

工學碩士 學位論文

인터넷에서 고품질 오디오 스트리밍
서비스를 위한 복합적 QoS 보장기법

A Hybrid QoS Guarantee Scheme for High-Quality
Internet Audio Streaming Services

指導教授 孫 周 永

2002 년 12 월

韓國海洋大學校 大學院

컴퓨터工學科 劉 成 日

工學碩士 學位論文

인터넷에서 고품질 오디오 스트리밍
서비스를 위한 복합적 QoS 보장기법

A Hybrid QoS Guarantee Scheme for High-Quality
Internet Audio Streaming Services

指導教授 孫 周 永

2002 년 12 월

韓國海洋大學校 大學院

컴퓨터工學科 劉 成 日

목 차

| | |
|---|----|
| 제 1 장 서론..... | 1 |
| 1.1 연구 배경..... | 1 |
| 1.2 연구 목적 및 제안기법..... | 2 |
| 제 2 장 관련 연구..... | 4 |
| 2.1 오디오 스트리밍 서비스 개요..... | 4 |
| 2.2 RTP (REAL-TIME TRANSPORT PROTOCOL)..... | 6 |
| 2.3 멀티미디어 데이터 스트리밍 서비스 QOS 보장 기법..... | 7 |
| 제 3 장 오디오 스트리밍 서비스 관련 기초 실험..... | 12 |
| 3.1 인터넷 데이터 손실률..... | 12 |
| 3.2 왕복시간..... | 15 |
| 3.3 전송률에 따른 패킷 간격..... | 17 |
| 제 4 장 복합적 QOS 보장 기법..... | 19 |
| 4.1 선택적 재전송..... | 20 |
| 4.2 클라이언트 버퍼 동적 할당..... | 22 |
| 4.3 전송률 동적 제어..... | 23 |
| 제 5 장 실험 및 결과..... | 26 |
| 5.1 재전송 소요시간..... | 26 |
| 5.2 재전송 완료시 버퍼 내의 남은 재생시간..... | 28 |
| 5.3 2분 동안 재생시 수신측 재생되지 않은 버퍼량 변화..... | 29 |
| 5.4 재전송시 송신측 패킷 간격 변화..... | 30 |
| 5.5 재전송시 수신측 패킷 간격 변화..... | 31 |
| 제 6 장 결론 및 향후 연구과제..... | 32 |
| 참고문헌..... | 33 |

A Hybrid QoS Guarantee Scheme for High-Quality Internet Audio Streaming Services

Seong-il Yoo

*Department of Computer Engineering
Korea Maritime University, Busan, Korea*

Abstract

This paper describes a hybrid QoS guarantee scheme for high quality audio streaming services on the Internet. One of the continuous multimedia data, digital audio, has the timing restrictions on transmission and playback over the packet switching networks. The continuous playback of the audio data requires the isochronous transmission of the audio data packet through the Internet. In order to retain the QoS at the ultimate destination (client) as the same as servers provide, the transmission protocols should consider the error conditions such as packet loss, and out of order delivery. Generally, the protocols supporting the transmission of continuous multimedia data do not recover the errors due to the packet loss, and out of order delivery. These are working good for the audio and video streaming services with the toll quality, but not for the high quality streaming services, such as the surround hi-fi sound/music playback. The hybrid QoS guarantee scheme includes the three mechanisms to overcome the problem. First, the selective retransmission is adopted to recover the packet loss with the minimal

overhead. Second, the adaptive buffering at client-side is designed to prevent from the starvation of audio data during the retransmission. Finally, the adaptive transmission rate at server-side is also used to maintain the isochronous transmission even after the retransmission due to packet loss. The experiments haven good results of the hybrid QoS guarantee scheme for the high quality audio streaming services on the Internet.

제 1 장 서론

1.1 연구 배경

인터넷을 사용하는 사용자의 급격한 증가와 더불어 인터넷을 통한 오디오 데이터의 전송도 급격히 증가하는 추세이다. 단순한 전화통화 수준의 음성통신에서 인터넷을 이용한 온라인 교향악 콘서트 음악 감상, Hi-Fi 스테레오 FM 라디오 방송의 실시간 중계, 주문형 돌비 5.1 채널 오디오 서비스 등 고품질 오디오 서비스가 요구되고 있다. 오디오는 특성상 최종 사용자 단에서 재생되는 서비스 품질(QoS : Quality of Service)이 매우 중요하다. 인간은 비디오보다 오디오에 더 민감하기 때문이다[1]. 인터넷에서 오디오 스트리밍 서비스를 하기 위해서는 적게는 초당 56Kbits에서 1.4Mbits의 오디오 데이터를 지속적(isochronous)으로 전송해야 한다. 많은 양의 오디오 데이터를 지속적으로 시간제약성에 맞추어 전달하는 과정에서 패킷 손실, 패킷 지연, 잘못된 순서로 도착하는 현상이 발생할 수 있으며, 이러한 현상의 정도에 따라 최종 사용자 단에서의 QoS가 결정된다.

인터넷에서 오디오 데이터를 전송하는 데 일반적으로 사용되는 UDP 전송 프로토콜은 신뢰성 있는 데이터 전달을 보장하지 않으며 전송 에러에 대한 정보도 제공하지 않는 문제를 가지고 있다[2]. 특히, 전송 에러 가운데 패킷 손실에 의한 수신측 재생 음질 저하현상이 UDP 전송 프로토콜을 이용하는 인터넷 오디오 스트리밍 서비스에서 주로 발생한다. 이 현상은 일반적인 음질(toll-quality) 서비스인 경우에는 사용자에게 감지되어도 크게 문제가 되지 않으나 고품질 오디오 서비스인 경우에는 현장감의 저하로 이어져 전체 서비스 불만 요인이 된다. 이 문제를 해결하기 위한 연구들이 진행되어 왔다[3,4,5]. 대표적인 기법으로 부가전송이나 피드백 제어 기법들이 제시되어 있다[3,4]. 이 기법들의 공통적인 특징은 패킷 손실에 대해 수신측이 주도적으로 행하는 것으로

부분적인 에러 복구 방법이다. 부가전송은 앞서 전송된 패킷의 손실된 데이터 및 오류복구 정보를 다음 패킷에 포함하여 전송하며, 피드백 제어는 수신측의 주기적인 피드백 정보에 의해 송신측의 전송률 및 부호화 방법을 다르게 하여 전송하는 기법이다.

1.2 연구 목적 및 제안기법

고품질의 오디오를 결함 없이 그대로 재생하기 위해서는, 앞서 제시된 두 가지 기법으로는 부족하다. 2.3절에 자세히 설명이 되어 있듯이 부분적인 에러 복구를 통해서도 수신측에서 완전하게 데이터를 수신하기가 불가능하기 때문이다. 본 논문에서는 RTP/UDP를 이용한 고품질 오디오 스트리밍 서비스에서 클라이언트에서의 재생 품질을 서버에서 제공하는 오디오 데이터 고유의 품질과 동일한 품질로 보장하기 위해 복합적 QoS 보장 기법을 제안한다.

복합적 기법에는 크게 세 가지 기법이 포함되어 상호 보완적으로 작동함으로써 최종적인 재생 품질의 보장을 담보한다. 첫째, 전송 중에 손실된 패킷을 RTCP 피드백을 이용하여 선택적으로 재전송하고, 둘째, 재전송에 의해 발생할 수 있는 오디오 데이터의 전송 등시성(isochronous transmission) 결손을 보장하기 위해 수신측 버퍼를 동적으로 할당하여 재전송에 필요한 시간을 보장 받는다. 셋째, 서버측에서 전송률 제어를 함께 시행하여 이러한 재전송에 의해 이어지는 다음 오디오 데이터의 전송률 저하 현상을 막는다. 세 가지 기법에 의해 작동되는 제안 기법이 실제 인터넷 상에서 운용되는 데 결정되어야 하는 파라미터(전송률, 에러에 의한 재전송률, 클라이언트에서 유지해야 하는 버퍼크기)를 얻기 위해 실제 인터넷에서 많은 실험을 통해 인터넷의 전송 특성을 측정하여 도출하였다.

실험은 재전송률과 깊은 관련이 있는 인터넷에서의 RTP/UDP 데이터 전송 손실률과, 송수신함에 있어 걸리는 왕복시간(RTT : Round Trip Time)을 측정하였다. 그 결과, 인터넷 전송 손실률은 시간대별로 가변적이거나 약 5% 내외로 밝혀졌고, 전송 왕복시간도 2홉을 거치는 실험에서 약 5.3ms로 밝혀졌다. 이를 기반으로 제안된 복합적 QoS 보장기법에서 규정해야 하는 전송률과 재전송에 따른 버퍼량, 전송률 변화율을 도출하였다. 이를 기반으로 설계된 오디오 스트리밍 서비스 서버, 클라이언트 시스템을 실제 인터넷에서 운용하고 실험한 결과, 기존 방식에 비해 적은 재전송률과 버퍼량으로 고품질 오디오 스트리밍 서비스의 품질이 확실하게 보장됨을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 인터넷에서의 오디오 재생 스트리밍 서비스에서 QoS를 보장하기 위한 기존의 연구 내용을 살펴보고, 3장은 실제 인터넷의 전송특성을 파악하기 위한 기초 실험 결과를 설명하고, 4장에서는 본 논문에서 제안하는 복합적 QoS 보장 기법인 동적 버퍼 할당 기법, 재전송 요청 기법, 그리고 전송률 동적 제어 기법을 자세하게 설명한다. 5장에서는 제안된 기법을 채택한 서버와 클라이언트 시스템을 구현하여 실제 인터넷에서 적용한 실험 결과를 보이며, 마지막으로 6장에서 향후 연구 방향과 결론을 내린다.

제 2 장 관련 연구

2.1 오디오 스트리밍 서비스 개요

인터넷 상에서 전송되는 데이터는 주로 이진(binary) 파일이나 텍스트 형태의 데이터이며 수신측에서 완전히 수신한 후 보여주는 다운로드 방식이 주를 이루고 있다. TCP/IP 기반의 데이터 전송에서는 프로토콜(protocol) 자체의 에러 처리 방법에 따라 에러 없는 데이터의 수신과 재생이 가능하다. 하지만 TCP 전송 프로토콜은 다음과 같은 특성에 의해 멀티미디어 데이터의 스트리밍 전송에 적합하지 못하다[2]. 첫째, 손실 없는 데이터 수신을 위해 과도한 ACK전송과 재전송으로 인해 스트리밍 서비스의 시간제약성을 만족하지 못한다. 둘째, TCP/IP와 같은 공격적 프로토콜은 네트워크 대역폭에 여유가 있을 경우 slow-start와 같은 기법에 의해 대역폭의 대부분을 차지하게 된다. 이러한 경우 수신측의 과도한 버퍼할당과 재생되는 데이터의 실시간성을 만족시키지 못한다[6]. 오디오 스트리밍 서비스는 전체 데이터를 다운로드 후 보여주는 방식과는 다르게 부분적인 데이터라도 수신과 동시에 재생이 된다. 송신측의 전송이 되는 동안 수신측에서는 재생이 되며 송신측의 전송완료와 동시에 수신측의 재생도 완료된다. 다운로드 완료 후 보여주는 방식은 전송하고자 하는 데이터가 파일과 같이 저장된 내용이지만 오디오 스트리밍 서비스는 인터넷 방송, 스포츠 중계와 같이 실시간으로 발생하는 데이터를 전송 및 재생할 수 있으며 최근 대두되고 있는 멀티미디어 데이터의 저작권에 관련된 문제도 해결할 수 있는 장점이 있다.

인터넷 스트리밍 서비스를 가능하게 하는 기술로는 다음 세 가지를 들 수 있다. 첫째, 서버측 스트리밍 기술이다. 실시간으로 발생하는 데이터를 수집, 압축하여 저장과 동시에 인터넷을 통해 전송한다. 비디오 캡이나 오디오 입력을 통해 수집되는 데이터를 스트림이 가능한 형태로 변환하여 스트림 서버에서 네트워크를 통해 전송한다[6].

둘째, IETF(Internet Engineering Task Force)에서 표준화한 RTP/RTCP 등 일

련의 멀티미디어 데이터 전송 프로토콜이다(그림 1).

| | | | | |
|----------|-----|------|------|------|
| RSVP | RTP | RTCP | RTSP | HTTP |
| | UDP | | TCP | |
| IP | | | | |
| Ethernet | | ATM | | |

그림 1. 인터넷 멀티미디어 전송 프로토콜 스택

RSVP(Resource Reservation Protocol)는 비디오와 다른 고 대역폭의 멀티캐스트 메시지 전송을 위해 인터넷 상의 채널들이나 경로들을 예약할 수 있도록 해 주는 프로토콜이다. RTP(Real-time Transport Protocol)는 오디오와 비디오와 같은 실시간 데이터를 전송하기 위한 인터넷 프로토콜로서 UDP를 기반으로 동작한다. 자세한 내용은 2.2절에 기술되어 있다. RTCP(Real-time Transport Control Protocol)는 RTP의 QoS를 유지하기 위해 RTP와 함께 쓰이는 시그널링 제어 프로토콜이다. RTP는 데이터 전송에만 관계하지만, RTCP는 데이터 전송을 감시하고, 세션 관련 정보를 전송하는 데 관여한다. RTSP(Real-Time Streaming Protocol)는 월드 와이드 웹 상에서 멀티미디어 스트리밍 데이터를 제어하는 방법에 대한 표준안이다. HTTP(Hypertext Transfer Protocol)는 웹 상에서 파일(텍스트, 그래픽 이미지, 사운드, 비디오 그리고 기타 멀티미디어 파일)을 주고받는 데 필요한 프로토콜로서 TCP/IP와 관련된 하나의 응용 프로토콜이다. UDP(User Datagram Protocol)는 IP를 사용하는 네트워크 내에서 컴퓨터들 간에 메시지들이 교환될 때 제한된 서비스만을 제공하는 통신 프로토콜이다. 예러 없는 데이터 전송을 보장하지는 않지만 전송지연 면에서 우수한 특성을 가지고 있기 때문에 시간 제약성을 가지는 멀티미디어 데이터 전송에 적합하다. TCP(Transmission Control Protocol)는 인터넷 상의 컴퓨터들 사이에서

데이터를 메시지의 형태로 보내기 위해 IP와 함께 사용되는 프로토콜이다. 앞서 설명한 이유로 멀티미디어 데이터 전송에는 부적절한 면을 가지고 있다. IP(Internet Protocol)는 인터넷 상의 한 컴퓨터에서 다른 컴퓨터로 데이터를 보내는 데 필요한 경로설정 프로토콜이다. 비연결형 위주로 동작하기 때문에 망 내의 트래픽 부하 균등, 결합 내성이 매우 강하다. 오디오 스트림 전송은 주로 RTP/UDP를 이용하여 전송하게 되는데 동기화 및 대역폭 측정 등과 같은 실시간 스트림에서 지원해야 하는 부가기능을 지원하지 못하는 문제점이 존재한다. 이러한 문제점을 해결하기 위해 XTP(eXpress Transport Protocol), HSTP(High Speed Transport Protocol), VMTP(Versatile Message Transaction Protocol) 등의 많은 프로토콜들이 제안되었다[7].

셋째, 클라이언트측 재생 기술이다. 전송된 멀티미디어 데이터의 압축 해제 및 재생, 그리고 안정적인 재생을 위한 버퍼 제어의 기능이 내포되어 있다. 인터넷 상의 대표적인 프로그램으로 넷스케이프사의 LiveMedia™ 기술, 마이크로소프트사의 WMT(Windows Media Technology)™에 기반한 NetMeeting, Windows Media Player, Real Network 사의 Helix™ 기술에 바탕을 둔 RealPlayer 등을 들 수 있다.

2.2 RTP (Real-time Transport Protocol)

| | | | | | | | |
|--|---|---|----|---|----------------------------|--------------------|--|
| V=2 | P | X | CC | M | PT | Sequence Number | |
| Timestamp | | | | | | | |
| Synchronization Source(SSRC) Indetifiers | | | | | | | |
| Contributing Source(CSRC) Identifiers | | | | | | | |
| Defined by Profile | | | | | Length of Header Extension | | |
| Header Extension | | | | | | | |
| Padding Octets | | | | | | Number of Paddings | |

그림 2. RTP Header 구조

RTP는 유니캐스트를 비롯하여 멀티캐스트 네트워크 상에서 오디오, 비디오 데이터와 같은 연속 미디어 데이터(Continuous Media Data)를 실시간으로 처리(retrieval)하는 응용에 맞게 데이터 전송 기능을 수행하는 수송계층 프로토콜이다[7]. 프로토콜 헤더 구조에서 V는 버전, P는 Padding, X는 Extension, CC는 CSRC 카운터, M은 Marker, PT는 Payload Type 을 의미한다. RTP는 실시간 데이터의 단말-대-단말 네트워크 전송 기능을 제공하지만 전송 품질을 보장하지는 않는다. 전송 품질 감시, 매체 식별, 및 제어 기능을 담당하는 제어 프로토콜인 RTCP 와 세트를 이루며 전송 및 네트워크 계층에 독립적으로 동작한다. 응용의 필요에 따라 헤더를 추가하거나 변경할 수 있는 일종의 맞춤형 프로토콜이다[8]. 이를 위해 RTP Payload의 종류와 형식을 명시하는 Profile 문서와 RTP Payload의 전송 방법을 명시하는 Payload Format 문서를 필요로 한다. RTP 헤더의 Marker 와 PT 필드는 응용에서 변경할 수 있고, RTP 헤더의 X 비트를 이용해서 응용은 필요한 확장 헤더를 추가할 수 있다. RTCP는 데이터 패킷과 같은 방법으로 제어 패킷을 주기적으로 전송하는 프로토콜이며[9] 데이터 전송 품질에 대한 피드백(SR, RR)을 제공한다. RTP 전송자의 식별을 위해 지속적인 식별자(SDES-CNAME)를 제공하며 최소한의 RTP 세션 제어 정보(SDES, BYE, APP)를 제공한다. RTCP 메시지의 종류로는 SR(Sender Report), RR(Receiver Report), SDES(Source Description), BYE(End of Session), APP(Application-defined)등이 있다[10].

UDP는 비연결형 서비스를 어플리케이션에게 제공하는 프로토콜로서 RTP의 기반이 되는 프로토콜이다. UDP는 TCP와는 달리, 메시지를 패킷으로 나누고, 반대편에서 재조립하는 등의 서비스는 제공하지 않으며, 도착하는 데이터 패킷들의 순서를 제공하지 않는다. UDP를 사용하는 멀티미디어 데이터 스트리밍 서비스에서는 전체 메시지가 올바른 순서로 도착했는지에 대해 확인할 수 있는 메커니즘이 있어야 한다는 것을 의미한다. 이를 위해 RTP가 UDP 위에서 활용되는 것이다.

2.3 멀티미디어 데이터 스트리밍 서비스 QoS 보장 기법

멀티미디어 데이터를 전송하는 프로토콜은 수신자에게 더 많은 정보를 전달하기 위해 RTP와 같은 응용계층의 프로토콜을 사용한다. 이러한 프로토콜을 사용하는 경우 수신자는 2 ~ 5%의 패킷 손실을 경험한다[11]. 이 경우 패킷 손실로 인한 수신측의 재생품질 저하현상이 발생하며 이러한 품질저하 현상을 극복하기 위해 여러 가지 에러 복구 방법이 제안되었다.

2.3.1 부가전송(Redundant Transmission)

멀티미디어 데이터 전송시 손실된 패킷에 따른 복구 문제를 수신측에서 처리할 수 있도록 부가정보(redundant information)를 후속 패킷에 추가하여 보내는 방식이다[3]. 추가지연 없이 패킷 손실에 대한 복구 정보를 제공하여 전방위 에러수정(forward error correction)이 가능한 장점이 있는 반면 보낸 패킷에 대한 부가정보를 다음 패킷에 추가함에 따라 정보전송에 따른 전송량 증가가 요구되는 단점을 가진다[13].

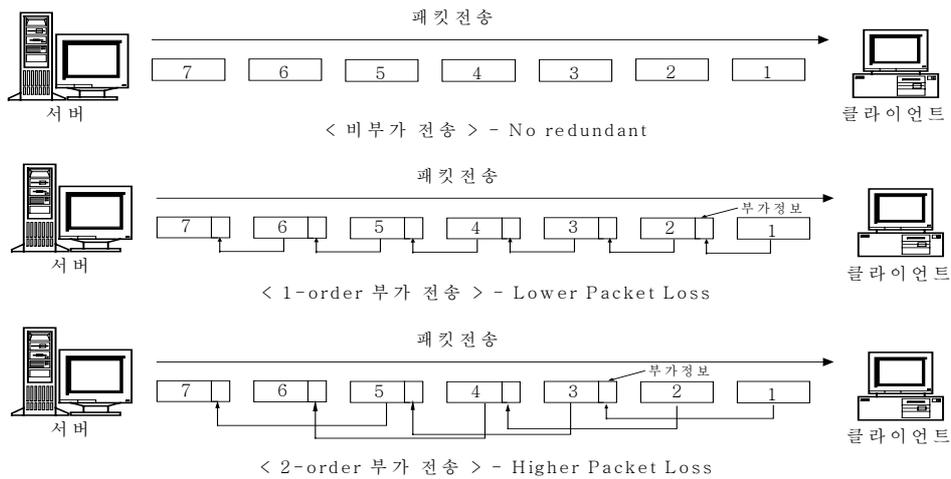


그림 3. 부가전송

부가전송 방식은 패킷 손실을 고려하여 수신측에서 복구할 수 있도록 송신측에서 패킷의 중복성을 허용하는 방식으로 이를 위한 RTP 패킷은 원본 데이터에 해당하는 PEB(Primary Encoded Block)과 패킷 손실에 대비한 부가정보에

해당하는 REB(Redundant Encoded Block)으로 구성된다[3]. REB는 패킷 손실량이 낮은 네트워크 환경에서는 REB 크기 만큼의 대역폭을 낭비하게 된다. 이 경우 수신측의 RTCP 피드백 정보를 이용하여 부가전송 메커니즘을 동적으로 적용할 수 있다.

패킷 손실이 없거나 매우 적은 경우 비부가전송에 해당하는 PEB만을 전송함으로써 네트워크 대역폭을 최소화할 수 있으며 부가전송 방식을 적용할 경우 REB를 몇 번째 후속 패킷에 부가할 것인가 하는 부가 전송 방식의 순서 문제로, 이는 패킷 손실의 연속성에 의존한다. 피드백된 패킷 손실률이 임계 값보다 큰 경우 부가전송 방식 알고리즘을 적용하여 손실된 패킷을 수신측에서 복구할 수 있도록 한다[3].

2.3.2 재전송(Retransmission)

재전송 방법은 일반적으로 오디오 데이터의 특성상 스트림 전송에 사용되지 않는 방법이다. 오디오 스트림 데이터는 송신측과 수신사이의 지연시간에 민감하기 때문에 재전송 방법을 사용할 경우 재전송된 데이터가 제한된 시간 이내에 수신측에 도착하지 않을 수 있기 때문이다. 하지만 재전송 방법이 모든 경우에 사용될 수 없는 것은 아니다. 종단간의 지연시간이 짧고 패킷 손실률이 적은 경우 사용이 가능하며 재전송에 의한 손실패킷 복구 방법은 가장 확실한 복구 방법이다[11]. 그러나 재전송 요청으로 인한 오버헤드가 증가하는 단점이 있다.

2.3.3 삽입기반 복구(Insertion-Based Repair)

삽입기반 복구 방법은 손실된 패킷을 단순한 데이터로 대체하는 방법이다. 수신측에서 수신도중 손실된 패킷 부분을 무음이나 특정 노이즈(noise), 앞서 전달된 패킷으로 대체하여 연속적인 재생이 가능하다[11]. 이 방법의 경우 부가정보 전송이나 재전송으로 인한 오버헤드는 전혀 발생하지 않으나 앞서 설명된 부가정보 전송 방법이나 재전송 방법과는 달리 손실된 패킷에 대한 복구가 전혀 이루어질 수 없기 때문에 재생품질이 현저히 저하되는 단점이 있다.

2.3.4 흐름 제어

송신측과 수신측간의 사용 가능한 대역폭 상태에 대한 수신측의 피드백 정보 전달 빈도수를 조절함으로써 효율적으로 대역폭을 이용하는 방법이다[9]. 대역폭의 부하가 적을 경우 피드백 빈도수를 적게 하고 혼잡 상태일 때는 빈도수를 많이 하여 통신망의 현재 상태를 송신측이 최대한 빨리 파악하도록 하여 전송률을 조절한다. 멀티미디어 데이터의 전송에 사용되는 UDP 전송 프로토콜이 TCP와 같은 프로토콜과 함께 사용될 경우 수신측의 피드백 정보 없이 고정적인 전송률로 전송한다면 TCP의 슬라이딩 윈도우 프로토콜(sliding-window-protocol)에 의해 대역폭이 점령당하게 되어 지속적인 스트림 전송이 어려워진다. 네트워크가 혼잡 상태일 때는 수신측의 피드백 정보 빈도수를 조절함으로써 송신측과 수신측간의 네트워크 상태를 송신측에 즉시 피드백 하여 송신측에서는 현재의 대역폭 범위에 맞는 전송을 통해 전송되는 데이터의 손실을 줄인다.

이러한 흐름제어 기법은 송수신측간의 피드백 정보를 통한 대역폭의 효율적 사용이라는 장점이 있으나, 고품질의 멀티미디어 데이터 전송시 발생하는 패킷 손실에 대해 복구할 수 없는 단점이 있다. 이것은 고품질 멀티미디어 데이터 일수록 고속의 전송률을 요구하며 흐름 제어 기법을 통해서도 패킷 손실에 대한 보장을 하기 어렵기 때문이다[5,12].

2.3.5 주기적 피드백 제어

수신측의 주기적 피드백 정보에 의해 송신측에서 부호화 방법과 전송률을 제어하는 방법이다[4]. 수신측에서 파악된 전송 상태를 송신측에 피드백 하여 전송되는 데이터량을 조절하기 위해 다음과 같은 사항을 고려해야 한다[14]. 첫째, 전송되는 데이터의 실시간성을 고려하여 시기 적절한 피드백을 제공하여야 한다. 둘째, 피드백 정보의 트래픽으로 인해 수신 정보의 품질이 영향을 받지 않아야 한다. 셋째, 지연시간에 민감한 네트워크의 혼잡상황에 대해 빠른 피드

백을 제공하여야 한다. 적절한 피드백 정보를 통해 전송 제어 효과를 얻을 수 있으나 주기적인 피드백 정보만으로는 수신측의 현재상태를 즉각적으로 피드백 할 수 없으며 통신망 혼잡으로 인해 피드백 정보가 유실되는 경우 피드백 정보 전송이 다음 피드백 정보의 전송까지 지연되므로 수신측 버퍼 고갈이나 오버플로우 현상이 나타날 수 있다.

제 3 장 오디오 스트리밍 서비스 관련 기초 실험

본 논문에서 제안하는 복합적 QoS 보장 기법이 현실적으로 적용 가능한지에 대한 기초 실험을 실시하였다. 재전송에 의한 QoS 보장 기법과 클라이언트측의 동반되는 동적 버퍼링 기법 그리고 서버측의 동적인 전송률 제어 기법이 현실적으로 적용가능성이 있는지 여부를 판단하기 위해 인터넷 전송 특성에 대한 사전 조사를 실시하였다.

3.1 인터넷 데이터 손실률

인터넷 상의 패킷 손실률은 재전송 빈도수와 비례하므로 손실률을 파악함으로써 재전송에 따른 추가적인 부하 정도를 예측할 수 있다. 이를 위해 현재 인터넷 상의 RTP 데이터가 손실되는 형태와 RTCP 피드백을 통한 재전송하는데 소요되는 시간의 측정이 선행되어야 한다. RTP 데이터가 손실되는 형태를 조사하기 위한 실험 방법은 RTP 데이터를 전송하는 서버 프로그램과 수신하는 클라이언트 프로그램을 제작하여 서버 프로그램에서 RTP 데이터 패킷을 전송하고 클라이언트 프로그램에서 패킷을 수신할 때 패킷 헤더 번호에 의한 손실된 패킷을 조사하였다.

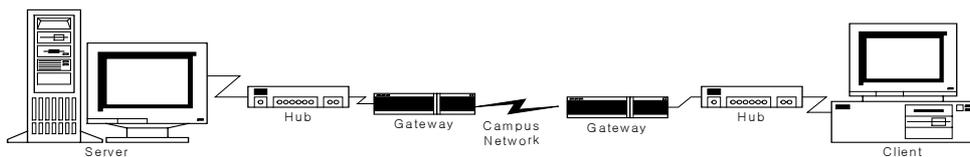


그림 4. 기초실험 환경

실험 환경은 RTP 패킷을 전송하는 서버와 패킷을 수신하는 클라이언트를 각각의 허브에 접속하여 인터넷을 경유하도록 하였다. 서버와 클라이언트 시스템

은 PC에서 동작한다. 운영체제는 Windows 2000이고, 개발도구는 Visual C++ 상에서 Win32 SDK를 이용하여 C 언어로 작성되었다.

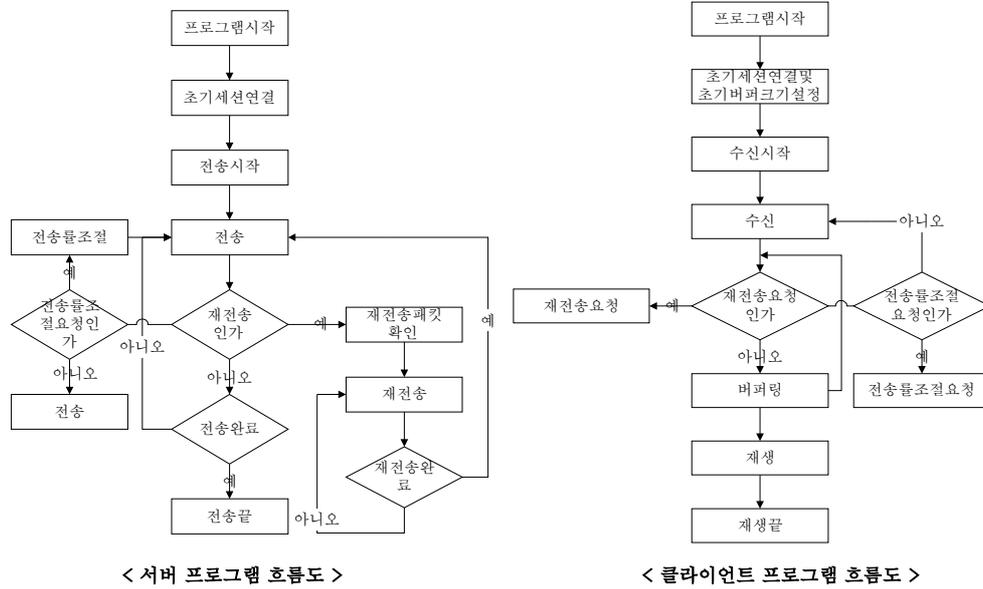


그림 5. 서버/클라이언트 프로그램 흐름도

서버 프로그램은 초기세션 연결 후 전송을 시작하며 클라이언트에서의 재전송 요청과 전송률 조절 요청을 받아들인다. 전송은 메인스레드(main thread)에서 담당하며 재전송 요청과 전송률 조절 요청은 서브스레드(sub thread)에서 받아들여 처리한다. 클라이언트 프로그램은 초기 세션 연결시 동적인 버퍼 크기를 결정하며 패킷수신 이후와 버퍼링 도중에 손실된 패킷의 재전송 요청과 전송률 조절을 요청한다.

서버와 클라이언트는 서로 다른 서브넷에 속하는 IP 주소를 가지고 있으며 각 서브넷의 게이트웨이를 경유하여 패킷을 전송하였다. 서버측에서 전송된 RTP 패킷은 서버가 속해 있는 네트워크의 게이트웨이를 경유하여 클라이언트가 속해 있는 네트워크의 게이트웨이에 도착하여 클라이언트로 전달된다. RTP

패킷은 총 2개의 게이트웨이를 경유하여 전송되었다. 실험 환경에서 RTP 데이터의 시간대별 손실률 실험을 실시하였다.

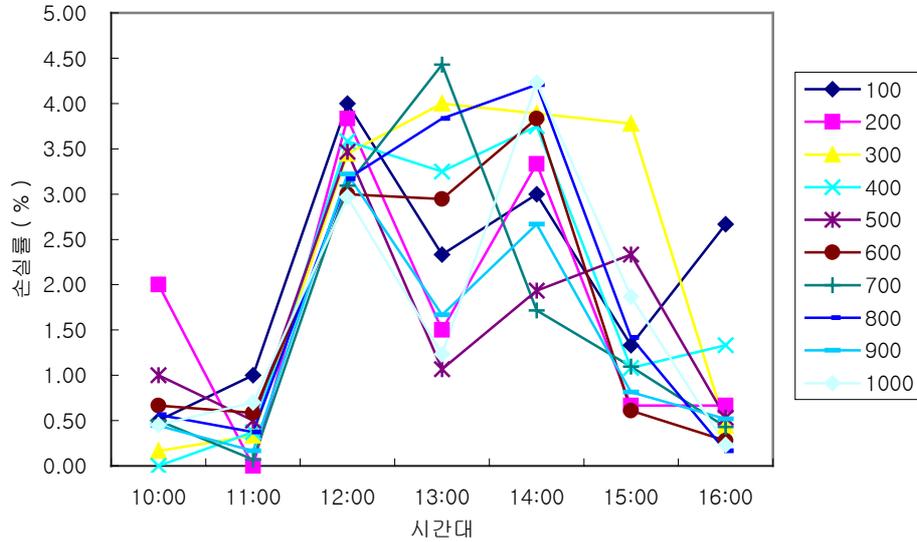


그림 6. 시간대별 데이터 손실률

서버에서 128 bytes의 패킷 크기를 가지는 100 개의 RTP 패킷을 256 Kbps 속도로 전송한 후 클라이언트에 도착한 패킷 개수를 측정하였다. 같은 방법으로 각각 200개부터 1000개의 RTP 패킷을 전송한 후 클라이언트에 도착한 패킷 개수를 측정하였다. 그림 6은 캠퍼스 내의 두 개의 게이트웨이(2 hop)를 경유하는 그림 4의 기초 실험 환경에서 256 Kbps의 속도로 3 일에 걸쳐 오전 10시부터 오후 4시까지의 RTP 패킷의 손실률 평균을 나타내었다.

실험결과를 보면 오전 11시부터 패킷 손실이 급격히 증가하여 오후 4시 이후에 점차 손실률이 줄어드는 것을 알 수 있다. 이것은 인터넷 상의 패킷들이 집중되는 시간대에 RTP 데이터의 손실도 함께 증가함을 보여주는 것이다. 최대 손실은 오후 1시에 700개의 패킷을 연속해서 전송하는 경우, 4.48%로 나타내

고, 오전 11시에 200개의 패킷을 연속해서 전송하는 경우에 0.1%로 나타났다. 실험한 인터넷에서 평균 손실률은 4.2% 미만이다. 이 값은 손실된 패킷을 재전송하는 경우 발생하는 트래픽의 증가량으로 볼 수 있다. 전체 RTP 패킷의 약 5%에 대한 재전송으로 완전한 데이터 전송이 가능하게 되므로 서버에서 제공되는 오디오 품질이 그대로 클라이언트에서 재생될 수 있다.

3.2 왕복시간

다음으로 실험된 내용은 재전송에 소요되는 시간이 오디오 데이터의 전송 등 시성을 만족시킬 수 있는지에 대한 여부에 관한 것이다. 오디오 데이터의 특성상 재전송에 기반 하여 완전한 데이터를 수신한 경우라도 재생되어야 할 시점 이후에 클라이언트에 도착하였다면 의미가 없다. 따라서 RTCP 피드백을 통한 손실된 패킷의 재전송에 소요되는 시간을 조사해 볼 필요가 있다. 실험 방법은 클라이언트의 RTP 패킷 재전송 요청 시점에서부터 요청된 패킷이 서버에서 전송되어 클라이언트에 도착하는 시점까지의 소요 시간, 즉 왕복시간(RTT : Round Trip Time)을 조사하였다.(그림 7)

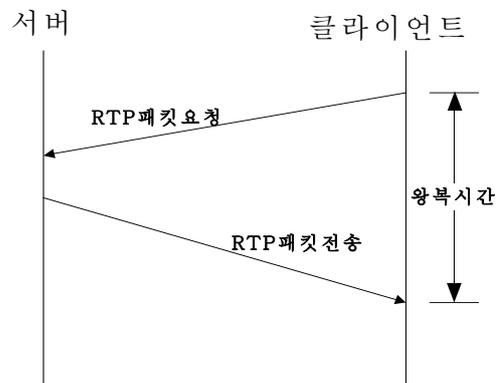


그림 7. 측정된 왕복시간

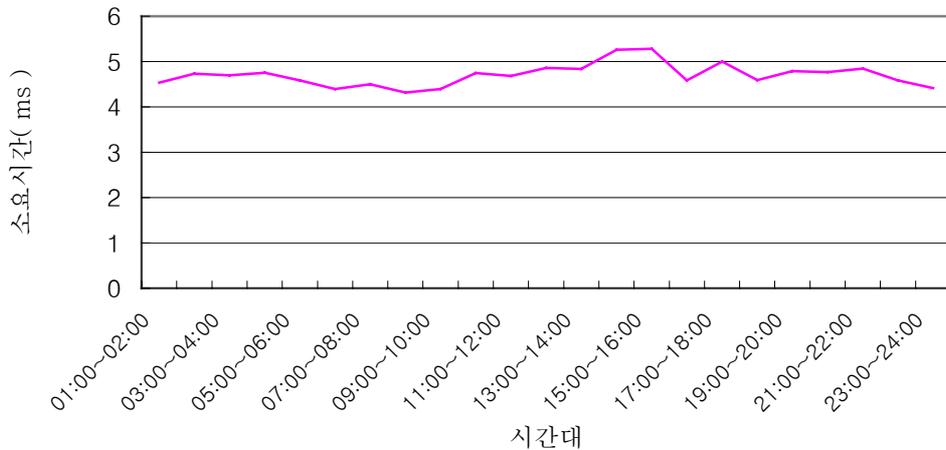


그림 8. 왕복 시간

그림 8 은 그림 4의 실험 환경에서 24 시간에 걸쳐 측정된 결과이다. 5분 간격으로 128 bytes 크기의 패킷 한 개를 클라이언트에서 서버로 전송하는 시점부터 서버에서 이를 다시 클라이언트로 전송하여 전송이 완료되는 시점까지의 시간을 측정하여 평균으로 나타내었다. 측정된 왕복시간은 오후 1시부터 오후 2시까지 최대 5.3 ms, 오전 9시부터 오전 10시 사이에 최소 4.4 ms로 나타났다. 또한 왕복 시간은 인터넷 사용량이 증가하는 시간대인 오전 11시에서 약간 증가하는 것을 볼 수 있으며, 일정 시간동안 4 ~ 6 ms를 소요하는 것을 알 수 있다. 이 실험에 의해 얻어진 결과에 의하면 클라이언트에서 유지해야 하는 버퍼크기가 전송률이 128Kbps인 경우는 98 bytes, 256Kbps인 경우는 49 bytes로 유지되면 된다. 이를 통해 매우 작은 버퍼만으로 선택적 재전송에 의해 패킷 손실에 대처할 수 있음을 알 수 있다.

현재 측정된 왕복시간은, 서버 프로그램이 RTP 데이터 패킷을 전송하지 않고 있고, 재전송 요청에 따른 처리를 고려하지 않은 상태에서 측정된 것으로 패킷의 재전송에 따른 서버 프로그램에서의 처리 시간(processing time)이 포함되

어 있지 않은 경우이다. 실제적인 RTP 데이터 패킷을 전송하는 동안 재전송에 대한 소요시간은 5장 실험 및 결과에 제시되어 있다.

3.3 전송률에 따른 패킷 간격

한번에 전송되는 패킷의 크기에 따라 서버에서 전송하는 데이터에 대한 패킷과 패킷간의 간격이 달라질 수 있다. 오디오 데이터의 전송률을 일정하게 유지하기 위해서는 패킷 크기가 커질수록 패킷 간격은 넓어지며 패킷 크기가 작아질수록 패킷 간격은 좁아져야 한다. 표 1은 2분간 재생되는 다양한 재생률을 가지는 오디오 데이터를 전송하기 위해 패킷 크기를 128 bytes로 고정할 경우 각 재생률에 따른 패킷 간격을 나타낸다.

표 1. 전송률에 따른 패킷 간격

| 전송률 (Kbps) | 샘플링률 (Hz) | 채널 | 패킷 크기 (Bytes) | 패킷 간격 (ms) |
|-----------------|----------------|--------|--------------------|-----------------|
| 320 | 44100 | Stereo | 128 | 3.125 |
| 256 | 44100 | Stereo | 128 | 3.906 |
| 128 | 44100 | Stereo | 128 | 7.812 |
| 96 | 44100 | Stereo | 128 | 10.416 |
| 56 | 22050 | Stereo | 128 | 17.857 |
| 32 | 11025 | Stereo | 128 | 31.25 |
| 20 | 11025 | Stereo | 128 | 50 |
| 64 | 8000 | Mono | 128 | 15.625 |

그림 7의 실험 결과와 표 1의 결과로 볼 때 128 Kbps의 stereo 수준의 오디오

데이터는 단독으로 손실된 패킷에 대한 재전송인 경우에는 수신측에서의 버퍼링 없이도 시간 제약성을 만족시키는 것을 볼 수 있다[4]. 그러나 그보다 나은 품질의 오디오 데이터는 단독 손실 패킷에 대해서도 재생 QoS에 영향을 받으며 더 나쁜 환경에서는 품질 열화가 충분히 예견된다. 이를 극복하기 위하여 본 논문에서 제시하는 복합적 QoS 보장 기법이 적용된다.

제 4 장 복합적 QoS 보장 기법

복합적 QoS 보장 기법은 세 가지 기법으로 구성된다. 첫째, 손실된 패킷에 대한 선택적 재전송, 둘째, 클라이언트 버퍼의 동적 할당, 셋째, 서버에서의 전송률 동적 제어 등이다. 컴퓨터 통신망에서 에러가 난 패킷을 재전송하는 대표적인 기법으로 정지대기(stop-and-wait), 후진-N(Go-Back-N), 그리고 선택적 재전송 (Selective Repeat) 등이 있다. 정지대기 기법은 각각의 전송된 패킷에 대한 응답을 받음으로 네트워크 대역폭 낭비가 극심하며[15], 후진-N 기법은 손실 발생시 손실되지 않은 패킷도 전송함으로써 대역폭의 낭비 및 실시간성을 만족하기에 부적절한 점이 있다[15]. 선택적 재전송 방법은 인터넷에서 신뢰성 있는 전송을 보장하는 TCP에서 사용하는 재전송 방법이다[15]. 3.2절의 기초 실험 결과에서 재전송에 소요되는 왕복시간은 극히 짧은 시간(약 5ms)임을 확인할 수 있으며, 손실 패킷 재전송에 요구되는 비용면에서 선택적 재전송 기법이 가장 적은 것으로 나타나있다[15,16]. 따라서 본 논문에서는 TCP에서의 흐름제어 및 재전송 방법에 사용되는 선택적 재전송 기법을 RTP/UDP 기반 오디오 데이터 전송의 재전송 기법으로 채택한다.

클라이언트 버퍼 동적 할당 기법은 송신측과 수신측의 초기 세션(session) 연결시 왕복시간을 측정하여 왕복시간동안 재생할 수 있는 크기만큼의 버퍼 공간을 할당하는 기법이다. 이를 통해 송수신측 간의 네트워크 상태를 고려한 수신측 버퍼 크기 할당을 통해 버퍼공간의 낭비를 최소화할 수 있을 뿐만 아니라 손실된 패킷의 재전송에 소요되는 시간을 보장 받을 수 있다.

서버에서의 전송률 동적 제어 기법은 송신측에서 일정한 패킷 간격으로 전송한 RTP/UDP 패킷이 수신측에 일정한 간격으로 도착하지 않는 경우에 발생할 수 있는 수신측 버퍼 고갈(starvation)이나 오버플로우(overflow)현상을 막기위

한 것이다. 수신측 버퍼의 상태와 동적인 왕복시간을 지속적으로 모니터링 하여 측정하고 그 상황에 따라 송신측에 현재 전송되고 있는 패킷들의 간격 조절 요청을 통해 전송률을 동적으로 제어한다.

4.1 선택적 재전송

클라이언트는 서버로부터 수신된 패킷에 대해 RTP 헤더의 순서번호를 확인한다. 만약 연속해서 수신된 패킷의 순서번호가 연속적이지 않은 경우, 연속적이지 않은 패킷 크기만큼 버퍼 공간을 채우지 않고 버퍼링을 진행한다. UDP를 기반으로 하는 프로토콜의 경우 패킷도착의 순서가 보장되지 않음으로 재전송 요청 이전에 패킷이 도착할 수도 있기 때문이다.

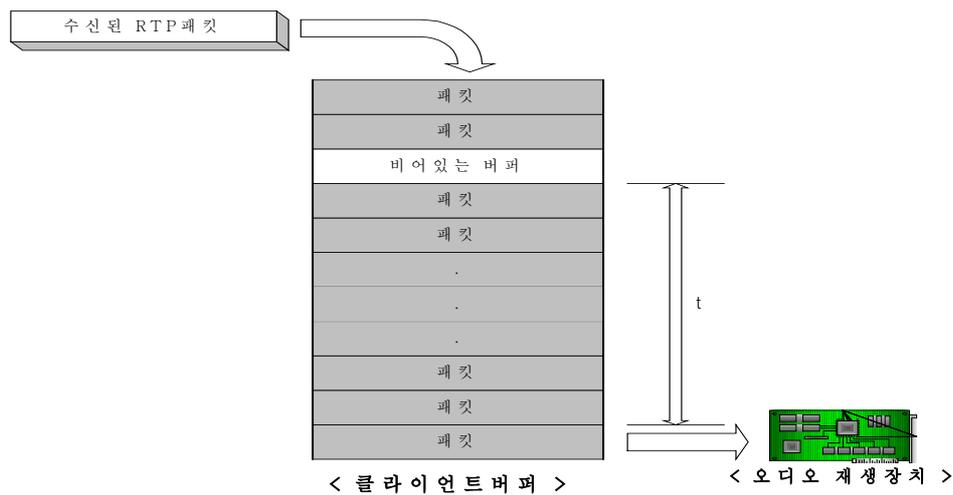


그림 9. 재전송 요청시점

재전송 요청 시점은 비어있는 버퍼 공간 직전까지 재생하는 데 소요되는 시간(t)이 왕복시간에 해당하는 시간이 되었을 시점($t \approx RTT$)이다. 재전송 요청

시점 이전에 도착하는 패킷은 비어있는 공간에 채워진다. 재전송 요청은 RTCP의 RR(Receiver Report) 메시지를 통해 이루어지고 재전송 성공시 버퍼 내의 비어있는 공간에 채워진다. 그림 10은 선택적 재전송에 의해 나타날 수 있는 클라이언트의 버퍼 상태를 모식화한 것이다.

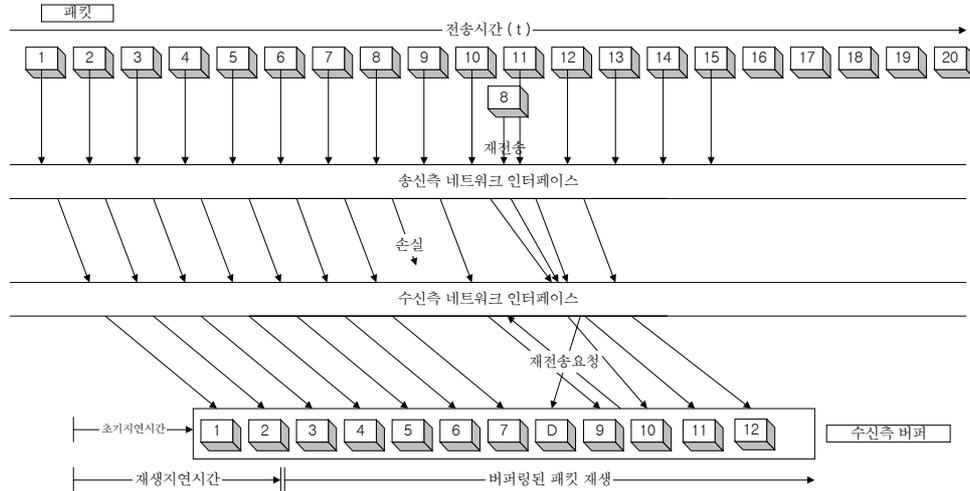


그림 10. 재전송 발생시 수신측 버퍼상태 변화

클라이언트에서는 초기 버퍼링 시간동안 지연이 발생하며 이 지연시간은 서버와 클라이언트 사이의 왕복시간동안 재생되는 데이터의 양을 기초로 한다. 초지연시간 이후 버퍼링된 패킷들을 재생하기 시작한다. 그림 9에서 서버에서 전송하는 RTP 패킷 중 8번 패킷이 손실이 발생한 경우 클라이언트에서는 8번 패킷 크기에 해당하는 버퍼공간 만큼을 채우지 않고 다음 패킷의 버퍼링을 진행한다. 9번 패킷 이후에 패킷의 손실이 확인되었고 8번 패킷까지의 재생시간이 왕복시간에 해당하는 시간이 되었을 경우 클라이언트에서는 재전송을 요청한다. 서버에서는 8번 패킷의 재전송 요청이 발생하였을 때 10번과 11번 패킷의 정상적인 패킷간격 사이에 손실된 8번 패킷을 전송한다. 클라이언트에서는 비어

있는 버퍼 공간에 8번 패킷을 할당하게 되므로 모든 RTP 패킷이 마감시간 이전에 클라이언트에 수신하게 되어 완전한 재생 품질을 유지한다.

4.2 클라이언트 버퍼 동적 할당

4.2.1 에러가 없는 네트워크 환경

송신측에서 전송된 패킷이 수신측에 손실 없이 도착할 수 있는 이상적인 네트워크 환경일 경우에는 수신측 버퍼 소모율, 즉 재생률을 r 이라 두고, 송신측 전송률을 s 라고 할 때 송신측 전송률과 수신측 소모율은 식 (1)과 같이 되어야 한다.

$$s = r \quad (1)$$

그러나 실제 인터넷상에서 전송되는 RTP/UDP 기반의 데이터는 송신측에서 일정 간격으로 패킷을 전송하여도 각 패킷의 전송지연 시간이 각각 다를 수 있기 때문에 수신측에 도착하는 패킷의 간격은 가변적이다. 이로 인해 전송동시성이 유지되지 못하는 현상이 발생할 수 있다. 수신측에서는 가변적인 패킷 도착 시간을 극복하기 위해 버퍼를 가져야 하며 그 버퍼 크기는 송수신측 간의 네트워크 상태를 고려하여 동적으로 조절되어야 한다.

4.2.2 클라이언트 초기 버퍼 크기

송신측과 수신측이 처음 세션을 연결할 때 그 당시의 왕복시간을 고려하여 클라이언트 초기 버퍼 크기를 결정한다. 클라이언트 초기 버퍼 크기는 송신측과 수신측간 왕복시간동안 재생되는 오디오 데이터의 양과 동일한 값을 가지게 된다. 단독 패킷 에러가 발생할 상황에 대비하여 반드시 만족되어야 하는 최소 초기 버퍼 크기 B 는 식 (2)와 같다.

$$B = \text{재생률} \times \text{왕복시간} \text{ (bit)} \quad (2)$$

B bits의 데이터를 수신하는 도중 패킷이 손실될 수 있으며 이에 대한 재전송에 소요되는 시간을 감안할 때, 수신측의 버퍼는 최소 버퍼 크기보다 커야 한다. 한 개의 패킷이 전송되는 중에 에러가 날 확률을 P 라 할 때 B 를 채우기 위해 서버가 전송하는 전송량 T 는 식 (3)과 같다[15].

$$T = \frac{B}{1-P} \text{ (bit)} \quad (3)$$

서버에서 한 패킷이 에러가 난 경우, 전송량의 증가가 초래됨으로써 재전송 후에는 전송률이 상향 조절되어야 클라이언트의 버퍼를 적어도 초기버퍼크기 이상으로 유지할 수 있다. 이에 따라 클라이언트의 버퍼량이 일시적으로 증가하는 현상이 발생할 수 있으나, 곧 이은 전송률 하향 조절로 버퍼량의 증가현상을 없앨 수 있다.

4.3 전송률 동적 제어

4.3.1 전송률 조정 요청 시점

전송률 동적 제어 기법은 송신측에서 전송하는 패킷 간격을 수신측의 요청에 의해 조절하는 기법이다. 수신측에서 송신측에 대한 전송 조정 요청 시점은 수신측의 버퍼 모니터링과 동적인 왕복시간 측정을 통해 가능하다[17,18]. 수신측에 도착한 패킷까지의 버퍼내 데이터 재생 시간이 왕복시간보다 짧은 경우에 전송률 증가 요청을 한다. 반면 수신측 버퍼 내 데이터 재생시간이 왕복시간보다 긴 경우에는 전송률 감소 요청을 한다. 전송률의 증가 또는 감소는 송신측에서 전송하는 패킷 사이의 간격을 수신측에서 요청한 ms 단위의 패킷 간격 조정 요청을 통해 이루어진다. 송신측에서 현재 전송중인 패킷의 간격을 1ms 조절할 경우 단위시간당 수신측에 도착하는 패킷의 수는 달라지며, 수신측 버퍼량은 점진적으로 증가 또는 감소한다. 앞서 제시한 표 1 경우에 대한 패킷 수 변화와 클라이언트 버퍼량 변화 예측 표를 표 2에서 보인다.

표 2. 1 ms 간격조절에 의한 패킷수와 버퍼량 변화

| 전송률 (Kbps) | 패킷 크기 (Bytes) | 패킷 간격 1ms 좁힘/넓힘 | |
|-----------------|--------------------|-----------------|---------------|
| | | 1초간 증가/감소 패킷수 | 1초간 증가/감소 버퍼량 |
| 320 | 128 | 103.2 | 13213 bytes |
| 256 | 128 | 65.6 | 8403 bytes |
| 128 | 128 | 16.4 | 2101 bytes |
| 96 | 128 | 9.2 | 1182 bytes |
| 56 | 128 | 3.1 | 403 bytes |
| 32 | 128 | 1.03 | 132 bytes |
| 20 | 128 | 0.4 | 52 bytes |

송신측에서 전송하는 패킷 간격을 1ms 조절할 경우 320Kbps에서는 1초 동안 13213bytes의 버퍼량 변화를 보이며 전송률이 가장 낮은 20Kbps에서는 52bytes의 버퍼량 변화를 보였다. 전송률이 높을수록 단위시간당 변화하는 버퍼량이 커지며 전송률이 낮을수록 단위시간당 변화하는 버퍼량이 작아진다.

4.3.2 송신측 패킷간격 변화

수신측 동적인 전송률 조절 요청 이외에 재전송 요청이 발생한 경우 송신측은 기존의 전송하는 패킷과 재전송이 요청된 패킷을 전송하게 된다. 재전송이 요청된 패킷은 기존의 패킷사이 간격 이내의 시간에 전송이 이루어지며 순간적으로 송신측에서의 재전송 요청시 패킷 간격은 기존 패킷 간격에 비해 절반 이하의 간격으로 좁아진다(그림 11).

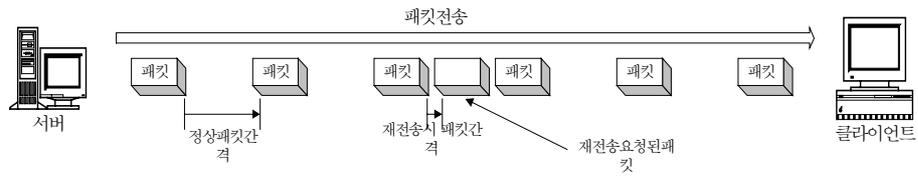


그림 11. 재전송시 패킷간격

이는 수신측에서는 비어있는 버퍼를 채우는 것일 뿐, 단위시간당 전송에서 변화가 아니므로 수신측 버퍼는 일정 크기를 유지할 수 있다. 수신측은 재전송이 완료되었을 경우 버퍼 내의 비어있는 공간을 재전송된 패킷으로 대체한다. 이것은 재전송 패킷의 재생 마감시간 이전에 모두 완료되어야 한다.

제 5 장 실험 및 결과

동적인 버퍼를 동적으로 할당할 수 있는 클라이언트와 선택적 재전송과 동적으로 전송률 제어가 가능한 서버 프로그램을 설계, 구현하여 RTP 데이터 트래픽을 모니터링 하였다. 서버와 클라이언트 프로그램은 Microsoft Windows 2000 운영체제 상의 Win32 SDK를 사용하여 C언어로 제작되었으며, 그림 4의 기초실험 환경에서 서버에서 클라이언트로 RTP 데이터 전송 중에 재전송 소요시간측정과 클라이언트 버퍼의 고갈 현상을 확인하기 위해 재전송 완료시 버퍼내의 남은 데이터량 측정과 클라이언트 버퍼량 변화를 확인하기 위해 2분간 서버에서 클라이언트로 RTP 데이터 전송시 클라이언트의 버퍼량을 측정하였다. 그리고 재전송 요청시 패킷 간격 변화를 확인하기 위해 같은 시점에 서버에서 전송되는 패킷 간격과 클라이언트에 수신되는 패킷 간격을 측정하였다.

5.1 재전송 소요시간

재전송이 요청된 패킷이 버퍼의 고갈 이전에 도착하는지 여부와 전송률 제어를 통해 버퍼가 적정크기로 유지되는지 여부를 확인하기 위해 오전 10시부터 오후 4시 사이에 30분 간격으로 서버에서 128 bytes 크기의 RTP 패킷 100개를 128Kbps 속도로 클라이언트로 전송하는 실험을 하였다. 전송도중 패킷 손실이 발생할 수 있으며 재전송 소요시간은 클라이언트에서 재전송 요청 패킷을 전송하는 시점부터 서버로부터 재전송된 패킷이 클라이언트에 전송 완료되는 시점까지이다. 클라이언트에서 손실된 패킷 확인 후 재전송 요청에 소요되는 시간은 포함되지 않은 시간이며 실제 인터넷 환경에서는 클라이언트에서 재전송 요청에 소요되는 처리시간을 합한 만큼의 재전송 소요시간이 필요하다. 그림 13에 손실된 패킷에 대한 재전송 소요시간을 나타내었다.

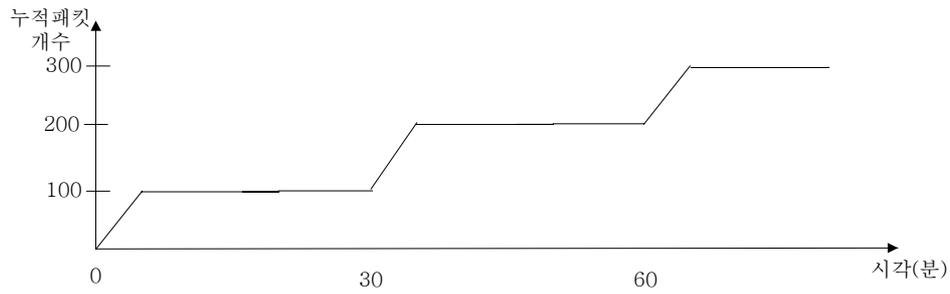


그림 12. 패킷 전송 누적 패턴

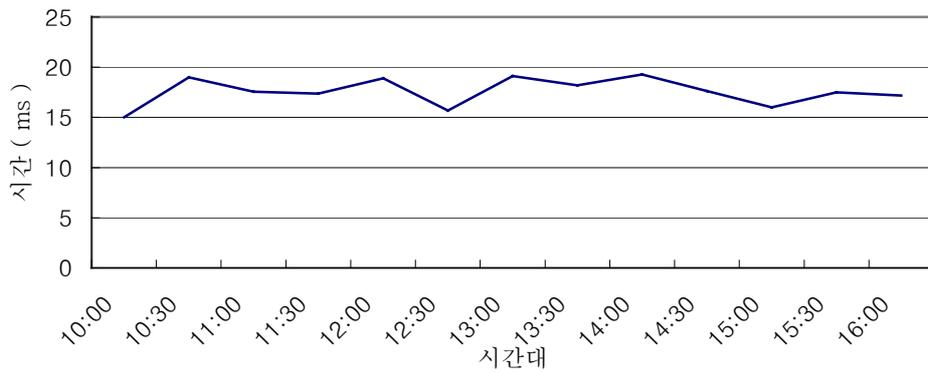


그림 13. 재전송 소요시간

재전송에 소요되는 시간은 클라이언트가 서버로 재전송 요청 패킷이 전달되는 시간과 요청된 패킷을 서버에서 클라이언트로 전송하는 데 소요되는 시간의 합이다. 실험 결과 재전송 소요시간은 최소 15.1 ms에서 최대 19.8 ms로 나타났다. 그림 8에서 측정된 왕복시간 4ms ~ 6ms 보다는 약 11.1ms ~ 13.8ms 시간이 더 소요되는 시간이다. 이는 서버에서 클라이언트로부터 재전송 요청된 패킷을 확인 후 재전송하기 위해 소요되는 처리시간이 더해 지기 때문이다. 서버에서는 재전송 요청된 패킷을 재전송하는데 약 12ms의 일정한 시간을 소요하지만 클라이언트의 재전송 요청에 소요되는 처리시간과 동적인 왕복시간을 기준으로 한 재전송 요청시점으로 인해 클라이언트의 버퍼크기는 재전송에 소요되는 시간동안 재생되는 데이터양보다 커질 수 있다. 다음 실험에서 실제 인

터넷 환경에서 측정된 전체 재전송 시간에 기반 한 재전송 완료시 클라이언트 버퍼 내의 남은 재생시간을 나타내었다.

5.2 재전송 완료시 버퍼 내의 남은 재생시간

그림 14는 재전송 완료시 클라이언트 버퍼 내의 재전송이 요청된 패킷 직전까지의 데이터 양을 데이터 재생률을 근거로 시간 값으로 표현한 것이다.

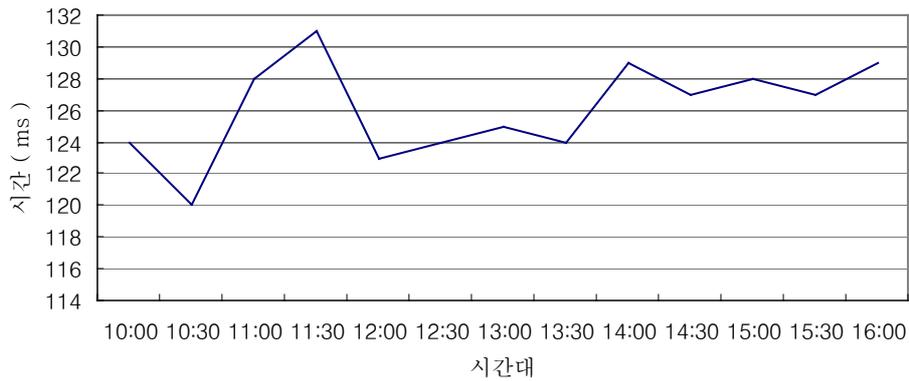


그림 14. 재전송 완료시 버퍼내의 남은 재생시간

송신측과 수신측의 왕복 시간에 기준 한 동적인 버퍼할당은 데이터 전송시 네트워크 상황을 고려한 버퍼크기이며 전송률 동적 제어를 통해 그림 14의 결과와 같이 버퍼 내의 데이터의 양을 일정하게 유지할 수 있다. 재전송 완료시 재전송 요청된 패킷 직전까지의 버퍼 내의 데이터 재생 여유 시간이 110ms ~ 130ms 임으로서 재전송 요청된 패킷이 재생 시점 이전에 도착함을 알 수 있다. 128 Kbps 이상의 전송률에 대하여 수신측 버퍼의 재생되지 않은 데이터 양을 확인하기 위해 수신측 버퍼 내의 남은 재생시간이 가장 적은 시간대인 오후 12시 부터 오후 2시 사이에 각각 128Kbps, 256Kbps 그리고 320Kbps 전송률로 2 분간 재생할 수 있는 데이터를 전송하였을 때 수신측의 재생되지 않은 버퍼량의 변화를 실험하였다.

5.3 2분 동안 재생시 수신측 재생되지 않은 버퍼량 변화

재생시간이 120초인 오디오 데이터를 스트리밍 방식으로 전송, 재생할 때 클라이언트 버퍼크기 변화를 측정하였다. 이를 통해 오디오가 재생되는 동안 항상 RTT 시간보다 상회하는 버퍼량이 클라이언트에 유지되는 것을 확인할 수 있고 재생품질의 열화현상이 전혀 발생하지 않음을 알 수 있는 것이다. 실험은 재생률이 각각 128Kbps, 256Kbps 그리고 320Kbps인 고품질 오디오 데이터에 대해 각각 실행되었다. 그림 15는 시간 기준으로 클라이언트 버퍼크기 변화를 보인다.

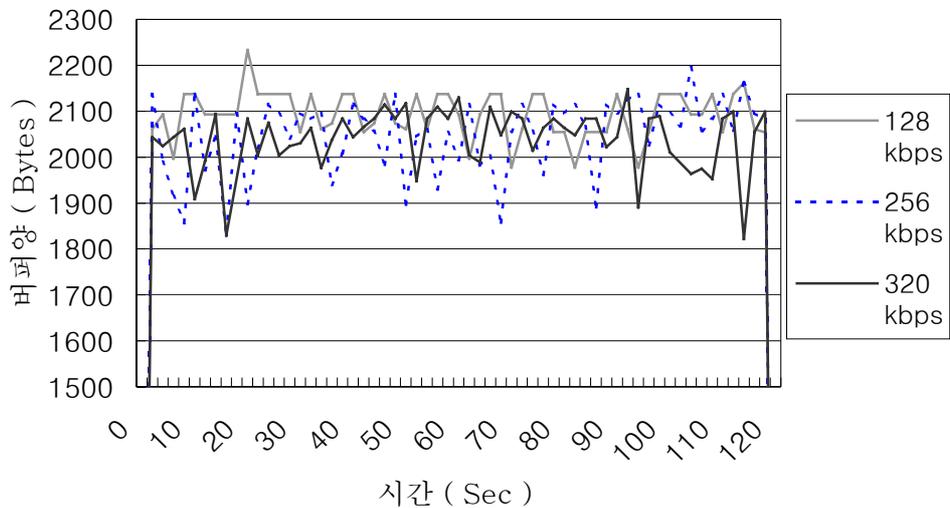


그림 15. 2분 간 재생시 수신측 재생되지 않은 버퍼량 변화

수신측 초기버퍼 버퍼링 이후 재생이 시작되며 버퍼링된 오디오 스트림의 재생 이후 수신측 재생되지 않은 버퍼량이 최소 1821 bytes ~ 최대 2195 bytes의 양을 유지하는 것을 확인 할 수 있다. 2분 간의 재생이 되는 동안 재전송 요청/완료 과정이 반복되게 되며 재생률에 따라 약간의 버퍼량 차이를 보이며 재생률이 높아짐에 따라 재생되지 않은 버퍼량의 차이가 크게 나타난다. 이는 전

송률이 높아짐에 따라 패킷 사이 간격이 좁아지며 가변적인 네트워크 상황에 의해 패킷 도착시점이 일정하지 않기 때문이다. 다음 실험에서는 손실된 패킷 발생시 송신측과 수신측의 패킷간격을 모니터링함으로써 실제 전송환경에서 패킷 간격의 변화를 확인한다.

5.4 재전송시 송신측 패킷 간격 변화

서버에서 전송되는 패킷과 패킷간의 시간 간격을 측정하였다. 패킷간의 시간 간격은 전송률을 나타내는 지표로서 동적으로 변하는 전송률을 파악하기 위한 것이다. 전송률이 각각 128Kbps, 256Kbps 그리고 320Kbps 인 경우에 대해 서버에서 전송되는 전체 패킷 가운데 100개의 패킷에 대한 것으로 표시하였다. 그림 16은 서버에서 인터넷으로 진입하기 직전 단에서 측정된 패킷 시간 간격이다.

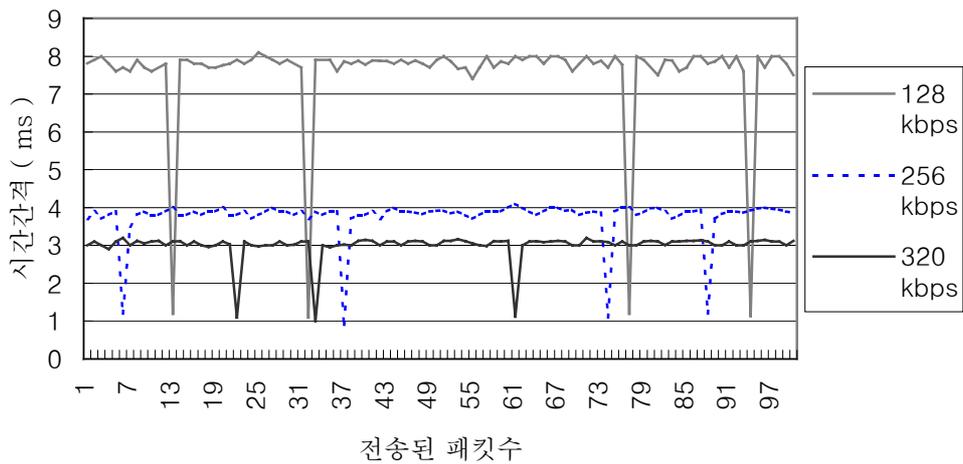


그림 16. 송신측에서 측정된 패킷 간격 변화

그림 16의 결과에서 정상적으로 에러 없이 패킷을 전송하는 경우 전송률이 128Kbps는 7.6ms ~ 8.1ms, 256Kbps는 3.5ms ~ 4ms, 320Kbps는 3ms ~

3.2ms 정도의 간격을 유지한다. 그러나 재전송 요청을 클라이언트로부터 받아서 서버가 재전송하는 경우에 서버측 패킷 간격이 최소 0.9ms ~ 최대 1.21ms의 간격을 보임을 알 수 있다. 이는 재전송 패킷을 전송할 때 서버는 정상적인 패킷과 패킷 사이에 재전송 패킷을 전송하기 때문이다.

5.5 재전송시 수신측 패킷 간격 변화

클라이언트에서 수신되는 패킷과 패킷 간의 시간 간격을 측정하였다. 그림 17에 정상적인 패킷과 재전송 요청에 의해 수신되는 패킷 사이의 간격을 나타내었다. 그림 16에 제시된 결과와 같은 시간에 클라이언트에 도착하는 패킷에 대한 모니터링 결과이며 서버측 재전송 시점에서 클라이언트에 도착한 패킷 간격을 확인할 수 있다.

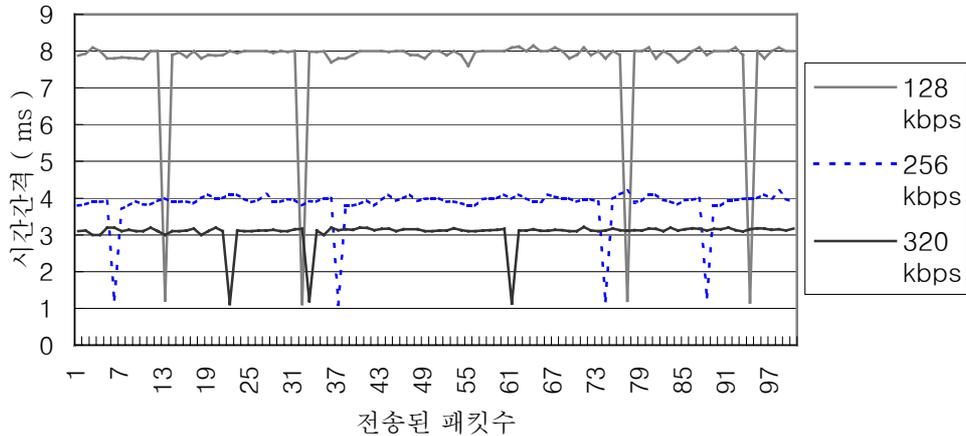


그림 17. 수신측에서 측정된 패킷 간격변화

그림 16과 그림 17의 비교분석 결과 송신측의 재전송시 패킷사이 간격은 좁아지며 수신측에 도착하는 패킷사이 간격도 최소 1.1ms ~ 최대 1.23ms로 좁아짐을 확인할 수 있다. 이는 일시적으로 정상적인 패킷과 재전송된 패킷이 연속적으로 도착하는 경우이다.

제 6 장 결론 및 향후 연구과제

인터넷에서의 오디오 스트리밍 서비스의 QoS를 결정하는 요인은, 전송중 패킷 에러율과 온전한 패킷이더라도 재생시각 이전에 도착해야 하는 것 등이다. 오디오 데이터는 동일하게 시간제약성을 가지는 비디오 데이터에 비해 전송률이 상대적으로 낮다. 따라서 현재의 인터넷에서도 완벽한 전송에 따른 완전한 QoS가 보장될 수 있다. 그러나 기존의 QoS보장 기법들은 완전한 데이터 전송을 보장하지 못하는 점과 수신측 상태를 즉각적으로 송신측에 알릴 수 없는 문제점을 가지고 있다. 손실된 패킷에 대한 선택적 재전송, 왕복시간을 기반으로 하는 동적인 수신측 버퍼 할당, 전송률 동적제어를 통한 버퍼유지를 통해 인터넷 상에서 클라이언트에서 서버에서 전송한 고품질 오디오 데이터를 손실이나 지연 없이 수신하여 서버가 제공하는 QoS 그대로 오디오를 재생하였다.

본 논문의 연구 결과로써 동적인 수신측 버퍼 할당과 유지를 통해 수신측 버퍼 크기를 최소화하여 과도한 버퍼 할당으로 인한 시스템자원 낭비를 막을 수 있으며 선택적 재전송 기법과 전송률 제어기법을 병행하여 적용함으로써 수신측에서는 송신측에서 전송한 데이터를 완벽히 수신하여 완전한 오디오 QoS를 보장할 수 있다. 이러한 연구결과의 응용으로는 인터넷을 사용한 고품질 음성 서비스, 인터넷 강의, 멀티미디어 서비스에 활용될 수 있다.

향후 연구 과제로는 고품질 오디오 스트리밍 서비스 기술을 이용하는 인터넷 콘서트, 현장감 넘치는 스포츠 중계, 그리고 고품질전화/음성회의 서비스를 연구할 계획이다.

본 논문에서는 고품질(128Kbps 이상 재생률을 가지는) 오디오 스트리밍 서비스를 위해 세 가지 기법을 복합적으로 적용하는 QoS보장 기법을 제안하였다.

참고문헌

- [1] Palmer. W. Agnew, and Anne S. Kellerman, *Distributed Miltimedia*, p. 30, Addison-Wesley, 1996.
- [2] James F. Kuros, Keith W. Ross, *Computer Networking : A Top-Down Approach Featuring the Internet*, pp. 167~268, Addison Wesley Press, 2001.
- [3] 강민규, 공상환, 김동규, “RTP/RTCP를 이용한 영상회의 시스템에서 오디오 패킷 손실 보상을 위한 동적 부가 전송 매커니즘 개발 및 성능 분석”, 한국정보처리학회 논문지, 제5권, 제10호, pp. 2641~2753, 1998.
- [4] 모수정, 안중석, “RTP/RTCP를 위한 확장성 있는 피드백 제어 기법”, 한국정보과학회 '98 가을 학술 발표 논문집(III), pp. 477~479, 1998.
- [5] 김태형, 스케줄링 기법을 적용한 VoIP환경에서의 QoS보장에 관한 연구, 성균관대학교 정보통신대학원, 석사학위논문, 2001.
- [6] 최정용, 네트워크상에서의 패킷 손실을 고려한 실시간 비디오-스트림 재전송 시스템의 구현, 성균관대학교 대학원, 석사학위논문, 2001.
- [7] 김명호, 이운주, 멀티미디어 개념 및 응용, 홍릉과학출판사, 1996.
- [8] 황연자, 차등 서비스 네트워크에서의 VoIP 성능 향상을 위한 연구, 고려대학교 산업정보대학원, 석사학위논문, 2001.
- [9] 정태욱, RTP/RTCP를 이용한 연속 미디어 데이터의 흐름제어 및 버퍼제어 기법, 광주대학교 대학원, 석사학위논문, 2000.
- [10] 문금지, VoIP Client에서의 H.323의 RAS Message 구현과 적용모델, 숭실대학교 대학원, 석사학위논문, 2000.
- [11] Colin Perkins, Orion Hodson, and Vicky Hardman, “A Survey of Packet Loss Recovery Techniques for Streaming Audio”, Readings in

- Multimedia Computing and Networking, pp. 607~615, Morgan Kaufmann, 2002.
- [12] Jean-Chrysostome Bolt, Sacha Fosse-Parisis, Don Towsley, “Adaptive FEC-Based Error Control for Internet Telephony”, Readings in Multimedia Computing and Networking, pp. 616~623, Morgan Kaufmann, 2002.
- [13] 김무중, VoIP 시스템에서 통화품질 향상을 위한 동적 제어 알고리즘의 성능분석, 대전대학교 대학원, 석사학위논문, 2001.
- [14] 한성우, 공간적 지역성을 이용한 RTCP 개선방안, 숭실대학교 대학원, 석사학위논문, 2000.
- [15] William Stallings, 데이터통신 및 컴퓨터통신 6/e, pp. 266~269, 사이텍미디어, 2001.
- [16] Bert J. Dempsey, Jorg Liebeherr, Alfred C. Weaver, “On retransmission-based error control for continuous media traffic in packet-switching networks”, *Computer Networks and ISDN Systems*, Vol 28, Issue 5, pp. 719~736, March 1996.
- [17] 김완규, 박규석, “VOD 시스템에서 클라이언트 버퍼를 위한 전송률 제어 알고리즘 설계 및 분석”, 멀티미디어학회 논문지, 제1권, 제1호, pp. 67~79, 1998.
- [18] 김윤호, 정두영, “가변 에러율 채널에 효과적인 Stop-and-Wait ARQ 방식”, 멀티미디어학회 논문지, 제5권, 제2호, pp. 198~205, 2002.

감사의 글

지난 대학원 생활 동안 많은 것을 가르쳐 주시고 본 논문이 완성되기 까지 끊임없는 격려와 조언으로 저를 이끌어 주신 손주영 교수님께 감사를 드립니다.

좀더 많은 것을 가르쳐 주시기 위해 따뜻한 격려와 엄한 질책을 아끼시지 않으시던 교수님께 다시 한번 감사 드립니다. 그리고 심사를 맡아 지도와 조언을 해주신 박휴찬 교수님과 김재훈 교수님께도 감사를 드립니다. 또한 대학원 생활동안 제게 가르침을 주신 류길수 교수님께도 감사를 드립니다.

그리고 지난 시간동안 연구실에서 같이 생활하며 동거동락 하던 연구실 동료 및 동생들에게 고마움을 전합니다. 실험을 도와주며 싫은 내색한번 안 하던 요나, 연구실의 홍일점 성미, 언제나 친구 같은 말벗이 되어준 진우, 연구실에 온 지 몇 달 안되었지만 착실한 진형, 가끔 연구실에 오지만 언제나 진지한 말벗이 되어준 정우, 그리고 이번에 입학한 신입생 현근이와 라경이, 그리고 다른 연구실이긴 하지만 여러가지로 도움을 주신 성대형과, 병수씨, 귀여운 후배 억종이, 이 모든 분들께 감사의 마음을 전합니다.

자식이 잘되기만을 바라시던 부모님과 할머니께도 감사 드립니다. 이제 좀더 어엿한 사회인으로서 부모님과 할머니께 효도할 것을 다짐합니다. 그리고 지난 10년간 나를 믿고 따라준 사랑스러운 예비신부 정은이에게 감사의 마음을 전합니다. 그리고 이 모든 일들을 가능하게 해주신 하느님께 영광을 바칩니다.