

Light-weight 소프트웨어 CODEC을 이용한 2K 영상의 실시간 네트워크 전송에 관한 연구

준회원 조진용*, 박종철**, 문정훈*, kwak재승*, 종신회원 김종원**°

Real-time Networked 2K Video Transfer with Light-weight Software CODEC

Jinyong Jo*, Jongchurl Park**, JeongHoon Moon*, Jaiseung Kwak* Associate Members,
Jongwon Kim**° Lifelong Member

요약

실시간 및 고품질 특성에 대한 멀티미디어 요구 증대는 응용의 광대역화를 촉진시키며 네트워크 및 시스템 측면의 요구 사항을 증대시킨다. 본 논문은 고화질 HD(high definition) 영상을 소프트웨어 기반으로 실시간 압축하는 DXT 부호화와 초고속 네트워크 전송을 효과적으로 결합하는 프로토타입 시스템의 구현을 다룬다. 특히 구현된 프로토타입 시스템을 통해 광대역 응용에 요구되는 미디어 압축과 전송을 위한 네트워크 및 시스템 특성을 살펴본다. 개발 구축된 시스템의 성능 검증을 위해 과학기술연구망에 테스트베드를 구축하고 중단 간 전송 실험을 통해 효용성을 입증한다.

Key Words : Broadband networked application, DXT compression, high-definition video

ABSTRACT

The growing availability of broadband multimedia applications are providing strict real-time services and enabling users to feel much interactivity as well as 'sense of presence'. In that sense, there would be increased demand for what network and system have to do to fulfill the user expectations. In this paper, we implement a real-time multimedia application which make use of real-time DXT CODEC and investigate what the challenges of networking and system are. We carry out various experiments over a research network test-bed to evaluate the performance of our proposals.

I. 서론

광 네트워크 기술의 발달은 첨단 응용 데이터의 고속 전송을 위해 수십 Gbps의 백본 대역폭을 제공하는 단계에 있다. 네트워크 인프라의 확대와 시스템 자원의 폭발적인 증가는 CineGrid^[10], 비압축

HDTV^{[5],[7]}, Super Hi-Vision^[12]과 같은 수백 Mbps에서 수 Gbps급 광대역 멀티미디어 응용의 현실화 가능성을 높이고 있다. 광대역화를 촉진시키는 원인으로서는 실시간성의 확보와 고해상도 영상에 대한 요구 증가를 들 수 있다.

H.264 하드웨어 CODEC은 10Mbps 이하의 네트

* 한국과학기술정보연구원 CNI 사업단 (jjiny92, jhmoon, jskwak}@kisti.re.kr),

** 광주과학기술원 정보통신공학과 (jjcpark, jongwon}@netmedia.gist.ac.kr) (° : 교신저자)

논문번호 : KICS2008-06-278, 접수일자 : 2008년 6월 20일, 최종논문접수일자 : 2008년 9월 5일

워크 대역폭으로 풀 HD급(1080p, 2K¹⁾) 영상의 실시간 전송(약 100ms 이하의 코덱 지연)을 가능하게 한다¹¹⁾. 24Gbps의 대역폭이 필요한 8K 영상은 H.264 하드웨어 CODEC을 이용해 약 128Mbps로 실시간 압축가능하다¹²⁾. 또한, JPEG 2000 하드웨어 CODEC은 디지털 시네마용 4K 영상을 약 500~600Mbps의 데이터율을 갖도록 실시간 압축할 수 있다. 하지만, 이러한 하드웨어 CODEC들은 유연성이 떨어지며 매우 고가이기 때문에 R&E(Research & Education) 영역에 적용하기 어렵고 상용 방송 등의 제한된 영역에서 활용될 것으로 보인다.

비압축 HDTV는, 하드웨어 코덱과 비교했을 때 상대적으로 낮은 구축비용, 저지연, 고품질 등의 장점이 있지만 기본적으로 1Gbps 이상의 네트워크 대역폭이 필요하다는 문제점이 있다. 국내외 연구망들이 수십 Gbps의 백본 대역폭을 제공하지만 대부분의 종단 사용자들은 1Gbps급 접속망(Access network)에 연동되어 있기 때문에 시스템의 활용이 극히 제한적인 상태이다.

따라서 본 논문은 소프트웨어 CODEC을 이용해 2K 영상을 실시간으로 부호화, 송수신 및 재생할 수 있는 시스템에 대해서 제안하고 소프트웨어 CODEC 및 전송 시스템의 성능에 대해서 분석한다. 상용 시스템의 개발과 서비스에 목적을 둔 4K 및 8K 시스템들이 수백 Mbps급의 네트워크 대역폭을 필요로 한다는 점에서 유사한 광대역 멀티미디어 응용들에 대한 연구는 시스템 및 네트워크 특성들을 이해하는데 중요한 요소가 된다. 본 논문은 소프트웨어 CODEC의 어셈블리 구현을 실시간 영상 스트리밍 환경에 적용한 최초의 연구 결과라는 점에서 의미가 있으며 광대역 멀티미디어 응용의 유·무선망 서비스를 위한 시스템 및 네트워크 요구 사항을 연구하는데 목적을 둔다.

개발된 시스템은 소프트웨어 CODEC을 이용해 비압축 HDTV에 준하는 실시간성을 제공하며 네트워크 대역폭에 대한 요구를 1/6으로 줄임으로써 시스템 및 네트워크 측면의 유연성(flexibility)을 높인다. 이에 따라, 연구망 환경의 종단사용자들은 3자 회의 등이 가능한 2K급 고품질·실시간 TelePresence 환경을 제공할 수 있다. 또한, 개발 시스템은 R&E 영역에

서 교육 및 문화예술 지원을 위한 플랫폼 기술로도 활용 가능하다. 연구 결과의 확장을 통해 4k 영상 전송 등 초고해상도 영상의 실시간 스트리밍 환경 적용을 기대한다.

본 논문의 II절에서는 DXT 코덱을 이용한 실시간 미디어 전송과 관련된 연구를 조사하고, III절에서는 DXT 부호화에 대해서 살펴본다. 개발된 실시간 미디어 전송시스템의 구조에 대한 설명은 IV절에서 기술한다. V절에서 실험 환경 및 네트워크 적용 실험을 수행하고, VI절에서 결론을 맺는다.

II. 관련 연구

일반적으로 2K급 영상의 부호화를 위해 MPEG2, MPEG4, H.264, JPEG 2000 등의 소프트웨어 CODEC을 이용할 수 있다. 본 논문에서 하드웨어 코덱은 논외로 둔다. 이러한 소프트웨어 CODEC들은 높은 압축률로 인해 실시간성에 제약을 받는다. 실제로, IA32(Intel architecture 32) Xeon 2.4Ghz 시스템 상에서 MPEG2 코덱의 SIMD(single instruction multiple data) 최적화를 통해 2K급 영상을 부호화하면 약 9.3fps의 성능을 보이며 H.264 코덱은 0.7fps의 낮은 실시간 부호화 성능을 보인다¹³⁾. C/C++은 일반 변수(scalar)를 직접 다루기 때문에 스칼라형(scalar version) 구현으로 칭하는데 SIMD 최적화에 비해 매우 낮은 부호화 성능을 나타낸다.

실시간성을 높이기 위한 소프트웨어 CODEC으로 DXT 부호화가 있다 DXT 부호화는 그래픽 카드 전문회사인 S3 사에 의해서 텍스처 압축용으로 개발되어 S3TC(S3 texture compression)으로 불리기도 하며, 1970년대에 개발된 BTC(block truncation coding)를 적응화하여 개발된 Color Cell Compression 기법과 매우 유사하다. DXT 역시 낮은 부호화 속도로 인해 주로 오프라인 처리를 위해 사용되었으나, SSE2(streaming SIMD extensions 2)를 이용한 어셈블리 구현을 통해 실시간 부호화가 가능해졌다¹⁴⁾. FastDXT¹⁵⁾는 이러한 어셈블리 구현을 보다 이식성(portability)있게 재구현한 개발 결과물이다. GPU(graphics processing unit)를 이용한 실시간 DXT 부호화도 가능하지만 그래픽카드의 픽셀 리드백(pixel read-back) 속도에 따라 성능이 결정되는 문제가 있다¹⁶⁾.

FastDXT를 이용한 스트리밍 응용들은 SAGE(scalable

1) 본 논문에서 사용된 '2K'(K=1,000) 용어는 영상의 가로 해상도를 나타내며 편의상 풀HD급 (1920x1080) 영상을 지칭하는데 사용한다. 참고로 '2K' 및 '4K'는 극장용 영상의 가로해상도를 나타내며 '8K'는 Super Hi-Vision 시스템의 가로 해상도를 나타낸다.

2) 부호화 성능은 화상그룹(group of picture) 및 시스템 성능에 종속되므로 해당 결과는 절대적인 값으로 볼 수 없다.

adaptive graphics environment)^[9] 와 UltraGrid^[5] 환경에 실험적으로 적용한 예^{[11],[6]}가 있다. Luc의 연구^[11]는 FastDXT를 이용해 2K 및 4K 영상을 실시간 전송한 실험이라는 점에서 의미가 있다. 하지만, [1]은 저장형 콘텐츠에 대한 전송실험으로 실시간 영상의 스트리밍 응용에 대한 성능 분석은 포함하고 있지 않다. Petr의 연구^[6]는 2K 영상의 실시간 DXT 부호화 전송 및 재생에 관한 최초의 실험이라는 점에서 본 논문의 내용과 유사성이 높다. 하지만, 다중쓰레드 기반의 스칼라형 구현을 통해 영상을 부호화했기 때문에 낮은 사양의 시스템에서는 실시간 부호화가 어려운 문제가 있다. 또한, YCbCr 색공간 형태를 변환과정 없이 직접 부호화하므로 영상의 일그러짐 현상이 발생할 때, 색상 변화가 띠 모양으로 보이는 밴딩(banding) 현상이 RGBA 입력에 비해 큰 단점이 있다.

고품질 음성의 지원은 사용자간 상호작용을 향상시키고 응용분야의 확장(문화예술 등)을 위해 필수적인 요소이다. SAGE 환경은 16비트 44.1Khz 스테레오 음성원(audio source)을 제공하며 UltraGrid는 기존의 인터넷 회의용 툴인 RAT(robust addio tool)을 이용한다. RAT과 같은 인터넷 음성 툴을 이용할 경우, 미디어간 동기화의 문제가 발생할 수 있으며 응용 툴의 특성상(손실 환경하의 적응성 고려) 고품질 음성을 활용하기 힘들다. 개발시스템은 6채널 24비트 48Khz 음성원을 제공한다.

본 논문의 실시간 영상전송 시스템은 UltraGrid 기반의 영상공유 시스템^[7]의 확장 형태로 개발되었다.

III. DXT 부호화

DXT 압축은 블록 절단 부호화(block truncation coding) 방식의 일종으로 손실압축을 사용하며 4:1 또는 6:1의 고정비트 압축률을 갖는다. DXT 압축 포맷은 DXT1, DXT2, DXT3, DXT4, DXT5로 구성된다. DXT1은 32비트 RGBA 이미지의 알파 채널(A)을 이용하지 않거나 1 비트 채널만을 가지므로 가장 높은 압축률(6:1 혹은 8:1)을 제공한다.

본 논문에서는 비압축 32비트 RGBA 이미지의 실시간 전송을 위해 그림 1과 같은 DXT1 압축을 적용한다. DXT1 압축을 위해 하나의 정지 영상은 겹치지 않는 4x4 픽셀 블록들로 분할되며, 블록내의 픽셀(p)들은 4단계 보간을 통해 양자화 된다.

블록 내에서 임의로 선택된 2개의 24비트 RGB 픽셀들은 16비트 RGB5:6:5 형태의 색상값(C_0, C_1)

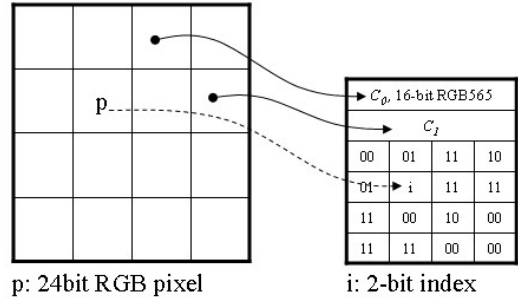


그림 1. DXT1 부호화

으로 변환되며 이들은 보간의 기준값이 된다. 4단계 보간을 위한 추가적인 색상값 C_2, C_3 는 다음과 같이 표현된다^[3].

$$C_2 = \frac{2}{3}C_0 + \frac{1}{3}C_1, C_3 = \frac{1}{3}C_0 + \frac{2}{3}C_1 \quad (C_0 > C_1)$$

$$C_2 = \frac{1}{2}C_0 + \frac{1}{2}C_1, C_3 = transparent \quad (C_0 \leq C_1)$$

블록내의 각 픽셀들은 4단계 보간값(색상값)들에 의해 표현될 수 있으므로 부호화 과정을 거치게 되면 픽셀 당 2비트를 갖는다. 즉, 2비트 인덱스 값들은 4개의 색상값을 검색테이블(lookup table)로 이용해 부호화 및 복호화를 수행한다. 결과적으로, 4x4 픽셀 블록이 부호화를 거칠 때, 2개의 16비트 색상값과 16개의 2비트 인덱스값을 갖게 되므로 총 64비트의 메모리 공간이 필요하다.

실효치(root mean square)에 의한 원본이미지와 DXT1 압축된 이미지의 평균 오차는 JPEG의 약 75% 수준이며 최악의 경우 25% 정도를 보인다^[11]. 실효치에 의한 평균 오차가 일정하지 않는 이유는 기준이 되는 색상값의 선택에 따라 영상품질이 결정되기 때문이다. 영상품질을 높이기 위해서는 원본이미지와 압축된 이미지 간에 최소 오차를 갖는 색상값을 선택해야 한다. 본 논문에서는 2K 영상의 부호화를 위해 FastDXT를 이용한다.

그림 2의 좌측은 256x256 크기의 PNG (portable network graphics) 포맷을 갖는 원본이미지이며 우측은 DXT1을 이용해 압축한 결과이미지이다. 시각적으로 확인하기는 쉽지 않지만, 압축이미지의 경우 경계면을 중심으로 화질열화가 발생함을 확인할 수 있다. 그림 3은 그림 2를 5배 확대 적용한 것으로써 우측의 DXT1 압축이미지가 4x4 픽셀 블록들을 중심으로 질저하가 발생하고 있음을 보여준다.

전체적으로 손실 압축의 특성 상 품질 저하 현상



그림 2. 원본이미지(좌)와 압축이미지(우)



그림 3. 원본이미지(좌)와 압축이미지(우)의 5배 확대

을 보이지만, 고품질 화상회의 등 영상 편집 이외의 용도로 사용된다면 매우 합리적인 수준의 품질을 유지할 수 있다. 또한, 소프트웨어적 구현을 통해 실시간 부호화가 가능하다는 점에서 네트워크 환경 하에서도 그 활용 가능성이 높다.

IV. 실시간 미디어 전송시스템

4.1 전송시스템 구성

FastDXT는 Intel사의 멀티미디어 명령어인 SSE2 나 스칼라 구현을 선택적으로 이용해 영상의 DXT1 부호화를 수행한다. 스칼라 구현을 이용할 경우, DXT1 부호화의 높은 계산복잡도를 충족시키기 위해서 최소 Intel Core2Quad 2.4Ghz 이상의 CPU 성능이 필요하다. 또한, 부호화를 위해 GPU를 직접 활용하기 때문에 하드웨어 가속이 가능한 그래픽카드를 이용한다. 개발 시스템은 x86 플랫폼에서 동작하는 32비트 리눅스 환경하에서 구현된다. 특별히 언급되지 않는 이상, 본 논문의 모든 실험은 Intel Core2Quad 2.4Ghz CPU를 이용한다.

그림 4는 개발된 고품질 미디어 전송시스템의 구성도이다. 미디어 캡처 인터페이스(Media capture interface)는 32비트 RGBA 색공간을 갖는 영상 프레임과 24비트 48Khz PCM 음성 프레임을 캡처해 프레임 버퍼에 저장한다. 캡처된 영상프레임은 어셈블리 구현을 이용해 실시간 압축된 후 패킷화를 거치며 음성 프레임은 압축과정 없이 패킷화된 후 송신된다.

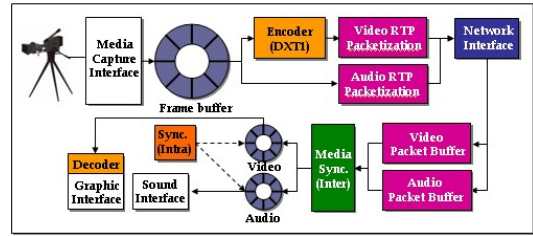


그림 4. 시스템 구성도

수신된 패킷들은 패킷 버퍼에 저장되며, 프레임화를 거친 직후 미디어간(inter-media) 동기화가 수행된다. 미디어간 동기화는 음성과 영상의 RTP timestamp를 이용한다. 미디어간 동기화를 완료한 음성 및 영상 프레임은 버퍼에 저장된 후, 미디어내(intramedia) 동기화를 수행한다. 미디어내 동기화는 정의된 타이머 인터럽트의 발생시간에 기초한다. 동기화를 완료한 영상 프레임은 재생 버퍼에 입력된 후 GPU 가속에 의해 복호화되며 재생을 위해 GBR 색공간의 형태로 변환된다. 음성의 경우, 6채널 24비트 PCM 데이터는 ALSA(advanced Linux sound architecture) 드라이버와의 호환성을 높이기 위해 2채널 16비트 형태로 하향 변환(down converting) 된 후 재생된다.

4.2 실시간 스트리밍을 위한 미디어 획득

DXT 부호화는 입력 색공간 형태로 RGB(A)를 수 용한다. 이를 위해서 8비트 YCbCr 색공간을 갖는 영상 프레임을 캡처해 32비트 RGBA 색공간으로 변환하는 방법과 32비트 RGBA를 직접 캡처해 부호화하는 방법이 있다.

색공간 변환을 위한 처리 부하(processing load)나 직접 캡처 후 복사할 때 발생하는 메모리 대입연산은 높은 시스템 부하를 발생시키므로 2K 영상의 실시간 처리 성능을 떨어트린다. 실제로, 표준 C의 memcpy() 함수로 32비트 RGBA 영상을 메모리 복사한 후 어셈블리 구현을 이용해 DXT1 부호화하면 대용량 메모리 복사의 영향으로 인해 초당 약 27 프레임 이하의 처리 성능을 보인다. 32비트 RGBA는 16비트 YCbCr에 비해 2배의 메모리공간을 필요로 하며, 초당 약 250 Mbytes의 실시간 메모리 복사가 요구 된다³⁾.

개발된 실시간 전송 시스템은 posix_memalign() 함수를 이용해 미디어 프레임이 저장될 메모리 주

3) DXT1 부호화를 위한 화상원(image source)의 크기는 8.2 Mbytes이지만, 부호화 과정에서 알파채널(A)이 사용되지 않기 때문에 6.2Mbytes 크기의 화상이 1.0Mbytes로 압축 된다. 결과적으로, 데이터율은 1.5Gbps에서 250 Mbps 낮아진다.

소를 정렬한 후, SSE2를 이용해 빠른 메모리 복사를 수행한다. SSE2의 *movdqa* 명령을 사용할 경우, 영상프레임을 위해 할당된 메모리의 첫 바이트주소는 16의 배수가 되어야 하기 때문에 메모리 정렬이 필수적이다.

4.3 네트워크 및 시스템 요구사항

부호화를 마친 영상프레임은 약 250Mbps의 네트워크 대역폭을 필요로 하며 12Mbps 대역폭이 추가적으로 필요한 6채널 PCM 음성을 포함해 총 262Mbps의 송신률을 갖는다. 단일 영상프레임의 크기는 1,036,800바이트이며 음성프레임은 49,152바이트이다. 네트워크 전송을 위해서 1,500바이트로 단편화할 경우 영상은 약 700개, 음성은 약 30개의 패킷으로 구성된다. 전체 프레임률이 29.97fps이므로 초당 약 22,000여개의 패킷들이 시스템에서 처리되어야 한다.

일반적으로 수 Gbps급의 대용량 데이터의 UDP 전송성능은 커널버퍼와 MTU(maximum transfer unit) 크기에 비례한다. 하지만, 262Mbps의 낮은 송신률을 갖는 개발시스템이 대형프레임(jumbo frame)을 사용함으로써 얻을 수 있는 전송성능의 이득은 매우 적다.

시스템 측면에서 살펴보면, 9,000바이트 MTU를 사용하게 되면 1,500바이트 MTU와 비교했을 때 보다 CPU 부하를 약 34% 줄일 수 있다. Intel Core2Quad 2.4Ghz 환경에서 9,000바이트 MTU는 56%의 CPU 부하를 보인 반면 1,500바이트의 경우에는 약 84%의 CPU 점유율을 보였다. CPU부하의 감소는 RTP 패킷 헤더와 페이로드(payload) 헤더값들에 대한 할당횟수가 줄어들기 때문이다.

단일코어(single core) CPU를 사용하거나 다중코어(multi-core) CPU상에서 단일쓰레드(single thread)를 이용해 부호화할 경우, 네트워크를 위해 발생하는 인터럽트가 부호화를 위해 사용되는 CPU와 중첩되므로, 높은 CPU 부하는 실시간 부호화 성능을 악화시킨다. 따라서, 9,000바이트 MTU를 사용해 CPU 부하를 최소화하거나, 부호화와 네트워크를 위해 발생하는 인터럽트들이 하나의 CPU로 집중되지 않도록 해야 한다.

리눅스 환경 하에서 사용자는 IRQ(interrupt requests)에 대한 처리를 특정 CPU에 할당할 수 있다. 다음 명령과 같이 이더넷 인터페이스에서 발생하는 IRQ를 처리할 CPU와 부호화를 담당할 CPU와 다르게 설정함으로써 네트워크 부하로 인한 부호화 성능악

화를 방지할 수 있다. n 은 CPU 번호로써 부호화를 위한 쓰레드의 개수보다 커야한다. $\langle irq \rangle$ 는 이더넷 인터페이스의 IRQ 번호를 나타낸다.

```
echo 2n > /proc/irq/<irq>/smp_affinity
```

이와 같이 실시간 부호화를 통해 영상의 데이터율이 줄면 일반 인터넷 환경에서 사용하는 표준 프레임을 이용할 수 있으므로 네트워크 측면의 유연성이 향상된다. 대형프레임을 사용하기 위해서는 네트워크 장비의 지원이 필요하다.

4.4 FastDXT 성능 분석

Pentium4 3.0Ghz 상에서 단일 쓰레드와 어셈블리 구현을 이용해 2K 해상도의 영상을 부호화할 경우에 약 48프레임을 실시간 부호화할 수 있다^[1]. 네트워크를 통한 실시간 스트리밍은 영상의 캡처와 송수신 등 추가적인 프로세싱을 필요로 하므로 부호화 성능이 떨어진다.

성능 측정을 위해 RDTSC(real time-stamp counter)를 이용해 μs 단위의 시간 해상도(time resolution)를 유지한다. 표 1은 어셈블리 구현을 이용해 2K 해상도를 갖는 영상을 실시간 부호화했을 때 나타나는 DXT1 부호화 성능이다. 2K 입력 신호의 프레임률은 초당 29.97이다. 어셈블리 구현 대신 스칼라 구현을 이용해 부호화할 경우(쓰레드 3개 사용), 평균 29.38fps를 보이며 CPU 부하는 코어 당 약 83%를 차지한다. 높은 CPU 부하로 인해, 우선순위가 높은 시스템 서비스의 인터럽트 요청이 발생하면 시스템의 부호화 성능이 크게 떨어진다. 따라서, 29.97fps를 갖는 2K 영상의 실시간 부호화를 위해서 스칼라 구현을 이용하는 것은 적합하지 않다.

단일 쓰레드를 적용했을 때, 영상 프레임을 부호화하는데 약 31.3822ms의 시간이 소요되며 쓰레드의 개수가 증가할수록 부호화 성능이 높아짐을 확인할 수 있다. 하지만 부호화 시간의 표준 편차도 함께 증가한다. 부호화 시간의 표준 편차가 증가하면 초당 부호화 프레임률(송신측에서 초당 부호화되는

표 1. DXT1 부호화 성능(ms)

항목	1 thread	2 threads	4 threads
부호화시간(avg.)	31.3822	22.3753	21.2077
부호화시간(std.)	0.1512	1.4140	1.8292
프레임률(avg.)	29.9700	29.9718	29.9717
프레임률(std.)	0.0000	0.1011	0.1070

프레임의 개수)의 표준편차도 함께 증가하기 때문에 송신측 또는 수신측에서 평활(smoothing)을 위한 부담이 증가한다(4개의 스트레드를 사용할 경우 부호화 프레임들의 표준편차가 약 0.1070로 증가한다). 평활은 미디어 동기화를 위한 추가적인 버퍼링 시간을 요구하기 때문에 바람직하지 않다.

다중 스트레드를 사용할 경우에 부호화 시간의 표준편차가 커지는 이유는 스트레드의 작업 간 전환(context switching)이 실시간으로 이루어 지지 않기 때문이다. 2K 이상의 해상도를 갖는 영상(예, 4K 디지털 시네마)의 부호화나 프레임들의 배가(doubling)가 필요한 영상(예, 시속차 스테레오스코픽(time-sequential stereoscopic) 영상)의 경우 실시간 부호화를 위해서 고사양의 시스템과 평활이 필요함을 유추할 수 있다. 표 1의 결과를 바탕으로 본 논문에서는 실시간 DXT1 부호화를 위해 단일 스트레드에 기반 어셈블리 구현을 이용한다.

4.5 실시간 전송 및 미디어 동기화

부호화된 영상과 음성은 각각의 RTP 세션을 구성하며 헤더 형태는 IEFT RFC 3550에 따른다. 음성 RTP 패킷은 PCM 음성원과 재생에 필요한 메타데이터를 포함한다. n 번째 영상과 음성 프레임은 $1/f_s$ (약 33.3ms) 간격을 두고 동일한 시간(t_n)에 캡처되며 수신측에서 동기화를 용이하게 하기 위해 RTP time-stamp값을 공유한다. f_s 는 영상의 재생 프레임률이다.

수신된 패킷들의 지연변이(j_i)를 흡수하기 위해 해당 프레임을 구성하는 첫 번째 패킷의 도착시간을 기준으로 최대 $1/f_s + 3 \cdot j_i$ 초의 시간 지연을 갖는다. 음성은 미디어간 동기화를 용이하게 하기 위해 영상의 재생 프레임률(예, 29.97fps)에 동기화시킨다. 이와 같은 음성 캡처방식은 소수점연산으로 인한 잉여 데이터가 발생하기 때문에 음성프레임의 시작 채널이 일정하지 않은 문제가 있다. 수신단에서는 채널 정보의 뒤섞임을 막기 위해 음성 채널을 재정렬한다.

수신단의 패킷버퍼(그림 5)에서 프레임화가 완료(t_n^v, t_n^a)되면 패킷의 RTP time-stamp를 읽어 미디어간 동기화를 수행한다. 미디어간 동기화는 음성과 영상의 프레임 크기 차에 따른 전송완료 시간의 차이와 네트워크 지연변이 등에 의한 미디어 프레임간 동기화 오차($t_n^v - t_n^a$)를 흡수한다. 개별 영상 프레임과 음성 프레임의 크기가 각각 S_v, S_a , 네트워

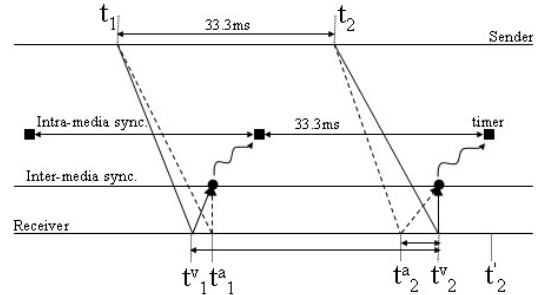


그림 5. 미디어 송수신 및 동기화

크 용량이 C 이고 지연변이가 영상과 음성 패킷에 동일하게 적용된다고 가정했을 때, 미디어간 동기화를 위한 추가 지연은 약 $|(S_v - S_a)/C|$ 이다. 패킷들이 연속적으로 손실되지 않는 이상 미디어 프레임이 손실되지 않으므로 프레임 손실로 인한 미디어간 동기화 문제가 발생할 확률은 매우 낮다. 프레임 손실이 발생하면 타임스탬프 값이 작은 미디어 프레임을 패킷 버퍼에서 삭제함으로써 동기화를 유지한다. 미디어간 동기화가 완료된 프레임들은 영상 및 음성 프레임버퍼에 저장된다.

미디어내 동기화를 위해 POSIX(portable operating system interface) 타이머를 이용한다. 타이머는 미리 정의된 재생 프레임률에 기초해 주기적으로 소프트웨어 인터럽트를 발생시킴으로써 영상 또는 음성의 미디어내 도착 시간의 차이($|t_n^v - t_{n-1}^v|, |t_n^a - t_{n-1}^a|$)로 인한 동기화 오차를 흡수한다. 인터럽트가 발생하면 프레임버퍼에서 해당 프레임을 인출해 미디어 인터페이스의 재생 버퍼로 이동시킨 후 재생(t_n^v)한다. 미디어내 동기화를 위해 각 프레임은 평균 33.3ms(1 프레임 시간)의 버퍼링을 한다.

V. 실험결과

본 절에서는 네트워크 전송 실험을 통해 개발시스템의 성능을 살펴본다.

네트워크 전송실험을 위해 국가 과학기술연구망인 KREONET 상에 그림 6과 같이 테스트베드를 구축했다.

중단 간 총 4홉의 3계층 스위치를 거치며 병목(bottleneck)지점의 대역폭은 1Gbps이다. 백그라운드 트래픽은 상향(좌로부터 우) 196Mbps이고 하향(우로부터 좌)으로 140Mbps이다. 상향 전송을 위해 1,500 바이트 표준 프레임을 이용했고 하향으로 9,000 바이트 대형프레임이 사용되었다. 실험은 총 150분간

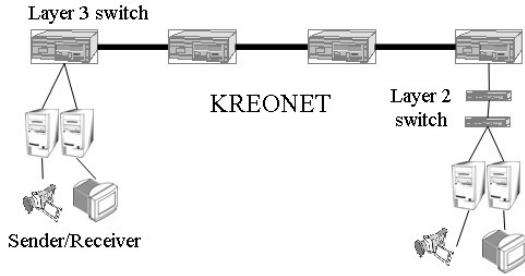


그림 6. 실험 환경

표 2. round-trip time 값 (ms)

항목	최소값	평균값	최대값	표준편차
9k 프레임	5.61	12.49	18.66	5.32
1.5k 프레임	3.44	8.02	16.47	5.49

표 3. 프레임간 시간간격(ms)

항목	음성	영상	미디어간
평균 시간간격	33.3669	33.3669	0.0084
최대 시간간격	92.7100	119.8401	0.0948
최소 시간간격	0.1699	1.719952	0.0040
표준편차	1.3532	1.993437	0.0016

진행되었다.

표 2는 하향으로 측정된 왕복전송지연(round-trip time)을 보여준다. 패킷 밀도(packet burstness)를 높이기 위해서 패킷 간 출발시간 간격을 0으로 설정한 후 측정했다.

표 3은 영상과 음성을 동기화하기 전에 측정된 미디어내(intra-media) 또는 미디어간(inter-media) 프레임 도착시간의 간격을 보여준다. 음성과 영상 프레임의 미디어간 도착시간의 차이는 평균 8.4 μ s이며 영상 또는 음성의 미디어내 도착시간의 차이는 평균 33.3668ms이다. 미디어간 도착시간의 최대 간격이 94.88 μ s이지만, 미디어내 도착시간 간격의 최대값이 119.8401ms(영상)와 92.7100ms(음성)를 보임으로써 미디어내 동기화에 대한 필요성이 미디어간 동기화에 대한 요구보다 높음을 알 수 있다. 발생한 미디어간 도착시간의 차이는 동기화를 통해 100% 흡수된다⁴⁾. 미디어내 동기화는 타이머 인터럽트에 의해 수행되며 평균 1 프레임 시간의 버퍼링이 필요하다. 영상 또는 음성 프레임이 미디어내 동기화

를 위해 버퍼링 되는 평균 대기시간은 31.79ms였으며 최대 79.22ms, 최소 0.22ms를 나타냈다.

압축과정을 생략하고 영상과 음성을 네트워크 전송한다고 가정했을 때, 송신 시스템에서 발생하는 최대 지연(worst case)은 33.3ms이며 이는 미디어 캡처를 위한 버퍼링 시간이다. DXT1 부호화된 영상을 네트워크 전송할 경우, RGBA 이미지의 메모리복사와 부호화 등으로 인해 버퍼링 시간이 증가된다. 실측된 버퍼링시간은 최소(best case) 33.3ms이며 최대(worst case) 66.6ms 이다. 비압축(uncompressed) 시스템과 비교했을 때, 약 1 프레임 시간 정도의 추가 지연이 송신측에서 발생한다. 수신 시스템에서도 동기화를 위해 최대 79.22ms 정도의 지연이 생기므로 송신과 수신 시스템에서 발생하는 최대 프로세싱 지연은 약 145ms(최선의 경우 약 34ms)이다.

DXT1 부호화 영상의 실효치에 의한 평균 품질은 JPEG의 75%^[1] 수준 이므로 영상에 대한 추가적인 품질 측정은 생략한다. 미디어의 캡처, 부호화, 송수신 및 재생을 위한 각 과정은 실시간 처리 성능에 상호 영향을 준다. 예를 들어, 부호화 성능이 낮을 경우 전체 송신률이 감소하며 미디어내 동기화 성능이 떨어질 경우 재생 프레임률의 표준편차가 커지게 된다. 송신 시스템에서 측정된 전송 프레임률은 약 29.9705fps이며 표준편차는 0.0123을 나타냈다. 수신측의 재생 프레임률은 약 29.9697fps이고 0.0104의 표준편차를 가졌다. 영상 신호의 입력이 29.97fps이고 리눅스 플랫폼의 낮은 시간 해상도를 감안할 때, 개발된 시스템이 부호화, 미디어 동기화 및 재생을 안정적으로 지원하고 있음을 확인할 수 있다.

그림 7은 개발 시스템을 활용한 고품질 화상회의 장면을 보여준다. 모니터(1920x1200)에서 재생되는 영상을 1배 확대 비율로 캡처 했다. 비월 주사(interlaced scanning) 방식의 신호를 디인터레이싱(deinterlacing) 하지 않기 때문에 움직임이 큰 영상일 경우 가공현



그림 7. 캡처 화면(1배)

4) 하드웨어(사운드 및 그래픽카드)의 내부 버퍼링으로 인해 발생하는 미디어간 오차는 고려하지 않았다.



그림 8. 확대 화면(8배)

상(artifacts)이 발생하지만 DXT1 손실 부호화로 인한 품질 저하를 시각적으로 구분하기는 쉽지 않다. 그림 8은 품질 저하를 확인하기 위해서 그림 7의 특정 부위를 8배 확대한 화면이다. 색상의 테두리를 중심으로 색번짐 현상이 있음을 알 수 있다.

VII. 결 론

본 논문은 고품질 멀티미디어 콘텐츠의 실시간 송수신을 위한 기본형 시스템을 구현하고 네트워크 전송실험을 통해 성능을 평가했다. 또한, 광대역 멀티미디어 응용의 R&E 환경 적용을 위한 시스템 및 네트워크의 요구사항에 대해 연구했다.

개발 시스템은 최악의 경우 약 150ms 이내의 프로세싱 지연을 보임으로써 멀티미디어 데이터의 실시간 처리 능력과 사용자 간 상호작용성을 극대화시켰으며 인터럽트 분산을 통한 표준 프레임의 사용은 네트워크 측면의 유연성을 높였다.

손실 압축의 특성 상 품질 저하를 피할 수는 없지만 시각적으로 원영상과 비교했을 때 매우 합리적인 수준으로 판단된다. 개발 시스템이 R&E 영역에 적용되고 광대역 멀티미디어 응용의 특성을 파악하는 기초 자료로 활용되기를 기대한다.

참 고 문 헌

[1] L. Renambot, B. Jeong, J. Leigh, "Realtime Compression for High-resolution Content," *Access Grid (AG) Retreat*, 2007.
 [2] Fast DXT: A real-time DXT compression library, <http://www.evl.uic.edu/cavern/fastdxt/>
 [3] S3 Texture Compression(S3TC), <http://en.wikipedia.org/wiki/S3TC>
 [4] J.M.P. van Waveren, Real-Time DXT Compression, May 2006, Id Software, Inc., <http://www.intel.com/cd/ids/developer/asmo-na/eng/324337.html>

[5] C. Perkins, L. Gharai, T. Lehman, and A. Mankin, "Experiments with delivery of HDTV over IP networks," *Proc. Packet Video Workshop*, April 2002.
 [6] P. Holub, "Updates on UltraGrid Platform," *Internet2 Spring Member Meeting*, April 2008.
 [7] J. Jo, J. Chae, M. Lee, J. Kim, and O. Byeon, "Uncompressed High Definition Visual Sharing System toward Interactive Networked Collaboration," *Proc. IMMERSCOM 2007*, Oct. 2007.
 [8] M. Alvarez, E. Salami, A. Ramirez, and M. Valero, "HD-VideoBench: A Benchmark for Evaluating High Definition Digital Video Applications," *Proc. IISWC 2007*, pp.120-125, Sept. 2007.
 [9] B. Jeong, L. Renambot, R. Jagodic, R. Singh, J. Aguilera, A. Johnson, J. Leigh, "High-Performance Dynamic Graphics Streaming for Scalable Adaptive Graphics Environment," *Proc. SC06*, Nov. 2006.
 [10] CineGrid <http://www.cinegrid.org/>
 [11] HaiVision, "MAKO-HD, The TelePresence Codec," <http://www.haivision.com/>
 [12] EE Times-Asia, "NHK bets on Super Hi-Vision as future TV," Sept. 2007.

조 진 용 (Jinyong Jo)

준회원



1999년 2월 전남대학교 컴퓨터 공학과
 2002년 8월 광주과학기술원 정보통신공학과 석사
 2007년 8월~현재 광주과학기술원 정보통신공학과 박사과정
 2003년 8월~현재 한국과학기술정보연구원 선임연구원

<관심분야> 멀티미디어 응용 및 네트워크 프로토콜

박 종 철 (Jongchurl Park)

준회원



2007년 2월 전북대학교 전자정보공학부
 2007년 3월~현재 광주과학기술원 정보통신공학과 석사과정
 <관심분야> 네트워크 및 멀티미디어 응용 설계

문 정 훈 (Jeonghoon Moon)

준회원



1997년 2월 경일대학교 컴퓨터
공학과 학사
1999년 2월 경북대학교 컴퓨터
공학과 석사
2000년 5월~현재 한국과학기술
정보연구원 선임연구원
<관심분야> 정보통신분야,

멀티미디어 응용 분야

곽 재 승 (Jaiseung Kwak)

준회원



1995년 2월 전남대학교 산업공
학과 석사
1999년 2월 전남대학교 산업공
학과 박사
2000년 6월~현재 한국과학기술
정보연구원 선임연구원
2007년 1월~현재 한국과학기술

정보연구원 연구망개발팀장

<관심분야> 정보통신공학, 광통신 공학

김 종 원 (Jongwon Kim)

중신회원



1994년 2월 서울대학교 학사,
석사, 박사
1994년~1999년 공주대학교 전
자공학부 조교수
1998년~2001년 Univ. of Southern
California Dept. of Engineering
- Systems 연구조교수

2001년 9월~현재 광주과학기술원 정보통신공학과
정교수

<관심분야> 네트워크 미디어 시스템 및 프로토콜