

클라이언트/서버 기반 스트리밍 시스템에서의 네트워크 적응형 QoS 기법

정 연 일[†] · 이 정 찬[†] · 이 승 룡^{††}

요 약

유·무선 인터넷 및 하드웨어의 발전으로 인터넷 방송, VoD(Video On Demand) 등의 응용 서비스가 증가하면서 QoS(Quality of Service) 보장 문제가 해결해야 될 중요한 이슈로 부각되고 있다. 현재 인터넷은 모든 패킷을 우선순위에 상관없이 FIFO 형태로 전달하는 최선형(Best Effort) 서비스 기법에 근간하고 있기 때문에 보다 안정적인 서비스 지원을 위한 새로운 QoS 프로토콜 개발, QoS 지원 라우터 개발 등의 연구가 진행 중이다. 하지만 이러한 새로운 QoS 기법들은 기존 인프라를 충분히 활용하지 못하고 새로운 하드웨어 및 소프트웨어를 사용해야 하기 때문에 고비용, 비효율성의 문제를 수반한다. 이러한 점에 착안하여, 본 논문에서는 기존의 인프라를 최대한 이용하는 클라이언트/서버 기반의 네트워크 적응형 QoS 제어기법을 제안한다. 제안된 QoS 제어기법은 스트리밍 시스템에서 실시간으로 서비스되고 있는 네트워크 대역폭 상태를 파악하여 QoS를 지원하기 때문에 유·무선 네트워크에 상관없이 적용할 수 있을 뿐만 아니라, 기존의 스트리밍 서비스에서 사용되는 프로토콜들을 이용하기 때문에 저 비용으로 구현할 수 있다. 또한, 스트리밍 서버와 클라이언트에 새로운 모듈 추가만으로 QoS를 지원하기 때문에 라우터나 네트워크 회선의 교체 없이도 QoS를 보장할 수 있는 장점을 지니고 있다. 인터넷의 트래픽이 폭주하는 경우에서 모의실험을 수행한 결과 일반 인터넷 환경에서는 클라이언트의 버퍼 메모리가 비어 스트리밍 재생 시 끊김 현상이 자주 발생하였으나, 제안한 기법을 사용한 인터넷 환경에서는 동영상의 끊김 현상이 없이(seamless) 재생될 수 있는 스트림이 클라이언트 버퍼 메모리에 항상 있음을 확인할 수 있었다.

Network Adaptive Quality of Service Method in Client/Server-based Streaming Systems

Yonil Zhung[†] · Jung-chan Lee[†] · Sungyoung Lee^{††}

ABSTRACT

Due to the fast development of wire&wireless internet and computer hardware, more and more internet services are being developed, such as Internet broadcast, VoD (Video On Demand), etc. So QoS (Quality of Service) is essentially needed to guarantee the quality of these services. Traditional Internet is Best-Effort service in which all packets are transported in FIFO (First In First Out) style. However, FIFO is not suitable to guarantee the quality of some services, so more research in QoS router and QoS protocol are needed. Researched QoS router and protocol are high cost and inefficient because the existing infra is not used. To solve this problem, a new QoS control method, named Network Adaptive QoS, is introduced and applied to client/server-based streaming systems. Based on network bandwidth monitoring mechanism, network adaptive QoS control method can be used in wire&wireless networks to support QoS in real-time streaming system. In order to reduce application cost, the existing streaming service is used in NAQoS. A new module is integrated into the existing server and client. So the router and network line are not changed. By simulation in heavy traffic network conditions, we proved that stream cannot be seamless without network adaptive QoS method.

키워드 : QoS(Quality of Service), 대역폭(Bandwidth), 클라이언트/서버(Client/server), 스트리밍(Streaming)

1. 서 론

멀티미디어 운영체제, 네트워크, 응용서비스 기술과 이를 지원할 수 있는 고성능 하드웨어의 발전에 힘입어 위성 방송, 유·무선 인터넷 환경에서 스트리밍 서비스가 점차 일반화되어 가고 있다. 하지만 인터넷 서비스가 폭증하면서 서비스 품질(QoS) 보장에 대한 요구도 더욱 증대되고 있다. 스트리밍 서비스 관리자에게는 최적의 네트워크 운용과

관리를 위한 효율적인 대역폭 관리가 필요하게 되었으며, 사용자는 안정적이고 고품질의 스트리밍 서비스를 요구하게 되었다[1].

이러한 문제들을 해결하기 위해 인터넷상에서 QoS 지원에 관한 연구가 많이 진행되어 왔으나, 실제 서비스 현장에서는 회선의 대역폭을 증설하는 방법이 주로 사용되어 왔다. 그러나 이러한 회선 증설은 높은 비용 부담을 감당해야 하며, 지속적으로 증가하는 트래픽 문제를 해결하는 데는 한계가 있다. 그리고 현재 인터넷은 모든 패킷을 우선순위에 상관없이 FIFO 형태로 동일하게 전달하는 등, 최선형(Best Effort) 서비스만을 제공하고 있다. 때문에 다양한 멀

[†] 준 회원 : 경희대학교 대학원 컴퓨터공학과

^{††} 종신회원 : 경희대학교 컴퓨터공학과 교수

논문접수 : 2003년 7월 21일, 심사완료 : 2003년 12월 9일

미디어 데이터의 특징에 맞는 트래픽 제어는 근본적으로 보장하기에 어려운 실정이다[2]. 이러한 근본적인 QoS 보장을 위해서, 새로운 네트워크의 자원 관리 프로토콜 및 QoS 전용 라우터 개발 등의 연구가 진행 중이다. 하지만, QoS 보장하기 위한 차세대 인터넷 프로토콜에서의 QoS 지원 방법과 이를 보완하는 기존의 네트워크 적응형 QoS 기법에는 몇 가지 문제점을 내포하고 있다.

먼저 근본적인 QoS를 보장하기 위한 방식은 QoS 전용 라우터 등의 새로운 하드웨어나 소프트웨어로 추가 및 교체로 인한 고비용의 문제가 생긴다. 또한 네트워크 트래픽의 계속적인 증가로 네트워크 구조 변경, 네트워크 트래픽의 분산, 높은 우선순위의 트래픽 처리 등의 새로운 방식을 추가 사용하게 될 경우, 다시 새로운 프로토콜의 적용, 새로운 하드웨어로 교체 및 추가 등으로 인한 비효율성과 고비용의 문제가 생기게 된다. 따라서 새로운 인프라를 구축해야 하는 근본적인 QoS 제어뿐만 아니라 현재의 인프라를 이용해서 QoS를 구현하기 위한 방법으로 애플리케이션 레벨에서의 QoS 지원 방식도 필요하다. 이러한 접근은 근본적인 QoS 보장을 해줄 수는 없지만 비효율성과 고비용의 문제점을 해결 할 수 있다. 그리고 근본적인 QoS 보장을 위한 차세대 인터넷 프로토콜 위에서도 사용이 가능하므로 두 가지를 동시에 사용하면 보다 더 확실한 QoS 제어가 가능하게 된다.

이러한 애플리케이션 레벨에서 네트워크 적응형 QoS 방식은 먼저 네트워크 상태를 측정하여 네트워크의 상태에 따라서 혼잡 제어, 프레임 드랍, 트랜스코딩 등과 같이 전송되는 데이터의 양을 조절하거나 재 인코딩 하는 등의 연구로 진행되고 있다[19-21, 24]. 이러한 방식은 네트워크의 상태가 스트리밍을 서비스하기에 안 좋은 상태로 계속 이어질 경우, 동영상이 끊어지는 현상이 계속적으로 발생하거나, 좋은 품질의 서비스를 받기 위해 클라이언트에서 버퍼링 대기 시간이 많아지는 등의 문제점들이 발생 한다. 이런 문제점은 네트워크의 상태를 예측 하는 것이 불가능한 일이며, 계속적으로 네트워크의 상태를 측정하고 네트워크 상태 변화에 대한 적절한 반응이 부족하기 때문에 일어난다. 또한 애플리케이션에서 제공할 수 있는 다른 QoS 기법으로는, 클라이언트의 버퍼 메모리 크기를 최대한 크게 하여 재생이 되기 전 충분한 크기의 스트리밍을 전송하여 재생하는 방법이 있다. 이러한 방법은 자원이 풍부한 컴퓨터에서는 사용이 용이 하지만 버퍼 메모리의 크기에 제약을 갖는 휴대폰이나 PDA 같은 클라이언트에서는 활용하기 어렵다.

이러한 문제점들을 해결하기 위한 방법의 하나로 본 논문에서는 클라이언트/서버 기반 인터넷 스트리밍 시스템에서 기존의 인프라를 최대한 이용한 네트워크 적응형 QoS 제어 기법을 제안한다. 제안된 QoS 기법의 특징은 다음과 같다. 첫째, 기존의 방식은 QoS를 지원하기 위하여 네트워크 전체에 변화를 주거나 패킷 등에 변화를 주는 새로운 방식을 사용하여 고비용 QoS를 지원하고 있다. 제안된 방식은 기존의 스트리밍 서비스에서 사용하는 네트워크 파라미터들을 이용한 QoS를 지원한다. 현재의 스트리밍 서비스에서 사용하고

있는 RTCP(Real-time Transport Control Protocol), RTP(Real-time Transport Protocol) 등의 프로토콜 네트워크 파라미터 값을 이용하여 서비스를 하는 스트리밍 서버와 서비스를 받는 클라이언트 간의 네트워크 대역폭 상황을 측정하여 QoS를 적용하기 때문에 커다란 변화 없이 현재의 인프라를 이용하여 QoS를 제공할 수 있다.

둘째, 제안된 기법은 스트리밍 서버와 재생 클라이언트에 새로운 모듈만을 포함하여 QoS 제어를 하게 된다. 서버에는 실시간 QoS가 가능한 네트워크 적응형 스트리밍 전송 모듈과 클라이언트에는 대역폭 측정 및 네트워크 파라미터 값 분석 모듈을 포함하게 된다. 따라서 기존 QoS 기법처럼 네트워크 관련 하드웨어나 소프트웨어의 전체적인 교체 없이 기존 매카니즘에 새로운 모듈만 포함하여 QoS 구현이 가능하다.

셋째, 제안하는 QoS 기법은 실시간으로 네트워크의 변화를 측정하여 측정된 결과를 토대로 실시간으로 QoS를 적용하며 대역폭 상태를 예측하는 알고리즘을 사용하여 QoS를 지원하게 된다. 따라서 네트워크 사용량 증가에 따른 QoS 제공을 위한 추가 비용이 적으며 QoS를 지원하는 스트리밍 시스템의 유지 보수가 쉽다. 또한 현재의 네트워크 대역폭을 실시간으로 측정하여 QoS 제어를 하기 때문에 기존의 네트워크나, PAN(Personal Area Network), BAN(Body Area Network)과 같은 새로운 형태의 차세대 네트워크에도 사용될 수 있어 고 효율적인 QoS 지원을 할 수 있다.

또한 기존의 네트워크 적응형 QoS 기법에서 사용하는 네트워크 상태에 따라 전송되는 데이터의 양을 조절한다거나 재 인코딩 하는 등의 변화가 아닌, 스트리밍 서버에서 실시간 대역폭 변화에 적당한 비트율로 인코딩된 다른 스트리밍을 서비스를 하는 변화를 채택하여 서비스 되는 동영상의 품질은 다소 떨어지더라도 끊어지지 않는 동영상을 계속적으로 제공하여, 클라이언트에서 끊어지는 현상을 줄이고 유·무선 네트워크의 특징에 상관없이 실시간 동영상 재생을 위한 기법으로 사용할 수 있는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 QoS 관련 연구를 소개하고, 3장에서는 제안된 클라이언트/서버 기반 스트리밍 시스템에서의 네트워크 적응형 QoS 알고리즘에 대하여 기술한다. 4장에서는, 네트워크 적응형 QoS 알고리즘에 대한 성능 평가 및 효율 분석을 한 뒤, 5장에서 결론을 맺는다.

2. 관련 연구

현재의 인터넷은 통신하는 호스트들 간에 세그먼트를 전달하기 위해서 최선의 노력을 하지만 어떠한 보장(Guarantee)도 하지 않는다. 따라서 QoS를 지원하기 위해서 하나의 서비스로 취급되는 IP 패킷의 흐름을, 특성에 따라 여러 종류의 서비스로 구분하여 QoS를 지원하는 새로운 IP 패킷 전달 방식에 대한 연구가 최근 수년간 연구되어 왔다[3, 4]. QoS 방식은 기존의 최선형(Best effort) 서비스 상에서 QoS를 제공하는 방식과 차세대 인터넷에서 QoS 제공을 위하여 제안된 방식으로 나눌 수 있으며 그 중 최선형 서비

스 상에서 QoS를 제공 하는 방식으로는 지터(Jitter)를 감소시키는 방법과 패킷 손실 처리 방법 등이 있다. 차세대 인터넷에서 QoS 제공하기 위한 방식은 RSVP(Resource reSerVation Protocol), IntServ(Integrated Service), DiffServ(Differentiated Service) 모델 등이 있다.

지터(Jitter)를 감소시키는 방법으로는 한계 지연 시간을 일정 상수로 사용 하는 Fixed Play-out Delay 기법[13], 가변적인 변수로 사용하여 상황에 맞는 시간 상수 값을 설정하여 지터를 없애거나, 감소시키는 Adaptive Play-out Delay 기법이 있다[14].

패킷 손실 처리 방법으로는 패킷이 손실될 때, 수신자가 부가적인 정보를 포함한 다음 패킷의 정보를 가지기 때문에 결과적으로 패킷의 손실을 없는 것으로 만들어 패킷 손실을 감추는 FEC(Forward-Error Correction) 기법[15], 패킷 손실의 영향을 줄이며, 오류를 사용자가 느끼지 못 할 정도로 최소한의 크기로 자료를 나누어 패킷의 전체 손실에 의한 데이터 스트림의 품질을 유지하는 Interleaving 기법[16], 기본적인 생각은 손실된 패킷의 원본 데이터와 비슷한 데이터로 대체하는 패킷을 만들어 내거나, 이전의 데이터와 비슷한 데이터를 반복하는 Receiver-based recovery 기법들이 있다[17].

차세대 인터넷에서 QoS 제공을 위하여 제안된 RSVP는 서비스 품질에 대한 요청이 데이터 흐름의 모든 경로에 따라서 이루어질 수 있도록 라우팅 데이터베이스를 참조하여 데이터 경로상의 각 노드에서 자원이 예약되도록 한다. Int Serv 모델은 실시간 응용 서비스에서 발생하는 패킷의 흐름을 단위로 하여 패킷을 전달한다. 즉 RSVP 프로토콜을 이용하여 연결 수락 제어와 자원예약을 수행하여 패킷의 전달 지연을 보장해 주는 모델이다. 보장된 서비스를 위해서는 IntServ를 제공하는 라우터가 망 전체에 설치되어야 한다. 이러한 IntServ 모델의 문제점으로 인하여 백본망에서는 현실적으로 수용하기에는 어려움이 있다[6]. IntServ 모델의 문제점을 극복하기 위해 제안된 DiffServ 모델은 IPv4 헤더에 TOS(Type of Service) 필드가 정의되어 있고, 작은 지연, 높은 수율, 낮은 손실율 등을 나타내기 위해 TOS 필드를 사용한다[7]. 기존 라우터에서는 이와 같은 응용의 요구를 거의 무시하고 모든 패킷을 동일하게 처리하여 근본적인 QoS 제어가 불가능 하였다. DiffServ가 사용되기 위해서는 에지(Edge) 라우터 및 코어 라우터가 모두 이를 지원하여야 하는데, 현재는 확실한 수익모델이 없어서 이러한 인프라에 대한 투자가 어렵기 때문에 실제 사용되기는 어렵다[5, 11, 18].

지터를 줄이거나 손실된 패킷을 처리하는 기법들은 네트워크에서 트래픽이 증가할 때 효과가 미비하고 사용자가 만족할 만한 QoS를 지원하기 어렵다. 또한 차세대 인터넷에서 QoS 제공을 위하여 제안된 RSVP, IntServ, DiffServ 등의 방식은 QoS를 위한 근본적인 해결방안은 될 수 있으나 실제 구현하기에는 고비용, 저효율의 문제점이 발생한다.

이러한 문제점들을 해결하기 위한 네트워크 적응형 QoS

제어 기법에 관한 연구는 다음과 같다. 클라이언트와 서버 간의 RTP 기반 스트리밍 서비스를 하면서 RTCP의 정보 중 패킷 손실에 기반 해서 클라이언트의 형태를 unloaded, hold, loaded 세 가지 형태로 구분하여 서버의 발송률(sending rate)을 조절하는 방식[19], CBR(Constant Bit Rate)가 아닌 VBR(Variable Bit Rate)환경의 VOD 시스템에서 VBR의 버스트한(Burst) 특성, 즉 특정 프레임이 다른 프레임보다 프레임 당 비트수가 지나치게 많은 경우가 발생하여 데이터 전송시에는 QoS를 보장하기 어려우므로 전송률(Transmission Rate)을 일정하게 유지 할 수 있도록 평활화 해주는 방식[20], 다양한 클라이언트 단말기에 따라 적절한 대역폭의 비디오 스트림이 서비스가 가능하도록 비트율 변환뿐만 아니라, 입력영상의 해상도 및 프레임을 변환을 적응적으로 수행하는 방식[21], 멀티미디어 데이터는 인코딩이 되면 비율이 고정되기 때문에 여러 네트워크가 묶여 있는 인터넷 환경에서 효율적으로 활용하기 위해서, 핵심이 되는 몇 군데 지역에 에지(edge) 서버를 두어서 접속하는 클라이언트와 가장 짧고 패킷 소실이 적은 서버로 접속하게 하고 서비스를 받게 하는 방식[22], 네트워크의 트래픽에 따라 전송하는 패킷을 조절하는 방식[23], 네트워크 환경에 적응하여 대역폭을 평활화 하는 방식 [24], 전송되는 패킷 전송 간격, 데이터그램 크기 등의 변화를 주는 방법으로 QoS를 제어하는 방법 등의 연구들이 진행 중에 있다.

이러한 연구들은 QoS 제어를 위해 서버의 부하가 많이 생성 되거나, 일부 스트리밍 데이터 특징이나 환경에 맞는 QoS 기법을 정의하여 해당 환경에서만 QoS 효과를 얻을 수 있으며 네트워크의 상태를 실시간으로 측정하여 상태를 반영하지 못하기 때문에 실시간 동영상의 재생 처리가 어렵고 네트워크의 상태가 계속해서 낮은 대역폭을 유지 할 경우 발생하는 클라이언트의 낮은 재생률에 대한 문제점들을 해결 할 수 없다. 하지만 제안하는 네트워크 적응형 QoS의 경우 애플리케이션 레벨에서 지원하는 QoS로서, 실시간 대역폭 변화를 반영하여 동영상의 품질은 떨어지더라도 동영상의 끊김 현상은 줄어들며, 기존의 클라이언트/서버 기반의 스트리밍 시스템에 사용되는 프로토콜을 사용하여 서버와 클라이언트에 적은 부하만을 생성시키고, 네트워크의 대역폭을 이용한 매카니즘을 이용하기 때문에 Diff Serv, RSVP, IntServ 모델과 같이 사용 될 수 있다.

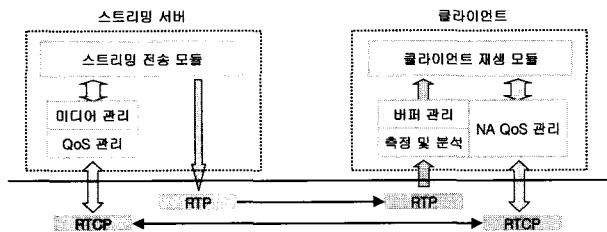
3. 클라이언트/서버 기반 네트워크 적응형 QoS 기법

3.1 클라이언트/서버 기반에서 네트워크 적응형 QoS 모델

현재의 스트리밍 서비스나 유·무선 인터넷 방송의 경우 각각 낮은 레벨부터 높은 레벨까지 인코딩 된 여러 단계의 스트리밍 소스를 제공하고 있다. 사용자는 스트리밍 서비스를 시작하려는 처음에 선택한 한 가지 레벨의 스트리밍으로 서비스를 받게 된다. 이럴 경우 QoS를 지원하는 시스템의 경우라도 네트워크의 대역폭 상태를 고려하지 못한 QoS는 사용자가 만족할 만한 QoS를 지원하기는 쉽지가 않

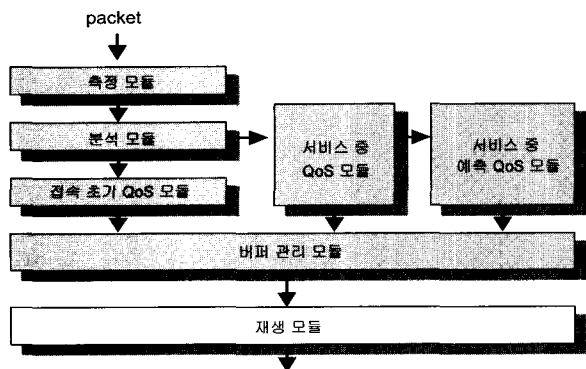
다. 서비스를 제공하는 서버 측에서 제공할 수 있는 QoS는 단순히 네트워크의 불안정한 대역폭 상태를 고려하여 버퍼의 크기를 크게 하여 사용자가 이용하고 있는 스트리밍 서비스를 끊어지지 않게 제공하는 것뿐이다.

제안한 클라이언트/서버 기반 스트리밍 시스템에서 네트워크 적응형 QoS(Network Adaptive QoS : NAQoS) 기법의 경우 스트리밍 서버와 서비스를 받는 클라이언트 간의 네트워크 대역폭 상태를 측정하여 그 측정된 값을 통하여 적합한 QoS를 하게 된다(그림 1). 측정하는 데이터도 새로운 프로토콜을 이용하는 것이 아니라 서버에서 클라이언트로 보내게 되는 RTP, RTCP 패킷을 이용하여 측정하게 된다. 클라이언트는 서버에서 받는 패킷들을 이용하여 네트워크 대역폭 및 QoS 파라미터 정보를 서버로 다시 보내게 된다. 서버에서는 클라이언트에서 보낸 데이터를 이용하여 서비스 되고 있는 스트리밍 인코딩 레벨 중 적합한 레벨로 서비스를 다시 하게 된다. 스트리밍 서버에서는 얼마나 많은 클라이언트가 연결이 되어있는지 네트워크 망이 유선인지 무선인지에 대한 내역은 상관없이 모든 네트워크에서 적용할 수가 있다. 결론적으로는 클라이언트에서는 기존의 프로토콜을 이용한 네트워크 대역폭 및 QoS 파라미터 측정 및 비교 모듈을 포함 하고 스트리밍 서버에서는 클라이언트 요구에 맞는 레벨의 스트리밍 소스의 전송 모듈 등 최소한의 모듈만 포함함으로써 QoS를 제공할 수 있다.



(그림 1) 스트리밍 클라이언트/서버에 포함되는 NAQoS 모델

MPEG-1의 동영상은 일정 블록 단위로 인코딩 되어 있어 헤더가 포함된 일정 부분만을 재생할 수 있다. 따라서 비트율이 다른 하나의 미디어라 하더라도 재생 클라이언트에서 재생 비트율을 자동으로 변경해 주면 재생이 가능하다.



(그림 2) 클라이언트에 포함되는 QoS 알고리즘 모듈

다. 본 논문에서 클라이언트에서 네트워크 대역폭을 측정하며 서버에 측정 결과를 전송하고 클라이언트의 버퍼 메모리 안에 서버에서 보내는 해당 비트율의 스트리밍의 양을 측정하는 것으로 동영상 재생을 대신 하였다.

(그림 2)는 클라이언트에 포함되는 QoS 알고리즘의 전체 모듈이다. 패킷이 네트워크를 통해 클라이언트로 들어오면 우선 측정 모듈에서 QoS 파라미터 값들을 측정하고 분석하게 된다. 분석이 끝난 뒤 서비스 접속 초기, 서비스 중에 따라서 각기 QoS 모듈로 데이터를 전달하게 되며 서비스 중 일정 시간동안 현 레벨의 스트리밍 서비스가 유지 될 경우 예측 QoS 모듈로 데이터를 전달하게 된다.

3.2 접속 초기 NAQoS 알고리즘

접속 초기 NAQoS 알고리즘은 클라이언트가 스트리밍 서비스를 하는 서버에 접속하여 클라이언트 스트리밍 플러이어가 스트리밍을 재생하기 전까지의 버퍼링을 하는 동안의 네트워크 적응형 QoS 알고리즘을 말한다(그림 3).

```

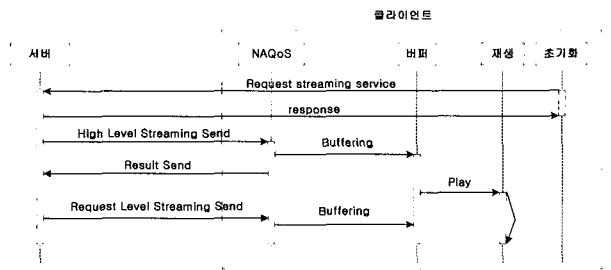
01 : Request () ;
02 : Send_Option () ;
03 : Send_DESCRIBE () ;
04 : Send_Setup () ;
05 : buffer = Receiver_Buffer_Initial () ;
06 : Send_Play () ;
07 : Member_variable_Initial () ;
08 : timer = Timer () ;
09 : while (end to streaming) {
10 :   receive_packet = Receive_RTP () ;
11 :   totalbandwidth = totalbandwidth + receive_packet ;
12 :   buffer = buffer + receive_packet ;
13 :   if (timer == t) {
14 :     currentbandwidth = totalbandwidth / t ; } }
    
```

(그림 3) 접속 초기 네트워크 적응형 QoS 알고리즘

(그림 3)에서 스트리밍 서비스를 받기위해 해당 서버에 연결요청을 한다(라인 1). RTSP 옵션을 설정 한다(라인 2). 서비스 받을 미디어에 대한 파일 속성을 요청한다(라인 3). RTP와 RTCP 포트 설정하고 서버에 전송한다(라인 4). Setup 명령이 성공하고 나면 미디어 재생을 위해 필요한 버퍼를 미리 초기화 한다(라인 5). 서버에 재생 명령을 내려서 스트리밍 서비스를 시작 한다(라인 6). 클라이언트의 각종 변수들을 초기화 한다(라인 7). 일정 시간 마다 대역폭 측정을 위한 타이머를 호출 한다(라인 8). 스트리밍 서비스가 끝날 때 까지 Receive_RTP() 함수로부터 받은 패킷 데이터를 receive_packet에 저장 한다(라인 10). 현재까지 받은 receive_packet을 합한다(라인 11). receive_packet은 재생을 위한 buffer로 보낸다(라인 12). 일정시간 t가 되던 현재까지의 전체 데이터를 시간으로 나눠서 현재의 대역폭을 계산 한다(라인 14).

(그림 4)는 접속 초기 네트워크 적응형 QoS 알고리즘의 시퀀스 다이어그램이다. 클라이언트는 서비스를 제공하는 서버에 접속하여 원하는 스트리밍 서비스에 대한 요청 신호(Request Streaming Service)를 보내게 된다. 스트리밍 서버

는 현재 제공하고 있는 스트리밍 서비스 중 가장 높은 레벨로 인코딩 된 스트리밍을 요청한 클라이언트로 보내게 된다. 클라이언트에서는 서버에서 들어오는 RTP 데이터를 버퍼로 보내기 전에 NAQoS 모듈에서 네트워크의 대역폭 및 QoS 파라미터 값을 실시간으로 계산 한다. 또한 클라이언트에서 스트리밍을 재생하기 전까지 버퍼링 시간 동안의 측정된 QoS값 결과를 RTCP 패킷을 이용하여 스트리밍 서버로 보낸다. 스트리밍 서버는 클라이언트에서 보낸 결과 정보를 이용하여 초기 버퍼링 후 보낼 스트리밍 소스의 레벨을 결정해서 적합한 스트리밍 데이터를 전송 한다. 클라이언트에서는 초기 버퍼링이 끝나면서 재생을 하게 되며 계속 해서 버퍼에 스트리밍 서버에서 보낸 스트리밍 소스를 저장하게 된다.



(그림 4) 접속 초기 네트워크 적응형 QoS

클라이언트가 서버에 접속하는 초기에는 기존의 RTP, RTSP, RTCP 패킷의 정보만 이용하여 대역폭 및 QoS 파라미터 값을 계산 하게 되며, 클라이언트에서 서버에 결과를 보낼 경우에도 RTCP 패킷에 같이 정보를 보낸다. 따라서 클라이언트에 스트리밍 서버에서 보내는 패킷을 처리하는 모듈만 포함하며 서버에서는 클라이언트의 요청에 대한 QoS 분석 모듈과 다른 레벨로 인코딩 된 스트리밍 소스를 전송해주는 모듈만 포함 하면 된다.

3.3 스트리밍 서비스 중 NAQoS 알고리즘

스트리밍 서비스가 시작이 되고 클라이언트에서 스트리밍이 재생이 되면 서버에서는 초기 버퍼링 동안의 평균 대역폭 정보를 이용하여 적합한 스트리밍 서비스를 하게 된다. 초기 버퍼링이 끝나고 스트리밍 서비스 중의 네트워크 적응형 QoS 알고리즘은 (그림 5)와 같다.

```

01 : while (end to streaming) {
02 :   receive_packet = Receive_RTP ();
03 :   totalbandwidth = totalbandwidth + receive_packet ;
04 :   buffer = buffer + receive packet ;
05 :   delay = Delay_Calculation ();
06 :   jitter = Jitter_Calculation ();
07 :   current_bandwidth = Current_Bandwidth ();
08 :   if (timer == t){
09 :     average_bandwidth = Average_Bandwidth ();
10 :     if (average_bandwidth < SERVICE_Bandwidth){
11 :       RR_set (jitter, delay, receive_time) ;
12 :       RR_extention_set (bandwidth) ; } }

```

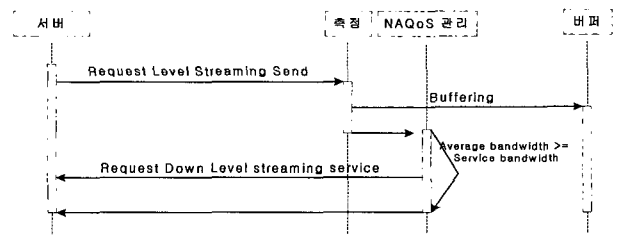
(그림 5) 스트리밍 서비스 중 네트워크 적응형 QoS 알고리즘

그림에서와 같이 RTP 패킷을 받아서 receive_packet에 저장하고 전체 대역폭을 계산한다(라인 2-3). 패킷을 버퍼에 저장한다(라인 4). 딜레이, 지터, 현재 네트워크 대역폭을 계산한다(라인 5-7). 버퍼 크기를 참고한 시간 t를 설정하여 평균 대역폭을 t초 마다 구한다(라인 8-9). 평균 대역폭이 서비스 허용 대역폭 보다 떨어질 경우 RTCP 패킷 확장부에 해당 데이터를 전송하여 다음 하위 단계의 스트리밍 서비스를 요구한다(라인 10-12).

클라이언트에서는 스트리밍 서비스를 받는 동안 계속해서 RTP, RTCP 데이터를 이용하여 대역폭, 평균 대역폭을 측정하게 된다. 다음 레벨의 스트리밍 서비스를 선택하는 기준은 QoS 파라미터 값들 중 평균 대역폭을 이용한다. 네트워크의 대역폭 상태는 불안정 하고 순간적으로 대역폭이 떨어지는 경우가 많다. 따라서 서비스되고 있는 대역폭 보다 대역폭이 순간적으로 떨어지더라도 이를 바로 스트리밍 서비스에 반영하는 것은 스트리밍 서버에 많은 부하를 일으키게 된다. 그러므로 순간적으로 떨어지는 대역폭은 의미가 없기 때문에 평균 대역폭 값을 이용하여 해당 레벨의 스트리밍 서비스를 결정한다.

평균 대역폭이 서비스되는 허용 대역폭 밑으로 떨어질 경우는 네트워크의 대역폭 상태가 해당 레벨의 스트리밍 서비스를 하기에는 적합하지 않기 때문에 다음 낮은 레벨의 스트리밍으로 서비스를 해야 한다. 여기서 평균 대역폭은 네트워크의 대역폭 상태를 클라이언트가 측정하여 결과를 서버에 전송한 뒤 낮아진 대역폭에 맞는 해당 레벨의 서비스를 다시 받기 위해서는 버퍼 크기의 60%에 해당되는 시간으로 지정하여 측정한다. 너무 짧은 시간 동안의 평균 대역폭은 현재 네트워크 대역폭 상태를 반영한 값으로는 보기 힘들기 때문에 최소 60%에 해당되는 시간을 이용해야 클라이언트에서 스트리밍을 재생 시 끊어지는 현상이 일어나지 않는다.

비디오 형태의 스트리밍 데이터는 일반적인 파일 전송과는 달리 신뢰성이 100% 보장될 필요는 없다[12]. 네트워크의 대역폭은 서비스 되는 레벨의 비트 당 전송률(bps) 보다 보통 20% 정도 낮은 대역폭 상태가 될 경우 클라이언트에서 재생되는 화면이 끊어지는 현상이 일어난다. 하지만 보다 높은 QoS 서비스를 하고자 할 경우는 클라이언트의 버퍼와 연관이 있지만 10% 이하 일 경우 다음 레벨의 대역폭 서비스로 이동하는 것이 좋다.



(그림 6) 서비스 중 네트워크 적응형 QoS

(그림 6)에서 평균 대역폭이 서비스 허용 대역폭 밑으로

떨어질 경우 다음 레벨의 스트리밍 서비스를 요구하게 되는데 이 요구는 스트리밍 서버에서 제공되는 레벨의 최하위 레벨의 스트리밍 서비스가 될 때 까지 대역폭 및 QoS 파라미터 값을 측정하여 스트리밍 서비스를 반영한다. 또한 최하위 레벨의 스트리밍 서비스보다 더 대역폭이 떨어질 경우는 스트리밍 서버에서 지정한 방법으로 서비스를 하게 된다. 지정한 서비스는 관리자가 결정을 하게 되는데 동영상의 경우 음성만을 서비스 할 것인지 아니면 서비스를 중단하고 사용자에게 이 상황을 통보 할 것인지에 대한 결정을 해서 서비스를 한다.

3.4 스트리밍 서비스 중 네트워크 예측 NAQoS 알고리즘

클라이언트에서 스트리밍이 재생되면서 측정 되는 대역폭이 현재 서비스 레벨 보다 떨어지는 경우 측정값을 바로 서버에 알려서 적합한 스트리밍 서비스를 받을 수 있다. 하지만 대역폭이 떨어진 상태에서 대역폭이 다음 단계로 떨어지는 네트워크 상황은 쉽게 적용할 수 있는 반면, 네트워크의 대역폭 상태가 좋아지는 상황을 정확히 측정하기는 불가능 하다. 따라서 스트리밍 서비스 중 떨어지는 대역폭은 3.2절에서 설명한 방식으로 계속하여 정확한 대역폭 상태를 서버에 전달하는 알고리즘을 이용해서 QoS를 할 수 있으며, 현재 상태를 유지하고 있는 경우라면 새로운 알고리즘을 적용해야 한다.

```

01 : while (end to streaming) {
02 :   receive_packet = Receive_RTP ();
03 :   totalbandwidth = totalbandwidth + receive_packet ;
04 :   buffer = buffer + receive packet ;
05 :   delay = Delay_Calculation ();
06 :   jitter = Jitter_Calculation ();
07 :   current_bandwidth = Current_Bandwidth ();
08 :   if (current_bandwidth == setting_t ) {
09 :     Estimate_Bandwidth();
10 :     if (current_bandwidth > SERVICE_Bandwidth){
11 :       RR_set (jitter, delay, receive_time)
12 :       RR_extention_set (bandwidth) } }

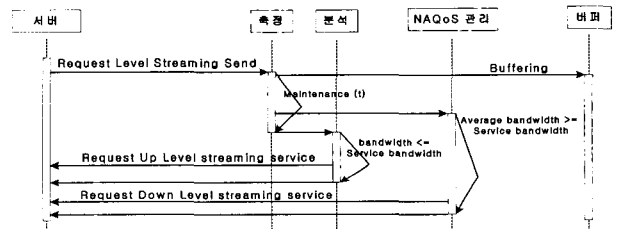
```

(그림 7) 스트리밍 서비스 중 네트워크 예측 NAQoS 알고리즘

측정되는 실시간 대역폭이 현 상태를 유지 하게 되는 경우는 두 가지 상태로 유추 할 수 있다. 첫 번째 상태는 대역폭이 현 상태를 계속 유지하고 있는 경우이고, 두 번째는 스트리밍 서버가 낮은 레벨의 QoS를 제공하고 있기 때문에 네트워크의 대역폭 상태가 좋아졌지만 측정하지 못하는 경우이다. 네트워크의 대역폭 상태가 상위 레벨 비트율 스트리밍 서비스를 할 만큼 좋아졌다면 다시 높은 단계 QoS를 제공해야 하지만 낮은 대역폭 상태의 QoS 상황에서 기존의 RTP, RTCP등의 패킷을 이용하는 네트워크 적응형 QoS는 더 이상 네트워크의 정확한 대역폭을 알 수가 없게 된다. 따라서 측정된 실시간 대역폭이 계속 현 상태를 유지할 경우 일정 시간(t) 마다 실제 네트워크의 대역폭 상태를 점검 할 필요가 있다. 여기서 일정 시간(t) 결정이 서버나 클라이언트의 성능에 커다란 영향을 끼치게 된다.

(그림 7)은 스트리밍 서비스 중 네트워크 예측 NAQoS 알고리즘이다. RTP 패킷을 받아서 receive_packet에 저장하고 전체 대역폭을 계산한다(라인 2-3). 패킷을 버퍼에 저장한다(라인 4). 딜레이, 지터, 현재 네트워크 대역폭을 계산한다(라인 5-7). 미리 지정한 시간 t 동안 현재의 대역폭이 유지가 될 경우 예측 함수를 호출한다(라인 8-9). 네트워크 대역폭이 서비스 허용 대역폭 보다 높을 경우 RTCP 패킷 확장부에 해당 데이터를 전송하여 다음 상위 단계의 스트리밍 서비스를 요구한다(라인 10-12).

(그림 8)은 스트리밍 서비스 중 네트워크의 대역폭 상태를 예측하기 위한 네트워크 적응형 QoS 알고리즘이다. 그림에서와 같이 측정 모듈에서 대역폭이 미리 정해진 일정 시간(t) 마다 분석 모듈에서 실제 대역폭을 측정하여 현재의 네트워크 대역폭 상태를 예측한다. 일정 시간 마다 실제 네트워크의 대역폭 상태를 점검하는 방식은 네트워크의 대역폭에 맞는 레벨의 스트리밍을 서비스 하는 도중 상위 단계 레벨의 스트리밍을 추가 하여 보내는 방식과 스트리밍과 관련 없는 임의의 데이터를 서버에서 보내는 방식을 사용한다. 첫 번째 방식의 경우 서버에서 다른 레벨의 스트리밍을 일정 시간마다 번갈아 가며 보내야 하기 때문에 서버에 부하가 많이 걸리는 편이고 두 번째 방식은 QoS를 하면서 불필요한 데이터를 네트워크를 통해 전달함으로써 서비스가 더 나쁜 상황으로 될 수도 있다는 단점이 있다.



(그림 8) 스트리밍 서비스 중 네트워크 예측 NAQoS

본 논문에서는 세 가지 사항을 고려해서 서버에서 관리자가 실제 대역폭을 확인 할 일정 시간을 결정하여 알고리즘에 포함 하도록 하였다. 세 가지 고려 사항은 첫째, 네트워크의 특징, 둘째, 서버의 상태, 셋째, 스트리밍 소스의 종류이다. 네트워크의 특징은 유·무선 네트워크의 여부와 네트워크의 트래픽이 많이 발생하는 시간과 적게 발생하는 시간의 차이 등을 말한다. 유·무선과 네트워크의 상태에 따라서 시간(t)을 변화 시켜야 서버에 많은 부하가 걸리지 않게 된다. 서버의 상태란 최대 서비스가 가능한 사용자의 수에 대한 실제 접속하여 서비스를 받고 있는 사용자의 수를 지칭한다. 스트리밍 소스의 종류는 크게 스트리밍 소스의 재생 시간과 사용자의 서비스 되는 빈도에 따른 분류를 말한다. 이 세 가지 고려 사항을 결정할 이유는 시간(t)을 한가지로 결정하게 되면 앞서 말한 상황에 따라 스트리밍 서버의 성능 차이가 많이 나며 또한 스트리밍 서버의 부하 차이가 생기기 때문이다. 본 논문에서 시간(t)은 <표 1>과 같이 세 가지 형태로 나누었다. 세 가지 형태의 일정

시간은 한 가지만 사용하는 것이 아니라 세 가지(A, B, C 형태)를 동시에 사용하게 된다.

<표 1> 네트워크 상태 예측을 위한 일정 시간(t)과 장단점

| 시간(t) 간격 | 특징 | 단점 |
|---------------------------|--|-----------------|
| • 10초(A 형태) | • 빠른 네트워크 상태 반영 • 스트리밍의 크기가 작은 경우 | • 서버의 부하가 큼 |
| • 10초×3회 + 5분 + 1분 (B 형태) | • 스트리밍의 크기가 큰 경우 • 빠른 네트워크 상태 반영 가능 | • 서버의 처리변수 증가 |
| • 1분 (C 형태) | • 서버의 부하 최소화 | • 느린 네트워크 상태 반영 |

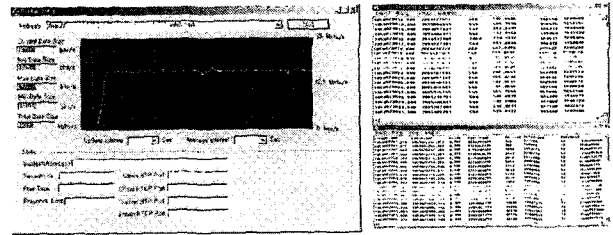
<표 1>에서 결정한 시간 t는 네트워크의 특징을 분석하여 결정된 값이다. 실제 네트워크의 패킷을 분석하여 정리한 결과 네트워크 상태의 특징은 다음 세 가지와 유사한 경향을 나타내었다. 첫째, 일정한 네트워크의 대역폭을 유지하다가 순간적으로 트래픽의 양이 늘어나면서 1초 내외의 네트워크 대역폭이 떨어졌다가 다시 원 상태로 유지되는 경우. 둘째, 트래픽의 양이 증가하면서 네트워크의 대역폭이 현저히 떨어졌고 그 상태가 약 3분에서 최대 7분 정도로 네트워크의 상태가 불안정하다가 처음 서비스 상태로 유지되는 경우. 셋째, 네트워크의 대역폭이 서서히 떨어지고 그 상태가 장 시간동안 유지되었다 원 상태로 복구되는 경우로 나타났다. 첫 번째와 같은 패턴이 가장 빈번하게 일어났으며 이러한 패턴을 가진 네트워크의 상태와 서비스하고자 하는 스트리밍의 종류 및 크기에 따라서 현재의 대역폭을 예측하기 위한 일정 시간(t)이 결정하여 스트리밍 서버 관리자가 결정 하면 된다.

4. 네트워크 적응형 QoS의 성능 평가

스트리밍 서버에서 보낸 패킷을 클라이언트에서 재생하기 전 포함되는 네트워크 적응형 QoS 관리 모듈은 앞 절의 (그림 1)과 같이 재생 클라이언트가 스트리밍을 재생하기 전 대역폭 분석 하며 이를 통해 NAQoS 모듈 부분에서 QoS 제어를 하게 된다. RTP 패킷은 서버에서 클라이언트로 전송이 되면 대역폭 측정 및 분석 모듈에서 QoS 파라미터 데이터를 분석하게 되며 버퍼 관리자에 의하여 스트리밍 패킷들이 관리 된다. 버퍼관리 및 대역폭 측정 및 분석 모듈은 QoS 제어 모듈에 의해 관리 된다. 또한 QoS 제어 모듈은 서버로 보내게 될 RTCP 패킷에 QoS 레벨을 요구 하는 신호를 포함하여 서버로 전송하는 역할을 한다. 제안한 네트워크 적응형 QoS 관리 모듈은 RTP, RTCP 등의 기존의 패킷들이 재생되기 위해 스트리밍 클라이언트의 공통 제어 모듈로 전송되기 전에 포함 하면 된다.

제안한 클라이언트/서버 기반 스트리밍 시스템에서 네트워크 적응형 QoS를 지원하기 위해서는 먼저 스트리밍 서버와 클라이언트간의 네트워크에 대한 측정이 필요하다. 여

기서 측정되는 QoS 파라미터 값은 실시간 대역폭, 평균 대역폭, 타임스탬프, 일련번호, 딜레이(delay), 지터(jitter) 등이 있다. (그림 9)는 서버와 클라이언트간의 대역폭 및 QoS 파라미터 데이터를 측정하고 저장하기 위해 제작된 툴이다. 측정은 스트리밍 서버에서 보내지는 RTP, RTSP, RTCP 패킷을 이용하여 클라이언트에서 측정하였다.

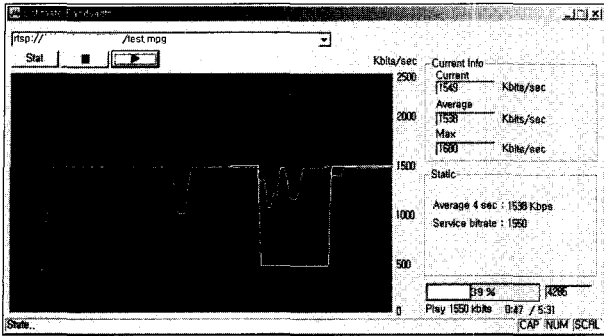


(그림 9) 네트워크 대역폭 및 QoS 파라미터 값 추출 툴 및 데이터

측정은 MPEG-1 형식의 데이터로 테스트 하였다. 용량이 큰 MPEG-1 데이터는 스트리밍 될 때 패킷화 된다. 각각의 패킷 헤더를 가지고 있고 헤더 또한 스트리밍의 대역폭을 사용하므로 대역폭을 측정의 대상은 패킷단위의 모든 헤더를 포함한 데이터로 하였다. RTP와 RTCP 프로토콜을 위해서 할당된 두 개의 포트로 들어오는 모든 스트리밍 데이터는 쓰레드를 통해 각각 분석모듈로 들어가고 대역폭을 분석 후 헤더정보를 분석 하였다. 측정된 데이터는 그래프 모듈로 값을 전송하고 현재의 대역폭과 버퍼로 보내질 실제 미디어의 데이터를 수치와 그래프로 표현하도록 하였다. 헤더의 분석을 마치고 남은 미디어 데이터는 재생을 위해서 버퍼를 채운다. 버퍼의 크기는 테스트를 위해서 FIFO형 버퍼를 5초 동안의 고정된 크기로 설정하고 버퍼가 100%되면 동시에 재생을 위한 대역폭만큼 버퍼에서 삭제하도록 필요한 부분만 제작하였다. 실제 서비스 시에서는 좀더 나은 버퍼 관리 알고리즘을 써야하지만 대역폭을 보기 위한 테스트이며 비트가 다른 동영상을 재생하는 클라이언트는 연구 중임으로 동영상 재생은 하지 않았다. 클라이언트/서버 기반 스트리밍 시스템에서 네트워크 적응형 QoS 성능을 평가하기 위한 평가 환경과 평가 항목은 <표 2>와 같다.

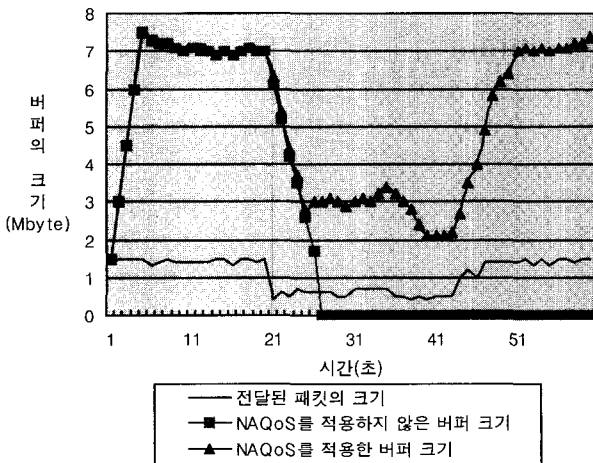
<표 5> NAQoS 알고리즘 성능 평가를 위한 평가 환경 및 평가 항목

| 항목 | 서버 환경 | 클라이언트 환경 |
|----------|--|---|
| 운영체제 | • Windows 2000 서버 | • Windows 98 |
| 하드웨어 환경 | • Pentium III 500 Mhz • RAM 512Mbyte • 10/100 Mbps NIC | • Pentium II 300Mhz • RAM 256Mbyte • 10Mbps NIC |
| 소프트웨어 환경 | • Morgan Streaming Server 1.0 • 1.5 Mbps MPG (MPEG-1), | • NAQoS 측정 툴 Ver 2.0 • 버퍼 크기 2Mb |
| 평가 항목 | • 퍼버 메모리의 상태 분석 • 네트워크 대역폭에 따른 서비스의 네트워크 반영 여부 | |



(그림 10) 네트워크 적응형 QoS 알고리즘이 포함된 QoS 측정 툴

평가 항목은 크게 두 가지로 나누어 설정하였다. 첫째는 실제로 서버에서 보낸 다른 비트율을 가진 하나의 스트리밍을 연결하여 재생 할 수 있는 클라이언트 개발은 진행 중이기 때문에, 트래픽이 갑자기 증가하여 현재 서비스 되고 있는 비트율보다 대역폭이 떨어지는 환경에서 NAQoS 알고리즘을 적용한 버퍼메모리와 적용하지 않은 시스템에서의 버퍼 메모리를 비교하여 클라이언트에서 재생할 동영상 대신 하였다. 둘째로 네트워크 대역폭을 반영한 스트리밍 서비스가 되고 있는 지 확인을 하기 위하여 현재의 대역폭과 서버에서 보내는 클라이언트의 비트율의 비교하여 현재의 네트워크 대역폭 상태를 잘 반영에 대하여 성능 평가를 하였다. (그림 10)은 <표 2>에서의 평가 환경에서 네트워크 적응형 QoS 알고리즘이 포함되어 실시간 대역폭, 평균 대역폭, 스트리밍 서버의 서비스 대역폭, 클라이언트의 버퍼 메모리의 상태 등을 측정하기 위해 제작된 툴이다.

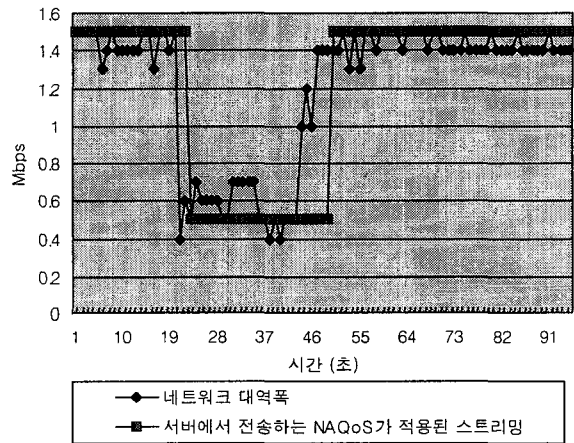


(그림 11) NAQoS를 적용하지 않은 버퍼와 적용한 버퍼 메모리의 비교

(그림 11)에서는 버퍼 메모리에 대한 측정은 실제 네트워크 대역폭과 NAQoS 알고리즘이 포함 안 된 클라이언트의 버퍼 메모리 크기와 NAQoS 알고리즘이 포함된 클라이언트에서의 버퍼 메모리의 크기를 비교하였다. 초기 5초간을 재생을 위한 초기 버퍼링 시간이며 그 후 클라이언트에서 스트리밍 재생을 시작한다. NAQoS를 적용하지 않은 버퍼

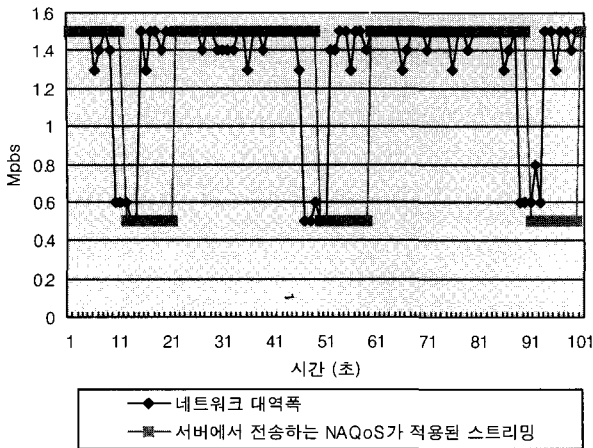
메모리의 경우 실제 서비스 대역폭 보다 아래로 떨어질 경우 버퍼 메모리의 스트리밍을 모두 재생 한 뒤에는 스트리밍을 재생하지 못하였고 다시 버퍼링을 시도하는 클라이언트 하더라도 대역폭이 낮은 상태라서 다시 초기 5초간 버퍼링을 하더라도 재생이 원활하지 않고 결국 다시 스트리밍을 재생하지 못했다. 하지만 NAQoS를 적용하였을 경우 실제 서비스 대역폭 보다 아래로 대역폭이 떨어지더라도 버퍼 메모리에 항상 스트리밍이 있어서 재생이 가능할 수 있음을 보여준다.

두 번째 평가 항목은 네트워크의 대역폭 상태에 따라 서버에서 전송하는 NAQoS가 적용된 스트리밍에 대하여 조사하여 QoS의 품질 여부를 확인 하였다. 네트워크 상에 트래픽이 적을 경우와 많을 경우 또한 트래픽이 자주 발생하는 경우와 트래픽이 계속적으로 증가하였을 경우에 대하여 서버에서 전송하는 스트리밍에 대하여 분석하였다. 스트리밍 서버에서 1.5Mbps의 동영상 파일을 1.5Mbps와 0.5Mbps의 두 가지 비트율의 서비스를 하고 있으며 NAQoS 적용이 되지 않은 클라이언트는 버퍼 메모리의 스트리밍이 없을 경우 서버에 다시 서비스 요청을 해서 서비스 받는 방식을 택하였다. 네트워크 추측 시간 간격은 10초로 설정 하였다.



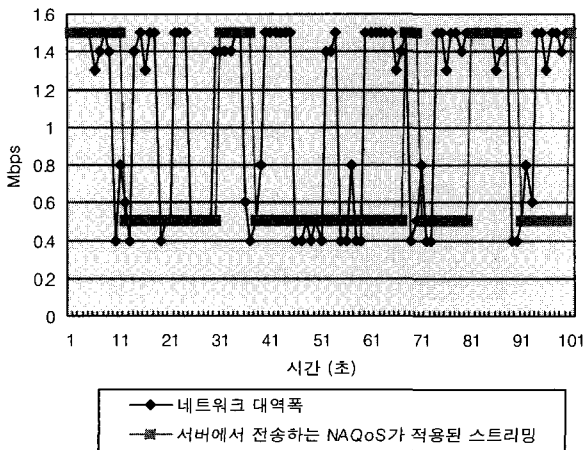
(그림 12) 트래픽이 일정 시간 동안 크게 증가한 경우 네트워크 반영 여부

(그림 12)는 트래픽이 일정 시간 동안 많이 발생하여 스트리밍 서비스가 허용 대역폭 밑으로 일정 시간 떨어진 경우의 네트워크 반영 여부에 대한 그림이다. 그림에서 보면 일정 시간 동안 네트워크에 많은 양의 트래픽이 발생하여 상위 비트율로 스트리밍 서비스를 하기에는 안 좋았을 경우 스트리밍 서버에서 보내는 스트리밍 또한 네트워크의 대역폭 상태를 반영하여 전송함으로 만족할 만한 QoS 제어를 하고 있음을 알 수 있다. 트래픽이 줄어들어 네트워크의 대역폭이 늘어나더라도 서버에서 상위 비트율의 스트리밍을 서비스를 하지 못하고 있는데 이것은 3.4절에서 일정시간(t)을 어떠한 형태로 결정하느냐에 따라 달라질 수 있다. (그림 12)의 경우에는 10초로 설정한 후 측정된 결과이다.



(그림 13) 트래픽이 일시적으로 증가할 경우 네트워크 반영 여부

(그림 13)은 트래픽이 일시적으로 증가하여 상위 비트율의 허용 대역폭 밑으로 떨어지는 경우의 네트워크 반영 여부를 나타낸다. 전체적으로 네트워크 대역폭 상태가 서비스 허용 대역폭보다 높지만 트래픽의 갑작스런 증가로 해당 비트율의 스트리밍 서비스를 할 수 없는 경우이다. (그림 13)에서도 NAQoS가 적용된 시스템에서는 네트워크 대역폭의 상태를 비슷하게 반영하고 있어서 만족할 만한 QoS를 제공하고 있다. 하지만 트래픽의 증가로 서비스 허용 대역폭 이하로 떨어진 시간에 비하여 서버에서 제공하고 있는 하위 서비스 대역폭 시간이 길다는 문제점이 있다.



(그림 14) 트래픽의 증가가 계속해서 발생하여 변화가 많은 경우 네트워크 반영 여부

(그림 14)는 네트워크 상에서 트래픽의 증가가 계속하여 발생하여 상위 비트율의 스트리밍 서비스를 지속적으로 할 수 없는 경우의 네트워크 반영 여부이다. (그림 14)에서 보면 실제 네트워크의 대역폭이 스트리밍 서비스하기에 좋은 대역폭 상태도 있지만 변화가 심하여 이런 네트워크의 상태에서는 사용자가 만족할 만한 스트리밍 서비스를 제공하는 것이 힘들다. 하지만 NAQoS가 적용된 스트리밍의 경우 전

체적으로 네트워크의 대역폭 상태 보다 낮은 레벨의 서비스가 이루어지고 있다. 또한 상위 비트율의 대역폭으로 서비스를 변경하였다가 다시 하위 비트율의 대역폭으로 떨어지는 경우도 볼 수 있다. (그림 12)와 (그림 13)의 경우는 클라이언트/서버 기반 스트리밍 시스템에서의 네트워크 적응형 QoS 기법을 적용하였을 경우 네트워크의 대역폭 상태를 반영하는 QoS 제어가 이루어졌으며 (그림 14)의 경우에는 만족할 만한 서비스를 제공하지는 않지만 클라이언트에서 스트리밍을 재생할 경우 끊어지지 않고 계속하여 서비스가 가능할 수 있음을 보여주고 있다.

5. 결론 및 향후 연구

기존의 인터넷 환경에서는 애플리케이션 프로그램의 안정성이 전송에 따른 지연보다도 중요 했으며, 데이터 전송 시 어느 정도의 지연은 서비스 특성상 그리 큰 문제가 되지 않았다. 그러나 사용자와 상호 작용하는 형태로 제공되는 인터넷의 멀티미디어 서비스에서 전송 지연은 연속되는 스트리밍 형태의 서비스 품질에 매우 심각한 문제를 일으켰다. 현재의 인터넷 환경은 이러한 실시간 형태의 멀티미디어 서비스에는 적합하지 않으며, 따라서 새로운 형태의 프로토콜과 라우터 및 호스트 환경 등 멀티미디어 네트워크 환경을 요구하고 있다.

하지만 이러한 QoS를 보장하기 위한 새로운 형태의 하드웨어나 소프트웨어의 개발은 고비용과 비효율성의 문제로 현실적으로 수용하기 어려운 부분이 많다. 또한, QoS가 지원되는 상황이라도 계속 적인 네트워크 사용량의 증가로 인하여 QoS를 지원을 위한 지속 적인 대책을 수립이 필요로 하게 된다.

따라서 본 논문에서는 인터넷에서 서비스의 QoS를 보장해 주기 위해서는 현재의 QoS 모델과는 다른 방식으로, 기존 인프라를 이용한 새로운 QoS 서비스 모델을 제안하였다. 제안된 클라이언트/서버 기반 스트리밍 시스템에서 네트워크 적응형 QoS 지원 기법은 기존의 프로토콜에서의 파라미터 추출을 통한 방식으로 실시간 QoS가 가능하여 저비용의 QoS 제어를 하며, 기존의 인프라를 그대로 이용하며 스트리밍 시스템의 서버와 클라이언트에 적은 변화만으로 최대의 효과를 얻을 수 있다. 또한, 네트워크 대역폭의 실시간 반영 기능으로 사용자가 요구하는 QoS를 제공 할 수 있는 장점이 있으며 네트워크의 종류에 관계없이 사용할 수 있어서 다른 QoS 기법에 비하여 유지 보수에 대한 비용이 적다.

제안한 NAQoS의 성능 평가 결과, 네트워크 대역폭 상태의 변화가 적을 경우에는 네트워크의 대역폭 상태에 관계 없이 만족할 만한 네트워크 상태를 반영하는 QoS 결과가 나왔다. 하지만 네트워크 대역폭의 대역폭 상태 변화가 많고 네트워크의 대역폭 상태가 전체적으로 상위 비트율 스트리밍 서비스를 하기에 안 좋을 경우에는 만족할 만한 네트워크 상태를 반영하지는 못하였으나 스트리밍이 클라이언트에서 끊어지지 않고 재생하는 것에 대해서는 만족할 만한 결과가 나왔다.

향후 연구로는, 현재 비트율이 다른 스트리밍 소스의 재

생에 대한 연구가 필요하며 스트리밍 서버에서 비트율이 다른 스트리밍 소스에 대한 전송 방식 및 버퍼링 및 캐싱 방법에 대한 연구가 필요하다. 또한 스트리밍 소스 콘텐츠에 대한 적용 알고리즘 또한 보완할 필요가 있으며 보안 QoS 파라미터를 이용하여 스트리밍 시스템의 보안에도 계속적인 연구가 필요하다.

참 고 문 헌

[1] D. Hazlett, "An interim solution to Internet congestion," Social Science Computing Review, 15, pp.181-189, 1997.
 [2] Vergsna, "IP Quality of Service," Cisco Press, 2001.
 [3] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture : An Overview," IETF RFC 1633, 1994.
 [4] J. Wroclawski, "The use of RSVP with IETF integrated services," RFC2210, 1997.
 [5] 전용희, 박수영, "DiffServe를 이용한 인터넷 QoS 보장 기술", 한국통신학회지, 제17권 제9호, pp.1152-1173, 2000.
 [6] Walter Weiss, "QoS with Differentiated Service," Bell Labs Technical journal, October-December, 1998.
 [7] J. Postel, "Service Mapping," RFC 795, 1981.
 [8] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," Work in progress, Internet-Draft, 1997.
 [9] A. Terzis, L. Wang, J. Ogawa, L. Zhung, "A Two-Tier Resource Management Model for the internet," In Proceedings of Global Internet Symposium '99, Rio de Janeiro, Brazil, 1999.
 [10] A. Kantawala, S. Norden, K. Wong, G. Parulkar, "DiSP : An Architecture for Supporting Differentiated Services in the Internet," Proceedings of INET99, 1999.
 [11] 김효곤, "Architectural and Practical Problems in Diff-Serv Deployment", 차세대인터넷 워크샵, 2000.
 [12] 진영민, "차세대 인터넷 기술 현황", 한국전산원 정보 동향 분석, 제4권 제21호, 1997.
 [13] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," INFOCOM '94. Networking for Global Communications. 13th Proceedings IEEE, Vol.2, pp. 680-688, 1994.
 [14] E. Steinbach, N. Farber, B. Girod, "Adaptive playout for low latency video streaming," Image Processing, 2001. Proceedings, 2001 International Conference on, Vol.1, pp.962-965 2001.
 [15] J. Bolot, S. Fosse-Parisis and D. Towsley, "Adaptive fec-based error control for internet telephony," Proceedings of IEEE INFOCOM, Vol.3, pp.1453-1460, 1999.
 [16] D. P. Olsen, D. Lanzinger, "Packet processing performance effects of interleaving and queueing," Aerospace Conference Proceedings, 2000 IEEE, Vol.1, pp.301-309, 2000.
 [17] Colin Perkins, Orion Hodson, Vicky Hardman, "A Survey of Packet-Loss Recovery Techniques," IEEE Network Ma-

gazine, Sep/Oct., 1998.
 [18] S. Floyd, V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks," IEEE/ACM Transactions on Networking, pp.365-386, 1995.
 [19] Ingo Busse, Bernd Deffner, Henning Schulzrinne, "Dynamic QoS Control of Multimedia Application based on RTP," Computer Communications, Vol.19, pp.49-58, May, 1995.
 [20] J. D. Salehi, et. al., "Supporting stored video : Reducing rate variability and end-to-end resource requirements through optimal smoothing," in Proc. of ACM SIGMETRICS, pp. 222-231, May, 1996.
 [21] P. Assuncao and M. Ghanbari, "A Frequency-domain video transcoder for dynamic bitrate reduction of MPEG-2 bitstreams," IEEE Trans. Circuits Syst. Video Technol., Vol.8, pp.935-967, Dec., 1998.
 [22] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," VCIP, January, 2001.
 [23] 박대훈, 허혜선, 홍윤식, "비디오 스트리밍 데이터 전송시 RTP를 이용한 효율적인 네트워크 트래픽 제어", 정보과학회 논문지, 제8권 제3호, pp.328-335, 2002.
 [24] 김재욱, 하 관, 차호정, "멀티미디어 전송을 위한 온라인 대역폭 평활화 기법", 정보과학회논문지, 제27권 제4호, pp.522-530, 2000.



정 언 일

e-mail : zhung@oslab.khu.ac.kr
 2000년 경희대학교 컴퓨터공학과 석사
 2003년 경희대학교 컴퓨터공학과 박사수료
 현재 경희대학교 컴퓨터공학과 박사과정
 관심분야 : 시스템 및 네트워크 보안, QoS



이 정 찬

e-mail : chan75@oslab.khu.ac.kr
 2002년 대불대학교 정보통신공학과 학사
 2002년~현재 경희대학교 컴퓨터공학과 석사과정
 관심분야 : 네트워크, 스트리밍, QoS



이 승 통

e-mail : sylee@oslab.khu.ac.kr
 1978년 고려대학교 재료공학과 졸업
 1986년 Illinois Institute of Technology 전산학과 석사
 1991년 Illinois Institute of Technology 전산학과 박사

1991년~1993년 Governors State University 조교수
 1993년~현재 경희대학교 컴퓨터공학과 교수
 관심분야 : 실시간 컴퓨팅, 실시간 미들웨어, 멀티미디어 시스템, 시스템 보안