특집논문-01-6-3-01

IP망을 통한 MP4 파일 스트리밍 시스템의 설계 및 구현

김 현 철*, 민 승 홍**, 서 덕 영**, 김 규 헌*, 김 진 웅*

Design and Implementation of An MP4 File Streaming System over IP Networks

Hyun-Cheol Kim*, Seunghong Min**, Doug Young Suh**, Kyuheon Kim* and Jinwoong Kim*

요 약

본 논문에서는 IP 망을 통한 MP4 파일 스트리밍 시스템을 소개한다. 사용자는 제안된 스트리밍 시스템을 이용하여 IP망을 통해 서버측의 MP4 컨텐츠를 엑세스할 수 있으며, 스트리밍되는 컨텐츠와 상호 작용도 가능하다. MP4 파일 형식은 MPEG-4 미디어 정보를 저장할 수 있도록 설계되었으며, 객체 지향적 구조를 이루고 있다. 본 논문에서 제시한 스트리밍 서버 시스템은 GUI, Session Manager, Splitter, SL Packetizer, Transmitter 등으로 구성되어 있으며, 클라이언트 시스템은 MPEG-4 참조 소프트웨어인 IM1-2D 플레이어를 기반으로 구현되었다. 또한 본 스트리밍 시스템은 실시간 전송이 필요한 미디어 데이터와 QoS 관리를 위해 RTP와 RTCP를 각각 사용하도록 설계되었으며, 초기객체기술자(IOD), 객체기술자(OD), BIFS와 같은, 스트리밍을 위해서 반드시 전송되어져야 할 데이터와 정지영상 및 텍스트 등과 같이 하나의 패킷으로 전송이 완료되는 미디어 데이터를 위해서는 TCP를 사용하도록 설계되었다.

Abstract

In this paper, we present an MP4 file streaming system over IP networks. Using the proposed system, a user can access MP4 contents in the server via IP networks and interact with the contents. The MP4 file format is designed to contain the media information of MPEG-4 and is object-oriented. The presented streaming server system contains GUI, session manager, splitter, SL packetizer and transmitter. In addition, we have implemented the client system based on the IM1-2D player, the MPEG-4 reference software. The presented streaming system is designed to use RTP for the media data requiring real-time streaming, RTCP for QoS management, and TCP for the data such as IOD(Initial Object Descriptor), OD(Object Descriptro), BIFS(Binary Format for Scene), which should be transmitted for the streaming, and the data, such as still image and text, which can be entirely transmitted in a packet.

I. 서 론

MPEG-4는 멀티미디어 부호화에 관한 ISO/IEC 국제 표준으로서 오디오, 비주얼, 시스템 등 현재 10개 분야로 나누어 표준화가 진행되고 있으며, 디지털 TV, 대화형 그래픽스 응용, 대화형 멀티미디어 분야에서 성공적으로 사용되고 있다[1]. 기존의 MPEG-1/2가 자연 영상과 오디오만을 지원하던 것에 비해, MPEG-4는 자연영상 뿐만 아니라, 정지영상, 컴퓨터 그래픽스, 합성 오디오 및 텍스트 등을 포함하는 멀티미디어의 부호화를 목적으로 하고 있다[2]. 또한 객체 기반 부호화를 사용하여 임의 모양 영상을 지원할 수 있으며, 객체들을 시간적 공간적 위치 정보에 따라 배치한 장면의 부호화를 지원한다. 장면 정보에는 시간적·공간적 위치 뿐만 아니라, 객체의 행위와 대화성(interactivity) 정보를 포함하므로

* 한국전자통신연구원 무선방송연구소 방송미디어연구부
Broadcasting Media Technology Dep., Radio & Broadcasting Research Lab., ETRI
** 경희대학교 전자정보학부
School of Electronics and Information, KyungHee University

MPEG-4 기반의 컨텐츠는 사용자와의 대화 기능을 제공한다. 또한 MPEG-4 미디어의 QoS 기술자(QoS descriptor)를 사용하여, 전송 채널에서 미디어 객체의 요구사항과 미디어 객체가 생성하는 트래픽의 특성을 전달할 수 있으므로 효율적인 멀티미디어 서비스를 가능하게 한다.

현재 인터넷 사용자의 꾸준한 증가로 인터넷을 통한 멀티미디어 전송 요구가 증가하고 있으며, 단방향 전송을 통한 단순 시청에서 사용자의 요구를 표시하고 전달할 수 있는 대화형 컨텐츠에 대한 요구 또한 증가하고 있다. 따라서 인터넷을 통한 객체기반 대화형 컨텐츠의 전달 방법에 대한 연구가 필요하다.

본 논문에서는 MPEG-4에 기반한 대화형 멀티미디어 컨텐츠를 인터넷을 통해 스트리밍하기 위한 시스템의 설계와 구현에 대해 설명한다. 본 논문에서 제시한 스트리밍 서버 시스템은 MPEG-4 컨텐츠를 저장한 MP4 파일의 전송 기능과, 클라이언트로부터 수신된 사용자의 요구에 대응하는 기능을 수행하며, 클라이언트 시스템은 서버에서 전송한 스트림을 각 미디어별로 수신하고 복호화한 후 이를 장면(scene) 정보에 맞게 구성하여 사용자에게 프리젠테이션 하는 기능과 사용자의 요구를 서버측으로 전송하는 기능을 가진다. MP4 파일은 장면을 구성하기 위해 필요한 정보와 장면을 구성하는 각 미디어 별로 분리되어 전송되며, 전송되는 데이터의 특성에 따라 사용하는 프로토콜을 달리한다. 미디어 객체가 아닌, 장면을 구성하기 위해 필요한 정보와 정지영상과 텍스트 정보처럼 한번의 전송으로 렌더링이 지속되는 미디어 데이터는 TCP(Transmission control protocol)를 사용하며, AV 객체와 같이 실시간 스트리밍이 요구되는 미디어 데이터는 RTP(Real-time transport protocol)와 UDP (User datagram protocol)를 사용한다. 그리고, 본 스트리밍 시스템은 네트웍 상태의 측정을 위하여 RTCP (RTP control protocol) 패킷을 송수신할 수 있도록 설계되었다.

본 논문은 다음과 같이 구성되어 있다. Ⅱ장에서는 MPEG-4 시스템과 MP4 파일에 대해 설명하고, Ⅲ장에서는 실시간 스트리밍을 위해 사용한 RTP와 RTCP에 대해서 설명한다. 그리고 Ⅳ장에서는 설계 및 구현한 MP4파일 스트리밍 시스템에 대해서 설명하고, Ⅴ장에서 실행 결과를 보인 후 Ⅵ장에서 결론과 향후 발전 방향에 대해 설명한다.

Ⅱ. MPEG-4 시스템

MPEG-4 시스템(ISO/IEC 14496-1)에서는 기존의 전형적인 시스템 기능이라 할 수 있는 동기화에 관련된 기능뿐만 아니라, 각 객체들의 시·공간적 관계를 나타내는 정보를 이용한 장면 구성 기능도 포함하고 있다. 본 장에서는 MPEG-4 장면구성 방법과 MPEG-4 장면을 저장하는 MP4 파일 형식에 대해 설명한다.

1. MPEG-4 시스템 장면 구성

그림 1에 MPEG-4 터미널 구조를 나타내었으며, MPEG-4 터미널은 전달 계층 (delivery layer), 동기 계층(sync layer), 압축 계층(compression layer) 등으로 구성된다.

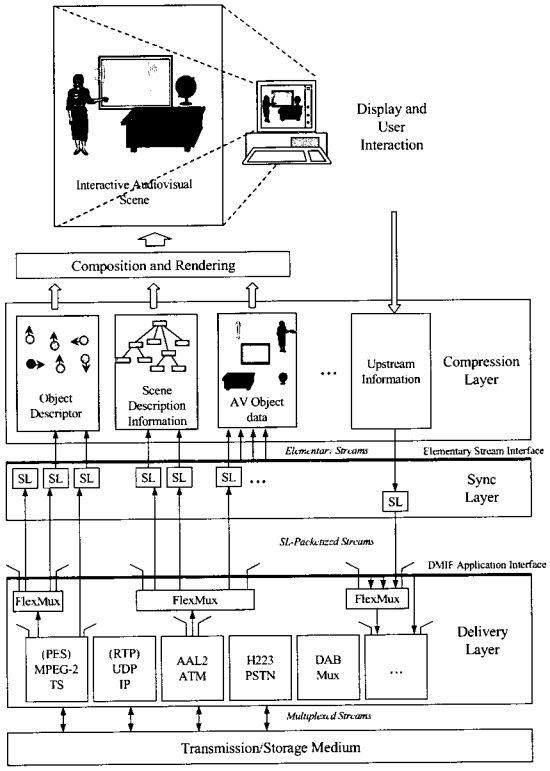


그림 1. ISO/IEC 14496 터미널 구조
Fig. 1. The ISO/IEC 14496 terminal architecture

수신 터미널에서의 각 계층은 다음과 같은 기능을 한다. 전달 계층은 DMIF(Delivery multimedia integration framework : ISO/IEC 14496-6)에서 표준화되며 MPEG-4 동기 계층과 네트웍 계층 사이의 인터페이스를 정의한다

[3][4]. 동기 계층은 객체들의 복호화 동기와 컴포지션 동기를 맞출 수 있도록, 전달 계층으로부터 전달된 SL 패킷화된 스트림에서 시간 정보와 동기화 정보를 추출한다. 압축 계층은 객체의 컴포지션과 렌더링에 이용할 수 있도록 객체의 부호화된 스트림을 입력받아 복호화하는 기능을 한다.

MPEG-4 장면을 구성하기 위해서는 일반적으로 초기객체기술자(IOD: Initial object descriptor), BIFS(Binary Format for Scene), 객체기술자(OD: Object descriptor), 장면을 구성하는 미디어 데이터들이 필요하다. 초기객체기술자는 MPEG-4 장면을 구성하기 위해 가장 먼저 해석되어야 하는 정보로, 각 미디어의 프로파일과 레벨을 기술하고, 장면기술자 스트림과 객체기술자 스트림에 대한 기초스트림 기술자(Elementary stream descriptor)를 포함한다. BIFS는 장면을 구성하는 AV 객체의 공간적·시간적 위치와 행위를 나타내는 정보로서 장면의 구조를 기술하는 노드(node)들의 집합으로 구성되며 AV 객체와는 독립적으로 코딩된다. BIFS에서 장면을 구성하는 객체들의 대화성(interactivity)과 행위를 표현하기 위해서 VRML(Virtual Reality Modeling Language, ISO/IEC 14772-1:1998)[5]에 정의되어 있는 이벤트 구조가 사용되며, 센서 노드와 라우트로써 기술된다. 센서 노드는 사용자의 작용이나 장면의 변화에 기초한 이벤트를 생성해 내는 노드이며, 라우트는 이벤트 발생 노드와 이벤트 수신 노드 사이의 연결로서 이벤트의 전파를 위해 사용된다. 객체기술자는 장면을 구성하는 각 미디어 객체의 기초스트림(Elementary stream)에 대한 정보를 기술하는 기술자(descriptor)들의 집합으로서, 각 미디어 객체의 기초스트림과 장면기술과의 연결을 제공한다. 그림 2는 객체기술자를 이용한 장면기술과 각 기초스트림들의 연결을 나타낸다.

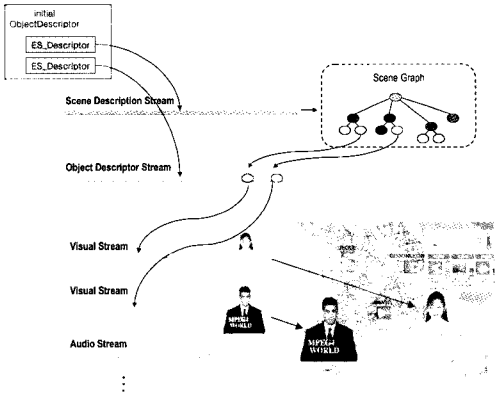


그림 2. 기초스트림과 장면기술을 연결하는 객체 기술자
Fig. 2. Object descriptors linking scene description to elementary streams

2. MP4 파일

MP4 파일은 MPEG-4 미디어 정보를 저장하기 위해서 디자인되었으며, 아톰(atom)이라 불리는 객체 지향적 구조로 구성되어 있다. 각 아톰은 크기(size)와 고유한 아톰 타입(atom type) 필드로 시작되고[6], 대부분의 아톰들은, 미디어 데이터의 크기나 위치, 트랙 ID등과 같은 메타 데이터를 기술하고 있으며, 'moov' 아톰에 포함된다. 반면, 각 미디어 데이터는 'mdat' 아톰에 저장되어 MP4 파일 내에 위치할 수도 있고, MP4 파일 외부에 위치하여 URL을 이용해 참조될 수도 있다. 그림 3은 'moov'와 'mdat' 아톰을 포함하고 있는 MP4 파일 구조이다.

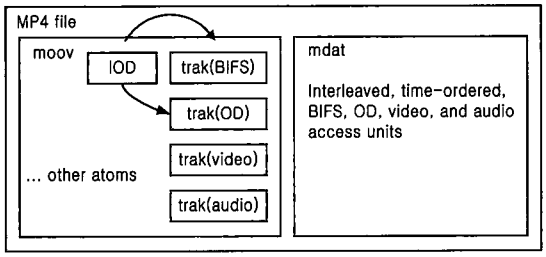


그림 3. MP4 파일 구조
Fig. 3. MP4 file structure

본 논문에 나타낸 스트리밍 시스템에서는 하나의 파일에 'moov' 아톰과 모든 미디어 데이터들이 저장된 파일 형태만을 고려하였다.

III. 실시간 전송 프로토콜

RTP는 AV 데이터와 같은 실시간 특성을 가지는 데이터의 종단간(end-to-end) 전송 서비스를 제공하기 위한 프로토콜이며, RTCP는 QoS를 측정하고, 진행중인 세션에 참가중인 참석자의 정보를 전달하기 위한 프로토콜이다[7]. 제안한 스트리밍 시스템은 미디어 데이터의 전송을 위해 RTP와 RTCP를 사용하도록 설계되었으며, 본 장에서는 RTP와 RTCP에 대해 기술하고자 한다.

1. RTP

TCP는 신뢰성을 강조한 프로토콜로 실시간 전송에 어려움이 있고, UDP는 신뢰성은 낮지만 TCP보다 빠른 전송을 할 수 있는 특징을 가진다[8]. 따라서 UDP를 사용할

경우 패킷의 손실이 발생할 수 있으며, 패킷의 순서가 바뀌어 도착할 수도 있다. 이러한 경우 RTP를 사용함으로써 패킷 손실이나 패킷 순서를 확인할 수 있는데, 이는 RTP 헤더의 열번호(sequence number)와 시간정보(time stamp)를 이용하여 가능하다. 그림 4에는 RTP 헤더 형식을 나타내었다.

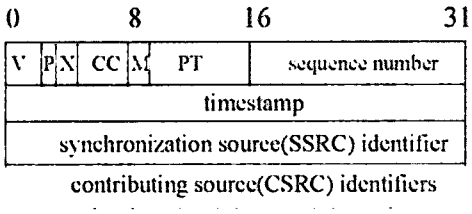


그림 4. RTP 헤더 형식
Fig. 4. RTP header format

RTP 패킷은 헤더와 페이로드(payload)로 구성되며, 헤더에는 그림 4에 나타낸 고정 헤더 필드(fixed header fields) 외에 헤더 확장(header extension)이 추가될 수 있다. 그리고 페이로드에는 실제 전송하고자 하는 데이터가 들어가게 된다.

2. RTCP

다양한 제어 정보를 전달하기 위한 RTCP 패킷은 몇 가지 타입으로 나뉘는데, SR(Sender report), RR(Receiver report), SDES(Source description), BYE, APP등이 있다. RTCP는 능동 송신자(active sender)인 참석자로부터 전달되는 SR 패킷 또는 능동 송신자가 아닌 참석자로부터 전달되는 RR 패킷을 사용하여 손실된 패킷 수나 지터(jitter)등 전송 품질과 관련된 통계적 정보를 제공한다. 그리고 SDES 패킷을 통해, 송신자를 식별하는데 필요한 정보를 제공할 수 있다. BYE 패킷은 세션을 탈퇴할 때 사용되며, APP 패킷은 새로운 응용이나 특징(feature)이 개발되었을 때 실험을 목적으로 사용된다[7][8].

Ⅳ. MP4 파일 스트리밍 시스템의 구조 및 동작

본 장에서는 스트리밍 시스템의 구조를 서버와 클라이언트로 나누어 기술하고, 스트리밍 시스템의 동작 방법도 서버와 클라이언트로 구분하여 설명하고자 한다.

1. 스트리밍 시스템 구조

1.1 서버 구조

본 논문에서 제시한 스트리밍 서버는 MP4 파일을 미디어 별로 분리하여 클라이언트로 전송하는 기능과, 클라이언트로부터 수신된 사용자의 요구에 대응하는 기능을 수행할 수 있도록 설계되었다. 서버는 전송하고자 하는 스트림의 종류에 따라 TCP 또는 RTP를 구분하여 사용하는데, 손실에 민감한 초기객체기술자, 객체기술자, BIFS 정보와 한번의 전송 후 일정기간 지속적으로 렌더링 되는 정지영상, 텍스트 등과 같은 미디어 데이터는 TCP로 전송되며, 지연과 지터에 민감한 AV 미디어 데이터는 RTP를 사용하여 전송되도록 설계되었다.

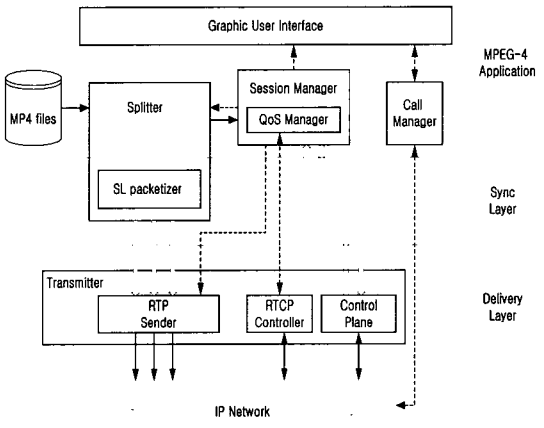


그림 5. 스트리밍 서버 구조
Fig. 5. The architecture of streaming server

그림 5는 본 논문에서 제안한 스트리밍 서버 구조이다. 실선 화살표는 데이터의 흐름을, 점선 화살표는 컨트롤 메시지의 흐름을 나타내며, 이중화살표는 쓰레드(Thread)의 동기를 위하여 사용한 미디어스트림(MediaStream) 객체이다. 미디어스트림은 IM1-2D 플레이어에서 제공되는 것과 동일한 것으로서, 메모리 버퍼와 FIFO(First-In First-Out) 메카니즘으로 구성되어 있으며 데이터 스트림의 버퍼링과 전달을 관리한다[9]. 미디어스트림은 RTP sender 모듈과 SL Packetizer 모듈 사이에, RTP를 사용하여 스트리밍되는 미디어 개수만큼 생성되고, Session Manager 모듈과 Control Plane 모듈 사이에, TCP 데이터의 송신과 수신을 위해 2개가 생성된다. 그림 5에 나타낸 스트리밍 서버는 크게, MPEG-4 응용 계층(MPEG-4 Application layer), 동기 계층(Sync layer), 전달 계층

(Delivery layer)의 3부분으로 나눌 수 있다.

1.1.1 MPEG-4 Application layer

MPEG-4 응용 계층은 사용자 인터페이스를 제공하고 전체 시스템을 제어하는 기능을 가진 계층으로서 Call Manager, Session Manager, Splitter 모듈 등을 포함한다.

- Call Manager : Call Manager는 클라이언트로부터 접속 요청을 기다리는 기능을 하며, 접속 요청이 있을 경우 이를 GUI로 알리는 기능을 한다. GUI로 접속 요청을 알린 후, 다시 접속 요청 대기 상태가 된다.
- Session Manager : Session Manager는 세션을 관리하는 모듈로, 초기객체기술자의 전송을 명령하고 클라이언트에서 서버로 전송되는 업스트림(upstream) 메시지를 분석하며, QoS Manager 모듈을 포함하고 있어 RTCP를 통한 네트워 상태를 전달받아 QoS를 관리하는 모듈이다.
- Splitter : Splitter는 II장 2절에서 설명한 MP4 파일의 'moov' 아톰을 분석하여 초기객체기술자, BIFS, 객체기술자와 각 미디어 객체별 스트림으로 분리한 후 각 기초스트림을 액세스 유닛(Access unit)단위로 세분하고 그 위치를 저장함으로써, 클라이언트측에서 요구하는 기초스트림을 스트리밍할 수 있도록 한다. 또한 SL Packetizer를 포함하고 있어 미디어 객체별로 분리된 기초스트림을 SL 패킷화하는 기능을 수행한다.

1.1.2 Sync Layer

스트리밍 데이터의 동기화를 위하여 각 미디어의 기초스트림을 SL 패킷화 스트림(SL- packetized stream)으로 만드는 기능을 Sync Layer에서 수행한다.

- SL Packetizer : SL Packetizer는 Splitter에 포함되어 있으며, Splitter에서 생성한 AU단위의 위치정보와 SLConfigDescriptor 정보를 이용하여 각 미디어의 기초스트림을 SL 패킷으로 만드는 기능을 한다.

1.1.3 Delivery Layer

Delivery Layer는 초기객체기술자, BIFS, 객체기술자와 미디어 데이터를 전송하고, 클라이언트에서 서버측으로 전달되는 업스트림을 수신하여 Session Manager에게 알리는 기능을 하며, RTP Sender, RTCP Controller, Control Plane 모듈 등으로 구성된다.

- RTP Sender : RTP Sender는 Splitter내의 SL Packetizer에서 전달받은 각 SL 패킷을 RTP 패킷화한 후 이를 클라이언트측으로 전송하는 모듈이다.
- RTCP Controller : RTCP Controller는 RTCP SR 패킷을 생성하고 이를 클라이언트 측으로 전송하는 기능과 클라이언트로부터 전달되는 RTCP RR 패킷을 수신한 후 이를 QoS Manager로 전달하는 기능을 수행함으로써 네트워 상태를 파악할 수 있도록 한다.
- Control Plane : TCP를 사용하는 초기객체기술자, 객체기술자, BIFS 및 정지영상과 같이 한번의 전송으로 렌더링이 지속되는 미디어 데이터를 전송하며, 클라이언트로부터 전달되는 업스트림을 수신하여 Session Manager로 전달함으로써 클라이언트의 요구에 대응할 수 있도록 한다.

1.2 클라이언트 구조

본 논문에서 제시한 클라이언트의 구조는 그림 6과 같으며, MPEG-4 참조 소프트웨어인 IM1-2D 플레이어를 기반으로 구현되었다. 본 클라이언트는 서버측에서 전송되는 데이터의 수신 및 복호화, 컴포지션, 렌더링을 수행하여 사용자에게 장면을 보여주고, 사용자로부터 장면에 대한 요구를 입력받아 서버측으로 전달하는 기능을 수행한다.

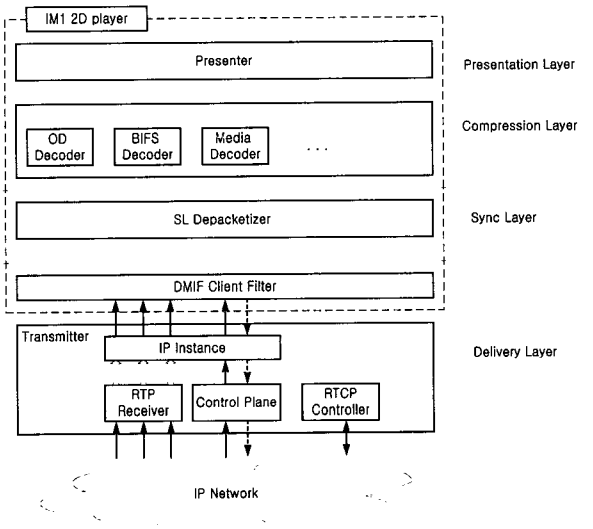


그림 6. 클라이언트 구조
Fig. 6. The architecture of a client

앞서 IV장 1.1절에서 설명한 서버 구조에서와 마찬가지로 실선 화살표는 데이터의 흐름을 나타내고, 점선 화살표는 컨트롤 메시지의 흐름을 나타낸다. 그리고 이중화살표는 미디어스트림 객체를 나타내는데, RTP Receiver 모듈과 IP Instance 모듈 사이에 RTP를 사용하여 스트리밍

되는 미디어 개수만큼 생성된다. 클라이언트는 MPEG-4 시스템에 따라 전달계층, 동기계층, 압축 계층, 프리젠테이션 계층 등으로 구분될 수 있으며, 동기 계층, 압축 계층, 프리젠테이션 계층은 IM1-2D 플레이어의 구조를 따른다. 그러나 현재 IM1-2D 플레이어는 원격 시나리오(remote scenario)를 지원하지 않으므로, 본 논문에서 제시한 시스템에서는 IP 망을 통한 스트리밍에 적합하도록 전달 계층을 설계하였다. 전달 계층은 RTP Receiver, RTCP Controller, Control Plane, IP Instance 모듈 등으로 구성된다.

- RTP Receiver : RTP Receiver는 스트리밍 서버로부터 전송된 RTP 패킷화된 미디어 데이터를 수신한 후 RTP 패킷헤더를 제거하여 SL 패킷으로 만들고 이를 DMIFClientFilter로 전달하는 역할을 한다. 여기서 DMIFClientFilter는 IM1-2D 플레이어에서 제공하는 객체로 IM1-2D 플레이어와 전달계층과의 통신을 위한 인터페이스로 사용된다.
- RTCP Controller : RTCP Controller는 RTCP SR 패킷을 입력받고 RTCP RR 패킷을 생성해서 서버측으로 전송하는 기능을 한다.
- Control Plane : TCP를 이용해서 서버측으로 업스트림을 전송하는 기능과, TCP를 통해 서버에서 클라이언트 측으로 전달되는 데이터를 수신하여 IP Instance로 전달하는 기능을 Control Plane에서 수행한다.
- IP Instance : IP Instance는 DMIFClientFilter로부터 전달되는 메시지를 이용하여 서버측으로 전송할 패킷을 생성한 후, Control Plane 모듈로 전달하는 기능과 Control Plane과 RTP Receiver를 통해 전달되는 데이터를 DMIFClientFilter로 전달하는 기능을 수행한다.

2. 스트리밍 시스템 동작

본 절에서는 제안한 스트리밍 서버와 클라언트 각각에 대한 동작을 호설정(Call setup) 과정과 데이터 전송(Data transmission) 과정으로 나누어 설명한다.

2.1 서버 동작

스트리밍 서버가 동작하는 과정을 그림 7에 나타내었다. 그림 7 상단의 GUI, Session Manager, Splitter등은 스트리밍 서버를 구성하는 클래스이며, 세로축은 시간의 흐름을 나타낸다. 그리고 스트리밍 서버에서 Splitter와 RTP Sender는 독립된 쓰레드로 동작하도록 설계되었다.

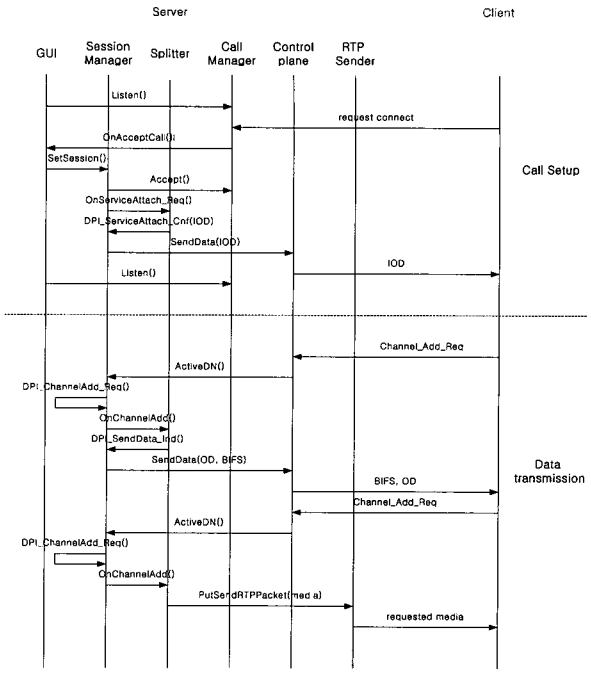


그림 7. 스트리밍 서버 동작 흐름도
Fig. 7. Event trace diagram of a streaming server

2.1.1 서버의 Call setup 과정

서버가 실행되면, Call Manager가 생성되어 대기(listen) 상태가 되며, 클라이언트로부터 접속 요청이 있을 경우 이를 GUI로 알린다. GUI는 Session Manager를 비롯한 필요한 모듈들을 생성하고 새로운 세션을 설정하게 되는데, 이때 Session Manager는 Splitter로부터 초기객체기술자를 넘겨받아 클라이언트 측으로 전송한다.

2.1.2 서버의 Data transmission 과정

서버는 클라이언트 측에서 초기객체기술자를 분석한 후 서버측으로 전송하는 BIFS와 객체기술자 요구 메시지를 Control Plane을 통해 수신하고 Session Manager로 전달한다. 이 요구 메시지에는 트랙 ID와 채널 ID 등이 포함되어 있다. Session Manager는 메시지의 종류를 분석한 후, 트랙 ID와 채널 ID를 이용해 Splitter에 BIFS와 객체기술자 스트림을 요구하게 된다. 이 요구에 따라 Splitter는 SL 패킷화된 객체기술자와 BIFS 정보를 Session Manager로 전달하고, Session Manager는 BIFS와 객체기술자를 Control Plane을 통하여 클라이언트 측으로 전달한다.

클라이언트에서는 BIFS와 객체기술자정보를 분석한 후,

현재 프리젠테이션 되어야 할 미디어의 전송을 요구하게 되는데, 서버측에서는 Control Plane을 통해 이 메시지를 수신하여 Session Manager로 전달하고, Session Manager는 메시지를 분석한 후, 스트리밍이 필요한 미디어의 트랙 ID와 채널 ID, 포트번호를 Splitter로 전달한다. Splitter는 요구한 트랙 ID에 해당하는 스트림 종류를 판단하여, 실시간 스트리밍이 요구되는 미디어 데이터인 경우, RTP Sender를 통해 클라이언트측으로 전송하고, TCP를 이용하는 미디어의 경우 SL 패킷화된 미디어 스트림을 Session Manager로 전달한다.

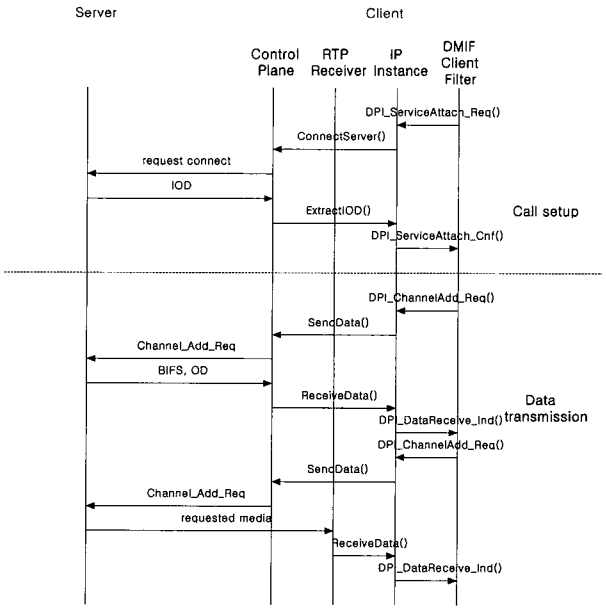


그림 8. 클라이언트 동작 흐름도
Fig. 8. Event trace diagram of a client

2.2 클라이언트 동작

그림 8에서는 클라이언트의 동작과정을 설명한다. 앞의 IV장 2.1절에서와 마찬가지로, 그림 8의 상단에 있는 Control Plane, RTP Reciever, IP Instance 등은 클라이언트에서 구현된 클래스이며, 세로축은 시간의 흐름을 나타낸다. 전달계층과 IM1-2D 플레이어와의 통신은 DMIFClientFilter 클래스를 통해서 이루어지며, Control Plane과 RTP Reciever는 독립된 쓰레드로 동작한다.

2.2.1 클라이언트의 Call setup 과정

사용자가 스트리밍 서버의 IP 주소를 입력하고 접속을 요청하면, DMIFClientFilter는 IP Instance에 서비스 생성을 요구하고, IP Instance는 Control Plane을 통해 서버의 well-known 포트로 접속 요청을 한다. 이후 서버측에서 초기객체기술자를 전송하면, 클라이언트의 IP Instance에서는 초기객체기술자를 분리하여 DMIFClientFilter로 전달한다.

2.2.2 클라이언트의 Data transmission 과정

IP Instance가 초기객체기술자를 DMIFClientFilter로 전달하면, 초기객체기술자를 분석한 IM1-2D 플레이어는 DMIFClientFilter를 통하여 BIFS와 객체기술자를 요구하게 되는데, IP Instance는 DMIFClientFilter가 요구하는 BIFS와 객체기술자의 트랙 ID와 채널 ID를 Control Plane을 통하여 서버측으로 전달함으로써 채널 생성을 요구하게 된다. 서버측에서 객체기술자 정보와 BIFS 정보를 전송하면 클라이언트는 Control Plane을 통해 수신한 후, IP Instance를 통해서 DMIFClientFilter로 BIFS와 객체기술자를 전달한다. IM1-2D 플레이어는 BIFS와 객체기술자 분석 후 프리젠테이션에 필요한 미디어들을 DMIFClientFilter를 통해 요구하게 되고, IP Instance는 DMIFClientFilter가 요구하는 미디어 객체의 트랙 ID와 채널 ID, 그리고 미디어 데이터를 수신하기 위한 포트번호 등을 Control plane을 통해 서버측으로 전달한다. 이후 프리젠테이션에 필요한 미디어들이 스트리밍되며, 클라이언트는 RTP Receiver 또는 Control Plane을 통해 미디어 데이터를 받아 들인 후, DMIFClientFilter로 전달함으로써 IM1-2D 플레이어를 통한 프리젠테이션을 가능하게 한다.

V. 실행 결과

스트리밍 실험은 서울(211.60.14.81)과 수원(163.180.123.113) 사이에서 이루어졌으며, 그림 9에 서버의 실행결과를 나타내었다. 스트리밍 서버의 사용자 인터페이스는 클라이언트의 접속 상황 및 스트리밍 진행 상황을 표시한다. 그림 10과 11에는 클라이언트의 실행 결과를 나타내었다. 그림 10은 장면을 구성하는 모든 미디어 데이터를 RTP를 이용하여 스트리밍한 경우이며, 그림 11은 미디어 데이터 중, 하나의 패킷으로 전송되어 지속적으로 렌더링되는 미디어 데이터는 TCP, 그 외 실시간 스트리밍이 요구되는 미디어 데이터는 RTP를 사용한 경우이다. 그림 10에서 RTP만으로 전송했을 경우 JPEG 이미지인 지도의 손실이 발생한 경우를 나타내었으며, 망 상태에 따라 지도

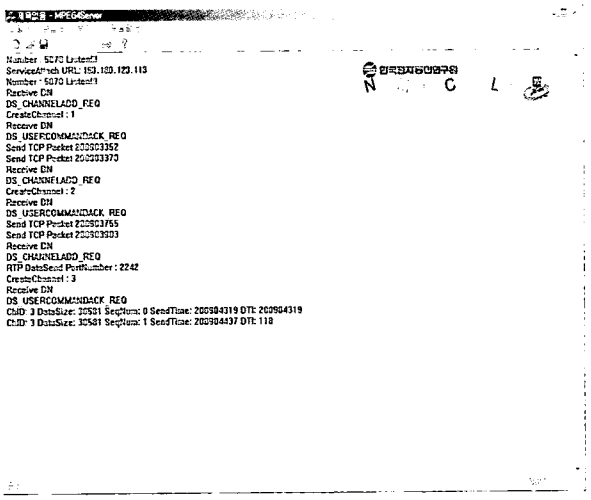


그림 9. 서버 실행 결과
Fig. 9. Result of server execution

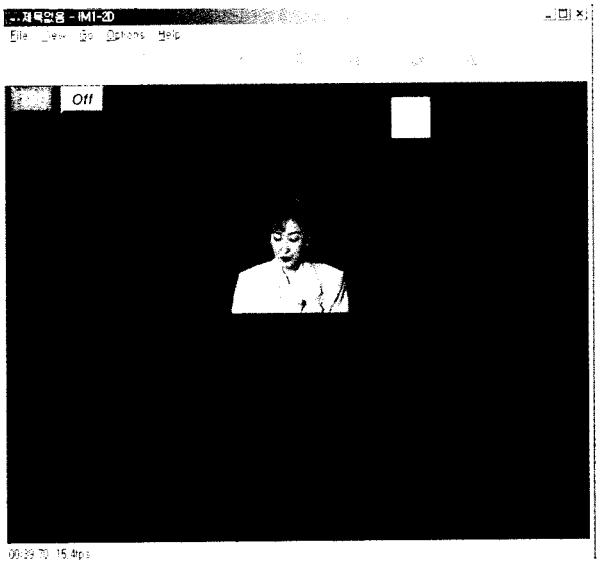


그림 10. 클라이언트 실행 결과 (RTP로만 전송한 경우)
Fig. 10. Result of client execution (using only RTP)

이미지 뿐만 아니라, 버튼이나 날씨 이미지들의 손실도 종종 발생한다. 이와 같은 패킷 손실은 전송되는 패킷의 크기에 비례하는 경향이 있다[10].

실험 결과에서 볼 수 있듯이, 한 번 전송된 후 계속 렌더링을 유지하는 정지영상의 손실이 발생했을 경우는 장면 구성에 매우 치명적일 수 있다. 따라서 정지영상과 텍스트 객체 와 같이 한번의 전송으로 전송이 완료되는 미디어 데이터는 TCP로 전송하는 것이 장면 구성 시 우수한 성능을 나타낸다.

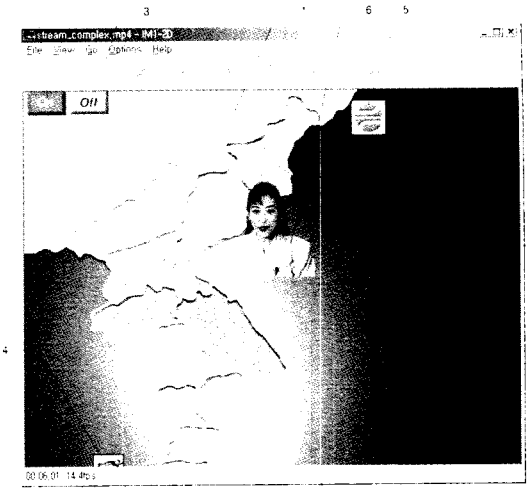


그림 11. 클라이언트 실행 결과(RTP와 TCP를 같이 사용한 경우)
Fig. 11. Result of client execution (using RTP and TCP)

실험에 사용된 컨텐츠는 MPEG-4 비디오 객체와, G.723.1 오디오 객체, JPEG 이미지 객체로 장면을 구성한 MP4 파일이 사용되었다. 그림 11의 클라이언트 실행 결과에서, 2번과 3번인 ON/OFF 버튼 이미지는 터치센서(TouchSensor)로 동작하는 객체로, ON 버튼 클릭 시 6번인 비디오 객체와, 오디오 객체가 서버로부터 스트리밍 되어 장면에 렌더링되고 OFF 버튼 클릭시 비디오, 오디오 객체가 장면에서 사라지도록 라우팅되어 있다. 5번 로고(logo)는 타임센서(TimeSensor)와 벨류에이터(Valuator) 노드를 사용하여 시간에 따라 4장의 JPEG 이미지가 교대로 렌더링 된다. 4번 날씨 이미지는 타임센서를 사용하여 0초부터 10초 사이에만 렌더링 되도록 설정되었다.

VI. 결론

기존의 스트리밍 시스템은 오디오와 비디오만으로 구성된 컨텐츠를 스트리밍하였으므로, 사용자와의 대화가 불가능했다. 이에, 본 논문에서는 사용자와의 대화가 가능한 MPEG-4 기반의 대화형 컨텐츠를 IP망을 통해 전송하기 위한 스트리밍 서버와 클라이언트 시스템 구조를 설계하고 그 구현 결과를 보였다.

MPEG-4 컨텐츠를 효율적으로 스트리밍 하기 위해서, MPEG-4의 구성 객체들을 각각 어떤 프로토콜로 전송해야 할 지를 결정해야 하며, 그 결정에 맞는 스트리밍 시스템을 구현해야 한다. 본 논문에서 제시한 스트리밍 시스템에서는, 미디어의 종류에 따라 TCP와 RTP로 구분하여 전송하

는데, 초기객체기술자, BIFS, 객체기술자와 한 번의 전송으로 렌더링을 지속하는 정지영상이나, 텍스트 등과 같은 미디어 객체는 TCP를 사용하여 전송하고 실시간 스트리밍이 요구되는 오디오, 비디오와 같은 미디어 정보는 RTP를 사용하여 전송함으로써 장면 구성에 적합하도록 설계하였다.

현재 구현된 스트리밍 시스템은 인트라넷(intranet)과 인터넷(internet)에서 모두 동작 가능하지만, 패킷 손실이 많은 인터넷에서는 비디오, 오디오의 열화가 발생한다. 이런 문제점을 보완하기 위해 향후 QoS 관리를 위한 기능들이 고려되어야 할 것이다. 그리고 동기화 정보를 이용한 미디어들 간의 동기를 고려해야 할 것이다.

참 고 문 헌

[1] ISO/IEC JTC1/SC29/WG11 N3536, "Overview of the MPEG-4 Standard," Jul. 2000.

[2] 고성제, 김성욱 역, "MPEG-4의 세계", 영풍문고, 2000.

[3] ISO/IEC JTC1/SC29/WG11 N2506, "Information technology-very-low bitrate audio-visual coding-Part 6: Delivery Multimedia Integration Framework(DMIF)," 1999.

[4] ISO/IEC JTC1/SC29/WG11 N3713, "Information technology-Coding of audio-visual objects-Part 6: Delivery Multimedia Integration Framework(DMIF)," 2000.

[5] ISO/IEC 14772-1:1997 "The Virtual Reality Modeling Language," 1997.

[6] ISO/IEC JTC1/SC29/WG11 N2501 "Information technology-Coding of audio-visual objects-Part 1: Systems," 1999.

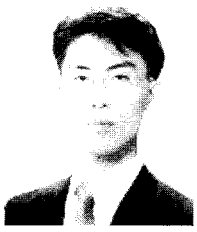
[7] RFC1889, "RTP: A Transport Protocol for Real-Time Applications," Jan. 1996.

[8] 김재균, "영상통신시스템", 영지문화사, 2000.

[9] ISO/IEC JTC1/SC29/WG11 MPEG97/M3111, "APIs for Systems Software Implementation," Nov. 1997.

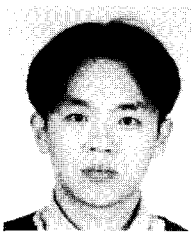
[10] 최지훈, 이상조, 서덕영, "Internet QoS 적응적 실시간 멀티미디어 전송에 대한 연구", *한국통신학회 하계종합학술발표회 논문집*, vol(하), pp.1200-1203, 2000. 7.

저 자 소 개



김 현 철

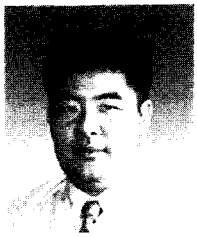
1998년 2월 : 경희대학교 전자공학과 졸업(공학사)
2000년 8월 : 경희대학교 대학원 전자공학과 졸업(공학석사)
2000년 7월~현재 : 한국전자통신연구원 연구원
주관심분야 : 멀티미디어 통신



민 승 홍

2001년 2월 : 경희대학교 전자공학과 졸업(공학사)
2001년 3월~현재 : 경희대학교 대학원 전자공학과 석사과정
주관심분야 : 멀티미디어 통신

저 자 소 개



서 덕 영

1980년 2월 : 서울대 핵공학과 졸업(공학사)
1986년 2월 : 미국 Georgia Tech. 핵공학 졸업(공학석사)
1990년 6월 : 미국 Georgia Tech. 전기공학과 졸업(공학박사)
1990년~1992년 : 상공부 생산기술 연구원, HDTV 연구개발단 선임연구원
1992년 3월~현재 : 경희대학교 전자정보학부 교수
주관심분야 : networked video

김 규 헌

1989년 2월 : 한양대학교 전자공학과 졸업(공학사)
1992년 9월 : 영국 University of Newcastle 전기전자공학과 졸업(공학석사)
1996년 7월 : 영국 University of Newcastle 전기전자공학과 졸업(공학박사)
1996년~1997년 : 영국 University of Sheffield reserach fellow
1997년~현재 : 한국전자통신연구원 선임연구원
주관심분야 : 영상처리, 대화형 방송 시스템

김 진 웅

1981년 2월 : 서울대학교 전자공학과 졸업(공학사)
1983년 2월 : 서울대학교 전자공학과 졸업(공학석사)
1993년 8월 : 미국 Texas A&M University 전기공학과 졸업
1983년~현재 : 한국전자통신연구원 책임연구원
주관심분야 : DSP, 멀티미디어 시스템, 대화형 방송 시스템