

## 주기적인 브로드캐스팅과 멀티캐스팅 기법을 이용한 멀티미디어 스트림의 제공 방법

최한기<sup>0</sup> 김진혁 박승규  
아주대학교 정보통신전문대학원  
{coffa<sup>0</sup>, gene, sparky}@madang.ajou.ac.kr

### The Policy of Providing Multimedia Streams via Periodic Broadcasting and Multicasting

Han-ki Choi<sup>0</sup> Jin-Hyuk Kim seung-Kyu Park  
Graduate school of Information and Communication, Ajou University

#### 요 약

인기있는 비디오 스트림을 서비스하는 VOD 시스템에서 다수의 사용자가 하나의 스트림을 공유할 수 있는 방법 중 하나로 주기적 브로드캐스팅 기법이 있다. 이 방법은 비디오 스트림을 주기적으로 브로드캐스팅해줌으로써 다수의 사용자에 대한 서버의 로드가 일정하게 유지되는 효과로 확장성(Scalability)을 갖지만 별도의 채널을 두지 않는 한 VCR 기능을 제공할 수 없고 서비스 응답 시간이 길게 된다. 본 논문에서는 기존의 방법에 On Demand 기능을 만족시키기 위해 프락시 기반 프리픽스 기술을 도입하고 VCR의 일부 기능을 구현할 수 있는 방법을 제안하였다.

#### 1. 서 론

VOD 서비스를 제공하는 방법에는 사용자가 서비스를 요청한 시점에서 서비스 제공자가 각 사용자에게 새로운 채널을 할당하는 것(True VOD)과 다수의 사용자가 하나의 스트림을 공유하거나 서비스 제공자가 비디오를 브로드캐스팅해주는 것(Near VOD)이 있다. 전자는 사용자의 요청을 바로 서비스 해 줄 수 있고 충분한 VCR 기능을 제공할 수 있는 반면에 각각의 사용자에게 채널을 할당함으로써 한정된 대역폭에서 많은 비디오 요청이 있는 경우 서비스가 제한될 수 있고 대역폭의 확장성이 낮으므로 채널에 대한 비용이 많이 들며 서버의 부하가 큰 단점이 있다. 후자는 브로드캐스팅 전송방법을 통해 다수의 사용자에 대해 같은 서버 로드를 가질 수 있어 확장성을 가지는 장점을 가지지만 별도의 채널을 두지 않는 한 VCR 기능을 제공할 수 없고 서비스 응답 시간이 긴 단점이 있다.

본 논문에서는 주기적 브로드캐스팅 기법과 프락시에서 멀티캐스팅 기법을 혼용하여 앞에서 언급한 두 가지 기법의 장점을 제공할 수 있는 방법에 대해 제안한다.

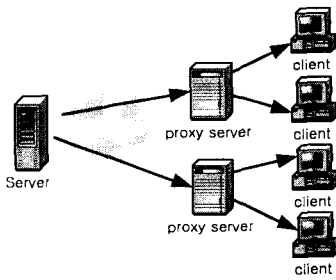
#### 2. 관련 연구

NVOD 시스템에는 스케줄된 멀티캐스팅과 주기적 브로드캐스팅의 두 가지 기법이 있다. 스케줄된 멀티캐스팅이란 일정 기간 동안 사용자들의 요청을 그룹화해서 하나의

멀티캐스트 채널을 통해 일괄적으로 서비스하는 것이다. 주기적 브로드캐스팅이란 비디오를 여러 세그먼트로 나누어 각 세그먼트 단위로 채널을 통해서 주기적으로 브로드캐스팅하는 기법이다. 따라서 사용자의 수에 관계없이 일정한 서버 부하를 유지하여 확장성을 가질 수 있다. 이러한 경우에는 세그먼트를 나누는 방법에 따라 네트워크 부하가 달라지게 된다. 논문[1, 2]에서는 세그먼트의 크기를 점점 증가시키는 피라미드 기법(pyramid-cased scheme)을 사용하여 비디오 세그먼트를 나누었다. 이 방법은 클라이언트는 비디오의 첫 부분부터 보기 시작한다고 가정하며 많은 비동기적인(asynchronous) 클라이언트의 요청에 대해 어느 시점에서 반응을 할 것인가를 결정해야 한다. 논문 [1]은 비디오의 첫 세그먼트의 주기가 시작하는 시점까지 기다린 후 클라이언트들의 요청이 동기화된 시점에서 서비스를 시작한다. 논문[3]은 프락시 서버를 두어 스트림의 첫 부분을 미리 저장해 둬으로써(prefix) 프락시 캐시의 이점과 동시에 비동기적인 클라이언트의 요청에 즉시 응답함으로써 시작 지연(startup delay)을 줄이는 방법을 제안하였다. 논문[4]에서는 또한 첫 번째 사용자가 캐시된 첫 세그먼트를 받는 동안 같은 세그먼트를 받아야 하는 두 번째 사용자에 대해 현재 진행중인 스트림과 이미 지나간 스트림을 동시에 받는 방법을 도입하였다. 이러한 방법을 패칭(patching) 기법[4]이라고 한다.

본 연구는 정보통신부지원 대학기초연구지원사업  
(정보통신기초기술연구지원사업)의 지원을 받아 진행됨

### 3. 시스템 구조



<그림 1> 시스템 모델

본 논문은 다음과 같은 구조를 가정한다.

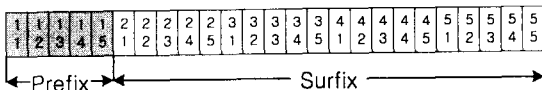
1. 서버는 매우 높은 용량을 가진 링크로 연결되어 있으며 Proxy 는 최종 사용자에게 가까이 위치해 있다.
2. Server-Proxy Proxy-Client 간에는 멀티캐스트를 지원하며 비디오는 CBR (Constant-Bit-Rate) 방식으로 1.15Mbps의 전송률을 가진다.
3. 클라이언트는 비디오 데이터의 첫 번째부터 요청하며 현재 재생 중인 세그먼트에서 다른 세그먼트로 Jumping 하는 기능을 요구한다.
4. 서버와 클라이언트는 하나의 비디오만을 고려하며 클라이언트는 비디오를 전송하거나 전송 받을 수 있는 충분한 공간과 대역폭을 가지고 있다.
5. 모델을 간단히 하기 위해 서버에서 클라이언트까지의 전송 지연은 없으며 스트림의 전송률과 스트림을 소비하는 재생률은 같다고 가정한다.

### 4. 제안된 기법

본 논문에서는 비디오의 크기를 V, 비디오 시간을 T, 채널의 수를 K, 세그먼트의 크기를 S로 나타내었다.

#### 4.1 비디오 데이터의 세그먼트 분할

연속적인 비디오를 일정한 크기의 세그먼트  $K^2$  개로 나눈다. 세그먼트 하나의 크기는  $S = V / (K * (K-1))$  이다. 그림 2는 K=5일 때 비디오를 세그먼트로 나눈 그림이다.



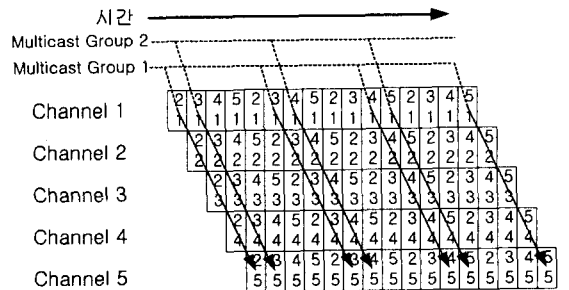
<그림 2> 세그먼트 분할 방법

프락시는 1~K개의 세그먼트를 캐시하고 있어야 하며 클라이언트의 요청이 들어오면 캐시된 세그먼트(prefix stream)를 전송해 준다.

#### 4.2 서버의 주기적 브로드캐스팅

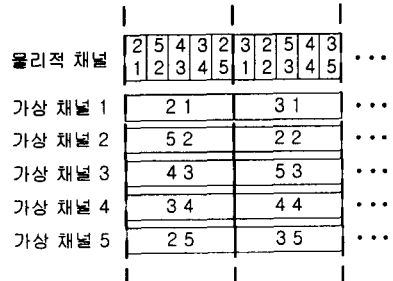
서버는 세그먼트를 <그림 3>과 같이 하나의 주기로 반복적으로 브로드캐스팅한다. 서버는 가상의 채널을 통해 K개의 세그먼트를 동시에 전송하며 서버로부터 프락시

까지 전송 받은 스트림은 멀티캐스트 그룹을 통해 순차적 세그먼트로 클라이언트로 전송된다.



<그림 3> 서버의 브로드캐스팅과 프락시의 멀티캐스팅

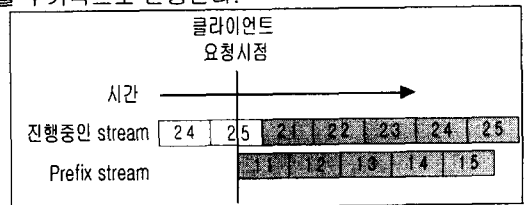
물리적 채널은 K개의 가상의 채널을 할한 필요 대역폭을 가져야 하는데 <그림 4>가 이것을 보여준다. 가상의 채널은 서버와 프락시 간의 물리적 채널을 통해 전송된다. 프락시는 물리적 채널을  $(T*S)/(V*K)$ 의 시간 단위로 나눠 가상의 채널로 변경한다.



<그림 4> 물리 채널과 가상 채널의 매핑

### 4.3 프락시와 클라이언트

프락시는 K개의 멀티캐스트 그룹을 가지며 클라이언트의 요구에 따라 각각의 Prefix 채널을 가진다. 하나의 멀티캐스트 그룹은 <그림 3>의 화살표의 방향 순서로 세그먼트를 주기적으로 전송한다.



<그림 5>서비스 요청 시 사용자가 받는 스트림

<그림 5>는 2개의 채널에서 클라이언트에게 전송되는 스트림을 설명하고 있다. 클라이언트는 첫 요청시 1~K까지의 세그먼트를 전송 받으면서 프리픽스 스트림을 수신 중 다음 세그먼트 (K+1 번째)가 시작되는 시점에서 동시에 스트림을 전송 받는다.

#### 4.3.1 클라이언트의 버퍼공간

클라이언트는 프리픽스 스트림을 재생하는 동안 멀티캐스트중인 스트림을 저장한다. 그러므로 프락시에 저장된 캐시공간과 같은  $K*S$ 의 공간을 가져야 한다.

### 4.3.2 VCR 기능

VCR 정품 기능은 현재 전송 중인 멀티캐스트 그룹을 변경함으로써 특정 비디오 위치로 이동할 수 있다. 정품 기능은 (K-1) 세그먼트 단위로 이동할 수 있도록 제한되어 있다. 예를 들면 그림2의 멀티캐스트 그룹1에서 (2,5) 세그먼트를 재생 중이라면 (2.1)로 Backward 하거나 (3,4), (4,3), (5,2) Forward 할 수 있다. 1~K 범위의 세그먼트로 Backward 할 경우 프락시 서버로부터 새 프리픽스 스트림을 전송 받는다.

## 5. 실험

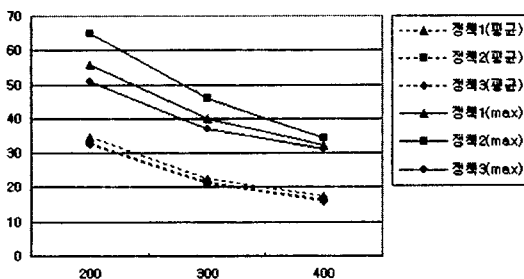
서버는 클라이언트의 요청에 관계없이 같은 채널을 가지므로 항상 일정한 로드를 가진다. 반면 프락시는 K개의 멀티캐스트 외에 클라이언트의 요청에 따라 프리픽스 스트림을 생성하게 되므로 로드와 항상 변한다. 서버와 프락시 간에 한정된 채널을 가진 경우 각 비디오의 채널을 얼마 만큼 할당하느냐에 따라 프리픽스의 크기가 달라지게 되므로 프락시는 다른 로드와 가지게 된다. 본 실험은 3가지 정책을 고려하여 프락시 로드와 효율적으로 분산시킨 정책을 찾는 것이다.

### 5.1 실험 방법

100개의 비디오에 대해 10000번의 요청이 일어났을 때 서버는 인기있는 20개의 비디오에 대한 요청만을 받아들인다. 비디오의 선택은 Zipf 분포를 따르며  $\theta = 0.271$  이다. 비디오의 요청 시정은 Poisson 분포를 따르며  $\lambda = 10$ 로 한다. 본 실험은 서버와 프락시 간에 할당된 전체 채널의 수를 200개, 300개, 400개로 증가시킴에 따른 프락시의 로드와 변화를 측정하였다. 프락시의 로드와 동시에 서비스하는 채널의 총 개수로 측정한다. 또한 동일한 요청에 대해 세가지 정책을 고려하여 평균, 할당된 채널의 최대값, 분산을 구하였다.

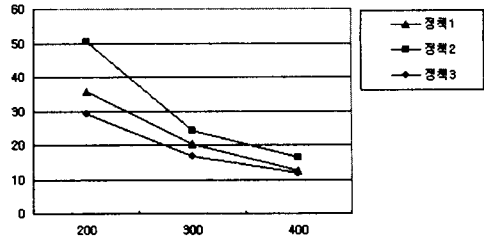
- 정책1 : 모든 비디오에 대해 같은 수의 채널을 할당
- 정책2 : 비디오의 인기도에 비례해 채널을 할당
- 정책3 : 비디오의 크기에 비례해 채널을 할당

### 5.2 성능 및 분석



<그림 6> 프락시 서버의 채널 평균과 채널의 최대값

세가지 정책 모두 프락시 로드와 평균값은 유사하였다. 그러나 할당된 채널의 최대값은 큰 차이가 났다. 이러한 차이는 프락시의 로드와 고르게 분산된 정도가 다르기 때문에 나타난다.



<그림 7> 프락시 채널의 분산

<그림 7>은 채널의 분산 값을 측정한 것이다. 정책2는 인기 높은 비디오 길이가 짧아지는 효율보다 인기 적은 비디오 길이가 길어지는 정도가 훨씬 크기 때문에 평균 채널의 수가 크게 좋아지지 않았으며 고르지 못한 스트림 길이 때문에 채널의 분산은 오히려 떨어졌다. 정책 3은 스트림의 길이를 모두 같게 만든 정책으로 프락시 로드와 가장 균일 하다. 본 실험에서는 프락시의 로드와 균일한 정책3이 가장 효율적임을 알 수 있었다.

## 6. 결론 및 향후 과제

본 논문에서는 기존의 연구에 On Demand 기능을 만족시키기 위해 프락시 기반 프리픽스 기술을 제안하였으며 본 논문에서 제안된 방법을 통해 VCR의 일부 기능을 구현할 수 있었다. 서버의 로드와 사용자의 요청에 관계없이 일정하였다. 또한 여러 개의 비디오에 대해 일정한 채널을 가지는 조건하에서 프락시의 로드와 줄이는 채널 할당 정책으로 전체 비디오 크기에 대해 비디오 크기의 비율만큼 채널을 할당하는 것이 가장 효율적임을 보였다.

차후에는 일반적인 VCR 기능(Forward, Backward, Pause, Jump Forward/Backward)을 구현할 수 있는 방법에 관한 연구가 수행되어야 할 것이다.

## 7. 참고 문헌

- [1] K. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," in Proc. ACM SIGCOMM, September 1997.
- [2] S. Viswanathan and T. Imielinski, "Pyramid broadcasting for video on demand service," in IEEE Multimedia Computing and Networking Conference, vol. 2417, pp. 66-77, 1995.
- [3] Y. Guo, S. Sen, D. Towsley "Prefix Caching assisted Periodic Broadcast: Framework and Techniques to Support Streaming for Popular Videos" in Proc. ACM Multimedia, 2001
- [4] K. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in Proc. ACM Multimedia, September 1998.
- [5] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," in Proc. IEEE INFOCOM, April 1999.