멀티미디어 스트리밍의 트랜스코딩과 캐싱의 협업을 위한 transco-prefix caching algorithm

설성운° 서대화
경북대학교 전자공학과
{sulsa21°}@palgong.knu.ac.kr, {dwseo}@ee.knu.ac.kr

Transco-prefix caching algorithm for cooperation between transcoding and caching

SeongWoon Seol° DaeWha Seo
Dept. of Electronics Engineering, Kyungpook National University

요 약

트랜스코딩에 관련된 연구들은 비디오 스트리밍에 대한 품질관련 연산들을 제공해 주었으나 트랜스코딩 과정에서 발생하는 연산지연과 전송지연에 대한 고려가 부족하였다. 또한 비디오 스트리밍에 대한 캐싱기법과 관련된 연구들은 비디오 스트리밍 서버와 클라이언트 사이의 전송지연을 줄여주는 장점을 제공하는 반면, 트랜스코딩과의 협업을 위한 연동들은 드물었다. 본 논문에서는 대역폭이 유선망에 비해 상대적으로 빈약한 무선망 채널을 통해 연속적인 비디오 스트리밍 서비스를 제공하는 과정 중에 발생하는 전송지연과 연산지연을 동시에 극복할 수 있도록 해주는 통합 시스템 아키텍처와 캐싱 알고리즘을 제안한다.

I. Introduction

VOD(video-on-demand)나 전자 도서관, 원격교육 등의 비디오 스트리밍 서비스에 대한 급격한 요구 증가와 맞물려서 최근 몇 년동안의 비디오 스트리밍에 관련된 연구가 증가하고 있다. 일반적인 비디오 스트리밍 시스템에서의 비디오 시퀀스는 먼저 high quality로 인코딩 된 후 서버에 저장된다. 사용자들은 무선망 네트워크등과 같은 채널들을 통해 스트리밍 서버에 접속하여 비디오 스트리밍 서비스를 받을 수 있다. 그러나 현재로서 이러한 무선망 네트워크에서의 가용한 대역폭들은 넉넉하지 못한 편이다. 따라서 원활한 비디오 스트리밍 서비스를 위해서는 높은 비트율의 비디오 스트림을 낮은 비트율로 바꾸어 주는 트랜스코더가 필요하다.

과거 몇 년동안 여러 가지 트랜스코딩 기법들이 소개되었다 .[1], [2], [3], [4]. 그러나, 위의 트랜스코딩 기법들은 트랜스코딩 연산 과정 중에 발생하는 연산지연의 특성들까지 충분히 고려되어 설계되지는 않았다. 특히 무선망에서의 비디오 스트리밍은 무선망의 제한된 대역폭으로 인해 지연에 더욱더 민감할 수밖에 없다.

한편 비디오 스트리밍 서비스 도중에 발생하는 지연을 줄이기 위한 여러 가지 비디오 스트리밍 캐싱 기법들이 소개되었다.[5][6][7]. 캐싱기법들은 스트리밍 서버와 모바일 호스트와의 전송 지연에 따른 비디오 스트리밍의 비연속성을 해결하는 데 도움을 준다. 그러나 비디오 품질과 관련된 연산을 수행하는 트랜스코딩 기법들이 충분히 반영되지는 못했다.

위의 두 내용들을 종합해 본다면, 트랜스코딩에 관련된 연구들은 비디오 스트리밍에 대한 품질관련 연산들을 제공해 주었으나 트랜스코딩 과정에서 발생하는 연산지연과 전송지연에 대한 고려가 부족하였다. 또한 비디오 스트리밍에 대한 캐싱기법과 관련된 연구들은 비디오 스트리밍 서버와 클라이언트 사이의 전송지연을 줄여주는 장점을 제공하는 반면, 트랜스코딩과의 협업을 위한 연동들은 드물었다. 따라서 대역폭이 유선망에 비해 상대적으로 빈약한 무선망 채널을 통해 연속적인 비디오 스트리밍 서비스를 제공하는 과정 중에 발생하는 전송지연과 연산지연을 동시에 극복할 수 있도록 해주는 통합 시스템 아키텍처와 캐싱 알고리즘이 필요하다.

본 연구에서는 트랜스코딩과 캐싱의 효과적인 연동을 통해 비디오 스트리밍 서비스 도중에 발생하는 연산지연과 전송지연을 줄일 수 있도록 하는 transco-prefix caching 알고리즘을 제안하였다. 그리고, 비디오 스트림 내용의 성격에 따라 real-time major service와 quality major service로 나누어 캐싱 정책을 다르게 하여 보다 나은 품질의 스트리밍 서비스를 제공할 수 있도록 했다.

2. 관련 연구

2.1. two-pass video coding system and transcoding architecture

원본 비디오 스트림들을 낮은 비트율의 비디오 스트림으로 바꾸는 트랜스코더는 보통 유선망과 무선망 사이에 위치한 프록시 서버에 설치된다 . 또한 프록시 서버는 기본적인 트랜스코딩의 기능뿐만이 아니라, 무선망 채널의 제한된 대역폭과 잡음 채널 에러에 민감한 특성까지 감안하여 이를 극복할 채널 코딩의 기능까지 구비해야 한다. 따라서 이와 같은 요구 조건들이 반영된 two-pass video coding system[8]과 트랜스코딩 프록시 시스템[9]이 제안되었다.

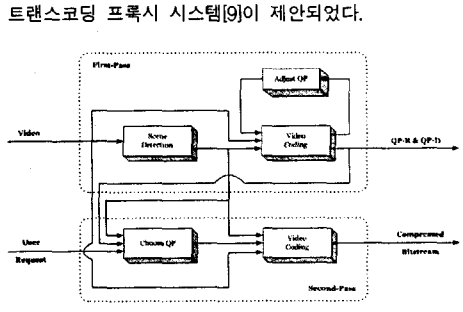

Fig. 1 two-pass video coding system [8]

Fig. 1은 비디오 스트리밍 서버쪽에서의 two-pass video coding system을 나타낸 그림이다. 서버의 coding system 에서는 비디오 시퀀스의 rate-distortion function(이 후 R-D function)들을 참조하고 quantization parameter(이 후 QP) 값들을 조절하여 비디오 스트림의 비트량을 제어하는 기법을 사용한다. R-D function 정보는 스트림 bit-rate 값과 스트림 distortion 값을 QP 값과 매핑시킨 테이블(QP-R & QP-D)로 구성된다.

이렇게 비디오 인코딩 서버에서 미리 생성된 이러한 R-D function 정보들은 비디오 스트림과 함께 프록시 서버로 전송되어 트랜스코딩할 때에도 사용할 수 있게 된다. 다시 말해 프록시 서버에서는 R-D function 정보 테이블을 참조하여 클라이언트와 프록시 서버 사이의 무선망 채널의 대역폭에 해당하는 QP 값을 찾은 뒤 그 QP값으로 원본 비디오 스트림을 트랜스코딩하게 된다.

보통의 프록시 시스템에서는 불완전한 무선망 대역폭의 변동으로 인해 잦은 트랜스코딩이 일어나게 되지만, 이 시스템에서는 비디오 스트리밍 서버에서 전송되는 rate-distortion function 과 quantizaton parameter와의 매핑정보를 이용하여 프록시 서버에서의 반복되는 트랜스코딩 연산을 줄일 수 있도록 하는 장점을 가지고 있다.

2.2. two-pass video coding system and transcoding architecture 에서 부족한 점

two-pass video coding system 에서의 트랜스코딩 아키텍처[9]에서는 R-D function 연산에 대한 부담을 서버에게 분담시켜 프록시 서버 시스템 구현에 있어서의 complexity를 낮추고 보다 빠른 스트리밍 서비스를 제공할 수 있도록 하였다. 그러나, 이 아키텍처에서는 트랜스코딩 자체에서 발생하는 지연에 대한 고려가 부족하다. 만약 동일한 비디오 스트림의 요청이 많아지는 경우에는 같은 트랜스코딩 연산이 계속 반복되어 전반적으로 비디오 스트리밍 서비스에 적지 않은 지연이 발생하게 된다. 또한 무선망 채널의 대역폭 변동이 잦은 경우에도 다양한 QP값에 따라 트랜스코딩 연산이 반복적으로 이루어져야 하기에 추가적인 지연이 발생한다. 그리고 이러한 트랜스코딩 연산지연 뿐만이 아니라 비디오 스트리밍 서버와 프록시 서버사이의 전송지연까지도 고려해야할 필요가 생긴다. 결국 이러한 지연들로 인해 발생되는 비효율성을 극복하고 보다 나은 성능의 스트리밍 서비스를 제공할 수 있는 프록시 서버 시스템이 필요하다.

3. transco-caching proxy server system

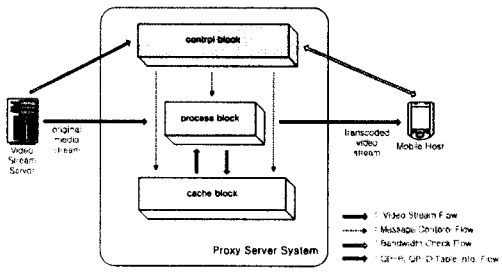

Fig. 2 proxy server system block

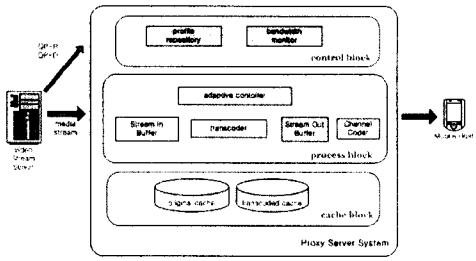

Fig. 3 transco-caching proxy server system

Fig. 2,3에서는 트랜스코딩 과정에서의 지연을 줄이고 같은 비디오 스트림에 대한 잦은 요청으로 인해 발생하는 잦은 트랜스코딩의 비효율을 줄일 수 있도록 하는 transco-caching 프록시 서버 시스템을 나타내었다. 본 프록시 서버 시스템에서는 위에서 기술한 two-pass video coding system 트랜스코딩 아키텍처[9]의 지연문제를 극복하기 위해 캐쉬들을 삽입하고, QP-R과 QP-D 테이블 정보를 참조하여 양자화 파라미터 별로 각각 트랜스코딩 된 비디오 스트림들이 transcoded cahce에 저장될 수 있도록 제어하는 역할을 수행하는 adaptive controller를 추가하였다.

프록시 서버 시스템은 각각의 모듈들로 구성되어 있다. 모듈들은 그 동작 특성에 따라 control block과 process block, 그리고 cache block으로 분류된다. process block에서는 비디오 스트리밍 서버로부터 전송되는 비디오 스트림의 실제적인 트랜스코딩 연산과 채널코딩 그리고, 버퍼링 연산들이 이루어진다. cache block 에서는 입출력 되는 스트림과 트랜스코딩 연산이 되고 있는 스트림들의 캐슁작업이 수행되며, control block에서는 process block과 cache block 에서의 연산 작업 때 필요한 각종 정보들을 수집하고, 수집된 정보들을 바탕으로 제어 메시지를 생성하는 역할을 수행한다.

프록시 서버 시스템을 구성하는 모듈들로는 다음과 같다. 원본 비디오 스트림만을 저장하는 original cache와 트랜스코딩이 된 스트림만을 저장하는 transcoded cache 등의 캐쉬모듈과 stream-in buffer와 stream-out buffer 로 이루어진 버퍼모듈, 비디오 스트림 데이터에 대한 정보가 담겨 있는 profile repository 모듈, 프록시 서버와 모바일 호스트 사이의 무선망 대역폭을 주기적으로 체크하는 bandwidth monitor 모듈, 채널 에러 보정을 위한 코드가 덧붙여 지는 channel coder 모듈, 그리고 각종 트랜스코딩 연산이 일어나는 transcoder 모듈과 프록시 서버 시스템의 각종 모듈들을 최종적으로 제어하는 역할을 하는 adaptive controller가 있다.

4. co-operation between transcoding and caching at the transo-caching algorithm

4.1 Basic operation

트랜스코딩과 캐슁의 협업을 위한 핵심 제어는 adaptive controller에서 수행된다. 프록시 서버 내의 캐쉬에 아무런 스트림 데이터가 저장되어 있지 않은 채 최초로 스트리밍 서비스가 시작되면 우선, QP-R, QP-D 테이블로부터 프록시 서버와 모바일 호스트 사이의 무선망 채널 대역폭 값에 해당하는 QP(quantization parameter)값을 찾아 프록시 서버로 들어오고 있는 비디오 스트림의 QP값과 비교를 한다. 만약 그 QP값이 서로 다를 경우에는 무선망 채널 대역폭에 맞추어 새롭게 갱신된 QP값으로 비디오 스트림을 트랜스코딩한다. 동시에 비디오 스트리밍 서버와 프록시 서버 사이의 유선망 전송 지연과 트랜스코딩 과정 중에 발생하는 지연만큼 해당하는 비디오 스트림 데이터의 영상 프레임들을 transcoded cache에 저장한다. 곧 전송지연과 연산지연 분량 만큼의 비디오 영상 프레임들을 캐쉬에 미리 저장시켜 놓아 차후에 같은 비디오 스트림 요청이 반복되었을 때 보다 빠른 prefetching이 가

능하도록 한 것이다.

특별히 비디오 스트리밍 서버와 프록시 서버 사이의 전송지연이 transcoder에서의 트랜스코딩 연산지연 보다 길어질 경우에는 원본 비디오 스트림 데이터를 복제하여 original cache에 저장을 한 후 다시 트랜스코딩 연산을 필요로 할 때 사용하도록 한다. 물론 비디오 스트리밍 서버와 프록시 서버 사이의 전송지연 분량 만큼을 캐쉬에 저장시키도록 한다.

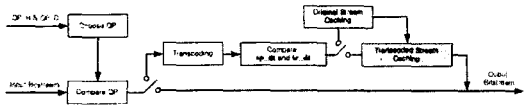

Fig. 4 Basic operation of transcoding and caching process in the adaptive controller

Soc = bit_rate * sp_dt (if, sp_dt > tr_dt)
Stc = Soc + bit_rate * tr_dt

where,
Soc : original cache에 캐싱되는 비디오 스트림의 양
Stc : transcoded cache에 캐싱되는 비디오 스트림의 양
sp_dt : 스트리밍 서버와 프록시 서버 사이의 전송지연 시간
tr_dt : transcoder에서 트랜스코딩 연산을 하는 데 걸리는 시간
bit_rate : 스트리밍 서버로부터 전송되는 비디오 스트리밍의 시간당 bit throughput

4.2 Caching algorithm

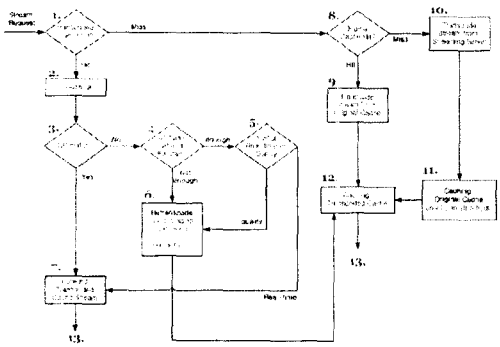

Fig. 5 transco-caching algorithm

① 모바일 호스트에서 비디오 스트림을 요청하면 프록시 서버에서 그 요청을 가로채어 해당 비디오 스트림이 transcoded cache에 존재하는지 확인한다. 존재하면(Hit) ②, 없다면(Miss) ⑧으로 간다.
② apative controller에서는 QP-R & QP-D 매핑 정보 테이블을 참조하여 현재 무선망 대역폭에 해당하는 QP값을 선택한다. ③으로 간다.
③ 선택된 QP값이 transcoded cache 스트림의 QP값과 같은지 체크한다. 같다면 ⑦, 다르면 ④로 간다.
④ transcoded cache 스트림의 QP값에 해당하는 bit-rate와 현재 무선망 대역폭과 비교한다. bit-rate가 대역폭보다 작아 바로 무선망 채널로 전송할 수 있다면 ⑤로, 재가공하여 전송해야 한다면 ⑥으로 간다.
⑤ 사용자가 보다 스트리밍 전송 속도에 민감하다면 ⑦로, 스트리밍 품질에 더 민감하다면 ⑥으로 간다.
⑥ 만약 해당 스트림의 original cache가 Miss이면 ⑩으로 가고, Hit이면 선택된 QP로 다시 트랜스코딩한다. ⑫로 간다.
⑦ 무선망 채널을 통해 transcoded cache 스트림을 모바일 호스트로 바로 전송한다. ⑬으로 간다.
⑧ original cache에 해당 스트림의 원본 비디오 스트림이 존재(Hit)한다면 ⑨로, 없다면(Miss) ⑩으로 간다.
⑨ original cache 스트림을 선택된 QP로 트랜스코딩한다. ⑫로 간다.
⑩ 스트리밍 서버로부터 해당 스트림을 직접받아 선택된 QP로 트랜스코딩한다. ⑪로 간다.
⑪ 전송지연이 연산지연보다 크다면 original cache에 스트리밍서버로부터 전송된 원본 비디오 스트림을 Soc만큼 저장한다. ⑫로 간다.
⑫ transcoded cache에 트랜스코딩된 비디오 스트림을 Stc만큼 저장한다. ⑬으로 간다.
⑬ End.

각각의 캐쉬에 Hit 혹은 Match된 스트림을 바로 모바일 호스트로 전송하면서 동시에, Soc, 혹은 Stc 이후부터의 playout 프레임을 스트리밍 서버로 요청한다. 스트리밍 서버로부터 Soc, Stc 이후의 playout 프레임이 프록시 서버로 도달하면 트랜스코딩 연산 후에, 이전 캐쉬에 저장되어 있던 이전 프레임들과 동기를 맞추는 작업이 필요하다. 이후 두 프레임 스트림들을 연결하여 사용자에게 연속적으로 스트리밍 서비스를 제공할 수 있도록 한다.

5. Conclusion

트랜스코딩과 캐싱과의 효과적인 연동을 통한 협업으로 연산지연과 전송지연을 극복할 수 있도록 하는 프록시 시스템을 제안하였다. 본 프록시 시스템에서는 비디오 스트리밍 서버에서 생성되는 quantization parameter와 bit-rate와의 매핑정보를 프록시 서버에서 참조하여, quantization parameter 별로 해당 비디오 스트림 데이터를 트랜스코딩하여 캐싱하는 방법을 사용하였다. 이러한 매핑정보는 불필요한 트랜스코딩 연산의 반복을 줄여줄 수 있을 뿐만 아니라 트랜스코딩 된 각각의 깊이 별로 캐싱하는 데에도 유용하게 사용되어 기존의 개별적인 트랜스코딩과 캐싱 기법들보다 보다 더 연속적인 스트리밍 서비스를 제공할 수 있도록 해준다. 또한 캐쉬의 저장공간의 효율을 높이기 위해 전송지연과 연산지연 만큼 prefetching 할 수 있을 정도로만 캐싱하였다. 그리고, 비디오 스트림 내용의 성격에 따라 real-time major service와 quality major service로 나누어 캐싱 정책을 다르게 하여 보다 나은 품질의 스트리밍 서비스를 제공할 수 있도록 했다.

6. 참고 문헌

[1] Peng Yin, Min Wu, Bede Liu: "Video Transcoding by Reducing Spatial Resolution", IEEE ICIP 2000
[2] Anthony Vetro, Peng Yin, Bede Liu, Huifang Sun: "Reduced Spatio-temporal Resolution Transcoding Using Intra-Refresh Technique", IEEE Intl. Sym. on Circuits and Systems, 2002
[3] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," IEEE Trans. on Circuits and Systems for Video Technology, pp. 953-967, Dec. 1998.
[4] H. Sun, W. Kwok, and J. Zdepski, "Architecture for mpeg compressed bitstream scaling," IEEE Trans. on Circuits and Systems for Video Technology, pp. 191-199, April 1996.
[5] Z.L. Zhang, Y. Wang, David H.C.Du, D. Su, "Video Staging : A Proxy-Server-Based Approach to End-to-End Video Delievery over Wide-Area Networks", IEEE/ACM TRANSACTIONS ON NETWORKING. VOL.8.NO.4,August 2000
[6] S. Sen, J. Rexford, D. Towsley, "Proxy Prefix Caching for Multimedia Stream", INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies.
[7] H. Fahmi, M. Latif, S. Sedigh-Ali, A. Ghafoor, P. Liu, L. H.Hsu, "Proxy Servers for Scalable Interactive Video Support",Computer , Volume: 34Issue: 9, Sep 2001,Page(s): 54-60
[8] J. Cai and C. W. Chen, "Two-pass video encoding for low bit rate streaming applications," in SPIE VCIP 2002, Jan. 2002.
[9] J. Cai and C. W. Chen, "A high-performance and low-complexity video transcoding scheme for video streaming over wireless links", IEEE WCNC 2002, Orlando, FL, March, 2002.