

효율적 멀티미디어 전송을 위한

다중 스트림 버퍼링 시스템

°김현태*, 김형진*, 이경근**, 나인호*

*군산대학교 정보통신공학과 멀티미디어 연구실

**호원대학교 전자계산학과

A Multi-Stream Buffering System for Efficient Transmission Transfer

°Hyoun-Tae Kim, Hyoung-Jin Kim, Kyoung-Kuen Lee, In-Ho Ra

Multimedia Lab., Dept. of Telecommunications Eng., Kunsan National Univ.

Dept. of Computer Science, Howeon Univ.

E-mail: ihra@ks.kunsan.ac.kr

요 약

다중의 연속 미디어 데이터를 출력 마감시간(playout deadline)전에 검색 및 전송하여 자연스럽게 처리하기 위한 가장 효율적인 처리방안은 메모리 버퍼를 이용하는 것이다. 본 논문에서는 다중 스트림의 데이터 전송 요구 및 해제의 버퍼 상태를 파악하여 버퍼를 동적으로 할당하고 물리적인 버퍼를 윈도우 크기의 논리버퍼로 분할하여 동적으로 관리하는 매핑관리자를 설계하여 버퍼이용률을 높일 수 있는 방안을 제시하였다. 또한, 서로 다른 소비율로 발생된 여분의 버퍼 공간을 활용할 수 있는 지능적인 선반입 기법과 마감시간을 고려한 전송 스케줄링 기법을 통해 실시간에 동기식으로 다중 미디어 스트림을 전송할 수 있는 동적 버퍼 관리 기법을 제안하였다. 마지막으로, 버퍼의 이용율을 높이기 위해 입출력 주기에 따라 버퍼 공간을 여러 스트림들이 파이프라이닝 방식으로 버퍼를 공유하는 기법과 잔여 버퍼공간의 할당 여부를 결정하기 위한 수락제어 기법을 적용하였다.

1. 서론

네트워크 성능의 향상과 더불어 저장 장치 및 입출력 인터페이스에 대한 기술 발전은 대용량의 멀티미디어 시스템을 보다 더 효율적으로 구축할 수 있는 발판을 제공하였다. 연속 미디어 즉, 오디오 및 비디오를 처리해야 하는 멀티미디어 시스템은 멀티미디어 데이터가 갖는 여러 가지 중요한 특징들을 효율적으로 다룰 수 있어야 한다. 첫째, 오디오나 비디오와 같이 지연에 민감한 미디어 데이터를 실시간에 처리할 수 있어야 한다. 둘째 멀티미디어 데이터가 요구하는 대량의 저장 공간과 입출력 대역폭을 제공할 수 있어야 한다. 셋째, 다양한 미디어가 서로 다른 스트림으로 전송될 경우에 다중 스트림 미디어 데이터 간의 출력을 동기화할 수 있어야 한다.

분산 환경에서 멀티미디어 데이터를 실시간에 연속적으로 사용자에게 보여주기 위해서는 멀티미디어 데이터를 검색 및 전송하여 연속적으로 출력하기 위해 커널 차원의 메모리 버퍼 관리 기능이 지원되어야 한다[1].

본 논문은 고정된 크기의 메모리 버퍼 공간과 충분하지 못한 디스크 대역폭만 제공되는 시스템 환경에서 멀티미디어 데이터의 처리율과 질의에 대한 응답시간을 최소화하고 각 미디어 스트림의 연속성 요구 조건을 충족시키면서 고정된 메모리 버퍼 공간을 최대한 효율적으로 이용할 수 있는 효율적인 버퍼링 모델과 동적 버퍼 관리 기법을 제안하는데 목적을 두고 있다[5].

최근 들어 Raymond[1] 등은 버퍼 공간과 디스크 전송율이 고정된 시스템 환경에서 멀티미디어 시스템의 처리율을 최대화할 수 있는 방법에 대해 연구하였으며, 이러한 연구를 통해 디스크의 대역폭과 제한된 버퍼 공간에 대해 공유 개념을 적용하면 전체 버퍼 요구량을 50%정도 절감할 수 있는 이론과 실험 결과를 발표한 바 있다. 그러나 지금까지 제안된 버퍼 관리 기법에는 다음과 같은 문제점들을 가지고 있다 [3][4][5][7].

- 커널 수준에서 동적으로 버퍼링 상태를 파악하고 버퍼링 시간과 조건을 동적으로 제어할 수 있는 방법을 제공하지 못했다.
- 물리적인 일차원 배열의 메모리 버퍼를 논리적으로는 다중 분할하고 버퍼의 생성 및 소비 상황을 파악하여 논리

본 논문은 '97 정보통신연구관리단의 대학기초 연구비 지원에 의해 연구되었음.

버퍼에 대한 저장 단위를 동적으로 변경하면서 미디어 데이터가 연속적으로 버퍼링될 수 있는 버퍼 관리 기법에 대해 고려하지 않았다.

- 파이프라이닝 기법을 이용하여 버퍼를 공유함으로써 버퍼의 이용률을 극대화하면서 버퍼 요구량을 최소화하는 방안에 대해 고려하지 않았다.
- 버퍼링 상태에 따라 응용 프로그램에서 멀티미디어의 서비스 품질을 조절할 수 있는 인터페이스를 제공하지 않았으며 버퍼링 속도를 동적으로 제어할 수 있는 관리 정보를 제공할 수 없었다.

이러한 문제점을 해결하기 위해 본 논문에서는 대용량의 다중 스트림 멀티미디어 데이터를 동기화하여 실시간에 처리할 수 있도록 제한된 버퍼 공간에 대한 버퍼 오버플로우나 언더플로우를 방지하면서 멀티미디어 데이터의 서비스 품질을 유지시킬 수 있는 방안에 연구하였다. 그 결과, 1차원 선형 버퍼를 논리적으로 다중 분할하여 여러 개의 스트림을 통해 서로 다른 속도로 전송되는 가변적인 다중 스트림 멀티미디어 데이터를 가변 크기의 논리 버퍼에 할당하여 이것들이 동시에 동기적으로 작동하도록 하는 다중 스트림 버퍼링 시스템을 제안하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 본 연구에서 제시한 새로운 버퍼링 모델의 구조와 특징에 대해 설명하고 각 모듈간의 인터페이스 구조와 동적 버퍼 관리 기법의 작동 방식에 대해 설명한다. 3장에서는 향후 연구 계획과 결론에 대해 기술한다.

2. 제안된 버퍼링 시스템 모델의 특징 및 구조

본 연구에서 제안한 버퍼링 시스템 모델은 VOD, 원격교육, 원격 의료와 같이 분산 환경에서 멀티미디어 데이터가 여러 개의 채널을 통해 다중 스트림 방식으로 전송 및 처리되어야 하는 경우 이것들을 효율적으로 버퍼링하여 동기식으로 출력하기 위해 필요한 것으로서 그림 1과 같은 시스템 환경에서 이용된다.

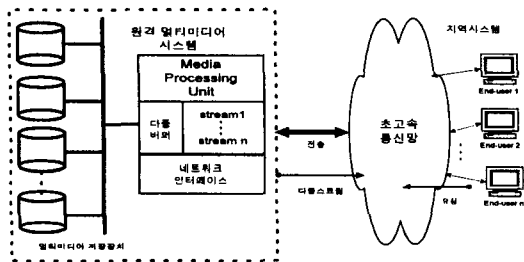


그림 1 초고속 통신망을 이용한 멀티미디어 전송 시스템 구조

2.1. 버퍼 관리 기법 설계시 고려사항

앞에서 제시한 기존 버퍼 기법들의 문제점들을 해결하기 위해서 본 논문에서는 다음과 같은 특징을 제공하는 버퍼 구조와 동적 버퍼 관리를 위한 추진 전략을 적용하였다.

1. 물리 버퍼를 윈도우 단위의 가변 크기로 분할한 논리 버퍼에 할당하고 주기 단위에 따라 파이프라인 방식으로 버퍼를 공유하여 버퍼 사용 공간을 최소화하고 이용률을 극대화한다.
2. 버퍼 할당과 회수를 가변적인 윈도우 크기로 수행한다. 이로 느린 속도의 대용량 미디어 스트림과 빠른 속도의 소용량 미디어 스트림 간의 동기화와 버퍼 공유를 효율적으로 처리할 수 있도록 하고 동적 버퍼 할당 및 회수를 가능하도록 한다.
3. 버퍼에서 최소 잔류시간을 이용한 선반입 버퍼링 기법을 적용하여 입력력 장치의 대역폭을 효율적으로 사용하고 입력력 응답 및 서비스 시간을 최소화하도록 한다.
4. 다중 스트림 데이터의 프리젠테이션 시간 정보와 선반입하기 위한 시간 정보를 구축하여 출력 동기화가 보장되도록 한다.
5. 멀티미디어 데이터의 서비스 품질과 네트워크 대역폭의 효율적으로 사용하기 위하여 새로운 스트림에 대한 서비스 수용 여부를 결정하기 위해 수락제어 기법을 사용한다.

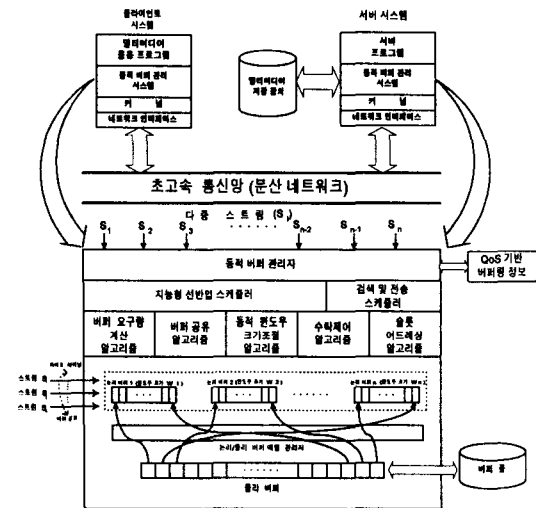


그림 2 윈도우 기반 동적 버퍼링 시스템 구조

2.2. 동적 버퍼 관리 시스템

동적 버퍼 관리 기법이란 미디어 스트림들의 버퍼 요구량에 따른 버퍼 스케줄링 정보와 네트워크 대역폭과 지연 변동에 대한 정보를 통해 각 스트림에 최소한의 필요 버퍼 공간을 할당하고 회수하는 기법으로서 실시간 동기식 출력을 지원하면서 서비스 품질을 유지하기 위한 기법이다.

이러한 목표를 달성하기 위해 제안된 버퍼 관리 기법에서는 지능형 선반입 스케줄러를 사용하여 활성화된 스트림에게 남아있는 잔류자원(remaining resource: 디스크 대역폭, 버퍼 공간 등)을 할당하고 여분을 대기중인 스트림에게 할당하여 다중 스트림의 개수를 최대화할 수 있도록 선반입 시기 및 방법을 결정하고 선반입된 버퍼 공간을 오랫동안 점유하고 있지 않도록 하는 버퍼 교체 알고리즘을 사용하였다.

2.2.1 동적 버퍼 관리자

동적 버퍼 관리자는 선반입 기법을 이용하여 디스크 대역폭, 버퍼 공간, 네트워크 대역폭의 이용율을 극대화하고 버퍼링 조건, 버퍼 요구량, 미디어 스트림의 개수, 멀티미디어 프리젠테이션 서비스 품질, 슬롯 어드레싱 정보, 물리 버퍼와 논리 버퍼 간의 매핑 정보 등을 이용하여 네트워크의 트래픽 부하 변동에 따른 네트워크 지연 변동을 흡수하여 미디어 데이터의 연속적인 출력이 가능하도록 미디어 스트림 간의 동기화를 수행하기 위한 버퍼 관리 기능을 총괄적으로 수행하기 위해 필요하며, 그림 2와 같은 구조로 구성된다.

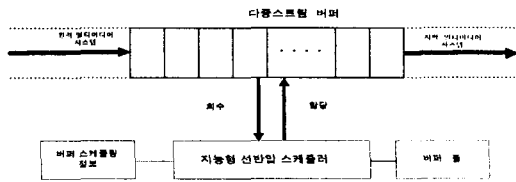


그림 3 선반입 버퍼 스케줄러 구조

2.2.2 버퍼 요구량 계산

서로 다른 소비율을 갖는 스트림의 최대 버퍼 요구량을 윈도우 크기 별로 계산하여 버퍼 요구량을 최소화 할 수 있도록 하고 스트림의 검색 요구된 데이터를 시간 슬롯으로 나누어 버퍼에 할당하여 버퍼 공유의 효율을 최대한 이용하도록 하는 버퍼 요구량 계산식을 사용한다.

$$B = \sum_{i=1}^n B_i = \frac{T * \rho - s}{P} \sum_{i=1}^n p_i * (R - p_i) \quad \text{식(1)}$$

(B: 전체 버퍼 요구량 n: 요구 스트림의 수 B_i: 각 스트림의 버퍼 요구량 T: 사이클 주기 ρ: 디스크 이용률 P: 전체 소비율 p_i: 각 스트림의 소비율)

2.2.3 수락 제어(admission control)

멀티미디어 서비스를 제공하기 위해서는 다중의 클라이언트 요구를 동시에 서비스하면서 요구된 QoS를 보장할 수 있도록 해야 한다. 새로운 사용자에게 서비스를 제공할 것인지 아닌지를 결정하기 위해서는 새로운 사용자가 요구한 서비스를 추가로 수행할 경우 기존의 사용자에게 영향을 미치지 않도록 해야 한다. 멀티미디어 서버는 제한된 자원(디스크 전송률, 버퍼크기)을 가지고 있기 때문에 당연히 일정한 사용자에게만 서비스를 제공할 수 있으며 이러한 자원제약조건을 기초로 하여 새로운 사용자에게 서비스 제공할 것인지를 판단해야 하며, 이러한 판단을 결정하기 위해서는 수락제어가 필요하다.

이를 위해서는 버퍼 오버플로우나 언더플로우가 발생하지 않도록 하면서 QoS를 유지할 수 있도록 데이터 검색 및 전송량을 제어하며 자원 요구량에 평균치를 기반으로 하여 추가적인 전송 서비스가 가능한데도 수행되지 않는 경우를 최대한 줄여야 한다. 본 논문에서는 가능한 한 최대의 연속 미

디어 스트림에 대한 디스크 검색 요구를 처리하기 위해 주기적 서비스 시간을 기반으로 각각의 미디어 스트림을 인터리빙 방식으로 처리하는 수락 제어 기법을 사용하였다.

하나의 주기내에서 인터리빙 방식으로 n개의 각 스트림을 디스크에서 검색할 때 소요되는 하나의 서비스 시간이 T라면 디스크 이용률은 $\rho = (\sum_{i=1}^n t_i + s) / T$ 과 같다(여기서 s는 전체 탐색시간이다). 따라서 각 스트림 S_i에 대해 버퍼 언더플로우 및 오버플로우를 일으키지 않을 조건은 다음과 같다.

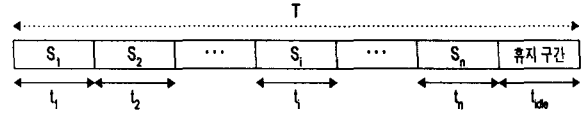


그림 4 한 서비스 시간의 동작중인 스트림들

- 조건 1: 각 스트림의 읽기량과 소비량이 같아야 한다.
- 조건 2: 각 스트림의 데이터가 해당 시간 슬롯내에 각 S_i의 데이터를 소비하기 위해서는 반드시 t_i내에 모두 읽기를 마치는 시점에서 최대의 버퍼를 요구해야 한다.

조건 1을 만족한다는 것은 최소의 버퍼공간을 요구하는 것을 의미하며, 이것에 의해 디스크 입출력 및 버퍼링에 필요한 서비스 시간(주기)의 하한을 결정할 수 있다. 즉,

$T \geq \frac{s * R}{R * \rho - P}$ 이 된다. 또한, 조건 2로부터 가능한 가장 버퍼를 이용한다면 서비스 시간의 상한을 결정할 수 있다. 즉, 읽기를 마치는 시점에서 버퍼 요구량은 식 (2)와 같다.

$$B_i = P_i * (R - P_i) * \frac{T * \rho - s}{P} \quad \text{식(2)}$$

그러므로 n개의 스트림에 대한 총 버퍼 공간 요구량은 식 (3)과 같고, 시스템의 자원제한성으로 인한 가장 버퍼크기가 B_{max}라면 반드시 $B \leq B_{max}$ 조건을 만족해야 하므로 식 (4)와 같이 최대의 서비스 시간(upper bound)을 정할 수 있다.

$$B = \sum_{i=1}^n B_i \quad \text{식(3)}$$

$$T \leq \frac{B_{max} * P}{\rho * \sum_{i=1}^n P_i * (R - P_i)} + \frac{s}{\rho} \quad \text{식(4)}$$

따라서 위의 식에 나타난 조건을 만족하도록 하면 각 스트림들의 연속성을 유지하면서 데이터가 손실되지 않도록 서비스 시간의 범위를 정확히 결정하여 여러 개의 미디어 스트림을 제시간에 서비스할 수 있다. 또한, T를 기반으로 새로운 사용자의 서비스 요구에 대한 수락제어(admission control)를 수행하면 제한된 자원의 이용율을 극대화하면서 가능한 한 많은 수의 미디어 스트림을 서비스할 수 있다.

2.2.3. 동적 윈도우 크기 조절 및 파이프라이닝 기법

제한된 버퍼 공간에 여러 개의 병행 스트림 요구가 발생했

을 때 각각의 활성 스트림 데이터가 요구하는 전체 버퍼 공간을 시간 단위로 세그먼트화하고, 각 세그먼트에 해당하는 크기 만큼의 버퍼 공간을 윈도우 단위로 구성된 시간 슬롯에 할당하는 버퍼 어드레싱 기법(buffer addressing scheme)을 사용하면 버퍼 소비량을 줄일 수 있다.

본 논문에서는 각 미디어의 데이터 세그먼트를 주기적으로 시간 슬롯(윈도우)에 할당되 데이터 세그먼트들이 분산적으로 저장되지 않도록 소비율에 따라 지정된 범위 내에서 다음 시간 슬롯에 라운드 로빈 방식으로 할당하는 방법을 적용하도록 하였다. 또한, 미디어 스트림의 단위 시간 별로 동일한 버퍼 공간을 요구하는 미디어들을 하나의 소비 그룹으로 분류하여 이것들이 버퍼 공간을 논리적 윈도우 단위로 공유할 수 있는 수 있는 파이프라이닝 방식을 적용하였다. 파이프라이닝 방식을 이용하면 동일한 주기 뿐만 아니라 반주기(half period) 또는 배주기(double period)를 가진 미디어 스트림 간의 버퍼 공간 공유가 가능해진다. 그러나 파이프라이닝 방식을 이용한 버퍼 공유에서는 버퍼공간에 대한 저장율과 소비율이 일정시간 동안에 동일한 속도를 유지되어야 하는 문제가 발생한다. 그렇지 않으면 파이프라이닝 효과가 발생하지 않으며 오히려 버퍼를 공유하기 위해 기다려야 하는 문제가 발생하여 연속적으로 미디어 스트림을 버퍼에 저장할 수 없다. 따라서 버퍼에 할당되는 미디어 스트림의 데이터 세그먼트 크기를 동적으로 변경시키고 각 시간 슬롯의 크기(윈도우)를 세그먼트 크기의 변화에 따라 동적으로 변경시킬 수 있는 기법이 필요하다. 이를 위해 본 논문에서는 디스크 이용률과 미디어 스트림 전송 속도에 따라 수신측의 버퍼 할당 단위를 동적으로 변경하여 버퍼 공간에 대한 저장율과 소비율을 다 단위 시간내에서 동기화 될 수 있도록 하는 동적 윈도우 크기 조절 기법을 제안하였다.

2.2.5. 선반입 스케줄링

어떤 하나의 스트림에 대한 새로운 요구가 발생하면 수락 제어기(admission controller)는 이것을 허용할 수 있는지를 결정하기 위해 디스크 전송률과 버퍼 공간에 여유가 있는지를 검사하여 만약에 충분한 자원이 있다면 그 질의를 활성화하고 그렇지 않으면 해당 요구는 대기큐에서 기다려야 한다.

일반적으로 시스템의 성능은 시스템의 처리력과 질의에 대한 응답시간으로 측정할 수 있다. 그러나 시스템 설계시에 버퍼 공간과 디스크 전송률이 미리 결정된다면 응답시간은 일차적으로 디스크 이용률과 버퍼 이용률에 의해서 결정된다.

본 논문에서는 선반입 기법을 이용하여 최대의 버퍼 이용률과 디스크 이용률을 제공하기 위한 방안에 대해 연구하였다. 선반입을 기법을 이용하면 큐에 대기중인 어떤 미디어 스트림이 활성 상태로 진입하기 전에 약간의 데이터를 미리 읽어들이므로써 여분의 디스크 이용률과 버퍼 소비율을 최대한 활용할 수 있다. 그러나 선반입으로 인하여 가용 자원의 고갈을 초래할 수 있고 이로 인하여 버퍼 공간에 대한 버퍼오버플로우가 발생할 수 있다. 따라서 선반입시에는 대기중인 미디어 스트림 중에서 최소의 버퍼 공간을 요구하는 것을 선택하여야 버퍼 오버플로우를 방지할 수 있다. 만약에 대기큐에 미디어 스트림이 우선 순위를 가지고 대기하는 경우에는 가장 먼저 서비스해야 할 미디어 스트림을 서비스할 수 있는

버퍼 공간을 예약하는 스케줄링 기법을 적용하여야 한다. 그림 5는 본 논문에서 제안한 지능형 선반입 스케줄링 알고리즘을 간략히 나타낸 것이다.

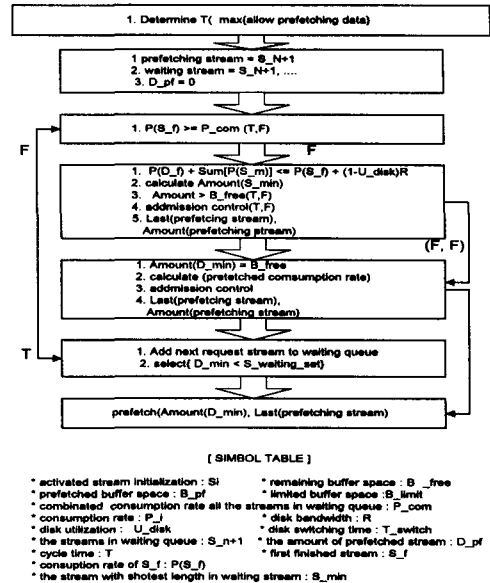


그림 5 선반입 스케줄링

2.2.6 모듈 인터페이스

제한된 디스크 입출력 속도와 버퍼 공간만으로 여러 개의 다중 미디어 스트림을 서비스하기 위해 본 논문에서 제안한 동적 버퍼링 시스템의 모듈 간 인터페이스 구조는 그림 6과 같다.

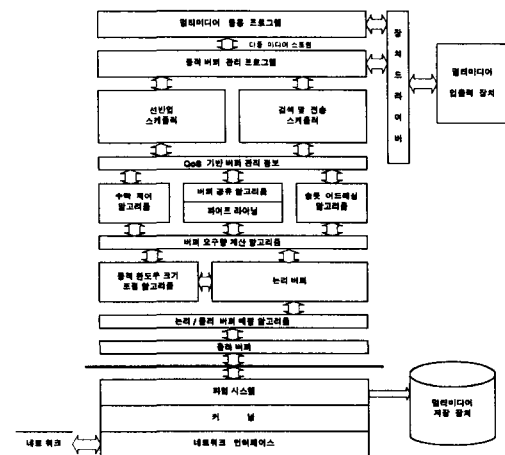


그림 6 버퍼링 시스템의 모듈 간 인터페이스 구조

3. 결론 및 향후 연구계획

본 논문에서는 분산 멀티미디어 환경에서 연속 미디어를 검색 및 처리하기 위해서 커널 수준에서 제공할 수 있는 버퍼링 시스템 모델 및 동적 버퍼 관리 기법을 제안하였다.

향후에는 제안된 기법의 성능을 평가하기 위해 버퍼링 시스템 모델과 여러 가지 관리 기법을 유닉스 시스템에 구현하고 다양한 멀티미디어 응용 프로그램을 통해 서비스품질, 프리젠테이션 동기성, 버퍼이용율 등에 대한 성능을 평가 및 분석할 예정이다.

[참고문헌]

- [1] J. Gemmel and S. Christodoulakis, "Principles of Delay Sensitive Multimedia Storage and Retrieval", ACM Trans. on Information System. Vol. 10. No. 1, pp. 51-90, Jan., 1992.
- [2] Kun-Lung Wu, Philip S. Yu, "Consumption-Based Buffer Management for Maximizing System Throughputs of News-on-Demand Multimedia Systems", IBM Research Division RC 20143, 1995.
- [3] Raymond T. Ng, Jinhai Yang, "An Analysis of Buffer Sharing and Prefetching Techniques for Multimedia Systems", Multimedia Systems 4(2): 55-69. 1996.
- [4] Yeon Seung Ryu and Kern Koh, "A Dynamic Buffer Management Technique for a Video-on-Demand Server", in Proc. of IPSJ International Symposium on Multimedia Systems, Yokohama, Japan, Mar. 1996
- [5] Dan Revel, Crispin Cowan, Dylar Mc Nameee, Calton Pu and Jonathan Walole, "An Architecture for Flexible Multimedia Prefetching", Workshop on Resource Allocation Problems in Multimedia Systems, Washington DC, December 1996.
- [6] Dwigh J. Makaroff and Raymond T. Ng, "Buffer Sharing Schemes for Continuous-Media Systems", Information systems, 20(6). pp 446-465. 1994.
- [7] S. Gollapudi and A. Zhang, "Buffer Management in Multimedia Database Systems", IEEE Int'l Conference (ICMCS'96), Hiroshima, Japan, June, 1996.