

무선 모바일 프록시 시스템에서 유사도 기반의 캐싱 손실 최소화

Similarity-based Caching Replacement Loss Minimization in Wireless Mobile Proxy Systems

이종득*

Chong-Deuk Lee*

요 약

무선 모바일 프록시 캐싱 구조에서 캐싱 교체로 인한 손실은 스트리밍 QoS에 중요한 영향을 미친다. 본 논문에서는 캐싱 교체 과정에서 발생하는 손실을 최소화하기 위하여 유사도 기반의 캐싱 손실 최소화 기법 SCLM (Similarity-based Caching Loss Minimization)을 제안한다. 제안된 기법은 객체 세그먼트를 분할 한 후 유사도 관계를 수행한다. 유사도 관계가 수행된 세그먼트들은 SRT (Similarity Relation Tree)가 생성되고 유사도가 측정된다. 유사도는 적합성 피드백을 결정하는 중요한 척도로서 적합성을 만족한 세그먼트들은 캐싱 교체를 위해 캐시 블록에 저장한다. 시뮬레이션 결과 제안된 기법은 prefix 캐싱 기법, segment 캐싱 기법, 그리고 bps 캐싱 기법에 비해서 캐싱 시작 지연 제어율, 캐시 처리율, 그리고 캐시 응답율의 QoS가 효율적임을 보인다.

Abstract

The loss due to caching replacement in the wireless mobile proxy caching structure has a significant effect on streaming QoS. This paper proposes a similarity-based caching loss minimization (SCLM) for minimizing the loss caused by the caching replacement. The proposed scheme divides object segments, and then it performs the similarity relation about them. Segments that perform the similarity relation generates similarity relation tree (SRT). The similarity is an important metric for deciding a relevance feedback, and segments that satisfy these requirements in the cache block for caching replacement. Simulation results show that the proposed scheme has better performance than the existing prefix caching scheme, segment-based caching scheme, and bi-directional proxy scheme in terms of QoS, average delayed startup ratio, cache throughput, and cache response ratio.

Key words : wireless mobile, streaming QoS, SCLM, segments

I. 서 론

무선 모바일 환경에서 스트리밍 손실은 이동성,

대역폭 제약, 채널 오류, 그리고 프록시 구조에서의 스트리밍 객체의 캐싱 불일치로 인해 발생된다. 이러한 캐싱 불일치에 따른 손실을 최소화하기 위해 여러

* 전북대학교 전자공학부(Division of Electronic Engineering, Chonbuk national University)

· 제1저자 (First Author) : 이종득

· 투고일자 : 2012년 4월 26일

· 심사(수정)일자 : 2012년 4월 28일 (수정일자 : 2012년 6월 23일)

· 게재일자 : 2012년 6월 30일

프록시 캐싱 기법들이 제안되어 왔으나 전통적인 프록시 캐싱 기법들은 비-스트리밍에 기반을 두고 있다. 이로 인하여 캐싱 손실이 크게 발생하고 있으며, QoS 성능 또한 떨어지고 있다[1, 2, 3].

이러한 문제를 해결하기 위하여 프록시에서 미디어 객체의 일부를 캐싱하기 위한 프리픽스(prefix) 캐싱 기법과 세그먼트 기반 캐싱 기법들이 제안되었다[1, 4, 5]. 프리픽스(prefix) 캐싱 기법은 미디어 데이터의 처음 일부를 캐싱하는 기법으로서 서버에서 프록시로의 데이터 패칭시 경로 상에서 발생하는 지연과 트래픽의 문제를 개선하기 위한 기법이다 [1, 4]. 그러나 이 기법은 포스트픽스(postfix) 미디어 객체를 스트리밍 할 때 사용빈도와 중요도 등 스트리밍 결정 요소가 고려되지 않아 데이터 패킷 처리 시 많은 손실이 발생한다. 그리고 세그먼트 기반 캐싱 기법은 미디어 객체 전체를 캐싱하는 것이 아니라 객체 일부를 캐싱하는 기법으로서 시작 지연을 줄이기 위해 제안되었다[5]. 그러나 이 기법은 객체 일부를 캐싱할 때 캐싱 교체에 따른 유사도 등이 고려되지 않아 연속적인 스트리밍 객체의 경우 프로кси 지터 문제가 발생하고 있다. 프로кси 지터는 스트리밍 끊김 현상과 혼잡을 유발하여 시스템의 성능을 저하시키고 이로 인하여 스트리밍 손실이 크게 발생한다. 또한 Wang et al. [6]은 프리픽스 캐싱에서 발생하는 트래픽 지연 문제를 해결하기 위하여 양방향 프로кси 스트리밍 기법 BPS (bidirectional proxy streaming)을 제안하였다. 그러나 이 기법은 캐싱 교체가 수행되는 세그먼트들에 대해서 캐싱 적합성이 고려되지 않아 교체로 인한 지연과 캐시 히트율이 떨어지는 문제가 발생한다.

본 논문에서는 이러한 문제를 해결하기 위하여 미디어 객체 일부를 세그먼트 단위로 분할하고, 분할된 객체 세그먼트들에 대해서 유사도를 적용한다. 적용된 유사도는 트래픽으로 인한 지연과 프로кси 지터로 인한 끊김 현상을 최소화함으로써 연속적인 스트리밍 미디어 서비스를 보장하도록 한다. 이를 위하여 프로кси 캐시 구조에서 스트리밍 미디어 객체는 다음과 같다고 가정한다. 첫째, 스트리밍 미디어 객체는 세그먼트 단위로 분할 가능하다. 크기가 큰 스트리밍 미디어 객체를 하나의 객체로 간주하여 객체 전체를 캐싱하는 것은 프로кси 지터를 발생하기 때문이다. 둘

째, 대역폭 제약, 캐시 용량 제약으로 인한 문제를 최소화하기 위해 세그먼트들을 로컬 캐시 블록에 저장된다. 셋째, 캐시에서 인코딩율과 디코딩율은 서로 일치된다. 즉 인코딩율과 디코딩율이 일치되지 않으면, 언더플로우로 인한 지터 지연과 오버플로우로 인한 혼잡이 발생되기 때문이다.

따라서 본 논문에서는 이와 같은 가정에 따라 프로кси 캐시에서 스트리밍 미디어 서비스를 향상시키고, 캐싱 교체로 인한 손실을 최소화하고자 한다. 이를 위해서 본 논문에서는 캐시 용량 및 세그먼트 길이를 고려하여 객체 세그먼트들을 분할하였으며, 인코딩율과 디코딩율을 고려하여 언더플로우와 오버플로우가 발생하지 않도록 유사도를 고려하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 살펴하며, 3장에서는 제안된 기법의 스트리밍 캐싱 손실 최소화 메커니즘에 대해서 살펴본다. 4장에서는 제안된 기법의 시뮬레이션 결과에 대해서 살펴보고, 끝으로 결론에 대해서 살펴본다.

II. 관련 연구

프록시 캐싱 기법은 클라이언트와 프로кси 사이의 트래픽을 줄이고, 클라이언트가 데이터를 요청할 때 접근 시간과 대기시간을 줄이기 위한 기법이다. 이 기법은 프로кси에서 사용빈도가 높거나 관심도가 높은 객체들을 우선적으로 캐싱하여 콘텐츠들을 효율적으로 전송한다. 프로кси에서 클라이언트가 요청한 미디어 객체들을 스트리밍하기 위한 캐싱 알고리즘으로는 NetBlotz[7]과 TranSquid[8]의 웹 객체 캐싱 기법이 있다. 그러나 이들 기법은 비스트리밍 객체에는 효율적이지만 크기가 큰 스트리밍 미디어 객체들에는 효율적이지 못하다. 미디어 객체들을 스트리밍하고 캐싱하는 일은 미디어 웹 객체들에 비해서 훨씬 복잡하고 어렵기 때문이다.

따라서 기존의 웹 캐싱 방법에서 벗어나 미디어 객체들을 스트리밍하고 캐싱할 수 있는 새로운 캐싱 기법이 요구되고 있다. 이러한 요구에 따라 프로кси에서 미디어 객체들을 캐싱하여 스트리밍하기 위한 프로кси 캐싱 기법들이 제안되고 있으며, 이들 기법은

프록시에서 미디어 객체 분할과 유사도에 따른 손실 등을 고려하고 있다 [9, 10, 11].

웹 캐싱 기법과 달리 미디어 캐싱 기법은 미디어 객체들의 일부만 캐싱할 수도 있고, 미디어 객체 전체를 캐싱할 수 있는 기법이다. 이를 위해서 Verschure et al.[12]는 서버 스케줄링을 이용한 부분 캐싱 기법을 제안하였다. 프리패칭 알고리즘은 트래픽을 줄이기 위하여 자주 참조되는 미디어 객체들을 프록시에 미리 프리패칭하여 캐싱하는 기법이다. 그리고 Lee et al [13]은 미디어 스트림 객체를 캐싱하기 위하여 요약기법을 제안하였다.

이외에도 Rejaie et al.[14]은 계층형 비디오 캐싱 기법을 제안하였고, Ma et al.[15]는 캐시된 데이터의 크기와 대역폭에 따라 각각의 미디어 객체를 여러 단계로 캐싱하는 점진적 캐싱 기법을 제안하였다. 그러나 이들 기법은 프록시 구조에서 스트리밍 되는 미디어 객체들에 대해서 인코딩율과 디코딩율을 고려하지 않 지터 지연과 혼잡으로 인한 패킷 손실 증가 그리고 캐시 히트를 감소와 같은 문제가 발생하고 있다.

III. 캐싱교체 손실제어 메커니즘

3-1 객체 세그먼트 분할

프록시 캐시 구조에서 교체로 인한 손실을 최소화 하기 위해서는 미디어 객체들을 세그먼트단위로 분할해야 한다. 세그먼트 분할 과정은 그림 1과 같다.

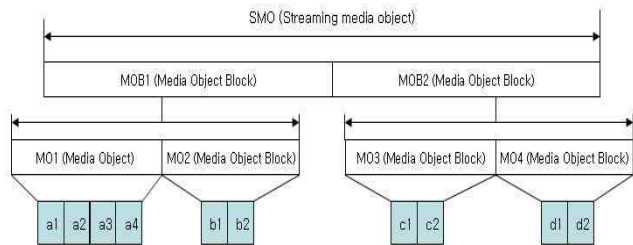


그림 1. 미디어 객체 분할
Fig. 1. Media Object Partition.

스트리밍 미디어 객체 SMO는 여러 미디어 객체 블록 MOB로 분할되며, MOB는 다시 미디어 객체 MO와 세그먼트 단위로 분할된다. 이때 분할된 세그

먼트들은 최소 전송 단위가 된다. SMO로부터 분할된 여러 세그먼트들은 캐시 교체과정을 거쳐 스트리밍되며, 캐싱 승인정책에 따라 각 세그먼트들에 캐싱 값이 할당된다. 즉 프록시 캐시에서 스트리밍 서비스가 수행될 때 세그먼트 크기는 서비스에 민감한 반응을 보이게 된다. 따라서 캐시에 따른 손실이 최소화 되도록 인코딩율과 디코딩율을 고려한 세그먼트 분할을 수행한다.

3-2 유사도 관계

그림 2의 SMO는 각기 구별된 세그먼트들로 구성된 스트리밍 루트 노드이다. 각각의 노드는 임의의 출구인 에지를 가지고 있으며, 루트 트리의 리프 노드들은 서로 다른 계층에 존재할 수 있다. LSMO는 SMO의 모든 레이블들의 집합이라고 하고, LLSMO는 SMO의 리프 노드들에 대한 모든 레이블들의 집합이라 하자. LLSMO는 원소 도메인이며, 이 경우 LLSMO는 중첩된 계층 구조를 구성하게 된다. 예를 들어 $LLSMO = \{b1, b2, \dots, s1, s2, \dots, m1, m2, \dots, c1, c2, \dots\}$ 일 때 그림 2와 같은 음악 SMO구조를 가정하자.

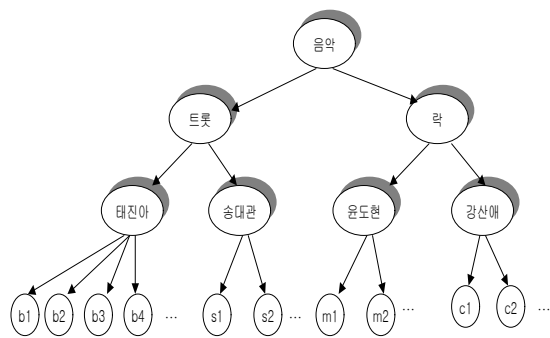


그림 2. 음악파일 SMO 구조
Fig. 2. SMO Structure for Music File.

유사도 관계는 LLSMO로부터 분할된 객체 세그먼트들로부터 결정되며, 유사도 관계를 결정하기 위해 유사도 함수가 적용된다. 유사도 함수는 객체 세그먼트들에 대한 상대적 중요도를 나타낸다. 유사도 관계가 결정된 세그먼트들은 계층적 구조를 구성하며, 이와 같이 구성된 계층구조를 유사도 관계 트리 SRT (Similarity Relation Tree)구조라 한다.

그림 3은 그림 2로부터 구성된 SRT 구조를 보여주

고 있으며, LLSMO로부터 5개의 서브 SRT그룹이 생성되게 된다.

- SRT1={ 음악, 트롯, 태진아, b1, b2}
- SRT2={ 음악, 트롯, 태진아, b3, b4}
- SRT3={ 음악, 트롯, 송대관, s1, s2}
- SRT4={ 음악, 락, 윤도현, m1, m2 }
- SRT5={ 음악, 락, 강산애, c1, c2}

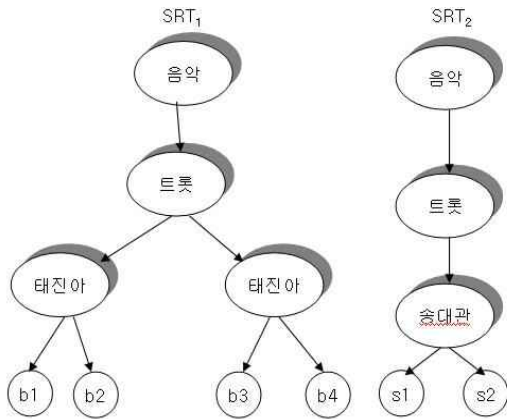


그림 3. 그림2의 SRT 구조
Fig. 3. SRT Structure for Fig. 2.

이때 SRT구조에서 노드의 깊이는 SMO에서 분할된 세그먼트까지 도달하는 에지들의 수이다. 따라서 SMO에서 임의의 두 리프 노드 leaf1, leaf2가 주어질 때, 유사도 관계 결정은 다음과 같이 정의된다.

정의 1. SMO에서 임의의 두 리프 노드 leaf1, leaf2가 주어질 때 리프 노드 leaf1, leaf2가 적어도 한 개의 유사도 관련 기반 공통조상 SRCA (Similarity Relation-based Common Ancestor)를 가지고 있으면 SRCA는 SRCA(leaf1, leaf2)이다.

예를 들어, SRCA가 적어도 한 개의 공통 조상을 가지고 있다고 하면, 그림2에서 SRCA(b1, b2)=태진아이며, SRCA(b1, s1)=트롯임을 알 수 있다.

3-3 OSS(Object Segment Similarity)

OSS 결정은 MOB의 각 객체원소들에 대해서 유사도를 측정하는 방법이며, 유사도 측정 후 각 객체원소들에 대한 가중치 평균을 구하여 두 집합 구조에

대한 유사도를 결정한다. 이때 객체원소의 유사도는 다른 집합구조와 해당 원소의 관계성에 의해 결정된다. 따라서 MO1과 MO2는 유사도가 수행할 객체집합이라 하고, s1과 s2는 유사도 관계가 결정되는 세그먼트라 할 때 s1과 s2의 SRCA는 $SRCA_{s_1, s_2}(l_1)$ 이 된다. 예를 들어 그림3에서 $SRCA_{SRT_1, SRT_2}(b_i) = \{\text{태진아, 트롯, 음악}\}$ 이다.

이것은 bi의 공통 조상이 SRT1과 SRT2이고, 공통조상에 대하여 결합성과 관계성이 수행되기 때문이다.

이처럼 SRCA가 존재할 때 미디어 객체 MO와 객체 세그먼트 s1, s2에 대한 결합성은 다음과 같이 정의된다.

정의 2. 객체 세그먼트 s1, s2에 대한 유사도 관계 $leafsim_{(SRT_1, SRT_2)}(s_i) = \frac{depthSRCA_{(SRT_1, SRT_2)}(s_i)}{depth(s_i)}$ 이다.

여기서 $leafsim_{(SRT_1, SRT_2)}(s_i)$ 는 객체 세그먼트 si가 SRT와 어느 정도 일치되는지는 나타내는 결합성이다. 이때 최대 결합성은 1이며, 최소 결합성 0이다.

예를 들어 그림 3에서 b1의 $leafsim_{(SRT_1, SRT_2)}(s_i)$ 는 0.66이며, b2도 0.66이다.

따라서 SRT1과 SRT2에 따른 세그먼트 원소 집합 E1과 E2에 대한 유사도는 다음과 같이 정의된다.

정의 3. $sim(E_1, E_2) = \frac{\sum_{s \in E_i} leafsim_{(SRT_1, SRT_2)}(s_i) \times \omega}{\sum_{s \in E_i} (leaf_i) \times \omega}$ 이다.

여기서 ω 는 가중치이다. 따라서 그림 3의 b1, b2에 대한 유사도는 0.66이며, 이것은 b1의 leafsim과 b2의 leafsim이 같기 때문이다.

3-4 유사도 기반 스트리밍

유사도가 결정된 객체 세그먼트들은 캐싱 교체를 위해 캐시 블록 CB (Cache Block)에 저장된다. CB에

저장된 객체 세그먼트들은 객체클래스를 구성하게 되며, 유사도에 따라 객체 세그먼트들이 재정렬된다. 이제 하나의 로컬 CB안에는 k 개의 객체 세그먼트가 있고, 객체 세그먼트 s_i 는 길이 l_j 를 가지고 있다고 가정하자. 그리고 로컬 CB 안에는 유사도가 측정된 n 개의 객체 세그먼트들이 있다고 가정하자. 그림 4는 유사도가 수행된 객체 세그먼트들의 캐시 구조이며, [10]에서 제안된 $M_{filtering}^{0.1 \cup 0.4}(s)$ 와 $M_{filtering}^{0.7 \cup 0.9}(s)$ 를 적용했을 때 CB구조는 그림 5, 그림 6과 같다.

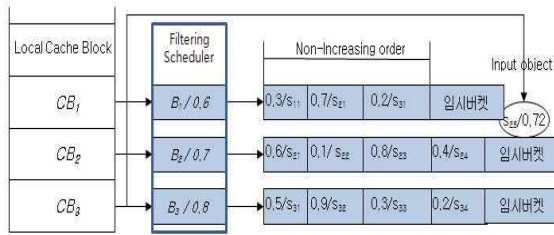


그림 4. CB 구조
Fig. 4. CB Structure.

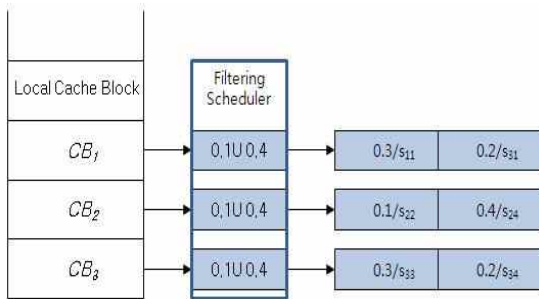


그림 5. $M_{filtering}^{0.1 \cup 0.4}(s)$ 일 때의 CB 구조
Fig. 5. CB Structure with $M_{filtering}^{0.1 \cup 0.4}(s)$.

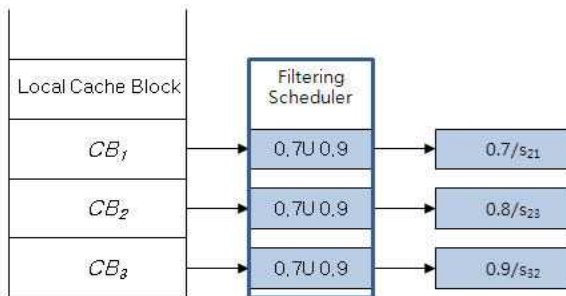


그림 6. $M_{filtering}^{0.7 \cup 0.9}(s)$ 일 때의 CB 구조
Fig. 6. CB Structure with $M_{filtering}^{0.7 \cup 0.9}(s)$.

이때 새로 입력된 객체세그먼트들은 캐시 블록 구조의 임시버킷에 저장한다. 그리고 임시 버킷에 저장된 세그먼트들은 필터링 스케줄러에 의해서 피드백 적합도가 수행되며, 피드백 적합도에 따라 캐싱 교체를 결정한다. 그림 4에서 보는바와 같이 새로 입력된 세그먼트 s_{25} 가 $s_{25}/0.72$ 이라 할 때, 이 객체는 적합도가 수행된 후 CB_2 에 저장된다. 이처럼 CB에 저장된 객체 세그먼트들은 캐싱 교체를 만족하는 세그먼트들로서 스트리밍에 따른 손실을 최소화하게 된다.

IV. 시뮬레이션 분석

4-1 시뮬레이션 환경

시뮬레이션을 위해 전체 미디어 객체들의 수는 N 으로 하였으며, 표준 비디오 프레임은 6,000개를 소스로 이용하였다. 본 논문에서는 시뮬레이션 편의를 위하여 미디어 객체의 크기를 최대 5Mbyte 이내로 제한하였으며, 이들 객체의 분포는 Pareto 분포에 기반을 두었다. 비트율은 최대 1.44Mbps, 패킷크기는 512kb, 링크 대역폭은 10/100Mbps, 그리고 평균 채널 대역폭은 약 2.4Mbps로 설정하였다. 전체 시뮬레이션 시간은 560s 동안 수행하였으며, 스트림의 t_s 는 [1, 20s], ω 는 $0 \leq \omega < 1$ 로 세트시켰다. 초당 교체되는 스트림 데이터는 10개 이내로 제한하였으며, 최대 PSNR(Peak Signal to Noise Ratio)은 30dB이내로 제한하였다. 미디어 객체들의 분포 측정은 Zipf-like 분포에 기반을 두었으며, 분포는 $\delta=0.7$ 로 설정하였다. 기본적인 캐시 용량은 $0.05 * (\sum \text{객체크기})$ 라고 가정하였으며, 서버로부터 미디어 객체들을 패치하는 데 걸리는 지연은 지수분포를 따른다고 가정하였다. 그리고 원본 서버로부터 미디어 객체를 캐시하는 데 걸리는 지연은 무시하였다.

4-2 시뮬레이션 결과

성능 평가를 위해서 본 논문에서는 캐시크기와 유사도를 변화시켜 가면서 시뮬레이션을 수행하였다. 시뮬레이션에서 사용된 주요 성능 척도는 캐싱 교체에 따른 평균 지연 제어율, 평균 캐시 처리율, 그리고

평균 캐시 응답율이다. 따라서 본 논문에서는 이들 성능 척도에 기반을 두고서 시뮬레이션을 수행하였으며, 성능 비교는 prefix 기법, segment 기법, BPS 기법, 그리고 제안된 기법으로 구분하여 비교 분석하였다.

4-2-1 유사도에 따른 평균 지연 제어율

첫 번째 성능평가는 유사도에 따른 평균 지연 제어율이다. 중요도가 0.5 이하일 때는 버퍼 캐시 제어에 영향을 주게 되며, 스트리밍 QoS에도 영향을 미치게 된다. 특히 중요도가 낮을 때에는 캐시 크기로 인한 지연율은 높아지며, 캐시 히트율 또한 떨어지게 된다. 이에 대한 시뮬레이션 결과는 그림 7과 같다.

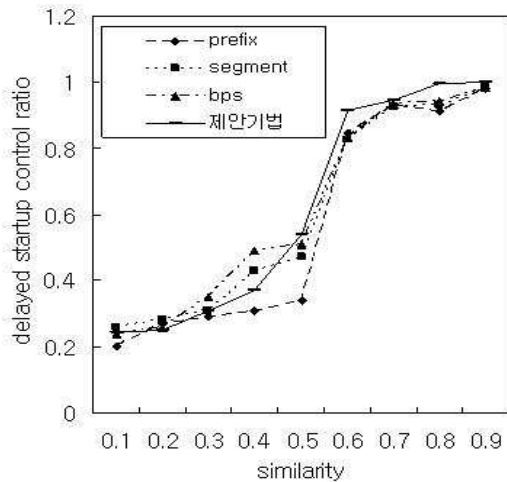


그림 7. 평균 지연 제어율
Fig. 7. Average delayed startup control ratio.

그림 7에서 보듯이 제안된 기법은 비교적 우수한 세그먼트 기법보다 시작 지연 제어율의 성능이 효율적임을 알 수 있다. 이것은 제안된 기법에서는 스트림 세그먼트들에 대해서 유사도와 적합성 피드백이 적용되었기 때문이다. 따라서 스트림 세그먼트의 크기를 프록시 캐시에 적합하도록 분할 할 때 캐시가 효율적으로 제어됨을 알 수 있다. 본 논문에서는 스트림 세그먼트가 너무 작거나 캐시 용량을 초과하는 미디어 스트림 데이터들에 대해서는 [10]의 μ 를 적용하여 필터링이 수행되도록 하였다.

4-2-2 캐시 크기에 따른 평균 캐시 처리율

두 번째 성능평가는 캐시 크기를 점차 증가시키면서 평균 캐시 처리율을 평가하였다. 스트림 세그먼트에 따른 캐시 처리 시뮬레이션 결과는 그림 8과 같다.

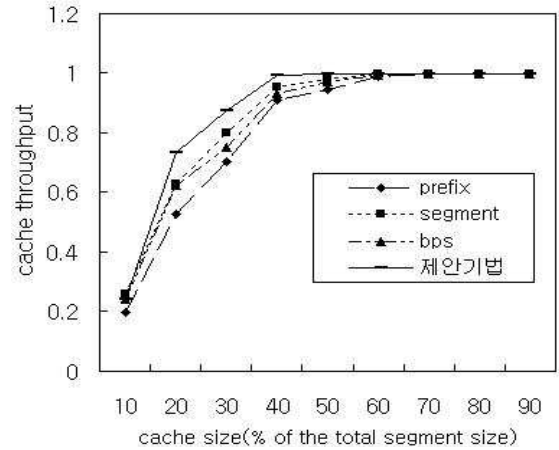


그림 8. 평균 캐시 처리율
Fig. 8. Average cache throughput.

그림 8에서 보듯이 캐시 크기가 증가할 때 제안된 기법의 캐시 처리율은 매우 효율적임을 알 수 있다. 특히 캐시 크기가 50% 이상일 때 패킷 손실율은 거의 발생하지 않았으며 또한 처리율이 매우 효율적임을 알 수 있다. 이것은 캐시 크기가 중요도에 영향을 받기 때문이다. 즉 중요도가 낮고, 캐시 크기가 작으면 스트림 세그먼트들에 대한 참조가 분산되기 때문에 지연으로 인한 패킷 손실이 크게 발생하고, 전체 캐시 처리율은 떨어지게 된다. 따라서 제안된 기법은 각 캐싱 교체 세그먼트들에 대해서 유사도와 적합성 피드백을 반영하였기 때문에 캐시 크기가 증가할 때 캐시 처리율이 증가하게 됨을 알 수 있다.

4-2-3 유사도에 따른 평균 캐시 응답율

세 번째 성능평가는 유사도에 따른 평균 캐시 응답율이다. 시뮬레이션을 위해 유사도 값을 0.1에서부터 0.9까지 변화시켜가면서 평균 캐시 응답율의 성능을 평가하였다. 이에 대한 분석 결과는 그림 9와 같다.

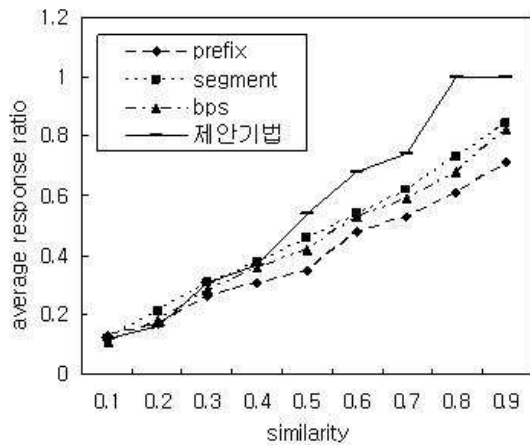


그림 9. 평균 캐시 응답율
Fig. 9. Average cache response ratio.

그림 9에서 보듯이 제안된 기법은 캐시 응답을 관점에서 볼 때 비교 기법들에 비해서 성능이 우수함을 알 수 있다. 특히 비교적 성능이 우수한 세그먼트 기법과 비교해 볼 때 약 10%의 성능이 향상되었음을 알 수 있다. 이에 반해서 프리픽스 기법과 bps 기법의 성능이 상대적으로 낮은 이유는 스트림 세그먼트 데이터의 크기 및 캐시 용량에 따른 유사도와 적합성 피드백이 고려되지 않았기 때문이다. 따라서 제안된 기법은 캐시 크기를 고려하여 스트리밍 미디어 객체에 유사도와 적합성 피드백을 반영하였기 때문에 크기가 큰 스트림 데이터가 새로 입력되거나 캐싱 교체가 수행될 때 캐시 제약으로 인한 손실을 줄일 수 있는 장점을 제공하게 된다.

V. 결 론

모바일 프록시 구조에서 스트리밍 미디어 객체는 캐싱 교체시 지터 지연, 혼잡 등으로 인한 손실이 크게 발생하고 있다. 본 논문에서는 캐싱 교체시 발생하는 손실을 최소화하기 위하여 유사도 기반의 손실 최소화 기법을 제안하였다. 제안된 기법은 유사도를 적용하기 위하여 미디어 객체를 세그먼트 단위로 분할하였으며, 분할된 객체 세그먼트들에 대해서 유사도 관계 트리 SRT를 구성하였다. SRT가 구성된 후 객체 세그먼트는 유사도에 따라 CB가 재구성되도록 하였으며, 이렇게 재구성된 CB의 객체 원소들에 대

해서 캐싱 교체가 수행되도록 하였다. 시뮬레이션 결과 제안된 기법은 prefix 캐싱 기법, segment 캐싱 기법, 그리고 bps 캐싱 기법에 비해서 캐싱 시작 지연 제어율, 캐시 처리율, 그리고 캐시 응답율의 성능이 우수함을 알 수 있었다.

참 고 문 헌

- [1] K. L. Wu, S. Yu, and J. L. Wolf, "Segmentation of Multimedia Streams for Proxy Caching," *IEEE Transactions on Multimedia*, Vol. 6(5), pp 770-780, 2004.
- [2] Z. Miao, A. Ortega, "Proxy caching for Efficient Video Services over the Internet," *In Packet Video Workshop*, pp. 1-21, 1999.
- [3] S. Krishnamachari, M. Van der Schaar, S. Choi, and X. Xu, "Video Streaming over Wireless LANs: A Cross-Layer Approach," *in Proceedings of Packet Video Workshop*, Nantes, France, pp. 1-9, 2003.
- [4] S. Sen, J. Rexford, and D. Towsley, "Proxy Prefix Caching for Multimedia Streams," *in Proc. IEEE INFOCOM, New York*, 1999, pp. 1310-1319, 1999.
- [5] S. Chen, B. Shen, S. Wee, and X. Zhang, "Segment-based Streaming Media Proxy: Modeling and Optimization," *IEEE TRANSACTIONS ON MULTIMEDIA*, Vol. 8, No. 2, pp. 243-256, 2006.
- [6] J. Wang and D. L. Lowhorn, "Proxy Caching for Wireless Multimedia Streaming," *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, pp. 1100-1105, 2005.
- [7] S. Acharya, H. Korth, and V. Poosala, "Systematic Multiresolution and its application to the World Wide Web," *in Proc. IEEE ICDE'99, Sydney, Australia*, Mar. 1999, pp. 40-49, 1999.
- [8] A. Maheshwari, A. Sharma, K. Ramamrithan, and P. Sheonoy, "Transquid: Transcoding and Caching Proxy for Heterogeneous e-commerce Environments," *in Proc. IEEE INFOCOM 2002, San Jose, CA*, pp. 50-59, 2002.
- [9] C. D. Lee, T. W. Jeong, "Fuzzy Filtering Based

Segment Grouping for User-Centered Multimedia Streaming Service in P2P Distribution Mobile Networks," *Journal of Internet Technology*, Vol. 11, No. 5, pp. 651-658, 2010.

- [10] C. D. Lee, "Object version Transcoding for streaming media service in Wireless Mobile Networks," *KONI*, vol. 15, no. 3, pp. 355-363. 2011.
- [11] C. D. Lee, "Multi-level streaming using Fuzzy Similarity in P2P Distribution Mobile Networks," *KONI*, vol. 15, no. 3, pp. 364-371. 2011.
- [12] O. Verscheure, C. Venkatramani, P. Frossard, and L. Amini, "Joint Server Scheduling and Proxy Caching for Video Delivery," *Computer Communications*, Vol. 25, No. 4, pp. 413-423, 2002.
- [13] S. J. Lee, W. Y. Ma, and B. Shen, "An Interactive Video Delivery and Caching System using Video Summarization," *Computer Communications*, Vol. 25, No. 4, pp. 424-435, 2002.
- [14] R. Rejaie, H. Yu, M. Handely, and D. Estrin, "Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Vol. 2, pp.980-989, 2000.
- [15] W. Ma and D. H. C. Du, "Design a Progressive Video Caching Policy for Video over Wide Area Networks with Proxy Server," *IEEE TRANSACTIONS MULTIMEDIA*, Vol. 6, No. 4, pp. 599-610, 2004.

이 종 득 (李鍾得)



1983년 2월 : 전북대학교 컴퓨터과 학과(이학사)

1989년 2월 : 전북대학교 컴퓨터과 학과(이학석사)

1998년 2월 : 전북대학교 컴퓨터과 학과(이학박사)

1992년 3월~2002년 2월 : 서남대학교

컴퓨터통신학과 교수

2002년 2월~2012년 5월 현재 : 전북대학교 전자공학부 교수

관심분야 : 무선 모바일 네트워크, 무선센서 네트워크, MIMO, 유비쿼터스 통신, 등