SPIHT 웨이브릿 기반의 비디오 데이터의 멀티스트림 전송 기법

강 경 원, 류 권 열, 권 기 룡, 문 광 석, 김 문 수

Multi-stream Delivery Method of the Video Data Based on SPIHT Wavelet

Kyung-Won Kang, Gwon-Yeol Ryu, Ki-Ryong Kwon, Kwang-Seok Moon, Moon-Soo Kim

요 약

본 논문에서는 현재의 인터넷 환경하에서 클라이언트가 가진 대역폭을 최대한 활용하여 최선의 서비스를 제공할 수 있는 웨이브릿 기반에서 SPIHT(set partitioning in hierarchical trees)를 이용한 비디오 데이터의 압축과 멀티스트림을 이용한 전송 기법을 제안한다. 실험 결과, 제안한 방법은 웨이브릿 기반 비디오 부호화기를 이용하였기 때문에 동일한 비트율에서 기존의 DCT(discrete consine transform) 기반 비디오 부호화보다 블록킹 현상이 없이 화질이 약 1.5dB가 상승하였으며, 점진적 전송이 가능하였다. 그리고, 고압축시 발생하는 에러 전파를 막기 위해 TCP(transmission control protocol)상에서 효율적인 전송과 네트워크의 지터에 강하며 클라이언트가 가진 대역폭을 최대한 활용할 수 있는 멀티스트림 전송을 통해 그 효용성을 확인하였다.

Abstract

In this paper, we proposed the compression technique of the video data using SPIHT(set partitioning in hierarchical trees) based on wavelet and the multi-stream delivery method for best-effort service as fully utilizing the clients bandwidth over the current Internet. The experiment shows that the proposed method provides about 1.5dB better picture quality without block effects than DCT(discrete consine transform) based coding schemes at the same bit rates because of using the wavelet video coder. In addition, this technique implements the multi-stream transmission based on TCP(transmission control protocol). Thus, it is provided with the best-efforts service which is robust to the network jitter problem, and maximally utilizes the bandwidth of the client's.

key words : video stream, internet, multi-streams, SPIHT, wavelet

Ⅰ. 서 론

인터넷은 WWW의 대중화에 힘입어 불과 30년이라는 짧은 기간 안에 양적, 질적으로 팽창하고 있다. 특히 스트리밍 기술의 대두로 인터넷과 멀티미디어가 결합한 서비스가 증가하고 있기 때문에, 멀티미디어 데이터의 효율적인 통신을 위해서는 고압축 부호화 기법의 적용이 필수적이다[1-3]. 지금까지 공간, 주파수, 시간 영역 상관성을 이용한 정보 압축 기법들이 제안되어 왔다. 특히 DCT(discrete cosine transform)로 대표되는 블록변환에 근거한 기법들은 JPEG, MPEG-1, MPEG-2 등 각종 동영상 부호화의 국제 표준 방식에서 활용되고 있다[4-6]. 그러나 DCT를 근간으로 하는 기존의 표준화된 부호화 기법을 사용할 경우 블록 변환 기법의 특성상 고압축 시 블록킹 현상(blocking effect)이 발생하여 화질 열화를 초래한다. 또한 고압축시 현재 프레임과 이전 프레임간의 데이터 의존성이 높아지기 때문에 패킷 손실이 발생할 경우 복원 영상에 심각한 화질 열화를 야기할 수 있기 때문에, 고압축과 더불어 신뢰성 있는 전송도 요구된다. 현재 사용되고 있는 통신망은 인터넷과 같은 비동기적 패킷 교환망이기 때문에 멀티미디어 데이터를 전송할 경우 생성된 미디어들의 원래 시간 정보는 네트워크와 시스템에서의 지터(jitter)에 의한 전송 지연의 영향을 받아 전송 성능이 떨어지는 단점을 가지고 있다[7].

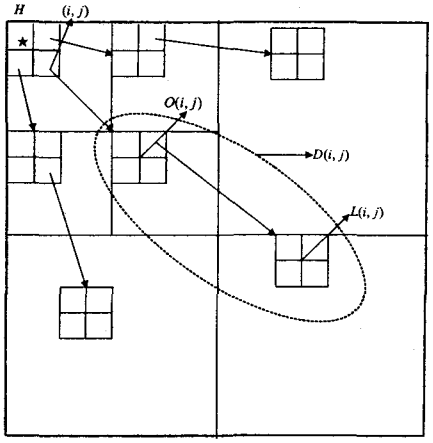
본 논문에서는 현재의 인터넷 인프라 환경하에서 클라이언트가 가진 대역폭을 최대한 활용하여 최선의 서비스를 제공할 수 있는 SPIHT(set partitioning in hierarchical trees) 웨이브릿 기반 비디오 데이터의 멀티스트림 전송

기법을 제안한다. 웨이브릿 기반 부호화는 인간의 시각특성과 유사하게 공간 및 주파수의 해상도를 가변적으로 표현함으로써 평탄하게 변하는 부분과 윤곽선과 같이 급격하게 변하는 부분을 동시에 효과적으로 다룰 수 있으며, 화소의 증가에 따른 계산량도 선형적으로 증가하므로 블록 변환 기법이 필요 없을 뿐만 아니라, 낮은 비트율에서 블록킹 효과가 없으며, 점진적 전송이 가능하여 고압축 부호화에 적합하다[8]. 그러나 고압축을 수행할 경우 패킷 손실로 인한 에러가 전파될 수 있기 때문에, 본 논문에서는 TCP에서 효율적인 전송과 네트워크의 지터에 강하며 클라이언트가 가진 대역폭을 최대한 활용할 수 있는 멀티스트림을 사용한 전송 방법을 제안한다.

II. SPIHT 웨이브릿기반 부호화

SPIHT는 EZW(embedded zerotrees wavelet) 부호화 방법을 개선하여 압축 효율을 향상시킨 방법이다[9,10]. 그림 1은 웨이브릿 부밴드간의 공간적 자기 유사성을 이용한 공간-방향 트리에서의 부모-자식 의존관계를 나타낸 것이다. 트리의 각 노드는 하나의 화소에 대응하며, 하나의 화소 좌표의 직접적인 자손(offspring)은 다음 레벨에서 같은 공간적 위치에 있는 화소들에 대응한다. 트리는 각 노드가 자식을 갖지 않거나 4개의 자식을 갖도록 정의되며, 이 4개의 자식들은 항상 2×2 이웃 화소들로 그룹을 형성한다. 그러나 그림 1에서 별표 표시된 부분들은 자손을 갖지 않는다. 부호화 과정은 초기화 과정과 분류 과정(sorting pass) 그리고 세분화 과정(refinement pass), 마지막으로 양자화 단계 갱신 과정(quantization-step update)으로 구성된다. 초기화 과정은 먼저 부분집합들의 중요성 테스트를 하는 순서가 중요하기 때문에 중요성 정보가 저장되는 목록을 다음과 같이 세 가지 순서 목록으로 구분한다. 세 가지 목록에는 중요하지 않은 집합들의 목록인 LIS(list of insignificant sets), 중요한 화소들의 목록인 LSP(list of significant pixels), 그리고 중요하지 않은 화소들의 목록인 LIP(list of insignificant pixels)가 있다. 이 목록들에는 좌표 (i,j)가 저장되며 이 좌표가 LSP와 LIP에서는 각 화소를 의미하지만 LIS에서는 그 노드 (i,j)의 모든 자손들의 좌표 집합인 $D(i,j)$를 의미하는 type-A과 모든 자식들의 좌표 집합을 뺀 $L(i,j)$를 의미하는 type-B로 표현된다

분류 과정에서는 LIP에 있는 모든 화소들이 테스트되며, 중요한 화소인 경우 LSP에 포함된다. 비슷한 방법으로 LIS에 있는 집합들도 테스트되며, 중요한 집합인 경우 목록에서 제거되면서 4개의 새로운 집합을 형성한다. 새로운 부분집합들은 하나이상의 구성요소들을 가지면 다시 LIS에 들어가게 되며, 하나의 좌표만 가지고 있으면 이 화소값이 중요도에 따라 LSP와 LIP에 각각 들어가게 된다. 그리고 세분화 과정에서는 LSP에 있는 모든 화소들의 MSB(most significant bit)는 수신측으로 전송된다. 이들 과정은 그림 2와 같이 단계별로 수행되며, 정해진 비트율에 도달할때까지 반복 수행된다.



$O(i,j)$: 노드 (i,j) 의 모든 자식들의 좌표 집합
$D(i,j)$: 그 노드 (i,j) 의 모든 자손들의 좌표 집합
H: 모든 공간 근원 트리 뿌리들의 좌표 집합

그림 1. 공간-방향 트리에서의 부모-자식 의존관계 예
Fig. 1. Examples of parent-offspring dependencies in the spatial-orientation tree.

<step 1> 초기화 과정
$n = \lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\}) \rfloor$, (단, $c_{i,j}$∈모든 화소)
LSP= ∅, LIP={ all pixels in lowpass subband }.
LIS={ pixels in lowpass subband | which have descendants}
→ type-A로 설정.
<step 2> 분류 과정
while((i,j) of LIP) {
LIP에 있는 각각의 (i,j)에 대해 $S_n(i,j)$을 전송.
if($S_n(i,j)$ == 1) (i,j)를 LSP에 포함, $c_{i,j}$의 부호를 전송.
}
while ((i,j) of LIS) {
if (type-A) {
$S_n(D(i,j))$을 전송.
if($S_n(D(i,j))$==1) $k,l \in O(i,j)$에 대해 $S_n(k,l)$을 전송.
if($S_n(k,l)$==1) (k,l)을 LSP에 포함, $c_{k,l}$의 부호를 전송.
if($S_n(k,l)$ == 0) (k,l)을 LIP의 끝에 추가.
if($L(i,j)\neq 0$) (i,j)를 LIS의 끝에 type-B로 포함.
else (i,j)를 LIS에서 제거.
}
if (type-B) {
$S_n(L(i,j))$를 보냄.
if($S_n(L(i,j))$==1) $(k,l) \in O(i,j)$를 LIS에 type-A로 추가.
(i,j)를 LIS에서 제거.
}
}
<step 3> 세분화 과정
마지막 분류 과정에서 포함된 것을 제외하고 LSP에 있는 모든 $|c_{i,j}|$의 n 번째 MSB를 전송.
<step 4> 양자화 단계 갱신 과정
$n = n-1$, step 2로 이동.

그림 2. SPIHT 부호화 알고리즘
Fig. 2. SPIHT encoding algorithm.

III. SPIHT 웨이브릿 기반의 비디오 부호화

현재 비디오 압축 부호화는 기본 알고리즘으로 DCT를 사용하고 있다. DCT의 단점은 정규화된 블록 단위의 연산이 수행되므로 양자화시 발생하는 블록간 이웃되는 픽셀들간의 불일치, 블록단위로 이뤄지는 움직임 예상과 움직임 보상에서 이웃된 블록들간에 생기는 불일치, 그리고 시스템 버퍼내의 참조 영상에서 일어나는 블록간의 불일치로 인한 블로킹 현상이 발생한다. 특히 고압축일 경우 블록킹 현상이 심하게 나타나는 단점이 있다. 이러한 단점을 극복할 뿐만 아니라 점진적 전송 특성을 얻기 위해 MPEG-2 비디오에 대해 웨이브릿 변환 영역에서 SPIHT를 이용한 비디오 부호화 방법이 제안되었다[11]. 본 논문에서는 저속 전송 매체를 위한 비디오 부호화를 위해 SPIHT를 이용한 비디오 부호화기를 사용하였다. 비디오 스트림은 I 프레임과 P 프레임으로 구성되어 있으며, I 프레임은 하나의 프레임으로부터 압축되어 독립적으로 복원이 가능한 프레임으로, 웨이브릿 기반에서 SPIHT와 산술 부호화를 사용하여 부호화하였다. P 프레임은 I 프레임 또는 다른 P 프레임을 기준으로 삼아 움직임 예측하여 움직임 벡터(*motion vector*)를 생성한 후 움직임 벡터를 이용한 보상 과정 후 생성된 나머지 차이 분을 SPIHT를 사용하여 부호화하였다. 움직임 추정은 16×16 블록에 대해 반화소 단위로 수행하였다. 그리고 움직임 보상을 위한 예측 영상은 H.263의 OBMC(overlapped block motion compensation)을 사용하였다[12]. 그림 3은 SPIHT를 이용한 고압축 비디오 부호화기의 블록 다이어그램을 나타낸 것이다. SPIHT 비디오 부호화기는 H.263의 기본 구조에 DCT 대신 DWT(discrete wavelet transform)와 SPIHT를 사용하였으며, 움직임 보상을 위해 공간영역에서 OBMC을 사용하였다.

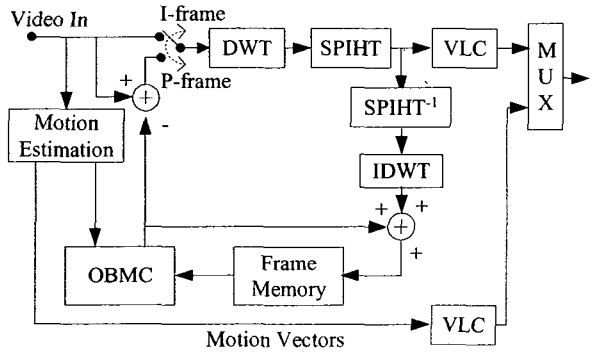


그림 3. SPIHT 웨이브릿 비디오 부호화기의 블록도
Fig. 3. Block diagram of the SPIHT wavelet video coder.

IV. 제안한 멀티스트림 전송 기법

현재 인터넷 방송에서 가장 많이 사용되고 있는 비디오 스트림들은 클라이언트와 미디어 서버간에 단일 스트림 형태로 접속이 된다. 따라서, TCP 기반에서 지터가 발생할 경우 클라이언트는 그 영향을 그대로 받게 되어 전송 성능이 떨어진다. 따라서 본 논문에서는 네트워크의 지터에 효율적으로 대처할 수 있을 뿐만 아니라 클라이언트 측의 대역폭을 가능한 최대로 활용할 수 있는 멀티스트림 전송 기법을 제안한다.

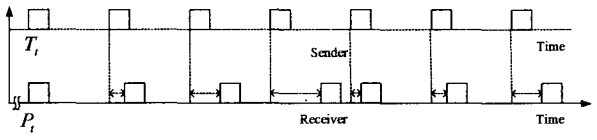


그림 4. 지연 지터의 예
Fig. 4. Example of delay jitter

그림 4는 송신측과 수신측에서 본 지터에 따른 상태를 보인 것이다. 송신측에서 동일한 시간간격으로 미디어를 전송하더라도 수신측에서는 불규칙적인 간격으로 미디어가 수신됨을 알 수 있다. 여기서 T_t는 송신측의 전송시간을 의미하며, P_t는 재생 시간을 의미한다.

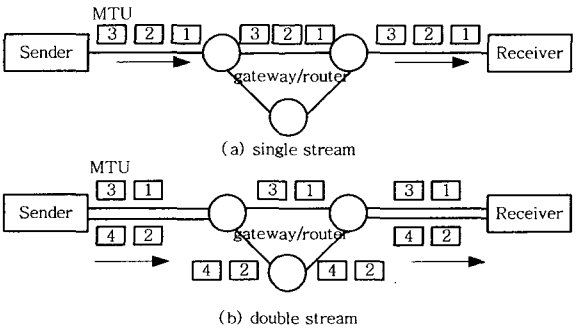


그림 5. 단일스트림과 이중스트림을 사용한 전송 예
Fig. 5. Example of the transmission using the single stream and double streams.

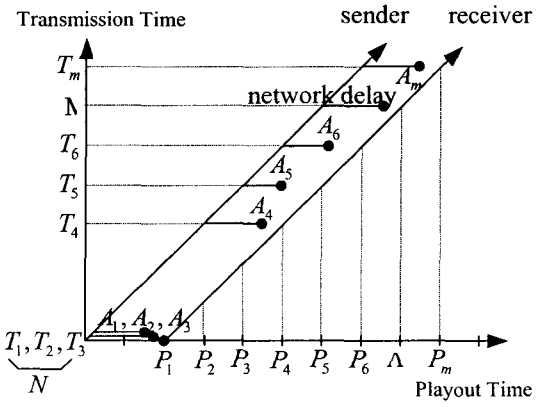


그림 6. 3개 스트림을 이용한 전송 예
Fig. 6. Example of delivery using three streams

그림 5는 멀티스트림의 효능을 설명하기 위해 기존의 인터넷 상에서 단일 스트림과 멀티스트림으로 전송하는

과정을 나타낸 것이다. 전송은 MTU(maximum transmission unit) 단위로 순차적으로 수행된다. 단일 스트림을 사용할 경우 하나의 MTU 단위로 데이터가 전송되지만, 멀티스트림으로 전송할 경우 각각의 스트림마다 개별적인 MTU 단위로 데이터가 전송되기 때문에 총 패킷의 크기가 전송 대역폭보다 작을 경우 모든 MTU를 병렬적으로 전송할 수 있어 전송 효율이 좋아질 뿐만 아니라, 하나의 스트림에 지터가 발생하더라도 다른 스트림은 계속 전송될 수 있기 때문에 네트워크 지터에도 강하게 된다. 그림 6은 본 논문에서 제안한 멀티스트림을 이용하여 전송하는 예를 나타낸 것이다. 여기서, T_m는 임의의 m번째 프레임에 대해 송신측 출발 시간을 의미하며, A_m은 수신측 도달 시간을 의미한다. 그리고, P_m은 도달한 프레임을 재생하는 시간을 의미하며, N는 스트림의 수를 의미한다. 처음 전송은 각각의 스트림을 통해 병렬적으로 전송한다. 수신측은 모든 데이터를 받은 후 재생을 수행하며, 하나의 프레임에 대한 재생이 끝난 경우 송신측은 다음 프레임을 전송하는 과정을 수행한다. 네트워크를 통해 데이터를 읽는 과정은 수신측에서 재생을 위해 데이터를 읽는 시간에 비해 많이 걸린다. 그러나 그림 6에서 보는 봐와 같이 멀티스트림을 사용할 경우 다소 지터가 발생하더라도 일정한 속도로 비디오 스트림을 재생할 수 있다.

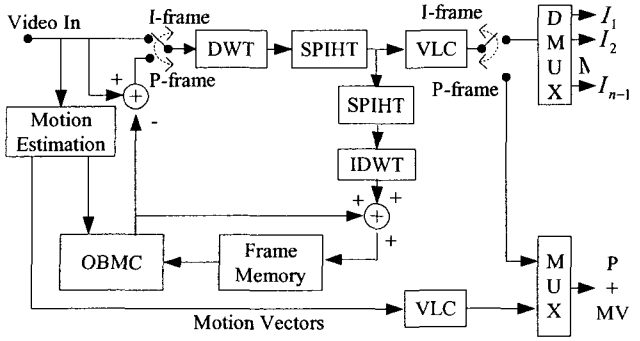


그림 7. 제안한 부호화기의 블록 다이어그램
Fig. 7. Block diagram of the proposed encoder

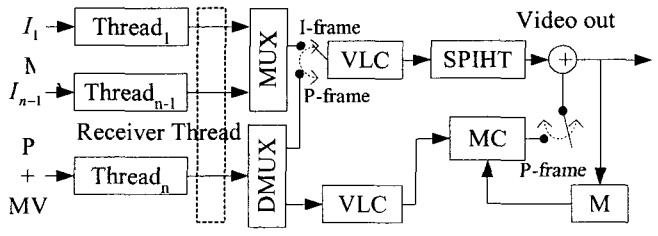


그림 8. 제안한 복호기의 블록 다이어그램
Fig. 8. Block diagram of the proposed decoder

그림 7과 그림 8은 멀티스트림을 위한 제안한 SPIHT를 이용한 비디오 부호기와 복호기의 블록 다이어그램을 나타낸 것이다. 부호화기의 기본 구조는 그림 3과 유사한 구조를 가지고 있으나, 멀티스트림 전송을 위해 비슷한 크기를 가지도록 I 픽쳐 스트림들과 P 픽쳐와 MV(Motion vector)가 결합된 스트림으로 구성하며, 복호화기는 개별적 스트림을 받아 하나의 스트림으로 복원한다.

V. 실험 및 고찰

5.1 실험 환경

본 논문에서는 SPIHT 기반 비디오 부호기의 성능평가를 위하여 표 1과 같이 5개의 비디오 스트림을 실험에 사용하였다. 비디오 스트림들은 176×144크기의 QCIF형식과 352×288 크기의 CIF형식으로, 24kbps~20Mbps의 저속 전송 매체를 위한 비디오 부호기의 실험을 위해 많이 사용되는 비디오 스트림들이다.

표 1. 실험에 사용된 비디오 스트림
Table 1. Video streams in the experiment.

스트림 형식	비디오 스트림 이름	I frame의 평균 크기(bytes)	P frame의 평균 크기(bytes)
QCIF	Akiyo	2322	89
	Miss American	1336	92
	Salesman	2791	130
	Carphone	2573	292
CIF	Foreman	8585	1103

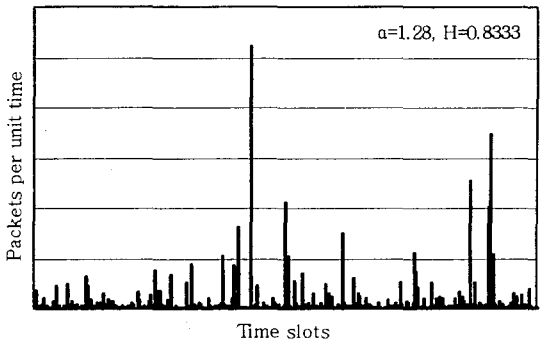


그림 9. α-stable self-similar 트래픽 모델
Fig. 9. α-stable self-similar traffic model.

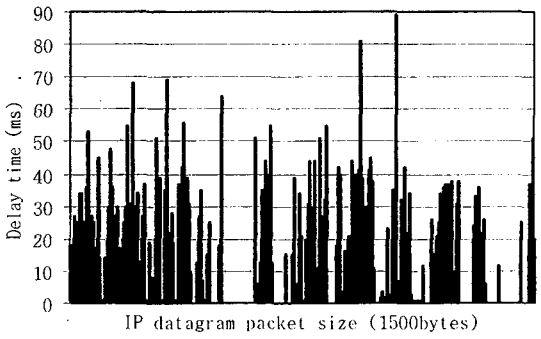


그림 10. 인터넷 트래픽 시뮬레이션을 위한 지연 모델
Fig. 10. Delay model for the simulation of Internet traffic.

실제 인터넷 환경하에서 실험은 제어하기가 어렵기 때문에 제어 가능한 실험을 위해 인터넷 트래픽 모델 시뮬레이션을 사용하였다. 기존에는 일반적으로 포아송(Poisson) 또는 마코비안(Markovian)을 기본으로 한 모델을 사용하였으나, 이 모델은 비교적 넓은 범위의 시간 스케일로 평균을 구했을 경우, 버스트한 특성이 없어진다[13]. 따라서 본 논문에서는 버스트한 트래픽 특성 시뮬레이션을 위해서 그림 9와 같이 α-stable self-similar 트래픽 모델을 사용하였다[14]. α-stable self-similar 트래픽 모델은 매우 버스터한 트래픽 특성을 표현할 수 있을 뿐만 아니라 트래픽이 가지는 self-similar 특성을 표현할 수 있다. 여기서 hurst 파라미터(H) 는 self-similar 특성을 나타내는 파라미터로서, www 트래픽 모델을 만들기 위해 0.8333으로 설정하였으며, 평균과 편차를 결정하는 α는 1.28로 설정하였다[14]. x 축은 시간 슬롯을 의미하며, y 축은 각 시간 슬롯 당 전송되는 데이터 양을 의미한다.

그림 10은 인터넷 트래픽 시뮬레이션이 그림 9와 같은 전송량을 가지도록 서버에 적용될 지연 모델을 나타낸 것이다. 즉, 단위 IP 패킷(1500bytes) 전송한 후 그림과 같이 지연 시킴으로써 전체 전송속도를 그림 9와 같은 형태가 되도록 하였으며, 실험을 위한 서버와 클라이언트 프로그램들은 windows 환경하에서 자바로 구현하였다.

5.2 실험 결과

본 논문에서는 이를 위한 척도로서 식 (1)과 같다. DR은 지터가 없는 상태에서의 기준 처리시간(t_{ref})에 대한 지터가 있는 상태에서의 현재 처리 시간(t_c)과 지터가 없는 상태에서의 처리시간(t_{ref})의 차에 대한 비로서, DR=0 이면 동기화가 되는 상태이고 DR=1이면 기준시간만큼 지연이 되어 동기화가 되고 있지 않음을 의미한다.

$$\text{DR(Delay Rate)} = \begin{cases} \dfrac{t_c - t_{ref}}{t_{ref}} & \text{if } t_c > t_{ref} \\ 0 & \text{if } t_c \le t_{ref} \end{cases} \quad (1)$$

그림 11은 앞에서 제시한 지연 모델을 이용하여 LAN 상에서 평균 전송 속도가 1000kbps, 700kbps, 400kbps, 200kbps, 100kbps, 그리고 50kbps 하에서 멀티스트림의 수에 따라 클라이언트 측의 수신 대역폭이 어떻게 변화하는지 비교한 것이다. 여기서 x축은 스트림 수를 의미하며, y축은 클라이언트에서 수신되는 평균 전송 속도(대역폭)를 의미한다. 그림에서 보는 바와 같이 하나의 스트림 평균 속도가 1000kbps나 700kbps와 같이 고속인 경우 스트림의 수가 증가하더라도 클라이언트 측의 수신 속도가 계속 증가하지 않고 일정값에 수렴함을 알 수 있다. 이것은 전송 속도는 전송로가 가진 대역폭을 초과할 수 없기 때문이다. 따라서 전송 속도가 100kbps나 50kbps와 같이 저속인 경우 전송로의 대역폭이 충분히 넓기 때문에 스트림의 수가 증가할 경우 선형적으로 증가함을 알 수가 있다

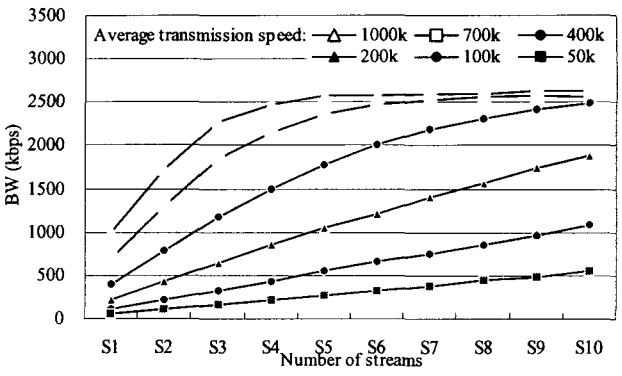


그림 11. 여러 평균 전송속도하에서 스트림 수에 따른 클라이언트의 대역폭 증가 비교

Fig. 11. Client's bandwidth according to the number of streams at several average transmission speeds

표 2는 스트림들에 대해 56kbps 비트율에서 H.263과 SPIHT 부호기로 생성한 I 프레임과 P 프레임의 평균 PSNR을 나타낸 것이다. 여기서 Y는 휘도 성분이며, C는 색차 성분을 나타낸다. 실험 결과 색차 성분에 대한 PSNR은 거의 비슷하나 휘도 성분에 대한 PSNR은 H.263 부호기를 사용한 경우 보다 SPIHT 부호기를 사용한 경우 동일한 비트율에서 약 1.5 dB 증가하였음을 알 수 있었다.

표 2. 각 스트림에 따른 PSNR 비교

Table 2. Comparison of PSNR results for each stream.

Stream name (56kbps)	I frame [dB]				P frame [dB]			
	H.263		SPIHT		H.263		SPIHT	
	Y	C	Y	C	Y	C	Y	C
Akiyo	35.05	39.26	37.45	39.31	34.96	39.34	36.74	39.79
Miss American	37.68	37.67	39.27	39.65	37.30	37.12	38.39	37.13
Salesman	32.24	38.71	33.35	39.11	31.92	38.61	32.91	38.98
Carphone	33.84	39.32	35.54	39.84	33.30	39.20	34.62	39.72
Foreman	32.92	38.65	34.60	39.65	31.72	38.32	32.67	38.41

인터넷 트래픽 모델 시뮬레이션을 이용해 평균 전송 속도가 약 150kbps, 100kbps가 되도록 설정한 후 CIF 형식의 Foreman 스트림을 제안한 멀티스트림(4개의 멀티스트림)을 이용한 방법과 기존의 단일 스트림 방법의 성능 비교를 그림 12와 그림 13에 나타내었다. 여기서, 비트율이 150kbps인 경우는 Foreman 스트림을 네트워크 지연이 없는 경우 연속적으로 재생 가능한 비트율을 의미하며, 100kbps인 경우에는 네트워크 지연이 없는 경우에도 연속 재생이 안되는 비트율을 의미한다. 4개의 멀티스트림을 이용한 제안한 방법과 기존의 단일 스트림 방법의 성능을 비교한 것이다. 단일 스트림을 이용한 방법은 전송 지연이 커짐에 따라 그 영향이 두드려지게 나타났지만, 4개의 멀티스트림을 이용한 제안한 방법은 150kbps에서는 일정한

처리 속도를 보였으며, 100kbps에서도 전송 지연에 강함을 알 수 있었다.

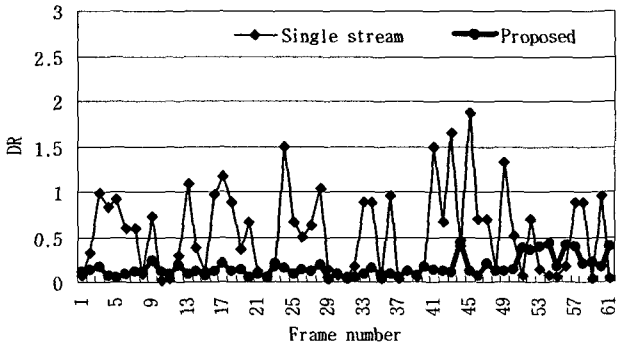


그림 12. 기존의 방법과 제안한 방법의 성능비교(150kbps)
Fig. 12. DR comparison of the proposed scheme with conventional scheme at 150kbps.

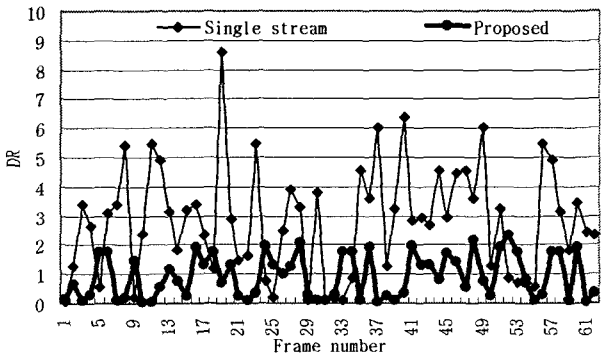


그림 13. 기존의 방법과 제안한 방법의 성능비교(100kbps)
Fig. 13. DR comparison of the proposed scheme with conventional scheme at 100kbps.

Ⅵ. 결론

본 논문에서는 현재의 인터넷 인프라 환경하에서 클라이언트가 가진 대역폭을 최대한 활용하여 최선의 서비스를 제공할 수 있을 뿐만 아니라 고압축 시에서도 블로킹 현상이 없으며 점진적 전송이 가능한 웨이브릿 기반 SPIHT를 이용한 멀티스트림 비디오 신호의 전송 기법을 제안하였다. 다양한 인터넷 트래픽 조건하에서 실험한 결과, 웨이브릿 기반 비디오 부호화기를 이용하였기 때문에 기존의 DCT 기반 비디오 부호화기보다 동일한 비트율에서 화질이 약 1.5dB가 상승하였다. 그리고, 네트워크 지연이 없는 경우 연속 재생이 가능한 비트율에서는 기존의 단일 스트림 전송인 경우 네트워크의 상태에 따라 DR의 값이 큰 값을 가지면서 심하게 변화하였으나, 제안한 멀티스트림을 이용한 전송인 경우는 DR의 값이 0에 근접하였으며, 네트워크 지연이 없는 경우에도 연속 재생이 안되는 비트율에서도 제안한 멀티스트림을 이용한 전송 방법이 기존의 단일스트림 전송방법 보다 전송 지연에 강함을 확인하였다. 따라서, 네트워크 지터에 적응적으로 대처할 수 있어 최선의 서비스를 제공할 수 있다. 향후 전송 속도에 따른 비디오 스트림에 대한 스킵과 에러 은닉에 관한 연구가 된다면 보다 효율적인 비디오 스트림 전송이 가능하리라 사료된다.

접수일자 : 2002. 5. 29 수정완료 : 2002. 7. 11

이 논문은 2001년 정보통신 연구 진흥원 대학 기초연구 지원 사업 연구 결과의 일부입니다.

참 고 문 헌

[1] Intelligent streaming of the Microsoft, available at http://msdn.microsoft.com/library/en-us/dnwmt/html/IntStreaming.asp, 2002.

[2] Surestream of the RealNetworks, availabe at http://www.realnetworks.com/products/servers/wp_surestream. html, 2002.

[3] Jan Pieper, Savitba Srinivasan, Byron Dom, "Stream-media knowledge discovery," Computer, pp. 68-74, Sep. 2001.

[4] ISO/IEC JTC1 CD 10918, "Digital compression and coding of continuous-tone still image," ISO, 1993.

[5] ISO/IEC JTC1 CD 11172, "Coding of moving pictures and associated audio for digital storage media up to 1.5Mbits/s," ISO, 1992.

[6] Draft ITU-T Recommendation H.262, "Generic coding of moving pictures and associated audio informaton: video," 1995.

[7] R. Steinmetz and K. Nahrstedt, Multimedia: Computing, Communications and Applications, Prentice Hall PTR, 1995.

[8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. Image Proccess, vol. 1, pp. 205-220, 1992.

[9] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Trans. Signal Process, vol. 8, pp. 155-167, 1993.

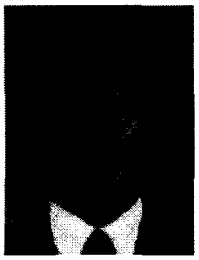
[10] A. Said and W. A. Pearlman, "A new fast/efficient image codec based on set partitioning in hierarchical tress," IEEE Trans. Video Technol. vol. 6, pp. 243-250, 1996.

[11] J. Karlekar and U.B. Desai, "SPIHT video coder," 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control, vol. 1, pp.45 -48, 1998

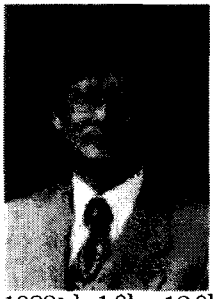
[12] Draft ITU-T Recommendation H.263, "Video coding for low bitrate communication," Dec. 1995.
[13] V. Paxon and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling", IEEE/ACM Transactions on Networking, vol. 3, no. 3, p. 226-244, Jun. 1995.
[14] J.R. Gallardo, D. Makrakis, and L. Orozco-Barbosa, "Use of Alpha-Stable Self-Similar Stochastic Processes for Modeling Traffic in Broadband Networks," Proceedings of 1998 SPIE Performance and Control Conference, pp. 218-296, Oct. 1998.

강 경 원(Kyung Won Kang),
準會員
1996년 2월 부경대학교 전자공학과 졸업(공학사)
1998년 2월 부경대학교 대학원 전자공학과 졸업(공학석사)
2002년 2월 부경대학교 대학원 전자공학과 졸업(공학박사)
1998년 3월~1999년 8월 미국 Jackson State University 연구원
2002년 3월~현재 위덕대학교 멀티미디어공학과 강의전담교수
관심분야: 멀티미디어 통신, 인터넷 방송, 영상신호처리 등

류 권 열(Gwon-Yeol Ryu)
正會員
1982년 8월 경북대학교 전자공학과 졸업(공학사)
1990년 8월 경북대학교 산업대학원 전산전공 졸업(공학석사)
1998년 2월 부경대학교 대학원 전자공학과 졸업(공학박사)
1986년 7월~1995년 4월 포항공과 대학교 전자계산소(전산과장)
1998년 4월~현재 위덕대학교 멀티미디어공학과 조교수
관심분야: 영상처리, 멀티미디어 통신



권 기 룡(Ki-Ryong Kwon)
正會員
1986년 2월 경북대학교 전자공학과 졸업(공학사)
1990년 2월 경북대학교 대학원 전자공학과 졸업(공학석사)
1994년 8월 경북대학교 대학원 전자공학과 졸업(공학박사)
1996년 3월~현재 부산외국어대학교 전자·컴퓨터공학과 조교수
관심분야: 웨이브릿변환, 적응신호처리, 능동잡음제거, 멀티미디어 통신등



문 광 석(Kwang Seok Moon),
正會員
1979년 2월 경북대학교 전자공학과 졸업(공학사)
1981년 2월 경북대학교 대학원 전자공학과 졸업(공학석사)
1989년 2월 경북대학교 대학원 전자공학과 졸업(공학박사)
1988년 1월~12월 일본 동경대학교 공학부 연구원
1997년 8월~1998년 7월 미국 Jackson State University 객원교수
1990년 3월~현재 부경대학교 전자컴퓨터정보통신공학부 교수
관심분야: 영상신호처리, 적응신호처리등

김 문 수(Moon Soo Kim)
正會員
1963년 2월 전남대학교 공과대학 졸업(공학사)
1981년 2월 동아대학교 대학원 전자공학과 졸업(공학석사)
1987년 2월 동아대학교 대학원 전자공학과 졸업(공학박사)
현재 부경대학교 전자컴퓨터정보통신공학부 교수
관심분야 : 신호처리, 회로 및 통신 시스템 등