

특집논문-05-10-3-04

TCP 친화적인 전송률 제어기법 기반의 효율적인 비디오 스트리밍

이 정 민^{a)}, 이 선 현^{a)}, 정 광 수^{a)*}

Efficient Video Streaming Based on the TCP-Friendly Rate Control Scheme

Jungmin Lee^{a)}, Sunhun Lee^{a)}, and Kwangsue Chung^{a)*}

요 약

최근 비디오나 오디오 스트리밍 서비스와 같은 멀티미디어 트래픽이 증가하고 있다. 이러한 트래픽들은 패킷을 전달하는데 대부분 UDP(User Datagram Protocol)기반의 RTP(Realtime Transport Protocol)를 사용한다. 하지만 UDP기반의 RTP는 기본적으로 혼잡 제어 메커니즘이 없으며 현재 인터넷의 주요 트래픽인 TCP(Transmission Control Protocol)와의 형평성을 보장하지 않는다는 문제점을 갖는다.

본 논문에서는 비디오 스트림의 TCP 친화적인 전송률 조절 기법으로 TF-RTP(TCP-Friendly RTP)를 제안하였다. TF-RTP는 네트워크 상태가 혼잡하여 패킷 손실이 발생할 경우, 개선된 파라미터 산출 방법을 사용하여 경쟁하는 TCP의 전송률을 보다 정확하게 계산하여 전송되는 비디오 스트림의 전송률을 조절함으로써 경쟁하는 TCP 트래픽과 친화적으로 동작하며 네트워크 대역폭을 보다 공평하게 사용하게 된다. 실험을 통해 제안하는 TF-RTP가 TCP의 전송률을 보다 정확하게 계산하며 TCP 트래픽과의 친화성, 형평성 측면에서도 성능 개선을 보임을 확인할 수 있었다.

Abstract

The multimedia traffic of continuous video and audio data via streaming service accounts for a significant and expanding portion of the Internet traffic. This streaming data delivery is mostly based on RTP with UDP. However, UDP does not support congestion control. For this reason, UDP causes the starvation of congestion controlled TCP traffic which reduces its bandwidth share during overload situation.

In this paper, we propose a new TCP-friendly rate control scheme called "TF-RTP(TCP-Friendly RTP)". In the congested network state, the TF-RTP exactly estimates the competing TCP's throughput by using the modified parameters. Then, it controls the sending rate of the video streams. Therefore, the TF-RTP adjusts its sending rate to TCP-friendly and fair share with competing TCP traffics. Through the simulation, we prove that the TF-RTP correctly estimates the TCP's throughput and improves the TCP-friendliness and fairness.

Keywords : Streaming protocol, Rate control, Congestion control

a) 광운대학교 전자공학부

Department of Electronics Engineering, Kwangwoon University

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성 지원사업과 한국과학재단 특정기초연구[R01-2005-0000-10934-0(2005)]의 지원에 의해 수행되었음.

I. 서 론

현재 인터넷에서 FTP(File Transfer Protocol)나 HTTP(Hypertext Transfer Protocol)와 같은 주요 트래픽들은 전

송 프로토콜로 대부분 TCP를 사용하고 있다. TCP는 종단 간(end-to-end)의 신뢰적인 패킷 전달을 수행하며 혼잡 제어 매커니즘(congestion control mechanism)을 통해 송신단의 전송률을 직접 제어하여 네트워크 혼잡 상황으로 발생하는 데이터의 부분별 손실을 막고 네트워크가 안정하게 동작할 수 있도록 한다^[1].

최근들어 비디오나 오디오 스트리밍 서비스와 같은 멀티미디어 트래픽이 증가하고 있다. 이러한 트래픽들은 지연에 민감하고 어느 정도의 패킷 손실을 허용하며 전송률을 기반으로 동작하는 특징을 가진다. 멀티미디어 트래픽의 이러한 본질적인 특징으로 인해 신뢰적인 패킷 전달을 수행하는 TCP는 스트리밍 서비스의 전송 프로토콜로는 적합하지 않다. 따라서 대부분의 멀티미디어 트래픽은 혼잡 제어를 수행하지 않는 UDP를 사용하게 되었다. 하지만 UDP는 TCP와 같은 혼잡 제어 매커니즘이 없으므로 네트워크 혼잡 상황을 야기하여 네트워크를 붕괴(congestion collapse)시킬 가능성이 있으며 기존의 주요 트래픽인 TCP와도 친화적(TCP-friendly)으로 동작하지 않는다는 문제점을 갖는다^{[1][2]}.

1990년대 후반부터 비디오 데이터의 스트리밍 기법과 관련하여 네트워크 안정성을 향상시키면서 기존 인터넷의 주요 트래픽인 TCP와의 친화성을 고려한 혼잡 제어 매커니즘의 적용에 대한 연구가 활발하게 진행되고 있다^[2-9]. 이러한 연구들은 TCP의 평균 전송률 계산을 기반으로, 인터넷에서의 표준 멀티미디어 스트리밍 프로토콜인 RTP의 전송률을 현재 네트워크의 상태에 따라 조절함으로써 네트워크의 안정화 및 TCP 기반 트래픽과의 형평성 보장을 목적으로 하고 있다. 하지만 이러한 연구들은 TCP의 평균 전송률을 계산하는 과정에서 본질적으로 부정확성을 가지며 이러한 문제점을 개선하기 위해 추가적인 알고리즘을 적용하여 비디오 스트림의 전송률을 조절한다는 한계점을 가지고 있다.

본 논문에서는 스트리밍 서비스에서 비디오 스트림을 전달하는 RTP의 전송률을 TCP 친화적이며 공평한 전송률로 조절하는 전송 기법으로 TF-RTP를 제안하였다. TF-RTP에서는 종단간 지연(RTT: Round Trip Time), 재전송 타임아웃(RTO: Retransmission TimeOut), 패킷손실률(packet

loss rate)의 파라미터들을 사용하여 경쟁하는 TCP의 평균 전송률을 보다 정확하게 계산한다. 경쟁하는 TCP의 평균 전송률에 대한 정확한 계산을 바탕으로 제안하는 TF-RTP는 기존 연구들이 가지는 추가적인 알고리즘의 적용없이 네트워크 상태에 따라 비디오 스트림 데이터의 전송률을 적절하게 조절하며 이를 통해 전송되는 비디오 스트림의 전송률 변화를 줄이고, 경쟁하는 TCP 트래픽과의 친화성 및 형평성을 크게 개선하였다.

논문의 2장에서는 비디오 스트림 전송 기법으로서 경쟁하는 TCP 트래픽과의 친화성 및 형평성을 향상을 고려한 기존 RTP 기반의 비디오 스트림 전송 기법에 대한 소개와 함께 문제점을 기술하였으며 3장에서는 기존 연구의 문제점을 검증하고, 본 논문에서 제안하는 TF-RTP의 세부 알고리즘을 소개하였다. 4장에서는 시뮬레이션을 통해 제안하는 TF-RTP의 성능을 평가하였으며 마지막으로 5장에서 결론을 맺었다.

II. 관련 연구

인터넷의 주요 트래픽에 사용되는 전송 프로토콜인 TCP는 비디오나 오디오와 같은 멀티미디어 전송에 적절하지 않으므로 대부분의 멀티미디어 스트리밍에서는 전송 프로토콜로 UDP를 기반으로 하는 RTP를 사용하게 된다. 하지만 혼잡 제어 매커니즘이 없는 UDP기반의 RTP를 전송 프로토콜로 사용함으로써 네트워크의 불안정을 발생시키게 되며 TCP를 사용하는 트래픽과 네트워크 자원을 공평하게 사용하지 못하는 문제점을 가지게 된다. 1990년대 후반부터 이러한 문제점을 개선하기 위해 IETF(Internet Engineering Tasking Force)의 표준 스트리밍 프로토콜인 RTP에 혼잡 제어 매커니즘을 적용한 스트리밍 전송기법에 관한 연구가 활발하게 진행되고 있다. 본 장에서는 RTP에 혼잡 제어 매커니즘을 적용하여 비디오 스트림의 전송률을 TCP 친화적으로 조절하는 것을 목적으로 가지는 기존의 연구들을 설명하고 기존 방법들이 가지는 문제점에 대해서 분석하도록 한다.

1. TCP 전송률 모델링 공식

Padhye는 [10]에서 TCP Reno를 기반으로 TCP 플로우의 평균 전송률을 중단간 지연과 재전송 타임아웃, 패킷 손실률과 같은 파라미터들을 사용하여 수학적 공식으로 모델링하였다.

$$T = \frac{S}{t_{RTT}\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1 + 32p^2)} \quad (1)$$

식 (1)은 Padhye가 제안한 TCP 플로우의 평균 전송률을 구하는 공식을 보이고 있다. T는 TCP 플로우의 일정 구간에서의 평균 전송률을 나타내며, t_{RTT} 는 비디오 스트림이 전달되는 네트워크에서의 중단간 지연을, S는 전송되는 비디오 데이터의 패킷 크기를 의미한다. p는 패킷 손실률을 의미하며 t_{RTO} 는 TCP가 가지고 있는 재전송 타이머의 타임아웃 값을 의미한다. 공식에 따라, TCP 플로우의 평균 전송률은 중단간 지연과 패킷 손실률에 의해 계산되며, 여기에서 패킷 손실률은 일정 구간에서 전체 보내진 패킷의 수에 대한 손실된 패킷의 비율로 계산된다^[10].

Padhye의 TCP 전송률 모델링 공식은 현재 네트워크의 상태에서 경쟁하는 TCP 플로우의 평균 전송률을 비교적 정확하게 계산할 수 있다는 장점을 가진다. 이러한 장점으로 인해 TCP 친화적으로 비디오 스트림의 전송률을 조절하고자 하는 관련 연구들에서 빈번하게 사용되고 있으며, 대표적으로 Padhye가 제안한 TFRCP(TCP-Friendly Rate Control Protocol)와 Floyd가 제안한 TFRC가 있다^{[4][5]}. TFRC는 일정 주기마다 식 (1)의 공식에 의해 TCP 친화적인 전송률을 계산하여 비디오 스트림의 전송률을 결정하는 것으로 중단간 지연이나 패킷 손실률의 변화에 따라 전송률의 변화가 심하게 발생한다는 단점을 갖는다. TFRC는 전송률 변화가 심하게 발생하는 TFRCP의 단점을 개선하기 위해, 하나의 RTT 구간에서 한번의 패킷 손실만을 인식하여 패킷 손실률을 계산하며, 인접한 최근 8개의 각 RTT 구간에서의 패킷 손실률에 EWMA(Exponentially

Weighted Moving Average) 기법을 적용하여 패킷 손실률을 계산함으로써 전송률의 변화를 줄이게 된다. 하지만 이러한 방법들은 전송 계층의 새로운 프로토콜로 제안된 것으로 현실적으로 사용하기 어렵다는 한계를 갖는다.

2. LDA 기법

Sisalem은 2.1절에서 설명한 Padhye의 TCP 전송률 모델링 공식을 표준 스트리밍 프로토콜인 RTP의 전송률 조절에 사용하는, RTP/RTCP 모델 기반의 비디오 스트림 전송 기법으로 LDA(Loss Delay-based Adaptation)를 제안하였다^[8]. LDA는 RTCP의 제어 메시지를 이용하여 TCP의 평균 전송률을 계산하는데 사용되는 파라미터인 중단간 지연, 패킷 손실률, 재전송 타임아웃을 계산한다^[11]. RTCP의 제어 메시지로부터 얻은 세 가지 파라미터들은 식 (1)의 공식에 적용되어 동일한 네트워크 환경에서 대역폭을 경쟁하는 TCP 플로우의 평균 전송률을 계산한다. 계산된 TCP의 평균 전송률은 LDA에서 전송하는 비디오 스트림의 전송률 조절에 사용되며, 이를 통해 경쟁하는 TCP 플로우와의 친화성 및 형평성을 향상시키게 된다.

If, $p > 0$: Congestion State

$$T_m = \text{MAX}(T_{m-1} \times (1 - \sqrt{p}), T_{TCP})$$

그림 1. LDA의 전송률 조절 기법
Fig 1. Rate control scheme in LDA

LDA는 그림 1과 같은 전송률 조절 기법을 이용하여 경쟁하는 TCP 플로우와 친화적이고 공평하게 네트워크 대역폭을 공유하도록 비디오 스트림의 전송률을 조절하게 된다. 패킷 손실이 발생하지 않을 경우, 즉 $p=0$ 이면, 네트워크 상태는 안정한 상태로 판단되며, 이러한 경우에는 일반적인 TCP와 마찬가지로 전송률을 선형적으로 증가시키게 된다. 패킷 손실이 발생할 경우, 즉 $p>0$ 이면, 네트워크 상태는 혼잡 상태로 판단되며, 이러한 경우에는 네트워크의 안정성 및 TCP와의 친화성을 고려하여 전송률을 감소시켜 나

가게 된다. RTCP 제어 메시지를 통해 얻어진 중단간 지연, 패킷 손실률, 재전송 타임아웃과 같은 파라미터들을 식 (1)에 대입하여 LDA는 경쟁하는 TCP의 평균 전송률인, T_{TCP} 를 계산할 수 있다. 하지만 T_{TCP} 를 그대로 비디오 스트림의 전송률로 적용할 경우, 전송률의 변화가 크게 발생하는 문제점을 갖는다. 이러한 문제점을 해결하기 위해 LDA는 패킷 손실률의 제공근에 반비례하게 전송률을 감소시키는 TCP의 성질을 이용하여, 그림 1과 같이 두 가지 전송률 성분중에서 큰 값으로 다음 주기의 전송률을 조절하게 된다.

LDA는 TCP 친화적인 전송률 조절을 위해 2.1절에서 설명한 Padhye의 공식을 사용한다. 하지만 네트워크 혼잡 상태에서, Padhye 공식의 산출값을 그대로 사용할 경우, 전송률의 변화가 크게 발생하는 문제점을 가지며, 이러한 문제점을 개선하기 위해서 추가적인 알고리즘을 사용하게 된다. 추가적인 알고리즘을 사용한다는 것은, LDA가 경쟁하는 TCP의 평균 전송률을 정확하게 계산할 수 없기 때문이며, 이것은 RTCP 제어 메시지를 통해 얻은 중단간 지연, 패킷 손실률, 재전송 타임아웃과 같은 파라미터들의 값이 부정확함을 의미한다.

3. SRTP 기법

2.2절의 LDA와 유사한 방법으로 Song이 제안한 SRTP(Smart RTP)가 있다. SRTP는 표준 스트리밍 프로토콜인 RTP에 전송률 조절 기법을 적용한 비디오 스트림 전송 기법으로서, 네트워크 혼잡 상태를 회피하고 TCP와 친화적으로 동작할 수 있도록 하며 네트워크 상태변화에 따른 전송률의 변화폭을 줄임으로써 사용자에게 서비스되는 비디오의 품질 변화를 줄이는 것을 목적으로 갖는다^[9].

SRTP는 LDA와 마찬가지로 RTP의 제어 프로토콜인 RTCP의 정보를 이용하여 중단간 지연, 패킷 손실률, 재전송 타임아웃의 파라미터들을 계산하고 이를 2.1절의 Padhye의 공식에 적용하여 경쟁하는 TCP와 친화적이며 공평한 전송률을 계산한다.

$$p = \frac{N_{real}}{(N_{max} - N_{first})} \quad (2)$$

$$t_{SRTT} = (\alpha \times t_{CurrentRTT}) + (1 - \alpha) \times t_{SRTT} \quad (3)$$

식 (2)는 SRTP에서 사용하는 파라미터인 패킷 손실률을 계산하는 공식으로, 분모의 N_{max} 는 가장 최근에 수신된 패킷의 최대 시퀀스 번호를 의미하며, N_{first} 는 최초로 수신된 패킷의 시퀀스 번호를 의미한다. 분자의 N_{real} 은 현재까지 수신된 전체 패킷의 수를 의미한다. 식 (3)은 중단간 지연의 변화 정도를 줄이기 위해 저대역 필터링 기법을 적용해서 SRTT(Smoothed RTT)를 구하는 식으로 일반적인 TCP와 유사하게 가중치, α 는 0.8로 설정하게 된다.

RTCP 제어 메시지를 통해 SRTP는 네트워크에서 패킷 손실이 발생했음을 알게 된다. 즉, $p > 0$ 이면, 네트워크 상태는 혼잡 상태로 판단되며, SRTP는 식 (2), (3)을 통해 얻어진 파라미터들을 Padhye의 공식에 적용하여 TCP 친화적이며 공평한 전송률을 계산한다. Padhye의 공식에 의해 계산된 전송률은 네트워크 혼잡 상태 회피와 TCP 친화성 및 공평성을 고려하여 그림 2과 같은 방법으로 비디오 스트림의 전송률을 조절하게 된다. 패킷 손실이 발생하지 않은, 즉 네트워크가 안정한 상태로 판단될 경우, SRTP는 일반적인 TCP와 유사하게 전송률을 선형적으로 증가시킨다.

2.2절의 LDA와 마찬가지로 SRTP는 Padhye의 공식에 의

$$\begin{aligned} & \text{If, } p > 0 : \text{Congestion State} \\ & \text{If, } (T_{S-RTP} > T_{TCP}) \\ & \text{New } T_{S-RTP} = \text{MAX} (\beta \times T_{S-RTP} + (1 - \beta)(T_{S-RTP}(1 - \sqrt{p})), T_{TCP}) \end{aligned}$$

그림 2. SRTP의 전송률 조절 기법

Fig 2. Rate control scheme in SRTP

해 산출된 값으로 전송률을 직접 조절하지 않는다. 이는 LDA와 유사하게 RTCP 제어메시지 정보를 기반으로 계산된 중단간 지연, 패킷 손실률, 재전송 타임아웃과 같은 파라미터들이 부정확하게 계산되기 때문이다. SRTP에서는 이를 보완하고 비디오 스트리밍에서 요구되는 부드러운 전송률 변화에 대한 요구사항을 만족시키기 위해서 그림 2와 같이, Padhye 공식의 계산값과 저대역 필터링 기법을 적용한 성분값 사이의 최대값으로 다음 주기의 전송률을 결정하게 된다.

Ⅲ. TF-RTP 알고리즘

본 장에서는 2장에서 지적한 기존의 TCP 친화적인 비디오 스트림 전송 기법들이 가지는 문제점을 간단한 실험을 통해서 검증하며, 이러한 문제들을 개선하기 위해서 본 논문에서 새롭게 제안한 TF-RTP(TCP-Friendly RTP)의 알고리즘에 대해 기술한다. 제안하는 TF-RTP 전송 기법은 기존 연구들에서 문제점으로 지적된, Padhye의 공식에 사용되는 파라미터들의 부정확성을 개선하여 파라미터 값들과 이를 이용한 TCP의 평균 전송률을 보다 정확하게 계산하게 된다. 이를 이용하여 전송되는 비디오 스트림의 전송률을 추가적인 알고리즘 적용없이 TCP 플로우와 친화적으로 조절하게 된다.

1. TF-RTP 개요

기존의 TCP 친화적인 비디오 스트림 전송 기법들은 경쟁하는 TCP의 평균 전송률을 계산하기 위해 식 (1)의 Padhye의 공식을 사용하였다. 하지만 Padhye의 공식에 사용되는 파라미터들의 부정확성에 의해 공식의 산출값은 실제 TCP의 평균 전송률에 비해 작은 값으로 계산되며 전송률의 변화가 크게 발생한다는 문제점을 갖는다. 이러한 문제점을 개선하기 위해 기존의 연구들은 추가적인 알고리즘을 적용하여 비디오 스트림의 전송률 변화를 줄이고자 하였다.

본 논문에서는 이러한 기존 연구들의 근본적인 문제점을 해결하기 위해 개선된 파라미터 계산을 통한 TCP 친화적인 비디오 스트림 전송 기법으로 TF-RTP를 제안하였다. TF-RTP는 RTCP의 제어 메시지를 이용하여 중단간 지연이나 패킷 손실률의 파라미터를 개선된 방법으로 계산하고 이를 기반으로 경쟁하는 TCP 플로우의 평균 전송률을 보다 정확하게 계산한다. 계산된 TCP 친화적인 전송률은 추가적인 알고리즘 적용없이 비디오 스트림의 전송률을 직접적으로 조절한다.

그림 3은 제안하는 TF-RTP의 중단간 구조를 나타낸 것이다. TF-RTP는 기존 연구들과 마찬가지로 표준 스트리밍 프로토콜인 RTP/RTCP 모델을 기반으로 한다. RTCP의 제

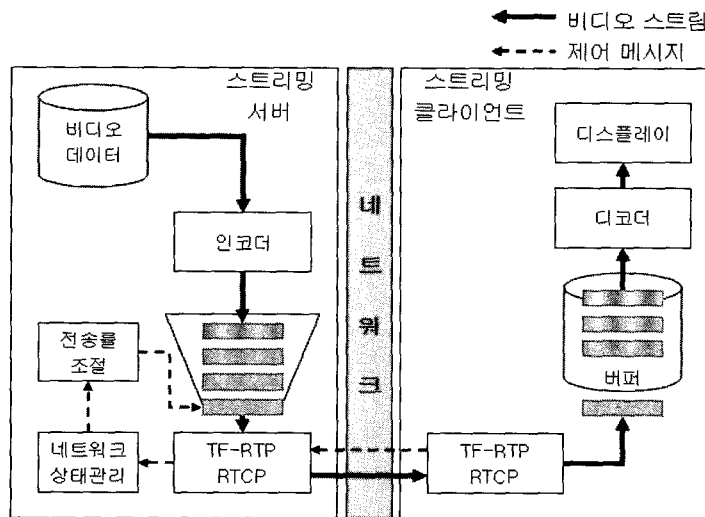


그림 3. TF-RTP의 중단간 구조
Fig 3. End-to-end architecture in TF-RTP

어 메시지를 이용하여 스트리밍 클라이언트는 종단간 지연이나 패킷 손실률등의 네트워크 상태정보를 주기적으로 스트리밍 서버에 보고한다. 클라이언트에서 보고된 RTCP 제어 메시지의 정보를 바탕으로 스트리밍 서버는 네트워크상태를 혼잡상태나 안정상태로 판단하게 되며, 각각의 상태에 따라서 전송되는 비디오 스트림의 전송률을 증가시키거나 감소시키게 된다.

이후 3.2절에서는 비디오 스트림의 전송과 관련하여 기존의 TCP 친화적인 전송률 조절 기법들이 가지는 문제점을 간단한 실험을 통해 분석하며 3.3절에서는 기존 연구들의 문제점을 개선하는 TF-RTP의 알고리즘에 대해 상세하게 설명한다. 그리고 3.4절에서는 TF-RTP의 개선된 파라미터 계산 방법을 기반으로 TF-RTP가 가지는 전송률 조절 방법에 대해 설명하도록 한다.

2. 기존 연구의 문제점 검증

네트워크 안정성 향상과 TCP 친화적인 전송률 조절을 목적으로 가지는 LDA나 SRTP는 2장에서 설명하였듯이

Padhye의 TCP 전송률 모델링 공식을 사용한다. 하지만 공식의 산출값을 이용하여 직적으로 비디오 스트림의 전송률을 조절할 경우, 경쟁하는 TCP에 비해 상대적으로 작은 전송률을 가지며 전송률의 변화도 크다는 문제점을 갖는다. 이러한 문제점을 보완하기 위해 LDA나 SRTP에서는 추가적인 알고리즘을 사용하여 이러한 문제점을 보완하게 된다.

기존의 LDA나 SRTP가 추가적인 알고리즘을 적용하는 근본적인 원인은 사용하는 Padhye 공식의 산출값이 부정확하기 때문이다. 이러한 부정확성은 LDA나 SRTP에서 RTCP 제어 메시지를 기반으로 얻게 되는 종단간 지연이나 패킷 손실률과 같은 파라미터의 정보가 정확하지 않기 때문이다.

그림 4는 기존 연구의 문제점을 검증하기 위해 수행한 실험 결과를 나타낸 것으로 10Mbps의 대역폭에서 TCP 플로우와 비디오 스트림을 전달하는 RTP 플로우를 경쟁시킨 결과이다. TCP 플로우에서는 혼잡 윈도우의 변화와 그에 따른 전송률 변화를 나타내었으며 RTP 플로우에서는 RTCP 제어 메시지의 정보로부터 얻은 종단간 지연의 변화와 LDA나 SRTP에서 계산되는 Padhye 공식의 산출값을

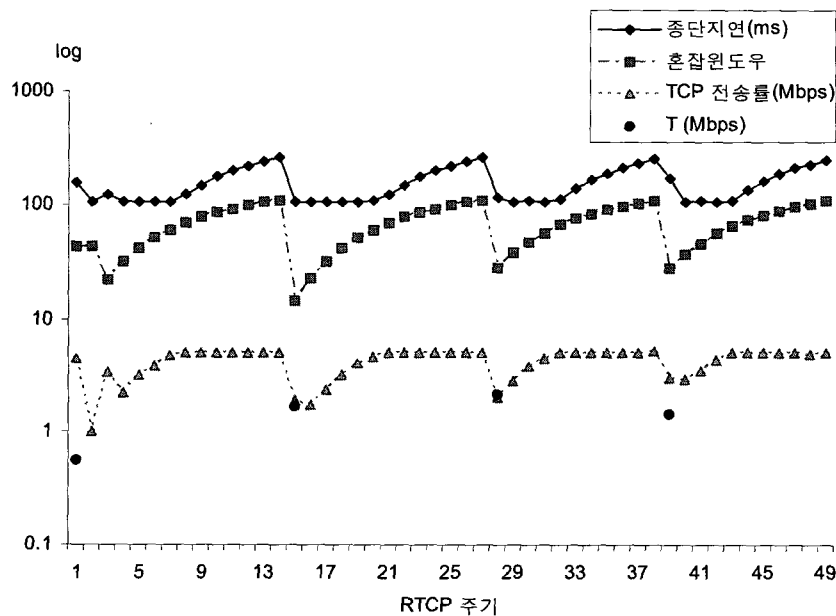


그림 4. LDA와 SRTP에서의 문제점 1

Fig 4. Problem 1 in the LDA and SRTP

나타내었다. 초기에 발생한 패킷 손실을 제외하면, 패킷 손실은 14, 28, 39번째 RTCP 주기에 발생한다. 패킷 손실이 발생한 시점에서의 종단간 지연, 패킷 손실률, 그리고 Padhye 공식의 산출값은 다음과 같다.

- 14번째 RTCP 주기 : 종단간 지연=0.105ms, 패킷 손실률=0.003, T=1.69Mbps
- 28번째 RTCP 주기 : 종단간 지연=0.116ms, 패킷 손실률=0.0015, T=2.17Mbps
- 39번째 RTCP 주기 : 종단간 지연=0.174ms, 패킷 손실률=0.0015, T=1.447Mbps

실험 결과를 통해 기존의 TCP 친화적인 비디오 스트림 전송 기법인 LDA나 SRTP에서 경쟁하는 TCP의 평균 전송률 계산, T는 실제 측정된 TCP 플로우의 평균 전송률인 4.255Mbps에 비해 작은 값으로 계산되는 것을 확인할 수 있으며 전송률의 변화폭이 크게 발생하는 것을 확인할 수 있다. 이것은 그림 4에서 볼 수 있듯이 패킷 손실이 발생하는, 즉 Padhye의 공식을 이용해서 TCP 친화적인 전송률을

계산하는 시점에서의 종단간 지연이 상대적으로 큰 값을 가지기 때문이다. 또한 식 (2)와 같은 패킷 기반의 패킷 손실률 계산을 사용할 경우, 패킷 손실률이 상대적으로 큰 값으로 계산되며 변화가 크게 발생하는 단점 때문이다.

그림 5는 그림 4의 실험을 확장하여 TCP 플로우 수를 추가하여 실험한 결과이다. 그림 4의 결과와 마찬가지로 Padhye의 공식에 의한 TCP 친화적인 전송률 계산은 평균 1.67Mbps로 경쟁하는 TCP의 평균 전송률인 3.0Mbps에 비해 매우 작은 값으로 계산된다. 또한 39, 40번째 RTCP 주기에서 확인할 수 있듯이 연속적인 패킷 손실이 발생할 경우, 기존의 LDA나 SRTP의 Padhye 공식의 산출값은 변화가 크게 발생한다. 이것은 다른 응용서비스와는 다르게 부드러운 전송률 변화를 요구하는 비디오 스트림 전송에 있어 부정적인 영향을 미치게 된다.

3. 개선된 파라미터 산출

3.2절에서는 기존의 TCP 친화적인 전송률 조절 기법이 가지는 문제점을 간단한 실험 결과를 통해서 검증하였다.

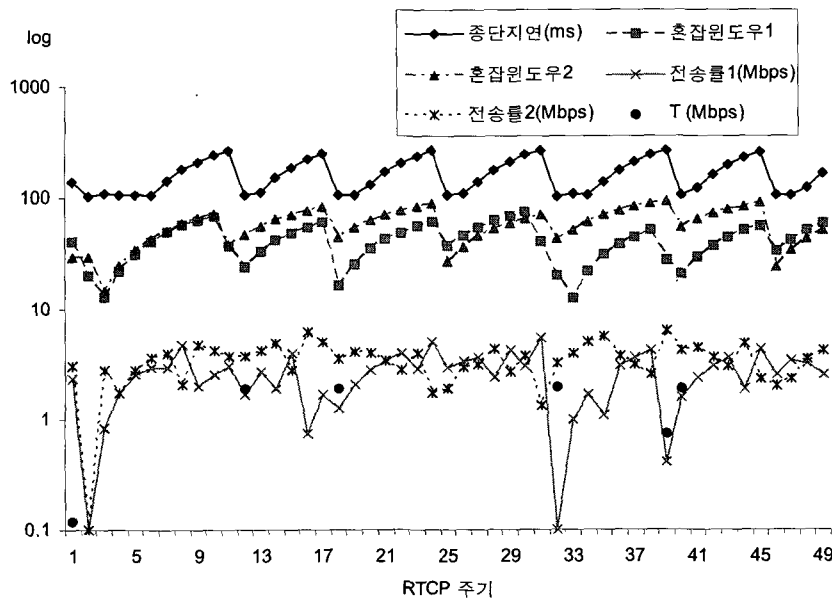


그림 5. LDA와 SRTP에서의 문제점 2
Fig 5. Problem 2 in the LDA and SRTP

실험 결과를 통해, 기존 연구인 LDA나 SRTP에서 TCP 친화적인 비디오 스트림의 전송률 조절을 위해 추가적인 알고리즘을 적용하는 근본적인 원인이 RTCP 제어 메시지를 통해서 얻게 되는 종단간 지연이나 패킷 손실률과 같은 파라미터의 부정확성에 있다는 것을 확인할 수 있었다. 본 절에서는 기존 연구들의 근본적인 문제점을 개선하기 위해 RTCP 제어 메시지를 사용하여 보다 정확한 파라미터 값을 얻기 위한 방법을 설명한다. 본 논문에서 제안하는 TF-RTP는 Padhye의 공식에 적용되는 파라미터들의 정확한 값을 얻기 위해 다음의 방법을 사용한다.

- i) 패킷 손실 이벤트 기반의 패킷 손실률 계산
- ii) 전송률 변화를 줄이기 위한 필터링된 종단간 지연 (smoothed RTT) 계산
- iii) TCP와 유사한 재전송 타임아웃 계산

2장에서 설명했듯이 LDA나 SRTP에서 사용하는 RTCP 제어 메시지의 패킷 손실률 정보는 식 (2)에서와 같이 패킷 수 기반의 패킷 손실률에 대한 정보를 제공한다. 패킷 수 기반의 패킷 손실률, p 는 3.2절에서 설명하였듯이 상대적으로 큰 값으로 결정되므로 실제 TCP의 평균 전송률에 비해 작은 값으로 TCP 친화적인 전송률을 계산하게 되며, 연속적인 패킷 손실이 발생할 경우, 전송률의 변화에 큰 영향을 끼치게 된다. TF-RTP에서 패킷 손실률, p 는 이러한 단점을 개선하기 위해 식 (4)와 같이 패킷 수 기반이 아닌 패킷 손실 이벤트 기반으로 계산된다. 패킷 손실 이벤트 기반의 패킷 손실률 계산은 하나 이상의 패킷 손실이 발생하더라도 하나의 패킷 손실 이벤트로 간주하므로 패킷 수 기반의 방법에 비해 패킷 손실률이 작고, 손실률의 변화도 작아진다. 이와같은 방법으로 비디오 스트림의 전송률을 안정하게 변화시킬 수 있으며 경쟁하는 TCP와의 형평성을 향상시킬 수 있다.

$$p = \frac{\text{loss event}}{\# \text{ of received packets} / \# \text{ of loss event}} \quad (4)$$

그림 6은 기존의 패킷 수 기반의 패킷 손실률 계산 방법

과 TF-RTP에서 사용하는 이벤트 기반의 패킷 손실률 계산 방법에 대한 비교이다. RTCP 주기마다 10개의 데이터 패킷이 전송된다는 가정에서, 임의의 패킷 손실 발생을 ‘x’로 표시하였으며 각각의 패킷 손실에 따른 두 가지 방법의 패킷 손실률 변화를 결과로 나타내었다. 결과를 통해 확인할 수 있듯이 이벤트 기반의 패킷 손실률 계산 방법은 기존의 패킷 수 기반의 패킷 손실률 계산에 비해 작은 값을 가지며 패킷 손실률의 변화도 크게 줄어드는 것을 볼 수 있다. 이러한 이벤트 기반의 패킷 손실률을 Padhye의 공식에 적용할 경우, 경쟁하는 TCP와의 형평성이 향상되는 동시에 전송률의 변화폭이 줄어드는 긍정적인 효과를 기대할 수 있다.

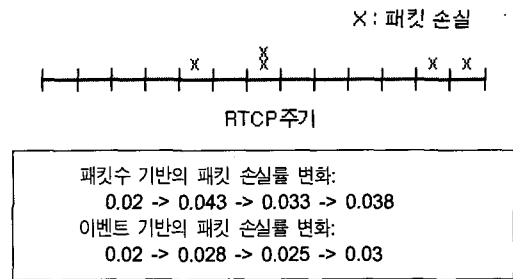


그림 6. 패킷 손실률 계산 방법 비교

Fig 6. Comparison between packet-based p and event-based p

TF-RTP의 평균 전송률 계산에 사용되는 종단간 지연과 재전송 타임아웃 파라미터는 TCP와 유사한 방법을 적용하여 보다 정확하고, 변화폭이 크지 않도록 계산한다. 종단간 지연의 경우, 지연의 변화 크기를 줄이기 위해 기존의 방법과 유사하게 식 (5)와 같은 저대역 필터링을 사용한다. 하지만 3.2절의 실험 결과를 통해 확인하였듯이 한번의 저대역 필터링 적용은 하나의 RTCP 구간 안에서는 변화 정도가 작지만 각 RTCP 구간마다의 종단간 지연은 비교적 크게 변화할 가능성이 있으므로 이러한 종단간 지연의 변화 정도를 줄이기 위해 각 RTCP 주기마다 보고된 종단간 지연에 대해 다시 한번 저대역 필터링을 적용시킨다. 즉, RTCP를 보고하는 수신단에서 한번의 필터링을, 그리고 각 RTCP를 수신하는 송신단에서 다시 한번 필터링을 적용함으로써 종단간 지연의 변화를 최소화하게 된다. 재전송 타임아웃 파라미터는 식 (6)과 같이 계산된

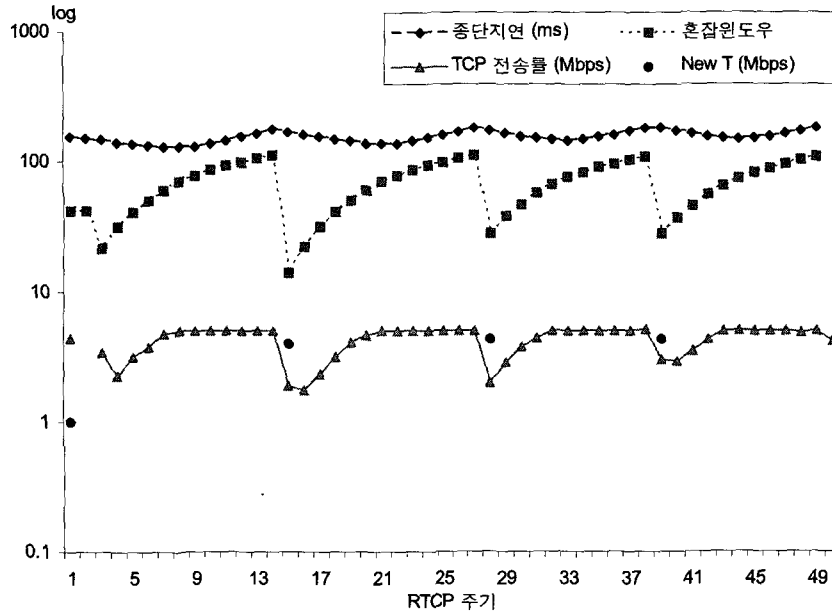


그림 7. 개선된 TCP 전송률 계산 1
Fig 7. Throughput estimation 1 in the TF-RTP

다. 일반적인 TCP에서 계산하는 방법과 같이 현재의 종단간 지연에, 이전의 종단간 지연에 대한 편차 성분을 합하여 계산한다^[12].

$$t_{SRTT} = (\alpha \times t_{CurrentRTT}) + (1 - \alpha) \times t_{SRTT} \quad (5)$$

$$t_{RTO} = t_{SRTT} + 4 \times SRTT_{Variation} \quad (6)$$

그림 7은 앞서 설명한 TF-RTP가 가지는 개선된 파라미터 계산 방법을 사용하여 3.2절에서의 성능 검증 실험을 수행한 것으로 TF-RTP의 개선된 파라미터 계산 방법을 통해 기존의 LDA나 SRTP가 가지는 문제점이 개선될 수 있음을 확인하는 실험 결과이다. 결과를 통해 확인할 수 있듯이 TF-RTP가 가지는 개선된 파라미터 계산은 기존의 방법에 비해 종단간 지연에 대한 저대역 필터링을 두 번 적용함으로써 해당 파라미터 값의 변화가 크게 줄어들었음을 볼 수 있다. 또한 이벤트 기반의 패킷 손실률을 계산함으로써 경쟁하는 TCP 플로우의 평균 전송률에 근사한 값으로 Padhye 공식의 결과값을 산출하게 된다. 초기 상태에서의

패킷 손실을 제외한 14, 28, 39번째 패킷 손실에서의 개선된 파라미터 값과 이것을 적용하여 산출한 Padhye 공식의 전송률 계산은 아래와 같았으며, 경쟁하는 TCP의 평균 전송률은 4.23Mbps, Padhye 공식의 전송률 계산은 4.17Mbps로 측정되었다.

- 14번째 RTCP 주기 : 종단간 지연=0.167ms, 패킷 손실률=0.0002, T=3.94Mbps
- 28번째 RTCP 주기 : 종단간 지연=0.171ms, 패킷 손실률=0.0002, T=4.32Mbps
- 39번째 RTCP 주기 : 종단간 지연=0.178ms, 패킷 손실률=0.0002, T=4.25Mbps

그림 8은 그림 5와 마찬가지로 그림 7의 실험에 TCP 플로우를 추가하여 확장한 실험 결과를 나타낸다. 종단간 지연은 그림 7에서와 같이 개선된 파라미터 계산을 사용하여 기존의 방법에 비교하여 보다 부드러운 변화를 나타내는 것을 확인할 수 있으며, 새로운 파라미터를 적용한 Padhye 공식의 계산값 역시 그림 7의 실험 결과와 유사하게 경쟁하는 TCP의 평균 전송률인 3.11Mbps와 근사한 3.03Mbps로

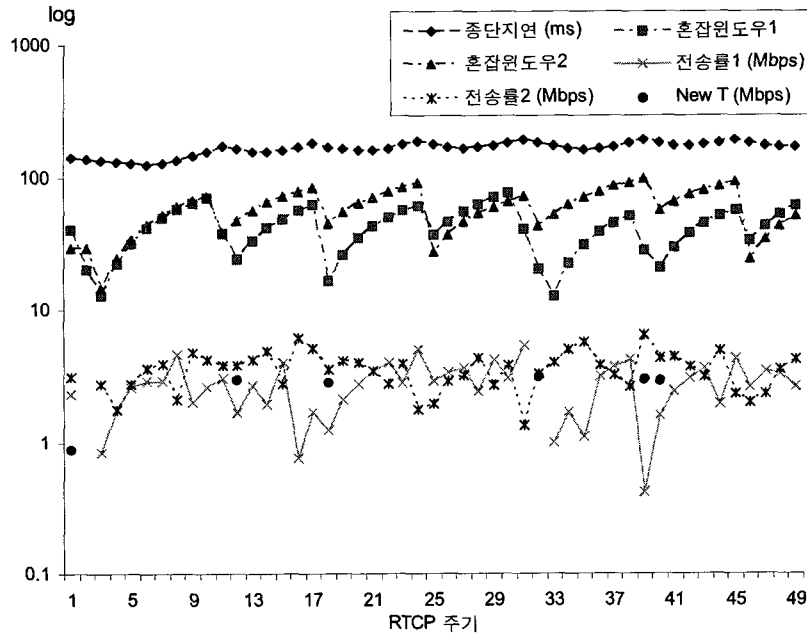


그림 8. 개선된 TCP 전송률 계산 2
Fig 8. Throughput estimation 2 in the TF-RTP

계산되었다. 또한 그림 5에서 확인하였듯이 기존의 방법의 경우, 연속적인 패킷 손실이 발생하면, 전송률의 변화가 크게 발생하는 문제점을 가지고 있었으나 TF-RTP의 개선된 파라미터를 이용한 Padhye 공식의 계산에서는 이벤트 기반의 패킷 손실률 계산과 중단간 지연에 대한 변화를 줄임으로써 그림 8의 실험 결과와 같이 전송률의 변화가 크게 줄어들었음을 확인할 수 있다.

3.2절의 실험 결과와 그림 7, 그림 8의 실험 결과 비교를 통해 기존의 연구들은 경쟁하는 TCP의 평균 전송률을 정확하게 계산할 수 없으며 이것은 Padhye의 공식에 적용하는 파라미터들의 부정확성에 근거한다는 것을 확인하였다. 하지만 제안하는 TF-RTP는 기존 연구의 근본적인 문제점을 개선함으로써 TCP 친화적인 전송률, New T가 경쟁하는 TCP 플로우의 평균 전송률과 거의 정확하게 계산할 수 있음을 확인할 수 있다.

2. TF-RTP의 전송률 조절 알고리즘

TF-RTP는 RTCP 제어 메시지의 패킷 손실률 정보를 이

용하여 네트워크 상태를 판단하게 된다. 즉, 패킷 손실률이 0보다 크면 네트워크를 혼잡 상태로 판단하게 되며, 패킷 손실이 발생하지 않으면, 네트워크를 안정한 상태로 판단하게 된다. 경쟁하는 TCP와 친화적인 전송률 조절을 위해 TF-RTP는 네트워크가 안정한 상태일 경우, TCP의 전송률 조절 기법인 AIMD(Additive Increase Multiplicative Decrease) 기법과 유사하게 중단간 지연시간마다 하나의 패킷만큼 전송률을 선형적으로 증가시킨다^[13]. 하지만 네트워크 상태가 혼잡 상태로 판단될 경우, 3.3절에서 설명한 개선된 파라미터를 사용하여 계산된 전송률로 비디오 스트림의 전송률을 조절한다.

그림 9는 제안하는 TF-RTP에서 비디오 스트림의 전송률 조절 알고리즘을 보여준다. 패킷 손실이 발생한, $p > 0$ 인 경우, TF-RTP는 패킷 손실 이벤트에 기반한 패킷 손실률을 계산하고 저대역 필터링을 적용하여 부드럽게 변화하는 중단간 지연과 그에 따른 재전송 타임아웃 파라미터를 계산한다. LDA나 SRTP의 기존 연구들과 달리, TF-RTP는 보다 정확하게 계산된 파라미터 값을 식 (1)에 대입하여 경쟁하는 TCP 플로우의 평균 전송률을

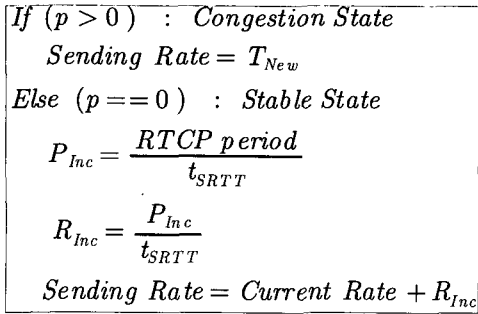


그림 9. TF-RTP의 전송률 조절 기법
 Fig 9. Rate control scheme in TF-RTP

보다 정확하게 계산하므로 이렇게 산출된 값을 다음 주기의 전송률로 추가적인 알고리즘 적용없이 직접적으로 적용하게 된다. 패킷 손실이 발생하지 않은 경우, TF-RTP는 네트워크 상태를 안정한 상태로 판단하고 일반적인 TCP와 마찬가지로 전송률을 선형적으로 증가시키게 된다. 기존의 TCP에서 전송률의 선형적인 증가는 하나의 종단간 지연마다 하나의 패킷만큼 증가시킨다. RTP/RTCP 모델 기반의 TF-RTP는 전송률의 조절주기가 종단간 지연이 아닌 하나의 RTCP 주기가 되므로 TCP의 전송률 조절주기에 맞추어 TF-RTP 플로우의 전송률을 조절할 필요가 있다. 따라서 하나의 RTCP 구간

에서 증가시킬 패킷의 수인, P_{Inc} 를 우선적으로 구하고, 이를 종단간 지연, t_{SRTT} 로 나누어 전송률 증가성분, R_{Inc} 를 구하며 이렇게 얻어진 전송률 증가성분을 현재의 전송률에 더함으로써 다음 주기에서의 TF-RTP 플로우의 전송률을 조절하게 된다.

IV. 실험 및 성능 평가

본 장에서는 TCP 친화적인 비디오 스트림의 전송률 조절 기법으로 본 논문에서 새롭게 제안한 TF-RTP(TCP-Friendly RTP)의 성능 평가를 위해 LBNL(Lawrence Berkely National Laboratory)의 ns-2(network simulator)를 사용하여 다양한 실험을 수행하였다^[14].

1. 실험 환경

제안한 TF-RTP의 성능을 평가하기 위해서 그림 10과 같은 실험 환경을 구성하여 성능 실험을 수행하였다. TF-RTP의 성능 분석을 위해 배경 트래픽으로 TCP 트래픽을 사용하였다. TCP를 배경 트래픽으로 사용하는 이유는, 비디오 스트림 데이터가 전달되는 인터넷 환경에서 대부분

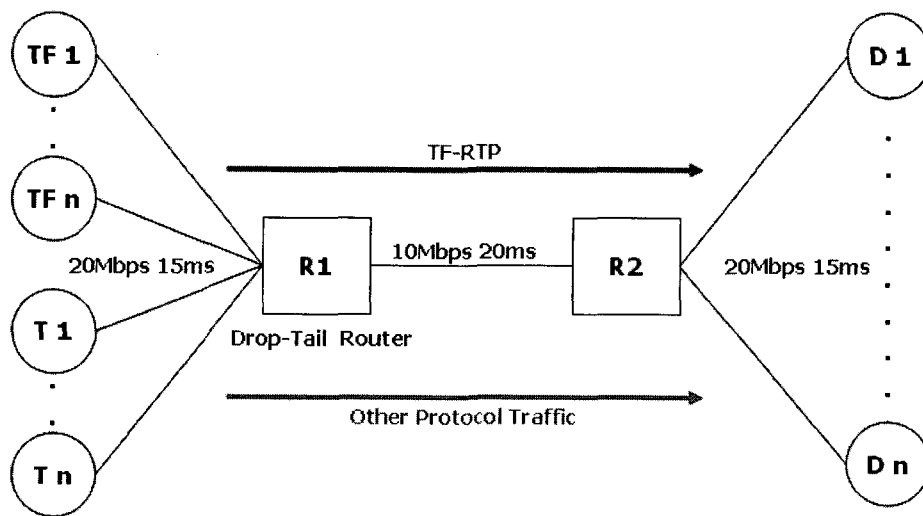


그림 10. 실험 환경
 Fig 10. Simulation environment

의 주요 트래픽들이 TCP를 기반으로 동작하기 때문이며, 또한 TF-RTP의 목적이 경쟁하는 TCP 플로우에 친화적인 전송률 조절과 형평성 향상이기 때문이다.

TF-RTP의 TCP 친화적인 전송률 조절 성능을 평가하기 위해 비디오 스트림을 전달하는 TF-RTP 플로우와 데이터 스트림을 전달하는 TCP 플로우의 전송률 변화와 함께 평균 전송률을 비교하였다. 추가적으로 경쟁하는 TCP 플로우와의 형평성에 대한 성능 평가를 위해 형평성 지수를 통해 성능을 분석하였다.

2. TF-RTP의 성능 검증

그림 11은 그림 10의 실험 환경에서 하나의 TCP 플로우와 하나의 TF-RTP 플로우를 경쟁시켰을 때의 실험 결과를 보여준다. 사용되는 TF-RTP 플로는 3장에서 설명한 개선된 파라미터들을 사용하여 경쟁하는 TCP 플로우의 평균 전송률을 계산하고, 이것을 TF-RTP의 전송률 조절 알고리즘에 사용한다. 초기 상태에서 TF-RTP는 세션 초기의 불안정성에 의해 높은 패킷 손실률을 가지며, 그로 인해 TCP의 평균 전송률을 상대적으로 작은 값으로 측정하여 전송

률을 조절한다. 하지만 초기 상태가 지난 일정 시간 이후부터는 경쟁하는 TCP 플로우의 전송률과 거의 근사한 값으로 비디오 스트림의 전송률을 조절하는 것을 확인할 수 있다. 통계적으로 전체 실험 시간에서 실제 TCP 플로우의 평균 전송률은 5.09Mbps으로 측정되었으며 TF-RTP 플로우의 평균 전송률은 3.96Mbps로 측정되었다. 하지만 초기 상태를 지난 시점부터의 TF-RTP의 평균 전송률은 4.32Mbps로, 경쟁하는 TCP 플로우와 근사한 전송률을 가지는 것을 확인하였다.

그림 11의 실험을 확장하여 플로우 수를 증가시키면서 실험을 수행하였으며 그 실험 결과를 그림 12와 그림 13에 나타내었다. 그림 12는 TCP 플로우를 2개로 증가시켜 실험한 결과를 나타낸 것으로 TCP는 기본적으로 전송률의 변화가 심한 고유의 특성을 그대로 보여준다. 하지만 TF-RTP는 상대적으로 전송률의 변화가 크게 발생하지 않는 것을 확인할 수 있으며 이것은 비디오 스트리밍 서비스에서 중요하게 요구되는 성질이라 할 수 있다. 2개의 TCP의 평균 전송률은 3.2Mbps로 측정되었으며 TF-RTP의 평균 전송률은 3.1Mbps로 TCP 친화적인 전송률로 비디오 스트림의 전송률을 성공적으로 조절하는 것을 확인할 수 있다.

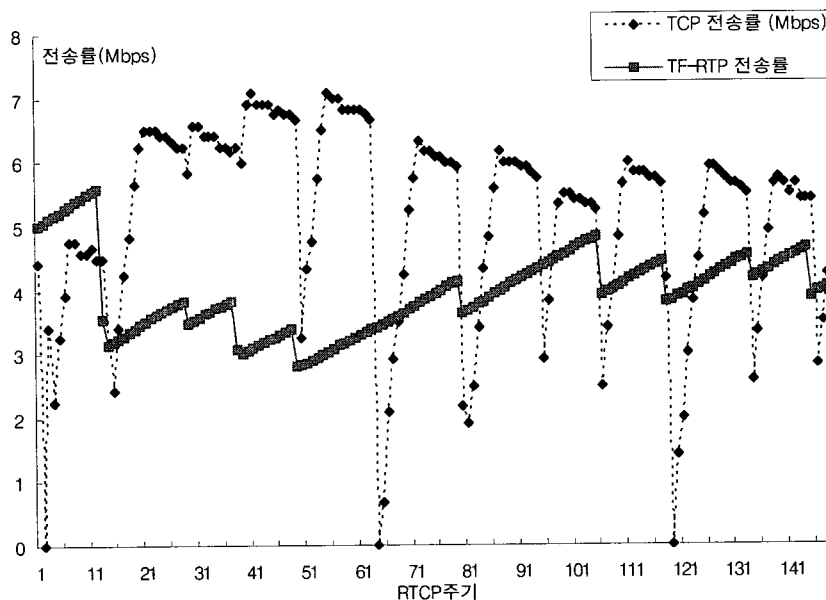


그림 11. 전송률 비교: TF-RTP vs. 1 TCP

Fig 11. Throughput comparison between TF-RTP and 1 TCP

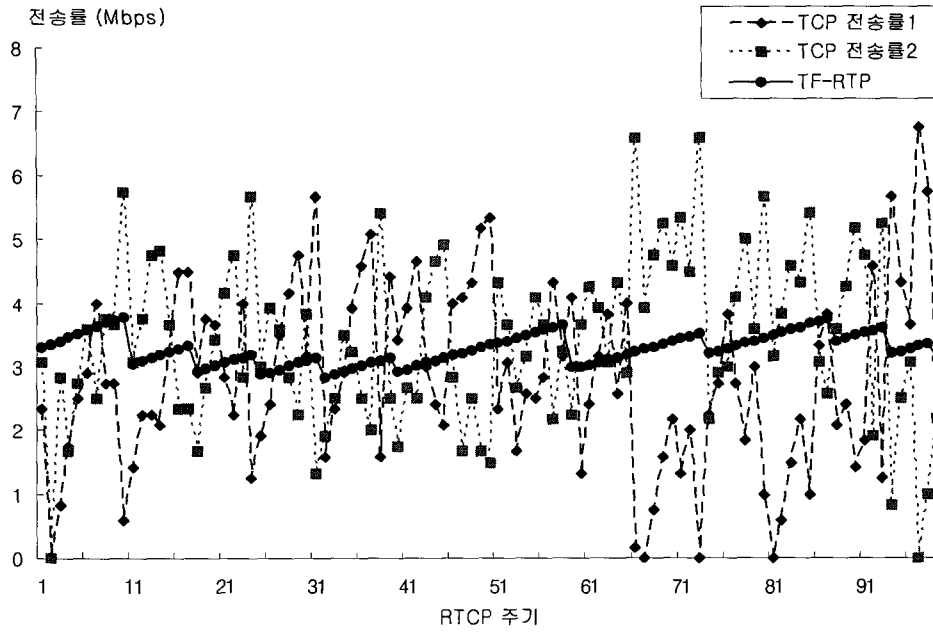


그림 12. 전송률 비교: TF-RTP vs. 2 TCP

Fig 12. Throughput comparison between TF-RTP and 2 TCP

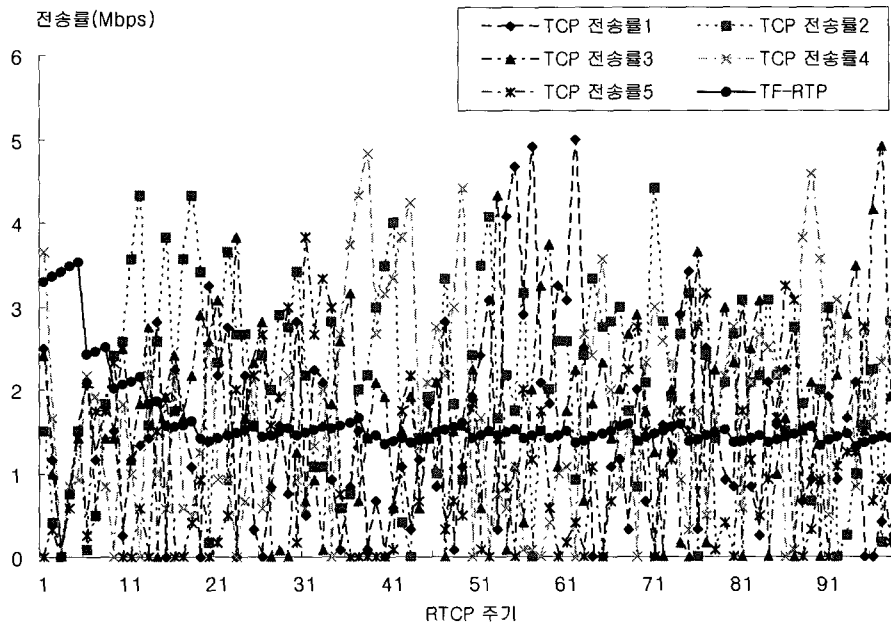


그림 13. 전송률 비교: TF-RTP vs. 5 TCP

Fig 13. Throughput comparison between TF-RTP and 5 TCP

실험을 좀더 확장하여 5개의 TCP 플로우와 TF-RTP를 경쟁시켰을 때의 실험 결과를 그림 13에 나타내었다. 앞선 실험과 마찬가지로 TF-RTP는 TCP와 달리 효율적인 비디오 스트림 전송을 위해 전송률이 부드럽게 변화하는 것을 확인할 수 있다. 5개 TCP 플로우의 평균 전송률은 1.71Mbps로 측정되었으며 TF-RTP의 평균 전송률은 1.63Mbps로, TCP 친화적인 전송률 조절을 수행하는 것을 확인할 수 있다.

표 1은 경쟁하는 TCP 플로우와 TF-RTP 플로우 사이의 네트워크 대역폭 사용에 대한 형평성을 분석한 실험 결과를 나타낸다. 실험 결과는 경쟁하는 TCP 플로우 수에 따라서, TCP 플로우의 평균 대역폭 점유와 TF-RTP 플로우의 평균 대역폭 점유에 대한 비율로 구하였으며 그 결과를 형평성에 대한 지수, Fairness로 나타내었다. 결과를 통해 플로우의 개수가 극히 작은 경우를 제외하고는 플로우 개수에 독립적으로 형평성이 일정하게 유지됨을 확인할 수 있었으며 이를 통해 제안하는 TF-RTP가 비디오 스트림을 전송하는데 있어, TCP 친화적으로 전송률을 조절함과 동시에 네트워크 대역폭 사용에 있어서도 경쟁하는 다른 플로우와 공평하게 사용한다는 것을 확인할 수 있었다.

표 1. 형평성 분석

Table 1. Fairness analysis

경쟁 플로우 수	TF-RTP			
	vs. 1 TCP	vs. 2 TCP	vs. 5 TCP	vs. 9 TCP
TCP 평균 전송률	4.78Mbps	3.2Mbps	1.71Mbps	1.02Mbps
TF-RTP 평균 전송률	4.32Mbps	3.1Mbps	1.63Mbps	0.97Mbps
Fairness	0.9	0.96	0.95	0.95

$$Fairness = \frac{Avg. TF - RTP's Throughput}{Avg. TCP's Throughput}$$

V. 결론

오늘날, 오디오나 비디오 데이터를 전송하는 멀티미디어 스트리밍에서 데이터 패킷의 전달은 대부분 UDP기반의 RTP를 통해 이루어지고 있다. 이러한 UDP기반의 RTP는 기본적으로 혼잡제어 메커니즘이 없으며 현재 인터넷의 주요 데이터 트래픽인 TCP와의 형평성을 보장하지 않는다는 문제점을 갖는다. 이러한 문제를 해결하기 위해 경쟁하는 TCP 플로우에 친화적으로 전송률을 조절하는 비디오 스트림 전송 기법으로 LDA나 SRTP가 제안되었으나 TCP의 평균 전송률을 정확하게 계산하지 못한다는 근본적인 한계를 가지며 이를 극복하기 위해 추가적인 알고리즘을 사용하게 된다.

본 논문에서는 이러한 LDA나 SRTP와 같은 기존 연구의 가장 큰 문제점인 TCP 평균 전송률 계산의 부정확성을 개선하기 위해 수정된 파라미터 계산 방법을 제안하였으며, 이를 통해 보다 정확하게 경쟁하는 TCP 플로우의 평균 전송률을 계산하여 비디오 스트림의 전송률을 조절하는 TF-RTP를 제안하였다. LDA나 SRTP가 TCP 친화적이면서 비디오 스트리밍에서 요구되는 안정적인 전송률 변화를 위해 추가적인 알고리즘을 사용하는 반면, 제안하는 TF-RTP는 정확한 파라미터 계산과 그에 따른 TCP 평균 전송률 계산을 바탕으로 추가적인 알고리즘 사용없이 TCP 친화적이고 공평하게 비디오 스트림의 전송률 조절이 가능하게 된다. 여러 실험을 통해 TF-RTP의 성능을 검증 및 분석하였으며 그 결과로 기존의 제시 방법에 비해 TCP 친화성과 네트워크 대역폭 사용에 대한 형평성이 크게 개선되었음을 확인하였다.

향후 과제로는 초기 상태의 불안정성에 대한 알고리즘 개선과 함께 전송률에 조절에 따른 비디오 스트림의 품질에 관한 연구를 수행할 것이다. 또한 실험 환경을 확장하여 다양한 환경에서 TF-RTP에 대한 성능 측정 및 분석에 관한 연구가 수행되어야 할 것이다.

참 고 문 헌

[1] S. Floyd and F. Kevin, "Router mechanisms to support end-to-end congestion control," Technical Report, LBL-Berkeley, Feb. 1997.

[2] S. Cen, C. Pu, and J. Walpole, "Flow and congestion control for internet streaming applications," Multimedia Computing and Networking, Jan. 1998.

[3] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate based congestion control mechanism for real-time streams in the Internet," IEEE INFOCOMM, Mar. 1999.

[4] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," International Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV), 1999.

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," In Proceedings of SIGCOMM, Aug. 2000.

[6] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP emulation at receivers - flow control for multimedia streaming," Technical Report, NCSU, Apr. 2000.

[7] D. Bansal, and H. Balakrishnan, "Binomial Congestion Control Algorithms," IEEE INFOCOMM, Apr. 2001.

[8] D. Sisalem, and H. Schulzrinne, "The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme," International Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV), Jul. 1998.

[9] B. Song, K. Chung, and Y. Shin, "SRTP: TCP-friendly congestion control for multimedia streaming," 16th International Conference on Information Networking, Jan. 2002.

[10] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," ACM SIGCOMM, 1998.

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF, RFC 1889, Jan. 1996.

[12] V. Jacobson, "Congestion Avoidance and Control," ACM SIGCOMM, Aug. 1988.

[13] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms", RFC2001, Jan. 1997.

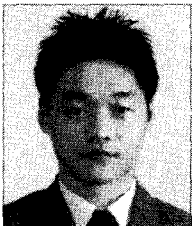
[14] The network simulator ns-2, <http://www.isi.edu/nanam/ns/>

저 자 소 개



이 정 민

- 2000년 : 광운대학교 전자공학부 학사
- 2002년 : 광운대학교 전자통신공학과 석사
- 2002년~현재 : 광운대학교 전자통신공학과 박사과정
- 주관심분야 : 비디오 스트리밍, 무선 전송 프로토콜, 인터넷 QoS



이 선 현

- 2003년 : 광운대학교 전자공학부 학사
- 2005년 : 광운대학교 전자통신공학과 석사
- 2005년~현재 : 광운대학교 전자통신공학과 박사과정
- 주관심분야 : 인터넷 QoS, 비디오 스트리밍, Wireless Sensor Network

저 자 소 개

**정 광 수**

- 1981년 : 한양대학교 전자공학과 학사
- 1983년 : 한국과학기술원 전기 및 전자공학과 석사
- 1991년 : 미국 University of Florida 전기공학과 박사(컴퓨터공학전공)
- 1983년~1993년 : 한국전자통신연구원 선임연구원
- 1991년~1992년 : 한국과학기술원 대우 교수
- 1993년~현재 : 광운대학교 전자공학부 교수
- 주관심분야 : 컴퓨터통신, 인터넷 프로토콜, 유무선 스트리밍 프로토콜, 유비쿼터스 센서 네트워크