

Adaptive MPEG-4 Video-Layered Streaming 시스템의 설계 및 구현

신정아*, 김상형*, 김현정*, 이흥기*, 이성인*, 유관종*, 김두현**
*충남대학교 컴퓨터학과
**한국전자통신연구원
e-mail:jasinn@cs.cnu.ac.kr

The Design and Implementation of Adaptive MPEG-4 Video-Layered Streaming System

Jeong-Ah Sinn*, Sang-Hyoung Kim*, Hyun-Jung Kim*, Heung-Ki Lee*, Sung-In Lee*, Kwan-Jong Yoo*, Du-Hyun Kim**

*Dept of Computer Science, Chung-Nam University
**Electronics and Telecommunications Research Institute

요 약

인터넷의 급속한 확산과 통신 기술의 발전으로 이용자들은 더욱 많은 콘텐츠를 요구하고 있고 이에 멀티미디어 스트리밍 기술이 많은 관심을 받고 있다. 이에 본 논문에서는 기존에 연구되어진 MPEG-4 Layering 기술을 바탕으로 용량이 큰 멀티미디어 데이터를 하아의 스트림으로 전송하는 방법이 아닌 여러 개의 스트림으로 분할하여 가변적인 네트워크 환경에 맞추어 전송할 수 있도록 하였으며, 클라이언트 측에서 QoS Level을 측정하여 이후 수신되는 스트림에 적용될 수 있도록 구현하였다.

1. 서론

인터넷의 급속한 확산과 통신 기술의 발전으로 인해 사용자들의 정보 이용이 기존의 텍스트 위주의 정적인 것으로 동적인 것으로 변하고 있다. 이러한 변화로 인해 동적인 멀티미디어 데이터 시스템에 대한 연구가 활발하게 진행되고 있다.

멀티미디어 데이터 시스템에서는 유동적인 네트워크의 변화에 따라 패킷 손실이나 병목 현상 등으로 시간적 제약을 지키지 못하는 문제가 발생할 수 있으므로 시스템 구성 시 전송 모듈이나 전송 후 발생할 수 있는 에러에 대한 처리 모듈이 문제를 해결하는데 중요한 역할을 하게 된다.

기존에 이용되던 텍스트나 정지화상 등의 정적인 데이터는 이러한 데이터를 전송함에 있어서 시간에 큰 구애를 받지 않았으나, 멀티미디어 데이터의 경우 시간의 연속성에 의존하기 때문에 전송 및 수신 시간에 따라 데이터의 내용이 달라질 수 있다. 특히, 본 논문에서 이용하고자 하는 MPEG-4의 경우 디코딩에 있어서 다른 프레임들을 참조하므로 어느 하나의 프레임의 손실이 하나만으로 그치는 것이 아니라, 손실된 프레임을 참조하는 다른 프레임

들의 연속적인 손실을 의미하므로 중요하다. [1]

이에 본 논문에서는 지금까지 연구해 온 MPEG-4 시스템에 대한 연구 자료를 바탕으로 유동적인 네트워크 환경에 적용할 수 있도록 MPEG-4 데이터를 Layer별로 나누는 기술을 이용[2]하여 온라인 상에서 서버에서 클라이언트의 스트리밍 서비스 가능하도록 Adaptive MPEG-4 Video-Layered Streaming 시스템을 위한 서버 및 클라이언트 모듈을 구현하고 문제점을 해결하는 내부적인 모듈에 대해 논의한다.

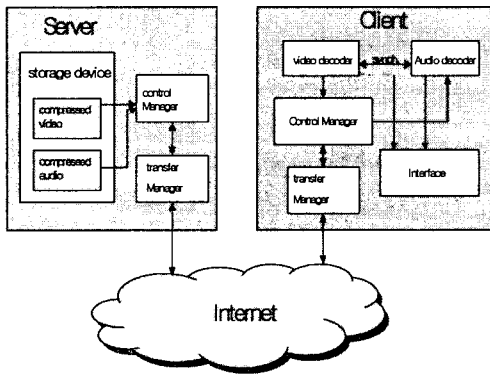
2. 관련 연구

앞서 제기한 문제점들은 멀티미디어 데이터를 다루는 시스템의 구현상의 방법이나 데이터의 압축 포맷에 따른 방법, 전송 프로토콜이나 그에 따른 전송 부분 구현 및 QoS(Quality Of Service) 모듈 방식 등에 따른 개선되어질 수 있다.

2.1 멀티미디어 스트리밍 시스템

일반적인 멀티미디어 스트리밍 시스템은 데이터를 제

공하는 서버와 이를 제공받는 클라이언트로 이루어지고 내부적으로는 용량이 큰 멀티미디어 데이터를 압축하거나 나누어 저장하는 기술을 접목한 저장 모듈, 효율적인 멀티미디어 스트림 서비스를 위한 전송 모듈, 네트워크 상황이나 클라이언트 환경에 맞춰 좀더 안정적인 멀티미디어 스트림 서비스를 제공하기 위한 QoS 관리 모듈, 비디오와 오디오 데이터 간의 동기화 모듈, 서버의 스트리밍 서비스 목록이나 플레이(play)창과 그 외 여러 메뉴 등을 이용할 수 있게 보여지는 사용자 인터페이스 부분으로 이루어진다.[3] <그림 1>은 이러한 모듈들로 이루어진 일반적인 멀티미디어 스트리밍 시스템을 보여준다.



< 그림 1. 일반적인 멀티미디어 스트리밍 시스템 구조 >

2.2 전송 프로토콜

새로운 스트리밍 기법을 도입해 기존의 TCP와 UDP를 병행하여 구현하거나, 멀티미디어 데이터 전송을 위한 RTP와 컨트롤을 위한 프로토콜인 RTCP, RTSP등을 이용하여 구현한다. [4] RTP를 이용하거나, UDP를 이용시, 프로토콜의 특성상 나타날 수 있는 패킷 손실 같은 문제를 해결하여 효율성을 높이기 위한 연구가 진행 중이다.

2.2.1 UDP

대개의 멀티미디어 스트리밍 전송 시스템에서 이용되고 있는 방법이다. 스트림 요청이나 QoS Level 정보 등의 데이터 전송에 중요한 역할을 하는 정보는 TCP를 이용하여 전송하고 실제 멀티미디어 데이터는 UDP를 이용하여 전송한다. 이것은 연결 지향형인 TCP를 이용하여 안정적이고 신뢰성 있는 전송을 보장하기 위해서가 아니라, 시간이 중요한 제약이기 때문에 시간에 대한 보장을 위하여 UDP를 이용하는 것이다.

TCP와 UDP를 전송 시스템에 적용할 경우, 기존에 데이터 전송 방법에서 크게 달라지는 점이 없다. 일반적으로 구현되어 사용되고 있는 설계 구조에 따라 구현이 가

능하다.

2.2.2 RTP

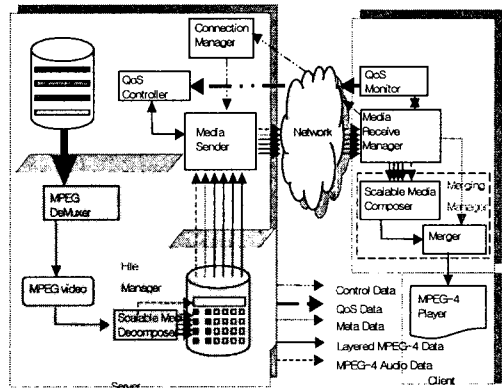
멀티미디어 데이터 전송에 적합하도록 고안된 RTP 프로토콜은 UDP의 빠른 데이터 전달 특성을 이용하였으나, 그 자체만으로는 QoS 보장이나 신뢰성을 제공하지는 못하므로 RTCP(Real-time Control Protocol)나 RTSP(Real-time Streaming Protocol) 등의 컨트롤 프로토콜들을 동시에 이용하여 구현한다.

RTP는 TCP/UDP 상위에서 있는 프로토콜로서 일반적으로 RTP는 UDP 프로토콜 상에서 실행되고[5], 컨트롤 프로토콜은 TCP 프로토콜 상에서 실행된다. RTCP는 RTP가 주기적인 전송의 보장이나 서비스 질에 관한 보장이 없다[6]는 단점을 보완하기 위해서 이용한다.

3. 시스템 설계 및 구현

본 논문에서 소개하는 시스템의 전체적인 구조는 다음 <그림 2>와 같다. 서버는 MPEG-4 파일을 Layer별로 나누는 기술을 이용[2]하여 MPEG-4 데이터를 저장하는 관리자와 전송을 담당하는 관리자, QoS를 처리하는 관리자로 크게 나뉘지며, 클라이언트는 서버로부터의 멀티미디어 데이터를 수신하는 관리자와 QoS를 체크하고 서버에 보내주는 관리자, 받아들이는 파일들을 디코딩하고 머징하기 위한 관리자로 나누었다.

MPEG-4 파일을 Layer별로 나누어 저장하는 기술 자체로 클라이언트의 수신 환경에 따라 Layer 차이로 QoS를 보장해주는 역할을 한다. 그러나, 기존 연구에서는 사용자 인터페이스에 의해 사용자가 수신 환경을 고려하여 Layer의 양을 직접 셋팅 해주었기 때문에 이용에 어려움이 있었다. 이에 본 시스템에서는 클라이언트에 수신되는 스트림과 스트림 전송 전 미리 받은 Info 파일을 바탕으로 QoS Level을 모듈 내에서 측정하여 이후 수신되는 스트림에 적용될 수 있도록 했다.



< 그림 2. Adaptive MPEG-4 Video-Layered Streaming System >

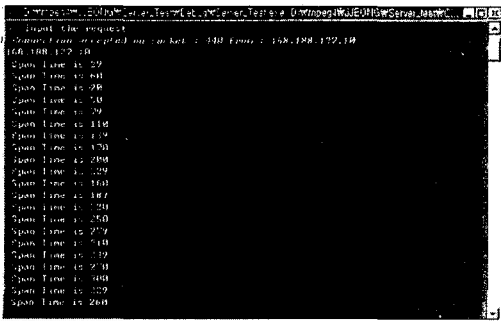
3.1 서버(Server)

■ 저장 관리자(File Manager) : 하나의 MPEG-4 스트림을 Layer별로 나누는 기술을 이용한 디바이더 모듈을 이용하였다. 여러 개의 데이터로 나누는 것은 전송 시 QoS를 보장하는 측면이 강하다. QoS Level에 대한 측정 은 서버 단독으로 처리하기 어려우므로 보통은 그 측정이 클라이언트에서 이루어진다.[2] 즉, 클라이언트에서 측정된 수용 가능한 수신 량을 서버 시스템으로 전송하여, 다음 전송 시 적용되는 화질에 대한 QoS 보장과 유동적인 대역폭에 따라 수신되는 량의 조절이 가능하다.

■ 연결/전송 관리자(Connection Manager) : TCP와 UDP를 이용하여 전송 관리자를 구성하였다. 서버 측에서 Listen, 클라이언트의 요청을 처리, 파일 전송 전담하는 각각의 쓰레드를 생성하는 Media Sender이다. 데이터 전송은 GoV 단위로 이루어진다. GoV는 하나의 레이어 파일을 가리키는 단위가 아니라, T1F1에서부터 T3Fn에 나뉘어 저장되어 있는 VOP의 모음이다. 디바이더 모듈에 의해 생성된 데이터의 Info 파일에서 GoV와 QoS 관리자에서 넘겨진 Level 정보를 바탕으로 전송한다.

■ QoS 관리자(QoS Controller) : 클라이언트에서 보내지는 QoS Level을 받아서 이후 보내지는 Layered MPEG-4 데이터에 적용할 수 있도록 전송 관리자와 저장 관리자에 전달해주며, 서비스되고 있는 데이터에 대한 로 그를 처리한다.

서버가 동작하고 클라이언트의 요청이 들어오면 클라이언트의 버퍼를 고려하여 서버 측에서 전송하는 시간을 체크한다. <그림 3>은 서버가 클라이언트에 MPEG-4 스트림을 서비스하는 모습으로 화면에 나타난 내용은 체크된 시간을 보여준다.



< 그림 3. 서버 동작 결과 >

3.2 클라이언트(Client)

■ 수신 관리자(Media Receive Manager) : 요청한 멀티미디어 데이터 수신(Media Receiver)을 하면서 메모리 버퍼와 파일 버퍼에 수신된 데이터를 저장한다. 버퍼의 구현에는 더블리 링크드 리스트(doubly linked list)를 이용하였다. 구현이 쉽고 전후 방향으로 이동할 수 있으므로 UDP를 이용하여 수신 받은 경우, Layered MPEG-4 데이터의 순서가 뒤바뀔 수 있기 때문에 머징 전에 이를 해결할 수 있었다. 두 가지 형태의 버퍼를 이용하는 이유는 메모리 버퍼의 빠른 속도를 이용하고 파일 버퍼의 넉넉한 용량을 이용하기 위해서 이다. 또한 MPEG의 특성상 어느 일정량의 전송이 이루어진 후 머징이 이루어지므로 현재 머징 되고 있을 경우, 머징에 이용되는 버퍼가 가득 차 수신된 데이터의 버퍼로부터 넘겨지지 못할 경우를 위해서도 두 개의 버퍼를 이용하는 것이 효율적이다.

■ QoS 관리자(QoS Monitor) : 수신 관리자를 통해 전송 상태를 체크하면서 QoS Level을 설정하여 다시 서버 측에 정보를 보낸다. Level 설정 시 데이터 수신 처음에 받아들인 Info file을 기초로 각 GoV의 플레이 타임을 알아내고 받아 들여서 버퍼에 있는 데이터와 머징로 보내져 머징 시 이용되는 버퍼에 있는 데이터의 계산으로 현재의 QoS Level을 앞으로 수신할 때 이용할 QoS Level로 설정하는 알고리즘은 <그림 4>와 같다.

■ Merger 관리자(Merging Manager) : 수신된 멀티미디어 데이터가 저장되어 있는 버퍼에 접근하여 읽어서 머징과 디코딩을 위한 버퍼에 채워 넣는다. 여러 개의 Layering되어진 파일을 하나의 YUV 파일로 생성해 주는 형태로 구현되어 있으나, 향후 플레이 할 수 있는 디스플레이어가 붙게 되면 하나의 YUV 파일로 생성되는 것이 아니라, 플레이 가능한 일정량이 만들어지면 바로 보여질 수 있는 형태가 된다.

클라이언트가 동작하면 서버에 요청을 보내고 수신되는 데이터를 버퍼링 하면서 머징 하는 과정을 보여준다. <그림 5>는 클라이언트의 동작 결과를 보여주는 화면으로 머징 시 참조되는 로그 파일로부터 읽어들이는 VOP 별 시간과 이를 디코딩 하는데 이용되는 다른 VOP의 정보를 보여준다.

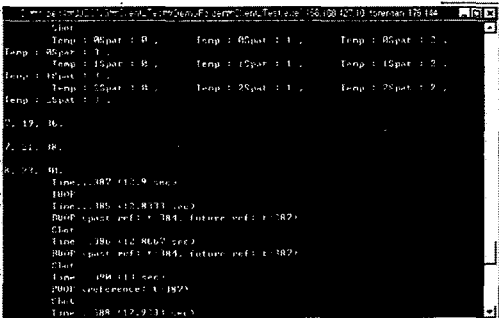
```

QoS_Level_Setting(int GoVNo)
{
    char buffer[100]
    int LastLayer, LyrSqnNo, RecvSqnNo, Temp, Spat
    long LyrSize

    // Info 화일로부터 현위치를 알아온다.
    LastLayer = StrmPosition();
    // 전역 변수인 mGoV를 하나 증가시킨다.
    mGoV = mGoV + 1

    for(Temp = 0 ;Temp < MAXTEMP ; Temp++)
    {
        // 현재 위치에서의 Spatial Number 만큼 Loop를
        // 돌면서 실제 받은 패킷 사이즈와 Info 화일에서의
        // 사이즈를 비교해서 현재까지 받아들인 Spatail
        // Number로 QoS Level을 셋팅한다.
        for(Spat = 0;Spat<ExistedLyrNo(Temp);Spat++)
        {
            LyrSize = getStrmSize(Temp, Spat)
            LyrSqnNo = (int) LyrSize / mPacketSize
            RecvSqnNo = getRecvSqnNo(mGoV , Temp,
Spat)
            if ( LyrSqnNo != RecvSqnNo) break
        }
        // 최상위 레이어의 패킷이 도착하지 않은 경우,
        // 하위 레이어의 QoS Level을 -1로 셋팅한다.
        if ( Spat ) mQoSLevel[mGoV][Temp] = Spat - 1
        else
            for (;Temp < MAXTEMP ; Temp++)
                mQoSLevel[mGoV][Temp] = -1
            break
        }
    // QoS Log 화일에 기록
}
    
```

< 그림 4. 클라이언트 QoS Level Setting Code >



< 그림 5. 클라이언트 동작 결과 >

4. 결론 및 향후 과제

지금까지 Adaptive MPEG-4 Video-Layered Streaming 시스템의 구성과 내부적으로 이용한 전송 방법, 전체적인 시스템 구조에 대해 서술했다. 본 논문에서 구현한 시스템은 기존에 큰 용량의 스트리밍 전송 시 가변적인 네트워크 상태로 인해 나타나는 문제를 여러 개의 스트림으로 분할하여 전송하는 방법을 이용하여 해소하고 이러한 분할된 계층을 이용해서 클라이언트 측에서 QoS Level을 설정하는 스트리밍 시스템의 온라인화가 이루어진 것에 큰 의미가 있다.

멀티미디어 데이터 서비스를 기존의 멀티미디어 시스템과 접목하고 위한 데이터 전송 방법은 UDP를 이용했으나, 앞서 소개한 RTP 같은 멀티미디어 데이터를 위해 만들어진 프로토콜을 이용하는 것도 가능성이 있다.

QoS 를 클라이언트 측에서 수신된 정보를 바탕으로 측정 후 다시 서버에 측정치만큼의 데이터를 요청하도록 구현하였기 때문에 QoS 정보를 서버측으로 전송 시 네트워크 상태가 급변한 경우에 대한 대책이 미흡하다. 향후 이러한 문제점을 수정하기 위해서 실시간 네트워크 상태에 대한 정보를 바탕으로 QoS 설정이 가능한 방식이나 Error Control 등에 대한 연구가 필요하다.

참고문헌

- [1] Universal Streaming 서비스를 위한 MPEG-4 스케일터빌리티 제공 기술 연구, 한국전자통신연구소 최종 연구보고서, 2001년 11월
- [2] 이흥기 외 6인, "이질적 환경에서의 Layering 되어진 멀티미디어 스트림 서비스, 한국정보과학회 춘계학술 발표대회 제 29권 제 1호, 2002년 4월
- [3] streaming video over the Internet : Approaches and directions, IEEE transactions on circuits and systems for video tech, vol.11, no.3, march 2001
- [4] 한국인터넷방송협회, www.kwn.or.kr
- [5] Terms Korea, www.terms.co.kr
- [6] RTP 이용 강좌, www.edu80.com