

무선 스트리밍 QoS를 위한 이득 기반 세그먼트 캐싱

Profit-based Segment Caching for Wireless Streaming QoS

이종득*

Chong-Deuk Lee*

요약

본 논문에서는 무선 채널 상에서 연속적인 스트리밍 QoS 보장을 위한 새로운 이득 중요도 기반의 세그먼트 캐싱 제어 메커니즘을 제안한다. 제안된 기법은 QoS (Quality of Service)보장을 위해 *SSCP* (Single Segment Caching Profit)와 *MSCP* (Multiple Segment Caching Profit)에 의해 캐싱 제어를 수행한다. 이들 *SSCP*와 *MSCP* 캐싱 제어는 프록시에서 스트리밍이 수행될 때 캐시의 성능을 최적화하기 위한 것이다. 제안된 기법의 성능을 알아보기 위하여 본 논문에서는 고정 분할 기법, 임계값 기반 기법, *SSCP* 와 *MSCP*의 성능을 비교하였으며, 그 결과 제안된 기법이 다른 기법들에 비해서 성능이 우수함을 보인다.

Abstract

This paper proposes a new profit popularity-based segment caching control mechanism for assuring a consecutive streaming QoS (Quality of Service) in the wireless channel. Then, the proposed mechanism operates *SSCP* (Single Segment Caching Profit) and *MSCP* (Multiple Segment Caching Profit) for assuring a QoS. *SSCP* and *MSCP* is to optimize the cache performance when is performed the streaming in the proxy. The proposed mechanism simulated to evaluate such mechanisms as fixed-partition mechanism, weight-based mechanism, *SSCP*, and *MSCP*. Simulation results show that the proposed mechanism has superior performance compared to other mechanisms.

Key words : wireless channel, channel fading, network congestion, *SSCP*, *MSCP*

I. 서론

최근에 스마트 모바일 서비스의 증가로 무선 채널 상에서 고품질의 멀티미디어 스트리밍 서비스를 위한 많은 연구가 제안되고 있다. 그러나 무선 채널은 유선 네트워크와 달리 대역폭 제약으로 인한 끊김 현상이 발생하고, 클라이언트에 대한 지연현상이 발생하는 문제점을 가지고 있다. 특히 단말 간 서비스되는 스트리밍 미디어 서비스는 높은 채널 비트 오류

율, 채널 페이딩, 그리고 핸드오프 등으로 인하여 QoS 저하 현상이 발생되고 있으며, 이로 인하여 스트리밍 품질이 떨어지는 문제가 발생하고 있다. 무선 채널 상에서 이러한 문제를 해결하기 위한 기법 중의 하나가 프록시 캐싱 기법이다[1]-[4].

프록시 캐싱 기법은 사용빈도가 높거나 관심도가 높은 스트리밍 객체들을 우선적으로 캐싱하여 프록시의 캐싱 오버헤드를 줄이기 위한 기법으로서 시스템의 효율성과 사용자 QoS를 향상시키기 위한 것이

* 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

다[5]-[7]. 이 기법은 클라이언트 근처에서 사용빈도가 높거나 관심도가 높은 스트리밍 객체들을 우선적으로 캐싱함으로써 네트워크 트래픽을 감소시키고, 바이트 히트율을 증가시키는 장점을 가지고 있다. 그러나 무선 채널 환경에서 프록시 캐싱은 텍스트, 이미지와 같은 이산 미디어 스트리밍에는 효과적이거나 멀티미디어와 같은 연속적인 스트리밍 미디어 캐싱 서비스에는 부적합하다. 이러한 이유로 연속 미디어는 이산 미디어에 비해 대용량이며, 대역폭, 채널 비트 오류율, 그리고 채널 페이딩과 같은 무선 스트리밍 조건들에 많은 제약을 받기 때문이다. 이러한 제약으로부터 능동적인 스트리밍을 보장하기 위해서는 미디어 객체 세그먼트들에 대한 이득을 측정하여 캐싱을 제어해야 한다. 제안된 이득 기반 세그먼트 캐싱 메커니즘은 캐시 크기 및 트랜스코딩을 적응적으로 수행하며, 프록시 캐시와 클라이언트 사이의 데이터 전송을 보다 능동적으로 지원한다.

최근에는 RTSP (Real-Time Streaming Protocol) 상에서 프리픽스 캐싱 서비스를 무선망에 적용시키고, 무선상황 및 핸드오프 시 캐싱 프록시의 성능을 향상시키기 위해 세그먼트 기반 프록시 캐싱 기법이 제안되고 있다[3]-[5]. 세그먼트 기반의 프록시 캐싱 기법은 미디어 객체 전체를 캐싱하는 것이 아니라 미디어 객체 일부를 선택해서 캐싱하는 기법이다. 기존의 세그먼트 기반의 프록시 캐싱 기법은 스트리밍 데이터를 릴레이하기 위해 객체 분할, 전송율 제어, 프레임 블럭화, 그리고 임계값을 적용한 캐싱 제어 기법 등이 주로 연구되었다[8]-[11]. 그러나 이들 기법은 클라이언트에서 스트리밍 전송을 연속적으로 보장하지 못하는 문제점을 가지고 있다. 프록시 캐시에서 미디어 객체 스트리밍은 세그먼트 단위로 캐시되기 때문에 캐시될 세그먼트와 그렇지 않을 세그먼트를 식별하는 일이 매우 중요하다. 만일 세그먼트를 식별하지 않는다면 제약조건으로 인하여 세그먼트들의 전송을 지연시키는 프록시 지터 (proxy jitter)가 발생한다. 프록시 지터는 스트리밍 미디어 서비스에 끊김 현상을 유발하며, 이로 인하여 혼잡과 전송지연이 증가하고, 패킷 손실이 증가하게 된다.

따라서 본 논문에서는 무선 채널 상에서 프록시 캐싱과 트랜스코딩에 적응적으로 대응하여 캐시 히

트율을 향상시키고, 프록시 지터를 최소화하여 스트리밍 QoS를 보장하기 위한 이득 기반 세그먼트 캐싱 제어 메커니즘을 제안한다. 이를 위해서 본 논문에서는 무선 채널 상에서 클라이언트가 요청한 미디어 스트리밍의 QoS를 보장하기 위해 단일 세그먼트 캐싱 이득 *SSCP*와 다중 세그먼트 캐싱 이득 *MSCP* 알고리즘을 제안하며, 제안된 알고리즘은 스트리밍 미디어 서비스에 적응적으로 대응하는 장점을 가지게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 살펴보고, 3장에서는 이득 기반 세그먼트 캐싱 제어 메커니즘에 대해서 살펴보고, 4장에서는 제안된 기법의 성능에 대해서 살펴본다. 끝으로 5장에서는 결론에 대해서 살펴본다.

II. 관련 연구

세그먼트 기반의 프록시 캐싱 기법은 미디어 객체들을 스트리밍 서비스하기 위한 기법으로서 세그먼트 전체를 캐싱하는 것이 아니라 세그먼트 일부를 선택해서 캐싱하는 기법이다. 전치캐싱 (Prefix Caching) 기법 [2]는 미디어 객체들을 세그먼트하기 위한 초기의 기법으로서 이 기법은 세그먼트를 전치와 후치 (suffix)로 구분하여 캐싱을 수행한다.

Rejaie et al. [8]은 세그먼트들을 같은 크기로 분할하여 캐싱하는 고정분할 캐싱기법 (fixed-partition caching)을 제안하였으며 이 기법은 클라이언트의 접근 패턴에 따라 세그먼트 길이를 일정한 크기로 분할하여 캐싱을 수행하는 기법이다. 그러나 이 기법은 일정한 크기로 인해 히트율은 증가시킬 수 있는 장점은 있으나, 가변적인 캐시 용량에 적응적으로 대응하지 못하는 문제점을 가지고 있다. [9]는 임계값을 이용한 스트리밍 미디어 서비스 기법을 제안하였으며, 이 기법은 임계값보다 큰 객체 프레임을 추출하여 캐싱을 수행하는 기법이다. 그러나 이 기법은 프록시 캐싱을 위해 스트리밍 대역폭에 적응적으로 대응할 수 있으나 프레임의 수가 증가되거나 프레임에 대한 프로파일 정보가 제공되지 않을 경우 바이트 히트율이 떨어지는 문제점을 가지고 있다. Wu et al. [12]에서는 캐싱을 수행하기 위하여 프레임을 블럭화하는 기법을 제안하였으나 이 기법은 비연속적인 프레임들을 스트리밍할 때 블럭화의 크기로 인하여 캐싱 성

능이 떨어지는 문제점을 가지고 있다.

Hefeeda et al. [13]은 P2P 기반의 미디어 스트리밍을 위해 선택 기법을 적용하여 스트리밍을 제어하는 기법을 제안하였다. 이 기법은 송신측 후보들의 토폴로지와 네트워크 통신 품질을 고려한 기법으로서 피어들의 선호도 (goodness)에 따라 송신자가 최상의 스트리밍 객체를 선택하여 스트리밍 서비스를 수행하도록 하였다. 그러나 이 기법은 전송율 및 미디어 객체 크기를 고려하지 않아 미디어 캐싱을 제어할 때 전송 효율이 떨어지는 문제와 네트워크 혼잡이 발생하는 문제가 발생하고 있다.

Zhu et al. [10]의 전송율 제어 기법은 미디어 프레임의 스트리밍 서비스를 보다 융통성 있게 제어해주는 기법으로서 인코더 버퍼와 디코더 버퍼의 전송율 불일치 문제를 해결하기 위해 제안되었다. 이 기법에서 인코더 버퍼와 디코더 버퍼의 전송율 불일치는 스트리밍 QoS에 영향을 미치며, 지터 지연, 네트워크 혼잡 등의 원인을 제공하게 된다.

따라서 본 논문에서는 이와 같은 문제들을 해결하고 미디어 세그먼트의 스트리밍 QoS를 향상시키고, 혼잡 등의 성능을 개선하기 위하여 이득 기반 세그먼트 캐싱 제어 메커니즘을 제안한다.

III. 제안된 이득 기반 세그먼트 캐싱 제어

제안된 이득 기반 세그먼트 캐싱 제어 기법은 세그먼트 이득 중요도에 따라 스트리밍 QoS를 보장하기 위한 기법으로서 프록시 캐시로 하여금 스트리밍 미디어 서비스를 극대화하기 위한 것이다. 이를 위해 세그먼트 이득에 따라 캐시를 제어할 수 있도록 단일 세그먼트 이득 함수 SSCP (Single Segment Caching Profit)와 다중 세그먼트 이득 함수 MSCP (Multiple Segment Caching Profit)을 제안한다. 그리고 이들 함수에 의해 캐싱 교체가 효율적으로 수행될 수 있도록 캐싱 교체 알고리즘을 제안한다.

3-1 이득 중요도

세그먼트 이득 중요도는 세그먼트 스트리밍에 따른 지연시간 절약과 혼잡을 제어하기 위한 척도로서 프록시 캐시에서 각 미디어 객체의 처리순서를 결정하기 위하여 사용된다. 먼저 각각의 이득 중요도는 미디어 객체의 단일 세그먼트 버전 이득을 평가하여 캐싱 여부를 결정한다. 그리고 난 후 단일 미디어 객체의 여러 세그먼트 버전들에 대한 이득 중요도를 측

정하여 이득 우선순위에 따라 캐싱이 수행되도록 구조화한다.

먼저 $O_{n,s}$ 를 세그먼트 이득 척도를 위한 미디어 객체 n 의 세그먼트라 하자. 다른 미디어 객체 세그먼트들에 대한 참조율은 통계적으로 서로 독립이라 가정하고, m,s 로 나타낸다. 여기서 m,s 는 미디어 객체 n 의 세그먼트들에 대한 평균 참조율이다. 캐시에서 미디어 객체 n 의 세그먼트 s 를 처리하는 처리율은 $C_{s-throughput}$ 로 표현하며, 처리율의 결과에 따라 이득 중요도 $\alpha(0 \leq \alpha \leq 1)$ 가 결정된다. 그리고 클라이언트에서 프록시 캐시에 세그먼트를 요청했을 때 발생하는 평균 요청지연시간은 R_{delay} 로 표현하며, 미디어 객체 n 의 세그먼트들에 대한 크기는 $P_{cache-capacity}$ 로 표현한다. 세그먼트 s 에 대한 인코딩율은 E_r 로 표현하며, 세그먼트 s 에 대한 디코딩율은 D_r 로 표현한다. 여기서 $D_r > E_r$ 이다. 표1은 세그먼트 캐싱을 위한 이득 파라미터들이며, 이득 중요도에 따라 세그먼트 캐싱 순서가 결정된다.

표1 세그먼트 캐싱을 위한 이득 파라미터
Table 1. Profit Parameters for Segment Caching

parameters	Meaning
$O_{n,s}$	객체 n 의 세그먼트 s
$r_{n,s}$	세그먼트 s 에 대한 평균 참조율
s_{length}	세그먼트 크기
$C_{s-throughput}$	캐시에서 세그먼트 s 의 처리율
$t_{n,s}$	임의의 세그먼트 s 를 처리하는데 걸리는 시간
α	참조율에 따른 중요도
br_s	세그먼트 s 의 비트율
R_{delay}	클라이언트에서 프록시 캐시에 세그먼트를 요청했을 때 발생하는 평균 요청지연시간
$P_{cache-capacity}$	캐시에서 세그먼트 처리 크기
E_r	세그먼트 s 의 인코딩율
D_r	세그먼트 s 의 디코딩율
$\omega(s,s')$	세그먼트 s 에서 세그먼트 s' 까지 캐싱하는데 드는 비용
B_w	세그먼트 s 의 대역폭

일반적으로 이득 중요도는 캐시 환경에 따라 변화될 수 있으며, 캐시에서 세그먼트 s 의 처리율에 따라 이득 중요도 α 가 결정된다. α 는 $C_{s-throughput}$ 에

의해 결정되며, 1에 가까울수록 이득 중요도가 높은 세그먼트로서 캐싱 제어를 효과적으로 수행하게 된다.

3-2 단일 세그먼트 캐싱 이득

단일 세그먼트 캐싱 이득은 미디어 객체의 다른 세그먼트들이 캐싱되지 않았을 때 캐싱의 우선순위를 결정하기 위한 것으로서 n 개의 세그먼트들에 대해서 각각의 캐싱 이득을 측정한다. 또한 세그먼트 캐싱 이득함수는 클라이언트들에게 세그먼트 트래픽으로 인한 지연시간을 최소화하여 응답시간을 보장하기 위한 것이다. 따라서 클라이언트가 요청하는 미디어 객체 세그먼트들을 최적으로 응답하기 위해서는 이득함수에 따른 대역폭 절약과 지연절약이 보장되어야 한다. 이처럼 대역폭 절약과 지연 절약을 보장되면 캐싱 이득은 극대화되며, 단일 세그먼트 캐싱 이득 $SSCP$ 는 식(1)과 같이 정의된다.

(정의1) $SSCP=$

$$\frac{r_{n,s} \times s_{length} \times R_{delay} \times P_{capacity} (D_r - E_r)}{O_{n,s} \times B_w \times br_s} \times \alpha \quad (1)$$

만일 클라이언트가 요청한 객체 세그먼트 s' 를 프록시 캐시가 지연을 발생하지 않고 캐싱하면, 객체 세그먼트 s' 에 대한 지연절약 DS (Delay Saving)은 식(2)와 같이 정의된다.

(정의2) $DS=$

$$\frac{r_{n,s} \times s_{length} \times R_{delay}' \times [(P_{capacity} (D_r - E_r)) - \omega(s, s')]}{O_{n,s} \times B_v \times br_s} \quad (2)$$

특히 프록시 캐시에서 미디어 객체 n 의 처음 몇 개의 세그먼트들을 캐싱하기 위해서는 지연을 최소화하기 위한 대역폭이 고려되어야 한다. 만일 사용자가 요청한 모든 세그먼트들을 미디어 서버가 직접 제공하면 이때 스트리밍 지연은 발생하지 않을 수도 있다. 그러나 $o_{n,s}$ 가 이미 캐시되고 $o_{n,s'}$ 가 새롭게 요

청되면, 이 과정에서 병목과 대기로 인한 지터 지연이 발생한다. 따라서 캐싱 $o_{n,s}$ 와 요청 $o_{n,s'}$ 사이에서 병목과 대기로 인해 발생하는 지터 지연을 최소화하기 위해 식(1)과 식(2)가 적용되게 된다.

3-3 다중 세그먼트 캐싱 이득

다중 세그먼트 캐싱 이득은 미디어 객체의 여러 세그먼트들을 서로 통합하였을 때 통합 이득에 따라 객체의 다중 세그먼트들을 동시에 캐싱하기 위한 것이다. 이 과정에서 객체 세그먼트들 사이의 캐시 용량과 세그먼트 크기가 고려되며, 객체 세그먼트들이 캐시의 용량보다 크면 캐싱율을 최적화하기 위해 객체 세그먼트들을 분할하게 된다. 이때 분할된 객체 세그먼트들은 $(s, s') \in [O_{n,s}]$ 에 추가되며, 분할에 따른 다중 세그먼트 캐싱 이득 $MSCP$ 는 식(3)과 같이 정의된다.

(정의3) $MSCP=$

$$\frac{\sum_{(s, s') \in [O_{n,s}]} r_{n,s} \times s_{length} \times R_{delay} \times P_{capacity} (D_r - E_r)}{O_{n,s} \times B_w \times br_s} \times \alpha \quad (3)$$

여기서 모든 $(s, s') \in [O_{n,s}]$ 는 세그먼트 s 에서 세그먼트 s' 까지의 캐싱을 의미한다.

$MSCP$ 또한 캐싱과정에서 분할된 다중 세그먼트들로 인하여 지터지연이 발생할 수 있으므로 대역폭과 캐시 용량을 고려하여야 한다. 특히 다중 세그먼트 분할 캐싱으로 인한 캐싱 QoS가 보장되도록 세그먼트들은 $(s, s') \in [O_{n,s}]$ 에서 관리된다.

3-4 캐싱 제어

다중 세그먼트 분할 캐싱은 객체 세그먼트가 캐시될 때 미디어 객체의 전체 세그먼트들의 이득 중요도를 측정하여 다중 세그먼트 버퍼 큐를 구성하는 기법이다. 이에 반해서 단일 세그먼트 캐싱은 각각의 객체 세그먼트의 이득 중요도를 측정하여 단일 우선순위 세그먼트 버퍼 큐를 구성한다. 다중 세그먼트 분할 캐싱은 먼저 이득 중요도를 구한 후 이들 이득 중

요도에 따라 객체 세그먼트들을 소트하며, 더 이상 캐싱할 객체 세그먼트가 없을 때까지 반복을 수행한다. 이때 가장 높은 이득 중요도를 가진 객체 세그먼트들을 반복해서 선택한다. 그러나 단일 세그먼트 캐싱은 세그먼트 이득 중요도와 지연절약을 모두 고려하여 각 캐시될 객체 세그먼트를 소트하며, 객체 세그먼트들에 대해 평균 접근 시간을 측정하여 캐싱 시간을 예측한다. 따라서 단일 세그먼트 캐싱은 객체의 세그먼트들이 더 이상 캐시 되지 않을 때까지 객체의 평균 접근 세그먼트들이 가장 큰 세그먼트들을 차례로 선택하여 캐싱을 수행하게 된다. 이때 선택된 객체 세그먼트만이 스트리밍을 위해 캐시되고, 이후 나머지는 세그먼트 버퍼에서 삭제를 수행한다. 알고리즘1은 다중 세그먼트 분할 캐싱을 위한 알고리즘이며, 입력 세그먼트에 따라 이득 중요도가 측정된다. 알고리즘에서 매개변수 BQ 는 정렬된 버퍼 큐이며, 이것은 캐시된 객체 세그먼트들을 저장하기 위해 사용된다. BQ 는 이득

중요도가 큰 값에 따라 캐시될 객체들을 오름 차순으로 정렬한다. 여기서 AS_{cached} 는 캐시될 모든 객체 세그먼트들이며, $O_{n,s}$ 은 객체 n 의 세그먼트 s 이다. 그리고 $P_{cache-capacity}$ 은 객체 세그먼트를 캐싱하기 위한 캐시 용량이며, s_{length} 는 세그먼트 크기이다.

알고리즘1 : 다중 세그먼트 분할 캐싱
 입력 : $O_{n,s}$
 출력 : 이득 중요도에 따라 정렬된 $O_{n,s}$
 // 캐시제어를 위한 제어변수($BQ, AS_{cached}, O_{n,s}, P_{cache-capacity}, s_{length}$)
 Insert $O_{n,s}$ into BQ
 if ($AS_{cached} < \text{Cache Capacity}$)
 $AS_{cached} = AS_{cached} + P_{cache-capacity} + s_{length}$
 for each segment s of object n in BQ
 Calculate $MSCP$
 SORT(BQ)
 else
 delete the segment j of object I

with minimum $MSCP$

$$AS_{cached} = AS_{cached} - P_{cache-capacity}$$

알고리즘2는 단일 세그먼트 캐싱 알고리즘이다. 알고리즘2에서 임의의 객체 세그먼트에 대하여 캐시 허락은 다음과 같이 수행된다.

1) 시스템이 접근한 모든 객체들을 캐시할 버퍼 큐가 충분하면 전체 객체 세그먼트는 평균 패칭지연과 관계없이 연속해서 캐싱을 수행할 수 있다.

2) 그러나 캐싱 시스템이 객체 세그먼트를 캐싱할 버퍼 큐가 충분하지 못하면 이득 중요도에 따라 높은 객체 세그먼트 순으로 캐시하며, 다음과 같다.

알고리즘2 : 단일 세그먼트 캐싱

입력 : $O_{n,s}$

출력 : $SSCP$ 와 DS 에 따라 정렬된 $O_{n,s}$

// 캐시제어를 위한 제어변수 ($BQ, AS_{cached},$

$O_{n,s}, P_{cache-capacity}, s_{length}$)

Insert $O_{n,s}$ into BQ

If($s_{length} < \text{Cache Capacity}$)

 Calculate $SSCP$ and DS

 SORT(BQ)

else

 Insert $C_{s-throughput}$ of $O_{n,s}$ into BQ

 while ($C_{s-throughput} > O_{n,s}$)

 Insert the segment j of object i with

 minimum DS into BQ

$AS_{cached} = AS_{cached} - P_{cache-capacity}$

IV. 시뮬레이션 분석

본 논문에서는 이벤트-지향 시뮬레이션을 통하여 성능을 분석하였다. 성능 분석을 위해 4.1절에서는 시뮬레이션 환경에 대해서 살펴보고, 그리고 4.2절에서는 시뮬레이션 결과들에 대해서 살펴본다.

4-1 시뮬레이션 환경

본 논문에서는 시뮬레이션을 위해 무선LAN과 유선 LAN이 결합된 유/무선 네트워크 환경을 적용하였다. 네트워크 토폴로지는 하나의 서버, 하나의 AP 그리고 5 개의 클라이언트 노드로 구성하였다. 프록시 노드는 AP와 상호 작용하여 위치를 공유하도록 하였으며, IEEE 802.11 상에서 무선 모바일 노드들과 통신을 수행하도록 하였다. 그리고 서버는 유선통신상에서 프록시/AP와 통신을 수행하도록 하였으며, 유선통신의 경우 포워드와 백워드가 모두 가능하도록 설정하였다. 프록시와 무선 링크간의 대역폭은 1.28Mbps로 설정하였으며, 무선 링크의 BER(Bit Error Rate)는 1.2×10^{-5} 로 설정하였다. 그리고 링크 레이어에서의 전송 지연은 $50 \mu s$ 이내로 설정하였으며, 최대 재전송 SD 패킷 수는 3개 이내로 제한하였다. 시뮬레이션 데이터로는 3개 그룹의 뉴스 비디오의 미디어 객체 세그먼트를 소스로 이용하였으며, 전체 시뮬레이션 시간은 560s로 설정하였다. 스트리밍을 위한 t_s 는 $[1, 20s]$, α 는 $0 < \alpha < 1$ 로 설정하였다. 연속적인 스트리밍 패킷 데이터는 10개로 제한하였으며, 최대 PSNR(Peak Signal to Noise Ratio)은 30dB 이내로 제한하였다. 전체 시뮬레이션 시간을 고려하여 서버로부터 미디어 객체들을 패치하는 데 걸리는 지연은 고려하지 않았으며, 원본 서버로부터 미디어 객체를 패치하는 데 걸리는 지연은 무시하였다.

4-2 시뮬레이션 결과

시뮬레이션에서 사용된 주요 성능척도는 대역폭 변화에 따른 혼잡제어, 이득 중요도에 따른 캐시 제어, 그리고 지연율에 따른 캐시 처리율이다. 따라서 이들 성능 척도에 기반을 두고서 시뮬레이션을 수행하였으며, 성능 비교는 고정분할 기법[11], 임계값 제어 기법[9], 그리고 제안된 기법으로 구분하여 수행하였다.

4-2-1 대역폭에 따른 혼잡제어

첫 번째 성능평가는 3개 그룹의 뉴스 비디오 클립 시퀀스를 이용하여 시뮬레이션을 수행하였으며 대역폭 변화에 따른 성능을 알아보기 위하여 단일 세그먼

트 캐싱과 다중 세그먼트 분할 캐싱으로 구분하여 혼잡 제어율을 측정하였다. 이에 대한 시뮬레이션 결과는 그림 1과 같다.

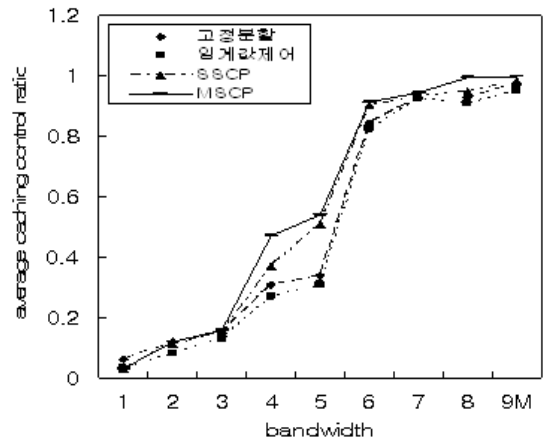


그림 1. 대역폭 변화에 따른 혼잡제어율
Fig. 1. Congestion Control Rate with Bandwidth.

그림 1에서 보듯이 대역폭이 증가할 때 평균 혼잡 제어율이 높게 나타남을 알 수 있으며, SSCP와 MSCP를 이용하였을 때 고정분할과 임계값 제어 기법에 비해서 성능이 우수함을 알 수 있다. 특히 무선 대역폭이 보장된다면 원격 미디어 서버가 오버로드이거나 프록시 캐시에서 혼잡이 발생할 때 클라이언트는 자신이 요청한 객체 세그먼트의 상태를 파악하여 캐시의 혼잡상태를 예방할 수 있게 된다. 따라서 캐시에서 혼잡이 발생할 때 비교 기법은 혼잡 제어 성능이 약간 낮게 나타났으며, 제안된 기법은 효율적인 캐싱 제어로 인하여 좋은 성능이 유지됨을 알 수 있다.

4-2-2 이득 중요도에 따른 캐싱 제어

두 번째 성능평가는 이득 중요도에 따른 캐시 제어율이다. 이득 중요도가 스트리밍 품질에 어떠한 영향을 미치는지에 대해서 알아보기 위하여 비교 기법에 대하여 5회 실험을 반복하였다. 그리고 이들에 대한 평균 캐시 제어율을 측정하였다. 이에 대한 시뮬레이션 결과는 그림 2와 같다. 그림 2에서 보듯이 이득 중요도가 0.7 이하일 때는 버퍼 캐시 히트율과 캐싱 제어에 비교적 큰 영향을 미치지 않음을 알 수 있다.

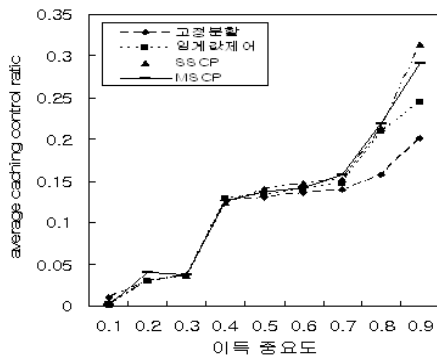


그림 2. 이득 중요도에 따른 캐싱제어율

Fig. 2. Caching Control Rate with Profit Popularity.

또한 그림 2에서 보듯이 이득 중요도가 증가할 때 SSCP가 MSCP에 비해 캐시 제어율이 보다 높게 나타남을 알 수 있는데 이것은 중요도가 낮은 순으로 객체 세그먼트를 보다 적응적으로 제어할 수 있기 때문이다. 결과적으로 캐싱 제어는 이득 중요도에 영향을 받는다는 것을 알 수 있으며, 이득 중요도가 낮을수록 스트림 패킷 데이터들에 대한 참조가 분산되기 때문에 캐시 제어율 또한 낮아지게 된다. 따라서 제안된 기법은 각 객체 세그먼트들에 대한 이득 중요도가 반영되었기 때문에 스트림 데이터 패킷 수가 증가하거나 크기가 증가할 때도 다른 기법들에 비해서 캐시 제어 성능이 높게 유지될 것으로 기대된다.

4-2-3 지연율에 따른 처리율

세 번째 성능평가는 지연율을 점차 증가시켜 가면서 평균 캐시 성능을 평가하였다. 지연율에 따른 캐시 처리 시뮬레이션 결과는 그림 3과 같다.

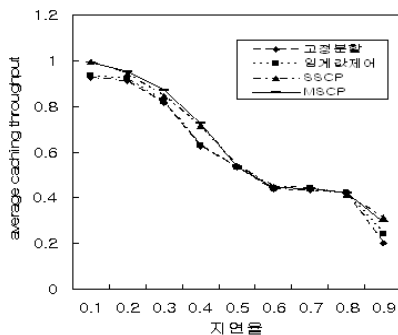


그림 3. 지연율에 따른 캐싱처리율

Fig. 3. Caching Throughput with Delayed Rate.

그림 3에서 보듯이 SSCP와 MSCP는 비교 기법에 비해서 캐싱 처리율이 약 6%정도 향상됨을 알 수 있으며, 지연율이 증가할 때 캐시 처리율의 성능은 매우 낮음을 알 수 있다. 특히 지연율이 0.3이상일 때 처리율이 매우 비효율적이며, 이러한 현상은 캐시 용량과 이득 중요도에 영향을 받기 때문이다. 즉 이득 중요도가 낮고, 캐시 용량이 작으면 스트림 세그먼트들에 대한 참조가 분산되기 때문에 지연으로 인한 패킷 손실이 크게 발생하고, 전체 캐시 처리율이 떨어지게 된다. 따라서 제안된 기법은 각 캐싱 교체 세그먼트들에 대해서 이득 중요도와 지연 절약을 반영하여 미디어 객체 세그먼트 크기가 증가할 때 SSCP와 MSCP 알고리즘으로 인한 캐시 처리율의 성능이 향상되게 된다.

V. 결 론

무선 채널 상에서 스트리밍 미디어 서비스는 높은 채널 비트 오류율, 채널 페이딩, 그리고 핸드오프 등으로 인하여 QoS 성능이 떨어지는 문제가 발생하고 있다. 본 논문에서는 이러한 문제를 해결하기 위하여 이득 중요도 기반의 새로운 세그먼트 캐싱 제어 기법을 제안하였다. 제안된 기법은 세그먼트 s에 대한 참조율을 반영하여 세그먼트 캐싱 이득을 측정하였으며, 순차적 객체 세그먼트 캐싱과 다중 객체 세그먼트 캐싱이 가능하도록 SSCP와 MSCP 알고리즘을 제안하였다. 제안된 기법에서 SSCP는 미디어 객체의 다른 세그먼트들이 캐싱되지 않았을 때 객체 세그먼트를 효율적으로 캐싱하는 장점이 있으며, MSCP는 크기가 큰 분할되지 않은 미디어 객체를 효율적으로 캐싱하는 장점이 있다. 따라서 제안된 기법은 프록시 캐시에 혼잡이 발생하고, 지터로 인한 지연이 발생할 때 시뮬레이션을 통하여 캐싱 구조를 적응적으로 제어함을 알 수 있었다.

참 고 문 헌

[1] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and

- Unicasy Real-time Video Streaming over Wireless LANs," in *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 6, pp. 524-534, 2002.
- [2] S. Chen, B. Shen, S. Wee, and X. Zhang, "Segment-based Streaming Media Proxy: Modeling and Optimization," *IEEE Transaction on Multimedia*, Vol. 8, No. 2, pp. 243-256, 2006.
- [3] C. D. Lee, T. W. Jeong, and J. Y. Ahn, "TRM-based Multimedia Streaming Optimization Scheme in Wireless Networks," *International Journal of Innovative Computing, Information and Control*, Volume 7, No. 4, pp. 1199-1209, 2011.
- [4] 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.
- [5] C. D. Lee, "Object version Transcoding for streaming media service in Wireless Mobile Networks," *KONI*, vol. 15, no. 3, pp. 355-363. 2011.
- [6] C. D. Lee, "Multi-level streaming using Fuzzy Similarity in P2P Distribution Mobile Networks," *KONI*, vol. 15, no. 3, pp. 364-371. 2011.
- [7] 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.
- [8] 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.
- [9] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing* 5, pp. 193-204, 2002.
- [10] P. Zhu, W. Zeng and C. Li, "Cross-layer design of source rate control and congestion control for wireless video streaming," *Advances in Multimedia*, Volume 2007, Issue 1, pp. 1-13, 2007.
- [11] C. D. Lee, T. W. Jeong, K. H. Choi, and J. Y. Ahn, "A Proxy Caching Scheme based on partition and mapping of media block segments," *International Journal of Innovative Computing, Information and Control*, Volume 8, No. 2, pp. 1493-1500, 2012.
- [12] 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.
- [13] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "Promise: Peer-to-Peer Media Streaming using Collectcast," in *ACM International Conference on Multimedia*, Berkeley, California, USA, pp. 1-10, 2003.

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