Journal of the Korea Academia-Industrial cooperation Society
Vol. 12, No. 6 pp. 2785-2792, 2011

DOI : 10.5762/KAIS.2011.12.6.2785

프록시 시스템에서 차별화된 스트리밍 미디어 서비스를 위한 퍼지 적합도 기반 트랜스 코딩

이종득[1*]
[1]전북대학교 전자공학부

Fuzzy Relevance-based Transcoding for Differentiated Streaming Media Service in the Proxy System

Chong-Deuk Lee[1*]
[1]**Division of Electronic Engineering, Chonbuk National University**

요 약 프록시 시스템에서 지연, 혼잡, 간섭 등의 문제는 QoS(Quality of Service)를 떨어뜨릴 뿐만 아니라 또한 스트리밍 미디어 서비스의 응답성과 신뢰성을 떨어뜨린다. 본 논문에서는 이러한 문제를 해결하기 위하여 새로운 퍼지 적합도 기반의 트랜스코딩 프록시 FRTP(Fuzzy Relevance-based Transcoding Proxy) 메커니즘을 제안한다. 제안된 FRTP메커니즘은 미디어 객체의 분할된 세그먼트 버전에 대한 퍼지 적합도를 분석하며, 분석된 퍼지 적합도에 따라 퍼지 적합도 트랜스코딩 그래프 FRTG(Fuzzy Relevance-based Transcoding Graph)를 생성한다. FRTG는 분할된 미디어 객체 세그먼트 버전들에 대한 트랜스코딩을 결정하게 되며, 결정된 트랜스코딩은 세그먼트 버전들 사이의 지연절약율 DSR(Delay Saving Ratios), 캐시 히트 정확률 CHPR(Cache Hit Precision Ratio), 그리고 캐시 히트 재현율 CHRR(Cache Hit Recall Ratio)을 향상시키게 된다. 제안된 기법의 성능을 알아보기 위하여 DSR, CHPR, 그리고 CHRR의 성능을 평가하였으며, 그 결과 제안된 기법이 비교 기법들에 비해서 DSR, CHPR, 그리고 CHRR의 성능이 향상됨을 알게 되었다.

Abstract Such problems as delay, congestion, and crosstalk in the proxy system degrade not only QoS (Quality of Service) but responsiveness and reliability of the streaming media service. To solve this problem this paper proposed a FRTP (Fuzzy Relevance-based Transcoding Proxy) mechanism. The proposed FRTP mechanism analyzes fuzzy similarity for partitioned segment versions of media objects to create a FRTG (Fuzzy Relevance-based Transcoding Graph). Created FRTG determines the transcoding for partitioned media object segment versions. Determined transcoding improves DSR (Delay Saving Ratios), CHPR (Cache Hit Precision Ratio), and CHRR (Cache Hit Recall Ratio). The proposed mechanism is simulated to evaluate such performance parameters as DSR, CHPR, and CHRR. Simulation results shows that the proposed mechanism outperforms in DSR, CHPR and CHRR compared with the other existing mechanisms.

Key Words : Mobile Streaming, Fuzzy Relevance, Transcoding, Object Segments Industrial

1. 서론

프록시 캐싱 기법은 효율적인 미디어 콘텐트 전송을 위한 기법으로서 미디어 서버와 프록시 간 또는 프록시와 모바일 클라이언트 간의 트래픽을 줄이고, 사용자가 요청한 대기시간을 줄이기 위해 사용되고 있다[1-4].

이 기법은 프록시에서 인기도 또는 사용빈도가 높은 객체들을 우선적으로 캐싱하여 콘텐트들을 전송한다. 최근에는 프록시상에서 트랜스코딩을 위한 여러 캐싱 알고리즘이 제안되고 있다.

트랜스코딩이란 코딩된 신호를 다른 신호로 바꾸는 과정을 말하며 멀티미디어의 경우 압축된 동영상을 다른

*교신저자 : 이종득(cdlee1008@jbnu.ac.kr)
접수일 11년 04월 08일 수정일 11년 05월 06일 게재확정일 11년 06월 09일

규격의 동영상으로 변환하거나 비트율, 공간 해상도, 프레임율, 오류 탄력성 등을 조절하는 기술을 의미한다[2]. 프록시 시스템에서 트랜스코딩은 서버 기반 트랜스코딩, 클라이언트 기반 트랜스코딩, 그리고 프록시 기반 트랜스코딩으로 분류된다[4].

서버 기반의 트랜스코딩에서 웹 객체들의 다중 버전 트랜스코딩은 오프라인 상에서 수행되며 이때 트랜스코드된 버전들은 서버 디스크에 저장된다. 서버 기반 트랜스코딩 기법의 장점은 클라이언트가 서버에 데이터를 요청 할 때 트랜스코딩으로 인한 지연을 줄일 수 있다는 점이다. 그러나 이 기법의 단점은 서버에서 객체의 여러 버전들에 대한 저장 공간을 개별적으로 유지하기 때문에 캐시저장 공간의 낭비가 크며, 이로 인하여 클라이언트의 요구사항을 제때에 반영하지 못한다는 점이다.

클라이언트 기반의 트랜스코딩은 모바일 클라이언트의 기능, 특성 등을 고려하여 트랜스코드를 수행한다. 클라이언트 기반의 트랜스코딩 기법의 장점은 원본 시멘틱 구조를 보존하여 프로토콜을 적응적으로 수행한다는 점이다. 그러나 클라이언트 측에서의 모바일 기기의 제한된 대역폭과 제한된 전력 에너지는 스트리밍 미디어 서비스의 장애가 되고 있다.

프록시 기반의 트랜스코딩은 서버와 클라이언트의 미들 레벨로서 프록시는 요청한 객체를 즉시 트랜스코딩하는 기능을 수행하며, 요청한 객체를 클라이언트에게 전송하기 전에 클라이언트가 요청한 요구사항을 분석하여 요청된 객체를 적절한 버전으로 트랜스코딩하는 기능을 수행한다. 전통적인 트랜스코딩 프록시[5-6]은 웹 프록시의 기능뿐만 아니라 또한 캐시 기능을 수행하고 있다. 그러나 전통적인 웹 프록시는 스트리밍 미디어 객체들을 스트리밍할 때 지터 지연, 혼잡, 간섭 등으로 인하여 QoS가 떨어지는 문제점이 발생하고 있다[7-8]. 이러한 문제를 해결하기 위하여 [9]는 coverage-기반 알고리즘과 demand-기반 알고리즘을 제안하였다. coverage-기반 알고리즘은 객체의 원본 버전을 캐시하기 위한 기법이며, 이에 반해 demand-기반 알고리즘은 트랜스코드된 버전을 캐시하기 위한 기법이다. 그러나 이들 기법들은 미디어 객체의 크기와 프록시 캐시 용량, 미디어 객체의 적합도, 미디어 객체의 세그먼트 버전 및 크기, 비트율, 프레임율 등을 반영하지 않아 혼잡과 지연의 문제점이 발생하고 있다.

본 논문에서는 이러한 문제점을 해결하기 위하여 퍼지 적합도 기반의 새로운 트랜스코딩 프록시 메카니즘 FRTP (Fuzzy Relevance-based Transcoding Proxy)을 제안한다. 제안된 FRTP 메카니즘은 스트리밍 미디어 객체의 분할된 세그먼트 버전들에 대하여 퍼지 적합도를 수행하며, 퍼지 적합도에 따라 퍼지 적합성 트랜스코딩 그래프 FRTG (Fuzzy Relevance-based Transcoding Graph)를 구성한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 살펴보며, 3장에서는 차별화된 스트리밍 미디어 서비스를 위한 FRTP 메카니즘에 대해서 살펴본다. 그리고 4장에서는 제안된 메카니즘에 대한 시뮬레이션 결과에 대해서 살펴보며, 끝으로 결론에 대해서 살펴본다.

2. 관련연구

최근에 대부분의 트랜스코딩 프록시 구조는 유사한 네트워크 링크 및 디바이스 프로파일에 일치되는 모바일 클라이언트 중심으로 진화되고 있다. 전통적인 미디어 프록시 기법들은 미디어 객체의 각 세그먼트 버전 분할 및 통합을 수행하지 않고 있다[9-11]. 이러한 문제를 해결하기 위하여 Chang과 Chen[1]은 캐싱이 수행될 때 객체의 각기 다른 버전들을 분할하는 기법을 제안하였다. 그러나 이 기법은 웹 객체들의 분할에는 효율적이지만 미디어 객체의 경우에는 크기 속성 제약이 발생하고 있다.

Miao[5]는 상영 품질(playback quality)을 향상시키기 위하여 선택적 캐싱(selective caching)기법을 제안하였다. 이 기법은 프레임들이 전송율에 따라 캐시될 수 있도록 한 “중간 프레임(intermediate frames)" 선택 기법이다. 그러나 이 기법에서 프록시는 시작 세그먼트(initial segment) 이후의 프레임들은 직접 캐시하지 못하는 문제가 있다.

Li[9]는 유연한 상영(playback)을 위해 동적 조절 비디오 버퍼 기법과 전송 스케줄링 기법을 제안하였다. 일반적으로 비디오 참조는 사용빈도가 높은 프레임과 사용빈도가 낮은 프레임으로 구분된다. 이와 같은 시나리오는 주로 라이브 뉴스(live news)와 같은 인터넷 비디오에서 나타나고 있다.

그리고 Ma[12]는 트랜스 코딩 프록시를 위해 계층적 캐싱 구조를 제안하였다. 이 기법에서 트랜스코딩 객체들은 계층적 경로 상에 분산되도록 하였다. 이 기법은 트랜스코딩 프록시의 로드를 줄일 수 있지만 네트워크 지연(latency)으로 인하여 트랜스코딩 비용이 높다는 문제점을 가지고 있다.

Shen[13]은 TEC (Transcoding-Enabled Caching)를 위한 3가지 캐싱 전략을 제안하였다.

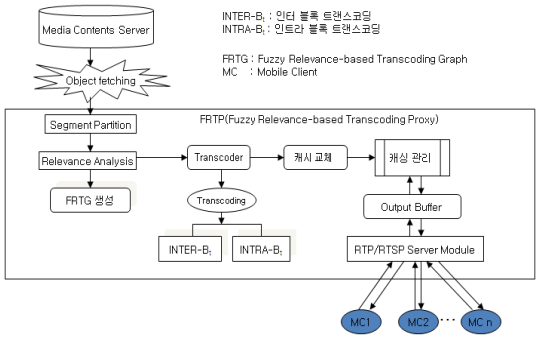
이 기법은 시스템 상에는 동일한 비디오의 변종(variants)들이 존재할 수 있다고 가정하였다. 이 기법에서 처음 두 개의 알고리즘은 비디오 객체 중 하나를 선택

하여 캐시하는 기법으로서 사용자가 비디오 버전을 요청할 때 비트율을 다르게 하여 처리하는 기법이다. 세 번째 알고리즘은 트랜스코드의 처리 부담(processing load)을 줄이기 위한 기법으로서 동일한 비디오 객체를 멀티 버전으로 캐시하는 기법이다. 그러나 이들 기법들은 세그먼트 버전들 사이의 트랜스코딩을 고려하지 않았기 때문에 DSR, CHPR, CHRR의 성능이 떨어지는 문제가 발생하고 있다.

따라서 본 논문에서는 이와 같은 문제들을 해결하기 위하여 FRTP를 제안한다. 제안된 메카니즘은 인코딩과 디코딩의 전송율 불일치 및 불균형, 그리고 언더플로우와 오버플로우로 인해 발생되는 지터 지연, 혼잡, 그리고 간섭 등의 오버헤드를 줄이는 효과를 제공하게 된다.

3. FRTP 메커니즘

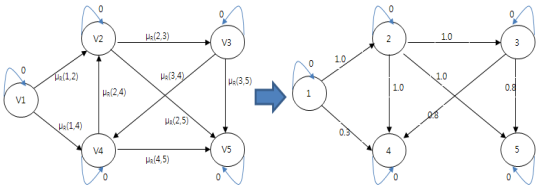
제안된 FRTP는 미디어 콘텐츠 서버와 MC (Mobile Client)들 간의 서로 다른 이종 네트워크를 상호 연결해 주는 미들웨어 서버이다. 그림 1은 본 논문에서 제안된 FRTP 구조이며, 적합도 분석, 트랜스코딩, 캐시교체, 캐시 관리 과정을 거쳐 스트리밍을 수행한다.



[그림 1] FRTP 시스템 구조
[Fig. 1] System structure of FRTP

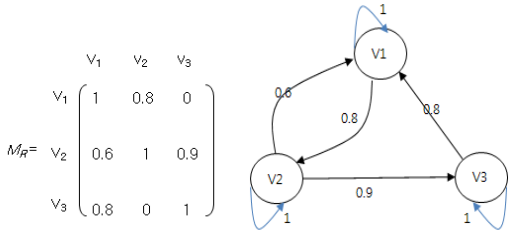
3.1 퍼지 적합도

퍼지 적합도 FR(Fuzzy Relevance)은 [0, 1]사이의 소속함수 μ로 정의되며, FR에서 각 아이템은 μ의 소속값을 가지게 된다. FR을 수행한 후 트랜스코딩을 보다 명세화하기 위하여 퍼지 적합도 그래프 FRG (Fuzzy Relevance Graph)를 생성한다. FRG G_i는 [0,1]사이의 소속함수 μ를 가진 방향성그래프로서 그림 2와 같다.



[그림 2] FRG
[Fig. 2] FRG

여기서 G_i는 트랜스코드가 가능한 객체 세그먼트i의 버전들 간의 퍼지 적합도 관계를 나타낸다. 각 정점(vertex) $v \in V[G_i]$에 대해서 v는 트랜스코드가 가능한 객체i의 버전이다. 만일 방향성 에지 $(u,v) \in E[G_i]$가 존재하면 객체 세그먼트i의 버전u는 버전 v로 트랜스코드 된다. 버전u에서 버전v 까지의 트랜스코딩 비용은 $\mu_R: u \times v \rightarrow [0,1]$로 나타내며, 이것은 버전$u$에서 버전 v로의 퍼지 적합도이다. 즉 FRG는 두 정점 u, v가 정점 l을 경유할 때의 에지의 퍼지 적합도 정도를 $\mu_R(u,l)$=0.8, $\mu_R(l,v)$=0.4로 표현한다. 따라서 l을 경유한 퍼지 적합도 관계는 $\mu_R(u,l) \wedge \mu_R(l,v)$=0.8 $\wedge$ 0.4=0.4가 된다. 이처럼 u에서 v를 경유하는 직, 간접의 모든 경로에 대한 퍼지 적합도를 구하여 이들 가운데서 최대값을 u에서 v로의 퍼지 적합도 정도라 한다. 예를 들어 미디어 블록 객체 o_1, o_2에는 각각 분할된 객체 세그먼트 버전이 o_1=o_2={v1, v2, v3}와 퍼지관계 R이 다음과 같은 퍼지 행렬 M_R로 주어져 있다고 가정하면 퍼지 행렬 M_R의 FRG는 그림 3과 같이 표현된다.



[그림 3] M_R과 FRG
[Fig. 3] M_R and FRG

3.2 미디어 객체 트랜스코딩

일반적으로 원본 미디어 객체 블록은 완전한 미디어 콘텐트 정보를 가지고 있으며, 원본 미디어 객체 블록은 여러 객체 세그먼트들로 분할되어 트랜스코딩 된다. 그러나 트랜스코딩과정에서 프록시에 요청한 스트리밍 미디

어 데이터에 트랜스코딩의 불확실성이 발생하면 지연, 혼잡, 간섭 등의 문제가 발생한다[2-4]. 본 논문에서는 이러한 문제를 해결하기 위하여 프록시로부터 패치된 각 미디어 객체를 n개의 세그먼트 버전 V로 사상한다. 이때 객체의 원본 버전을 V_1라 하고, 더 이상 트랜스코드 되지 않는 버전을 V_n이라 가정한다. 그리고 V_2, ..., V_{n-1}을 중간 버전들이라 가정한다. 만일 $i < j$이면 각 버전i, j에 대해서 V_i는 V_j보다 더 상세한 버전이라 하며, 원본 버전은 가장 상세한 버전이며, 트랜스코드된 버전일수록 덜 상세한 버전이 된다. 그러나 원본버전이라 하더라도 V_i가 V_j에 대한 트랜스코딩 정보 즉 퍼지 적합도 정보를 가지지 않으면 V_i는 V_j로 트랜스코드 되지 않는다. 본 논문에서는 트랜스코딩을 위해 퍼지 적합도가 큰 객체 세그먼트 버전들을 순차적으로 선택하며, 이를 위해 인터블록 트랜스코딩 $INTER-B_t$와 인트라블록 트랜스코딩 $INTRA-B_t$로 구분하여 트랜스코딩을 수행한다.

3.2.1 $INTER-B_t$ 트랜스코딩

$INTER-B_t$는 미디어 블록들 간의 적합도 관계이다. $INTER-B_t$는 블록들의 크기와 버스트 동기화에 의해 결정되며 블록들의 크기는 캐시 용량을 초과해서는 안 된다. $INTER-B_t$는 미디어 객체 세그먼트 분할로 인하여 캐시 교체가 빈번하게 발생하는 문제를 줄이기 위한 정책이며, 미디어 블록 A에 대한 퍼지 적합도는 다음과 같이 정의된다.

(정의1) A_α={$x|\mu_A(x) \geq \alpha$, $x \in X$, $\alpha \in [0,1]$}이다.

여기서 만일 $a \in R$이면, $\mu_a(x)=\begin{cases}1 & x=a \\ 0 & x \neq a\end{cases}$로 표현된다. 우리는 퍼지수가 정규이고, 적합도가 매우 큰 R상의 퍼지부분집합을 $\exists_x \leftarrow R$, $\bigcup_x \mu_A(x)$=1로 표현한다. 이것은 R상의 퍼지집합의 소속 값(membership value)의 최대 값이 1임을 의미하며, 이는 적합도가 매우 높음을 의미한다. 적합도는 A_α=[$a_1^{(\alpha)}$, $a_2^{(\alpha)}$]의 성질을 만족해야 하며, (정의2)와 같은 포함관계를 가진다.

(정의2) $(\alpha' < \alpha) \Rightarrow (a_1^{(\alpha')} \leq a_1^{(\alpha)}, a_2^{(\alpha')} \geq a_2^{(\alpha)})$이면, $(\alpha' < \alpha) \Rightarrow (A\alpha \subset A\alpha')$이다.

3.2.2 $INTRA-B_t$ 트랜스코딩

$INTRA-B_t$는 미디어 블록 객체가 프록시 캐시 용량을 초과할 때 미디어 블록 객체를 분할하여 스트리밍 미디어 캐시를 최적화하기 위한 전략이다. 이를 위하여 본 논문에서는 $INTRA-B_t$의 객체 세그먼트들에 대해서 퍼지 적합도 관계 FRR(Fuzzy Similarity Relation)을 수행한다. FRR이 수행된 객체 세그먼트들은 우선순위가 부여되며, 다음과 같이 정의된다.

(정의3) X상의 퍼지 유사도 관계 FRR은 다음과 같은 성질을 만족하는 퍼지관계이다.

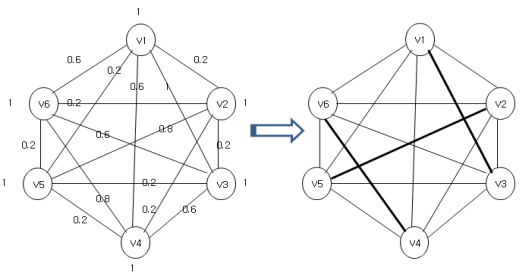
(1) 반사 관계(Reflexive Relation) : $\mu_R(x,x)$=1, $\forall_x \in X$

(2) 대칭 관계(Symmetric Relation) : $\mu_R(x,y)$= $\mu_R(y,x)$

(3) 전이 관계(Transitive Relation) : $\mu_R(x,z)$=$\bigcup_y$ $\{\mu_R(x,y) \cap \mu_R(y,z)\}$

예를 들어 R상에서 그림 4와 같은 퍼지 행렬 M_R을 가정하자.

$$M_R = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{pmatrix} 1 & 0.2 & 1 & 0.6 & 0.2 & 0.6 \\ 0.2 & 1 & 0.2 & 0.2 & 0.8 & 0.2 \\ 1 & 0.2 & 1 & 0.6 & 0.2 & 0.6 \\ 0.6 & 0.2 & 0.6 & 1 & 0.2 & 0.8 \\ 0.2 & 0.8 & 0.2 & 0.2 & 1 & 0.2 \\ 0.6 & 0.2 & 0.6 & 0.8 & 0.2 & 1 \end{pmatrix} \end{array}$$



[그림 4] 퍼지 유사도 관계 M_R
[Fig. 4] Fuzzy similarity relation M_R

그림 4에서 보듯이 $\alpha \geq 0.2$일 때는 $\{v1,v2,v3,v4,v5,v6\}$로의 적합도 관계가 결정되며, 이때는 트랜스코딩 관계가 매우 낮음을 의미한다. 그러나 만일 $\alpha \geq 0.8$이면, 그림 5에서처럼 적합도 관계는 $\{v_1,v_3\}$, $\{v_2, v_5\}$, 그리고 $\{v_4, v_6\}$의 순서로 결정되게 된다. 따라서 알파-레벨이 $\alpha \geq 0.6$, $\alpha \geq 0.8$, $\alpha \geq 1.0$으로 증가할 때는 트랜스코딩 관계가 보다 명료해지게 된다.

3.3 캐시교체

본 논문에서 수행되는 캐시교체는 캐시 용량, 객체의 크기, 그리고 FRR을 고려하여 수행하며, 제약조건 C가 $c \in C$ 일 때 교체 사상은 $disjunction(\cup)$사상, $conjunction(\cap)$사상, 그리고 필터링 사상으로 구분된다.

3.3.1 $disjunction(\cup)$사상

$disjunction(\cup)$사상은 캐시용량, 객체의 크기를 모두 고려한 사상으로서 $Max-Min$ 관계를 가진 객체들을 교체하기 위한 정책이다. 캐시 용량을 고려한 교체 사상은 $M_{disjunct}^{C,C}$ 으로 표현하고, 객체의 크기를 고려한 교체 사상은 $M_{disjunct}^{s}$,으로 표현한다.

따라서 $disjunction(\cup)$에 의한 사상은 다음과 같이 정의된다.

(정의4) $M_{disjunct}^{(C,C)\cup s}$: $\forall c,c' \in C$이고 c는 $c \cup c'$이다.

여기서 C,C는 Cache Capacity이며, s는 객체의 크기이다.

3.3.2 $conjunction(\cap)$사상

$conjunction(\cap)$사상은 캐시용량, 객체의 크기, 그리고 FRR을 고려한 사상으로서 $Min-Max$ 관계를 가진 객체들을 교체하기 위한 정책이다. 캐시 용량을 고려한 교체 사상은 $M_{conjunction}^{C,C}$, 객체의 크기를 고려한 교체 사상은 $M_{conjunction}^{s}$, 그리고 FRR을 고려한 교체 사상은 $M_{conjunction}^{FRR}$ 으로 표현한다.

따라서 $conjunction(\cap)$에 의한 사상은 다음과 같이 정의된다.

(정의5) $M_{conjunction}^{(C,C)\cap s\cap FSR}$: $\forall c,c' \in C$이고 c는 $c \cap c'$이다.

여기서 $\forall c,c' \in C$, $\varnothing$은 c의 $null$이다.

3.3.3 $filtering$ 사상

$filtering$ 사상은 $disjunction(\cup)$사상과 $conjunction(\cap)$사상을 만족하지 않는 객체들을 제거하기 위한 정책이다. $filtering$ 사상은 $disjunction(\cup)$사상과 $conjunction(\cap)$사상을 통해서 수행되며, $filtering$을 위한 사상은 $f_i \in F$일 때 $M_{filtering}^{FRR}$로 표현한다. 따라서 $filtering$을 위한 교체 사상 $M_{filtering}^{FRR}$은 다음과 같이 정의된다.

(정의6) $M_{filtering}^{FRR}$={ $M_{filtering}^{FRR}$(O) | $Min\{\alpha \leq \mu(B_i)$ and $\mu(S_j) \geq \alpha\}$ 또는 $M_{filtering}^{FRR}$={ $M_{filtering}^{FRR}$(O) | FRR<0.6}이다.

예를 들어 아래 표 1에서 미디어 블록 CB_1, CB_2, CB_3의 각각에 대해서 $M_{filtering}^{0.1\cup 0.4}(o)$을 수행하면 CB_1={o_1, o_3}, CB_2={o_2,o_4}, CB_3={o_3,o_4}의 객체가 필터링 된다. 결과적으로 이들 객체는 객체의 크기가 너무 작게 분할된 객체이거나 또는 너무 크게 분할된 객체들로서, 이들 객체들이 캐시 블록 CB에서 필터링 된다.

[표 1] 캐시 참조 테이블
[Table 1] Cache reference table

CBo	o_{11}	o_{12}	o_{13}	o_{14}
CB_1	0.3	0.7	0.2	
CB_2	0.6	0.1	0.8	0.4
CB_3	0.5	0.9	0.3	0.2

4. 시뮬레이션 분석

이 장에서는 이벤트-지향 시뮬레이션(event-driven simulation)을 통하여 제안된 FRTP 메카니즘의 성능을 살펴본다. 성능분석을 위해 4.1절에서는 시뮬레이션 모델에 대해서 살펴보며, 그리고 4.2절에서는 시뮬레이션 결과들에 대해서 살펴본다.

4.1 시뮬레이션 모델

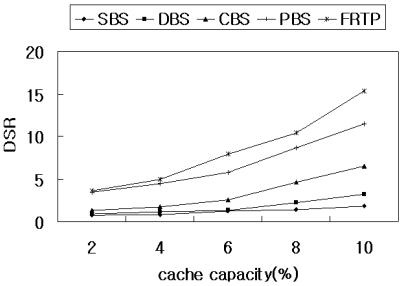
시뮬레이션을 위해 전체 미디어 객체들의 수는 N으로 하였으며, 표준 비디오 시퀀스는 각각 350개, 450개, 500개의 프레임을 소스로 이용하였다. 본 논문에서는 시뮬레이션 편의를 위하여 미디어 객체의 크기를 최대 5Mbyte 이내로 제한하였으며, 이들 객체의 분포는 Pareto 분포에 기반을 두었다. 비트율은 최대 1.28Mbps, 패킷크기는 512kb, 링크 대역폭은 10/100Mbps, 그리고 평균 링크 대역폭은 약 1.2Mbps이내로 제한하였다. 전체 시뮬레이션 시간은 560s 동안 수행하였으며, 스트림의 t_s는 [1, 20s], μ는 $\mu \geq 0.7$, α는 $0<\alpha<1$로 세트시켰다. 그리고 모바일 클라이언트들은 무선 IP를 통하여 IP백본 망에 연결되어 있다고 가정하였다. 연속적인 스트림 데이터는 10개 이내로 제한하였으며, 최대 PSNR(Peak Signal to Noise Ratio)은 30dB이내로 제한하였다. 인코딩율과 디코딩율에 따른 트랜스코딩율은 초당 100Kbyte 이내로 제한하였으며, 인코더 버퍼크기와 디코더 버퍼크기는 모두 550Kbyte 이내로 제한하였다. 미디어 객체들의 분포 측정은 Zipf-like 분포에 기반을 두었으며, 분포는 δ=0.7로 설정하였다. 기본적인 캐시 용량은 0.05*($\sum$객체크기)라고 가정하였으며, 서버로부터 미디어 객체들을 패치하는 데 걸리는 지연은 지수분포를 따른다고 가정하였다. 그리고 원본 서버로부터 미디어 객체를 트랜스코딩하는 데 걸리는 지연은 무시하였다.

4.2 시뮬레이션 결과

성능 평가를 위해서 우리는 캐시용량, 미디어 객체 수, 미디어 객체의 크기 그리고 α값을 변화시켜 가면서 시뮬레이션을 수행하였다. 시뮬레이션에서 사용된 주요 성능 척도는 DSR, CHPR, 그리고 CHRR이다. CHPR이란 적합도를 만족하는 미디어 객체들이 프록시 캐시에 얼마나 많이 히트되었는지의 비율이며, CHRR이란 적합도를 만족하는 미디어 객체들 중에서 적합도가 가장 큰 미디어 객체들이 얼마나 많이 히트되었는가의 비율이다. 본 논문에서는 제안된 기법의 성능을 위해 단순 서버 기반 기법인 SBS(Simple-Based Server), CBS(Coverage-Based Server), DBS(Demand-Based Server), 그리고 PBS(Profit-Based Server)기법과의 성능을 비교분석하였다.

4.2.1 캐시 용량 및 FR에 따른 DSR분석

첫 번째 시뮬레이션에서는 캐시 용량과 FR 즉 α값을 변화시켜가면서 제안된 캐시 교체 알고리즘의 성능을 분석하였다. 이에 대한 분석 결과는 그림 5와 같다.



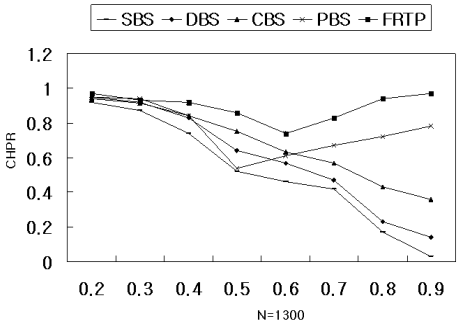
[그림 5] 캐시용량과 FR에 의한 DSR
[Fig 5] DSR by cache capacity and FR

그림 5에서 보듯이 제안된 기법은 지연 절약율(DSR) 관점에서 볼 때 다른 기법들에 비해서 성능이 우수함을 알 수 있다. 특히 SBS기법과 DBS기법을 비교해 볼 때 평균 성능이 각각 14%, 12% 정도가 향상되었다. 그리고 비교적 성능이 우수한 PBS기법과 성능을 비교해 볼 때 약 5%가 향상되었음을 알 수 있다. SBS기법과 DBS기법의 성능이 좋지 못한 이유는 이 알고리즘들은 객체들의 크기 및 캐싱 용량에 따른 적합도를 고려하지 않았기 때문이다. 따라서 멀티미디어 객체는 웹 객체와는 달리 트랜스코딩 프록시 환경에서 지연 문제가 더 크게 발생된다.

그러나 제안된 FRTP기법은 적합도가 반영되었기 때문에 트랜스코딩 프록시 환경에서 크기가 큰 미디어 객체가 새로 입력되거나 트랜스코딩이 수행될 때 지연 절약을 효율적으로 수행하게 된다.

4.2.2 미디어 객체 수에 따른 CHPR과 CHRR분석

두 번째 시뮬레이션에서는 미디어 객체 수가 N=1,300일 때 α값을 변화시키면서 CHPR과 CHRR의 성능을 평가하였다. 여기서N=1,300으로 설정한 것은 비디오 시퀀스를 프레임 소스로 사용하였기 때문이다. 미디어 객체 수 및 FR에 따른 CHPR과 CHRR의 시뮬레이션 결과는 그림 6과 같다.

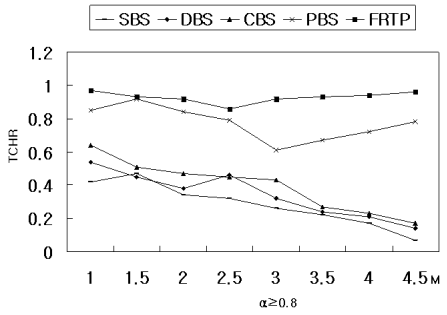


[그림 6] 미디어 객체 수에 의한 CHPR과 CHRR
[Fig 6] CHPR and CHRR by media object number

그림 6에서 보듯이 α값이 증가하면 제안된 FRTP의 CHPR과 CHRR의 성능이 증가하게 된다. 따라서 객체의 적합도가 크면 클수록 제안된 FRTP기법의 CHPR과 CHRR의 성능 또한 향상되게 된다. 그리고 지연 절약율 DSR도 증가하게 된다. 그러나 α값이 작으면 성능이 떨어짐을 알 수 있다. 결과적으로 CHPR과 CHRR의 성능은 α값에 영향을 받게된다. α값이 작으면 객체들에 대한 참조가 분산되기 때문이다. 따라서 제안된 FRTP의 기법은 각 객체에 대한 적합도가 고려되었기 때문에 다른 기법들에 비해서 성능이 향상되게 된다.

4.2.3 미디어 객체 크기에 따른 캐시히트율과 DSR분석

세 번째 시뮬레이션에서는 미디어 객체의 크기에 따른 전체 캐시히트율 TCHR(Total Cache Hit Ratio)와 DSR의 성능을 평가하였다. 미디어 객체의 크기가 크면 전체 캐시 히트율과 DSR에도 영향을 받는다. 이에 대한 시뮬레이션 결과는 그림 7과 같다.



[그림 7] 미디어 객체의 크기에 의한 전체 캐시히트율과 DSR

[Fig 7] Total cache hit rate and DSR by media object size

그림 7에서 보듯이 제안된 FRTP의 TCHR은 비교적 우수한 PBS 기법보다 성능이 향상되었음을 알 수 있다. 이것은 제안된 FRTP기법에서는 미디어 객체들의 적합도와 유사도 관계를 고려하였기 때문이다. 따라서 미디어 객체의 크기가 증가해도 TCHR과 DSR의 성능은 크게 영향을 받지 않는다. 그러나 미디어 객체의 크기 관점에서 볼 때 α값은 TCHR과 CHPR, CHRR 에 영향을 미친다. 만일 α값이 α<0.8일 때는 미디어 콘텐츠 서버로부터의 패치 지연이 크게 발생하기 때문에 TCHR과 DSR의 성능에 영향을 미치게 된다. 따라서 본 논문에서는 이러한 문제를 피하기 위하여 $\alpha \geq 0.8$로 하여 시뮬레이션을 수행하였다. 그 결과 미디어 객체의 크기가 작거나 캐시 용량을 초과하는 객체들에 대해서는 필터링이 수행되었다. 따라서 제안된 FRTP기법은 α값에 의해 반복적인 트랜스코딩 연산을 피할 수 있게 되며, 그 결과 트랜스코딩과 스트리밍 미디어 서비스를 안정적으로 수행하게 된다.

5. 결론

최근에 무선 모바일 네트워크상에서 낮은 네트워크 대역폭, 이동성 그리고 제한된 자원 등은 트랜스코딩 프록시 기법을 어렵게 하고 있다. 본 논문에서는 이러한 문제를 해결하기 위하여 새로운 퍼지 적합도 기반의 트랜스코딩 프록시 메카니즘 FRTP를 제안하였다. 제안된 FRTP 메카니즘은 미디어 객체의 분할된 객체 세그먼트 버전에 대한 퍼지 적합도 FR를 수행하여 트랜스코딩 여부를 결정하도록 하였다. 또한 효율적인 트랜스코딩을 위하여 트랜스코딩 그래프 FRTG를 구성하였다. 이처럼 구성된 FRTG는 분할된 미디어 객체 세그먼트 버전들에 대한 트랜스코딩을 결정하게 되며, 결정된 트랜스코딩은 세그먼트 버전들 사이의 DSR, CHPR, CHRR의 성능을 결정하게 된다. 그리고 본 논문에서는 트랜스코딩 프록시에 새로운 미디어 객체가 입력될 때 트랜스코딩의 불확실성과 모호성을 줄이기 위하여 $INTER-B_t$와 $INTRA-B_t$를 수행하였다. 제안된 기법의 성능 척도는 캐시용량, 미디어 객체 수, 미디어 객체의 크기 그리고 FR의 값 즉 α값을 변화시켜 가면서 시뮬레이션을 수행하였다. 그 결과 제안된 기법의 성능을 비교한 결과 SBS 기법과 DBS기법에 비해서 평균 성능이 각각 14%, 12% 정도가 향상되었으며, 비교적 성능이 우수한 PBS기법과 성능을 비교한 결과 약 5%의 성능이 향상되었음을 알 수 있었다.

References

[1] Cheng-Yue Chang and Ming-Syan Chen, "On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxies," IEEE TRANSACTIONS ON PARELLEL DISTRIBUTED SYSTEM, VOL. 14, NO. 6, pp. 611-624, 2003.

[2] Chi-Feng Kao and Chung-Nan Lee, "Aggregate Profit-Based Caching Replacement Algorithms for Streaming Media Transcoding Proxy Systems," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 9, NO. 2,

pp. 221-230, 2007.

[3] Songqing Chen, Bo Shen, Susie Wee, and Xiaodong Zhang, "Segment-Based Streaming Media Proxy: Modeling and Optimization," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 8, NO. 2, pp. 243-256, 2006.

[4] Kuei-Chung Chang, Tien-Fu Chen, "Efficient Segment-based Video Transcoding Proxy for Mobile Multimedia Services," Journal of Systems Architecture 53, pp. 833-845, 2007.

[5] Zhourong Miao, Antonio Ortega, "Proxy caching for Efficient Video Services over the Internet," in:Packet Video Workshop, pp. 1-21, 1999.

[6] Dapeng Wu, Yiewei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang and Jon M. Peha, "Sreaming Video over the Internet : Approachs and Directions," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 11, NO. 3, PP. 282-300, 2001.

[7] S. Acharya, H. Korth, and V. Poosala, "Systematic Multiresolution and its Application to the World Wide Web," IEEE ICDE'99, Sydney, Australia, pp. 40-49, 1999.

[8] 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.

[9] Danjue Li, Chen-Nee Chuah, Gene Cheung and S. J. Ben Yoo, "MUVIS : Multi-source Video Streaming Service over WLANs", Journal of Communication and Networks(JCN), vol.7, pp. 144-156, 2005.

[10] C. D. LEE, "Proxy Caching Grouping by Partition and Mapping for Distributed Multimedia Streaming Service," KIIS, Vol 19, No, 1, pp. 40-47, 2009.

[11] C. D LEE, T. W. JEONG, J. Y. AHN, "A Streaming Service based on Partition and Mapping of Media Block," ISME2009, The 6th International Symposium on Management Engineering, 5-7 August, Dalian, China, CD, pp. 1-6, 2009.

[12] W. Ma and D. H. C. Du ,"Design a Progressive Video Caching Policy for Video Proxy Server," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 6, NO. 4, pp. 599-610, 2004.

[13] Bo Shen, Shun-Ju Lee, Sujoy Basu, "Caching Stratigies in Transcoding-Enabled Proxy Systems for Streaming Media Distribution Networks," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 6, NO. 2, pp. 375-386, 2004.

이 종 득(Chong-Deuk Lee) **[정회원]**

- 1989년 2월 : 전북대학교 일반대학원 컴퓨터과학 (이학석사)
- 1998년 2월 : 전북대학교 일반대학원 컴퓨터과학 (이학박사)
- 1992년 3월 ~ 2002년 2월 : 서남대학교 정보통신공학과 교수
- 2002년 3월 ~ 현재 : 전북대학교 전자공학부 교수

<관심분야>
유비쿼터스 컴퓨팅, 무선 모바일, 무선 센서 네트워크 등