

# 멀티미디어 응용을 위한 동적 버퍼 관리 기법\*

김 제 욱\*\*, 하 란\*\*  
\*홍익대학교 정보공학과  
\*\*홍익대학교 컴퓨터공학과

## Dynamic Buffer Management for Multimedia Applications

Jae-wook Kim\*, Rhan Ha\*\*

\*Dept. of Information Engineering, Hongik University

\*\*Dept. of Computer Engineering, Hongik University

### 요 약

컴퓨터의 성능향상과 네트워크 기술의 발전으로 제공되는 서비스도 텍스트 기반에 머물지 않고 멀티미디어 분야의 서비스로 다양해지고 있다. 특히 인터넷의 보급은 멀티미디어 응용에 분산된 데이터를 이용 가능하게 하였다. 그러나, 멀티미디어 데이터는 대용량이며 주기적인 실시간 전송을 요구하는 특성을 가지는데 기존의 네트워크는 이런 요구들을 보장하지 못한다. 따라서 이러한 네트워크 상에서 멀티미디어 데이터를 효율적으로 서비스하기 위한 버퍼 관리 기법들이 제안되었다. 기존의 제안된 버퍼 관리 기법은 크게 네트워크의 적응성을 높이기 위한 기법과 서비스 품질의 급격한 변화를 방지하여 안정적인 서비스에 초점을 둔 기법으로 분류된다. 본 논문에서는 두 가지 버퍼 관리 기법을 절충하여 안정된 서비스 품질과 네트워크에 대한 적응성을 보장하기 위한 동적 버퍼 관리 기법을 제시하고, 모의 실험을 통해 제안된 버퍼 관리 기법이 멀티미디어 응용에서 좋은 결과를 나타냄을 보인다.

### 1. 서 론

컴퓨터의 성능 향상과 네트워크 기술의 발전은 원거리의 사용자들을 위한 다양한 멀티미디어 서비스를 가능하게 하였다. 특히 데이터 압축 기술의 발전은 상대적으로 적은 대역폭으로 동영상 정보의 전송을 가능하게 함으로서 화상회의, 주문형 비디오 등 많은 응용 프로그램의 확대를 가져왔으며, 이런 추세는 앞으로 더욱 가속화 될 것이다. 이러한 멀티미디어 데이터들은 대용량이며 주기적인 전송을 필요로 하는 특징을 가지지만 기존의 TCP/IP 기반의 네트워크에서는 필요한 만큼의 대역폭을 미리 할당하거나 주기적인 전송을 보장해줄 수 없다. 따라서 기존의 네트워크 상에서 이러한 멀티미디어 데이터를 효율적으로 전송하기 위해서 데이터 버퍼링 기술이 반드시 필요하다. 이러한 버퍼 관리 기법은 크게 네트워크에 대한 적응성을 높이기 위한 버퍼 관리 기법과 서비스의 품질에 초점을 둔 버퍼 관리 기법으로 나누어 볼 수 있다.

네트워크의 적응성에 초점을 둔 버퍼 관리 기법은 변화하는 네트워크 대역폭에 대응할 수 있는 한 방안을 제공한다. MBONE에서의 화상회의 도구인 VIC[1], 버클리 대학의 Continuous Media Toolkit[2], 오레곤 공대의 Quasar's Adaptable Streaming Video Player[3] 등이 그 예인데, 이들은 모두 서비스 품질의 안정적인 보장보다는 네트워크 대역폭 변동에 따라 발생할 수 있는 지터를 제거하는 데 초점을 두었다. 필요할 경우 데이터의 품질을 결정하는 요소를 제어하여 대역폭의 요구 수준을 낮추어 동적인 네트워크 환경에 적응할 수 있도록 하는데, 경우에 따라 응용 프로그램의 서비스 품질이 급격히 나빠질 수 있다는 단점이 있다. 반면, 안정적인 서비스 품질에 초점을 둔 버퍼 관리 기법은 버퍼를 이용하여 데이터의 품질이 급격하게 나빠지는 상황을 방

지하고자 버퍼를 이용한다. 이를 위해 Feng[4][7]은 저장된 멀티미디어 데이터에 대한 정보를 미리 수집하여 데이터 대역폭의 변화가 최소화 될 수 있도록 버퍼를 이용하였으며, 이를 통해 서비스 품질의 급격한 변화를 방지함으로써 안정적인 서비스를 제공하고자 했다. 그러나, 이렇게 사전에 구해진 정보를 통해 결정된 최소 변화 대역폭의 서비스는 네트워크의 예기치 않은 지터나 환경 변화가 예상되는 동적인 환경에서는 적용하기 어렵다는 문제점이 있다.

본 논문에서는 두 가지 버퍼 관리 기법을 절충하여 네트워크 환경 변화에서도 잘 적응하며 서비스의 품질을 최대한 보장할 수 있는 동적 버퍼 관리 기법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 제시된 여러 버퍼 관리 기법들의 문제점에 대해 설명하고 본 논문에서 제시하는 동적 버퍼 관리 기법의 모델을 소개한다. 3장에서는 기존의 기법들과 본 논문에서 새로 제안된 동적 버퍼 관리 기법에 대한 실험 결과를 비교 분석하고 4장에서는 결론 및 향후 연구 과제에 대해 논한다.

### 2. 동적 버퍼 관리 기법

기존에 제안된 버퍼 관리 기법 중 네트워크의 적응성에 중점을 둔 기법은 서버에서 전송 받은 데이터를 주어진 버퍼에 선적제한 후 사용하기 시작하고, 네트워크의 변화나 지터가 발생했을 때 적재된 데이터를 사용하여 멀티미디어 데이터의 주기적인 특징을 만족 시켜 준다. 그러나, 주어진 버퍼의 언더플로우를 막기 위해 데이터의 품질을 결정하는 요소들을 제어하여 대역폭의 요구 수준을 낮추어 버퍼의 적체에 만 초점을 두어 응용 프로그램의 서비스 품질이 급격하게 낮아질 수 있는 단점이 있다. 반면 안정적인 서비스 품질에 초점을 둔 기법은 사전에 전체 데이터의 정보를 수집하고, 프레임 크기에 비해 상대적으로 낮은 요구 대역폭의 데이터 전송으로 네트워크 대역폭에 여유가 있을 때, 미리 데이터를 버퍼에 적재하였다가 높은 대역폭을 요구하는 프레임

\* 본 연구는 정보통신연구진흥원 '99 대학기초연구지원사업의 후원으로 연구되었습니다. (과제번호 : C1-1999-1234-00)

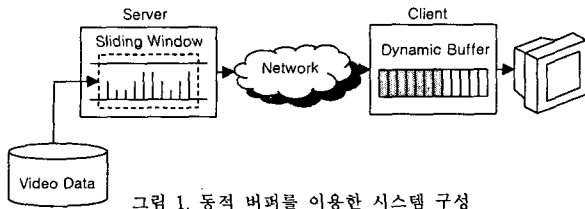


그림 1. 동적 버퍼를 이용한 시스템 구성

임을 서비스할 때, 적재된 데이터를 이용하게 하여 일정한 대역폭의 전송을 최대한 보장하고 서비스 품질의 손실을 최소화하도록 하였다. 그러나 이렇게 사전에 구해진 정보를 통해 결정된 최소 변화 대역폭의 서비스는 네트워크의 예기치 않은 지터나 환경 변화가 예상되는 동적인 환경에서는 적용하기 어렵다는 문제점이 있다.

따라서 두 가지 효과를 절충하기 위한 동적 버퍼 관리 기법은 그림 1에서와 같이 크게 서버 쪽의 슬라이딩 윈도우와 클라이언트 쪽의 버퍼로 구성하였으며 주어진 버퍼는 그림 2에서와 같이 일정 크기의 두 부분을 가상으로 나눈 후, 한 부분은 네트워크의 변화나 지터에 대비하기 위해 선적재를 하는데 이용하고, 나머지 부분은 서버내의 윈도우와 결합하여 안정적인 서비스 품질의 보장을 위해 전송 계획을 수립하는데 이용한다. 가상으로 나누어진 클라이언트 버퍼의 두 부분은 동적 네트워크에 대한 적응성과 안정적인 서비스 품질을 위해 네트워크의 상태에 따라 각각 그 크기가 변하며 상호 보완적으로 동작한다.

## 2.1. 대역폭 평활화

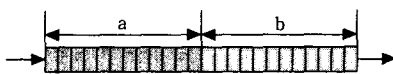
MPEG등의 표준 압축 기법에서는 프레임에 따라 크기가 다르기 때문에 전송에 필요한 요구 대역이 다양하다. 따라서 기존에 제안된 데이터 평활화를 위한 버퍼 관리 기법에서는 사전에 전체 비디오 파일의 정보를 수집하여 클라이언트에 구성되어 있는 버퍼의 크기에 따라 서버에서 전송 대역폭을 결정하여 네트워크 대역폭의 여유가 있을 때, 미리 데이터를 버퍼에 적재하였다가 큰 크기의 프레임 영역에서 소모하도록 하는 방법으로 요구 대역폭의 변화의 영향이 최소화 되도록 전송 계획을 수립하게 된다. 그러나 동적인 네트워크 환경에서 이렇게 긴 시간 동안의 전송 계획을 수립한다는 것은 불가능한 일이다. 따라서 동적 버퍼 관리 기법에서는 네트워크의 지터나 환경 변화에도 클라이언트 버퍼의 오버플로우가 일어나지 않도록 서버에서 일정 프레임 단위를 가진 윈도우를 이용하여 주기적으로 전송계획을 수립한다.

$$\forall i \leq k \leq j, \sum_{k=i}^j (frame_k - r) \leq b \quad (1)$$

서버 내의 슬라이딩 윈도우 W의 시작과 끝을 각각  $i, j$ 라 하고 클라이언트 내의 적재 가능한 버퍼의 크기를  $b$ 라고 하면 전송할 대역폭  $r$ 은 버퍼의 오버플로우가 일어나지 않게 하기 위해 (1)을 만족하는 범위 내에서 서비스 품질이 최대가 되도록 결정한다.

### 2.1.1. 슬라이딩 윈도우 평활화

멀티미디어 데이터 전송대역의 평활화를 위한 윈도우의 종류에는 크게 두 가지가 있다. 그림 3에서와 같이 각각 W프레임의 길이를 가진 윈도우가 연속적으로 서로 중첩되지 않게 평활화를 하는 홉핑 윈도우(Hopping-Window) 평활화 기법[5]과 윈도우 크기 W 보다 작은  $\alpha$  시간마다 이전 윈도우와 중첩되게 평활화 하는 슬라이딩 윈도우(Sliding-Window) 평활화 기법[5]이 있다. 홉핑 윈도우 평활화 기법의



- a : 네트워크 지터 대비 선적재 크기
- b : 평활화 전송 계획 수립에 사용할 크기

그림 2. 클라이언트 내의 버퍼 구조

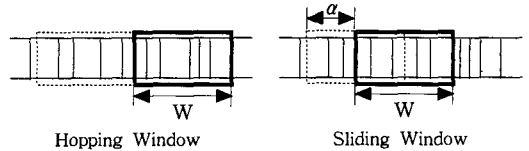


그림 3. 평활화 윈도우의 종류

경우 윈도우의 앞쪽에 I 프레임 같이 높은 전송률을 요구하는 프레임이 있으면 선적재가 불가능하여 평활화에 실패하게 된다. 또한, 윈도우를 이용하여 수립된 전송 계획에 따라 전송하더라도 네트워크의 상태의 변화나 지터의 발생으로 클라이언트의 버퍼에 언더플로우가 발생할 수 있다. 따라서 홉핑 윈도우 평활화에 비해 평균  $W/\alpha$ 배만큼의 계산 오버헤드가 있지만 네트워크의 변화에 더 민감하게 반응하고 윈도우의 경계를 넘어 데이터의 선적재가 가능한 슬라이딩 윈도우 평활화가 온라인 상에 적합하다. 따라서 본 기법에서는 [5]에서 실험 결과, 가장 효율이 좋은  $\alpha = W/2$  슬라이딩 윈도우 평활화 기법을 사용한다.

### 2.1.2. 윈도우와 버퍼의 크기

서버 내에서 대역폭 평활화를 위하여 사용하는 윈도우의 크기와 클라이언트 내 버퍼의 크기는 서로 밀접한 관련이 있으며 데이터의 평활화에 있어 큰 영향을 준다. 작은 크기의 윈도우의 경우, 평활화 효과가 떨어지지만 작은 크기의 버퍼와 빠른 응답 시간을 보장한다. 반면 큰 크기의 윈도우는 긴 시간 동안 좋은 평활화 성능을 보이지만 그에 대응하는 커다란 크기의 버퍼가 필요하다. 만일 큰 크기의 윈도우에 작은 크기의 버퍼를 이용하여 평활화를 시도한다면 버퍼로 필요한 만큼의 데이터의 선적재가 불가능하므로 요구 대역폭을 낮추기 위해 서비스 품질의 손실이 불가피하게 된다. 따라서 원하는 정도의 대역폭 평활화를 위해서는 이에 대응하는 버퍼의 크기가 필요하다. 제안된 동적 버퍼 관리 기법에서 윈도우의 크기와 버퍼의 크기는 서로 비례하여 늘어나거나 감소하도록 하였다. 초기 윈도우의 크기와 버퍼의 크기는 서비스를 요구하는 시스템의 환경이나 시스템의 목적, 사용자의 요구에 의해 달라질 수 있다.

## 2.2. 네트워크의 적응성

제안된 동적 버퍼 관리 기법에서는 크게 네트워크의 상태에 따라 초기 상태, Congestion 상태, Steady 상태, Slow Start 상태로 변하게 되고 가상으로 정해진 각 영역의 크기는 상태에 따라 변하게 된다.

### 2.2.1 초기상태

할당된 버퍼의 크기를  $b$ 라 할 때, 분할 버퍼의 초기 상태에는 네트워크 지터를 대비하기 위해  $b/2$  만큼의 선적재를 하고 데이터의 평활화를 위해  $b/2$ 의 동일한 크기를 이용한다.  $b/2$ 크기의 버퍼를 이용하여 결정된 전송 계획에 따라 버퍼에 적재된 데이터는 지터 대비 버퍼 영역에 데이터가 모두 적재된 후, 사용되기 시작한다.

### 2.2.2 Steady 상태

네트워크의 상태가 양호하여 지터가 발생하지 않거나 전송 대역폭이 충분하면 동적 버퍼는 서서히 더 넓은 영역의 평활화를 시도한다. 기본적으로 전송 대역폭의 평활화를 위하여 사용하는 버퍼 영역의 크기를 늘리고 이에 비례하여 슬라이딩 윈도우의 크기를 증가시킨다. 여기서 윈도우의 크기는 한 프레임씩 서서히 증가시키고 그에 따라 요구 대역폭의 평활화를 위해 사용하는 버퍼의 크기는, 윈도우의 크기를 W라 하고 평활화를 위해 사용하는 버퍼의 크기를  $b$ 라 할 때,  $b/W$ 만큼씩 증가시킨다. 너무 많은 크기의 증가로 인하여 데이터의 적재량이 줄었을 때 지터가 발생하면 버퍼 전체의 언더플로우를 초래할 수 있으므로 최대 증가 크기는 전체 버퍼 크기의 75%로 제한한다.

### 2.2.3 Congestion 상태

네트워크의 환경 변화나 지터로 인해 적재한 데이터의 소모가 미리 만들어진 전송 계획과 다르게 진행되면, Congestion 상태로 들어가고 서버로 이를 통보함으로써 전송하는 멀티미디어 데이터의 품질을 제어하여 대역폭의 요구 수준을 낮추도록 한다. 요구 대역폭의 평활화를

위해 사용하는 버퍼의 크기를 반으로 줄이고 그에 비례하여 슬라이딩 윈도우 크기를 W/2로 줄인다. 이에 따라 서버에서는 상대적으로 늘어난 지터 대비 버퍼 크기에 맞추어 재적재가 가능하도록 데이터의 품질을 제어하게 되고 윈도우의 크기가 작아짐에 따라 슬라이딩의 폭이 좁아져 더 빈번하게 전송계획을 수립하게 된다.

2.2.4 Slow Start 모드

지터 대비 버퍼영역의 적재를 마쳐 언더플로우의 상태에서 벗어나면 Slow Start 상태로 들어오게 된다. 이 상태에서는 버퍼가 초기 상태로 돌아가기 위한 준비를 하게 된다. 전송 가능한 데이터의 파악 없이 너무 급격하게 평활화 버퍼 크기의 증가를 시도하여 다시 Congestion 상태로 빠지는 것을 방지하기 위해서 서서히 서버의 슬라이딩 윈도우의 크기와 이에 비례하는 클라이언트 쪽의 요구 대역폭 평활화를 위한 버퍼의 크기를 늘려가기 시작하는데 기본적인 증가 방법은 Steady 상태에서의 방법과 동일하다.

3. 성능 평가

멀티미디어 시스템에서 동적 버퍼 관리 기법을 사용했을 때 얻어지는 효과를 모의 실험을 통하여 측정하였다. 실험은 지터가 빈번하게 발생하는 네트워크 상에서 동적 버퍼 관리 기법의 적용 여부와 요구 대역폭의 평활화 효과를 알아보는데 초점을 두었다. 실험을 위한 데이터를 위해 750프레임의 MPEG-1 비디오 파일을 사용하였으며 네트워크의 지터는 랜덤하게 발생하도록 하였고 지터가 발생하였을 때는 전송한 데이터가 모두 손실 되도록 하였다. 첫 번째 실험에서는 10kb의 버퍼 크기와 40프레임의 윈도우 크기를 가진 동적 버퍼 기법과 Feng 이 [4]에서 성능 분석을 위해 비교하였던 단순 버퍼 관리 기법이, 지터가 빈번하게 발생하는 네트워크에 환경에 잘 적응하는지 비교하여 보았다. 단순 버퍼 관리 기법은 전체 비디오 파일의 사전 정보를 얻어 요구 대역폭의 평균값으로 전송하고 주어진 버퍼에 적재를 한 후, 서비스를 시작하도록 하였다. 그림 4에서는 각 버퍼 관리 기법의 버퍼 적재량을 나타내었는데 단순 버퍼 관리 기법의 경우 초기에 많은 데이터를 적재하여 두지만 네트워크 지터가 발생했을 때 적재하여 둔 데이터를 소모하는 방법만을 사용하기 때문에 곧 버퍼의 언더플로우를 초래하였으나 동적 버퍼 관리 기법은 네트워크 상태에 따라 버퍼의 크기를 변화시키고 그 크기에 따라 데이터의 품질을 제어하여 요구 대역폭을 낮춤으로서, 재적재를 시도하기 때문에 단순 버퍼 관리 기법보다 더 좋은 버퍼의 활용도를 보였다. 두 번째에서는 같은 가상의 네트워크 환경에서 40프레임, 100프레임의 윈도우 크기와 10kb와 20kb의 버퍼 크기를 각각 조합하여 실험하였다. 그림 5에서 알 수 있듯이 네트워크의 지터 발생으로 버퍼 내 데이터의 적재량이 떨어지면 Congestion 상태에 들어가 윈도우 영역이 좁아지기 때문에 평활화 효과가 떨어지는 했지만 그 외의 부분에서는 좋은 요구 대역폭의 평활화 효과를 얻을 수 있었다. 40프레임의 윈도우보다 100 프레임의 윈도우에서, 10kb의 버퍼보다는 20kb의 버퍼에서 더 좋은 평활화 효과를 보였다. 결과적으로 동적 버퍼 관리 기법 역시 기본에 제안된 두 가지 버퍼의 특징을 모두 수용하고 있기 때문에 더 큰 버퍼와 윈도우 크기에 좋은 성능을 나타낼 수 있었다.

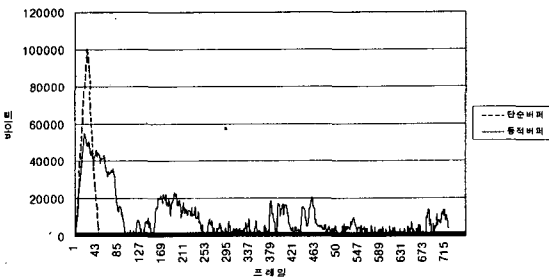


그림 4. 네트워크 변화 상태에서 버퍼의 활용도 비교

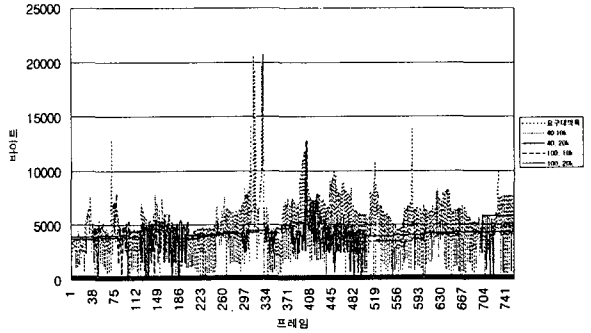


그림 5. 동적 버퍼 관리기법의 요구 대역폭 평활화

4. 결론

응용 프로그램 수준에서의 버퍼 관리 기법은 크게 동적인 네트워크에 대한 적응성을 높이기 위한 기법과 안정적인 서비스 품질의 보장에 초점을 둔 기법으로 나누어 볼 수 있다. 본 논문에서는 이 두 가지 기법을 절충한 동적 버퍼 관리 기법을 제안하였고, 제안된 버퍼 관리 기법은 기본적으로 서버 내의 슬라이딩 윈도우와 클라이언트 내의 버퍼로 구성하였고 이 버퍼를 다시 가상적인 두 부분으로 나누어, 네트워크의 지터 대비를 위해 데이터의 선적재를 위한 부분과 서버 내의 윈도우를 함께 데이터를 평활화 할 부분으로 구분하였다. 가상적으로 나눈 두 부분은 네트워크의 상태에 따라 상호 보완적으로 그 크기가 변하며 변화되는 버퍼의 크기에 비례하여 서버 내 윈도우 크기를 조정하도록 함으로써 네트워크의 상태에 민감하게 전송 데이터의 품질을 조정하도록 하였다. 또한 모의 실험을 통해 지터가 발생하는 네트워크 환경에서도 잘 적용하며 전체적으로 요구 대역폭의 평활화 효과도 보이고 있음을 알 수 있었다. 현재 제안된 동적 버퍼 관리 기법을 이용하여 실제 멀티미디어 시스템을 구현하고 있으며 향후 실제 구현된 시스템에서 기존에 제안되었던 버퍼 관리 기법과의 자세한 성능 비교를 계획하고 있다.

참고 문헌

- [1] Steve McCanne and V. Jacobson, "VIC: A Flexible Framework for Packet Video", In Proceedings of ACM Multimedia, November 1995.
- [2] L. A. Rowe, K. Patel, B. C. Smith, and K. Liu, "MPEG Video in Software Representation, Transmission and Playback," In Proceedings of IS&T/SPIE International Symposium. on Electronic Imaging: Science and Technology, Februaury 1994.
- [3] Jonathan Walpole, Rainer Koster, Shanwei Cen, and D. Towsley, "Optimal Buffering for the Delivery of Compressed Prerecorded Video", in Proceedings of ACM SIGMETRICS, pages 222-231, May 1996.
- [4] Wu-Chi Feng, Brijesh Krishnaswami, and Arvind Prabhudev, "Proactive Buffer Management for the Streamed Delivery of Stored Video", In Proceedings of ACM Multimedia, 1998.
- [5] J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online Smoothing of Live, Variable-Bit-Rate Video", University of Massachusetts computer science technical report 98-27, July 1998.
- [6] Wu-chi Feng and Stuart Sechrest, "Smoothing and buffering for delivery of prerecorded compressed video", In proceeding of IS&T/SPIE Multimedia Computing and Networking, Feb. 1995, San Jose, CA, pp. 234-242.
- [7] Wu-chi Feng and S. Sechrest, "Critical BAndwidth Allocation for the Delivery of Compressed Prerecorded Video", Computer Communications, Vol. 18, No. 10, Oct. 1995, pp. 709- 717