

계층적 MPEG 스트리밍을 위한 실시간 Rate Control 알고리즘

박준호*, 나윤주*, 김철용**, 남지승*
*전남대학교 컴퓨터 공학과
**전남대학교 대학원 정보통신 협동과정

e-mail:entia3@hanmail.net

Algorithm on the Rate Control Technology with Real-Time for Layered MPEG Streaming

Jun-Ho Park*, Yun-Joo Na*, Cheul-Young Kim**, Jie-Seung Nam*
*Dept of Computer Engineering, Chonnam National University
**Interdisciplinary Program of Information and TeleCommunication,
Chonnam National University

요약

현재 인터넷의 서비스 중 동영상은 많은 대역폭을 차지하고 있으며 실시간으로 전송되어야 하는 특성을 가지고 있다. 이러한 조건을 만족하기 위해서 Layered coding의 특성을 이용한 스케일러블 전송이 MPEG2를 통해 가능하게 되었다. 본 논문에서는 MPEG2의 스케일러블 전송을 바탕으로 송신측에서의 폭주제어와 함께 실시간 서비스를 제공하는 알고리즘을 제안한다. 본 알고리즘에서는 폭주 제어를 위해 Congestion Window를 사용하며 네트워크 속도를 측정하여 QoS(Quality of Service)를 보장한다.

1. 서론

멀티미디어 데이터 중 하나인 동영상 데이터는 데이터의 양이 크며 실시간으로 전달되어야 한다는 특성을 갖고 있다. 이러한 대용량의 데이터는 통신망을 오가는 대역폭의 많은 부분을 차지한다. 이에 따라 효율적인 통신망 환경 조성을 위해서 비디오를 효율적이고 안전하게 전송할 수 있는 방법으로 비디오 데이터의 layered coding과 scalable 전송, 그리고 데이터 크기를 줄여서 전송하는 압축이 필요로 하게 되었다. 현재 이런 실시간 멀티미디어 데이터의 효율적인 전송을 위해 국제 표준안으로 제안된 압축기법 중 하나가 MPEG이다. 하지만 송수신 PE(Processing Element) 간의 처리 속도 불균형과 데이터 흐름의 집중 현상, 통신망의 용량을 초과하는 데이터 유입 등으로 네트워크 통신 효율은 떨어져 결국 혼잡 현상이 발생한다. 이러한 혼잡 상태로 인해 비디오나 동영상, 음성 신호와 같은 멀티미디어 데이터를 상실했을 경우에는 막대한 손해를 입는 경우가 생긴다. 그렇기 때문에 최소한의 화질을 유

지하면서 네트워크 상의 고장이나 폭주를 진단해서 처리할 수 있어야 한다.[1][3][4]

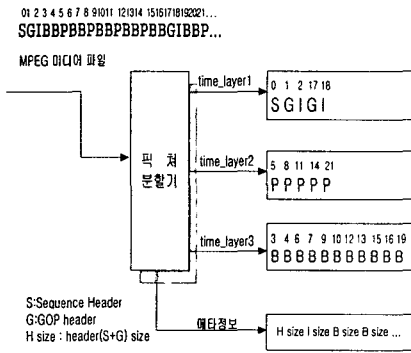
본 논문에서는 혼잡이 발생하였을 때의 흐름제어 방법과 더불어 MPEG의 계층형 특성을 이용한 전송률 제어기(Rate Control)에 대하여 알아본다.

2. MPEG 관련 기술

2.1 시간적 스케일링

시간적 스케일링은 MPEG 비디오 파일의 픽처 층을 구성하는 픽처 타입(I, P, B 픽처)에 따라 3개의 계층으로 나눈 다음, 네트워크의 QoS가 충분하면 3개의 레이어를 모두 전송하고 QoS가 낮아지면 3개의 레이어 중 일부만을 전송하여 전송률을 QoS에 맞춰 조정하는 기법이다. MPEG 비디오 파일에 시간적 스케일링 과정을 적용하면 time_layer1에서 time_layer3까지 3개의 계층으로 분리된다. (그림 1)과 같이 time_layer1은 I 서브스트림과 GOP, 시퀀스의 헤더정보들을 모아놓았고, time_layer2는 P 서브스트림이, time_layer3는 B 서브스트림만으로 구

성되어 있다. 픽처 계층보다 상위 계층의 부가 정보인 시퀀스 헤더와 GOP 헤더 정보는 time_layer1 계층에만 포함되어 있기 때문에 독자적으로 디코딩될 수 있다. 반면에 time_layer2(P 서브스트림)와 time_layer3(B 서브스트림)은 존재하지 않아도 재생이 가능하다. 그러나 이들 2개 계층이 누락되게 되면, 프레임 Drop으로 인하여 비디오의 동작이 부자연스럽게 된다. 계층간의 종속성을 살펴보면, time_layer2계층의 디코딩을 위해서는 time_layer1계층이 필요하며 time_layer3계층의 디코딩은 time_layer1, time_layer2 계층이 필수적이다. 실험 결과에 의하면 time_layer1이 차지하는 비율은 25% 내외이고, time_layer2는 35%내외이며 time_layer3은 대략 40%정도이다.[1]



(그림1) 시간적 스케일링 기법

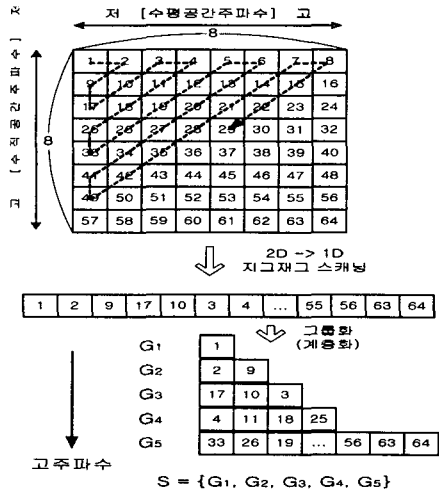
2.2 충실도 스케일링

충실도 스케일링은 MPEG-2 비디오 문법 구조의 최하위 층인 블록을 코딩할 때 산출된 DCT 계수를 조정함으로써 화질을 다양하게 계층화하는 미디어 스케일링 기법이다. 블록은 DCT를 수행하는 단위로써 8×8픽셀 크기를 가지며, DCT 변환 후에는 하나의 DC와 63개의 AC로 이루어진다. (그림2)와 같이 2차원 블록을 변환한 64개의 2차원 DCT 계수는 스케닝 과정을 통하여 1차원 계수열로 변환한다. 64개의 1차원 DCT 계수는 5개의 계수 그룹 G1~G5로 분할하였으며 이 계수 그룹들은 순서 있는 집합(ordered set)인 S의 부분집합으로써 각각이 하나의 계층을 이룬다.

이는 5개의 계층으로 구분된다. 그룹 G1은 DC 성분(1번 계수)을 가진 fidelity_layer1(평균27%)이라고 하고, AC1과 AC2(2, 9번 계수)를 갖는 그룹 G2를

fidelity_layer2(평균6%)라 하며, 그룹 G3의 AC3~AC5(3, 10, 17번 계수)를 fidelity_layer3(평균7%), 그룹 G4의 AC6~9(4, 11, 18, 25번 계수)를 fidelity_layer4(평균20%), 그룹 G5는 나머지 고주파 성분 AC10~AC63(33~64번 계수)의 fidelity_layer5(평균40%)로 하였다.

기본적으로 충실도 스케일링을 적용한 계층은



(그림2) 충실도 스케일링 기법

fidelity_layer1만을 제외한 다른 계층들은 독립적으로 재생이 불가능하다. 따라서 fidelity_layer2 이상의 계층들은 이전의 계층들을 하나로 병합한 후 재생하여야 한다.[1]

2.3 통합된 분할 기법을 적용한 스케일링

네트워크 상황에 맞추어 전송하는 MPEG 데이터의 양을 조절하기 위해서 주어진 MPEG-2 비디오 비트 스트림에 시간적 스케일링과 충실도 스케일링 방법을 함께 적용하면 스케일링 기법을 개별적으로 적용하는 것보다 더욱 융통성 있게 QoS에 적용할 수가 있다. 분할 순서는 시간적 스케일링 기법을 먼저 적용하여 픽처를 I, P, B에 따른 분류 후, 분할된 픽처의 DCT 블록에 대해 충실도 조정 기법을 적용한다. 시간적 스케일링 기법을 적용한 계층에 충실도 스케일링 기법을 적용하게 되면 3개의 time_layer마다 5개의 fidelity_layer가 생성되어 총 15개의 계층화 파일이 생성된다. 데이터양은 T1F1에서 T3F5까지 다양하게 변화하여 나타난다. 끊임없는 시각적인 효과를 고려한다면 T1F1에서 T2F1로 T3F1 식으로 스케일

링을 변화시키고 화면의 품질을 중요시하면 T1F1에서 T1F2 식으로 충실도를 높여야한다.

3. 적응형 MPEG 스트리밍을 위한 흐름제어 방식

3.1 실시간 데이터 전송을 위한 흐름제어

통신망의 자원을 최대한 활용하는 형태로 개발된 비연결형 통신 프로토콜을 사용하는 통신망에서는 할당된 자원의 관리가 매우 어려워 자원 할당 기법을 일반적으로 사용하지 않는다. 따라서 사용자의 QoS 요구사항을 맞추어 주기가 매우 어렵다. 또한 통신망에 진입하는 데이터의 양을 예측하기 어렵기 때문에 다양한 흐름제어 방식이 개발되어 왔다.

인터넷에 적용이 가능한 이러한 기존의 흐름제어 기법은 망에 진입하는 데이터의 양을 조절하기 위해 전송하고자 하는 데이터를 망에 진입시키는 순간을 지연시키는 방법을 사용하였다. 이러한 방법은 비실시간 데이터의 경우에는 목적인 바를 달성하고 있다. 그러나 실시간 미디어의 경우에 전달되는 데이터는 일정한 시간 내에 수신 노드에 도착해야 전달된 데이터가 유효하게 사용될 수 있다. 정해진 시간 내에 수신 노드에 도착하지 못하는 데이터는 유효하게 사용될 수 없으므로 전달하는 과정에서 통신망에 부하를 가중하지 않도록 전송을 하지 않는 것이 좋다. [3][4]

최근에 들어 이를 적용하여 미디어 스케일링 기법을 통하여 멀티미디어 데이터의 양을 조절함으로써 채널에 실리는 부하를 경감시키려는 연구가 진행되고 있다. 이러한 접근 방식은 응용계층에서 사용자가 전송하는 데이터의 특성에 따라 요구하는 QoS를 제공하기 위한 노력으로서 의미를 가지고 있다.

3.2 흐름제어를 위해 고려할 변수

본 논문에서 개발하고자 하는 흐름제어는 트랜스포트 계층의 UDP나 RTP 프로토콜을 이용한 응용프로그램 계층에서 개발하고 있다. 이러한 환경에서 흐름제어를 위한 어떠한 Parameter를 사용하는 것이 가능한지를 알아본다.

(1) 패킷손실 (Packet Loss)

대부분의 패킷 손실은 망에 데이터 패킷이 너무 많이 진입하거나 특정 라우터에 몰려 혼잡이 발생하여 라우터나 수신노드의 버퍼가 받아들일 수 없어 패킷을 버리게 되어 발생한다.

(2) 지연 (Delay)

송수신 시스템이 가지고 있는 시간의 차이가 존재하기 때문에 패킷의 전송 지연시간을 측정하는 것은

어려운 일이다. 그러나 패킷을 수신하는 노드에서 응답 패킷을 되돌려 줌으로 해서 이를 이용하여 Turn-around time을 측정할 수 있다. 이러한 방식은 전송 지연시간 뿐만 아니라 패킷이 손실 없이 전달되었는지 송신노드가 알아내는 방법으로 사용될 수 있다.

(3) 버퍼의 변화 (Threshold 증가유)

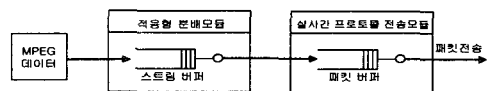
데이터를 송신하는 시스템이나 수신하는 시스템 그리고 라우터는 순간적으로 변하는 처리량의 변화에 대처하기 위하여 버퍼를 이용하고 있다. 하위 계층의 프로토콜에서는 버퍼에 쌓이는 데이터의 양을 이용하여 이러한 흐름제어를 수행하는 다양한 방법이 연구되고 있다. 역시, 이러한 기법을 상위계층에서도 QoS 관리에 이용할 수 있다.

4. 제안하는 흐름제어 알고리즘

· 망의 대역폭에 수렴하는 QoS 제어 기법

시간적 스케일링과 충실도 스케일링을 적용하여 최소 스케일의 MPEG 비디오 스트림을 전송하기 시작한다. 이후 인터넷에서 혼잡 없이 원활한 전송이 이루어지면 점차적으로 시간적 스케일링과 충실도 스케일링을 높여 나간다. 이러한 방법을 통하여 네트워크에서 현재 제공하는 대역폭에 맞추어 적절한 화질의 비디오 데이터를 전송할 수 있다.

인터넷에서 현재 제공하는 전송률을 알아내기 위하여 송신 노드는 다음과 같이 2단계 버퍼를 사용한다(그림3). 첫째 버퍼는 영상저장 장치에서 가져온 비디오 스트림과 전송률 제어기 모듈에서 가져온 QoS 정보를 조절하는 스트림 버퍼이다. 둘째 버퍼는 스케일링을 적용한 GOP 단위의 영상데이터를 하위 프로토콜에서 전송하기에 적절한 패킷들로 분할하여 보관하는 패킷 버퍼이다.

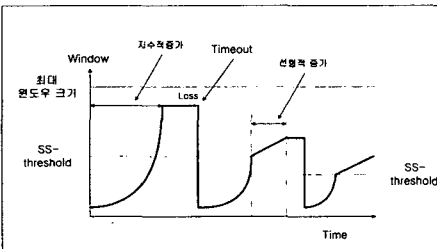


(그림 3) 송신 노드의 2중 버퍼 구조

패킷 버퍼에서 만일 흐름제어 없이 데이터를 지속적으로 전송한다면 송신노드의 전송 능력에 따른 속도로 패킷이 전송 될 것이다. 그러나 이 전송 속도는 인터넷에서 제공하는 전송 속도에 비하여 일반적으로 매우 빠르다. 따라서 인터넷에서는 혼잡현상이 발생하여 인터넷의 성능을 저하시킬 것이다. 인터넷 속도에 맞추어 데이터를 전송하기 위해서는 이에 맞는 흐름제어 기법이 필요하다. 인터넷 능력에 맞추어 데이터를 전송하는 흐름 제어 기법에는 Stop

and wait 기법, Slide window 기법, Congestion window를 이용한 Slow-start 알고리즘 등이 있다. 이 중 Congestion window를 이용한 흐름제어 기법을 패킷버퍼에 있는 데이터를 전송하는데 적용한다면 데이터 전송속도는 네트워크에 현재 상황에 맞춘 전송속도로 패킷을 전송하게 될 것이다.

Congestion window은 전송한 세그먼트에 대한 타이머가 만료되면 현재 폭주윈도우의 1/2을 ss-threshold로 설정하고 폭주윈도우의 크기를 1로 설정한 후 slow start 단계로 들어가 손실된 세그먼트부터 다시 재 전송하는 방법으로 폭주윈도우의 크기가 ss-threshold에 이르거나 세그먼트의 손실이 감지될 때까지 계속 증가하게 된다. 또한 폭주윈도우의 크기가 ss-threshold와 같아지거나 커지면 congestion avoidance 단계로 들어가는데 이 단계에서 송신측은 하나의 완전한 폭주윈도우가 전부 ACK 되면 폭주윈도우의 크기를 하나 증가시키는 방법으로 현재의 폭주윈도우의 크기를 x로 가정하면 매 ACK 마다 1/x 만큼의 폭주윈도우가 증가하게 된다. 이는 폭주윈도우의 크기가 선형적으로 증가하며 최대 윈도우 크기(maximum window size)에 도달하거나 세그먼트의 손실이 감지될 때까지 계속된다. Slow start와 혼잡 회피(congestion avoidance)와 타임아웃이 적용될때의 모습은 (그림4)에서 볼 수가 있다.[2][3][4][5]



(그림4) Congestion Window를 사용한 Slow Start 방법

이와 같이 진행된 데이터 전송 흐름으로부터 우리는 인터넷에 평균전송 속도를 구해 낼 수 있다. 구해내는 방법은 GOP 데이터의 패킷전송을 시작하는 시간(T_1 sec)과 전송이 끝난 시간(T_2 sec)을 측정하여 전송하는데 소모된 시간($T_2 - T_1$)을 얻고 GOP 데이터의 크기(M bits)를 이 값으로 나누게 되면 현재 인터넷의 전송 속도(S_b bit/sec)에 해당하는 값을 얻을 수 있다. 즉, 다음 수식과 같게 된다.

$$S_b = M / (T_2 - T_1) \text{ bits/sec}$$

만일 스케일링이 적용 안된 MPEG 비디오 스트림의 GOP가 N 개의 프레임으로 구성되어 있다면 인터넷의 프레임 전송속도는 $S_r = N / (T_2 - T_1)$

frames/sec 이다. 구해진 S_r 값은 스케일링이 적용된 GOP 단위의 프레임을 전송하였기 때문에 사실은 N 개보다 작은 수의 프레임을 전송하였을 것이다. 일반적으로 S_r 값이 초당 30 frames 정도일 때 시각적으로 자연스럽다. GOP 단위의 패킷 전송을 수행하면서 S_r 값이 30보다 큰 경우마다 충실도 스케일링 수준 혹은 시간적 스케일링 수준을 점차 높여준다. 만일 S_r 값이 30보다 작은 경우에는 네트워크에서 제공하는 전송속도보다 더 많은 양의 데이터를 전송하고 있다는 것을 의미한다. 이러한 경우 네트워크에서는 이를 감당할 수 없게 되어 혼잡이 발생할 것이다. 따라서 충실도 스케일링 혹은 시간적 스케일링 수준을 낮춰 주어야 한다. 이와 같은 스케일링 조절을 수행하는 과정에서 전송하는 MPEG 비디오 스트림의 화질은 네트워크의 전송속도에 수렴된다.

V. 결론

현재 인터넷의 멀티미디어 동영상은 데이터의 양이 크며 실시간으로 전달되어야 한다는 특성을 가지고 있다. 이를 위해서는 인터넷의 성능에 맞추어 망에 진입하는 데이터의 양을 조절할 수 있는 스케일링 전송방법이 필요하다. 본 논문에서는 시간적 스케일링과 공간적 스케일링 전송을 통해 실시간 데이터 전송을 만족하는 흐름제어 알고리즘을 제안하였다. 본 알고리즘의 목표는 효율적인 자원의 이용과 데이터의 실시간 전송이다. 향후 과제는 알고리즘의 구현과 더불어 멀티캐스트 분야로의 확장이다.

참고문헌

- [1]김형철, "QoS 적용형 MPEG 비디오의 미디어 스케일링 방안 고찰" 한국 전자 통신 연구소 기술 문서, 1999
- [2]Yeali S.Sun, F-M Tsou, Meng Chang Chen and Zsehong Tsai, "A TCP-Friendly Congestion Control Scheme for Real-time Packet Video Using Prediction", IEEE GLOBECOM'99, pp.1818-1821, 1999
- [3]Douglas E. Comer, David L. Stevens, "internetworking with TCP/IP", vol.1, Prentice-Hall, Inc, pp.179-228, 1995
- [4]Douglas E. Comer, Rajendra. Yavatkar, "A Rate-Based Congestion Avoidance and Control Scheme for Packet Switched Network", Proc of 10th ICDCS, IEEE, pp.390-397, 1990
- [5]Syllabus, "Computer Network", <http://www.cs.columbia.edu/networks97/sched97.html>