

# 임베디드 웹 스트리밍에서의 미디어 스트림의 실시간 제어 기술

(A Real-Time Control Technique of Media Stream in Embedded Web Streaming)

이 현 주 <sup>†</sup>      김 상 옥 <sup>\*\*</sup>  
(Hyunju Lee)      (Sangwook Kim)

**요 약** 웹 스트리밍 환경은 제한된 환경을 제공하기 때문에 그에 적합한 비디오 재생 서비스 기술과 실시간 전송을 위한 스트리밍 기술 그리고 효율적인 제어 기술이 필요하다. 본 논문에서는 리눅스 환경에서의 고품질 미디어 재생기를 웹 브라우저와 연동하기 위한 기술로서, 실시간 재생되는 미디어 스트림에 대한 효율적인 제어 기술을 제안한다. 제안하는 스트림 제어 기술은 웹 브라우저의 미디어 재생 화면에서 OSD에 의해 제어 데이터를 표시하고, 사용자 입력 데이터를 처리한다. 그래픽 사용자 인터페이스에 독립적으로 동작하며, 제어 데이터에 대한 추가와 확장을 고려하여 설계되었다.

**키워드** : 미디어 콘텐츠, 실시간 미디어 스트리밍, 스트림 제어 기술, 플러그인, OSD

**Abstract** Web streaming environment need a suitable video playback service technique, a streaming technique for real-time delivery and an efficient control technique due to its limited environment. In this paper, we propose an effective control technique for real-time media stream playback as a connection technique between the high-quality media player and web browser based on Linux environment. The proposed stream control technique presents control data through OSD on media playback screen of web browser, and process the user's input data. It acts GUI-independently, and is designed considering addition and extension for control data.

**Key words** : media contents, stream control technique, plug-in, real-time media streaming, OSD

## 1. 서 론

분산 환경에서의 대화형 스트리밍 멀티미디어 콘텐츠 서비스는 사용 목적과 네트워크 환경에 따라 콘텐츠를 표현하는 기술이 각각 다르게 구성된다. 웹 환경은 가장 대표적인 개방형 분산 구조로 다수의 사용자를 위한 다양한 형태의 스트리밍 기술과 비디오 압축 기술이 사용되고 있지만, 임베디드 웹 기반 환경에 적합한 미디어 스트리밍 재생에 대한 개발은 미약하다. 임베디드 웹 환경은 일반 응용 프로그램 환경보다는 여러 가지 제한된 환경을 제공하기 때문에 그에 적합한 최상의 품질을 서비스 할 수 있는 최적의 미디어 재생 기술이 필요하다.

본 논문에서는 홈 네트워킹[1] 시스템에서의 실시간

스트리밍 비디오 재생 기술의 일부로서, 고품질의 미디어 재생기를 웹과 연동한 후 실시간 재생되는 미디어 스트림에 대한 효율적인 제어 기술을 제안한다. 제안하는 제어 기술은 셋탑박스나 같은 소형 시스템에서 최적화된 재생 기술로 웹 브라우저의 미디어 재생 화면에서 OSD에 의해 제어 데이터를 표시하고 사용자 입력 데이터를 처리하는 기술이다. 이 기술에는 브라우저와 재생 기사이의 효율적인 데이터 전달 방법이 포함된다. OSD에 의한 제어 기술은 홈 네트워킹 시스템뿐만 아니라 제한된 리소스를 제공하는 모바일 환경의 비디오 스트리밍 재생 기술에도 포함될 수 있는 응용 기술로서, 결과적으로 사용자 QoS를 향상시키는 기술이라고 할 수 있다.

본 논문의 2장에서는 논문과 관련된 연구에 대해서 소개하고, 3장에서는 제시하는 임베디드 웹 브라우저에서의 스트리밍 데이터의 제어 기술에 대해 설명한다. 4장에서는 성능평가 및 구현을 나타낸다. 마지막으로 5장에서는 결론을 맺는다.

<sup>†</sup> 학생회원 : 경북대학교 컴퓨터학과  
hyunju@cs.knu.ac.kr

<sup>\*\*</sup> 정 회 원 : 경북대학교 컴퓨터학과 교수  
swkim@cs.knu.ac.kr

논문접수 : 2005년 2월 18일

심사완료 : 2005년 9월 3일

## 2. 관련연구

다양한 멀티미디어 하드웨어에 대한 개발 기술이 발전되면서 대용량의 멀티미디어 데이터를 웹 브라우저에 향상된 기법으로 표현하고 처리하는 기술이 많이 개발되고 있다. 대표적인 미디어 데이터의 웹 브라우저 처리 기술인 플러그인[2]은 넷스케이프 네비게이터나 인터넷 익스플로러 같은 웹 브라우저에 첨가되어 브라우저의 일부처럼 동작하면서 미디어 콘텐츠에 대한 요청이 있을 때마다 자동으로 인식되고, 웹 문서에 통합되어 멀티미디어를 자동으로 재생시키는 프로그램이다. 많이 알려진 플러그인을 통한 미디어 재생 기술로는 퀵타임 플레이어(QuickTime Player), 리얼 플레이어(RealPlayer), 윈도우 미디어 플레이어(Windows Media Player), 비디오라이브(VDOlive), 속웨이브(Shockwave) 등이 있다. 이런 재생 기술들은 마치 웹 브라우저에서 직접 실행하는 것과 같은 효과를 제공하기 때문에 편리성뿐만 아니라 멀티미디어를 웹에서 구현하는 데에 유용하게 쓰인다. 또한, 미디어 자료 파일을 사용자의 컴퓨터에 다운로드하지 않고도 실행할 수 있는 장점을 가지고 있으며, 실시간으로 미디어를 재생하는 다양한 인터넷 방송 도구 또는 주문형 비디오 기능까지 다양한 동영상 재생 서비스 기능을 제공하고 있다. 그러나 이런 재생 기술들은 실시간 스트리밍으로 재생중인 미디어 콘텐츠에 대한 제어는 지정된 사용자 인터페이스(GUI)에서만 가능하여 뛰어난 미디어 재생 성능을 제공함에도 불구하고 지원하는 운영체제나 플랫폼에 대한 의존성이 강하기 때문에 플러그인 차원에서 다른 기능 수행을 위한 추가적인 수정이 용이하지 않다.

OSD(On Screen Display)는 사용자가 필요로 하는 정보나 알아야 할 정보를 모니터 화면에 직접 표시하는 기능으로 일반적으로 화면의 조정상태나 진행상태를 화면에 표시하는 기능이다. OSD를 응용하여 스트림 제어를 위한 명령을 다른 화면 조정 상태와 함께 OSD 메뉴를 이용하여 처리한다면, 특정 사용자 인터페이스에 종속적이지 않기 때문에 시스템의 자원 활용을 높일 수 있으며, 스트림 제어를 위한 사용자 인터페이스의 호출이 필요 없기 때문에 보다 쉽게 사용자 중심적인 제어를 할 수 있게 된다. OSD에 의한 제어는 사용자 인터페이스와 상관없이 재생화면에 호출되어 스트림을 실시간으로 제어할 수 있고, 컨트롤 메시지를 기반으로 미디어 재생기에 직접 메시지를 전달할 수 있기 때문에 소형 시스템인 가정용 셋탑박스에서의 미디어 재생 제어 기술로 유용하게 사용될 수 있다. 또한 작은 크기의 화면을 제공하는 모바일 기기에서의 미디어 재생이나 또는 미디어의 재생 형태가 전체 화면 재생 모드일 경우

에 좀 더 사용자 중심적인 제어가 가능하여 그만큼 활용도가 높다고 할 수 있다.

## 3. 임베디드 웹 스트리밍의 실시간 미디어 제어 기술

스트리밍 기술[3-5]은 데이터 전체를 전송한 후 재생하는 것이 아니라 일부 데이터를 전송 후 즉시 재생하는 방식을 사용함으로써 실시간 특성을 가진 멀티미디어 데이터의 전송에 적합하다. 기존의 스트리밍 미디어 콘텐츠에 대한 제어 기술은 특정 그래픽 사용자 인터페이스(GUI)의 지원에 그 서비스가 제한되어 있어서, 다른 시스템과의 연동 및 서로 다른 운영체제와 네트워크 환경을 지원할 수 있는 투명성과 유연성이 부족하여 추가적인 확장이 쉽지 않았다. 그러므로, 보다 다양한 스트리밍 재생환경에서 사용자 인터페이스에 종속적이지 않으면서 스트리밍 데이터를 효율적으로 제어하는 기술이 필요하다.

본 논문에서는 미디어 재생기를 웹 브라우저와 연동시키고 실시간 스트리밍 형태로 재생되는 미디어 콘텐츠에 대한 제어 처리를 위한 스트림 컨트롤러를 구성한다.

그림 1은 콘텐츠 스트리밍 서비스 [6]를 수행하는 시스템의 전체적인 구조와 웹과의 연동 및 제어 기술 처리를 위한 스트림 컨트롤러의 동작 흐름을 나타낸다. 스트리밍 서비스 시스템은 콘텐츠를 제공하는 서버와 제공되는 콘텐츠를 해석하여 재생하는 미디어 재생기 그리고 제어 명령을 입력으로 받아서 최종 표현되는 공간인 웹 브라우저로 구성된다.

서버는 다양한 미디어 포맷의 파일을 저장하고 있으며, 실시간 스트리밍 서비스 제공을 위해 실시간 프로토콜(RTSP: Real-Time Streaming Protocol)[7] 환경하에 동작한다. 미디어 재생기는 초기 미디어 서버와의 연결설정을 수행하고, 실제 미디어 데이터 송수신을 수행한다. 또한 미디어 재생기는 서버에서 제공하는 다양한 포맷의 미디어 콘텐츠를 재생하기 위해 최적화된 디코딩 모듈을 포함하고 있으며, 웹 브라우저와의 연동 기술을 포함하고 있다.

### 3.1 임베디드 웹 브라우저와의 연동 및 스트리밍 기술

웹 브라우저를 통해 미디어 콘텐츠를 재생하기 위해서는 고품질의 미디어 재생기를 웹 브라우저와 연동하는 기술이 필요하다. 그 결과 미디어 재생기는 자동으로 인식되어 사용하기 쉽게 동작해야 하며, 브라우저에 효율적으로 통합되어야 한다. 또한, 재생 화면의 OSD로 표현된 제어 메뉴가 실제 재생기에 효율적으로 적용되기 위해서는 발생된 명령에 대한 적절한 데이터 전달 기술이 필요하다.

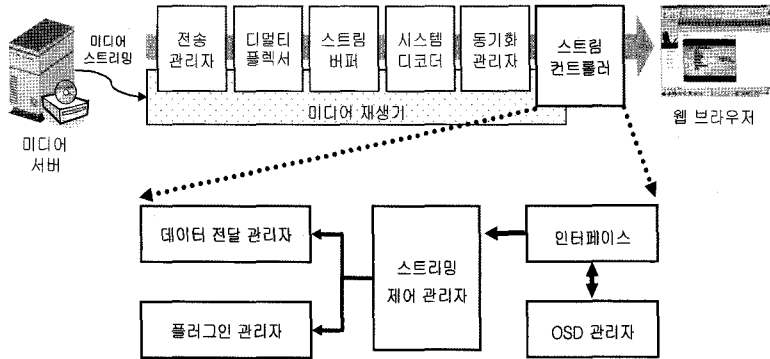


그림 1 스트리밍 서비스 시스템 구조 및 동작 흐름

본 논문에서는 미디어 재생기와 웹 브라우저와의 연동을 위한 기술로 플러그인 관리자를 정의하며, OSD에 의한 미디어 서버로의 직접적인 제어 데이터 전달을 위한 기술로 데이터 전달 관리자를 정의한다.

플러그인 관리자는 웹 문서 관리 규칙과 브라우저 자동 인식 규칙을 가지고 있다. 웹 문서 관리 규칙은 브라우저의 메인 화면을 구성할 문서를 정의할 때 적용하는 규칙으로 HTML이나 XML 형식의 문서를 구성할 수 있도록 한다. 브라우저 자동 인식 규칙은 플러그인의 기능을 재생되는 웹 문서에 통합되도록 하며, 웹 과의 연동을 브라우저에 의해 자동으로 인식되도록 한다. 결과적으로 웹 브라우저에서 직접 미디어를 실행하는 것과 같은 효과를 제공한다. 연동 결과는 웹 브라우저를 통해서 플러그인 라이브러리를 확인할 수 있으며 또한 플러그인에서 지원 가능한 미디어 포맷을 확인할 수 있게 된다.

데이터 전달 관리자는 플러그인과 미디어 재생기 사이의 사용자 입력 제어 데이터 전달에 관련된 기능을 담당한다. 플러그인 모듈과 브라우저에 임베디드된 미디어 재생기 사이의 정확한 제어 데이터 전달을 수행하며, 스트리밍 제어 관리자에서 넘어오는 인자를 바탕으로 적절한 명령 메시지를 작성하여 서버에 전달한다. 웹과 연동하는 미디어 재생기가 여러 기능을 가지도록 재생하고자 하는 미디어에서 중요 기점을 표시해 둔다. 미디어 콘텐츠의 설정된 기점으로 데이터 탐색이 가능하고 또한 데이터의 반복적 사용이 가능하도록 함으로써 기존 미디어 재생기에 대한 이용성을 높여 다양한 분야에서 사용될 수 있다. 이를 위하여 재생기를 제어하는데 컨트롤 메시지를 기반으로 하여 서버와 미디어 재생기에 직접 메시지를 전달한다.

3.2 실시간 스트리밍 데이터의 제어 기술

본 논문에서는 시스템의 자원 활용을 높이고 사용자

중심적인 제어를 위하여 미디어 콘텐츠에 대한 제어 기술로서 OSD를 이용한다. OSD 메뉴는 VCR 컨트롤로 처리되던 재생 제어 명령들을 모두 포함하도록 구성된다.

그림 2는 제어 데이터를 화면에 표시하고, 콘텐츠가 재생되는 환경을 설정하는 OSD 관리자의 구조 및 동작 흐름을 나타낸다.

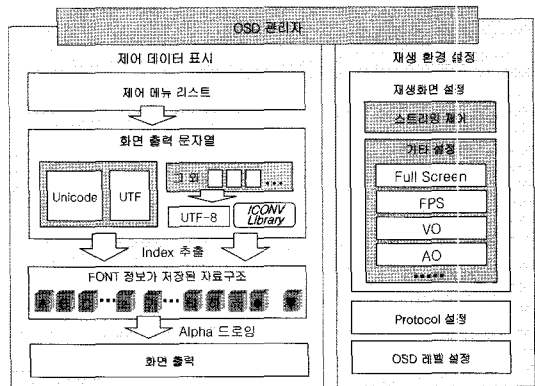


그림 2 제어 데이터 표시 과정과 재생 환경 설정 내용

OSD는 웹 브라우저에 재생중인 콘텐츠 화면에 표시되며, 현재 재생과 관련해 사용중인 기능과 사용 가능한 기능을 보여준다. OSD 문자열 처리는 영어와 한글을 포함한 모든 문자열 처리를 위해 유니코드일 경우와 유니코드가 아닐 경우로 나누어 3 바이트 크기의 UTF-8 문자열 표현 방식을 적용하여 처리한다. 유니코드일 경우는 불려온 폰트 데이터의 인덱스 값을 그대로 추출하여 폰트 정보 자료구조에 저장하고, 유니코드가 아닌 문자열 즉, 한글을 포함한 2바이트 크기 이상의 모든 문자열은 ICONV 라이브러리를 이용하여 리코딩 과정을 거쳐 3바이트 크기의 UTF-8로 변환한 후에 인덱

스 값을 폰트 정보 자료구조에 저장한다. 이러한 문자열 처리방법은 다양한 재생환경을 고려한 것으로, 운영체제 및 기반 소프트웨어에 따라 폰트 처리의 지원 범위와 방법이 다르기 때문이다. 재생화면인 웹 브라우저에서 OSD 메뉴 호출이 발생하면 OSD 관리자는 제어 메뉴 리스트에 저장된 문자열을 그 특성에 따라 자료구조에 폰트 정보를 저장하고, 저장된 폰트 정보는 적절한 알파 형태로 화면에 그려지게 된다.

사용자 입력에 의해 OSD 제어 표시 명령이 발생하면 메시지 큐에 발생 명령을 저장한 후 재생 화면에 OSD 제어 메뉴를 표시한다. 이때, 제어 메뉴는 나중에 발생할 수도 있는 메뉴 내용에 대한 추가 및 수정을 고려하여 미리 별도의 리스트를 작성해 둔다. 리스트에는 화면에 표시되는 모든 제어 메뉴가 항목별로 저장된다.

화면에 표시되는 데이터는 일시정지/다시재생, 멈춤, 점프 등의 일반적인 스트리밍 제어 부분과 파일열기, 도움말, 콘솔 등의 기타 설정 부분으로 표시한다. 재생 환경 설정은 재생 화면설정, 프로토콜 설정 그리고 OSD 레벨 설정으로 나눌 수 있다. 재생 화면 설정은 미디어 콘텐츠에 대한 음량이나 밝기, 대비, 초당 재생되는 프레임 수, 오디오/비디오 디바이스 설정 등을 할 수 있고, 프로토콜 설정은 실시간 스트리밍의 특성을 잘 적용할 수 있도록 RTSP 프로토콜이 설정되어 있으며, OSD 레벨 설정은 화면에 표시될 OSD 문자열의 종류와 색깔, 출력되는 위치, 크기 등을 설정할 수 있다.

웹과 연동되는 미디어 데이터에 대한 사용자 QoS를 높이기 위해서는 스트림의 흐름을 제어할 때 요구되는 모든 메시지를 효율적으로 정의 및 전달하고, 시스템 동작에 필요한 모든 자원을 관리할 필요가 있다. 본 논문에서는 스트리밍 제어 관리자를 통해 전송 메시지 관리

및 시스템 자원 관리를 한다.

그림 3은 스트리밍 제어 관리자를 중심으로 미디어 스트림이 제어되는 과정을 나타낸다.

OSD 제어 메뉴 중 특정 명령이 발생하면 스트리밍 제어 관리자는 현재 입력된 OSD의 ID를 통하여 명령의 종류를 파악하고 해당하는 파일 길이나 전체 패킷 수 같은 부가 정보를 파악한 뒤 그 값들을 인자로 해서 데이터 전달 관리자를 호출한다. 데이터 전달 관리자는 스트리밍 제어 관리자에서 넘어오는 인자를 바탕으로 적절한 명령 메시지를 작성하여 서버에 전달한다. 또한, 스트리밍 제어 관리자는 명령을 수행함으로써 요구되는 시스템에 대한 자원을 관리한다. 자원 관리는 시스템에 동작하면서 요구하는 컴퓨터 자원을 할당하고 삭제하는 일로, 프로그램이 시작할 때는 요구되는 메모리와 디바이스를 할당하고 종료할 때는 프로그램이 가지고 있던 메모리와 디바이스 등의 자원을 해제하는 동작을 수행한다. 자원 관리는 재생과 종료뿐만 아니라 점프 제어가 발생한 경우에도 필요하다. 점프 제어가 발생되면 수행하던 버퍼링을 종료시키고 그 후에 사용하던 자원을 초기화하는데, 이 과정에서 스트림의 임시 저장 장소인 스트림 버퍼와 디코딩 버퍼, 프리젠테이션 버퍼의 내용을 모두 비우며, 점프 제어가 발생할 때마다 초기화 과정을 수행한다. 스트리밍 제어 관리자가 처리하는 제어 명령은 일반적으로 GUI에서 제공하는 VCR 컨트롤의 기본 제어 명령들이 포함된다.

웹 브라우저의 메인 페이지에서 콘텐츠의 요청은 재생(Play) 명령으로 하며, 이 명령으로 초기 연결 설정이 이루어지면 데이터를 전송 받기 시작한다. 초기 연결 과정에서 현재 사용자가 선택한 파일에 대한 길이와 전체 재생 시간, 전체 패킷 수, 패킷 당 재생 시간 등의 정보

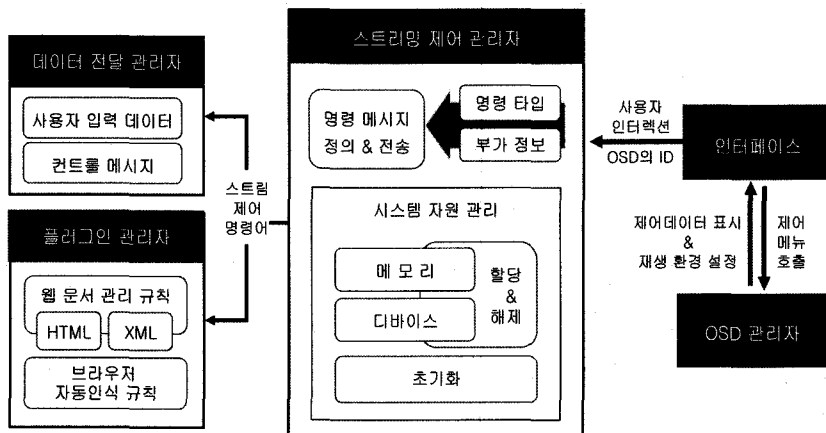


그림 3 미디어 스트림 제어 과정

를 알게 된다. 연결 설정된 세션은 유지하되 서버로부터 스트리밍 진행을 중지시키는 명령이 일시정지(Pause) 명령이다. 다시 연결 설정할 필요 없이 중지된 스트림으로부터 스트리밍을 다시 수행할 수 있는 명령이 다시 재생(Resume) 명령이다. 재생중인 스트림의 임의의 위치로 이동하여 재생하는 명령이 점프(Jump) 명령이다. 초기 연결 설정에서 알게 된 각종 파일정보를 기반으로 점프해야 될 임의의 위치를 점프 메뉴 리스트를 미리 작성하여 호출하는 형식으로 처리한다. 점프 메뉴 리스트에는 OSD에 출력될 메뉴 명령과 점핑을 하게 될 절대적인 시간 값이 저장된다. 점프 메뉴는 재생될 콘텐츠의 특성에 따라 메인 페이지를 작성할 때 정의하며, 절대 시간 값은 전체 재생시간과 콘텐츠의 중요 기점을 고려하여 정의한다. 입력 값과 다음에 전송 받아야 할 패킷 번호를 계산하여 서버에 점프 명령과 함께 전송하면 서버는 해당하는 패킷에서부터 비디오를 재전송한다. 현재 스트리밍되는 미디어에 대한 완전 종료가 멈춤(Stop) 명령이다. 이 명령은 서버로부터 데이터 전송을 중지하고 서버와 재생기 사이에 맺어진 연결 설정을 완전히 종료한다. 이때, 웹 브라우저의 플러그인 상태는 유지하면서 재생 되었던 스트림의 연결 설정만 종료하게 된다.

4. 성능 평가 및 구현 예

본 논문에서 제시한 미디어 데이터의 웹 브라우저 처리 기술을 평가하기 위하여 임베디드 웹 스트리밍 기술 부분과 미디어 스트림의 실시간 제어 처리 기술 부분으로 나누어 비교하였다. 임베디드 웹 스트리밍 기술 부분은 고품질의 미디어 재생기를 웹과 연동하는 기술로서의 지원 범위와 기능을 나타냈고, 스트림의 실시간 제어 처리 기술 부분은 개방형 기술로서 OSD의 성능을 최적화하여 사용자가 많고 어플리케이션 적용이 용이한 애플플레이어[8]와의 기능을 비교하였다. 구현결과로는 제어

기술이 적용된 재생기가 웹 브라우저와 연동되는 상태와 OSD 실행 예를 제시한다.

4.1 성능 평가

제안하는 임베디드 웹 스트리밍 기술의 성능을 평가하기 위하여 웹 브라우저와의 연동을 위한 기능 및 기술의 지원 범위를 나타냈다.

표 1은 제안하는 임베디드 웹 스트리밍 기술의 성능을 나타낸 것이다.

제안하는 OSD 제어 기술의 성능을 평가하기 위하여 애플플레이어(MPlayer 1.0pre5)의 OSD 성능과 비교하였다. 제안하는 OSD 제어 기술은 KNU 재생기(Ver. 2.0)에 적용하여 실험하였다. 두 재생기는 동일한 운영체제를 기반으로 하며, 각각의 독립적인 컴포넌트들로 구성되어 다양한 미디어 포맷을 지원하는 등 비슷한 성능을 가지고 있다. 특히, 애플플레이어에서도 OSD에 의한 미디어 스트림 제어가 가능하기 때문에 본 논문에서 제시하는 제어 기술과의 비교가 용이하다고 할 수 있다. 비교되는 핵심 사항은 OSD에 의한 제어 구성 능력이다.

표 2는 제안하는 기술과 애플플레이어의 OSD에 의한 제어 구성 능력을 비교한 것이다.

제안하는 OSD 제어 기술 중 점프 제어는 정의된 메뉴 리스트에 따라 분기하는 형식을 가지고 있다. 지정된 초 단위의 분기는 각각의 재생 환경 설정 부분에 정의된 탐색(seek) 명령을 통해 가능하다. 별도의 점프 리스트를 구성하는 이유는 분기되는 지점을 미디어 콘텐츠 별로 재생 특성과 목적에 맞도록 다르게 지정할 수 있다는 점과 점프될 메뉴 리스트를 화면에 표시해 줌으로써 분기되는 기점에 대한 사용자의 이해도 및 콘텐츠 재활용성을 높인다는 점이다. 반면 애플플레이어는 OSD 스트리밍 제어 기술이 제안하는 기술에 비해 극히 제한적으로 지원하고 있다. 점프 기능은 탐색(seek) 명령에 의해서만 수행이 가능하고, 만약 사용자가 분기되는 지점에 대한 수정을 원하더라도 점프되는 탐색 시간 값은

표 1 임베디드 웹 스트리밍 기술의 성능

지원 항목	제안하는 임베디드 웹 스트리밍 기술
운영체제	리눅스(Linux)
구성요소간의 독립성	지원 (기능 추가 및 접근의 용이성 향상)
실시간 스트리밍 제어	지원 (OSD, 그래픽 사용자 인터페이스를 통하여 가능)
다양한 운영체제에 대한 투명성	제한적 지원 (재생기의 모든 구성요소 및 스트림 컨트롤러 가능)
기능의 확장성	지원 (구성요소의 독립적인 관리를 통하여 가능)
적용 대상	PC, 셋탑박스, 모바일 단말
응용 분야	인터넷 실시간 유틸리티(원격교육방송, 온라인 게임), 이미지/음성/동영상 관련 유틸리티 등

표 2 OSD 성능 비교

지원항목	제안하는 기술	엠펙플레이어
재생환경설정 (음량, 대비, 밝기, 탐색(seek) ...)	지원	지원
스트리밍 제어 설정 (메뉴호출방식)	일시정지	지원
	다시재생	지원
	멈춤	지원하지 않음
	점프	지원하지 않음
기타 설정 (파일열기, 도움말, 콘솔 ...)	지원	지원
OSD 문자열	유니코드 (Unicode)	지원
	비-유니코드 (Non-Unicode)	지원 (3byte UTF-8)
	OSD 레벨 설정 (위치, 컬러, 크기, 폰트)	지원

재생기의 환경 설정 부분에 정의되어 있어서 일반 사용자의 임의적인 수정이 용이하지 않다. OSD 문자도 한글을 포함한 2바이트 이상의 문자는 지원하지 못하며, OSD 레벨 설정도 크기와 폰트 지정만 가능하다.

4.2 구현 예

제안하는 스트림 제어 기술은 리눅스 운영체제의 셋탑박스 시스템 기반에서 임베디드 웹 브라우저에 미디어 재생기를 연동시키고 OSD 기술에 의해 스트리밍 미디어에 대한 최적의 스트림 제어 성능을 나타냈다. 구현 환경은 실시간 스트리밍을 제공하는 서버와 웹 브라우저는 모질라(Mozilla ver. 1.8)를 사용하였으며, 재생기는 제안하는 스트림 제어 기술이 적용된 KNU 플레이어를 테스트 환경으로 사용하였다. 미디어 콘텐츠는 H.264 비디오와 MPEG-4 AAC 오디오로 인코딩된 MPEG-4 콘텐츠를 스트리밍 하였다.

사용자가 웹 브라우저의 메인 페이지에 접속하여 콘텐츠를 요청하면, 웹 브라우저는 플러그인으로 콘텐츠를 확인하고 서버와의 연결 설정 확인 후 스트리밍 서비스를 시작하게 된다.

그림 4의 (a)는 웹 브라우저의 플러그인을 이용하여 스트리밍 서버에 접속한 후 미디어를 스트리밍 서비스하는 메인 화면으로, 가정용 인터넷 방송 프로그램의 예이다. 미디어 재생기는 웹 브라우저에 임베디드 되었고, 사용자는 OSD 메뉴를 호출하여 미디어 재생기를 직접 컨트롤하는 화면이다. OSD에 의한 스트림의 제어는 기존의 미디어 재생기의 인터페이스에서 제공하던 VCR 컨트롤을 모두 포함하고 있다.

그림 4의 (b)와 (c)는 OSD에 의한 미디어 스트림 제어 예로 특정 스트림 부분으로의 이동 즉, 점프 제어를 나타낸다. 점프 제어 메뉴 리스트에 정의된 내용이 화면에 표시되며, 이러한 정보를 바탕으로 전체 데이터에서 현재 재생중인 화면의 상대적인 위치를 계산하여 OSD

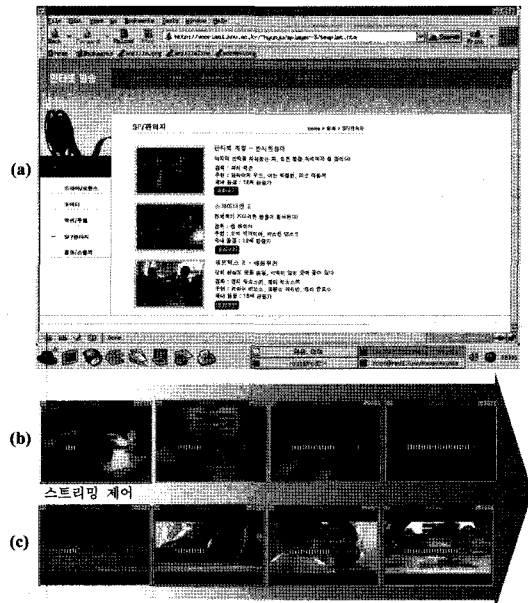


그림 4 미디어 스트리밍 서비스와 스트림 제어 예

막대로 사용자에게 알려준다.

5. 결론

본 논문에서는 리눅스 환경에서 고품질의 미디어 재생기를 웹과 연동한 후 재생 화면에서 콘텐츠를 실시간 제어하는 기술을 제안하였다. 스트림 제어 기술은 기반 기술과 개발 환경에 많은 제약 사항을 가지고 있는 웹 환경에서 시스템의 자원 절약과 사용자 중심적인 제어 처리를 목적으로 미디어 재생 화면에서의 OSD에 의한 제어 데이터 표시 및 사용자 입력 처리를 담당할 수 있다. 또한 서버와 미디어 재생기와 브라우저 사이의 효율적인 데이터 전달을 포함하고 있다. 지정된 웹 브라우저

의 메인 페이지에서 미디어 재생기는 최적의 성능으로 동작할 수 있게 하였으며, 이러한 요소들을 테스트하기 위해 웹 브라우저에 임베디드한 스트리밍 비디오 재생 어플리케이션을 설계하고 구현하였다.

향후 연구 과제는 제안된 스트림 제어 기술을 홈 네트워크 같은 특수한 환경이나 용도에 맞춰 최적화된 비디오 재생 기술 개발에 응용하는 것이며, OSD의 구조와 성능을 향상시켜 이질적인 플랫폼이나 운영체제의 다양한 미디어 재생기에도 적용하는 것이다. 이는 재생 환경에 종속적이지 않으면서 기술의 활용 범위를 폭 넓게 하기 위한 것이다.

컴퓨터의 상호작용, 임베디드 멀티미디어 시스템



김 상 욱

1979년 경북대학교 컴퓨터공학으로 학사 학위 취득. 1981년 서울대학교 컴퓨터과 학과 석사학위 취득. 1989년 서울대학교 컴퓨터과학과 박사학위 취득. 현재 경북대학교 전자전기컴퓨터학부 교수. 관심분야는 모바일 멀티미디어 시스템, 멀티미

디어 콘텐츠 저작 및 인간과 컴퓨터의 상호작용

### 참 고 문 헌

- [1] R. Bill, "Home Networks a standards Perspective," IEEE Communications Magazine, December 2001.
- [2] 김상욱, 이송록, 임영순, 이현주, 김상욱, 조창식, 마평수, "스트리밍 재생기를 이용한 웹 미디어 서비스 설계 및 구현", 한국멀티미디어학회논문지 Vol. 7, No. 2, pp.237-240, November 2004.
- [3] Dapeng Wu, Yiwei Thomas Hou, Ya-Qin Zhang, Jon M. Peha, "Streaming Video over Internet: Approaches and Directions," Circuits and Systems for Video Technology, IEEE Transactions on, Vol.11, Issue 3, pp.282-300, March 2001.
- [4] Kellerer, W., Steinbach, E., Eisert, P., Girod, B., "A real-time Internet streaming media testbed," 2002 IEEE International Conference on Multimedia and Expo, ICME '02, Vol. 2, pp.453 - 456, August 2002.
- [5] Jason But, Greg Egan, "Designing a scalable video-on-demand system," Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, Vol. 1, pp.559 - 565, June 2002.
- [6] Microsoft Corporation, "Inside Windows Media 인터넷 방송을 위한 스트리밍 기법의 모든 것", Com & Books, 2000 6월.
- [7] Gregory J. Conklin, Gary S. Greenbaum, Karl O. Lillevold, Alan F. Lippman, Yuriy A. Reznik, "Video coding for streaming media delivery on the Internet," Circuits and Systems for Video Technology, IEEE Transactions on, Vol. 11, Issue 3, pp.269 - 281, March 2001.
- [8] MPlayer, <http://www.mplayerhq.hu/>



이 현 주

2004년 경북대학교 컴퓨터학과 석사학위 취득. 2004년 3월~현재 경북대학교 컴퓨터학과 박사과정. 관심분야는 멀티미디어 콘텐츠의 응용, 멀티미디어 데이터의 관리기법 및 장면구성기술, 인간과