

실시간 멀티미디어 스트림을 위한 분산 전송 플랫폼

(The Distributed Transport Platform for Real-Time Multimedia Stream)

송 병 훈 [†] 정 광 수 [‡] 정 형 석 ^{***}

(Byunghun Song) (Kwangsoo Chung) (Hyungseok Chung)

요약 기존의 분산 객체 미들웨어가 제공하는 RPC(Remote Procedure Call) 기반의 메시지 전송 방식은 내부 오버헤드로 인해 실시간성을 요구하는 스트림 데이터의 전송에는 부적합하다. 그래서 OMG (Object Management Group)에서는 분산 환경에서도 실시간 스트리밍을 지원 할 수 있는 새로운 AV (Audio and Video) 스트림 서비스 참조 모델을 제안하였다. 그러나 이 모델은 오직 참조 모델이기 때문에 실제 구현 방법에 따라 재 정의해야 할 부분이 많이 남아 있다. 특히 실제 네트워크 환경을 고려한 혼잡 제어와 같은 기능이 없기 때문에 스트림 전송의 QoS를 제어할 수가 없다. 이러한 고려는 최근 연구되어 지고 있는 다양한 스트림 전송 플랫폼들이 가져야 할 고급 기능으로 전체 네트워크의 효율 증가를 위해서도 반드시 필요하다. 본 논문에서는 분산 환경의 장점을 최대한 부각 시킬 수 있는 OMG의 스트림 서비스 참조 모델을 재 정의한 스트림 전송 플랫폼을 설계하고 구현하였다. 제안한 플랫폼은 스트림 전송을 위한 새로운 TCP-Friendly 프로토콜인 SRTP(Smart RTP)를 하부 구성 요소로 제공하며 이를 이용하여 본 플랫폼을 통한 스트리밍 응용들의 효율을 향상 시켰다.

키워드 : 미들웨어, 스트리밍, CORBA, TCP-Friendly

Abstract The traditional distributed object middleware platform is not suitable for the transmission of stream data, because RPC (Remote Procedure Call)-based message transmission have a great overhead. Therefore, the OMG (Object Management Group) proposes the AV (Audio and Video) stream reference model for streaming on the distributed object middleware platform. But, this reference model has not a detail of implementation. Particularly it also has not congestion control scheme for improvement of network efficiency on the real network environment. It is a very important and difficult technical issue to provide the stream transmission platform with advanced congestion control scheme. In this paper, we propose an architecture of a distributed stream transport platform and deal with the design and implementation concept of our proposed platform. Also, we present a mechanism to improve streaming utilization by SRTP (Smart RTP). SRTP is our proposed TCP-Friendly scheme.

Key words : middleware, streaming, CORBA, TCP-Friendly

1. 서 론

· 본 연구는 한국 과학재단 목격기초연구[R01-2002-000-00179-0(2002)] 지원으로 수행되었음

† 비회원 : 광운대학교 전자통신학과
byungh@adams.kwangwoon.ac.kr

‡ 종실회원 : 광운대학교 전자통신공학과 교수
kchung@daisy.kwangwoon.ac.kr

*** 비회원 : 한국전자통신연구원 연구원
chunghs@etri.re.kr

논문접수 : 2002년 3월 20일

심사완료 : 2002년 12월 31일

컴퓨터 처리기술, 멀티미디어 부호화 기술의 급속한 발전으로 인하여 새로운 형태의 서비스들이 많이 등장하고 있다. 또한, 네트워크의 고속화가 급속도로 발전함에 따라 주문형 비디오와 같은 새로운 유형의 멀티미디어 서비스에 대한 요구가 빠르게 늘어나고 있다. 특히, 무엇보다도 ISP(Internet Service Provider)의 서비스가 대형화 분산화 되어 가는 추세에서 대규모 가입자를 동시에 서비스 할 수 있는 멀티미디어 스트리밍 플랫폼에 관한 연구가 더욱 중요하게 인식되고 있다[1]. 스트리밍

서비스의 대형화를 위해선 어떠한 것들을 추가로 고려하여야 하는가? 현재 진행되고 있는 많은 관련 연구들을 종합하여 보면, 다음과 같은 두 가지 요인들에 주목할 수 있다.

첫째로 스트리밍 프레임워크에 대한 고려이다. 스트리밍 프레임워크는 멀티미디어 응용들이 스트리밍 데이터의 전송 경로를 효율적으로 설정하고 제어할 수 있는 메커니즘을 제공하는 일종의 미들웨어 플랫폼이다. 이를 통해 송신자와 수신자는 쉽게 스트리밍 데이터의 송수신 경로를 설정할 수 있으며 다른 하드웨어 플랫폼, 운영체제, 링 환경 하에서도 상호 운용성을 보장받을 수 있게 된다. 현재 연구 또는 개발중인 대표적인 스트리밍 프레임워크로는 ISO/IEC JTC1 SC29/WG11 MPEG 그룹에서 정의한 DSM-CC(Digital Storage Media - Command and Control)와 분산 객체 관리 구조에 대한 표준을 정의하고 있는 OMG(Object Management Group)의 CORBA (Common Object Request Broker Architecture) AV 스트리밍 서비스(Audio Video Streaming Service) 모델 등이 있다[2,3,4,5]. 특히 본 논문에서는 대표적인 산업계 분산 환경 표준이면서 확장성이 매우 뛰어난 CORBA를 이용하는 OMG의 AV 스트리밍 모델을 기반으로 새로운 스트리밍 전송 플랫폼을 제안하고자 한다.

다음으로 고려할 사항은 현재의 인터넷 통신 환경과 조화 할 수 있는 스트리밍 전송의 혼잡제어 방법에 관한 것이다. 현재의 인터넷은 최선형의 서비스만을 제공하고 있기 때문에 혼잡 상황으로 인한 패킷 손실이 빈번하게 일어난다. 즉, 어떠한 트래픽에 대해서도 QoS(Quality of Service)를 보장 할 수 없다. 이러한 환경 속에서, 기존의 AV 스트리밍 서비스들은 실시간성 데이터라는 전송 미디어의 특성과 적은 프로토콜 오버헤드를 위해 UDP와 같은 전송 프로토콜을 많이 사용하고 있다. 그러나 네트워크의 상황을 고려할 수 없는 UDP 기반의 스트리밍은 오히려 네트워크의 혼잡 상황시 경쟁하는 다른 정상적인 TCP 연결들의 전송률을 잠식하는 "불공정성 문제"를 야기시킬 수 있다. 이러한 현상의 근본적인 원인은 UDP는 TCP와 같은 혼잡 제어 방법이 없기 때문이다. 전역적인 망에서 스트리밍 서비스를 성공적으로 실현하기 위해서는 현재의 인터넷의 주요한 트래픽인 TCP와 스트리밍 트래픽이 잘 조화 될 수 있어야 한다. 그러므로 같은 네트워크 환경에서 경쟁하는 TCP 기반의 많은 인터넷 응용들과의 공평한 대역폭 분배는 성공적인 스트리밍 전송 플랫폼을 평가하는데 매우 중요한 요인으로 작용한다[6].

본 논문에서는 대역폭 분배의 불균형성 문제를 해결

할 수 있는 새로운 스트리밍 전송 플랫폼을 제안하고 설계하였다. 제안한 플랫폼은 스트리밍을 위한 기본적인 연결 설정 및 승인제어, 관리 기능을 상호 다른 시스템 환경에서 투명하게 제공하며 제안한 프로토콜인 SRTP(Smart RTP)를 이용하여 전송률을 증대 효과를 가능하게 한다.

본 논문은 총 5장으로 기술되어 있다. 먼저 2장에서는 본 연구의 배경 및 관련 연구에 관해 전반적으로 기술하였고, 3장에서는 본 논문에서 제안한 실시간 스트리밍 전송 플랫폼에 대해 기술하였다. 4장에서는 스트리밍 전송 플랫폼의 구현 및 성능 평가에 대해 기술하였으며 마지막으로 5장에서는 결론 및 향후 연구 과제에 대해 기술하였다.

2. 배경 및 관련 연구

대형화, 분산화되어 가고 있는 컴퓨팅 환경에서 이기 중간의 새로운 서비스를 효과적으로 제공하기 위해 개발된 것이 미들웨어 기술이다. 그 중에서도 대표적인 산업계 표준인 CORBA는 새로운 서비스를 디자인 할 때 확장성과 관리측면에서 매우 탁월한 장점을 지닌다. 본 장에서는 우선 이러한 CORBA 미들웨어의 기본적 구성과 스트리밍 서비스를 고려할 때 발생하는 문제점, 그리고 이것을 보완한 최근 연구에 대해 알아 보도록 한다. 또한 본 논문에서 스트리밍 전송 플랫폼의 중요한 기능으로 주목되고 있는 혼잡 제어 방법에 대해 살펴 본다.

2.1 CORBA 미들웨어 구성

CORBA는 분산 컴퓨팅 기술과 객체 지향 기술을 접목시킨 분산 객체 컴퓨팅의 대표적인 산업계 표준으로서 OMG라는 비영리 단체에 의해 다양한 표준화가 진행되고 있다. CORBA 미들웨어 환경에서 클라이언트 객체는 서버 객체의 위치, 구현된 언어, 운영 체제, 하드웨어 플랫폼, 통신 프로토콜 및 네트워크 등에 대한 사전 지식 없이도 서버 객체에게 오퍼레이션의 수행을 요구할 수 있다. 이러한 투명성을 보장하기 위한 CORBA는 복잡한 미들웨어의 구조를 가진다[7,8].

2.2 분산 객체 미들웨어의 문제점과 해결 방안

그림 1과 같은 CORBA를 포함한 분산 객체 미들웨어의 요구/응답 시맨틱(semantic)은 시간 제약적인 특성을 갖는 오디오, 비디오 스트리밍 데이터의 전송에는 부적합하다. 그러나 분산 객체 미들웨어는 뛰어난 이식성, 유연성 및 재사용성을 제공한다는 장점을 갖고 있기 때문에, 시간 제약성을 갖는 스트리밍 데이터의 전송에 대한 문제만 보완한다면 이상적인 스트리밍 전송 플랫폼으로 활용이 가능할 것이다.

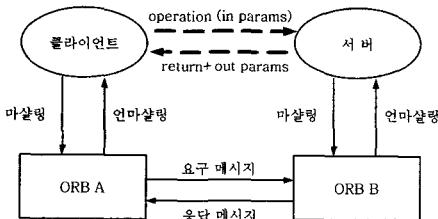


그림 1 분산 객체 미들웨어의 요구/응답 시맨틱

최근 OMG에서는 이러한 문제점을 해결하고자 스트리밍 데이터 제어 정보는 요구/응답 시맨틱을 따라 교환되며 실제 데이터 전송은 별도의 전송 경로를 통해 교환하는 즉, 제어 경로와 전송 경로를 별도로 정의하고 관리하는 방법이 제안되었으며, 이를 통해 다양한 관련 연구가 활발히 진행되고 있다[9].

2.3 OMG AV 스트리밍 서비스 플랫폼

기존의 분산 객체 미들웨어의 요구/응답 시맨틱의 문제를 보완하여 스트리밍 데이터도 시간적 제약을 극복하면서 효과적으로 전송할 수 있도록 연구하는 대표적인 스트리밍 프레임워크로는 OMG에서 연구, 발표된 AV 스트리밍 서비스 플랫폼을 들 수 있다[2]. 그림 2는 OMG AV 스트리밍 서비스 참조 모델에서 정의한, 별개의 제어 경로와 전송 경로를 갖는 분산 스트리밍 참조 모델을 도시한 것이다.

스트리밍 참조 모델은 크게 세 부분으로 나뉘어진다. 컨트롤/매니저 객체는 스트리밍 종단 객체 및 이들 간의 스트리밍을 생성, 유지, 삭제하는 역할을 담당하는 객체이다. 객체 어댑터는 서버측 CORBA 미들웨어의 구성 요소를 나타내며, 스트리밍 서비스를 위해 별도의 변경이 필요치 않은 부분이다. 스트리밍 종단은 내부적으로 세 부분으로 다시 나뉘어 진다. OMG AV 스트리밍 서비스 사양에서는 컨트롤/매니저 객체와 스트리밍 인터페이스 컨트롤 객체에 대한 표준 인터페이스를 제공하며 그림 2의 1번과 2번 인터페이스가 여기에 해당한다. 반면 본

참조 모델은 3번과 4번 인터페이스에 대해서 정의하고 있지 않는데 이는 구현 이슈로 여러 기능이 있을 수 있다. OMG의 스트리밍 참조 모델은 새로운 스트리밍 플랫폼을 디자인할 때 기본적인 참조 모델로 사용할 수 있다. 그리고 이렇게 디자인된 새로운 플랫폼은 기존의 CORBA 미들웨어가 가진 여러 분산 환경을 위한 장점을 그대로 계승 할 수 있는 장점을 가진다.

2.4 TCP-Friendly 혼잡 제어 기법

서론에서 언급한 UDP 기반의 스트리밍 전송 프로토콜들이 인터넷의 주요 전송 프로토콜인 TCP와 불공정성 문제를 야기 시키는 현상은 기존의 스트리밍 전송 프로토콜들이 해결해야 할 대표적인 문제점이며 이러한 문제에 대한 해결은 전체 네트워크의 효율을 높이는데 크게 기여 할 수 있다. 특히 TCP는 현재의 인터넷에서의 주요 트래픽이기 때문에 이러한 문제는 반드시 해결되어야 한다.

TCP-Friendly 혼잡 제어는 TCP가 아닌 프로토콜이 TCP와의 불균형성을 종단간에서 해결하여 보려는 연구들 중에 가장 대표적인 방법이다[6]. TCP-Friendly란 같은 연결, 상황에서 TCP가 아닌 연결의 평균 전송률이 결국 TCP의 평균 전송률과 유사하게 분포하는 성질을 의미한다. 특히 TCP-Friendly의 정도를 나타내는 공정성 지수 F를 사용하여 이 성질을 어느 정도 만족하는지를 나타낼 수 있다. 아래 식 (1)은 공정성 지수 F의 산출 방법이다. TTCP는 TCP 연결의 실제 전송률(goodput)을 나타낸다.

$$F = \frac{T_{\text{제안한프로토콜}}}{T_{\text{TCP}}} \quad (1)$$

TCP-Friendly 혼잡 제어 방법은 크게 전송 방법의 지원 유형을 기준으로 단일과 다중 전송 방법으로 구분되며 각 유형들은 다시 전송률과 윈도우 기반의 혼잡 제어 방식으로 나뉘어 진다. 특히 AV 스트리밍 전송과 같은 응용들을 위한 연구로는 전송률 기반의 TCP-Friendly 혼잡 제어 방법이 많이 연구되어지고 있다. 그 이유는 이 방법이 다른 방법들에 비해 실시간성과 연속적인 특성을 갖는 AV 스트리밍의 전송률 제어 방법으로 적합하기 때문이다.

전송을 기반의 TCP-Friendly 성질을 만족하면서 전송률(T)을 산출하는 방법은 방정식 (2)에 기초한다[10]. 여기서 t_{RTT} 는 종단간 지연시간을 t_{RTO} 는 재전송 시간을 그리고 s 는 세그먼트의 크기, b 는 각각의 ACK를 받은 ACK 패킷의 수, p 는 패킷의 손실률을 나타낸다.

$$T = \frac{s}{t_{RTT} \sqrt{\frac{2bp}{3} + t_{RTO} \min\left(3\sqrt{\frac{3bp}{8}}, p(1+32p^2)\right)}} \quad (2)$$

이 방정식은 그림 3과 같이 Reno-TCP의 네트워크 상

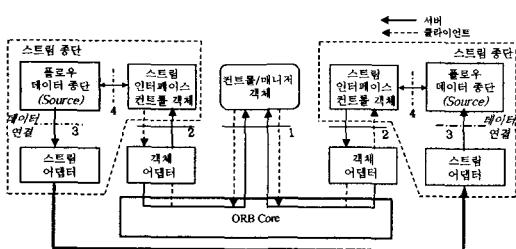


그림 2 OMG 스트리밍 참조 모델

태에 따른 TD(Triple-Duplicate ACK)와 TO(Time-Out) 이벤트에 근거한 TCP 윈도우 크기의 변화를 수학적으로 모델링한 것이다. 따라서 S1 구간에서의 TCP의 전송률은 방정식 (3)과 같이 정의 할 수 있다[10].

$$S1 = Z1 + Z2 = \frac{Z1\text{구간의 페킷수} + Z2\text{구간의 페킷수}}{Z1\text{시간} + Z2\text{시간}} \quad (3)$$

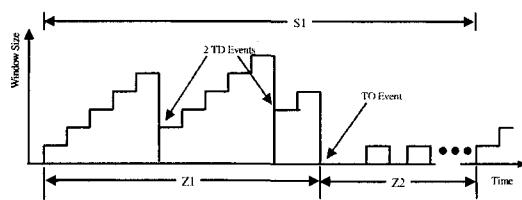


그림 3 TD와 TO를 고려한 TCP의 윈도우 크기의 변화 모델

3. 실시간 스트림 전송 플랫폼의 설계

지금까지 고찰해 본 바와 같이 OMG AV 스트리밍 서비스 플랫폼은 기존의 스트림 전송 플랫폼이 제공하지 못했던 하드웨어 플랫폼에 독립적인 분산 미들웨어 환경의 장점을 그대로 수용할 수 있다. 이러한 장점은 본 플랫폼을 다양한 형태의 통신망과 시스템에 쉽게 적용할 수 있게 만든다. 그러나 OMG AV 스트리밍 서비스 플랫폼은 스트림 데이터의 전송 및 제어를 위한 플로우 설정/해제, 미디어 디바이스 선택/변경, 멀티캐스트/유니캐스트 전송 메커니즘 등을 제안하고 있지만, 효율적인 실시간 데이터 전송을 위해 필수적인 네트워크 자원의 승인 제어 기법과 빈번한 네트워크 혼잡 상황을 고려한 스트림 혼잡 제어 방법에 대해서는 고려하고 있지 않다.

본 논문에서 제안한 실시간 스트림 전송 플랫폼은 OMG AV 스트리밍 서비스에서 정의된 기본적인 장점을 수용하면서 추가로 멀티미디어 스트림을 위한 새로운 공통 서비스들을 지원한다. 다음은 제안한 실시간 스트림 전송 플랫폼이 제공하는 공통 서비스들이다. a), b) 공통 서비스는 원래 OMG 스트리밍 참조 모델이 제공하는 기본 기능이며 c), d) 기능은 본 논문에서 재 정의한 실시간 스트리밍 플랫폼이 지원하는 새로운 기능이다.

- a) 스트림 플로우 설정 및 해제 기능
- b) 서비스 품질 협상 및 예약 기능
- c) 네트워크의 트래픽 상황에 따른 승인 제어 기능
- d) SRTP: TCP-Friendly 혼잡 제어 기능

3.1 실시간 스트림 전송 플랫폼의 구성

그림 4는 제안한 실시간 스트림 전송 플랫폼의 전체 구성을 나타낸다. 앞서 언급한 것과 같이 제안한 실시간 스트리밍 플랫폼은 기존의 OMG 스트리밍 참조 모델에 기반한 플랫폼으로 기본적으로 스트림 컨트롤러, 리소스 핸들러, 스트림 핸들러와 같은 부분은 표준 참조 모델을 따르며 스트림 어댑터는 그 기능을 재 정의하여 TCP-Friendly한 혼잡 제어를 지원하도록 확장하였다. 마지막으로 리소스 매니저는 본 플랫폼을 위해 새롭게 추가한 부분으로 현재 가능한 자원을 바탕으로 새로운 스트림의 생성에 대한 승인 제어 기능을 수행함으로써 예약된 스트림을 보호하는 기능을 수행한다.

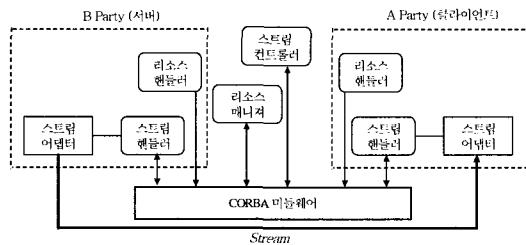


그림 4 실시간 스트림 전송 플랫폼 구조

3.2 스트림 컨트롤러

본 플랫폼을 이용하는 응용은 스트림 컨트롤러가 제공하는 API를 통해 클라이언트와 서버간 스트림을 생성, 해제하고 데이터 전송을 시작, 중지시킬 수 있다. 클라이언트로부터 스트림 생성 요구가 발생되면 스트림 컨트롤러는 양측의 리소스 핸들러를 결합하여 클라이언트와 서버를 연계시키고, 추가적인 스트림 핸들러 및 스트림 어댑터의 생성을 유발한다.

3.3 리소스 매니저

리소스 매니저는 네트워크 자원의 효율적인 관리를 위해 스트림 서비스에 대한 승인제어 기능을 담당한다. 하나의 도메인 내에서 분산 AV 스트리밍 전송 플랫폼을 사용하는 종단간의 연결에서 최대 복목 지점의 대역폭을 B_{bn} 이라 가정하면 이 값은 스트림 종단간의 최대 사용 가능한 대역폭이 된다고 볼 수 있다. 그러므로 B_{bn} 의 값이 0으로 가까워질수록 종단간의 사용 가능한 대역폭이 줄어들게 된다. 이러한 상황이 계속되면 결국 네트워크의 혼잡 상황이 가중되며 되어, 기존에 서비스를 잘 받고 있는 스트림 연결들도 전체적인 전송률을 낮출 수 밖에 없게 되거나, 최악의 경우에는 서비스의 단절도 일어나게 된다. 실시간 스트리밍 플랫폼의 리소스 매니저는 이러한 상황을 사전에 방지하는 기능을 제공한다.

리소스 매니저는 주기적으로 네트워크 상태 측정 패킷을 종단간에 전송하여 B_{bn} 의 값을 구한다. 측정은 패킷 쌍(Packet Pair) 방법을 이용한다. 이 방법은 서버쪽 플랫폼에서 연속하는 두 패킷 쌍을 보낼 때 클라이언트 플랫폼에 도달하는 시간차를 이용하는 방법으로 방정식 (4)와 같은 식을 기초로 하여 B_{bn} 의 값을 구한다[11]. S_2 는 두 번째 패킷의 크기를 나타내며, t_1 , t_2 는 첫 번째와 두 번째 패킷의 도착 시간을 나타낸다.

$$B_{bn} = \frac{S_2}{t_2 - t_1} \quad (4)$$

실시간 스트림 플랫폼을 이용하는 새로운 종단간 서비스 연결을 R_{new} 라고 할 때 리소스 매니저는 그림 5와 같은 승인 제어 알고리즘을 기준으로 승인 제어를 수행한다.

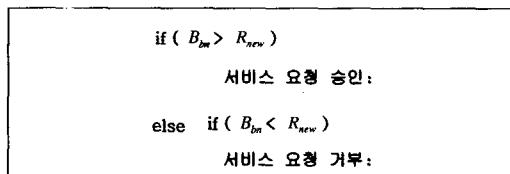


그림 5 리소스 매니저의 승인 제어 알고리즘

3.4 리소스 핸들러

리소스 핸들러는 양단간에 하나씩 존재하며 스트림 설정을 위한 연결 협상 기능을 담당한다. 스트림 컨트롤러로부터 스트림 생성 요구 발생시 전송 플랫폼의 승인 제어를 위해 리소스 매니저 정보를 교환하며, 승인이 성공적으로 이루어진 경우 스트림 핸들러를 기동시켜 실제적인 스트림 생성 과정을 지시한다. 그림 6은 양단

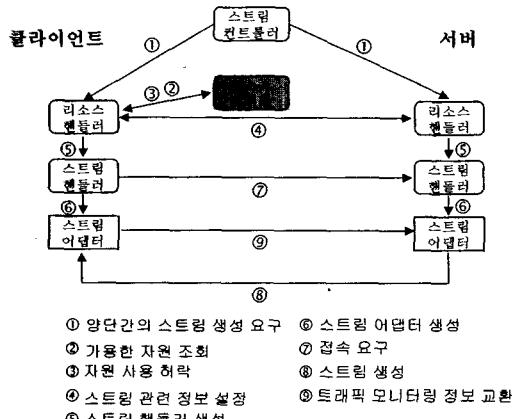


그림 6 스트림 생성 절차

간 스트림 생성을 위한 절차를 도시한 것이다.

3.5 스트림 핸들러

스트림 핸들러는 양단간 스트림 생성 및 제어 기능을 담당하는 모듈로서 스트림 서비스의 접속/해제, 시작/종료 등의 스트림 제어 기능을 제공한다.

3.6 스트림 어댑터

본 논문에서 구현한 스트림 어댑터는 기존의 TCP 기반의 많은 인터넷 응용들과의 전송률 조화를 보이면서도 전체적인 스트림 전송률의 평활화 특성을 유도하는 TCP-Friendly 혼잡 제어를 지원한다. 특별히 제안한 전송률 기반의 TCP-Friendly 혼잡 제어 프로토콜을 SRTP(Smart RTP)라 부른다. 이 프로토콜은 IETF(Internet Engineering Task Force)에서 권고하는 실시간 AV 전송 프로토콜의 표준인 RTP(Real-time Transport Protocol)를 기반으로 그 기본 기능을 확장하여 새로운 혼잡 제어 기능을 추가한 프로토콜이다[12].

RTP는 기본적으로 UDP 기반의 전송 프로토콜이면서도 AV 스트림 전송에 최적화된 전송 및 기본적인 피드백 정보 처리 기법을 제공한다. 그러나 이러한 RTP 역시 앞서 언급한 UDP와 같은 대역폭 공정성 문제를 근본적으로 해결하지는 못한다. SRTP는 이러한 문제를 해결하면서 실시간 스트림 전송 플랫폼에 결과적으로 "AV 스트리밍을 위한 TCP-Friendliness 혼잡 제어" 기능과 "완만한 전송률(rate smoothness) 평활화" 기능을 제공한다. 전체적인 SRTP의 동작 과정은 다음과 같다.

(i) RTP의 제어 프로토콜인 RTCP의 SR(Sender Report), RR(Receiver Report) 속의 패킷 정보를 기반으로 식 (5), (6)을 통해 네트워크의 손실률 p 와 지연 값으로 대응되는 t_{RTT} 값을 구한다. 특히 식 (6)은 t_{RTT} 값의 변화의 민감성을 고려한 방정식으로 본 논문에서는 임계 값($\alpha = 0.8$)을 전통적인 TCP의 상수에 비해 낮게 설정한다. 이러한 설정 이유는 지연 값을 예측하는데 있어서 현재의 지연 정보($\overline{t_{RTT}}$)에 대한 비중을 높이기 위해서다. 이렇게 구한 p , t_{RTT} 값은 식 (2)에 대입되어 최대 전송 가능한 TCP-Friendly 스트림의 전송률(T)을 구하는데 사용한다. SRTP는 이렇게 얻은 T 값을 기준으로 초기 전송률을 결정한다. N_{first} 와 N_{max} 는 전송 받은 패킷의 최초, 최대 번호이며, N_{real} 은 실제 전송받은 패킷의 수이다. 또한, T_j 는 마지막 수신한 SR의 타임 스텝프를 나타내며, T_i 는 RR을 수신받고 전송하는데 걸린 시간의 차를 의미한다. t 는 RTCP의 RR에 의해 알 수 있는 패킷의 도착 시간을 나타낸다.

$$p = \frac{N_{real}}{(N_{max} - N_{first})} \quad (5)$$

$$t_{RTT} = (\alpha \cdot \overline{t_{RTT}}) + (1 - \alpha)(t - T_i - T_j) \quad (6)$$

(ii) 혼잡 상황 시 스트림 어댑터는 다음과 같은 SRTP의 혼잡 제어 방법을 이용하여 동적으로 전송률을 변화시켜 TCP와 비슷한 방법으로 네트워크를 안정 상태로 유도한다. 그림 7은 제안한 SRTP가 판단하는 패킷 손실률 p 에 따른 네트워크 상태이며 결론적으로 혼잡, 비 혼잡 상태에 따라 스트림의 전송률을 증가, 감소시키게 된다. 특히 네트워크 상태에 따른 SRTP 스트림의 패킷 손실률 p 와 자연 값 t_{RTT} 은 경쟁하는 TCP 연결들의 동일한 상태에서의 손실 및 지연 정보 값과 유사하므로 이 값에 근간한 SRTP의 전송률은 TCP-Friendly하다고 할 수 있다. 다시 말해서 SRTP의 전송률의 변화로 인한 추가적인 혼잡 상황은 발생되지 않게 된다. 또한 이렇게 동적으로 결정되는 SRTP의 전송률은 장시간의 관점으로 볼 때 스트리밍에 최적화된 트래픽 형태인 작은 전송 진동 폭을 갖는 트래픽 특성을 나타낸다.

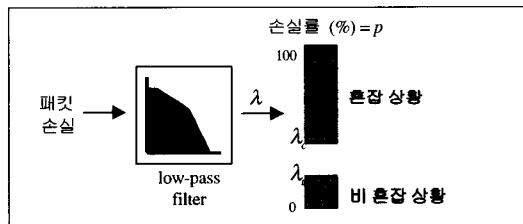


그림 7 네트워크 상태 분류

그림 8은 TCP-Friendly 특성을 고려한 SRTP의 전송률 판단 알고리즘이다. $N+1$ 상태의 측정값(p, t_{RTT})을 기반으로 식 (2)에 의해 얻어진 전송률이 N 상태, 즉 현재의 스트림 전송률($\overline{T_{RTT}}$)보다 작을 경우 이를 혼잡 상황으로 간주하고 그림 8의 알고리즘처럼 전송률을 감소시킨다. 손실률에 따른 전송률 감소($T_{RTT} = \overline{T_{SRTP}}(1 - \sqrt{p})$) 비례식은 식 (2)를 P 에 의해 정리하여 얻은 관계식에 근거한

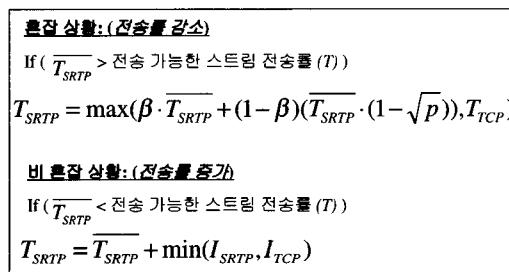


그림 8 상태에 따른 전송률 판단 알고리즘

다. T_{TCP} 는 TCP 플로우의 전송률을 나타내며 (는 전송률 판단에 관한 비례 상수이다. 그러므로 혼잡상황에서 SRTP는 경쟁하는 TCP보다 적어도 같거나 약간 큰 값으로 전송률을 감소시키므로 보다 빠르게 네트워크 혼잡 상황을 극복할 수 있다.

반면 $N+1$ 상태의 측정 값으로 얻은 $N+1$ 상태의 전송률이 현재 N 상태의 전송률보다 클 경우는 SRTP의 전송률을 증가 시킨다. I_{TCP} 는 제안한 스트림 전송 플랫폼의 리소스 매니저에 의해 측정된 TCP의 하나의 RTT당 증가하는 전송률 증가 값을 나타내며 I_{SRTP} 는 SRTP의 전송률 증가 값을 나타낸다. 여기서 I_{SRTP} 는 스트림 미디어의 품질 특성을 반영 할 수 있는 상수를 의미한다. 예를 들어 MPEG의 경우 특정 B나 P 프레임의 개수를 I_{SRTP} 의 상수 값으로 정의하여 추가적인 품질 향상의 기준으로 사용할 수 있다[13,14]. 그러므로 비 혼잡 상태에서는 적어도 경쟁하는 TCP 정도의 전송률 증가를 이를 수 있으며 상태에 따라서 스트림 미디어의 전송률 품질을 크게 향상시킬 수 있다.

4. 실시간 전송 플랫폼의 구현 및 성능 평가

본 장에서는 설계하고 구현한 실시간 스트림 전송 플랫폼에 대해 기술한다. 또한 구현한 플랫폼의 성능을 평가하기 위해서 적절한 시험 망을 구성하였고 본 플랫폼을 통한 스트림 트래픽의 특성을 시험하고 분석하였다. 또한 제안한 플랫폼을 구성하는 각 기능 모듈의 세부 기능을 정의한 후 이를 OMG IDL(Interface Definition Language)을 통해 표준화된 인터페이스 형식으로 표기하였다.

4.1 인터페이스 구현

표 1은 플랫폼 구성 모듈과 표준 인터페이스와의 관계를 나타낸 것이다. 스트림 컨트롤러는 두 개의 인터페이스로 분리하여 구현하였다. *BasicStreamCtrl* 인터페이스에는 다른 인터페이스를 통해 호출되는 오퍼레이션들이 구현되어 있다. 용용이 *BasicStreamCtrl*의 오퍼레이션을 호출할 경우 스트림 컨트롤러는 해당 인터페이스의 오퍼레이션을 호출하게 된다. *StreamCtrl* 인터페이스는 *BasicStreamCtrl*를 상속하며, 클라이언트와 서버의 리소스 핸들러를 바인딩하기 위한 오퍼레이션으로 구현되어 있다.

ResourceMgr 인터페이스는 클라이언트측 리소스 핸들러가 호출하는 서비스 요구의 승인을 결정하고, 종단간의 최대 병목지점의 값(B_{bn}), TCP 연결의 상태 등의 기능을 위한 오퍼레이션들을 본 플랫폼에 제공한다. 또한, 리소스 핸들러는 두개의 IDL 인터페이스로서 구현되어 있

다. *MDevice* 인터페이스는 스트림 핸들러의 생성 및 삭제를 위한 오퍼레이션을 수행하거나, 상대 *MDevice*와 직접 바인딩을 맺기 위한 오퍼레이션을 제공한다. *Negotiator* 인터페이스는 스트림 데이터 전송에 필요한 자원을 *ResourceMgr*에서 조회하고 승인된 자원을 송신 측에 알려주기 위한 오퍼레이션을 수행한다.

마지막으로 구현한 *StreamHdl* 인터페이스는 스트림 설정 및 전송을 위한 실제적인 오퍼레이션은 수행한다. 그리고 *StreamHdl_A*와 *StreamHdl_B* 인터페이스는 단지 스트림 생성을 위한 접속 순서를 정의한 인터페이스이다.

표 1 구성 모듈별 인터페이스 정의

구성 모듈	IDL 인터페이스
스트림 컨트롤러	BasicStreamCtrl, StreamCtrl
리소스 매니저	ResourceMgr
리소스 핸들러	MDevice, Negotiator
스트림 핸들러	StreamHdl, StreamHdl_A, StreamHdl_B

4.2 실시간 스트리밍 전송 플랫폼의 동작 과정

설계한 인터페이스 객체들은 그림 9와 같은 오퍼레이션들을 호출하며 상호 동작한다.

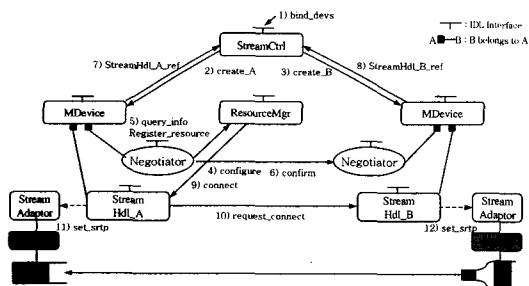


그림 9 실시간 스트리밍 전송 플랫폼의 인터페이스
상호 동작

옹용으로부터 bind_devs란 오퍼레이션이 호출된다.
StreamCtrl 객체는 A측과 B측의 *MDevice* 객체에 대해 각각 *create_A*와 *create_B* 오퍼레이션을 호출하여 *StreamHdl* 객체의 생성을 요구한다. *MDevice* 객체는 우선 *Negotiator* 객체를 생성한 후 스트림 설정에 필요한 정보를 교환한다. 연결 설정 정보를 바탕으로 A측의 *Negotiator* 객체는 *ResourceMgr* 객체에게 서비스 승인을 요청한다. *ResourceMgr* 객체로부터 서비스가 승인되면 A측 *Negotiator*는 B측에 대해 *confirm* 오퍼레이션을 호출하여 최종적으로 설정 정보를 전달한다. 구현

체적으로 *ResourceMgr* 객체는 주기적인 *StreamHdl* 와의 팩트 쌍의 정보를 통해 B_{bn} 값을 구하게 되고 이 정보에 근간하여 새로운 서비스에 대한 승인제어를 담당한다.

다음 단계로 *MDevice* 객체는 *StreamHdl_A*와 *StreamHdl_B* 객체를 생성하고 이에 대한 객체 참조를 *StreamCtrl* 객체에게 반환한다. *StreamCtrl* 객체는 *StreamHdl_A* 객체에 대해 connect 오퍼레이션을 호출함으로써 A측의 스트림 핸들러와 B측의 스트림 핸들러가 스트림 경로를 설정할 것을 지시한다. *StreamHdl_A*가 B측에 대해 request_connect 오퍼레이션을 호출하면 *StreamHdl_B*는 스트림 어댑터를 통해 NSAP(Network Service Access Point)를 생성하고 주소 정보를 매개변수에 기록하여 반환한다. *StreamHdl_A*는 자신의 스트림 어댑터를 통해 해당 주소로 접속한다. 연결이 제안한 플랫폼 상에서 모두 설정되고 스트림이 전송 SRTP 모듈을 동작시켜 모든 준비 상태를 마치게 된다. 여기까지의 과정을 통해 데이터 전송 경로가 설정되어 이후에 응용으로부터 start 오퍼레이션이 호출되면 *StreamCtrl* 객체는 내부적으로 *StreamHdl_B*의 start 오퍼레이션을 호출하여 스트림 전송을 개시한다.

4.3 시험방의 구성

구현한 실시간 스트리밍 전송 플랫폼의 기능 및 성능을 평가하기 위해서 그림 10과 같은 시험 망을 구성하였다. 본 시험망은 기본적인 아령형(dumbbell) 네트워크 구조로 설계 하였으며 각각 n개의 짧은 TCP 트래픽(TWWW), 긴 TCP 트래픽(FTFP), 그리고 AV 스트리밍 트래픽(SRTP)을 일정 기간 전송할 수 있게 하였다. 여러 번의 시험을 통해 구현한 실시간 스트리밍 전송 플랫폼은 망의 구성의 형태와는 직접적인 영향이 거의 없음을 확인 하였기에 본 논문에서는 기본적인 망 구성에서 시험한 결과에 대해서만 기술하였다.

시협망에 존재하는 서버 및 클라이언트들은 모두 CORBA 기반의 분산 환경을 지원하며 스트림 플랫폼을 통해 전송되는 기본적인 미디어는 MPEG-4를 사용하였다. 각 시스템은 PC기반의 원도우와 리눅스 운영체제 상

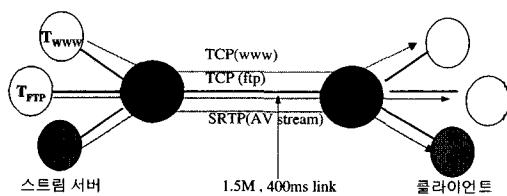


그림 10 시험 망의 구성

에서 동작하며 네트워크 링크의 설정을 위해선 Dummynet [15]을 사용하였다. 그리고 백그라운드 트래픽을 발생시키기 위해서 SURGE toolkit [16] 등을 이용하였다. 구현에 사용한 CORBA 미들웨어 툴킷은 MICO 2.2.7 라이브러리를 사용하였다.

4.4 성능 평가

4.4.1 스트리밍 전송 플랫폼의 기능 및 표준화 구현 검증 실시간 전송 플랫폼의 세부 동작을 검증하기 위해서, 본 논문에서는 OMG의 스트리밍 서비스 참조 모델을 기반으로 구현한 기능들을 비교하였고 결과적으로 각 동작들이 정확하게 표준에 따라 수행됨을 확인할 수 있었다. 특히, 2장에서도 언급했듯이, OMG의 AV 스트리밍 서비스 사양에서는 플로우 테이타 종단과 스트리밍 어댑터 그리고 스트리밍 인터페이스 컨트롤 객체와 플로우 테이타 종단 사이의 인터페이스에 대해서는 명세를 정확히 정의하고 있지 않다. 이것은 일반적으로 스트리밍 응용의 제어 방법과 스트리밍 환경을 구현하는데 가장 중요한 전송 방법을 네트워크 환경에 맞게 구현할 수 있도록 분리하기 위함이다. 본 논문에서는 이러한 OMG AV 참조 모델의 기본 취지에 입각하여 실제 스트리밍 전송 부분의 구현을 최선형 서비스를 제공하는 현재의 인터넷 환경에 적합하게 재정의하였다. 그리고 이를 고려하여 스트리밍 어댑터에 구현한 것이 TCP-Friendly한 혼잡 제어를 지원하는 SRTP이다.

4.4.2 기존의 스트리밍 전송 플랫폼과의 비교

기존의 OMG 스트리밍 참조 모델 기반의 스트리밍 전송 플랫폼으로는 IONA의 Orbix MX[17]와 NEC C&C 연구소의 스트리밍 서버[18], 그리고 가장 널리 알려진 워싱턴 대학의 TAO(The ACE ORB) AV 플랫폼 [5] 등이 있다. 특히 TAO 플랫폼은 CORBA 기반의 AV 스트리밍 플랫폼 연구에 많은 영향을 미쳤으며 현재까지도 많은 프로젝트들이 TAO의 연구 성과를 기반으로 새로운 연구들을 시도하고 있다. 표 2는 본 논문에서 제안한 실시간 스트리밍 전송 플랫폼과 대표적인

표 2 스트리밍 전송 플랫폼간의 비교

	제안한 플랫폼	TAO AV 플랫폼
스트리밍 컨트롤러	OMG 참조 모델 기반	OMG 참조 모델 기반
리소스 매니저	네트워크 트래픽 상황에 따른 승인 제어 기능 지원	기능 없음
리소스 핸들러	향상된 자원 협상 지원	기본적 자원 협상 지원
스트리밍 핸들러	OMG 참조 모델 기반	OMG 참조 모델 기반
스트리밍 어댑터	SRTP를 이용한 TCP-Friendly 지원	UDP 지원

OMG 스트리밍 참조 모델인 TAO AV 플랫폼을 구성 요소의 기능들을 기준으로 비교한 것이다.

표 2에서도 잘 나타난 것처럼 본 논문에서 제안한 플랫폼은 TAO와 비교해 볼 때 다음과 같은 큰 특징이 있다. 첫째는 리소스 매니저의 기능을 통한 네트워크 트래픽 상황에 따른 승인 제어 기능이다. 이것은 효과적인 스트리밍 전송 플랫폼을 위한 가장 기본적인 고려이다. 둘째는 리소스 핸들러를 통한 향상된 자원 협상 방법의 지원이다. 그리고 마지막으로 스트리밍 어댑터를 통한 스트리밍 전송의 TCP-Friendly 성질의 지원이다. 이것은 최근 스트리밍 전송에 관한 진보된 이슈이자 대역폭 불공정성 문제를 해결하기 위해 매우 중요한 기능이다.

4.4.3 TCP-Friendly한 SRTP의 혼잡 제어 기능 검증

스트리밍 어댑터를 통한 스트리밍 전송 기능에 대한 성능 검증은 그림 10에 도시된 트래픽 특성에 잘 나타난다. 그림 11은 각각 $n=20$ 일때 각각의 연결들이 병목 연결인 첫번째 라우터에서의 각 트래픽의 대역폭 점유율을 나타낸다. 시험 결과에서 알 수 있듯이 일반적인 TCP 기반의 혼잡 제어는 큰 폭의 전송 진동 폭을 가진다. 이러한 전송 진동폭의 급격한 변화는 전체 망의 이용 효율을 빈번하게 변화시킨다. 더욱이 이러한 트래픽 유형은 연속적인 스트리밍 전송에 매우 취약하다. 반면 제안한 플랫폼의 스트리밍 어댑터를 통한 SRTP 스트리밍은 기존의 TCP와 호환되는 TCP-Friendly한 특성을 보이며 전체적인 전송률 진동 폭의 변화를 크게 줄인다. 또한, 이 트래픽은 장기간 일정한 전송률을 유지하며 전송률의 평활화를 유도하기 때문에 연속적인 스트리밍 전송에 매우 적합하다.

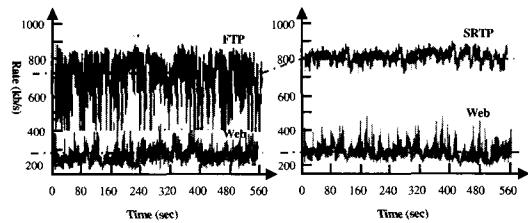


그림 11 WWW/FTP/SRTP의 대역폭 점유율

그림 12는 연속적으로 네트워크의 상태가 혼잡, 비혼잡 상태로 반복되는 실제의 네트워크의 상황과 유사하게 백그라운드 트래픽을 발생시켰을 때 나타나는 SRTP의 공정성 지수 F의 변화를 도시한 것이다. 이것은 각각의 네트워크 상태에서 구해진 손실률과 지연값을 기반으로 TCP의 실제 전송률 T_{TCP} 와 같은 상황에서의

T_{SRTP} 과 구하여 이를 식 (1)에 대입하여 얻은 결과이다. 결론적으로 SRTP의 공정성 지수 F가 거의 1에 수렴하고 있으므로 SRTP를 이용하는 스트림 트래픽들은 기존의 TCP 트래픽들과의 대역폭 불공정성 문제를 해결하면서 조화를 이룬다고 말할 수 있다.

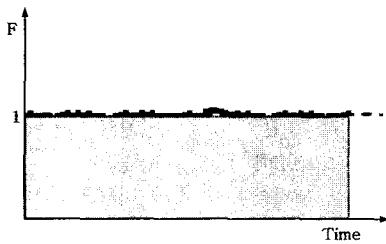


그림 12 SRTP의 공정성 지수(F)의 변화

5. 결 론

최근 폭발적으로 증가하고 있는 스트림 전송 응용들을 분산 환경이 지원되는 대규모 미들웨어 환경에서 제공하기 위한 연구들이 많이 진행되고 있다. 특히 CORBA와 같은 분산 미들웨어는 이미 성공한 서비스 플랫폼으로 평가 받고 있기 때문에 새로운 서비스를 구현하는데 매우 중요한 기반 기술로 인식되고 있다. 이러한 이유 때문에 OMG에서 제안한 AV 스트리밍 참조 모델은 스트리밍 플랫폼으로서 매우 주목 받고 있다. 그러나 이 참조 모델은 스트리밍을 운영하기 위한 프레임워크는 있지만 구체적인 하부 구현 방법에 대한 기술이 없는 실정이다.

본 논문에서는 OMG의 스트리밍 참조 모델을 기반으로 새로운 실시간 스트림 전송 플랫폼을 제안하고 이를 구현하였다. 제안한 플랫폼은 CORBA 서비스 형태로 기존의 CORBA 미들웨어 상에 위치할 수 있다. 특히 본 플랫폼의 스트림 제어 및 자원 승인 결정 및 관리 기능을 OMG IDL의 표준 인터페이스로 제공함으로써 CORBA의 장점인 다양한 시스템 간의 상호 운용성을 충분히 보장 할 수 있었다. 또한 제안한 스트림 전송 플랫폼은 현재의 인터넷 환경에 적합한 스트리밍 전송 방법으로 TCP-Friendly 혼잡 제어를 지원하는 SRTP를 하부 서비스로 제공한다. 이를 통하여 본 플랫폼을 통한 스트림은 기존의 TCP 연결들과 대역폭을 공정하게 분배하면서도 전송 효율을 최적화 할 수 있었다. 특히 SRTP는 연속적인 멀티미디어 스트림을 위한 적합한 트래픽 특성인 전송률의 평활성도 제공한다.

향후 연구 과제로는 본 플랫폼을 현재의 유니캐스트

환경뿐만 아니라 멀티캐스트 환경도 지원 할 수 있는 구조로 그 기능을 확대하는 것에 관한 연구가 이루어져야 할 것이다.

참 고 문 헌

- [1] D. Reininger, D. Raychaudhuri, and M. Ott, "A Dynamic Quality of Service Framework for Video in Broadband Networks," IEEE Network, Nov. 1998.
- [2] ISO/IEC JTC1/SC29/WG11, Digital Storage Media-Command and Control, ISO/IEC 13818-6 N1100, Oct. 1995.
- [3] Object Management Group, Control and Management of A/V Streams Specification, OMG Document Telecom, Oct. 1998.
- [4] J. W. Hong, J. Kim, J. Park, "A CORBA-Based Quality of Service Management Framework for Distributed Multimedia Services and Applications," IEEE Network, March 1998.
- [5] S. Mungee, N. Surendran, D. Schmidt, "The Design and Performance of a CORBA Audio/Video Streaming," Proceeding of HICSS-32, Jan. 1999.
- [6] J. Widmer, R. Denda, and Martin Mauve, "A Survey on TCP-Friendly Congestion Control," IEEE Network, May/June 2001.
- [7] OMG, "The Common Object Request Broker: Architecture and Specification," Ver. 2.0, July 1995.
- [8] OMG, "The CORBAservices: Common Object Services Specification," OMG Doc. 95-3-31, March 1995.
- [9] A. Gokhale and D. C. Schmidt, "Optimizing the Performance of the CORBA Internet Inter-ORB Protocol Over ATM," Proceeding of SIGCOMM '97, Aug. 1997.
- [10] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP Throughput: A Simple Model and Its Empirical Validation," Proceeding of SIGCOMM 98, Aug. 1998.
- [11] K. Lai and M. Baker, "Measuring Bandwidth," Proceeding of INFOCOMM 99, 1999.
- [12] H. Schulzrinne, "RTP: A Transport Protocol for Real-Time Application," IETF Audio-Video Transport Working Group, RFC 1889, Jan. 1996.
- [13] 송병훈, 최상기, 정광수, "QoS 적용 기능을 갖는 멀티미디어 스트리밍 서비스 구현", 한국정보과학회 논문지 (C), 5권 5호, 1999. 10.
- [14] B. Song, K. Chung, and Y. Shin, "SRTP:

- TCP-Friendly Congestion Control for Multimedia Streaming," Proceeding of ICOIN16, Jan. 2002.
- [15] Dummynet, <http://www.iet.unipi.it/~luigi/>.
- [16] Scalable URL Reference Generator(SURGE), <http://www.cs.bu.edu/>.
- [17] IONA, "IONA Orbix MX," www.iona.com.
- [18] J. Redlich, "A Distributed Object Architecture for QoS sensitive Networking," Proceeding of Open Arch, April 1998.



송 병 훈

1998년 광운대학교 전자통신공학과 학사
 2000년 광운대학교 전자통신공학과 석사
 2000~현재 광운대학교 전자통신공학과 박사과정



정 광 수

1981년 한양대학교 전자공학과 학사 1983년 한국과학기술원 전기 및 전자공학과 석사. 1991년 미국 University of Florida 전기공학과 박사(컴퓨터공학전공). 1983~1993년 한국전자통신연구원 선임연구원 1991~1992년 한국과학기술원 대우 교수. 1993년~현재 광운대학교 전자공학부 교수(정보통신 연구원)



정 형 석

1998년 광운대학교 전자통신공학과 학사
 2000년 광운대학교 전자통신공학과 석사
 2000~현재 한국전자통신연구원 선임연구원