碩士學位論文

SIP 기반 실시간 멀티미디어 통신을 위한 성능 요소 분석

2005 年

漢城大學校 一般大學院

컴 퓨 터 工 學 科

컴퓨터 工學 專攻

李　　宰　　雨

碩 士 學 位 論 文
指導敎授 황 기 태

SIP 기반 실시간 멀티미디어 통신을 위한 성능 요소 분석

An analysis of performance parameters the SIP-Based real-time communication

위 論文을 컴퓨터工學 碩士學位論文으로 提出함

2004年 12月 日

漢城大學校 一般大學院

컴퓨터 工學科

컴퓨터 工學 專攻

李 宰 雨

碩 士 學 位 論 文
指導敎授 황 기 태

SIP 기반 실시간 멀티미디어 통신을 위한 성능 요소 분석

An analysis of performance parameters the SIP-Based real-time communication

2004年 12月 日

漢城大學校 一般大學院

컴퓨터 工學科

컴퓨터 工學 專攻

李 宰 雨

李宰雨의 工學 碩士學位論文을 인정함

2004年 12月 日

심사위원장 박 영 환 (인)

심사위원 정 인 환 (인)

심사위원 황 기 태 (인)

목 차

표 목 차

그 림 목 차

제 1 장 서론

1.1 연구배경

인터넷 환경이 성장함에 따라 SIP(Session Initiation Protocol)[1]를 기반으로 하는 여러 응용프로그램들이 실생활에 적용이 되고 있다. SIP는 인터넷상에서 통신하고자 하는 지능형 단말기들이 서로를 식별하여 그 위치를 찾고, 그들 상호간에 멀티미디어 통신 세션을 생성하거나 삭제, 종료하기 위한 절차를 명시한 응용 수준의 시그널링(signaling) 프로토콜이다[2]. 이러한 응용프로그램들 중 가장 대표적인 것이 멀티미디어 통신이라 할 수 있다. 이는 멀티미디어 서비스 세션을 생성하고 수정하고 종료하기 위한 시그널링 프로토콜이므로. 시그널링 프로토콜의 역할은 메시지 교환을 원하는 주체들 간에 메시지 세션을 제어하기 위한 정보를 교환하는데 있다[3].

일반 전화망에서 통화를 하기 위해 다이얼 번호를 누르고, 상대가 전화 수화기를 들기 전까지 대기음을 듣고, 통화가 끝난 뒤 회선 자원을 반납하는 과정들이 시그널링을 통한 제어 메시지 전달을 통해 이뤄진다. 이와 비슷하게 인터넷을 기반으로 하는 인터넷 텔레포니 서비스(Internet Telephony service), 원격 화상회의, 음성 메일 등에 사용될 수 있는 시그널링 프로토콜 중에 하나가 바로 IETF(Internet Engineering Task Force)의 MMUSIC(Multiparty Multimedia Session Control)에 의해 제안된 SIP이다[4]. 향후 모든 단말기들이 IP를 갖고 있어 마치 컴퓨터를 연상케 할 것이다. 이를 All-IP라 한다. 따라서, All-IP 기반의 차세대 유/무선 통합 통신망 상에서 다양한 형태의 지능형 단말기들을 이용한 다자간 멀티미디어 협업(collaboration)서비스를 구현하는데 필요한 핵심 프로토콜로 자리잡게 되었다[5].

한편, SIP는 RTP(Real Time Protocol)와 보통 짝을 이루어서 멀티미디어 응용에 주로 사용되고 있다[6]. RTP는 IETF RFC 1889에서 표준화한 프로토콜로, 실시간 데이터의 종단 전송 기능을 수행하는 프로토콜이다

[7, 8]. 따라서 호(call) 처리는 SIP를 이용하고 데이터 전송은 RTP를 이용한다. SIP기반 미디어 멀티미디어 시스템 분석을 위해서는 패킷을 실시간으로 전송할 수 있어야 하는데, 이런 역할을 RTCP가 해주고 있다.

1.2 연구 목적

본 논문에서는 첫째, SIP 기반 응용 프로그램 중 현재 활발히 연구되어지고 있는 SIP 기반 실시간 멀티미디어 플레이어의 시스템 구조를 분석하였다.

둘째, 임베디드 시스템에 포팅하기 앞서 분석된 SIP 기반 실시간 멀티미디어 플레이어의 성능 요인을 측정하였다.

셋째, 성능 요인을 측정한 데이터를 바탕으로 각 성능 요인들을 분석하였다.

본 논문의 구성은 다음과 같다.

● 2장 관련연구
  - VOVIDA사[9]에서 개발한 SIP 시스템에 대해 기술한다.
● 3장 SIP기반 실시간 멀티미디어 서비스
  - SIPSET 1.5와 MPEG4IP 1.0[10]으로 구성된 SIP기반 실시간 멀티미디어에 대해서 기술 한다.
● 4장 RTCP
  - 성능 요인을 측정하기 위한 RTCP에 관한 내용에 대해 기술 한다.
● 5장 성능 요인 측정 및 분석
  - SIP기반 실시간 멀티미디어의 성능 요인을 측정하기 위한 실험 환경과 패킷 분석을 하기 위한 프로세싱 과정, 실험 및 분석에 대해 기술 한다.
● 6장 결론 및 향후 연구과제

제 2 장 관련 연구

2.1 SIP

2.1.1 SIP 개요

SIP는 IETF의 MMUSIC 작업그룹에서 개발한 것으로 매우 간단한 텍스트 기반의 응용계층제어 프로토콜이다. 특히 VoIP(Voice over Inter Protocol)가 단순한 음성전달 차원을 넘어 완벽한 통합 메시징 기능을 수행하고 나아가 다자간 회의, 온라인 면회 등의 부가서비스를 구현하도록 하는데 있어 SIP 기반의 솔루션 개발이 필수적으로 요구된다.

SIP는 같은 텍스트기반의 SMTP와 HTTP 개발 이후에 설계됐다. 또 SIP는 클라이언트들이 호출을 시작하면 서버가 그 호출에 응답을 하는 클라이언트서버 구조에 기반을 두고 있다.

SIP는 이러한 기존의 텍스트 기반 인터넷 표준들을 준수함으로써 시스템 유지보수와 최적화 및 응용서비스의 구현이 상대적으로 용이한 특징을 갖고 있다.

이와 함께 SIP는 프록시와 리다이렉트 등을 통해 사용자의 이동성을 지원할 뿐 아니라 하위계층의 프로토콜로부터 독립적인 성격을 갖는다.

2.1.2 SIP 네트워크 구조

그림 1은 SIP의 네트워크 구조를 나타내고 있다. SIP는 클라이언트들이 호출을 시작하면 서버가 그 호출에 응답을 하는 클라이언트/서버 구조에 기반을 두고 있다. 각각의 역할은 다음과 같다.

■ UA(User Agent)

실제 사용자와 동작하는 부분을 말한다. 요청하는 UAC(client)와 응답

하는 UAS(Server)로 구분될 수 있지만 실제적으로 두 부분 모두가 하나의 프로그램에 포함되어야 한다.

■ Proxy server

요청 메시지를 실제 메시지가 전달되어야 할 곳으로 넘겨주는 역할을 한다. Proxy mode는 이후 연속되는 메시지들을 계속 전달받아 넘겨주지만 redirect mode에서는 자신을 거치지 않고 바로 요청 할 수 있는 SIP-URI를 클라이언트에게 전달한다.

■ Registrar

UA는 자신을 REGISTER 메소드를 이용해 registrar에게 등록하고 다른 UA로부터 요청 받을 시 그 등록된 정보를 그 UA에게 제공함으로서 자신에게 메시지가 전달될 수 있도록 한다. 또한 인증 과정을 통해 정당한 사용자인지를 구별하는 역할도 한다.

■ Location server

SIP UA가 아니라 사용자의 위치를 등록해놓은 DB라고 볼 수 있다. Location server와 다른 SIP 구성원간에는 SIP가 아닌 다른 방법으로 정보가 교환된다.

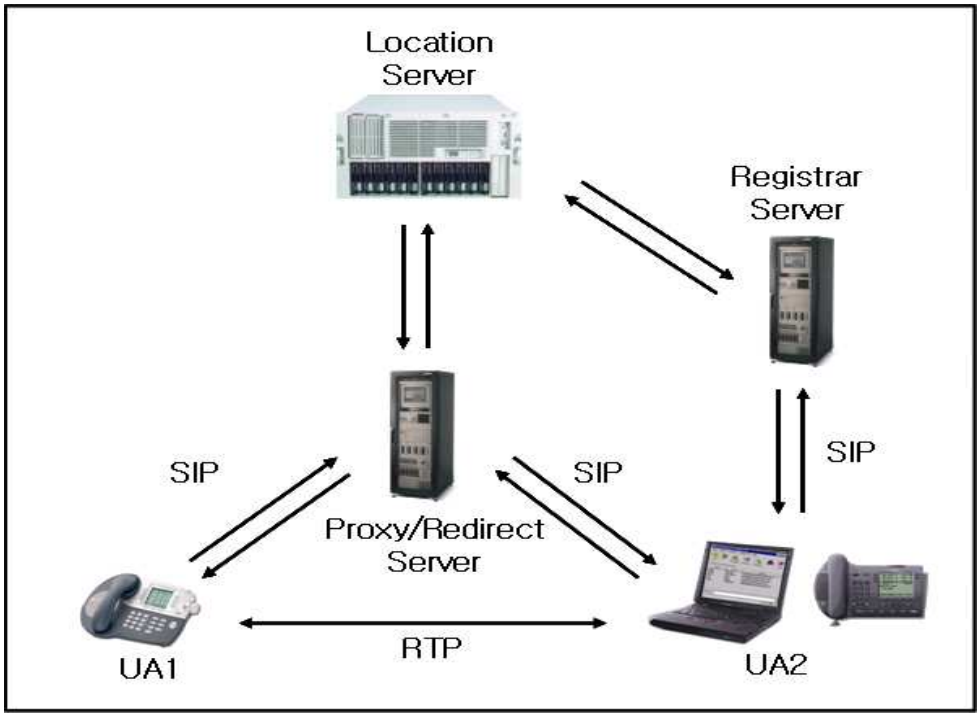


그림 1 SIP 시스템 구성도

2.2 SIP 특징

SIP의 특징으로 IETF에 충실히 부합되게 설계되었다. 그래서 인터넷에서 사용되는 다른 많은 프로토콜과 결합하여 다양한 서비스들을 만들 수 있는 유연성과 확장성을 제공한다. 일반적인 예로서 SAP(Session Advertise Protocol)로 세션에 대한 정보를 관심 있는 그룹에 제공하고 SIP를 통해 대화를 원하는 상대가 세션에 참가하도록 요청(INVITE)하며 SDP(Session Description Protocol)[11]를 통해 열고자 하는 미디어 형태에 대한 정보를 교환한다. 또한 SDP에 기술된 RTP등을 이용해서 실시간 멀티미디어 서비스를 제공할 수 있게 한다[12].

SIP는 응용계층 프로토콜로서 TCP와 UDP 모두 사용할 수 있으며 요청 / 응답 구조이다. 또한 HTTP와 SMTP의 구문과 의미 등 많은 부분을 그대로 사용하며 HTTP와 유사한 기능을 한다. 텍스트 기반이기 때문에 H.323[13]에 비해 구현이 용이한 장점도 있다.

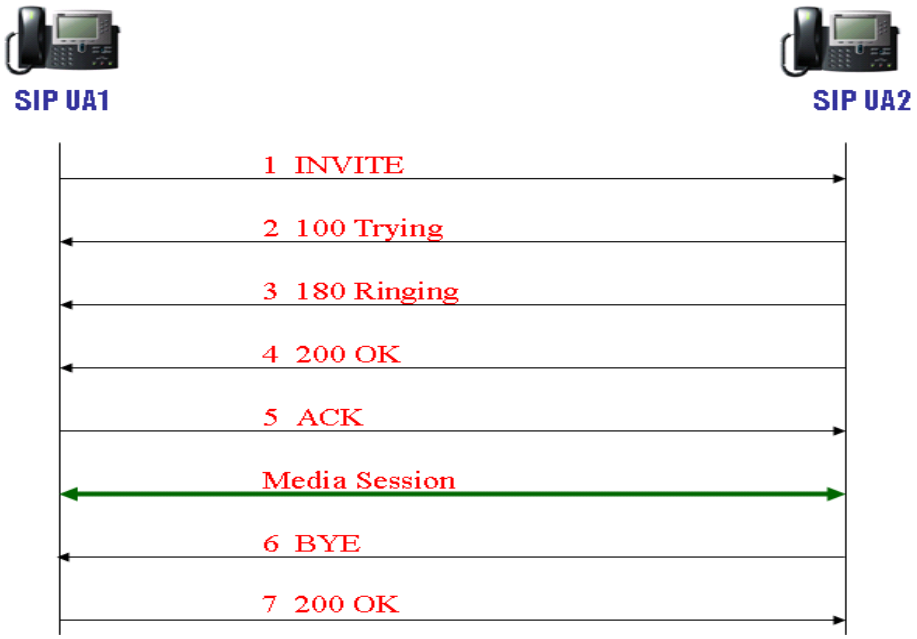


그림 2 SIP동작 원리

SIP는 각 사용자들을 구분하기 위해 이메일 주소와 비슷한 SIP URL을 사용함으로서 사용자는 IP주소에 종속되지 않고 서비스를 제공받을 수 있다.

그림 2의 예를 통해 간단히 SIP의 동작을 알아보면 SIP UA1은 INVITE 메소드를 이용해서 SIP UA2와 통화할 수 있도록 요청한다. 이 과정에서 SDP에 SIP UA1의 미디어 타입에 대한 정보가 같이 전달된다. INVITE 메시지가 SIP UA2의 전화기에 전달되면 전화벨을 울리고 그 상태를 다시 SIP UA1에게 알려준다. SIP UA2가 전화를 받으면 통화를 하려는 의지가 있다는 메시지를 SIP UA2의 미디어 타입에 대한 정보와 함께 SIP UA1에게 전해주고 SIP UA1은 ACK 메시지를 보냄으로써 통화를 시도하는 미디어 타입이 일치함을 알리며 둘 간에 실제 통화를 하게 된다. 통화가 끝난 뒤 SIP UA2가 전화 수화기를 내림으로서 BYE 메소드가 SIP UA1 에게 전달되고 세션을 끝내는 BYE 메소드 요청이 잘 처리되었음을 알리는 200 OK 메시지를 보냄으로써 비로소 세션을 끝마치게 된다.

제 3 장 SIP 기반 실시간 멀티미디어 플레이어

3.1 SIP 기반 실시간 멀티미디어 플레이어의 소프트웨어 구조

3.1.1 시스템 구조

본 논문은 VOVIDA사의 오픈 소스인 리눅스 기반의 SIPSET 1.5와 MPEG4IP 1.0을 이용한 멀티미디어 통신을 하는 플레이어를 기반으로 분석을 수행하였다[14]. 그림 3은 분석된 SIP UA 구조를 표현하였다.

SIP UA는 현재 SIP의 스택 기능을 하는 GUA Process (SIP Stack)와 스트림을 수신 및 디스플레이 하는 GUI Process(SIP Video)로 나뉘며 프로세스 간에는 파이프(PIPE)를 이용하여 통신하고 쓰레드들 간에는 큐를 이용하여 통신한다.

GUA Process는 call 수신을 위해 SIP 메시지를 처리하는 프로세스이다. Sip Receive 쓰레드는 caller인 UAC로부터 요청 메시지를 수신하며 Sip Parser 쓰레드는 요청 메시지를 이벤트 형태로 변환한다. Worker 쓰레드는 이벤트 메시지를 처리하며 GuiEvent 쓰레드는 GUI Process로부터 메시지를 수신하여 처리한다. Sip Send 쓰레드는 SIP 응답 메시지를 UAC에게 보낸다.

GUI Process는 call이 설정된 후에 비디오 프레임을 수신하는 프로세스이다. RTP 통신을 하기위한 미디어 세션이 설정이 되면 비디오와 오디오 개체가 각각 생성된다.

RTP Receive 쓰레드는 네트워크를 통해 수신한 인코딩 된 패킷을 받는 역할을 수행하며 이 패킷은 ByteStream 큐에 저장하게 된다. 처음 몇 초간은 지터의 영향을 줄이기 위해 초기 버퍼링 지연시간을 1~3초로 설정된다. 이 시간의 길이가 길면 지터로부터의 안전성은 확보되나 초기 지연이 길어지는 단점이 있다. 그 이후 RTP Receive 쓰레드는 Decode 쓰레드에게 START 메시지를 보내 비디오 스트리밍 플레이를 시작하게 된다. Decode 쓰레드는 비디오 프레임을 디코딩 후 버퍼링과 렌더링을 하기위해

Sync 쓰레드에게 넘겨준다. 또한 비디오 코덱은 윈도우 메신저와 통신하기 위해 H.261 혹은 MPEG4 등으로 디코딩 한다. Sync 쓰레드는 비디오와 오디오를 동기화 하기위한 쓰레드로서 동기화된 각각의 미디어 데이터를 출력한다.

3.1.2 SIP 세션 설정

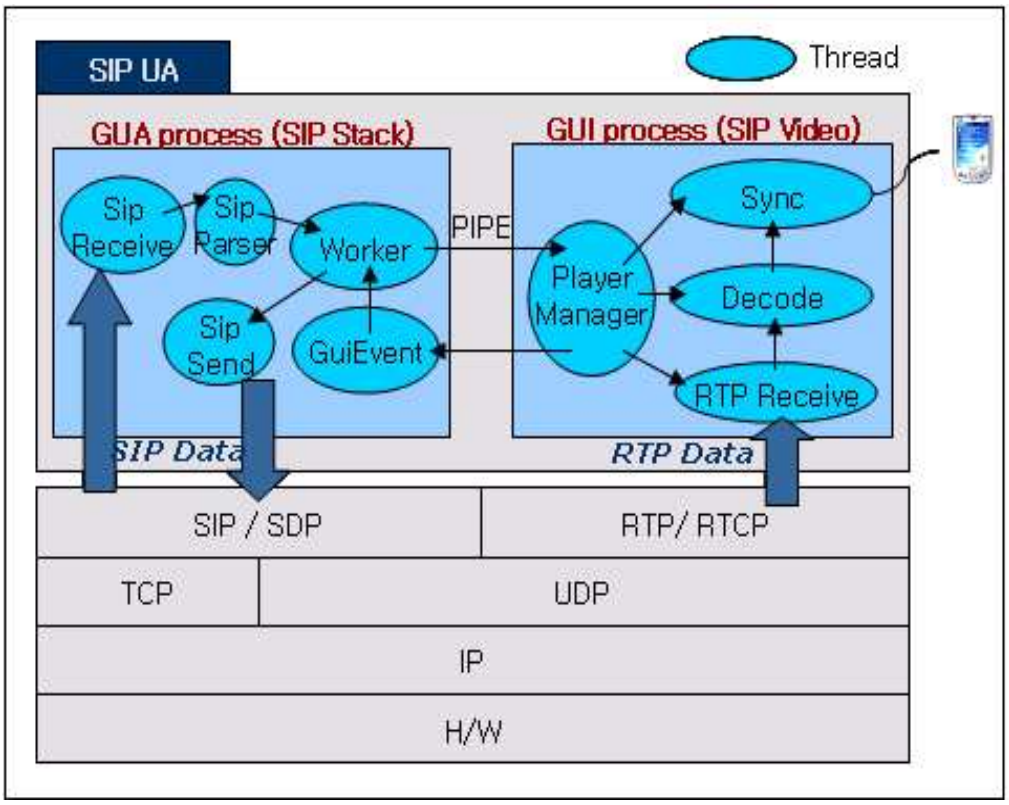


그림 3 멀티미디어 플레이어 소프트웨어 시스템 구조

그림 3은 멀티미디어 플레이어 소프트웨어 시스템 구조를 나타낸 그림이다. SIP 세션 설정 과정은 기본적인 SIP의 호출 흐름과 동일하며 다음과 같다.

Step 1) caller인 UAC가 callee인 UAS에게 요청 메시지를 보내면 Sip Receive 쓰레드는 요청 메시지를 수신하여 Sip Parser 쓰레드 큐에 삽입한다.

Step 2) Sip Parser 쓰레드는 큐에 들어온 요청 메시지를 이벤트 메시지 형태로 변환하여 Worker 쓰레드 큐에 삽입한다.

Step 3) Worker 쓰레드는 큐에 들어온 이벤트 메시지를 처리하여 파이프를 통해 Player Manager 쓰레드에게 메시지를 보낸다.

Step 4) Player Manager 쓰레드는 사용자에게 call을 알려준다. 사용자의 응답 메시지에 따라 파이프를 통해 확인 메시지를 GuiEvent 쓰레드에게 보낸다.

Step 5) GuiEvent 쓰레드는 파이프를 통해 메시지를 받아 Worker 쓰레드 큐에 삽입한다.

Step 6) Worker 쓰레드는 큐에 들어온 메시지를 SIP 응답 메시지로 생성하여 Sip Send 쓰레드 큐에 넣는다.

Step 7) Sip Send 쓰레드는 SIP 응답 메시지를 UAC에게 전송한다.

Step 8) 단말기간 SIP 세션이 설정된다.

3.1.3 시스템 구조

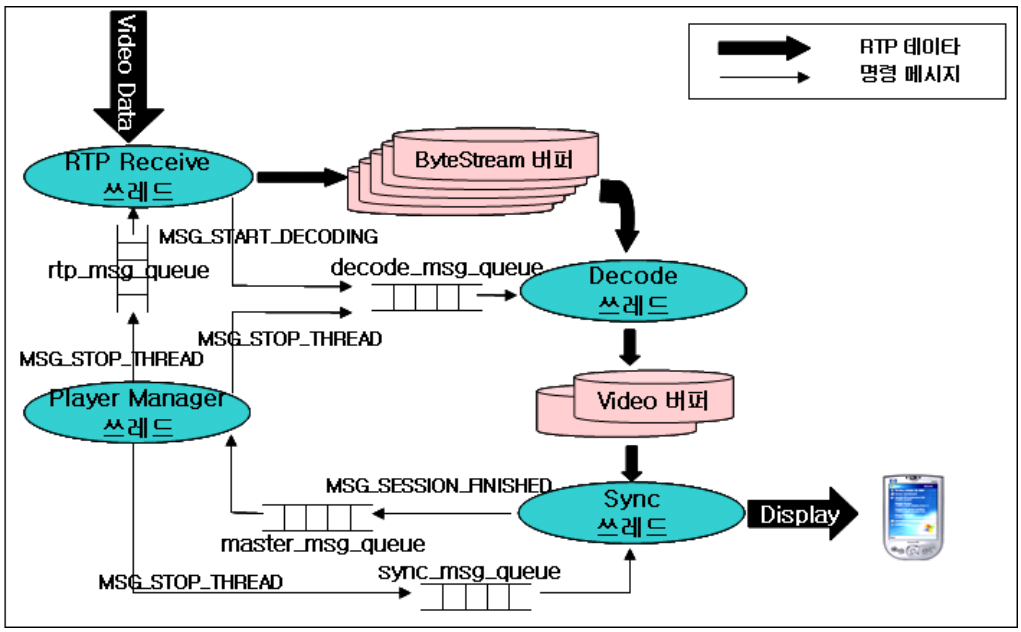


그림 4 SIP 비디오 프로세스

그림 4는 GUI Process의 쓰레드들간 미디어 데이터 통신을 나타내는 그림이다.

네트워크를 통해 들어온 패킷은 RTP Receive 쓰레드가 받아 하나의 비디오 프레임을 만들어 ByteStream 큐에 저장하며 Decode 쓰레드가 비디오 프레임을 읽어 Video 버퍼에 저장하게 된다. Sync 쓰레드는 오디오와 비디오의 동기화를 위해 존재하며 Video 버퍼에 있는 비디오 프레임을 출력한다.

1) RTP Receive 쓰레드

미디어 플레이어 스트리밍은 SIP 세션이 설정된 후 시작되며 RTP Receive 쓰레드는 상대방으로부터 인코딩 된 비디오 데이터를 받아 하나의 프레임으로 만든 후 Decode 쓰레드로 넘겨주는 쓰레드이다.

Decode 쓰레드는 3초간 버퍼링을 하고 상대방으로부터 비디오 패킷을

받아 그것을 헤더 부분과 데이터 부분을 나누는 프로세싱 작업을 거친 후 패킷의 크기를 보고 하나의 비디오 프레임으로 만들어 ByteStream 버퍼에 삽입한다.

2) Decode 쓰레드

Decode 쓰레드는 RTP Receive 쓰레드로부터 인코딩 된 비디오 데이터를 받아 비디오 프레임을 디코딩 하는 쓰레드이다.

Video 버퍼는 더블 버퍼링을 하며 각각의 비디오 프레임에 대한 동기화를 하기위해 비디오 프레임에 Timestamp 값을 넘겨주게 된다. Timestamp 값을 넘겨받은 비디오 프레임은 Video 버퍼로 저장된다.

3) Sync 쓰레드

Sync 쓰레드는 Decode 쓰레드로부터 비디오 / 오디오를 동기화하기 위한 쓰레드이다. Sync 쓰레드는 비디오 / 오디오 미디어가 생성되고 세션이 연결되었을 때 비디오 / 오디오의 초기 설정을 담당하고 있으며 Sync 쓰레드 내에 State Machine이 메시지 큐로 오는 각각의 메시지 형태에 따라 Play 하고 Stop 메시지를 Player Manager 쓰레드에게 보낸다. 또한 비디오와 오디오를 동기화하기 위한 Timestamp 값을 각각의 미디어에서 얻어 지연된 값을 조정한다.

3.2 실시간 멀티미디어 플레이어의 동작 모습

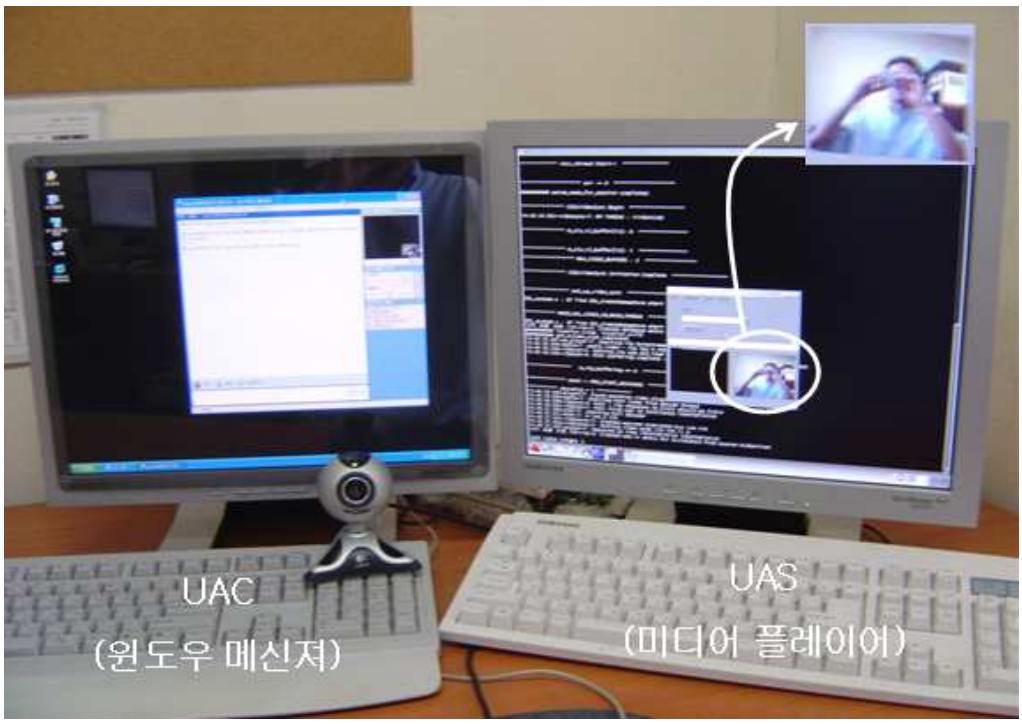


그림 5 멀티미디어 플레이어의 동작 모습

제 4장 RTCP 패킷 분석

4.1 RTCP

4.1.1 RTCP 개요

RTCP는 RTP 데이터의 피드백 정보를 제공하며, 피드백 정보는 최소한의 QoS를 제공하는데 사용된다. 하나의 세션에 참가한 참가자들은 주기적인 제어 패킷을 보내어 다음과 같은 기능을 수행하게 된다[15].

4.1.2 RTCP 기능

RTCP는 기본적으로 세션의 모든 참가자들에게 주기적인 제어 패킷을 보내어 다음의 기능들을 수행한다.

첫째, 데이터 분배의 품질에 대한 피드백을 제공한다. 이 기능은 RTCP Sender Report(SR)와 Receiver Report(RR)에 의해 수행된다.

둘째, RTP 소스의 식별을 위해 지속적인 식별자를 수송한다. 이 식별자는 CNAME(Canonical Name) 이라 부른다. SSRC는 충돌이 발생하거나 프로그램이 다시 시작될 경우에 변경될 수 있기 때문에 수신자는 각 참가자들을 일관성 있게 유지하기 위해서 CNAME을 필요로 한다. 이 두 기능은 모든 참가자들이 RTCP 패킷을 송신할 것을 요구한다. 따라서 RTP가 많은 수의 참가자를 수용할 수 있기 위해서는 그 송신 비율이 제어되어야 한다. 각 참가자가 모든 다른 참가자들에게 제어 패킷을 전송하기 때문에 각 참가자는 전체 참가자의 수를 파악할 수 있게 된다. 이 참가자 수를 이용해서 제어 패킷 전송 간격을 조정한다.

셋째, 최소한의 세션 제어 정보를 수송한다. 이 기능은 참가자가 마음대로 가입, 탈퇴할 수 있는 허술하게 제어되는 세션에서 유용하게 이용될

수 있는 기능으로 예를 들어 새로운 참가자의 신분 또는 이름을 제공함으로써 다른 모든 참가자의 사용자 인터페이스에 새로운 참가자의 이름을 알릴 수 있게 된다.

4.1.3 RTCP 패킷 형식

표 1 RTCP 패킷 형식

PT (Packet Type)	Message
200	Sender Report (SR)
201	Receiver Report (RR)
202	Source Description (SDES)
203	Good Bye (BYE)
204	Application specific

각 RTCP 패킷은 RTP 데이터 패킷과 비슷하게 고정부로 시작을 해서 패킷의 종류에 따라 달라지는 가변 길이의 구조화된 요소들이 뒤따르고 항상 32비트 경계로 끝난다. 이러한 정렬 방식과 고정부의 길이 필드는 RTCP 패킷을 "Stackable"하게 만든다. 즉, 다수의 RTCP 패킷들로 복합 RTCP 패킷을 만들어 하나의 하위 프로토콜 패킷으로 전송할 수 있게 한다. 복합 패킷을 이루는 RTCP 패킷의 수는 하위 프로토콜의 최대 길이에 의해 한정된다. 복합 패킷의 개별 RTCP 패킷들은 독립적으로 처리되지만 프로토콜의 기능을 수행하기 위해서 다음의 제약이 주어진다.

■ 송수신 보고(SR, RR)는 대역폭 제약이 허용하는 한 자주 보내서 송수

신 통계 치의 정밀도를 최대화시켜야 한다. 따라서 주기적으로 전송되는 복합 RTCP 패킷에는 반드시 보고가 포함되어 있어야 한다.

■ 새로운 수신자는 가능한 빨리 소스의 CNAME을 수신해서 소스를 식별하고 Lip-Sync와 같은 목적을 위해 그것을 매체와 연결해야 한다. 따라서 각 복합 RTCP 패킷은 SDES CNAME도 포함해야만 한다.

■ 모든 RTCP 패킷들은 적어도 두개의 개별 패킷으로 구성된 복합 패킷 형태로 전송되어야 한다. 이때 다음의 형식이 권고된다.

1) Encryption Prefix : 복합 패킷이 암호화 될 경우에만 무작위 32비트의 값이 전치된다. 이 값은 매번 전송되는 복합 패킷에 대해 새로운 값으로 선택된다.

2) SR 또는 RR : 복합 패킷의 첫 RTCP 패킷은 항상 보고 패킷이어야 한다. 이것은 수신되거나 송신된 데이터가 없을 경우에도 마찬가지로, 이 경우에는 빈 RR이 전송된다. 복합 패킷의 나머지 하나가 BYE일 경우에도 마찬가지로 보고 패킷이 첫 패킷으로 등 장해야 한다.

3) 추가 RR들 : 수신 보고 소스의 수가 31을 넘어서면 첫 보고 패킷에 이어 첨가 RR 패킷이 나와야 한다.

4) SDES : CNAME을 포함하는 하나의 SDES 패킷은 각 복합 패킷에 반드시 포함 되어야 한다. 다른 소스 설명 항목들도 응용에서 필요로 하면 대역폭 제한에 따라 포함될 수도 있다.

5) BYE 또는 APP : 주어진 SSRC/CSRC에 대해 마지막 패킷으로 전송되어야 하는 BYE를 제외한 나머지 패킷들은 어떤 순서로도 이어질 수 있다.

변환기와 혼합기들은 다수의 소스로부터 전달되는 개별 RTCP 패킷들을 가능하면 항상 복합 패킷으로 합쳐서 보내는 것이 바람직하다. 그렇게 함으로써 패킷 오버헤드를 줄일 수 있기 때문이다. 복합 패킷의 전체 길이가 하위 프로토콜의 MTU를 초과할 경우는 복합 패킷을 잘라서 여러 개의

하위 프로토콜 패킷에 포함시켜 보낼 수 있다.

4.1.4 전송간격

RTP는 수명에서 수 천명의 참가자를 하나의 세션에 참가시킬 수 있도록 설계되었다. 오디오 회의의 경우에 데이터 패킷은 참가자의 수에 상관없이 비교적 일정한 비트율을 가지지만(언제나 발언하는 사람은 한 두 사람이므로) 제어 패킷의 경우에는 참가자의 수에 비례하여 비트율이 증가(각 참가자가 나머지 모두에게 일정한 간격으로 제어 패킷을 전송하기 때문에)하게 된다. 따라서 제어 패킷의 전송 간격은 제어되어야 한다. RTCP에 할당되는 대역폭은 세션 대역폭의 5%로 고정되는 것이 바람직 하다.

1) 세션 멤버 수 관리

새로운 참가자에 대한 새로운 SSRC를 가지는 다수의 패킷을 수신할 때 까지 그 참가자는 유효하지 않은 것으로 간주한다. 각 참가자는 어떤 사이트가 5개의 RTCP 보고 간격 동안 RTP 혹은 RTCP 패킷을 보내지 않으면 그 사이트를 비활성화 상태로 표시하거나 삭제한다. 일단 사이트가 유효화되면 나중에 비활성 상태로 표시되어도 30분 까지는 그 사이트의 상태는 유지되고 RTCP 대역폭을 공유하는 전체 사이트의 수에 계속 계산된다.

2) SDES 대역폭 할당

SDES의 필수 항목인 CNAME 이외에 NAME, EMAIL 등과 같은 부가적인 정보에 제어 대역폭을 할당하는 것을 신중하게 고려해야 한다. 이유는 부가 정보들이 수신 보고와 CNAME의 전송 간격을 늘려서 전체 프로토콜의 성능 저하를 유발할 수 있기 때문이다. 일반적으로 한 참가자에 할당된 RTCP 대역폭의 20% 미만이 이러한 정보들을 수송하는데 이용되도록 권유 된다. 더욱이 모든 SDES 항목이 모든 응용에 포함될 필요는 없

다. 포함된 것들은 사용 정도에 따라 적정 비율의 대역폭을 할당받으면 된다.

4.2 송신자 / 수신자 보고

RTP 수신자는 자신이 동시에 송신자일 경우와 아닌 경우에 각각 SR와 RR를 보내서 수신 품질에 대한 피드백을 제공한다. 패킷 종류 코드 이외에 두 패킷의 다른 점은 SR이 활성화된 송신자들에 의해서 이용되는 20바이트의 송신자 정보 섹션을 가지고 있다는 것이다. 한 사이트가 최종 보고 이후 RTCP 전송 간격 동안 아무 데이터 패킷이라도 보냈으면 SR을 보내고 그렇지 않았을 경우에는 RR을 보낸다.

SR이나 RR 모두 0개 이상의 수신 보고 블럭을 가진다. 보고 블럭은 이 수신자의 최종 보고 이후에 이 수신자에게 RTP 데이터 패킷을 보낸 SSRC들 각각에 대해 하나씩 존재한다. CSRC 리스트에 나열된 CSRC들에 대해서는 보고를 보내지 않는다. SR 이나 RR 패킷에는 최대 31개의 수신 보고 블럭이 포함될 수 있기 때문에 그것을 초과하는 경우에는 추가적인 RR 패킷들이 초기 SR 또는 RR 패킷에 연이어 쌓일 수 있다.

4.2.1 SR : 송신자 보고 RTCP 패킷

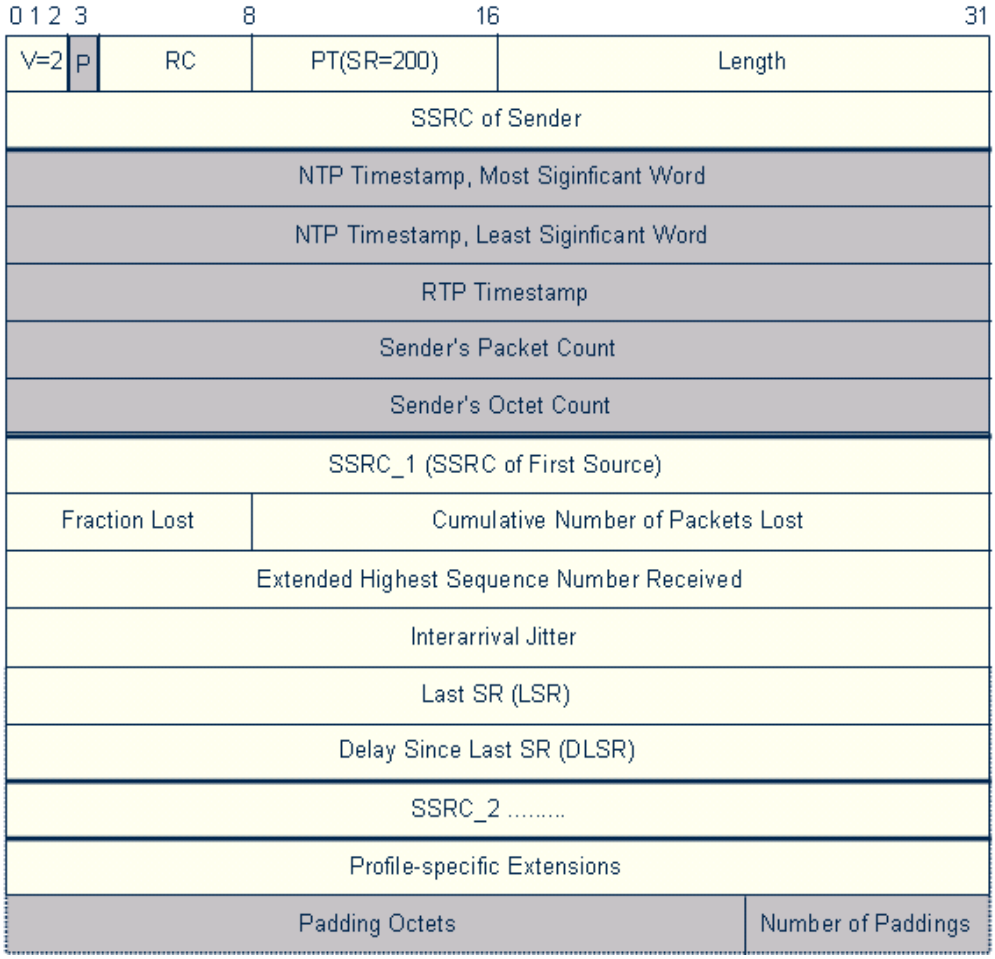


그림 6 Sender Report 패킷

그림 6은 송신자 보고 패킷구조를 나타낸 것으로 송신자 보고 패킷은 세 개의 섹션으로 구성되며, 프로파일에 따른 네 번째 확장 섹션이 추가될 수도 있다.

● 헤더

■ Report Count(RC) : 5비트 필드로 이 패킷에 포함된 수신 보고 블록

의 수를 나타낸다.

- Packet Type(PT) : 8비트 필드로 RTCP SR 패킷의 경우에는 200 값을 가진다.
- Length : 16비트 필드로 32비트 워드로 계산된 전체 패킷의 길이에서 1을 뺀 값으로 헤더와 채워넣기를 모두 포함한다. 0 값도 유효하다.

● 송신자 정보

- NTP Timestamp : 64비트 필드로 이 보고가 보내진 시간을 나타내고 Round-Trip 지연 계산에 이용된다.
- RTP Timestamp : 32비트 필드로 NTP Timestamp의 시간에 상응한다. 그러나 단위와 무작위 오프셋은 데이터 패킷의 RTP Timestamp들과 같다. 이것은 RTP Timestamp 계수와 실제 시간 간의 관계를 이용해서 해당 NTP Timestamp로부터 계산된다.
- 송신자 패킷 계수 : 32비트 필드로 전송 시작에서부터 이 SR 패킷이 생성될 때 까지 이 송신자에 의해 전송된 RTP 데이터 패킷의 총 수를 나타낸다. 이 값은 송신자가 SSRC 식별자를 변경하면 초기화 된다.
- 송신자 바이트 계수 : 32비트 필드로 전송 시작에서부터 이 SR 패킷이 생성될 때까지 이 송신자에 의해 전송된 RTP 데이터 패킷에서 헤더와 채워넣기를 제외한 페이로드 바이트의 수를 나타낸다. 송신자가 SSRC 값을 변경하면 이 필드는 초기화 된다.

● 수신 보고 블록

- SSRC_n : 32 비트 필드로, 이 수신 보고 블럭의 정보가 해당되는 소스의 SSRC 식별자를 나타낸다.
- Fraction lost : 8비트 필드로, 이전 SR 또는 RR 패킷이 송신된 이후 분실된 RTP 데이터 패킷의 비율이 고정 소수로 표현된다. 이것은 분실 비율을 256으로 곱한 후 정수부를 취하는 것과 같다. 이 값은

분실된 패킷의 수를 기대된 패킷의 수로 나누어 얻어진다. 중복 수신에 의해 분실이 음의 값을 가질 경우 분실 비율은 0이 된다.

■ Cumulative number of packets lost : 24비트 필드로, 소스 SSRC_n으로부터 수신 시작 이래로 분실한 총 RTP 데이터 패킷 수로 예측된 패킷 수에서 실제 수신된 패킷 수를 뺀 값으로 정의된다. 실제 수신된 패킷 수에는 지연 도착된 것과 중복 도착된 것도 포함된다.

■ Extended highest sequence number received : 32비트 필드로, 하위 16비트는 소스 SSRC_n으로부터 수신된 RTP 데이터 패킷의 최고 순번을 포함하고 상위 16비트는 그 순번을 알고리즘에 따라 관리되는 순번 사이클의 해당 계수로 확장 한다.

■ Interarrival jitter : 32비트 필드로, Timestamp 단위로 측정되고 비부호 정수로 표현되는 RTP 데이터 패킷 도착 시간 간의 통계적 가변성의 측정값을 나타낸다. Si가 패킷 i로부터의 RTP Timestamp이고 Ri가 패킷 i의RTP Timestamp 단위로 측정한 도착 시간일 때, 패킷 i와 j의 지터 D는 D(i,j) = (Rj-Ri) - (Sj-Si) = (Rj-Sj) - (Ri - Si)와 같이 표현되고, 도착간 지터 J는 J = J + (|D(i-1, i)| - J)/16로 정의된다. 여기서 1/16은 Gain Parameter로 적당한 수렴 속도를 보장하면서 좋은 소음 감쇄율을 보장 한다.

■ Last SR Timestamp(LSR) : 32비트 필드로, 소스 SSRC_n으로부터 최근에 받은 RTCP SR의 일부인 64비트 NTP 타임스탬프의 중간 32비트를 나타낸다. SR을 아직 받지 않았으면 이 필드는 0 값을 가진다.

■ Delay since last SR(DLSR) : 32비트 필드로, 소스 SSRC_n으로부터의 최종 SR 패킷 수신과 이 수신 보고 블럭 송신간의 지연을 나타내며, 1/65536초 단위로 표시된다. SSRC_n으로부터 아직 SR 패킷을 받지 못했을 경우 이 필드의 값은 0으로 설정된다.

4.2.2 RR : 수신자 보고 RTCP 패킷

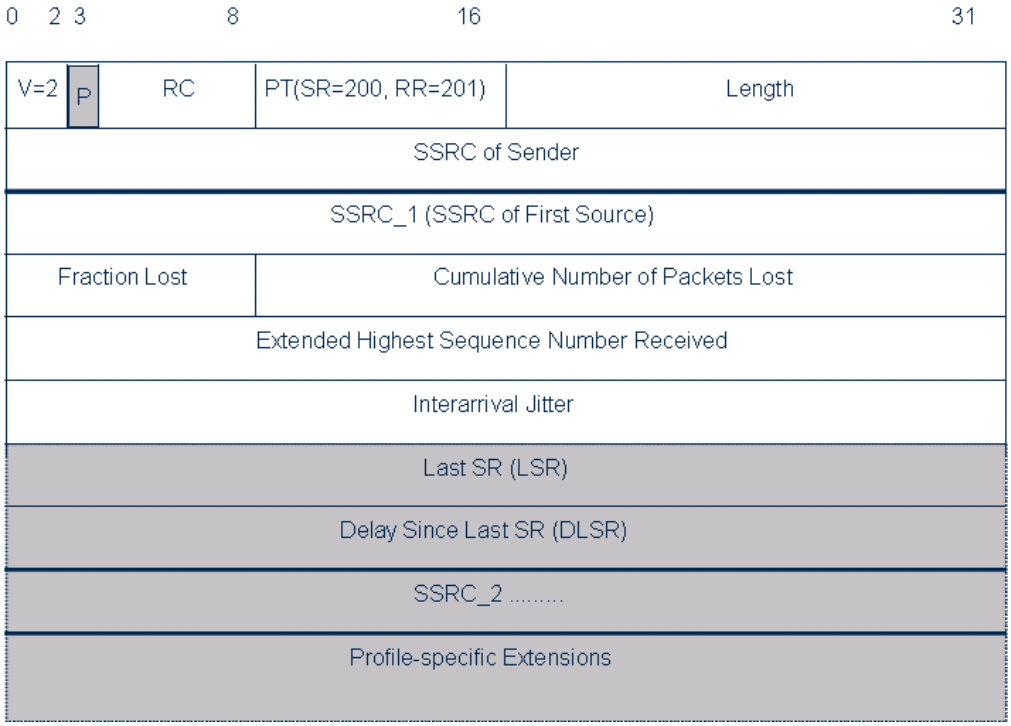


그림 7 Receiver Report 패킷

그림 7은 수신자 보고 패킷의 구조를 나타낸 것으로 RR 패킷의 형식은 패킷 종류 필드가 201 값을 가지고 송신자 정보 5워드(NTP/RTP Timestamp, Sender Packet/Byte Counter)가 생략되어 있다는 것을 제외하고는 SR과 같다. 보고할 데이터의 전송이나 수신이 없을 경우에는 복합 RTCP 패킷의 전단에 RC 값이 0인 빈 RR 패킷을 삽입한다.

1) 송신자 수신자 보고 확장

송신자 또는 수신자에 대해서 주기적으로 보고 되어야 할 추가 정보가 있을 경우 프로파일은 프로파일에 따른 (profile-specific) 또는 응용에 따른(application-specific) 송신자와 수신자 보고에 대한 확장을 정의해야 한

다. 추가적인 송신자 정보가 필요할 경우, 우선은 송신자 보고의 확장부에 포함되어야 한다. 수신자에 관계되는 정보가 포함될 경우 그 데이터는 기존의 수신 보고 블럭의 배열에 병치하는 블럭의 배열로 구조화될 것이다. 즉, 블럭의 수가 RC 필드에 반영될 것이다.

2) 송신자 수신자 보고 분석

수신 품질 피드백은 송신자뿐만 아니라 다른 수신자들이나 제삼자 모니터들에도 유용할 것이다. 송신자는 피드백에 따라서 전송 율을 변경할 수 있고, 수신자는 문제가 지역적인 것인지, 전역적인 것인지를 결정할 수 있으며, 망 관리자는 멀티캐스트 분배를 위한 그들의 망의 성능을 평가하기 위해서 RTCP 패킷만을 수신하는 프로파일에 무관한 모니터를 이용할 수 있다.

송신자 정보와 수신자 보고 블럭에 모두 누적 계수가 이용되기 때문에 두 보고 사이의 차이점을 계산하여 단기 그리고 장기에 걸친 측정을 하고 보고의 분실에 대한 회복력을 제공한다. 수신된 최근 두 보고의 차이가 분배의 최근 품질을 측정하는데 이용될 수 있다. 두 보고 사이의 기간 동안 이러한 차이점으로부터 여러 가지 Rate들이 계산될 수 있도록 NTP 타임스탬프가 포함되었다. 타임스탬프는 데이터 인코딩을 위한 클럭 속도와 무관하기 때문에 인코딩과 프로파일에 무관한 품질 감시기를 구현할 수 있다.

4.2.3 SDES : 소스 설명 RTCP 패킷

SDES는 헤더와 0개 이상의 Chunk로 구성된 3-계층 구조로 각 Chunk는 그 Chunk에 식별된 소스를 설명하는 항목들로 구성된다.

■ Packet Type(PT) : 8비트 필드로, RTCP SDES 패킷의 경우에는 202

값을 갖는다.

■ Source Count(SC) : 5비트 필드로, SDES 패킷에 포함된 SSRC/CSRC Chunk의 수를 나타낸다. 0 값도 유효하지만 아무런 의미도 없다.

각 Chunk는 SSRC/CSRC 식별자와 SSRC/CSRC에 관계되는 정보를 수송하는 0개 이상의 항목들의 리스트로 구성된다. 각 Chunk는 32비트 경계에서 시작한다. 각 항목은 8비트의 종류 필드와 텍스트의 길이를 나타내는 8비트의 바이트 계수, 그리고 텍스트 자체로 구성된다. 텍스트는 255 바이트까지 이용할 수 있지만 RTCP 대역폭 소비에 관계된다는 것을 인지해야 한다.

텍스트는 ISO 10646 Annex F에 명시된 UTF-2(UTF-8 또는 UTF-FSS로도 알려짐) 인코딩 방법에 따라 인코딩 된다. US-ASCII는 이 인코딩 기법의 부분집합이기 때문에 부가적인 인코딩을 필요로 하지 않고, 캐릭터의 최상위 비트를 1로 설정함으로써 멀티-바이트 인코딩 기법으로 인코딩 된 사실을 알린다.

종단 시스템은 고정 RTP 헤더의 SSRC와 같은 값의 소스 식별자를 포함하는 하나의 SDES 패킷을 보낸다. 각 항목들 가운데 CNAME 만이 필수이다.

CNAME(Canonical End-point Identifier, 1) : CNAME은 무작위로 할당된 SSRC 식별자에 충돌이 발생할 수 있고, 프로그램이 재시작 되었을 때 SSRC 식별자가 변경될 수 있기 때문에 새로운 SSRC 식별자와 상시적으로 남아있는 소스를 위한 식별자와의 바인딩을 제공한다. SSRC 식별자처럼 CNAME도 RTP 세션 내에서 유일해야 한다.

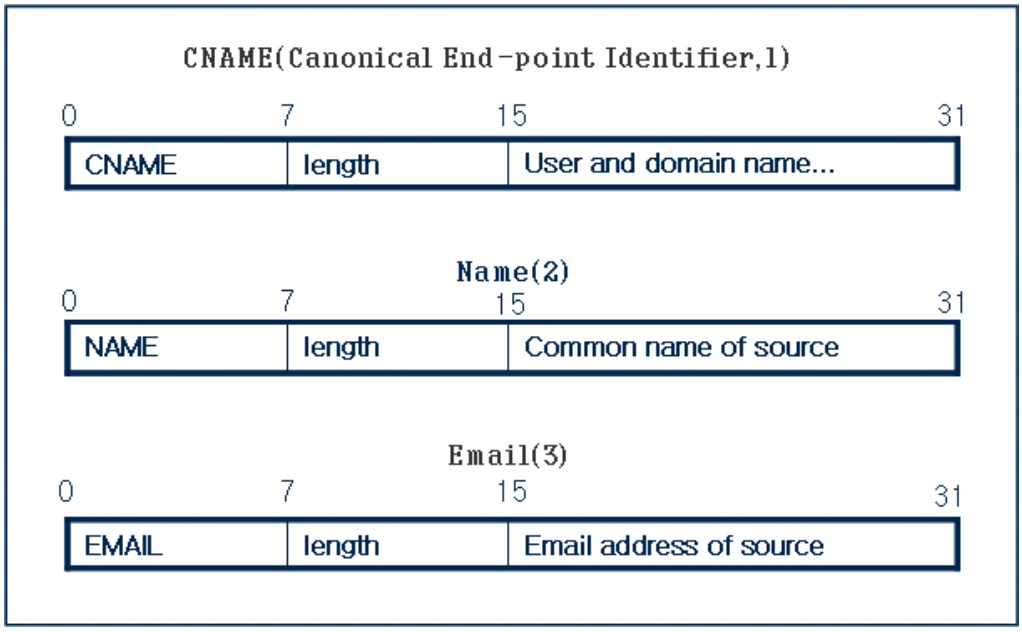


그림 8 CNAME

그림 8에서 CNAME은 가능하면 자동적으로 만들어져야 한다. 이러한 요구 사항을 만족시키기 위해서 프로파일이 별도로 명시하지 않은 한 "user@host" 또는 "host"의 형식이 이용되어야 한다. 이러한 형식의 장점은 NAME 아이템을 별도로 전달할 필요가 없다는 점이고, 단점은 한 사용자가 한 호스트에서 다수의 소스를 생성할 경우 각 소스에 대해 유일한 식별자가 제공되지 못한다는 점이다.

- NAME(User name, 2) : 소스를 설명하는데 이용되는 실명으로, 사용자가 원하는 어떤 형태로도 기록될 수 있다. NAME은 적어도 세션 기간 동안에는 변함이 없어야 하며, 세션 내에서 유일할 필요는 없다. 프로파일에서 각 항목들의 우선순위를 정하여 전송되는 횟수를 결정하는데 도움을 줄 수 있다.
- EMAIL(Electronic mail address, 3) : RFC822에 명시된 형식의 전자메일 주소로, 세션 기간 동안에는 변함이 없어야 한다.

PHONE(Phone number, 4) : 국제 접속 코드를 + 기호로 변경한 형태의 전화번호이다. 예) +1 908 555 1212

- LOC(Geographic user location, 5) : 이 항목은 응용에 따라서 정밀도가 달라진다. "Murray Hill, New Jersey", "Room 2A244, AT&T BL MH"와 같이 표현될 수 있다. 정밀도의 결정은 구현자나 사용자에 달려있지만 각각의 형식과 내용은 프로파일에 서술되어 있을 수 있다. 이동 호스트를 제외하고는 세션 기간 동안에는 변함이 없어야 한다.
- TOOL(Application or tool name, 6) : 실시간 스트림을 생성하는 응용의 이름과 버전 등을 나타낸다. 이 정보는 디버깅 목적에 유용하다. 이 정보 또한 세션 기간 동안에는 변함이 없어야 한다.
- NOTE(Notice/status, 7) : 소스의 현 상태를 설명하는 임시 메시지(예, "on the phone, can't talk")를 전달하거나 세미나 중일 경우 발표의 제목을 전달하는 등의 목적으로 이용되는 부분 이지만 프로파일에 의해 다른 용도로 정의될 수도 있다. NOTE는 예외적인 정보를 수송하는 데만 이용되어야 하고 모든 참가자들이 일상적인 용도로 이용해서는 안 된다. 그렇게 되면 제어 패킷 대역폭이 낭비되어 CNAME과 같은 중요한 정보가 전달되는 간격이 늘어나기 때문이다. NOTE 항목이 활성화된 동안에는 디스플레이하는 것이 중요하기 때문에 NOTE 항목이 RTCP 대역폭을 차지할 수 있도록 CNAME이 아닌 다른 항목들의 전송을 자제하는 것이 좋다. 수신측에서 NOTE 항목 반복 기간의 몇 배 기간 동안 또는 20~30 RTCP 기간 동안 NOTE를 받지 못하면 NOTE 항목이 비활성화 된 것으로 간주해야 한다.
- PRIV(Private extensions, 8) : 이 항목은 실험적인 또는 응용에 따른 SDES 확장을 정의하는데 이용된다. 이 항목은 길이-스트링 쌍으로 구성된 전치부(prefix)와 항목의 나머지를 채우고 원하는 정보를 수송하는 값 스트링(value string)을 포함한다. 전치부 길이 필드는 8비트 필드이고 전치 스트링은 PRIV 항목을 이 응용

이 수신할 다른 PRIV 항목들에 대해 유일하게끔 정의하는 이름으로 사람에 의해서 선택된다. SDES PRIV 전치부는 IANA에 등록되지 않는다. 대신 어떤 종류의 PRIV 항목이 일반적인 유용성을 가지고 있는 것으로 판명되면 IANA에 등록된 정규 SDES 항목 타입을 할당되어 전치부를 필요하지 않게 해야 한다. 이렇게 하는 것이 사용을 단순화 시키고 전송 효율을 증가시킨다.

4.2.4 BYE : Goodbye RTCP 패킷

BYE 패킷은 소스가 더 이상 활성화 상태가 아님을 알린다.

- Packet Type(PT) : 8비트 필드로, RTCP BYE 패킷은 203 값을 갖는다.
- Source Count(SC) : 5비트 필드로 BYE 패킷에 포함된 SSRC/CSRC 식별자의 수를 나타낸다. 0 값도 유효하지만 의미는 없다.

BYE 패킷이 혼합기에 도착하면 혼합기는 SSRC/CSRC 식별자 변경 없이 그대로 전달한다. 혼합기가 수행이 중지될 때 자신의 SSRC 식별자와 자신이 처리하는 모든 제공 소스들을 나열한 BYE 패킷을 보내야 한다. 부가적으로 BYE 패킷은 8비트 바이트 계수와 계수 바이트 크기의 탈퇴 이유(예, "camera malfunction", "RTP loop detected")를 나타내는 텍스트를 포함할 수 있다.

4.2.5 APP : Application-defined RTCP 패킷

새로운 응용이나 새로운 기능이 개발되었을 때 패킷 종류 값을 등록할

필요 없이 실험을 목적으로 사용되기 위한 패킷이다. 인식할 수 없는 이름을 가진 APP 패킷은 무시되어야 한다. 실험 후에 광범위한 이용성이 판명되면 각 APP 패킷은 부 타입과 이름 필드 없이 재정의 되어 RTCP 패킷 타입을 이용해서 IANA(Internet Assigned Number Authority)에 등록하도록 권고된다[11].

- Subtype : 5비트 필드로 일련의 APP 패킷들이 하나의 유일한 이름으로 정의될 수 있도록 하는 부 타입으로 사용되거나 응용에 종속되는 데이터를 위해 사용될 수 있다.
- Packet Type(PT) : 8비트 필드로 RTCP APP 패킷은 204 값을 가진다.
- Name : 4바이트 필드로 이 응용이 수신할 다른 APP 패킷들에 대해 유일하도록 APP 패킷을 정의하는 이름으로 사용자에 의해 선택된다.
- Application-specific Data : 가변 길이의 필드로 APP 패킷에 나타날 수도, 나타나지 않을 수도 있다. 이 필드는 RTP에 의해 해석되지 않고 응용에 의해 해석된다. 이 필드의 길이는 32비트의 배수여야 한다.

제 5장 실험 및 분석

5.1 실험

5.1.1 실험 환경

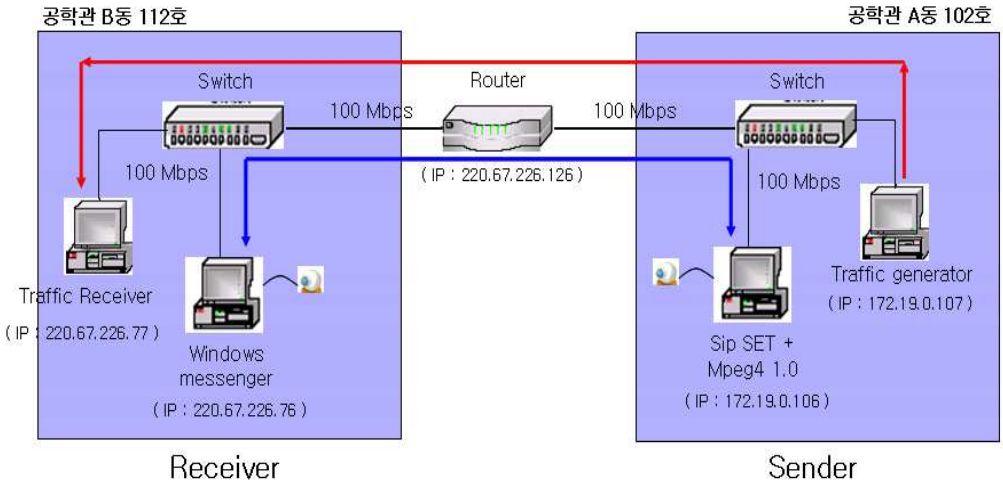


그림 9 실험 환경

그림 9는 SIP 기반 실시간 멀티미디어의 성능분석을 위한 실험환경이다.

Receiver는 한성대학교 공학관 B동 112호에 위치하고 Windows Messenger 4.7[16]을 사용하였다. Sender는 한성대학교 공학관 A동 102호에 위치하고 SIP Set + MPEG4IP 1.0을 사용한 미디어 플레이어를 사용하였다. Receiver와 Sender 사이에는 Router가 1개 존재하며 100Mbps의 LAN으로 연결되었다. 이 라우터는 Receiver와 Sender에 있는 각각의 스위치를 통해 미디어 세션 연결 시 멀티미디어 데이터를 주고받을 수 있도록 구성했다.

멀티미디어 세션이 연결되고 SIP기반 실시간 멀티미디어 통신을 위한

성능을 분석하기 위해 트래픽 생성기를 통해 순수한 네트워크의 스트레스를 주기 위해 Receiver쪽에 트래픽 생성기에서 발생된 패킷을 받는 Traffic Receiver를 설치하였다.

5.1.2 시스템 사양

표 2 시스템 사양

	112호 (RR : Receiver Report)	102호 (SR : Sender Report)
시스템 사양	■ Pentinum4 2.8GHz ■ 512MB RAM	■ Pentinum4 1.7GHz ■ 256MB RAM
응용 프로그램	■ Microsoft Excel 2003 ■ XML Parser(Xalan)	■ 리눅스 기반 Ethereal 0.10.5 ■ Traffic generator - Sniffer Pro 4.70

본 실험에서 102호와 112호와의 영상 통신을 하기위해 Windows Messenger가 제공하는 H.261 코덱을 사용하였으며, 비디오 크기는 178x144(QCIF)로 설정하였다. 카메라는 초당 30프레임을 지원하는 Logitech Quick cam을 사용하였다. 성능요소들을 측정하기 위한 절차는 다음과 같다

1) Ethereal 0.10.5 에서 RTCP패킷 캡쳐
2) XML[17]문서로 변환
3) 파싱을 통한 데이터 검출
4) 성능 요인들의 값 추출
5) 추출된 데이터를 Microsoft Excel 2003으로 분석

5.1.3 시스템 환경 설정

Ethereal은 리눅스를 OS로 하는 시스템에는 '전체설치'시 기본적으로 제공되어 진다.

본 논문에서는 최신 버전을 사용하기 위해 Ethereal 0.10.5[18]으로 업그레이드를 하였다. Ethereal 0.10.5를 다운로드 가능한 곳은 http://www.ethereal.com 에 가면 다운로드가 가능하다. 트래픽 발생기는 윈도우즈 기반 Sniffer Pro 4.70[19]을 사용하였으며 http://www.sniffer.com에 가면 다운로드가 가능하다. XML Parser는 Xalan을 사용하였다. 이유는 Ethereal 0.10.5가 패킷을 저장할 때 XML 문서로 저장가능하기 때문에 이 XML문서를 편집하여 성능요인들을 추출하기 위한 작업을 하기위해 사용했다. Xalan[20] 설치 과정은 아래와 같다. 본 연구에서는 윈도우즈 XP를 OS로 하는 시스템에서 설치를 하였다

1) http://www.apache.org/xalan-j/에서 xalan.jar을 다운로드한다.

2) xalan.jar을 JDK가 설치된 폴더의 lib폴더에 (C:\jdk1.3.1_07\lib) 다운받는다.

3) 환경변수를 등록한 후 java org.apache.xalan.xslt.Process을 입력하고 그림 10과 같은 화면이 나오면 설치가 완료된 것이다.

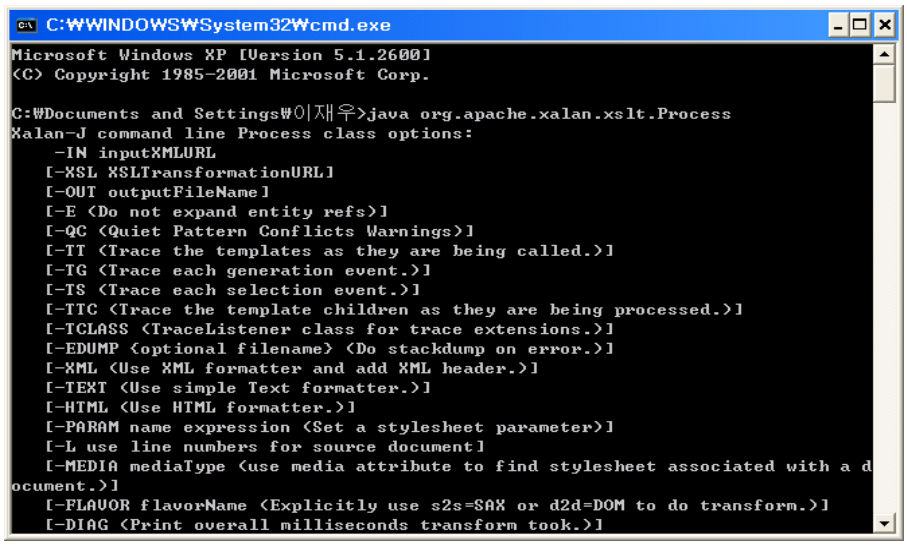


그림 10 Xalan 설치

5.2 성능 요인

5.2.1 고려해야 할 성능 요인

실시간 멀티미디어 전송시 트래픽은 트래픽 파라미터 및 네트워크 설계와 같이 신중하게 고려해야 하는 다양한 문제들이 있다[21, 22].
주요 고려 사항들은 다음과 같다.

1) End-to-end Delay
   - 패킷이 네트워크를 통해 종단 간에 전달되는데 소요되는 시간을 측정하였다. (Round-Trip Delay)
2) Jitter
   - 패킷이 도착할 예정시간과 패킷이 실제로 도착한 시간 간의 시간(편차)를 측정하였다.
3) Packet loss rate
   - 비디오 프레임을 전송시 패킷의 손실로 인한 비디오의 품질을 결정하는 중요한 요인이 된다.

5.2.2 파싱을 통한 성능 요인 데이터 검출

본 논문에서는 End-to-end Delay, Jitter, Packet loss rate를 계산하기 위한 패킷 분석기인 Ethereal 0.10.5를 사용하였다. 패킷 분석기에서 RTCP 패킷만을 XML문서로 저장한다[23, 24].

그림 11은 네트워크 트래픽이 없는 상태에서 저장된 XML문서를 나타내고 있다.

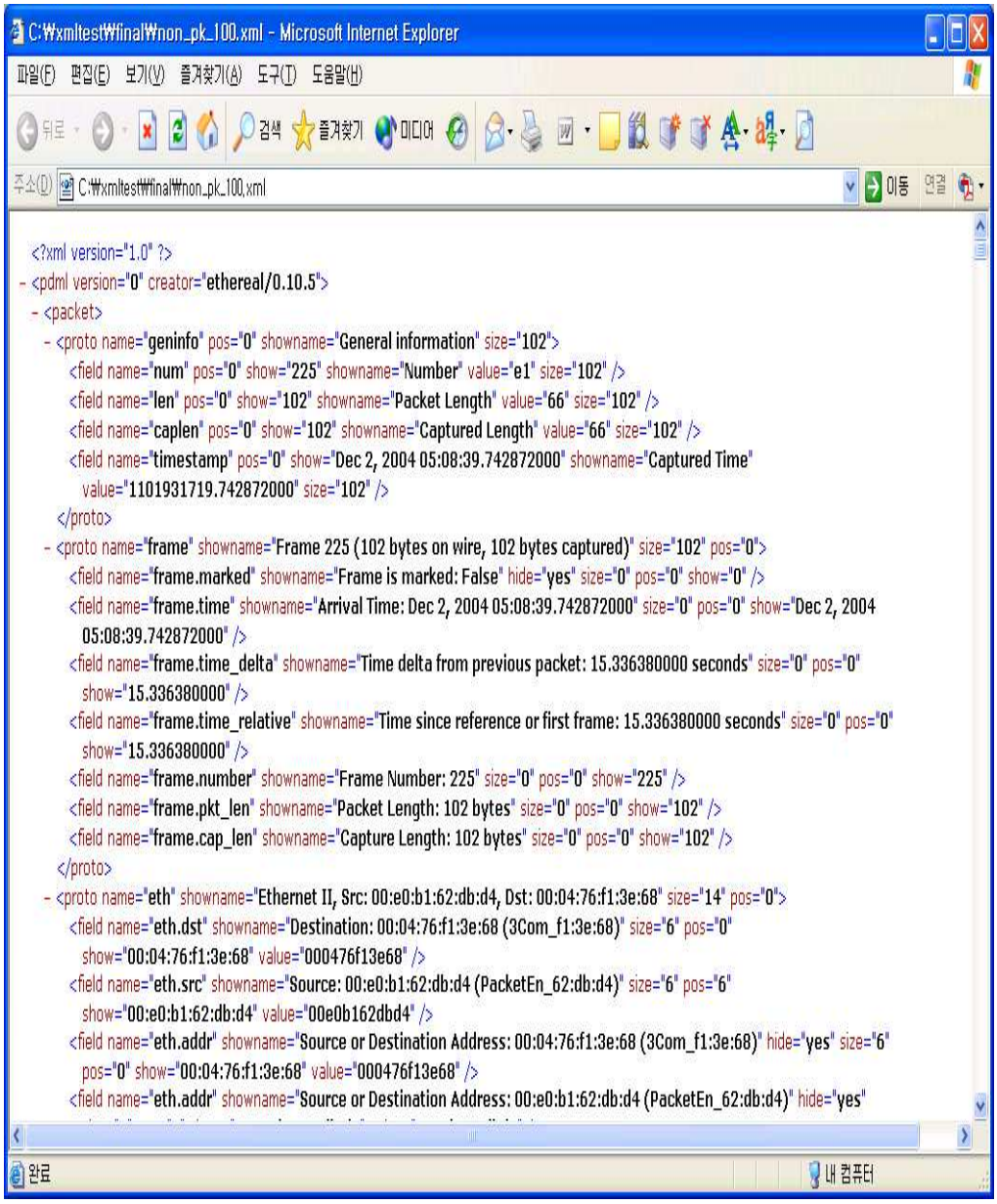


그림 11 RTCP 패킷 XML 문서

본 연구에서 필요한 End-to-end Delay, Jitter, Packet loss rate를 계산하기위해 Xalan Parser를 사용하여 XML문서를 변환하는 작업을 수행하였다. 프로세싱 프로그램은 그림 12와 같다[25].

그림 12 성능 요인 데이터 추출을 위한 XML 프로세서

그림 12는 프로세싱과정이 올바르게 됐는지 검사하기위한 절차로 HTML문서로 변화하는 작업을 수행한다. 아래 그림은 파싱 하기위한 소스

와 결과를 나타내고 있다. 그림 13-A와 그림 13-B는 HTML문서로 변환하기 위한 XML 프로세서 와 파싱 결과를 나타내고 있다.

process.xsl - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
<?xml version="1.0" encoding="EUC-KR"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" encoding="EUC-KR"/>

<!-- 문서 -->
        <xsl:template match="/">
         <html>
          <xsl:apply-templates select="root"/>
         </html>
        </xsl:template>

<!-- 패킷 리스트 -->
        <xsl:template match="root">
         <body>
          <xsl:apply-templates select="packet"/>
         </body>
        </xsl:template>

<!-- 패킷 -->
        <xsl:template match="packet">
         <table border='1'>
          <tr>
           <td><xsl:apply-templates select="RR"/><xsl:apply-templates
select="SR"/></td>
          </tr>
          <tr>
           <td><xsl:apply-templates select="pksize"/></td>
          </tr>
          <tr>
           <td>Source / port</td><td><xsl:apply-templates
select="srcadd"/></td><td><xsl:apply-templates select="srcport"/></td>
          </tr>
          <tr>
           <td>Destination / port</td><td><xsl:apply-templates
select="dstadd"/></td><td><xsl:apply-templates select="dstport"/></td>
          </tr>
          <tr>
           <td>Packet log time</td><td><xsl:apply-templates
select="pktimestamp"/></td>
          </tr>
          <tr>
           <td>Sender's packet count</td><td><xsl:apply-templates
select="pkcount"/></td>
```

그림 13-A HTML문서로 변환하기 위한 XML 프로세서

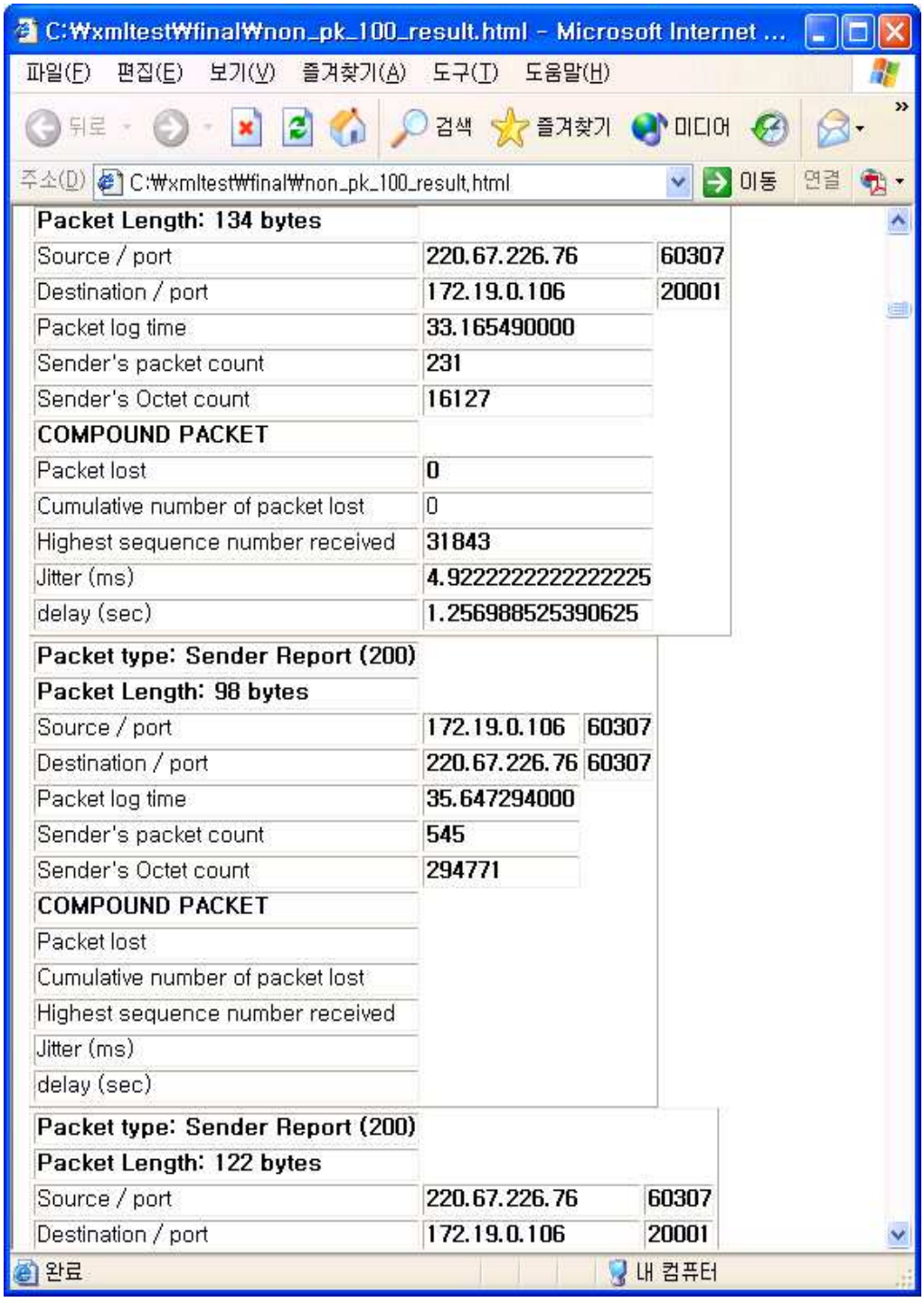

C:₩xmltest₩final₩non_pk_100_result.html - Microsoft Internet ...
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기 미디어
주소(D) C:₩xmltest₩final₩non_pk_100_result.html
이동 연결
Packet Length: 134 bytes
Source / port 220.67.226.76 60307
Destination / port 172.19.0.106 20001
Packet log time 33.165490000
Sender's packet count 231
Sender's Octet count 16127
COMPOUND PACKET
Packet lost 0
Cumulative number of packet lost 0
Highest sequence number received 31843
Jitter (ms) 4.9222222222222225
delay (sec) 1.256988525390625
Packet type: Sender Report (200)
Packet Length: 98 bytes
Source / port 172.19.0.106 60307
Destination / port 220.67.226.76 60307
Packet log time 35.647294000
Sender's packet count 545
Sender's Octet count 294771
COMPOUND PACKET
Packet lost
Cumulative number of packet lost
Highest sequence number received
Jitter (ms)
delay (sec)
Packet type: Sender Report (200)
Packet Length: 122 bytes
Source / port 220.67.226.76 60307
Destination / port 172.19.0.106 20001
완료
내 컴퓨터

그림 13-B 파싱 결과

5.3 데이터 계산

5.3.1 성능 요인의 계산을 위한 데이터 추출

성능 요인을 위한 각각의 End-to-end Delay, Jitter, Packet loss rate를 계산하기 위해 아래의 그림은 파싱된 데이터를 Microsoft Excel 2003으로 변환한 데이터 중 일부를 나타낸 것이다. 이를 바탕으로 각각의 성능요인 End-to-end Delay, Jitter, Packet loss rate를 계산하는 것을 기술 하고자 한다. 그림 14는 프로세싱으로 추출된 데이터를 나타내고 있다.

pktimestamp	srcadd	dstadd	srcpor	dstport	cumulpklost	RTPRecv	jitter	delay
63.791467	172.19.0.106	220.67.226.76	58923	58923				
68.58781	220.67.226.76	172.19.0.106	58923	20001	0	28883	3.96666667	4.79598999
69.949887	172.19.0.106	220.67.226.76	58923	58923				
72.682883	220.67.226.76	172.19.0.106	58923	20001	0	28986	3.73333333	2.731994629

그림 14 프로세싱으로 추출된 데이터

5.3.2 End-to-end Delay

End-to-end Delay계산은 Receiver Report 패킷의 필드 중 32비트로 된 Delay since last SR(DLSR) 필드 부분을 계산한 것이며 1/65536 초를 나타낸다.

그림 15는 End-to-end Delay를 계산하기위해 도식화해 낸 그림이며, Com1이 Sender, Com2는 Receiver를 나타내고 있다. 그림15에서 보듯이 Round-Trip Delay = t1 + t2 이므로,
Round-Trip Delay = RR 메시지를 받은 시간 - SR 메시지를 보낸 시간 - RR 메시지 생성시간을 구하면 Round-Trip Delay값을 구할 수 있다.

Round-Trip Delay = 68.58781 - 63.791467 - 4.79598999
= 0.353009766 (sec) ▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪

▪▪▪▪▪▪▪▪▪▪▪▪ <수식 1>

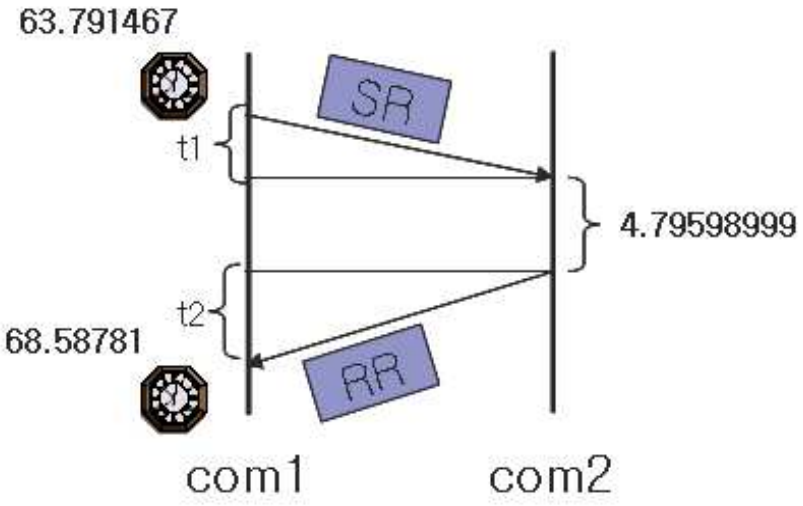


그림 15 End-to-end Delay 계산

5.3.3 Jitter

Receiver Report 패킷 필드 중 Interarrival jitter 필드 값을 90kHz로 샘플링 한 결과를 나타냈다.

이 값을 얻기 위해 페이로드 형식(Payload Type)을 알아야 한다. 페이로드 형식(Payload Type)은 오디오 또는 비디오 부호화와 같은 특별한 페이로드의 내용을 RTP로 전송하는 방법을 정의한다. 현재 RTP를 위해 정의된 페이로드 종류[26]가 표 3에 나열되어 있다. RTP/AVP (Audio/Video Profile)[27]는 응용 프로그램에서 사용될 페이로드 형식의 집합을 위한 페이로드 숫자를 할당한다.

본 논문에서는 Microsoft Windows Messenger와 통신하기 위해 코덱을 H.261을 사용하였다. 따라서 90kHz로 샘플링을 하였다[28].

표 3 페이로드 종류

페이로드 종류 (PT)	코덱 이름	오디오/비디오 (A/V)	표본화 속도 (Hz)	채널 수
0	PCMU	A	8000	1
1	1016	A	8000	1
2	G.721	A	8000	1
3	GSM	A	8000	1
5	DVI4	A	8000	1
6	DVI4	A	16000	1
7	LPC	A	8000	1
8	PCMA	A	8000	1
9	G.722	A	8000	1
10	L16	A	44100	2
11	L16	A	44100	1
12	QCELP	A	8000	1
13	CN	A	8000	1
14	MPA	A	90000	
15	G.728	A	8000	1
16	DVI4	A	11025	1
17	DVI4	A	22050	1
18	G.729	A	8000	1
25	CelB	V	90000	
26	JPEG	V	90000	
28	nv	V	90000	
31	H.261	V	90000	
32	MPV	V	90000	
33	MP2T	AV	90000	
72-76	Reserved	N/A	N/A	N/A
96-127	Dynamic	?		

그림 14에서의 Jitter값은 계산된 값으로 프로세싱과정에서 그 값을 미리 계산하여 얻은 값이다. 계산은 아래와 같다.

Receiver Report 패킷 필드 중 Interarrival jitter 필드 값 : 357 이고 PT가 H.261이므로 90kHz로 샘플링 하였으므로,

Jitter = 357 / 90 = 3.96666667 (ms) ······································ <수식 2>

의 값을 얻을 수 있다.

5.3.4 Packet loss rate

Receiver Report 패킷에 cumulative number of packet lost 부분은 패킷이 loss된 누적 개수를 나타낸다. 또한 Receiver Report 패킷 필드 중 Highest Sequence Number Received 이 부분은 RTP 데이터의 sequence number중 가장 최근에 받은 sequence number를 나타낸다.

Packet loss rate = (패킷 손실 개수 / 수신한 총 RTP 패킷의 개수) * 100 ··· <수식 3>

=> (32 - 19) / (31619 - 31507) * 100

= 8.837209 (%)

라는 결과를 얻어 Packet loss rate는 8.837209 (%) 의 값을 얻을 수 있다.

5.4 분석

5.4.1 실제 트래픽 측정

SIPSET 1.5 + MPEG4IP 1.0의 시스템의 Round-Trip Delay, Jitter, Packet Loss Rate를 측정하기 위해 102호에 위치한 트래픽 발생기를 통해

공학관 A동과 공학관 B동 사이에 있는 라우터에 스트레스를 주었다. 실험은 트래픽 발생기로 네트워크에 10%씩 증가 시켜가며 트래픽을 발생시켰다.

트래픽 발생기를 통해 100Mbps 네트워크에서 실제 100%의 실제 트래픽은 62.6Mbps인 것을 확인 하였다. 그러므로 최대 가능 트래픽은 62.6Mbps로 결론 짓는다.

5.4.2 트래픽에 증가에 따른 성능 요인

표 4는 대역폭 100Mbps에서 각 성능 요인의 평균값을 나타낸 표이다. 이는 20번의 표본 데이터를 바탕으로 평균을 기준으로 가장 높은 값 2개, 가장 낮은 값 2개를 제외한 16번의 표본 데이터를 평균을 내어 평균에 가장 가까운 값을 표본 데이터로 결정하였다. 그 결과는 표 4와 같다.

표 4 대역폭 100Mbps(최대가능 트래픽 62.6 Mbps)에서 각 성능 요인의 평균값

100Mbps의 데이터 전송을 지원하는 네트워크 환경에서 초당 58.4Mbps의 전송시 93.4%의 대역폭을 차지함으로 측정값은 의미 없는 데이터(X)로 취급 했으며 각각의 평균을 이용하여 다음과 같은 그래프를 얻을 수 있었다. 표 4의 내용은 다음과 같다.

(1) 실제 트래픽 : 100Mbps의 데이터 전송을 지원하는 네트워크가 실제 전송할 수 있는 최대 트래픽 비율.

(2) 실제 네트워크 로드 : 최대가능 트래픽 62.6Mbps를 100%로 설정 하였을 때 발생시킨 상대적 트래픽.

(3) 이론적 네트워크 로드 : 100Mbps의 데이터 전송을 지원하는 네트워크가 최대 100Mbps를 모두 지원한다고 가정 했을 때 실제 트래픽에 의한 비율.

성능요인 / 트래픽			Jitter (ms)	Round-Trip Delay (ms)	Packet Loss rate (%)
실제 트래픽 (Mbps)	실제 네트워크 로드 (%) 실제트래픽/최대가능트래픽)	이론적 네트워크 로드 (%)			
0	0	0	4.408918	0.57563451	0
9.6	15.3	9.6	17.83282	15.90951004	11.91854
19.2	30.7	19.2	17.83509	17.22594885	25.89112
28.8	46.0	28.8	18.72687	27.86714	38.91879
38.4	61.4	38.4	22.07338	57.46871159	52.8349
48.8	78.0	48.8	25.620155	77.08492101	68.2184419
58.4	93.4	58.4	9.958915(x)	109.8297(x)	82.91131(x)
62.4	99.7	62.4	x	x	x
62.4	99.7	62.4	x	x	x
62.4	99.7	62.4	x	x	x

그림 16은 15.3%의 트래픽을 발생시 나타나는 Jitter의 값의 변화를 보여주는 그래프이다.

그림 16 트래픽 15.3% 발생시 Jitter

그림 16에서 X축은 미디어 세션이 연결된 후에 미디어데이터를 받는 진행 시간을 나타냈고, Y축은 X축에 해당하는 로그된 미디어데이터 패킷의 Jitter값을 나타낸다.

트래픽이 없는 경우 평균 4.108681 (ms)의 Jitter 값을 확인 할 수 있었다. 그래프에서 보듯이 150초를 전후로 하여 네트워크 트래픽을 10% 증가시 평균 17.88282(ms) 의 Jitter 값을 확인 할 수 있었다.

그림 17은 15.3%의 트래픽을 발생시 나타나는 Round-Trip Delay의 값의 변화를 보여주는 그래프이다.

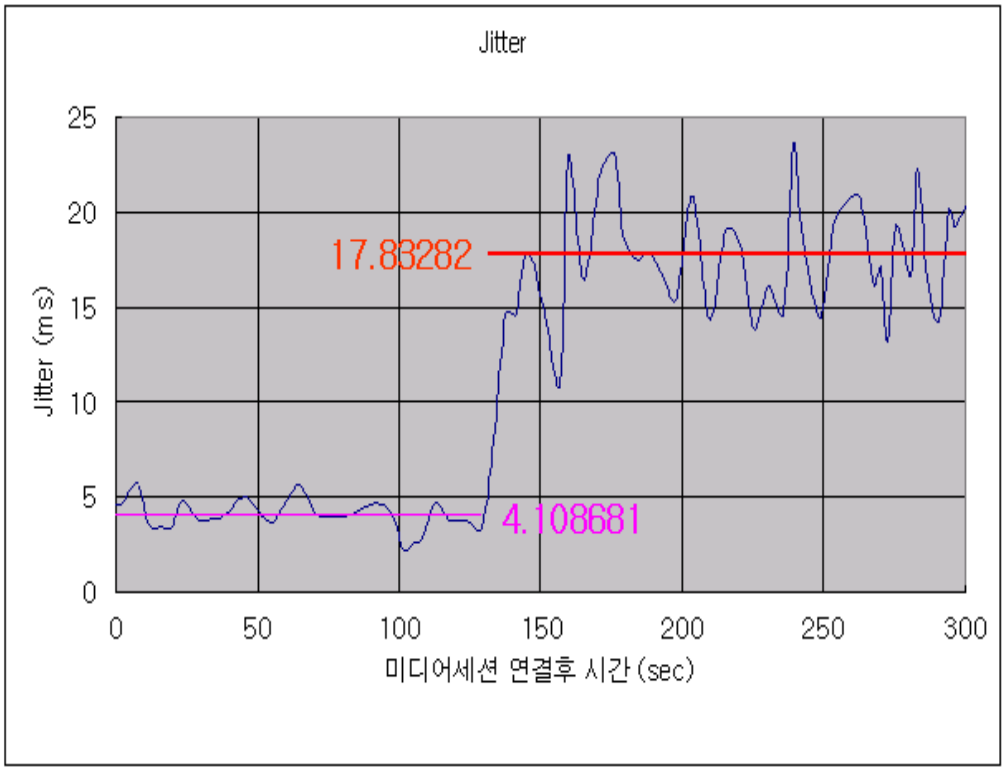


그림 17 트래픽 15.3% 발생시 Round-Trip Delay

그림 17에서 X축은 미디어세션이 연결된 후에 미디어데이터를 받는 진행 시간을 나타냈고, Y축은 X축에 해당하는 로그된 미디어데이터 패킷의 Round-Trip Delay값을 나타낸다.

트래픽이 발생하지 않을 경우 대부분 1ms 이내의 Round-Trip Delay를 보였으나 200초를 전후로 하여 15.3%의 트래픽을 증가 시 평균 15.91(ms)의 Round-Trip Delay를 값을 확인 할 수 있었다.

그림 18은 15.3%의 트래픽을 발생시 나타나는 Round-Trip Delay의 값의 변화를 보여주는 그래프이다.

그림 18 트래픽 15.3% 발생시 Packet Loss Rate

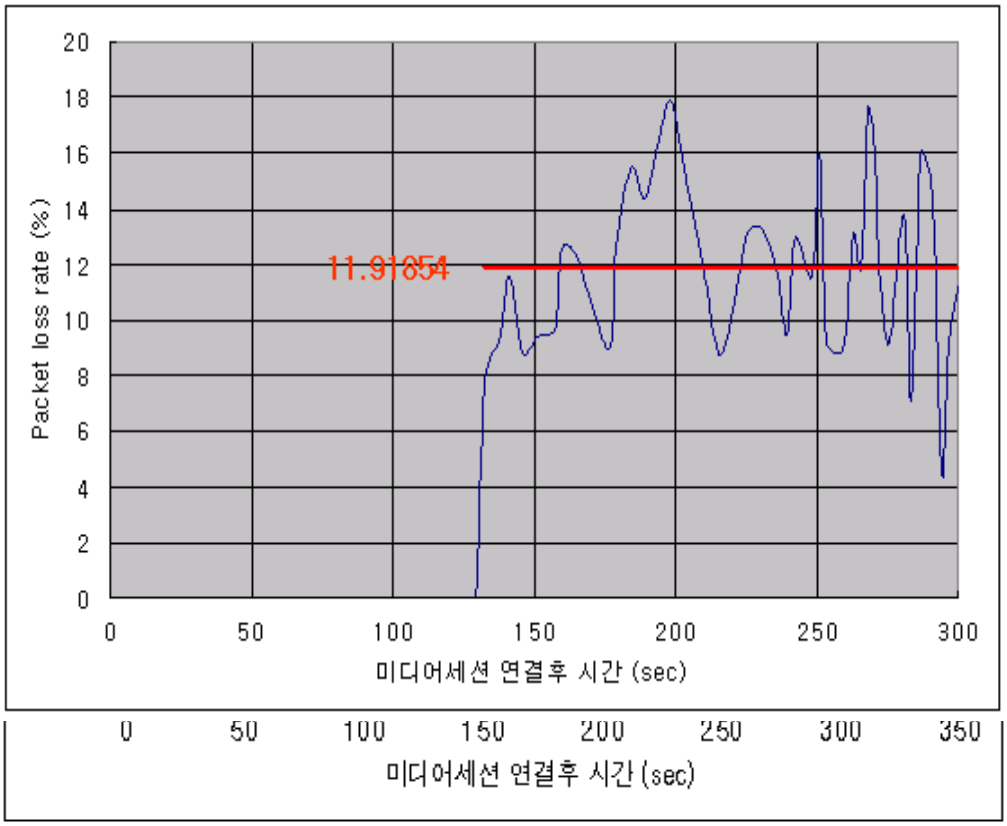


그림 18에서 X축은 미디어세션이 연결된 후에 미디어데이터를 받는 진행 시간을 나타냈고, Y축은 X축에 해당하는 로그된 미디어데이터 패킷의 Packet Loss Rate값을 나타낸다.

트래픽이 발생하지 않을 경우 패킷 손실이 0% 였으나 120초를 전후로 하여 15.3%의 네트워크 트래픽을 증가 하였더니 평균 11.9%의 손실을 보였다. 각 트래픽 증가율에 대한 그래프는 별첨을 참고하기 바란다. 각 평균값은 다음과 같은 그래프를 통해 결과 나타내었다.

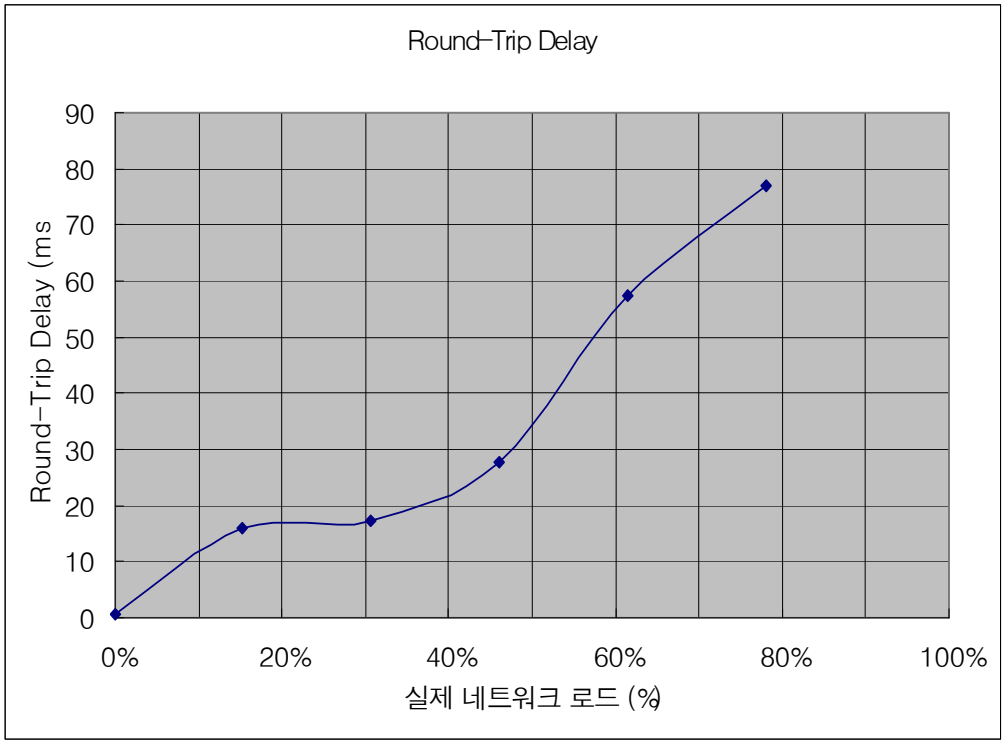


그림 19 100Mbps 데이터 전송을 지원하는 네트워크 환경에서 트래픽 증가에 따른 Round-Trip Delay

그림 19는 100Mbps 데이터 전송을 지원하는 네트워크 환경에서 트래픽 증가에 따른 그래프이다. Round-Trip Delay는 종단간의 Delay값을 측정한 수치로써 위 그래프에서 볼 수 있듯이 50% 정도의 트래픽 증가 시 낮은 비율의 증가 추세를 보이다가 50%이상의 트래픽 증가 시 급격하게 변화하는 것을 볼 수 있었다. 이로써 Round-Trip Delay는 트래픽이 증가하면 할수록 그 크기가 커진다는 것을 알 수 있다.

그림 20 100Mbps 데이터 전송을 지원하는 네트워크 환경에서 트래픽 증가에 따른 Jitter

그림 20은 100Mbps 데이터 전송을 지원하는 네트워크 환경에서 트래픽 증가에 따른 Jitter를 나타내는 그래프이다. Jitter는 RTP패킷의 도착 시간 편차를 측정한 수치로써 위 그래프에서 볼 수 있듯이 10%의 트래픽을 발생시켰을 경우 급격한 변화를 보이는 것으로 보아 트래픽에 상당하게 민감한 반응을 보이는 것을 알 수 있었다. 그 후 트래픽이 증가함에 따라 비슷한 증가 추이를 보였다.

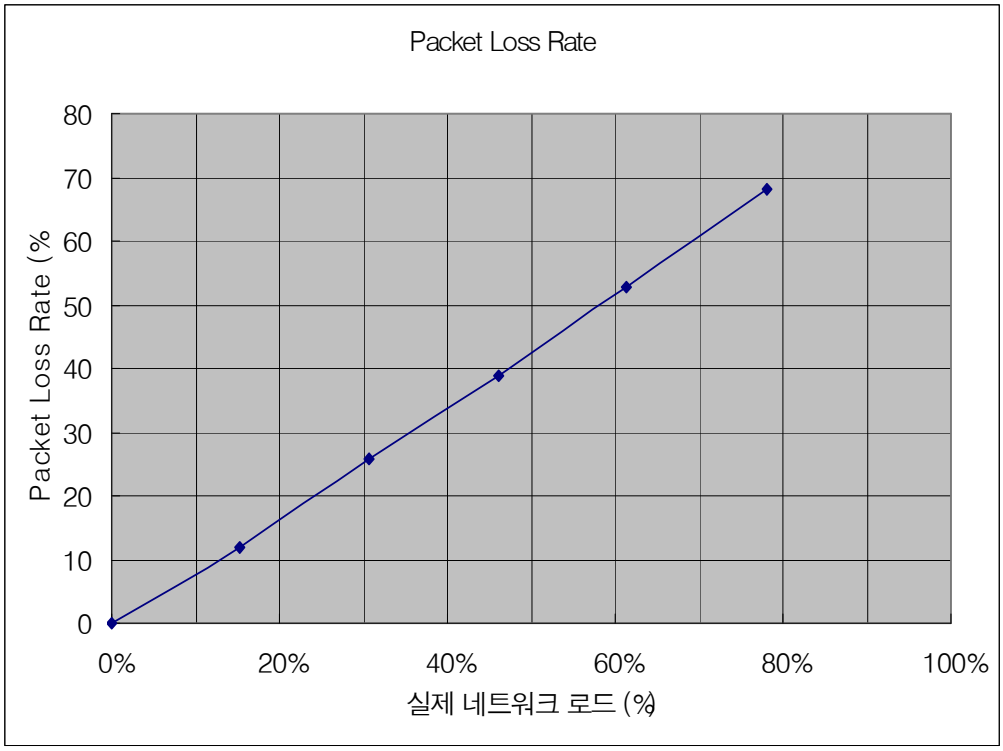


그림 21 100Mbps 데이터 전송을 지원하는 네트워크 환경에서 트래픽 증가에 따른 Packet Loss Rate

그림 20은 100Mbps 데이터 전송을 지원하는 네트워크 환경에서 트래픽 증가에 따른 Packet Loss Rate를 나타내는 그래프이다. Packet Loss Rate는 Receiver Report에서 다음 Receiver Report까지 보고 받은 RTP 패킷의 패킷 손실률을 측정한 수치로써 위 그래프에서 볼 수 있듯이 트래픽 증가 시 구간 간 일정한 증가 추이를 보였다. 그 이유는 패킷 손실은 앞에서 정의한 것과 같이 Jitter값이 시간편차가 큰 패킷은 다음 Receiver Report에서 패킷 손실로 간주 하여 실제 Receiver Report 에서 다음 Receiver Report까지의 RTP 데이터의 손실과 이전 Jitter에 의해 이전 Receiver Report에 보고 받지 못한 패킷이 추가되어 낮은 비율의 트래픽에서도 패킷 손실을 볼 수 있었다.

제 6장 결론 및 향후 연구

현재 실시간성과 서비스의 품질 보증을 요구하는 많은 응용 서비스들이 있다. 본 논문에서는 이들 응용프로그램 중 SIP 기반 실시간 멀티미디어 서비스의 성능을 분석 하였다. 무선 네트워크 환경에서 임베디드 단말기로 Sender와 Receiver의 QoS를 제어하기 위한 유선 네트워크에서 일차적인 실험을 하였다. 각각의 성능요인들을 다음과 같이 분석할 수 있었다.

첫째, Round-Trip Delay에서는 50% 정도의 트래픽 증가 시 낮은 비율의 증가 추세를 보이다가 50%이상의 트래픽 증가 시 급격하게 변화하는 것을 보였다.

둘째, Jitter 에서는 10%의 트래픽을 발생시켰을 경우 급격한 변화를 보이는 것으로 보아 트래픽에 상당하게 민감한 반응을 보였다.

셋째, Packet Loss Rate에서는 트래픽 증가 시 구간 간 일정한 증가 추이를 보였다. 이것은 트래픽의 증가로 인한 RTP 패킷의 지연이 커짐에 따라 예상했던 시간 이후 보다 많은 시간 편차를 보이며 그 다음에 도착하는 Receiver Report에 도착하여 이를 패킷 손실로 판단하게 되는 현상이 일어나게 된다. 이로 인해 패킷 손실률 또한 10%의 트래픽을 발생시켰을 경우 기존 연구와는 다른 그래프의 양상 즉, 높은 수치의 손실률을 보이는 것을 확인 할 수 있었다.

따라서, 각각의 성능 요인 그래프에서 알 수 있듯이 트래픽이 증가 할수록 각 성능 요인이 증가하는 것을 알 수 있으며 50%이상의 트래픽 증가 시 50%이하의 각 성능 요인의 증가 비율이 커지는 것을 알 수 있었다.

멀티미디어 통신 서비스에서 각 종단간의 멀티미디어 데이터가 원활한 데이터 통신을 하기 위해서는 이러한 수치를 보였을 경우 RTCP Report를 보고 Sender쪽에서는 프레임 전송률을 기존 초당 30프레임을 전송하는 것에서 초당 프레임을 30프레임보다 낮은 프레임 전송할 수 있도록 제어를

해야 한다. Receiver쪽에서는 버퍼링을 통해 이러한 현상들을 줄일 수 있다.

이 데이터를 바탕으로 임베디드 시스템에 SIPSET + MPEG4IP를 포팅시 Sender 또는 Receiver에서 QoS를 제어하는데 필요한 자료가 될 것이다. 현재 SIPSET 1.5에서는 RTCP 패킷 중 Sender Report Message 만이 구현이 되어 있다.

향후 Receiver Report Message를 구현함으로 써 Sender와 Receiver의 QoS를 효율적으로 제어하는데 필요하다. 또한, 향후 연구로 현재 나타나고 있는 Jitter의 값이 트래픽 포화시 증가추이를 보이는 것이 아니라 감소추이를 보이는 정확한 연구가 요구된다.

마지막으로 본 논문에서 분석한 SIPSET + MPEG4IP 시스템은 오디오와의 연동이 되지 않는 상태에서 비디오 데이터만으로 실험된 데이터로써 측정된 성능 요인으로 오디오기능을 추가하여 실제로 오디오와 비디오의 멀티미디어 데이터의 성능요인을 측정하는 연구가 요구된다.

參 考 文 獻

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, IETF, June 2002.

[2] M.handley et al., "SIP : Session Initiation Protocol," RFC 2543, Mar.1999.

[3] Henry Sinnreich, Alan B. Johnston, "Internet Communications Using SIP," Willey, 2001.

[4] Luan Dang, Cullen Jennings & David Kelly, "Practical VoIP Using VOCAL," O'REILLY.

[5] Jonathan Rosenberg, Bell Laboratories Jonathan Lennox and Henning Schulzrinne, "Programming Internet Telephony Services," IEEE Network, May/June 1999.

[6] H. M. Radha, M. V.D. Schaar, Y. Chen, "MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming Over IP," IEEE Transactions On Multimedia, Vol. 3, No. 1, Mar. 2001.

[7] H. Schulzrinne et al., "RTP : A Transport Protocol for Real-Time Applications," IETF, RFC 1889.

[8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF, RFC 3550, July 2003.

[9] VOVIDA SIPSET 1.5, http://www.vovida.org/.

[10] MPEG4IP 1.0, http://mpeg4ip.sourceforge.net/.

[11] M. Handley, V. Jacobson, "SDP: Session Description Protocol," RFC 2327, April 1998.

[12] "Session Initiation Protocol (SIP) Extension for Instant Messaging," IETF RFC 3428, Dec. 2002.

[13] ITU-T H.323, "Packet-based Multimedia Communications Systems,"

September 1999.
[14] 이재우, 황기태, 이재문, 김남윤, “SIP 기반 미디어 플레이어의 소프트웨어 시스템 분석“, 제 22회 한국정보처리학회 추계학술발표대회 논문집 제 11권 제 2호, p1465~p1468, 2004. 11.
[15] RFC 1889 - RTP: A Transport Protocol for Real-Time Applications.
[16] http://www.microsoft.com/download/details.asp/.
[17] T. Bray, J. Paoli, and C. M. Sperberg-McQueen, "Extensible markup language (XML) 1.0," W3C REC-xml-19980210, Feb. 1998; http://www.w3.org/TR/REC-xml/.
[18] http://www.ethereal.com/.
[19] http://wwww.sniffer.com/.
[20] http://www.apache.org/xalan-j/.
[21] O.Festor and A.Pras, □□A Methodology for Performance Analysis of Real-Time Continuous Media Applications,□□12th International Workshop on Distributed Systems: Operations and Management DSOM□□2001 Nancy France, October 15-17, 2001.
[22] http://www.netpredict.com, "Performance Analysis for Video Streams across Networks", White Paper, December 2003.
[23] J. Rosenberg, "The Extensible Markup Language(XML) Configuration Access Protocol(XCAP)," draft-ietf-simple-xcap-03.txt, July 2004.
[24] J. Rosenberg, "A Session Initiation Protocol(SIP) Event Package for Modification Events for the Extensible Markup Language(XML) Configuration Access Protocol(XCAP) Managed Documents,"draft-ietf-simple-xcap-package-01, Feb. 2004.
[25] 손우사, 강창석, “ 예제로 배우는 XSLT", 인포북.
[26] http://www.iana.org/assignments/rtp-parameters, RTP Parameters.
[27] Colin Perkins, "RTP : audio and video for the Internet," Addison-Wesley.
[28] RFC 2032 - RTP Payload Format for H.261 Video Streams.

Abstract

An analysis of performance parameters the SIP-Based real-time communication

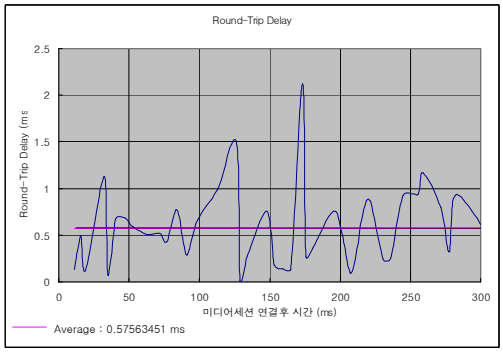
Lee, Jae-woo
Department of Computer Engineering
Graduate School, Hansung University

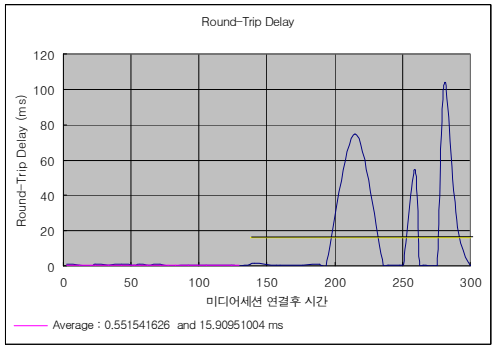
As the Internet environment grows more, application programs based on the SIP(Session Initiation Protocol) are becoming more popular in real life.

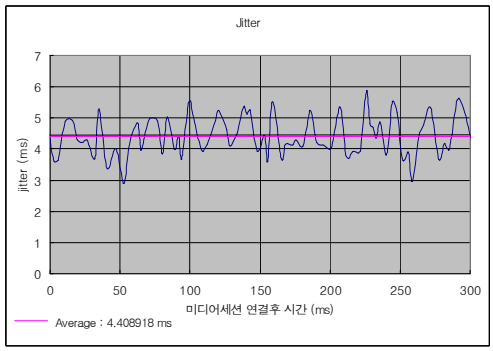
First, This study analyzed a system structure of the SIP-based real-time player which is the hot topic in the Internet. And then specifically, this thesis analyzed the architecture of the SIP system developed by the VOVIDA corporation, and then constructed an experimental environment that is equipped a router and a traffic generator between the SIP devices, which run a SIP-based real-time player based on SIPSET 1.5 and MPEG4IP 1.0 and a windows messenger respectively. Finally, this thesis shows measurements of the performance parameters that are the End-to-end Delay, Jitter and Packet Loss Rate.

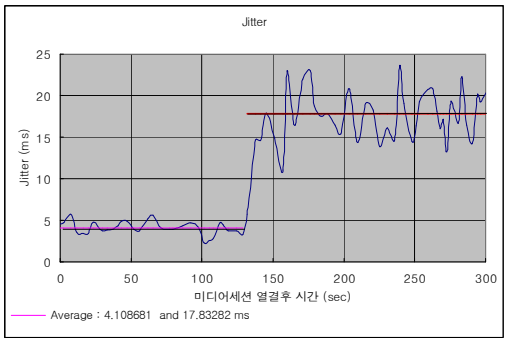
This thesis is expected as an important data the design of multimedia applications on the embedded system and QoS control mechanisms.

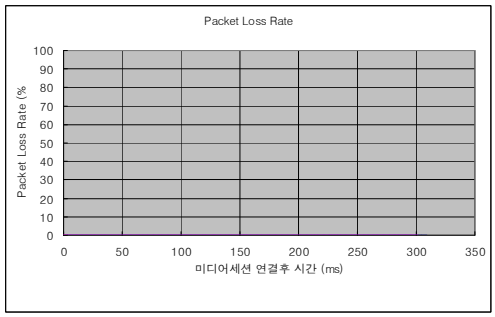
별 첨

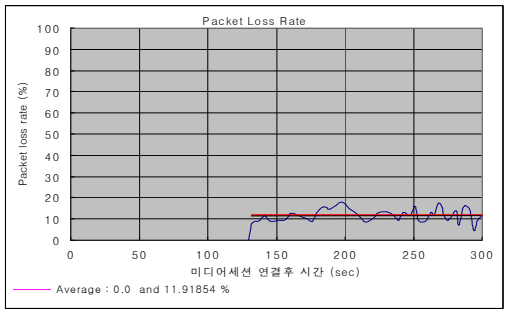






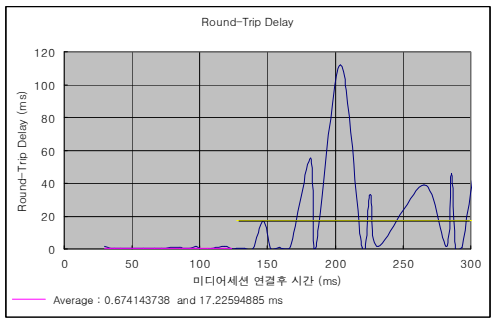


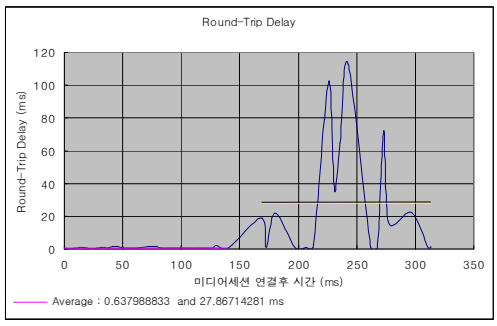




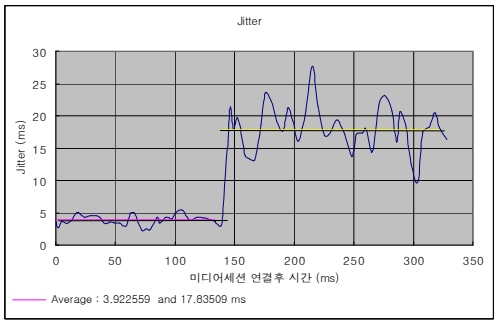
< 네트워크 로드 0% >

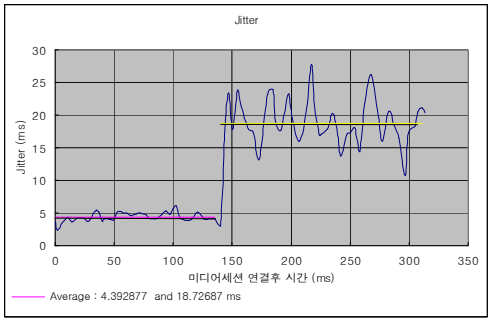
< 네트워크 로드 15.3% >

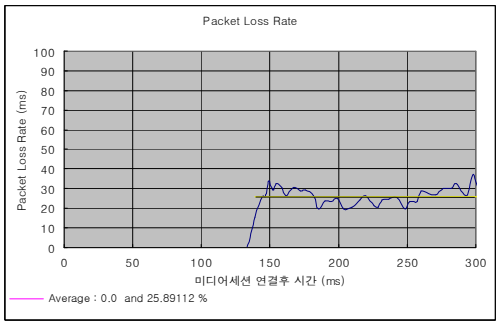


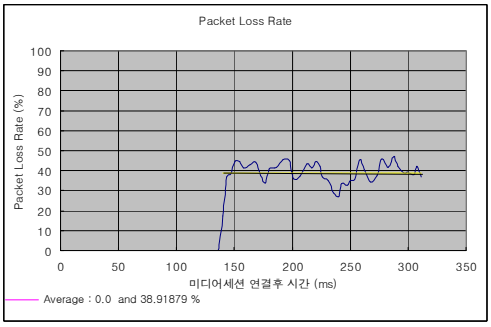


별 첨



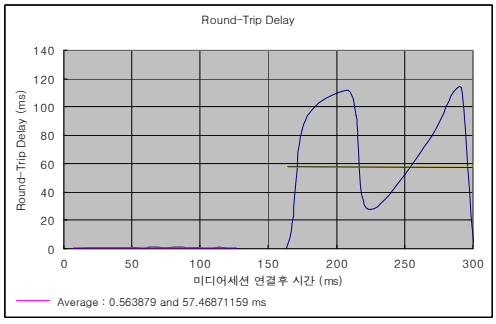


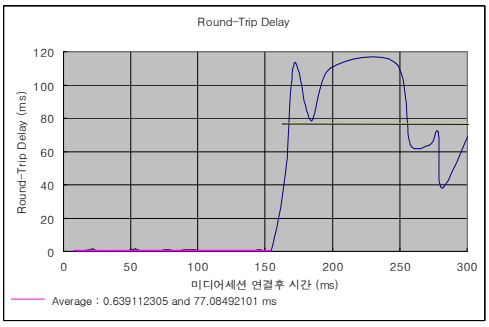


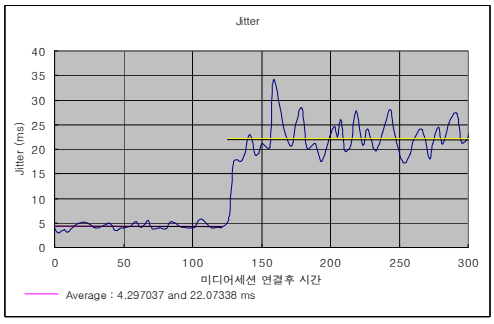


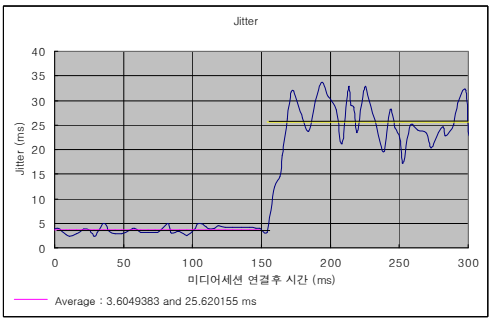
< 네트워크 로드 30.7% >

< 네트워크 로드 46.0% >

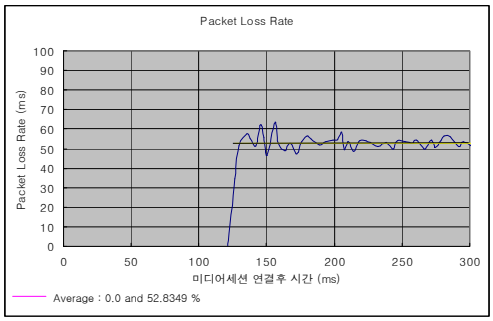




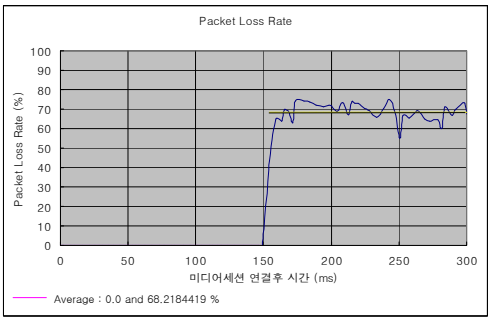




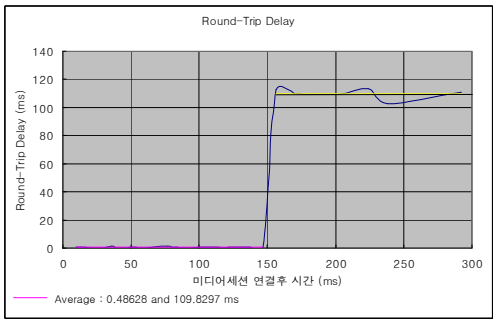
 # 별 첨

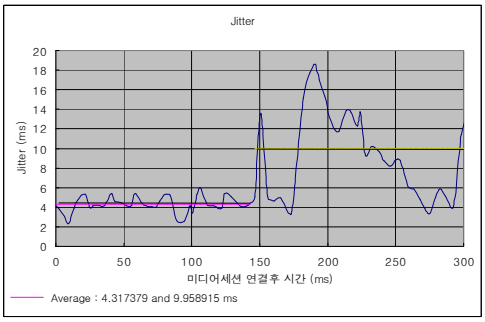


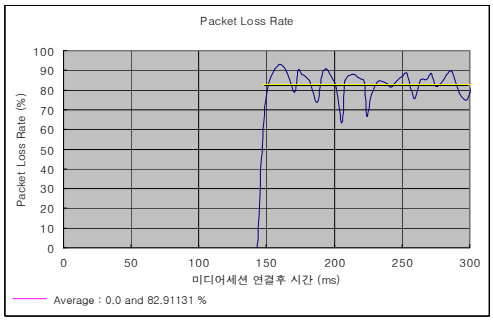
< 네트워크 로드 61.4 %>



< 네트워크 로드 78.0%>







< 네트워크 로드 93.4% >