

# RTSP에 기반한 MPEG-4 스트리밍 시스템

## (A RTSP-based MPEG-4 Streaming System)

이 상 현 <sup>†</sup> 이 종 민 <sup>\*\*</sup> 차 호 정 <sup>\*\*\*</sup> 박 병 준 <sup>\*\*\*\*</sup>  
 (Sanghyun Lee) (Jongmin Lee) (Hojung Cha) (Byungjoon Park)

**요약** 본 논문은 RTSP에 기반한 MPEG-4 스트리밍 시스템의 구현에 대해 기술한다. MPEG-4 스트리밍 시스템은 콘텐츠 저작 도구, 서버, 클라이언트로 구성된다. 기존에 제공되는 멀티플렉싱 방법에 의한 콘텐츠 스트리밍시 발생하는 지연 시간 및 서버 부하 등의 문제점을 해결하기 위한 새로운 멀티플렉싱 방법을 제안한다. 구현된 시스템은 전송 제어 프로토콜로 RTSP를 사용하여 DMIF를 구현하기 위한 새로운 전송 계층의 구현에 따른 부담을 줄였으며, 로컬 재생뿐만 아니라 스트리밍 재생시에도 사용자 상호 작용 기능을 지원한다. 멀티플렉싱 기능의 지원 여부에 따른 효율성을 서버의 성능 분석을 통해 검증하고 제안된 MPEG-4 스트리밍 시스템이 올바르게 작동함을 보인다.

**키워드** : RTSP, MPEG-4, 스트리밍 시스템

**Abstract** This paper describes the implementation of a RTSP-based MPEG-4 streaming system. The implemented system consists of content authoring tool, streaming server and streaming client. The existing MPEG-4 multiplexor has been modified with improvement to generate a file format with which the server load and streaming delay are reduced. The streaming system uses RTSP, instead of DMIF, as its communication protocol. With RTSP, the server provides extensive control functionalities as in DMIF and its implementation effort is greatly simplified. The implemented system supports the user interactivity for both streaming and local file playback. The effectiveness of the proposed multiplexing scheme has been validated with the various experiments on the fully functional server.

**Key words** : RTSP, MPEG-4, streaming system

### 1. 서론

최근 컴퓨터와 통신망 관련 기술의 발달로 인해 주문형 비디오 서비스(VOD)가 활성화되고 있으며 TV와 같은 단방향 서비스뿐만 아니라 사용자와 상호 작용이 가능한 양방향 서비스에 대한 요구와 필요성이 증가하고 있다. 네트워크를 통한 양방향 멀티미디어 서비스를 위

해서는 네트워크 대역폭의 한계와 사용자와의 상호 작용에 따른 서비스의 특성을 고려하여야 하며, 이로 인해 낮은 대역폭과 높은 비디오 품질을 제공하는 MPEG-4 [1]가 주목받고 있다.

MPEG-4는 동영상 관련 신호 압축 및 복원에 관한 기술 표준으로 음성이나 영상과 같은 멀티미디어 자료를 객체로 관리한다. 또한, BIFS(Binary Format for Scene)[2]를 규정하고 있어 사용자 상호 작용성을 지원한다. 플레이어는 재생시 객체에 이벤트가 발생하였을 경우 BIFS를 참조하여 객체를 이동, 삭제, 추가하거나 장면 전체를 갱신할 수 있다. MPEG-4는 동영상 압축에 관련된 분야 이외에 멀티미디어 자료의 전송에 관한 분야까지 고려를 하고 있다. 멀티미디어 자료의 전송과 관련된 부분을 DMIF(Delivery Multimedia Integrated Framework)[3]라 정의한다. DMIF는 DAI(DMIF-Application Interface)라는 응용 프로그램과의 인터페이스를 제공하며 이를 통해 응용 프로그램은 DMIF와 상

· 본 연구는 정보통신기술연구지원사업(2001-07603)과 과학재단 특장 기초연구사업(R01-2002-000-00141-0)에 의해 부분적으로 지원되었음.

<sup>†</sup> 비회원 : 필링크 연구원

shlee@feelingk.com

<sup>\*\*</sup> 비회원 : 연세대학교 컴퓨터과학과

jmlee@cs.yonsei.ac.kr

<sup>\*\*\*</sup> 종신회원 : 연세대학교 컴퓨터과학과 교수

hjcha@cs.yonsei.ac.kr

<sup>\*\*\*\*</sup> 비회원 : 광운대학교 컴퓨터과학과 교수

bjpark@cs.kwangwoon.ac.kr

논문접수 : 2002년 2월 18일

심사완료 : 2002년 10월 8일

관없는 투명한 통신 방식을 제공받는다. DMIF는 멀티미디어 스트리밍을 관리하기 위한 세션 수준의 프로토콜을 정의하며 종단간의 QoS와 하위 전송 계층에서의 효율적인 전송을 위한 방법 등을 정의하고 있다. 이러한 DMIF의 규약에 따른 새로운 전송 계층의 구현은 복잡도가 크기 때문에 기존에 정의된 RTSP[4]와 같은 프로토콜을 사용하여 MPEG-4의 전송 계층을 구현하려는 연구가 다양하게 진행 중이다. RTSP는 스트리밍을 위한 국제 표준으로서 원격의 미디어 객체에 대해 VCR에서 제공하는 것과 같은 재생, 정지, 순간정지, 빠른 재생과 같은 다양한 제어기능을 제공한다. 또한 SDP(Session Description Protocol)[5]를 사용하여 세션 관리의 기능도 제공하고 있다. RTSP의 이러한 기능은 DMIF가 정의하는 미디어 컨트롤 기능과 세션 관리 기능을 충분히 제공할 수 있다.

MPEG-4 스트리밍에 관한 연구의 대표적인 예로는 IBM[6]과 AT&T[7], 콜롬비아 대학[8] 등에서의 연구가 있다. IBM은 MPEG-4를 이용하여 인트라넷상에서의 원격 교육을 할 수 있는 시스템을 개발하였다. 이 시스템의 서버로는 기존의 IBM 비디오 서버를 확장하여 작성되었고 클라이언트는 순수 자바로만 작성되었다. 사용되는 미디어 데이터로는 음성과 영상 스트림이 캡슐화된 MPEG-4 시스템 화일을 사용하였다. 프로토콜로는 RTSP와 RTP를 사용하였으며 멀티캐스트를 지원하여 원격지의 사용자가 임의의 시간에 원격 강의에 참가할 수 있도록 하였다. AT&T는 MPEG-4의 데이터 스트림과 제어 스트림의 효율적인 전송을 위해 MPEG-4 데이터 스트림 전송을 위한 RTP 페이로드 형식을 제안하고 구현하였다. 서버에서의 전송은 MPEG-4를 구성하는 여러 미디어 스트림을 개별적인 각각의 RTP 스트림으로 전송하도록 하였으며 클라이언트는 IM-1[9] 플레이어를 사용하였다. 콜롬비아 대학은 사용자와 상호작용이 가능한 MPEG-4 스트리밍 시스템을 구현하였다. 클라이언트는 IM-1 플레이어를 사용하였으며 전송 프로토콜로는 Xbind사에서 개발한 XDMIF API[10]를 사용하여 네트워크상에서 재생이 가능하도록 하였다. 서버에서의 스트림은 개별적인 스트림을 멀티플렉싱하여 송신하며 사용자와의 상호 작용을 위하여 독자적인 매커니즘을 구현하였다.

기존의 MPEG-4 스트리밍에 관한 연구들은 사용자와의 상호 작용 지원, 사용되는 미디어 스트림의 멀티플렉싱 지원 등의 문제점을 갖고 있다. IBM과 AT&T의 MPEG-4 스트리밍 시스템은 사용자와의 상호 작용 기능을 제공하지 못하고 있으며, 스트리밍시 스트림의 멀

티플렉싱 기능도 제공하지 못하고 있다. 콜롬비아 대학에서 구현한 시스템은 사용자와의 상호 작용을 지원하고 있으나 전송 계층인 DMIF를 Xbind사의 API를 사용하고 있다. 기존에 제공되는 멀티플렉서에 의해 생성된 시스템 화일을 스트리밍 서비스할 경우 멀티플렉싱되는 순서에 의한 처리와 사용자와의 상호 작용에 의한 이벤트 처리 과정에서의 미디어 스트림간의 이동을 처리하기 위한 지연 시간 증가 및 서버 부하 등의 문제점이 있다.

본 논문은 RTSP에 기반하여 MPEG-4에서 정의하는 사용자와의 상호 작용이 가능한 스트리밍 시스템의 구현에 대하여 기술한다. 기존의 멀티플렉싱 방법으로 생성된 MPEG-4 시스템 화일을 스트리밍 서비스에 사용할 경우에 발생 가능한 지연 시간 증가 및 서버 부하 등의 단점에 대해 살펴보고 이러한 단점을 해결할 수 있는 새로운 멀티플렉싱 방법을 제안하며 실험을 통해 이를 검증한다. 제안된 멀티플렉싱 방법으로 생성된 시스템 화일을 효율적으로 전송할 수 있는 서버 모델의 구조와 구현에 대하여 설명하고 실험을 통해 MPEG-4 데이터 전송시 스트리밍 멀티플렉싱의 효율성에 대하여 기술한다. 또한, 전송 계층인 DMIF를 표준 스트리밍 프로토콜인 RTSP를 사용하여 구현함으로써 기존 프로토콜의 활용성과 DMIF를 RTSP로서 대체 구현할 수 있음을 보이고 스트리밍시에도 사용자와의 상호 작용이 가능함을 보인다.

논문의 구성은 다음과 같다. 2 장에서는 MPEG-4 시스템의 개요와 저작 도구, 스트리밍 서버, 클라이언트의 구조에 대해 살펴본다. 특히, 새로운 멀티플렉싱 방법, RTSP로 DMIF를 구현하는 방법에 대해 설명하고, 사용자 상호 작용을 지원하는 시스템의 구조 및 구현에 대해 기술한다. 3 장에서는 성능 분석을 통해 제안된 멀티플렉싱 방법을 검증하며 구현된 시스템이 올바르게 동작함을 보이고, 4 장에서 결론을 맺는다.

## 2. 시스템 구현

다음은 MPEG-4 스트리밍 시스템의 전체 구조와 시스템의 동작 과정을 기술한다. 또한, MPEG-4 스트리밍을 위한 기존의 방법들의 문제점을 알아보고 이를 해결하기 위한 새로운 방법을 제안한다. MPEG-4 스트리밍 시스템의 구조에 대해 살펴보고, RTSP를 사용하여 DMIF를 구현하기 위한 방법 및 사용자와의 상호 작용 기능을 제공하면서 스트리밍 하기 위한 방법들을 살펴본다. 또한, 구현된 MPEG-4 스트리밍 시스템을 콘텐츠 저작 단계, 스트리밍 서버, 스트리밍 클라이언트의 세

부분으로 세분화하여 각 부분들의 구조 및 구현에 대해 기술한다.

2.1 시스템 구조

MPEG-4 스트리밍 시스템은 멀티미디어 스트림들을 멀티플렉싱된 MPEG-4 시스템 화일로 만들어내는 콘텐츠 저작 부분과 사용자의 요청에 따라 MPEG-4 개별 스트림을 네트워크를 통해 클라이언트로 전송하는 스트리밍 서버, 그리고 서버로부터 네트워크를 통해 데이터를 전송받아 이를 재생하고 사용자 이벤트를 다시 서버로 전송하는 클라이언트로 구성된다. 구현된 MPEG-4 스트리밍 시스템 구조는 그림 1과 같다.

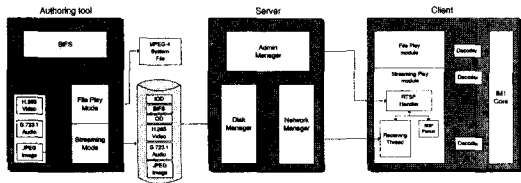


그림 1 시스템 구조

콘텐츠 저작 단계에서는 스트리밍을 위한 멀티미디어 데이터를 MPEG-4 형식으로 생성한다. MPEG-4 형식의 화일은 여러 개의 미디어 스트림들로 구성되며 콘텐츠 저작 단계는 이러한 여러 개의 미디어 스트림들을 멀티플렉싱된 MPEG-4 시스템 화일로 생성하는 과정이다. 서버는 생성된 화일을 네트워크를 통해 클라이언트로 전송하는 기능을 수행하며, 클라이언트는 서버로부터 전송 받은 멀티미디어 데이터를 BIFS에 근거하여 원본과 같이 재생하고 사용자에 의해 이벤트가 발생했을 경우 이를 네트워크를 통해 서버로 전송하게 된다. MPEG-4에서 멀티미디어 자료의 전송과 관련된 부분을 정의한 DMIF를 RTSP로 구현하기 위한 방법은 다음과 같다.

2.2 RTSP에 기반한 DMIF

DMIF는 멀티미디어 스트리밍을 관리하기 위한 세션 수준의 프로토콜을 정의하며 종단간의 QoS와 하위 전송 계층에서의 효율적인 전송을 위한 방법 등을 정의하고 있다. 이러한 DMIF의 규약에 따른 새로운 전송 계층의 구현은 구현의 복잡도가 크고, 구현된 전송 계층의 효율성 검증에 많은 시간과 노력이 필요하다. 따라서, 기존의 여러 논문에서 효율성이 입증된 RTSP를 사용하여 DMIF를 구현함으로써, 이러한 구현의 복잡도 및 효율성 검증에 따른 시간과 부담을 줄일 수 있다.

RTSP를 사용하여 DMIF의 구현을 만족시키기 위해서는 RTSP를 사용하여 DMIF가 제공하는 API인 DAI

의 의미에 적합하게 구현되었는지를 보여줄 수 있어야 한다. DAI는 응용 프로그램에게 네트워크 전송과정에 대한 투명성을 제공하기 위해 정의된 인터페이스이다. 이를 통해 응용 프로그램은 DMIF와 상관없는 투명한 통신 방식을 제공받는다. 그림 2는 구현된 RTSP와 DAI간의 의미가 서로 적합함을 보여주고 있다.

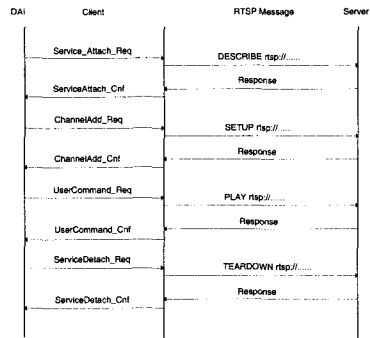


그림 2 DAI와 RTSP

DMIF 사용자들의 서비스 관리를 위한 서비스 관련 DAI인 ServiceAttach\_Req와 ServiceAttach\_Cnf는 RTSP의 서비스 정보를 얻기 위한 명령인 DESCRIBE 명령과 그 응답으로 대처할 수 있다. 채널을 관리하기 위한 DAI인 ChannelAdd\_Req와 ChannelAdd\_Cnf는 RTSP의 서비스 요청 메시지인 SETUP 메시지와 그 응답 메시지로, 데이터 전송을 위한 DAI인 UserCommand\_Req와 UserCommand\_Cnf는 RTSP의 미디어 자료 요청 명령인 PLAY 메시지와 그 응답 메시지로, 서비스 종료를 위한 DAI인 ServiceDetach\_Req와 ServiceDetach\_Cnf는 RTSP의 서비스 종료 메시지인 TEARDOWN 메시지와 그 응답 메시지로 대처 가능하다. DMIF가 제공하는 인터페이스인 DAI를 이러한 방법으로 RTSP 메시지로 대응시킴으로써 RTSP를 사용하여 DMIF를 구현할 수 있으며, DMIF 규약에 의한 새로운 전송 계층의 구현에 따른 복잡성과 효율성 검증에 대한 시간과 부담을 줄일 수 있다.

MPEG-4 시스템의 가장 큰 특징은 사용자와의 상호 작용을 지원한다는 점이다. 사용자와의 상호 작용이란 재생되는 화면에 대해 마우스 클릭과 같은 이벤트가 발생했을 때 새로운 스트림이 재생, 선택된 미디어 객체가 이동, 삭제되는 것들을 말한다. MPEG-4 시스템에서 정의하고 있는 사용자 상호 작용은 다음과 같다.

- Insert : 현재의 장면에 특정 미디어 객체를 삽입한다.

- Delete : 현재의 장면에서 특정 미디어 객체를 제거한다.
- Replace : 현재의 장면에서 특정 미디어 객체를 제거하고 대신 다른 객체로 대체한다.
- Scene Replace : 현재의 장면을 모두 초기화하고 새로운 장면으로 대체한다.

그림 3은 클라이언트 내에서의 이러한 사용자 이벤트 처리과정을 보여준다. 사용자의 마우스 클릭 등에 의해서 화면 위에 이벤트가 발생하면 이벤트 처리를 담당하는 Presenter는 이벤트가 발생한 위치를 이벤트 배열에 저장한다. Presenter는 화면의 각 구역과 그 구역에 해당하는 객체들의 정보를 미리 알고 있다. Presenter는 매 루프마다 이벤트 처리를 감지하여 이벤트가 발생한 위치에 기준하여 그 위치에 해당하는 객체의 이벤트 핸들러에게 위치 정보와 발생한 이벤트에 대한 정보를 넘겨준다. 실제 이벤트를 처리할 미디어 객체들은 BIFS에 의해 구성된 장면 구성 트리의 일부분으로 존재한다. 장면구성 트리에 있어서 중요한 점은 트리의 각 노드들은 장면을 구성하는 미디어 객체들이며, 각 노드들에는 객체의 특성을 기술하는 부분과 함께 이벤트 처리만을 담당하는 'Proxy node'가 같이 존재한다는 점이다. Presenter로부터 이벤트가 발생했음을 알리는 호출이 오면 장면 구성 트리의 해당 노드는 Proxy 노드 객체를 통해 BIFS 디코더에 접근한다. BIFS 디코더는 발생한 이벤트의 종류에 따라 현재의 화면에 대해 Insert, Delete, Replace, Scene Replace 이벤트를 수행한다.

스트리밍 재생시의 상호 작용은 크게 두 가지로 나눌 수 있다. 이벤트에 의해 새로운 스트림에 대한 요청이 발생하지 않는 클라이언트 측의 상호 작용과 새로운 스트림의 요청이 발생하는 서버 측의 상호 작용이다. 특정 이미지에 마우스 포인터를 옮겼을 때, 그 이미지 위에 지정된 텍스트가 겹쳐 보이는 것은 클라이언트 측 상호 작용의 좋은 예라 할 수 있다. 스트리밍 재생시 서버 측 상호 작용 예는 클라이언트에서 원하는 스트림이 클라이언트의 로컬 지역에 존재하지 않을 경우 서버와의 통신에 의해 해당 스트림을 가져와야 하는 경우를 들 수 있다. 이벤트 발생에 의해 서버에게 새로운 스트림을 요청할 경우 구현된 RTSP 서비스 모듈을 통해 이루어지게 된다. BIFS에서는 외부의 MPEG-4 객체에 대한 링크 역할을 하는 'Inline'이라는 객체가 존재한다. 특정 객체에 이벤트가 발생하여 새로운 스트림으로 대체되거나 새로운 스트림이 삽입될 경우 Inline 객체가 HTML 태그의 링크와 동일한 역할을 하여 새로운 스트림에 대한 위치를 알려주게 된다. 본 논문에서는 이러한 Inline

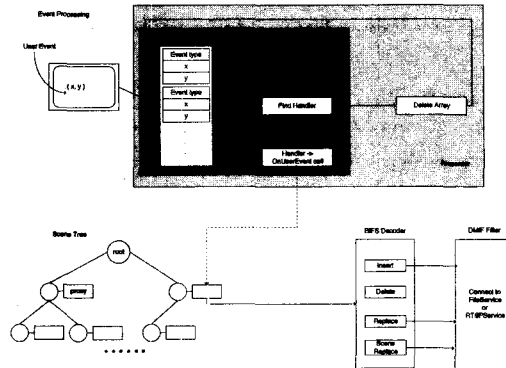


그림 3 클라이언트에서의 이벤트 처리과정

객체에 RTSP URL 정보를 지정함으로써 스트리밍시에도 RTSP를 통하여 서버와의 상호 작용이 가능하도록 하였다. 이벤트 발생에 의해 새로운 스트림 요청이 생기면 BIFS 디코더는 해당 객체의 Inline 객체에서 RTSP URL을 얻는다. 얻어진 RTSP URL과 함께 다시 DMIF 필터를 통해 클라이언트의 RTSP 서비스 모듈을 적재시키고 앞서 설명했던 방식에 따라 새로운 스트림을 요청하게 된다. 이러한 방법으로 구현된 서버와 클라이언트는 MPEG-4 시스템에서 정의하고 있는 모든 사용자 상호 작용을 지원한다. 구현된 MPEG-4 스트리밍 시스템을 콘텐츠 저작, 서버, 클라이언트 세 부분으로 세분화하여 살펴보면 다음과 같다.

### 2.3 콘텐츠 저작

콘텐츠 저작 단계에서 가장 중요한 부분은 여러 개의 미디어 스트림들을 멀티플렉서를 통해 하나의 MPEG-4 시스템 화일로 만드는 부분이다. MPEG-4 시스템 그룹에서는 MPEG-4 시스템 화일을 만들 수 있는 멀티플렉싱 도구를 제공하고 있다. 제공되는 멀티플렉서에 의해 시스템 화일이 구성될 경우, BIFS에 지정된 순서대로 화일이 멀티플렉싱되는 것이 아니라 단순히 Mux 스크립트 화일에 기술된 순서대로 화일이 멀티플렉싱된다. 따라서 시간상으로 화일의 가장 앞부분에 위치해야 할 데이터가 Mux 스크립트에 기술된 순서에 의해 가장 뒷부분에 위치할 수도 있다. 이러한 방법으로 생성된 MPEG-4 시스템 화일은 로컬 화일 재생시에는 문제가 없으나 스트리밍시에는 문제점을 갖게 된다. BIFS에 지정된 순서대로 멀티플렉싱된 시스템 화일을 스트리밍 플레이어에 사용할 경우 시간상으로는 가장 먼저 재생되어야 하지만 화일의 뒷부분에 있는 데이터를 재생하기까지 지연이 발생할 수 있기 때문이다.

시스템 화일을 스트리밍에 사용할 경우 생길 수 있는

또 다른 문제는 사용자의 이벤트에 의하여 서버로부터 새로운 스트림을 요청할 경우 서버에서 하나의 멀티플렉싱된 스트림에서 임의의 위치 이동을 해야 하는 부담이 있다는 점이다. MPEG-4 시스템 화일은 일반적인 MPEG 동영상 데이터처럼 하나의 프레임 단위로 저장되는 것이 아니라 Access Unit이라는 작은 단위로 저장되고 또 이러한 Access Unit들이 모여 SL(Sync Layer) 패킷이라는 형태로 캡슐화되기 때문에 일반적인 동영상 데이터 내에서의 이동보다 서버에서의 부담을 증가시킬 수 있다.

이러한 기존 멀티플렉싱 방법의 문제점을 해결하기 위해 본 논문에서는 기본적인 MPEG-4 시스템 화일 생성 기능을 제공하는 동시에 스트리밍 모드로 실행 될 때에는 시스템 화일의 단점을 줄이기 위하여 하나의 멀티플렉싱된 화일을 생성하지 않고 각각의 미디어 데이터를 개별적인 MPEG-4 화일로 구성하는 새로운 멀티플렉싱 방법을 제안한다. 개별적으로 생성된 화일을 사용자의 요청에 따라 필요한 미디어 데이터만 전송하면 되기 때문에 지연이 생기지 않고 또한 이벤트 발생으로 인해 새로운 스트림을 요청할 때에도 화일 내에서의 임의의 위치 이동이 필요 없다. 각각의 미디어 데이터를 개별적인 MPEG-4 화일로 구성하는 새로운 멀티플렉서는 MPEG-4 시스템 그룹에서 제공하는 멀티플렉서를 수정하여 구현하였다. 그림 4는 본 논문에서 새롭게 제안된 멀티플렉싱 과정을 보여준다.

MPEG-4 시스템 화일의 저작 과정은 다음과 같다. 먼저 장면 구성 정보를 담고 있는 BIFS 스크립트 화일을 생성한다. BIFS 스크립트 화일은 VRML(Virtual Reality Model Language) 스펙을 기준으로 하여 MPEG-4 시스템에 정의되어 있다. BIFS는 장면을 구성하는 객체의 시간적, 공간적 정보와 속성들에 대한 정보로 구성되며, 각각의 객체들은 트리를 구성하는 하나의 노드로써 표현된다. 또한, 각 미디어 데이터의 특징

과 클라이언트가 필요로 하는 디코더의 구성에 관한 정보를 기술하는 멀티플렉서 스크립트 화일을 생성한다. 본 논문에서는 일반 텍스트 에디터를 사용하여 수동으로 BIFS 스크립트 화일과 멀티플렉서 스크립트 화일을 작성하였다. 그림 5는 작성된 BIFS 스크립트 화일과 멀티플렉서 스크립트 화일의 예를 보여준다.

```

BIFS Script
DEF Node1 Transform1
  position 100 100
  scale 1.0 1.0
  children 0
END

MPEG4SystemAccessUnit
  elementaryStreamID 1
  streamRate 1.00 1.00
  startTS 0
  endTS 0
  ...

Mux Script
MPEG4SystemAccessUnit
  elementaryStreamID 0
  streamRate 1.00 1.00
  startTS 0
  endTS 0
  ...
    
```

그림 5 BIFS 스크립트와 멀티플렉서 스크립트의 예

작성된 BIFS 스크립트 화일과 멀티플렉서 스크립트 화일, 미리 준비된 비디오, 오디오 및 이미지 화일들은 수정된 멀티플렉서로 입력되어 멀티플렉싱 과정을 거치게 되며, 멀티플렉싱 과정을 통해 컨텐츠의 복잡성을 나타내는 프로파일과 레이블 정보 등의 클라이언트를 초기화하는데 필요한 정보를 가지고 있는 IOD(Initial Object Descriptor) 화일, 전송될 각 스트림의 정보를 가지고 있는 OD(Object Descriptor) 화일, 장면 구성 정보를 가지고 있는 바이너리 형태의 BIFS 데이터 화일, 실제 장면을 구성하는 시간 정보가 포함된 멀티미디어 화일들을 생성해 낸다.

2.4 스트리밍 서버

스트리밍 서버는 RTSP를 사용하여 클라이언트와 통신하며 저작 단계에서 멀티플렉서에 의하여 생성된 개별적인 멀티미디어 데이터를 클라이언트의 요청에 따라 전송한다. 그림 6은 멀티플렉싱 기능을 지원하는 서버 구조를 보여준다.

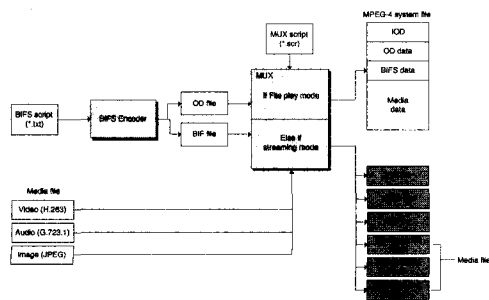


그림 4 수정된 멀티플렉싱 과정

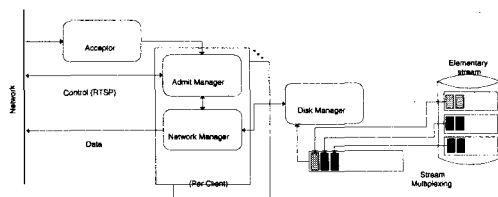


그림 6 서버 구조

서버는 Acceptor, 사용자 수용제어기(Admin Manager), 네트워크 관리자(Network Manager), 디스크 관리자(Disk Manager)의 네 부분으로 구성 되어있다. Acceptor는 클라이언트로부터의 새로운 접속을 담당하며 실행요구를 사용자 수용제어기로 넘겨준다. 사용자 수용제어기는 클라이언트로부터의 RTSP 메시지를 처리하며 RTSP 메시지 단계에 따라 사용자 수용제어를 하며 데이터 전송을 요구할 경우 전송을 위해 네트워크 관리자를 생성한다. 네트워크 관리자는 사용자 수용제어기에 의해 생성되며 데이터 전송을 담당한다. 디스크 관리자는 네트워크 관리자로부터 요청을 받아 클라이언트가 요청한 데이터를 로컬 디스크에서 읽어오는 역할을 한다. MPEG-4 시스템의 가장 큰 특징 중의 하나는 하나의 장면을 구성하는 멀티미디어 객체가 일반 주문형 비디오 시스템과는 달리 여러 개가 될 수 있다는 것이다. 하나의 장면이 하나의 비디오와 오디오로만 구성되는 것이 아니라 여러 개의 비디오, 오디오, 이미지 등이 복합적으로 구성될 수 있다. 서버에서는 장면을 구성하는 각각의 멀티미디어 객체들을 디스크 관리자에서 일정 크기로 읽어 들여 이를 선반입 버퍼에 멀티플렉싱 하여 네트워크 관리자를 통해 클라이언트로 전송하게 된다.

DMIF의 규약에 따른 새로운 전송 계층의 구현의 복잡도를 줄이기 위해 본 논문에서는 RTSP를 사용하여 DMIF를 구현한다. RTSP는 서버와 클라이언트간의 컨트롤 메시지를 주고받는데 사용된다. 서버에서 컨트롤 메시지인 RTSP를 처리하는 부분은 수용 제어기이다. 수용 제어기에서 처리할 수 있는 RTSP 메시지는 DESCRIBE, SETUP, PLAY, SET\_PARAM, GET\_PARAM, TEARDOWN 이다. 수용 제어기가 각각의 RTSP 메시지들을 수신했을 때의 동작은 다음과 같다.

- DESCRIBE : 이 메시지를 수신하면 서버는 현재의 서비스 내역폭, 자원 상태표에 근거하여 사용자 수용 여부를 결정짓는다. 사용자 수용이 결정되면 클라이언트로부터의 RTSP URL에 근거하여 사용자가 원하는 비디오가 서버의 로컬 디스크에 존재하는지 검사를 하게 된다. 원하는 비디오가 존재하지 않을 경우에는 RTSP 에러 메시지를 전송한다. 원하는 비디오가 존재할 경우 RTSP OK 메시지와 함께 SDP를 통해 미디어의 특성에 대한 정보와 세션에 대한 정보를 알려 준다.
- SETUP : 이 메시지를 수신하면 서버는 응답 메시지와 함께 해당 비디오에 속하는 BIFS와 OD 화일을 전송한다. 하나의 MPEG-4 장면을 구성하는 데이터

들은 IOD, BIFS, OD, 멀티미디어 객체로 나뉠 수 있다. 구현된 서버에서는 DESCRIBE 메시지에 대한 응답과 함께 IOD를 전송하고 SETUP 메시지에 대한 응답과 함께 BIFS, OD 화일을 전송하도록 하였다. 클라이언트는 BIFS, OD 화일을 수신하여 장면구성을 위한 트리를 구성하고 수신 받은 미디어 스트림에 대한 각 디코더들을 초기화할 수 있다.

- PLAY : 이 메시지를 수신하면 실제 미디어 데이터를 전송하기 위해 데이터 전송을 담당할 새로운 포트를 생성하고 이 포트로 클라이언트로부터 데이터 요청이 올 때까지 대기하게 된다. 서버에서는 전송해야 할 스트림들의 이름과 개수를 조사하여 클라이언트 구조체에 저장하고 나중에 디스크 관리자에서 이를 참조하여 여러 스트림을 멀티플렉싱할 수 있도록 한다. 수용 제어기는 새로운 사용자를 위한 구조체들을 초기화시킨다. 또한, 데이터 전송을 위한 네트워크 관리자를 Suspend된 상태로 생성한다. 이후 클라이언트의 최초 전송을 위해 디스크 관리자에게 화일 읽기 요청을 보낸다. 디스크 관리자는 요청을 받아들여 스트림 멀티플렉싱을 하지 않는 경우에는 요청이 들어온 화일만을, 스트림 멀티플렉싱을 할 경우에는 일정량의 데이터를 읽어 들여 선반입 버퍼를 채우고 수용 제어기에게 버퍼가 채워졌음을 알린다. 수용 제어기는 Suspend 상태로 되어있는 네트워크 관리자를 Resume 시킴과 동시에 다른 메시지를 받기 위해 반환되고 이후 네트워크 관리자는 선반입 버퍼에 채워진 최초 전송 데이터를 전송하고 수용 제어기로부터 전송 요청이 있을 때까지 대기한다.

- SET\_PARAM : 이 메시지는 클라이언트에서 서버로 보내질 경우 서버의 특정 파라미터 값을 설정하는 역할을 한다. 목적이 뚜렷하게 정의 되어있지 않아서 구현시 다양한 목적으로 사용될 수 있는 장점이 있다. 현재 서버의 모델은 클라이언트가 자료를 필요로 할 때 서버로 요청하여 가져가는 풀(pull) 모델의 서버이기 때문에 서버는 클라이언트로부터 전송 요청이 있을 때만 데이터를 전송하게 된다. RTSP에서 데이터 전송을 요구하는 메시지는 PLAY 메시지가 있지만 그 의미가 풀 모델의 서버에는 적합하지 않다. 따라서 SET\_PARAM 메시지를 클라이언트로부터의 스트림 전송을 요청하는데 사용한다. SET\_PARAM 메시지에 추가적으로 요청을 원하는 스트림의 ID를 같이 보내면 서버는 이 ID를 참조하여 그 ID에 해당하는 화일을 읽어들이어 해당 ID를 전송하는 네트워크 관리자를 실행시켜 추가적인 데이터를 전송한다.

- GET\_PARAM : 이 메시지는 SET\_PARAM과 유사하게 클라이언트에서 서버로 보내질 경우 서버의 특정 파라미터 값을 얻어오는 역할을 한다. 클라이언트 측에서 멀티플렉싱된 스트림을 개별적인 스트림으로 정확히 분리해내기 위해서는 각 개별적인 스트림의 전체 크기를 아는 것이 중요하다. 전송 초기에 서버의 디스크 관리자가 화일로부터 한번에 읽어 들이는 크기가 각 스트림의 크기보다 작을 경우에는 클라이언트에서 멀티플렉싱된 스트림을 그 크기만큼 분할하면 되지만 디스크 관리자가 화일로부터 한번에 읽어 들이는 크기가 각 스트림의 크기보다 클 경우에는 각 스트림의 크기가 서로 다르기 때문에 클라이언트가 복원하는데 문제가 발생한다. 따라서 클라이언트는 PLAY 메시지를 전송하여 멀티플렉싱된 스트림을 전송 받기 전에 개별적인 각 스트림의 크기를 알고 있어야 하며 이를 위해 GET\_PARAM 메시지를 사용한다. 클라이언트는 크기를 알고자 하는 스트림의 ID를 GET\_PARAM 메시지와 함께 보내면 서버는 ID에 해당하는 스트림의 크기를 응답 메시지와 함께 전송한다.
- TEARDOWN : 클라이언트로부터 이 메시지를 수신하면 서버는 해당 클라이언트와의 접속을 종료하고 관련되어 있던 네트워크 관리자들을 제거한다. 또한 사용자 대역폭과 자원 테이블을 갱신하여 새로운 사용자 접속시 수용 제어에 적용될 수 있도록 한다.

**2.5 스트리밍 클라이언트**

MPEG-4 시스템 그룹에서는 MPEG-4의 규약에 맞는 참조 소프트웨어를 제작하는 참조 소프트웨어 그룹이 있다. 이 그룹에서 MPEG-4 시스템 화일을 재생할 수 있는 IM-1 플레이어를 개발하였다. 본 논문에서는 화일 재생만이 가능한 IM-1 플레이어를 스트리밍이 가능하도록 수정하여 스트리밍 클라이언트로 사용하였다. 플레이어에서 데이터 전송을 담당하는 DMIF 계층 내에 네트워크 플레이어가 가능하도록 새로운 모듈을 추가하였고 RTSPService라고 명명한다. 그림 7은 DMIF내에 추가된 RTSPService 모듈의 구조를 보여준다.

사용자가 입력하는 URL을 근거로 하여 DMIF는 화일 서비스를 위한 모듈을 적재할 것인지 네트워크 플레이어를 위한 RTSPService 모듈을 적재할 것인지를 판단한다. 사용자가 입력한 URL이 RTSP URL일 경우 DMIF 필터는 실시간에 DLL로 작성된 RTSPService 모듈을 적재한다. 스트림 멀티플렉싱을 수행하지 않을 때의 RTSP 서비스 모듈은 크게 RTSP 메시지 처리기, 데이터를 수신하는 쓰레드, SDP 파서로 나뉜다. RTSP

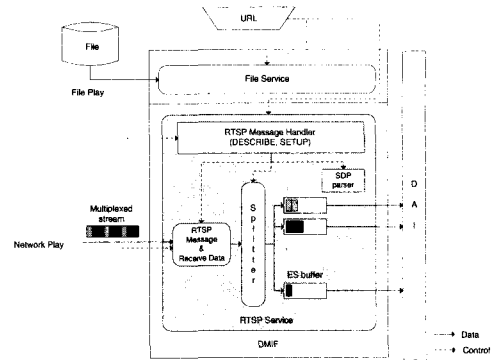


그림 7 RTSPService 모듈

메시지 처리기는 RTSP 메시지 중 DESCRIBE와 SETUP 메시지만을 처리한다. 서버로부터 DESCRIBE 메시지에 대한 응답과 함께 SDP, IOD 데이터가 전송되면 RTSP 메시지 처리기는 SDP 파서를 이용하여 SDP의 내용을 분석한다. SDP로부터 같이 전송된 IOD 데이터의 크기와 앞으로 전송될 미디어 스트림에 할당된 포트 번호를 알 수 있다. 또한 IOD 데이터를 파싱하여 전송될 BIFS, OD 데이터의 ID와 미디어 스트림의 ID들을 파악하게 된다. 서버로부터 SETUP 메시지에 대한 응답이 오면 이와 함께 BIFS, OD 데이터도 함께 전송이 된다. RTSP 메시지 처리기는 전송된 BIFS와 OD 데이터를 각각의 디코더로 전송하여 장면 구성 트리와 각 미디어 디코더를 초기화 할 수 있도록 한다. 이후 서버로부터 PLAY 메시지를 전송할 때 RTSP 메시지 처리기는 PLAY 메시지에 개별 스트림의 ID를 덧붙여서 전송을 하고 각 개별 스트림을 전송 받을 수 있는 쓰레드를 생성한다. 이후 서버로부터 스트림을 전송 받으면 splitter에 의해 각 미디어 스트림들은 개별적인 버퍼로 저장된다. 특정 버퍼가 일정량 미만일 경우 SET\_PARAM 메시지와 함께 요청을 원하는 스트림의 ID를 같이 전송하게 된다. 스트림을 분할하는데 있어서 중요한 것은 각 스트림들을 원래의 크기와 같게 분할하는 것이다. 분할 초기에는 서버로부터 일정 크기씩 전송되기 때문에 문제가 없지만 누적된 크기가 각 스트림의 원래 크기에 이르면 각 스트림의 크기가 서로 다르기 때문에 분할 할 때에 주의를 해야 한다.

멀티플렉싱된 스트림에서 각각의 스트림을 원래 크기로 복원하기 위해서는 각 스트림의 원래 크기를 알고 있어야 한다. 본 논문에서는 이를 위해 RTSP 메시지 중 GET\_PARAM 메시지를 이용하였다. GET\_PARAM 메시지 역시 SET\_PARAM 메시지와 마찬가지로

지로 용도가 특별히 규정되어 있지 않아서 실제 구현시 다양한 용도로 쓰일 수 있다. GET\_PARAM 메시지는 클라이언트에서 서버로 전송되는 경우 서버의 파라미터 값을 요청하는 것으로서 본 논문에서는 GET\_PARAM 메시지에 크기를 알고자 하는 스트림의 ID를 덧붙여 전송함으로써 서버로부터 각 스트림의 크기를 얻을 수 있도록 하였다. 스트림의 크기를 알고 난 후 스트림의 전송이 필요로 할 경우 전송 받은 횟수와 한번에 전송 받은 크기를 곱함으로써 현재까지 전송 받은 크기를 알 수 있고 전체 스트림의 크기에서 이 크기를 뺀으로써 현재 받을 스트림의 크기도 알 수 있다. 지금까지 전송 받은 크기와 새로 전송 받은 크기가 개별 스트림의 크기보다 작다면 새로 전송 받은 스트림을 서버에서 전송하는 크기만큼씩 분할하여 각 스트림 버퍼로 저장하면 된다. 지금까지 전송 받은 크기와 새로 전송 받은 크기의 합이 전체 스트림의 크기보다 크다면 그 스트림의 끝 부분을 전송하는 것이 되므로 그 스트림을 분할할 때는 차이만큼만 분할하여 스트림 버퍼로 저장하면 된다. 그림 8은 스트림의 split 과정에 대한 흐름도를 보여 준다.

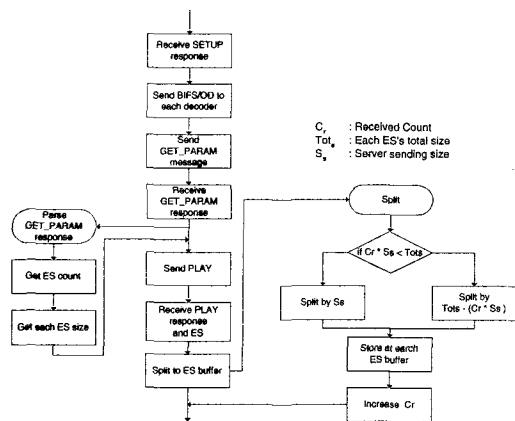


그림 8 스트림 split 과정

### 3. 결과 분석

다음은 구현된 클라이언트의 실행 모습을 살펴보고 각 서버 모델의 성능을 분석한다. 구현된 시스템은 Windows 2000 플랫폼에서 실행된다. 서버는 각 컴포넌트들의 실시간 클래스 변경기능 때문에 Windows NT 4.0 버전 이상에서 동작한다. 시스템을 구성하는 멀티플렉서, 스트리밍 서버와 클라이언트는 마이크로소프트 Visual C++ 6.0을 사용하여 구현하였다.

표 1 예제 데이터

이름	종류	크기(KB)
champ128.jpg	JPEG	4
classic.jpg	JPEG	8
kaesel128.jpg	JPEG	4
linda.jpg	JPEG	6
logos.jpg	JPEG	31
news.kpg	JPEG	9
ofen256.jpg	JPEG	11
pop.jpg	JPEG	11
qmeteo.jpg	JPEG	33
salami128.jpg	JPEG	5
tomate128.jpg	JPEG	4
good.h263	H.263 video	173

#### 3.1 콘텐츠 저작

MPEG-4 시스템 화일의 저작 과정은 다음과 같다. 먼저 장면 구성 정보를 담고 있는 BIFS 스크립트 화일과 장면을 구성하는 미디어 데이터의 특징 및 클라이언트가 필요로 하는 디코더의 구성에 관한 정보를 기술하는 멀티플렉서 스크립트 화일을 일반 텍스트 에디터를 사용하여 생성한다. 작성된 BIFS 스크립트 화일과 멀티플렉서 스크립트 화일, 미리 준비된 비디오 및 이미지 화일들을 이용하여 수정된 멀티플렉서로 멀티플렉싱 과정을 통해 클라이언트를 초기화하는데 필요한 정보를 가지고 있는 IOD화일, 전송될 각 스트림의 정보를 가지고 있는 OD화일, 장면 구성 정보를 가지고 있는 바이너리 형태의 BIFS 데이터 화일, 실제 장면을 구성하는 시간 정보가 포함된 멀티미디어 화일들을 생성한다. 표 1은 사용된 예제 데이터에 대한 상세 내용이다. 표 1에서 보는 바와 같이 예제 MPEG-4 화일을 구성하는 각 객체는 11개의 JPEG 데이터와 하나의 H.263 비디오 화일이다.

초기 화면은 8개의 이미지들로 구성이 되어 있다. 각 이미지들을 클릭하면 새로운 화면으로 갱신이 된다. 즉, 각 이미지들은 HTML 문서의 링크 역할을 하고 있다. 사용자 상호 작용을 위해 각 이미지들 위에 마우스를 위치하면 이미지 위에 특정 글자들이 나타나도록 한다. 여러 이미지들 중 'Pizza'라는 글씨가 겹쳐 보이는 이미지를 선택하면 서버로부터 자료를 다시 스트리밍을 받아 새로운 화면으로 갱신하도록 한다. 새로 갱신된 화면은 3개의 이미지로 구성되고, 가운데 있는 그림을 선택하면 서버에게 다시 스트리밍을 요청하게 되며, 현재 화면 위에 비디오가 겹쳐져서 스트리밍 재생이 된다.

#### 3.2 클라이언트 실행결과

그림 9, 10, 11, 12는 클라이언트의 실행 과정을 보여



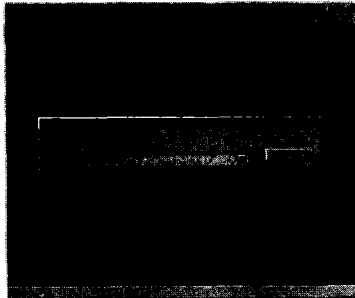


그림 9 RTSP URL 입력

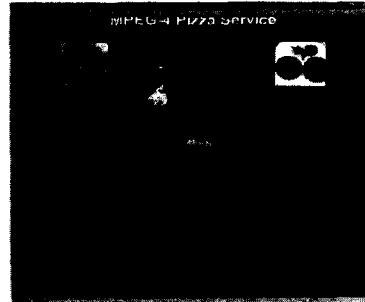


그림 11 이벤트에 의한 장면 전환

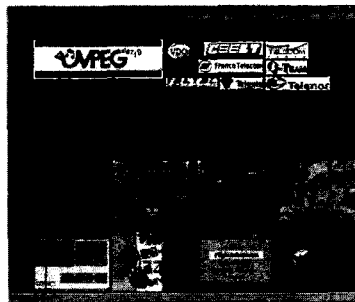


그림 10 초기 화면

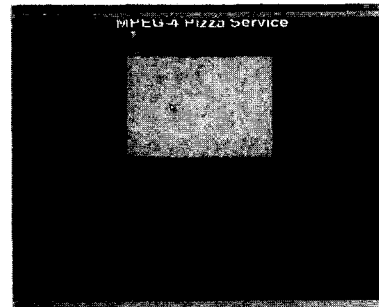


그림 12 이벤트에 의한 비디오 재생

준다. 그림 9는 클라이언트 실행 초기에 사용자로부터 RTSP URL을 입력받는 장면이다. 그림 10은 입력받은 서버의 주소로 접근하여 데이터를 처음 스트리밍 완료했을 때의 장면이다. 초기 화면은 그림에서 보는 바와 같이 여러 개의 JPEG 이미지들로 구성되어 있다. 하단에 실제로 보이는 8개의 이미지 외에도 마우스를 이미지 위로 가져갔을 때 선택된 효과를 나타내기 위하여 각 이미지에 대해 색상이 반전된 이미지들이 같이 스트리밍 된다. 각 이미지에 마우스 포인터를 가져가면 이미지가 반전되거나 특정 텍스트가 출력되는 것을 볼 수 있다. 그림에서는 하단의 그림에 마우스 포인터를 가져갔을 때 'Pizza'라는 텍스트가 표시되고 있다. 이는 클라이언트 측 상호 작용의 예이다.

그림 11은 그림 10에서 텍스트가 표시된 이미지를 클릭하여 장면이 전환된 모습이다. 이미지를 클릭하면 클라이언트는 BIFS 디코더를 통해 이미지와 연결된 Inline 객체로부터 새로운 스트림을 전송 받을 주소를 얻게 된다. 이때 이 주소가 RTSP URL일 경우 초기에 스트리밍을 수행하는 것과 유사하게 RTSPService 모듈을 적재하여 서버로부터 새로운 스트림을 전송 받게 된다. 이는 서버 측 상호 작용의 예로서 장면전환 이벤트가 일어난 것이다. 그림 11에서도 각 이미지에 마우스

를 가져갔을 때 중앙 하단에 있는 박스에 텍스트들이 표시되는 것을 볼 수 있다. 그림 12는 그림 11에서 가운데 이미지를 클릭한 후의 모습이다. 기존의 이미지들이 없어지고 좌측 상단에 새로운 비디오 데이터를 전송 받으며 재생하고 있다. 이 또한 서버 측 상호 작용의 예로서 MPEG-4에서 규정한 상호 작용 중 장면 내 새로운 객체의 삽입, 장면 내 객체 제거의 예이다. 이상의 그림에서와 같이 스트리밍시에도 MPEG-4에서 규정하고 있는 모든 상호 작용을 지원함을 알 수 있다.

### 3.3 서버 성능분석

서버의 성능측정에는 Windows 2000을 사용하는 Pentium III 550을 사용하였다. 성능 측정에 사용된 MPEG-4 화일은 표 1의 예제 데이터를 사용하여 생성하였으며, MPEG-4 화일을 구성함에 있어서 BIFS 스크립트 내에서는 H.263 비디오 데이터가 가장 먼저 재생되고 비디오 재생이 끝나면 이후 2초 간격으로 각 이미지 데이터들이 재생되도록 하였다. Mux 스크립트는 BIFS의 시간 순서와는 상관없이 표 1에 기술된 순서대로 작성하였다. 성능 측정은 MPEG-4 시스템 그룹에서 제공되는 기존의 멀티플렉서를 사용하여 생성된 화일을 스트리밍 할 때와 본 논문에서 구현된 멀티플렉서를 통하여 생성된 화일을 스트리밍 할 때의 시간을 측정하였다.

표 2 세션 준비 지연 시간

MPEG-4 화일 생성 방법	세션 준비 지연시간
MPEG-4 시스템 그룹의 멀티플렉서	3.96 sec
논문에서 제안된 멀티플렉서	2.58 sec

표 2에 표시된 시간은 클라이언트가 최초로 접속하여 디코딩 버퍼로 비디오 데이터 처음 받기까지의 시간을 측정된 것이다. 기존의 MPEG-4 시스템 그룹에서 제공하는 멀티플렉서를 통해 생성된 화일을 스트리밍 할 경우 본 논문에서 구현한 멀티플렉서로 생성한 화일로 스트리밍 할 때보다 약 1.4초 정도의 지연이 발생하는 것을 볼 수 있다. MPEG-4 시스템 그룹에서 제공하는 멀티플렉서를 사용하여 화일을 저장할 경우 BIFS에 지정된 시간 순서와 상관없이 Mux 스크립트 화일에 지정된 순서로 화일이 저장되므로 실제 가장 먼저 재생되어야 할 비디오 데이터는 화일의 가장 끝부분에 위치하게 된다. 따라서 화일의 끝부분에 있는 비디오 데이터를 모두 받을 때까지 대기해야만 한다. 이러한 이유로 기존의 MPEG-4 시스템 그룹에서 제공하는 멀티플렉서를 통해 생성된 화일의 스트리밍 재생은 본 논문에서 구현한 멀티플렉서를 통해 생성된 화일을 스트리밍 재생할 때보다 더 많은 지연시간을 필요로 한다.

#### 4. 결론

본 논문은 RTSP에 기반한 MPEG-4 스트리밍 시스템의 구현에 대하여 기술하였다. 콘텐츠의 저작 과정에서는 기존의 멀티플렉서에 의해 생성되는 MPEG-4 시스템 화일을 스트리밍할 때 생기는 지연 시간 및 사용자 상호 작용 기능을 지원하기 위한 미디어간의 임의 위치 이동에 따른 서버 부하 등의 문제를 해결하기 위해 화면을 구성하는 멀티미디어 객체들을 개별적인 MPEG-4 화일로 생성하도록 하는 새로운 멀티플렉서를 제안하였다. 서버와 클라이언트간의 통신은 DMIF 규약에 의한 새로운 전송 계층 구현의 복잡성을 피하기 위해 기존의 연구들에서 효율성이 입증된 RTSP를 사용하여 DMIF를 대체할 수 있도록 하였고, 이러한 RTSP에 기반한 MPEG-4 데이터를 효율적으로 전송할 수 있는 시스템을 설계하고 구현하였다. 또한, 구현된 시스템이 스트리밍 서비스시에도 MPEG-4에서 규정하고 있는 모든 사용자 상호 작용 기능을 지원함을 보였다.

MPEG-4는 낮은 대역폭과 높은 비디오 품질을 제공하며 사용자와의 상호 작용 기능을 제공하여 인터넷을 통한 양방향 서비스를 가능하게 한다. 또한, 무선 인터넷의 발달로 노트북, PDA, 휴대폰 등을 통한 모바일 멀티미디어 서비스에도 적극 이용될 전망이다. 구현된 RTSP기반 MPEG-4 스트리밍 시스템은 이러한 차세대 멀티미디어 서비스의 중요한 참조 모델이 될 수 있다. 향후 연구 과제로는 콘텐츠 저작을 위한 저작 도구의 개발이 요구된다. 또한, 클라이언트 접속시 모든 BIFS 데이터를 전송하는 것이 아니라 서버에서 BIFS에 지정된 시간을 인지하여 그 시간에 맞게 알맞은 BIFS 데이터를 전송할 수 있는 BIFS 스트리밍에 관한 연구도 필요하다.

#### 참고 문헌

- [1] Rob Koenen, "Overview of MPEG-4 Standard," ISO/IEC JTC1/SC29/WG11 N3747, October 2000.
- [2] ISO/IEC 14496-1, "Coding of Audio-Visual objects : System, final draft international standard," ISO/IEC JTC1/SC29/WG11 N2501, October 1998.
- [3] MPEG-4 DMIF Group, "Generic Coding of Audio-Visual Objects : Part6 - DMIF," ISO/IEC JTC1/SC29/WG11 N2506 FDIS of ISO/IEC 14496-6, November 1998.
- [4] H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol(RTSP)," RFC 2326, April 1998.
- [5] V. Jacobson, M. Handley, "SDP : Session Description Protocol," RFC 2327, April 1998.
- [6] Peter Westerink, Lisa Amini, Sundar Veliah, Will Belknap, "A live intranet distance learning system using MPEG-4 over RTP/RTSP," IEEE International Conference on Multimedia and Expo, Vol.2, pp.601-604, February 2000.
- [7] A. Basso, S. Varakliotis, "Transport of MPEG-4 over IP/RTP," IEEE International Conference on Multimedia and Expo, Vol.2, pp.1067-1070, February 2000.
- [8] H. Kalva, L. Tang, J.F. Huard, G. Tselikis, J. Zamora, L. Cheok, A. Eleftheriadis, "Implementing Multiplexing, Streaming, and Server Interaction for MPEG-4," IEEE Transactions on Circuits and Systems for Video Technology, pp.1299-1311, December 1999.
- [9] ISO/IEC 14496-1, "IM1 Tools," ISO/IEC JTC1/SC29/WG11 N2523, October 1998.
- [10] Xbind, "XbindIP," <http://www.xbind.com>, 1998.



이 상 현

2000년 광운대학교 컴퓨터과학 학사  
 2002년 광운대학교 컴퓨터과학 석사. 현재 (주)필링크 선임 연구원. 관심분야는 멀티미디어 시스템, 운영체제, 컴퓨터 네트워크



이 중 민

1999년 광운대학교 컴퓨터과학 학사  
 2001년 광운대학교 컴퓨터과학 석사  
 2002년 현재 연세대학교 컴퓨터과학과 박사과정. 관심분야는 멀티미디어 시스템, 운영체제, 컴퓨터 네트워크



차 호 정

1985년 서울대학교 컴퓨터공학 학사  
 1987년 서울대학교 컴퓨터공학 석사  
 1991년 영국 University of Manchester 전산학 박사. 1993년 ~ 2001년 광운대학교 컴퓨터과학과 부교수. 2001년 ~ 현재 연세대학교 컴퓨터과학과 부교수  
 관심분야는 멀티미디어 시스템, 운영체제, 분산 컴퓨팅



박 병 준

1984 서울대학교 컴퓨터공학 학사. 1988 미국 미네소타대학교 컴퓨터과학 석사  
 1997 미국 일리노이대학교(UIUC) 컴퓨터과학 박사. 미국 Epic Systems, SPSS 등에서 연구원으로 활동하였으며, 현재 광운대학교 컴퓨터과학과 교수. 관심분야는 인공지능, 데이터마이닝, 기계학습 등