

# 이동 단말기를 위한 동영상 스트리밍 소프트웨어<sup>†</sup>

연세대학교 이종민\* · 차호정\*\*

서강대학교 이상민 · 낭종호\*\*

고려대학교 정진환\* · 최진영\*\*

## 1. 서 론

최근 네트워크 기술의 발달로 인해 무선 네트워크의 대역폭이 증가하고 있으며, 하드웨어 기술의 발전으로 인한 진보된 이동 단말기가 등장함에 따라 이동 단말기에서 기존의 음성 통신 뿐만 아니라 무선 네트워크를 이용하는 다양한 응용 프로그램들을 탑재하려는 시도가 이루어지고 있다[1]. 특히, 인터넷의 보급으로 인해 기존의 텍스트 기반의 서비스에서 벗어나 보다 향상된 동영상 서비스의 수요가 급증하고 있으며, 이동 단말기와 무선 네트워크를 이용하여 동영상 서비스를 제공하려는 시도가 많이 이루어지고 있다.

그러나, 기존 인터넷 방송과 같은 유선 네트워크 기반의 동영상 시스템을 이동 단말기를 위한 동영상 시스템으로 이식하여 사용하려는 시도는 유선 네트워크와는 다른 무선 네트워크의 특성과 서비스 수신자의 특성 차이로 인해 종종 실패로 끝난다. 무선 네트워크는 유선 네트워크와는 달리 링크 오류로 인한 패킷의 손실과 지연이 많다. 패킷 손실은 지터를 발생시키며 스트리밍 된 동영상 스트림을 재생할 경우 화면의 왜곡을 발생시킨다. 또한, 이동 단말기는 기존의 데스크탑 컴퓨터와는 달리 연산 능력, 메모리 크기, 색상 수, 화면 크기 등의 시스템 자원에 있어서 많은 제약을 가지고 있다. 따라서, 현재 유선 네트워크 기반의 동영상 스트리밍 시스템에서 많이 사용되는 H.26x나 MPEG 같은 동영상 압축 표준들은 이동 단말기에서는 시스템 자원 부족으로 인해 올바르게 재생할 수 없다.

모바일 환경에서의 효율적인 스트리밍 서비스를 위하여 3GPP (The Third Generation Partnership

Project)[2]에서는 프로토콜과 코덱을 포함한 포괄적인 표준화 작업을 진행하고 있다. 그러나, 이러한 표준화 방안은 복잡하고 많은 연산 처리를 필요로 하기 때문에 현재의 모바일 환경에 적용하기에는 부적절하다. 따라서, 업계에서는 자체적으로 개발한 프로토콜과 코덱으로 모바일 스트리밍 서비스를 수행하려는 시도가 이루어지고 있다.

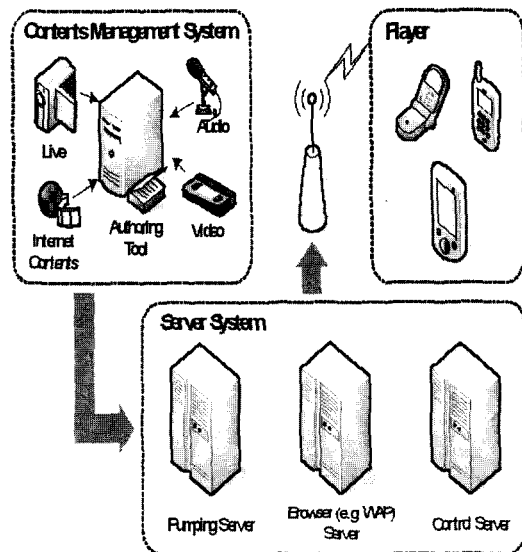


그림1 시스템 구조

모바일 동영상 스트리밍 시스템은 크게 스트리밍 서비스 제공을 위한 서버 시스템, 콘텐츠 관리를 위한 콘텐츠 관리 시스템 그리고 스트리밍 서비스를 받는 수신자의 세 가지로 구분할 수 있다. 그림 1은 모바일 동영상 스트리밍 시스템의 구조를 보여준다. 콘텐츠 관리 시스템은 기존에 작성되어 있던 동영상 콘텐츠나 실시간 방송을 수신자가 재생 가능한 콘텐츠의 형식으로 변환하고 변환된 콘텐츠를 관리하는 기능을 수행한다. 콘텐츠 관리 시스템의 중요한 요소로는 기존의 콘텐츠를 이동 단말기에서 재생 가능한 형식으로 변환하는 콘텐츠 변환 도구가 있다. 콘텐츠의 변환은 영상 크기, 색상 수, 화

<sup>†</sup> 본 글은 과학재단 특정기초연구사업(R01-2002-000-00141-0)에 의해 지원되었음.

\* 학생회원

\*\* 종신회원

질, 프레임율 등을 조정하는 작업이며 시스템 자원의 제한이 있는 이동 단말기의 특성을 반영해야 한다. 서버 시스템은 스트리밍 서비스를 제공하기 위한 시스템으로 사용자에게 콘텐츠 목록 제공 및 선택을 하도록 하기 위한 브라우징 서버, 사용자에게 콘텐츠를 스트리밍 및 다운로드하기 위한 스트리밍 서버 그리고 서버의 상태 감시 및 로그 분석 등을 처리하기 위한 제어 서버로 구성된다. 서버 시스템의 중요한 요소로는 스트리밍 서버이며, 효율적인 스트리밍 및 다운로드 서비스를 위해서는 높은 오류 발생률과 지연의 특성이 있는 무선 네트워크의 특성과 제한된 시스템 자원을 가지고 있는 이동 단말기의 특성을 반영한 적절한 전송 정책과 프로토콜이 필요하다. 수신자는 이동 단말기로 스트리밍 및 다운로드된 콘텐츠를 재생기를 통해 재생한다. 이동 단말기에서 재생되는 스트림의 서비스 품질을 높이기 위해서는 재생기의 경량화가 이루어져야 한다.

본고에서는 이동 단말기를 위한 동영상 스트리밍 시스템을 구축하고자 할 경우 발생하는 문제점과 시스템 설계시 고려해야 할 사항들에 대해 기술한다. 또한, 현재 상용화되어 서비스중인 모바일 동영상 스트리밍 시스템의 구현 예를 들어 모바일 동영상 스트리밍 시스템의 설계 및 구현에 대한 기준을 제시하고자 한다. 이동 단말기를 위한 동영상 재생기, 스트리밍 서버 그리고 콘텐츠 저작 도구에서 고려해야 할 사항들과 구현 예는 다음과 같다.

## 2. 이동 단말기를 위한 동영상 재생기

다음에는 표준 코덱으로 인코딩된 동영상을 이동 단말기에서 재생할 때 발생하는 문제점과 이를 해결하기 위한 LVF (Lightweight Video Format) 코덱에 관해 기술한다.

### 2.1 이동 단말기와 동영상 표준 코덱

초창기 저장 공간의 문제점을 해결하기 위한 MPEG-1 동영상 표준을 시작으로 하여 현재 MPEG-x, H.26x 등 다양한 동영상 표준들이 나름대로의 특징을 가지며 활용되고 있다. 일례로, MPEG-1, 2와 H.261의 코딩 효율성을 목표로 한 동영상 표준안에서, 최근의 MPEG-4 [3]나 H.263+[4]는 코딩의 효율성을 향상시키는 기법 뿐만 아니라 무선 통신과 같은 환경에서도 잘 적용할 수 있는 다양한 기법과 도구를 가지고 쓰이고 있다. 특히, MPEG-4의 경우 무선 환경을 고려한 것으로서 이것은 대표적 무선 장치인 이동 단말기에서 활용될 수 있는 가능성을 가지고 있음을 의미한다.

그러나 일반적으로 동영상 재생은 압축 알고리즘의

복잡성으로 인해 매우 많은 연산을 필요로 한다. 특히, 현재 널리 보급된 MPEG-x, H.26x와 같은 표준 동영상 형식들은 코딩의 효율성을 극대화하는 기술들로서 동영상 재생을 위해 상당 수준의 하드웨어 자원을 필요로 한다. 비록 이러한 동영상 표준안들이 나름대로의 계산량과 압출률 사이의 관계를 고려하여 여러 단계의 프로필 구성을 마련하고 있지만 그 기본 목표가 코딩의 효율성임은 틀림없다. 따라서 복잡한 코딩 알고리즘을 수행하여야만 하는 여러 동영상 표준안들이 무선 환경을 고려한 여러 기법과 도구를 가지고 있다고 하지만 제한된 연산 능력을 가진 휴대 전화에서 만족스러운 수준으로 동영상을 재생할지는 의문이다. 더욱이 MPEG-4에서 언급하는 무선 환경에 대한 고려는 코딩된 데이터 오류 검출, 복구 등에 관한 것이지 재생을 위한 이동 단말기의 성능에 관한 것은 아니기 때문이다.

표 1 휴대 전화 프로세서

칩셋	프로세서	속도*	캐쉬
MSM-3100	ARM7TDMI	19/19MHz	N/A
MSM-5000*	ARM7TDMI	13.5/27MHz	N/A

\*CDMA-2000 1x 칩셋, \*\*유휴 상태/통화 상태

표 1은 현재 널리 유통되고 있는 CDMA-2000 칼라 휴대 전화기의 칩셋과 그 칩셋에 포함되어 있는 프로세서 타입과 수행 속도이다. 코어 프로세서로 ARM7 계열의 저전력 32 bits 프로세서를 사용하며 캐쉬 없이 동작을 한다. 프로세서는 전력 소비 절감을 위해 유휴 상태와 통화 상태의 수행 속도가 다르다. 국내 휴대 전화의 경우 프로세서는 32 bits로 수행 가능한 것이지만, RISC 기반 프로세서의 단점인 실행 코드의 크기 문제 때문에 메모리 절감을 위해 Thumb mode라는 16 bits 형태로 저장된 실행 코드를 수행시킨다. 표 1에서 보듯이 프로세서의 기본 속도는 매우 낮으며 이 또한 통화에 필요한 성능에 맞추어져 있다. 보통 통화 상태에서는 전체 프로세서의 70% 가량을 통화 관련 태스크와 자체 기기 관리를 수행하는데 쓰여진다. 이러한 환경에서, 현재 무선 환경에 잘 맞게 되어 있다는 MPEG-4를 휴대 전화에 탑재하는 것이 적당한지를 알아보기 위해 먼저 MPEG-4 디코딩에 필요한 성능을 분석하였다.

표 2는 두 개의 MPEG-4 파일의 디코딩 속도를 측정 한 것이다. MPEG-4를 비롯한 일반적으로 널리 알려진 표준 동영상 코덱은 블록 기반의 압축 방식으로써 트랜스폼 코딩, 양자화, 엔트로피 코딩의 과정을 거친다. 가령 MPEG-x나 H.26x의 경우 DCT 변환, 양자화, 허프만 변환을 하게 된다. 이 때문에 MPEG-4의 디코딩 속도를 보다 자세히 알아보기 위해서 먼저 코딩 알고리즘을 세분

화하고 이를 각각 측정하였다. 즉, 표 2는 이 세 가지의 주요한 코딩에서 연산량을 알아보기 위해 분석한 것이다. 각 파일은 I, P 블록 별로 구분해 놓은 것이고 표 2에서 VLD는 허프만 디코딩, IDCT는 DCT 디코딩, Parsing은 움직임 보상 및 문법 분석, Dequant는 양자화 변환을 뜻한다. 컨텐츠는 QCIF 크기로서 파일 이름은 압축률을 뜻한다. 측정 환경은 Pentium III 750MHz 프로세서이며 단위는 us/block 이다.

표 2 MPEG-4 디코딩 성능 분석

동영상	51.m4v		38.m4v	
	Intra	Inter	Intra	Inter
VLD	2.11	1.15	2.25	1.39
IDCT	2.81	2.77	2.59	2.77
Parsing	2.70	1.01	2.71	1.79
Dequant	1.36	0.16	1.38	0.14
합계	8.98	5.09	8.93	6.09
압축률	12.08	52.21	11.65	38.67

표 2에서 보듯이 IDCT와 VLD가 매우 많은 연산을 필요로 함을 알 수 있다. 특히 IDCT는 전체 성능의 30%~50%를 차지할 만큼 매우 큰 영향을 미친다. 이것은 IDCT 알고리즘 자체가 매우 복잡하며 반복적인 연산이 많기 때문이다. IDCT가 아무리 효율적인 알고리즘 - AAN IDCT 알고리즘[5] 경우 블록당 144회 곱셈과 464회 덧셈 - 을 사용하더라도 기본적으로 곱셈과 덧셈의 수가 많기 때문에 연산에 많은 시간이 필요로 한다. VLD는 효율적인 비트 스트림을 생성해 주지만 디코딩 과정에서 빈번한 메모리 작업을 하기 때문에 비교적 많은 시간을 소모한다. VLD의 경우 비트 연산 위주이기 때문에, 캐쉬가 없고 메모리 성능이 낮은 휴대 전화에서는 보다 큰 부담이 될 가능성이 크다. Parsing은 Intra 블록 부분에서 많은 연산을 필요로 하지만 보통의 경우 Inter 블록이 매우 많기 때문에 평균적으로 볼 때 크게 성능을 저하시키는 요인은 아니다.

표 2의 실험 결과를 이용하여 보았을 때 평균적으로 하나의 프레임을 처리하는데 대략 4000us가 소모된다. 실제 휴대 전화에서는 어느 정도의 디코딩 시간이 소모되는지를 계산하기 위해 이를 표 1의 ARM7TDMI 프로세서에 디코딩 한다고 가정하고 단순 적용 - 실제 Pentium III는 클럭 당 ARM7 보다 많은 수의 코드를 실행한다. 따라서 ARM7이 실제로는 코드를 더 느리게 수행한다. - 시켜 보았을 때, QCIF 크기에서 대략 750ms(747963 us = 4039 us \* (750 MHz / 13.5 (유휴 상태의 MSM-5000) MHz) \* (100 % / 30 (여

유 프로세서 시간) %))가 필요함을 알 수 있다. 즉, 휴대 전화에서는 출력에 필요한 연산까지 고려할 때 MPEG-4로 저장된 동영상은 초당 하나의 화면도 재생할 수가 없다. 실험에 사용된 MPEG-4 코드는 상당 수준 최적화 된 것으로서 디코딩 시간은 보다 더 최적화를 하더라도 해결하기에는 매우 긴 시간이다.

위와 같은 문제점을 해결하기 위해 먼저 연구된 방법은 하드웨어적인 해결법이다. 동영상 재생을 위해 프로세서를 충분히 빠른 것으로 장착하거나 특정 미디어 프로세서를 휴대 전화에 추가로 장착하는 것이다. 특히 후자의 경우는 칩셋의 하드웨어적인 변경 없이 가능하기 때문에 휴대 전화 생산 업체에서 적용하기 쉬운 방안이다. 위의 두 가지 모두 빠른 프로세서를 이용하거나 추가적인 하드웨어를 이용하였기 때문에 비교적 좋은 품질로 MPEG-4 동영상을 처리할 수 있다.

하지만, 하드웨어적인 접근 방법은 빠른 프로세서와 추가적인 하드웨어로 인해 비용이 상승되며 휴대 전화의 가장 중요한 자원인 전력 문제도 야기된다. 보다 빠른 프로세서는 보다 많은 전력을 소모하지만 현대 충전지의 발전 속도는 매우 느리기 때문에 휴대 전화의 이동성에 악영향을 끼치게 된다. 또한 특정 동영상을 위해서는 특정 미디어 프로세서를 장착하기 때문에 다양한 동영상에 대한 유연성이 떨어진다. 이러한 접근 방식은 고가의 고성능 휴대 전화에서 일부 사용되는 방식이다.

## 2.2 LVF 코덱

기존 동영상 표준안이 휴대 전화에 그대로 쓰일 수 없다는 점과 추가적인 하드웨어로 인한 부담을 해결하기 위하여 하드웨어 변경이나 추가 없이 휴대 전화 성능을 고려한 동영상 코덱을 설계하고 소프트웨어만으로 동영상을 재생하는 방법이 있다. 소프트웨어 기반의 동영상 재생기로 국내에 상용화되어 쓰이고 있는 것으로는 LVF가 있다[6]. LVF는 동영상 재생에 가장 큰 병목 현상을 일으키는 IDCT와 VLD에 해당되는 코딩을 개선한 것으로 구체적인 개선 방법은 다음과 같다.

첫 번째로 앞서 언급된 것처럼 IDCT는 그 특성상 필요 계산이 정해져 있기 때문에 최적화에 한계가 있다. 때문에 LVF에서는 효율적인 압축률과 계산량을 조정할 수 있는 프랜스폼 코딩으로써 DCT 코딩 다음으로 많이 쓰이는 Wavelet 코딩[7]을 사용하였다. Wavelet 코딩은 어떤 기본 함수를 사용하느냐에 따라 필요 연산량과 압축률을 조절할 수 있다. 경우에 따라서는 DCT 보다 더 많은 연산을 하여 보다 높은 압축률과 화질을 얻을 수 있고 반대로 압축률은 다소 낮더라도 연산량을 크게 줄일 수도 있다. 연산량과 영상신호 처리 효율의 조절이

Wavelet 기저 함수의 변화로 용이하다는 점에서 저성능 휴대 전화에 적합하다. 특히 HAAR Wavelet의 경우 매우 간단한 덧셈 연산만으로 영상을 다루는 함수로 알려져 있다.

두 번째로 VLD의 성능을 높이기 위해 허프만 방식보다는 RLE를 개량한 방법을 사용하였다. RLE는 코딩 효율은 비록 허프만 보다 떨어지지만 디코딩 수행 속도가 매우 빠르다. RLE 사용에 앞서 RLE 압축률 개선을 위해 인코딩 과정에서 Normalization을 하였다. RLE는 최악의 경우(입력 데이터의 반복이 대부분 1인 경우) 오히려 데이터 크기가 커지는 단점이 있으나 이러한 것을 방지하기 위해 RLE 앞 과정에서 Normalize 단계를 거쳐 최악의 경우(데이터 반복이 전혀 없는 경우)가 발생하지 않게 하여 압축 효율을 높였다.

그림 2는 LVF를 H.263과 비교하여 정량적으로 보인 것이다. LVF의 경우 화질이 비록 H.263보다 약간 떨어지지만 처리 속도는 매우 뛰어나다.

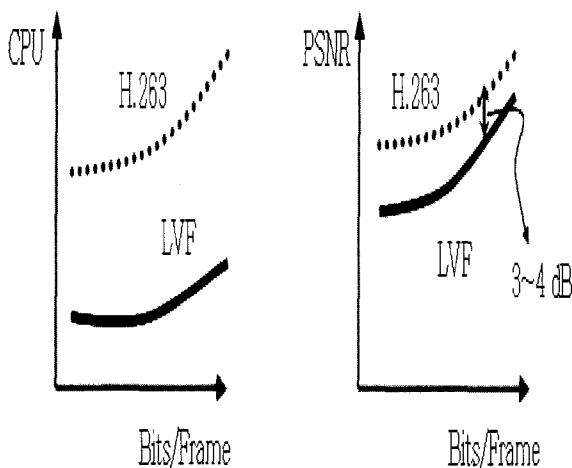


그림 2 LVF와 H.263 비교

LVF의 PSNR은 대략 3-4 dB 정도 낮지만 수행 속도는 H.263에 비해 무려 수십 배 가량 빠르다. 일반적으로 볼 때, PSNR 3-4 dB는 화질의 차이를 분명히 말할 수 있지만 휴대 전화에서는 그렇지 않다. 휴대 전화에서는 LCD가 제한된 색상을 가지며 저전력을 위해 픽셀의 특성이 보통의 LCD보다 성능이 떨어지기 때문에 화질이 나빠졌다고 단정짓기는 어렵다. 실제, LVF 코덱은 알고리즘적인 성능 향상뿐만 아니라 이러한 특성도 고려하였기 때문에 더욱 성능을 높일 수 있었다.

결과적으로 볼 때, 그림 2에서 보다시피 Wavelet 기반의 LVF는 약간의 화질을 희생하여 매우 큰 수행 성능을 보여 주었다. 이것은 휴대 전화에서 소프트웨어 방식

으로 동영상 재생이 가능함을 의미하는 것이다.

### 3. 무선 스트리밍 서버

다음은 무선 스트리밍 서버의 설계시 고려해야 할 사항을 살펴보고 무선 네트워크 및 이동 단말기의 특성을 반영한 프로토콜과 전송률 조정 정책에 대해 구현 예를 들어 기술한다.

#### 3.1 스트리밍 서버 구조

스트리밍 서버의 설계시 고려해야 할 사항으로는 확장성, 안전성 그리고 효율성이 있다. 각각의 항목에 대해 고려해야 할 사항은 다음과 같다.

**확장성 :** 스트리밍 서버가 설치되는 시스템의 확장으로 인한 스트리밍 서버 성능의 향상을 의미한다. 스트리밍 서버는 수백 혹은 수천의 동시 서비스 요구를 처리할 수 있어야 하며 많은 연산 처리를 필요로 한다. 이를 해결하기 위해 스트리밍 서버는 다중 프로세서 시스템에 설치되는 경우가 많으며, 시스템 활용도를 높이기 위해 스트리밍 서버는 멀티 쓰레드 구조로 설계되어야 한다. 멀티 쓰레드 구조로 스트리밍 서버를 작성할 경우 쓰레드에 어떠한 작업을 할당할 것인지를 결정하는 쓰레드 모델을 결정해야 한다.

쓰레드 모델은 일반적으로 구현이 쉬운 세션당 쓰레드를 하나씩 할당하는 방법을 가장 많이 사용한다. 그러나, 세션당 쓰레드를 할당하는 방법은 세션 개수가 늘어남에 따라 쓰레드의 개수가 늘어나며 많은 쓰레드의 사용으로 인해 쓰레드간의 컨텍스트 교환에 소비되는 부하의 증가로 오히려 시스템의 성능을 떨어뜨릴 수 있다. 이러한 문제점으로 세션당 쓰레드 모델은 중소 규모의 스트리밍 서버에서 사용되며 대규모 스트리밍 서버에서는 사용하지 않는다.

세션당 쓰레드 모델의 단점을 해결하기 위해 세션 처리 과정을 특성에 따른 몇 가지의 단계로 나누고 각각의 단계에 쓰레드를 할당하여 쓰레드에서는 특화된 작업을 분담하게 하는 방법이 있다(7). 이 방법은 세션당 쓰레드 모델보다 상대적으로 적은 수의 쓰레드 개수를 유지하며 다중 프로세서 시스템의 효율성을 증가시킬 수 있다. 쓰레드의 개수가 적으면 시스템의 효율성이 떨어지며 쓰레드의 개수가 많으면 쓰레드간의 컨텍스트 교환에 따른 부하 증가로 효율성이 떨어진다. 따라서, 작업 특성에 따라 쓰레드를 할당하는 모델은 다중 프로세서 시스템의 효율성을 높일 수는 있으나 시스템의 확장으로 인한 스트리밍 서버 성능의 확장을 충분히 기대할 수 없다.

구현된 동영상 서버에서는 적당한 개수의 쓰레드 개수를 유지하기 위하여 쓰레드 풀 모델을 사용한다.

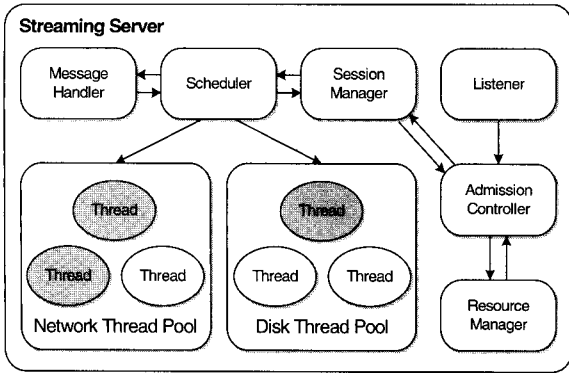


그림 3 스트리밍 서버 구조

그림 3은 스트리밍 서버의 구조를 보여준다. 쓰레드 풀 모델은 특화된 작업을 수행하는 쓰레드들의 집합을 생성하고 작업 부하의 증감에 따라 풀에 속해 있는 쓰레드의 개수를 적절히 조절하는 방법이다. 쓰레드 풀 모델의 사용으로 다중 프로세서 시스템에 적합한 쓰레드의 개수를 유지할 수 있으며 시스템의 확장시에도 스트리밍 서버의 성능 향상을 기대할 수 있다.

**안전성 :** 스트리밍 서버에서 제공해야 할 안전성은 서비스 되는 스트림에 대한 사용자 QoS 보장과, 예외 상황에 대한 서버 내부의 복구 능력이 있다. 일반적으로 스트리밍 서버에서는 서비스 되는 콘텐츠의 대역폭을 인식하고 콘텐츠 대역폭으로 자료를 전송하는 서버-푸시 모델을 사용한다. 서버-푸시 모델의 경우 전송되는 각 패킷은 전송 되어야 할 마감 시간을 가지고 있으며, 이러한 패킷의 전송 마감 시간을 만족시키기 위하여 스트리밍 서버에서는 효율적인 스케줄러가 필요하다. 스케줄러는 크게는 세션의 처리를 담당하는 쓰레드들의 실행 순서 및 동기화를 조절하는 기능을 수행하고 작게는 각 쓰레드 내부에서 처리해야 할 작업의 수행 순서를 조절한다. 상용화된 스트리밍 서버에서는 스케줄러가 일정한 주기로 실행되어 스케줄링 작업을 수행하는 주기 모델을 사용한다. 구현된 스케줄러는 주기적으로 실행되어 해당 주기 내에 수행되어야 할 작업의 종류 및 양을 결정한다. 작업 순서의 결정은 각 작업의 마감시간에 따른 위급도를 계산하여 결정된 위급도에 따라 결정한다. 스케줄러는 사용자 QoS 보장을 위해 중요한 역할을 수행하므로 그 정확성을 높이기 위해 우선 순위를 가장 높게 하여 실행하며 시스템에서 제공하는 정밀한 시간 함수를 사용한다.

무선 동영상 스트리밍 서비스에서는 수신자 환경의 다양성과 무선 네트워크 환경의 급격한 변화로 예상치 못한 상황이 많이 발생한다. 스트리밍 서버는 이러한 예

외 상황에 대해 적절히 대처하여야 하며, 하나의 세션에서 발생한 문제에 대해 전체 다른 세션에 영향을 주지 않도록 하여 서비스 중인 다른 스트림들에 대한 사용자 QoS를 보장해 줄 수 있어야 한다. 이를 위해 구현된 스트리밍 서버에서는 별도의 세션 관리자가 존재하여 각 세션의 상태를 감시하고 세션에 대한 상태 전이 과정을 유지하여 잘못된 상태 전이를 하는 세션을 발견할 경우 해당 세션에 대한 서비스를 중단하도록 한다.

**효율성 :** 스트리밍 서버는 서비스 요구시 디스크 대역폭, 네트워크 대역폭, 메모리 등의 시스템 자원을 해당 서비스를 위해 할당한다. 스트리밍 서버에서는 서비스 중인 스트림의 사용자 QoS 보장이 중요하지만, 효율적인 시스템 자원관리를 통해 많은 수의 서비스 요구를 수용하는 것 또한 중요하다. 이를 위해 적절한 자원 관리 방법과 효율적인 사용자 수용 제어 방법이 필요하다. 스트리밍 서비스에 사용되는 동영상 콘텐츠는 각각의 대역폭을 가지며, 다양한 콘텐츠의 종류로 인해 콘텐츠가 요구하는 대역폭도 다양하게 된다. 또한, 콘텐츠 요구 대역폭은 대부분 가변 대역폭으로 시간에 따라 요구되는 대역폭이 변하게 된다. 이러한 다양한 서비스 요구에 대해 고정된 일정 자원을 서비스가 요구하는 최대 혹은 평균 값으로 할당하게 되면 시스템 자원의 낭비 혹은 시스템 자원 초과로 인한 사용자 QoS 보장을 못하게 되는 경우가 발생한다.

구현된 스트리밍 서버에서는 많은 수의 사용자에 대한 사용자 QoS 지원과 효율적인 자원 관리를 위해 동적 자원 관리 정책을 사용한다. 이를 위해 스트리밍 서버 내부의 자원 관리자는 프로세서 사용량, 디스크 대역폭, 네트워크 대역폭, 메모리 크기 등의 시스템 자원을 실시간 모니터링하고 갱신하여 항상 정확하고 최신의 정보를 유지하도록 한다. 자원 관리자에 의해 유지되는 시스템 자원은 사용자 수용 제어기에서 사용자 수용 제어의 중요한 기준으로 사용된다. 사용자 제어기는 시스템의 사용 가능한 자원량과 요청된 서비스의 콘텐츠 대역폭에 근거한 예상 자원 소모량에 근거하여 요청된 스트림의 수용 제어 여부를 결정한다. 이는 시스템 자원이 허용하는 범위 내에서만 서비스를 허용함으로써 서비스 중인 세션의 사용자 QoS를 보장하기 위해서이다.

### 3.2 전송률 조정 정책

스트리밍 서비스의 경우 서비스 수신자는 네트워크를 통해 패킷을 전송 받으면서 동영상 스트림을 디코딩 해야 한다. 그러나, 이동 단말기는 연산 처리 능력과 메모

리 크기 등의 시스템 자원에 있어서 제약을 가지고 있기 모든 작업을 동시에 처리하기에 많은 어려움이 있다. 따라서, 이동 단말기에서 사용자에게 보다 좋은 동영상 품질을 보장하기 위해서는 네트워크 패킷의 처리 시간에 드는 비용을 최소화 해야 한다. 연산 처리 비용을 줄인 최적화된 프로토콜의 사용과 스트리밍 서버에서의 효율적인 전송 정책으로 이동 단말기에서의 네트워크 패킷을 처리하는데 소모되는 비용을 줄일 수가 있다.

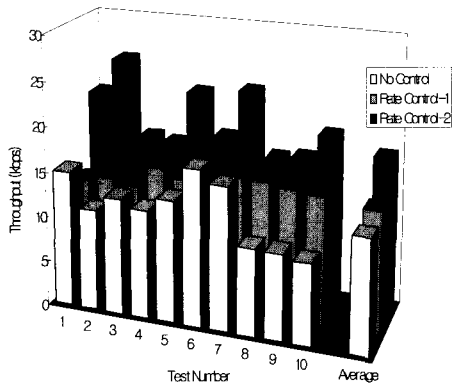


그림 4 전송 성능 비교

스트리밍 서비스는 실시간성이 중요시되기 때문에 UDP에 기반한 프로토콜을 사용한다. 스트리밍 서버는 스케줄러를 통해 네트워크로 전달되는 패킷의 마감 시간을 만족하도록 패킷을 전송한다. 미디어 스트림 전송의 시간 만족도를 충족시키기 위해 원본 콘텐츠의 전송률로 자료를 전송하면 전송률이 가변하게 되며 순간 자료 전송량이 급증하여 패킷의 손실을 유발할 수 있다. 또한, 전송률이 가변적일 경우 무선 채널의 할당과 해제에 소비되는 시간이 발생할 수 있으며, 이는 패킷 전달 시간을 증가시키고 지터를 발생시킨다. 지터는 재생 화면의 깨짐을 발생시켜 사용자 QoS 수준을 만족시키지 못하게 된다. 따라서, 이동 단말기의 수신 버퍼와 네트워크 상태를 고려한 효율적인 전송률 정책이 필요하며 스트리밍 서버에서 전송되는 자료의 전송률을 일정하게 유지하는 것이 필요하다.

다운로드 서비스일 경우 안정적인 자료의 전송을 위해 TCP에 기반한 전송 프로토콜을 사용한다. 최대의 전송 성능을 얻기 위해 스트리밍 서버는 네트워크가 허용하는 최대 대역폭으로 자료를 전송해야 한다. 그러나, 무선 환경의 경우 무선 상태에서의 전송 오류로 인한 패킷의 손실이 자주 발생하며, TCP에서는 이를 네트워크의 혼잡 상황이라 인식하고 혼잡 제어 정책에 따라 전송률을 줄이게 된다. 무선 네트워크에서의 링크 오류에 의

한 잦은 패킷의 손실은 TCP에서의 전송률 감소를 가져오며 이로 인해 충분한 전송 성능을 얻을 수 없다. 이를 해결하기 위해 무선 환경에 적합한 WTCP[9] 이나 I-TCP[10]와 같은 프로토콜을 사용하는 방법이 있으나, 대부분 전체 네트워크의 프로토콜 스택의 변경을 요구하여 실생활에 적용하기 어려운 단점이 있다.

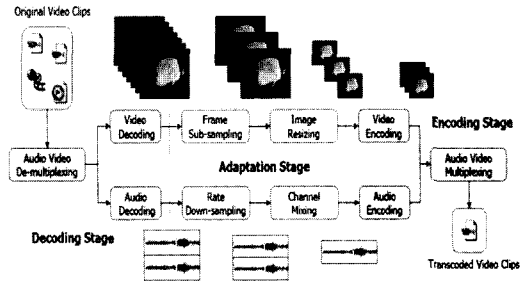


그림 5 트랜스코딩 과정

구현된 스트리밍 서버에서는 스트리밍 서비스와 다운로드 서비스에 네트워크 패킷 처리 연산 비용을 고려하여 설계된 프로토콜을 사용한다. 스트리밍 서비스에서는 다중화된 스트림의 분리, 적절한 패킷 크기의 결정을 통해 이동 단말기에서의 프로토콜 처리에 드는 연산 비용을 최소화 하도록 한다. 또한, 스트리밍 전송률을 네트워크의 상태와 이동 단말기의 수신 버퍼 크기를 고려하여 적절히 조정함으로써 무선 네트워크에서의 패킷 손실과 지연을 줄이도록 한다. 다운로드 서비스에서는 위한 응용 계층에서의 새로운 전송률 조정 방법을 사용한다. 이 방법은 TCP에서 사용되는 혼잡 윈도우 조절 방법과 유사하며, 응용 프로그램에서 네트워크 상태 변화에 따라 자료 전송률을 조절함으로써 수행된다. 그림 4는 전송률 조정 정책에 따른 TCP 전송 성능의 향상을 보기 위하여 전송률을 조정하지 않은 방법(No Control)과 응용 계층에서의 서로 다른 전송률 조정 정책(Rate Control-1, Rate Control-2)의 전송 성능을 CDMA-2000 네트워크에서 비교한 것이다. 10회의 실험을 통해 전송률 조정 방법이 전송률을 조정하지 않았을 경우보다 약 40~50%의 성능 향상이 있음을 볼 수 있으며, 적절한 전송률 조정 정책으로 전송 성능을 향상시킬 수 있음을 알 수 있다.

#### 4. 콘텐츠 저작 시스템

다음은 기본적인 동영상 변환 기법과 구현된 동영상 저작 도구에 대해서 설명한다.

##### 4.1 동영상 변환 기법

이동 단말기에게 동영상 서비스를 하기 위해서는 다

양한 형식의 동영상 데이터를 이동 단말기에 맞는 형태로 바꾸어 주기 위해 그림 5와 같은 동영상 변환(Transcoding) 과정을 거쳐야 한다. 동영상 변환은 크게 복호화 과정(decoding stage), 적응변환 과정(adaptation stage), 부호화 과정(encoding stage)으로 나뉘어진다. 각각의 단계에 대한 자세한 과정은 다음과 같다.

**복호화** : 압축된 동영상 데이터를 동영상과 오디오 데이터로 나눈 다음, 동영상 데이터는 압축되지 않은 낱장의 Bitmap 형식의 그림들로 변환하고 오디오 데이터는 압축되지 않은 PCM 형식으로 변환한다.

**적응변환** : 동영상 데이터를 압축되지 않은 형태로 변환한 다음에는, 이동 단말기에 맞는 형식으로 바뀌어야 한다. 이러한 적응변환 과정에는 동영상 데이터의 프레임 율을 낮추기 위한 작업(Frame Sub-sampling)과 그림의 해상도를 줄이는 작업(Image Resizing)이 포함된다. 단말기의 화면에 맞도록 색상 수를 줄이기 위한 디터링도 필요한데 이는 보통 이동 단말기의 복호기에서 수행된다. CDMA-2000 무선망에서 112x96 픽셀 크기를 갖는 이동 단말기의 경우, 해상도는 112x96의 크기로 줄이면 되지만 프레임 율은 무선망의 대역폭과 단말기의 성능 중에 더 낮은 쪽에 맞추어 낮추어야 한다. LVF 인코딩 기술을 사용하면 동영상 데이터에 움직임의 양이 많은 경우에도 CDMA-2000 무선망의 대역폭에서 초당 10프레임 이상의 동영상을 제공할 수 있지만 단말기의 성능제약으로 인해 실제로는 초당 10프레임 이하로 낮추어야만 한다[6]. 오디오 데이터는 단말기의 오디오 재생 능력에 맞추어 샘플링 빈도를 낮추거나(Rate Down-sampling) 스테레오를 모노로 변경하는 작업(Channel Mixing)을 수행한다.

부호화 : 적응 변환된 동영상과 오디오 데이터는 이동 단말기가 지원하는 동영상 부호기에 의해 다시 동영상 데이터로 압축된다. 이 과정에서 데이터 양을 무선망의 대역폭에 맞추기 위해 압축률을 조정하게 되며 이에 따라 화질과 음질에 열화가 발생하게 된다. 이렇게 부호화된 두개의 데이터를 적절한 타임스탬프와 함께 다중화(Multiplexing)함으로써 동영상 변환이 완료되게 된다. 이러한 과정을 거쳐 완성된 동영상 데이터는 파일로 저장해서 보관하거나 스트리밍 서버를 통해 무선 단말기에게 전송하게 된다.

동영상 변환 과정에서 가장 많은 시간이 소모되는 부분은 동영상 데이터의 복호화와 부호화 부분이다. 따라서 변환 시간을 줄이기 위해 복호화와 부호화를 적게 하면서 적응변환을 할 수 있도록 복호화 방법을 제안하는 연구[11]가 계속되고 있다. 하지만 현재 실제로 쓰이고 있는 많은 동영상 데이터들은 AVI, MPEG, ASF, WMV와 같은 다양한 형식을 사용하므로, 각각의 압축방식에 맞는 최적화된 변환기를 모두 개발하는 것은 매우 힘든 일이다. 마이크로 소프트웨어사의 윈도우즈 환경에서는 DirectShow 기술을 이용하면 동적으로 동영상 데이터의 특정 압축형식에 맞는 복호기를 찾아내어 압축되지 않은 형태의 데이터를 생성해준다. 따라서 이러한 방식을 이용하면 윈도우즈 환경에서 재생 가능한 동영상 데이터라면 어떤 동영상이라도 쉽게 다른 형태로의 변환이 가능하게 된다.

동영상 변환 과정에서 가장 많은 시간이 소모되는 부분은 동영상 데이터의 복호화와 부호화 부분이다. 따라서 변환 시간을 줄이기 위해 복호화와 부호화를 적게 하면서 적응변환을 할 수 있도록 복호화 방법을 제안하는 연구[11]가 계속되고 있다. 하지만 현재 실제로 쓰이고 있는 많은 동영상 데이터들은 AVI, MPEG, ASF, WMV와 같은 다양한 형식을 사용하므로, 각각의 압축방식에 맞는 최적화된 변환기를 모두 개발하는 것은 매우 힘든 일이다. 마이크로 소프트웨어사의 윈도우즈 환경에서는 DirectShow 기술을 이용하면 동적으로 동영상 데이터의 특정 압축형식에 맞는 복호기를 찾아내어 압축되지 않은 형태의 데이터를 생성해준다. 따라서 이러한 방식을 이용하면 윈도우즈 환경에서 재생 가능한 동영상 데이터라면 어떤 동영상이라도 쉽게 다른 형태로의 변환이 가능하게 된다.

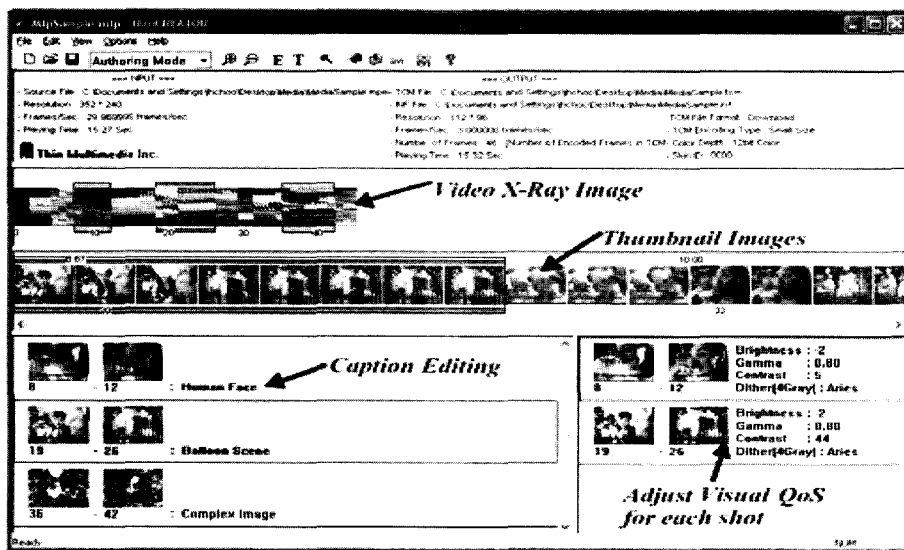


그림 6 동영상 제작 도구

## 4.2 동영상 저작 도구

무선 환경의 동영상 서비스에서 여러 가지 이동 단말기의 요구사항을 만족시키기 위해서는, 각각의 요구사항에 맞는 다양한 동영상 데이터들을 각각 동영상 변환 과정을 거쳐 생성해주어야 한다. 그림 6은 구현된 동영상 저작 도구의 그래픽 사용자 환경의 화면이다. 동영상 저작 도구는 이러한 다양한 형태의 동영상 데이터를 생성하기 위해 해상도, 프레임 율, 색상 비트 수, 동영상 압축률, 오디오 소리 크기 값 등을 조절할 수 있는 다양한 기능을 제공해야한다. 또한 이동 단말기의 LCD 화면은 대개 PC가 갖고 있는 CRT나 TFT LCD 화면에 비해 어둡기 때문에 동영상 데이터의 화질을 높이기 위해서 밝기나 선명도를 조절하거나 히스토그램 평활화 같은 이미지 프로세싱 기법을 사용하기도 한다.

만약 이러한 다양한 형태의 동영상들을 콘텐츠가 등록될 때마다 미리 만들어놓는다면, 콘텐츠의 수가 증가하면 할수록 서버의 저장 공간이 수백 개의 파일들로 인해 금방 가득 차버리는 상황이 발생할 수도 있다. 이러한 환경에서는 각각의 동영상 데이터들을 미리 생성해두는 대신 저작 도구에서 적응변환에 필요한 각각의 설정 값들만 보관해 두었다가, 무선 단말기로부터 요청이 있을 때에 요청된 형태의 동영상 데이터를 실시간으로 변환하여 보내주는 방법을 사용할 수 있다.

## 5. 결론

본고에서는 이동 단말기를 위한 동영상 스트리밍 시스템을 구축할 경우 고려해야 할 사항과 실제 국내에 상용화되어 사용되고 있는 시스템의 예를 기술하였다. 이동 단말기를 위한 동영상 시스템은 이동 단말기, 스트리밍 서버, 콘텐츠 저작 시스템 등의 시스템 전반에 걸쳐 제한된 시스템 자원을 가지고 있는 이동 단말기의 특성과 오류 발생 확률이 많고 지연 시간이 긴 무선 네트워크의 특성을 고려하여야 한다. 이동 단말기에서 원활한 동영상 재생기를 위해서는 경량화된 코덱의 사용이 필요하며 통신에 사용되는 비용이 최소화하도록 하는 것이 필요하다. 스트리밍 서버의 경우 확장성, 안전성, 효율성이 중요시되며 사용자 QoS 보장을 위한 적절한 스케줄링 정책과 무선 네트워크의 환경과 이동 단말기를 고려한 효율적인 전송 정책이 필요하다. 콘텐츠 저작 시스템은 이동 단말기의 사용자 환경을 고려한 적절하고 효율적인 트랜스 코딩과 콘텐츠 관리 방법이 필요하다.

본고에서 구현 예로 든 동영상 스트리밍 시스템은 상용화되어 쓰이고 있는 서비스중인 시스템이다. 향후 연구로는 LVF의 화질 개선 및 다양한 플랫폼으로의 이식, 스트리밍 서버의 정확한 무선 상태 감지에 따른 효율적

인 전송 정책 연구, 콘텐츠 변환 시스템에서의 이질 미디어간 콘텐츠 변환 기술 연구 등이 있다.

## 참고문헌

- [1] L. Garber, 'Will 3G Really Be the Next Big Wireless Technology?', IEEE Computer, Vol. 35, No. 1, pp. 26-32, January 2002.
- [2] 3GPP, <http://www.3gpp.org>
- [3] R. Talluri, 'Error-resilient video coding in the ISO MPEG-4 standard,' IEEE Comm. Magazine, Vol. 36, pp. 112-119, June 1998.
- [4] ITU-T, 'Video Coding for Low Bit Rate Communication,' ITU-T Recommendation H.263.
- [5] Arai, Y., T. Agui, and M. Nakajima, 'A Fast DCT-SQ Scheme for Images,' IEEE Transactions of the IEICE, Vol. E71, No.11, pp. 1095-1097, November 1988.
- [6] Jin-Hwan Jeong and Chuck Yoo, 'A Video Streaming System for Mobile Phones: Practice and Experience,' Proceedings of International Conference on Wireless Networks and Emerging Technologies, 2003.
- [7] Amara Graps. 'An Introduction to Wavelets,' IEEE Computational Science and Engineering, Vol. 2, No. 2, pp. 50-61, Summer 1995.
- [8] H. Cha, J. Lee, J. Oh, 'Constructing a Video Server with Tertiary Storage: Practice and Experience,' Multimedia Systems, ACM/Springer Verlag, Vol.8, No.5, pp. 380-394, December 2002.
- [9] P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan, 'WTCP: A reliable transport protocol for wireless widearea networks,' Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, WA, pp. 231-241, August 1999.
- [10] Ajay V. Bakre and B. R. Badrinath, 'I-TCP: Indirect TCP for mobile hosts,' Proceedings of 15th International Conference on Distributed Computing Systems, Vancouver, Canada, pp. 136-143, May 1995.



[11] H. Radha, M. van der Schaar, Y. Chen,  
'The MPEG-4 Fine-Grained Scalable  
Video Coding Method for Multimedia

Streaming over IP,' IEEE Transaction  
on Multimedia, Vol. 3, No. 1, pp. 53-68,  
March 2001.

**이 종 민**



1999 광운대학교 컴퓨터과학 학사  
2001 광운대학교 컴퓨터과학 석사  
2003~현재 연세대학교 컴퓨터과학과  
박사과정  
관심분야: 멀티미디어 시스템, 운영체제,  
컴퓨터 네트워크  
E-mail : jmlee@cs.yonsei.ac.kr

**남 종 호**



1986 서강대학교 전자계산학과 학사  
1988 한국과학기술원 석사  
1992 한국과학기술원 박사  
1992~1993 Fujitsu 연구소 연구원  
1993~현재 서강대학교 컴퓨터학과 교수  
관심분야: 비디오 인덱싱, 디지털 비디  
오 라이브러리  
E-mail : jhnang@ccs.sogang.ac.kr

**차 호 정**



1985 서울대학교 컴퓨터공학 학사  
1987 서울대학교 컴퓨터공학 석사  
1991 University of Manchester 전산학  
박사  
1993~2001 광운대학교 컴퓨터 과학과  
부교수  
2001~현재 연세대학교 컴퓨터과학과  
교수  
관심분야: 멀티미디어 시스템, 운영체제,  
내장형 시스템  
E-mail : hjcha@yonsei.ac.kr

**정 진 환**



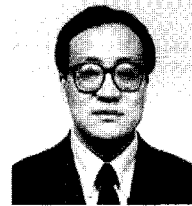
1997 고려대학교 컴퓨터과학 학사  
1999 고려대학교 컴퓨터과학 석사  
2003~현재 고려대 컴퓨터학과 박사과정  
관심분야: 소형 기기를 위한 비디오 처  
리, 멀티미디어 스트리밍, 운영체제  
E-mail : jhjeong@os.korea.ac.kr

**이 상 민**



1999 서강대학교 전자계산학과 졸업  
2001 서강대학교 컴퓨터학과 석사  
2001~현재 서강대학교 컴퓨터학과 박사  
과정  
관심분야: 멀티미디어 어댑테이션, 동  
영상 압축  
E-mail : lesmin@nownuri.net

**최 진 영**



1982 서울대학교 컴퓨터공학 학사  
1986 미국 University of Drexel 응용  
전산학 석사  
1993 미국 University of Penns-ylvania  
전산학 박사  
1996~현재 고려대 컴퓨터학과 부교수  
관심분야: 실시간 컴퓨팅, 정형 기법,  
보안, 소프트웨어 엔지니어링, 프  
로토콜 엔지니어링  
E-mail : choi@formal.korea.ac.kr

• The 9th International Conference on Database Systems for Advanced Applications •

- 일 자 : 2004년 3월 17~19일
- 장 소 : 제주도
- 주 최 : 데이터베이스연구회
- 상세안내 : <http://aitrc.kaist.ac.kr/~dasfaa04>