 따른 전송시간의 증가율을 TRDMP와 TCP를 사용하여 테스트 한 결과이다. (그림 5.1)을 통해 일-대-다의 전송시 TRDMP 이용하여 데이터를 전송하면 수신자의 수에 관계없이 전송시간이 일정한 것을 볼 수 있다.

(그림 4.1)의 실험환경에 차별화된 제어 서비스를 적용하였을 경우 그 결과는 (그림 5.2)와 같으며, 차별화 된 제어 서비스에 대해 성능평가를 실험하였다. 차별화 된 제어 서비스를 통해 임의의 A와 B 두개의 전송 그룹으로 나누었다. A 그룹내의 참가자에서 패킷 손실은 0%로 B 그룹내의

패킷 손실을 10%로 설정하였으며, 전송지연을 0.2msec/ MSS로 설정하여 총 200kbytes을 전송하였다. 그 결과 A 그룹의 평균 전송시간은 30초 정도이며 B 그룹의 전송 시간은 35초가 나왔다. 데이터의 전송량이 많거나 또는 패킷 손실이 많을 경우 A와 B 그룹간의 전송시간 차이는 더 증가한다.




(그림 5.2) 차별화 된 제어 그룹

## 6. 결 론

신뢰성 있는 전송서비스는 현재 다양한 분야에서 연구되고 있다. 신뢰성 프로토콜들은 오류 제어, 흐름 제어, 정체 제어를 효율적으로 제공되어야 한다. 본 논문에서는 이러한 필요 조건들을 효율적으로 제공하기 위해 일대다 응용에 적합한 트리 기반의 신뢰성 있는 전송 프로토콜(TRDMP)을 설계 및 구현하였다. TRDMP는 다음과 같은 특징을 갖는다. 첫째, 송신자 기반과 수신자 기반의 신뢰성 있는 프로토콜들의 문제점을 해결하기 위해 계층적인 제어 트리를 구성하였다. 즉 수신자 기반의 프로토콜에서 발생되는 버퍼 넘침과 전송 지연에 대한 문제점을 해결하였고, 송신자 기반의 프로토콜의 ACK 폭주를 해결하였다. 둘째, TCP와 유사한 효율적인 흐름 제어와 정체 제어를 제공하였다. 그리고 그 흐름 제어와 정체 제어 기법을 멀티캐스트 환경에 효율적으로 적용하는 방안을 제시하였다. 셋째, 차별화 된 제어 서비스를 제공하였다. 마지막으로 각 그룹 관리자가 장애 발생시 fail-over 대책을 마련하였다.

이 TRDMP는 전송지연과 오류 복구 지연시간을 최소화하였으며, 그룹 관리자의 결함 허용을 빠르게 처리하였다. 또한 각각 차별화된 흐름 제어와 정체 제어 서비스를 제공하였다. 차별화된 제어 서비스의 장점은 분산 파일 전송인 경우 각 수신자들이 하나의 흐름 제어와 정체 제어를 통해 데이터를 전송 받을 때 모든 수신자들의 데이터 수신 시간이 유사한 것에 비해, 차별화 된 서비스를 통해 각 수신자들의 데이터 수신 시간에 차이가 발생한다. 이러한 서비스는 웹 캐싱이나 분산 파일 전송과 같은 응용에 적용된다면 더 많은 효율성을 제공 받을 수 있다.

## 참 고 문 헌

[1] S. Deering, "Host Extension for IP multicasting," RFC 1112, August, 1989.

[2] S. Armstrong, A. Freier and K. Marzullo, "Multicast Trans 'port Protocol," RFC 1301, February, 1992.

[3] S. Casner, "Frequently Asked Questions (FAQ) on the Multicast Backbone (MBone)," Aug. 1994.

[4] A. Mankin, A. Romanow, S. Brader, and V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols," RFC 2357, June, 1998.

[5] S. Pingali, D. Towsley and J. Kurose, "A Comparison of Sender Initiated and Receiver-Initiated Reliable Multicast Protocols," In Proceedings of ACM SIGMETRICS, May, 1994.

[6] S. K. Kasera, J. Kurose, and D. Towsley, "A Comparison of Server Based and Receiver Based Local Recovery Approaches for Scalable Reliable Multicast," CMPSCI Technical Report TR 97 69, Dec. 1997.

[7] B. N. Levine and J. Garcia Luna Aceves, "A Comparison of Known Classes of Reliable Multicast Protocols," Proceedings of IEEE ICCC, November, 1996.

[8] W. R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, Jan. 1997.

[9] R. Yavatkar, J. Griffioen, and M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications," *In Proceedings of the ACM Multimedia '95 Conference*, November, 1995.

[10] Markus Hoffman, "Scalable Multicast Communication in Wide Area Networks," Ph.D. Thesis in German, University of Karlsruhe, February, 1998.

[11] S. K. Kasera, J. Kurose and D. Towsley, "Scalable Reliable Multicast Using Multiple Multicast Groups," CMPSCI Technical Report TR 96 73, October, 1996.

[12] R. Yavatkar and J. Griffioen, "Reliable Dissemination for Large Scale Wide Area Information Systems," In the Proceedings of the 3rd IEEE HPCS'95, August, 1995,

[13] J. C. Lin and S. Paul, "RMTP : a reliable multicast transport protocol," in IEEE Infocom, (San Fransisco, Californiali fornia). Mar. 1996.

## 김 영 재

e-mail : ceo@nanumbiz.com

1996년 건국대학교 산업대학원 전자계산학과 (공학석사)

2000년 건국대학교 대학원 컴퓨터공학과 (박사수료)

1982년~1985년 국방부 회계주사

1985년~1999년 KHF 과장

1997년~현재 안양대학교 겸임교수

2000년~현재 ㈜나눔비즈닷컴 CEO

관심분야 : 멀티캐스트, 리룩스, 분산처리

## 박 은 용

e-mail : eypark@cclab.konkuk.ac.kr

1998년 건국대학교 전자계산학과(공학사)

2000년 건국대학교 대학원 컴퓨터공학과 (공학석사)

2000년~현재 건국대학교 대학원 컴퓨터공학과 박사과정

관심분야 : 멀티캐스트

## 안 상 준

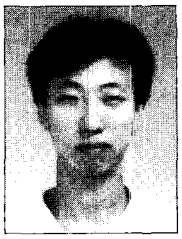
e-mail : sjahn@cclab.konkuk.ac.kr

1994년 서울산업대 전자계산학과 (공학사)

1996년 건국대학교 대학원 컴퓨터공학과 (공학석사)

1999년 건국대학교 대학원 컴퓨터공학과 (공학박사)

관심분야 : CDN, RMT, Traffic Measurement, Multicast NMS



## 현 호 재

e-mail : hjhyun@cclab.konkuk.ac.kr

1997년 동국대학교 전자계산학과 졸업 (공학사)

1999년 건국대학교 컴퓨터공학과 졸업 (공학석사)

1999년~현재 건국대학교 컴퓨터공학과 박사 재학 중

관심분야 : Multicast, Security, QoS, Mobile IP, Key Management

## 한 선 영

e-mail : syhan@cclab.konkuk.ac.kr

1977년 서울대학교 전산학과(학사)

1979년 한국과학기술원 전산학(석사)

1988년 한국과학기술원 전산학(박사)

1981년~현재 건국대학교 전산학과 교수

관심분야 : 인터넷 프로토콜, 네트웍 멀티미디어