

인터넷 혼잡상황에서 멀티미디어 스트리밍 서비스의 QoE 향상을 위한 전송률 제어기법

(A Novel Rate Control for Improving the QoE of Multimedia Streaming Service in the Internet Congestion)

구 자 현[†] 정 광 수^{**}
(Jahon Koo) (Kwangsue Chung)

요 약 IPTV나 Mobile IPTV와 같은 실시간 멀티미디어 스트리밍 시스템의 설계에 있어 중요한 요소 중 하나는 변화하는 네트워크 특성과 상태에 효율적으로 적응하여 멀티미디어 데이터를 전달하는 것이다. 네트워크 적응적 데이터 전송은 네트워크의 혼잡상황에서 네트워크의 안정성과 프로토콜간 공정성을 향상시킬 수 있지만 영상재생의 연속성과 같은 콘텐츠 재생 특성을 고려하지 않아 사용자에게 양질의 QoE(Quality of Experience)를 제공하지 못하는 문제점을 가지고 있다.

본 논문에서는 콘텐츠의 재생 특성을 고려하지 않아 발생하는 문제점을 해결하기 위해 인터넷 혼잡상황에서 멀티미디어 스트리밍 서비스의 QoE 향상을 위한 새로운 전송률 제어기법인 NCAR(Network and Client-Aware Rate control)을 제안하였다. 제안한 NCAR 기법은 네트워크 인지형 혼잡제어(Congestion Control)와 클라이언트 인지형 흐름제어(Flow Control)를 기반으로 동작하고 있다. 네트워크 인지형 혼잡제어는 멀티미디어 스트리밍의 공정성과 안정성을 향상시키고 높은 링크 활용도와 전송률의 변화를 감소시키며 클라이언트 인지형 흐름제어는 미디어 재생의 불연속성을 제거하고 안정된 버퍼할당과 낮은 재생지연시간의 좋은 특성을 제공한다. 시뮬레이션 결과를 통해 제안한 방법의 유효성을 확인하였다.

키워드 : 혼잡제어, 흐름제어, QoE, 전송률 제어, NCAR

Abstract The delivery of multimedia that efficiently adapts its bit-rate to changing network characteristics and conditions is one of the important challenging tasks in the design of today's real-time multimedia streaming systems such as IPTV, Mobile IPTV and so on. In these work, the primary focus is on network congestion, to improve network stability and inter-protocol fairness. However, these existing works have problems which do not support QoE (Quality of Experience), because they did not consider essential characteristics of contents playback such as the media continuity. In this paper, we propose a novel rate control scheme for improving the QoE of multimedia streaming service in the Internet congestion, called NCAR (Network and Client-Aware Rate control), which is based on network-aware congestion control and client-aware flow control scheme. Network-aware congestion control of the NCAR offers an improving reliability and fairness of multimedia streaming, and reduces the rate oscillation as well as keeping high link utilization. Client-aware flow control of NCAR offers a removing the media discontinuity and a suitable receiver buffer allocation, and provides a good combination of low playback delay. Simulation results demonstrate the effectiveness of our proposed schemes.

Key words : Congestion Control, Flow Control, QoE, Rate Control, NCAR

· 본 연구는 지식경제부와 한국산업기술진흥원의 전력기술인력양성사업으로 수행된 결과임

† 학생회원 : 광운대학교 전자통신공학과
jhkoo@cclab.kw.ac.kr

** 중신회원 : 광운대학교 전자통신공학과 교수
kchung@kw.ac.kr

논문접수 : 2009년 7월 29일

심사완료 : 2009년 9월 2일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 정보통신 제36권 제6호(2009.12)

1. 서론

현재 인터넷에서의 주요 트래픽들은 전송 프로토콜로 대부분 TCP(Transmission Control Protocol)를 사용하고 있다. TCP는 종단간(end-to-end)의 신뢰적인 패킷 전달을 수행하며 혼잡 제어 메커니즘(congestion control mechanism)을 통해 송신단의 전송률을 직접 제어하여 혼잡 상황으로 발생하는 데이터의 무분별한 손실을 막고 네트워크가 안정하게 동작할 수 있도록 한다[1].

최근 들어 오디오나 비디오 스트리밍과 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 멀티미디어 트래픽의 전송 프로토콜로는 적합하지 않다. 따라서 대부분의 멀티미디어 트래픽은 혼잡 제어를 수행하지 않는 UDP(User Datagram Protocol)를 기반으로 데이터를 전송하게 되었다. 하지만 UDP는 TCP와 같은 혼잡 제어 메커니즘이 없으므로 네트워크 혼잡 상황을 야기하여 네트워크를 붕괴(congestion collapse)시킬 가능성이 있으며 기존의 주요 인터넷 트래픽인 TCP와도 친화적(TCP-friendly)으로 동작하지 않는다는 문제점을 갖는다[1,2].

1990년대 이후로 스트리밍 전송기법과 관련하여 네트워크 안정성을 향상시키기 위해, 혼잡 제어 메커니즘 적용에 대한 연구가 활발하게 진행되고 있다[2~13]. 이러한 연구들은 네트워크의 상태에 따라 전송률을 조절함으로써 무분별한 패킷 손실을 막고, 네트워크의 안정성을 향상시키는 것을 목적으로 하고 있다. 하지만 이러한 기존의 연구들은 네트워크상태의 안정화만을 고려하므로 스트리밍 응용 프로그램의 본질적인 특성에 대한 고려가 부족하다는 한계를 가진다. 또한 스트리밍 서비스의 기본적인 요소인 미디어 스트림의 특성에 대한 고려가 부족하다는 한계점을 가지고 있다.

본 논문에서는 현재의 네트워크 상태와 전송되는 비디오 데이터의 특성을 모두 고려하는 전송률 조절기법으로서, NCAR(Network and Client-Aware Rate control) 기법을 제안한다. 제안한 NCAR 기법은 네트워크 상태에 따라 전송률을 조절하는 네트워크 인지형 혼잡 제어(Congestion Control)와 클라이언트 버퍼 내 서비스되는 비디오 데이터의 특성을 고려하여 동작하는 클라이언트 인지형 흐름제어(Flow Control)를 기반으로 동작하고 있다. 즉, 기존의 관련 연구들이 추구하는 네트워크 안정성 향상을 만족시키면서 사용자 관점에서 끊임없이 부드러운 비디오 재생을 위해 비디오 품질 및 전송률을 조절하게 된다.

본 논문의 2장에서는 스트리밍 전송기법과 관련하여 기존 연구들을 분류 및 기술하였고, 3장에서는 제안하는 NCAR 기법을 운영하기 위한 멀티미디어 스트리밍 시스템을 소개하였고, 제안하는 NCAR기법의 알고리즘에 대해 상세히 기술하였다. 4장에서는 시뮬레이터를 이용하여 제안하는 알고리즘의 성능을 검증하고, 기존 스트리밍 전송기법과의 성능 비교 결과를 기술하였으며, 마지막으로 5장에서 결론을 맺었다.

2. 관련연구

인터넷 환경에서의 효율적인 비디오 스트림 전송을 위해 기존의 여러 가지 기법들이 제안이 되었다. 본 장에서는 기존에 제안된 비디오 스트림 전송 기법들을 크게 두 가지로 분류하여 각각에 대하여 기술하고, 기존 연구들이 가지는 문제점에 대하여 설명하도록 한다.

2.1 TCP 친화적인 전송률 조절 기법

혼잡제어 메커니즘을 가지지 않는 UDP를 전송프로토콜로 사용하는 비디오 스트리밍 응용에서 네트워크 혼잡상황에 의한 패킷 손실과 전송 지연은 스트리밍 서비스 품질을 저하시키는 원인이 된다. 이러한 문제점을 해결하기 위해 혼잡제어 메커니즘을 적용한 멀티미디어 전송프로토콜에 대한 연구가 진행되었으며, 비디오 스트림의 전송률을 네트워크의 가용 대역폭에 적합하도록 조절함으로써 혼잡상황 발생을 감소시키게 된다[7].

혼잡제어 메커니즘을 적용하여 네트워크 혼잡상황을 예방하고 패킷 손실을 감소시키고자 하는 목적을 가졌던, 네트워크 관점의 전송기법들은 단순히 혼잡제어 메커니즘의 적용에서 만족하지 않고, 인터넷의 주요 트래픽인 TCP와의 친화성까지 고려하는 것으로 발전하게 된다. 이러한 연구의 결과로, TCP와 유사한 혼잡제어 메커니즘을 수행하면서 동시에 비디오 스트림 전송에 적합하도록 전송률의 변화를 줄인, RAP나 SQRT, TEAR와 같은 기법들이 제안되게 된다[2-4]. RAP는 TCP와 유사한 혼잡제어 메커니즘을 사용하지만 ACK 기반의 윈도우 조절방식이 아닌 전송률 조절방식을 사용하여 전송률의 변화를 감소시키게 된다. SQRT는 TCP의 AIMD(Additive Increase and Multiplicative Decrease) 기법의 파라미터 값 변경을 통해, 비디오 전송에 적합하도록 전송률의 변화를 줄이게 된다. TEAR는 TCP의 혼잡제어 메커니즘을 수신단에서 수행하여 네트워크 상태에 적합한 전송률을 송신단에 알려줌으로써 TCP 친화적으로 전송률을 조절하는 것으로, 윈도우 기반 방식과 전송률기반 방식이 혼합된 형태의 전송기법이다.

TCP 친화적인 전송률 조절 기법의 대표적인 예는 Padhye가 제안한 TCP 전송률 모델링 공식을 이용하는

TFRC(TCP-Friendly Rate Control) 기법이다. Padhye는 [8]에서, 식 (1)과 같이, 세그먼트 크기 $Packet\ Si$, 패킷 손실률 p , 종단간 지연 RTT , 재전송 타임아웃 t_{RTO} 을 이용하여 TCP 커넥션의 평균 전송률(T)을 계산하는 모델링 공식을 제안하였으며, TFRC 기법은 이러한 Padhye의 공식을 이용하여 비디오 스트림의 전송률을 TCP 친화적으로 조절하게 된다[5].

$$T = \frac{Packet\ Size}{RTT \sqrt{\frac{2p}{3}} + t_{RTO} \left(3 \sqrt{\frac{3p}{8}} \right) p(1+32P^2)} \quad (1)$$

수신단에서 송신단으로 피드백되는 패킷 손실률, 종단간 지연, 재전송 타임아웃을 이용하여 TCP 커넥션의 평균 전송률을 계산하는 Padhye의 공식은 동일한 네트워크 환경에서 경쟁하는 TCP 커넥션의 평균 전송률을 비교적 정확하게 추정할 수 있으며, 이러한 장점으로 인해 이후에 제안된 TCP 친화적인 전송률 조절 기법들에서 빈번히 사용되어 왔다[9].

기존의 네트워크 관점으로 접근한 비디오 전송 기법들은 혼잡제어 메커니즘 적용을 통해 패킷 손실을 줄여 혼잡상황을 줄이고, 경쟁하는 TCP 트래픽과 친화적으로 동작하며, 전송률의 변화를 줄여 스트리밍 서비스에 적합하다는 장점을 가진다. 하지만, 최선형(Best-Effort)의 서비스를 제공하는 인터넷 환경에서, 네트워크 정보를 미리 알 수 없으며, 여전히 전송률의 변화가 크게 발생할 수 있다는 문제점을 가진다. 또한 단순히 네트워크 관점에서의 혼잡제어 메커니즘 적용이나 TCP 친화적인 전송률 조절 기법이 실제 서비스되는 비디오 스트림의 품질 향상을 보장하지 않는다는 한계를 가진다[6]. 인터넷을 통한 비디오 스트리밍 응용에서 서비스 품질의 향상을 위해서는 네트워크 상태 변화에 따른 품질 적응 기법 적용이 필요하며 이와 관련한 연구들을 다음의 2.2절에서 설명하도록 한다.

2.2 네트워크 상태기반 품질 적응 기법

스트리밍 서비스의 품질 향상을 위한 다른 연구로 인터넷을 통한 비디오 스트리밍에서 네트워크의 안정성을 향상시키면서 사용자에게 제공되는 서비스 품질의 향상을 위해 품질 적응 기법이 제안 및 연구되었다. 품질 적응 기법은 혼잡제어 메커니즘에 의해 수시로 변화하는 네트워크의 가용 대역폭에 적합하도록, 서비스되는 비디오 스트림의 품질을 결정하는 동시에 높은 품질의 서비스를 가능한 오랫동안 제공하는 것을 목적으로 가진다. 이러한 네트워크 상태 기반 품질 적응 기법은 오래 전부터 연구되었던 주제로서, 이와 관련하여 여러 연구들이 수행되었다[10-21].

인코딩된 비디오 스트림의 품질을 변화시키는 기법들은 몇 가지로 구분할 수 있다. 첫 번째는 네트워크 상태

에 적합한 전송률을 가지는 데이터 전송을 위해, 이미 인코딩된 비디오 스트림을 디코딩 후, 양자화 파라미터를 수정하여 다시 인코딩하여 전송하는 방법으로, 디코딩과 인코딩을 다시 수행하는 과정에서 CPU 부하가 크고 다수의 클라이언트에 대한 서비스가 어렵다는 문제를 갖는다[10-12]. 또 다른 방법은 여러 버전으로 인코딩된 스트림들을 준비한 상태에서, 네트워크 가용 대역폭의 변화에 따라 적절한 데이터율을 가지는 스트림을 선택하여 전송하는 방법이다.

계층적인(Hierarchical) 인코딩을 사용하는 방법에서 서버는 계층적으로 인코딩된 비디오 스트림을 갖는다[13-16]. 네트워크 대역폭에 여유가 있을 경우, 서버는 비디오 스트림의 더 높은 레이어까지 전송을 하며, 네트워크 대역폭이 줄어들 경우, 현재 활성화된 레이어를 제거하여 전송함으로써 네트워크 상태변화에 적응적으로 전송률을 조절하게 된다. 계층적인 인코딩을 사용하는 품질 적응 기법은 최소한의 비디오 스트림 품질을 보장하는 기본 레이어(Base-Layer) 데이터의 클라이언트 측 버퍼 언더플로우를 예방하는 동시에 적절한 상위 레이어(Enhancement-Layer) 데이터의 추가 및 제거를 통해 사용자에게 제공되는 비디오 스트림 품질의 최대화를 목적으로 한다.

Rejaie는 [17,18]에서 계층적으로 인코딩된 비디오 스트림을 TCP의 혼잡제어 메커니즘인 AIMD 기법에 의해 결정된 전송률로 전송하는 품질 적응 기법을 제안하였다. 상대적으로 오랜 시간의 전송률 변화는 계층적으로 인코딩된 비디오 스트림의 레이어 조절에 의해 처리하며, AIMD 기법의 순간적인 전송률 변화는 클라이언트 버퍼링을 통해 처리하게 된다.

Rejaie의 품질 적응 기법에서는 네트워크 상태기반 레이어 조절 메커니즘에 추가적으로 활성화된 레이어들 사이에서의 대역폭 할당 메커니즘을 제안하고 있다. 활성화된 레이어간의 대역폭 할당 메커니즘은 현재 클라이언트의 버퍼링된 데이터양을 기반으로 각 레이어별로 대역폭을 할당하는 방법으로, 네트워크의 가용 대역폭이 줄어들 경우, 상위 레이어의 비활성화를 통해 제공되는 서비스 품질이 감소하는 것을 최대한 예방하게 된다[18].

Rejaie의 품질 적응 기법이 제안된 이후에도 여러 가지 관련 연구들이 수행되었다. Nick은 [19]에서 Rejaie의 품질 적응 기법을 확장하여 혼잡제어 메커니즘으로 AIMD 기법대신, TCP 친화적인 전송 프로토콜인 SQRT를 사용하여 품질 결정에 필요한 버퍼 리소스를 감소시켰으며, SQRT 사용에 따른 적절한 대역폭 할당 및 관리 메커니즘을 제안하였다. Naoki는 기존의 TCP 트래픽과 네트워크 대역폭을 공평하게 사용하는 MPEG-4 (Moving Picture Experts Group) 비디오 스트림 전송

기법을 제안하였다[20]. 2.1절에 설명된 TFRC를 이용하여 TCP 친화적인 전송률을 계산하고, 계산된 전송률에 적합한 비디오 스트림 전송을 위해 비디오 스트림의 품질을 결정하는 기법을 소개하였으며, 계층적 인코딩의 한 종류인 FGS(Fine Granular Scalability) 코딩된 비디오 스트림을 사용하였다.

인터넷 환경에서 효율적인 비디오 스트리밍을 위해서는 네트워크 상태를 기반으로 하는 TCP 친화적인 전송률 조절 기법만으로는 한계가 있다. 네트워크의 안정성 향상이 사용자에게 서비스되는 비디오 스트림의 품질 향상을 보장하지는 않기 때문이며, 이러한 한계를 극복하기 위해 TCP 친화적인 전송률에 맞추어 서비스되는 비디오 스트림의 품질을 결정하는 품질 적응 기법들이 제안이 되었고, 이러한 품질 적응 기법들은 계층적인 인코딩 방식을 기반으로 하였다. 하지만, 재생의 연속성과 같은 콘텐츠 재생 특성을 고려하지 않아 사용자에게 QoE(Quality of Experience)를 제공하지 못하는 문제점을 가지고 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 인터넷 혼잡상황에서 멀티미디어 스트리밍 서비스의 QoE 향상을 위한 새로운 전송률 제어기법인 NCAR(Network and Client-Aware Rate control)를 제안하였다. 다음의 3장에서 본 논문에서 제안하는 네트워크와 클라이언트 상태를 이용한 비디오 스트림 전송률 조절 기법에 대해서 설명하도록 한다.

3. NCAR(Network and Client-Aware Rate control)

3.1 중단간 멀티미디어 스트리밍 시스템

기존 멀티미디어 스트리밍 시스템의 주요 동작 모드는 스트리밍 서버 내 혼잡상황에 따라 전송률을 조절하는 전송률 적응(Rate Adaptation) 모듈이다. 이 전송률 적응 모듈을 통해 패킷 손실을 줄여 혼잡상황을 줄이고, 경쟁하는 TCP 트래픽과 친화적으로 동작하며, 전송률의 변화를 줄여 스트리밍 전송에 적합한 동작을 수행한다. 이 모듈은 클라이언트의 네트워크 계층으로부터 수신된 패킷들의 네트워크 상태 정보를 피드백 받아 동작하는 네트워크 인지형 전송률 조절 기법을 사용하고 있다. 그러나, 네트워크 적응적인 전송률 조절 기법은 멀티미디어 스트리밍 서비스의 사용자 측면의 품질을 고려하지 않아 멀티미디어 스트리밍 서비스의 주요 요소인 QoE를 제공하지 못한다.

멀티미디어 스트리밍 서비스의 QoE를 개선하고 네트워크의 안정성과 공정성을 제공하기 위해, 본 논문에서 제안한 NCAR(Network and Client-Aware Rate control) 기법을 서비스 하기 위한, 중단간 멀티미디어 스트리밍 시스템을 그림 1과 같이 제안하였다.

제안한 시스템은 다음과 같은 동작을 수행한다. 먼저, 스트리밍 서버 내 SVC(Scalable Video Coding) 부호화 기법으로 인코딩된 비디오 데이터는 전송하기 전 흐름제어 모듈(Flow Control Module: FCM)을 통해 SVC 비트스트림 Extractor가 제어된다[23]. 이 제어를 통해 SVC로 인코딩된 비디오 데이터는 적당한 비트스트림의 품질로 조절된다. 그리고, TCP 친화적인 프로토콜을 기반으로 동작하는 혼잡제어 모듈(Congestion Control Module: CCM)을 통해 이 비트스트림은 클라이언트로 전송된다. 스트리밍 클라이언트에서는 수신된 비디오 데이터가 클라이언트 버퍼에 버퍼링되는 동안 패킷

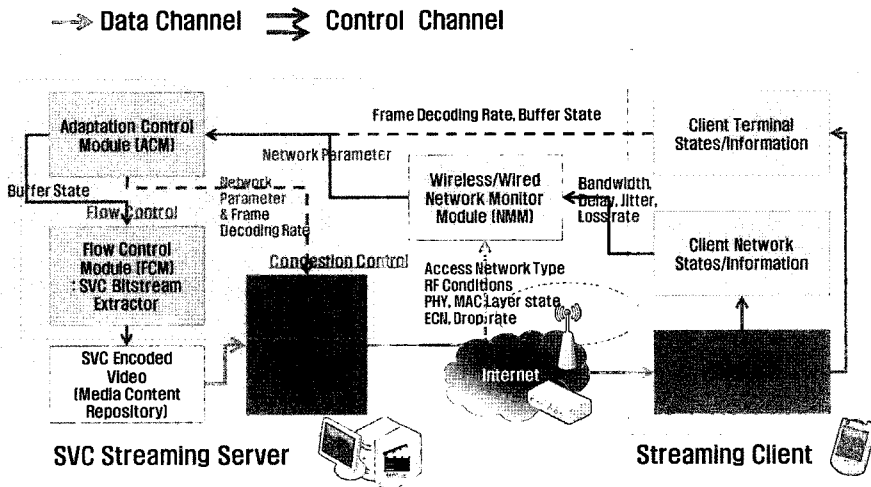


그림 1 QoE를 개선하기 위한 중단간 멀티미디어 스트리밍 시스템

유실, 패킷 지연 그리고 클라이언트 버퍼 상태를 수집하여 네트워크 모니터 모듈(Network Monitor Module: NMM)에 전달한다. 결과적으로 서버는 이렇게 피드백되는 정보를 이용하여 반복적으로 네트워크와 클라이언트 상태를 예측하고 멀티미디어의 품질과 전송률을 조절한다. 이 결과 제안한 시스템은 클라이언트의 버퍼상태와 네트워크 대역을 예측하여 사용자에게 좋은 영상 및 서비스 만족도를 보장할 수 있다.

제안한 멀티미디어 스트리밍 시스템의 주요 요소는 네트워크 모니터 모듈(NMM), 적응제어 모듈(Adaptation Control Module: ACM), 혼잡제어 모듈(CCM)과 흐름제어 모듈(FCM)로 크게 4가지로 분류 할 수 있다. 먼저, 네트워크와 단말의 상태를 감시하여 수집하는 NMM은 주기적으로 네트워크 상태를 감시하여 스트리밍 서버와 클라이언트 사이의 네트워크 상태를 알기 위해 다양한 네트워크 정보를 수집 및 계산한다. 혼잡제어와 흐름제어를 상황에 따라 선택하여 판단하는 ACM은 가용 대역폭 내 공정성을 제공하는 범위에서 스트리밍 서버의 전송 비트율을 통제하고, 이 조절을 위한 제어 방법을 선택한다. 혼잡제어를 수행하는 CCM은 ACM을 통해 통제된 전송 비트율에 맞게 TCP 친화적으로 패킷

간 시간 간격을 조절한다. 마지막으로 흐름제어를 수행하는 FCM은 ACM을 통해 통제된 정보를 기반으로 SVC 비트스트림 Extractor를 이용하여 멀티미디어 스트리밍의 비트스트림 품질을 조절한다.

즉, ACM은 주기적으로 NMM을 통해 수집된 정보로 네트워크와 클라이언트의 상태를 예측하여 FCM 또는 CCM 중 적합한 방법을 선택하여 서버의 최종 전송률을 조절 및 제어한다. 클라이언트의 버퍼 상태에 따라 적응하기 위해 FCM을 통해 비트스트림의 품질 등급이 조절되거나, 네트워크의 가용대역폭 상태에 적응하기 위해 CCM을 통해 전송 비트율이 조절된다. 이를 통해 스트리밍 서버의 전송률을 조절한다.

3.2 멀티미디어 스트리밍의 전송률 조절

그림 2는 3.1절에서 제안한 멀티미디어 스트리밍 시스템을 통하여 전송되는 멀티미디어 스트리밍 전송률 조절 동작을 도시화한 것이다. 그림 2와 같이 제안하는 전송률 조절 방법은 크게 두 개의 전송률 조절 방법을 가지고 있다. 하나는 혼잡제어이고 다른 하나는 흐름제어를 기반으로 전송률 조절 방법이다. 혼잡제어를 기반으로 하는 동작은 그림 3과 같이 다양한 네트워크 상태 정보를

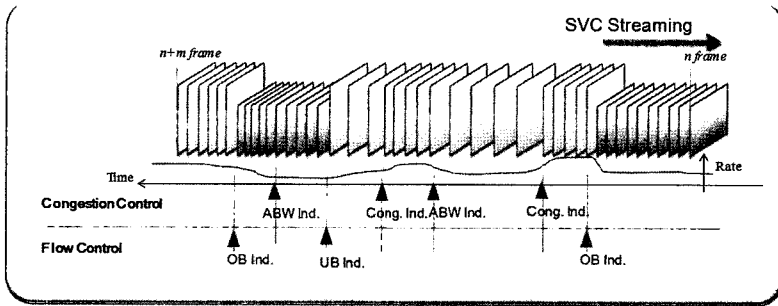


그림 2 제안하는 멀티미디어 스트리밍 동작 도시

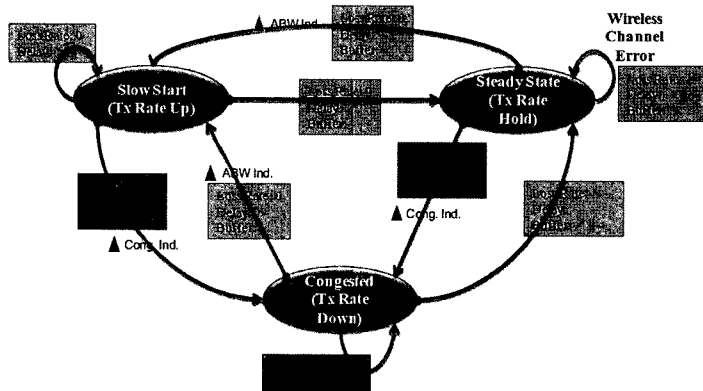


그림 3 다양한 네트워크 상태 정보

Cong. Ind.)과 가용 대역폭의 발생(Available Bandwidth Indication: ABW Ind.)으로 네트워크의 혼잡 정도를 판단하여 서버의 전송 비트율을 조절하는 네트워크 인지형 전송률 조절 기법을 제안하였다. 또한, 혼잡 상태를 정확히 인지하기 위해 패킷 유실이나 패킷 지연과 같은 정보 이외에도 대역폭의 변화율, 전송률 조절 주기와 같은 정보를 추가적으로 이용한다.

그러나, 제안한 전송률 기법 내 혼잡제어 동작만으로는 오버플로우나 언더플로우와 같은 클라이언트에서 발생하는 스트리밍 데이터의 흐름제어를 충족 시키는 못한다. 본 논문에서는 네트워크 인지형 전송률 조절 방법에 추가적으로 클라이언트의 상태를 인지하여 흐름제어를 기반으로 동작하는 전송률 조절 기법을 제안하였다. 제안한 전송률 기법 내 흐름제어 동작은 오버버퍼링 발생(Overbuffering Indication: OB Ind.)과 언더버퍼링 발생(Underbuffering Indication: UB Ind.)으로 클라이언트의 상태를 판단하며, 이를 통해 전송하는 스트리밍 데이터의 비트스트림 크기를 조절하여 서버의 미디어 전송률을 관리한다. 클라이언트 측면의 품질을 보장하기 위해, 흐름제어 동작은 혼잡제어 동작과 구분되어 혼잡 상황 및 클라이언트 상태에 따라 스트리밍 데이터의 흐름을 제어해야 한다. 그리고, 흐름제어 동작을 정확히 구분하기 위해 클라이언트가 서버로 보내는 버퍼의 크기, 화면 크기, 배터리 상태 등 다양한 정보를 이용하여 흐름제어를 수행할 수 있다. 본 논문에서는 영상 재생의 연속성을 보장하기 위해 클라이언트 버퍼의 크기 정보를 피드백 받아 SVC Extractor로 비트스트림 크기 조절한다.

3.3 AIHD(Additive Increase Heuristic Decrease)

멀티미디어 스트리밍의 전송률 제어 방법들에 있어서 단순히 가용대역폭에 따라 전송률을 조절하는 것 보다 다른 트래픽에 친화적으로 전송률을 조절하여 부드러운 재생품질을 보장하는 것이 좋다. 스트리밍 서버의 중단 간 혼잡제어 모듈은 가용대역폭을 파악하고 혼잡상황을 감시하여 데이터 전송률을 조절한다. 이러한 전송률 변화는 클라이언트에서 수신하는 데이터의 변화를 야기시켜 잦은 영상의 품질 변화를 발생시키거나 변화를 수용하여 부드럽게 재생할 수 있도록 많은 버퍼를 요구한다. 따라서, 멀티미디어 스트리밍의 전송률 조절시 전송률의 변화의 폭을 줄여야 한다.

본 논문에서는 전송률 변화의 폭을 줄이기 위해 새로운 혼잡제어 알고리즘인 AIHD(Additive Increase Heuristic Decrease)를 제안하였다. 제안한 알고리즘은 TCP의 AIMD(Additive-Increase/Multiplicative-Decrease) 방법과 비교하여 동일 혼잡상황에 대해서 낮은 전송률 변화를 보였다. 제안한 알고리즘의 전송률 증가(Incre-

ase: I)와 감소(Decrease: D)는 다음 식 (2), (3)을 따른다.

$$I : R_n = R_c + \left(\frac{Packet\ Size}{RTT} \right) \cdot U_f \quad (2)$$

$$D : R_n = \beta \times R_t + (1 - \beta) \times R_c ; 0 < \beta < 1 \quad (3)$$

R_c 는 현재 서버가 전송하는 전송률, R_n 는 증가와 감소 동작을 통해 결정되는 새로운 전송률, R_t 는 TCP model을 기반으로 현재 시점에서 피드백 정보를 통해 예측되는 가용대역폭, β 는 AIHD 알고리즘의 파라미터, 피드백 정보의 간격에 따른 업데이트 가중치 $U_f = \Delta T / RTT$ 이다.

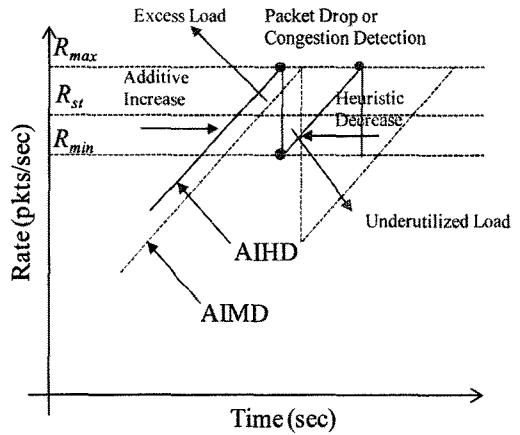


그림 4 AIHD 알고리즘의 전송률 변화

AIHD의 전송률 증가와 감소의 동작은 그림 4와 같이 동작한다. AIHD 알고리즘의 전송률 증가는 보통 R_{min} 부터 최대 지속 전송률(Maximum Sustainable Rate)인 R_{st} 를 지나 증가한다. 그리고, 패킷 유실, 혼잡 상황 인지를 통해 R_{max} 로부터 R_{min} 로 휴리스틱하게 전송률을 감소 시킨다.

스트리밍 서버가 전송률 변화 없이 안정된 상태로 전송률을 전송하기 위해서는 현재 전송률 R 이 R_{st} 을 초과하는 데이터량(Excess Load)과 R 이 R_{st} 보다 낮은 구간의 이용하지 못하는 데이터량(Underutilized Load)이 같아야 한다. 그래서, 최대 지속 전송률은 식 (4)을 만족해야 한다.

$$R_{st} > \frac{R_{min} + R_{max}}{2} \quad (4)$$

또한, 본 논문에서 제안한 AIHD 알고리즘인 식 (3)을 이용하여 식 (5)와 같이 정의할 수 있다.

$$R_{st} > R_{max} - \beta \times \left(\frac{R_{max} - R_t}{2} \right) \quad (5)$$

그리고, AIMD보다 전송률 변화를 줄이기 위해서는 $R_{st}(AIHD) > R_{st}(AIMD)$ 를 만족시켜야 하므로 식 (6)

과 같은 조건을 충족 시켜야 한다.

$$\frac{R_{max}}{4} > \beta \times \left(\frac{R_{max} - R_t}{2} \right) \quad (6)$$

일반적으로 $R_{max} > R_t$ 를 만족시키므로, 식 (6)을 만족시키기 위해서는 식 (7)을 충족해야 한다.

$$1 < 2\beta / (2\beta - 1) \quad (7)$$

따라서, β 는 식 (8)과 같은 조건을 만족시켜야 한다.

$$\beta > 0.5 \quad (8)$$

본 논문에서 제안하는 AIHD는 식 (8)을 만족시킬 경우 기존 AIMD 기법의 혼잡제어 방법 보다 낮은 전송률 변화폭을 가진다. 그리고, 이를 통해 작은 버퍼와 품질의 변화를 최소화시킬 수 있다.

3.4 네트워크 인지형 혼잡제어

멀티미디어 스트리밍 서비스 응용에서 네트워크의 혼잡 상태에 따라 스트리밍의 전송률을 조절하는 것은 네트워크의 안정성 및 혼잡제어를 위한 중요한 기능이다. 이를 통해 패킷 손실율을 줄이고 TCP 플로우와 같은 다른 트래픽들과 공정하게 대역폭을 공유 할 수 있다. 본 논문에서는 AIHD를 이용하여 부드러운 전송률 조절 동작 특성을 제공하며, 패킷 손실, 패킷 지연 및 대역폭의 변화 등을 고려하여 동작하는 네트워크 인지형 혼잡제어 방법을 제안하였다.

본 논문에서 제안한 NCAR의 혼잡제어 동작은 다음과 같이 동작한다. 먼저, 클라이언트는 네트워크 상태를 감시하고 그와 관련된 정보를 수집한다. 서버는 클라이언트가 수집한 정보들을 이용하여 현재 네트워크의 가용 대역폭을 예측하고 그 대역폭에 적합한 전송률을 조절한다. 서버는 수집한 정보를 기초로 Padhye가 제안한 TCP 전송률 모델링 공식인 식 (1)을 이용하여 현재 네트워크의 가용 대역폭을 예측한다.

예측한 가용 대역폭을 이용하여 그림 5와 같은 방법으로 네트워크 혼잡 여부를 판단하고, AIHD를 이용하여 전송률의 증가와 감소를 조절한다. 이러한 동작은 제안한 시스템의 ACM에서 판단하여 CCM에서 관련 동작을 처리한다.

```

Calculating RateABW(Target Rate) by using equation based TCP model (Padhye et al)
if (RateABW > Ratecurrent)
    ABW Indication:
        Slow Start (Is > It)
else if (RateABW < Ratecurrent)
    Cong Indication:
        Congestion Control (Ic > It)
else
    Steady State (Is = It)
    
```

그림 5 NCAR의 혼잡제어 동작

ACM에서 판단하는 NCAR 기법의 혼잡제어 동작을 정리하면 다음과 같다.

- 가용 대역폭(Rate_{ABW})이 현재 전송률(Rate_{current})보다 크다면, 혼잡이 없다고 판단하여 주기적으로 CCM을 통해 전송률을 증가시킨다.

- 가용 대역폭(Rate_{ABW})이 현재 전송률(Rate_{current})보다 작다면, 혼잡이 있다고 판단하여 즉시 CCM을 통해 전송률을 감소시킨다.

이러한 혼잡제어를 위해 NCAR의 CCM은 본 논문에서 제안한 AIHD를 이용하여 전송률을 증가와 감소시킨다. NCAR는 패킷간 전송 간격을 조절하여 전송률을 변경한다. 전송률을 점진적으로 증가시키기 위해 전송 간격은 AIHD의 증가 공식인 식 (2)를 기반으로 식 (9)와 같이 계산한다.

$$I_n = I_c - \frac{(I_c)^2}{(RTT + I_c)} \cdot U_f \quad (9)$$

여기서, I_n 새로운 패킷간 전송 간격이고, I_c 는 현재 패킷간 전송 간격이다.

혼잡상황이 확인 후 전송률을 배수적으로 감소시키기 위해 전송 간격은 AIHD의 감소 공식인 식 (3)을 기반으로 식 (10)과 같이 계산한다.

$$I_n = I_c \times \left(\frac{I_t}{\beta \cdot I_c + (1-\beta) \cdot I_t} \right) \quad (10)$$

여기서, I_t 는 예측한 가용 대역폭(Rate_{ABW})을 패킷간 전송 간격으로 환산한 값이다. 그리고, 잦은 전송률 조절에 따른 전송률 변화폭을 줄이기 위해 NCAR의 혼잡제어 동작은 클라이언트로부터 수신한 피드백 메시지의 주기에 따라 동작한다.

3.5 클라이언트 인지형 흐름제어

본 논문에서 제안한 NCAR은 멀티미디어 스트리밍 서비스 중 서버와 클라이언트 사이의 전송되는 비트스트림의 비트율을 관리하기 위해 클라이언트의 수신 버퍼 상태와 수신된 미디어의 상태의 특성에 따라 SVC 비트스트림 Extractor를 이용하여 전송 영상의 품질을 조절하는 흐름제어 기법을 가지고 있다. 이를 통해 클라이언트 버퍼의 오버플로우나 언더플로우 상황에서도 연속적이고 부드러운 재생이 가능하도록 미디어 재생의 불연속성을 제거하고 안정된 버퍼할당과 낮은 재생지연 시간 특성을 제공한다.

본 논문에서 제안한 NCAR의 흐름제어 동작은 다음과 같이 동작한다. 먼저, 클라이언트는 수신 버퍼의 상태감시하고 그와 관련된 정보를 수집한다. 서버는 클라이언트가 수집한 정보들을 이용하여 앞으로의 버퍼 상태의 변화를 예측하고 미디어 재생의 연속성을 유지시키기 위해 전송 영상의 품질 등급을 조절한다. 버퍼 상태의 변화는 현재 버퍼의 길이($qlen_c$), 전송 영상의 비트스트림 품질 등급 그리고 품질 등급 별 버퍼의 최대($qmax_{th}$), 최소($qmin_{th}$) 임계 값을 이용하여 예측한다.

품질 등급의 변화를 선택하는 최대, 최소 임계 값은 식 (8), (9)와 같다.

$$qmax_{th} = \int_{t_0}^{t_0 + T_{max}} \frac{R_{svc}(t)}{Packet\ Size} dt \quad (11)$$

$$qmin_{th} = \int_{t_0}^{t_0 + T_{min}} \frac{R_{svc}(t)}{Packet\ Size} dt \quad (12)$$

여기서, $R_{svc}(t)$ 은 현재 시간(t_0) 서비스하는 SVC 비트스트림의 전송률, 클라이언트에서 기대하는 최대 수용 가능한 버퍼링 시간 T_{max} 과 기대하는 최소 요구 버퍼링 시간 T_{min} 이다. 이 최대, 최소 버퍼링 시간은 미디어의 특성 및 제공하는 서비스 사업자의 요구 사양에 따라 선택된다.

현재 클라이언트 버퍼의 자원 상태를 예측한 후, 그림 6과 같이 혼잡제어를 수행한 후 흐름제어를 수행한다. 각 혼잡제어 구간에 따라 $qlenc_c$ 와 $qmin_{th}$, $qmax_{th}$ 을 이용하여 버퍼의 오버플로우와 언더플로우 상황을 판단한다. 이러한 동작은 제안한 시스템의 ACM에서 판단하여 FCM에서 관련 동작을 처리한다.

ACM에서 판단하는 NCAR 기법의 흐름제어 동작을 정리하면 다음과 같다.

- 가용 대역폭($Rate_{ABW}$) 이 현재 전송률($Rate_{current}$) 보다 크고 현재 버퍼길이($qlenc_c$)가 최대 임계 값($qmax_{th}$) 보다 크다면, 오버플로우 상황으로 판단하여 전송 영상의 품질 등급을 한 단계 올린다.
- 가용 대역폭($Rate_{ABW}$)이 현재 전송률($Rate_{current}$) 보다 작고 현재 버퍼길이($qlenc_c$)가 최소 임계 값($qmin_{th}$) 보다 작다면, 언더플로우 상황으로 판단하여 전송 영상의 품질 등급을 한 단계 내린다.

NCAR의 흐름제어 동작은 SVC 비트스트림 Extractor를 이용하여 전송 영상의 품질을 조절한다. SVC 비

트스트림 Extractor는 시간, 공간, 품질의 인코딩 방법에 따라 다양하게 비트스트림의 품질을 조절 할 수 있는 확장성을 제공한다. 이 SVC 비트스트림은 하나의 기본 계층(base layer)에 다수개의 상위 계층(enhancement layer)으로 구성되어 있다. 따라서, 상위 계층의 개수가 증가하면 영상의 품질이 좋아지지만 이 영상을 스트리밍하기 위한 요구 대역폭이 증가한다.

NCAR는 버퍼의 오버플로우 상황이 감지되면 영상의 품질 등급을 올려 서비스에 요구되는 대역폭을 올린다. 그래서, 클라이언트에서 소비하는 데이터의 크기는 증가 되고 오버플로우를 해결한다. 또한, 현재 서비스하는 영상 등급보다 높은 등급의 영상을 서비스하여 좋은 화질을 제공하고, 이를 통해 멀티미디어 스트리밍 서비스의 화질에 대한 QoE를 향상 시킬 수 있다. 반대로 언더플로우 상황을 감지하면 영상의 품질 등급을 내려 서비스에 요구되는 대역폭을 줄인다. 그래서, 클라이언트에서 소비하는 데이터의 크기는 감소되고 언더플로우 문제는 해결된다. 또한, 언더플로우 현상을 해결하여 미디어 재생의 불연속성을 제거하고, 이를 통해 멀티미디어 스트리밍 서비스의 연속 재생에 대한 QoE를 향상 시킬 수 있다.

멀티미디어 스트리밍 서비스의 특성상 적정 품질을 유지하기 위해 본 논문에서 제안한 NCAR는 다음과 같은 예외 처리 동작을 수행한다.

현재 전송 영상의 품질 등급이 최상 등급이고 NCAR의 FCM에서 흐름제어를 위해 그 이상의 등급으로 조절을 요구할 때, 오버플로우 현상을 피하기 위해 NCAR는 패킷간 전송간격을 조절하는 CCM을 이용하여 식 (13)과 같이 전송간격을 조정한다.

```

Calculating RateABW (Target Rate) by using equation based TCP model (Padhye et al)

if ( RateABW > RateCurrent ) {
    ABW Indication :
    CC - Slow Start ( Ic > In )
    if ( qlencc > qmaxth )
        OB Indication:
        Flow Control - SVC Extractor Quality (Up)
} else if ( RateABW < RateCurrent ) {
    Cong Indication :
    CC - Congestion Control ( In > Ic )
    if ( qlencc < qminth )
        UB Indication :
        Flow Control - SVC Extractor Quality (Down)
} else
    Steady State ( Ic = In )
    
```

그림 6 NCA 알고리즘의 의사코드

$$I_n = \omega \times \frac{Packet\ Size}{Rate_{svc_Extractor}(Profile\ Maximum\ Level)\ Current}} \quad (13)$$

여기서, ω 는 감소변수이고 $Rate_{svc_Extractor}(Profile\ Maximum\ Level)\ Current$ 는 현재 서비스하는 영상 품질 등급의 요구 전송률이다.

그리고, 현재 전송 영상의 품질 등급이 최하 등급이고 NCAR의 FCM에서 흐름제어를 위해 그 이하의 등급으로 조절을 요구할 때, 미디어 재생의 연속성을 보장하기 위해 NCAR는 패킷간 전송간격을 식 (14)와 같이 유지시킨다. 멀티미디어 스트리밍 서비스의 특성상 단절 없이 실시간적으로 영상의 재생을 보장시켜야 한다.

$$I_n = \frac{Packet\ Size}{Rate_{svc_Extractor}(Profile\ Minimum\ Level)\ Current}} \quad (14)$$

4. 실험 및 평가

본 장에서는 혼잡 네트워크에서의 NCAR의 성능을 실험 및 평가하였다. NCAR의 성능을 평가하기 위해 LBNL(Lawrence Berkeley National Laboratory)의 ns 2(Network Simulator)를 이용하여 시험 및 평가하였다 [23]. 그리고, 시험을 위해 "SOCCER_352x288_30_orig_02_yuv" 영상을 JSVM(Joint Scalable Video Model) [24]을 이용하여 표 1과 같이 4개의 Quality Layer를 가지는 SVC 비트스트림 영상을 구성하였으며, 다음 항목에 대해 시험하였다.

- 혼잡 네트워크에서의 성능 평가
- 혼잡 정도에 따른 성능 비교 평가

성능 비교를 위해 멀티미디어 스트리밍 서비스 환경에서 네트워크 인지형 전송률 제어 방법을 사용하는 경우와 전송률 제어 방법을 사용하지 않는 경우, 그리고 본 논문에서 제안한 NCAR를 사용하는 경우에 대한 성

능을 평가하였다. 시뮬레이션 네트워크 환경은 그림 7과 같으며 동일 네트워크 환경에서 스트리밍 서버의 전송률 제어 방법을 3가지 방법으로 선택하여 시험하였다.

그림 7에서 R1과 R2 라우터 사이에 혼잡 구간을 구성하였고 혼잡 구간의 지연은 20ms로, 대역폭은 2Mbps로 설정하였다. 혼잡상황을 만들기 위해 이 혼잡 구간은 2개의 TCP 플로우와 1개의 비디오 세션 플로우가 공유하게 설정하였다. 비디오 세션의 스트리밍 방법은 RTP기를 기반으로 전송하게 설정하였으며 RTCP RR 메시지를 통해 클라이언트로부터 네트워크, 단말관련 정보를 피드백 받도록 구성하였다. RTCP RR의 전송 주기는 0.5 초로 설정하였다. 제안한 NCAR의 β 값은 0.75로 설정하였다.

4.1 혼잡네트워크에서의 성능 평가

본 절에서는 제안한 NCAR 기법의 동작 특성에 따른 성능 평가를 그림 7과 같은 네트워크 환경에서 각 전송률 제어 기법을 비교시험 하였다. 시뮬레이션은 총 200초 동안 진행되며 비디오 세션 플로우는 0초부터 200초 구간까지 전송되고, 시뮬레이션 시작 20초 후 경쟁 트래픽으로 첫 번째 TCP 플로우의 트래픽 전송이 시작된다. 그리고, 시뮬레이션 시작 50초 후 또 다른 경쟁 트래픽으로 두 번째 TCP 플로우의 트래픽이 전송이 시작되고 50초 후 종료된다. 이 시험에서는 시뮬레이션 구간 50초부터 100초 사이의 극심한 혼잡구간이 생성된다. 먼저, 제안한 NCAR 기법의 동작특성을 도식화 하면 그림 8과 같다.

그림 8(a)는 각 플로우의 전송률 변화와 혼잡 정도 및 클라이언트의 버퍼 상태에 따라 SVC의 품질 등급의 변화를 보여주고 있다. 그림 8(a)에서 보는 것처럼 본 논문에서 제안한 NCAR 기법은 다른 TCP 플로우와 친화적으로 네트워크를 공유함을 알 수 있다. 그림 8(b)는

표 1 시험 SVC 비트스트림의 특성

JSVM Layer	Resolution	Frame Rate	Bitrate (kbps)	Spatial level	Temporal level	Quality level	Y-PSNR	U-PSNR	V-PSNR
Layer 20	352x288	30	594.8	1	4	0	31.0413	39.7593	41.7566
Layer 33	352x288	30	813.5	1	4	1	32.4077	40.8846	42.9231
Layer 34	352x288	30	955.3	1	4	2	33.4442	41.3644	43.3295
Layer 35	352x288	30	1083.4	1	4	3	33.4442	41.3644	43.3295

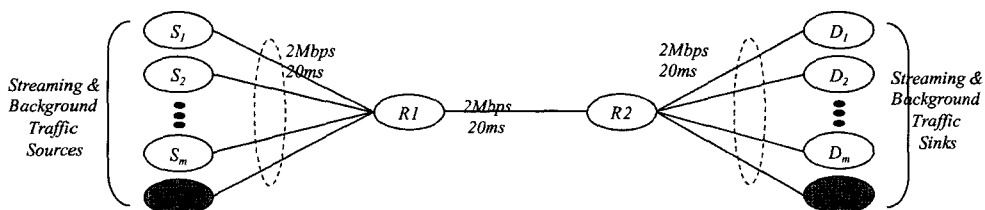
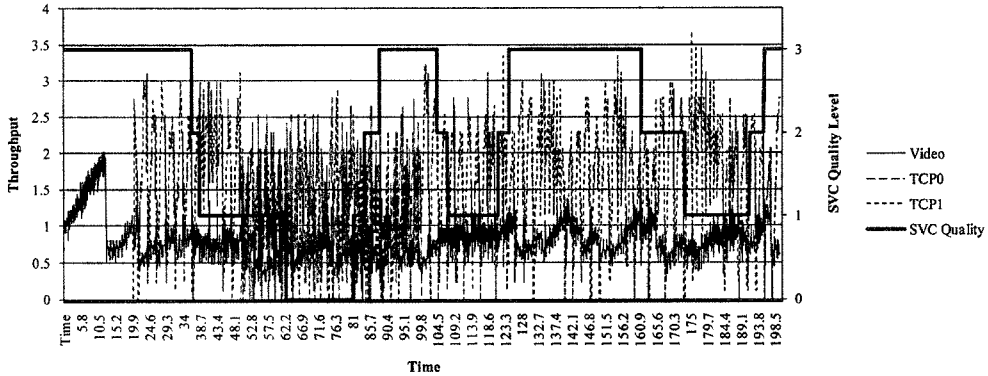
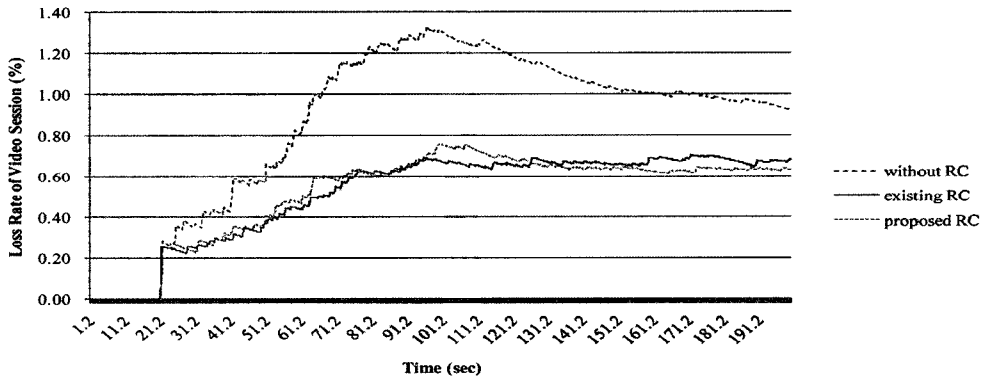


그림 7 NCAR 기법 시험을 위한 시험 네트워크 환경



(a) 각 플로우의 전송률 변화와 SVC 품질 등급



(b) 스트리밍 클라이언트의 버퍼길이와 SVC 품질 등급

그림 8 NCAR 기법의 동작결과

NCAR 기법을 사용하는 클라이언트 버퍼의 크기와 버퍼링된 정보 량에 따른 전송 미디어의 품질 등급 조절의 변화를 확인 할 수 있다. 그림 8(b)에서 보는 것처럼 네트워크의 혼잡상황이 가중되어 전송률을 감소시킬 경우 클라이언트 버퍼 내 버퍼링된 정보량이 감소하여 미디어 재생의 연속성이 보장될 수가 없어진다. NCAR는 이러한 상황을 감지하여 미디어의 품질 등급을 낮춰 최소 요구 버퍼링량을 충족시켜 미디어의 연속성을 보장한다. 그리고, 반대의 경우 오버플로우를 방지하기 위해 버퍼 내 버퍼링된 정보량이 커질 경우 영상의 품질 등급을 올려 화질을 개선하고 오버플로우를 방지함을 알 수 있다.

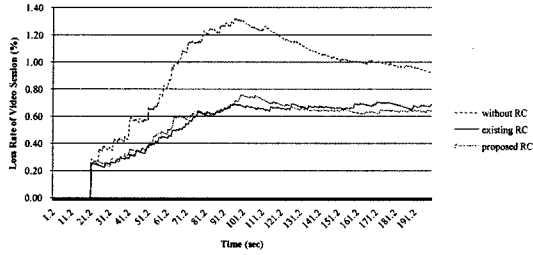
본 논문에서 제안한 NCAR 이외에 다른 전송률 제어 (Rate Control: RC)기법과의 시험결과는 그림 9와 같다.

제안한 NCAR 기법은 그림 9(a)와 같이 기존 전송률 제어 방법과 동일한 패킷 유실을 보여준다. 그림 8(a)와 그림 9(a)를 통해 NCAR 기법은 패킷 유실 감소와 높은 링크 활용도, 그리고 플로우간 공정성을 제공함을 알

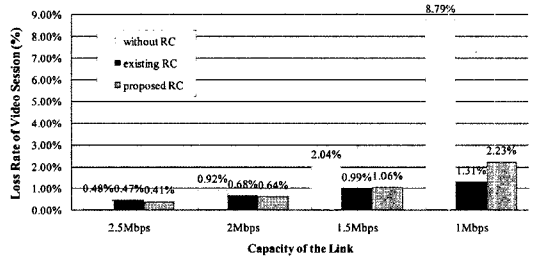
수 있다. NCAR는 그림 9(b)와 같이 기존 전송률 제어 방법에서 요구하는 많은 버퍼공간과 넓은 버퍼길이 변화를 요구하지 않는다. 이를 통해 안정된 버퍼할당 및 적정 버퍼링 데이터를 유지함을 확인 할 수 있다. 그리고, 그림 9(c)와 같이 기존 전송률 제어 방법은 객관적인 품질지표인 PSNR (Peak Signal to Noise Rate)의 변화는 없지만 클라이언트 상태 및 미디어 재생 특성에 대한 고려가 없어 동작하여 사용자에게 부드럽고 연속적인 서비스를 제공하지 못한다. 그러나, NCAR는 낮은 PSNR 변화를 보이지만 시물레이션 전 구간 멀티미디어 스트리밍 서비스의 끊어짐 없이 미디어 재생의 연속성을 보장하였다. 이는 IPTV나 Mobile IPTV와 같은 실시간성을 요구하는 멀티미디어 스트리밍 서비스 사용자의 서비스 연속성에 대한 QoE를 개선하고 보장시켜 준다.

4.2 혼잡 정도에 따른 성능 비교 평가

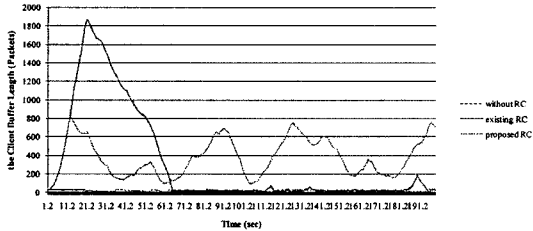
본 절에서는 혼잡 정도에 따라 NCAR 기법의 성능을 다른 전송률 기법과 비교 평가하였다. 그림 7과 같은 동



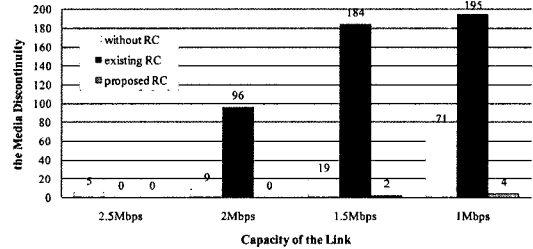
(a) 패킷 유실을 변화



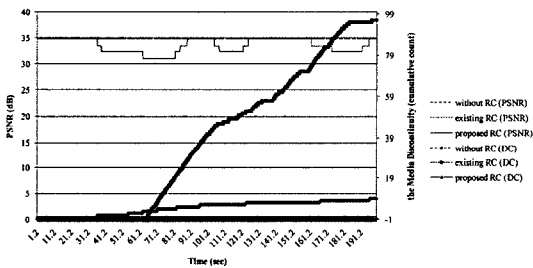
(a) 비디오 세션의 패킷 유실율



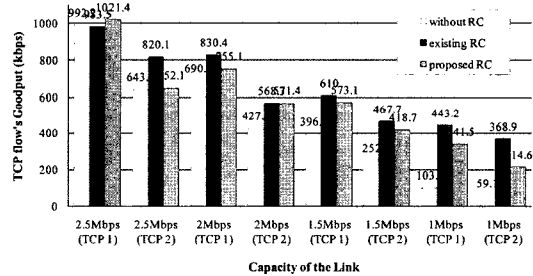
(b) 클라이언트 버퍼 길이 변화



(b) 미디어 재생의 불연속성 횟수



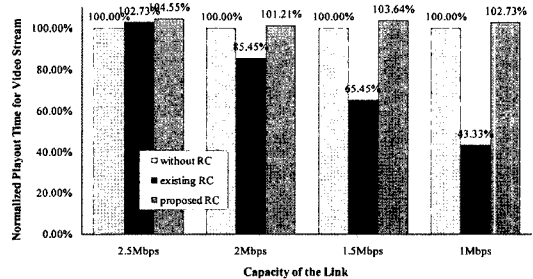
(c) PSNR과 미디어 불연속성 횟수
 그림 9 각 전송률 제어기법 성능비교



(c) 각 TCP 플로우의 유효전송률

일 네트워크 환경에서 혼잡 구간의 링크 대역폭을 1Mbps에서 2.5Mbps로 변경하며 성능을 평가하였다. 비디오 세션의 패킷 유실율, 미디어 재생의 불연속성, 각 TCP 플로우의 유효전송 속도 그리고, 재생시간에 대해서 각 전송률 기법 별 시험 결과는 그림 10과 같다.

그림 10(a)와 (c) 같이 링크의 자원이 부족한 환경일 수록 전송률 제어 방법을 사용하지 않는 경우보다 사용하는 경우 낮은 패킷 유실율과 불공정성 문제가 발생함을 알 수 있다. 이는 혼잡 정도가 극심한 환경일수록 스트리밍 서버에서 전송률 제어 방법을 사용해야 링크 자원의 활용도가 높아짐을 알 수 있다. 본 논문에서 보장하려고 한 QoE 측면에서 미디어의 연속성은 그림 10(b)와 같이 NCAR 기법이 매우 우수함을 알 수 있다. 하지만, 클라이언트의 상황을 인지하지 않는 일반적인 전송률 제어 방법의 경우 전송률 제어를 하지 않는 방법보다 매우 나쁜 품질을 제공한다. 이는 네트워크 친화적인 동작을 위해 QoE에 대한 고려 없이 지나치게 전송



(d) 비디오 스트림의 재생시간
 그림 10 혼잡 정도에 따른 성능비교

률을 조절해서 발생하였다. 마지막으로 시뮬레이션 구간 동안 정상적으로 비디오 스트림이 재생된 시간을 평가하였다. 그림 10(d)와 같이 네트워크 상황만 인지하여 전송률을 조절하는 방법은 혼잡 정보가 극심한 환경일 수록 실시간 특성을 가지는 스트리밍 서비스인 IPTV와

같은 응용서비스의 서비스가 불가능 함을 보여준다. 하지만, 본 논문에서 제안한 NCAR 기법은 네트워크 상황과 클라이언트의 상황을 인지하여 미디어의 품질 등급을 적절히 조절하여 완벽하게 재생시간을 보장한다. 따라서, 본 논문에서 제안한 NCAR는 Mobile IPTV와 같이 대역폭이 작고 네트워크 상황이 가변적인 환경에서도 좋은 서비스 제공이 가능할 것이다.

본 논문에서 제안한 NCAR 기법의 네트워크 인지형 혼잡제어는 멀티미디어 스트리밍의 공정성과 안정성을 향상시키고 높은 링크 활용도와 전송률의 변화를 감소시킨다. 그리고, NCAR의 클라이언트 인지형 흐름제어는 미디어 재생의 불연속성을 제거하고 안정된 버퍼할당과 낮은 재생지연시간 및 우수한 재생 보장 특성을 제공하였다.

5. 결론

기존 멀티미디어 스트리밍 시스템의 전송률 제어 기법은 네트워크 혼잡상황에서 네트워크의 안정성과 프로토콜간 공정성을 향상시키지만 미디어 재생의 연속성과 같은 특성을 고려하지 않아 양질의 QoE(Quality of Experience)를 제공하지 못하는 문제점을 가지고 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 인터넷 혼잡상황에서 멀티미디어 스트리밍 서비스의 QoE 향상을 위한 새로운 전송률 제어기법인 NCAR(Network and Client-Aware Rate control)을 제안하였다. 제안한 NCAR 기법은 멀티미디어 스트리밍의 공정성과 안정성을 향상시키고 높은 링크 활용도와 전송률의 변화를 감소시키고 흐름제어를 통해 미디어 재생의 연속성을 보장하고 안정된 버퍼할당과 낮은 재생지연시간을 제공하였다.

향후 연구과제로는 클라이언트 버퍼 수신 정보 이외에 배터리, 단말 화면크기, 사용자 선호도 등 다양한 클라이언트 상태 정보를 이용하여 QoE를 보장하는 전송률 제어기법 연구와 Mobile IPTV와 같은 이동성을 보장하는 네트워크 환경에서의 효율적인 스트리밍 서비스를 위한 전송률 제어기법에 대한 연구가 필요하다.

참고 문헌

- [1] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath live streaming via TCP: scheme, performance and benefits," *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, December 2007.
- [2] 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*, 1999.
- [3] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," *IEEE INFOCOM*, 2001.
- [4] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP emulation at receivers - flow control for multimedia streaming," *Technical Report, NCSU*, 2000.
- [5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," *ACM SIGCOMM*, 2000.
- [6] T. Kim and M. H. Ammar, "Optimal quality adaptation for MPEG-4 fine-grained scalable video," *IEEE INFOCOM*, 2003.
- [7] J. Yan, K. Katrinis, M. May, and B. Plattner, "Media- and tcp-friendly congestion control for scalable video streams," *Multimedia, IEEE Transactions on*, vol.8, no.2, pp.196-206, 2006.
- [8] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," *NOSSDAV*, 1999.
- [9] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM*, 1998.
- [10] J. Bolot and T. Turletti, "A rate control mechanism for packet video in the internet," *IEEE INFOCOM*, pp.1216-1223, 1994.
- [11] A. Ortega and M. Khansari, "Rate control for video coding over variable bit rate channels with applications to wireless transmission," *IEEE Image Processing*, 1995.
- [12] W. Tan and A. Zakhori, "Error resilient packet video for the internet," *IEEE Image Processing*, 1998.
- [13] J. Lee, T. Kim, and S. Ko, "Motion prediction based on temporal layering for layered video coding," *ITC-CSCC*, vol.1, pp.245-248, 1998.
- [14] S. McCanne, "Scalable compression and transmission of internet multicast video," *Ph.D. thesis, University of California Berkeley, UCB/CSD-96-928*, 1996.
- [15] S. McCanne and M. Vetterli, "Joint source/channel coding for multicast packet video," *IEEE Image Processing*, pp.776-785, 1995.
- [16] M. Vishwanath and P. Chou, "An efficient algorithm for hierarchical compression of video," *IEEE Image Processing*, pp.275-279, 1994.
- [17] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for unicast audio and video," *ACM SIGCOMM*, 1999.
- [18] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for Internet video streaming," *IEEE Journal on Selected Areas of Communications*, 2000.
- [19] N. Feamster, D. Bansal, and H. Balakrishnan, "On the interactions between layered quality adaptation and congestion control for streaming video," *Packet Video Workshop*, 2001.
- [20] N. Wakamiya, M. Miyabayashi, M. Murata, and H. Miyahara, "MPEG-4 video transfer with TCP-friendly rate control," *IFIP/IEEE MMNS*, 2001.

- [21] T. Kim and M. H. Ammar, "Optimal quality adaptation for scalable encoded video," *IEEE Journal on Selected Areas of Communications*, 2005.
- [22] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuits Syst. Video Techn.*, 2007.
- [23] The network simulator ns-2, <http://www.isi.edu/nanam/ns/>
- [24] Joint Video Team (JVT) of ISO/IEC MPEG, ITU-T VCEG, "Joint scalable video model JSVM-9," *JVT-V202*, January 2007.

구 자 현

정보과학회논문지 : 정보통신
제 36 권 제 5 호 참조

정 광 수

정보과학회논문지 : 정보통신
제 36 권 제 5 호 참조