

SPIHT 기반 비디오 신호의 적응적 멀티스트림 전송기법

강경원, 정대일*, 류권열, 권기룡**, 문광석***
위덕대학교 컴퓨터멀티미디어공학부, *동명정보대학교 정보공학부,
부산외국어대학교 전자공학과, *부경대학교 전자컴퓨터정보통신공학부

Adaptive Multi-stream Transmission Technique Based on SPIHT Video Signal

Kyung-Won Kang, Tae-Il Jeong, Gwon-Yeol Ryu, Ki-Ryong Kwon, Kwang-Seok Moon
Division of Computer and Multimedia Engineering, Uiduk University,
*Division of Information Engineering, Tongmyong University of Information Technology,
**Department of Electronic Engineering, Pusan University of Foreign Studies,
***Division of Electronic, Computer and Telecommunication Engineering, PuKyong Nat'l University
E-mail : kwkang@mail.uiduk.ac.kr, tijeong@hanmail.net, kyryu@mail.uiduk.ac.kr,
krkwon@taejo.pufs.ac.kr, moonks@pknu.ac.kr

요약

본 논문에서는 현재의 인터넷 환경 하에서 최선의 서비스를 제공하기 위해 SPIHT 비디오 데이터에 대한 적응적 멀티스트림 전송 기법을 제안한다. 제안한 방법은 QoS가 보장되지 않는 비동기적인 패킷 망에서 클라이언트의 수신 버퍼 상태를 이용한 네트워크 상태에 따른 정보 전송량 조절과 TCP 상에서 멀티스트림을 이용한 병렬 전송을 수행함으로써 신뢰성 있는 전송과 네트워크의 지터에도 강하며, 클라이언트가 가진 대역폭을 최대한 활용할 수 있는 최선의 서비스를 제공한다.

1. 서론

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. 지금까지 공간, 주파수, 시간 영역 상관성을 이용한 정보 압축 기법들이 제안되어 왔다. 특히 DCT로 대표되는 블록변환에 근거한 기법들은 JPEG, MPEG-1, MPEG-2 등 각종 영상 부호화의 국제 표준 방식에서 활용되고 있다[1-3]. 그러나 DCT를 근간으로 하는 기존의 표준화된 부호화 기법들은 고압축 시 블록킹 현상(blocking effect)이 발생하여 화질 열화를 초래할 뿐만 아니라, 프레임간의 종속성이 존재하여 패킷 손실이 발생할 경우 에러가 전파된다[4]. TCP는 연결 지향적 구조를 가지고 있기

때문에 신뢰성 있는 전송은 가능하나, 지터가 발생 할 경우 클라이언트가 그 영향을 그대로 받아 전송 성능이 떨어지는 단점이 있다. 따라서, 본 논문에서는 현재의 인터넷 인프라 환경 하에서 클라이언트가 가진 대역폭을 최대한 활용하여 최선의 서비스를 제공할 수 있는 SPIHT 비디오 데이터의 적응적 멀티스트림 전송 기법을 제안한다. 제안한 방법은 현재의 비동기적인 패킷망 상에서 최선의 서비스를 위해 클라이언트의 수신 버퍼 상태에 따라 네트워크 상태를 추정하여 전송할 스트림의 비트율을 설정한다. 그리고 TCP의 전송 성능 향상을 위해 멀티스트림을 통한 병렬 전송함으로써 네트워크 트래픽의 증가에 의해 발생하는 전송 지연에 덜 민감할 뿐만 아니라, 클라이언트 측의 대역폭을 최대한 이용할 수 있는 최선의 서비스를 제공한다.

2. SPIHT 기반 비디오 부호화

현재 비디오 압축 부호화는 기본 알고리즘으로 DCT를 사용하고 있다. 그러나 이들 부호화들은 고압축을 수행할 경우 블록킹 현상이 심하게 나타나는 단점이 있다. 이러한 단점을 극복할 뿐만 아니라 점진적 전송 특성을 얻기 위해 웨이브릿 변환 영역에서 웨이브릿 영역에서 SPIHT를 이용한 비디오 부호화 방법이 제안되었다[5]. SPIHT는 EZW 부호화 방법을 개선하여 압축 효율을 향상시킨 방법으로[6,7], 그림 1과 같이 SPIHT 웨이브릿 비디오 부호화기를 구성한다. I-프레임은 독립적으로 복원이 가능한 프레임으로서, SPIHT를 이용하여 정지영상부호화 방법과 같은 방법으로 부호화를 수행하며, P-프레임은 I 또는 다른 P 프레임 기준으로 삼아 움직임 예측하여 움직임 벡터(motion vector)를 생성한 후, 움직임 벡터를 이용한 보상 과정 후 생성된 나머지 차이 분만을 SPIHT를 이용하여 부호화한다. 움직임 추정은 16×16 블록에 대해 반화소 단위로 수행하며, 움직임 보상은 블록화 현상을 줄이기 위해 H.263의 중첩 움직임 보상 기법(OBMC)을 사용한다[8].

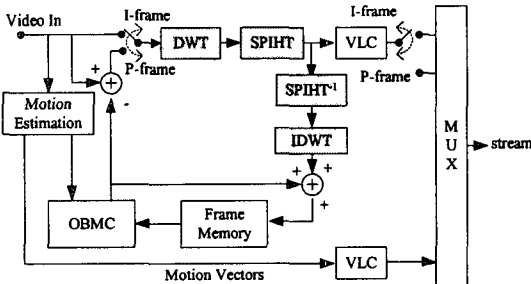


그림 1. SPIHT 웨이브릿 비디오부호화기의 블록 다이어그램

3. 제안한 적응적 멀티스트림 전송

현재의 비동기적인 패킷 망에서 통신망의 효율적으로 사용하기 위해 통신망의 대역폭에 따라 전송 가능한 크기의 스트림을 생성하여 전송할 경우, 통신 자원의 낭비를 방지하여 통신망의 효율을 높일 수 있다. 또한 생성된 스트림을 멀티스트림으로 전송할 경우 각각의 스트림마다 개별적인 MTU(maximum transmission unit) 단위로 데이터가 전송되기 때문에 총 패킷의 크기가 전송 대역폭보다 작을 경우 모든 MTU를 병렬적으로 전송할 수 있어 전송 대역폭을 증가되어 클라이언트가 가진 대역폭을 최대한 활용할

수 있을 뿐만 아니라, 전송 시 하나의 스트림에 지터가 발생시 다른 스트림을 통해 계속 전송할 수 있기 때문에 클라이언트의 버퍼링을 통한 네트워크 지터를 극복할 수 있다[9].

따라서, 본 논문에서는 현재의 네트워크상에서 고압축 시 발생하는 블록킹 현상과 에러 전파가 없는 최선의 서비스를 제공하기 위해 네트워크의 대역폭에 따른 SPIHT 비디오 데이터의 프레임 패턴의 적응적 설정과 SPIHT의 임베디드 특성을 이용한 멀티스트림 생성을 통해 효율적인 전송 기법을 제안한다. 그림 2는 적응적 멀티스트림을 구현하기 위하여 네트워크의 상태에 따라 프레임 패턴을 적응적으로 설정하는 과정을 도식한 것이다. 비디오 스트림은 그림 2와 같이 I 프레임과 P 프레임으로만 구성된 2개의 스트림으로 구성한다. 적응적 프레임 패턴 제어기는 클라이언트의 버퍼링 상태를 검사하여 클라이언트 측의 수신 버퍼 크기(B)의 충만도에 따라 전송할 프레임 패턴을 적응적으로 변경한다. 즉, 클라이언트 측의 수신 버퍼에 버퍼링된 스트림의 상태가 $0.75B$ 보다 큰 경우에는 그림 2와 같이 화질을 중시한 많은 정보량을 전송하며, 버퍼링된 상태가 $0.75B$ 보다 작은 경우에는 기존의 I 프레임 수를 감소시켜 단위 시간당 전송되어야 하는 정보량을 줄여 네트워크를 통해 전송한다. 버퍼링된 상태가 $0.25B$ 보다 더 작은 경우에는 네트워크의 상태가 나쁜 상태이므로, 단위 시간당 전송하는 정보량을 최소화하여 전송함으로써, 연속적인 재생을 가능하게 한다. 제안한 방법은 서버 측에서 기억공간의 낭비를 초래할 수 있으나, 다양한 프레임 패턴을 생성시킬 수 있어 통신망의 상황에 따라 다양한 패턴으로 전송할 수 있어 통신망의 효율을 높일 수 있다.

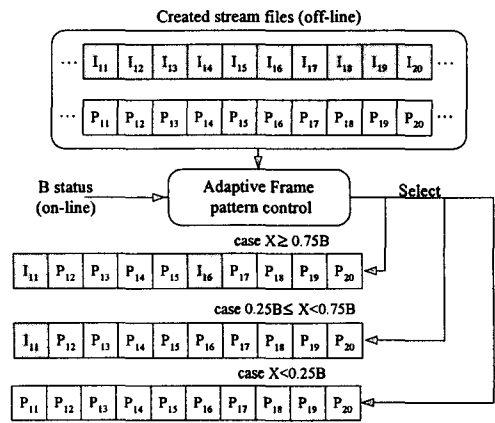


그림 2. 적응적 프레임 패턴 제어기를 이용한 스케일러빌리티

그림 3은 제안한 서버 구조의 블록도로 나타낸 것이다. 서버는 부호화 부분과 전송 부분으로 구성된다. 전송 부분은 적응적 프레임 패턴 제어기와 멀티스트림 생성기 그리고 이 두 부분들을 제어하는 제어 부분으로 구성된다. 적응적 프레임 패턴 제어기는 네트워크 상태에 따라 전송할 프레임 패턴을 결정하는 곳으로 스케일러블한 전송이 가능하게 하며, 멀티스트림 생성기를 통해 효율적으로 전송이 가능하다.

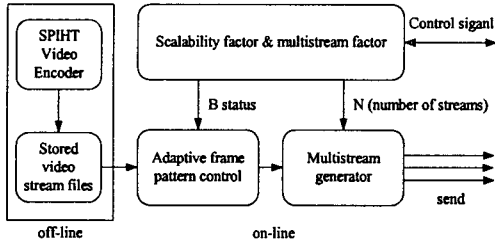


그림 3. 제안한 서버 구조의 블록도

4. 실험 및 고찰

본 논문에서는 SPIHT 비디오 부호기의 성능평가를 위하여 표 1에서와 같이 24kbps~20Mbps의 저속 전송 매체를 위한 비디오 부호기의 실험을 위해 많이 사용되는 비디오 스트림들을 실험에 사용하였다.

표 1. 실험에 사용된 비디오 스트림

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

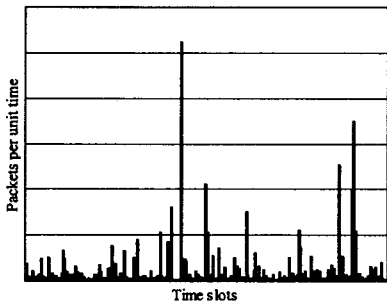


그림 4. α-stable self-similar 트래픽 모델(α=1.28, H=0.8333)

인터넷 환경 하에서 실험은 제어하기가 어렵기 때문에 제어 가능한 실험을 위하여 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[10]. 따라서 본 논문에서는 버스트한 트래픽 특성을 시뮬레이션하기 위해서 그림 4와 같이 α-stable self-similar 트래픽 모델을 사용하였다[11]. α-stable self-similar 트래픽 모델은 매우 버스트한 트래픽 특성을 표현할 수 있을 뿐만 아니라 트래픽이 가지는 self-similar 특성을 표현할 수 있다. 여기서 hurst 파라미터(H)는 self-similar 특성을 나타내는 파라미터로써, 0.8333로 설정하였고, 평균과 편차를 결정하는 알파는 1.28로 두었다. x 축은 시간 슬롯을 의미하며, y 축은 각 시간 슬롯 당 전송되는 데이터 양을 의미한다.

실험을 위한 서버와 클라이언트 프로그램들은 windows 환경 하에서 자바로 구현하였다. 수신측에서 재구성된 멀티미디어 데이터의 표현 서비스 품질은 표현이 평활하게 나타나야 한다. 그리고 일정한 재생 시간 간격을 유지하여 연속성을 보장해야 한다. 따라서 본 논문에서는 이를 위한 척도로서 지터가 없는 상태에서의 기준 처리 시간(t_{ref})에 대한 지터가 있는 상태에서의 현재 처리 시간(t_c)과 지터가 없는 상태에서의 기준 처리시간(t_{ref})의 차의 비를 사용하였으며, 식 (1)과 같다.

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

DR은 지터가 없는 상태에서의 기준 처리시간에 대한 지터가 있는 상태에서의 현재 처리 시간과 지터가 없는 상태에서의 처리시간의 차에 대한 비로써, DR=0이면 동기화가 되는 상태이고 DR=1이면 기준시간만큼 지연이 되어 동기화가 되고 있지 않음을 의미한다.

그림 5는 앞에서 제시 한 지연 모델을 이용하여 LAN상에서 평균 전송 속도가 1000kbps, 700kbps, 400kbps, 200kbps, 100kbps, 그리고 50kbps 하에서 멀티스트림의 수에 따라 클라이언트 측의 수신 대역폭이 어떻게 변화하는지 비교한 것이다. 여기서 x 축은 스트림 수를 의미하며, y 축은 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 일정 값에 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 증가 할 경우 선형적으로 증가한다. 즉, 평균 전송 속도가 낮을수록 멀티스트림의 효과가 극대화됨을 알 수가 있다.

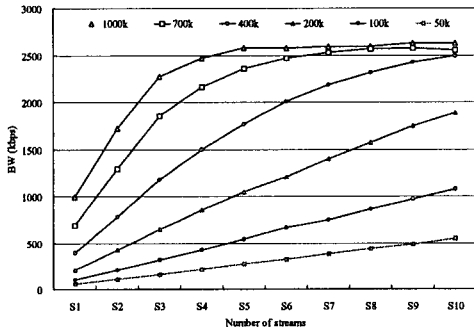


그림 5. 스트림 수에 따른 클라이언트의 대역폭 증가 비교

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. 비디오 스트림에 따른 PSNR 비교

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
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 모델에 의해 평균 전송 속도가 약 150kbps, 100kbps가 되도록 설정한 후 CIF 형식의 Foreman 스트림을 제안한 적응적 멀티스트림을 이용한 방법과 4개의 스트림을 이용한 방법, 그리고 기존의 단일 스트림 방법의 성능 비교를 그림 6과 그림 7에 나타내었다. 여기서, 비트율이 150kbps인 경우는 Foreman 스트림을 네트워크 지연이 없는 경우 연속적으로 재생 가능한 비트율을 의미하며, 100kbps인 경우에는 네트워크 지연이 없

는 경우에도 연속 재생이 안 되는 비트율을 의미한다.

그림 6과 같이 평균속도가 150kbps인 연속 재생이 가능한 상태에서 전송할 경우 지터의 영향을 받아 단일 스트림인 경우 DR의 값이 큰 값으로 많이 변화하여 연속 재생이 힘든 상태를 의미하나, 4개의 멀티스트림을 이용한 경우나 제안한 적응적 멀티스트림을 이용한 경우 DR의 값이 거의 0에 근접하는 값을 나타내어 지터의 영향을 별로 받지 않으면서 연속 재생이 가능함을 의미한다.

그림 7에서는 평균속도를 100kbps로 줄였을 때, 재생과정을 도식한 것이다. 단일 스트림인 경우 지터의 영향을 많이 받아 DR의 값이 아주 심하게 변화하는 것을 볼 수 있었으며, 4개의 멀티스트림을 사용한 경우에도 지터의 영향을 받아 연속 재생이 부자연스러운 정도의 DR 값을 나타내었다. 그러나 제안한 방법인 경우 초기에는 4개의 멀티스트림과 같이 지터의 영향을 받지만, 버퍼링이 되는 양을 검사하여 송신측에서 적응적 프레임 패턴을 제어하여 전송되는 데이터 량을 100kbps이하로 줄임으로써 연속 재생이 가능하게 하였다. 적응적 프레임 패턴 제어는 의해 I 프레임 대신 P 프레임들이 전송함으로 약간의 화질 저하를 야기시킬 수 있으나 효율적인 전송을 가능하다.

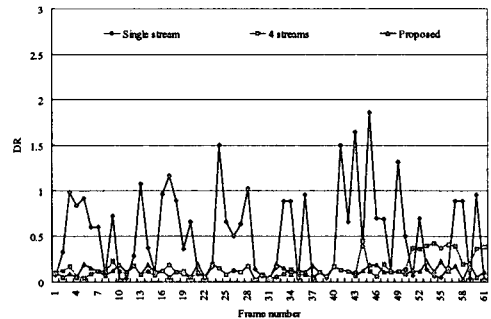


그림 6. 기존의 방법과 제안한 방법의 성능 비교(150kbps)

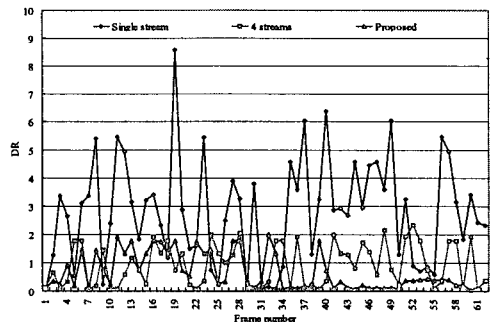


그림 7. 기존의 방법과 제안한 방법의 성능 비교(100kbps)

6. 결론

본 논문에서는 QoS가 보장되지 않는 현재의 비동기적인 패킷 망에서 SPIHT 비디오 데이터의 적응적 멀티스트림 전송기법을 제안하였다. 제안한 방법은 고압축에서도 블록킹 현상이 없으며 에러 전파 없는 신뢰성 있는 전송과 연속 재생을 위한 최선의 서비스를 위해 SPIHT 비디오 부호화 기반 비디오 스트림을 생성하여 적응적 프레임 패턴 제어기와 TCP 상에서 멀티스트림을 이용해 전송하였다. 적응적 프레임 패턴 제어기는 클라이언트의 수신 버퍼상태를 검사하여 네트워크의 상태를 추정한 후 전송할 정보량을 조절하는 것으로 스케일러블한 전송을 가능하게 하였으며, 멀티스트림 전송은 TCP상에서 전송성능을 향상시킬 수 있을 뿐만 아니라, 네트워크의 지터에도 강하며, 클라이언트가 가진 대역폭을 최대한 활용할 수 있었다. 실험한 결과, SPIHT 비디오 부호기를 이용한 적응적 멀티스트림 전송을 기법인 경우 기존의 DCT 기반인 H.263 비디오 부호기를 이용한 스트림 전송 기법보다 동일한 비트율에서 화질이 약 1.5dB 향상되었을 뿐만 아니라, 기존의 단일 스트림 전송인 경우 네트워크의 상태에 따라 DR의 값이 큰 값을 가지면서 심하게 변화하였으나, 제안한 적응적 멀티스트림을 이용한 전송인 경우에는 DR의 값이 0에 근접함을 확인할 수 있었다. 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.

IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control, vol. 1, pp. 45-48, 1998.

- [6] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-3462, December 1993.
- [7] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.
- [8] Draft ITU-T Recommendation H.263, "Video coding for low bitrate communication," Dec. 1995.
- [9] 강경원, 문광석, "멀티스트림을 이용한 비디오 스트림의 평활화," *신호처리·시스템 학회 논문지*, 제 3권, 1호, pp.21-26, 2002.
- [10] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling", *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, p. 226-244, June 1995.
- [11] 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, October 1998.

[참고문헌]

- [1] ISO/IEC JTC1 CD 10918, "Digital compression and coding of continuous-tone still image," ISO, 1993.
- [2] ISO/IEC JTC1 CD 11172, "Coding of moving pictures and associated audio for digital storage media up to 1.5Mbits/s," ISO, 1992.
- [3] Draft ITU-T Recommendation H.262, "Generic coding of moving pictures and associated audio information: video," 1995.
- [4] Jan Ozer, "Audio, Video Live from the web," *PC Magazine*, pp. 100-136, March, 1996.
- [5] J. Karlekar and U. Desai, "SPIHT video coder,"