

이종 클라이언트들을 위한 멀티레이트 TCP 비디오 스트리밍에 관한 연구

정희원 정 영 하*, 종신회원 최 윤 식*

Multi-Rate TCP Video Streaming for Client Heterogeneity

Young H. Jung* *Regular Member*, Yoonsik Choe* *Lifelong Member*

요 약

본 논문에서는 동일 서버를 통하여 이종 클라이언트들에게 동시에 비디오 스트리밍 서비스를 제공하는 환경에서 서비스 품질 보장을 위한 스트리밍 알고리즘을 제안하였다. 이러한 환경에서 네트워크 자원을 효율적으로 사용하고 신뢰성 있는 전송을 수행하기 위해 전송계층 프로토콜로서 TCP를 사용 하여 비디오 스트리밍 서비스를 제공하면 각 스트리밍 세션들을 위한 TCP 커넥션들은 링크 대역폭을 공평하게 공유하게 된다. 이로 인해 상대적으로 높은 비트레이트를 요구하는 스트리밍 클라이언트는 재생버퍼 언더런과 같은 품질저하를 겪게 될 수 있는데 본 논문에서는 이러한 문제를 해결하기 위하여 각 클라이언트들의 요청 대역폭과 재생버퍼 크기를 고려하여 적응적으로 다중 TCP 커넥션을 사용하는 방안을 제안하였다. 제안 알고리즘을 통하여 이종 클라이언트들의 환경에 맞는 스트리밍 대역폭을 보장할 수 있게 됨을 실험결과를 통하여 확인 할 수 있다.

Key Words : 멀티레이트 비디오, 이종 단말 지원, TCP 스트리밍, QoS

ABSTRACT

In this paper, we propose a video streaming server that guarantees a certain level of quality when a server should serve video streaming service to multiple heterogenous clients simultaneously with TCP transport. If each heterogeneous client requests video streaming service in according to its own requirement such as bitrate of content and these requests are accepted by a server, then TCP flows for each video streaming session fairly share limited uplink bandwidth of the server. At this time, because TCP's bandwidth fair-share characteristics can result in bandwidth shrinkage of higher bitrate video streaming session, the client of higher bitrate video may suffer sluggish playback which is related with streaming QoS degradation. To tackle this problem, our proposed server system uses multiple TCP connections adaptively for each video streaming session depending on the anticipated status of the client playout buffer. Simulation results show that our proposed algorithm can successfully reduce the occurrence of playout buffer underrun and enhance streaming quality for whole video clients.

I. 서 론

최근 브로드밴드 인터넷 접속 서비스의 비약적인

확대에 따라 다양한 형태의 멀티미디어 스트리밍 서비스들이 활발히 사용되고 있다. 이러한 인터넷상에서의 멀티미디어 스트리밍 서비스를 위하여 전통

※ 본 연구는 지식경제부 산하 정보통신 연구진흥원의 대학IT 연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-(C1090-0801-0011))

* 연세대학교 전기전자 공학과 영상정보 연구실 ({crosscom, yschoe}@yonsei.ac.kr)

논문번호 : KICS2007-12-551, 접수일자 : 2007년 12월 07일, 최종논문접수일자 : 2008년 03월 21일

적으로는 전송계층 프로토콜로서 UDP를 사용하는 것이 당연한 것으로 여겨져 왔다. 이는 TCP 프로토콜이 재전송, 혼잡제어 등의 알고리즘과 확인 메시지 수신 등으로 인해 실시간 어플리케이션의 사용에 있어 적합하지 않다고 생각되어져왔기 때문인데, 최근 브로드밴드 인터넷 접속과 인터넷 인프라의 발달로 인해 TCP를 이용한 멀티미디어 스트리밍 서비스가 도입될 수 있게 되었다^[1]. TCP를 이용한 멀티미디어 스트리밍은 인터넷상에서의 방화벽 이슈, 다른 어플리케이션 플로우들과의 대역폭 공평성 문제, 신뢰성 있는 전송 등의 장점을 갖고 있기 때문에 다양한 응용환경 하에 점점 더 활발하게 사용되고 있다.

본 논문에서는 TCP를 이용한 다양한 멀티미디어 스트리밍 환경 중에서 하나의 스트리밍 서버에서 PC, PDA, 모바일 등의 다양한 단말들로 스트리밍 서비스를 제공하는 환경을 고려하였다. 이러한 환경에서 TCP를 이용하여 이중 단말들이 각각 요구하는 비트레이트로 스트리밍 서비스를 제공하면 TCP 플로우간의 공평분배 (fair-share) 특성으로 인해 서버의 전송 대역폭을 TCP 플로우들이 균등하게 점유하게 된다. 즉, 서로 다른 재생 비트레이트를 갖는 스트리밍 세션들이 전송계층의 동작으로 인해 전송 대역폭을 균등하게 사용 점유하게 된다.

서로 다른 비트레이트를 갖는 스트리밍 서비스들이 TCP로 인해 전송대역폭을 균등하게 점유하게 되면, 상대적으로 높은 비트레이트를 요구하는 스트리밍 세션의 가용대역폭이 비디오의 재생 비트레이트 이하로 감소하게 될 수 있다. 이렇게 되면 해당 스트리밍 세션은 재생버퍼 (Play-out-Buffer)의 고갈을 겪게 되고 이는 리버퍼링을 위한 재생중단으로 이어지게 된다. [2][3]과 같은 최근의 많은 연구들이 스트리밍 서비스의 품질적으로 사용하듯이 멀티미디어 스트리밍 도중 발생하는 리버퍼링을 위한 잦은 재생 중단은 사용자 입장에서 심각한 서비스 품질 저하로 간주된다. 이와 같이 TCP를 이용한 멀티미디어 스트리밍 중 사용대역폭 감소는 스트리밍 품질 저하와 직결되기 때문에 보다 고품질의 스트리밍 서비스를 보장하기 위해서는 서버 단에서 대역폭 보장을 위한 알고리즘이 필수적으로 제공되어야 한다.

Gürses 등은 [4]에서 제한된 대역폭을 갖는 서버 채널 상에서 TCP 기반의 스트리밍을 위하여 SFD (Selective Frame Discard) 알고리즘을 적용하여 사용가능한 대역폭에 맞게 왜곡이 최소화 되도록 스

트리밍 레이트를 조정하도록 하였다. TCP 단에서 예측된 가용대역폭 이하로 프레임 드롭을 통해 스트리밍 레이트를 줄이되 수신 단이 경험할 왜곡이 최소화되도록 드롭될 프레임을 선정하였다. SFD 알고리즘을 본 논문에서 고려하고 있는 상황, 즉 서로 다른 요구 비트레이트와 플레이어아웃 버퍼를 갖는 이중 스트리밍 단말들을 동시에 지원하는 상황에 적용한다면 상대적으로 큰 비트레이트를 요구한 스트리밍 플로우에서 프레임 드롭에 의한 왜곡이 많이 발생할 가능성이 크기 때문에 서비스 형평성 상의 문제를 야기하게 된다.

Nguyen 과 Cheung은 [5]에서 TCP 전송을 이용하여 멀티미디어 스트리밍을 할 때 사용 대역폭의 단기간변화 (short-term fluctuation)에 대응하기 위하여 다중 TCP (Multiple TCP:이하 MultiTCP) 커넥션을 이용하는 방안을 제안하였다. 멀티미디어 스트리밍을 위하여 TCP를 사용할 때 어플리케이션과 TCP 계층 중간의 MultiTCP 컨트롤 유닛을 통하여 고정된 개수의 여러 TCP 세션을 열고 이를 통하여 스트리밍 서비스를 시도한다. 이렇게 여러 TCP 세션을 사용함으로써 혼잡제어 상황에서 단일 세션을 사용할 때보다 적은 폭의 혼잡윈도우 감소를 경험하게 되고 이로 인해 단기간 대역폭 변화가 감소하게 된다. [5]에서 제안된 MultiTCP의 사용은 단기간 대역폭 변화에 잘 대처할 수 있지만, 모든 스트리밍 세션들에 대해 고정적인 개수의 TCP 커넥션을 사용하기 때문에 서로 다른 비트레이트를 갖는 영상들을 스트리밍할 때 적용 시에는 경우에 따라 성능이 제한적일 수밖에 없다.

본 논문에서는 이중 단말들이 존재하는 서비스 상황에서 하나의 스트리밍 서버로 이들 단말들에 대해 품질이 보장된 TCP기반의 스트리밍 서비스를 제공하기 위하여 다중 TCP 연결을 각 단말들의 전송상황에 따라 적응적으로 사용하는 방식을 제안하였다. TCP 플로우간의 공평분배가 존재하는 상황에서 멀티미디어 스트리밍 세션의 대역폭을 보장하기 위하여 적응적으로 다중 TCP 세션을 설정하도록 하였다. 이때 추가되는 부수적인 TCP 세션들의 생성은 이중 단말들의 서로 다른 재생 버퍼 상태를 기준으로 하여 각 네트워크 상황에 따라 적응적으로 생성하게 하여 각 클라이언트의 버퍼 언더런 발생을 최소화 할 수 있도록 고안되었다. 이를 위하여 서버 단에서는 클라이언트 버퍼 상태를 피드백 받고 클라이언트가 현재 수신 가능한 가용대역폭의 통계적 특성을 토대로 이후의 버퍼 상태를 예측하

고 필요할 경우 추가 TCP 세션을 생성/삭제하는 알고리즘을 제안하였다. 다수의 TCP 커넥션을 사용하여 멀티미디어 스트리밍에 활용하는 방안은 [5]에서 제안된 방식과 유사하지만, 서로 다른 비트레이트를 갖는 스트리밍 플로우들 간의 공평 분배특성을 극복하여 스트리밍 대역폭을 확보하고 클라이언트의 재생버퍼 상태를 예측하여 적응적으로 추가TCP 세션들을 생성하는 점에서 [5]의 연구와 차별된다.

본 논문의 이후 구성은 다음과 같다. II장에서는 초점을 맞추고 있는 네트워크 및 서비스 환경, 가정들에 대해서 설명을 하고, III장에서는 제안된 적응적 다중 TCP 스트리밍 알고리즘에 대해서 설명한다. IV장에서는 네트워크 시뮬레이터를 이용한 시뮬레이션 결과들 통하여 제안방식의 효과를 검증하고 V장을 통해 결론과 앞으로의 연구방향에 대해 설명하고자 한다.

II. 서비스 모델 및 망구조

본 논문에서 고려하는 스트리밍 서비스망의 구조는 그림 1과 같다. 인터넷을 통하여 스트리밍 서비스를 수행하는 서버가 브로드밴드 인터넷 접속을 통해 존재하고 클라이언트들은 서로 다른 네트워크 및 시스템 요구사항을 갖고 각각의 유무선 네트워크 접속 환경을 이용하여 스트리밍 서비스를 제공받는다. 보통 스트리밍 서버가 브로드밴드 인터넷 접속을 통하여 인터넷에 연결되어 있기 때문에 스트리밍전송을 보내주는 업링크 대역폭에는 일정한 제한이 있게 된다. 이렇게 한정된 업링크 자원을 통하여 이종 단말들에게 스트리밍 서비스를 제공하는 환경을 본 논문에서는 고려한다.

그림 1의 서비스 환경에서는 I장에서 설명한 바와 같이 인터넷 접속 시 방화벽 통과와 이슈등과 재전송이 보장된 전송을 수행하기 위하여 모든 스

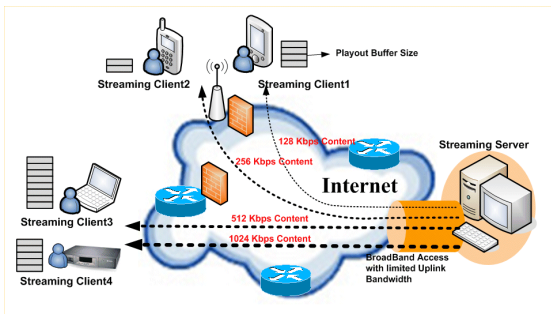


그림 1. 이종 클라이언트들의 스트리밍 서비스 구조

트리밍 서비스는 TCP 전송계층 프로토콜을 이용하였다. TCP를 이용하여 저장된 비디오 콘텐츠에 대해 비디오 스트리밍을 수행할 때 [1]에서 언급한바와 같이 비디오 스트리밍 서버는 활용 가능한 네트워크 대역폭을 최대한 충분히 사용하기 위하여 TCP 대역폭이 지원하는 한 최대한 빠르게 비디오 스트리밍 데이터 전송을 시도한다고 가정하였다. 즉, FTP와 같은 대용량데이터 전송 시와 같이 비디오 어플리케이션 레이어는 TCP 레이어의 송신버퍼가 수용할 수 있는 한도 내에서 최대한 빠르게 전송계층 레이어로 전송한다. 또한 고려된 서비스 환경에서 비디오 스트리밍에 대한 수락제어는 요청 콘텐츠의 비트레이트를 기준으로 수행중인 스트리밍 세션의 비트레이트들과의 합이 서버의 업로드 대역폭을 초과하지 않을 경우에 수락되는 것으로 가정하였다.

III. 이종 클라이언트 지원을 위한 적응적 다중 TCP 스트리밍 기법

3.1 실시간 클라이언트 재생버퍼 예측

제안 알고리즘에서 이종 클라이언트별 스트리밍을 위해 스트리밍 서버는 각 클라이언트들의 재생버퍼의 상태를 확인하고 이를 통해 가까운 미래의 상태를 예측한다. 이러한 재생 버퍼의 상태 예측을 위해 그림 2와 같은 스트리밍 모델을 사용하였다. $A(t)$ 를 시간 t 에서 현재까지 수신된 스트리밍 데이터의 바이트 수라 표현하고, $P(t)$ 를 시간 t 에서 현재까지 재생된 비디오 스트리밍 데이터의 바이트 수라 표현한다. 실제 스트리밍 시스템에서 사용하듯이 스트리밍 클라이언트 어플리케이션은 네트워크전송 도중 발생하는 전송지연의 효과를 줄이기 위하여 초기버퍼링 (pre-roll buffering)을 수행하고 이러한 초기버퍼링 시간을 τ_a 한다. 대부분의 스트리밍 시스템에서와 같이 재생 버퍼 언더런 발생시 τ 시

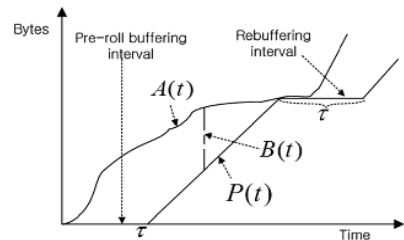


그림 2. 스트리밍 서비스 모델

간 만큼의 리버퍼링을 수행한다고 가정하였다. 또한 고정비트레이트 (Constant Bit Rate : 이하 CBR)의 부호화 비트레이트 혹은 재생 비트레이트를 μ 라 표시하도록 한다. $P(t)$ 는 식(1)과 같이 근사화 되어 모델링 될 수 있다.

$$P(t) \approx \begin{cases} \mu(t - \tau(1 + Count_{RB})), & \text{if } t > \tau \\ 0, & \text{if } t \leq \tau \end{cases} \quad (1)$$

식 (1)에서 $Count_{RB}$ 는 스트리밍 대역폭 감소 시 발생하게 되는 재생버퍼 고갈로 인한 리버퍼링 횟수를 의미한다. $B(t)$ 를 클라이언트 재생버퍼의 t 시간까지 남아있는 바이트 수라 한다면 식 (2)에 의해 계산될 수 있다.

$$B(t) = A(t) - P(t) \quad (2)$$

클라이언트 재생 버퍼의 가까운 미래 상태를 예측하기 위해서는 $B(t_{curr} + \Delta t)$ 를 구해야 한다. $t \leq t_{curr}$ 의 구간에서 $A(t)$ 는 클라이언트로부터 서버로 전송되는 주기적 피드백 정보를 통해 알려져 있지만, $A(t_{curr} + \Delta t)$ 는 네트워크의 다이내믹한 특성으로 인해 랜덤한 값을 갖게 된다. $A(t)$ 는 비감소 함수(non-decreasing function)이기 때문에 식(3)과 같은 윈도우 기반의 선형회귀함수(linear regression function)로 모델링 할 수 있다.

$$A(t) = \alpha + \beta t + \epsilon \quad (3)$$

여기서 α 와 β 는 회귀함수 파라미터들이고 ϵ 은 평균이 0인 가우시안 랜덤 변수를 나타낸다. 이전의 w 개의 도착된 바이트 샘플들로부터 서버의 제안 알고리즘 모듈은 최소자승 에러를 보장하는 회귀함수 파라미터를 구하기 위하여 식 (4)를 사용한다.

$$\beta = \frac{\sum_w t_i A_i - \bar{t} \sum_w A_i}{\sum_w t_i^2 - w \bar{t}^2}, \quad \alpha = \bar{A} - \beta \bar{t} \quad (4)$$

식 (4)에서 \bar{A} 와 \bar{t} 들은 이전의 w 개의 클라이언트 수신 바이트들과 샘플 시간들의 평균값을 의미한다. 제안 시스템에서, 스트리밍 서버는 각 클라이언트들로부터 현재까지의 수신바이트 정보와 현재까지 경험된 리버퍼링 횟수를 포함하고 있는 피드백 데이터를 수신하고 이 때 위의 식 (2), (3), (4)등을 사용하여 $B(t_{curr} + \Delta t)$ 를 계산한다. 만약 예측된 버퍼 상태가 미리 정해진 임계값보다 낮을 경우 3.4장의 알고리즘을 따라 부수적인 TCP 커넥션을 연

결하여 스트리밍 대역폭을 보장하도록 한다.

3.2 보호구간 예측 방법

각 클라이언트의 예측된 버퍼 상태에 따라 새로운 TCP 연결들이 설정되면, TCP의 기본 동작에 의해 새로운 TCP는 혼잡제어 알고리즘으로 현재 활용 가능한 네트워크 대역폭을 탐색하게 된다. 다른 TCP 플로우의 생성과 같이 링크내의 네트워크 변화가 없으면 각 TCP 커넥션의 대역폭은 일정한 수준으로 안정화 되게 된다. 이때까지 걸리는 시간을 TCP 보호시간 (Guard-Interval) 이라 명하였다. 제안 알고리즘에서 스트리밍 세션을 위하여 TCP 커넥션을 하나 혹은 여러 개 추가하게 되면, 이 보호 시간 동안 해당 스트리밍 세션을 위하여 TCP 커넥션을 더 생성하지 않아야 한다. 이는 TCP 보호시간 자체가 새로운 TCP 커넥션이 효과를 보이기까지 대기해야할 시간을 의미하기 때문이다.

TCP 보호 시간은 네트워크 상태에 따라 변할 수 있기 때문에 고정된 값을 사용하지 않고 실시간으로 예측을 해야 한다. 보호시간을 결정하기 위하여 서버 모듈은 클라이언트로부터 피드백되어 전송되는 데이터 수신레이트의 이동평균(moving average)을 이용하여 예측한다. 즉, 스트리밍을 위한 첫 TCP 생성 혹은 부수적인 TCP 세션들이 생성될 때의 시간 t_{init} 과 그 이후에 식 (5)를 만족하는 t_{stab} 를 찾은 후 이들의 차를 식 (6)과 같이 보호 시간으로 사용하도록 하였다.

$$R_{avg}(t) = m \times R_{avg}(t-1) + n \times R_{meas}(t) \\ t = t_{stab}, \text{ if } |R_{avg}(t) - R_{meas}(t)| \leq R_{avg}(t) \times \sigma \quad (5)$$

$$t_{guard} = t_{stab} - t_{init} \quad (6)$$

식 (5)에서 $R_{meas}(t)$ 는 클라이언트의 어플리케이션 계층에서 측정된 스트리밍 데이터의 수신 레이트를 의미하며 $R_{avg}(t)$ 는 이들의 이동평균을 의미한다. $R_{meas}(t)$ 값은 피드백 정보를 통하여 클라이언트로부터 서버로 전달된다. 또 식 (5)에서 m 과 n 은 이동평균 파라미터를 나타내며 σ 는 허용 가능한 네트워크 대역폭의 변동비를 의미한다.

3.3 자기 제한적 소스 전송률 제어

II장에서 가정한 바와 같이 TCP를 이용한 저장 비디오의 스트리밍시 벌크데이터의 전송 경우와 유사하게 네트워크의 가능 대역폭을 최대한 활용하기 위하여 서버 어플리케이션은 TCP 전송버퍼가 수용

할 수 있는 범위 내에서 최대한 빨리 전송계층 레이어로 스트리밍 데이터를 보낸다. 각 클라이언트의 네트워크상태와 재생버퍼크기를 고려하여 부수적인 TCP 커넥션을 추가/삭제하는 제안 알고리즘은 어플리케이션 계층에서 수행되기 때문에, 이렇게 벌크 데이터 전송방식으로 어플리케이션에서 전송계층으로 짧은 시간 내에 모두 보내버리면, 어플리케이션 계층에서는 이 후에 발생하는 네트워크 변화에 대처하여 절적이 대역폭 확보를 할 수가 없게 된다.

이러한 문제를 극복하기 위하여 제안 알고리즘에서는 어플리케이션 계층에서 전송계층으로 스트리밍 데이터를 보낼 때 스트리밍 비디오의 부호화 비트레이트 이상의 일정 레이트로 보내도록 하였고 이를 자기 제한적 어플리케이션 전송률 제어 (self-limiting source rate control)로 이름 하였다. 네트워크의 활용가능 대역폭을 최대한 이용하고 재생 비트레이트를 보장하기 위하여 어플리케이션의 자기 제한 레이트는 그림 3의 리키버킷 모델을 이용하여 조절이 되는데, 이때 버킷의 토큰 레이트는 식 (7)에 의해 결정된다.

$$R_{avg} = \text{arg max} (R_{avg}, \mu) \quad (7)$$

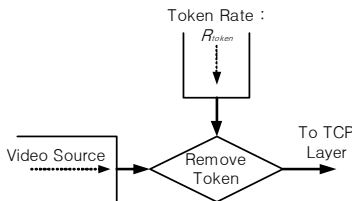


그림 3. 자기 제한적 소스 전송률 제어 리키 버킷

3.4 적응적 다중 TCP 연결

제안 알고리즘에서 스트리밍 서버는 각 클라이언트로부터 피드백 정보를 수신 후 클라이언트들의 재생 버퍼의 가까운 미래 상태를 예측한 후 이 값이 버퍼 최소임계값보다 작을 경우 스트리밍 대역폭의 추가 확보를 위한 부수적인 TCP 커넥션을 생성한다. 스트리밍 서버의 복잡도를 고려하여 추가적으로 생성할 TCP 커넥션의 수는 식 (8)의 비례식을 이용하도록 한다.

$$\frac{N_{tot} : k_{curr} = R_{tot} : R_{avg}}{N_{tot} + K_{new+} : k_{curr} + k_{new+} = R_{tot} : \mu} \quad (8)$$

식 (8)에서 N_{tot} 은 현재 스트리밍 서버가 갖고 있

는 총 TCP 커넥션 수를 의미하며, k_{curr} 는 해당 비디오 스트리밍 세션을 위해 생성한 TCP 커넥션 수를 의미하고 k_{new+} 는 스트리밍 대역폭 확보를 위해 추가적으로 생성해야할 TCP 커넥션 수를 의미한다. 또한, R_{tot} 는 서버가 전송할 수 있는 최대 업링크 대역폭을 의미하며 이는 스트리밍 서버 어플리케이션에 미리 설정되어 있다고 가정한다. 비례식 (8)을 구하고자 하는 k_{new+} 에 대해서 풀면 식 (9)의 결과를 얻을 수 있다.

$$k_{new+} = \lfloor \frac{(\mu - R_{avg})R_{tot}}{(R_{tot} - \mu)R_{avg}} \times k_{curr} \rfloor \quad (9)$$

한편, 예측된 미래의 재생버퍼 상태가 버퍼의 최대임계값을 초과할 경우, 스트리밍 서버는 부수적인 TCP 커넥션들을 줄여나간다. 이때, TCP 커넥션의 삭제는 갑작스러운 스트리밍 대역폭 감소를 방지하기 위하여 다중으로 증가시키던 방식과 달리 단조 감소 방식으로 삭제해 나간다.

IV. 실험 결과

본 장에서는 이중 클라이언트들에 동시 스트리밍 시 제안알고리즘의 성능에 관한 평가를 하고자 한다. 시뮬레이션을 위하여 트레이스 기반의 네트워크 시뮬레이터인 NS-2 [6]을 사용하였고 비디오 스트리밍 모듈을 사용하기 위하여 EvalVid [7]을 TCP 스트리밍 환경에 맞도록 수정하여 사용하였다. 각 클라이언트의 재생 동작 상태를 살펴보기 위하여 간단한 스트리밍 환경을 그림 4와 같이 설정하였다. 즉, 2.6Mbps의 최대 업로드 대역폭을 갖는 스트리밍 서버가 4개의 이중 클라이언트에게 스트리밍하는 상황을 가정하여 대역폭 내에서 최대한의 스트리밍 세션이 수락된 상황을 모델링 하였다. 4개의 이중 클라이언트들의 요청 스트리밍 비트레이트(μ)와 재생버퍼 크기(τ)는 표 1과 같다. 각 비디오 스트리밍에 사용된 비디오 콘텐츠는 다양한 해상도의

표 1. 시뮬레이션을 위한 이중 클라이언트들의 사양

	요청컨텐츠 평균 부호화 레이트 (bps)	요청 컨텐츠 해상도	클라이언트 초기 재생버퍼 사이즈 (초)
클라이언트 1	1024 Kbps	720x480	5 Sec
클라이언트 2	768 Kbps	352x288	4 Sec
클라이언트 3	512 Kbps	352x288	3 Sec
클라이언트 4	128 Kbps	177x144	2 Sec

영상을 초당 30 프레임으로 MPEG4 인코더를 사용하여 부호화 하여 각각 총 79.969 초 길이의 비트 스트림을 각 클라이언트로 동시에 스트리밍하도록 실험환경을 설정하였다. 성능 평가를 위하여 기존의 방식대로 하나의 스트리밍 세션을 위하여 하나의 TCP커넥션만을 사용한 경우, MultiTCP [5]에서 제안된 바와 같이 하나의 스트리밍 세션을 위하여 5개의 고정된 TCP커넥션을 사용한 경우와 본 논문에서 제안한 적응적 다중 TCP (Adaptive Multiple TCP connection: 이하 A-MultiTCP)를 사용한 경우를 같은 상황에서 실험하여 비교하였다.

그림 5, 6과 7은 기존 방식과 제안방식을 사용할 때 각 클라이언트별 재생버퍼의 상태를 나타낸다. 그림 5에서 볼 수 있듯이, 기존의 방식을 사용할 때 상대적으로 낮은 품질의 스트리밍을 요청한 클라이언트들은 재생 버퍼의 언더런이 발생하지 않는 반면 높은 품질의 스트리밍을 요청한 클라이언트 1은 비교적 빈번한 버퍼 언더런을 경험하게 된다. 또한 기존의 방식에서는 클라이언트 4의 재생버퍼 크기가 크지 않음에도 불구하고 재생비트레이트 보다 네트워크의 가용 대역폭이 크기 때문에 상대적으로 많은 양의 비디오 스트림을 저장하게 되어 재생버퍼를 비효율적으로 사용하게 됨을 확인할 수 있다. 이는 실제 단말에서는 버퍼 오버 플로우로 연결되어 스트리밍 품질의 저하가 발생할 수도 있다. 또한 그림 6을 통하여, MultiTCP [5]을 사용할 경우에도 상대적으로 높은 품질의 스트리밍을 요청한 단말들이 빈번한 재생버퍼 언더런을 경험하게 되고 재생버퍼의 비효율적 사용이 개선되지 않음을 확인할 수 있다. 이에 반하여 제안방식을 사용한 시스템에서의 결과는 그림 7에서 나온 것 과 같이 높은 비트레이트의 스트리밍을 요청한 클라이언트 1과 2에서도 재생버퍼 언더런이 발생하지 않는 것을 확인할 수 있다. 또한 재생버퍼가 크지 않은 모바일 단말과 같은 클라이언트 4에서도 기존 방식과 달리 스트리밍 데이터가 재생버퍼를 과도하게 점유하지 않게 되고 각각의 클라이언트들이 자신들이 보유하고 있는 재생버퍼의 크기에 맞게 효율적으로 스트리밍 데이터를 저장하는 것을 확인할 수 있다.

*표 2는 설정된 시나리오로 각 클라이언트들이 스트리밍 서비스를 받을 때 경험하게 되는 리버퍼링 횟수와 79.969 초 길이의 비디오 데이터를 스트리밍하는데 걸리는 총 시간에 대해서 기존 방식과 제안된 A-MultiTCP 방식에서의 차이를 나타낸다. 기존의 방식과 MultiTCP [5]의 방식에서는 가장 높

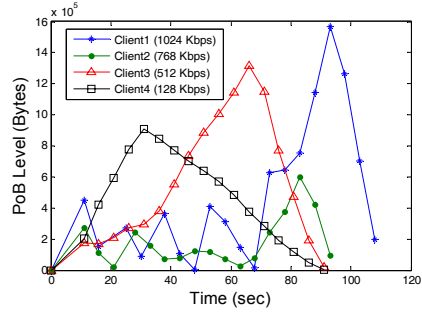


그림 5. 각 클라이언트별 재생버퍼의 상태 (기존 방식)

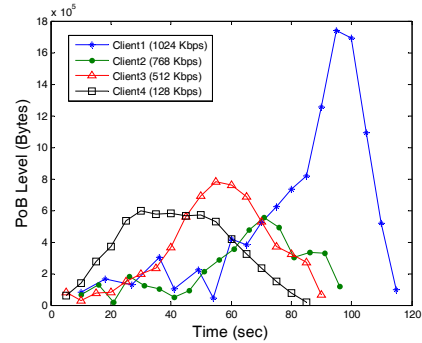


그림 6. 각 클라이언트별 재생버퍼의 상태 (MultiTCP^[5])

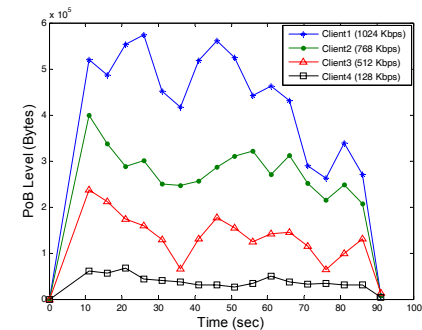


그림 7. 각 클라이언트별 재생버퍼의 상태 (제안 방식)

표 2. 기존 방식들과 제안 방식의 버퍼 언더런에 의한 리버퍼링 횟수 및 재생시간 비교

	버퍼언더런에 의한 리버퍼링 횟수		비디오 콘텐츠 스트리밍에 소요된 시간			
	기존방식	MultiTCP[5]	제안방식	기존 방식	MultiTCP[5]	제안 방식
클라이언트1	4	5	0	99.969	104.969	79.969
클라이언트2	0	2	0	79.969	87.969	79.969
클라이언트3	0	1	0	79.969	82.969	79.969
클라이언트4	0	0	0	79.969	79.969	79.969

은 품질의 스트리밍을 요청한 단말들 (1024Kbps 요청 단말들)이 동일한 길이의 콘텐츠를 시청하는데 20.6~23.8% 가량 더 많은 시간이 걸리는 것을 확

인할 수 있다.

그림 8, 9와 10은 각각 기존 방식과 제안 방식을 사용했을 때 네트워크상에서 측정된 비디오 스트리밍 대역폭의 비교를 나타낸다. 그림 8의 기존 방식에서는 0~30 초 구간까지 4개의 클라이언트들로부터 생성된 4개의 TCP 커넥션들이 서버의 링크 대역폭의 25%씩 공평하게 사용하기 때문에 클라이언트 1과 클라이언트 2는 그림 5에서처럼 버퍼 언더런을 경험하게 된다. 이후 가장 낮은 비트레이트를 요구한 클라이언트 4의 스트리밍이 모두 종료된 후 나머지 3개의 TCP 커넥션은 링크 대역폭의 33%씩 공유하여 사용하게 되고 클라이언트 3의 스트리밍이 종료하고 난 후에 클라이언트 1과 클라이언트 2는 각각 요구한 비트레이트 이상의 스트리밍 대역폭을 확보하여 안정적인 스트리밍을 제공받을 수 있게 된다. 이러한 이유로 인해 표 2에서와 같이 클라이언트 1과 클라이언트 2는 기존 방식에서 스트리밍 시간이 원래의 스트리밍 시간보다 길어지게 되어 스트리밍 품질 저하를 경험하게 된다. MultiTCP [5]를 이용한 시스템에서도 이러한 경향은 그대로 나타나게 된다. 이는 MultiTCP [5] 알고리즘을 사용하는 비디오 스트리밍 서버는 각 이종 단말들에게 동일하게 5개의 TCP 커넥션을 할당하기 때문에 네트워크 혼잡상태, 단말 재생버퍼의 상태 등에 대응할 수 없기 때문이다. 따라서 해당 방식의 그림 9는 기존방식의 각 단말별 대역폭 점유 (그림 8)와 비슷한 양상을 나타내게 되고 이로 인해 그림 6과 표 2에서 확인한 바와 같이 높은 비트레이트의 스트리밍을 요청한 단말이 빈번한 재생버퍼 언더런을 경험하게 된다. 그러나 제안 방식에서는 그림 10에서 나타난 바와 같이 각 클라이언트들은 요청한 비디오 재생 비트레이트와 비슷한 수준으로 스트리밍 대역폭을 점유하기 때문에 모든 클라이언

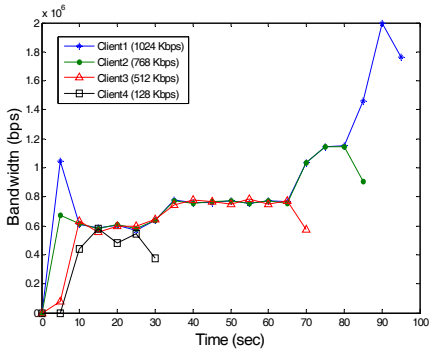


그림 8. 각 클라이언트별 스트리밍 대역폭 (기존 방식)

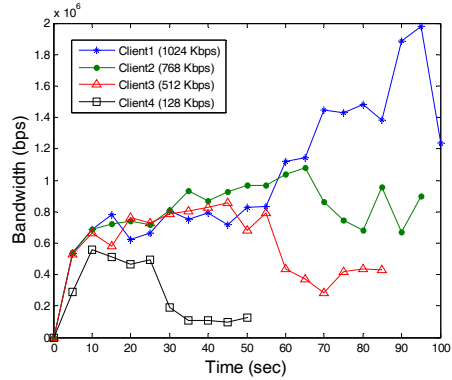


그림 9. 각 클라이언트별 스트리밍 대역폭 (MultiTCP^[5])

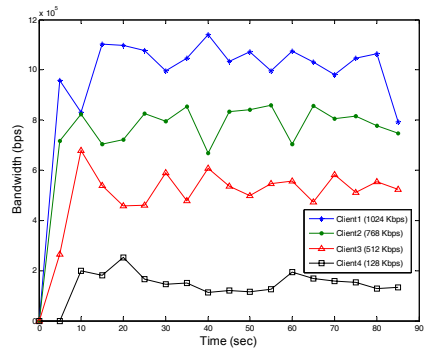


그림 10. 각 클라이언트별 스트리밍 대역폭 (제안 방식)

트들이 안정적인 스트리밍을 제공받을 수 있게 되고 이로 인해 표 2의 결과와 같이 스트리밍 품질의 저하가 발생하지 않게 된다.

V. 결론

본 논문에서는 비디오 스트리밍 서버의 업링크 대역폭에 제한이 있는 일반적인 상황에서 각 클라이언트별로 해상도와 재생버퍼크기가 다른 이종 클라이언트들에 대해 동시에 고품질의 스트리밍을 제공할 수 있는 TCP 스트리밍 방식을 제안하였다. 제안 알고리즘은 각 클라이언트들의 재생버퍼 상태와 요청 대역폭등을 고려하여 적응적으로 다중 TCP를 사용하도록 고안되었다. 실험결과를 통하여 제안 방식을 적용할 경우 네트워크의 대역폭을 최대한 활용할면서 각 클라이언트별로 재생버퍼를 효율적으로 사용할 수 있도록 유도하며 고품질의 스트리밍이 가능한 것을 확인할 수 있었다. 따라서 본 제안 알고리즘은 OS나 네트워크 장비에서의 지원 없이 어플리케이션에서 제어가 이루어지기 때문에 BcN과

같은 통합 서비스망에서 이중 네트워크/클라이언트들을 통해 동시에 비디오 스트리밍을 제공할 경우 고품질 비디오 스트리밍을 위하여 쉽게 적용할 수 있는 장점이 있다.

참 고 문 헌

[1] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia Streaming via TCP: An Analytic Performance Study", In Proc. of ACM Multimedia, Oct. 2004, pp. 908-915

[2] A. C. Dalal, and E. Perry, "A New Architecture for Measuring and Assessing Streaming Media Quality", In PAM workshop 2003, Apr. 2003, pp. 223-231

[3] N. Terada, E. Kawai, and H. Sunahara, "Extracting Client-Side Streaming QoS Information from Server Logs", In IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Aug.2005, pp.621-624.

[4] Eren Gürses, Gozde Bozdagi Akar and Nail Akar, "A simple and effective mechanism for stored video streaming with TCP transport and server-side adaptive frame discard", Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 48 ,Issue 4 2005), pp 489-501

[5] Thinh Nguyen and Sen-ching S. Cheung, "Multimedia Streaming Using Multiple TCP Connections", in IPCCC 2005, April. 2005, pp 215- 223

[6] ns-2 Network Simulator, available from "http://nslam.isi.edu/nslam/index.php/Main_Page", (2007).

[7] C. Ke, C. Lin, C. Shieh, W. Hwang, "A Novel Realistic Simulation Tool for Video Transmission over Wireless Network", IEEE International Conference on SUTC2006, June 5-7, 2006, Taichung, Taiwan

정 영 하 (Young H. Jung)

정회원



1997년 2월 연세대학교 전기공학과 졸업
 1999년 2월 연세대학교 전기공학과(석사)
 2000년 3월~현재 삼성전자 네트워크 사업부 책임연구원
 2006년 3월~현재 연세대학교 전기전자공학과(박사과정)

<관심분야> Video Communication, 3GPP Network

최 윤 식 (Yoonsik Choe)

종신회원



1979년 2월 연세대학교 전기공학과 졸업
 1987년 5월 펜실베이니아 주립대 전기공학 M.S.
 1990년 2월 Purdue 대학 전기공학 Ph.D
 1993년 3월~현재 연세대학교 전기전자공학과 정교수

<관심분야> 영상신호처리, 영상코딩, 멀티미디어 통신 등