

SVC 스트리밍을 위한 시간 계층 기반의 동적 큐 관리 알고리즘

(An Active Queue Management Algorithm Based on the
Temporal Level for SVC Streaming)

구 자 현 [†]

정 광 수 ^{**}

(Jahon Koo)

(Kwangsue Chung)

요약 최근 광 대역 통합 네트워크에서 고품질의 멀티미디어 서비스에 대한 사용자 요구가 증가하고 있다. 또한, 사용자 단말기기의 다양화 및 대화면 디스플레이 장치의 보급으로 다양한 형태의 서비스 품질(QoS)에 대한 요구도 증가하고 있다. 이를 위해 네트워크 관점에서 동적 큐 관리 알고리즘과 같은 인터넷 성능을 개선하여 서비스 품질을 보장하는 연구와 종단 관점에서 미디어의 품질을 보장하기 위한 SVC(Scalable Video Coding) 부호화 기법에 대한 연구가 활발히 진행 중에 있다. 그러나, 기존 동적 큐 관리 알고리즘은 비디오 부호화 기술의 본질적인 특성에 대하여 고려하지 못하여 서비스 품질을 보장하는 못하는 문제점을 가지고 있다.

본 논문에서는 현재 혼잡제어 알고리즘의 문제점을 개선하기 위해 NAL(Network Abstract Layer)의 헤더 내 TID(Temporal ID)를 통해 SVC 부호화 기술의 특성을 파악하여 프레임간 의존성이 낮은 프레임의 패킷에 대하여 차등적으로 패킷을 폐기하는 TS-AQM(Temporal Scalability - Active Queue Management) 알고리즘을 제안하였다. 제안한 TS-AQM 알고리즘은 혼잡상황 시 차등적인 패킷 폐기를 통해 SVC 부호화 기법을 이용하는 스트리밍 서비스에 대하여 안정적인 비디오 복호화를 통해 멀티미디어 서비스 품질을 보장하였다.

키워드 : 동적 큐 관리, 스케일러블 비디오 코딩, 시간 계층, NAL, TS-AQM, TID

Abstract In recent years, the user demands have increased for multimedia service of high quality over the broadband convergence network. These rising demands for high quality multimedia service led the popularization of various user terminals and large scale display equipments, which needs a variety type of QoS (Quality of Service). In order to support demands for QoS, numerous research projects are in progress both from the perspective of network as well as end system; For example, at the network perspective, QoS guaranteeing by improving of internet performance such as Active Queue Management, while at the end system perspective, SVC (Scalable Video Coding) encoding scheme to guarantee media quality. However, existing AQM algorithms have problems which do not guarantee QoS, because they did not consider the essential characteristics of video encoding schemes.

In this paper, it is proposed to solve this problem by deploying the TS-AQM (Temporal Scalability Active Queue Management) which employs the differentiated packet dropping for dependency of the temporal level among the frames, based on SVC encoding characteristics by exploiting the TID (Temporal ID) field of the SVC NAL unit header. The proposed TS-AQM guarantees multimedia service quality through video decoding reliability for SVC streaming service, by differentiated packet dropping when congestion exists.

Key words : AQM, SVC, Temporal Level, NAL, TS-AQM, TID

· 본 연구는 지식경제부의 전자기술개발사업의 지원을 받아 수행된 연구결과임

† 학생회원 : 광운대학교 전자통신공학과

jhkoo@cclab.kw.ac.kr

** 종신회원 : 광운대학교 전자통신공학과 교수

kchung@kw.ac.kr

논문접수 : 2009년 3월 31일

심사완료 : 2009년 7월 29일

Copyright©2009 한국정보과학회 : 개인 목적이거나 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 제5호(2009.10)

1. 서론

최근 전 세계적으로 고도화된 IT 기술을 이용한 다양한 초고속 인터넷 서비스가 개발되고, NGN(Next Generation Network)등 광 대역 통합 네트워크 구축이 진행되고 있다. 또한, 유무선 통합 및 방송통신 융합이 점점 가속화 되고 있으며, 이를 기반으로 새로운 응용 서비스 및 네트워크 기술이 활발하게 연구 및 개발되고 있다. 특히 Mobile TV, IPTV와 사용자 제작 콘텐츠(UCC)등 다양한 멀티미디어 응용 서비스들이 개발 및 이용 되고 있다. 이로 인해 광 대역 통합 네트워크에 오디오나 비디오 스트리밍 서비스와 같은 멀티미디어 트래픽의 이용량이 지속적으로 증가하고 있다. 이를 반영하듯 광 대역 통합 네트워크에서 고품질의 멀티미디어 서비스에 대한 사용자 요구도 증가하고 있다. 또한, 사용자 단말 기기의 다양화 및 대화면 디스플레이 장치의 보급으로 다양한 형태의 서비스 품질(QoS)에 대한 요구도 증가하고 있다[1].

이를 위해 네트워크 관점에서 인터넷 성능을 개선하여 서비스 품질을 보장하는 연구와 종단 관점에서 미디어의 품질을 보장하기 위한 연구가 활발히 진행 중에 있다. 먼저, 종단 관점에서 다양한 네트워크 및 이 기종 단말 환경에서 미디어 품질 보장을 위해 공간적, 시간적, 품질적 확장성을 제공하는 H.264/AVC를 확장한 부호화 기술인 SVC(Scalable Video Coding)에 대한 연구 및 표준화가 진행 중에 있다[2]. 그리고, 지속적으로 증가하는 인터넷 트래픽으로 인하여 발생하는 혼잡상황에 대하여, 네트워크 관점에서 인터넷 서비스 성능을 개선하기 위해 혼잡상황을 효과적으로 제어하는 라우터 알고리즘에 대한 연구가 진행 중에 있다. 그 중에서도 구현이 용이하며 적은 복잡성을 가지는 동적 큐 관리(Active Queue Management) 알고리즘에 대한 연구가 활발히 진행되고 있다[3,4].

그러나, 기존 동적 큐 관리 알고리즘은 비디오 부호화 기술의 본질적인 특성에 대하여 고려하지 못하여 서비스 품질을 보장하는 못하는 문제점을 가지고 있다. 즉, 동적 큐 관리 알고리즘은 네트워크 혼잡상황 발생시, 혼잡제어를 위해 임의의 패킷을 폐기한다. 이는 상호 의존성을 가지는 프레임 정보를 유실 시켜, 비디오 복호화시 갑자기 움직이는 화면(Jerky Picture)등을 야기시킨다. 이로 인해 전체적인 멀티미디어 서비스 품질이 저하된다.

본 논문에서는 현재 혼잡제어 알고리즘의 문제점을 개선하기 위해 비디오 부호화 기술의 특성을 파악하여 프레임간 의존성이 낮은 프레임의 패킷에 대해 차등적으로 패킷을 폐기하는 TS-AQM(Temporal Scalability

Management) 알고리즘을 제안하였다. 제안한 TS-AQM 알고리즘은 혼잡상황 시 차등적인 패킷 폐기를 통해 SVC 부호화 기법을 이용하는 스트리밍 서비스에 대하여 안정적인 비디오 복호화를 제공하고 이를 통해 멀티미디어 서비스 품질을 보장하였다.

2장에서는 관련연구로 SVC 부호화 기법과 AQM 기술에 대하여 알아보고 기존 AQM 기술의 문제점에 대하여 지적하였다. 3장에서는 새로 제안한 TS-AQM 알고리즘에 대하여 기술 하였다. 4장에서는 시뮬레이션을 통해 제안한 알고리즘의 성능을 검증하였다. 마지막으로 5장에서 결론을 맺었다.

2. 관련연구

2.1 Scalable Video Coding(SVC)

H.264/AVC를 확장한 부호화 기술인 SVC는 기존의 MPEG-2, MPEG-4등에서 시도한 계층 부호화 기반의 확장성(Scalability)이 갖는 문제점인 낮은 압축효율, 복합 확장성의 부재, 높은 구현 복잡성의 문제를 해결하기 위해 개발된 새로운 확장 비디오 부호화 기법이다. 확장형 부호화 기술인 SVC는 크게 VCL(Video Coding Layer) 계층과 NAL(Network Abstraction Layer)계층 두 개의 계층으로 구분된다. 먼저, SVC의 VCL 계층에서는 영상 정보의 부호화 정보인 기본 계층 부호화 정보와 상위 계층의 확정성 부호화 정보를 슬라이스(Slice) 단위로 생성한다. VCL에서 생성한 각 슬라이스는 네트워크를 통해 효율적으로 전송되기 위해, NAL 계층에서 NAL Unit 단위로 만들어지게 되며, NAL Unit 단위로 구성된 SVC 비트스트림(Bitstream)은 IP 네트워크 등을 통해 전달할 수 있는 구조를 가지게 된다[2,5,6].

이러한, SVC 부호화 기술은 그림 1과 같이 공간적, 시간적, 품질적 확장성을 지원하는 복합적인 확장 비디오 부호화 방법을 가지고 있다. SVC를 이용하여 한번 부호화된 비트스트림은 두 개 혹은 그 이상의 의존적인 계층으로 구성될 수 있다. 이와 같은 경우 SVC 비트스트림은 하나의 기본계층(Base layer)과 기본계층 위에 연속적으로 쌓을 수 있는 다수의 상위계층(Enhancement layer)들로 구성된다. 여기서 기본계층에 연속되는 상위계층의 정보를 함께 이용하면, 기본계층의 비트스트림 보다 개선된 화질을 제공하는 비트스트림을 만들 수 있다. 일반적으로 기본계층은 기본 품질을 제공하고, 연속된 상위 계층은 이전 계층들로 만들어진 비디오보다 높은 품질을 갖도록 부호화한다.

공간적인 확장성은 상·하향 표본화를 통해 여러 해상도를 가지는 영상들을 계층적으로 구성하여 제공한다. 우선 하위계층의 크기가 작은 영상을 부호화한 후, 상위

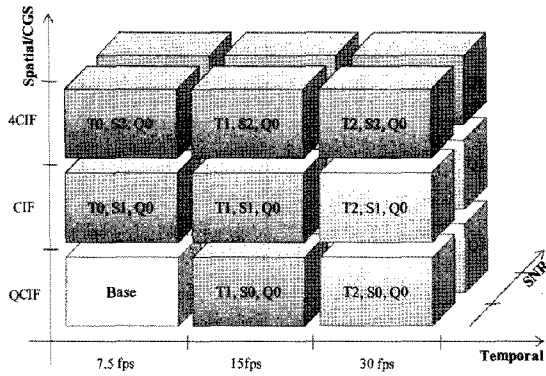


그림 1 SVC의 Layered Scalable Bitstream

계층은 부호화된 하위계층의 정보를 이용함으로써 부호화 효율을 높인다. 시간적 확장성은 MCTF(Motion Compensated Temporal Filtering) 또는 Hierarchical B Picture 기법을 이용한다. 이는 매 프레임마다 구별된 시간적 레벨을 지정함으로써 다양한 프레임율을 제공하며 GOP(Group of Pictures) 단위로 처리한다. 하나의 GOP에서는 적어도 하나의 주요 Pictures를 가지며 다른 Picture들은 이 주요 Picture의 의존성을 가진다. 마지막으로 화질 확장성은 CGS(Coarse Grain Scalability)와 FGS(Fine Grain Scalability)에 의해 구현된다. CGS는 계층 기반인 측면에서 공간적 확장성과 동일한 방식이지만, 계층 간에 동일한 해상도를 가지는 반면, 양자화(QP: Quantization Parameter) 값을 다르게 할당하여 상위 계층일수록 높은 화질을 제공한다. FGS는 Bit-plane 부호화 기법과 유사한 Cyclic 부호화 기법과 2-scan 부호화 기법으로 만들어지며, CGS보다 세밀한 비트 양 및 화질 차이를 제공할 수 있다.

SVC의 시간적 확장성 예로 그림 2는 0부터 4까지 5개의 Temporal Level을 가지는 확장성 비디오 비트스트림을 보여주고 있다. 이 비트스트림은 16개의 GOP Size를 가지고 있으며 Hierarchical B-Picture 기법을 이용하여 총 5개의 Temporal Level를 가지고 있다. 주요 Picture에 해당하는 I-picture는 Temporal Level 0을 가진다. 각 화살표는 Reference Picture로부터 Predicted Picture의 연관성에 대하여 나타낸다. 그림 2와 같이 I/P-picture의 경우 높은 Temporal Level을 가지는 B0, B1, B2-Picture들과 연관성을 가짐을 알 수 있다[2,5,6].

SVC의 NAL 계층은 영상정보를 부호화한 VCL 계층 정보를 NAL Unit 단위로 분배하여 전송계층과 연결한다. 이처럼 VCL과 NAL 계층을 통해 생성된 SVC 부호화 정보는 NAL Unit 단위의 비트스트림으로 저장되며, 각 NAL Unit은 1byte의 NAL Unit 헤더를 가지고 있다. 그리고, 그림 3과 같이 NAL Unit 헤더 생성 시 추적인 정보 필드를 가지고 있다[7].

일반적으로 SVC 부호화 정보는 기본계층에서 생성된 NAL Unit과 확장계층에서 생성된 NAL Unit의 헤더 구조를 가진다. 기본계층의 NAL Unit은 H.264/AVC와 호환이 되어야 하므로 H.264/AVC의 NAL Unit 헤더 구조와 동일한 필드 3가지인 F(forbidden_zero_bit), NRI(nal_ref_idc), Type(nal_unit_type)으로 구성되며 총 1바이트 크기를 갖는다. 상위 계층에서 생성된 NAL Unit 헤더의 경우 기본계층 NAL Unit 헤더에 추가적으로 3바이트 크기를 갖는 새로운 헤더 필드를 가지고 있다. 그림 3에서 TID(Temporal_ID), DID(Dependency_ID), QID(Quality_ID) 필드는 각각 시간적, 공간

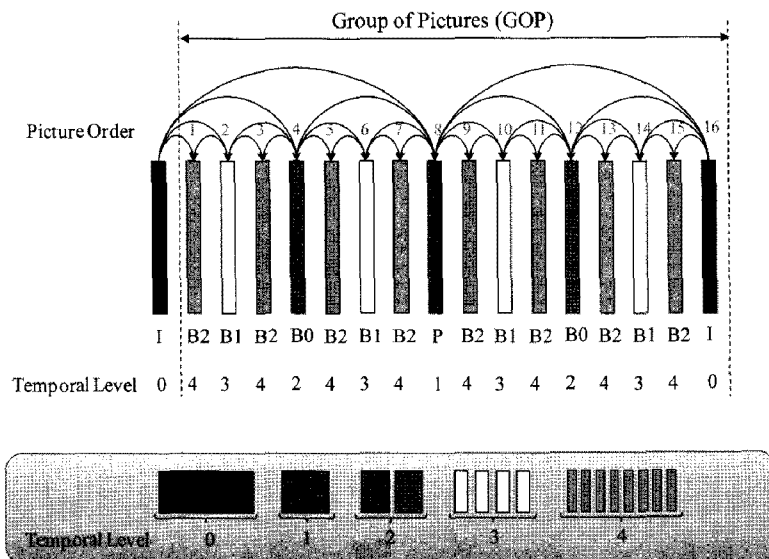


그림 2 Hierarchical B-Picture 기법을 사용한 SVC 비디오 부호화 예시

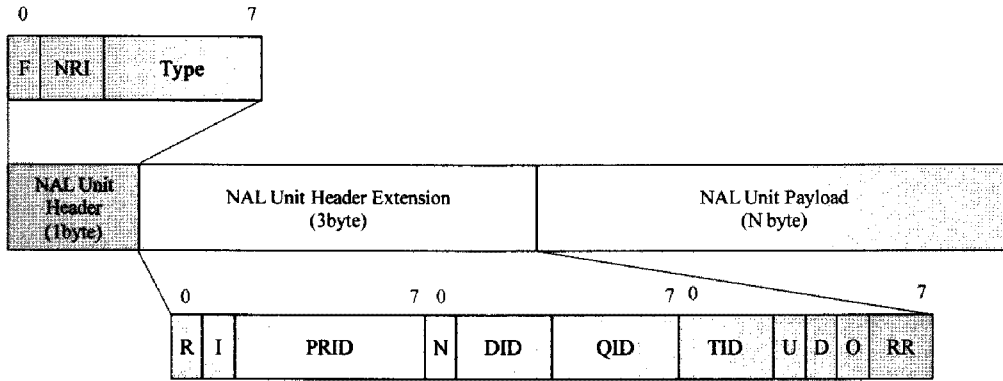


그림 3 SVC NAL Unit Header와 Extension Header 구조

적, SNR 확장성에서 각 계층간의 시공간적 품질관련 체계를 나타낸다. 즉, TID는 시간적 확장성을 위한 Temporal level간의 체계(hierarchy)를 나타내고, DID는 공간적 확장성의 계층간 예측(inter-layer prediction)에 있어서 상·하위 계층 간의 종속 체계(dependency hierarchy)를 나타내며, QID는 SNR 확장성 지원을 위한 FGS 계층간의 계층 체계를 나타낸다. TID, DID, QID 값은 모두 0 이상의 정수이며, 이 값들의 조합으로부터 각 NAL Unit에 대한 시공간적 품질계층 관계를 유도해 낼 수 있다.

2.2 큐 관리 알고리즘

Per-hop 단위로 패킷 처리를 하는 IP 네트워크에서의 혼잡제어 방법은 혼잡성이 주로 발생하는 라우터나 게이트웨이의 출력(output interface)링크에서 링크 자원인 큐를 효율적으로 관리하는 라우터 알고리즘을 이용하는 방법이다. 일반적으로 라우터는 유입되는 패킷을 라우팅하기 위하여 라우터로 들어오는 패킷들을 큐에 임시적으로 저장한다. 이때 가장 간단하게 저장하는 방법으로 FIFO(First-In First-Out) 큐가 사용된다. 만약 라우터로 들어오는 패킷의 입력 트래픽이 라우터를 통하여 나가는 출력링크로 처리할 수 있는 트래픽을 넘는 경우 임시로 큐에 저장을 하고 저장이 안된 패킷들은 모두 버려지게 된다. 이러한 기본적인 FIFO 처리 방식을 Drop Tail이라 한다. 이 방법은 큐 관리가 가장 간단하여 구현하기는 쉽지만 플로우(flow)간의 불공정한 출력링크 할당, 전역 동기화(global synchronization)에 따른 평균지연 시간의 증가 및 성능 감소, 그리고 혼잡 붕괴의 가장 큰 원인이 되는 “풀 큐(full queue)”와 “락아웃(lock out)”을 발생시키는 문제점을 가지고 있어 혼잡상황 제어에 적합하지 않다[3,4].

현재 인터넷의 라우터나 게이트웨이에서 발생하는 혼잡상황을 해결하기 위해서 여러 가지 혼잡제어 방법에 대한 연구가 활발히 진행되고 있다. IETF의 RFC 2309 문서에서 권고하고 있는 라우터나 게이트웨이의 큐를 관

리하는 혼잡제어 알고리즘은 크게 두 가지로 분류 할 수 있다. 첫 번째는 스케줄링 알고리즘으로 대표적인 방법으로는 FQ(Fair Queueing) 알고리즘이 있다. 이 알고리즘은 각 플로우에 대하여 개별적인 큐를 분리하여 관리하는 per-flow방식을 사용한다. 그러나 흐름들에 관한 정보를 유지하고 처리하기 위해서 복잡한 알고리즘을 필요로 하기 때문에 고속의 라우터 처리 능력을 요구한다. 또한, 많은 플로우를 갖는 네트워크 환경에서 널리 사용하기에는 많은 어려움과 오버헤드가 필요하다[4].

두 번째로 처음부터 간단한 구조로 설계된 능동적 큐 관리 알고리즘이 있다. 이 방법은 간단하면서도 어느 정도의 공정성 및 혼잡상황을 제어하는 기능을 제공한다. 대표적인 능동적 큐 관리 알고리즘으로는 RED 알고리즘이 있다[8]. RED 알고리즘은 모든 플로우가 하나의 큐를 통하여 처리가 되며 네트워크 혼잡정도에 따라 큐가 관리된다. 따라서 스케줄링 알고리즘보다 적은 비용으로도 네트워크에 발생하는 혼잡 상황 문제를 해결할 수 있다.

RED 알고리즘은 혼잡 상황이 발생하기 이전에 평균 큐 크기를 기반으로 그림 4와 같이 평균 큐 크기의 변화의 정도를 기반으로 혼잡 상황을 판별하며, 혼잡 상태에 대한 피드백 정보를 패킷의 폐기나 패킷 헤더에 혼잡 상황을 표시(marking)하는 ECN(Explicit Congestion Notification)을 이용하여 종단 호스트인 송신 측에게 혼잡정보를 알려 준다.

RED 알고리즘은 EWMA(Exponential Weighted Moving Average)를 이용하여 평균 큐 크기를 계산하며 혼잡상황이 발견되면 평균 큐 크기에 따라 비례적으로 적절한 패킷 폐기 확률 값을 계산한다. 예를 들어 평균 큐 크기가 최소 경계 값(minimum threshold: min_{th})을 초과하면 패킷들은 패킷 폐기 확률 값에 따라 랜덤(random)하게 폐기되거나 IP 패킷 헤더의 ECN bit를 이용하여 표시한다. RED 알고리즘은 네트워크 혼잡상황을 피하기 위해서 TCP와 같이 피드백 정보를 기반으

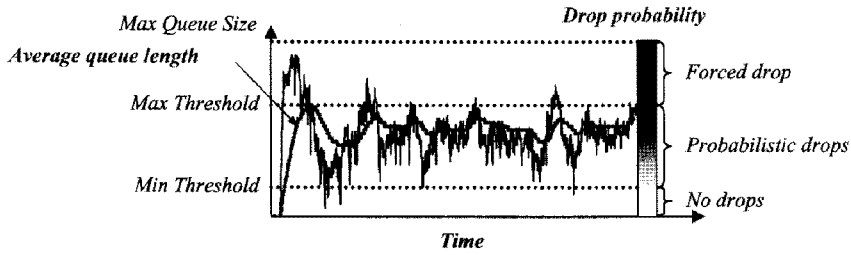


그림 4 RED 알고리즘의 동작

로 혼잡제어를 하는 전송 프로토콜과 상호 동작한다. 그리고 평균 큐 크기가 최대 경계 값(maximum threshold: max_{th})을 초과할 경우 모든 패킷들은 폐기 또는 표시한다[8].

RED 알고리즘의 동작 방법은 이처럼 기존의 혼잡제어 방법과는 다르게 혼잡상황을 예상하여 라우터에 도착하는 임의의 플로우의 패킷을 폐기하기 때문에 drop-tail 방법이 가지고 있는 “락 아웃”과 “풀 큐” 문제를 해결 할 수 있다. 또한 미리 혼잡상황을 발견하여 TCP 등의 전송률을 줄여 혼잡제어를 수행하기 때문에 불필요한 패킷 손실로 발생하는 네트워크 자원의 낭비를 Drop Tail 방법 보다 줄여 좋은 성능을 나타낼 수 있다. 그리고 효과적인 큐 관리로 평균 큐 크기를 작게 유지할 수 있어 큐잉(queueing)으로 생기는 전송지연 시간을 줄일 수 있다. 일반적으로 RED 알고리즘을 이용하는 방법이 Drop Tail 방법을 사용하는 것보다는 좋은 성능을 제공한다[4,8].

2.3 문제점과 프레임 복호화 에러 확률

인터넷 서비스 성능 개선을 위해 개발된 큐 관리 알고리즘 들은 동작 특성상 SVC 부호화 기법을 이용하는 스트리밍 서비스에 대하여 적합한 동작 및 서비스를 제공하지 못한다. 동적 큐 관리 알고리즘은 혼잡제어를 위해 각 플로우 속성에 대한 구분 없이 모든 플로우에 대하여 혼잡상황이 판단되면 임의의 패킷을 폐기한다. 이러한 동작은 단편(Fragment)되어 있는 SVC 비트스트림의 GOP내 특정 Picture의 복호화를 어렵게 하거나 실패하게 한다. 즉, 단편된 특정 패킷의 유실로 전체 Picture을 복호화하지 못하게 하거나, 다른 Picture의 복호화를 의존성을 제공하는 경우 패킷 유실로 자신 이외의 다른 Picture까지 복호화가 되지 않아 종단간 측면에서의 네트워크 효율(Efficiency)은 떨어지게 된다. 그리고, 각 Picture간 의존성이 높은 SVC 기반의 스트리밍 서비스에 대하여 최악의 경우 잘못된 혼잡제어 동작을 통해 화면에 영상이 출력되지 않는 서비스 붕괴를 야기 시킬 수 있다.

먼저 Level 계층간 의존성을 가지는 Temporal Level 별 패킷 폐기 확률을 계산하면 다음과 같다. 대표적인

큐 관리 알고리즘인 RED의 경우 식 (1)과 같은 패킷 폐기 확률을 가진다. 여기서, ω 는 평균 큐 크기이다.

$$\therefore P(\omega) = \frac{P_{max}}{(max_{th} - min_{th})} \times (\omega - min_{th}) \quad (1)$$

총 4개의 Temporal Level을 가지는 SVC 비트스트림의 경우 다음 식 (2), (3), (4), (5)와 같은 패킷 폐기 확률을 가진다. 여기서 n_{Levelx} 은 각 Temporal Level별 패킷 수를 의미한다.

$$P_{Total}^{Level4} = \int_{min_{th}}^{max_{th}} \frac{n_{Level4}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (2)$$

$$P_{Total}^{Level3} = \int_{min_{th}}^{max_{th}} \frac{n_{Level3}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (3)$$

$$P_{Total}^{Level2} = \int_{min_{th}}^{max_{th}} \frac{n_{Level2}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (4)$$

$$P_{Total}^{Level1} = \int_{min_{th}}^{max_{th}} \frac{n_{Level1}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (5)$$

이 패킷 폐기 확률은 min_{th} , max_{th} 의 구간 크기 이외에 각 Temporal Level별 패킷 개수에 비례하며, 만약 각 Temporal Level별 동일 개수의 패킷을 가질 경우 각 Temporal Level의 패킷 폐기 확률은 동일할 수 있다. 그러나 그림 5와 같이 각 Picture를 표현하는 각 프레임의 패킷 수는 다른 개수를 가지고 있다.

그림 5와 같은 패킷 개수를 가질 경우 다른 프레임들과 복호화시 높은 의존성을 가지는 Temporal Level이

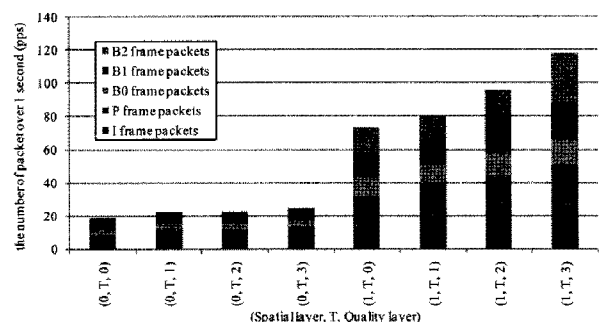


그림 5 SVC의 Temporal Level별 패킷 개수 비교

낮은 I 프레임에 가장 높은 패킷 폐기 확률이 적용된다. 따라서, 일반적인 동적 큐 관리 알고리즘을 사용 할 경우 인터넷 서비스를 개선할 수 있지만, SVC 기반의 스트리밍 서비스의 품질을 저하시키는 문제점을 가지게 된다.

그리고, 이렇게 상호 의존성을 가지는 Temporal Level별 각 프레임의 정상적인 복호화 성공율을 나타내는 $rate_{frame}^{decoding}$ (Decoding Success Rate)과 IP 패킷 단위로 단편 된 각 프레임의 정상적인 조립 성공율을 나타내는 $rate_{frame}^{assemble}$ (Assemble Success Rate)은 다음과 같은 식 (6), (7), (8), (9), (10)의 관계를 가진다. 참고적으로 본 논문에서는 패킷 손실에 따른 복호화 성공율을 비교하기 위해 SVC 부호화 기법이 제공하는 FEC (Forward Error correction) 특성을 배제하여 다음 수식을 유도하였다.

$$rate_{l frame}^{decoding} = rate_{l frame}^{assemble} \quad (6)$$

$$rate_{p frame}^{decoding} = rate_{l frame}^{decoding} \times rate_{p frame}^{assemble} \quad (7)$$

$$rate_{B0 frame}^{decoding} = 0.5 \times rate_{l frame}^{decoding} \times rate_{B0 frame}^{assemble} + 0.5 \times rate_{p frame}^{decoding} \times rate_{B0 frame}^{assemble} \quad (8)$$

$$rate_{B1 frame}^{decoding} = 0.25 \times rate_{l frame}^{decoding} \times rate_{B1 frame}^{assemble} + 0.25 \times rate_{p frame}^{decoding} \times rate_{B1 frame}^{assemble} + 0.5 \times rate_{B0 frame}^{decoding} \times rate_{B1 frame}^{assemble} \quad (9)$$

$$rate_{B2 frame}^{decoding} = 0.125 \times rate_{l frame}^{decoding} \times rate_{B2 frame}^{assemble} + 0.125 \times rate_{p frame}^{decoding} \times rate_{B2 frame}^{assemble} + 0.25 \times rate_{B0 frame}^{decoding} \times rate_{B2 frame}^{assemble} + 0.5 \times rate_{B1 frame}^{decoding} \times rate_{B2 frame}^{assemble} \quad (10)$$

위 (10) 수식과 같이 Temporal Level이 높은 B2 프레임의 경우 Temporal Level이 낮은 I 프레임에 가장 많이 의존한다. 즉, Temporal Level 증가할수록 아래 Temporal Level에 의존함을 알 수 있다. 따라서, 높은 Temporal Level을 성공적으로 복호화 하기 위해서는 기초가 되는 낮은 Temporal Level의 프레임을 성공적으로 복호화 해야 한다. 각 프레임 별 $rate_{frame}^{assemble}$ 는 식 (11)과 같이 표시할 수 있다. 여기서, E_{frame} 는 해당 프레임이 정상적으로 조립되지 못할 프레임 에러 확률이다.

$$rate_{frame}^{assemble} = 1 - E_{frame} \quad (11)$$

수신 단에서 여러 개의 패킷으로 단편화된 프레임이 정상적으로 조립되지 못할 프레임 에러 확률은 식 (12)로 표현 할 수 있다. 여기서, L 은 하나의 프레임이 단

편화된 총 패킷 수이고, P_{frame} 은 해당 프레임의 패킷이 네트워크에서 폐기될 확률이다.

$$E_{frame} = \sum_{i=1}^L i P_{frame} (1 - P_{frame})^{L-i} \quad (12)$$

이 프레임 에러 확률은 확률론에 따라 식 (13)과 같이 변환이 가능하다[9].

$$E_{frame} = L + 1 + \frac{(1 - P_{frame})^{L+1} - 1}{P_{frame}} \quad (13)$$

따라서, 이 프레임 에러 확률은 $L > 1$ 이고 $1 \geq P_{frame} \geq 0$ 에서 L 과 P_{frame} 에 비례한 식 (14)와 같은 관계로 표현할 수 있다.

$$\therefore E_{frame} \propto L \times P_{frame} \quad (14)$$

식 (14)를 통해 알 수 있듯이 각 프레임의 에러 확률은 단편화된 패킷 수(L)와 각 프레임 별 패킷 폐기 확률(P_{frame})에 비례한다. 그런데, 그림 5에서와 같이 대화면 디스플레이 장치를 위한 Spatial Layer의 확장이나, 고품질 영상을 위한 Quality Layer의 확장은 L 값 및 P_{frame} 를 증가시킨다. 따라서, 현재의 라우터 기법으로는 대화면 고품질 영상에 대한 비디오 부호화가 매우 어려워 질 수 있다. 또한, 이러한 부호화 실패는 네트워크의 효율을 저하시켜 인터넷 성능을 저하시킨다.

본 논문에서는 임의의 패킷을 폐기하는 현재 큐 관리 알고리즘의 문제점을 개선하기 위해 비디오 부호화 기술의 특성을 파악하여 패킷의 중요도에 따라 차등적으로 패킷을 폐기하는 TS-AQM(Temporal Scalability - Active Queue Management) 알고리즘을 제안한다. 제안한 TS-AQM 알고리즘은 혼잡상황 시 Temporal Level별로 차등적인 패킷 폐기를 통해 SVC 부호화 기법을 이용하는 스트리밍 서비스에 대하여 안정적인 비디오 복호화를 제공하고 이를 통해 미디어 서비스 품질을 보장한다.

3. TS-AQM 알고리즘

본 논문에서는 혼잡상황에 대하여 Temporal Level에 따라 차등적으로 패킷을 폐기하는 새로운 TS-AQM (Temporal Scalability - Active Queue Management) 알고리즘을 제안한다. 각 Temporal Level은 SVC 부호화의 NAL 계층에서 제공하는 NAL Unit 헤더 내 TID 정보를 이용하였다. 제안한 TS-AQM은 그림 6과 같은 처리 구조를 가지고 있다. 입력된 패킷들의 EWMA기반의 평균 큐 크기를 이용하여 혼잡여부를 인지하고, 평균 큐 크기의 값에 따라 혼잡정도를 판별하여, 이에 비례한 확률로 수신된 패킷을 폐기한다.

그리고, 그림 7과 같이 TS-AQM은 혼잡정도에 따라 입력된 패킷에 대하여 SVC 스트리밍 패킷을 구분하고

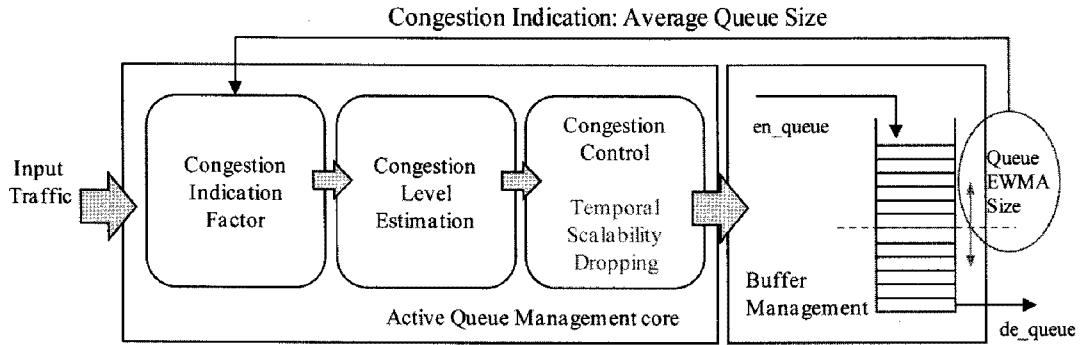


그림 6 TS-AQM 알고리즘의 처리 구조

```

For each packet arrival
  Calculate the average queue size  $avg_q$ 

  if  $min_{th} \leq avg_q < max_{th}$ 
    if (SVC Packet)
      Temporal Scalability Dropping
    else
      Calculate probability  $P_a$ 
      Drop the arriving packet with probability  $P_a$ 
  else if  $max_{th} \leq avg_q$ 
    Drop the arriving packet
    
```

그림 7 TS-AQM 알고리즘

킷 폐기 확률의 정도를 결정하는 P_{max} 값을 Temporal Level에 따라 그림 8과 같이 임계 구간별 다르게 증가시킨다. 따라서, 동일 혼잡구간에서 높은 TID값을 가지는 패킷이 낮은 TID값을 가지는 패킷보다 높은 확률로 폐기된다.

이러한 TSD 기법을 도시화 하면 그림 9와 같다. 그림 9와 같이 각 Temporal Level에 따른 패킷 폐기 확률은 혼잡정도의 구간에 따라 차등적으로 적용됨을 알

혼잡정도가 특정 조건을 만족할 경우 TSD(Temporal Scalability Dropping) 기법에 따라 패킷을 폐기한다.

TS-AQM은 기존 NAL 헤더에서 확장된 3byte의 NAL Unit 헤더 내 3bit의 TID값을 이용하여 그림 8과 같이 TID 값에 따른 차등적인 패킷 폐기를 수행한다. TSD 기법은 Temporal Level에 따라 차등적인 동작을 위해 임의의 임계 값 구간을 정의하고 있다. 이 임계값 구간에 따라 각 Temporal Level은 구분되며 또한, 혼잡정도에 따라 Temporal Level별 다른 패킷 폐기 확률을 가지게 된다. 이 동작은 프레임간 복호화 의존성이 없고, 우선 순위가 낮은 Temporal Level에 대하여 혼잡구간에서 우선적으로 패킷을 폐기한다. 이를 위해 패

```

Temporal Scalability Dropping
if  $min_{th} \leq avg_q < level3_{th}$ 
  if (TID > 3)
    Packet Dropping with probability  $\alpha \cdot (TID - 3) \cdot P_a$ 
  else if  $level3_{th} \leq avg_q < level2_{th}$ 
    if (TID > 2)
      Packet Dropping with probability  $\alpha \cdot (TID - 2) \cdot P_a$ 
  else if  $level2_{th} \leq avg_q < level1_{th}$ 
    if (TID > 1)
      Packet Dropping with probability  $\alpha \cdot (TID - 1) \cdot P_a$ 
  else if  $level1_{th} \leq avg_q \leq max_{th}$ 
    if (TID > 0)
      Packet Dropping with probability  $\alpha \cdot (TID) \cdot P_a$ 
    
```

그림 8 Temporal Scalability Dropping 기법 (GOP8의 경우)

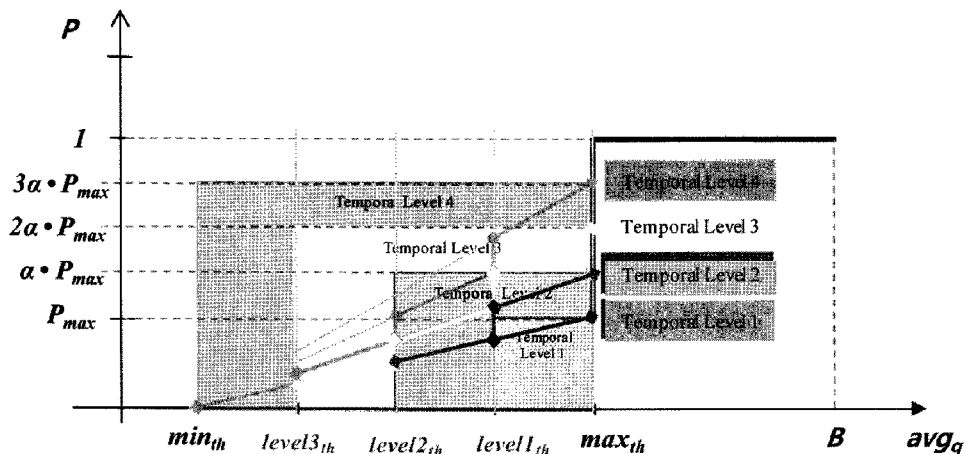


그림 9 TSD 기법의 동작 도시화 (GOP8의 경우)

수 있다. 즉, 각 Temporal Level에 대하여 동일 혼잡정도 구간에서 TID 값이 높을 경우 TID 값에 비례적으로 패킷 폐기 확률 값이 증가한다.

혼잡정도를 구분하는 임계값들은 SVC에서 제공하는 최대 Temporal Level 수에 따라 조정되며 이 Temporal Level은 라우터가 동작 중 주기적으로 최대 TID 값(TID_{MAX})을 수집하여 이용한다. 예를 들어, GOP 크기가 16인 경우 TID_{MAX} 는 4를 가지며 총 5개의 Temporal Level을 가지므로 총 5개의 임계 값 구간을 가지게 된다. 각 구간 별 임계값들은 다음 식 (15)와 같이 정의한다.

$$level N_{th} = min_{th} + \frac{(max_{th} - min_{th}) \times (TID_{MAX} + 1 - TID_N)}{TID_{MAX} + 1} \quad (15)$$

총 4개의 Temporal Level을 가지는 SVC 비트스트림의 경우 Temporal Level에 따라 차등적으로 패킷을 폐기하는 TS-AQM 알고리즘은 각 Temporal Level별 식 (16), (17), (18), (19)와 같은 패킷 폐기 확률을 가진다.

$$P_{Total}^{Level4} = \int_{min_{th}}^{level3_{th}} P(\omega) d\omega + \int_{level3_{th}}^{level2_{th}} \frac{n_{Level4}}{n_{Level4} + n_{Level3}} P(\omega) d\omega + \int_{level2_{th}}^{level1_{th}} \frac{n_{Level4}}{n_{Level4} + n_{Level3} + n_{Level2}} P(\omega) d\omega + \int_{level1_{th}}^{max_{th}} \frac{n_{Level4}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (16)$$

$$P_{Total}^{Level3} = \int_{level3_{th}}^{level2_{th}} \frac{n_{Level3}}{n_{Level4} + n_{Level3}} P(\omega) d\omega + \int_{level2_{th}}^{level1_{th}} \frac{n_{Level3}}{n_{Level4} + n_{Level3} + n_{Level2}} P(\omega) d\omega + \int_{level1_{th}}^{max_{th}} \frac{n_{Level3}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (17)$$

$$P_{Total}^{Level2} = \int_{level2_{th}}^{level1_{th}} \frac{n_{Level2}}{n_{Level4} + n_{Level3} + n_{Level2}} P(\omega) d\omega + \int_{level1_{th}}^{max_{th}} \frac{n_{Level2}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (18)$$

$$P_{Total}^{Level1} = \int_{level1_{th}}^{max_{th}} \frac{n_{Level1}}{n_{Level4} + n_{Level3} + n_{Level2} + n_{Level1}} P(\omega) d\omega \quad (19)$$

TS-AQM은 식 (16), (17), (18), (19)와 같이 TID가 낮은 패킷에게는 낮은 패킷 폐기 확률을 TID가 높은 패킷에게는 높은 폐기 확률을 제공하여 혼잡상황 발생 시 SVC 부호화 미디어 특성에 맞게 패킷을 폐기한다. 따라서, GOP내 각 프레임의 복호화 성공을 위해 상대적으로 중요도 및 우선 순위가 낮은 패킷부터 폐기된다. 사용자 체감 품질(QoE: Quality of experience) 측면에

서 복호화 실패 프레임 수를 줄여 시간에 따라 화질의 변화를 최소화 시켜 품질을 개선할 수 있다. 그리고, 수신 패킷의 프레임들에 대하여 전반적으로 복호화 성공율을 높일 수 있다. 이를 통해 서비스 이용자에게는 미디어 재생 시 좋은 품질을 제공하고, 나아가 종단간 측면에서의 네트워크 효율 (Efficiency)를 개선할 수 있다.

추가적으로 TS-AQM을 사용하는 코어(Core) 네트워크 내 라우터는 서버에 혼잡제어 기능을 가지고 있지 않은 스트리밍 서비스에 대하여 차등적인 Temporal Level 패킷 폐기를 통해 네트워크 적응적인 스트리밍 전송률 제어(Rate Control) 기능을 제공할 수 있다. 그리고, 혼잡제어 기능(Congestion Control)을 가지고 있는 스트리밍 서버에게는 혼잡상황이 발생하기 이전에 혼잡에 대비 할 수 있는 정보를 제공할 수 있다. 이는 사전에 라우터 큐에 오버플로우(Overflow)가 발생하기 전 혼잡정도에 따라 Temporal Level 단위로 폐기하고, 스트리밍 송신측에서 Temporal Level 별 손실 패킷의 비율을 인지하여 네트워크 상황에 보다 친화적이고 적응적인 전송률 제어를 수행하게 할 수 있다.

4. 성능 시험 및 평가

본 장에서는 SVC 부호화 기술의 특성을 파악하여 프레임간 의존성이 낮은 프레임의 패킷에 대하여 차등적으로 폐기하는 TS-AQM 알고리즘의 성능을 평가하기 위해 아래와 같은 사항의 성능을 ns(Network Simulator)를 이용하여 시험 및 평가하였다[10].

- i. 혼잡상황에서 프레임 복호화 성공을 비교
- ii. 이웃 플로우와의 친화성(Friendliness), 패킷 유실, 지연 성능 비교

4.1 혼잡상황에서 프레임 복호화 성공을 비교

본 논문에서 제안한 TS-AQM 알고리즘의 프레임 복호화 성공율을 시험하기 위해 그림 10과 같은 네트워크 시뮬레이션 환경을 설정하였다. 혼잡상황이 발생하는 라우터 R1에 TS-AQM 알고리즘과 RED 알고리즘, Drop Tail 방법을 적용하여 비교 시험하였다. SVC 스트리밍 Session은 RTP/RTCP로 구성하였고 임의의 혼잡상황을 유발시키기 위해 배경트래픽으로 Expoo Traffic 특성을 가지는 UDP Session 설정하여 아래와 같은 시나리오로 시험하였다[11].

- i. 첫 번째 시험은 4개의 다른 Quality Layer를 가지는 영상에 대해서 혼잡상황에 따른 복호화 성공율을 계산하였다. 혼잡상황을 발생시키기 위해 2Mbps의 병목구간에 Quality Layer별로 배경트래픽을 각 1.6, 1.5, 1.4, 1.3 Mbps로 전송하였다.
- ii. 두 번째 시험은 SVC 스트리밍 Session에 대해서 혼잡정도를 0%~50%까지 비례적으로 조정하며

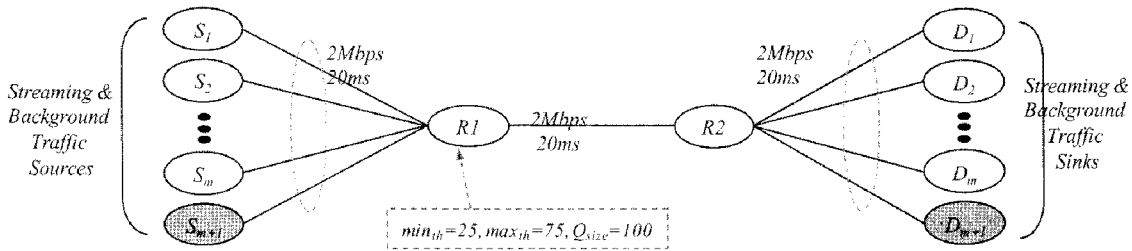


그림 10 네트워크 시뮬레이션 환경

배경 트래픽을 전송하는 환경에서의 복호화 성공율을 계산하였다.

시험을 위해 "SOCCER_352x288_30_orig_02_yuv" 영상을 JSVM을 이용하여 Layer 20(595kbps), Layer 33(814kbps), Layer 34(956kbps), Layer 35(1084kbps)으로 Quality Layer를 가지는 SVC 부호화 영상을 구성하였으며, 부호화된 영상에 대한 특성을 이용 기반으로 ns 환경에서 시험하였다[12]. 그리고, 각 시나리오 별로 시뮬레이션은 총 200초 동안 진행하였으며 30~140초 구간에서 약 110초 동안 혼잡상황을 발생시켰다.

첫 번째 시나리오는 4개의 Quality Layer 별로 임의의 혼잡상황을 발생 시켰을 때 각 큐 관리 알고리즘 별 프레임 복호화 성공율을 계산하였다. 일반적인 큐 관리 방법인 Drop Tail과 대표적인 큐 관리 기법인 RED 알고리즘, 본 논문에서 제안한 TS-AQM 알고리즘을 사용할 때의 각 프레임별 복호화 성공율은 그림 11과 같다.

RED 알고리즘을 사용할 경우 그림 11과 같이 혼잡발생시 임의의 패킷에 대하여 폐기하는 동작으로 인하여 2장에서 설명한 것과 같이 상호 의존성이 가장 높은 I Picture의 프레임에 대한 패킷이 가장 많이 폐기가 된다. 이로 인하여 가장 많은 수를 가지고 있는 B2 Picture의 프레임의 복호화가 되지 않아 전체적인 복호화 성공율이 나빠진다. 이로 인하여 Drop Tail 보다도 낮은 복호화 성공율을 보인다. 그리고, 고화질의 SVC 부호화 영상으로 갈수록 복호화 성공율의 저하가 보다 심각해 짐을 알 수 있다.

TS-AQM의 경우 혼잡정도에 따라 NAL Unit 헤더 내 TID 정보를 이용하여 Temporal Level별 차등적인 패킷 폐기 동작을 수행하여 복호화 의존성이 낮은 B2 Picture의 프레임부터 차등적으로 폐기하여 전체적인 복호화 성공율이 우수하다. 그리고, Drop Tail이나 RED와 다르게 고화질의 SVC 부호화 영상에 대해서도 극심한 저하를 보이지 않는다.

두 번째 시나리오는 Layer 35(1084kbps)의 SVC 부호화 영상에 대하여 병목 구간의 혼잡 정도를 부호화 영상의 0%~50% 비율로 혼잡정도를 조절할 경우 각 라우터 큐 관리 알고리즘에 대한 복호화 성공율을 계산

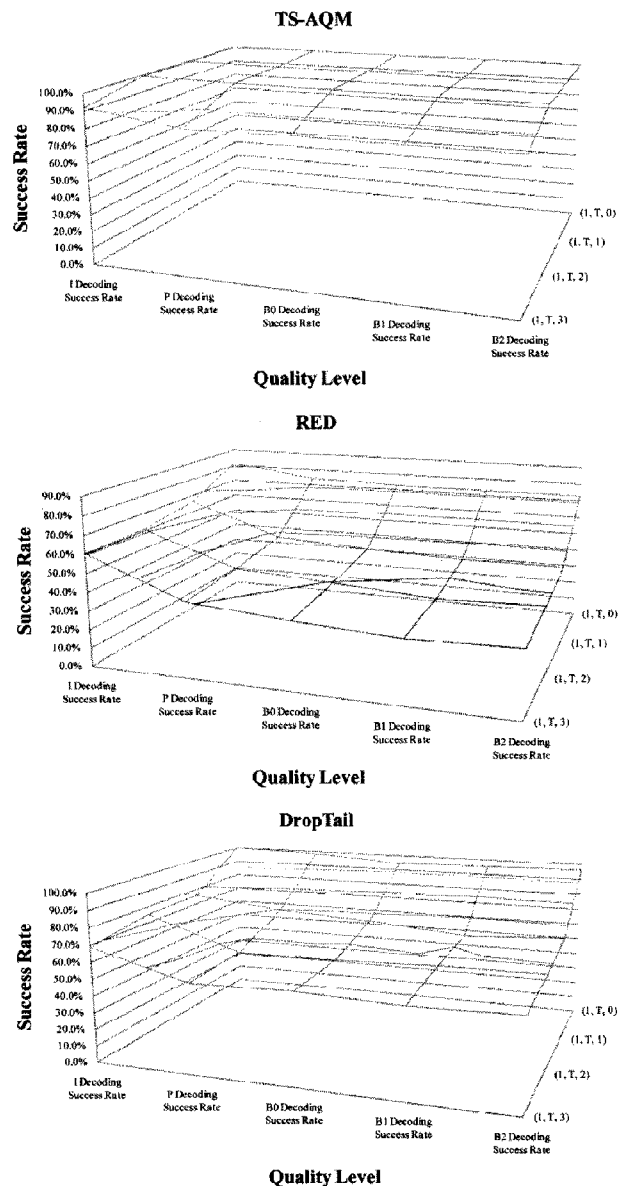


그림 11 Quality Level에 대한 각 프레임 별 복호화 성공율 하였다. 혼잡정도에 따른 각 프레임 별 복호화 성공율은 그림 12와 같다.

RED 알고리즘을 사용할 경우 다른 알고리즘 보다 이른(Early) 패킷 폐기동작 및 의존성이 높은 프레임 패킷 폐기로 전반적으로 프레임의 복호화 성공율이 빠르

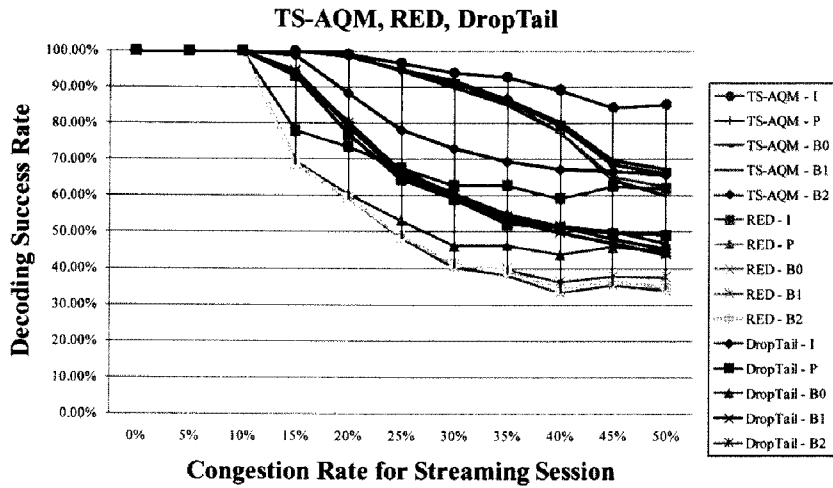


그림 12 혼잡정도에 대한 각 프레임 별 복호화 성공률

게 나빠짐을 알 수 있다. 그리고, Drop Tail의 경우도 임의의 구간부터는 오버플로우로 인하여 버스트(Burst) 하게 패킷이 폐기되어 급격하게 복호화 성공율을 낮아 짐을 알 수 있다.

TS-AQM의 경우 Temporal Level별 차등적인 패킷 폐기를 통해 의존성이 낮은 프레임부터 우선적으로 폐기되어 혼잡정도에 따라 완만하게 복호화 성공율이 낮아 짐을 알 수 있다. 따라서, SVC 기반의 스트리밍 서비스를 Drop Tail이나 RED와 같이 임의의 패킷 폐기를 수행하는 네트워크에 서비스시 잘 못된 패킷 폐기로 인하여 비디오 영상의 품질을 심각하게 저하 시킬 수 있다.

4.2 이웃 플로우와의 친화성(Friendliness), 패킷 유실, 지연 시간 성능 비교

본 논문에서 제안한 TS-AQM 알고리즘의 동작 특성을 파악하기 위해 SVC 스트리밍 플로우의 성능 비교 이외에 다른 플로우와의 친화성에 대하여 시험하였다. 그리고, 네트워크 효율 및 성능을 알아보기 위해 SVC 스트리밍 플로우의 패킷 유실을 및 단방향 지연(One-way Delay)시간에 대한 성능을 계산하였다. 네트워크 시뮬레이션 환경은 그림 10과 동일하며 시험 절차는 2개의 Reno TCP 플로우와 혼잡상황을 유발 시키는 1개의 가변 UDP Session, 그리고 SVC 스트리밍을 함께 전송하였다. 여기서 UDP Session의 전송률을 조절하여 혼잡정도의 증가에 따라 TS-AQM 알고리즘과 RED 알고리즘, Drop Tail 방법의 동작 특성을 시험하였다. 사용한 SVC 부호화 영상은 Layer 35(1084kbps)를 사용하였고 TCP 플로우 생성을 위해 2개의 FTP 서비스와 Expoo Traffic 특성의 UDP Session을 이용하였다. 그리고, 총 200초 동안 시뮬레이션하여 30~140초 구간에 걸쳐 혼잡상황을 유발시켜 측정하였다.

먼저, 큐 관리 동작으로 인하여 가장 많은 동작 특성을 보이는 TCP 플로우에 대하여 알아보기 위해 2개의 TCP Flow의 평균 전송률을 측정하였다. 각 큐 관리 알고리즘 별 두 TCP 플로우의 평균 전송률은 그림 13과 같다.

그림 13과 같이 TS-AQM 알고리즘 동작으로 다른 큐 알고리즘과 같이 이웃 플로우의 성능을 저하시키지 않음을 알 수 있다. 즉, SVC 스트리밍에 대한 차등적인 패킷 폐기는 SVC 스트리밍 Session에 영향을 주며, 다른 이웃 플로우들은 RED 알고리즘과 동일한 과정을 통

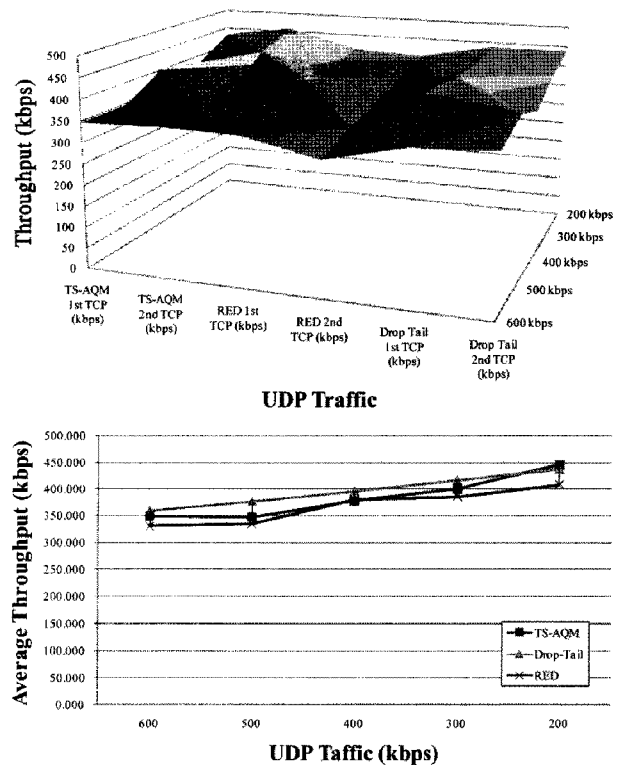


그림 13 두 TCP 플로우의 평균 전송률

해 처리되어 플로우간 불공정성이나 성능 저하를 유발시키지 않는다.

다음으로 동일 시험 환경에서 각 큐 관리 알고리즘으로 발생하는 SVC 스트리밍 Session의 평균 패킷 손실율을 비교하였다. RED 알고리즘과 Drop Tail의 경우 앞의 시험결과와 같이 낮은 복호화 성공율을 제공하며, 그림 14와 같이 많은 패킷 손실율을 가진다. 이는 Temporal Level별 차등적인 패킷 폐기를 하지 않으며, 혼잡 상황으로 임의의 패킷을 폐기함으로써 발생하는 문제이다. TS-AQM의 경우 혼잡 정도에 따라 차등적으로 패킷을 폐기하는 동작 이외에도 각 Temporal Level별 다른 패킷 폐기 확률을 제공한다. 이를 통해 I Picture의 프레임 임처럼 패킷 수가 많고 TID가 낮은 패킷들은 보호하여 RED 알고리즘과 Drop Tail 보다 낮은 패킷 손실율을 제공한다.

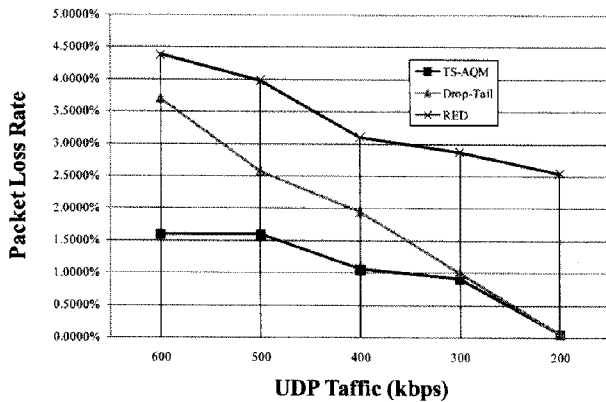


그림 14 SVC 스트리밍 Session의 평균 패킷 손실율

마지막으로 동일 시험환경에서 각 큐 관리 알고리즘별 SVC 스트리밍 Session의 단방향 지연 시간을 계산하였다. 단방향 지연 시간을 측정하기 위해 주기적으로 수신측에서 송신측으로 전송되는 RTCP RR(RTP Control Protocol Receiver Report)내 지연 시간 정보를 수집하였으며 수집된 결과는 그림 15와 같다.

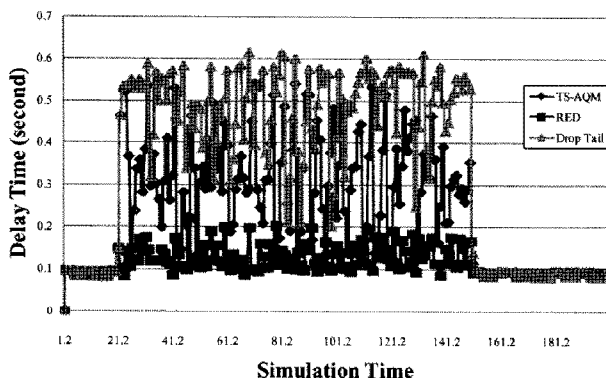


그림 15 SVC 스트리밍 Session의 단방향 지연시간

그림 15와 같이 RED 알고리즘은 동적 큐 관리 알고리즘 특징상 낮은 큐잉 지연시간을 제공하여 다른 알고리즘 보다 낮은 지연 시간인 평균 140ms를 가진다. 이와는 다르게 Drop Tail의 경우 오버플로우를 통해서 패킷 폐기 동작을 수행하기 때문에 최대 큐 크기에 가까운 큐잉 지연시간을 제공하여 평균 500ms의 높은 지연 시간을 가진다. 본 논문에서 제안한 TS-AQM의 경우 고품질 SVC 스트리밍에 대하여 차등적인 패킷을 폐기하여 RED 알고리즘 보다는 떨어지지만 기존 Drop Tail 방법 보다는 좋은 평균 300ms의 낮은 지연시간을 제공한다. 따라서, TS-AQM의 동적 큐 관리 특성으로 인하여 단방향 지연 시간이 개선됨을 알 수 있다.

본 논문에서 제안한 TS-AQM은 기존 Drop Tail 보다 개선된 성능을 제공하며 차등적인 패킷 폐기를 통해서 패킷 손실율과 우수한 프레임 복호화 성공율을 보장한다. 이를 통해 서비스 이용자에게는 미디어 재생 시 좋은 품질을 제공하며 나아가 End-to-End 측면에서의 네트워크 효율을 개선 할 수 있다. 그리고, 혼잡구간에 따라 Temporal Level별 폐기를 통해 SVC 스트리밍 전송을 제어(Rate Control)을 코어 망에서 부분적으로 제공할 수 있다.

5. 결론

고품질의 멀티미디어 서비스 및 서비스 품질(QoS)에 대한 요구를 충족하기 위해 네트워크 관점에서 동적 큐 관리 알고리즘과 같이 인터넷 성능을 개선하여 서비스 품질을 보장하는 연구와 종단 관점에서 미디어의 품질을 보장하기 위한 SVC(Scalable Video Coding) 부호화 기법에 대한 연구가 활발히 진행 중에 있다. 그러나, 기존 동적 큐 관리 알고리즘은 비디오 부호화 기술의 본질적인 특성에 대하여 고려하지 못하여 SVC 부호화 기법의 스트리밍 서비스의 품질을 저하시키는 문제점을 가지고 있다.

본 논문에서는 현재 혼잡제어 알고리즘의 문제점을 개선하기 위해 NAL(Network Abstract Layer)의 헤더 내 TID(Temporal_ID)를 통해 SVC 부호화 기술의 특성을 파악하여 프레임간 의존성이 낮은 프레임의 패킷에 대하여 차등적으로 폐기하는 TS-AQM(Temporal Scalability - Active Queue Management) 알고리즘을 제안하였다. 제안한 TS-AQM 알고리즘은 혼잡상황 시 차등적인 패킷 폐기를 통해 SVC 부호화 기법을 이용하는 스트리밍 서비스에 대하여 안정적인 비디오 복호화를 제공하며 이를 통해 화질의 변화를 최소화 시켜 미디어 서비스 품질을 보장하였다. 나아가 End-to-End 측면에서의 네트워크 효율(Efficiency)를 개선하였다.

향후 연구 과제로는 TS-AQM 알고리즘을 적용하여

다양한 네트워크 환경에서 시험 및 성능을 평가해야 할 것이다. 그리고, 네트워크 특성과 SVC 부호화 특성 이외에도 종단 노드의 전송 특성에 대하여 고품질의 멀티미디어 서비스와 품질을 보장하는 프레임 워크에 대한 연구가 필요하다.

참 고 문 헌

- [1] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet Protocol Television (IPTV): The Killer Application for the Next-Generation Internet," *IEEE Communications Magazine*, November 2007.
- [2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable H.264/MPEG4-AVC Extension," in *Proc. IEEE International Conference on Image Processing*, pp. 161-164, Atlanta, USA, October 2006.
- [3] S. Floyd and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control," *LBL Technical report*, February 1997.
- [4] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet," *IETF RFC (Informational) 2309*, April 1998.
- [5] J. Reichel, H. Schwarz, and M. Wien, "Working Draft 3 of ISO/IEC 14496-10: 2005/AMD1 Scalable Video Coding," *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 Document N7310*, July 2005.
- [6] J. Ohm, "Introduction to SVC Extension of Advanced Video Coding," *ISO/IEC JTC1/SC29/WG11 N7315*, July 2005.
- [7] D. Singer, M. Visharam, and Y. Wang, "Information Technology-Coding of audiovisual objects-Part 15: AVC file format, Amendment 2: File format support for scalable video coding," *ISO/IEC 14496-15/FPDAM 2(SVC file format)*, July 2007.
- [8] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transaction on Networking*, August 1993.
- [9] Hansen, A Table of Series and Products, Prentice Hall, Englewood Cliffs, NJ, 1975.
- [10] The Network Simulator ns-2, <http://www.isi.edu/nasnam/ns/>
- [11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *IETF, RFC 1889*, January 1996.
- [12] Joint Video Team (JVT) of ISO/IEC MPEG, ITU-T VCEG, "Joint scalable video model JSVM-9," *JVT-V202*, January 2007.



구 자 현

1999년 광운대학교 전자통신공학과 학사
 2001년 광운대학교 전자통신공학과 석사
 2003년~2008년 ㈜이노와이어리스 정보통신연구소 과장. 2001년~현재 광운대학교 전자통신공학과 박사과정. 관심분야는 인터넷 QoS, 유.무선 비디오 스트리밍, 광대역 무선인터넷 기술

정 광 수

정보과학회논문지 : 정보통신
 제 36 권 제 2 호 참조