

ATM 망에서 비디오 서비스를 위한 동적 VBR 트래픽 특성화 (Dynamic VBR traffic characterization for video service in ATM network)

황재철* 조미령** 이상원*** 이상훈****
(Jae-Chul Hwang) (Mee-Ryoung Cho) (Sang-Won Lee) (Sang-Hun Lee)

요 약

본 논문에서는 ATM상에서의 효율적인 VBR 비디오 소스 전송을 위한 트래픽 특성화를 다루었다. 트래픽 특성화를 위하여 전송중 트래픽이 적은 모니터링 기법을 사용하였고 지연 요구사항을 만족시키는 동적 VBR 특성화 방식을 제안하였다.

동적 VBR 방식은 누적 제한 함수 (Cumulative Constraint Function)를 이용한 토큰 버킷 알고리즘 버퍼링을 사용하였다. 누적 제한 함수는 초기에는 peak rate로 전송을 시작하여 일정시간 동안 모니터링 한 후, 다음 트래픽을 예측하여 적절한 버퍼를 제공하였다. 누적 프레임의 갱신을 통하여 네트워크 자원 예약량을 감소시켰고 이전 프레임 정보를 가지고 rate를 변경시켰다. 이는 적은 모니터링을 하며 오버헤드도 적다. 그리고 이전의 결정적 제한 함수 (Deterministic Constraint Function)의 복잡도를 $O(n)$ 까지 낮추었고, 토큰 버킷의 토큰 rate와 토큰 depth를 구하여 매핑하였다. 본 논문은 시뮬레이션을 통한 결과치를 비교 분석하여 이전의 방식보다 적은 네트워크 자원이 소모되는 것을 보여준다.

ABSTRACT

This paper is focused on the traffic characterization for the efficient transmission of the VBR video source in the ATM network. For the traffic characterization, low traffic monitoring technique is applied and the dynamic VBR characterization method is suggested to satisfy the delay requirement.

The dynamic VBR method uses the token bucket algorithm buffering though Cumulative Constraint Function. According to the Cumulative Constraint Function, the packet initially started transferring at the peak rate and the token bucket provided proper amount of buffer for traffic after a certain period of monitoring. It also reduced the network resource bandwidth through renewal of the cumulative frame and changed the rate from the previous frame information. It requires only small amount of monitoring and causes little overhead. In addition, it lowered the complexity of Deterministic Constraint Function to $O(n)$ and mapped the token rate and token depth to the token bucket. This study shows less network resource consumed than the previous method, comparing and analyzing the result of simulations.

* 정회원 : 광운대학교 전자공학과 대학원 석사과정 졸업
** 정회원 : 광운대 일반대학원 전자공학(박사) 재학
*** 정회원 : 광운대학교 대학원 전자공학과 박사과정 수료
**** 중신회원 : 광운대학교 전자공학부 교수

논문접수 : 2001. 4. 12.
심사완료 : 2001. 4. 28.

1. 서론

현대 정보화 사회에서 네트워크 서비스는 음성, 데이터, 화상 등 멀티미디어 서비스를 주로 제공하게 된다. 이러한 멀티미디어 서비스를 제공하기 위해 B-ISDN(Broadband-Integrated Service Digital Networks)은 다양한 서비스를 만족시킬 수 있는 여러 범위의 대역폭을 수용할 수 있어야 한다. 이와 같은 특성화된 서비스 클래스에 따른 다양한 QoS (Quality of Service)를 만족시키기 위해 ATM (Asynchronous Transfer Mode)을 전송 및 교환의 기본 방식으로 채택하였다.[1],[2]

ATM Forum에서는 영상 전송 시 CBR (Constant Bit Rate) 트래픽 서비스를 사용할 것을 권장하고 있지만 이것은 실시간 멀티미디어 서비스를 위해서는 바람직하지만 네트워크의 효율 면에서는 권장할 만한 사항은 아니다.[4]

기존의 회선 교환망에서 비디오 전송은 CBR로 이루어진다. 그 결과 모든 영상에 대하여 균일한 화질을 유지할 수 없다.[5] 이러한 문제점은 비디오 소스에 Peak Bit Rate로 할당하여 해결할 수 있지만, 영상의 내용이 복잡하지 않은 부분에서는 많은 대역의 낭비를 초래할 수 있다.[6] 따라서, 대역의 낭비를 줄이고, 전송된 양상의 균일한 화질을 유지할 수 있도록 필요한 대역을 가변적으로 할당할 수 있어야 한다. 이를 위해서는 비디오 소스의 특성을 바탕으로 발생 트래픽량을 정확히 예측할 수 있는 방법이 필요하다. 이중 real-time Variable Bit Rate (rt-VBR)은 버스티 (Bursty) 소스로부터의 실시간 트래픽 뿐만 아니라 대역의 효율적 면이나 예측 기법상 가장 적합한 서비스이다. 이 rt-VBR은 보통 다른 소스들로부터의 트래픽을 갖는 통계적 다중화 방식을 허용하며 Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), 그리고 Maximum Burst Size (MBS)를 사용할 수 있다.[7]

최근 ATM을 기반으로 한 B-ISDN에서의 VBR을 갖는 비디오의 전송에 관한 연구가 활발히 진행되고 있다. ATM은 대역 이용률을 높이기 위하여 고정된 크기의 셀 (cell)을 통계적으로 다중화하여, 음성 및 비디오와 같은 다양한 범위의 서비스를 동시에 제공할 수 있다.

영상 VBR 연결은 상당히 많은 대역폭을 필요로 하는 동시에 최소한의 QoS, 예를 들어 bound delays, minimum throughput, bounded jitter 의 보장을 요구한다. 많은 고품질 비디오 응용은 worst-case QoS 를 보장해주는 결정적 (deterministic) 서비스를 요구하고 있다.[8],[9]

결정적 서비스의 구현을 위해 가장 중요한 요소는 연결에서 트래픽을 구체화할 수 있는 트래픽 특성화이다. 결정적 서비스를 이용한 네트워크 상에서의 트래픽 특성화는 수락 제어 함수와 정책 결정 방법에서 사용된다. 연결 설定时, 수락 제어 함수가 존재하는 다른 연결의 QoS를 보장하면서, 새로운 연결을 설정할 수 있는지에 대해 결정을 하며, 연결이 결정된 후에는 정책 결정 방법은 네트워크에 연결된 모든 연결들의 QoS를 보장한다.

결정적 서비스에 의한 트래픽 특성화는 정확해야만 한다. 만약 트래픽 특성화가 실제 트래픽보다 많은 트래픽을 규정한다면 수락 제어 함수는 새로운 연결 설정 시, 네트워크 자원을 과대 평가하게 되며, 이는 곧 네트워크의 효율을 낮추게 된다. 실시간에서의 정교한 억제 함수는 결국 복잡한 정책 결정 방법을 필요로 하게 된다. 트래픽 특성화의 선택은 결국 정교한 수락 제어와 정책 결정 방법을 구현하기 위해 요구되는 간소함의 사이의 tradeoff 이다.

트래픽 특성화 함수는 토큰 버킷 방식에 매핑되어 사용할 수 있다. 토큰 버킷은 효율적으로 구현될 수 있으며, VBR 영상을 정확히 특성화시킬 수 있다.[10],[11]

본 논문에서는 두 가지 기본적인 영상 특성화에 대하여 연구하였다. 첫 째로 영상 트래픽을 정확히 특성화를 하기 위해 얼마나 많은 양의 정보를 필요로 하는가에 대한 것으로 이를 위해 누적 제한 함수 (Cumulative Constraint Function)를 제안하였으며, 정보의 양을 변화시켜가면서 누적 제한 함수의 정확도를 측정하였다. 두 번째로 이 함수를 이용하여, 토큰 버킷의 토큰 rate (r)과 버킷 depth (b)에 매핑하였다.

2. 트래픽 특성화 제어

네트워크에서 트래픽 제어가 이루어지는 시점에 따라 크게 예측 제어와 반응 제어로 나눌 수 있다. 예측 제어라 함은 네트워크 내로 사용자 트래픽을 유입시키기 전 호 설정 단계에서 네트워크 트래픽 폭주가 발생하지 않도록 미리 트래픽 제어를 하는 방법을 말하며, 반응 제어는 네트워크로부터 트래픽 발생원으로 귀환되는 네트워크의 트래픽 상태에 관한 정보에 따라 전체적인 네트워크의 폭주를 동적으로 제어하는 방법을 말한다. 이중에서 ATM Forum에 규정된 다섯 가지 서비스중 ABR 트래픽은 반응 제어를 사용하며 나머지 트래픽에서는 예측 제어가 사용된다. 즉 rt-VBR 서비스에서 대역폭을 예측하여 할당할 수 있다. 실제의 트래픽의 흐름을 모니터링하여 실시간으로 서버의 서비스율을 가변시켜 준다. 트래픽 소스의 평균 대역폭과 현재의 대역폭, 표준 편차를 이용하여 다음 프레임의 대역폭을 할당하는 방법과, 소스의 평균 대역폭과 현재의 대역폭과 자기 상관 함수를 이용하여 대역폭을 예측하는 방법 등이 있다.[12],[13]

지연에 민감하지 않은 트래픽의 특성화는 smoothing 기법으로 해결할 수 있으나 영상 신호와 같이 지연에 민감한 경우는 크게 통계적 접근 방식(statistical approach)과 연산적 접근 방식(operational approach)으로 해결할 수 있다. 전자는 소스의 통계적 성질 즉, 셀의 퍼크, 평균, 그리고 표준편차와 같은 트래픽 파라미터를 예측하여 이용할 수 있다. 이 방식은 잠재적으로 높은 네트워크 효율성을 갖지만, 자기 다른 성격을 지닌 영상 트래픽의 burstiness와 시간적 상호 관계(timely correlation)에 적합하지 못하며, 구현 자체가 복잡하다는 단점을 가지고 있다. 그리고 확률적 보장만을 해준다는 것이다. 그리고 후자는 무손실 서비스를 위한 것이며 유한정된 n개의 프레임 정보를 모니터링을 통해서 트래픽 강제 함수를 구한 뒤 그 유형을 반복하여 상한 경계치를 구해 특성화시키는 방법이다. 확률론적 방식(stochastic approach)이나 결정적 방식(deterministic approach)도 모두 연산적 방식에 포함된다. 그러나 연산적 유한 모니터링 기법은 아래와 같은 몇 가지 문제점이 있다.

- ① 유한적 모니터링으로는 전체의 트래픽 경향을 알기 어렵다.
- ② 계산 복잡도가 높다. 즉, n개의 프레임을 모니터링 했을 경우에, 복잡도는 적어도 $O(n)$ 이다. 만약 복잡도가 낮다면, 정확도가 매우 낮게 된다.
- ③ 초기 모니터링은 시간 비용을 수반한다. 즉 지연에 민감한 상위 레벨의 요구 사항을 충족시키기가 어렵다.

확률론적 방식은 VBR 프레임 크기별 분포등의 특성을 이용하여 확률 과정론에 기반한 접근 방식으로 잠재적으로 네트워크 자원을 효율적으로 사용한다. 그리고 이 방식은 Bursty에 약하고 약간 복잡한 단점을 가지고 있다. 결정적 서비스 방식은 무손실 서비스를 위한 것이며, 한정된 유한개의 정보를 모니터링한 후 트래픽을 특성화한다. 단점으로는 억제 함수의 복잡도가 높다.

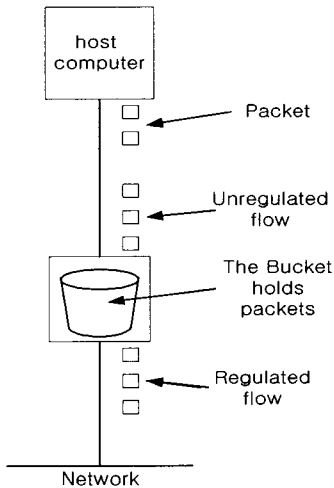
3. 트래픽 정형화

폭주 트래픽의 중요한 원인 중에 하나는 버스티하다는 것이다. 그러므로 네트워크내에서 이러한 트래픽의 여러 특성을 원하는 형태로 변경해주어야 한다. 이러한 일련의 방법을 트래픽 정형화(traffic shaping)라 한다. 여기에는 버퍼링, 스페이싱(spacing), 스케줄링(scheduling) 등 여러 가지가 있다. 이 중 본 논문에서는 버퍼링의 방법 중 리키-버킷 알고리즘과 토큰-버킷 알고리즘을 사용하였다.

3.1 리키-버킷 알고리즘

리키-버킷 알고리즘(Leaky-Bucket Algorithm)은 네트워크의 평균 허용 속도에 대하여 감시를 하는 리키 버킷과 가상 리키 버킷으로 정의하며, 평균 셀 속도와 최대 셀 속도를 동시에 고려하는 가상 리키 버킷으로 나눌 수 있다. 리키 버킷 알고리즘들에서 셀은 단지 버킷으로 전송되어 이 버킷을 통과하여 전송된다. 즉, 여러 호스트들이 연결되어 있거나 또는 빠른 셀이 들어와도 버킷은 항상 일정한 출력만을 전송하게 된다. 만일 버킷의 용량이상이 입력되었

을 경우 패킷은 버려진다. [그림 1]은 매 셀 도착마다 각각 하나씩 실어서 셀을 전송하는 가장 일반적인 방법을 보여주는 리키 버킷 그림이다. 그림에서 입력되는 패킷은 버킷에 쌓이게 된다. 그리고 버킷을 지난 패킷들은 입력 속도와는 상관없이 항상 일정한 속도로 출력되어 전송된다.

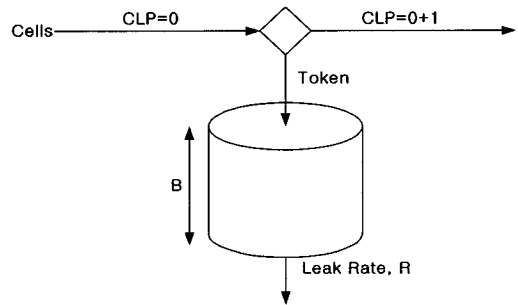


[그림 1] 리키 버킷 알고리즘

[Fig. 1] Leaky-Bucket Algorithm

그리고 연속-상태 리키-버킷 알고리즘(Continuous-state Leaky-Bucket Algorithm)은 크기가 B 인 버킷에서는 고정 전송률 R 로 이동하는 토큰이 포함되어

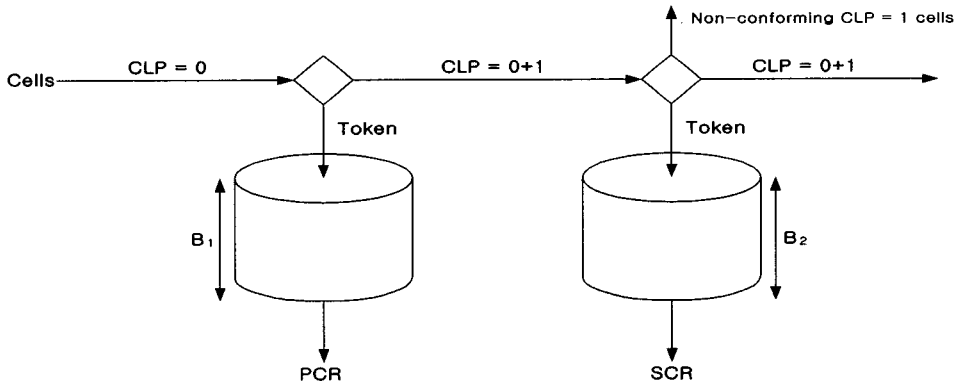
있다. 셀이 도착했을 때 버킷의 크기가 B 보다 작으면 이 셀은 순응 셀이 되고, 이 버킷에 토큰이 추가된다. 셀이 도착했을 때 버킷이 가득 차있으면 이 셀은 비 순응 셀이 되며, 이 셀은 버려지거나 또는 태그가 붙여져서 네트워크로 전송된다. 비 순응 셀은 버킷에 토큰이 추가되지 않는다. [그림 2]는 연속-상태 리키-버킷 알고리즘의 흐름을 보여주고 있다, 전송률 R 은 소스에 대해 협전된 전송률이며, 버킷 크기 B 는 소스가 전송률 R 로 전송할 수 있는 최대 버스트의 크기이다.



[그림 2] 연속 상태 리키버킷 알고리즘의 흐름

[Fig. 2] Flow of the Continuous-state Leaky-Bucket Algorithm

이중 리키-버킷 기법은 첫 번째 버킷에서의 최대 셀 전송률과 두 번째 버킷에서의 유지 가능한 셀 전송률을 이루기 위해 사용될 수 있다. 이중 가상 스케줄과 마찬가지로, 비 순응 셀들은 첫 번째 버킷에 데



[그림 3] VBR 트래픽에 사용된 이중 리키버킷 알고리즘

[Fig. 3] Using double Leaky-Bucket Algorithm in VBR Traffic

그가 붙는다. 만약 첫 번째 버킷에 태그가 붙은 셀이 두 번째 버킷에서 비 순응이라면 버려진다. 그렇지 않다면 태그는 변하지 않은 채 네트워크로 전송된다. 또한, 만약 첫 번째 버킷에서 순응 셀이 두 번째 버킷에서 비 순응이라면 여기에 태그가 붙고 네트워크로 전송된다. B1, B2 크기의 버킷을 선택함으로써 원하는 셀 지연 변이를 얻는다.

3.2 토큰-버킷 알고리즘

리키-버킷 알고리즘은 버스티 트래픽에 상관없이 항상 일정한 패킷의 출력력을 갖는다. 많은 응용 부분에서 큰 버스트가 도착했을 때 데이터의 손실 없이 이를 해결해줄 수 있는 것이 필요하게 되었다. 그래서 토큰 버킷 알고리즘이 만들어 졌다.

토큰 버킷 알고리즘 (Token Bucket Algorithm)은 일정한 시간 ΔT 마다 계속해서 토큰을 생성시킨다. 이를 토큰 rate라하며 R이라 표시한다.

[그림 4]는 토큰-버킷 알고리즘이다. 여기서 셀은 단지 토큰-버킷으로부터 토큰을 받을 수 있을 때만 전송된다. 토큰은 네트워크의 평균 허용 속도 R로 생성되고 토큰-버킷에 저장된다. 버킷은 B, 즉 토큰 depth라는 유한한 크기를 가지는 양으로 표시된다. 그러므로 셀은 최대 B 만큼의 양이 한번에 전송 가

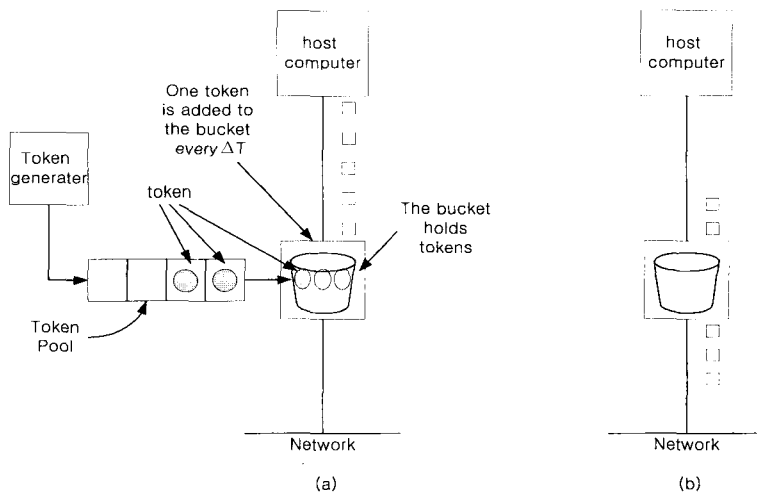
능하다. 이를 다시 말하면 B는 최대 허용 가능한 버스트 길이로서 생각될 수 있다. 토큰-버킷이 가득 채워지면 이외의 다른 토큰은 버려진다.

[그림 2]의 리키-버킷과 [그림 4]의 토큰-버킷은 몇 가지 다른 traffic shaping을 가지고 있다. 리키-버킷 경우는 host로부터의 입력이 없을 경우 나중의 큰 버스트를 위해 버퍼링을 계속하지 않는다. 반면 토큰-버킷의 경우는 버킷의 최대 용량까지 버퍼링을 한 후 n개의 패킷이 버스티 되었을 때 버퍼링 한 양만큼 한 순간에 출력력을 낼 수 있다. 그러므로 빠른 전송 또한 가능하다. 또 다른 차이점은 버킷의 최대 용량 이상의 입력이 들어왔을 시 토큰-버킷의 경우는 입력될 패킷을 절대 버리지 않는다. 그러나 리키-버킷의 경우는 패킷을 버리게 된다. 대부분의 VBR 트래픽 제어에서 리키-버킷을 사용해 왔으나 본 논문에서는 더 효율적인 토큰-버킷 알고리즘을 사용하였다.

식 (1) 토큰-버킷에서의 최대 버스트율을 나타낸다.

$$C + \rho S = MS \tag{1}$$

- 여기서 C : token bucket capacity (bytes)
- S : burst length (sec)
- ρ : token arrival rate (bytes/sec)
- M : maximum output rate (bytes/sec)
- MS : the number of bytes in maximum-speed burst of length S seconds



[그림 4] 토큰-버킷 알고리즘
[Fig. 4] Token-Bucket Algorithm

이 모델은 IETF IntServ (Internet Engineering Task Force integrated Services), RSVP (ReSerVation Protocol), CR-LDP (Constraint Routing-Label Distribution Protocol) 등에서 Traffic Specification (Tspec)을 기술하는데 사용되고 있다.[14~16] 이 모델에서 Tspec은 5 가지 파라미터로 구성되며, R (token bucket rate), B (token bucket depth), p (peak bandwidth), m (minimal policed size), M (maximum policed size)으로 이루어진다. 여기서 R과 B는 트래픽 특성화에 있어서 가장 중요한 파라미터이다. 이들은 서로 상관 관계가 있다. $R > B$ 일 때는 속도 우세, $B > R$ 일 때는 버퍼 우세 구간이라 한다. 버퍼 우세 구간에서는 리키 버킷을 사용하게 되면 큰 문제는 없다. 반면 속도 우세 구간에서는 리키 버킷을 사용하게 되면 패킷의 손실을 우려해야 한다. 그러므로 속도 우세 구간인 네트워크에서는 토큰-버킷을 사용하여야 한다.

4. 동적 VBR

이 방식은 Peak Rate (PR)로 서비스를 시작하여 주기적으로 자원을 모니터링 한 후 누적 함수에 맞추어 버퍼링을 하는 방식이다. 이는 네트워크 영역을 효율적으로 관리하기 위해 버퍼의 크기를 늘리면서 속도는 낮출 수 있다. 여기서 Peak Rate는 ATM 커넥션의 최대 속도를 규정하는 파라미터인 Peak Cell Rate (PCR)와 흡사한 성질을 가진다. PCR은 ATM의 전송 매체인 Cell의 수를 규정하는 반면 PR는 영상 프레임의 전송 Bytes의 양을 규정한다. 즉 시간당 전송되는 Bytes의 양을 규정한다. 단위는 Bytes/시간이다.

본 논문에서는 처음 모니터링은 예측 기법을 사용하여 PR로 버퍼의 크기를 할당해 준 뒤에 이 후의 버퍼는 실제로 입력된 영상 프레임의 크기를 측정 한 후 다시 버퍼를 할당해주는 방식을 사용하였다. Knightly와 Zang는 Deterministic Binding Interval Dependent (D-BIND) 트래픽 모델을 제시하여 n-segment linear function 을 만드는 rate-interval 쌍 $\{(R_j, I_j) \mid 1 < j < n\}$ 의 합으로 비디오 트래픽을 특성화시킨다.[8] D-BIND 모델은 정확히 VBR 비디오 트래픽을 특성화시킬 수 있으나, 파라미터 선택의 문제에 대해 언급하지 않았으며 그 복잡도가 $O(n^2)$ 이

다. Dallas E. Wrege와 Jorg Liebeherr는 D-BIND 모델을 변형시켜 억제함수를 구했지만 복잡도는 $O(kn)$ 으로 여전히 높다.[9]

본 논문에서는 누적 제한 함수를 이용하여 결정적 방식의 문제점인 복잡도를 $O(n)$ 으로 낮추며, 실제 영상 소스에 가깝게 억제 함수를 만든 후 토큰-버킷 알고리즘에 매핑하는 연구를 하였다.

누적 제한 함수는 Tspec의 변경 시점을 판단하기 위해서는 모든 프레임 부분합 (SS)을 이용하는 누적 제한 함수를 통해 얻어진 곡선을 따라 그 변화에 알맞게 토큰-버킷의 rate (R)과 depth (B)를 매핑시킨다.

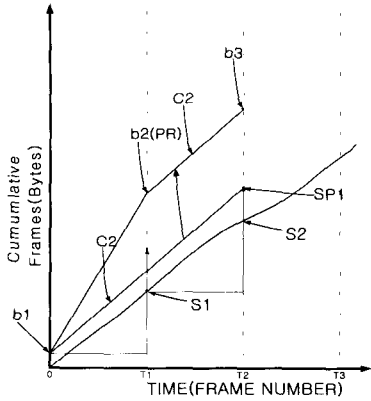
누적 제한 함수는 두 가지 특징을 가지고 있다. 하나는 초기 peak-rate로 Tspec을 기술해 주기 때문에 연결 요청자가 트래픽 특성화를 할 때 요구되는 시간 비용이 없다는 점이다. 두 번째 특징은 Tspec을 전송 중에 네트워크 자원 사용 효율을 높이는 방식으로 바꾼다는 점이다. 전자는 지연에 민감한 트래픽의 경우 선호되는 특성이고, 후자의 경우는 네트워크 사용 효율을 높일 수 있다는 것이다.

트래픽 변화의 경향은 적어도 수십에서 수백 개의 프레임 정보가 있어야 한다. 반드시 프레임 단위가 아니라도 상위 레벨에서 네트워크 레벨로 넘겨지는 단위의 데이터들도 마찬가지로 할 수 있다.

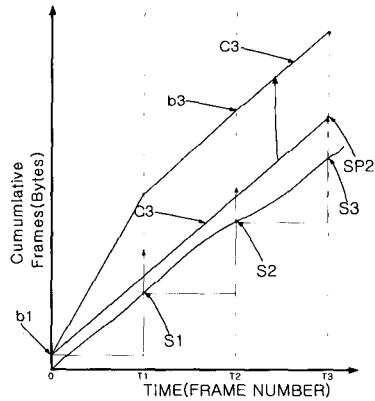
[그림 5](a)의 첫 구간에서 시작 포인트 b_1 은 처음 입력 프레임 값이며, 그리고 $0 \sim T_1$ 까지는 PR값으로 버퍼링해 준다. 다시 말하면 b_1 에서 시작하여 PR만큼의 버퍼를 할당한다. 버퍼링된 값은 b_2 이며 그 함수를 C_1 으로 표시한다. 그리고 입력된 프레임의 합은 S_1 으로 표시한다.

$$P_{peak} = \text{Max 프레임} / \text{interframe 타임} \quad (2)$$

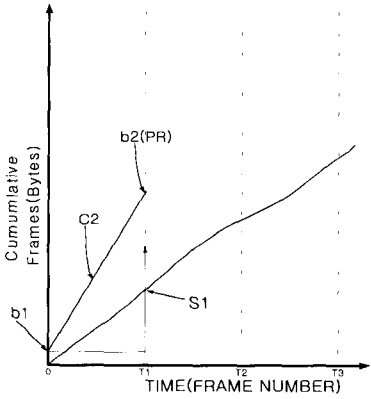
여기서 interframe 타임은 프레임 헤더에 명시되며 일반적으로 0.0388sec를 가진다.



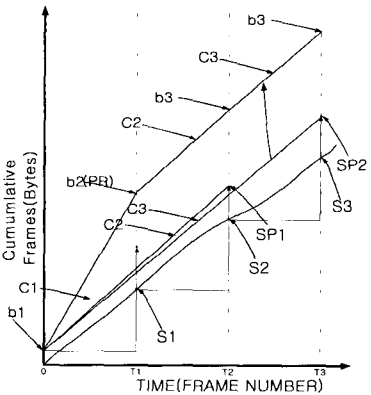
(a) 시간 0~T1
(a) time 0~T1



(d) 시간 0~T2
(d) time T1~T2



(b) 시간 T1~T2
(b) time T1~T2



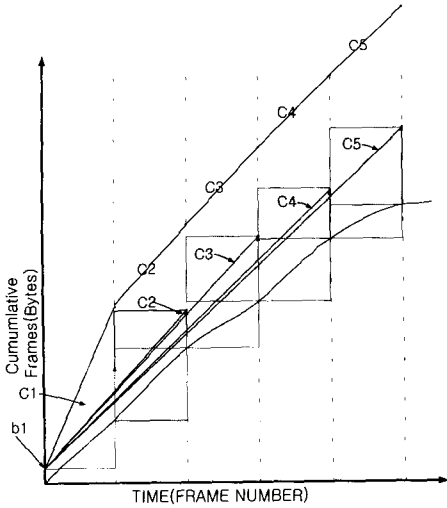
(c) 시간 T2~T3
(c) time T2~T3

[그림 5] 누적 함수

[Fig. 5] Cumulative Function

[그림 5](b)는 T1~T2 동안은 S1값에 최대 프레임 값을 합하여 첫 번째 누적 포인트 SP1을 만든다. 그리고 b1에서 SP1까지의 함수 C2를 만든 후 0~T2까지의 함수를 T1~T2까지의 구간으로 이동시킨다. 여기서 이 C2는 다음의 식 (3)로 구할 수 있으며, 이 값은 current rate이다. 그리고 이동된 함수의 시작 포인트는 b2라 표시하였고 그 값은 아래의 식 (4)로 구할 수 있으며 그 값은 토큰 depth이다. 그리고 실제 입력된 프레임의 합은 S2로 표시한다.

[그림 5](c)는 T2~T3 동안은 역시 S2값에 최대 프레임 값을 합하여 두 번째 누적 포인트 SP2를 만든다. 그리고 b1에서 SP2까지의 함수 C3를 만든 후 0~T3까지의 함수 C3를 T2~T3까지의 구간으로 이동시킨다. 물론 시작점은 식 (4)를 이용하여 b3 구한 후 연결시킨다. 위의 방법으로 계속되는 누적 프레임 함수를 나타낸다.



[그림 6] Cumulative Function의 합
[Fig. 6] Sum of Cumulative Function

따라서 [그림 6]는 영상소스의 누적 그래프와 그 위를 지나는 누적 제한 함수 (Cumulative Constraint Function : CCF)를 나타낸다. [그림 6]에서와 같이 기술기의 변화 시점은 매 부분마다 합해주며, 그 변화량은 그 다음 부분합까지 연속적으로 최대 프레임 (Max_Fame)이 들어온다는 가정 하에 기존의 누적 프레임의 정보에 합하여 변화량을 결정한다. 즉 [그림 6]에서 회색 사각형은 다음 누적 함수동안 연속적으로 최대 프레임이 들어왔을 경우를 나타내며, 아래 사각형은 실제 영상 소스를 나타낸다. 그리고 여기서 C1은 Peak Rate를 나타내며 b1은 초기의 토큰 Depth를 나타낸다. 그리고 마지막으로 위 함수를 실 함수에 가깝게 접근하기 위하여 α 라는 adjustment constant를 사용하였다.

$$C_j = \frac{\sum_{i=1}^j S_i + Peakrate * T_j - b_1}{t_j} \quad (3)$$

$$b_j = (C_{j-1} - C_j) * t_j * \alpha + b_{j-1} \quad (4)$$

$$CCF = \{ C_i * t_{S_i} + b_i | i \geq 1, t_i \geq 0 \} \quad (5)$$

- 여기서 C1=Peak Rate
- b1=Initial bucket depth
- j ≥ 2 Cj : jth current rate
- Sj : jth total byte of frames

- tj : total amount of time until Sj
- Tj : interval of Sj
- tSi : time of Si
- I : interframe time
- bj : token depth
- α : adjustment constant (0.13)

식 (3)은 각 프레임 부분합의 변화 시점에서 현재까지의 프레임 정보($\sum_{i=1}^j S_i$)와 다음 프레임 부분합을 만나기 전까지 연속적으로 최대 프레임이 들어왔을 경우의 프레임 양 (Peak rate * T)의 합을 총 누적 시간으로 나누어 토큰 속도를 결정한다. 결정된 토큰 속도에 따른 토큰 버킷의 크기 (b)를 구하기 위한 식은 (4)이다. 누적 제한 함수는 초기 peak-rate로 전송을 시작하기 때문에 처음은 실제 프레임 정보보다 높은 그래프를 그리나, 거시적으로 보았을 경우 평균 속도로 낮추어 가는 것을 실험 결과 알 수 있다. 식 (3), (4)에 의해 결정된 부분 선형의 (Cj, bj) 쌍들에 의해 식(5)와 같이 CCF가 만들어진다. CCF는 초기 peak-rate로 전송을 시작하기 때문에 처음 프레임 부분합은 실제 프레임 정보보다 높은 그래프를 그리며, 네트워크 자원을 과대하게 사용하게 된다. 초기 네트워크 자원의 과대 사용은 계속적으로 프레임 정보에 쌓이기 때문에 이 문제를 해결하기 위하여, adjustment constant ($\alpha:0.13$, 보정상수)를 식 (4)에 적용하여 전체적 CCF 함수를 소스 트래픽에 접근하게 함으로써, 실제 소스의 변화에 민감하게 대처하며 정확도를 높인다. 강제 함수에서 나온 부분 선형의 값은 토큰-버킷 파라미터 (rj, bj)는 (Cj, bj)로 매핑이 된다. 즉 위 억제 함수에서 나온 부분 선형의 값은 곧 토큰-버킷의 rate값 R과, 토큰-버킷의 depth값 B로 매핑된다.

5. 시뮬레이션 및 결과

5.1 시뮬레이션 준비

VBR의 비디오 트래픽 예로써 압축 Moving Pictures Expert Group-1 (MPEG-1) 알고리즘을 사용하였다. MPEG 소스 트래픽은 일반적으로 큰 버스트와 셀 도착에 있어서 높은 상관성을 갖는다. VBR 전송 모드에서 전송률은 변동이 심하다. 또한 프레임의 타입에 따라서도 그 크기의 변화가 심하게 된다.

MPEG 부호화기 (MPEG Encoder)는 3가지 I, P, B 프레임을 생산한다. 이중 I (Intra coded) 프레임은 프레임내 (intraframe) 압축에 사용되며 움직임 보상을 이용하지 않고 그 프레임만을 부호화한다. 그리고 P (Predicted coded) 와 B (Bidirectionally predicted coded) 프레임은 I 프레임 사이에 전송되며 프레임간 (interframe) 압축에 사용되며 움직임 보상을 한 후에 부호화를 한다. 그러나 방법에는 약간의 차이가 있다. I 프레임은 이전의 프레임과 독립적이므로 손실이 발생했을 경우 복구가 어려운 반면에 P 프레임의 경우는 이전의 프레임과 연관성이 있어서 이전의 프레임이 완전하다면 어느 정도의 손실은 복구할 수 있다. 부호화는 Group of Picture (GOP)단위로 이루어진다.

본 논문에서는 트래픽의 측정 단위는 아래와 같은 이유에서 프레임보다는 슬라이스를 이용하였다.

첫째, 발생된 셀 수에 대한 Probability Mass Function (PMF)가 더 좋은 특성을 갖는다.

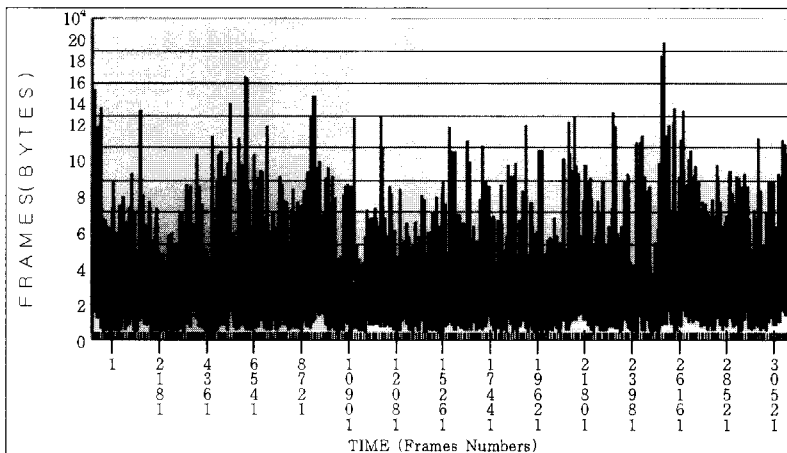
둘째, 슬라이스 데이터는 시·공간적으로 상관성을 분리할 수 있다.

셋째, 셀 손실이 발생할 경우 슬라이스가 독립적으로 패킷화되기 때문에 수신측에서 재동기화를 빠르게 수행할 수 있다

본 논문의 시뮬레이션을 위해 UC Berkeley MPEG-I software encoder에 의해 코딩된 여러 영상 소스 (TV sports, news, movie etc)[17]를 사용하였다. 모든 영상 소스는 아래와 같이 코딩되었다.

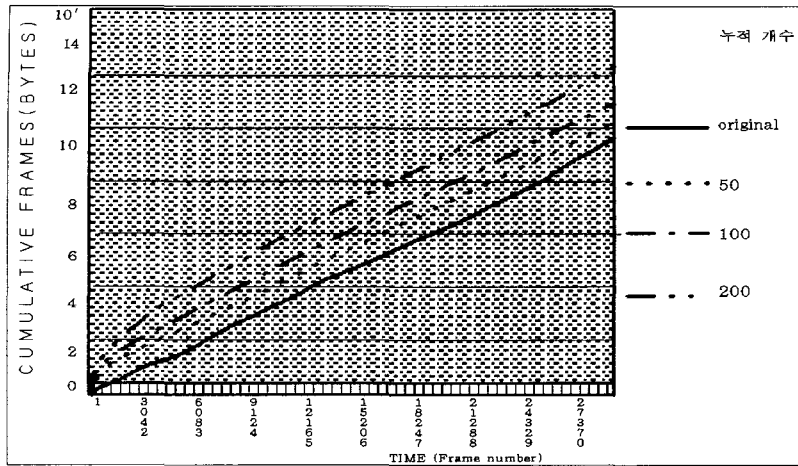
- 각 프레임은 하나의 슬라이스로 되어있다.
- GOP pattern : IBBPBBPBBPBB (12 frames)
- Quantizer scales :10(I), 14(P), 18(B)
- 각 시퀀스 당 프레임수 : 40000(약 30분가량의 영상)

영상 소스의 원본 프레임 용량을 나타낸 그래프는 [그림 7]이며 동적 VBR에서의 누적 제한 함수 중 보정 상수를 사용하지 않은 그래프는 [그림 8]이다. 그리고 보정 상수를 사용하여 이상적인 누적 제한 함수를 나타낸 그래프는 [그림 9]이다. 마지막으로 영상 소스에 대한 시뮬레이션을 통하여 얻은 토른 레이트를 [그림 10]에서 나타낸다.



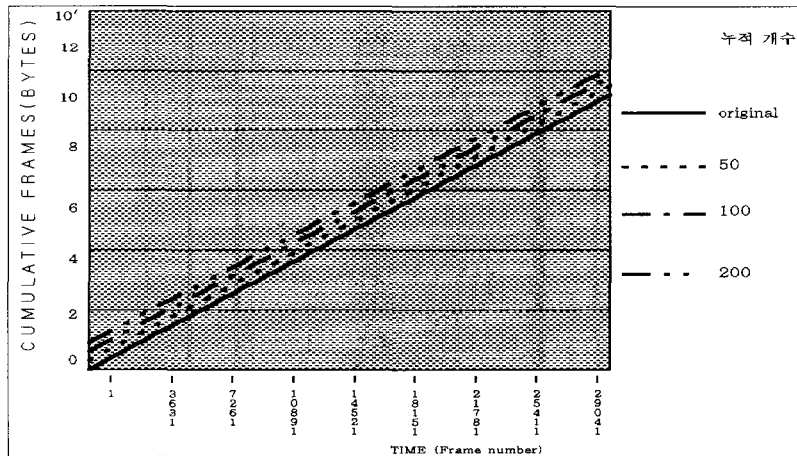
[그림 7] 영상 소스의 원본 프레임

[Fig. 7] Original Frames of video source



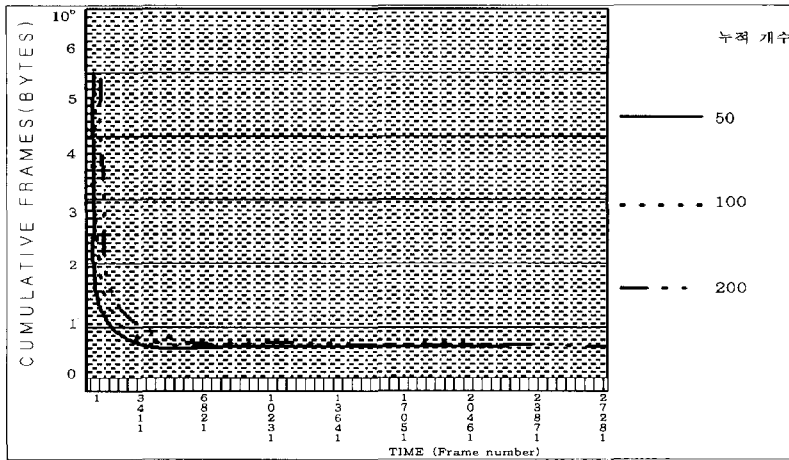
[그림 8] 영상 소스의 누적 함수(보정 상수 제외)

[Fig. 8] Cumulative Function of video source(except Adjustment constant)



[그림 9] 영상 소스의 이상적 누적 함수

[Fig. 9] Ideal Cumulative Function of video source



[그림 10] 영상 소스의 토큰 레이트
 [Fig. 10] Token rate of video source

5.2 결과 및 고찰

VBR로 기술된 트래픽에서 평균 속도 r 값만으로는 실제 소모되는 네트워크 자원을 확실히 기술할 수 없어 EB (Efficient Bandwidth) 개념이 쓰인다.[18] 간단한 계산을 위해 Lucent의 CAC (Call Admission Control)에 적용된 손실이 없는 다중화시 적용된 EB 계산식을 사용하여 CBR, aVBR (Average VBR), CCF에 의한 VBR (ssVBR)의 네트워크 자원 예약량을 비교하여 CCF의 효율성을 측정하였다.[18],[19] EB 계산식을 소개하면 다음과 같다. 토큰 버킷 depth b 값 (B_T)은 네트워크 내 스위치의 입력단 buffer 요구량으로 이해될 수 있고 스위치 입력 큐의 길이 (B)와 스위치 처리 능력 (C)에 따라 $B_T(B/C)$ 만큼의 대역폭을 우선적으로 할당받게 되는데, 토큰 버킷 rate 값 ($r=C$)이 크거나 작은 경우에 따라 식 (6)과 같은 EB값이 된다.

$$EB = \begin{cases} \frac{PR}{1 + \frac{B/C}{B_T}(PR - CR)} & \text{if } CR \leq \frac{B_T}{B/C} \\ CR & \text{if } \frac{B_T}{B/C} \leq CR \leq PR \end{cases} \quad (6)$$

peak rate로 서비스할 때에 비교하여 CCF방식의 네트워크 자원 절약능력을 측정하기 위해서는 CCF 방식이 $rate(C_j)$ 를 변경시키는 시점에 대한 적분치 비교를 이용한다. 네트워크 자원 사용치 R 은 아래 식 (7), (8), (9)와 같이 계산된다.

$$R_{CBR} = Peak\ rate * T_{total} \quad (7)$$

$$R_{CCF} = \sum_{j \in J} [EB(j) * T(j)] \quad (8)$$

$$R_{aVBR} = EB * T_{total} \quad (9)$$

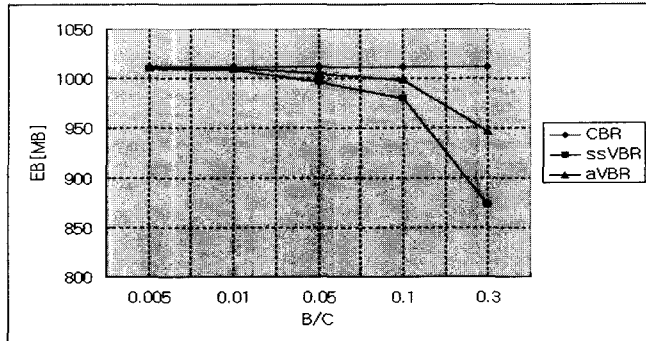
where T_{total} : total runtime

$$EB(j) = C \rightarrow C(j), B_T \rightarrow B_T(j) \in \quad (10)$$

$T(j)$: runtime with j -th(C, B_T)

VBR과 CCF를 이용한 VBR의 자원 예약량을 비교하기 위해 아래와 같은 두 가지 전송 서비스 경우를 고려한다.

- CBR : peak rate 전송, $b = peak\ rate/l$
- aVBR : 전송 속도 r 은 전체 트래픽의 평균 속도 값을 갖고, b 값은 그 r 값에 대한 최적 속도를 갖는 VBR 서비스



[그림 11] 영상 소스의 EB값 비교

[Fig. 11] Compare EB of video source

앞의 식 (6), (7), (8), (9)를 토대로 수치를 분석하였다. 시뮬레이션은 사용된 자료의 전체 프레임이 40000개이고, B/C값을 5가지로 나누어 측정하였다.

그 결과 <표 1>과 같이 구할 수 있다. 그리고 [그림 11]는 <표 1>에 의한 영상 소스의 EB값을 그래프로 나타낸 것이다. 그림과 같이 CCF를 이용한 VBR 방

<표 1> B/C 값과 SS 값에 대한 네트워크 자원 예약량 [MB]

<Table. 1> Network source value of B/C and SS[MB]

B/C Source		0.005	0.01	0.05	0.1	0.5
asterix	CBR	737	737	737	737	737
	ssVBR	735	734	721	706	605
		736	735	729	722	667
		734	734	7229	724	685
aVBR	737	736	735	733	716	
silence of lamb	CBR	671	671	671	671	671
	ssVBR	670	668	661	651	581
		670	670	665	659	614
		669	668	665	660	626
aVBR	670.7	670	667	663	635	
m_tv	CBR	1257	1257	1257	1257	1257
	ssVBR	1254	1252	1235	1214	1068
		1256	1254	1245	1232	1143
		1253	1252	1245	1236	1169
aVBR	1256	1256	1253	1249	1217	
news	CBR	949	949	949	949	949
	ssVBR	948	946	932	916	802
		948	947	939	930	862
		946	945.6	940	933	883
aVBR	948.6	948	942	935	879	
race	CBR	1012	1012	1012	1012	1012
	ssVBR	1010	1009	996	980	873
		1011	1010	1003	994	926
		1009	1008	1003	996	946
aVBR	1011	1011	1005	998	947	

<표 2> 프레임 부분합 단위별 Loss Rate(%)

<Table. 2> Loss Rate of Sub-sum(%)

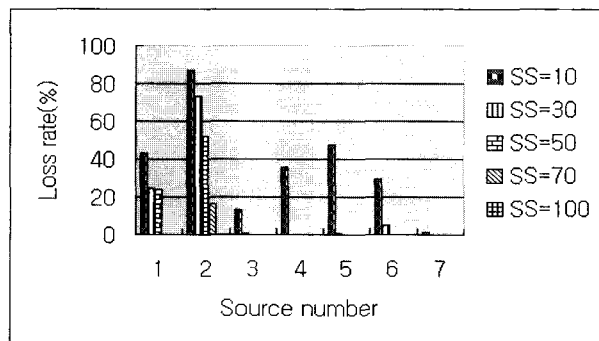
SS Loss(%)	10	30	50	70	100	200
asterix	43	24.2	23.7	0	0	0
movie 1	0	0	0	0	0	0
movie 2	87	73	51.7	16	0	0
lambs	0	0	0	0	0	0
m_tv	13.2	0.6	0	0	0	0
news	0	0	0	0	0	0
race	0	0	0	0	0	0
atp	0	0	0	0	0	0
kino	0	0	0	0	0	0
fuss	0	0	0	0	0	0
Mr.bean	0	0	0	0	0	0
mtv	35.6	0.06	0	0	0	0
super bowl	47.2	0.58	0	0	0	0
simpsons	0	0	0	0	0	0
talkshow	0	0	0	0	0	0
talkshow2	0	0	0	0	0	0
term	0	0	0	0	0	0
video	0	0	0	0	0	0
soccer	29.3	5.2	0	0	0	0
starwar	0.95	0	0	0	0	0

식은 기존의 CBR과 aVBR 방식에 비해 네트워크 자원이 절약됨을 알 수 있다. 또한 B/C 값이 커질수록 그 수치가 낮아짐을 알 수 있다.

<표 2>는 프레임 부분 합 단위 별 loss rate를 나타낸 것이다. 이 표를 보면 SS를 10~70 사이의 값으로 결정하였을 경우 부분적으로 loss가 일어나는

것을 볼 수 있다. 이와 함께 SS의 크기가 증가함에 따라 loss 또한 현저히 줄어드는 것을 볼 수 있다. (SS=70 이후 loss는 0)

[그림 12]는 21개 소스 중 loss가 발생한 7개 소스 (1: asterix, 2: james bond, 3: MTV 4: MTV II, 5: super bowl, 6: soccer, 7: starwars)에 대한 SS별



[그림 12] 소스별 Loss Rate(%)

[Fig. 12] Loss Rate of each source(%)

loss rate를 표로 나타난 것이다. 위 그림에서와 같이, loss가 발생한 7개 소스는 SS가 10일 때 가장 큰 loss rate를 발생시키며, SS가 작아짐에 따라 급속도로 loss rate가 감소하는 것을 알 수 있다.

10~70 사이의 프레임 정보는 CCF를 결정하기에는 부족한 양임을 알 수 있다. 이는 CCF 자체가 속도 결정 시 통계적 방식에 기반을 둔 확률적인 접근을 함에 기인한다. 즉 다음 SS까지의 최대 프레임 정보를 가정하고 계산하는 중 부족한 프레임 정보로 인하여, 충분한 buffer를 할당함으로써 발생한다.

이러한 실험 결과 여타 다른 방식의 CBR과 VBR보다 CCF가 네트워크 자원을 절약할 수 있으며, CCF를 만족시키기 위해서는 최소 70 프레임 이상의 정보가 필요하다는 것을 알 수 있다. 또한 CCF는 여타 소스에 비해 news 및 토크쇼와 같은 소스에 잘 적용된다.

[그림 8]~[그림 10]은 영상 소스에 대한 강제 함수와 토크 rate의 변화를 나타낸다. SS를 50, 100, 200으로 변화시켜 가면서, 구한 강제 함수를 실험을 통해 보면, 프레임 부분 합을 적게 결정할수록 실 영상 그래프에 가까워지는 것을 볼 수 있다. 또한 토크 rate의 변화는 프레임 부분합의 단위가 클수록 그 변화가 적으면서 평균 속도에 천천히 접근해감을 알 수 있다. 이 외에 21개의 서로 다른 소스에 적용해 본 결과 모든 소스에 CCF가 적용 가능하며, EB값이 여타 방식에 비해 절감되었음을 알았다.

6. 결론

지금까지 VBR 트래픽 특성화에 관해 나온 많은 결정적 방식의 가장 큰 문제는 계산량이 많고, 강제 함수를 만들기 위해서는 많은 프레임 정보가 선행되어 필요하다는 문제점을 가지고 있었다. 본 논문에서는 peak rate로 전송을 시작함으로써, 정교한 admission control function을 간단히 하여 연결시간을 단축시킨다. 동적 VBR방식은 소스의 특징에 맞추어 갱신 시점을 자동으로 변화하며 토크 버킷 parameter들을 정의해 준다. 갱신을 통하여 망 자원 예약량인 effective bandwidth를 감소 시켜준다. 그리고 처음 SS를 만날 때, 그 전까지의 프레임 정보를 가지고, rate를 변경시키는 시도를 하였다. 이는 프레

임의 모니터링을 거치도 않아도 되며 모니터링으로 인한 오버헤드도 거의 없다. 그 결과 모든 영상 소스에 특정 프레임 정보 (100프레임 이상)만 가지고 적용이 가능하다는 것을 알 수 있었다. 시뮬레이션 결과에서 동적 VBR은 CBR보다 B/C 값에 따라 3~34%까지, aVBR보다는 2~28% 정도의 망 자원을 절약할 수 있음을 보였다. 또한 SS 단위를 줄임으로써, 실제 프레임 합수에 근접할 수 있음을 알 수 있었다. 그러나 SS이 너무 작을 경우 loss가 생기며 자주 rate를 변화시켜야 하는 단점이 동시에 생긴다. 프레임에 따라 B/C값과 SS의 값의 적절한 조합으로 최상의 강제 누적 함수를 구할 수 있었다.

그리고, 앞으로는 강제함수를 통한 토크-버킷에 매핑시 buffer에서의 서스펜스 현상을 줄여주며, SS값과 상관없이 적용될 수 있는 강제 함수의 연구와 적합한 수치의 자동적인 변화가 필요하다고 사료된다.

※ 참고문헌

- [1] Uyless D Black "ATM VOLUME I' Foundation for Broadband Networks' ", Prentice Hall, 1999.
- [2] Raif O. Onvural " Asynchronous Transfer Mode Networks: Performance Issues" Artech House, Inc. 1994.
- [3] David E.McDysan and Darren L.Spohn, "ATM Theory and Application", McGraw-Hill, 1995.
- [4] The ATM Forum "ATM Traffic Management Specification Version 4.0", Apr. 1996. Available Through <ftp://ftp.atmforum.acom/pub/approved-specs/aftm-0056.000.ps>
- [5] D.reiniger, D.Raychaudhuri, "Statistical Multiplexing of VBR MPEG compressed Video on ATM networks" IEEE INFOCOM'93, Vol.2, pp.919-926,1993.
- [6] Pramod Pancha, Magda El Zarki, " A look at the MPEG video coding standard for variable bit rate video transmission" IEEE INFOCOM'92, pp.85-94,1992

- [7] 김영범 "ATM 망의 트래픽 제어 기술 및 현황" 한국 통신 학회지 제 15권 제 3호 pp.116-123, 1998년 3월.
- [8] E. Knightly and H. Zhang, "Traffic characterization and switch utilization using deterministic bounding interval dependent traffic models", In Proc. IEEE INFOCOM95, pages 1137-1145, April 1995.
- [9] Dallas E. Wrege and Jorg Liebeherr, "Video Traffic Characterization for Multimedia Networks with a Deterministic Service", Proc. IEEE INFOCOM 96 pp.537-544, March 1996.
- [10] Puqi Perry Tang, Tsung-Yuan Charles Tai, "Network Traffic Characterization Using Token Bucket Model", Proc. IEEE INFOCOM 99 Vol.1. 51-62, March 1999.
- [11] Andrew S.Tanenbaum " Computer Networks, Third Edition" pp.379-384. 1996
- [12] P.Pancha and M.El Zarki. "Bandwidth allocation schemes for variable-bit-rate MPEG sources in ATM networks", IEEE trans. Circuits Syst. Video Technol. vol.2 .pp 19093, June 1993.
- [13] Jean-lien C. wu, Chia_chang shiu, Traffic Modeling and Bandwidth Allocation for MPEG Video sources in ATM Networks, IEEE Grobecom, pp.902-909.1993.
- [14] J.wroclawski, The Use of RSVP with IETF Integrated Services, RFC 2210, Network Working Group, September.
- [15] B.Branden, L. Zhang, s. berson, S Herzog, and S. Jamin Resoure Resrvation Protocol (RSVP)-Vrsionl Functional specification, RFC 2205, September 1997.
- [16] Bilel Jamoussi, Constraint-Based Lsp setup using LDP, Internet draft, IRTF MPLS working Group, availble at <http://www.ietf.org/internet-draft-ietf-mpls-cr-10-03.txt>.
- [17] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems", Technical Report 101, Institute of Computer Science, University of Wurzburg, February 1995. The traces used in this paper are available via anonymous ftp from the site <ftp-info3.informatik.uni-wuerzburg.de> in the directory /pub/MPEG/.
- [18] A. Elwalid and D. Mitra, Effective bandwidth of general Markovian traffic sources and admission control of high speed networks, IEEE/ACM Trans. On Networking, vol. 1, no. 3, pp. 329-343, June 1993.
- [19] Dapeng Wu and H. Jonathan Chao, "Efficient Bandwidth Allocation and Call Admission Control For VBR Service Using UPC Parameters", Proc. IEEE INFOCOM 99, Vol.3.
- [20] Marwan Kruz " Bandwidth Allocation Strategies for Transporting Variable-Bit-Rate video traffic. IEEE Communications Magazine, January 1999
- [21] Zhi-Li Zhang, Jim Kurose, et al Smoothing, "Statistical Multiplexing, and Call Admission Control for Stored Video", IEEE Journal on selected areas in communications. VOL.15, August 1997.
- [22] E.W. Knightly, "H-BIND: A New Approach to Providing Statistical Performance Guarantees to VBR Traffic", Proc. IEEE infocom'96, April 1996.

황재철



1991년 3월 광운대학교
전자공학과 입학
1995년 2월 광운대학교
전자공학과 졸업
1998년 9월 광운대학교
전자공학과 대학원
석사과정 입학
2001년 2월 광운대학교
전자공학과 대학원
석사과정 졸업

이상원



1991년 3월 광운대학교
전자공학과 입학
1995년 2월 광운대학교
전자공학과 졸업
1995년 3월 광운대학교 대학원
전자공학과 석사과정 입학
1997년 2월 광운대학교 대학원
전자공학과 석사과정 졸업
1997년 3월 광운대학교 대학원
전자공학과 박사과정 입학
2000년 8월 광운대학교 대학원
전자공학과 박사과정 수료

조미령



1997년 2월 방송통신대(독학사)
전자계산학 학위 취득
1997년 8월~1999년 8월
광운대 일반대학원 전자공학
(석사) 졸업
2000년 3월 광운대 일반대학원
전자공학(박사) 재학

이상훈



1983년 2월 광운대학교
응용전자과 졸업(공학사)
1987년 8월 광운대학교 대학원
전자과 졸업(공학석사)
1992년 2월 광운대학교 대학원
전자과 졸업(공학박사)
1991년~현재 광운대학교
전자공학부 교수