비디오 데이터의 효율적 전송을 위한 적응적 멀티스트림 기반 오류 정정 부호화 기법

강경원, 이경환, 류권열, 권기룡*, 문광석**
위덕대학교 컴퓨터멀티미디어공학부, *부산외국어대학교 디지털정보공학부,
**부경대학교 전자컴퓨터정보통신공학부

Error Correcting Coding Based on Adaptive Multi-stream Technique for Effective Transmission of the Video Data

Kyung-Won Kang, Kyeong-Hwan Lee, Gwon-Yeol Ryu, *Ki-Ryong Kwon, **Kwang-Seok Moon
Division of Computer and Multimedia Engineering, Uiduk University,
*Division of Computer and Multimedia Engineering, Pusan University of Foreign Studies,
**Division of Electronic, Computer and Telecommunication Engineering, PuKyong Nat'l University
E-mail : kwkang@mail.uiduk.ac.kr, kyryu@mail.uiduk.ac.kr, moonks@pknu.ac.kr

요 약

최근, 인터넷을 통한 VOD와 같은 멀티미디어 서비스의 증대로 대용량 파일 서비스의 효율적인 저장 시스템과 전송 기법들이 필요하다. 본 논문에서는 현재의 인터넷 환경 하에서 VOD 서비스와 같은 대용량 파일 서비스를 위한 효율적인 부호화 기법을 제안한다. 제안한 방법은 QoS가 보장되지 않는 비동기 패킷 망에서 클라이언트 측의 네트워크 상태에 따라 정보 전송량을 조절함과 동시에 다수의 비디오 스트림을 이용한 병렬 전송과 체크 스트림을 이용한 에러 정정을 통해 버퍼링시 미전송된 블록을 복원함으로서 신뢰성 있는 전송과 네트워크의 지터에도 강하며, 클라이언트가 가진 대역폭을 최대한 활용 할 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 저장 시스템과 전송 기법들이 필수적이다.

현재 영상 부호화의 국제 표준은 DCT를 기반으로 하는 JPEG, MPEG-1, MPEG-2 등이 있다[1-3]. 그러나 DCT를 근간으로 하는 기존의 표준화된 부호화 기법들은 고압축시 블록킹 현상(blocking effect)이 발생하여 화질 열화를 초래할 뿐만 아니라, 프레임간의 종속성이 존재하여 전송시 패킷 손실이 발생할 경우 에러가 전파된다[4]. 인터넷에서 이용되는 TCP 전송은 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 고압축시 블록킹 현상이 없으며, 클라이언트가 가진 대역폭을 최대한 활용하여 최선의 서비스를 제공할 수 있는 SPIHT 비디오 데이터에 대한 적응적 멀티스트림 기반 오류 정정 부호화 기법을 제안한다.

제안한 방법은 현재의 비동기적인 패킷망 상에서 최선의 서비스를 위해 클라이언트의 수신 버퍼 상태를 참조하여 네트워크 상태를 추정한 후 전송할 스트림의 비트율을 조정한다[5]. 그리고 TCP의 전송 성능 향상을 위해 멀티스트림을 통한 병렬 전송과 더불어 에러 정정을 위한 하나의 체크 스트림을 추가하여 전송한다. 체크 스트림은 버퍼링시 임의의 하나의 스트림에서 미수신된 블록이 발생할 경우 체크 블록을 이용한 오류 정정을 수행하여 미수신된 원래의 블록을 복원함으로써 네트워크 트래픽의 증가에 의해 발생되는 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[6]. SPIHT는 EZW 부호화 방법을 개선하여 압축 효율을 향상시킨 방법으로[7,8], 그림 1과 같이 SPIHT 웨이브릿 비디오 부호화기를 구성한다.

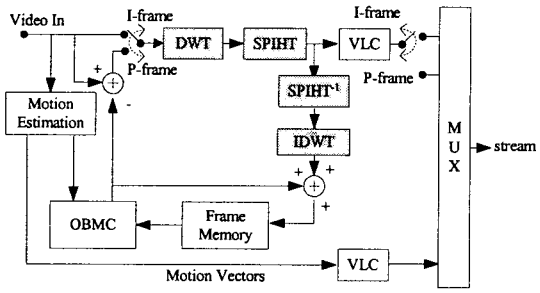

그림 1. SPIHT 웨이브릿 비디오부호화기의 블록 다이어그램

3. 제안한 멀티스트림 기반 오류 정정 부호화

현재의 비동기적인 패킷 망에서 통신망의 대역폭에 따라 전송 가능한 크기의 블록을 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 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[5].

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 설정한 후 임베디드 특성을 이용한 동일한 크기의 블록으로 구성된 다수의 스트림들로 스트립핑(stripping) 한 후 병렬 전송한다.

그림 2는 적응적 멀티스트림을 구현하기 위하여 네트워크의 상태에 따라 프레임 패턴을 적응적으로 설정하는 과정을 도식한 것이다. 비디오 스트림은 그림 2와 같이 I 프레임과 P 프로임으로만 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 초래 할 수 있으나, 다양한 프레임 패턴을 생성시킬 수 있어 통신망의 상황에 따라 다양한 패턴으로 전송할 수 있어 통신망의 효율을 높일 수 있다.

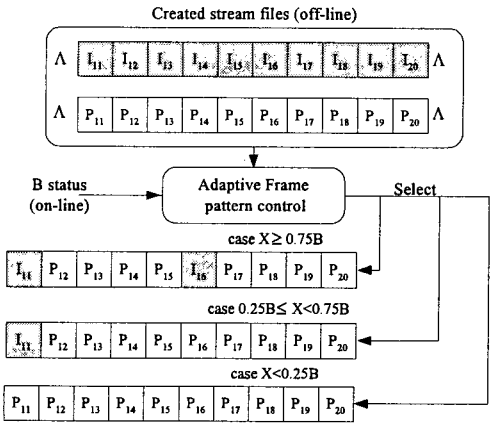

그림 2. 적응적 프레임 패턴 제어기를 이용한 스케일러빌리티

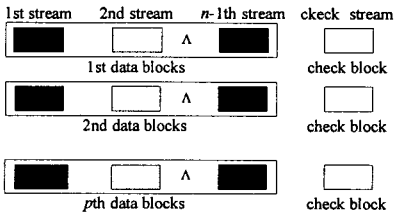

그림 3. 체크 블록 스트림

생성된 스트림은 그림 3과 같이 전송을 위해 생성된 일정한 크기의 블록에 패리티 체크를 위한 체크

블록을 첨가한다. 체크 블록은 클라이언트 측에서 버퍼링시 시간내에 수신되지 못한 블록이 있을 경우 미수신된 블록의 복원을 위해 사용된다. 미수신된 블록이 있는 스트림은 재접속을 통해 일정 일정 블록 이후의 블록부터 재전송을 통해 지속적인 전송과 클라이언트의 대역폭을 최대한 이용할 수 있다.

4. 실험 및 고찰

본 논문에서는 SPIHT 기반 비디오 부호기의 성능평가를 위하여 표 1과 같이 4개의 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

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

인터넷 환경 하에서 실험을 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 특성(burst properties)이 없어진다[9]. 따라서 본 논문에서는 트래픽 특성을 시뮬레이션하기 위해 α-stable self similar 트래픽 모델을 사용하였다[10]. α-stable self similar 트래픽 모델은 매우 버스트한 트래픽 특성뿐만 아니라 트래픽이 가지는 self similar 특성도 표현할 수 있다. 여기서 hurst 파라미터(H) 는 self similar 특성을 나타내는 파라미터로써, WWW 트래픽 모델을 만들기 위해 0.8333으로 설정하였으며, 평균과 편차를 결정하는 α는 1.28로 설정하였다[10].

수신측에서 재구성된 멀티미디어 데이터의 표현 서비스 품질은 그 표현이 연속적으로 나타나야 한다. 그리고 일정한 재생 시간 간격을 유지하여 연속성을 보장해야 한다. 따라서 본 논문에서는 이를 위한 척도로서 지터가 없는 상태에서 기준 처리 시간(t_{ref})에 대하여 지터가 있는 상태에서의 현재 처리 시간(t_c)과 지터가 없는 상태에서의 기준 처리시간(t_{ref})과의 차성분에 대한 비를 사용하였으며, 그 식은 (1)과 같다.

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

즉, DR은 지터가 없는 상태에서의 기준 처리시간에 대한 지터가 있는 상태에서의 현재 처리 시간과 지터가 없는 상태에서의 처리시간의 차에 대한 비로써, DR=0 이면 연속 재생이 가능한 상태이고 DR=1이면 기준 시간만큼 지연이 되어 연속 재생이 되고 있지 않음을 의미한다.

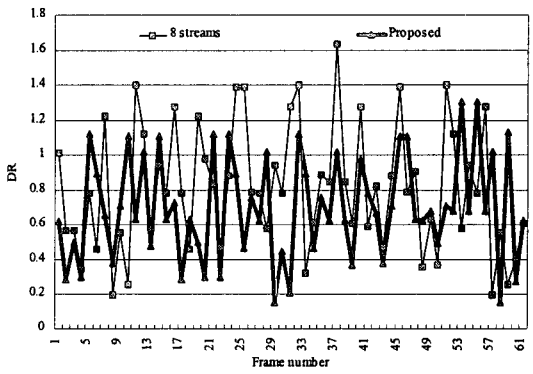

그림 4. 기존의 방법과 제안한 방법의 성능 비교 (150kbps)

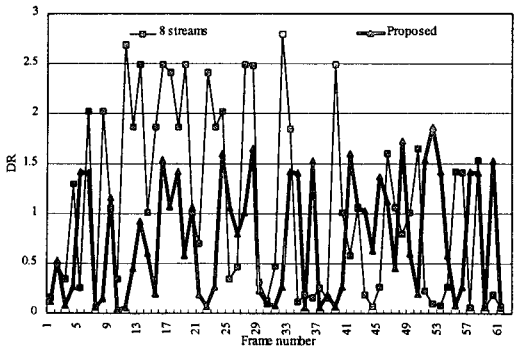

그림 5. 기존의 방법과 제안한 방법의 성능 비교 (100kbps)

인터넷 트래픽 모델 시뮬레이션에 의해 평균 전송 속도가 약 150kbps, 100kbps가 되도록 설정한 후 CIF 형식의 Foreman 스트림을 제안한 오류 정정 스트림을 가지는 8개의 멀티스트림을 이용한 방법과 오류 정정 스트림이 없는 8개의 적응적 멀티스트림을 이용한 방법에 대한 성능 비교의 결과를 그림 4와 그림 5에 나타내었다. 그림 4의 비트율이 150kbps인 경우는 Foreman 스트림에 있어서 네

트워크 지연이 없는 경우 연속적으로 재생 가능한 비트율을 나타내며, 그림 5의 100kbps인 경우는 네트워크 지연이 없는 경우에도 연속 재생이 되지 않는 비트율을 의미한다.

그림 4와 같이 평균속도가 150kbps인 연속 재생이 가능한 상태에서 전송할 경우 두 경우 모두 DR의 값이 거의 0에 근접하는 값을 나타내어 지터의 영향을 별로 받지 않으면서 연속 재생이 가능함을 알 수 있다. 그림 5는 평균속도를 100kbps로 줄였을 때, 재생과정을 도식화한 것이다. 오류 정정 스트림이 없는 단순한 8개의 멀티스트림을 사용한 경우에도 초기에는 지터의 영향을 받아 연속 재생이 부적합할 정도의 DR을 나타냈었으나 50 프레임부터 클라이언트의 버퍼링 상태에 따라 송신측에서 적응적 프레임 패턴을 제어하여 전송되는 데이터 양을 100kbps이하로 줄임으로써 연속 재생을 수행함을 보여준다. 그러나 제안한 방법은 초기 상태부터 버퍼링시 도달하지 못한 블록들은 지연 없이 체크 블럭을 통해 복원과 적응적 스트림 제어를 통해 연속 재생이 가능함을 보여준다.

5. 결론

본 논문에서는 현재의 비동기적인 패킷망에서 SPIHT 비디오 데이터에 대한 적응적 멀티스트림 기반 오류 정정 부호화 기법을 제안하였다. 제안한 방법은 QoS가 보장되지 않는 비동기적인 패킷망에서 비디오 스트림의 신뢰성 있는 전송과 연속적으로 재생하기 위한 최선의 서비스를 위해 TCP 상에서 적응적 프레임 패턴 제어기에 의한 스케일러블 프레임 패턴을 생성하여 멀티스트림을 이용해 전송하였다. 적응적 프레임 패턴 제어기는 네트워크 상태에 따라 클라이언트의 버퍼가 채워지는 상태가 따르기 때문에 이 상태 정보를 참조하여 전송할 정보량이 스케일러빌러티를 가지도록 하였으며, 멀티스트림 전송을 통해 TCP상에서 전송 성능을 향상시킬 수 있을 뿐만 아니라, 네트워크의 지터에도 강하며, 클라이언트가 가진 대역폭을 최대한 활용할 수 있는 제어성능을 가지고 있었다. 또한 비디오 스트림이 정해진 버퍼링 시간내에 완전하게 수신되지 않더라도 체크 스트림을 이용하여 복원한 후 재생이 가능하였다. 실험결과, 제안한 체크 스트림을 포함된 적응적 멀티스트림을 이용한 전송인 경우 체크 스트림이 포함되지 않은 멀티스트림 보다 네트워크 지터에 적응적으로 대처할 수 있어 최선의 서비스를 제공할 수 있었다. 향후 UDP 상에서 오류 정정과 오류 은닉에 관한 연구가 계속된다면 보다 효율적인 비디오 스트림 전송이 가능할 것으로 사료된다.

[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.

[5] 강경원, 정태일, 류권열, 문광석, "SPIHT 기반 비디오 신호의 적응적 멀티스트림 전송기법," 멀티미디어학회 논문지, 제5권, 제6호, pp.697-703, December 2002.

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

[7] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-3462, December 1993.

[8] 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.

[9] 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, June 1995.

[10] 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.