

무선 환경에 적합한 사용자 이동성 기반 협동 프락시 구조

백창현⁰ 신승훈 박승규

아주대학교 정보통신전문대학원

{labmedia⁰, sihnhsh, skypark}@ajou.ac.kr

A mobility-aware collaborative proxy architecture applied to mobile environments for effective multimedia services

Changhyun Baek⁰, Seung Hoon Sohn, and Seungkyu Park

Graduate School of Information and Communication, Ajou University

요 약

최근 무선 통신 기술의 발전으로 인해 무선 사용자를 대상으로 하는 멀티미디어 서비스가 보편화 되고 있다. 그러나 무선 단말기의 제한된 능력으로 인해 유선 환경과 동일한 방식으로 무선 환경에서 서비스하는 경우, 이를 무선 단말기 상에서 유연하게 처리하기 어렵기 때문에 멀티미디어 데이터를 무선 단말기에 적합한 형태로 변환시킨 후 무선 사용자에게 전송해야 한다. 이에 따라 무선 사용자에게 효과적인 멀티미디어 서비스를 제공하기 위한 멀티미디어 오브젝트의 프레임 사이즈나 프레임 비율 등을 변환시키는 트랜스코딩 기법에 대한 연구가 현재 활발히 진행되고 있다. 본 논문에서는 트랜스코딩을 수행하는 프락시 간의 협동 구조를 바탕으로 사용자 이동성을 기반으로 한 멀티미디어 데이터 서비스의 구조적 효율성을 높이는 방식을 제안한다.

1. 서 론

무선통신과 컴퓨터 하드웨어 분야의 진보는 무선 사용자를 대상으로 하는 멀티미디어 서비스를 가능케 하였다. 그러나 무선 단말기에는 해상도나 처리 능력과 같은 제약이 여전히 존재한다. 이에 따라 서비스 제공자는 사용자의 단말기에 최적화된 데이터를 제공함으로써 사용자가 서비스를 원활하게 이용할 수 있도록 해야한다. 이와 같이 원본 데이터를 특정 디바이스에 맞게 변환하는 작업을 트랜스코딩이라고 하며 최근 이에 대한 연구가 활발히 진행되고 있다[1][3][4].

본 논문에서는 인접한 프락시 간의 로드밸런싱을 통해 시스템의 구조적 효율성을 높일 수 있도록 하며, 또한 사용자의 이동성을 반영하여 프락시에서 수행되어지는 트랜스코딩 작업의 효율성을 고려하도록 하였다. 각각의 프락시는 미디어 서버에서 보내온 데이터를 사용자의 단말기에 맞게 트랜스코딩한 후 사용자에게 전달한다. 그러나 트랜스코딩과 같은 처리 작업은 프락시 내의 CPU 자원을 많이 요구하기 때문에 특정 프락시로 트랜스코딩이 집중되는 경우에서 서비스 블로킹 상태를 초래하게 된다. 따라서 본 논문에서 제안하는 기법은 인접한 프락시가 서로 협동하여 트랜스코딩 작업을 처리함으로써 시스템 전체적인 효율을 높이는 동시에, 트랜스코딩 시 멀티미디어 오브젝트(MPEG)의 프레임 타입을 고려하여 트랜스코딩 작업의 성능을 향상시키는 것이 목적이다.

또한 사용자의 이동성을 반영하기 위한 방법으로 사용자의 과거 이동 패턴을 기반으로 예상되는 이동패턴을 알아낸 후, 사용자가 특정 베이스 스테이션(Base Station, BS)에 등록할 때 마다 이동패턴을 바탕으로 사용자가 이동할 경로를 예측하여 프락시 협동 구조에 이를 반영하였다. 이를 통해 프락시의 캐시 히트율(Hit Ratio)을 높이며, 시스템 전체적인 네트워크 비용을 절감할 수 있도록 한다.

본 논문의 구성은 2장에서는 관련연구를 살펴보고, 3장에서는 협동 프락시 구조에 대한 설명을, 그리고 4장에서는 이동성 반영을 위한 사용자 로그 히스토리 구성을, 5장에서는 트랜스코딩 프락시 선정 알고리즘에 대해 설명하였다. 끝으로 6장에서는 각각 본 논문에서 제시하는 구조의 효율성을 입증하기 위한 시뮬레이션 결과를 기술하였다.

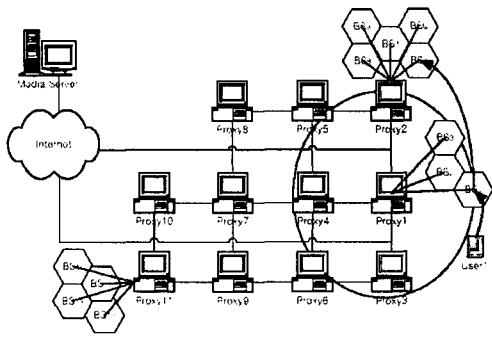
2. 관련 연구

Inter-frame 기법은 높은 압축률을 얻기 위해 MPEG와 같은 비디오 인코딩 알고리즘에 의해 보편적으로 사용된다. 이 기법은 인접 그림 간의 정보의 차이가 하나의 그림에서 포함하는 데이터의 양보다 훨씬 적다는 원리를 이용한다. 이 기법을 통해서, 비디오 스트림은 I/B/P 세 가지 프레임으로 인코딩되며, 이들 각각은 사이즈와 트랜스코딩에 요구되는 CPU 자원(I:B:P=1:3:3.5)이 서로 다른 특성을 가지게 된다. 이에 따라 Xueyan 등은 프레임 타입을 기반으로 캐싱 비디오 오브젝트를 위한 파티셔닝 기법을 제안하였다[2].

또한 무선 환경의 VOD 서비스를 위해 프락시 간에 트랜스코딩을 위한 협동 구조가 활발히 연구되고 있다[1]. 그러나 프락시 내의 캐시 사이즈와 비교하여 멀티미디어 오브젝트의 사이즈가 상대적으로 크기 때문에 일반적인 웹 캐싱에서의 히트율에 못미치는 결과를 내기 쉽다. 또한 저조한 히트율은 프락시 내에서 트랜스코딩의 가능성을 높이기 때문에 높은 CPU 리소스를 요구하게 된다.

3. 시스템 구조

본 논문에서 제안하는 멀티미디어 시스템 구조는 [그림 1]과 같이 미디어 서버와 프락시, 그리고 무선 사용자로 구성된다.



[그림1] 협동 프락시 구조

미디어 서버는 멀티미디어 데이터의 원본을 가지고 있으며, 하나의 MPEG 멀티미디어 오브젝트는 I/B/P 색션으로 나누어 구성된다.

프락시는 본 시스템에서 핵심 요소이다. 프락시는 미디어 서버에서 받은 멀티미디어 오브젝트를 사용자의 단말기에 맞는 데이터로 트랜스코딩을 수행한다. 또한 자신이 다른 작업으로 인해 트랜스코딩을 수행할 수 없는 상태가 되었을 때 인접 프락시 중 자신을 대신할 후보 프락시를 찾아 협력을 요청한다.

무선 사용자는 데이터 요청시 자신의 단말기 상태를 멀티미디어 오브젝트 요청 메시지에 포함시킨다. 즉 프락시에서는 사용자에서 보낸 요청 메시지 내부의 단말기 특성을 확인한 후에 트랜스코딩의 형태를 결정한다.

4. 사용자 로그 히스토리 구성

무선 사용자의 이동성을 프락시 캐싱 시스템에 반영하는 것은 프락시 내부의 캐시 히트율에 큰 영향을 미친다. 즉 사용자의 이동을 미리 예측하여 사용자가 이동할 곳에 미리 데이터를 저장하도록 하여 캐시 히트율을 높이며, 또한 인접 프락시에게 트랜스코딩 작업을 요청할 경우, 사용자 이동 후 유효 프락시가 되도록 협력 요청을 위한 인접 후보 프락시를 선정할 때 사용자의 이동패턴을 반영하도록 한다.

사용자 ID	현재위치	이동한위치	빈도수
1	A	B	100
1	A	E	50
2	C	E	25

[표1] 사용자 로그 히스토리

각 사용자는 BS에 등록 될 때마다 사용자 로그 히스토리에 자신의 위치 정보를 저장한다. 사용자 로그 히스토리는 [표1]과 같이 구성되며, 프락시는 협력 작업을 위한 인접 후보 프락시를 선정할 때 이 DB 정보를 이용한다.

무선 사용자가 프락시에 멀티미디어 오브젝트를 요청할 때, 프락시는 이 사용자의 로그 히스토리 정보를 검색한 후 빈도에 따라 인접 프락시에게 가중치를 적용한다. 따

라서 사용자 로그 히스토리를 이용하여 특정 사용자가 이동 중에 자신이 직접 연결되는 프락시가 트랜스코딩 작업을 수행하는 확률을 높이고자 한다.

5. 트랜스코딩 프락시 선정 알고리즘

트랜스코딩 프락시 선정 알고리즘(Transcoding Proxy Choosing, TPC)은 하나의 프락시에서 이미 다른 사용자의 요청을 처리하기 위해 CPU 자원을 모두 사용했을 때 인접 프락시에게 도움을 요청하기 위해 사용하게 된다. 제안된 알고리즘은 [표2]와 같은 환경변수를 이용한다.

변수	정의
CS_i	무선 호스트 i 의 현재 단말기 특성
LP_i	프락시 i 의 위치 우선권
CP_i	프락시 i 의 CPU 능력 우선권
AP_i	프락시 i 의 인접 프락시 리스트
APP_i	프락시 i 의 인접 프락시의 우선권 리스트
PAP_i	각 프레임에 유효한 프락시 조합 리스트
N_i	AP_i 의 수
w	CP_i 와 LP_i 사이의 가중치

[표2] TPC 알고리즘의 환경변수

5.1 유효한 인접 프락시 검색

본 단계에서는 사용자 요청에 대해 캐시 히트되지 않을 때, 요청을 받은 프락시가 사용자 단말 상태에 맞게 트랜스코딩을 수행한다. 그러나 요청을 받은 프락시가 사용자 요청을 처리할 만한 CPU 리소스를 가지고 있지 않으면 트랜스코딩을 수행할 수 있는 인접한 프락시 리스트를 찾는다. 즉 본 단계에서 트랜스코딩을 대신 수행할 수 있는 인접 프락시의 리스트를 구성하게 된다.

무선 사용자는 {사용자 ID, 비디오 ID, 세그먼트 ID, CS_i }와 같은 형태로 단말기의 특성(CS_i)을 포함하는 서비스 요청 메시지를 프락시에 보낸다.

```

if(check_cache_hit(mh_req) == CACHE_HIT)
    /* 캐시 히트일 경우 바로 서비스 */
else
    if(proxyi != BUSY)
        req_to_server(mh_req, proxyi);
    else
        /* 인접의 유효한 프락시 검색 */
        evaluate_priority(APi[j]); // step2
    
```

[그림2] 유효 인접 프락시 검색

5.2 후보 프락시 결정

이 단계에서는 인접 프락시 상태 검색에서 얻어진 유효 프락시 리스트를 기반으로 최적의 프락시를 선정한다. 이때 사용자 로그 히스토리 정보와 인접 프락시의 남은

자원을 살펴보고 후보 프락시를 결정한다.

즉 환경변수 CP 와 LP 값에 따라 인접 프락시를 평가한 후 최대 값을 가지는 인접 프락시가 트랜스코딩을 위한 후보가 된다.

```

evaluate_priority{
    for j = 1 to Ni step 1
        APPi[j] = (1-W)*CPi(API[j]) + W*LPi(API[j]);
    endfor
}
find_candidate{
    candidate = Max(APPi);
}
    
```

[그림3] 후보 프락시 찾기

5.3 파티셔닝 트랜스코딩

위의 후보 프락시 결정 단계에서 후보 프락시를 찾지 못할 경우 파티셔닝 트랜스코딩을 위한 프락시를 다시 찾게 된다. 이때 하나의 멀티미디어 오브젝트를 I/B/P 섹션으로 나누어 트랜스코딩하게 된다. 트랜스코딩에 요구되는 CPU 자원이 각 프레임에 따라 1:3:3.5의 비율을 가지므로 [2], 사용자의 요청을 받은 프락시를 포함한 인접 프락시 리스트에 대해 I/B/P 프레임 중 하나 이상을 만족하는 프락시를 찾게 된다. 세가지 타입 모두에 대해 트랜스코딩이 가능한 프락시 조합이 될 경우 서비스 블로킹하지 않고 트랜스코딩 후에 서비스를 한다.

5.4 선택적인 트랜스코딩 프락시 재구성

인접 프락시 간에 캐시와 트랜스코딩 부분을 공유하게 되므로, 시간이 흐름에 따라 비효율적인 프락시에서 트랜스코딩을 진행할 수 있다. 예를들어 프락시 A가 무선 사용자 U1을 위해 비디오 V1에 대한 트랜스코딩을 하고 있을 때, 인접한 프락시 B 쪽에서 10명의 사용자가 동일한 비디오를 A를 통해서 받는 경우 전체적인 네트워크 비용을 감안하여 트랜스코딩을 담당하는 프락시를 B로 교체하는 과정이다. 이때에는 2단계에서 프락시를 평가하는 과정을 다시 거치게 된다.

6. 시뮬레이션

본 논문에서 제안하는 서비스 구조의 성능을 측정하기 위해서 [표4]를 기반으로 800개의 사용자 요청을 시뮬레이션 하였다. 각 사용자는 [표3]과 같은 확률로 생성되었다. 또한 프락시에서 트랜스코딩에 요구되는 CPU 리소스는 사용자 클래스에 7.5를 곱한 값으로 설정하였다. 사용자 클래스 0은 원본 멀티미디어 오브젝트를 요청하는 사용자이다.

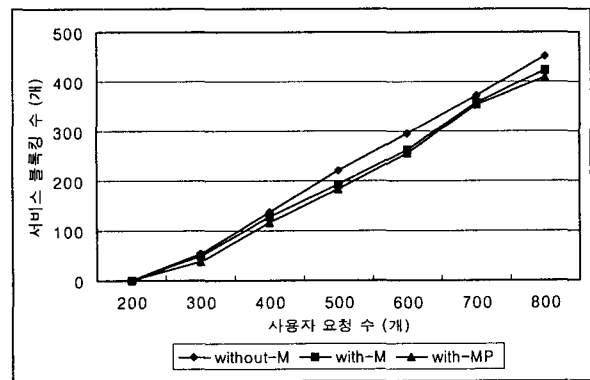
	0	1	2	3	4	5
접근 확률	0.15	0.2	0.2	0.2	0.15	0.1
CPU 리소스	0	7.5	15	22.5	30	37.5

[표3] 사용자 클래스

변수	설정 값
미디어 서버의 멀티미디어 오브젝트 개수	100
총 프락시 개수	16
비디오 요청에 대한 Zipf-분포 매개변수 θ	0.729
사용자 간의 평균 요청 시간 간격(Poisson 변수)	20초
무선 사용자의 평균 핸드오프 기회	5
핸드오프시 인접 프락시가 담당하는 BS로 이동성	0.2

[표4] 실험에서 이용한 변수

[그림 5]는 사용자 이동성을 반영하지 않은 협동구조 (without-M)와 이동성을 반영한 구조(with-M)와 이동성과 파티셔닝기법 모두를 도입한 구조(with-MP)에 대한 성능비교를 나타낸다.



[그림5] 실험 결과

실험결과에서 with-MP가 가장 적은 서비스 블로킹 수를 보임을 알 수 있으며, with-M과 with-MP의 차이는 하나의 프락시에서 단독으로 트랜스코딩을 처리할 수 없는 상태가 되었을 경우에 두 구조의 성능 차를 제시하고 있다.

7. 참고 문헌

[1] Chang Hyun Baek, Hyun Kyung Kim, Seung Hoon Sihn, and Seung Kyu Park, "A cooperative proxy architecture applied to mobile environments for effective VOD service", ITC-CSCC2003, JULY 2003.

[2] Xueyan Tang, Fan Zhang, and Samuel T. Chanson, "Streaming Media Caching Algorithms for Transcoding Proxies", ICPP'02, 2002.

[3] Bruce Zenel, "A general purpose proxy filtering mechanism applied to the mobile environment", Wireless Networks 5, 391-409, 1999.

[4] Kevin Curran and Gerard Parr, "A framework for the transmission of streaming media to mobile devices", International Journal of Network Management, 12:41-59, 2002.