

무선통신 환경에서 비디오 스트리밍의 버퍼 언더플로우와 오버플로우를 해결하기 위한 토큰버킷과 AMP 기법의 결합

이 현 노*, 김 동 회^o

Combination of Token Bucket and AMP Schemes to Solve Buffer Underflow and Overflow of Video Streaming in Wireless Communication

Hyun-no Lee*, Dong-hoi Kim^o

요 약

무선통신 네트워크에서 비디오 스트리밍 서비스를 제공 할 때 수신측 재생 버퍼에 있는 비디오 스트리밍 패킷 정보량은 네트워크의 상태에 따라 가변한다. 패킷 정보량이 특정 버퍼량보다 작아지면 버퍼 언더플로우(buffer underflow)문제가 발생하게 된다. 반대로 패킷 정보량이 특정 버퍼량보다 많아지면 버퍼 오버플로우(buffer overflow)문제가 발생한다. 목적지 수신 버퍼에서 비디오 스트리밍을 재생 할 때 이러한 버퍼 언더플로우와 오버플로우 문제는 영상의 멈춤이나 프레임 손실로 인한 스킵 현상을 유발하여 재생의 불연속성을 발생시킨다. 따라서 본 논문에서는 무선통신 환경에서 비디오 스트리밍의 버퍼 언더플로우와 오버플로우 문제를 해결하기 위하여 군집 트래픽을 제어하기 위한 토큰 버킷(Token Bucket)기법과 수신측 재생 속도를 가변적으로 변화시키는 AMP(Adaptive Media Playout)기법을 결합하여 그 효과를 비교 분석하였다. 모의실험을 통하여 2가지 방법의 결합이 언더플로우와 오버플로우 발생 회수, 언더플로우와 오버플로우로 인해 발생된 재생의 멈춤 지속시간 및 제거 되는 프레임 수 및 PSNR(Peak Signal to Noise Ratio)측면에서 우수함을 확인 할 수 있었다.

Key Words : Video streaming, Adaptive Media Playout, Token Bucket, Underflow, Overflow, PSNR

ABSTRACT

In wireless communication network, the amount of packet data for the video streaming in the playout buffer of the receiver is changed with time according to network condition. If the amount of packet data is less than a specific buffer amount, the buffer underflow problem is generated. On the contrary, if the amount of packet data is more than a given buffer amount, the buffer overflow problem is occurred. When the playout of the video streaming is processed, these buffer underflow and overflow problems cause stop and skip phenomenons and then provide the discontinuity of playout. Therefore, to solve the buffer underflow and overflow problems of the video streaming in wireless communication network, This paper analyzes the combined effect of Token Bucket scheme,

* First Author : Dept. of Electrical and Electronic Engineering, College of Information Technology, Kangwon National University, lechn900115@naver.com, 학생회원

^o Corresponding Author : Dept. of Electrical and Electronic Engineering, College of Information Technology, Kangwon National University, donghk@kangwon.ac.kr, 종신회원

논문번호 : KICS2014-10-421, Received October 19, 2014; Revised June 29, 2015; Accepted July 9, 2015

which controls the bursty traffic, and AMP(Adaptive Media Playout) scheme, which adaptively changes the playout speed of receiver. Through simulation, we found that the combination of Token Bucket and AMP schemes provides the superiority in terms of the occurrence number of buffer underflow and overflow, the stop duration time and the number of removed frames generated by underflow and overflow, and PSNR.

1. 서 론

비디오 스트리밍 서비스의 중단 간 품질에 있어 중요하게 고려할 사항 중 하나는 수신측 비디오 재생에 있어서 연속성이 보장되어야 한다는 점이다. 특히 무선통신 환경에서는 혼잡으로 인해 재생의 불연속이 발생 할 수 있고 이러한 현상은 사용자 체감 품질을 저하시키는 원인이 된다. 수신측 재생 버퍼에 채워져 있는 비디오 패킷의 양은 시간에 따라 가변하는데 네트워크 혼잡으로 인해 패킷의 유입이 지연될 경우 버퍼의 고갈 현상이 일어 날 수 있다. 이러한 경우 버퍼 언더플로우가 발생하며 버퍼 내에는 재생하기에 충분한 양의 데이터가 없기 때문에 버퍼링으로 인하여 영상의 멈춤 현상이 생길 수 있다. 반대로 균질적인 트래픽이 그대로 재생 버퍼에 유입될 경우 버퍼의 최대량을 넘어 패킷이 손실되는 버퍼 오버플로우가 발생할 수 있다. 손실된 패킷으로 인하여 특정 프레임과 그에 중속성을 가지는 상위 프레임들이 정상적으로 복호화 되지 못하므로 화질의 열화 및 영상의 스킵 현상이 발생 할 수 있다^[1,2]. 이러한 재생의 연속성을 방해하는 현상들을 해결하고 재생 품질을 보장하기 위해 여러 가지 연구가 진행되어 왔다. 그 중 하나로 무선 네트워크에서 교차 계층에 기반한 네트워크 적응형 비디오 스트리밍 기법에 대해 제안하는 연구가 있다. 비디오 전송자는 MAC과 물리계층의 상태를 측정하여 시간적으로 변하는 무선랜의 채널 상태를 예측한다. 예측된 채널 상태 정보는 네트워크 적응형 비디오 전송을 제어하기 위해 사용된다^[3]. 그리고 무선 채널 상태를 인지하여 비디오의 전송률과 품질을 조절하는 네트워크 기반의 품질 적응 스트리밍 기법인 Mobile Adaptive Rate Control(MARC)기법 또한 연구되었다^[4]. 또 다른 연구로는 사용자 체감 품질을 만족시키기 위하여 최소 재생 버퍼 레벨을 보존하여 가변적인 무선 네트워크 환경에서도 재생의 끊김 및 급격한 품질변화를 방지하는 비트율 적응 알고리즘도 제시되었다^[5]. 또한 단지 네트워크 지터에 대해 보상하기 위해 재생 지연이나 버퍼 사이즈를 동적으로 조절하는 지터 버퍼 컨트롤 메커니즘이 있으며 이 경우에 오디오 애플리케이션에 대해서는 효과적이다^[6]. 그

러나 실시간 비디오 스트리밍의 품질을 향상시키기에 충분하지 않다. 콘텐츠 인지형 AMP 기법의 경우 비디오 장면의 내용에 기초하여 재생 속도를 변경하는 방식이다. 예를 들어, 비디오 장면 중 움직임이 적은 장면일 경우에 재생 속도를 감소시켜 언더플로우의 발생을 줄이는 효과를 가지게 된다^[7]. 하지만 이러한 메커니즘의 경우 매우 높은 복잡성을 가지며 움직임이 많은 스포츠나 액션 영화 같은 경우에는 어려움이 있다. 그리고 본 논문에서 사용한 일반적인 AMP(Adaptive Media Playout)기법이 있다. AMP 기법은 버퍼에 있는 패킷 또는 버퍼량의 변동 횟수에 따라 동적으로 재생 속도를 조절하게 된다^{[8][9]}. 버퍼에 임계값을 설정하고 현재 측정된 버퍼량이 특정 임계값에 도달했을 경우 재생 속도를 증가 또는 감소시켜 버퍼의 고갈 및 포화 상태를 방지하는 기법이다. 즉, 버퍼가 포화상태에 가까워져 버퍼 오버플로우의 위험이 생기면 재생 속도를 증가시켜 단위 시간동안 더 많은 데이터를 재생하고, 버퍼의 고갈상태에 가까워져 언더플로우의 위험이 생기면 재생 속도를 감소시켜 버퍼의 안정 상태를 유지한다. 또한 본 논문에서는 수신측 재생 버퍼로 불규칙하게 들어오는 패킷들을 제어하기 위해 토큰 버킷(Token Bucket)기법을 사용하게 된다. 토큰 버킷은 단위 시간동안 생성되는 토큰의 수와 버킷의 용량을 파라미터로 설정함으로써 토큰에 따라 버퍼로 유입되는 트래픽을 제어 할 수 있다. 이러한 토큰 버킷 기법은 흐름제어와 데이터 트래픽의 성형을 위한 통신시스템에 널리 적용되고 있다. 비디오 서비스에서 품질을 높이기 위해 토큰버킷을 이용한 연구들도 있었다. 낮은 대역폭을 가지는 네트워크를 통해 비디오를 전송하기 위해 토큰버킷을 이용한 실시간 비트율 제어 기법이 있었으며^[10] 멀티미디어 서비스에서 서비스 품질을 보장하기 위해 토큰버킷을 이용한 대역폭 할당 방식이 있었다^[11]. 본 논문에서는 AMP 기법과 토큰버킷 기법을 결합하는 방식을 제안한다. 무선 통신 환경에서 트래픽의 혼잡으로 인해 불규칙 하게 들어오는 경우 그 정도가 심해지면 AMP만을 사용할 경우 재생속도 변경에 있어서 매우 많은 변화를 줘야 하거나 재생속도 변경만으로는 버퍼의 언더플로우 및 오버플로우 현상을 해소하기 어려운 부

분이 있을 수 있다. 따라서 재생 버퍼로 불규칙하게 들어오는 비디오 데이터를 1차적으로 토큰버킷을 이용하여 어느 정도 규칙적으로 성형해 주게 되고 2차적으로 재생 버퍼 내에서 AMP 제어를 할 경우 비디오 스트리밍 서비스에서 우수한 성능을 나타낸다는 것을 시뮬레이션 성능 분석을 통하여 나타내고 있다. 본 논문의 II장에서는 제안하는 기법을 이용하여 언더플로우와 오버플로우를 해결하기 위한 시스템 모델에 대해 설명하며, III장에서는 시뮬레이션 결과 분석을 하고 마지막으로 IV장에서는 결론을 제시하였다.

II. 언더플로우와 오버플로우를 해결하기 위한 시스템 모델

본 논문에서는 재생 품질을 저하시키는 원인인 버퍼 언더플로우와 버퍼 오버플로우를 해결하기 위해 토큰버킷 기법과 AMP 기법을 사용하게 된다. Fig 1은 토큰버킷 기법과 AMP 기법을 결합한 수신기 구조를 나타내는 그림이다. 네트워크의 영향으로 인해 수신측으로 균집적인 데이터가 유입 되려고 할 때 토큰버킷 기법을 이용하여 제한된 토큰의 개수 내에서의 패킷 유입만을 허용하게 한다. Fig 1과 같이 먼저 불규칙하게 유입되는 패킷들은 임시 버퍼(Temporary buffer)에 저장되고 그 후 하나의 패킷에 대해 하나의 토큰을 소비하여 재생 버퍼(Playout buffer)로 패킷을 유입시키도록 한다. 추가적으로 재생버퍼의 오버플로우나 언더플로우를 방지하기 위해 AMP 기법을 적용하여 버퍼에 있는 패킷량에 따라 재생속도를 변경하게 된다.

좀 더 자세히 설명하자면 Fig 1에서 보듯이 수신측으로 들어오는 비디오 패킷은 네트워크의 혼잡 및 지연 지터에 의해 불규칙하게 들어오는 형태를 가정한다. 이러한 형태의 Data traffic pattern은 임시 버퍼에 먼저 저장된다. 이후 토큰버킷 기법을 적용하여 하나의 토큰 당 하나의 패킷만을 재생 버퍼로 유입시켜 재생 버퍼 내의 패킷의 양을 제어하게 된다. 버킷 안에 있는 토큰이 존재하지 않는 경우에는 더 이상 재생버퍼에서는 비디오 패킷을 받을 수 없다. 그리고 초당 생성되어 버킷에 추가되는 토큰은 버킷의 용량이 최대일 경우 더 이상 추가 될 수 없다. 초당 생성되는 토큰의 수를 r , 토큰이 저장되는 버킷의 최대 용량을 c 라고 할 때 임의의 시간간격 Δt 동안에 재생 버퍼로 들어올 수 있는 비디오 패킷의 최대 수(N_{max})와 토큰 버킷의 최대 평균 속도(AV_{max})는 식 (1)과 식 (2)로 표현할 수 있다.

$$N_{max} = r \times \Delta t + c \quad (1)$$

$$AV_{max} = \frac{N_{max}}{\Delta t} \quad (2)$$

식 (1)과 식(2)를 통하여 토큰 버킷을 사용함으로써 네트워크의 평균 패킷 속도를 제한할 수 있다는 것을 의미한다^[12]. 본 논문에서 제안하는 수신기의 구조에서 토큰버킷 기법을 사용하여 버퍼의 포화를 방지하기 위해서는 적절한 파라미터 값을 설정해 주어야 한다. 그러기 위해 수신단에서는 재생 버퍼의 최대 용량(B_{max}), 단위 시간당 재생량($L_{playout}$), 패킷 최대 크기(P_{txmax} bytes)를 고려하여 파라미터를 정해줘야 한다. 임시 버퍼에 충분한 패킷이 있고 재생 버퍼가 비어있다고 가정할 경우 토큰 버킷에 의해 유입될 수 있는 최대 패킷량은 토큰 생성에 따른 유입 가능한 최대 패킷량($P_{tokenmax}$ bytes)과 버킷 용량에 따른 유입 가능한 최대 패킷량($P_{bucketmax}$ bytes)의 합으로 나타 낼 수 있으며 각각 식 (3)과 식 (4)로 나타 낼 수 있다.

$$P_{tokenmax} = r \times P_{txmax} \quad (3)$$

$$P_{bucketmax} = c \times P_{txmax} \quad (4)$$

그리고 토큰 버킷에 의해 재생버퍼로 비디오 패킷이 들어온 후에 재생 버퍼에서는 일정 프레임이 재생된다. 버퍼의 포화상태를 방지하기 위해서는 식 (5)와

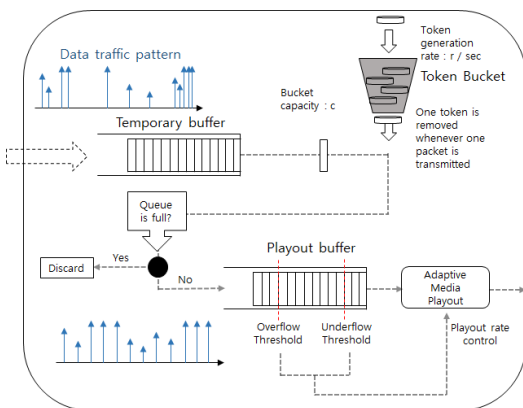


그림 1. 토큰버킷 기법과 AMP기법의 결합을 가진 비디오 스트리밍 수신기
Fig. 1. Video streaming receiver with the combination of Token Bucket and AMP algorithms

같이 토큰버킷으로 인해 유입될 수 있는 비디오 패킷의 양이 재생 버퍼의 최대값과 단위 시간당 재생량을 합한 값보다 작거나 같아야 한다.

$$P_{tokenmax} + P_{bucketmax} \leq B_{max} + L_{payout} \quad (5)$$

식 (5)에 따라 오버플로우를 해결하기 위한 파라미터를 설정하기 위해 r 값과 c 값을 구해야 한다. 따라서 식 (5)를 식 (6)과 같이 변경하여 r 값과 c 값을 더한 값을 구할 수 있다.

$$r + c \leq \frac{B_{max} + L_{payout}}{P_{txmax}} \quad (6)$$

토큰 버킷 파라메타인 r 값과 c 값을 더한 값을 구할 경우 가능한 값 선택 범위 내에서 가장 큰 값을 선택해야 한다. 그렇지 않은 경우에는 지나친 패킷 유입 제한으로 인해 오히려 버퍼 언더플로우가 발생 할 수 있기 때문이다. 따라서 식 (7)을 사용하여 $r+c$ 값을 정하게 된다.

$$r + c = \left\lfloor \frac{B_{max} + L_{payout}}{P_{txmax}} \right\rfloor \quad (7)$$

토큰 버킷을 거쳐 재생 버퍼로 들어온 패킷들에 대해서는 2차적으로 언더플로우 및 오버플로우를 방지하기 위해 AMP 기법을 적용한다. 재생 버퍼가 고갈되거나 포화되는 것을 방지하기 위해 언더플로우 임계값(Underflow Threshold)와 오버플로우 임계값(Overflow Threshold)을 정하고 버퍼의 패킷량을 측정하여 해당 임계값에 도달했을 때 프레임의 재생속도를 적응적으로 변화(Playout rate control)시키게 된다. AMP 기법에서 언더플로우일 경우에 변경하는 재생 속도($V_{underflow}$)는 식 (8)과 같이 표현되며 오버플로우일 경우에 변경하는 재생 속도($V_{overflow}$)는 식 (9)와 같이 표현된다.

$$V_{underflow} = V_{normal} - V_{normal} \times C_{payout} \quad (8)$$

$$V_{overflow} = V_{normal} + V_{normal} \times C_{payout} \quad (9)$$

여기에서 V_{normal} 은 정상 재생속도를 의미하며

C_{payout} 은 재생 속도 변경율을 의미한다. 예를 들어, 원래의 재생 속도가 30 fps이며 AMP 기법 적용 시 재생 속도 변경율을 $\pm 20\%$ 로 설정한 경우에 실시간으로 재생 버퍼의 패킷량을 측정하여 언더플로우 임계값 이하로 내려간 경우에는 언더플로우 발생 위험이 있으므로 재생 속도를 $30 \text{ fps} - (30 \text{ fps} \times 20/100) = 24 \text{ fps}$ 로 감소시켜 재생하여 버퍼 안에 머무는 패킷량을 증가시켜 언더플로우의 위험을 피하게 하였다. 이런 경우에 버퍼내의 비디오 패킷의 부족으로 인한 재생의 멈춤 현상을 방지 할 수 있을 것이다. 반대로 버퍼의 패킷량이 오버플로우 임계값 이상인 경우에는 오버플로우 발생 위험이 있으므로 재생 속도를 $30 \text{ fps} + (30 \text{ fps} \times 20/100) = 36 \text{ fps}$ 로 증가시켜 재생하여 버퍼에 머무는 비디오 패킷을 더 많이 꺼냄으로써 버퍼내의 패킷량을 감소시켜 포화 상태에 따른 오버플로우 위험을 피한다. 상기와 같은 조치에 따라 패킷의 손실을 막을 수 있으며 정상적인 복호화로 화질의 열화 및 재생의 스킵현상을 방지 할 수 있다.

III. 시뮬레이션 결과분석

3.1 시뮬레이션 기법

시뮬레이션은 NS-2(Network Simulator 2)와 JSVM(Joint Scalable Video Model)을 이용하여 진행하였다. 시뮬레이션 대상이 되는 비디오 시퀀스는 단일 계층으로 인코딩 된 영상을 사용하였으며 JSVM을 이용하여 인코딩 하였다. NS-2에서는 비디오 trace기반의 시뮬레이션으로 진행한다. 비디오 trace는 실제 비디오 스트림으로부터 중요한 특징들을 추출하여 텍스트 파일 형식으로 만든 후 이 데이터를 바탕으로 NS-2를 통해 전송하여 시뮬레이션 하는 기법이다^[1]. trace에 기록되어 있는 주요 정보로는 패킷의 전송간격 및 프레임의 길이 및 프레임의 부호화 유형 등 실제 비디오 스트림으로부터 추출되어 저장된 형태이다. 이러한 trace 파일을 바탕으로 전송단에서는 자동으로 해당하는 프레임에 대한 정보를 패킷의 형태로 생성하여 전송해 주고 수신단에서는 이 패킷들의 정보를 수신시간과 함께 다시 trace 파일의 형태로 기록하게 된다. 이 수신된 trace 파일을 가지고 재생의 불연속성 및 패킷 손실을 등을 계산 할 수 있으며 일련의 과정을 거쳐 JSVM을 이용하여 다시 디코딩을 하게 되면 비디오 시퀀스로 복원 할 수 있다. 이후 JSVM에서 제공하는 PSNR static틀을 이용하여 PSNR 측정을 할 수 있고 플레이어를 이용한 화질을 볼 수 있다. 이 모든 시뮬레이션 과정 들은 MyEvalSVC를 참조하여 진

행하였다^[13,14].

3.2 시뮬레이션 분석

시뮬레이션 대상이 된 비디오 파일은 YUV 형식으로 JSVM 인코더를 사용하였으며 Table 1과 같이 구성되어 있다^[15].

총 3개의 비디오 시퀀스를 사용하였으며 모두 300 frame으로 이루어져 있다. 또한 각 영상들은 공간적 및 화질적 확장성을 가지지 않은 단일 계층으로 인코딩된 영상이다. foreman_cif 영상과 crew_cif 영상은 352 x 288 크기의 해상도를 가지며 각각 영상 내 object들의 움직임은 다르다. 또한 704 x 288의 크기를 가지는 보다 고해상도 영상인 hall_2cif 영상을 가지고 시뮬레이션을 진행한다.

시뮬레이션 파라미터는 Table 2와 같다. 전파 모델은 Two-ray ground 모델을 사용하였으며 경쟁 트래픽을 생성하여 네트워크의 혼잡을 주었다. 토폴로지는 Fig 2와 같다. 먼저 각 트래픽 서버와 AP는 유선으로 연결되어있으며 유선 대역폭은 각 1 Mbps이며 무선 대역폭은 2 Mbps로 설정하였다. S2 노드는 CBR 트

표 1. 시뮬레이션을 위한 YUV 파일들
Table 1. YUV files for simulation

	foreman_cif	crew_cif	hall_2cif
Frame number	300	300	300
GOP size	16	16	16
Frame rate	30	30	30
Video resolution	CIF (352x288)	CIF (352x288)	2CIF (704x288)
Total video packet size	418334 byte	629038 byte	420175 byte

표 2. 시뮬레이션 파라미터들
Table 2. Simulation parameters

Used simulation tool	NS-2, JSVM		
Node number	sender 3, receiver 3, AP 1		
Propagation model	Two-Ray ground model		
Token generation rate (r) for Token Bucket	foreman_cif 4/0.1 sec	crew_cif 5/0.1 sec	hall_2cif 5/0.1 sec
Bucket capacity (c) for Token Bucket	18	19	17
Maximum packet size (Ptxmax)	1,500 Byte		
Buffer size (Bmax)	30,000 Byte		
Normal Playout speed (Vnormal)	30 fps (4183 Byte /0.1 sec)	30 fps (6290 Byte /0.1 sec)	30 fps (4202 Byte /0.1 sec)
Underflow threshold for AMP	30%		
Overflow threshold for AMP	70%		
Playout speed change rate for AMP (Cplayout)	±10%, ±20%, ±30%		

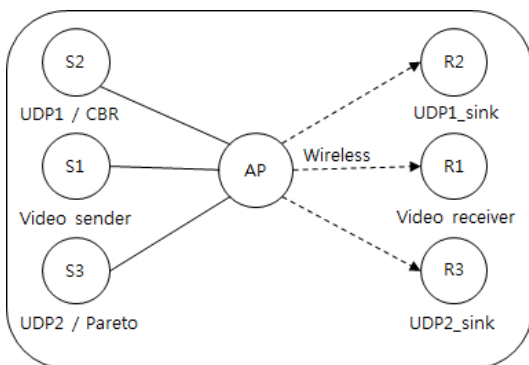


그림 2. 시뮬레이션 토폴로지
Fig. 2. Simulation topology

래픽을 생성하여 R2로 전송하고 S3 노드는 Pareto 트래픽을 생성하여 R3로 전송하게 된다. 이 때, 비디오 서버는 초당 30 frame의 정보를 수신단에게 전송하고 S2는 시뮬레이션 시간 내 특정 구간에서만 전송을 시도하고 종료하여 무선 대역폭을 차지하게 된다. S3는 비디오 트래픽 전송과 동시에 pareto 트래픽을 발생시킨다. pareto 트래픽은 트래픽을 발생시키는 On기간과 트래픽을 발생시키지 않는 Off기간이 pareto 분포에 따라 결정되는데 본 시뮬레이션에서 설정한 값은 burst_time이 0.5 Sec, idel_time이 1 Sec로 평균 0.5 초의 ON기간과 평균 1초 동안의 OFF기간을 가진다.^[1] 또한 AP의 Queue 사이즈를 충분히 크게 하여 경쟁 트래픽으로 인하여 혼잡이 발생할 경우 패킷이 AP Queue에 쌓여 R1이 받는 비디오 패킷량이 줄어드는 현상이 발생하며 경쟁 트래픽이 발생하지 않는 경우에는 AP에 쌓여있던 S1의 비디오 패킷이 많은 대역폭을 점유하여 균질적인 패킷을 전송하게 되는

환경을 만들었다. 전송되는 각 패킷들의 최대 사이즈는 1500 byte이며 전송된 비디오 패킷은 수신측 재생 버퍼에 저장되었다가 0.1초당 3 frame만큼 재생하게 된다. 토큰 버킷 파라미터 중 초당 생성되는 토큰의 수와 버킷이 저장할 수 있는 토큰의 최대 개수는 재생 버퍼의 크기(30000 byte)를 고려하여 식 (7)을 사용하여 설정하였다. 식 (7)을 사용하면 r과 c를 더한 값을 정할 수 있는데 송신단에서 생성되어 전송되는 비디오 데이터 패킷의 양은 영상마다 각각 다른 양상을 보이기 때문에 테스트 시뮬레이션을 통하여 적절한 초당 생성되는 토큰의 개수 r을 정하였으며 Table 3과 같다. AMP 기법 적용 시 적응적으로 변경하는 재생 속도 변경을 파라미터는 총 세 가지로 $\pm 10\%$, $\pm 20\%$, $\pm 30\%$ 로 변경해 가며 진행하였다.

또한 성능분석 시 본 논문에서는 토큰 버킷 기법과 AMP 기법의 결합을 통한 비디오 스트리밍의 성능 분석을 Table 4와 같이 총 4가지의 시나리오를 통하여 진행하였다. Case 1의 경우 아무 기법도 사용하지 않은 경우이며 Case 2의 경우 토큰버킷 기법만 적용한 경우, Case 3은 토큰 버킷 기법은 사용하지 않고 AMP 기법만 적용한 경우, Case 4의 경우 본 논문에서 제안하는 토큰 버킷 기법과 AMP 기법을 모두 적용한 경우이다.

Fig 3은 비디오 중 foreman_cif의 비디오 데이터를 송신했을 때 수신측으로 유입되는 비디오 스트리밍 패킷의 패턴에 대해 나타낸 그림이다. 특히 버퍼의 오버플로우가 일어나는 구간 중 1초 동안(9.5 Sec ~ 10.5 Sec)의 구간에 대해 토큰버킷을 사용하지 않았을 경우(Case 1, Case 3)와 토큰버킷을 사용하여 유입되는 트래픽에 대해 성형을 거친 경우(Case 2, Case 4)에

표 3. 토큰버킷 파라미터들에 따른 언더플로우와 오버플로우 발생 횟수

Table 3. Occurrence number of buffer underflow and overflow according to Token Bucket parameters

	Summation of r and c	r/c	Number of underflows	Number of overflows
foreman_cif	22	4/18	14	0
		5/17	10	11
		6/16	10	31
crew_cif	24	4/20	40	0
		5/19	14	0
		6/18	10	10
hall_2cif	22	4/18	13	0
		5/17	4	0
		6/16	4	7

표 4. 비디오 스트리밍 시뮬레이션을 위한 네 가지 시나리오 Table 4. Four scenarios for video streaming simulation

Scenario	Token Bucket Algorithm	Adaptive Media Playlist
Case 1	X	X
Case 2	O	X
Case 3	X	O
Case 4	O	O

대해 나타내었다. Fig 3에서 토큰버킷을 사용하지 않은 경우에는 비교적 패킷들이 불규칙하게 유입되며 군집 트래픽 또한 그대로 버퍼로 유입되는 것을 볼 수 있다. 다른 비디오 시퀀스인 crew_cif와 hall_2cif의 경우도 Fig 3과 마찬가지로 토큰 버킷 기법을 사용하지 않을 때 패킷들이 비교적 일정한 간격을 두고 수신측 재생 버퍼로 들어오는 결과를 나타낸다.

Fig 4는 각각의 시뮬레이션 대상 비디오 데이터에 대해 Case 1과 Case 2에서 언더플로우와 오버플로우의 발생 횟수를 비교한 그래프이다. foreman_cif 영상의 경우를 보면 Case 1의 경우에는 언더플로우가 10

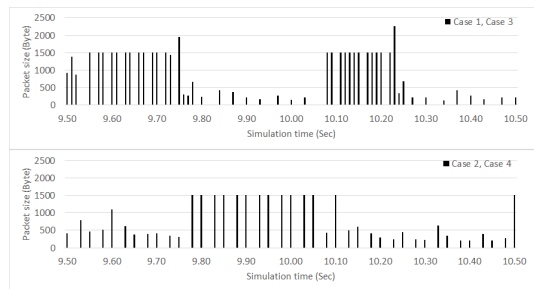


그림 3. 수신측 버퍼로 들어오는 비디오 패킷의 패턴 Fig. 3. Pattern of video packet size coming in the receive buffer

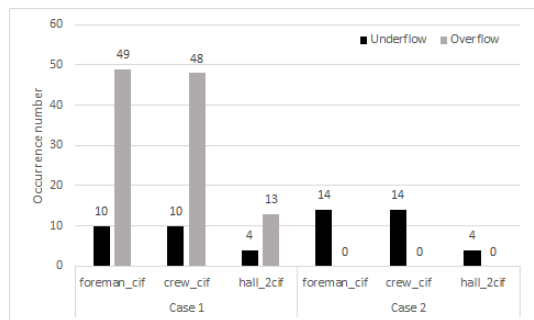


그림 4. Case 1과 Case 2에서 언더플로우 및 오버플로우 발생 횟수 Fig. 4. Occurrence number of underflow and overflow in Case 1 and Case 2

회, 오버플로우는 49회가 나타나며 토큰버킷을 사용한 Case 2의 경우에는 언더플로우는 14회, 오버플로우는 0회로 나타난 것을 알 수 있다. 전체적으로 토큰기법을 사용한 Case 2의 경우 모든 비디오 시퀀스들에 대해 오버플로우를 해소하는 것을 확인할 수 있으며 토큰을 이용한 패킷 유입의 제한으로 인해 foreman_cif 영상과 crew_cif 영상에서는 언더플로우 횟수가 약간 증가한 것을 볼 수 있다. 이와 같은 경우에 영상의 재생에 있어서 프레임의 손실은 일어나지 않고 300 frame모두 정상적으로 재생이 되며 영상의 멈춤 현상이 Case 1보다는 좀 더 지속될 수 있다는 것을 알 수 있다.

Fig 5는 Case 3과 Case 4에서 언더플로우와 오버플로우의 발생 횟수를 비교한 그림이다. 그림에서 아래쪽 10%, 20%, 30%는 AMP 동작 시 재생 속도를 원래 재생 속도보다 얼마만큼 더 빠르거나 느리게 재생했는지에 대한 재생 속도 변경율을 의미한다. Case 3은 토큰버킷 기법을 적용하지 않은 상태에서 AMP 기법만 적용한 경우이다. Case 3에서 보던 대체적으로 재생 속도 변경율을 10%정도만 조절했을 경우엔 높은 언더플로우 횟수와 오버플로우 발생 횟수를 나타내고 있고 재생 속도 변경율을 20%와 30%로 각각 더 높은 변경율을 적용했을 때 10%로 설정했을 경우보다 좀 더 좋은 성능을 내는 것을 확인할 수 있다. 즉 AMP 기법을 적용할 경우 오버플로우와 언더플로우 횟수를 모두 낮출 수 있다는 것을 알 수 있다. 하지만 토큰버킷만 사용한 Case 2와 비교해 보면 Case 3의 경우에는 언더플로우나 오버플로우 중 어느 한 쪽도 완벽하게 해결하지 못하여 영상의 멈춤 및 스킵현상 모두가 일어날 수 있다는 것을 알 수 있다. 반면 Case 4의 경우에는 오버플로우와 언더플로우가 매우

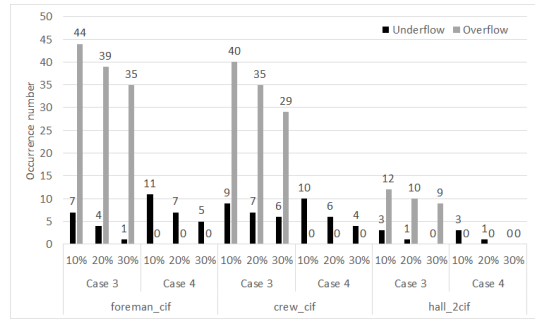


그림 5. Case 3과 Case 4에서 언더플로우 및 오버플로우 발생 횟수
Fig. 5. Occurrence number of underflow and overflow in Case 3 and Case 4

감소한 결과를 보이고 있다. 토큰기법을 적용한 상태에서 AMP 기법을 적용하였기 때문에 Case 2에서 봤던 결과와 같이 오버플로우는 모든 경우에 0으로 나타나며 언더플로우의 경우에는 AMP 기법까지 적용시킨 Case 4에서 더 낮은 수치를 보이고 있다. Case 4도 마찬가지로 재생속도 변경율을 높일수록 더 좋은 결과를 보이는 것을 알 수 있다. Table 5는 시뮬레이션 대상이 된 각 비디오에 대해 Case 1부터 Case 4까지 성능 분석을 했을 때 각 현상들에 따른 영상의 멈춤 현상 지속시간 및 제거되는 프레임의 수를 나타낸 표이다. 언더플로우 현상이 발생 할 경우 퍼퍼링으로 인해 영상의 멈춤 현상이 생기며 그 현상의 지속시간을 msec 단위로 표시하였다. 그리고 오버플로우 발생 시 패킷이 손실되는데 손실된 패킷에 해당하는 프레임이 정상적으로 복호화 되지 않게 된다. 또한 정상적으로 복호화 되지 않아 제거된 프레임에 종속성을 가지는 상위 프레임들 또한 제거된다. 따라서 위의 나타난 수치는 손실된 패킷들로 인해 최종적으로 제거된

표 5. 모든 경우들에서 언더플로우와 오버플로우로 인해 발생된 재생의 멈춤 지속시간 및 제거되는 프레임 수의 비교
Table 5. Analysis of the stop duration time and the number of removed frames generated by underflow and overflow in all

	foreman_cif		crew_cif		hall_2cif	
	Stop duration time (msec)	Number of Removed frames	Stop duration time (msec)	Number of Removed frames	Stop duration time (msec)	Number of Removed frames
Case 1	1000	79	1000	67	400	31
Case 2	1400	0	1400	0	400	0
Case 3	10%	700	900	63	300	31
	20%	400			100	
	30%	100			0	
Case 4	10%	1100	1000	0	300	0
	20%	700			100	
	30%	500			0	

모든 프레임의 수에 대해 표기하였다. Case 1과 Case 3의 경우 토큰버킷이 적용되지 않아 오버플로우가 발생하여 패킷들이 손실된다. 또한 Case 3의 경우에는 재생속도 변경율을 높일수록 오버플로우의 발생회수는 줄어드는데 위의 결과를 보면 제거된 프레임의 거의 다 비슷한 결과를 보이고 있는 것을 알 수 있다. 그 이유는 재생속도 변경율을 높임으로써 전체적으로 손실된 패킷의 개수는 줄었지만 여전히 특정 I 프레임이 손실되어 그에 종속성을 가지는 다른 프레임들이 제거되기 때문이다. 토큰버킷을 사용한 Case 2와 Case 4의 경우 오버플로우가 발생하지 않아 제거되는 프레임이 없다. 제안하는 기법인 Case 4의 경우 30%의 재생속도 변경율을 적용하였을 때 crew_cif와 hall_2cif에서는 가장 좋은 환경을 보인다. foreman_cif의 경우 Case 4에서 영상의 멈춤 현상이 0.5초 발생하며 Case 3의 30%에서보다 멈춤 현상이 약 0.4 초 더 오래 지속되는 것을 볼 수 있다. 하지만 Case 3의 경우에는 프레임이 손실되어 모든 영상을 볼 수 없는 문제가 발생하기 때문에 차라리 영상의 손실이 일어나지 않는 Case 4의 경우가 더 좋은 환경이라고 할 수 있다.

Fig 6과 Fig 7은 Case 3과 Case 4에서 전송된 비디오에 대해 PSNR을 측정한 그래프이다. 오버플로우로 인하여 정상적으로 복호화 되지 않은 프레임에 대해서는 이전 프레임 정보를 복사하여 총 300 frame에 대해 모두 PSNR을 구하였다. 따라서 손실된 프레임은 낮은 PSNR 수치를 보이게 된다. Fig 6에서 보듯이 AMP 기법만 사용한 Case 3의 경우 군집적인 트래픽이 그대로 유입된 상태에서 AMP만을 적용했기 때문에 오버플로우를 모두 해결하지 못하여 패킷의 손실이 일어나고 모든 영상에 대해 약 200~300 사이 특정 프레임들이 손실되는 것을 확인 할 수 있다. 반면 제안한 기법인 Case 4의 경우에는 토큰버킷을 사용하여 패킷의 손실을 방지했기 때문에 Fig 7과 같이 모든

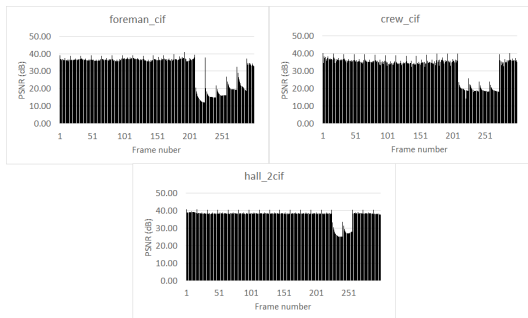


그림 6. Case 3에서 수신된 영상의 PSNR
Fig. 6. PSNR of image received in Case 3

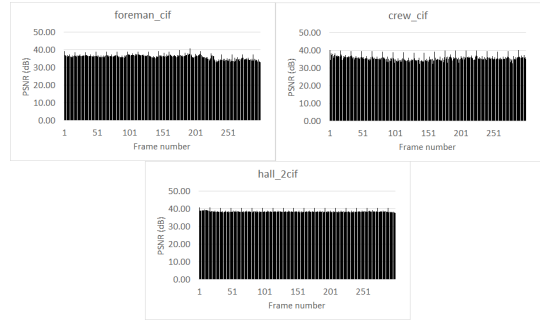


그림 7. Case 4에서 수신된 영상의 PSNR
Fig. 7. PSNR of image received in Case 4

프레임에 대해 높은 PSNR 수치를 보이는 것을 알 수 있다. 즉 영상의 손실 없이 모든 프레임들을 볼 수 있다는 것이다.

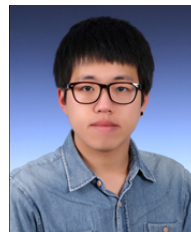
IV. 결론

무선 통신 환경에서 비디오 스트리밍의 버퍼 언더플로우와 오버플로우 문제를 해결하기 위한 여러 연구들이 있었으며 본 논문에서는 그 중 하나인 AMP 기법을 적용하고 추가로 토큰버킷 기법을 결합하여 비디오 스트리밍에 대한 성능 분석을 하였다. 불규칙적으로 들어오는 트래픽들에 대해 제어하고 오버플로우를 방지하기 위하여 토큰버킷 기법을 1차적으로 적용하였다. 재생 버퍼에는 2차적으로 언더플로우와 오버플로우 위험을 모두 낮추기 위해 AMP 기법을 적용시켰으며 버퍼의 패킷량에 따른 비디오 패킷의 포화나 고갈을 방지하기 위해 재생 버퍼량을 기준으로 두 개의 임계치를 적용하고 AMP 동작 시 재생 속도 변경율을 여러 가지로 바꾸어 성능분석을 수행하였다. 결과적으로 AMP 기법과 토큰버킷 기법을 함께 사용했을 경우에는 다른 경우들 보다 우수한 성능을 나타내었으며 30%의 재생 속도 변경율을 적용하였을 때에는 재생의 멈춤이나 스킵을 대부분 방지 할 수 있다는 것을 알 수 있었다. 하지만 재생 속도 변경율이 크면 클수록 오버플로우나 언더플로우에 대한 효과는 크지만 그만큼 사용자가 체감하는 영상의 변화율이 커지게 되고 그로 인해 사용자 품질을 저하시킬 수 있다. 따라서 향후 연구로는 비디오 스트리밍에서 연속성을 보장하면서 사용자 체감 품질을 고려할 수 있는 기법을 연구해야 한다.

References

- [1] Y.-M. Jang, *The application of NS-2 Network Simulation*, Hongrung Publishing Company, 2008.
- [2] J. Jo and J. Kim, "Synchronized one-to-many media streaming employing server-client coordinated adaptive playout control," *J. KICS*, vol. 28, no. 5c, pp. 493-505, May 2003.
- [3] S. H. Park, "Cross-layered network-adaptive video streaming over wireless networks," GIST doctorate thesis, 2008.
- [4] J. Koo and K. Chung, "Adaptive rate control of improving the QoE of streaming service in broadband wireless network," *J. KICS*, vol. 35, no. 2, pp. 334-344, Feb. 2010.
- [5] D. Suh, I. Jang, and S. Pack, "A video bitrate adaptation algorithm for DASH-based multimedia streaming services to enhance user QoE," *J. KICS*, vol. 39B, no. 6, pp. 341-349, Jun. 2014.
- [6] B. Steyaert, K. Laevens, De D. Vleeschauwer, and H. Bruneel, "Analysis and design of a playout buffer for VBR streaming video," *Ann. Oper. Res.*, vol. 162, no. 1, pp. 159-169, Sept. 2008.
- [7] H.-C. Chuang, C. Y. Huang, and T. Chiang, "Content-aware adaptive media playout controls for wireless video streaming," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1273-1283, 2007.
- [8] E. Steinbach, N. Farber, and B. Girod, "Adaptive playout for low latency video streaming," *Int. Conf. Image Process.*, vol. 1, pp. 962-965, Oct. 2001.
- [9] M. C. Yuang, T. L. Shih, Yu G. Chen, and C. L. Shen, "Dynamic video playout smoothing method for multimedia applications," *IEEE Int. Conf. Commun.*, vol. 3, pp. 1365-1369, Dallas, TX, Jun. 1996.
- [10] S.-H. Park and W.-G. Oh, "Real-time rate control with token bucket for low bit rate video," *J. KIICE*, vol. 10, no. 12, pp. 2315-2320, Dec. 2006.
- [11] R. Kim and H. Ryu, "The multi-queue bandwidth regulation method for multimedia QoS," *KIISE Fall Conf.*, vol. 2, pp. 469-471, Korea, Nov. 2005.
- [12] Behrouz A. Forouzan, *Data Communications and Networking*, 5th Ed., McGraw-Hill Education, 2012.
- [13] C.-H. Ke, "myEvalSVC : An integrated simulation framework for evaluation of H.264./SVC transmission," *KSII Trans. Internet and Inf. Syst.*, vol. 6, no. 1, pp. 378- 393, Jan. 2012.
- [14] C.-H. Ke, *How to do H.264 SVC transmission simulations(2014)*, Retrieved Aug, 2014, from <http://csie.nqu.edu.tw/smallko/ns2/svc.htm>
- [15] JSVM Software Manual(2006)

이 현 노 (Hyun-no Lee)



2014년 2월 : 강원대학교 IT대학 전기전자공학부 공학사
 2014년 3월~현재 : 강원대학교 IT대학 전기전자공학부 석사과정
 <관심분야> 차세대 이동통신 및 차세대 무선 네트워크

김 동 회 (Dong-hoi Kim)

한국통신학회 논문지 (통신이론 및 시스템) 제38A권 제4호 참조