

---

# MCVoD system의 향상된 성능 평가 - Server Channel 할당 기법 개선

강 석 훈\*

Performance Analysis of an Improved Channel Allocation Scheme in MCVoD System

Seok-Hoon Kang\*

---

이 논문은 2006년도 인천대학교 연구비를 지원받았음

---

## 요 약

본 논문에서는 MCVoD system의 VoD Server Channel 할당 기법 변경을 통한 성능향상에 대해 논하고, 이에 따른 MCVoD system의 전체 성능을 분석하도록 한다. 또한 시뮬레이션을 통해, 동일한 VoD Server Channel 할당 기법을 사용하는 타 VoD system과 성능을 비교해봄으로써, VoD Server 및 Interchange Agent의 동시 사용에 따른, MCVoD system 성능을 평가한다.

## ABSTRACT

This paper discusses improvement of MCVoD System by changing the channel allocation of VoD Server. And we analyzed performance of MCVoD System. Using simulation, we compare MCVoD with another VoD server using same channel allocation. In case of average channel allocation, MCVoD System using interchange agent, represents three times better than previous VoD server.

## 키워드

VoD, Interchange Agent, Channel Allocation

## I. 서 론

MCVoD (Multicast and Cache Video-on-Demand) 시스템은 멀티캐스트 기법과 프록시의 캐시를 이용한 Video-on-Demand 시스템이다[1]. MCVoD 시스템은 멀티캐스팅을 통해 동일한 사용자의 비디오 요청을 하나로 묶어 전송하며, 멀티캐스팅에 의해 발생하는 시간 지연 문제를 프록시의 캐시를 이용하고, 요청한 비디오의

첫 번째 비디오 스트림을 전송함으로써, 실시간으로 VoD 서비스를 제공할 수 있는 시스템이다. 또한 Interchange Agent (IA)를 이용하여 다중 프록시를 관리함으로써, 프록시 캐시의 사용 효율을 높이고 있으며, 다중 프록시의 임시 저장 장소를 이용하여, 사용자의 인기 비디오 순위 경향을 반영하도록 하고 있다. 이러한 IA를 통한 MCVoD 시스템은 VoD 서비스를 제공함에 있어, 높은 효율성을 제공하게 된다[1].

MCVoD 시스템의 성능 향상을 위한 방안으로, 본 논문에서는 VoD 서버의 채널 할당 기법 개선을 통해 전체 성능 향상을 보인다. 현재 MCVoD 시스템의 VoD 서버 채널 할당 기법으로는 **Batching** 기법을 사용하고 있다. 해당 채널 할당 기법은 채널을 할당함에 있어, 다른 할당 기법에 비해, 채널의 낭비를 가져오는 잠재적인 문제점을 내포하고 있으며, 따라서 이를 개선함으로써, VoD 서버의 성능 향상 및 나아가 MCVoD 시스템의 전체 성능을 향상 시킬 수 있다.

본 논문에서는 MCVoD 시스템 중 개선된 VoD 서버 채널 할당 기법 및 성능 분석을 하도록 하며, 해당 기법 개선에 따른 추가 비용에 대한 MCVoD 시스템의 전체 성능을 분석하도록 한다. 또한 시뮬레이션을 통해 VoD 서버 채널 할당량 및 MCVoD 시스템 전체 비용을 비교 하도록 하며, 동일한 VoD 서버 채널 할당 기법을 사용하는 타 VoD 시스템과 성능을 비교해봄으로써, VoD 서버 및 **Interchange Agent**의 동시 사용에 따른 MCVoD 시스템 성능을 비교하도록 한다.

## II. 본론

### 채널 할당 기법 개선 및 성능 분석

MCVoD 시스템의 VoD 서버의 채널 할당은 **Batching** 기법을 사용하고 있다. **Batching**은 VoD 서비스를 위한

서버 채널 할당 알고리즘이 간단해 질 수 있다는 장점이 있는 반면, 서비스 시작 직후, 동일 서비스 요청이 들어올 경우, 새로운 **Batching Window** 후에 하나의 채널을 할당함으로써, VoD 서버 채널의 낭비를 가져올 수 있다는 단점도 있다[2,3,4,5].

본 장에서는 개선된 VoD 서버의 채널 할당 기법들에 대해 알아보며, 이의 적용을 통한 VoD 서버 성능을 분석 하도록 한다. 덧붙여, VoD 서버 성능 향상에 따른 MCVoD 시스템의 전체 시스템 성능 향상에 대해서도 분석 하도록 한다.

#### 2.1. VoD 서버 채널 할당 기법

변경된 MCVoD 시스템의 VoD 서버 채널 할당 기법은 **Batching**, **Patching**, 그리고 **Batching Patch** 기법을 모두 사용하고 있다. 해당 기법들을 MCVoD 시스템상에 적용하기 위해서는, 클라이언트에 **Patching**에 필요한 버퍼를 반드시 제공해야 하며, 이와 관련한 시스템 비용이 발생하게 된다. VoD 서버의 향상된 채널 할당 기법은 다음과 같다.

##### 2.1.1. MCVoD 시스템의 Batching

VoD 서버 채널 할당 기법 중 **Batching** 기법은 그림1과 같다. VoD 서버가 사용자의 비디오 요청을 받은 시간을 **u1**이라 하고, **pt**의 시간 이전에 동일 비디오 요청을 위한 채널 할당이 이루어지지 않았다면, VoD 서버는 **x+u1**시

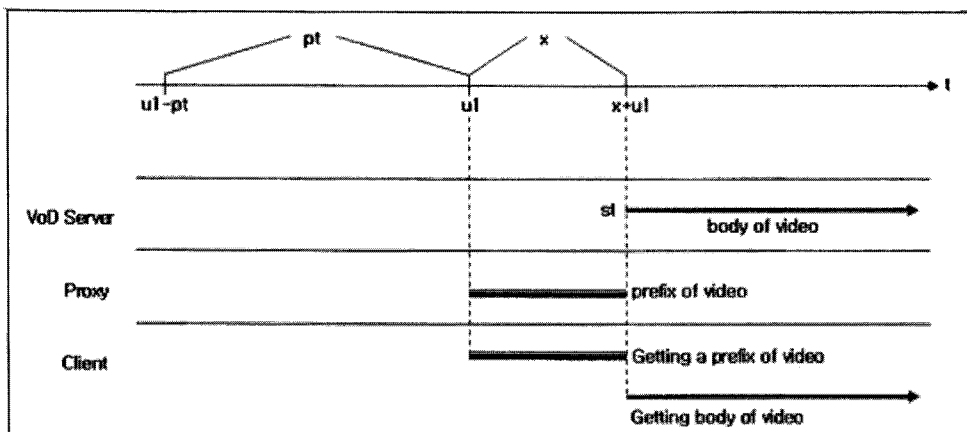


그림 1. VoD 서버 채널 할당의 배치 방법  
Fig. 1 Batching Method of VoD Server Channel Allocation

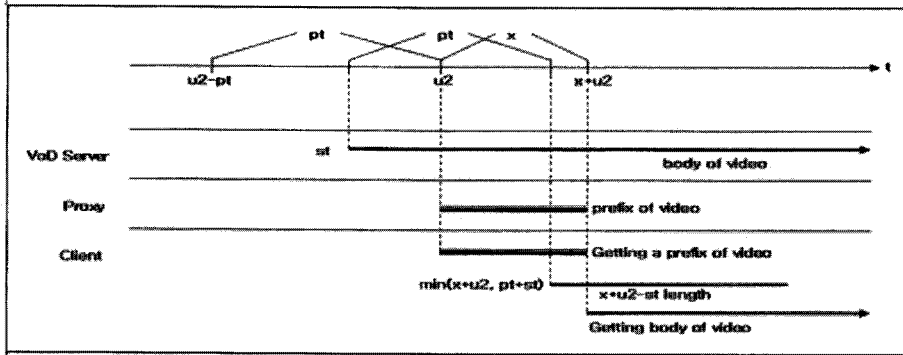


그림 2. VoD 서버 채널 할당의 패치 방법  
 Fig. 2 Patching Method of VoD 서버 채널 Allocation

간에 하나의 유니캐스트 채널을 할당하여 사용자에게 비디오를 서비스 시작한다.  $pt$ 는 Patching Window의 크기를 나타내며,  $x$ 는 프록시에 저장된 인기 비디오 Prefix의 길이를 나타낸다. 클라이언트는 다중 프록시로부터 비디오 Prefix를 받아 비디오 상영을 시작하며, 해당 Prefix가 끝나는  $x+u1$  시점에서 VoD 서버로부터 전송된 Video Body를 받아 상영함으로써, 실시간 서비스를 구현할 수 있다.

만약 사용자가 비인기 비디오를 요청했을 경우, VoD 서버는  $x+u1$  시간에 비디오 전체의 길이(비디오 Prefix+비디오 body)를 전송해야 하며, 이로 인해 발생하는  $x$  시간만큼의 지연시간 처리를 위해,  $x$ 의 값은 사용자의 비디오 요청 이탈 시간을 넘지 않는 범위 내에서 결정되어야 한다.

2.1.2. MCVoD 시스템의 Patching

VoD 서버 채널 할당 기법 중 Patching 기법은 그림2와 같다. VoD 서버가 사용자의 비디오 요청 받은 시간을  $u2$  라 하고,  $pt$  시간 이전에 동일비디오에 대한 채널이  $st$  (Service Time) 시간에 할당되었다면, VoD 서버는  $\min(x+u2, pt+st)$  시간에 하나의 Patching 채널을 할당하여  $x+u2-st$  길이만큼의 비디오를 전송하며,  $x+u2$  시간 이후의 비디오는 기존 채널에 묶어 해당 비디오 서비스를 시작한다. Patching 채널은  $x+u2-st$  시간 길이만큼의 서비스 후 사용 가능한 채널로 반환되게 된다. 클라이언트는 다중 프록시로부터 비디오 Prefix를 받아 비디오 상영을 시작하며, 해당 Prefix가 끝나는  $\min(x+u2, pt+st)$  시점에서 VoD 서버로부터 전송된 Video Body를 받아 상영함으로써, 실시간 서비스를 구현할 수 있다.

만약 사용자가 비인기 비디오를 요청했을 경우, VoD 서버는  $\min(x+u2, pt+st)$  시간에 비디오 전체의 길이(비디오 Prefix+비디오 body)를 전송해야 한다. 이로 인해 발생하는  $x$  시간만큼의 지연시간 처리를 위해,  $x$ 의 값은 사용자의 비디오 요청 이탈 시간을 넘지 않는 범위 내에서 결정되어야 한다.

2.1.3. MCVoD 시스템의 Batching Patch

VoD 서버 채널 할당 기법 중 Batching 기법은 그림3과 같다. VoD 서버가 사용자의 비디오 요청 받은 시간을  $u3$  라 하고,  $u2$  시간에 동일 비디오를 위한 patching 채널을 이미 스케줄링 하였다면, VoD 서버는  $\min(x+u2, pt+st)$  시간에 동일한 Patching 채널을 통해  $u3$  시간에 들어온 요청을 처리할 것이다. Client는  $u3$  시간부터  $x+u3$  시간 까지 다중 프록시로부터 비디오 Prefix를 전송 받으며, 나머지는  $u2$ 의 요청과 동일하게 처리된다.

2.2. 채널 할당 기법 변경을 통한 VoD 서버 성능 분석

MCVoD 시스템은 그림4와 같이 전체  $N$ 개의 비디오를 VoD 서버가 저장하고 있으며, 그 중  $K$ 개의 인기 비디오의 첫 번째 비디오 스트림 30개를 다중 프록시의 캐시에 저장하고 있다. 첫 번째 비디오 스트림의 길이는  $x$ 로,  $K$ 개(30개)의 인기 비디오의 경우 전체 스트림의 길이  $L$ 에서  $x$ 를 뺀 길이만큼을 서버에 저장하고 있다. 또한 비인기 비디오의 경우, 다중 프록시에 첫 번째 비디오 스트림이 저장되어 있지 않으므로, 서버상의 비인기 비디오의 전체 스트림의 길이는  $L$ 이다.

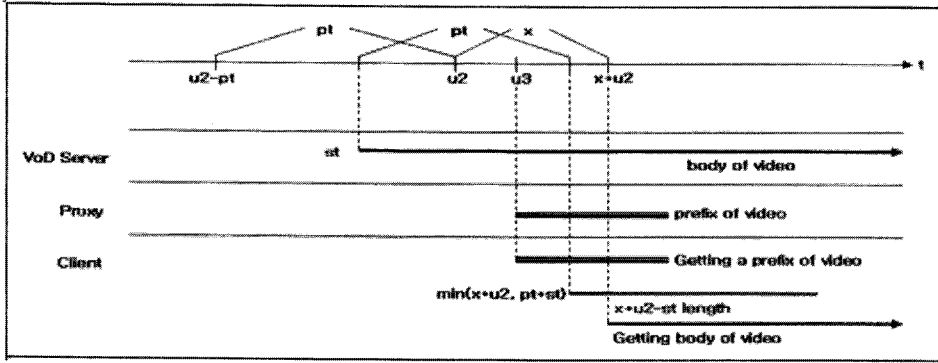


그림 3. VoD 서버 채널 할당의 배치 패치 방법  
Fig. 3 Batching Patch Method of VoD 서버 채널 Allocation

VoD 서버의 사용자 비디오 요청은 인기 비디오를 위한 요청과 비인기 비디오를 위한 요청으로 나누어 볼 수 있으며, 따라서 VoD 서버의 전체 채널 할당량( $S_T$ )은 아래와 같다.

$$S_T = \text{인기비디오를 위해 할당된 채널의 수} + \text{인기 비디오를 위해 할당된 채널의 수}$$

VoD 서버는 하나의 인기비디오를 위해 Patching window (PW)동안, 서버가 할당하는 Patching 채널의 수는 VoD 서버가 전송해야 하는 비디오 전체 길이를 Patching Window로 나눈  $L-x/PW$  과 같다. 사용자의

인기비디오 요청이 Poisson 분포 및 Zipf 분포를 따른다고 가정하고 비디오 요청 도착율을  $\lambda$ 라고 할 때,  $i$ 번째 인기비디오를 위한 사용자의 요청 도착율( $\lambda_i = \rho_i \times \lambda$ )에 대한 서버의 채널 할당 수식은 수식1과 같으며,  $k$ 개의 인기비디오를 위한 서버의 채널 할당 수식은 수식2와 같다.

$$\frac{L-x}{PW}, \max(1, I_i) \tag{1}$$

$\lambda_i$ 는 요청 도착율에 Poisson 분포 공식을 적용함으로써, 나오는 일반 수식이며, VoD 서버가 전송해야 하는

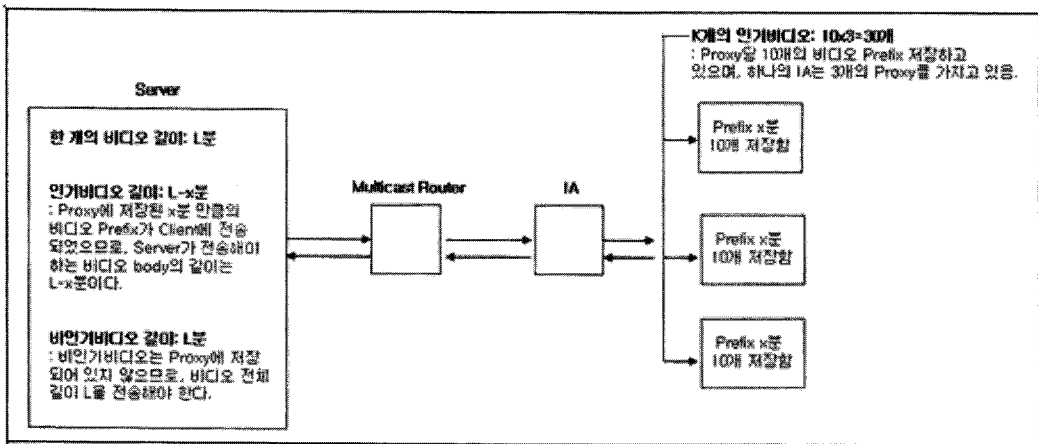


그림 4. MCVoD 시스템의 구조  
Fig. 4 Structure of MCVoD 시스템

video body의 길이(L-x)를 patching 시간으로 나눈 것에, 각patching 시간에 특정 비디오 i가 도착할 확률을 곱함으로써, 특정 비디오 i에 대한 VoD 서버 채널 할당 수를 구할 수 있다.

$$S_p = \prod_{i=1}^K \left( \frac{L-x}{PW} \max(1, I_i) \right) \quad (2)$$

수식(1)에서 특정 비디오 i를 위한 채널 할당 수를 구하였으므로, 수식1에 K개만큼 더함으로써, 인기비디오 K개를 위한 전체 채널수를 구할 수 있다.

또한 N-K개의 비인기 비디오를 위한, 서버의 전체 채널 할당 수식은 수식(3)과 같다.

$$S_U = \prod_{i=K+1}^N \left( \frac{L}{PW} \max(1, I_i) \right) \quad (3)$$

비인기비디오의 경우, VoD 서버가 클라이언트로 보아야 하는 video body의 길이는 L분(L Minutes)이다. 따라서 VoD 서버가 전송해야 하는 video body의 길이(L)를 patching 시간으로 나눈 것에, 각 patching 시간에 특정 비인기비디오 i가 도착할 확률을 곱함으로써, 특정 비디오 i에 대한 VoD 서버 채널 할당 수를 구할 수 있으며, 이를 비인기 비디오 N-K개만큼 더함으로써, 비인기 비디오를 위한 전체 채널 할당수를 구할 수 있다.

따라서 위의 수식을 이용하여 VoD 서버 전체 채널 할당량(S<sub>T</sub>)을 구하면 수식(4)와 같으며, VoD 서버 전체 채널 할당량을Patching Window의 횟수 (NoPW)로 나눔으로써, VoD 서버의 평균 채널 할당량(S<sub>a</sub>)를 수식(5)와 같이 구할 수 있다.

$$S_T = \prod_{i=1}^K \left( \frac{L-x}{PW} \max(1, I_i) \right) \prod_{i=K+1}^N \left( \frac{L}{PW} \max(1, I_i) \right) \quad (4)$$

NoPW의 값은 VoD 서버가 동작하는 전체 시간의 미한다. 따라서, 평균을 내기 위해, 전체 시간을 서버 동작 주기 시간으로 나눔으로써 NoPW(Number Patching

Window)를 구했으며, VoD 서버 전체 채널 할당수를 NoPW로 나눔으로써, VoD 서버 평균 채널을 구할 수 있다.

### 2.3. VoD 서버 성능 향상에 따른 MCVoD 시스템 성능 분석

MCVoD 시스템의 전체 비용(C<sub>sys</sub>)은 VoD 서버의 변경된 채널 할당 기법 적용 및 클라이언트의 버퍼 영역에 따른 비용을 추가함으로써 아래와 같이 나타낼 수 있다.

이때 멀티캐스트 라우터 및 프록시의 유니캐스트 채널의 비용은 전체비용에서 제외하도록 하며, 네트워크 비용은 해당 개체의 전체 채널로 나타낸다.

$$\begin{aligned} C_{sys} = & \text{서버비용(네트워크비용+저장비용)} \\ & + \text{IA비용(네트워크비용)} \\ & + \text{다중프록시비용(첫 번째 스트림 저장시간비} \\ & \text{용+cache저장공간비용)} \\ & + \text{Client비용(Buffer 저장공간비용)} \end{aligned}$$

따라서 VoD 서버의 네트워크 비용은 수식4의 S<sub>T</sub>로 나타낼 수 있다.

클라이언트의 버퍼 저장공간비용의 경우, Patching 기법 사용에 따른 것으로, 하나의 프록시에 연결 가능한 클라이언트 수를 Cn이라 하고, Client상의 Buffer 길이를 z로 나타낸다면, 하나의 프록시당 클라이언트의 버퍼 저장공간비용은 수식6과 같다.

$$\sum_{i=1}^{Cn} z_i \quad (6)$$

n개의 프록시당 클라이언트 버퍼 저장공간 비율은  $\sum_n \left( \sum_{i=1}^{Cn} z_i \right)$  이 된다.

위의 수식을 적용한 MCVoD 시스템 전체 비용(C<sub>sys</sub>)은 수식7로 나타낼 수 있다.

$$C_{sys} = S_T + \alpha \left( \sum_{i=1}^N L_i \right) + \sum_n \left[ IA_{nm} + \sum_i \left( \beta_i + \omega_i \sum_{i=1}^{Cn} z_i + \sum_{i=1}^{Cn} z_i \right) \right] \quad (7)$$

이때 각 항목은 다음과 같다.

-  $S_T$ : 서버 전체 채널 할당수

-  $\alpha \left( \sum_{i=1}^N L_i \right)$ :  $\alpha$ 는 VoD 서버가 비디오를 저장함에 있어 걸린 시간이며, 나머지 수식부분은 하나의 비디오 길이  $L$ 분이라 하고,  $N$ 개의 전체 비디오를 저장했을 경우 저장공간비용을 나타낸다.

-  $IA_m$ : 시스템상에  $m$ 개의 IA가 존재하고, IA 하나의 전체 채널 할당수 ( $IA_T$ )라고 했을 때,  $m$ 개의 IA 네트워크 비용을 나타낸다.

-  $\beta_n$ : 하나의 IA에  $n$ 개의 프록시가 존재할 경우 (시스템 전체에  $m \times n$ 개의 프록시가 존재함), 프록시 하나의 첫 번째 비디오 스트림 10개를 저장하기 위한 시간 비용을 나타낸다.

-  $\omega_n \sum_{i=1}^{11} x_i$ :  $\omega_n$ 은 첫 번째 스트림 11개를 저장하는 공간 비용이며, 나머지는 총 11개의 첫 번째 스트림의 길이 ( $x$ )를 모두 더한 값이다. 따라서 전체적으로 프록시 하나의 11개 첫 번째 스트림을 저장하기 위한 저장공간을 나타낸다.

-  $\sum_{i=1}^{C_n} Z_i$ : Client상의 Buffer 저장공간을 나타낸다.

### III. Simulation을 이용한 성능 분석

본 장에서는 시뮬레이션을 통해, VoD 서버 채널 할당 기법 변경에 따른 성능을 분석하도록 한다. 성능 분석의

첫 번째 단계로 MCVoD 시스템상에서 VoD 서버 채널 할당 기법만을 변경했을 경우, 동일한 사용자 비디오 요청 수에 따른, VoD 서버 채널 할당량을 비교 분석 하도록 한다. 두 번째, 클라이언트 상의 저장영역 추가로 인한 MCVoD 시스템 전체 비용 변화량을 비교 분석 하도록 한다. 세 번째, 개선된 VoD 서버 채널 할당 기법을 동일하게 사용하는 VoD 시스템과 MCVoD 시스템의 성능 비교를 통해, VoD 서버 및 Interchange Agent의 동시 사용에 따른 그 효율성을 입증하도록 한다.

#### 3.1. VoD 서버 채널 할당량 비교 분석

MCVoD 시스템의 VoD 서버 채널 할당 기법을 Batching 기법에서 Batching, Patching, 및 Batching Patch로 변경함으로써, VoD 서버 성능을 알아보도록 한다. 그림5, 그림6은 사용자의 요청에 따른 VoD 서버 전체 채널 할당량 및 빈번하게 발생하는 인기비디오를 위한 채널 할당량에 대한 시뮬레이션 결과를 나타내고 있다.

이 결과에서 개선된 VoD 서버 채널 할당 효율이, 기존 VoD 서버 채널 할당 효율보다, 3배 정도 높음을 알 수 있으며, 사용자의 요청이 많아짐에도 불구하고, 개선된 VoD 서버 채널 할당량은 사용자 요청 증가에 비해, 소폭으로 상승함을 보이고 있다. 이 결과를 통해, 개선된 VoD 서버 채널 할당 기법이 기존보다 성능이 개선되었음을 알 수 있다.

#### 3.2. MCVoD 시스템 전체 비용 변화량 비교 분석

VoD 서버 채널 할당 방식 개선에 따라, Client의 Buffer가 추가됨으로써, MCVoD 시스템의 전체 비용 변화량을 분석 하도록 한다. 아래의 그림7은 개선된 VoD

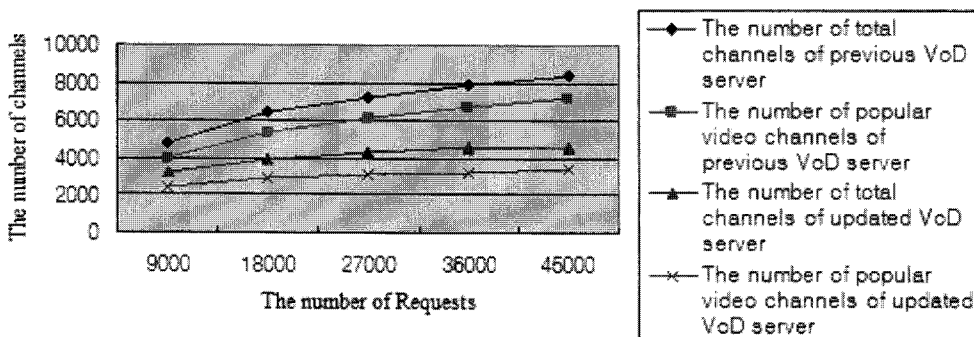


그림 5. MCVoD 시스템의 VoD 서버 채널 할당량 비교  
Fig. 5 Compare VoD 서버 채널 Allocation of MCVoD 시스템

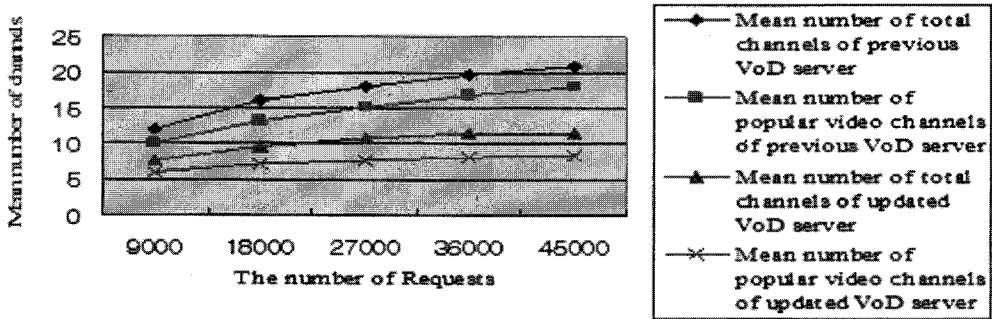


그림 6. MCVoD 시스템의 VoD 서버 평균 채널 할당량 비교  
 Fig. 6 Compare VoD 서버 Average 채널 Allocation of MCVoD 시스템

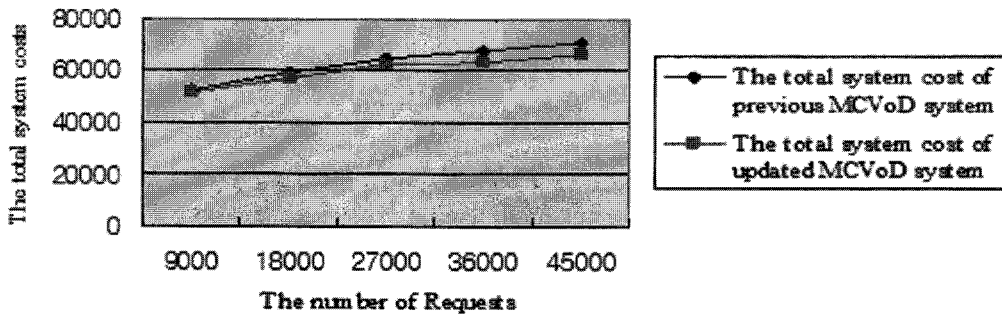


그림 7. MCVoD 시스템 전체 시스템 비용 비교  
 Fig. 7 Compare MCVoD 시스템 total 시스템 cost

서버 채널 할당 기법을 적용한, MCVoD 시스템 전체 비용을 비교하여 나타내고 있다.

위의 결과에서, 개선된 VoD 서버 채널 할당 기법에 따른 VoD 서버 네트워크 비용의 감소가, Client의 저장 영역 추가 비용 증가에도 불구하고, MCVoD 시스템 전체 비용이 줄어들었음을 알 수 있다. 따라서 개선된 VoD 서버 채널 할당 기법의 효율이 높음을 알 수 있다.

### 3.3. VoD 서버와 Interchange Agent 동시 사용에 따른 성능 비교

동일한 VoD 서버 채널 할당 기법을 사용하는 VoD 시스템과 VoD 서버 및 Interchange Agent를 모두 사용하는 MCVoD 시스템의 성능을 비교 분석한다.

그림8과 그림9는 사용자의 요청에 따른, VoD 서버의 전체 채널 할당량과 요청이 빈번한 인기비디오에 대한 채널 할당량을 보이고 있다.

아래의 결과에서, 동일한 VoD 서버 채널 할당 기법을

사용하는 VoD 서버보다, MCVoD 시스템의 VoD 서버의 성능이 더 좋음을 알 수 있다.

## IV. 결 론

MCVoD 시스템[1]은 멀티캐스팅 방식과 Interchange Agent를 이용한 실시간 Video-on-demand 서비스를 제공하는 VoD 시스템이다. Interchange Agent를 통한 MCVoD 시스템의 성능 평가[1]는 앞서 발표되었으며, 본 논문에서는 MCVoD 시스템의 성능을 개선할 수 있는 방안을 연구하였다.

MCVoD 시스템의 성능 향상을 위해, 본 논문은 VoD 서버의 채널 할당 기법의 변경을 통해, 전체적인 시스템 성능이 향상됨을 보였다. 기존의 MCVoD 시스템에서 사용하는 채널 할당 기법으로는 Batching 기법을 사용하였으며, 변경을 통한 Batching 기법의 잠재적인 단점을

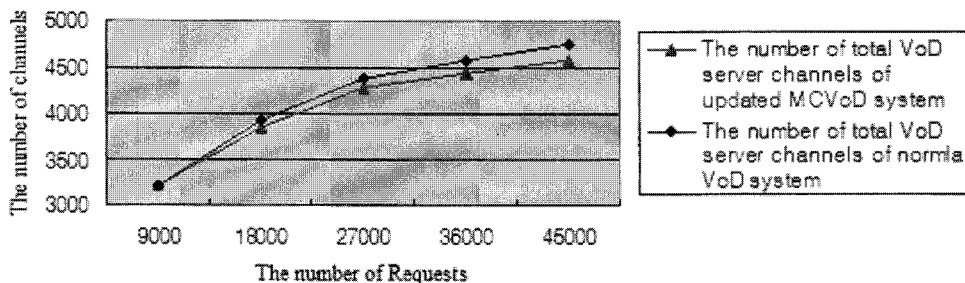


그림 8. VoD 서버 채널 할당량 비교  
Fig. 8 Compare VoD 서버 채널 Allocation

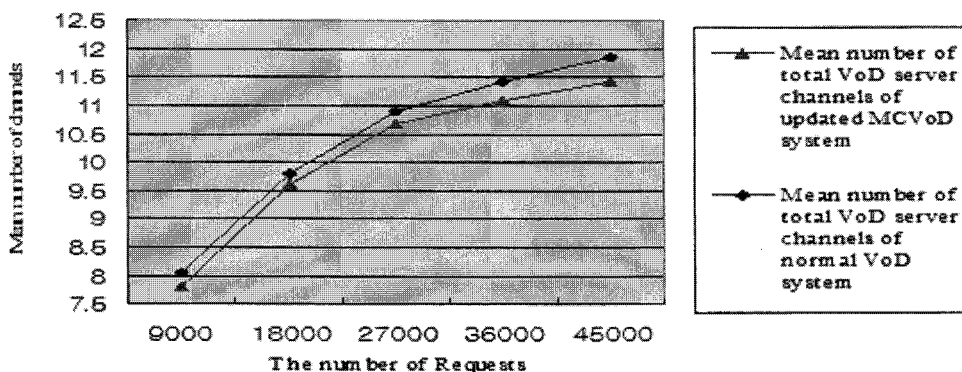


그림 9. VoD 서버 평균 채널 할당량 비교  
Fig. 9 Compare VoD 서버 average 채널 Allocation

보완함으로써, VoD 서버 및 전체 시스템의 성능을 향상시킬 수 있었다. 향상된 VoD 서버의 채널 할당 기법으로는 **Batching**, **Patching** 그리고 **Batching Patch** 방식을 사용하였다.

변경된 VoD 서버 채널 할당 기법은 **Batching** 방식의 문제점을 보완한 방식으로, 동일한 비디오 요청이 **Patching Window** 내에 다시 요청될 때, 지나간 시간만큼의 비디오 스트림을 **Patching** 채널 할당을 통해 전송하며, 남은 비디오 **Body**를 기존 채널에 묶어 전송하는 방식을 사용하고 있다. 또한 동일한 비디오 요청이 같은 **Patching** 채널 할당 전에 다시 들어올 경우, 이를 하나의 **Patching** 채널로 묶어 할당 하는 방식을 사용하고 있다. 따라서 이러한 **Patching** 채널은 일정 시간 이후, 사용 가능한 채널로 반환됨으로써, VoD 서버의 성능을 향상되었으며, 이를 통해, MCVoD 시스템의 전체 성능이 향상되었음을 확인하였다.

본 논문에서는 VoD 서버 채널 할당 기법을 **Batching** 기법에서 **Batching**, **Patching** 및 **Batching Patch** 기법으로의 변경에 따른 VoD 서버의 성능을 분석하였으며, 분석 모델을 기반으로 기존 **Batching** 기법을 사용하는 MCVoD 시스템과의 시뮬레이션을 이용한 성능 비교를 통해, 기존 MCVoD 시스템 보다 성능 및 비용이 향상되었음을 입증하였다. 또한 동일한 채널 할당 기법을 사용하는 타 VoD 시스템과의 시뮬레이션을 이용한 성능 비교를 통해, VoD 서버 및 **Interchange Agent**의 동시 사용에 따른 효율성을 입증하였으며, 나아가 MCVoD 시스템의 성능이 동일한 VoD 서버 채널 할당 기법을 사용하는 타 VoD 시스템 보다 성능향상이 이루어 졌음을 보였다.



참고문헌

- [1] SeokHoon Kang, IkSoo Kim, Yoseop Woo "Analysis of Performance for MCVoD System" ICCSA 2004 pp. 270-277
- [2] Ramesh, S. Injong Rhee, Guo, K. "Multicast with cache (Mcache): an adaptive zero-delay video-on-demand service" IEEE, INFOCOM 2001
- [3] Back Hyun Kim, Tae June Hwang, Ik Soo Kim. "Multicast VOD System for Interactive Services in the Head-End-Network" 정보처리학회논문지B, Vol.11, No.3
- [4] K. A. Hua, Y. Cai, and S. Sheu. Patching: A multicast technique for true video-on-demand services. In Proc. of ACM SIGCOMM, Sept 1997.
- [5] L. gao and D. Towsley. Supplying instantaneous video-on-demand services using controlled multicast. In Proc. of IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, Jun 1999.

저자소개



강 석 훈(Seok-Hoon Kang)

한양대학교 전자통신공학과 공학박사  
(1995)

인천대학교 멀티미디어시스템공학과  
부교수(2004~)

※ 관심분야: 임베디드시스템, 멀티캐스트, RTOS