

다양한 운영체제를 지원하는 MMS 재생기+

박한성 김장하^o 이동훈 김상욱 조창식* 마평수*

경북대학교 컴퓨터과학과, *한국전자통신연구원

{hspark, jhkim^o, leedh, swkim}@woorisol.knu.ac.kr, {cscho, pmah}@etri.re.kr

MMS Player supporting Various Operating System

Hanseong Park Jangha Kim^o Donghoon Lee Sangwook Kim Changsik Cho* Pyeongsoo Mah*

Dept. of Computer Science, Kyungpook National University

*Electronics and Telecommunications Research Institute

요 약

MMS 프로토콜과 WMV 코덱은 마이크로소프트에서 만든 스트리밍 프로토콜과 비디오/오디오 코덱이다. 오늘날 많은 스트리밍 서비스가 이 프로토콜과 코덱을 이용하고 있다. 그러나 프로토콜과 코덱이 공개되어 있지 않아서 스트리밍 서비스의 재생 환경이 윈도우즈 기반의 운영체제로 제한되어 있다. 따라서 서비스를 제공하는 서버와 그 서비스를 이용하는 클라이언트가 반드시 윈도우즈 환경에서만 동작할 수 있는 문제점을 가진다. 본 논문에서는 윈도우 운영체제가 아닌 리눅스와 같은 공개된 운영체제 환경에서 MMS 프로토콜을 사용하여 스트리밍 서비스를 받을 수 있도록 하는 방법을 제시한다. 이를 위하여 본 논문에서는 기존의 많은 연구 결과를 기반으로 MMS 프로토콜을 분석하고 공개 프로젝트인 MPlayer에서 윈도우 스트리밍 서버와의 VCR 컨트롤 지원을 위한 방법을 제시한다.

1. 서 론

실시간 스트리밍 기술과 코덱의 발전으로 고화질, 고음질의 미디어 콘텐츠를 스트리밍하는 서비스들이 늘어나고 있다. 사용자는 다양한 콘텐츠를 스트리밍 서비스를 통해 이용할 수 있게 되었다. 이를 가능하게 해주는 대표적인 기술로 Microsoft의 Media 9 Series[1]가 있다. 이는 파일 포맷을 WMV로 스트리밍 기술은 MMS 프로토콜을 사용한다. WMV의 고화질, 고음질 콘텐츠와 MMS의 안정적인 스트리밍으로 사용자는 질적으로 좋은 콘텐츠의 스트리밍 서비스를 받는다.

그러나 이런 기술들은 Microsoft에서 개발하고 공개하지 않아 서비스를 받는 재생 환경이 윈도우즈 기반의 운영체제로 제한된다. 재생 환경의 제한으로 인해 서비스 사업자들은 일부의 사용자를 포기하거나 MMS 프로토콜외의 스트리밍 서비스도 제공해주어야 하는 문제점이 생긴다. 이런 문제를 해결하기 위해 본 논문에서는 공개 운영체제 환경에서 MMS 프로토콜을 지원하는 스트리밍 재생기를 제시한다.

WMV 코덱 및 MMS 프로토콜이 공개 되지 않은 것을 해결하기 위해 많은 사람들이 WMV를 재생하는 기술과 MMS프로토콜을 분석하고 있다. 현재까지 알려진 기술로 WMV를 윈도우즈가 아닌 환경에서 재생할 수 있고 MMS 프로토콜의 재생과 관련된 부분들이 알려져서 스트리밍 받은 데이터를 저장할 수 있는 방법까지 제시 되었다[2]. 이런 방법을 통해서 공개 운영체제에서 실행되는 여러 미디어 플레이어에서 MMS 프로토콜을 이용해 스트리밍 서비스를 받을 수 있게 했다. 그러나 이런 플레이어에 MMS와 관련된 기능들은 윈도우즈 환경에 미디어 플레이어의 기능을 모두 만족 시키지 않고 있다.

대표적인 공개 프로젝트로 xine[3], mplayer[4], vlc[5]가 있다. 그러나 이들은 윈도우즈용 재생기에서 할 수 있는 WMV9의 재생과 기본적인 VCR 기능을 모두 할 수 있는 재생기는 없다. 본 논문에서는 윈도우즈 미디어 플레이어의 기본적인 기능을 제공하는 리눅스용 재생기를 제시한다. 이 재생기는 기존에 리눅스와 같은 운영체제에서 개발되어 온 공개 프로젝트를 기반으로 하였으며 Windows Media Player에서 제공하는 수준의 스트리밍 서비스를 공개 운영체제에서 제공할 수 있게 하기 위하여 구현 되었다. 본 논문의 구성은 다음과 같다. 먼저 2장에서는 MMS 프로토콜에 대한 분석 내용을 기술한다. 3장에서는 논문에서 제시 하는 재생기를 구현하는 방법을 그리고 4장에서는 구현 결과를 기술하며 5장에서는 결론과 앞으로 연구해야 할 방향을 제시한다.

2. MMS 프로토콜

MMS 프로토콜은 재생과 관련된 정보를 주고받기 위한 커맨드 패킷과 실제 미디어 정보가 있는 미디어 패킷으로 구성된다.

2.1 커맨드 패킷

커맨드 패킷은 접속 요청, 스트리밍 프로토콜 설정, pause, seek과 같은 커맨드를 주고받기 위한 패킷이다. 기본적으로 40바이트의 데이터를 갖고 있으며 커맨드의 종류에 따라 40바이트 뒤에 필요한 부가 정보들이 오게 된다. 표 1은 커맨드 패킷의 앞 40바이트의 내용을 보여 준다. 40바이트 중에서 마지막 4바이트중 앞의 2바이트가 커맨드 값을 나타내는데 값에 따른 역할은 4장에서 다룬다.

2.2 미디어 패킷

미디어 패킷은 미디어에 대한 정보를 갖고 있는 헤더

+ 본 연구는 한국전자통신연구원 컴퓨터소프트웨어연구소의 지원으로 수행되었음.

와 실제 미디어 데이터를 말한다. MMS 프로토콜을 통해 전송되는 미디어 패킷은 모두 그림 1의 헤더 정보를 포함해서 온다.

표 1 커맨드 패킷의 기본 40바이트

40바이트시작 -->

4bytes 01 00 00 [00]	보통 00 00 00 00 으로 나타나며 일부 서버에서 보내지는 메시지에서 4번째 값이 00이 아닌 경우가 있다.
4bytes CE FA 0B B0	커맨드 패킷을 표시하기 위해 "CE FA 0B 0B" 라는 값으로 고정된다.
4bytes	커맨드 패킷에서 남은 데이터의 길이를 Byte단위로 나타낸다.
4bytes 4D 4D 53 20	프로토콜 타입 (MMS)
4bytes	남은 데이터의 길이. 8바이트를 1로 설정
4bytes	Sequence number. 0으로 시작.
8bytes	네트워크 타이밍을 위한 time stamp값
4bytes	남은 데이터의 길이. 8바이트를 1로 설정
2bytes Command	커맨드 값.
2bytes	3일 경우 서버로 보내지는 메시지
4bytes Direction	4일 경우 클라이언트로 보내지는 메시지

-->40바이트 끝

4 bytes Sequence number	1 byte Packet ID type	1 byte UDP sequence TCP flags	2 bytes Packet length
----------------------------	--------------------------	----------------------------------	--------------------------

그림 1 미디어 패킷의 헤더 정보

그림 1에서 Sequence Number는 Packet Sequence Number를 뜻하고 Packet ID Type은 2일 경우 패킷에 포함된 데이터는 미디어 헤더 정보를 뜻하고 4일 경우 미디어 데이터를 뜻한다. UDP Sequence는 전송 프로토콜을 UDP로 사용할 경우 Sequence Number를 뜻하며 TCP flags는 TCP를 사용할 경우 패킷의 순서에 대한 정보를 뜻한다. 이 값이 0일 경우 패킷의 중간 데이터를 4일 경우 패킷의 첫번째 데이터를 8일 경우 패킷의 마지막 데이터를 C일 경우 패킷이 오직 하나만 있다는 것을 뜻한다. 마지막으로 Packet Length는 전송되는 패킷의 총 길이를 뜻한다.

3. 구현 방법

본 논문에서는 운영체제를 리눅스로 플레이어는 기존에 존재하는 것 중 Mplayer를 기반으로 한다. Mplayer는 스트리밍으로 재생할 때 Seek기능이 없고 서버로부터 오는 커맨드의 값들을 처리하지 않는 문제점이 있다. 예를 들어 서버로부터 스트리밍의 종료 커맨드가 오더라도 재생기의 재생을 멈추지 않고 마지막 화면이 그대로 나온다. 이런 문제가 있지만 WMV9를 지원하고 프로그램의 구조가 비교적 간단하기 때문에 Mplayer를 선택하게 되었다.

3.1 재생을 위한 커맨드 및 전송 순서

재생기의 입력으로 mms URL이 들어오게 되면 이를

파싱한 후에 서버와의 접속 및 스트리밍 시작과정이 필요하다. 이는 그림2에 나타나있다. 그림2에서 보이는 숫자는 커맨드 패킷에서 커맨드 부분에 들어가는 값으로 이런 과정을 거친 후에 재생에 필요한 미디어 데이터의 전송이 서버로부터 시작된다.

그림2 중에서 서버에서 보내는 커맨드 중 06이나 05와 같은 커맨드는 커맨드 뒤에 많은 부가 정보들이 오며 이런 정보들은 재생을 하면서 Seek과 같은 복잡한 기능을 구현하기 위해서 필요한 값들이다.

대표적인 예를 들자면 앞으로 재생할 미디어의 총 packet수, 하나의 패킷의 길이 그리고 총 재생시간과 같은 값이다. 이런 값들을 알고 있어야 만이 Seek 연산과 같은 사용자의 입력에 제대로 된 결과를 돌려줄 수 있게 된다. 본 논문에서 구현한 재생기는 이러한 값들을 저장해두고 커맨드에 따라 이 값들을 사용한다.

그림2와 같은 순서로 커맨드를 주고받게 되면 서버에서는 미디어 데이터를 클라이언트가 선택한 프로토콜로 전송한다. 이렇게 받은 미디어 데이터를 재생기에서 재생하기 시작한다.

재생을 하고 있는 중에 서버는 1분마다 한번씩 1B 커맨드를 보내게 되는데 이것은 클라이언트가 접속된 채로 제대로 스트리밍을 받고 있는가를 확인하는 것이다. 1B 커맨드를 받은 클라이언트는 반드시 1B 커맨드로 답을 해주어야 서버는 접속이 유지된 것으로 판단하고 계속적인 스트리밍을 한다.

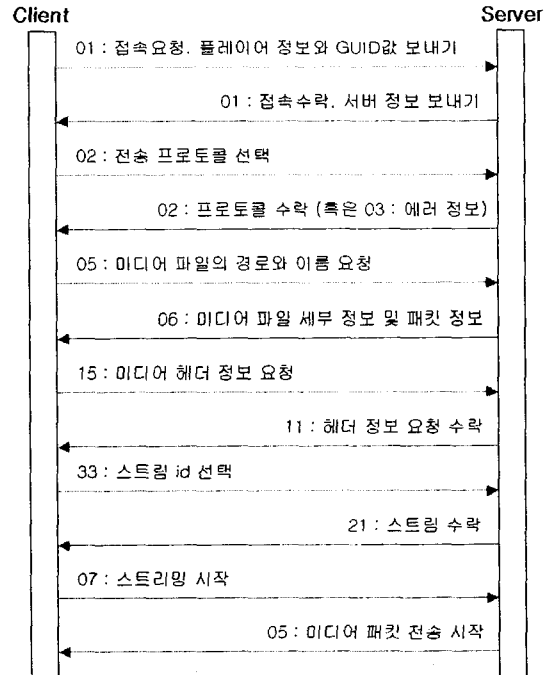


그림 2 재생을 위한 커맨드와 전송 순서

3.2 Seek 기능 구현

본 논문에서 선택한 Mplayer에서는 Seek기능을 제공하지 않는다. 그러나 스트리밍 서비스에서 Seek 기능은 매우 중요하므로 반드시 제공되어야 한다. 그림3은 Seek를 위한 MMS 커맨드 전송 순서이다.

먼저, 현재 스트리밍 되고 있는 데이터의 전송을 멈추기 위해 09 커맨드를 통해 스트리밍 중지를 서버에 요청한다. 클라이언트가 서버에 보내는 재생 정지 요청 커맨드는 09와 0D 커맨드가 있는데 09는 접속을 유지하면서 스트리밍만 멈추고 0D 커맨드는 접속을 끊을 때 사용한다. 09 커맨드를 받은 서버는 1E 메시지로 재생 정지 커맨드를 보낸다. 1E는 서버에서 스트리밍의 정지를 뜻하는 커맨드로 스트리밍이 완료되었거나 클라이언트의 정지 요청이 있을 경우 보내는 커맨드이다. 1E 메시지를 받은 클라이언트는 32 커맨드를 통해 클라이언트의 여러 정보(운영체제, 플레이어버전 등)를 보내게 되는데 이는 윈도우즈 미디어 플레이어일 경우이며 실제로 Seek를 하기 위해 반드시 보내야 하는 메시지는 아니므로 실제 구현에서는 보내지 않는다. 스트리밍이 정지된 상태에서 클라이언트가 스트리밍을 요청하는 07커맨드는 원하는 위치부터 재생이 될 수 있도록 값을 설정하는 부분이 있다. 8바이트로 된 재생 위치 시간 값과 4바이트로 된 재생 위치 패킷번호 값 이렇게 두 가지를 쓸 수 있는데 본 논문에서는 패킷번호를 사용한다. 이것을 위해 처음에 재생할 때 서버로부터 06 커맨드의 값 중 총 패킷의 수를 저장해두고 Seek를 할 때 그 숫자 범위 내의 값을 07커맨드에 설정하여 서버로 커맨드를 보내게 된다. 07 커맨드를 받은 서버는 패킷번호를 보고 그 위치의 패킷부터 전송하기 시작한다. 이 미디어 데이터를 받은 클라이언트는 재생을 시작한다.

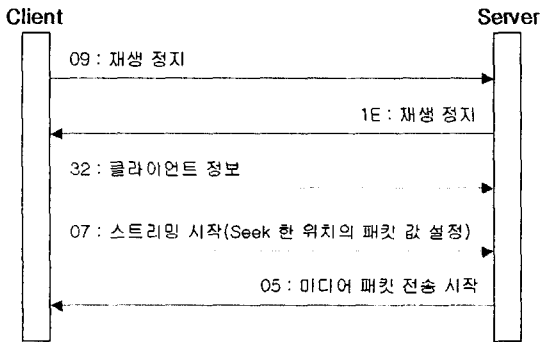


그림 3 Seek를 위한 커맨드의 전송 순서

4. 실행 결과

본 논문에서 제시한 재생기는 펜티엄4-1.3GH시스템에서 리눅스 커널 2.4.20-8 운영체제와 XFree86 4.3.0의 X11으로 실행되며 그림 4와 같은 실행 결과를 보여준다. 재생하는 파일은 WMV9 포맷으로 재생시간은 1시간 1분 29초, 총 패킷의 수는 89616개이며 하나의 패킷의 길이는 1444바이트이다.

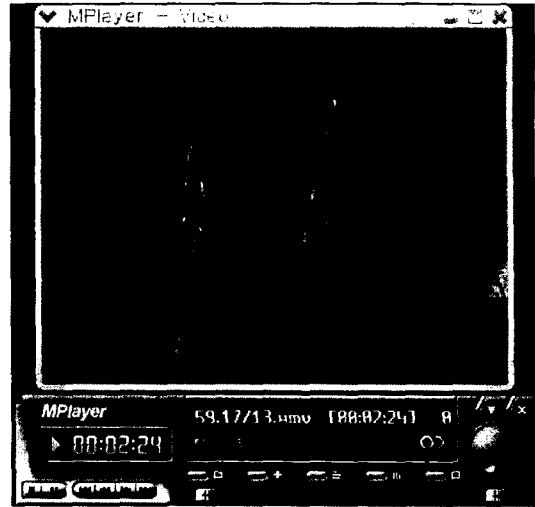


그림 4 재생 결과

5. 결론 및 향후 연구 과제

본 논문에서는 MMS 프로토콜을 지원하는 공개 운영체제 기반의 재생기를 제안하였다. 이를 위하여 기존에 리눅스 기반의 공개 프로젝트를 기반으로 그들의 성능과 문제점을 분석하였으며, Windows Media Player 수준의 스트리밍 서비스를 제공하기 위하여 WMV9 포맷 지원, Seek와 같은 VCR 컨트롤 기능을 강화 등에 초점을 두어 스트리밍 재생기를 구현하였다.

본 논문에서 제시한 재생기는 MMS 서버가 동작하는 윈도우 환경과 이질적인 운영체제에서 스트리밍 서비스를 제공할 수 있으며, 사용자에게 제공되는 VCR 컨트롤을 이용하여 서버와의 인터랙션을 수행할 수 있다.

향후 연구과제로 MMS 프로토콜에서 분석되지 않은 값이 하는 역할을 알아낼 때 마다 재생기의 지속적인 업데이트가 필요하며 서버와의 연결 설정 시 요구되는 콘텐츠 보호와 관련된 부분도 구현이 되어야 한다. 또한 이러한 재생기를 최적화하여 셋탑박스나 무선 단말과 같은 내장형 운영체제 환경에서 구현함으로써 궁극적으로 MMS 프로토콜을 이용한 스트리밍 서비스가 다양한 환경에서 제공될 수 있을 것이다.

참고문헌

[1] <http://www.microsoft.com/windows/windowsmedia/default.aspx>
 [2] <http://sdp.ppona.com/>
 [3] <http://www.xinehq.de/>
 [4] <http://www.mplayerhq.hu>
 [5] <http://www.videolan.org>