한국항행학회 논문지 제14권 제3호 2010년 6월

무선 네트워크상에서 멀티미디어 스트리밍 최적화를 위한 전송율 기반의 오버헤드 모니터링

Transmission Rate-Based Overhead Monitoring for Multimedia Streaming Optimization in Wireless Networks

이종득*

Chong-Deuk Lee*

요 약

무선 네트워크상에서 혼잡과 지연은 네트워크 내에 존재하는 패킷의 수가 과도하게 증가하거나 송신측과 수신측의 전송 균형이 일치되지 않을 때 주로 발생한다. 이러한 혼잡과 지연은 패킷 손실 (pack loss)의 원인이 되며, 패킷손실은 멀티미디어 스트리밍의 성능을 떨어뜨릴 뿐만 아니라 오버헤드를 증가시킨다. 본 논문에서는 무선네트워크의 패킷 손실을 최적화하고 멀티미디어 스트리밍의 QoS 향상을 위한 전송율 기반의 멀티미디어 스트리밍의 최적화 메카니즘을 제안한다. 제안된 기법은 전송율 모니터링과 오버헤드 모니터링에 기반하여 최적화를 수행한다. 이러한 목적을 위하여 본 논문에서는 소스율 제어에 의한 최적화를 수행하도록 하며, 이것은 혼잡, 지연 등의 이슈들을 최적화하기 위한 것이다. 성능 평가는 RED (Random Early Detection), TFRC (TCP-friendly Rate Control) 그리고 제안된 기법으로 수행하였으며, 시뮬레이션 결과 제안된 기법이 RED, TFRC기법에 비해서 패킷손실율, 처리율, 평균 응답율이 보다 효율적임을 알게 되었다.

Abstract

In the wireless network the congestion and delay occurs mainly when there are too many packets for the network to process or the sender transmits more packets than the receiver can accept. The congestion and delay is the reason of packet loss which degrades the performance of multimedia streaming. This paper proposes a novel transmission rate monitoring-based optimization mechanism to optimize packet loss and to improve QoS. The proposed scheme is based on the trade-off relationship between transmission rate monitoring and overhead monitoring. For this purpose this paper processes a source rate control-based optimization which optimizes congestion and delay. Performance evaluated RED, TFRC, and the proposed mechanism. The simulation results show that the proposed mechanism is more efficient than RED(Random Early Detection) mechanism and TFRC(TCP-friendly Rate Control) mechanism in packet loss rate, throughput rate, and average response rate.

Key words : congestion, delay, multimedia streaming, overhead monitoring., source rate control

I. 서 론

최근에 무선네트워크상에서 멀티미디어 스트리밍을 위한 프로토콜로서 TCP 프로토콜이 사용되고 있

* 전북대학교 공과대학 전자공학부(Div. of Electronic Engineering, Chonbuk National University)
· 제1저자 (First Author) : 이종득
· 투고일자 : 2010년 5월 4일
· 심사(수정)일자 : 2010년 5월 6일 (수정일자 : 2010년 6월 18일)
· 게재일자 : 2010년 6월 30일

다. 그러나 TCP 기반의 프로토콜은 지연과 혼잡에 매우 민감하며 이로 인하여 빈번한 끊김 현상과 재전송으로 인한 패킷손실 그리고 혼잡 등의 문제가 발생되고 있다[1],[2],[3],[4]. 특히 멀티캐스트 관점에서 볼 때 무선 네트워크의 멀티미디어 스트리밍은 TCP 기반의 스트리밍을 어렵게 하고 있다. 그리고 유니캐스트 관점에서 볼 때 TCP 기반의 스트리밍은 대역폭이 크고 단말 지연이 적을 때 효율적이다[4],[5]. 그러나 단말 지연이 크고 대역폭이 제한될 때는 혼잡과 지연으로 인한 전송 신뢰성 문제가 발생한다. 이러한 문제를 해결하기 위하여 TCP-friendly 기반의 혼잡 제어 기법 들이 제안되고 있다[4],[5]. 혼잡 제어 기법에는 window-based schemes와 rate-based schemes이 사용되고 있으며, rate-based schemes은 다시 probe-based schemes과 equation-based schemes로 분류된다[4][6]. equation-based scheme 은 패킷 크기, 패킷 손실율, RTT (Round Trip Time)의 조건이 일정한 응용에서는 전송율 향상이 제공된다.

이에 비해서 TFRC (TCP-friendly Rate Control) 는 equation-based scheme으로서 패킷 크기가 일정한 경우 유니캐스트 플로우를 위한 기법이다[6]. 그러나 멀티미디어의 패킷 크기는 가변적이기 때문에 이로 인해 혼잡이 발생한다. RED는 이러한 혼잡을 줄이기 위한 기법이다[7]. 무선 네트워크에서 혼잡 제어는 처리율 개선에 중요한 역할을 하며, 이처럼 혼잡제어는 프로토콜의 공정성과 응답성을 좋게 유지해 줄 뿐만 아니라 멀티미디어 응용의 상영 품질을 좋게 유지해 준다[8],. 따라서 본 논문에서는 프로토콜의 공정성과 응답성을 좋게 유지하여 멀티미디어 스트리밍의 처리율을 향상시키기 위한 스트리밍 최적화 메카니즘을 제안한다. 제안된 메카니즘은 네트워크 내에 존재하는 패킷의 수가 과도하게 증가되고 있는지 혹은 송신측과 수신측의 전송이 균형을 이루는지를 모니터링한다. 이러한 과정은 멀티미디어 스트리밍의 처리율에 매우 중요하다. 제안된 기법은 멀티미디어 스트리밍의 QoS의 요구사항을 만족해 주며, 우리는 이러한 기능을 통해서 단말 지연과 혼잡에 따른 패킷 손실을 최소화한다. 우리는 시뮬레이션 결과를 통해서 제안된 기법이 다른 기법들에 비해서 보다 효율적인지를 살펴본다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해서 기술하며, 3장에서는 제안된 기법의 전송율 기반의 스트리밍 최적화 기법에 대해서 살펴본다. 4장에서는 제안된 기법의 시뮬레이션 결과에 대해서 살펴보며, 끝으로 결론에 대해서 살펴본다.

Ⅱ. 관련연구

2-1 혼잡제어에 의한 최적화

무선 네트워크에서 혼잡제어는 전송 프로토콜의 공정성 (fairness)과 응답성 (responsibility)을 제공해 줄 뿐 아니라 멀티미디어 응용 도메인에 보다 나은 미디어 품질 (media quality)을 제공한다[9],[10]. 무선 네트워크는 유선 네트워크와는 달리 채널 대역폭 제약 (channel bandwidth limit), 자원 제약(resource limit), 그리고 전송율 (transmission rate) 불균형으로 인하여 혼잡 문제가 발생한다. 최근에는 이러한 문제를 해결하기 위하여 많은 연구들이 제안되고 있다. [9]는 Multi-hop 무선 네트워크상에서 혼잡제어와 플로우 제어를 위해 크로스-레이어 설계 (cross-layer design) 기법을 제안하였다. [11],[12]에서는 패킷이 도중에 도착하여 패킷이 손실될 때, 이러한 문제를 해결하기 위하여 경험적 기법을 적용하였다. 이 기법은 혼잡 또는 무선 오류가 발생하였을 때 패킷에게 손실 과정을 보여주도록 하는 기법이다. 이외에도 Akan과 Akyildiz는 equation 기반의 기법을 제안하였으며, 이 기법은 무선 네트워크에서 멀티미디어 트래픽을 제어하기 위하여 ARQ (Analytical Rate Control) 기법을 적용하였다 [13],[14]. 그러나 이 기법은 패킷 손실이 발생할 때 손실로 인한 오류의 원인을 정확하게 찾을 수 없다는 문제점이 있다. 그리고 Chen과 Zakhor는 end-to-end 기법을 제안하였으며, 이 기법은 충분한 대역폭을 이용하여 패킷 손실과 이로 인한 혼잡을 제어하였다[15]. 그러나 이 기법은 무선 채널 상에서 패킷 손실을 최소화하기 위하여 채널 대역폭 조건을 분명하게 알고 있어야 하는 문제점이 있다.

2-2 소스율제어에 의한 최적화

무선 네트워크에서 소스율 제어는 보다 나은 미디어 품질을 제공하기 위하여 소스율을 채널 조건과 일치시키는 기능이다[15],[16]. 소스율은 미디어 스트리밍의 서비스 품질을 향상시키기 위해 사용되는 원시 데이터의 사용율로서 소스율 제어는 수신측에서 미디어 스트리밍 속도 조절을 통해 수행된다. 이를 위해서 [1]은 송신측에서 소스율 제어 기법을 제안하였다. 이 기법은 송신측에서 응용 도메인의 채널 조건과 QoS 요구사항을 분석하여 소스율을 제어하는 기법이다. [17]은 인코더 버퍼와 단말 지연 제약에 따른 비트스트림을 위한 가상 네트워크 버퍼 관리 알고리즘을 제안하였다. 그러나 이들 기법들은 송신측과 수신측 사이의 동적인 전송율 상태를 고려하지 않았기 때문에 전송율의 공정성과 응답성이 떨어지는 문제점이 발생하고 있다. 따라서 본 논문의 목적은 전송율의 공정성과 응답성을 보장하고, 오버헤드를 최소화하기 위한 최적화된 미디어 스트리밍 기법을 제안하는데 있다.

2-3 크로스-레이어 설계기법에 의한 최적화

무선 네트워크에서 노드의 이동성과 대역폭 제약은 패킷 손실의 주요한 원인이 되고 있다. 이러한 문제를 해결하기 위하여 크로스-레이어 기법이 제안되고 있으며, 최근에는 미디어 스트리밍의 성능을 향상시키기 위하여 많은 크로스-레이어 설계기법이 제안되고 있다[2],[3],[9] 그러나 기존의 크로스-레이어 기법들은 MAC/물리계층 (Physical layer)에서의 단순한 정보 처리를 위해 사용되고 있다. 그리고 크로스-레이어 설계 기법은 TCP-friendly 기반의 전송율 제어와 Multimedia-friendly 기반의 전송율 제어를 모두 고려해야 한다. [9]는 TCP-friendly 적응성 (adaptability) 기법을 제안하였으며, 이 기법은 멀티미디어 응용 도메인에서 최소 대역폭과 최대 대역폭 그리고 전송율을 고려하여 스트리밍 QoS를 향상시켰다. [3]은 인터넷상에서 발생하는 혼잡 제어를 위해 응용계층 (application layer)에서는 소스율을 제어하도록 하였으며, 그리고 전송계층 (transport layer)에서는 전송율을 제어하도록 하였다.

그러나 이들 기법은 혼잡이 발생할 때 패킷 손실이 똑 같이 발생할 수 있다는 점이며, 무엇보다 이들 기법은 무선 채널 환경에서 QoS가 떨어진다는 점이다.

Ⅲ. 스트리밍 최적화 메카니즘

TCP-friendly 프로토콜l은 네트워크 공정성을 유지하면서 멀티미디어 응용을 만족시키는 프로토콜이다[6]. 그러나 이 프로토콜은 애드혹 네트워크 (ad-hock networks), 센서 네트워크 (sensor networks)와 같은 무선 네트워크에서 기존의 TCP와는 달리 네트워크 혼잡, 지연, 라우팅 실패, 링크 오류 등에 민감하다. 이러한 제약조건에도 불구하고, TCP-friendly는 UDP와 함께 무선네트워크의 적합한 대안이 되고 있다. 이 절에서 우리는 TCP-friendly환경 하에서 스트리밍을 최적화하는 기법에 대해서 살펴본다.

3-1 TCP-friendly 기반의 전송율 제어

이 절에서는 멀티미디어 스트리밍을 위해 TCP-friendly 기반의 전송율 제어와 대역폭 요구조건 그리고 단말간 지연 조건 등을 고려한다. 제안된 메카니즘은 TFRC기반 위에서 수행되며 TFRC처럼 TCP-friendly 전송을 수행한다[6]. 그러나 TCP-fiendly의 데이터 스트림을 전송할 때 충돌을 고려해야 하며, 실제로 충돌이 발생하면 혼잡과 지연이 발생하게 된다. 혼잡과 지연이 발생하면 패킷 손실을 줄이기 위해 전송율을 조절해야 하며, 이 과정은 송신측과 수신측의 전송 피드백을 통해서 수행한다. 송신측과 수신측의 전송율이 일치되지 않으면 패킷 손실이 발생되며, 이를 알아보기 위한 과정은 수신된 패킷으로부터 현재 손실된 패킷 p를 계산한다. 다음 식 (1)은 TCP-friendly의 전송율을 계산하는 대표적인 식이다[6].

$$T=\frac{s}{R\sqrt{\frac{2p}{3}}+t_{RTO}\min\left(1,3\sqrt{\frac{3p}{8}}\right)p\left(1+32p^{2}\right)} \quad (1)$$

여기서 T는 전송률로서 초당 전송되는 전송 바이트 (sending rate in bytes/s)율을 나타낸다. s는 패킷크기, R은 RTT (Round Trip Time)이다. 그리고 tRTO는 retransmit timeout value이다. 이후 수신측은 송신측에게 TCP-friendly 전송을 반복하게 된다. 만일 p가 매우 작으면 식(1)은 식(2)처럼 간단히 할 수 있다.

$$T=\frac{s}{R\sqrt{\frac{2p}{3}}} \qquad (2)$$

여기서 식(2)는 전송률의 상한 (upper bound)을 나타낸다.

3-2 전송율 모니터링

본 논문에서는 무선 네트워크에서의 멀티미디어 스트리밍의 패킷 손실을 줄이고 QoS를 향상시키기 위한 모니터링 기법을 제안한다. 제안된 기법은 전송율과 오버헤드 모니터링을 통해서 전송율을 최적화하고 모니터링 함수를 이용하여 언더플로우와 오버플로우로 인한 오버헤드를 조절한다. 전송률 모니터링함수는 스트림 데이터가 고정적 크기로 전송되는지 또는 가변적 크기로 전송되는지를 모니터링 한다. 또한 이것은 스트림이 긴 버스트인지 아니면 짧은 버스트인지를 모니터링한다. 일반적으로 짧은 버스트는 지연과 혼잡을 줄일 수 있지만 긴 버스트는 그렇지 않다. 이처럼 제안된 모니터링 함수는 송신측과 수신측 사이에서 중요한 역할을 하며, 혼잡과 지연을 줄이기 위해 패킷 손실을 모니터링한다. 그리고 전송율 조절은 패킷 손실 모니터링을 통해서 수행되며, 전송률 모니터링구조는 그림1과 같다.

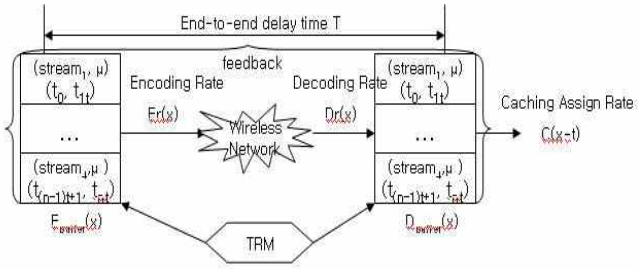

그림1 전송률 모니터링구조
Fig 1. Transmission Rate Monitoring Structure

이제 우리는 스트리밍을 위한 스트림 데이터 x를 가정하자. 스트림 데이터 x가 인코더 버퍼큐에 입력될 때 인코더 버퍼는 Ebuffer(x), 디코더 버퍼는 Dbuffer(x), 그리고 전송률 모니터링함수는 TRM(x)로 표시한다. 그리고 S(x)는 x 번째 스트림 데이터의 크기, Burst(x)는 송신측에서 전송된 버스트, B(t)는 임의의 t시간 동안에 수신측에서 실제 수신된 버스트라 하자. 일반적으로 손실이 발생하지 않으면 네트워크에서 처리율은 향상된다고 가정한다. 그러나 무선 네트워크에서 스트리밍은 네트워크 혼잡 등의 오버헤드에 의해 영향을 받는다.

따라서 패킷 손실이 발생하였을 때 인코더 버퍼, 디코더 버퍼에서의 전송률 모니터링함수를 통한 전송율 모니터링은 각각 다음과 같이 수행된다.

$$E_{buffer}(x)=\{E_{buffer}(x-1)+S(x)\}-B_{urst}(x)$$
$$D_{buffer}(x)=\{D_{buffer}(x-1)+B(t)\}-C(x-t)$$
$$TRM(x)=\{D_{buffer}(x)-E_{buffer}(x)\}\times\mu \qquad (3)$$

여기서 t는 스트림 데이터를 스트리밍할 때 발생되는 단말간 지연 시간이다. 그리고 C(x-t)는 캐시에 할당된 임의의 스트림 데이터이고, $\mu(0\leq\mu\leq1)$는 전송율 조절 계수이다.

패킷 손실을 측정한 후에 우리는 지연 시간을 고려한다. 지연 시간 측정은 임의의 t시간 동안에 전송된 스트림 데이터 x를 이용하며, 지연은 인코딩율과 디코딩율에 의해 좌우된다. 따라서 인코딩율과 디코딩율에 의한 지연 모니터링은 다음과 같이 수행된다.

$$D_{buffer}(x+t)=$$
$$\{\sum_{k=x+1}^{x+t} C(k)\}-\{E_{buffer}(x)+TRM(x)\} \qquad (4)$$

$$D_{buffer}(x)=$$
$$\{\sum_{k=x-t+1}^{x} S(k)\}-\{E_{buffer}(x)+TRM(x)\} \qquad (5)$$

만일 우리가 인코딩율을 고정한다면 혼잡과 지연으로 인한 패킷 손실은 줄어 들것이다. 그러나 멀티미디어 스트리밍은 크기가 가변적이기 때문에 인코

딩율이 고정적이지 않다. 무엇보다 인코더 버퍼에서 인코딩율과 디코더 버퍼에서의 디코딩율을 적절하게 유지해 준다면 오버플로우와 언더플로우로 인한 오버헤드는 줄어 들 것이다. 그러나 디코더 버퍼는 크기의 제약을 받으며, 디코딩율은 디코더 버퍼크기에 좌우된다. 따라서 오버플로우와 언더플로우로 인해 발생되는 오버헤드를 조절하기 위한 모니터링은 다음과 같다.

$$\sum_{k=x+1}^{x+t} C(k) \leq \{E_r(k)+D_r(k)+TRM(x)\} \quad (6)$$

여기서 Er(k)는 임의의 스트림 데이터에 대한 인코딩율이며, Dr(k)는 디코딩율이다.

식(6)에서 C(k)가 너무 크면 디코더 버퍼는 지나친 인코딩율로 인하여 오버플로우가 발생된다. 이러한 오버플로우를 피하기 위해서는 대역폭을 조절해야 하며, 이때 대역폭을 너무 크게 설정하면 전송율 문제가 뒤 따르므로 대역폭의 크기를 고려해야 한다.

3-3 오버헤드 모니터링

식(6)이 적절하게 제어되지 않으면 오버플로우로 인한 오버헤드가 발생할 수 있다. 이 절에서 우리는 오버헤드를 줄이기 위해 수신측 버퍼가 오버플로우인지 또는 언더플로우인지를 모니터링 한다. 우리는 이들을 모니터링하기 위해 먼저 임의의 t시간 동안에 스트리밍을 수행할 스트림들을 순차적으로 탐색한다. 그리고 난 후 수신측 버퍼에서 스트림 데이터 Dbuffer(x), ... , Dbuffer(x+t-1)을 체크를 하고 이 스트림 데이터들이 언더플로우인지 또는 오버플로우인지를 모니터링하게 된다.

ⅰ) 언더플로우 모니터링

언더플로우 모니터링은 인코딩율과 디코딩율 사이의 관계를 먼저 계산한다. 이때 인코더 버퍼에서 버스트가 긴 스트림들은 언더플로우를 예방하기 위해 μ가 가장 작은 순으로 스트림들을 제거한다. 우리는 인코더 버퍼에서 인코딩율을 적절히 조절하기 위해 μ를 $\mu \leq 0.5$, $0.6 \leq \mu \leq 0.7$, $\mu \geq 0.8$ 3단계로 분류한다. 그리고 우리는 디코더 버퍼에서의 디코딩율을 조절하기 위해 식(3)과 식(6), 그리고 소스 대역폭을 고려한다. 만일 이전의 모든 스트림 데이터가 최소한의 품질이 보장된 스트리밍이라면 S(x)는 최소한의 대역폭을 할당받게 된다. 그러나 S(x)를 수행하기 위한 대역폭이 디코딩율보다 크면 언더플로우가 발생한다. 따라서 우리는 언더플로우를 예방하기 위하여 디코더 버퍼에서 디코딩율을 조절해야 하며, 언더플로우를 예방하기 위한 UM(x)는 다음과 같다.

$$UM(x)= \{ \sum_{s=0}^{k} B(x-t+s) \times BW(x-1)+D_r \} \quad (7)$$

여기서 BW (Bandwidth)는 스트림 데이터에 대한 대역폭이다.

ⅱ) 오버플로우 모니터링

우리는 디코더 버퍼에서 디코딩율에 따른 오버플로우를 예방하기 위해 전송율과 디코딩율을 조절한다. 먼저 디코더 버퍼에서의 전송율을 계산한다. 전송율이 실제 디코딩율보다 크면 오버플로우가 발생하게 된다. 이처럼 디코딩율에 따른 오버플로우를 예방하기 위한 OM(x)는 다음과 같다.

$$OM(x)=\{ \sum_{s=0}^{k} B(x-t+s) \times BW(x-1)-D_r \} \quad (8)$$

직관적으로 볼 때 Er>Dr이면 오버플로우가 발생하고 Er<Dr이면 언더플로우가 발생한다. 그러나 이들을 조절하는 일은 쉬운 일이 아니다. 본 논문에서 우리는 오버플로우와 언더플로우로 인한 오버헤드를 모니터링하기 위하여 인코딩율과 디코딩율을 고려하였다.

3-4 스트리밍 최적화

전송율 모니터링과 오버헤드 모니터링이 끝 난 후에는 우리는 모니터링에 따른 스트리밍 최적화를 수행한다. 최적화 과정은 전송율에 의한 최적화와 송신측/수신측에 의한 최적화 과정으로 구분된다.

ⅰ) 전송율 모니터링에 의한 최적화

스트리밍이 최적화되기 위해서는 전송율은 인코딩율 Er과 디코딩율 Dr사이에 위치해야 한다. 이러한 과정을 위반하면 오버플로우와 언더플로우가 발생한다. 우리는 최적의 스트리밍을 고려하기 위해 버퍼의 상한과 하한을 결정하며 전송율을 고려한 최적화 RO(x)는 다음과 같다.

$$RO(x)=B_{urst}(x)-\frac{T-(E_r+D_r)}{2}\times\mu \qquad (9)$$

여기서 T는 전송율이다. 식(9)에서 μ값이 작으면 인코딩율 변량은 커진다. 따라서 최적화를 위해서는 인코딩율의 변량을 고려해야 한다. 본 논문에서는 $\mu \geq 0.7$일 때 인코딩율의 최적화가 시작되었다.

ⅱ) 송신측/수신측 모니터링에 의한 최적화

송신측과 수신측의 전송율이 균형을 이루면 전송율에 따른 최적화가 수행된다. 그러나 송신측에서의 인코딩율은 가변적이기 때문에 항상 최적화가 유지되는 것은 아니다. 이 절에서 우리는 송신측 버퍼와 수신측 버퍼사이의 스트림 데이터를 순차적으로 스트리밍하기 위한 최적화를 고려한다.

- 수신측 : (x-1)번째의 스트림 데이터의 스트리밍을 끝마친 후에는 다음 스트림 데이터Next(x)를 결정해야 한다. 이때 스트림 데이터 Next(x)는 크기가 고정일 수도 있고 또는 가변적일 수도 있다. 그러나 멀티미디어 데이터는 일반적으로 크기가 가변적이다. 따라서 우리는 가변적 크기만을 고려한다. 가변적 크기란 스트림 데이터의 크기가 서로 다른 버스트를 가진 스트림을 의미하며 최적화를 위해 인코딩율과 대역폭을 고려한다. 따라서 수신측에서의 Next(x)에 대한 스트리밍 최적화 RSO(x)는 다음과 같다.

$$RSO(x)=N_{ext}(x)\times T_{synchro}-\frac{E_r}{BW}\times SL(x) \qquad (10)$$

여기서 Tsynchro는 Next(x)에 대한 시간 동기화이며, SL(x)는 i 번째의 스트림 데이터의 크기, Er은 SL(x)에 대한 평균 인코딩율이다.

RSO(x)가 결정되면 Next(x) 대한 버퍼 크기를 고려해야 하며, Next(x) 대한 버퍼 크기 NSbuffer(x)는 다음과 같이 정의 한다.

$$NS_{buffer}(x)=T_{synchro}\times SL(x)\times\frac{Er-BW}{Er} \qquad (11)$$

만일 RSO(x)=0이면 최적화는 중단되며 이때에는 오버헤드가 발생된다. 그러므로 이러한 문제를 피하기 위해 RSO(x)≠0인 (Er/BW)ㅓth 번째의 스트림 데이터들에 대해서 스트리밍을 수행해야 한다.

- 송신측 : 수신측에서 혼잡 및 지연 없이 스트리밍을 계속하기 위해서는 스트림 데이터의 인코딩율을 지속적으로 모니터링해야 한다. 그러나 스트리밍 과정에서 인코딩율과 디코딩율 사이에는 실제 전송된 데이터의 양과 최적으로 전송된 데이터양에는 차이가 발생한다. 차이가 크면 최적화는 줄어들며, 반대로 차이가 적으면 최적화는 증가하게 된다. 앞에서 우리는 RSO(x)=0일 때를 살펴보았다. 즉 RSO(x)=0이면 오버헤드로 인해 전송효율이 감소하게 되므로 이러한 현상을 가능한 한 피해야 한다. 다음을 살펴보자.

1) T×Tsynchro>S(x)이고 RSO(x)>0 일 때, 이 경우는 대역폭의 크기가 증가함을 의미한다. 대역폭의 크기가 크면 전송율은 영향을 받기 때문에 T×Tsynchro를 조절해야 한다. 송신측에서 인코딩율을 조절하면 수신측에서의 응답성은 보다 좋아진다. 따라서 송신측에서 응답성을 고려한 인코딩율 조절 EA(x)는 식(12)와 같이 수행한다.

$$EA(x)=\{S(x)-S(x-1)\}+T\times T_{synchro}\times\delta \qquad (12)$$

여기서 $\delta(0<\delta<1)$는 응답 계수이다.

만일 EA(x)=0 이면 우리는 인코딩율 조절을 중단하고 처음부터 다시 시작해야 한다.

2) T×Tsynchro<S(x)이고 RSO(x)<0 일 때, 이 경우는 대역폭의 크기가 감소함을 의미한다.

대역폭의 크기가 감소할 때에는 최적화를 위해 인코딩율 T×Tsynchro을 조절해야 한다. 인코딩율 조절은 δ에 의해 결정되며, δ가 크면 응답성은 좋아지게 되기 때문에 TCP-friendly도 계속적으로 유지되게 된다. 그러나 δ가 작으면 응답성은 부분적으로 보장되지만 TCP-friendly는 무시된다. 그리고 $\delta \neq 0$이면 응답성이 좋아질 뿐만 아니라 전송 효율도 좋아지게 된다.

Ⅳ. 시뮬레이션 결과

이 절에서 우리는 NS-2 시뮬레이터를 이용하여 제안된 기법의 성능을 시뮬레이션 하였다[18]. 비트율은 1.28Mbps, 스트리밍을 위한 스트림 데이터는 200개의 비디오 프레임과 1,500개의 이미지 패킷 데이터를 이용하였다. 시뮬레이션에서 패킷 스트리밍을 위한 패킷크기는 512kb, 링크 대역폭은 10/100Mbps, 평균 링크 대역폭은 약 1.2Mbps로 설정하였다. 그리고 시뮬레이션 time은 420s 동안 수행하였으며, 스트림의 ts는 [1, 20s], μ는 $\mu \geq 0.7$, δ는 $0<\delta<1$로 설정하였다. 표준 비디오 시퀀스는 200개의 프레임을 가진 영화 비디오 소스를 사용하였으며 MPEF-4 FGS (Fine Granularity Scalable)를 이용하여 인코드하였다. 연속적인 스트리밍을 위한 스트림 데이터는 20개 이내로 제한하였으며, 최대 PSNR (Peak Signal to Noise Ratio)은 30dB이내로 제한하였다. 우리는 제안된 기법의 성능을 알아보기 위하여 RED, TFRC 그리고 우리의 기법의 성능을 비교하였다. 시뮬레이션을 위한 프로토콜은 IEEE 802.11에서 수행하였으며 각 기법의 무선링크는 10Mbps 이내로 제한하였다. 그리고 인코딩율과 디코딩율은 각각 100kb로 세트시켰으며, 인코더 버퍼크기와 디코더 버퍼크기는 모두 550kb로 세트시켰다. 모바일 터미널은 수신측에 멀티미디어 플로우가 있고, 멀티미디어의 맨처음 플로우는 무선이라 가정하였다. 전체 시뮬레이션은 평균 패킷 손실률, 처리율, 평균 응답률을 측정하였다. 우리는 서로 다른 링크 전송 지연 시간 t를 20ms에서 70ms까지 변화해 가면서 평균 패킷 손실율과 처리율을 시뮬레이션 하였으며, 시뮬레이션 결과는 그림2, 그림3과 같다.

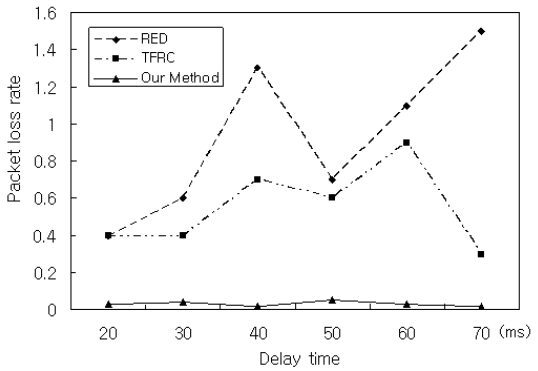

그림2 전송지연시간에 따른 패킷 손실율
Fig 2. Packet Loss Rate by Transmission Delay

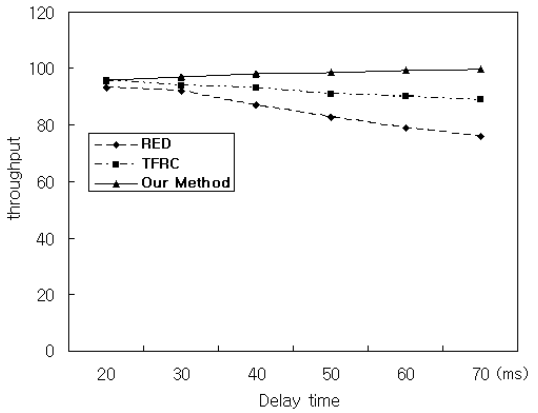

그림3 전송지연에 따른 throughput
Fig 3. Throughput by Transmission Delay

그림3에서 보듯이 링크 지연 시간t가 클 때 제안된 기법은 처리율이 크게 변하지 않지만 RED와 TFRC는 성능이 감소함을 알 수 있다. 그림4는 스트림 데이터 수에 따른 평균 응답률이다. 그림4의 770, 1060, 1570은 비디오 프레임을 포함한 수이다.

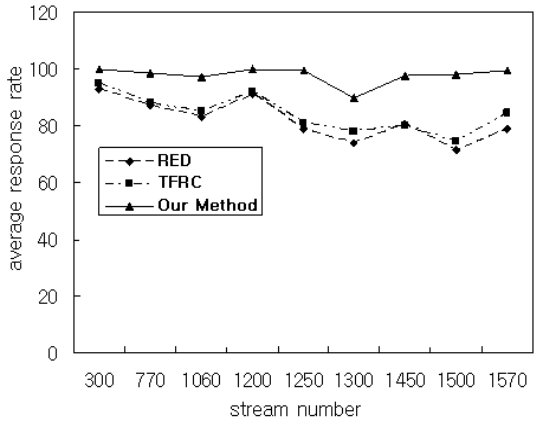

그림4 스트림 데이터 수에 따른 평균 응답율
Fig 4. Average Response Rate by Stream Data Number

그림4에서 보듯이 스트림 데이터의 수가 770, 1060, 1570일 때는 평균 응답률이 약간 감소하였다. 이것은 비디오 프레임이 포함되었기 때문이다. 전체 시뮬레이션 결과는 패킷 손실률과 처리율이 최적화됨을 보여 주었다. 그리고 스트림 데이터 수에 따른 평균 응답율 또한 최적화됨을 보여 주었다. 따라서 우리는 시뮬레이션을 통하여 제안된 기법이 RED와 TFRC기법에 비해서 성능이 최적화됨을 알 수 있다.

Ⅴ. 결 론

무선 네트워크상에서 혼잡과 지연은 네트워크 내에 존재하는 패킷의 수가 과도하게 증가하거나 송신측과 수신측의 전송 균형이 일치되지 않을 때 주로 발생한다. 이러한 혼잡과 지연은 패킷 손실과 오버헤드의 원인이 되며, 패킷손실과 오버헤드는 멀티미디어 스트리밍의 성능을 떨어뜨린다. 본 논문에서 우리는 무선네트워크의 패킷 손실을 최적화하고 멀티미디어 스트리밍의 QoS 성능 향상을 위한 스트리밍 최적화 메카니즘을 제안하였다. 제안된 기법은 전송율 모니터링과 오버헤드 모니터링의 trade-off 관계에 기반을 두었다. 본 논문에서는 최적화를 위해 인코딩율과 디코딩율을 고려하였으며, 그리고 언더플로우 모니터링과 오버플로우 모니터링을 고려하였다. 또한 우리는 전송율에 의한 최적화와 송신측/수신측에 의한 최적화를 고려하였다. 우리는 시뮬레이션을 통해서 제안된 기법의 성능을 알아보았으며, 그 결과 제안된 기법이 RED, TFRC 에 비해서 성능이 효율적임을 알 수 있었다.

참 고 문 헌

[1] A. Shegal, O. Verscheure, and P. Frossard, "Distortion-buffer optimized TCP video streaming", *in Proc. IEEE Int. Conf. Image Processing(ICIP),* pp.2083-2086, 2004.

[2] Peng Zhu, Wenjun zeng, and Chunwen Li, "Joint Design of Source Rate Control and QoS-Aware Congestion Control for Video Streaming Over the Internet", *IEEE Trans. on Multumedia, vol. 9. no.2*, pp.366-376, 2007.

[3] Peng Zhu, Wenjun zeng, and Chunwen Li, "Cross-Layer Design of Source Rate Control and Congestion Control for Wireless Video Streaming", *Hindawi Publishing Corporation Advances in Multimedia*, pp.1-13, 2007.

[4] N.R Sastry and S.S Lam, "CYRF: A Theory of Window-based Unicast Congestion Control", *IEEE/ACM Trans. on Networking, vol. 13, no. 2*, pp.330-342, 2005.

[5] R. Rejaie, M. Handley, and D. Estrin, "RAP: an end-to-end rate-based congestion control mechanism for realtime streams in the internet", *in Proceedings of the 18th Annual Joint Conf. of the IEEE Computer and Communications Societie(INFOCOM'99)*, vol. 33, pp.1337-1345.

[6] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP-Friendly rate control(TFRC): protocol specification", *IETF RFC 3448*, 2003.

[7] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss", *in Proc. IEEE Infocom 2003, SanFrancisco, California, USA*, Apr. 2003.

[8] B. Wang, J. F. Kurose, P. J. Shenoy, and D. F. Towsley, "Multimedia Streaming via TCP : an analytic performance study", *Proc. ACM Multimedia'04,* pp.908-915, 2004.

[9] Kai-Ting Yang, Wei Kuang Lai and Chin-Shiuh Shieh, "A Cross-layer Approach to Packet Size Adaptation for Improved Utilization in Mobile Wireless Networks", *International Journal of Innovative Computing, Information and Control*, *vol.5, no.11(B)*, pp.4335-4346, 2009.

[10] Takuo Nakashima, "Properties of the Correlation between Queue Length and Congestion Window Size under Self-similar Traffics", *International Journal of Innovative Computing, Information and Control, vol.5, no.11(B)*, pp.4373-4382, 2009.

[11] Ali Moarefianpour and Vahid Johari Majd, "Input-to-state Stability in Congestion Control Problem of Computer Networks with Nonlinear Links", *International Journal of Innovative Computing, Information and Control, vol.5, no.8,* pp.2091-2106, 2009.

[12] S. Cen, P. C. Cosman, and G. M. Voelker,"End-to-End differentiation of congestion and wireless losses", *IEEE/ACM Transctions on Networking, vol. 11,* pp.703-717, 2003.

[13] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links", *IEEE/ACM Transactions on Networking, vol.5, no.6,* pp.756-769, 1997.

[14] D.Barman and I. Matta, "Effectiveness of loss labeling in improving TCP performance in wired/wireless network", *in proceeding of the 10th IEEE International conference Network Protocols(ICNP'02)*, pp.2-11, 2002.

[15] M. Chen and A. Zakhor, "Multiple TFRC connections based rate control for wireless networks", *IEEE Transactions on Multimedia, vol.8, no.5*, pp.1045-1062, 2006.

[16] B. Xie and W. Zeng, "Rate-distortion optimized dynamic bit-stream switching for scalable video streaming", *in IEEE international Conference On Multimedia and Expo(ICME'04), vol.2*, pp.1327-1330, 2004.

[17] J. Yan, K. Katrinis, M. may, and B. Platter, "Media-and TCP-friendly congestion control for scalable video streams", *IEEE Transactions on Multimedia, vol.8, no.2*, pp.196-206, 2006.

[18] NS-2 simulator, http//:www.isi.edu/nanam/ns

이 종 득 (李鍾得)

1983년 2월 : 전북대학교 컴퓨터과학과(이학사)
1989년 2월 : 전북대학교 컴퓨터과학과(이학석사)
1998년 2월 : 전북대학교 컴퓨터과학과(이학박사)
1992년 3월~2002년 2월 : 서남대학교 컴퓨터통신학과 교수
2002년 2월~현재 : 전북대학교 전자공학부 교수
관심분야 : 무선 모바일 네트워크, 무선센서 네트워크, 무선모바일 애드혹 네트워크, 유비쿼터스 통신 등