

# 다양한 클라이언트 요청율에 적용가능한 주문형 비디오 스트리밍을 위한 동적 스킴 트랜지션 기법

(Dynamic Scheme Transition for On-Demand Video  
Streaming Adaptable to Variable Client Request Rate)

김 현 주 <sup>†</sup>                      엄 현 영 <sup>\*\*</sup>  
(Hyun-Joo Kim)                  (Heon-Young Yeom)

**요 약** 네트워크를 통한 주문형 비디오 스트리밍 서비스를 제공하기 위해서는 클라이언트의 서비스 요청이 많은 비디오 뿐만 아니라 서비스 요청이 적은 비디오에 대한 서비스도 효율적으로 제공할 수 있어야 한다. 본 논문에서는 클라이언트의 다양한 서비스 요청율에 대해 동적으로 서비스 스킴을 변환시켜가면서 클라이언트의 QoS를 비슷한 수준으로 유지시키고 효율적으로 서비스를 제공할 수 있도록 하는 동적 비디오 스트리밍 기법을 제안한다. 클라이언트의 서비스 요청이 적을 때는 휴리스틱 기법에 의해 서비스를 제공하고 서비스 요청이 많을 때는 주기적 방송(Periodic Broadcasting) 기법 중 Striping Broadcast를 사용하여 서비스를 제공한다. 서버는 사용하는 채널 수로 클라이언트 요청율의 변화를 감지하며 이에 따라 더 효율적인 기법으로 서비스 스킴을 전환한다. 이 같은 스킴 트랜지션(scheme transition)에 대해 서비스 요청율을 변화시킨 것과 현재 스트리밍 서비스를 제공하고 있는 업체에서 얻은 트레이스를 통해 시뮬레이션을 수행하였으며 합당한 수준의 성능이 나옴을 확인할 수 있었다.

**키워드** : 주문형 비디오 스트리밍 서비스, 주기적 방송, Striping 방송, 서비스 요청율

**Abstract** To provide on-demand video streaming services through the network, video objects with both high and low client request rates should be served efficiently. In this paper, we propose a dynamic on-demand streaming approach for providing services efficiently regardless of the client request rate. This approach can maintain QoS by transitioning the service scheme according to the request rate. The server provides services by the Heuristic broadcasting approach when the client request rate is low and by Striping broadcast, one of many Periodic broadcast approaches, when the client request rate is high. The server knows the variations of client request rates by the number of service channels and decides the transition to more efficient service schemes on its own. We simulated this scheme transition with various client request intervals and traces from a company providing streaming services. The results show that the performance of our approach is reasonable.

**Key words** : On-demand Video Streaming Service, Periodic Broadcast, Striping Broadcast, Client Request Rate

## 1. 서 론

네트워크로 비디오 서비스를 제공하기 위해서는 스트리밍 해주는 서버와 서비스 받는 클라이언트가 존재한다. 클라이언트 측면에서는 버퍼링과 디코딩을 위한 캐

싱에 대해 많은 연구가 이루어져왔으며, 서버 측면에서의 연구는 크게 실시간 스트리밍을 위한 인코딩, 동일 비디오에 대한 서비스 성능을 개선하기 위한 캐싱, 한정된 서버 대역폭으로 많은 수의 클라이언트에게 서비스를 제공하기 위한 서비스 측면으로 나눌 수 있다.

서비스 측면에서 지금까지 연구되어온 기법들은 유니캐스트, 멀티캐스트, 패칭(patching), 주기적 방송(periodic broadcast)으로 분류해볼 수 있다. 멀티캐스트는 현재 일반적인 인터넷 환경에서는 사용할 수 없기 때문에 오버레이(overlay) 네트워크로 트리를 구성해서 클라

<sup>†</sup> 학생회원 : 서울대학교 컴퓨터공학과  
hjkim@dcslab.snu.ac.kr

<sup>\*\*</sup> 종신회원 : 서울대학교 컴퓨터공학과 교수  
yeom@snu.ac.kr

논문접수 : 2005년 10월 14일

심사완료 : 2006년 3월 24일

이언트가 다른 클라이언트에게 서비스를 제공하는 형태의 연구가 많이 이루어지고 있다. 이 경우 늦게 서비스 요청을 한 클라이언트는 그 이전의 비디오 데이터를 받지 못하는데 이 부분은 서버와 따로 유니캐스트 채널을 형성하여 받는다. 이렇게 함으로써 서버는 전체 비디오 데이터 재생 시간만큼 클라이언트와 채널을 형성하는게 아니라 앞부분의 데이터에 대한 짧은 시간 동안만 채널을 형성하면 되므로 더 많은 수의 클라이언트에게 서비스를 제공할 수 있게 된다. 이를 패칭이라고 한다. 오버레이 네트워크로 비디오 데이터를 서비스 하는 기법들의 단점은 기반 네트워크 연결구조(topology)를 고려하지 않으면 비효율적인 노드간 경로가 형성되어 대용량의 비디오 데이터가 네트워크를 이동하면서 대역폭을 낭비할 수 있다는 것이다.

주기적 방송 기법에서는 각 비디오를 서비스하기 위해 일정 수의 멀티캐스트 채널을 할당하고 비디오를 세그먼트로 나눈 후 각 채널에 정해진 세그먼트를 주기적으로 반복하여 전송한다. 클라이언트에서는 채널을 바꿔가며 세그먼트를 받게 되는데, 하나 이상의 채널에 동시에 접근해서 데이터를 받게 된다. 또한 클라이언트에서는 현재 재생하는데 필요한 데이터가 아니면 디스크에 저장해야 하므로 저장공간을 많이 필요로 하게 된다. 채널을 많이 할당할수록 클라이언트에서 서비스 받기 위해 기다리는 시간이 줄어들게 되지만 서버에서는 하나의 비디오를 서비스하기 위해 필요한 대역폭이 늘어나게 되고 클라이언트에서 미리 받게 되는 데이터가 많아지므로 필요한 저장공간이 커진다. 이 기법은 클라이언트의 요청이 많을 때 유리하다.

최근에 제안된 Striping Broadcasting(StB)[1]에서는 클라이언트가 최대 3개의 서버 채널에만 동시에 연결하면 되고 striping을 통해 클라이언트의 저장공간 요구량을 줄일 수 있으며, 클라이언트의 초기 지연시간(initial delay)도 줄일 수 있다. 저자들은 StB를 Harmonic[2], Fast[3], Pagoda[4], Skyscraper[5]등의 주기적 방송 기법들과 수치적으로 비교하고 있는데, 서비스를 제공받기 위해 비디오 데이터 크기의 25%만큼의 클라이언트 저장공간을 필요로 하며 이는 Fast의 50% 요구량의 절반에 불과하다. StB는 구현까지 되었는데, 구현후 실험한 결과는 클라이언트 초기 지연시간과 클라이언트 저장공간 모두 수치분석 값보다는 크게 나왔다. 주기적 방송 기법은 특성상 구현이 어렵다는 문제점이 있어서 구현된 기법이 거의 없는데 StB이외에 구현된 주기적 방송 기법으로는 Fast가 있다[6].

클라이언트의 서비스 요청이 왔을 때에만 세그먼트를 스케줄링하여 전송하는 휴리스틱 기법들도 제안이 되었는데, 각 세그먼트는 클라이언트에서 재생되기 이전에만

전송되면 된다. 이와 관련된 기법들은 요청율이 적은 경우에 주로 유용하며 세그먼트를 미리 받음(prefetching)으로 인한 저장공간이 많이 필요하고 서비스 요청율이 높은 경우에는 서버에서 필요로 하는 대역폭이 주기적 방송 기법 보다도 많아지는 단점이 있다.

최근에는 클라이언트의 요청율이 낮은 경우와 높은 경우 모두를 동시에 고려하기 위해 채널 트랜지션(channel transition)[7-9]기법들이 제안되고 있는데, 주기적 방송 기법 중 하나를 기본으로 하여 서비스 도중 클라이언트 요청율에 따라 동적으로 채널할당을 늘이거나 줄임으로써 서버 대역폭을 효율적으로 사용하고자 하는 기법이다. 대부분 Fast를 기반으로 하고 있는데, 채널 트랜지션은 기반으로 하는 주기적 방송 기법의 성능의 한계를 벗어날 수 없다. 또한 요청율이 작은 경우에는 채널 수를 줄이므로 한 세그먼트의 크기가 길어지게 되어 클라이언트의 초기 지연시간이 너무 길어질 수 있다. 채널 수의 변경으로 세그먼트 크기가 달라지면서 각 채널의 슬롯(slot)의 경계가 서로 일치하지 않을 수 있는데, 이를 해결하기 위해 불필요한 세그먼트가 들어가기도 한다. 또한 harmonic계열의 주기적 방송 기법처럼 채널의 대역폭이 다른 경우에는 적용하기가 어렵다.

클라이언트 요청율은 예측 가능한 것이 아니라 어느 시점에 급격히 늘어날 수도 있고, 다시 작아질 수도 있다. 지금까지 요청율의 변화에 따라 서버 대역폭을 조정하면서 서비스를 제공하는 기법에 대한 연구는 몇 개의 채널 트랜지션 기법들에 불과하며 위에서 언급한 한계점들을 갖는다. 따라서 본 논문에서는 다양한 클라이언트 요청율에 따른 융통성있는 스트리밍 서비스를 제공하기 위하여 Dynamic Heuristic Broadcasting(DHB)[10] 기법과 주기적 방송 기법을 혼합하여 사용하는 방법을 제시한다. 즉, 클라이언트 요청율이 작은 경우는 DHB로 서비스를 제공하다가 요청이 증가하여 미리 정해진 수 이상의 채널을 사용하게 되면 주기적 방송 기법으로 트랜지션한다. 본 논문에서 사용하는 주기적 방송 기법은 Striping Broadcasting (StB)를 기반으로 한다.

2절에 본 논문에서 기반으로 하는 DHB와 StB를 기술하고, 3절에서는 이들간의 변환 기법에 대해 기술한다. 4절에서는 시뮬레이션을 통해 본 논문에서 제안된 변환 기법의 특성과 성능이 기술되며 5절에서 결론을 맺는다.

## 2. 관련연구

### 2.1 Dynamic Heuristic Broadcasting(DHB)

DHB는 어떠한 주기적 방송 기법에도 기반하지 않는 휴리스틱 알고리즘을 적용한다. 비디오 오브젝트를  $n$ 개의 세그먼트로 나누고 클라이언트의 서비스 요청이 있

을 때마다 서버 채널에 스케줄링 되어 있지 않은 세그먼트만을 스케줄링하여 브로드캐스트 한다. 세그먼트는 클라이언트에서 재생되어야 할 슬롯 이전의 슬롯에 할당하며 서버 채널에 이 조건을 만족하는 빈 슬롯이 존재하지 않는 경우 새로운 서버 채널을 할당하고 세그먼트가 재생되어야 할 슬롯의 위치에 스케줄링한다. 그림 1은 비디오 오브젝트를 6개의 세그먼트로 나누었다고 가정할 때 세그먼트 스케줄의 예이다. (a)는 슬롯 1에 처음으로 서비스 요청이 온 경우이다. 요청을 받은 다음 슬롯부터 순서대로 세그먼트를 스케줄링한다. (b)는 슬롯 3에 두번째 서비스 요청이 온 경우이다. 슬롯 4부터 해당 클라이언트에 세그먼트가 브로드캐스팅 되는데, 이때 서버 채널에 존재하지 않는 S<sub>1</sub>, S<sub>2</sub>가 새로 스케줄링된다. S<sub>1</sub>은 슬롯4에, S<sub>2</sub>는 슬롯5에 재생되어야 하는데 채널 0에는 빈 슬롯이 없으므로 새로운 채널 1의 재생 슬롯의 위치에 스케줄링된다. 알고리즘 1에 DHB의 알고리즘을 기술하였다.

슬롯	1	2	3	4	5	6	7	8	9
채널 0	-	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	-	-
채널 1	-	-	-	-	-	-	-	-	-

(a)

슬롯	1	2	3	4	5	6	7	8	9
채널 0	-	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	-	-
채널 1	-	-	-	S <sub>1</sub>	S <sub>2</sub>	-	-	-	-

(b)

그림 1 DHB 세그먼트 스케줄의 예

**알고리즘 1. Dynamic Heuristic Broadcasting 프로토콜**

**가정 :**

- 슬롯 k에 이미 세그먼트 m<sub>k</sub> 가 할당되어 있음
- 비디오는 k의 세그먼트로 나뉘어 있음
- 슬롯 i에 새로운 서비스 요청이 발생함

**알고리즘 :**

```

for j:=1 to n do
  search slots i+1 to i+j for an already
  scheduled instance of Sj
  if not found then
    let mmin := min {mk | i+1≤k≤i+j}
    let kmax := max {kl | i+1≤k≤i+j
    and mk = mmin}
    schedule one instance of Sj in
    slot kmax
  end if
end for loop
    
```

**2.2 Striping Broadcast(StB)**

StB에서는 클라이언트에서 필요로 하는 저장공간을 줄이기 위해 큰 세그먼트를 동일한 크기의 세그먼트 두 개로 더 나누어주는데 이를 stripes라고 한다. i∈[N,K]

에 해당하는 세그먼트에 대해서만 두개로 나누는데 첫 번째 부분은 S<sub>11</sub>에 할당하고 두번째 부분은 S<sub>12</sub>에 할당한다. StB에서 사용하는 중요한 기호의 의미는 아래와 같다.

- K : 비디오 데이터에 할당된 채널의 수로서 각 채널의 대역폭은 동일하다. C<sub>1</sub>~C<sub>K</sub>의 채널이 한 비디오 데이터를 서비스하는데 필요하다.
- N (2≤N≤K) : 가장 큰 세그먼트의 크기를 조정하기 위해 사용되는데 이로 인해 서비스 지연시간이나 클라이언트에서 필요로 하는 저장공간의 크기가 조정된다.
- L<sub>i</sub> : 세그먼트 i의 크기로서 비디오 데이터를 재생하는데 걸리는 총 소요시간인 L로부터 계산된다.

$$L_i = 2^{i-1} \cdot L_1 \quad i \in [2, N-1],$$

$$2^{N-1} \cdot L_1 \quad i \in [N, K],$$

$$L_1 = L / [(K-N+2) \cdot 2^{N-1}-1] \quad (1)$$

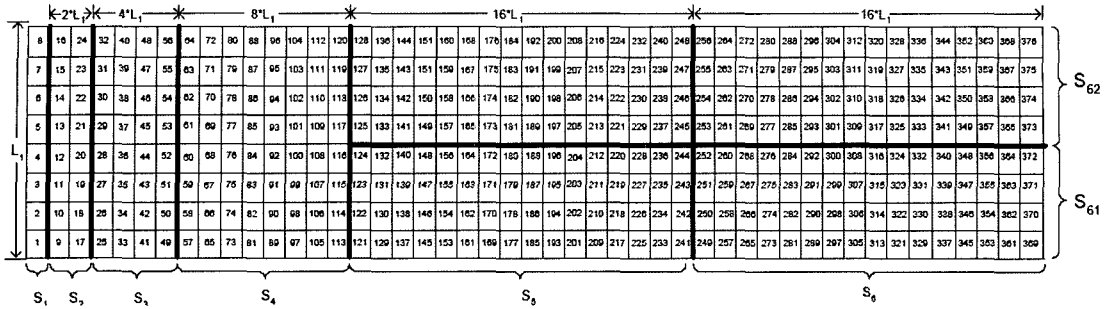
그림 2(a)는 N = 5, K = 6으로 하였을 때 세그먼트 할당에 대한 예이다. 각 블록은 블록 아이디로 표시되어 있으며 이 순서로 재생된다. 채널의 수는 6이므로 비디오 데이터는 S<sub>1</sub>~S<sub>6</sub>까지 6개의 수퍼블록으로 나뉘게 되고 각 수퍼블록은 (1)의 식에 따라, S<sub>1</sub>은 8개의 블록으로, S<sub>2</sub>는 16개, S<sub>3</sub>은 32개 S<sub>4</sub>는 64개로 구성된다. 채널 N부터 K까지는 세그먼트의 크기가 계속 커지는 것을 방지하기 위해 같은 크기의 세그먼트로 나눈다. 따라서 S<sub>5</sub>, S<sub>6</sub>는 128개 블록의 동일한 크기로 구성된다. 또한 N번째 채널부터는 쪼개져 S<sub>51</sub>, S<sub>52</sub>, S<sub>61</sub>, S<sub>62</sub>로 나누고 각각 C<sub>51</sub>, C<sub>52</sub>, C<sub>61</sub>, C<sub>62</sub>의 채널에 브로드캐스트 된다. 그림 2(b)는 이에 대한 브로드캐스트 스케줄을 나타낸다.

**3. 동적 스트리밍 서비스 스킵 트랜지션 기법**

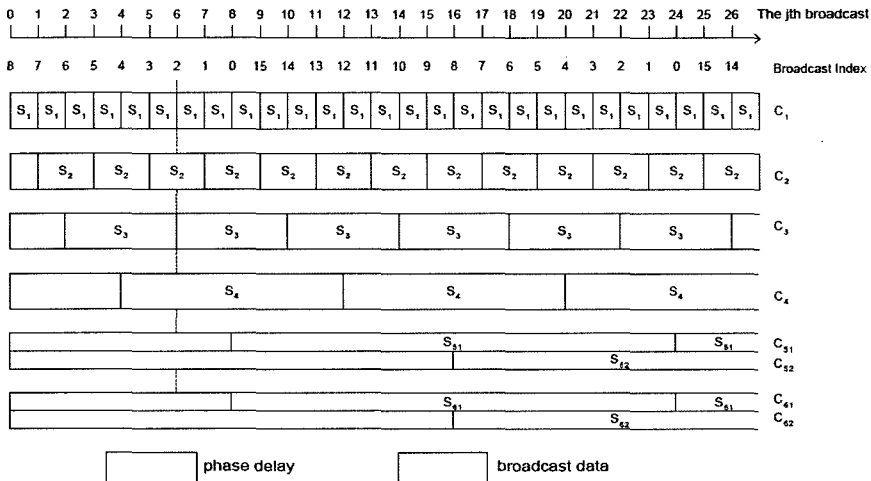
본 논문에서 제안하는 동적 스트리밍 서비스 스킵 트랜지션 기법은 클라이언트 요청율에 따라 유리한 기법으로 전환해가면서 스트리밍 서비스를 제공하자는 것이다. 처음에는 DHB로 스트리밍 서비스를 시작하는데 클라이언트의 요청이 작을 때는 DHB를 사용하여 스트리밍 서비스를 제공하고, 클라이언트 요청율 증가로 미리 정해진 수만큼의 서버 채널을 사용하게 되어 주기적 방송 기법을 사용하는 것이 더 유리해지면 StB로 전환한다. 또한 클라이언트 요청이 줄어들어 StB로 서비스하는 것이 서버 대역폭을 낭비하는 것이 되면 다시 DHB로 기법 전환을 한다. 이 절에서는 본 논문에서 적용하기 위해 수정된 DHB과 StB와 이 두 기법간 트랜지션 방법에 대해 기술한다.

**3.1 수정된Dynamic Heuristic Broadcasting(DHB)**

동적 스트리밍 서비스 스킵 트랜지션 기법에서는 각 비디오 데이터가 사용할 수 있는 최대 채널의 수를 미



(a) N=5, K=6에서의 세그먼트 할당



(b) 비디오 데이터에 대한 브로드캐스트 스케줄

그림 2 Striping Broadcast (StB)

리 정하고 그 채널 수에 맞게 비디오를 세그먼트로 나눈다. 이때 정하는 최대 채널의 수는 StB에서 사용할 채널의 수를 고려하여 결정한다. 이렇게 함으로써 StB를 사용한 경우와 같은 수의 세그먼트를 사용할 수 있게 되어 기법 전환시 슬롯의 경계가 불일치함으로써 발생할 수 있는 비효율을 막을 수 있다. 예를 들어, 한 비디오에서 사용할 수 있는 최대 채널 수를 4개로 정했다면, StB에서는 비디오를 15개의 세그먼트로 나누므로 DHB에서도 동일한 수의 세그먼트로 나눈다.

각 채널은 세그먼트 크기와 같은 크기의 슬롯으로 나누어져 있다고 보고, 각 슬롯에 세그먼트를 할당하여 전송한다. 기존의 DHB와 달리 본 논문에서 사용하는 DHB에서는 첫번째 채널에 1번 세그먼트만을 주기적으로 브로드캐스트 한다. 두 번째 세그먼트부터는 그 다음 채널의 빈 슬롯들 중 클라이언트에서 재생하게 될 슬롯을 기준으로 가장 최근의 빈 슬롯에 할당한다. 세그먼트의 스케줄은 DHB와 동일한 방식으로 이루어지게 되는데, 앞쪽 세그먼트는 채널에 더 자주 할당되므로 채널

내에서 빈 슬롯을 찾지 못하는 경우가 많이 생긴다. 이 경우 새로운 채널을 할당하여 사용하며 클라이언트 요청이 늘어나게 되면 사용하게 되는 채널 수도 늘어나게 된다.

그림 3은 비디오를 15개의 세그먼트로 나누어 서비스하는 경우, 슬롯 6까지 매 슬롯마다 클라이언트 요청이 발생했을 때 세그먼트 할당이 어떻게 이루어지는지에 대한 예를 보여준다. S<sub>1</sub>은 항상 채널 0으로 전송하고, S<sub>2</sub>부터는 채널 1이후에 할당하여 전송한다. 슬롯 0에 비디오 데이터에 대한 요청을 처음 받게 되면 세그먼트들은 슬롯 1부터 할당되는데, 채널 0의 슬롯 1에 S<sub>1</sub>을 할당하고 채널 1에 S<sub>2</sub>부터 S<sub>15</sub>까지 할당한다. 슬롯 1에 다시 서비스 요청이 발생하면 슬롯 2부터의 세그먼트들은 기존의 스케줄을 이용해서 서비스가 가능하다. 따라서 S<sub>2</sub>부터 S<sub>15</sub>까지의 세그먼트는 채널 1에 있는 것을 그대로 사용하고 슬롯 2 이후에 존재하지 않는 S<sub>1</sub>만 새로 채널 0의 슬롯 2에 할당해서 전송한다. 슬롯 2에 또다시 클라이언트 요청이 왔을 경우 S<sub>3</sub>부터 S<sub>15</sub>는 그대로 사

슬롯	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
채널 0		S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>									
채널 1			S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	
채널 2					S <sub>2</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>5</sub>		S <sub>6</sub>				

그림 3 DHB 세그먼트 할당

슬롯	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
채널 0	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>
채널 1		S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>
채널 2			S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>
채널 3					S <sub>8</sub> <sup>1</sup>	S <sub>9</sub> <sup>1</sup>	S <sub>10</sub> <sup>1</sup>	S <sub>11</sub> <sup>1</sup>	S <sub>12</sub> <sup>1</sup>	S <sub>13</sub> <sup>1</sup>	S <sub>14</sub> <sup>1</sup>	S <sub>15</sub> <sup>1</sup>	S <sub>8</sub> <sup>1</sup>	S <sub>9</sub> <sup>1</sup>	S <sub>10</sub> <sup>1</sup>	S <sub>11</sub> <sup>1</sup>
								S <sub>8</sub> <sup>2</sup>	S <sub>9</sub> <sup>2</sup>	S <sub>10</sub> <sup>2</sup>	S <sub>11</sub> <sup>2</sup>	S <sub>12</sub> <sup>2</sup>	S <sub>13</sub> <sup>2</sup>	S <sub>14</sub> <sup>2</sup>	S <sub>15</sub> <sup>2</sup>	

그림 4 수정된 StB 스케줄

용하고 없는 세그먼트인 S<sub>1</sub>, S<sub>2</sub>를 새로 할당한다. S<sub>2</sub>는 빈 슬롯이 없으므로 새로 채널 2를 할당하여 전송한다. 만일 세그먼트가 재생되어야 할 위치에 바로 할당될 수 없으면 그 이전의 빈 슬롯을 찾아 할당하고 빈 슬롯도 없는 경우 새로 채널을 할당하여 그 세그먼트가 재생되어야 할 위치에 할당한다.

3.2 수정된 Striping Broadcast (StB)

클라이언트 요청이 많으면 DHB가 StB보다 더 많은 채널을 필요로 하게 되므로 이 때에는 StB를 사용하여 브로드캐스트한다. 두 기법간의 전환을 쉽게 도식화하기 위해 그림 2 (b)의 StB 스케줄을 그림 4와 같이 바꾸었다. 세그먼트는 S<sub>i</sub>의 크기를 기준으로 표시하였으며 본문에서는 StB의 채널 split은 마지막 채널에서만 이루어진다고(N=K) 가정했다. S<sub>i</sub><sup>1</sup>은 S<sub>i</sub>의 첫번째 stripe을 의미하고 S<sub>i</sub><sup>2</sup>은 S<sub>i</sub>의 두번째 stripe을 의미한다.

3.3 스킴 트랜지션(Scheme Transition)

3.3.1 StB로의 스킴 트랜지션

그림 3의 예에서 슬롯 7에 클라이언트 요청이 오게 되면 슬롯 8부터 서비스가 시작되고, 현재 채널에 존재하지 않는 S<sub>1</sub>, S<sub>2</sub>, S<sub>7</sub>을 재할당하게 된다. 이때 S<sub>2</sub>는 슬롯 8이나 슬롯 9에 할당되어야 하는데, 현재 사용중인 채널에 빈 슬롯이 없으므로 할당할 수 없다. 따라서 S<sub>2</sub>를 위해 새로운 채널이 필요하다. 이로 인해 필요한 채널의 수는 미리 정해놓은 최대 사용 채널 수인 4개가 되며, StB를 사용하여 4개 채널에 주기적 방송하는 것에 비해 채널 사용상 이득이 없게 된다. 따라서 StB로의 스킴 트랜지션을 하게 된다.

DHB에서 StB로의 트랜지션은 적용시점에 따라 두가지로 나누었다.

• **Immediate Transition (IT)** : 서비스에 사용되는

채널의 수가 최대 사용 가능한 채널 수와 같아지는 시점부터 즉시 StB로 트랜지션. 현재 슬롯부터의 클라이언트 요청은 모두 StB의 세그먼트 스케줄에 따라 서비스받게 된다.

• **Delayed Transition (DT)** : 서비스에 사용되는 채널의 수가 최대 사용 가능한 채널 수와 같아지는 시점에 트랜지션을 준비하지만, DHB의 세그먼트가 모두 서비스된 이후 슬롯부터 StB의 세그먼트가 할당된다. 즉, 트랜지션을 결정해도 바로 StB 세그먼트 스케줄로만 서비스 받는 것이 아니며, StB와 DHB의 스케줄이 혼합되어 서비스된다.

IT를 사용하면 즉시 StB에 의해 서비스 받게 되므로 이후 슬롯부터는 클라이언트 요청들과 상관없이 서비스를 제공할 수 있다. 하지만, StB 스케줄에 따라 세그먼트를 할당하기 위해서는 이미 할당되어 있는 세그먼트의 일부를 다른 채널로 옮겨야 하는 경우가 발생할 수 있다. 이로 인해 이미 서비스를 받고 있는 클라이언트의 스케줄이 변경되므로 서버에서는 변경된 스케줄을 클라이언트에게 다시 알려주고 클라이언트는 이 변경된 스케줄에 따라 채널에서 세그먼트를 받아야 한다. 그림 5(a)는 그림 3에서 클라이언트가 슬롯 7에서 서비스를 요청한 경우 IT로 인한 스케줄의 변화를 보여준다. 흰색 슬롯은 이미 사용되고 있는 슬롯이고, 회색 슬롯은 빈 슬롯이다. 슬롯 7에서 클라이언트 요청을 받게 되면 S<sub>2</sub>를 할당하기 위한 새로운 채널이 필요하며 이로 인해 서버에서 사용하는 채널의 수가 StB로 서비스하는데 필요한 채널의 수와 같아진다. 따라서 스킴 트랜지션이 이루어진다. 첫번째 채널인 채널 0은 DHB에서와 같이 S<sub>1</sub>을 브로드캐스트하는데 사용된다. S<sub>2</sub>, S<sub>3</sub>은 빈번하게 반복되며 새로 할당이 되어야 하므로 비어있는 채널 3에

배치함으로써 기존 클라이언트에서 받을 영향을 최소화하도록 한다. 첫번째 채널과 마지막 채널은 이후 슬롯이 모두 비어 있으므로 StB의 스케줄에 따라 세그먼트가 할당되는데 문제가 없다. 하지만 채널 1과 채널 2는 StB 세그먼트가 할당되어야 하는 슬롯이 이미 사용되고 있으므로 DHB 세그먼트와의 조정이 필요하다. IT의 경우 즉시 StB로 서비스가 시작되어야 하므로 채널 2의 슬롯 10부터 S<sub>4</sub>~S<sub>7</sub>까지의 세그먼트가 주기적으로 할당된다. 이로 인해서 DHB 스케줄에 의해 할당되어 있던 슬롯 10의 S<sub>5</sub>와 슬롯 12의 S<sub>6</sub>이 충돌하게 된다. S<sub>6</sub>는 StB의 스케줄과 일치하므로 그대로 두고 S<sub>5</sub>는 현재 위치해 있는 슬롯 이전의 빈 슬롯을 찾아 채널 3의 슬롯 8에 재배치한다. 채널 1의 슬롯 12부터는 split되어 S<sub>8</sub>~S<sub>15</sub>가 할당되어야 하므로 S<sub>12</sub>~S<sub>15</sub>는 새로 옮겨갈 빈 슬롯이 필요하다. 하지만 현재 서비스되고 있는 채널 내에는 빈 슬롯이 없으므로 추가로 채널을 할당하여(트랜지션 채널) S<sub>12</sub>~S<sub>15</sub>를 재배치한다. 트랜지션 채널은 DHB에서 StB로의 트랜지션시 끊김없는 서비스를 제공하기 위해서 임시로 할당되어 사용되는 채널이다.

트랜지션 채널의 사용은 비록 일시적이긴 하나 StB만을 사용하여 서비스하는 것에 비해 채널을 하나 더 사용하게 되므로 부가적인 비용이 든다. 만일 이후에도 클라이언트의 서비스 요청이 계속 많다면 StB로의 IT가 현명한 결정이 될 것이다. 하지만, 클라이언트 요청은 예측 불가능한 것이므로 이후에 오는 요청이 적다면 StB로의 IT는 불필요한 것일 수도 있다. 따라서 DT에서는 이후 실제로 클라이언트 요청이 왔을 때 필요하면 트랜지션 채널을 사용한다. 이를 위해 현재 DHB 스케줄에 의한 서비스가 모두 완료되고 채널이 비는 슬롯

이후에 StB 세그먼트를 할당한다. 그림 5(b)는 DT의 예이다. S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>는 IT에서와 같은 방식으로 할당되며 S<sub>4</sub>~S<sub>7</sub>은 채널 2에서 DHB 스케줄에 의한 서비스가 끝난 이후 슬롯 14부터 반복된다. S<sub>8</sub>~S<sub>15</sub>는 채널 1에서 슬롯 16부터 할당된다. DT에서는 StB가 바로 적용되는 것이 아니므로 재생되는 시점까지 전송(on-time delivery)이 안되는 세그먼트가 생길 수 있는데 이러한 세그먼트는 DHB로 채널에 할당한다. 예를 들어 그림 5(b)에서 슬롯 8부터 시작되는 서비스는 S<sub>7</sub>을 제 시간에 전송할 수 없다. 따라서 S<sub>7</sub>은 DHB에 따라 스케줄링하여 채널 2의 슬롯 13에 할당한다. 즉, DT에서는 스킴 트랜지션 후 두 기법의 브로드캐스트 방식이 혼합되어 적용되다가 어느 시점 이상이 지나야 변환된 StB의 방식대로 브로드캐스트 된다. 즉, 슬롯 13부터가 StB 스케줄만으로 서비스가 가능해지므로 슬롯 7부터 11에 온클라이언트 요청에 대해서는 혼재된 방식이 적용된다. 만약 슬롯 8에 클라이언트 요청을 받았다면 슬롯 9부터 새로운 서비스가 시작되고 S<sub>4</sub>가 현재 채널에 존재하지 않으므로 새로 채널 2의 슬롯 11에 할당한다. 슬롯 9 이후에 받은 클라이언트 요청에 대해서는 채널에 있는 세그먼트만으로도 서비스가 가능하다. DT를 사용하면 IT에 비해 트랜지션 채널 사용으로 인한 비용을 줄일 수 있다.

IT는 클라이언트 요청이 지속적으로 많은 경우에 새로 발생하는 요청에 미리 대처하는 순행 트랜지션(proactive transition)이지만 기존에 서비스 받고 있는 클라이언트의 스케줄이 변경될 수 있으므로 이를 재전송해야 하는 오버헤드와 트랜지션 채널 사용으로 인한 부가적인 비용이 발생한다. 반면에 DT는 StB 세그먼트 스케줄링을

슬롯	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
채널 0	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>
채널 1	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>8</sub> <sup>1</sup>	S <sub>9</sub> <sup>1</sup>	S <sub>10</sub> <sup>1</sup>	S <sub>11</sub> <sup>1</sup>	S <sub>12</sub> <sup>1</sup>	S <sub>13</sub> <sup>1</sup>	S <sub>14</sub> <sup>1</sup>	S <sub>15</sub> <sup>1</sup>	S <sub>8</sub> <sup>1</sup>	S <sub>9</sub> <sup>1</sup>	S <sub>10</sub> <sup>1</sup>	S <sub>11</sub> <sup>1</sup>
						S <sub>8</sub> <sup>2</sup>	S <sub>9</sub> <sup>2</sup>	S <sub>10</sub> <sup>2</sup>	S <sub>11</sub> <sup>2</sup>	S <sub>8</sub> <sup>2</sup>	S <sub>9</sub> <sup>2</sup>	S <sub>10</sub> <sup>2</sup>	S <sub>11</sub> <sup>2</sup>	S <sub>12</sub> <sup>2</sup>	S <sub>13</sub> <sup>2</sup>	S <sub>14</sub> <sup>2</sup>	S <sub>15</sub> <sup>2</sup>
채널 2	S <sub>2</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>
채널 3		<b>S<sub>5</sub></b>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>
트랜지션채널						<b>S<sub>12</sub></b>	<b>S<sub>13</sub></b>	<b>S<sub>14</sub></b>	<b>S<sub>15</sub></b>								

(a) Immediate Transition (IT)

슬롯	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
채널 0	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub>
채널 1	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>8</sub> <sup>1</sup>	S <sub>9</sub> <sup>1</sup>	S <sub>10</sub> <sup>1</sup>	S <sub>11</sub> <sup>1</sup>	S <sub>12</sub> <sup>2</sup>	S <sub>13</sub> <sup>2</sup>	S <sub>14</sub> <sup>2</sup>	S <sub>15</sub> <sup>2</sup>
						S <sub>6</sub>	<b>S<sub>7</sub></b>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>
채널 2	S <sub>2</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>5</sub>		S <sub>6</sub>	<b>S<sub>7</sub></b>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>4</sub>	S <sub>5</sub>
채널 3			S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>2</sub>

(b) Delayed Transition (DT)

그림 5 DHB에서 StB로의 스킴 트랜지션 후 세그먼트 스케줄

미룸으로써 서비스 받는 클라이언트의 재스케줄링을 방지하고 트랜지션 채널의 사용을 줄인다. 하지만 이후에도 클라이언트 요청이 지속적으로 발생하는 경우에는 트랜지션 채널의 사용을 줄일 수 없으며 DHB와 StB의 스케줄이 혼합 사용되는 구간이 길어지게 된다. 채널 1에서 IT의 슬롯 12~15와 DT의 슬롯 16~19의 두번째 서브채널이 비는데 이것을 활용하기 위해 첫번째 서브채널에 할당되어 있는 세그먼트의 두번째 부분을 할당한다.

3.3.2 DHB로의 스킵 트랜지션

DHB로의 트랜지션은 StB로의 트랜지션보다 간단하다. StB 규칙에 의해 브로드캐스트 하다가 정해진 한계 슬롯 수( $T_{max}$ ) 이상 클라이언트 요청이 오지 않을 경우 해당 비디오에 대한 클라이언트의 요청이 줄어들었다고 보고 DHB로 스킵 트랜지션을 함으로써 서버 대역폭을 줄인다. 그림 6은 StB에서 DHB로의 트랜지션 후 세그먼트 할당을 보여주는데,  $T_{max}$ 개의 슬롯 동안 클라이언트 요청이 없어서 DHB로 트랜지션 하기로 결정하면 이전 서비스를 위해 사용되는 세그먼트는 그대로 전송한 후 채널을 릴리즈해야 한다. 채널 릴리즈를 어느 슬롯부터 할 수 있는지는  $T_{max}$  값에 따라 달라지는데 현재 슬롯 이후의 세그먼트를 서비스 받는 클라이언트가 없을 만큼  $T_{max}$ 를 충분히 크게 잡았다면 채널을 바로 릴리즈시킬 수 있을 것이고  $T_{max}$ 를 작게 잡았다면 이전의 요청을 서비스하기 위한 세그먼트를 모두 전송한 이후에 채널을 릴리즈해야 한다. 그림 6에서 회색으로 표시된 세그먼트는 슬롯 48에 온 클라이언트 요청을 서비스하기 위한 세그먼트를 나타낸다.  $T_{max}$ 가 1이라면 슬롯 50에서 DHB로의 트랜지션이 결정되는데 StB에 의한 세그먼트 스케줄링을 중지하고 이후 클라이언트 요청을 받았을 때 채널에 없는 세그먼트만 DHB로 스케줄링한다. 각 채널은 회색 슬롯 이후에 릴리즈 되는데, 만일 슬롯 50에서 다시 클라이언트 요청을 받은 경우 채널에 없는  $S_1, S_2, S_3, S_8$ 에 대해서만 세그먼트 할당이 이루어진다. 그림 6에서 굵게 표시된 부분이다. 반면에  $T_{max}$ 가 13이고 슬롯 61까지 새로운 서비스 요청이 없었다면

슬롯 62에서 DHB로의 트랜지션을 결정하고 모든 채널을 바로 릴리즈 할 수 있다.

4. 시뮬레이션

4.1 실험 환경

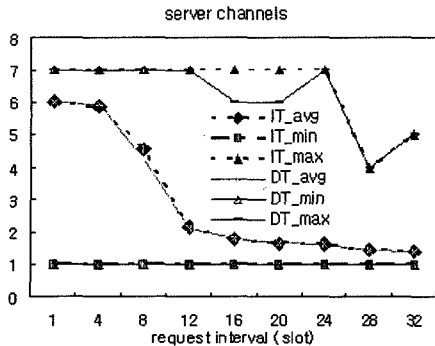
시뮬레이션에서 사용되는 비디오는 하나라고 가정하였고 이를 위해 서버에서 사용 가능한 최대 채널의 수는 StB에서 사용하는 채널의 수와 같다. 본 실험에서 StB에 사용되는 채널의 수는 6개로 하였다. 따라서 비디오는  $2^6 - 1 = 63$ 개의 세그먼트로 나뉘고 2시간 길이라고 가정하였을 때 한 세그먼트의 크기는 대략 1.9분에 해당한다. 이는 클라이언트의 최대 지연시간과 같다. 실험은 클라이언트의 서비스 요청 간격을 변화시켜가며 수행한 것과 실제 서비스를 제공하고 있는 업체에서 얻은 트레이스(trace)를 사용한 두 가지로 나누어 실험을 수행하였다. 서비스 요청 간격은 슬롯을 기준으로 하고 포와송 분포(Poisson distribution)을 따른다. 트레이스는 CDnetworks[11]에서 실제로 서비스를 제공하면서 발생한 클라이언트 요청에 대한 트레이스를 사용하였다. 클라이언트는 비디오 재생 도중에 서비스를 멈출 수도 있으나, 여기서는 끝까지 서비스를 받는다고 가정한다. StB로 서비스를 제공하다가 미리 정한 한계값 이상의 연속된 슬롯동안 클라이언트 요청이 없으면 다시 DHB로 트랜지션하게 되는데,  $T_{max}$ 는 30슬롯(비디오 재생 시간의 반 정도)으로 하였다. 시뮬레이션 시간은 10000 슬롯으로 하였다.

4.2 실험 결과

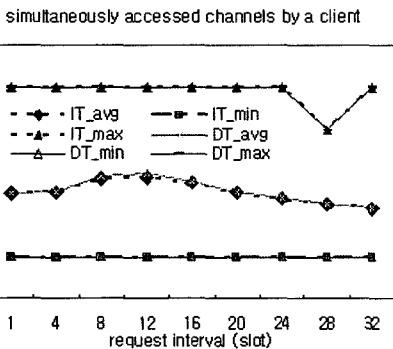
그림 7은 Immediate Transition(IT)과 Delayed Transition(DT)을 비교한 결과를 보여준다. 서비스 요청 간격은 1부터 32까지 변화시켰으며 서버에서 사용하는 채널의 수와 클라이언트가 서비스를 받을 때 동시에 접근해야 하는 최대 채널의 수, 클라이언트에서 미리 받은 비디오 데이터를 저장하기 위한 디스크의 양을 비디오 크기에 대한 %로 표현하였다. 점선이 IT를 사용한 것이고 실선은 DT를 사용한 것이며, 최대, 평균, 최소 값으로 나누어 표현하였다. 요청이 빈번할때는 DHB로 서

슬롯	49	50	51	52	53	54	55	56	57	58	59	60	61
채널 0	$S_1$		$S_1$										
채널 1	$S_2$	$S_3$		$S_2$	$S_3$					$S_8$			
채널 2	$S_5$	$S_6$	$S_7$	$S_4$	$S_5$	$S_6$	$S_7$						
채널 3	$S_{15}^1$ $S_{11}^2$	$S_8^1$ $S_{12}^2$	$S_9^1$ $S_{13}^2$	$S_{10}^1$ $S_{14}^2$	$S_{11}^1$ $S_{15}^2$	$S_{12}^1$ $S_8^2$	$S_{13}^1$ $S_9^2$	$S_{14}^1$ $S_{10}^2$	$S_{15}^1$ $S_{11}^2$				

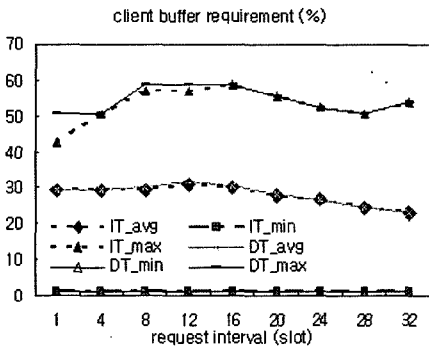
그림 6 StB에서 DHB로의 스킵 트랜지션 후 세그먼트 스케줄



(a) 서버 채널의 수



(b) 클라이언트의 동시 접근 채널 수



(c) 클라이언트의 저장공간 요구량

그림 7 Immediate Transition과 Delayed Transition의 성능 비교

비스를 시작하여도 곧 StB로 트랜지션한 후 계속 StB로 서비스된다. 반면에 요청이 계속 들어오는 간격 28 이후부터는 StB로의 트랜지션이 발생하지 않으며 시뮬레이션이 종료될때까지 DHB로만 서비스가 이루어진다. 양방향 스킴 트랜지션이 발생하는 구간은 4부터 24까지인데 가장 빈번하게 기법간의 트랜지션이 발생하는 구간은 요청 간격이 8일 때이다. IT과 DT는 대체적으로 비슷한 성능을 보여주고 있는데, 최대 서버 채널의 수와 클라이언트에서 필요로 하는 최대 저장공간의 양에 있

어서만 약간의 차이를 보여준다. 두 기법의 차이는 DHB에서 StB로의 트랜지션시 일시적으로 사용되는 트랜지션 채널을 얼마나 줄여줄 수 있는나에 있기 때문이다. IT의 경우 DHB에서 StB로의 트랜지션이 발생하는 요청 간격 24까지의 최대 서버 채널의 수가 7이다. 이는 기법 트랜지션을 하면서 트랜지션 채널을 사용하는 경우가 최소한 한번 이상 있다는 것을 의미한다. 반면에 DT는 요청이 많을 때는 IT와 그다지 차이가 나지 않지만 요청이 많지 않으면서 기법간의 트랜지션이 종종 발생하는 경우에는 서버 채널의 수를 줄여준다는 것을 볼 수 있다. 서비스 요청 간격이 4~12일 때는 최대 서버 채널의 수가 IT와 같지만 평균 서버 채널의 수는 미미하나 DT가 더 적고, 서비스 요청 간격 16~20일 때는 최대 서버 채널의 수에서 DT가 IT보다 적다는 것을 그래프에서 확인할 수 있다. 클라이언트에서 필요로 하는 저장공간의 양은 DT가 더 많은데 이는 StB로의 기법 트랜지션이 지연되면서 DHB의 적용구간이 IT에 비해 더 길어지기 때문이다. 평균값으로 봤을 때 클라이언트 요청이 적으면 DHB로 서비스되므로 StB를 사용하는 것에 비해 비용이 적으며 클라이언트 요청이 많아져도 StB의 성능을 따르므로 DHB의 단점이 가려진다.

트랜지션 채널은 서버 대역폭 사용에 있어서 추가 비용을 발생시키지만, 그림 7(a)의 평균 서버 채널 수를 보면 트랜지션 채널을 사용하는 경우는 적다는 것을 알 수 있다. 또한 최대 서버 채널 수에서 트랜지션 채널이 한 개만 사용됨을 알 수 있다. 실제로 StB에서 사용하는 채널의 수를 변화시키며 실험을 수행한 결과 한 개보다 많은 트랜지션 채널을 사용하는 경우는 발생하지 않았다.

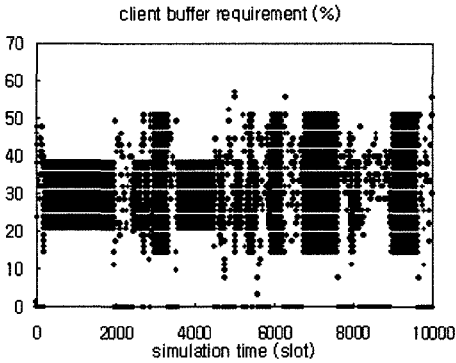
그림 8은 IT와 DT에 대해서 서비스 요청 간격이 8일 때 시뮬레이션 시간 동안 클라이언트 저장공간 요구량의 변화를 보여준다. IT는 DHB의 경우와 StB의 경우에 대한 구분이 비교적 명확하고 기법 트랜지션 구간(예를 들어, 3500, 6500, 7800 슬롯부근)이 짧게 나타나는 것을 볼 수 있다. 반면에, DT는 DHB에서 StB로의 트랜지션 구간이 더 길고 명확하지 않은 것을 볼 수 있다.

그림 9는 트레이스를 가지고 실험한 결과를 보여준다. 트레이스에 나타난 대부분의 비디오는 클라이언트 요청율이 작아서 DHB로만 서비스가 수행되고 StB로의 트랜지션은 발생하지 않았다. 기법 트랜지션이 발생한 비디오 중에서 가장 서비스 요청율이 높은 두 개 비디오에 대해 실험결과를 나타내었으며 이에 대한 정보는 표 1과 같다. 각 비디오 데이터는 200KB의 CBR로 서비스된다. 그림 9(a) sleeping은 시뮬레이션 시간 6000 슬롯부근에서 두번의 DHB에서 StB로의 트랜지션이 발생했으며 서비스 요청의 갑작스런 증가로 매번 트랜지션 채

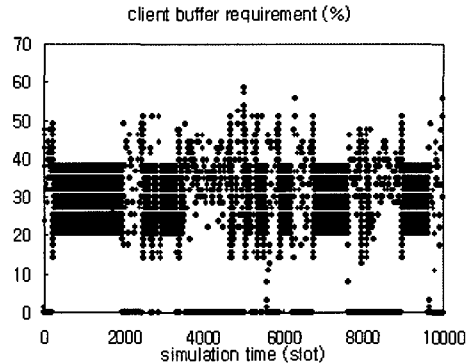


널을 사용하였음을 볼 수 있다. 클라이언트 저장공간 요구량의 경우 2000 슬롯 이전에 발생한 클라이언트 요청에 대해서는 DHB로만 서비스가 이루어졌기 때문에 높

게 나타났으며 6000 슬롯 부근에서는 StB로의 트랜지션이 이루어졌으므로 비디오 크기의 20~40% 사이가 진하게 나타났다. (b) possiblechanges의 경우 5800 슬롯

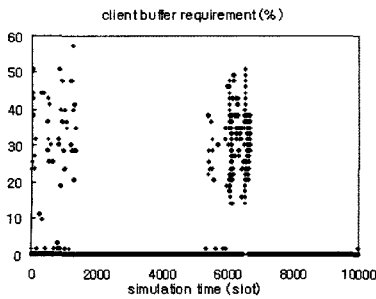
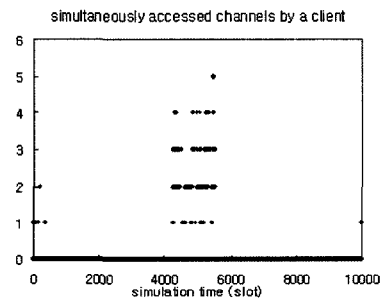
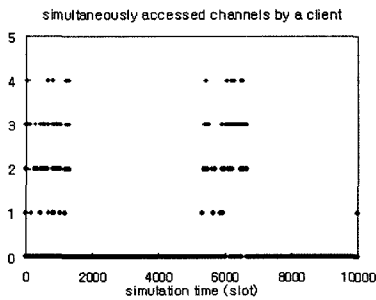
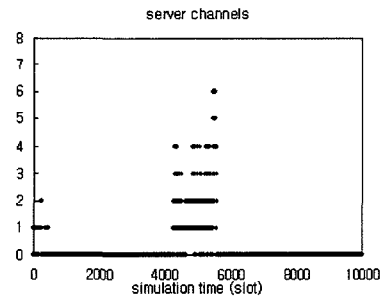
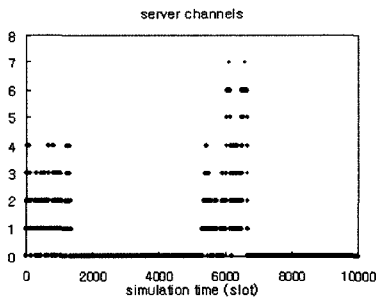


(a) Immediate Transition

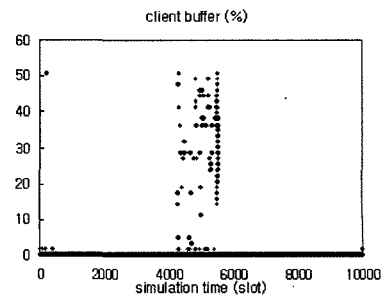


(b) Delayed Transition

그림 8 서비스 요청간격이 8일 때 두 기법의 클라이언트 저장공간 요구량 비교



(a) sleeping



(b) possiblechanges

그림 9 트레이스로 실험한 성능비교 (Delayed Transition)

부근에서 한번의 DHB에서 StB로의 트랜지션이 발생하였으며 요청의 수가 sleeping의 경우보다 작았기 때문에 트랜지션 채널까지는 사용하지 않았음을 볼 수 있다.

영화의 경우 소수의 비디오 오브젝트에 많은 클라이언트 요청이 발생하는 경향을 보이지만 교육 비디오 같은 경우 다수의 비디오 오브젝트에 고른 클라이언트 요청이 발생하는 경향을 보인다. 따라서 주기적 방송 기법만을 사용하여 서비스를 제공한다는 것은 서버 대역폭의 낭비라는 것이 분명하다. 하지만 주기적 방송 기법에서 서버 대역폭을 절약하기 위해 채널의 수를 작게 할 당하면 그만큼 서비스 받는 클라이언트의 초기 지연시간을 증가시키므로 비록 적은 수의 클라이언트일지라도 클라이언트 측면에서의 QoS는 떨어지게 된다. 이 경우 DHB기법으로 서비스를 제공하게 되면 서버 대역폭을 절약하면서 클라이언트의 초기 지연시간도 작게 할 수 있다. 다만 DHB로 서비스할 경우 클라이언트 저장공간 요구량은 커지게 되는데, 현재 디스크의 가격이 낮다는 점을 감안하면 이 비용은 네트워크 비용에 비해 수용가능하다.

표 1 트레이스에서 얻은 비디오에 대한 정보

비디오 파일	파일 크기 (KB)	재생 시간 (분)	서비스 요청 수	StB로의 트랜지션 수
sleeping	822,506	68.5	150	2
possible changes	858,790	71.6	77	1

5. 결론

본 논문에서는 클라이언트의 서비스 요청율이 낮을때는 DHB기법으로 서비스하고 높을때는 주기적 방송 기법으로 서비스함으로써 다양하게 변하는 요청율에 대해 동적으로 기법을 변화시켜가는 스킴 트랜지션을 제안하였다. 이같은 기법 변환을 통해 서버와 클라이언트 어느 한쪽의 QoS의 희생 없이 양 측면의 QoS를 어느 정도 수준으로 유지하면서 서비스를 할 수 있음을 시뮬레이션을 통해 보여주었다. 따라서 본 논문에서 제안한 스킴 트랜지션은 클라이언트가 적거나 많은 경우로 각각 초점을 맞추어 진행되었던 이전 연구와는 달리 실제로 다양하게 변할 수 있는 클라이언트 요청율에 대해 폭넓게 적용 가능하다는 것에 공헌이 있다.

참 고 문 헌

[1] Sheu, S., W. Tavanapong, and K. A. Hua, "A Video Broadcasting System," Tech. Rep. TR-03-03, Department of Computer Science, Iowa State

Univ., 2003.  
 [2] Li-Shen Juhn and Li-Ming Tseng, "Harmonic Broadcasting for Video-on-Demand service," in IEEE Transactions on Broadcasting, vol. 43, no. 3, 100-105, Sept., 1997.  
 [3] Li-Shen Juhn and Li-Ming Tseng, "Fast data Broadcasting and Receiving Scheme for Popular Video service," in IEEE Transactions on Broadcasting, vol. 44, no. 1, 100-105, March, 1998.  
 [4] Paris, J.-F., Carter, S.W., and Long, D.E., "A Hybrid Broadcasting Protocol for Video on Demand," in Proceedigs. Computing and networking Conference, San Jose, CA, 317-326, Jan, 1999.  
 [5] K. A. Hua and S. Sheu, "Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems," in Proc of SIGCOMM, 89-100, Sept., 1997.  
 [6] Z.-Y. Yang, "The Telepresentation System over Internet with Late-comers Support," Ph.D. dissertation, Nat'l Central Univ., Taiwan, R.O., C., 2000.  
 [7] Yu-Chee Tseng, Chi-Ming Hsieh, Ming-Hour Yang, Wen-Hwa Liao, and Jang-Ping Sheu, "Data Broadcasting and Seamless Channel Transition for Highly Demanded Videos," in IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 727-736, March, 2000.  
 [8] Yu-Chee Tseng, Yu-Chi Chuèh, and Sheu, J.-P., "Seamless channel Transition for the Staircase Video Broadcasting Scheme," in IEEE/ACM Transactions on networking, vol. 12, no. 3, 559-571, June, 2004.  
 [9] Wei-De Chien, Yuan-Shiang Yeh, and Jia-Shung Wang, "Practical channel Transition for Near-VOD services," in Proc of IEEE International Conference on Multimedia and Expo, 1843-1846, June, 2004.  
 [10] Carter, S.R., Paris, J.-F., Mohan, S., and Long, D.D.E., "A Dynamic Heuristic Broadcasting Protocol for Video-on-Demand," in Proc of International Conference on Distributed Computing Systems, 657-664, April, 2001.  
 [11] <http://www.cdnetworks.co.kr>



김 현 주

1996년 이화 여자 대학교 물리학과 졸업(학사). 1996년~1999년 쌍용 정보 통신 재직. 2002년 서울대학교 컴퓨터공학과 졸업(석사). 2002년~현재 서울대학교 컴퓨터공학과 박사과정. 관심분야는 멀티미디어, P2P, 분산시스템



엄 현 영

1984년 서울대학교 계산통계학과(학사). 1986년 Texas A&M 컴퓨터공학과(석사) 1992년 Texas A&M 컴퓨터공학과(박사). 1986년~1990년 Texas Transportation institute에서 시스템 분석가로 활동. 1992년~1993년 삼성 데이터 시스템 1993년~현재 서울대학교 교수로 재직. 관심분야는 분산시스템, 그리드 등