트리기반 멀티캐스트 환경에서 멀티미디어 스트림을 위한 선행에러복구 방안

김 기 영† · 윤 미 연† · 신 용 태††

요 약

멀티캐스트 방식은 네트워크 자원을 효율적으로 사용할 수 있다는 장점을 갖는 반면 신뢰성을 기본적으로 제공하지 않기 때문에 이를 해결하기 위한 다양한 연구가 진행되어 왔다. 하지만 기존 연구는 패킷기반의 신뢰성 보장에 국한되어 있기 때문에 멀티미디어 데이터를 전송하는 경우에는 프레임간의 의존성과 적절한 시간에 수신측에서 재생되어야하는 제약을 해결하지 못한다. 따라서 멀티미디어의 특성을 고려한 신뢰성과 확장성을 보장하는 연구가 필요하다. 본 논문에서는 이와 같은 문제를 해결하기 위해서 신뢰성을 지원하는 트리기반 멀티캐스트 상에서 H.263의 특성을 인식하여 확장성을 보장하는 선행에러복구(PER)을 제안한다. 제안하는 방식은 수신노드의 상태를 고려하여 제어패킷의 발생을 줄이고 언더플로우 상태의 수신노드의 스트림 버퍼의 빠른 복구를 가능하게 한다. 성능분석 결과, 제안하는 방식은 멀티미디어 데이터에 적용하는 경우 확장성에서 기존 연구보다 우수함을 보였다.

Preceding Error Recovery Algorithm for Multimedia Stream in the Tree-based Multicast Environments

Kiyoung Kim† · Miyoun Yoon† · Youngtae Shin††

ABSTRACT

IP Multicast is required of more little network resources than one in unicast. Furthermore, reliable multicast has been researched for supporting reliability at IP Multicast mechanism. Although these studies are carried out, they only have focused on general data. In other words, in case that real-time packet, they can not support reliability since they do not consider real-time properties such as dependency of inter-frame and playback in time. Besides, we also request to support scalability because we are based on Mobile IP network together with internet. Thus, we need a mechanism to guarantee reliability and scalability of real-time stream data. In this paper, we propose PER (Preceding Error Recovery) that reflect characteristics of the real-time data, especially for H.323. PER provides scalable reliability because it is based on tree-based multicast basically and helps to support scalable relibility as reducing control packet and recovers stream buffer space from underflow status as soon as possible. PER shows much better scalable and reliable than existing works.

키워드 : 멀티캐스트(Reliable Multicast), 신뢰성, QoS, 패킷기반(Packet-base), H.263

1. 연구 배경

인터넷의 폭 넓은 보급은 사용자와 데이터를 다양화시키고 있다. 초기 텍스트(Text) 기반으로 시작된 인터넷은 현재 음성, 영상과 같은 멀티미디어 데이터를 전송하는 중요한 백본망으로 사용되고 있다. 하지만 멀티미디어 데이터는 텍스트 데이터보다 많은 네트워크 자원을 필요로 하며, 일정범위 내의 에러를 허용하고 시간에 민감한 특징을 갖는다.

인터넷 프로토콜들은 신뢰적인 텍스트 전송을 고려하여 설계되었기 때문에 멀티미디어 데이터 전송을 전송하는 경우에 에러발생시 지연(Delay)이나 지터(Jitter) 발생하게되며, 동일한 멀티미디어 데이터가 중복 전송되는 문제점이 있다. 따라서, 멀티미디어 데이터를 효율적으로 전송하기 위해서는 신뢰성 보장 외에 멀티미디어 데이터 전송에 적합한 프로토콜이 필요하다. 이와 문제를 해결하기 위해 IETF RMT (Reliable Multicast Transport) WG에서 연구가 활발히 진행 중에 있다.

† 준 회 원 : 숭실대학교 대학원 컴퓨터학과
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2002년 10월 9일, 심사완료 : 2003년 4월 24일

멀티캐스트 전송 방식은 멀티미디어 데이터 전송에서 발생되는 데이터의 중복 전송을 해결하여 네트워크 자원의 효율적인 사용을 가능하게 하였다[1]. 멀티캐스트는 네트워크 계층에서 동작되며 라우팅(Routing)과 흐름제어(Flow Control) 기법으로 구분될 수 있다. 라우팅 기법으로는 DVMRP[2], MOSPF[3], CBT[4]등이 있으며, 멀티캐스트에 필요한 라우팅 기능을 제공한다.

하지만 멀티캐스트는 기본적으로 신뢰성을 보장하지 않기 때문에 신뢰성을 지원하는 흐름제어 기능이 추가되어야 한다. 멀티캐스트 환경에서 신뢰성 보장을 위한 흐름제어 기법은 송신자 기반(Sender-Initiate)과 수신자 기반(Receiver-Initiate)방식으로 분류되며, 확장성 지원을 위해 지역 재전송을 이용하여 송신노드의 부하와 제어패킷을 줄이도록 한 지역회복 기법[5] 등이 제시되었다. 지역그룹 방식(Local Group Concept)은 네트워크 상의 물리적 거리를 이용하여 수신노드들의 패킷손실을 지역별로 복구하는 방식으로, 그룹 내의 다수의 수신노드가 동일한 패킷에 대해 패킷손실로 인한 제어패킷 NAK의 폭주를 해결하였다.

수신자 기반은 패킷의 신뢰적인 전송 책임을 수신자가 수행하는 방식으로 에러가 발생하는 경우만 제어패킷 NAK (Negative Acknowledgement)을 전송한다. 신뢰성은 보장하지만 많은 수의 수신노드에서 패킷손실이 발생하면 제어패킷 NAK 폭주로 인해 확장성을 보장하지 못하는 단점이 있지만 트리기반일 경우에는 수신자가 증가하여도 확장성을 보장할 수 있다[6-8].

하지만 현재까지의 연구들은 신뢰성과 확장성을 고려하여 진행되었기 때문에 H.261[10], H.263[11]과 같은 멀티미디어 데이터 압축 알고리즘에서 사용하는 시간성(Temporal), 공간성(Spatial), 적시성(Timing)의 특성을 고려하지 않았다. 따라서, 패킷간에 데이터 의존성을 갖는 멀티미디어 전송에 적용하는 경우, 신뢰성과 확장성을 보장하지만 적시성에 위반된 데이터를 불필요하게 재전송하기 때문에 지연이 발생하고 적절한 QoS를 얻을 수 없다는 단점이 있다.

이와 같은 문제점을 해결하기 위해서는 패킷손실이 발생한 경우, 수신노드는 패킷내의 멀티미디어 데이터의 특성과 자신의 수신상태를 고려하여 재전송을 요청하여야 한다. 기존 네트워크 계층에서 지원되는 라우팅 기능 외에 멀티미디어 데이터의 특성을 인식할 수 있는 추가 구조가 필요하다. 따라서, 본 논문에서는 각각의 수신노드의 상태와 멀티미디어 데이터의 특성을 고려하여 선별적 선행에러를 지원하는 선행에러복구(PER : Preceding Error Recovery) 알고리즘을 제안한다.

제안한 PER은 트리기반의 멀티캐스트 환경이며, 수신 노드와 미디어제어자가 상호 동작한다. 수신노드는 패킷손실이 발생하면 자신의 수신상태(스트림 버퍼의 상태)와 손실된 패킷의 중요도를 제어패킷 NAK 내에 PER 플래그를 설정하여 미디어 제어자에게 전송하고, 제어패킷 NAK을 수신한 미디어 제어자는 제어패킷 NAK 내의 PER 플래그 설정을 검사하여 에러복구를 결정하게 된다. 따라서, 미디어 제어자는 제어패킷 NAK을 수신할 때 수신노드의 상태를 확인하게 되어 오버헤드 없이 수신노드들의 상태를 파악할 수 있고 적시성을 위배한 패킷을 재전송하지 않는다. 수신노드의 QoS를 저하시키지 않고 네트워크 자원을 빠른 시간 내에 복구할 수 있도록 해준다.

본 논문의 구성은 2장에서 신뢰성과 확장성을 지원하는 멀티캐스트 기술에 대해 살펴보고, 3장에서 미디어제어자와 수신노드에서 동작되는 PER 알고리즘을 대해 설명하도록 한다. 4장에서는 제안한 알고리즘을 기존연구와 비교하여 효율성을 증명하였으며, 5장에서 결론 및 향후연구를 제시한다.

2. 관련 연구

2.1 신뢰성을 지원하는 멀티캐스트

일대다 방식의 멀티캐스트에서 신뢰성을 보장하기 위한 방법으로 송신자 기반(Sender-Initiate), 수신자 기반(Receiver-Initiate)으로 구분할 수 있다.

송신자 기반 프로토콜은 송신측이 수신노드별로 ACK 목록을 관리하는 방식이다. 수신노드에서 패킷손실이 발생하면 선택적 재전송(Selective Repeat) 방식을 이용하여 에러복구를 하고, 패킷을 수신한 경우에는 점대점(Point-to-Point) 방식을 이용하여 ACK을 송신측에 전송한다[7]. 한편, ACK을 수신한 송신노드는 해당 ACK 목록을 갱신하고, 수신노드가 전송한 ACK의 손실을 탐지하기 위해 타이머를 이용한다. 타이머에 설정된 시간 내에 모든 수신노드로부터 ACK을 수신하지 못하면 송신노드는 패킷손실이 발생한 것으로 간주하고 타이머를 재 설정하고 패킷을 재전송하게 된다. 하지만 다수의 수신노드에서 패킷손실이 발생하면 수신노드에서 ACK 폭주가 발생하는 단점이 있다.

수신자 기반 프로토콜은 수신노드가 신뢰성 있는 패킷전송의 책임을 담당하는 방식으로 송신노드는 제어패킷 NAK을 수신할 때까지 계속해서 새로운 패킷을 전송한다. 송신노드는 제어패킷 NAK을 수신하면 요청 받은 패킷을 재전송한다. 송신노드는 패킷손실에 따른 재전송을 위해 송신자 기반에서 사용하는 타이머를 이용한다. 수신노드는 수신해야할 패킷 순서번호보다 큰 패킷을 수신하면 패킷손실이 발생한 것으로 판단한다. 송신노드는 전송할 데이터가 없는 경우 수신측의 타이머 종료에 따른 제어패킷 NAK 발생을

방지하기 위해 주기적인 상태정보의 전송이 필요하다. 송신자 기반 프로토콜 방식보다 제어패킷의 발생을 감소시키지만 다수의 수신노드에서 패킷손실이 발생하는 경우 제어패킷 NAK이 폭주하는 문제점이 있다.

집중형 제어와 순서적인 멀티캐스트 전달을 보장할 수 있도록 한 RMTP(Reliable Multicast Transfer Protocol)[12]와 SRM(Scalable Reliable Multicast)[13]은 제어패킷(ACK/NAK) 폭주를 해결하였지만 많은수의 수신노드가 존재하는 경우 병목현상이 발생될 수 있다. 특히, 멀티미디어 데이터 전송에 적용하는 경우에는 제어패킷의 발생회수가 급격히 증가하고 확장성을 보장하지 않기 때문에 적합하지 않다.

2.2 확장성을 지원하는 멀티캐스트

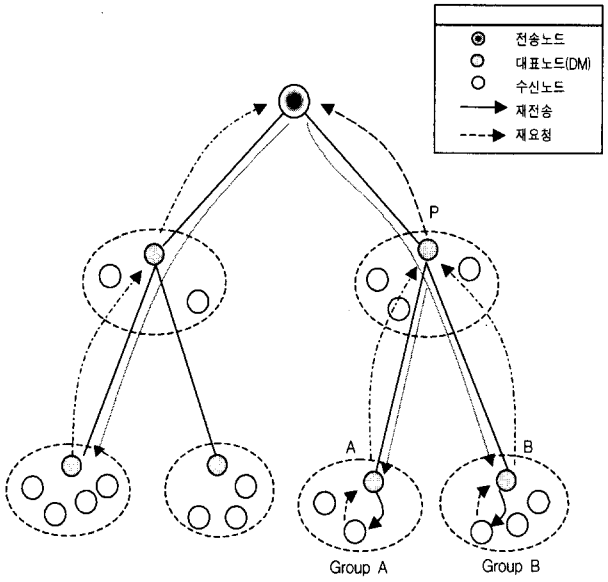
신뢰성과 확장성을 보장하는 TMTP(Tree-base Multicast Transport Protocol)[4]는 트리구조를 이용 흐름제어를 지원하는 방식으로 수신노드들은 동적으로 DM(Domain Manager)을 통해 트리에 가입한다.

수신자 기반에서 발생되는 제어패킷, 제어패킷 NAK의 폭주현상을 해결하기 위해 DM은 버퍼를 이용하여 자신에 속한 수신노드들에게 신뢰성 있는 전송을 보장한다. 만일 수신노드로부터 수신한 제어패킷 NAK에 해당하는 패킷이 DM의 버퍼에 없으면 DM은 부모 DM에게 제어패킷 NAK을 전송한다. 부모 DM의 버퍼 내에 패킷이 존재하면 재전송을 해주고 없으면 다시 상위 부모 DM에게 재전송 요청을 하는 방식으로 제어패킷 NAK의 발생을 감소시켜 확장성을 보장한다.

따라서, DM을 이용하여 제어패킷 NAK에 해당하는 패킷을 송신노드에 직접 요청하지 않고, 그룹 내에서 특정 수신노드가 패킷을 손실한 경우에는 해당 DM에 캐쉬 되어 있는 패킷을 재전송하게 된다. DM이 재전송할 수 없는 경우에는 트리로 구성되어 있는 DM을 이용하여 재전송을 시도하게된다. 그룹의 대표노드들은 송신노드까지의 홉의 수로 구성되기 때문에 트리구조에 따라 재 요청하게 되면 송신노드에 재 요청을 하지 않더라도 빠른 시간 내에 손실된 패킷을 재 요청할 수 있다.

한편, DM에서 발생되는 제어패킷 NAK 폭주를 줄이기 위해 TMTP는 제어패킷 NAK 억제(NAK Suppression)을 사용한다. 제어패킷 NAK 억제는 수신노드를 DM 단위로 그룹화하여 그룹 내의 수신노드들이 동일한 패킷손실에 대해 이론적으로 한 개의 제어패킷 NAK이 발생된다. 제어패킷 NAK 억제는 그룹에 속한 수신노드가 패킷을 손실했을 경우, 즉시 제어패킷 NAK을 발생시키지 않고 임의의 시간 동안 자신의 네트워크 인터페이스를 감시하여 동일한 패킷에 대한 제어패킷 NAK을 탐지하면 자신의 제어패킷 NAK을 취소하는 방식이다.

(그림 1)은 지역그룹 방식의 에러복구와 제어패킷 NAK의 흐름을 보여준다. 그룹 A에 속한 수신노드에서 패킷손실이 발생하면, 수신노드는 자신의 DM에게 제어패킷 NAK을 전송한다. DM은 제어패킷 NAK에 해당하는 패킷을 가지고 있는 경우에는 지역적으로 재전송 해주고 상위 DM에 추가적인 제어패킷 NAK을 전송하지 않는다. 그룹 A, 그룹 B에 속한 각각의 수신노드에서 동일한 패킷에 대해 다수의 수신노드에서 패킷손실이 발생한 경우에는, 수신노드들은 제어패킷 NAK 억제를 이용하여 자신의 그룹을 담당하는 DM에게 제어패킷 NAK을 전송한다. 제어패킷 NAK을 수신한 DM A와 B는 자신의 버퍼에 해당 패킷이 존재하면 지역적으로 재전송을 하며, 없는 경우에는 상위 DM인 P에 제어패킷 NAK을 전송한다. 이때 DM P가 수신하는 제어패킷 NAK의 수는 수신노드에 관계없이 자식 노드에 해당하는 DM에 수에 비례하게 된다.



(그림 1) 지역그룹 방식의 에러복구와 제어패킷 NAK 흐름

(그림 1)의 경우에는 2개의 제어패킷 NAK이 전송되게 된다. DM P가 재전송을 할 수 없으면 최종적으로 송신노드에게 재전송을 요청하고, 자신의 버퍼 내에 패킷이 존재하면, 상위노드인 송신노드에게 제어패킷 NAK을 전송하지 않고 지역적으로 재전송을 수행한다. TMTP는 트리내의 DM이 지역적인 신뢰성을 보장하기 때문에 패킷손실에 따른 제어패킷 NAK의 수가 감소하게 되므로 제어패킷 NAK 기반의 수신자 기반의 문제점을 해결하였다.

하지만 멀티미디어 데이터전송에 적용하는 경우에는 다음과 같은 문제점이 발생된다. 첫째, 멀티미디어 데이터는 지속적인 패킷전송이 발생하고 텍스트 데이터 보다 많은

수의 제어패킷 NAK이 발생한다. 둘째, 에러를 허용하지 않기 때문에 불필요한 재전송이 발생한다. 멀티미디어는 적시성에 위배되어 수신된 패킷은 패킷이 손실된 경우와 같은 결과를 가져오기 때문에 신뢰성만을 고려한 경우에는 오버헤드가 증가하게 된다. 셋째, DM은 수신노드들의 버퍼상태를 고려하지 않고, 제어패킷 NAK을 기반으로 재전송하기 때문에 서로 다른 수신노드의 버퍼상태를 고려하여 재전송할 수 없다는 단점이 있다.

이와 같은 문제점을 해결하기 위해 본 논문에서는 수신노드의 버퍼 상태를 제어패킷 NAK과 함께 전송하고 트리내에 위치하는 MC(Media Controller)노드가 재전송을 판단하도록 하는 PER 알고리즘을 제안한다.

3. 미디어 제어 시스템

3.1 제안하는 시스템 구성

앞 절에서 살펴본 바와 같이 기존 멀티캐스트의 연구는 네트워크 자원을 효율적으로 사용하는 동시에 신뢰성을 보장하는 것이다. 따라서, 신뢰성 보장을 고려하는 경우, 다수의 수신노드에서 패킷손실이 발생하게 되면 제어패킷 NAK의 폭주로 인해 네트워크 자원을 비효율적으로 사용하는 결과를 초래한다.

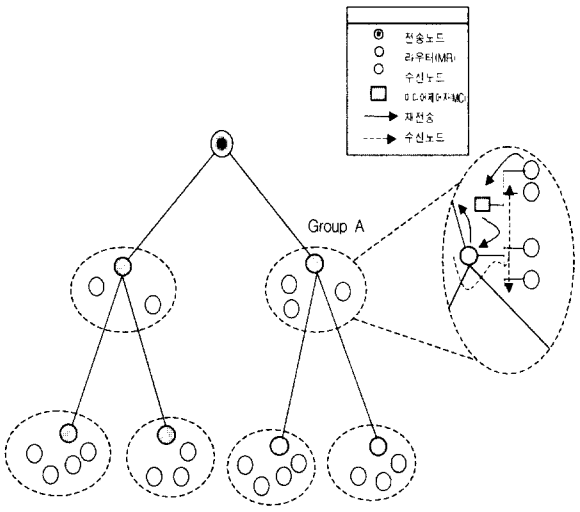
멀티미디어를 전송하는 경우에는 수신노드의 데이터 수신율이 데이터 재생율보다 같거나 커야 하는 시간제약성 때문에 신뢰성과 확장성만을 고려한 기존 멀티캐스트 방식[4]이 지원하는 패킷 복구는 시간제약성을 만족하지 못한다. 따라서, 수신노드의 QoS를 보장할 수 없으며 다수의 수신노드에서 패킷손실이 발생하면 제어패킷 NAK이 폭주하게 된다.

제안한 PER 알고리즘은 TMTP를 기반으로 하여 확장성을 보장하고, 수신노드의 버퍼상태와 손실된 패킷의 데이터 특성을 파악하여 시간성을 위배하는 경우에 선택적으로 에러복구를 포기하고, 다음 프레임의 첫 번째 패킷을 요청하도록 하여 불필요한 재전송을 감소시킨다. 따라서, 네트워크 자원을 효율적으로 사용하게 하고 수신노드의 버퍼상태를 빠른 시간 내에 복구할 수 있다.

PER 알고리즘을 적용하기 위해 기존 DM의 기능을 확장한 미디어 제어자(MC)를 사용하여 손실된 패킷을 복구하는 하도록 한다. 미디어 제어자는 RTP 헤더에 미디어 정보 필드를 이용하여 GOP 단위로 버퍼에 저장하고, I-프레임에 대해서 능동적인 재전송을 하는 방식이다.

미디어 제어자는 트리내의 노드이며 미디어 제어자에 속하는 수신노드는 브로드캐스트 환경으로 (그림 2)와 같다.

그룹 A 내에서 패킷손실이 발생한 수신노드는 제어패킷 NAK을 미디어 제어자에게 전송하고, 미디어 제어자는 자신의 버퍼에 제어패킷 NAK에 해당 패킷이 있으면 재전송한다. 이때, 미디어 제어자의 버퍼에 해당 패킷이 없는 경우에는 제어패킷 NAK 내의 PER 플래그 설정을 검사하여 송신노드에 제어패킷 NAK을 전송할 것인지 무시할 것인지 판단하게 된다. 제어패킷 NAK에 포함되어 전송되는 PER 플래그는 패킷손실이 발생한 수신노드에서 자신의 버퍼의 상태를 고려하여 설정하는 플래그이다. 이 플래그는 수신노드가 자신의 버퍼상태에 따라 제어패킷 NAK과 함께 전송하게 된다. 수신노드에서 패킷손실이 발생한 경우에만 전송되기 때문에 미디어 제어자는 각 수신노드의 버퍼상태를 확인하기 위한 추가 제어패킷 NAK을 발생시키지 않는다. 수신노드의 버퍼상태 보고는 패킷손실이 발생한 수신노드가 담당하고, 미디어 제어자는 제어패킷 NAK 수신시에 자신의 버퍼에 해당 패킷이 없는 경우에만 PER 플래그를 검사하기 때문에 미디어 제어자에 오버헤드가 집중되지 않는다.



(그림 2) 제안하는 시스템 구조

3.2 미디어 제어자의 에러복구 알고리즘

제어패킷 NAK을 수신한 미디어 제어자는 자신의 버퍼에 해당 패킷이 있으면, 단순히 재전송을 수행한다. 따라서, 미디어 제어자의 버퍼에 제어패킷 NAK에 해당하는 패킷이 저장되어 있는 경우는 기존의 지역회복 기법과 절차와 유사하다. 하지만 미디어 제어자의 버퍼 내에 제어패킷 NAK에 해당하는 패킷이 존재하지 않으면 <표 1>의 알고리즘에 따라 재전송의 수행을 판단하게 된다. 제어패킷 NAK내의 PER 플래그가 설정된 경우는 수신노드의 버퍼에 재생할 프레임이 재전송 시간동안 재생할 수 있는 데이터 보다 작다는 것을 표시한다. 미디어 제어자는 PER 플래그가 설

정된 제어패킷 NAK에 대해서는 자신의 버퍼정보를 이용하여 이후의 GOP를 요청하는 선행에러복구를 수행한다. 음성, 영상의 경우 적시성(timing)이라는 특성을 갖기 때문에 제때에 수신할 수 없는 패킷에 대해 재전송을 하지 않아도 것은 QoS에 영향을 미치지 않는다(미디어의 경우 일정 시점 이후에 수신하는 것은 의미가 없으며 영상보다 음성의 경우 그 정도가 더 크다). 미디어 제어자에서 알고리즘은 다음과 같다.

〈표 1〉 미디어 제어자의 에러복구 알고리즘

```
Primary Algorithm 1 :
1 : while (not fault) {
2 : wait until receive NAK from mi ;
3 : if (true == seek queue by sequence number)
4 :         retransmit packet, correspond to NAK ;
5 : else
6 : {
7 :         retrieve and check the information on frame type in
            NAK ;
8 :         if (flag == PER)
9 :         {     /* PER flag is set by receiver lost a packet */
10 :                    if (I-frame)
11 :                                request next First of GOP
12 :        }
13 :        else
14 :                    request retransmit to sender
15 : }
```

<표 1>에서와 같이 미디어 제어자는 제어패킷 NAK을 수신하면 해당 패킷이 자신에 버퍼에 존재하는 경우 재전송한다(4행). 미디어 제어자 내에 패킷이 없는 경우는 자신에 속한 모든 수신노드가 패킷을 수신하지 못한 경우이므로, 제어패킷 NAK 내의 PER 플래그를 확인하여 설정되어 있는 경우에는 상위 미디어 제어자에 다음 GOP의 첫 번째 패킷을 요청한다(8-11행). 미디어 제어자는 다음 GOP 해당하는 패킷을 이미 수신한 경우이면 멀티캐스트로 전송하여 패킷을 손실한 노드가 수신할 수 있도록 하고 아직 수신되지 않았으면 폐기한다. PER 플래그의 설정은 패킷손실이 발생한 수신노드들이 자신의 스트림 버퍼의 상황을 고려하여 독립적으로 제어패킷 NAK과 함께 전송하기 때문에 미디어 제어자는 추가 계산 없이 단순히 제어패킷 NAK의 플래그를 확인하고 해당루틴을 수행한다(8-12행).

3.3 미디어 제어자의 버퍼구조

트리구조로 구성된 미디어 제어자는 자신에 속한 수신노드들에 대해 지역 재전송을 담당하고, 하위 미디어 제어자의 재전송에 대비하여 버퍼에 GOP 단위로 저장한다. 저장된 GOP들은 타이머가 종료될 때 버퍼에서 삭제되고, 삭제된 공간은 GOP의 저장 공간으로 재할당한다. 타이머의 설정은 송신노드 측에서 통신 초기 RTT(Round Trip Time)를 이용하여 설정하거나 상위 미디어 제어자 하위 미디어 제어자간의 RTT를 이용하여 설정할 수 있다. 송신노드에서 전체 RTT를 이용하는 경우에는 많은 버퍼 공간을 필요로 하는 단점이 있다. 따라서, 본 논문에서는 상위 미디어와 하위 미디어 제어자간의 RTT를 이용하여 타이머를 설정하여 버퍼공간을 최소화하면서 미디어제어자 간의 패킷 손실을 가능하게 하였다. 미디어 제어자는 트리구조로 구성되기 때문에 하위 미디어 제어자의 재전송만을 고려하여도 전체 트리에 속한 미디어 제어자들의 재전송 요청을 지원할 수 있다.

한편, 수신노드로부터 수신한 제어패킷 NAK 내의 PER 플래그가 설정된 되어 다음 GOP의 첫 번째 패킷을 전송하기 위해 수신한 패킷은 GOP 단위로 버퍼에 저장한다. 이러한 구조는 상대적으로 데이터 양이 큰 I-프레임의 재전송 시간과 제어패킷 NAK 발생량을 줄일 수 있다.

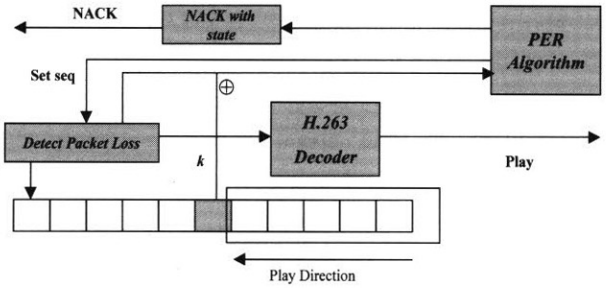
3.4 수신노드의 에러복구 알고리즘

멀티미디어 데이터 통신에서 수신노드들은 기본적으로 스트림 방식을 이용하여 미디어를 재생을 한다. 스트림 방식은 각각의 수신노드들이 일정크기의 버퍼를 이용하여 재생할 데이터를 버퍼에 저장한 후 재생을 시작하는 방식이다. 따라서, 패킷손실이 발생하여도 버퍼에 저장되어 있는 데이터를 모두 재생할 때까지 발생되는 지연과 지터를 방지하게 된다. 예를 들어, 화면이 초당 30프레임으로 구성되는 경우에 30초 크기의 버퍼를 사용하면 900프레임 이하의 에러는 재전송을 통해 복구할 수 있다. 이와 같은 구조는 음성, 화상통신에서 일반적으로 사용되고 있지만 연속된 패킷의 손실이 발생하여 버퍼내의 데이터 재생 속도가 수신되는 데이터 속도보다 큰 경우에 QoS를 만족시키지 못한다.

영상압축의 일종인 H.263으로 부호화된 영상인 경우는 I-프레임은 P-프레임의 복호화를 위한 기준 정보로 사용된다. 수신 노드에서 I-프레임을 손실하게 되면 다음 프레임에 해당하는 패킷을 정상적으로 수신하여도 완전하게 복호화 할 수 없다. 따라서, I-프레임의 손실은 최소 30프레임의 손실에 해당한다(압축률에 따라 그 이하 이상일수 있음). 이와 같은 결과는 P-프레임이 I-프레임을 기준으로 변경된 부분의 위치정보로 구성되어 있기 때문이다. 수신측에서 정확한 I-프레임을 수신하지 못하는 경우에는 공간, 시간적 에러전파로 인해 후속의 P-프레임을 에러 없이 수신하여도 QoS가 저하된다.

따라서, 기존 스트림 방식에서 연속적인 패킷손실 발생으로 인한 지연과 지터의 발생 회수를 줄이기 위해 패킷손실이 발생하면, 수신노드는 제어패킷 NAK 내의 PER 플래그

를 설정하여 패킷손실과 언더플로우 정보를 미디어 제어자에게 제공한다. PER 플래그의 설정은 패킷손실이 발생한 수신노드 별로 수행된다.



(그림 3) 수신노드의 에러복구

(그림 3)과 같이 수신노드는 현재 재생되고 있는 프레임 위치정보를 변수 k에 저장하고 패킷손실을 탐지했을 때, 제어패킷 NAK내의 PER플래그를 설정하여 재전송 요청과 수신노드의 버퍼상태 정보를 미디어 제어자에게 전달한다. 패킷에러가 발생한 수신노드는 변수 k에 저장되어 있는 현재 프레임 위치와 임계값을 비교하여 임계값보다 k의 값이 작으면, 제어패킷 NAK의 플래그를 설정하여 미디어 제어자가 자신의 상태를 파악하고 재전송을 결정하도록 한다.

수신패킷의 에러확률을 P(i), $buff_{size}$는 수신 노드의 스트림 버퍼의 크기이고, 수신 노드의 최대 버퍼 재생시간을 δ라 하면 일정 시점(t)에서 버퍼내의 데이터는 B는 식 (1)과 같이 표현할 수 있다.

$$B = \left\{ buff_{size} + \sum_{i=1}^{t} \lfloor \varepsilon P(i) - \tau \rfloor \right\} \quad (1)$$

$$D_t = B \times \frac{\delta}{buff_{size}} \quad (2)$$

임계값 Dt는 식 (2)를 이용하여 얻을 수 있으며, 일정 시점(t)에서 k값이 D_t보다 작으면 스트림 버퍼 내의 데이터가 에러복구를 하는 시간보다 작다는 것을 의미한다.

따라서, 수신노드는 미디어 데이터의 적시성을 보장하면서 에러복구를 할 수 없을 때, PER 알고리즘을 수행하게되어 불필요한 에러 복구에 따른 재전송을 억제하게 된다. 수신노드는 k값을 높여 스트림 버퍼의 언더플로우(Under Flow)에서 신속히 복구하는데 있다.

수신노드에서 동작하는 PER의 세부 알고리즘은 <표 2>와 같다. D_t가 k보다 크면 PER 플래그를 설정하지 않고 제어패킷 NAK을 전송하고, 작으면 PER을 플래그를 설정하여 제어패킷 NAK을 미디어 제어자에게 전송한다(8행). 이때, 미디어 제어자는 <표 1>의 알고리즘에 따라 자신의 버퍼에 재 요청한 GOP_n이 존재하는 경우 자신이 재전송을 수행하고 없는 경우에만 GOP_{n+1}의 첫 번째 I-프레임을 요청한다. 따라서, 미디어 제어자가 PER의 최종 동작을 결정하고 수행한다. 수신노드는 자신이 PER을 시도한 것을 플래그를 통해 저장하였기 때문에 도착한 패킷을 정상적으로 처리하게 된다.

〈표 2〉 수신노드에서의 PER 알고리즘

Primary Algorithm 2 :

```
1 : while (on session)
2 : {
3 :     wait until receive error ;
4 :       if (decision threshold(Dt) <= current buffer pointer(k))
5 :           do usual error recovery ;
6 :           continue ;
7 :     else
8 :           set PER flag in NAK ;
9 :
10      next sequence number := sequence of first frame in the
        next GOP
11 :    continue ;   /* prepare to be discarded NAK at MC */
12 : }
```

4. 분석 및 평가

4.1 분석환경

본 절에서는 제안한 알고리즘의 효율성을 입증하기 위해 확장성과 제어패킷, 제어패킷 NAK의 발생에 따른 대역폭의 활용도를 기존연구와 비교 분석하였다. 분석을 위한 네트워크 환경은 다음과 같다. 임의의 네트워크 N에서 미디어 제어자의 집합을 MC_G = { MC_1, MC_2, …, MC_i}, (0 <i < n)라고 정의한다. 미디어 제어자는 차수가 d일때 d트리로 구성되며 근 노드는 송신노드가 된다. 따라서, 전체트리 깊이가 k일때 임의의 i번째 미디어 제어자의 깊이(Level)는 식 (3)과 같다. MC_i를 근노드로 하는 서브트리를 구성하는 미디어 제어자의 수 Sub_MC(i)는 식 (3)을 이용하여 식 (4)와 같이 표현할 수 있다. 송신노드와 동일한 깊이에 i에 위치한 미디어 제어자들의 집합은 MC_{Gn}는 식 (5)와 같이 정의된다.

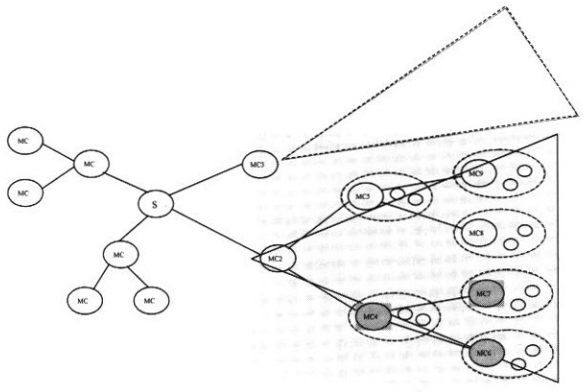
$$Level(i) = \lfloor \log_d i \rfloor + 1 \quad (3)$$

$$Sub_MC(i) = 2^{k - Level(MC_i)} \quad (4)$$

$$MC_{Gn} = \{ MC^*_{G_i} \}$$

$$where \left\{ \begin{array}{ll} G_n = 1, & i = 1 \\ 2^{level(i)-1} \le G_n \le 2^{level(i)-1} + 2^{level(i)-1} - 1, & i > 1 \end{array} \right. \quad (5)$$

네트워크 환경은 (그림 4)와 같으며 송신노드와 직접 연결된 MC_2와 MC_3로 구성된다.



(그림 4) 성능분석 네트워크 환경

MC_3의 서브 트리는 MC_2의 서브트리와 동일한 구성이며, 그림에서는 생략하였다(점선부분의 삼각형). 미디어 제어자 MC_6에서 그룹내의 수신노드의 수를 변화시켜 발생되는 제어패킷 NAK의 발생량을 분석하였으며, 확장성을 분석하기 위해 MC_4에서 서브트리의 깊이를 변화시키면서 제어패킷 NAK의 발생량을 측정하였으며 성능분석의 객관성을 위해 비교 대상인 지역그룹 방식도 동일한 조건을 가정하였다.

4.2 성능 분석

기존 방법과 비교 분석을 위해 제어패킷 발생량을 기준으로 성능을 비교하였다. 대역폭 활용도 측면에서 데이터 패킷의 양은 클수록 에러복구를 위한 제어패킷의 양은 작을수록 신뢰적이고 네트워크 자원을 사용하는 효율성이 높기 때문이다. 실험과 분석식에서 사용되는 실험변수는 <표 3>과 같다.

〈표 3〉 실험 변수

파라미터	설 명
N	임의의 네트워크
MC_G	전체 미디어 제어자(MC)의 집합
k	트리의 깊이
d	트리의 차수
m	멤버의 수
P_1	수신 노드에서 에러가 발생할 확률
F	GOP의 크기
P_2	손실된 패킷이 GOP의 I-프레임에 해당할 확률
P_{size}	패킷의 크기
i	GOP를 구성하는 I-프레임의 수

PER을 사용하지 않는 기존 지역그룹 방식(LGC)에서 발생되는 전체 평균제어패킷의 발생량 LGC_{Ts} [1, 5]에서 제시된 수식을 확장하여 식 (6)과 같이 표현할 수 있다.

$$LGC_{Ts} = 2^k - 1 \sum_{e=1}^{m} \left\{ \binom{m}{e} (1-P_1)^{m-e} P_1^e P_{size} \right\} \quad (6)$$

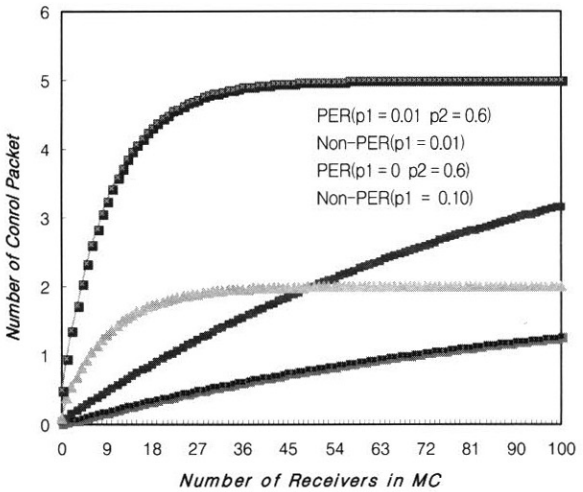
미디어 제어자를 근노드로 하는 서브트리의 깊이가 k, 멤버의 수를 m, 수신노드에서 에러가 발생 확률을 P_1이라고 하면 임의의 미디어 제어자에서 처리해야 하는 평균 제어패킷 발생량 T_s는 식 (7)과 같다.

$$T_s = Sub_MC(i) \sum_{e=1}^{m} \left\{ \binom{m}{e} (1-P_1)^{m-e} P_1^e E(i) \right\} \quad (7)$$

식 (7)에서 E(i)는 PER을 적용을 적용했을 때 발생되는 평균제어패킷 크기를 나타낸 것이다. F는 GOP의 크기이고 GOP를 구성하는 I-프레임의 수를 i, 수신노드에서 패킷손실이 발생했을 때 손실된 패킷이 GOP의 I-프레임에 해당할 확률을 P_2라하면 식 (8)과 같이 표현할 수 있다.

$$E(i) = \sum_{i=1}^{F} \binom{F}{i} (1-P_2)^{F-i} P_2^i P_{size} \quad (8)$$

(그림 5)는 분석결과를 바탕으로 임의의 미디어 제어자에서 처리해야하는 제어패킷의 크기를 제안한 방법과 기존의 방법의 지역복구 기법과 비교한 것이다.



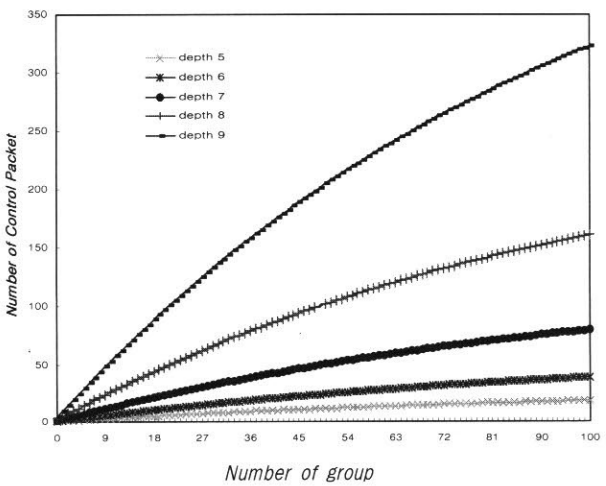
(그림 5) 에러율에 의한 제어패킷 분석

수신노드에서 패킷손실이 발생할 확률 P_1이 0.01, 패킷손실시 I-프레임일 확률 P_2가 0.6, GOP의 크기가 100일때 미디어 제어자가 처리해야하는 평균 제어패킷의 크기가 기존 방법보다 감소하는 것을 확인할 수 있다. 패킷손실시에 재전송을 통해 에러복구를 시도하지 않고 패킷내의 프레임 형식과 자신의 스트림 버퍼의 상태를 고려하여 에러복구를 하기 때문이다. 한편, 패킷손실이 높아지면 제어패킷의 발생량이 급격히 증가하지만 그룹내의 수신노드가 16개 이상인 경우에는 일정수준을 유지하기 때문에 다수의 그룹이 통신에 참여하는 환경에서 보다 효율적으로 네트워크 자원을 사용할 수 있다. 또한 패킷손실이 낮은 환경(P_1 = 0.01)에

서 발생되는 평균 제어패킷은 그룹내의 수신노드가 증가할수록 기존 방법보다 현저히 감소한다.

송신노드와 최종 수신노드 물리적 거리에 따른 패킷손실은 (그림 6)과 같다. X축은 각각의 미디어 제어자에 속한 수신노드이며 Y축은 최종 송신노드 사이의 링크에서 발생되는 제어패킷을 나타낸다.

기존 방법과 같이 미디어 제어자로 구성된 트리구조의 깊이가 깊을수록 제어패킷의 발생크기가 증가하는 결과를 확인할 수 있다. 트리의 깊이가 깊을수록 미디어 제어자의 수가 증가하기 때문이며, 미디어 제어자를 구성하는 트리의 깊이가 $k \leq 8$일때 발생되는 제어패킷의 발생 회수가 급격히 감소하는 것을 알 수 있다. 한편 트리깊이가 7이하로 구성되는 경우에는 평균제어패킷의 수가 수신노드의 수에 관계없이 일정해지는 것을 확인할 수 있다. 이는 30개의 수신노드를 멤버로 갖는 미디어 제어자가 이진트리 깊이 8개 이하로 구성되는 환경에서 수신노드 당 0.1개의 제어패킷을 처리하는 성능으로 확장성이 매우 높다는 것을 확인할 수 있다.



(그림 6) 미디어 제어자의 계층 수에 따른 제어패킷 수

한편, 멀티캐스트로 전송되는 GOP가 클수록 발생되는 제어패킷의 수가 증가하게 되는데 이는 압축비율에 따라 I, P-프레임의 비율이 결정되기 때문이며, GOP의 구성은 송신측에서 결정하는 것으로 송신측에서 그룹 수, 계층 수와 같은 통신환경에 따라 설정하여 효율성을 높일 수 있다.

5. 결 론

본 논문에서는 트리기반의 멀티캐스트 환경에서 패킷손실이 발생하는 경우 실시간 데이터의 특성을 이용하여 기존 방법보다 수신노드의 QoS를 저하시키지 않고 불필요한 제어패킷을 감소시키는 방안을 제안하였고, 구체적인 알고리즘과 수신노드에서 알고리즘 수행절차를 제시하였다. 제안한 알고리즘의 우수성을 입증하기 위해 그룹의 수, 그룹내의 수신노드에 따른 제어패킷의 평균 크기의 분석을 통해서 기존 트리방식에 널리 사용되는 지역그룹 방식과 비교를 시도하였다.

분석결과 제안한 PER 알고리즘을 사용하지 않는 기존의 트리기반의 멀티캐스트 방식인 지역그룹 방식에 비해 낮은 제어패킷이 발생하며, 패킷손실이 높은 환경에서도 동일한 결과를 확인하였다. 또한, 그룹내의 수신노드 증가하거나 트리의 깊이가 큰 경우에도 제어패킷의 발생회수는 급격히 증가하지 않아 확장성을 보장하는 것을 확인하였다.

이와 같은 결과는 제안한 PER이 각 수신노드의 스트림 버퍼의 상태에 따라 동작되고, 미디어 제어자 내에 버퍼링된 패킷을 이용하여 재전송을 수행하기 때문이다. PER은 수신노드와 미디어 제어자에서 수행된다. 각 수신노드는 패킷손실이 발생하면 자신의 스트림 버퍼 내에 버퍼링되어 있는 패킷이 재전송되는 시간동안 재생될 패킷보다 작은 경우, NAK 내의 PER을 설정하여 미디어 제어자에게 전송하여 미디어 제어자가 수신노드의 상태와 자신의 버퍼링 정보를 이용하여 재전송을 판단할 수 있도록 한다. 따라서 미디어 제어자는 NAK 내의 PER 설정을 확인하는 것만으로 수신노드의 스트림 버퍼 상태를 고려하여 재전송을 수행할 수 있다.

한편 실시간 데이터의 적시성(timing) 제약을 고려하여 NAK을 수신한 미디어제어자는 자신의 버퍼에 해당 패킷의 존재 유무를 확인하여 있는 경우에는 지역적 재전송을 통해 에러 복구을 수행하고, 존재하는 않는 경우에 NAK 내의 PER 플래그의 설정을 확인하여 PER이 설정되지 않은 경우에는 상위 미디어 제어자에 재전송 요청을 전송하고, PER이 설정된 경우에는 재전송을 무시하게 된다. 이는 PER이 설정된 경우에는 재전송을 통해 손실된 패킷을 복구하여도 수신측에서는 화면의 끊김 현상이 발생하기 때문에 불필요한 에러복구에 해당한다. 따라서 실시간 데이터의 적시성에 위배되는 에러복구를 수행하지 않기 때문에 에러복구를 위한 제어패킷의 량이 기존 방법보다 감소하며, 기존 방법과 같은 수준의 QoS를 보장할 수 있다.

결과적으로, 제안한 PER 알고리즘은 수신노드와 미디어 제어자가 상호 동작하는 방식으로 미디어 제어자에 오버헤드를 발생시키지 않고 수신노드의 버퍼상태를 고려하여 재전송을 결정하기 때문에 다수의 수신노드 참여시 발생할 수 있는 제어패킷 NAK 폭주를 해결하였고 실시간 데이터의 적시성을 고려한 에러복구 구조를 통해 수신노드에서 발생되는 제어패킷에 해당하는 NAK의 발생량을 감소시켜 제어패킷으로 인해 네트워크의 자원이 잠식되는 문제를 해

결하였다.

본 연구의 결과를 멀티캐스트 환경의 생방송, 화상회의에 적용하는 경우, 수신노드의 상태에 따라 일정 수준의 QoS를 보장하며 네트워크 자원의 효율적인 운영을 가능하게 할 것으로 기대되며, 제안한 PER 알고리즘이 H.263 형식만을 고려하였기 때문에 다양한 압축형식을 인식하고 지원할 수 있는 미디어 제어자에 대한 연구가 진행되어야 한다.

참 고 문 헌

[1] Sridhar Pingali, Don Towsley and Jim Kurose, A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols. IEEE Journal on Selected Areas in Communications, 15(3), pp.398-406, April, 1997.

[2] D. Waitzman, C. Partrigde and S. Deering(editors), Distance Vector Multicast Routing Protocol, RFC 1075, BBN STC and Stanford University, Nov., 1988.

[3] J. Moy, "Multicast Routing Extensions for OSF," Communications of the ACM, Vol.37, No.8, pp.61-66, August, 1994.

[4] R. Yavatkar, J. Griffoen and M. Sudan, "A reliable Dissemination Protocol for Interactive Collaborative Applications," Proceedings of ACM Multimedia, San Francisco, CA USA, ACM, pp.333-344, 1995.

[5] M. Hofmann, "A Generic Concept for Large-Scale Multicast," B. Plattner, Ed., Proc. International Zuerich Seminar, Volume 1044 of LNCS, Springer Verlag, pp.95-106, February, 1996.

[6] S. Paul, K. K. Sabnani, J. C. Lin, S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," IEEE Journal on Selected Areas in Communications, Vol.1,5 No.3, pp.407-421, April, 1997.

[7] B. Levine and J. J. Garcia-Luna-Aceves, "A comparison of known classes of reliable multicast protocols," Proc. Conference on Network Protocols (ICNP-96), Columbus, Ohio, 14*th* IEEE TRANSACTIONS ON NETWORKING Oct., 1996.

[8] R. Yavatkar, J. Griffioen and M. Sudan, "A reliable dissemination protocol for interactive collaborative applications," in Proc. ACM Multimedia, pp.333-44, 1995.

[9] S. Deering, "Host Extensions for IP Multicasting," *Internet RFC 1112*, Aug., 1989.

[10] Video codec for audiovisual services at p×64kbits, ITU-T Recommendation H.261, December, 1990, March, 1993(revised).

[11] Video coding for low bitrate communication, ITU-T Recommendation H.263, 1995.

[12] J. C. Lin and S. Paul, "RMTP : A reliable multicast transport protocol," in Proc. IEEE INFOCOM, pp.1414-1425, March, 1996.

[13] Jacobson, V., et al. A Reliable Multicast Framework for Lightweight Sessions and Application-Level Framing. in ACM SIGCOMM '95, 1995.

[14] Maufer, T. A., Deploying IP Multicast in the Enterprise, 1st Ed., Prentice-Hall Inc., New Jersey, pp.102-144, 1998.

[15] Ballardie, T., Francis, P. and Crowcroft, J., "Core Based Trees(CBT)," SIGCOMM '93, pp.85-95, September, 1993.

[16] B. Whetten, T. Montgomery, S. Kaplan, "A High Performance Totally Ordered Multicast Protocol," Submitted to INFOCOM '95, April, 1995.

[17] H. Schulzrirme, S. Casner, R. Frederick and V. Jacobson, "RTP : A Transport Protocol for Real-Time Applications," Internet Draft, Audio/Video Transport Working Group, IE TF, draft-ietf- avt-rtp-new-08.txt. Work in prgress, 2000.

[18] B. Girod, K. Stuhlmuller, M. Link and U. Horn, "Packet loss resilient internet video streaming," in Proceedings of SPIE Visual Communications and Image Processing, San Jose, CA, pp.833-844, January, 1999.

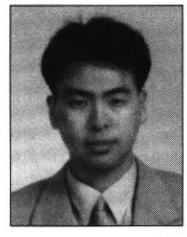
[19] Christian Maihofer and Kurt Rothermel, "a delay analysis of tree-based Reliable Multicast Protocols," In Proceedings of the International Conference on Computer Communications and Networks (ICCCN 01), January, 2001.

[20] hen Xiao, Kenneth P. Birman, Robbert van Renesse, "Optimizing Buffer Management for Reliable Multicast," IEEE International Conference on Dependable Systems and Networks(DSN '02), June, 2002.

[21] Anirban Chakrabarti and G. Manimaran, "Dynamic Multicast Routing with QoS and Reliability Constraints," DCNL Tech. Report, May, 2002.

[22] P. Druschel, M. Castro, A.-M. Kermarrec and A. Rowstron, Scribe : A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in Communications, 20, 2002.

[23] H.-Y. Tyan, J. Hou, B. Wang and Y.-M. Chen, "QoS Extension to the Core Based Tree Protocol," NOSSDAV '99, 1999.



김 기 영

e-mail : ganet89@kingdom.ssu.ac.kr

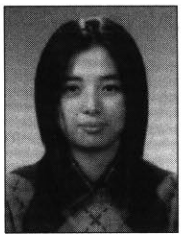
1996년 상지대학교 전자계산학과 학사

1995년~1997년 삼보정보통신 기술연구소

1999년 숭실대학교 컴퓨터학과 공학석사

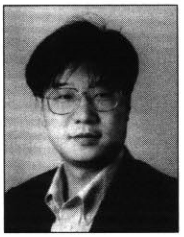
1999년~현재 숭실대학교 컴퓨터학과 박사과정

관심분야 : 멀티캐스트, 실시간 프로토콜, Mobie-IP, 정보보호



윤 미 연

e-mail : myyoon@cherry.ssu.ac.kr
2000년 가톨릭대학교 수학과/컴퓨터학과 학사
2002년 숭실대학교 컴퓨터학과 공학석사
2002년~현재 숭실대학교 컴퓨터학과 박사과정
관심분야 : 멀티캐스트, 실시간 프로토콜, Mobie-IP, 정보보호



신 용 태

e-mail : shin@comp.ssu.ac.kr
1985년 한양대학교 산업공학과 졸업(학사)
1990년 Univ. of Iowa 전산학과(석사)
1994년 Univ. of Iowa 전산학과(박사)
1994년~1995년 Michigan State Univ. 전산학과 객원교수
1995년~현재 숭실대학교 컴퓨터학부 조교수
관심분야 : 멀티캐스팅, 실시간통신, 이동인터넷 통신, 전자상거래 등