

이질적인 네트워크 환경에서 MPEG 비디오 서비스를 위한 미디어 계층화 기법

유 우 종[†] · 이 흥 기^{††} · 이 성 인^{†††} · 이 중 인[†] · 유 관 종^{††††}

요 약

연속성과 대용량의 특징을 가지는 MPEG을 이용한 비디오 서비스에서는 대용량의 네트워크 대역폭을 필요로 하게 된다. 대부분의 사용자들은 이질적인 네트워크 환경을 가지고 있기 때문에 동일한 비디오 스트림을 이용하여 모든 사용자에게 서비스 한다는 것은 네트워크 자원의 효율성 측면에서 상당한 낭비가 있게 된다. 이를 해결하기 위해 비디오 서비스를 요청한 사용자의 환경을 고려하여 그에 적절한 용량의 비디오 데이터를 전송할 수 있는 기술이 비디오 서비스 시스템에 필요로 하게 되었다. 따라서, 본 논문에서는 불필요한 네트워크 자원의 낭비를 막으면서 사용자에게 적절한 MPEG 비디오 스트림을 서비스 해주기 위해, TFS기법을 이용하여 MPEG 비디오 스트림을 계층화 시킨 후 네트워크 QoS에 따라 선택적으로 전송하는 기법을 제안한다. 본 논문에서 제안하는 모델을 이질적인 환경을 갖는 망들이 서로 연동되는 인터넷에 적용하여 인터넷 VOD 나 인터넷 방송 등의 서비스에 응용한다면, 대역폭 변화를 동적으로 반영하며 실시간으로 비디오를 서비스할 수 있는 환경을 구축할 수 있고, 불필요한 통신 자원의 낭비를 막음으로써 통신 자원의 효율성을 향상시킬 수 있을 것으로 기대한다.

Media Scaling Technology for MPEG Video Service on Heterogeneous Network Environment

Woo-Jong Yoo[†] · Heung-Ki Lee^{††} · Sung-In Lee^{†††} · Jung-In Lee[†] · Kwan-Jong Yoo^{††††}

ABSTRACT

The MPEG Video Service of having a property of continuity and large capacity needs large network capacity. Because most of users have a heterogeneous network environment, it's not efficient way that all users have same size of video data to provide them with real time video service. Therefore, for the purpose of an efficient and appropriate utilization of network resources, it requires to develop and deploy a new scalable transmission technique in consideration of respective network environment and individual clients computing power. The purpose of this paper is to develop a technology that can adjust the amount of data transmitted as an MPEG video stream according to its given communication bandwidth, and a technique that can reflect dynamic bandwidth while playing a video stream. For this purpose, we propose a TFS (Temporal-Fidelity Scaling) technique that splits the MPEG video stream into various substream according to picture type or resolution. Those methods proposed her can facilitate an effective use of network resources, and provide multimedia MPEG video services in real-time with respect to individual client computing environment

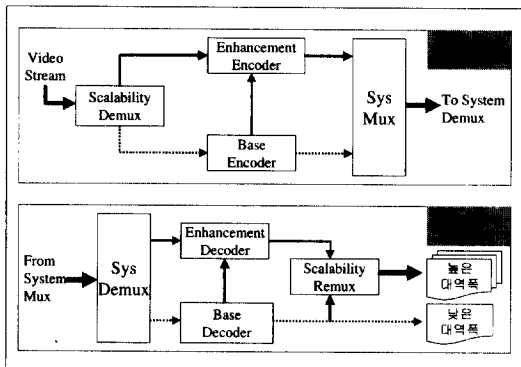
* 본 연구는 한국전자통신연구원의 지원을 받았음
† 정 회 원 : 대전보건대학 컴퓨터정보처리과 교수
†† 준 회 원 : 충남대학교 컴퓨터학과

††† 정 회 원 : 대전보건대학 멀티미디어사무용과 교수
†††† 정 회 원 : 충남대학교 정보통신공학부 교수
논문접수 : 2000년 8월 1일, 심사완료 : 2000년 12월 20일

1. 서 론

저장매체와 네트워크 기술의 발달은 컴퓨터 네트워크를 이용한 VOD(Video-on-Demand) 응용을 가능하게 하였다. 이와 같은 응용은 실시간성과 멀티미디어 데이터 스트림(예, 비디오 또는 오디오)을 전송하기 위한 대용량의 네트워크 대역폭이 필요하다는 특성을 가지고 있다. 따라서, 이와 같이 대용량 데이터의 전송을 실시간으로 전송하기 위해서 네트워크와 저장매체의 효율적인 사용을 위한 새로운 기술이 필요하게 되었다.

계층적 코딩기법은 이러한 새로운 기술 중의 하나로 멀티미디어 데이터를 기본 레이어(Base Layer)와 고위 레이어(Enhancement Layer)으로 나누어 멀티미디어 데이터를 재생하는 사용자의 컴퓨터 환경이나 네트워크 환경을 고려하여, 필요한 계층을 선택한 후 복호화하는 기법이다(그림 1)[1-3].



(그림 1) 스케일러빌리티 코딩 구조

지금까지의 연구 결과를 살펴보면 동일한 비디오 스트림을 사용자 대역폭의 QoS(Quality of Service)에 따라 서버에 여러 개 중복하여 저장한 후 클라이언트와 연결 설정시 결정된 QoS에 적합한 비디오 스트림을 선택하여 서비스 함으로써 저장매체의 기억공간을 낭비하였다. 최초 연결 설정시에 결정한 QoS가 재생 중에 변하게 되어 변화된 QoS를 반영하려면, 처음부터 변화된 QoS로 비디오를 다시 보아야 하는 문제점을 가지고 있다. 결국 QoS 변화를 동적으로 반영할 수 있는 방법이 없어, 사용자에게 최적의 서비스를 제공하지 못했다. 또한, 국제 동영상 압축의 표준인 MPEG-2 비디오에서는 스케일러빌리티로 SNR(Signal to Noise Ratio), 공간적(Spatial), 시간적(Temporal), 그리고 Data

Partitioning의 4가지 형태로 계층화된 코딩기법을 제공하고 있으나, 이 스케일러빌리티를 구현할 경우 인코더가 복잡해지고 압축 효율이 떨어질 뿐만 아니라, 기본 레이어만으로 된 데이터 양이 커서 낮은 대역폭 환경에서 이용하기가 어렵다[4-10]. 이러한 이유를 해서 실제로 스케일러빌리티를 지원해주는 상용화된 MPEG 인코더의 사례는 찾아보기가 힘든 형편이다.

따라서, MPEG-2에서 제공되는 4 가지 형태의 스케일러빌리티 대신에, MPEG-2의 기본 레이어 만을 계층적으로 코딩하여 통신 대역폭 변화에 동적으로 적용할 수 있는 새로운 연구가 요청되었다[11]. 이를 위해서는 저장 데이터를 QoS에 따라 송수신하기 위해 서버측에 저장되어 있는 미디어를 여러 개의 레이어로 분리하는 기능과, 현재의 QoS에 적합한 레이어를 사용자 간섭없이 자동으로 선택하는 기능, 그리고 선택된 레이어를 송신하고 레이어 별로 수신된 스트림을 하나로 통합하는 기능 등이 필요하다[12-14].

따라서, 본 논문에서는 QoS 적용형 시스템 모델을 서버 및 클라이언트로 구분하여 설계한 결과를 바탕으로, 압축 저장된 MPEG 비디오 스트림을 QoS에 따라 다양하게 서비스할 수 있도록 시간적으로는 픽처를 픽처 타입에 따라 계층화하고, SNR적으로는 화면을 화질의 선명도에 따라 계층화하는 TFS(Temporal-Fidelity Scaling)기법을 제안한다. 또한, 계층화된 비디오 스트림을 재생 가능한 형태로 취합하는 분할기와 이들을 구현하는 알고리즘에서 사용하는 메타정보를 생성하고 이용하는 기법도 함께 제안한다. 본 논문에서 제안하는 기법을 이용하여 MPEG 비디오 스트림을 계층화하면 시간적으로는 3개의 레이어로 분할되고 화질적으로는 5개의 레이어로 분할된다. 이 2가지 방법을 조합하게 되면 총 15개의 레이어가 만들어 지며, 이를 자신의 QoS 환경에 맞게 활용하면 대역폭이 낮은 환경에서도 실시간으로 MPEG 비디오 스트림을 재생시킬 수 있다. 또한, 클라이언트의 QoS에 적합한 레이어로 서비스를 받게 됨으로써 통신 자원의 낭비를 막고 성능의 효율성을 제고 시킬 수 있을 것으로 기대한다.

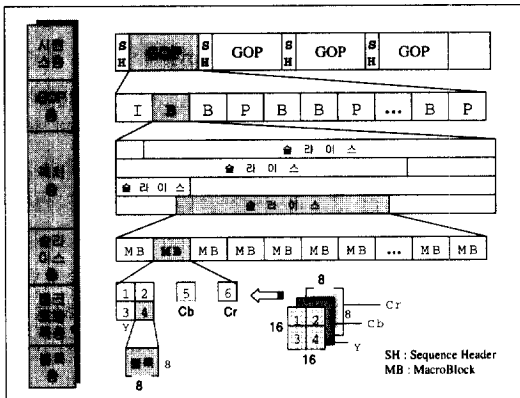
본 논문의 구성은 다음과 같다. 먼저 2장에서는 MPEG의 특징에 대해 간단히 소개하며, 3장에서는 제안된 시스템의 모델을 소개하며, 4장에서는 비디오 데이터를 시간적, SNR적으로 계층화 시키는 TFS기법에 대해 기술한다. 5장에서는 제안한 시스템을 이용한 시뮬

레이션 결과를 보여주며 마지막으로 6장에서는 결론을 맺는다.

2. MPEG 비디오의 개요

2.1 MPEG비디오의 특징

MPEG비디오는 시퀀스 층, GOP(Group of Picture) 층, 픽처 층, 슬라이스 층, 매크로블록 층, 블록 층의 계층구조로 이루어져 있다. 시퀀스 층은 일련의 같은 속성을 갖는 화면 그룹으로 화면의 크기와 프레임 율, 비트 율, 버퍼크기 등을 설정한다. GOP층은 I픽처 층에서 다음 I픽처까지의 픽처들을 포함한다. 픽처 층은 한 장의 화면을 나타내며 이 픽처 층은 코딩방식에 따라서 I 픽처, P 픽처, 그리고 B픽처로 나누어지게 된다 [6]. 슬라이스 층은 데이터의 손실을 고려하고 에러전이를 막기위해 하나의 화면에 포함된 매크로블록 들의 그룹으로 구성된다. 매크로블록 층은 1616 화소로 움직임 보상을 수행하는 기본단위이며, 블록 층은 88화소로 구성된 최소의 단위이다(그림 2).



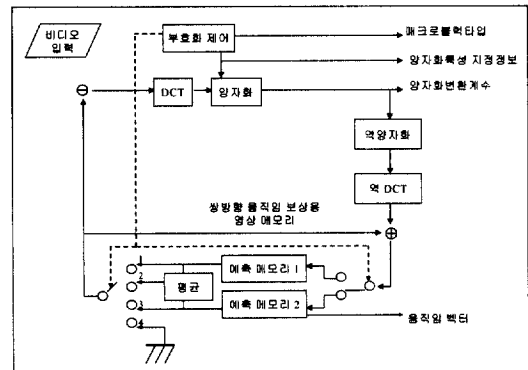
(그림 2) MPEG 영상 데이터의 계층구조

픽처는 해당 화면정보만으로 부호화되는 화면으로, 프레임간 예측을 쓰지 않고 생성된다. GOP내에서는 랜덤 액세스를 위해 최저 한 장의 I 픽처가 필요하고 I 픽처 내의 모든 매크로블록 타입은 인트라이며, 한 화면의 정보만을 이용하여 부호화하기 때문에 다른 픽처에 비해 가장 많은 데이터를 가지게 된다. P픽처는 I픽처 또는 P픽처로부터 예측을 수행함에 따라 생성되는 화면으로, 일반적으로 P픽처 내의 매크로 블록 타입은 인트라 프레임(Intra Frame; 화면내 부호화)과 인터 프

레이م(Inter Frame; 순방향 프레임간 예측화면) 양쪽을 모두 포함한다. B픽처는 MPEG의 특징인 쌍방향 예측에 의해 생기는 화면이다. MPEG구조의 최하위 층인 블록 층은 DCT(Discrete Cosine Transform)를 수행한 DCT 계수를 포함하고 EOB로서 종료되며 계수의 VLC(Variable Length Code)가 64개 있을 때에도 EOB(End of Block)를 추가하게 된다. 인트라 DC는 다른 AC성분과 다르게 독자적인 VLC를 사용한다.

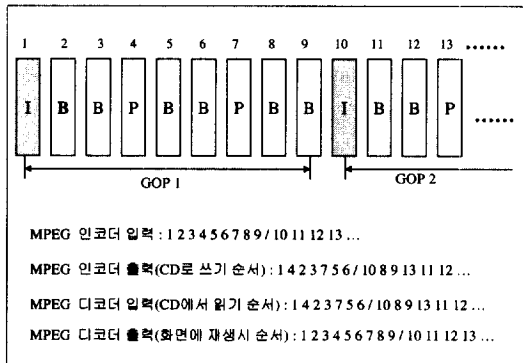
2.2 인코딩 및 디코딩 과정

인코더는 입력된 데이터를 MPEG-2 비트 열로 만드는 것으로 MPEG-2 비디오 표준에는 인코더의 역할과 기능에 대해 자세히 언급하고 있지 않다. 이는 비트 열을 만드는 방법보다는 어떤 비트를 만드는 지가 중요하기 때문이다. 우선 입력 데이터는 양방향 예측을 하는 B프레임 때문에 프레임 재순서화를 거치게 된다. 프레임 재순서화를 거친 데이터는 DCT를 이용해 DCT 계수로 코딩되고, 양자화를 거친 후 가변장부호화 된다. 대부분의 압축은 DCT보다는 양자화와 가변장부호화에서 일어나게 되며 이 결과에 따라 압축율이 결정된다. 여기의 출력 데이터는 버퍼를 거쳐 비트 스트림으로 출력되는데, 이 과정이 인트라 블록 코딩이다. 양자화를 거친 데이터가 역양자화와 IDCT(Inverse DCT)를 거치면 다시 DCT를 거치기 전의 데이터와 거의 같은 데이터로 복원된다. 이때 DCT를 거치기 전과 데이터의 차이는 양자화 오류로 이 데이터들은 프레임 메모리와 움직임 보상 및 예측기를 거쳐서 움직임 벡터를 추출하게 된다. 이 값과 IDCT를 결합한 값으로 데이터를 복원한다. 이 데이터를 재생된 데이터라고 한다[6]. 이 과정이 인터 블록 코딩이다(그림 3).



(그림 3) MPEG 부호기의 인코딩 과정

비디오 디코딩 과정이라는 것은 인코더에서 코딩된 데이터 즉, MPEG-2 비디오 스트림으로 만들어진 데이터를 바로 화면에 출력할 수 있는 데이터로 만드는 것을 의미한다. 대체적으로 디코딩 과정은 인코딩 과정의 역순으로 각 과정을 간단히 살펴보면 우선, 코딩된 데이터를 저장할 수 있는 버퍼가 있고, 버퍼를 거친 데이터는 가변장복호화 과정을 통해 인트라 블록에서의 DC계수와 나머지 AC계수들로 나누어 디코딩한 다음 역스캔(Inverse Scan)을 한다. 이때 역스캔은 zigzag 스캔과 대체 스캔 중 하나이다. 그런 후 역양자화와 IDCT(Inverse DCT)를 거치면 디코딩된 샘플이 나오게 된다. 이때 움직임 보상을 하기 위해서는 참조 프레임이 필요한데 참조 프레임은 프레임 메모리에 저장되어 있다. 물론 여기서 나온 디코딩된 샘플들은 화면에 디스플레이 하기 위해서는 프레임 재순서화(Reordering) 과정을 거쳐야 한다[6]. 프레임 재순서화 과정이 (그림 4)에 나타나 있다.



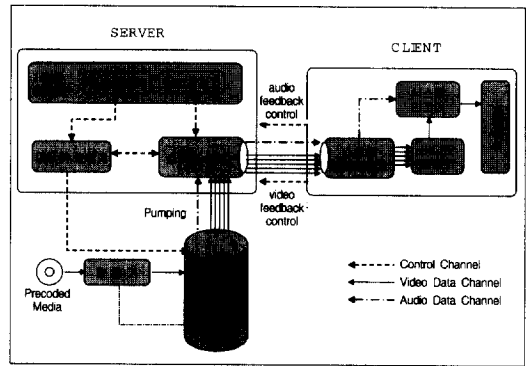
(그림 4) MPEG의 부호화순서와 재생순서

이상에서 살펴본 MPEG 비디오의 특성과 인코딩 및 디코딩 과정은, 본 논문에서 제안하는 TFS 기법을 이용한 계층화 기법에 중요한 개념이다. 즉, 픽처 타입은 시간적 계층화의 기본 단위이고, DCT를 수행하는 기본 단위인 블록 층은 화질 선명도에 따른 계층화 기법의 기본 단위이다.

3. 시스템 모델

본 논문에서 인터넷과 같은 이질적인 네트워크 환경 하에서 VOD 서비스를 위해 제안하는 시스템 모델은 서버와 클라이언트 구조로 나뉘어져 있다(그림 5). 이

구조는 기존의 방법과는 달리 하나의 MPEG 비디오 스트림을 QoS에 따라 여러 개 중복 저장하는 것이 아니라 계층적으로 분할하여 저장함으로써 저장매체의 기억공간 낭비를 방지할 수 있고, 재생중 QoS 변화를 반영할 수 있어 사용자에게 최적의 서비스를 가능하게 한다. 또한, 기존의 인코더와 디코더 및 MPEG 파일의 수정 없이 본 논문에서 제안하는 방법을 적용할 수 있다.

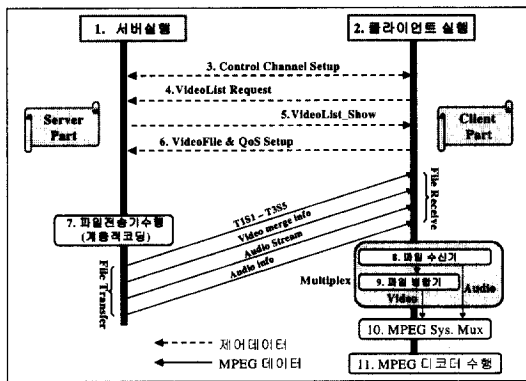


(그림 5) 스케일러블 VOD 서비스 모델

그림에 있는 각 모듈들의 의미와 VOD 서비스를 위한 시나리오는 다음과 같다. 서버는 압축된 MPEG 비디오 데이터 스트림을 분할기(Decomposer)를 이용하여 계층화하여 저장공간에 보관하고 있으며, 클라이언트는 자신의 단말기에서 온라인으로 비디오 서비스를 받고자 하면 서버 매니저(Server Manager)에게 연결설정을 요청한다. 연결설정 요청을 받은 서버는 서비스 가능여부를 확인한 후 연결관리자를 통해 연결설정을 허락하며, 설정이 완료되면 자신이 보유하고 있는 MPEG 비디오 목록을 클라이언트에게 보낸다. 서버로부터 서비스 가능한 비디오 목록을 받은 클라이언트는 이중 하나를 선택한다. 그러면 스트림 클라이언트가 자동적으로 활성화되어, 현재의 통신망 대역폭 및 클라이언트 단말기 성능을 분석하여 서버와 클라이언트 간에 서비스 가능한 QoS를 결정한 후, 피드백 제어 채널(Feedback Control Channel)을 통하여 서버의 인증제어기(Admission Controller)에게 보낸다. 서버의 인증제어기는 클라이언트로 수신한 QoS에 부합되는 레이어를 선정하여 스트림 서버로 MPEG 비디오 및 오디오를 보내도록 미디어 분배기(Media Distributor)에게 지시한다. 스트림 서버는 자신이 받은 오디오 스트

림과 계층화된 비디오 스트림을 스트림 클라이언트에게 송신하며, 이를 수신한 스트림 클라이언트는 오디오 오는 시스템 인코더에게 직접 보내고 비디오 서버 스트림들은 비디오 병합기(Video Merger)에게 송부한다. 비디오 병합기는 계층화된 레이어 파일들을 재생 가능한 하나의 비디오 파일로 통합하여 시스템 인코더에게 보내며, 시스템 인코더는 자신이 수신한 MPEG 오디오 스트림과 비디오 스트림을 믹싱(Muxing)하여 하나의 MPEG 시스템 스트림을 생성한 후 플레이어에게 재생되도록 한다.

재생 중에 초기 연결 설정시에 결정했던 QoS가 변할 경우를 대비하여 스트림 클라이언트가 주기적으로 QoS를 모니터링하여 결과를 인증제어기에게 보내고, 인증제어기는 새로운 QoS에 적합한 계층화된 비디오 서버 스트림들을 스트림 서버에게 보내도록 미디어 분배기에게 지시한다. 결과적으로 클라이언트에 있는 플레이어는 항상 현재의 QoS에 맞는 MPEG 시스템 스트림을 재생할 수 있어, 사용자에게 최적의 서비스가 가능해진다. 지금까지 클라이언트가 MPEG 비디오 스트림을 서비스 받기 위해 서버와 주고받은 제어 데이터와 MPEG 데이터의 흐름도를 간략하게 그림으로 도시하면 (그림 6)과 같다.



(그림 6) 서버 클라이언트간 흐름도

이와 같은 서비스를 위해서 가장 중요한 것은 계층적 코딩 기법과 비디오 병합기 기능, 그리고 QoS 모니터링 기능 및 QoS 인증제어 기능이다.

지금까지의 연구결과에 반해 본 논문에서 제안하는 방법을 사용하면 통신 자원의 낭비를 최소화한 상태에서 성능의 효율성을 추구할 수 있다. 이에 대한 구체적인 것은 앞으로 4장과 5장에서 기술하겠다.

4. TFS (Temporal-Fidelity Scaling) 기법

TFS 기법은 프레임 율을 조정하여 픽처 타입에 따라 시간적으로 계층화하는 Temporal 스케일링 기법과 화질의 선명도에 따라 DCT 블록을 계층화하는 Fidelity 스케일링 기법으로 이루어져 있다.

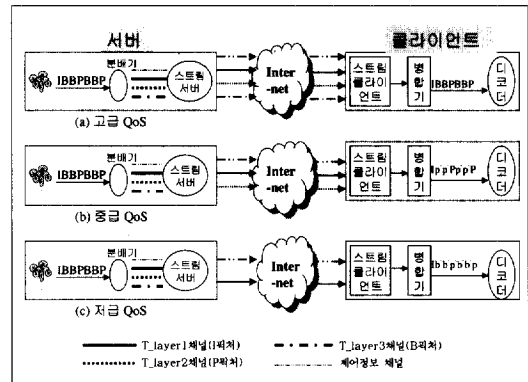
4.1 Temporal 스케일링 기법

본 기법은 I 픽처의 비율이 전체 MPEG-2 비디오 스트림 크기의 25% 내외이고 P 픽처는 35% 내외이며, B 픽처는 40% 내외인 것에 착안하여 원래의 비디오 데이터를 서버의 분할기에서 독립적으로 3개의 Layer로 나눈 것이다[2, 13, 14].

T_layer1(Temporal Layer 1)은 MPEG 비디오 스트림에서 픽처 층 이상의 헤더 정보 및 I 픽처들의 모임으로 구성되어 있고, T_layer2는 P 픽처들로 구성되어 있으며, T_layer3는 B 픽처들로 구성되어 있다. 픽처 분할 이전의 원본 MPEG 파일을 MP라 하고, 분할한 3개의 레이어 파일을 각각 MP₁(T_layer1), MP₂(T_layer2), MP₃(T_layer3)라 했을 경우에 다음과 같은 수식이 성립한다.

$$MP = \sum MP_t, \quad \text{단, } t = 1, 2, 3$$

즉, 분할된 레이어별 파일들은 서로 중복됨이 없는 직교성을 갖는다. Temporal 스케일링 기법을 이용하여 MPEG 비디오 스트림을 레이어로 분할한 후 QoS에 따라 이를 송수신 하는 서버와 클라이언트 간의 구조가 (그림 7)에 도시되어 있다.



(그림 7) Temporal 스케일링을 이용한 송수신

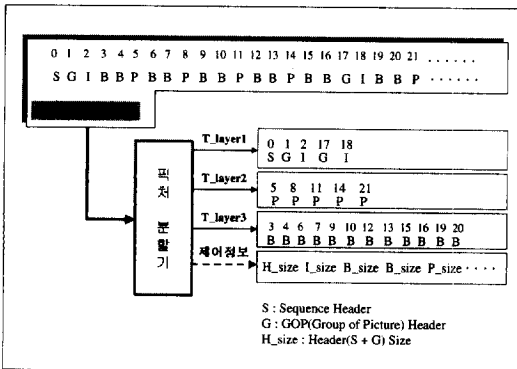
(그림 7)에 있는 각 모듈의 의미와 제어 및 데이터의 흐름을 서버와 클라이언트로 분리하여 (그림 5)와 관련 지어 살펴보면 다음과 같다.

4.1.1 서버

한 개의 스레드로 구성된 분할기는 MPEG-2 비디오 파일(확장자 *.m2v)을 입력 받아 필요한 제어정보를 만들고 원래의 비디오 데이터를 3개 레이어로 나누는 일을 하며, 총 4개의 스레드로 구성된 분배기는 제어 정보를 전송하는 1개의 스레드와 3개의 레이어를 각각 전송하는 3개의 스레드로 구성되어 있다.

① 분할기

분할기의 모듈은 디코딩의 과정과 거의 비슷하나, 파일에 저장하는 기법과 픽처의 크기를 저장하는 부분만 다르다. 먼저 입력을 받은 파일을 읽어 들여 픽처 헤더에 있는 픽처 타입으로 I, P, B를 구분하며 이를 제어정보 파일에 저장한다. 또한 다음 픽처가 나올 때까지 카운트를 하여 픽처 크기를 알아내어 역시 제어정보 파일에 저장한다. 픽처 타입으로 구분된 I 픽처는 T_layer1 파일에, P 픽처는 T_layer2 파일에, B 픽처는 T_layer3 파일에 저장한다(그림 8).



(그림 8) 픽처분할기 실행결과

(그림 8)은 (그림 5)에 있는 분할기를 좀더 상세하게 표현한 것이다. 분할기는 스레드로 구성되어 있고 비디오 데이터 스트림을 재생하기 이전에 오프라인 형태로 실행되므로 분배기와는 독립적이다.

② 분배기

분배기는 인증제어기로부터 QoS를 수신하여 현재 서비스 중인 QoS와 비교한다. 만일 수신한 QoS가 서

비스 중인 QoS와 상이하면, 5장에서 기술하는 QoS에 따른 레이어 선택 알고리즘을 실행시켜 새로운 레이어를 결정한 후, 스트림 서버에게 저장매체에서 읽어가는 비디오 스트림의 레이어를 변경하도록 지시한다. 스트림 서버는 분할기가 생성한 제어정보 파일을 참조하여 새로운 QoS에 맞는 레이어로 천이한다.

③ 스트림 서버

스트림 서버는 분배기가 넘겨준 QoS에 따른 레이어를 저장매체에서 읽어, 이를 스트림 클라이언트로 송신하는 역할을 담당한다. 서버에 있는 4개의 스레드 중에서 설정된 채널(Channel)에 따라 고급 QoS인 경우 제어정보, T_layer1, T_layer2, 그리고 T_layer3 등 4개 전송 스레드를 모두 사용하며, 중급 QoS인 경우는 4개 스레드 중에서 T_layer3 전송 스레드를 제외한 3개의 스레드를 사용하고, 저급 QoS인 경우에는 제어정보와 T_layer1 전송을 위한 2개의 스레드를 사용한다. 제어정보는 바로 전송하면 되고, 레이어별 데이터는 각 채널에 의해 연결된 레이어별 파일에서 픽처 크기만큼 읽어 들여 전송하면 된다. 따라서, 어떠한 클라이언트가 서버와 연결 설정시 사용하는 채널 수는 통신 대역폭과 재생하는 단말기 성능에 따라 달라지게 된다. 즉, QoS가 좋은 환경에서는 3개의 기본 레이어를 위한 채널에 제어정보 송신을 위한 1개의 채널을 추가하여 4개의 채널을 설정하며, QoS가 좋지 않은 환경에서는 T_layer1을 위한 채널에 제어정보 채널을 더하여 2개의 채널을 설정한 후 비디오 데이터를 송수신하게 된다. 채널 수가 많아지면 서버가 클라이언트에 더 많은 데이터를 송신하므로 클라이언트는 좀더 좋은 품질의 비디오를 재생할 수 있게 된다.

4.1.2 클라이언트

클라이언트 쪽에 존재하는 스트림 클라이언트도 스트림 서버와 마찬가지로 4개의 수신 스레드를 가지고 있다. 이들 4개의 수신 스레드는 스트림 서버에 의해 보내진 여러 개의 레이어별 데이터를 병합기가 하나로 모아서 MPEG 비디오 스트림을 만들어 주면서 실시간으로 재생시키도록, 레이어별로 비디오 스트림을 수신하는 역할을 담당한다. 고급 QoS인 경우에는 수신한 제어정보대로 각 레이어별 버퍼에서 픽처 크기만큼 읽어와서 원래의 MPEG 파일을 만들고, 중급 QoS인 경우 스트림 클라이언트로부터 받은 제어정보에서 픽처

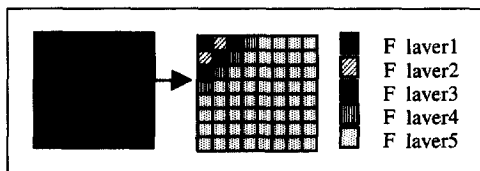
타입이 B픽처인 경우 이전 픽처와 동일한 픽처(p')임을 나타내는 Skipped 픽처를 삽입하여 준다. 저급 QoS인 경우도 받은 제어정보에서 픽처타입이 P나 B 픽처인 경우 Skipped 픽처(p')를 삽입하여 주어 재생 가능한 MPEG 비디오 스트림을 만든다. Skipped 픽처는 중간에 생략된 픽처 위치에 삽입되어 이전 픽처와 동일한 픽처임을 나타내고 재생시간의 프레임율을 맞추기 위해 사용하는데, 화면을 부호화하지 않고 이전 픽처를 이용하여 화면을 구성하는 최소한의 정보만을 포함하고 있다[2].

4.2 Fidelity 스케일링 기법

화질의 선명도에 따라 레이어 단위로 분할하는 Fidelity 스케일링 기법은 MPEG 비디오 스트림의 최하위 계층인 블록 층의 계수들을 이용하여 계층화하는 기법이다(그림 9). DCT를 수행하는 단위인 블록은 8×8의 크기를 가지며 이 블록이 하나의 DC와 63개의AC로 이루어지는 것에 착안하여 이를 5개의 계층으로 분할하였다[3, 12, 14].

4.2.1 인코딩 과정에서의 계층적 코딩

인코딩 과정을 이용한 계층적 코딩은 일반적인 MPEG 인코딩 과정과 유사하다. 인코딩시 DCT를 통한 블록은 인트라 블록과 비인트라 블록으로 나누어 지는데, 인트라 블록 내에는 영역내의 평균값인 하나의 DC계수와 영역내의 평균값으로부터 매우 정교한 최고 주파수의 영상 성분인 63개의 AC 성분이 배열되어 있다. 비인트라 블록 또한 64개의 AC성분이 분포되어 있다. 본 연구에서는 5개의 레이어로 나누었는데 F_layer1(Fidelity Layer 1)은 블록 층을 제외한 기존 데이터와 DC로 구성되어 있으며, F_layer2는 AC1과 AC2로 구성되어 있다. 또한 F_layer3는 AC3에서 AC5까지로 구성되어 있고, F_layer4는 AC6에서 AC9로 구성되어 있고, 마지막 F_layer5는 AC10에서 AC63으로 구성되어 있다(그림 9).

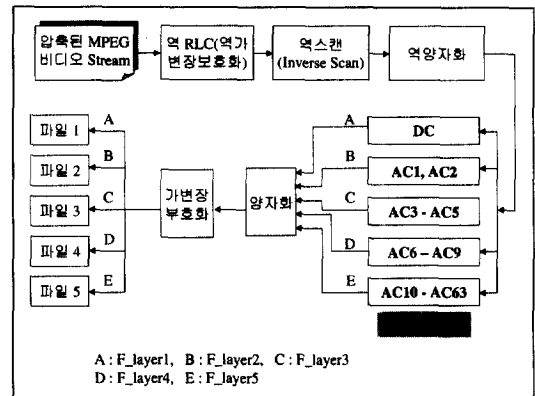


(그림 9) DCT 블록을 이용한 화질분할 개념

F_layer1에는 재생을 위해 필요한 최소한의 성분인 DC 성분이 포함되어 있으며 비디오 데이터의 경우 F_layer1만으로도 재생이 가능하다. 이 F_layer1을 기본으로 하여 주어진 QoS에 따라 나머지 레이어의 일부 또는 전부가 전송채널을 통해 MPEG 디코더가 있는 클라이언트로 전송되며, 전송된 레이어를 하나로 통합하기 위해 비디오 병합기에 의해 전처리 과정을 거쳐 하나의 블록으로 만든 후 재생시킨다.

4.2.2 디코딩 과정에서의 계층적 코딩

디코딩 과정을 이용하는 계층화는 기존의 인코딩 되어진 데이터를 이용하여 블록 층을 변환하기 때문에 블록 층을 제외한 나머지 부분(Sequence, GOP, Picture, Slice Header, Macroblock)은 원래 데이터를 그대로 사용하게 된다. 이는 기본 전송채널을 통해 데이터가 그대로 전송 되어짐을 의미한다. 블록을 디코딩하고 다시 양자화하는데 필요한 양자화 특성치와 양자화 매트릭스 등의 정보가 블록에 포함되어 있으므로 블록 변환기에 이 정보가 저장된다. 블록을 디코딩할 때 역양자화 후 인코딩시 DCT를 수행한 8×8 블록과 유사한 블록 F[v][u]를 얻게 된다. 인트라 블록내에는 하나의 DC 계수와 63개의 AC 성분이 배열되어 있으며, 이 블록을 인코딩 과정에서의 계층적 코딩과 같은 기법으로 5개 레이어로 분할한 후 레이어별로 양자화와 가변장부호화를 통해 파일로 출력되게 된다(그림10).



(그림 10) 화질분할기 구조

디코딩 과정을 이용한 변환은 기존의 인코딩된 데이터를 이용하여 블록 층을 변환하기 때문에 실시간에

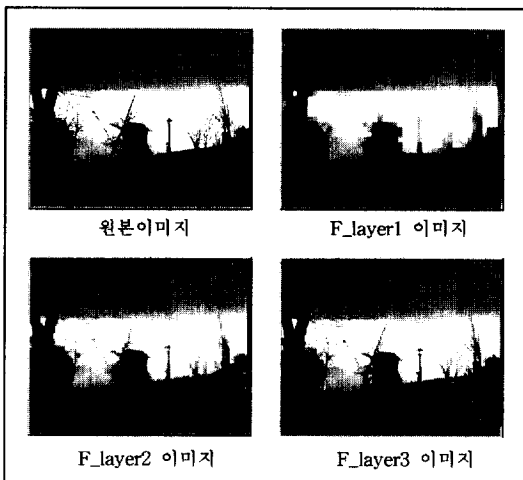
가깝게 계층적 코딩을 할 수 있고 전송도 가능하다. 레이어화 이전 원래 MPEG 파일을 MP라 하고 레이어화된 파일을 MP_f라 하면 이들은 다음과 같은 수식으로 나타낼 수 있다.

$$MP = \sum MP_f, \quad \text{단, } f = 1, 2, 3, 4, 5$$

즉, MP는 5 종류의 그룹으로 분리되고 각 그룹은 i ($0 < i \leq 64$) 개의 연속된 DCT 계수를 가지며, 그룹간에는 계수의 중복이 없다.

4.2.3 디코딩과 인코딩 과정에서의 장단점

블록 층 변환을 적용하는 방법에는 앞서 언급한 것처럼 2종류가 있는데, 하나는 인코딩시 DCT를 수행한 블록의 계수들을 이용하여 계층적 코딩을 하기 위해 블록을 변환하는 방법이고, 다른 하나는 압축된 MPEG-2 비디오 스트림 파일을 디코딩 과정을 이용해 변환하는 방법이다. 전자의 경우에는 인코딩을 위한 특별한 H/W가 필요로 하게 되지만, 후자의 경우는 기존의 MPEG-2 데이터를 이용할 수 있다는 장점과 실시간으로 계층화하여 서비스를 제공할 수 있다는 장점을 들 수 있다. 본 논문의 경우는 후자의 방법을 선택하여 설계 및 구현하였다. (그림 11)은 원래의 비디오 화질과 DCT블록을 레이어화 하였을 때 F_layer1, 3, 4에 해당하는 화질의 정도를 보여주고 있다.



(그림 11) 화질 레이어 별 선명도

영상의 범주에 따라 다소 차이는 있겠으나 (그림 11)에서 보는 바와 같이 F_layer3 이상의 영상을 이용

하면, 인터넷에서 비디오 서비스를 하는데에는 큰 문제는 없을 것으로 보여진다. 각 레이어별 파일 크기는 5장에서 기술한다.

5. 실험결과 및 분석

5.1 통합 이전의 TFS 실험

본 논문에서 제안한 Temporal 스케일링 및 화질 선명도에 따른 Fidelity 스케일링 모델의 타당성을 검증하기 위해 MSSG(MPEG Software Simulation Group) 비디오 코덱을 사용하여 구현한 후, 다양한 MPEG 파일에 대해 실험을 하였다[15-17]. 이 실험에서 N은 15, M은 3, 그리고 초당 프레임 율은 30으로 제한하였다. N은 GOP내의 픽처 개수를 의미하며, M은 I, P가 다음에 나올 주기를 의미하며, 프레임 율은 초당 보여주는 픽처 수를 의미한다. 프레임 율에 따른 Temporal 스케일링을 통해 6종류의 MPEG 비디오 스트림을 픽처단위로 분할하였다<표 1>.

<표 1> 픽처분할 결과

파일 \ QoS	T_layer1	T_layer2	T_layer3
ani.m1v (723KB)	284KB (39%)	485KB (67%)	726KB (100%)
icehaki.m1v (4898KB)	632KB (13%)	2668KB (54%)	4906KB (100%)
ad3.m2v (26666KB)	3093KB (12%)	12111KB (45%)	26683KB (100%)
station.m2v (1996KB)	473KB (24%)	1206KB (60%)	1998KB (100%)
flower.m2v (2752KB)	849KB (30%)	1933KB (70%)	2757KB (100%)
train.m2v (2750KB)	995KB (36%)	2023KB (74%)	2755KB (100%)
비 고	I 픽처 합	I, P 픽처 합	I, P, B 픽처 합

이는 MPEG의 기본 레이어만으로 구성된 원본파일과의 레이어별 크기 비율을 측정한 결과며, <표 1>에서 레이어별 수치는 이전 레이어를 누계한 결과값이다. 본 실험 결과에 의하면 T_layer1의 평균 백분율은 20.6%로 나왔으나, 제어정보와 같은 별도의 부가 정보를 네트워크로 보내주어야 함으로 실제로는 20~25% 정도를 차지한다. T_layer2는 32% 내외가 되고, T_layer3는 43% 내외인 것으로 나타났다.

화질의 선명도에 따른 Fidelity 스케일링 기법을 통

해 4종류의 MPEG 비디오 스트림을 DCT 블록의 DC 와 AC를 이용하여 분할하였다<표 2>.

<표 2> DCT 블록 분할 결과

QoS	파일		ad3 (26666K)	icehack (4989 KB)	flower (2752 KB)
	Layer	DCT			
F_layer 1	DC		5300 KB (20%)	1469 KB (30%)	742 KB (27%)
F_layer 2	DC, AC1 -AC2		7878 KB (30%)	1817 KB (37%)	873 KB (32%)
F_layer 3	DC, AC1 -AC5		10199 KB (38%)	2268 KB (46%)	1063 KB (38%)
F_layer 4	DC, AC1 -AC9		13540 KB (51%)	2782 KB (57%)	2420 KB (88%)
F_layer 5	DC, AC1 -AC63		26666 KB (100%)	4898 KB (100%)	2752 KB (100%)

이것 역시, 5개의 레이어로 분할한 결과값을 원본파일과 레이어별 크기 비율을 측정한 결과며, 여기서도 레이어별 수치는 이전 레이어를 누계한 결과값이다. 이 실험 결과에 의하면 F_layer1이나 F_layer2의 경우 화질은 많이 열화 되지만 파일 크기는 F_layer1은 25% 내외, F_layer2는 35% 내외로 감소됨을 알 수 있다.

지금까지의 결과를 종합하여 보면 T_layer1에서 T_layer3, 그리고 F_layer1에서 F_layer5까지의 모든 레이어 파일들의 크기가 다양하여, QoS에 변화에 적용하는 MPEG 비디오 스트림을 인터넷과 같은 실질적인 망 환경하에서 서비스 가능함을 알 수 있었다. 그러나, 이를 개별적으로 적용하기 보다는 하나로 통합하여 더 많은 레이어로 분할할 수 있다면, 좀더 다양한 QoS에 적합한 비디오 스트림을 서비스할 수 있을 것이다. 이에 본 논문에서는 이들을 통합한 분할기법을 설계 및 구현하였다.

5.2 통합 후의 TFS 실험

지금까지 실험결과에 의해, Temporal 및 Fidelity 스케일링 코딩기법에서 각각 원본 비디오의 약 40% 정도의 데이터만으로도 서비스가 가능함을 알 수 있었다. 그러나, 이러한 데이터량도 망의 상황이 급변하는 인터넷 환경에서 서비스를 하기에는 다소 많은 데이터가 포함되어 있을 수 있기 때문에, 두 기법을 통합하여 보다 세분화된 레이어를 이용하여 사용자 및 네트워크 환경의 QoS 변화에 민감하게 적용할 수 있는 분할 기법을 개발하였다.

통합된 TFS는 먼저 Temporal 스케일링 기법을 적

용하여 원본 MPEG 비디오 스트림을 3개의 레이어로 나눈다. 그런 다음, 나뉘어진 3개 레이어 각각에 Fidelity 스케일링 기법을 적용하여 5가지로 분할하여 총 15개의 레이어를 만들어 낸다. 이 15개 레이어 파일에 부여한 의미와 이들 파일이 어떠한 정보를 소유하고 있는지가 <표 3>에 있다.

<표 3> 15개 파일의 의미와 보유정보

파일	보유 내용	비고
T_1F_1	Sequence Header, GOP Header, I Picture Header, Slice Header, Macroblock, DC	i가 1이면 I 픽처, i가 2이면 P 픽처, i가 3이면 B 픽처에서의 DCT 블록 계수
T_2F_1	P Picture Header, Slice Header, Macroblock, DC	
T_3F_1	B Picture Header, Slice Header, Macroblock, DC	
T_iF_2	DCT Coefficient 1 ~ 2	
T_iF_3	DCT Coefficient 3 ~ 5	
T_iF_4	DCT Coefficient 6 ~ 9	
T_iF_5	DCT Coefficient 10 ~ 63	

T_iF_j 는 T_layer i와 F_layer j를 의미하는 것으로 예를 들어 T_3F_4 는 B픽처로 구성된 T_layer3와 DCT 계수 AC6에서 AC9까지로 구성된 F_layer4를 의미한다. 통합된 TFS 기법의 시스템 구조는 3장의 시스템 모델에서 제안한 (그림 5)와 같다. 비디오 스트림을 서비스하는 서버에서는 압축된 MPEG 파일을 시스템 디코더를 이용해 비디오 스트림과 오디오 스트림으로 분리한다. 분할기는 분리된 비디오 스트림을 15개의 레이어로 계층화함과 동시에, 계층화된 레이어 파일을 병합기에서 조합할 수 있도록 제어정보 파일을 부가적으로 생성한다. 클라이언트의 병합기에서는 서버에 의해서 보내진 레이어별 비디오 스트림과 제어정보 파일을 이용하여 MPEG비디오 스트림을 재구성하고, 시스템 인코더에서는 함께 전송된 오디오 스트림과 멀티플렉싱 (Multiplexing)하여 플레이어를 통해 재생되도록 한다.

통합된 분할기법을 이용하여 실험한 결과는 <표 4>와 같다.

<표 4>는 분할되어 있는 각 스케일러블 미디어 객체 그룹의 전송에 필요한 데이터 양을 분할 이전의 원본 MPEG-2 비디오 스트림의 데이터 양에 대한 백분율을 나타낸 것으로, T_iF_j 는 i와 j의 수치가 자신보다 작은 레이어들을 합산한 결과다. 즉 T_2F_3 는

$$T_2F_3 = \sum \sum T_iF_j, \quad \text{단, } i = 1, 2 \quad j = 1, 2, 3$$

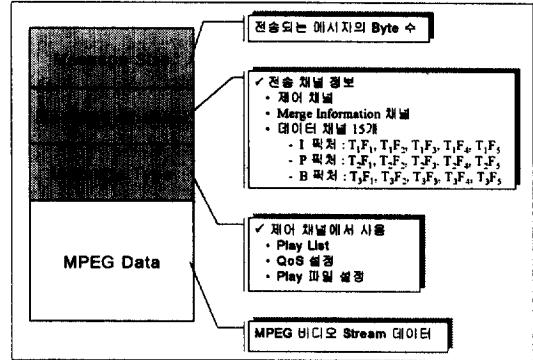
요구된 QoS에 대하여 스케일러블 미디어 객체 파일 T_1F_1 이 선택된 경우라면 원래 비트 스트림의 1% 만을 전송하면 되고, T_2F_4 객체 파일이 선택된 경우에는 원본에 비해 23%만 전송하면 된다. <표 4>을 통하여 스케일러블 미디어 객체 파일의 크기를 좀더 다양하게 조절할 수 있음을 알 수 있었으며, 결과적으로2가지 TFS 분할기법을 개별적으로 적용하기 보다는 통합하여 분할함으로써 동적으로 변화는 다양한 QoS 요구에 적절히 적용시킬 수 있었다. 여러 MPEG 파일을 실험해 본 결과, 실험대상 MPEG-2 파일에 따라 <표 4>의 결과는 달라진다는 것을 알 수 있었다[18]. 그러나, 모든 경우에 레이어별 파일크기가 다양하여 본 논문에서 제안하는 계층화 기법을 이용한 스케일러블 미디어 전송기법에 충분히 활용할 수 있었다. 경험적 실험결과에 의하면 T_1F_1 의 경우 원본 파일과의 크기 비율이 5~7%가 되어 통신망 트래픽을 상당히 감소시킬 수 있었으나 화질의 열화가 심하여 이미지 인식이 어려웠고, 원본 비디오에 준하는 비디오를 보기 위해서는 T_2F_3 이상이어야 가능함을 알 수 있었다. T_2F_3 의 경우 원본 비디오 크기의 20%정도 이므로 데이터 양을 상당히 줄일 수 있어, 대역폭이 낮은 환경에서도 실시간 재생이 가능함을 알 수 있었다.

<표 4> 통합된 TFS 적용결과

파일 이름 : ad3.m2v, 파일 크기 : 26,666 KB				
분할기법	T_layer1	T_layer2	T_layer3	보유내용
F_layer1	381 KB (1 %)	1,828 KB (7 %)	5,318 KB (20 %)	DC
F_layer2	875 KB (3 %)	3,367 KB (13 %)	7,893 KB (30 %)	DC, AC1 ~ AC2
F_layer3	1,134 KB (4 %)	4,489 KB (17 %)	10,213 KB (38 %)	DC, AC1 ~ AC5
F_layer4	1,479 KB (6 %)	6,034 KB (23 %)	13,556 KB (51 %)	DC, AC1 ~ AC9
F_layer5	3,093 KB (12 %)	12,111 KB (45 %)	26,683 KB (100 %)	DC, AC1 ~ AC63
보유내용	I 픽처	I, P 픽처	I, P, B 픽처	

5.3 계층적 코딩으로 인한 전송량 오버헤드

송신기와 수신기 사이를 오고 가는 비디오 Stream 은 전송오류를 최소화하고 정확한 정보를 전달하기 위해, 본 논문에서 제안하는 메시지 형태로 패킹(Packing)하여 전송하고 수신측에서는 역으로 해독(Unpacking)하는 과정을 거치게 된다(그림 12).



(그림 12) 메시지 구조

Layer화 후의 Layer별 파일 크기와 감소량, 그리고 메시지 패킹으로 인한 Layer별 추가되는 전송량 및 비율이 <표 5>에 나타나 있다.

<표 5> Layer화로 인한 전송량 오버헤드

ad3.m2v (26,666KB)				
Layer	Layer파일 크기(KB)	크기감소 (KB)	패킷크기 (KB)	패킷 비율(%)
PT ₁ PF ₁	381	26,302	2.3	0.5866 ~ 0.5906
PT ₁ PF ₂	875	25,808	5.2	
PT ₁ PF ₃	1,134	25,549	6.7	
PT ₁ PF ₄	1,479	25,204	8.7	
PT ₁ PF ₅	3,093	23,590	18.2	
PT ₂ PF ₁	1,828	24,855	10.7	
PT ₂ PF ₂	3,367	23,316	19.8	
PT ₂ PF ₃	4,489	22,194	26.4	
PT ₂ PF ₄	6,034	20,649	35.4	
PT ₂ PF ₅	12,111	14,572	71.1	
PT ₃ PF ₁	5,318	21,365	31.2	
PT ₃ PF ₂	7,893	18,790	46.3	
PT ₃ PF ₃	10,213	16,470	60.0	
PT ₃ PF ₄	13,556	13,127	79.6	
PT ₃ PF ₅	26,666	0	156.5	

<표 5>에서 Layer 파일크기는 재생을 위한 레이어를 의미하는 것으로, PT_iPF_j 는 i 이하의 T_layer와 j 이하의 F_layer의 합으로 PT_3PF_5 는 원본 파일크기와 동일하다.

크기 감소한 원본 비디오 스트림 크기에서 각 레이어 별 파일크기를 뺀 경우로, 통신망대역폭의 QoS를 절약하는 효과를 나타내며 다음과 같은 수식으로 표현할 수 있다.

$$\text{크기감소} = (\text{원본파일크기} - PT_iPF_j \text{ 크기}), \text{ 단 } (1 \leq i \leq 3, 1 \leq j \leq 5)$$

패킷크기란 전송시 메시지 패킹으로 인해 증가되는 데이터량으로, (그림 12)의 메시지 구조에 따라 한번 패킹할 때 마다 12바이트씩 증가하게 된다. 비디오 Stream은 전송을 위한 기본 프로토콜로 RTP를 사용하고, RTP의 전송단위가 2K이므로 2K 마다 12바이트씩 전송 오버헤드가 존재하게 된다. 그러나 대부분의 비디오 스트림의 경우 크기가 정확하게 2K로 나누어지지 않기 때문에 맨 마지막에 보내는 패킷에서는 크기가 2K가 아님에도 불구하고 메시지 패킹으로 인한 12바이트 전송 오버헤드는 감수해야 한다. 원본 비디오 Stream은 이와 같은 경우가 1번만 존재하지만 레이어화 된 비디오 스트림의 경우는 전송되는 레이어 개수만큼 존재하게 된다. 따라서 메시지 패킹으로 인한 전송량 오버헤드는 다음과 같은 수식으로 표현할 수 있다. 수식에서 PT_aPF_b 와 T_iF_j 는 바이트 단위로 표시되었음을 가정한 것이며, 수식의 결과값 역시 바이트 단위로 나온다.

$$\begin{aligned}
 & \text{전송량오버헤드}(PT_aPF_b) \\
 & = (((\sum \sum T_iF_j / 2048) \times 12) + (a \times b \times 12)) \\
 & \text{단, } (1 \leq a \leq 3, 1 \leq b \leq 5)
 \end{aligned}$$

마지막으로, 패킷비율은 전송될 파일에 대한 전송량 오버헤드인 패킷크기의 상대적인 크기 비율을 나타낸 것으로 패킷크기를 Layer파일크기로 나눈 것에 대한 백분율이다.

<표 5>의 실험결과에 의하여 레이어화로 인해 감소되는 데이터 양은 12%~98% 사이의 비율을 갖는데 비하여, 레이어 단위로 비디오 스트림을 전송하기 위한 전송량 오버헤드(패킷비율)는 전송될 파일크기에 비해 0.5866%~0.5984% 사이에서 증가되는 것으로 확인되었다. 이는 레이어화로 감소되는 데이터량 및 파일크기에 비해 미소하여 메시지 패킹으로 인한 전송량 오버헤드는 아주 작음을 알 수 있다. 따라서, 본 논문에서 제안하는 계층적 코딩기법의 전송 데이터량 감소 및 통신망 부하 감소효과에 대한 타당성을 검증해주고 있다.

5.4 QoS에 따른 레이어 결정

대역폭에 따른 QoS 파라미터가 주어졌을 때 이에 해당하는 레이어의 조합이 하나이상 나올 수 있으므로, 여러 개의 조합 중 적절한 조합을 선택해야 할 필요가 있다. 이에 선택된 조합 중에서 최상의 레이어를 사용

자 간섭 없이 자동으로 선택하는 알고리즘은 <표 6>과 같다.

<표 6>은 서버의 분배기에게 인증제어기로부터 수신한 QoS와 현재 서비스중인 QoS와 비교하여 서로 다른 경우, 클라이언트에게 어떠한 레이어의 비디오 스트림을 서비스할 것인가를 결정하기 위해 사용하는 알고리즘이다. QoS 파라미터로는 알고리즘의 단순화를 위해서 네트워크 대역폭과 사용자가 선호하는 분할기법(Temporal 또는 Fidelity)을 채택하였다. 알고리즘에서 입력으로는 네트워크 대역폭과 사용자선호 분할기법, 그리고 각 레이어들을 조합한 파일의 비트 율을 포함하는 배열이 들어오며, 결과값으로는 T_layer 레벨과 F_layer 레벨이 된다. 그리고 여러 개의 조합이 나올 수 있을 때, 대역폭에 가장 가까운 2개의 조합만을 가지고 선택알고리즘을 적용하면 된다.

<표 6> QoS에 따른 레이어 선택 알고리즘

```

Input: (network bandwidth), p (user preference), X (array)
Output: i (T_layer level), j (F_layer level)
begin
  if (p = temporal) then k := 3; l := 5;
  else k := 5; l := 3;
  count := 0;
  for t := k downto 1 do
    for s := l downto 1 do
      if (count >= 2) then break; // 후보자 선택
      if (X[t][s] ≠ t) then
        Add(t, s); count = count + 1;
      continue;
  if (Delete(&k, &l) = false) then service is not available
  if (Delete(&m, &n) = false) then i := k; j := l;
  if ((k-m) > 1) then i := k; j := l;
  else if (l-n) > 1) then i := m; j := n;
  else i := k; j := l;
end
  
```

<표 6>의 알고리즘에서는 우선적으로 사용자의 선호도를 반영해서 후보조합을 선택하지만, 만약 사용자가 선택한 선호도 조합과 나머지 조합의 레벨이 많은 차이가 있을 경우 다른 선호도를 선택하여, 주어진 대역폭에서 최상의 서비스를 할 수 있게 한다. 즉, 사용자가 프레임 율을 선호하면 자연스러운 화면을 요구한 경우로, 나머지 하나의 조합이 T_layer 레벨의 차가 T₃와 T₂ 같이 1만큼의 차이가 난다면 F_layer 레벨을

보고, T_layer 레벨의 차이가 2이상이었을 때에는 사용자가 선호한 기법을 우선으로 선택한다. 그렇지 않을 경우에는, 두 번째 조합을 분석하여 레이어를 선택하게 된다. 예를 들어,

선호도가 프레임 올인 상태에서 위에서 기술한 알고리즘을 적용한 결과 후보자 조합이 T_3F_2 와 T_2F_4 가 나왔을 때 먼저 사용자가 선호한 T_layer 레벨인 3과 2를 비교하여 1만큼의 차이가 나므로 F_layer 레벨을 살펴보게 되며, 이때 레벨의 차이가 2이상인 차이가 나므로 T_2F_4 를 선택하게 된다. 다른 예로 T_3F_3 , T_1F_5 가 있을 때 사용자가 선호한 T_layer의 차가 2 이상이기 때문에 F_layer를 고려하지 않고 바로 T_3F_3 를 선택해주게 된다. 이와 같이 하는 이유는 가능하면 사용자의 간섭을 최소화한 상태에서 최상의 품질로 서비스하기 위함이다. 사용자가 화질을 선호한다는 의미는, 화면의 자연스런 움직임 보다는 선명한 화질로 서비스 받고자 한다는 의미다.

이 알고리즘을 이용하면, 초기 연결 설정시의 QoS에 부합되는 서비스해야 할 레이어가 결정되고, 재생 중에도 QoS 변화에 따른 레이어 이동이 가능해진다 [19, 20].

8. 결 론

인터넷의 사용이 보편화 되고 ADSL(Asynchronous Digital Subscriber Line), Cable Modem등과 같은 초고속인터넷 서비스의 등장으로 기존의 텍스트를 기반으로 하는 서비스들이 점차 음성이나 영상과 같은 멀티미디어를 이용한 서비스로 변화되어 가고 있다. 초고속통신망 서비스 업체는 기존의 모뎀을 이용한 통신에 비교할 수 없을 정도로 고속의 인터넷을 이용할 수 있다는 광고를 하고 있고, 사용자들은 이런 초고속 통신 서비스로 몰리고 있다. 늘어나는 사용자에 반해 기술력의 한계로 인해 대부분의 사용자들은 기존의 통신 속도에 비해 많은 차이가 없는 서비스를 받고 있다. 사용자들은 보다 빠른 인터넷을 통해 비디오 서비스와 같은 멀티미디어 서비스를 원하고 있지만, 추가적인 H/W 장비를 설치하지 않고서는 어려운 실정이다. 현재의 시스템에 비용을 더 투자하지 않은 상태에서 이와 같은 문제를 해결하기 위한 해결점을 찾기 위해 시작한 것이 본 연구이다.

본 연구를 통해 MPEG 비디오 스트림을 통신 대역

폭에 따라 데이터 양을 조정하는 기술과 대역폭 변화를 비디오 스트림 재생 중 동적으로 반영하기 위한 기법을 연구하였다. 이를 위해 TFS 기법으로 Temporal 스케일링 기법과 Fidelity 스케일링 기법을 조합한 통합된 TFS 방법을 제안하였다. 그리고, 분할시 함께 생성한 메타정보를 이용하여 스케일러블하게 미디어를 전송하는 스트림 서버와 이를 수신하여 통합하는 비디오 병합기를 제안하였다. 동적인 대역폭 변화를 반영하기 위한 기법으로는 클라이언트 스트림으로부터 오는 피드백 제어정보와 인증제어 모듈, 그리고 분배기에서 사용하는 레이어 선택 알고리즘을 제안하였고, 지금까지 열거한 모든 기능을 통합하여 서버와 클라이언트가 유기적으로 연동하는 스케일러블 VOD 서비스 시스템 구조를 제안하였다.

또한, 본 논문에서 제안한 미디어 스케일링 기법을 스케일러블 VOD 서비스 시스템 구조에 적용하면, MPEG 비디오 스트림을 전체 크기는 변화되지 않는 상태에서 다양한 크기를 갖는 여러 개의 레이어 파일로 분할할 수 있고, 분할된 파일들은 재생 중에 QoS에 따라 다른 레이어로 전송된 후 병합되어 재생될 수 있음을 실험을 통해 확인할 수 있었다. 이로 인해, 기존의 연구결과가 같은 비디오 데이터를 QoS에 따라 여러 개 중복 저장함으로써 기억공간을 낭비하고, 사용자 간섭 없이 QoS를 결정하지 못하는 단점과, 재생 중에 동적으로 변하는 QoS를 서비스에 반영하지 못하는 문제점을 해결할 수 있었다.

따라서, 본 논문에서 제안하는 모델을 이질적인 환경을 갖는 망들이 서로 연동되는 인터넷에 적용하여 인터넷 VOD 및 인터넷 방송 등의 서비스에 응용한다면, 대역폭 변화를 동적으로 반영하며 실시간으로 비디오를 서비스할 수 있는 환경을 구축할 수 있고, 불필요한 통신 자원의 낭비를 막음으로써 통신 자원의 효율성을 향상시킬 수 있을 것으로 기대한다.

참 고 문 헌

- [1] Harrick Vin, "Heterogeneous Networking," IEEE Multimedia, pp.84-87, 1995.
- [2] 김태형, 유우종, 김형철, 공상한, 유관중, "스케일러블 전송을 위한 MPEG-2 비디오 Temporal Layered Coding에 대한 설계 및 구현", 한국정보과학회 추계 학술발표 논문집, 제25권 제2호, pp.462-464, 1998.

- [3] 손호신, 유우종, 김형철, 유관중, "동적인 QoS 변화에 적응시키기 위한 MPEG-2 Video Layered Coding", 한국정보처리학회 추계 학술발표 논문집, 제5권 제2호, pp.213-216, 1998.
- [4] International Standard ISO/IEC 13818-2, Information technology - Generic coding of moving pictures and associated audio information : Video, 1996.
- [5] 정제창, 그림으로 보는 최신 MPEG, 교보문고, 1995.
- [6] H. H. I Berin, "MPEG-1 and MPEG-2 Digital Video Coding Standards," http://www.wam.hhi.de/mpeg-video/paper/sikora/mpeg1_2/mpeg1_2.htm.
- [7] R. Arvind, R. Civanlar, and R. Reibman, "Packet Loss Resilience of MPEG-2 Scalable Video Coding Algorithms," IEEE Trans. circuit and systems for video tech. Vol.6. No.5, pp.426-435, 1996.
- [8] J. Y. Lee, T. H. Kim, and S. J. Ko, "Motion Prediction Based on Temporal Layering for Layered Video Coding," Proceeding of ITC-CSCC'98, Vol.1, July, 1998.
- [9] A. Puri, L. Yan and B. G. Haskell, "Syntax, semantics and description of Temporal Scalability," ISO/IEC JTC1/SC29/WG11 MPEG Doc. 93/795, Sept. 1993.
- [10] S. R. McCanne, "Scalable Compression and Transmission of Internet Multicast Video," Report No. UCB/CSD - 96 - 928, 1996.
- [11] 김형철, 강민규, 김두현, "QoS 적응형 스트리밍 서비스를 위한 MPEG 비디오의 미디어 스케일링 기법", 한국정보처리학회 추계 학술발표 논문집, 제6권 제2호, 1999.
- [12] 손호신, 유우종, 김형철, 유관중, "Scalable Transmission을 위한 Network QoS 결정기법과 SpatioTemporal LC 기법에 관한 연구", 한국정보처리학회 추계 학술발표논문집, 제6권 제2호, pp.DB195-DB198, 1999.
- [13] 김태영, 손호신, 유우종, 김형철, 유관중, "QoS를 고려한 SpatioTemporal Layered Coding과 Scalable Transmission에 관한 연구", 한국정보과학회 추계 학술발표 논문집, 제26권 제2호, pp.614-616, 1999.
- [14] W. J. Yoo, T. Y. Kim, H. S. Son, K. J. Yoo, H. S. Sin, H. C. Kim, "A study on the Layered Compression and Transmission of MPEG Video," International Conference on Information Technology and Computer Education (ICITCE'99), Chonju, Korea, pp.287-297, 1999.
- [15] MPEG Committee, MPEG-2 Test Model 0-5, ISO/IEC JTC1/SC29/WG11 N0400, April, 1993.
- [16] MPEG Committee, MPEG Software Simulation Group(MSSG) Video Codec, <http://www.mpeg.org/MPEG/MSSG>
- [17] 구준모, "MPEG-2 Source," http://www.infoage.co.kr/maso/MPEG_9706.zip
- [18] M. Krunz, R. Sass, and H. Hughes, "Statistical Characteristics and Multiplexing of MPEG Streams," Proceeding of INFOCOMM'95, pp.455-462, 1995.
- [19] W. J. Yoo, H. S. Son, T. G. Kang, S. I. Lee, D. H. Kim, K. J. Yoo, "The Design of QoS Filter Module to Transmit MPEG Video," Proceeding of PDCS 2000, IASTED, 2000.11.
- [20] W. J. Yoo, H. G. Lee, H. J. Kim, H. Y. Song, H. S. Shin, K. J. Yoo, "A Design of Scaler and Merger for Scalable MPEG Data Transmission on Heterogeneous Environments," Proceeding of ICADL 2000, 2000.12.



유우종

e-mail : wjyoo@tjhealth.ac.kr
 1987년 충남대학교 계산통계학과 (이학사)
 1990년 충남대학교 대학원 계산통계학과(이학석사)
 2001년 충남대학교 대학원 전산학과(이학박사)

1987년~1993년 한국전자통신연구원 선임연구원
 1993년~현재 대전보건대학 컴퓨터정보처리과 조교수
 관심분야 : 멀티미디어 응용, 실시간 미디어처리, VOD, 병렬처리



이흥기

e-mail : helius@nspplab.cnu.ac.kr
 2000년 충남대학교 컴퓨터학과 (이학사)
 2000년 충남대학교 컴퓨터 과학과 석사과정

관심분야 : 멀티 미디어 데이터의 전송 방법, 분산 시스템



이성인

e-mail : silee@tjhealth.ac.kr
 1987년 충남대학교 계산통계학과 (이학사)
 1989년 충남대학교 대학원 계산통계학과(이학석사)
 1996년 충남대학교 대학원 전산학과 박사과정수료

1989년~1991년 한국건설기술연구원 정보관리실 기술원
 1991년~현재 대전보건대학 멀티미디어사무용용과 조교수

관심분야 : 병렬처리, VOD, 멀티미디어 응용



이중인

e-mail : jilee@tjhealth.ac.kr
 1987년 충남대학교 계산통계학과 (이학사)
 1992년 충남대학교 대학원 계산통계학과(이학석사)
 1998년 충남대학교 대학원 전산학과 박사과정수료

1987년~1993년 충남대학교 전자계산소
 1993년~현재 대전보건대학 컴퓨터정보처리과 조교수
 관심분야 : 멀티미디어, 데이터베이스



유관종

e-mail : kjyoo@cs.cnu.ac.kr
 1976년 서울대학교 계산통계학과 졸업(이학사)
 1978년 서울대학교 대학원 전산학 전공(이학석사)
 1985년 서울대학교 대학원 전산학 전공(이학박사)

1987년 3월~8월 충남대학교 전산학과 학과장

1987년~1989년 충남대학교 전자계산소 소장

1990년 1월~12월 캘리포니아 대학(Irvine) 방문교수

1979년~현재 충남대학교 공과대학 정보통신공학부 교수

1995년~현재 한국정보과학회 이사

관심분야 : Agent, Scalable Coding, 멀티미디어 응용, VOD, 병렬처리, 컴파일러 설계