

# 무선 멀티미디어 스트리밍을 위한 협동적 캐싱 프록시 시스템

고덕주<sup>o</sup> 차호정  
연세대학교 컴퓨터과학과  
{djko,hjcha}@cs.yonsei.ac.kr

## A Cooperative Caching Proxy System for Multimedia Streaming in Wireless Networks

Dukju Ko<sup>o</sup> Hojung Cha  
Dept. of Computer Science, Yonsei University

### 요 약

본 논문은 기존에 제시되었던 협동적 캐싱 프록시 서버의 구조에 프록시 서버에 독립적인 prefix 배치와 프록시 서버간의 suffix 공유라는 동작을 추가하여, 무선 상에서의 효율적인 멀티미디어 콘텐츠 스트리밍이 가능하도록 하였다. 제안한 구조에 적합한 동작방법과 알고리즘 및 이에 따른 prefix의 특성을 제시하며, 콘텐츠 초기 지연 시간의 감소와 서버 저장 능력 향상에 대한 성능 개선 효과를 기술한다.

### 1. 서론

무선 통신 사용량의 폭발적인 증가와 함께, 무선 환경에서의 멀티미디어 콘텐츠 스트리밍에 대한 연구의 필요성은 점차 부각되고 있다. 클라이언트의 시작 지연시간을 감소시키고 전체 네트워크의 혼잡률을 낮추기 위해, 이미 기존의 많은 연구들이 유선에서의 멀티미디어 콘텐츠 스트리밍에 캐싱 프록시를 사용하는 방법에 대해서 이루어져 왔는데, 그 대표적인 예로는 prefix caching[1], segment-based caching[2], cooperative proxy caching[3] 등의 방법 등이 있다. Prefix caching은 클라이언트에서 메인 콘텐츠 서버에 스트리밍을 요청했을 경우에 발생하는 초기 지연시간을 줄이기 위해 클라이언트에 가까이 있는 프록시 서버에 콘텐츠의 앞부분만을 미리 캐싱해 놓는 방법이고, segment-based caching은 콘텐츠를 여러 개의 세그먼트로 분할한 후 각 세그먼트의 긴급 정도에 따라 차등을 두어 프록시 서버에 캐싱하는 방식이다. 또한 cooperative caching의 경우는 클라이언트의 스트리밍 요청에 대하여 프록시 서버에서 캐시 미스가 발생했을 경우에 메인 콘텐츠 서버에 스트리밍을 요청하기에 앞서 메인 콘텐츠 서버보다 좋은 대역폭을 보장할 수 있는 인접 프록시 서버에 스트리밍을 요청하는 방식을 사용한다. 하지만, 유선 환경에 비해 상대적으로 낮은 대역폭을 가지는 무선 환경의 특성이 존재하기 때문에 기존 유선 환경에서의 연구결과를 무선 환경에 그대로 적용하기에는 효율성이 떨어진다. 즉, 무선망과 같은 경우에는 그 대역폭이 매체의 특성상 가까운 거리에 있는 유선 프록시 서버들의 대역폭보다 작을 수밖에 없고 따라서, 프록시 서버끼리의 충분한 대역폭이 보장된다고 하더라도 프록시 서버에서 모바일 호스트까지의 대역폭이 제한되어 있기 때문에 프록시 서버에서의 상대적으로 큰 대역폭은 충분히 활용되지 못한다. 이러한 문제를 해결하기 위해서 무선망의 멀티미디어 콘텐츠 스트리밍에 cooperative caching의 방법을 적용시킬 수 있지만[4], 기존의 방법은 콘텐츠 재생시 초기 지연시간이 prefix caching보다 평균적으로 크며, 호출 빈도가 높은 콘텐츠가 많이 존재하는 프록시 서버로의 네트워크 트래픽이 집중될 수 있다는 문제를 가지고 있다.

본 논문에서는, 위에서 제시되었던 각각의 캐싱 프록시 메커

니즘을 효율적으로 혼합하여 무선에서의 멀티미디어 콘텐츠 스트리밍에 적합하게 개선된 협동적 캐싱 프록시 구조를 제안하고, 이 구조가 가지고 있는 특징과 이에 따른 성능향상에 대해 기술한다.

본 논문의 구성은 2장에서 전체적인 시스템의 구성과 제안한 협동적 캐싱 프록시 시스템의 구조를 살펴보고, 3장에서는 그 구조를 분석하며, 4장에서 결론을 맺는다.

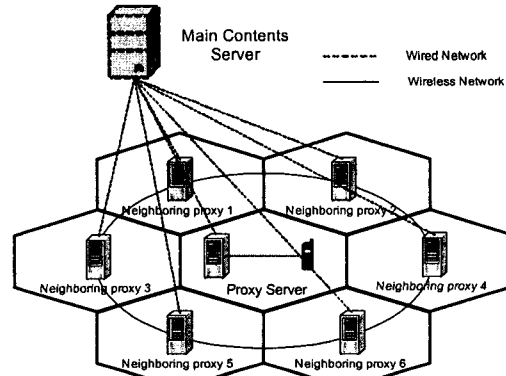


그림1. 시스템 구성도

### 2. 협동적 캐싱 프록시 시스템

다음은 본 논문에서 제안한 협동적 캐싱 프록시 시스템의 구조 및 동작과정을 기술한다.

#### 2.1 시스템 구조

본 논문에서 제안하는 시스템의 전체적인 기본 구조는, 그림 1과 같이 일반적인 셀 구조로 이루어진다. 각각의 셀에서는 베이스 스테이션과 하나씩 묶여 있는 프록시 서버가 존재한다. 이 프록시 서버는 서로 고속 유선망으로 연결되어 있으며, 모든 프록시 서버는 메인 콘텐츠 서버와도 고속 유선망으로 연결되어 있어 모바일 호스트의 스트리밍 요청에 대해 모든 프록시 서버에서 캐시 미스가 일어났을 경우 메인 콘텐츠 서버에서의 캐시 교체가 빠르게 일어날 수 있도록 한다. 본 논문에서 제안하는 협동적 캐싱 프록시는 기본적으로 두 가지 종류의 캐싱 세그먼트를 갖는다. 즉, 모바일 호스

• 본 연구는 한국과학재단에서 지원하는 특장기초연구사업으로 수행하였음 (과제번호 : R01-2002-000-00141-0)

트의 요청에 대하여 최소한의 지연시간으로써 스트리밍을 시작할 수 있게 해주는 prefix부분과 prefix 뒷부분의 멀티미디어 스트리밍을 계속 할 수 있게 해주는 suffix부분이 존재한다.

prefix부분들은 크기가 전체 콘텐츠의 크기에 비해 상대적으로 매우 작기 때문에 각 프록시 서버에 개별적으로 많은 수의 prefix를 두어 모바일 호스트의 스트리밍 요청에 대해서 최소한의 지연 시간을 가지고 스트리밍을 시작할 수 있게 보장한다.

suffix부분에서 콘텐츠의 초기 재생부분을 제외한 나머지 모든 부분이 재생되기 때문에 그 크기는 prefix부분에 대하여 상대적으로 크며, 따라서 동일한 suffix부분을 모든 프록시 서버마다 캐싱하는 것은 비효율적이라고 할 수 있다. 따라서 제안된 개선된 협동적 프록시 서버 캐싱의 방법에서는 이러한 suffix 세그먼트를 프록시 서버끼리 공유하게 된다.

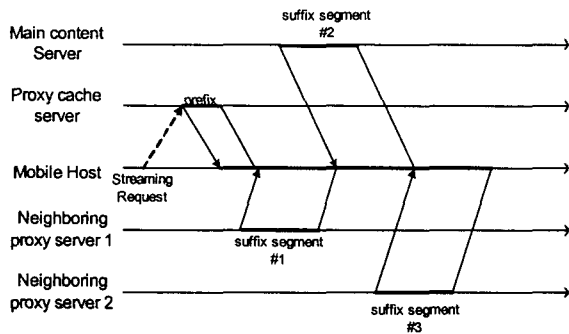


그림 2. 협동적 캐싱 프록시 시스템의 동작

그림 2는 모바일 호스트가 콘텐츠에 대한 스트리밍을 요청했을 경우에 자신이 속해 있는 프록시 서버와 인접한 프록시 서버들이 어떠한 방법으로 협력하는지 예를 보여준다. 모바일 호스트는 자신이 속해 있는 셀의 프록시 서버로부터 prefix부분을 스트리밍 받음으로써 초기 지연시간을 줄일 수 있고, 인접 프록시 서버 1과 2에 분산되어 저장되어 있는 suffix 세그먼트들을 스트리밍 받음으로써 각 프록시 서버의 저장 능력에 효율성을 갖는다.

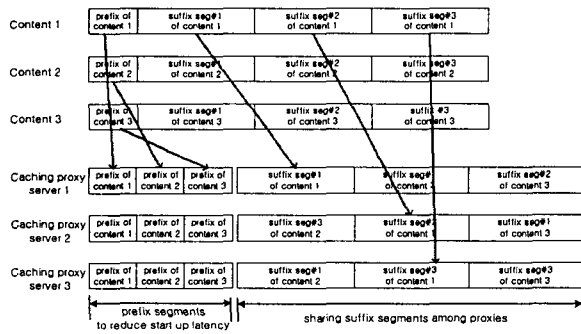


그림 3. 콘텐츠 세그멘테이션 및 세그먼트 분배

## 2.2 콘텐츠 세그멘테이션 및 세그먼트 분배

하나의 멀티미디어 콘텐츠를 프록시 서버들끼리 공유하기 위해서는 멀티미디어 콘텐츠가 적절한 크기와 갯수의 세그먼트로 나누어져 있는 것이 효율적이다. 하나의 멀티미디어 콘텐츠를 세그멘테이션하여 여러 프록시 서버에 나누어 저장하고 스트리밍함으로써, partial caching[5]의 알고리즘이 멀티미디어 콘텐츠의 세그먼트 단위를 기본 단위로 하여 효과적으로 일어 날 수 있도록 한다. 또한,

특정 멀티미디어 콘텐츠의 집중적인 호출이 있을 경우 발생할 수 있는 하나의 프록시 서버에 대한 과도한 부하를 분담시킬 수 있다. 그림 3은 여러 개의 콘텐츠가 각각 prefix와 suffix로 세그멘테이션된 후, 각 프록시 서버로 분배되는 예를 나타낸다. content 1,2,3의 모든 prefix세그먼트들은 프록시 서버 1, 2, 3에 모두 캐싱되며, suffix 세그먼트들은 프록시 서버 1, 2, 3 중의 하나에만 캐싱된다.

각 콘텐츠의 suffix부분은 세그멘테이션된 후 그 세그먼트들이 이웃에 존재하는 프록시 서버에 골고루 나뉘어져 분배되어야 하며, 본 논문에서는 세그먼트 분배 과정이 모바일 호스트의 세그먼트 요청을 통해서 이루어지도록 제안한다. 제안하는 구조에서는 콘텐츠의 prefix부분과 suffix부분이 각각 나뉘어서 관리되어야 하기 때문에 각각의 세그먼트 특성에 맞는 분배 정책이 수행되어야 한다.

**Prefix 세그먼트 분배** Prefix 세그먼트는 모든 프록시 서버에 개별적으로 분배되어야 한다. 따라서, 모바일 호스트의 요청이 있는 모든 prefix는 메인 콘텐츠 서버로부터 각각의 프록시 서버로 분배된다

인자	기술
proxy <sub>MH</sub>	모바일 호스트가 속해 있는 프록시
B <sub>other_proxy_prefix</sub>	요구하는 콘텐츠의 prefix가 존재하는 프록시 서버와 proxy <sub>MH</sub> 와의 스트리밍 대역폭
B <sub>other_proxy_suffix</sub>	요구하는 콘텐츠의 suffix가 존재하는 프록시 서버와 proxy <sub>MH</sub> 와의 스트리밍 대역폭
B <sub>MCS</sub>	메인 콘텐츠 서버와 proxy <sub>MH</sub> 의 스트리밍 대역폭

표 1. 세그먼트 분배 참조 인자

**Suffix 세그먼트 분배** Suffix 세그먼트는 각각의 프록시 서버마다 분배되지 않고 협동적인 프록시 서버 그룹에 하나씩만 존재한다. 따라서, 모바일 호스트에서 요청하는 suffix 세그먼트가 협동적인 프록시 서버의 그룹에 존재하지 않거나 존재하더라도 메인 콘텐츠 서버보다 낮은 대역폭으로 전송이 가능할 경우에 메인 콘텐츠 서버로부터 요청하는 모바일 호스트가 속해 있는 프록시 서버로 suffix 세그먼트가 분배된다.

```

if(prefix segment hits in proxyMH) {
    start streaming of prefix segment;
} else if (prefix segment misses in proxyMH) {
    if(Bother_proxy_prefix < BMCS) {
        start streaming of prefix from Main Contents server to mobile host;
        replace prefix segment in proxyMH;
    } else if(Bother_proxy_prefix > BMCS) {
        start streaming of prefix segment from neighboring proxy that has maximum Bother_proxy_prefix to MH;
        replace prefix segment in proxyMH;
    } while(mobile host requests STOP or suffix segment is ended) {
        if(BMCS < max(Bother_proxy_suffix)) {
            start streaming of suffix segment from neighboring proxy that has maximum Bother_proxy_suffix to MH;
        } else if(BMCS < max(Bother_proxy_suffix)) {
            start streaming of suffix segment from main content server to MH;
        } replace suffix segment in proxyMH;
    }
}
    
```

그림 4. 세그먼트 분배 알고리즘

그림 4는 표 1의 세그먼트 참조 인자를 사용해 세그먼트 분배 과정을 알고리즘으로 나타낸 것이다. 위와 같은 과정을 통해 prefix 세그먼트는 각각의 프록시 서버에 각각의 참조율에 따라 독립적

로 분배되게 되고, suffix 세그먼트는 프록시 서버간에 협력 할 수 있는 데이터 전송 대역폭의 범위에 따라 공유되어 프록시 서버들에 분산된다.

3. 시스템 분석

제안한 협동적 캐싱 프록시 서버는 기존의 협동적 프록시 서버와는 다르게 prefix부분이 각 프록시 서버마다 포함되어 있기 때문에 콘텐츠 재생의 초기 지연시간을 현저히 줄일 수 있는데, 협동적 캐싱 프록시 서버에 추가된 prefix의 크기와 개수에 따라 그 성능이 좌우되게 된다. 즉, prefix의 개수가 많을수록 캐쉬 미스가 줄어들어 콘텐츠 재생 초기 지연시간을 줄일 수 있고, 그 크기가 클수록 협동할 수 있는 프록시 서버의 수가 늘어날 수 있지만, 프록시 서버마다 개별적으로 저장되어 있는 prefix는 프록시 서버의 저장 효율에 오버헤드로 존재한다. 따라서 기존의 협동적 캐싱 프록시보다 효율적인 서버 저장능력을 보장하기 위해서는 제안한 협동적 캐싱 프록시 서버의 prefix가 다음과 같은 방법에 의해 개수가 결정되어야 한다. 표 2는 각 수식에서 사용되는 인자에 대한 기술이다.

인자	기술
$P_i$	제안한 협동적 캐싱 프록시 서버에서 협동할 수 있는 프록시 서버의 개수
$P_c$	기존의 협동적 캐싱 프록시 서버에서 협동할 수 있는 프록시 서버의 개수
$S$	각 프록시 서버의 저장 능력
$d$	프록시 서버간의 최대 지연 시간
$R$	스트리밍 콘텐츠의 재생 비트레이트
$n$	하나의 프록시 서버에 저장되는 각 prefix 세그먼트의 개수
$m$	기존의 협동적 프록시 서버에서 각 프록시 서버마다 평균적으로 갖는 콘텐츠의 갯수
$t$	모바일 호스트에서 모바일 호스트가 속해 있는 프록시 서버까지의 평균 지연 시간
$\alpha, \beta$	비례 상수

표 2. 수식 참조 인자

기존의 협동적 캐싱 프록시 서버보다 우수한 콘텐츠 재생 초기 지연 시간을 보장하기 위해서는 기존의 협동적 캐싱 프록시 시스템내에 존재하는 모든 콘텐츠들의 prefix를 가지고 있어야 한다.

$$m \times P_c < n \quad \dots \dots \dots \text{식(1)}$$

또한, 프록시 서버간의 최대 지연시간은 협동할 수 있는 프록시 서버의 수에 비례하기 때문에 다음과 같이 정의할 수 있다.

$$d = \beta \times P_i \quad \dots \dots \dots \text{식(2)}$$

기존의 협동적 캐싱 프록시는 prefix부분이 존재하지 않기 때문에 콘텐츠 재생의 초기 지연시간을 고려한다면 제안한 협동적 캐싱 프록시에 비해 작은 수의 협동적 프록시 서버를 가질 수 밖에 없다. 따라서, 다음과 같이 정의할 수 있다

$$P_i = \alpha \times (d \times R)^2 \times P_c \quad \dots \dots \dots \text{식(3)}$$

제안한 협동적 캐싱 프록시에서 프록시 서버마다 가지게 되는 prefix 세그먼트들의 크기는  $n \times d \times R$ 이 된다. 따라서, 제안한 방법이 더 효율적인 서버 저장 공간 성능을 가지기 위해서는 기존의 방법보다 프록시 서버끼리 공유할 수 있는 콘텐츠의 양이 더 많아야 하며, 따라서 다음과 같은 관계가 성립해야 한다.

$$S \times P_c < (S - n \times d \times R) \times P_i \quad \dots \dots \dots \text{식(4)}$$

식 (1), (2), (3) 와 (4)를 이용하면, 다음과 같은 결론을 얻을 수 있다.

$$\frac{m \times P_i}{\alpha \times (\beta \times P_i \times R)^2} < n < \frac{(\alpha \times \beta \times P_i \times R - 1) S}{\alpha \times (\beta \times P_i \times R)^2}$$

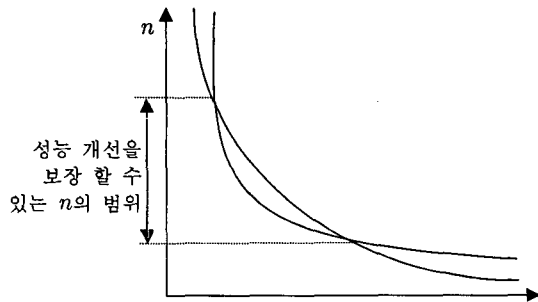


그림 3. 협동할 수 있는 프록시 서버의 수와  $P_i$  prefix 세그먼트 수와의 관계

따라서, 협동할 수 있는 프록시 서버의 개수  $P_i$ 와 콘텐츠의 재생 비트레이트  $R$ 에 따라 하나의 캐싱 프록시 서버에 저장되는 각 prefix 세그먼트의 개수  $n$ 이 결정될 수 있고, 제안한 협동적 캐싱 프록시 서버는 기존의 협동적 캐싱 프록시 서버보다 작은 초기 지연 시간과 서버 저장 능력에 효율성을 보장할 수 있다. 즉, 기존의 협동적 프록시 서버에서 프록시 서버 내에서의 캐쉬 히트에 평균적으로  $t + d/2$ 의 콘텐츠 재생 지연시간을  $t$ 로 감소시킬 수 있으며, 프록시 서버 공간 효율성을  $S \times P_c$ 에서  $\alpha \times (S - n \times d \times R) \times P_c$ 로 증가시킬 수 있다.

4. 결론

본 논문에서는 무선 상에서의 효율적인 멀티미디어 콘텐츠 스트리밍을 위한 협동적 캐싱 프록시 서버에 prefix와 suffix 세그먼트의 개념을 추가한 방법을 제시하였다. 제안한 구조에 따른 프록시 서버의 동작 과정과 알고리즘을 소개하였고 이를 바탕으로 하여 일반적인 협동적 캐싱 프록시 서버에 비해 어떠한 성능 개선이 일어나는지를 수식으로 표현해 보았다.

앞으로의 연구 과제로는 제안한 방법을 실제로 구현이나 시뮬레이션 등을 통해 실험을 함으로써 제안한 방법에 대한 검증은 하는 것이다.

참고문헌

[1] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams" *Proc. of INFOCOM*, New York, NY, March 1999.  
 [2] Kun-Lung Wu, Philip S. Yu, and Joel L. Wolf. "Segment-Based Proxy Caching of Multimedia Streams." *Proceedings of the Tenth International World Wide Web Conference*, Hong Kong, May 2001.  
 [3] Sandra G. Dykes and Kay A. Robbins, "Limitation and Benefits of Cooperative Proxy Caching", *IEEE Journal on Selected Areas in communications*, Vol.20, NO.7, September 2002  
 [4] Zhe Xiang, Zhun Zhong, Yuzhou Zhong, "A Cache Cooperation Management for Wireless Multimedia Streaming", *Info-tech and Info-net, Beijing. 2001 International Conferences on*, p328 -33 Volume: 5, 29 Oct.-1 Nov. 2001  
 [5] Shudong Jin, Azer Bestavros, Arun Iyengar, "Accelerating Internet Streaming Media Delivery using Network-Aware Partial Caching", *22nd International Conference on Distributed Computing Systems (ICDCS'02)*, Vienna, July 2002