

멀티스트림을 이용한 비디오 스트림의 평활화

Video Stream Smoothing Using Multistreams

강 경 원, 문 광 석

Kyung-Won Kang, Kwang-Seok Moon

요 약

비디오 스트림들은 사용된 압축 알고리즘의 구조와 화면의 복잡도 등에 따라 다양한 형태의 트래픽이 발생하므로, 송신측과 수신측 사이의 자원할당을 어렵게 할 뿐만 아니라, 현재의 인터넷과 같은 패킷 통신망에서는 연속적인 계층을 어렵게 한다. 따라서, 본 논문에서는 멀티스트림을 이용한 비디오 스트림의 평활화 방법을 제안한다. 제안한 방법은 스트림의 형태에 따라 LDU(logical data unit)를 정의한 후 일정한 크기로 다수의 스트림으로 생성하여 전송함으로써, 평활화와 신반입 과정에서 발생하는 버퍼링 시간을 줄일수 있을 뿐만 아니라 네트워크의 지터에도 강하면, 클라이언트의 대역폭을 최대한 활용할 수 있는 효율적인 전송 특성을 얻을 수 있다.

Abstract

Video stream invoke a variety of traffic with the structure of compression algorithm and image complexity. Thus, it is difficult to allocate the resource on the both sides of sender and receiver, and playout on the Internet such as a packet switched network. Thus, in this paper we proposed video stream smoothing using multistream for the effective transmission of video stream. This method specifies the type of LDU(logical data unit) according to the type of original stream, and then makes a large number of streams as a fixed size, and transfers them. So, the proposed method can reduce the buffering time which occurs during the process of the smoothing and prefetch be robust to the jitter on network, as well. Consequently, it has the effective transmission characteristics of fully utilizing the clients' bandwidth.

Key words : video stream, internet, multistreams, smoothing

I. 서 론

미국 국방성의 ARPANET(the advanced research projects agency network)에서 시작한 인터넷(Internet)은 컴퓨터, 저장 장치, 통신, 압축 기술 등의 발전과 WWW(world wide web)의 대중화에 힘입어 불과 30년이라는 짧은 기간 안에 양적, 질적으로 팽창하였으며, 사용자들도 인터넷을 통해 문자나 숫자 위주에서 그래픽, 영상, 음성, 오디오 및 비디오와 같은 멀티미디어 데이터에 대한 욕구가 증가하고 있다[1]. 이미 주문형 비디오 시스템(VOD: video on demand system)이나 전자 도서관(digital library), 화상 회의(video conference)와 같은 멀티 미디어 응용들이 실생활에서 사용 중이거나 개발 중에 있다.

영상 데이터는 양이 방대하므로 고압축이 필수적이다. 그러나 비디오 알고리즘은 그 목적에 따라 고정 비트율과 가변 비트율의 두 가지 비트 스트림을 생성할 수 있다. 고정 비트율 방식으로 생성된 비디오 스트림은 전송, 재생 시 단위 시간에 처리해야 하는 데이터량이 일정하기 때문에 시스템 자원을 효율적으로 사용할 수 있는 장점이 있는 반면 압축율이 떨어지고 화질이 가변적인 단점이 있다. 가변 비트율 방식은 화질을 일정하게 유지하면서 압축율을 높이기 위해 가변적인 비트 스트림을 생성한다. 그러나, 네트워크의 입장에서 보면 유입되는 트래픽의 비트율 변화를 감소시킴으로써 보다 큰 통계적 다중화 이득을 얻을 수 있다. 대표적인 가변 비트율 비디오 스트림의 비트율 평활화 방법에는 최대 비트율 제어

(peak bit rate control) [2]. 압축되는 모드에 따라 서로 성격을 달리하는 프레임들을 분리해서 처리하는 방법(frame decomposition) [3], 재생할 때 중요도에 따라 프레임들을 분리하는 방법(frame prioritization) [4], 데이터량이 상대적으로 많은 프레임들을 한번 읽어 보내는 것이 아니라 분산시켜 여러 번에 걸쳐 처리하는 방법(prefetched/delayed smoothing) [5-7] 등이 있다.

프레임 모드 분리 방법은 인코딩 모드에 따라 각각 성격을 달리하는 프레임(I, P, B)들을 분리해서 처리하는 방법으로, 각 모드 별로 분리된 프레임 크기의 자기상관 계수 (autocorrelation coefficient)가 스트림 전체 프레임에 대한 것보다 훨씬 높게 유지되기 때문에 유용한 방법이다. 그러나 클라이언트 시스템에서는 분리되어 들어오는 각 프레임간의 동기화를 시켜야 한다. 선반입 평활화 방법은 간단하면서도 효과적으로 가변 비트율 비디오 스트림을 평활화 할 수 있으나 초기 버퍼링에 의한 지연이 필요하다.

본 논문에서는 효율적인 평활화를 위해 멀티스트림을 이용한 비디오 스트림 평활화 방법을 제안한다. 제안한 방법은 원래의 스트림을 분석하여 LDU의 형태를 정의한 후, 일정한 크기로 다수의 스트림을 생성하여 전송함으로써, 평활화와 선반입 과정에서 발생하는 버퍼링 시간을 줄일 수 있을 뿐만 아니라 네트워크의 지터에도 강하면, 클라이언트의 대역폭을 최대한 활용할 수 있는 효율적인 전송 특성을 얻을 수 있다.

II. 멀티스트림 전송

클라이언트와 서버사이의 데이터 전달은 TCP/IP 전송 프로토콜을 통해 연결, 데이터 전송, 제어 등이 수행된다. TCP/IP 프로토콜에서 데이터 전달 프로토콜에서 데이터 전달 프로토콜은 연결 지향적(connection-oriented) 구조인 TCP(transmission control protocol)와 비연결(connectionless) 구조인 UDP(user datagram protocol)로 나눌 수 있다[8,9]. 일반적으로 TCP는 신뢰성 있는 서비스를 제공하는 트랜스포트 프로토콜로서 종점간 접속 설정, 흐름제어, 접속 종료 등의 서비스를 제공한다. 비연결형 서비스인 데이터그램 방식의 트랜스포트 프로토콜인 UDP와는 달리 TCP는 신뢰성 있는 서비스를 제공하기 위해 재전송 기법을 이용한다. 즉, 인터넷을 통해 전송되는 데이터들은 전송되는 경로상에 있는 라우터의 버퍼 한계, 망의 대역폭의 한계 등의 이유로 인해 손실된다. 망을 통과하는 데이터가 손실되었을 때 TCP는 그 손실된 데이터를 재전송 함으로써 종점간의 신뢰성 있는 서비스를 제공하는 것이다. TCP는 망으로 유입되는 데이터의 양을 조절하기 위해 슬라이딩 윈도우 프로토콜(sliding windows protocol)을 사용한다. 일반적으로 TCP는 재전송 메커니즘에서 기인하는 중단간 지연의 증가로 멀티미디어의 실시간성을 제공하지 못하며, 슬라이딩 윈도우 프

로토콜 메커니즘은 대용량의 멀티미디어 데이터 전송에 적합하지 않다고 인식되고 있다. 그러나 전송 에러가 발생할 수 있는 환경 하에서 고압축 비디오 데이터를 전송 시, 프레임간의 종속성에 의해 현재 프레임에서 발생한 패킷 에러가 다음 프레임으로 전파될 수 있다. 따라서 이런 환경 하에서 고압축 비디오 데이터를 전송할 경우 TCP가 UDP보다 더 나은 성능을 보인다. 특히 TCP의 슬라이딩 윈도우 프로토콜은 트래픽의 평활화 효과를 나타낸다는 것을 실험을 통해 입증되었다[10-12].

TCP/IP 상에서 데이터의 전송은 MTU(maximum transmission unit) 단위로 순차적으로 수행된다. 단일 스트림을 사용할 경우 하나의 하나의 소켓을 통해 MTU 단위로 데이터가 전송되지만, 멀티스트림으로 전송할 경우 다수의 소켓을 통해 각각의 스트림마다 개별적인 MTU 단위로 데이터가 전송되기 때문에 총 패킷의 크기가 전송 대역폭보다 작을 경우 모든 MTU를 병렬적으로 전송할 수 있어 전송 대역폭을 증가시켜 클라이언트가 가진 대역폭을 최대한 활용할 수 있다. 또한 전송 시 하나의 스트림에 지터가 발생하더라도 다른 스트림은 계속 전송할 수 있기 때문에 클라이언트의 버퍼링을 통해 네트워크 지터를 극복할 수 있다. 그림 1은 멀티스트림의 효능을 설명하기 위해 기존의 인터넷 상에서 단일 스트림과 멀티스트림으로 전송하는 과정을 나타낸 것이다. 만약 대역폭이 2개의 패킷을 동시에 전송할 만큼 충분한 경우, 그림 1의 (a)에서 보는 바와 같이 단일 스트림인 경우 패킷 1이 전송된 후 패킷 2가 순서대로 전송되지만, 멀티스트림을 사용할 경우에는 그림 1의 (b)와 같이 패킷 1과 패킷 2가 동시에 전송할 수 있어 대역폭을 충분히 활용한 효율적인 전송이 가능하다.

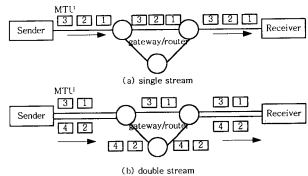


그림 1 단일스트림과 이중스트림을 사용한 전송 예
Fig. 1 Example of the Transmission using the single stream and double streams.

III. 제안한 스트림 평활화 기법

부호화시 가변 비트율로 인코딩된 비디오 스트림은 비

트를 변화가 심하여 시스템 자원을 효율적으로 사용할 수 없다. 따라서, 시스템 자원의 효율적인 사용을 위해 스트림 평활화 과정이 필요하다. 일반적으로 가변 비트율로 부호화된 비디오 스트림은 비트를 변화가 심하다. 그림 2는 akiyo 비디오 스트림에 대한 시간에 따른 전송량의 변화를 나타낸 것이다. 이미 저장되어 있는 비디오 스트림의 재생 서비스를 주로 하는 비디오 서버 시스템의 환경에서는 비디오 스트림의 생성, 저장 그리고 재생 사이의 시간차가 매우 크다. 따라서 저장하기 전에 각 비디오 스트림의 비트를 변화에 대한 정보를 충분히 알 수 있다. 이 정보에 따라 비디오 스트림을 저장하기 전에 비트율을 평활화 시키는 작업을 미리 수행할 경우 효율적인 평활화 작업을 수행할 수 있다.

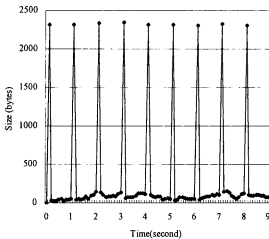


그림 2 akiyo 비디오 스트림의 프레임 형태
Fig. 2 Frame pattern of the akiyo video stream.

임의의 가변 비트율 비디오 스트림의 상영 시간이 T 이고, 임의의 상영 시점 $t_i (1 \leq t_i \leq T)$ 에서의 한 프레임의 크기를 $f(t_i)$ 라 표현하면, 재생에 필요한 전체 데이터량 D 는 식 (1)과 같으며, 평균 데이터량 D_{ave} 는 식 (2)와 같다.

$$D = f(1) + f(2) + \dots + f(T) = \sum_{i=1}^T f(t_i) \quad (1)$$

$$D_{ave} = \left\lfloor \frac{D}{T} \right\rfloor \quad (2)$$

따라서, D_{ave} 만큼씩 읽어서 재생하면 T 시간 동안 재생에 필요한 모든 데이터(D)를 T 시간 안에 읽어 재생할 수 있다. 이 방법은 복잡하게 변화하는 비트율을 하나의 평균 크기로 단일화시킬 수 있다. 그러나 D_{ave} 만큼씩 읽어서 재생할 경우, 재생되어야 하는 시점 t_i 에 필요한 데이터 $f(t_i)$ 를 $t_i (t_i < T)$ 시점에 읽는 상황이 생길

수 있다. 즉, 프레임 크기를 평균 프레임 크기로 맞추는 과정에서 $f(t_i) > D_{ave}$ 일 때, $f(t_i)$ 의 전부가 t_i 또는 그 이전 시점에 배치된다는 보장이 없기 때문에 연속성을 보장할 수 없다.

본 논문에서는 연속성을 보장하면서 평활화와 효율적인 전송을 위해 LDU라는 단위를 설정한 후, 멀티스트림을 통해 전송하고자 한다. LDU는 n 개의 프레임으로 구성되며, 구간 i 에서 LDU $_i$ 는 식 (3)과 같이 구성한다. 이때 평균 크기를 나타내는 함수 $g(t_i)$ 는 식 (4)와 같이 정의할 수 있다. 즉, 매 시간마다 $g(t_i)$ 의 크기의 전송할 경우 스트림의 평활화를 얻을 수 있다.

$$LDU_i = f_i + f_{i+1} + \dots + f_{i+n} \quad (3)$$

$$g(t_i) = \left\lfloor \frac{LDU_i}{n} \right\rfloor \quad (4)$$

여기서, LDU의 형태는 멀티스트림의 수와 부호화된 스트림의 특성에 따라 선택한다. H.263과 같은 고압축 부호화에 의해 부호화된 스트림은 I 프레임과 P 프레임으로 구성된다. 일반적으로 I 프레임의 평균 크기가 P 프레임 보다 10~30배 크다. 따라서, P 프레임의 평균 크기에 대한 I 프레임의 평균 크기의 비는 식 (5)와 같이 PIR(P frame size to I frame size ratio)로 정의한다.

$$PIR = \frac{I \text{ frame average size}}{P \text{ frame average size}} \quad (5)$$

효율적인 전송을 위한 스트림의 수 N 는 식 (6)에서 구하며, 식 (7)에 의해 LDU의 프레임 수를 설정한다. 여기서 $\lceil \cdot \rceil$ 는 양의 값보다 작지 않는 최소의 정수 값을 의미한다. 즉, I 프레임인 경우 손실되면 모든 P 프레임에게 영향을 미치게 되므로, P 프레임으로 구성된 스트림의 크기를 I 프레임에 비해 조금 할 수 있도록 설정한다.

$$N = \left\lceil \frac{PIR}{10} \right\rceil + 1 \quad (6)$$

$$LDU \text{의 프레임 수 } n = \left\lceil \frac{PIR}{N-1} \right\rceil + 1 \quad (7)$$

PIR 값이 26인 경우 식 (6)에 의해 멀티스트림의 수 N 는 4이므로, 식 (7)에 의해 LDU의 프레임 수는 11 프레임이다. 즉, 1개의 I 프레임과 10개의 P 프레임으로 구성된다. 따라서, 스트림의 수가 4개이므로, 전송을 위한 멀티 스트림은 그림 3과 같이 배치된다. 4개의 멀티스트림 중 P 프레임 보다 약 26배가 큰 I 프레임은 3개의 스트림으로 나누어져 전송되며, 크기가 작은 P 프레임은 10개의 프레임이 하나로 묶어져 하나의 스트림으로 전송된다. 각각의 스트림들은 비슷한 크기를 가지게 되어 평활화 된다. 그림 4는 PIR의 값에 따른 식 (6)과 식 (7)에 의

해 설정되는 기본 프레임 패턴을 멀티스트림의 형태로 나타낸 것이다.

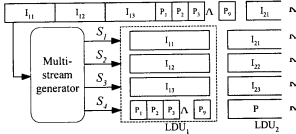


그림 3 멀티스트림 생성의 예 (N=4, LDU=10 frames)
 Fig. 3 Example of multistream generation(N=4, LDU=10 frames).

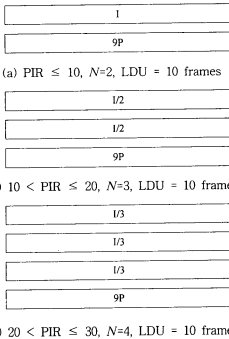


그림 4 PIR에 따른 기본 패턴 설정
 Fig. 4 Default pattern setting according to PIR.

IV. 실험 및 고찰

본 논문에서는 실험을 위해 Telenor사의 TMN2인 H.263 부호화기를 수정하여 부호화하였다. 실험을 위해 5개의 176×144 크기를 가지는 QCIF 형식의 비디오 스트림들을 사용하였다. 이들 비디오 스트림들은 24Kbps~20Mbps의 지속 전송 매체를 위한 비디오 부호화를 위해

제안된 H.263 부호화기의 실험을 위해 문헌들에서 빈번하게 사용되는 유명한 것들이다.

표 1은 비디오 스트림들을 대해 평균 I 프레임의 크기와 평균 P 프레임의 크기 그리고 P 프레임의 크기에 대한 I 프레임의 크기 비 PIR을 나타낸 것이다. PIR은 비디오 스트림에 따라 최대 29.44, 최소 3.62로 비트를 변화가 다양함을 알 수 있다. 이 비디오 스트림의 평균 I 프레임의 크기는 2322 bytes 이며, P 프레임의 평균 크기는 89로 PIR이 약 30이다. 즉, 프레임 별로 전송시 비트의 변화가 많아 네트워크 자원을 효율적으로 이용할 수 없게 된다. 따라서 스트림에 대한 평활화 작업을 통해 비트율을 일정하게 만든다.

표. 1 비디오 스트림의 P 프레임 크기 대비 I 프레임 크기 비
 Table 1 P frame size to I frame size ratio of the video stream.

| 스트림 형식 | 비디오 스트림 이름 | I frame의 평균 크기 (bytes) | P frame의 평균 크기 (bytes) | PIR |
|--------|---------------|------------------------|------------------------|-------|
| QCIF | Akiyo | 2322 | 89 | 26.09 |
| | Miss American | 1336 | 92 | 14.52 |
| | Salesman | 2791 | 130 | 21.47 |
| | Carphone | 2573 | 292 | 8.81 |
| | Foreman | 3066 | 367 | 8.35 |

그림 5는 그림 2와 같은 원래의 Akiyo 비디오 스트림을 LDU 단위로 평활화를 수행한 비디오 스트림을 나타낸 것이다. 여기서 LDU는 10 프레임, 20 프레임, 30프레임 그리고 40 프레임으로 설정하였다. LSU의 크기를 늘수록 평활화를 간단히 구현할 수 있으나 버퍼링에 의한 지연이 길어진다. 이 지연은 비디오 스트림의 재생할 경우 서비스 품질 저하를 야기 시킬 수 있다. 표 2는 LSU의 크기에 따른 제안한 멀티스트림과 기존의 단일스트림에 의한 지연 시간을 나타낸 것이다. 멀티스트림 수는 PIR의 값에 의해 식 (6)과 식 (7)에 의해 4로 설정되며, 스트림은 각각 3개의 I 스트림과 1개의 P 스트림으로 구성된다. 따라서, 그림 6에서 보는 바와 같이 제안한 멀티스트림을 이용한 평활화 방법의 경우 멀티스트림 전송에 의해 지연시간이 약 1/4로 감소함을 알 수 있었다.

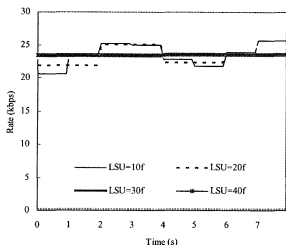


그림 5 LDU를 이용한 평활화 (Akiyo)
Fig. 5 Smoothing using LDU (Akiyo)

표 2 LSU의 크기에 따른 지연 시간 비교

Table 2 Comparison of Delay time according to the size of LSU.

| LSU의 크기 | 스트림수 기존의 단일스트림 (seconds) | 제한된 멀티스트림 (seconds) |
|-----------|--------------------------------|------------------------|
| 10 frames | 0.87 | 0.21 |
| 20 frames | 1.86 | 0.47 |
| 30 frames | 2.92 | 0.73 |
| 40 frames | 3.99 | 0.98 |

V. 결론

본 논문에서 비디오 스트림의 효율적인 전송을 위해 멀티스트림을 이용한 비디오 스트림의 평활화 기법을 제안하였다. 제안한 방법은 LSU의 선반입에 의한 초기 지연을 효과적으로 감소시키기 위해 현재의 인터넷 인프라 환경 하에서 클라이언트가 가진 대역폭을 최대한 활용한 멀티스트림 전송을 통해 평활화 작업을 수행하였다. 멀티스트림의 수는 비디오 스트림의 PIR 값을 이용하여 각 스트림 별로 평활화 할 수 있는 수준 찾아 전송함으로써, 각 스트림의 평활화도 수행하였다. 실험 결과 LSU의 크기가 클수록 평활화의 효과가 크게 나타남을 확인하였으며, 멀티스트림의 수가 n 인 경우 비피딩을 위한 지연 시간은 $1/n$ 로 감소함을 확인하였다. 향후 네트워크 트래픽에 따라 프레임에 대한 동기화에 관한 연구가 된다면 효율적인 비디오 스트림 전송이 가능하리라 사료된다.

접수일자 : 2001. 12. 4 수정완료 : 2002. 1. 23

본 논문은 2001년 정보통신부 대학 기초 연구지원 사업에 의한 연구 결과의 일부임

참고문헌

- [1] Richard Muriden, "Use of audio and video on the internet," *Proc. INET '95*, 1995.
- [2] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta, and J. Hill, "Statistical multiplexing of VBR MPEG compressed video on ATM networks," *Proc. of IEEE INFOCOM '93*, pp. 919-926, 1993.
- [3] D. R. Kenchammana-Hosekote, and J. Strivastava, "Retrieval techniques for compressed video streams," *Proc. of multimedia computing and networking (MMCN) '96*, Jan. 1996.
- [4] S. Jung and J. S. Meditch, "Critical bandwidth allocation for the delivery of compressed video," *Computer communications*, vol. 18, no. 10, pp. 709-717, Oct. 1995.
- [5] P. Pancha, and M. E. Zarki, "Prioritized transmission of variable bit rate MPEG video," *Proc. of IEEE GLOBECOM '92*, pp. 1135-1139, 1992.
- [6] R. P. Tsang, D. H. C. Du, and A. Pavan, "Experiments with video transmission over an asynchronous transfer mode(ATM) network," *ACM multimedia systems 4*, pp. 157-171, 1996.
- [7] T. Ott, T. V. Lakshman, and A. Tabatabai, "A scheme for smoothing delay-sensitive traffic offered to ATM networks," *Proc. of IEEE INFOCOM '92*, pp. 776-785, 1992.
- [8] J. Feng and S. Sechrest, "Critical bandwidth allocation for the delivery of compressed video," *Computer communications*, vol. 18, no. 10, pp. 709-717, Oct. 1995.
- [9] J. Postel, "Transmission control protocol," RFC 793, Sep. 1981.
- [10] J. Postel, "User datagram protocol," RFC 768, Aug. 1980.
- [11] R. Tsan, D. Du, A. Pavan, "Experiment with video transmission over an asynchronous transfer mode(ATM) network," *Multimedia system*, vol. 4, pp. 157-171, 1996.



강경원(Kyung Won Kang),

正會員

1996년 부경대학교 전자공학과
(공학사)

1998년 부경대학교 대학원 전자공학과
(공학석사)

1998년~현재 부경대학교 대학원
박사 과정 재학중

1998년 3월~1999년 8월, 미국 Jackson State University
연구원

관심 분야 : 멀티미디어 통신, 영상신호처리



문광석(Kwang Seok Moon),

正會員

1979년 2월 경북대학교 전자공학과
(공학사)

1981년 2월 경북대학교 대학원
전자공학과(공학석사)

1989년 2월 경북대학교 대학원
전자공학과(공학박사)

1988년 1월~12월 일본 동경대학교 공학부 연구원

1997년 8월~1998년 7월, 미국 Jackson State University

객원교수

1990년~현재 부경대학교 전자공학과 교수

관심 분야 : 영상신호처리, 적응신호처리,
멀티미디어 통신 등.
