

# 큐잉 모델을 이용한 대규모 주문형 비디오 서버의 성능분석

正會員 조진성\*, 신현식\*

## Performance Analysis of a Large-scale Video-On-Demand Server using Queueing Model

Jinsung Cho\*, Heonshik Shin\* *Regular Members*

### 요약

본 논문에서는 대규모 주문형 비디오 서비스를 위한 서버의 큐잉 모델을 설정하고 이를 통하여 성능을 분석한다. 주문형 비디오 서비스는 향후 초고속정보통신망에서 가장 큰 시장을 형성할 것으로 예측되고 있으며 이는 수천편의 영화를 저장하고 수만명의 사용자를 동시에 서비스할 수 있는 대규모 서버를 필요로 한다. 본 논문에서는 중소형 컴퓨터로 구성된 분산 시스템을 대상으로 하며 분산 시스템 내의 각 서버 노드들이 독립적으로 서비스를 수행한다. 본 논문에서 제시하는 큐잉 모델은 M/M/1 큐로 구성된 열린 큐잉 네트워크 모델로 패킷 손실 확률을 구하여 서버의 성능을 분석한다. 패킷 손실 확률은 사용자의 패킷 요청이 패킷의 종료시한내에 서비스 되지 못할 확률로 정의된다. 또한, 제시된 큐잉 모델은 시뮬레이션에 의해 그 정확성이 검증되므로 대규모 주문형 비디오 서버의 설계시 도구로 유용하게 사용될 수 있다.

### ABSTRACT

In this paper, we propose a queueing model of a server for large-scale video-on-demand services and analyze its performance. VOD services are expected to be spread over high-speed network. This service requires a large-scale server which stores thousands of movies and services tens of thousands of clients concurrently. This paper adopts a distributed architecture for the server which consists of PC or workstation. The open queueing network model proposed in this paper consists of several M/M/1 queues. The performance of the server derives from the packet loss probability that the client request for a packet is not serviced within its deadline. The queueing model have been validated through simulation, and thus, can be used in designing a large-scale VOD server.

\*서울대학교 컴퓨터공학과  
 論文番號:97236-0712  
 接受日字:1997年 7月 12日

## 1. 서 론

최근들어 멀티미디어 기술은 컴퓨터와 통신 시스템의 발달과 함께 급속한 발전을 이루고 있다. 특히, 정부에서 진행하고 있는 초고속 통신망 사업과 현재 서울을 중심으로 상용화되고 있는 Cable TV 망을 토대로한 멀티미디어 응용 서비스가 매우 활발히 개발되고 있다. 그중 사용자와 대화식으로 동작하여 동영상 데이터를 실시간으로 전송하는 주문형 비디오 서비스(Vidco-On-Demand service)는 향후 초고속정보통신망 상에서 가장 커다란 시장을 형성할 것으로 예측되고 있다[1].

이러한 대규모 주문형 비디오 서버를 운영하기 위해서는 수백내지 수천편의 영화를 저장하고 수천내지 수만명의 사용자를 동시에 서비스 할 수 있는 대규모 주문형 비디오 서버가 필수적이다. 0.5 Bps의 비트율을 가진 MPEG-1 스트림을 저장하더라도 100분 길이의 영화는 3 GB의 저장공간이 요구되며 2천개의 영화를 위해서는 6 TB가 필요하다. 이것은 5 GB 디스크 1,200개를 필요로하며 이때 디스크들의 병렬성을 최대한으로 활용하고 디스크의 전송률이 10MB/s 이라면 24,000명이 동시에 서비스를 받을 수 있다.

최근 몇년동안 여러 대학과 회사에 의해 프로토타입 주문형 비디오 서버가 개발되고 있다. 이러한 연구에서는 사용자에 동영상 데이터를 실시간에 전송해야 하는 대전제하에 서버의 자원을 최대한으로 이용하여 가능한 많은 사용자에 서비스를 수행하기 위한 서버의 구조 및 효율적인 알고리즘 등을 제시하고 있다[2, 3, 4, 5]. 그러나 많은 실험결과로부터 하나의 PC급 서버로는 수~수십개만의 스트림을 서비스할 수 있음이 명확해졌고, 따라서 위에서 언급한 규모의 주문형 비디오 서비스 시스템은 초병렬 대용량 컴퓨터에 의한 병렬 시스템 또는 중소형 컴퓨터에 의한 분산 시스템으로 구성되어야 한다. 이러한 대규모 주문형 비디오 서버에서는 입출력 대역폭(I/O bandwidth)으로 대표되는 자원의 관리가 매우 중요하며, 구축할 서비스 시스템의 규모에 따라 적절한 자원이 할당되어야 한다.

본 논문에서는 대규모 주문형 비디오 서비스를 위한 서버의 큐잉 모델을 설정하고 이를 통하여 성능을 분석한다. 제시된 큐잉 모델은 M/M/1 큐로 구성된

열린 큐잉 네트워크 모델로 패킷 손실 확률을 구하여 서버의 성능을 평가한다. 여기에서 패킷 손실 확률이란 사용자의 패킷 요청이 패킷의 종료시한(deadline) 내에 서비스 되지 못할 확률을 말한다. 또한, 본 논문에서 제시한 큐잉 모델은 시뮬레이션에 의해 그 정확성이 검증된다. 따라서 대규모 주문형 비디오 서버의 설계시 도구로 유용하게 사용될 수 있다.

본 논문의 구조는 다음과 같다. 2절에서는 본 논문이 대상으로 하고 있는 대규모 주문형 비디오 서비스 시스템의 모델을 설명하고, 3절에서는 서버의 큐잉 모델을 설정한다. 큐잉 모델을 기반으로 다양한 매개변수에 의한 성능 분석 결과를 4절에 제시하고 5절에서 결론을 맺는다.

## II. 시스템 모델

대규모 주문형 비디오 서비스 시스템의 구성은 초병렬 대용량 컴퓨터에 의한 병렬 시스템(예: Oracle [3]) 또는 중소형 컴퓨터에 의한 분산 시스템(예: Microsoft[4], Bellcore[5])으로 양분하여 고려할 수 있다. 차세대 주문형 비디오 서버의 구조로 위의 두가지가 크게 대두되고 있으며 각각의 장단점이 논의되고 있다. 그런데, 분산 시스템으로 구성된 위의 두 예의 접근 방법이 다르다. [4]에서는 시스템의 모든 노드에 하나의 영화를 스트라이핑 기법[6]에 의해 저장하므로 보다 높은 디스크 대역폭과 노드의 부하균배를 얻는다. 따라서, 각 노드가 분산 스케줄링에 의해 협동하여 동작함으로써 서비스가 이루어진다. 이와 달리, [5]에서는 각 노드가 독립적으로 서비스를 수행하며 노드들을 관리하는 조정자(coordinator)가 존재

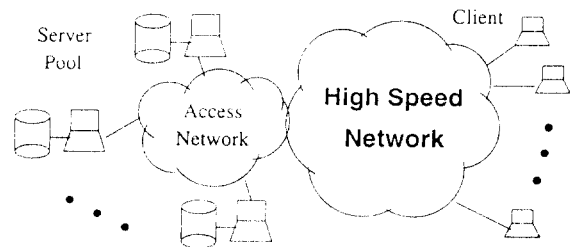


그림 1. 시스템 모델

한다. 이 방법은 노드들의 부하가 균일하지 않을 가능성이 존재하나 사용자의 요청이 특정 영화에 집중된다는 점을 고려하여[6] 노드에 인기있는 영화를 중복배치함으로써 부하균배를 얻을 수 있다.

본 논문에서는 [5]와 유사한 시스템 구성을 대상으로 한다(그림 1 참조). 사용자는 초고속정보통신망을 통하여 연결된 서버 풀(pool)에 접속하여 원하는 목록을 저장하고 있는 서버노드를 선택받아 서비스를 받는다. 이때 조정자가 서버노드를 선택하여 주며 서버 풀내의 서버노드의 부하균배를 고려하여 선택한다. 조정자와 사용자간의 연결 설정 기간은 그리 크지 않으므로 시스템의 성능에는 영향을 미치지 않는다고 가정한다. 서버노드들은 영화의 인기도를 고려하여 영화를 중복 저장하여 특정 서버노드의 용량초과(overflow)를 초래하지 않도록 한다. 각 서버노드는 높은 디스크 대역폭을 위한 디스크 배열등의 기법과 효율적인 자원 이용 알고리즘을 이용하여 성능을 최적화시킨다.

그림 1의 시스템 모델을 선정한 이유는 다음과 같다: 첫째, 초병렬 대용량 컴퓨터에 의한 접근 방법은 시스템 설계시 규모의 융통성에 제약을 가할뿐 아니라 시스템의 확장이 곤란하다. 분산 시스템의 접근 방법은 성능 대 비용면에서 유리한 중소형 컴퓨터를 이용하므로 그에 비해 유리하다. 둘째, [4]와 같이 분산 시스템의 모든 노드가 동기화되어 동작하여야 하는 경우는 시계 동기화의 오버헤드, 복잡한 스케줄링 등의 문제점이 존재한다. 또한 영화의 추가/삭제/변경 시 모든 노드에서 작업이 수행되어야 한다. 셋째, 구현 및 관리가 용이하다. 그러나 이러한 시스템 모델의 가장 큰 문제는 특정 서버노드로의 사용자 요구가 집중되는 것이다. 위에서 언급한대로 영화의 인기도를 고려하여 영화를 중복 저장하여 서버노드의 부하균배를 이루게 하는 것이 중요하며 이 작업은 주기적으로 수행되어야 한다.

그림 1에서 접근망(access network)은 서버 풀내의 서버노드간의 연결 및 초고속정보통신망에의 접근을 담당하는 장치 또는 망을 말한다. 본 논문에서는 서버 풀과 접근망까지를 모델화 대상으로 하며 초고속정보통신망의 대역폭은 접근망의 대역폭보다 크다고 가정한다.

### III. 큐잉 모델

2절에서 언급한 시스템은 그림 2와 같이 열린 큐잉 네트워크 모델(open queueing network model)로 나타내어질 수 있다. 우선 우리는 영화를 위한 비디오 스트림을 포아송(Poisson) 분포를 갖는 확률변수로 모델화한다. 이는 MPEG등의 압축에 의한 가변 비트율 스트림(variable bit rate stream)을 잘 모델화 할 수 있다[7]. 따라서 각 사용자,  $C_1, \dots, C_m$ 의 데이터 패킷 요청은 각각  $\lambda_1, \dots, \lambda_m$ 을 매개변수로 하는 포아송 프로세스로 모델화된다. 사용자의  $m$ 개의 포아송 프로세스는 네트워크 입력 큐에서 모아져(superposition) 매개변수  $\lambda$ 의 새로운 포아송 프로세스를 생성하고, Burke의 정리[8]에 의해 포아송 프로세스로 네트워크 입력 큐를 떠난다. 그리고, 네트워크 입력 큐를 떠난 포아송 프로세스는 서버노드  $i$ 에 확률  $p_i$ 로 분할(decomposition)되어 서비스 받은 후 네트워크 출력 큐에서 다시 모아진다.

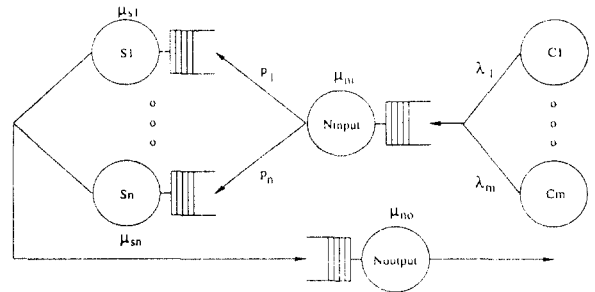


그림 2. 큐잉 네트워크 모델

따라서 그림 2의 큐들은 다음과 같이 모델화 된다.

- 네트워크 입력:  $M/M/1, \lambda = \sum_{i=1}^m \lambda_i, \mu_{ni}, \rho_{ni} = \lambda / \mu_{ni}$
- 서버노드  $i$ :  $M/M/1, p_i \lambda, \mu_{si}, \rho_{si} = p_i \lambda / \mu_{si}$
- 네트워크 출력:  $M/M/1, \lambda, \mu_{no}, \rho_{no} = \lambda / \mu_{no}$

여기에서, 우리는 요청된 데이터 패킷의 서버내에서의 지연시간을 구하여 비디오 스트림의 연속성을 만족하는지 여부를 살펴보고자 한다. 각 큐에서의 지연 시간  $W_{ni}, W_{si}, W_{no}$ 에 대한 확률밀도함수(p.d.f.)[10]는

M/M/1 큐의 분석으로부터 다음과 같이 구할 수 있다[9].

$$\begin{aligned} f_{W_{ni}}(t) &= \mu_{ni}(1-\rho_{ni})e^{-\mu_{ni}(1-\rho_{ni})t} \\ f_{W_{si}}(t) &= \mu_{si}(1-\rho_{si})e^{-\mu_{si}(1-\rho_{si})t} \\ f_{W_{no}}(t) &= \mu_{no}(1-\rho_{no})e^{-\mu_{no}(1-\rho_{no})t} \quad (t \geq 0) \end{aligned} \quad (1)$$

이때 서버에서의 전체 지연시간  $W = W_{ni} + W_{si} + W_{no}$ 에 대한 확률밀도함수  $f_W(t)$ 는  $f_{W_{ni}}(t)$ ,  $f_{W_{si}}(t)$ ,  $f_{W_{no}}(t)$ 의 콘볼루션(convolution) 연산에 의해 구할 수 있다.

$$f_W(t) = f_{W_{ni}}(t) \otimes f_{W_{si}}(t) \otimes f_{W_{no}}(t) \quad (2)$$

그런데 콘볼루션 연산은 그 복잡도가 매우 크므로 라플라스 변환(Laplace transform) 기법에 의해  $f_W(t)$ 를 구하고자 한다.  $f_W(t)$ 의 라플라스 변환인  $F_W^*(s)$ 는 다음에 의해 구해진다.

$$F_W^*(s) = F_{W_{ni}}^*(s) \cdot F_{W_{si}}^*(s) \cdot F_{W_{no}}^*(s) \quad (3)$$

여기에서  $F_{W_k}^*(s)$  ( $k = ni, si, no$ )는

$$F_{W_k}^*(s) = \int_0^{\infty} f_k(t)e^{-st} dt = \mu_k(1-\rho_k)/(\mu_k(1-\rho_k) + s) \quad (4)$$

와 같이 구해진다. 여기에서  $F_W^*(s)$ 로부터 역 라플라스 변환을 취하면 다음과 같이  $f_W(t)$ 를 구할 수 있다<sup>1)</sup>.

$$\begin{aligned} f_W(t) &= B_1 e^{-a_1 t} + B_2 e^{-a_2 t} + B_3 e^{-a_3 t} \\ a_1 &= \mu_{ni}(1-\rho_{ni}) \quad a_2 = \mu_{si}(1-\rho_{si}) \quad a_3 = \mu_{no}(1-\rho_{no}) \\ B_1 &= \frac{a_1 a_2 a_3}{(a_2 - a_1)(a_3 - a_1)} \quad B_2 = \frac{a_1 a_2 a_3}{(a_1 - a_2)(a_3 - a_2)} \quad (5) \end{aligned}$$

$$B_3 = \frac{a_1 a_2 a_3}{(a_1 - a_3)(a_3 - a_3)}$$

식 (5)의 정확성은 식 (6)에 의해 검증될 수 있다.

$$E(W) = E(W_{ni}) + E(W_{si}) + E(W_{no}) \quad (6)$$

식 (5)로부터 사용자의 패킷 요청이 종료시한내에 서비스 되지 못할 확률은 다음과 같다.

$$\begin{aligned} P[\text{패킷손실}] &= \int_D^{\infty} f_W(t) dt \\ &= \frac{B_1}{a_1} e^{-a_1 D} + \frac{B_2}{a_2} e^{-a_2 D} + \frac{B_3}{a_3} e^{-a_3 D} \end{aligned} \quad (7)$$

여기에서  $D$ 는 패킷의 종료시한으로 데이터의 재생시간을 의미한다. 이 값은 서버내에서의 미리읽기(read-ahead) 기법에 의해 증가할 수 있다.

#### IV. 성능 분석

본 절에서는 3절에서 제시한 큐잉 모델에 근거하여 대규모 주문형 비디오 서버의 성능을 분석한다. 표 1은 성능에 영향을 미치는 매개변수들을 나타낸다.

표 1. 본 논문에서 사용된 매개변수

기호	설명	사용된 값
$m$	사용자의 수	
$n$	서버노드의 수	10
$R_s$	스트림의 비트율	1.5 Mbps
$R_n$	접근망의 대역폭	1 Gbps
$R_d$	디스크의 대역폭	10 MBps
$B_{req}$	사용자 요청 패킷의 크기	1 KB
$B_{data}$	사용자 데이터 패킷의 크기	256 KB
$p_i$	서버노드 $i$ 에서 서비스 받을 확률	
$k$	미리읽기 상수	2

표 1에 나타난 매개변수들을 이용하여 3절에 사용된 변수를 나타내면 다음과 같다:

$$\mu_{ni} = \frac{B_n}{B_{req}}, \mu_{no} = \frac{B_n}{B_{data}}, \mu_{si} = \frac{R_d}{B_{data}}$$

$$\lambda = \sum_{i=1}^m \lambda_i = m\lambda_s = m \cdot \frac{R_s}{B_{data}}$$

$$D = k \cdot \frac{1}{\lambda_s} = k \cdot \frac{B_{data}}{R_s}$$

여기에서 모든 사용자가 같은 비트율의 스트림(예: MPEG-1)을 요청하고 모든 서버노드는 동일하다고

1) 본 논문에서는 inspection 기법[9]에 의해 역 라플라스 변환을 구하였다.

가정한다. 또한, 서버노드들간의 부하가 균일하게 분배되어  $p_i = 1/n$ 로 가정한다. 이것은 서버의 설계시 가장 중요하게 고려해야할 사항이다.

그림 3은 큐잉 모델과 시뮬레이션 결과의 비교를 나타낸다. 시뮬레이션은 MPEG 트레이스(trace) 데이터를 사용하였으며 그외의 매개변수는 동일하게 하였다. 시뮬레이션 프로그램은 SMPL[11]을 이용하여 작성되었다. 그림 3에서 나타난 약간의 차이는 시뮬레이션에서 MPEG 트레이스 데이터를 사용한 때문인 것으로 사료된다. 그러나 패킷손실확률이 0.2 이상의 결과는 무의미하므로 본 논문에서 제시한 큐잉 모델은 정확하다고 할 수 있으며 이하는 큐잉 모델에 의한 결과를 제시한다.

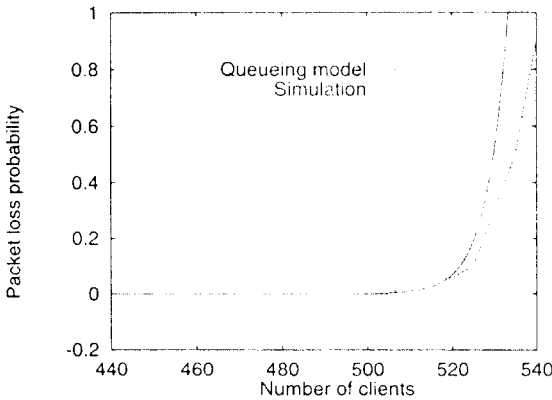


그림 3. 시뮬레이션과의 비교

주문형 비디오 서버의 성능에 가장 큰 영향을 미치는 변수는 디스크 대역폭( $R_d$ )과 접근망 대역폭( $R_n$ )이다. 특히 접근망의 대역폭은 서버노드 수와 밀접한 관련을 가지므로 서버의 설계시 고려해야할 부분이다. 즉,

$$R_n \geq R_d \times n$$

을 만족해야 한다. 디스크 대역폭과 접근망 대역폭이 성능에 미치는 영향을 그림 4에 나타내었다.  $z$ 축은 동시에 서비스 가능한 사용자의 수를 말하며 패킷손실률이 0.05이하의 사용자 수이다. 그림 4에서 보듯이

디스크의 대역폭과 접근망의 대역폭이 증가할수록 많은 사용자를 서비스 할 수 있다. 그러나, 두 대역폭 중 어느 한쪽이 병목이 될 수 있으며 가장 이상적인 조합을 유도해야 하며 이는 그림 4의 결과로부터 알 수 있다.

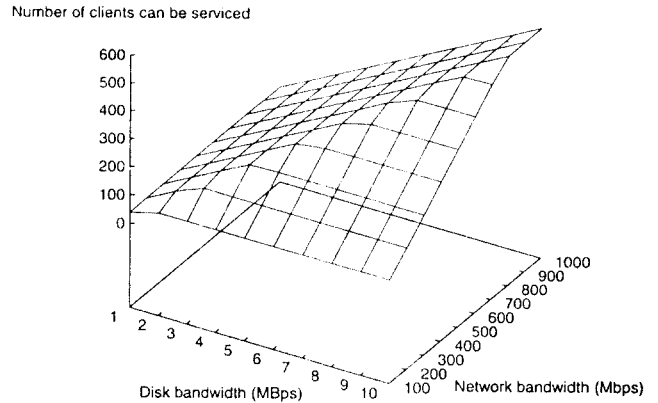


그림 4. 디스크와 네트워크 대역폭의 영향

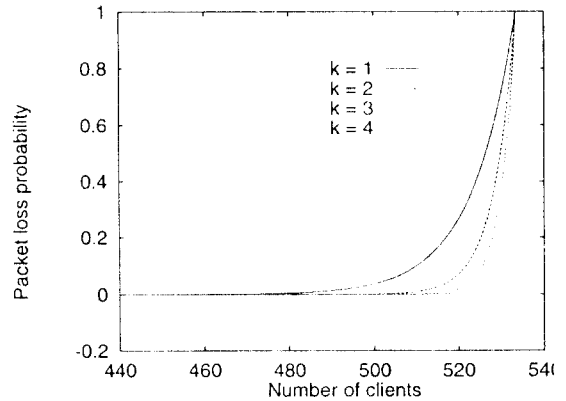


그림 5. 미리읽기 상수 k의 영향

그림 5는 미리읽기 상수  $k$ 의 영향을 나타낸다. 예측할 수 있듯이  $k$ 가 클 수록 좋은 성능을 보이나 그 효과는 점차 줄어들음을 알 수 있다. 그런데, 시스템의 성능에 영향을 미칠 것으로 예상되었던 데이터 패킷의 크기는 그림 6과 같이 성능에 거의 영향을 미치지 않는 것으로 나타났다. 이는 데이터 패킷의 크기가

커짐에 따라 스트림의 요청 시간 간격이 커지는 반면, 각 큐에서의 지연시간이 늘어나기 때문에 두 영향이 상쇄되어 같아지는 것으로 해석될 수 있다. 그러나 데이터의 크기는 디스크 서브 시스템의 성능에 큰 영향을 미치며 일반적으로 디스크의 효율적 이용을 위해 데이터의 크기는 어느 정도 이상이 되어야 한다. 이는 디스크 탐색시간이 데이터 전송시간보다 상대적으로 작아야 하기 때문이며 대부분 256KB 이상을 그 값으로 한다.

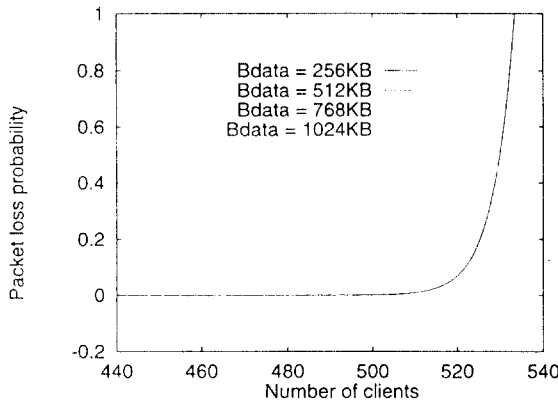


그림 6. 데이터 패킷 크기의 영향

서버노드의 버퍼는 미리읽기 상수  $k$ 와 데이터 패킷의 크기  $B_{data}$ 와 관련된다. 그림 5와 그림 6의 결과를 분석해보면, 데이터 패킷의 크기를 크게 하는 것보다  $B_{data}$ 를 작게 하고 미리읽기를 많이 하는 것이 유리할 것으로 사료되며 이는 가변 비트율 스트림을 고려할 때 타당하다.

### V. 결 론

본 논문에서는 대규모 주문형 비디오 서비스를 위한 서버의 큐잉 모델을 제시하였다. 수천편의 영화를 저장하기 위한 구조로서 중소형 컴퓨터로 구성된 분산 시스템을 대상으로 하였다. 대규모 서버를 네트워크 입력 큐, 네트워크 출력 큐, 각 서버노드 큐로 이루어지는 열린 큐잉 네트워크 모델에 의해 모델화 하여 비디오 스트림의 연속성을 보장하기 위한 사용자 패킷의 종료시한 만족여부를 구하였다. 다시 말하면, 패

킷의 손실 확률을 패킷의 응답시간을 이용하여 유도하고 여러 매개변수를 이용하여 성능 분석 결과를 제시하였다.

성능 분석 결과, 디스크 대역폭과 접근망의 대역폭에 병목현상이 발생하지 않는 값을 유도하여 서버를 설계해야 하며, 데이터 블록의 크기를 크게 하는 것 보다는 미리읽기를 많이 하는 것이 좋은 성능을 보임을 알 수 있었다.

한편, 서울대학교 컴퓨터공학과 컴퓨터시스템 연구실에서는 본 논문에서 제시한 시스템 모델에 기반하고 웹(WWW) 환경의 사용자 인터페이스를 갖춘 주문형 비디오 서비스 시스템의 프로토타입을 구축하였다.

### 참 고 문 헌

1. 박순철, 안희일, 임기욱, "멀티미디어 산업 동향 및 응용 기술," 주간기술동향 97-06, pp. 20-30, 1997.
2. S. Ghanderharizadeh and L. Ramos, "Continuous retrieval of multimedia data using parallelism," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 658-669, 1993.
3. A. Laursen, J. Olkin, and M. Porter, "Oracle media server: Providing consumer based interactive access to multimedia data," *Proc. of ACM SIGMOD 94*, pp. 194-201, 1994.
4. W. J. Bolosky, et. al., "The Tiger video filesaver," *Proc. of the 6th International Conference on Network and Operating System Support for Digital Audio and Video*, pp. 97-104, 1996.
5. A. Heybey, M. Sullivan, and P. England, "Calliope: A distributed, scalable multimedia server," *Proc. of USENIX 1996 Annual Technical Conference*, 1996.
6. J. Cho and H. Shin, "Scheduling algorithms in a large-scale VOD server," *Proc. of IPPS'97 Workshop on Parallel Processing and Multimedia*, pp. 17-25, 1997.
7. R. Tewari, et. al., "Design and performance tradeoffs in clustered video servers," *Proc. of International Conference on Multimedia Computing and*

*Systems*, pp. 144-150, 1996.

8. P. G. Harrison and N. M. Patel, *Performance Modelling of Communication Networks and Computer Architectures*, Addison Wesley, 1993.
9. L. Kleinrock, *Queueing Systems*, Volume I, John Wiley & Sons, Inc., 1975.
10. A. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*, 2nd Ed., Academic Press, Inc., 1990.
11. M. H. MacDougall, *Simulating Computer Systems : Techniques and Tools*, MIT Press, 1987.



조 진 성(Jin Sung Cho) 정회원  
1992년 2월: 서울대학교 컴퓨터  
공학과(공학사)  
1994년 2월: 서울대학교 컴퓨터  
공학과(공학석사)  
1994년 3월~현재: 서울대학교 컴  
퓨터공학과 박사과정  
※주관심분야: 멀티미디어 서버,

디스크 입출력 시스템, 실시간 시스템



신 현 식(Heon Shik Shin) 종신회원

1973년 2월: 서울대학교 응용물  
리학과(공학사)

1980년 12월: 미국 텍사스대학교  
의공학과(공학석사)

1985년 5월: 미국 텍사스대학교  
컴퓨터공학과(공학  
박사)

1986년~현재: 서울대학교 컴퓨터공학과 교수

※주관심분야: 분산 시스템, 실시간 시스템, 멀티미디  
어 시스템, 병렬 시스템, 결합허용 시  
스템