

유무선 네트워크에서 프레임 수신 순서 재할당 방법을 사용한 동영상 재생

강동진*, 김동회**

요약

NS-2를 사용하는 시뮬레이션에서 동영상 서비스가 수행될 때 프레임이 수신되는 순서대로 동영상이 재생된다. 기존의 수신된 프레임 순서 기반 동영상 재생 방법에서는 수신기와 송신기의 프레임 순서들이 일치하지 않아서 수신버퍼는 송신 버퍼의 프레임들 사이의 패킷들이 규칙적인 크기를 유지하는 효과를 가지지 못하고 수신버퍼에서의 패킷 군집 또는 성김 현상이 일부 프레임의 순서들의 예상치 못한 뒤바뀜에 기인하는 불규칙한 패킷 크기에 의해서 발생할 수 있다. 수신버퍼의 프레임들에서 발생한 이러한 패킷 군집 또는 성김 현상은 버퍼 오버플로우와 언더플로우의 발생확률을 증가시키게 된다. 이러한 문제점을 방지하기 위하여 제안하는 프레임 수신 순서 재할당 방법에서는 송신버퍼의 프레임 순서와 동일하게 수신버퍼의 프레임 순서를 재배치하는 별도의 재생버퍼를 추가하여 수신한 프레임들이 수신버퍼에서 일정한 패킷 크기를 유지하게 하였다. NS-2와 JSVM을 이용한 시뮬레이션을 통하여 기존 방법과 제안된 방법에서의 버퍼 오버플로우와 언더플로우의 발생 횟수 그리고 최대 신호 대 잡음비 성능을 비교하였다. 그 결과, 제안방법이 기존 방법보다 더 우수함을 확인할 수 있었다.

키워드 : 비디오 스트리밍, 프레임 수신 순서, 버퍼 오버플로우, 버퍼 언더플로우, 최대 신호 대 잡음비

Video Replay by Frame Receive Order Relocation Method in the Wire and Wireless Network

Dong-Jin Kang*, Dong-Hoi Kim**

Abstract

When video service is performed in simulation using NS-2(Network Simulation-2), the video replay is performed as the received frame order. In the existing video replay method based on the received frame order, as the frame orders of receiver and transmitter are different, the receiver buffer does not have the effect that the packets between the frames of transmitter buffer holds a regular size and packet dense and sparsity phenomenon in the receiver buffer is made by the irregular packet size due to the unpredictable reversed order of received partial frames. The above dense and sparsity phenomenon increases the probability of buffer overflow and underflow generation. To prevent these problems, the proposed frame receive order relocation method adds an extra replay buffer which rearranges the order of receive frame as the order of transmit frame, so it has the effect that the packets between the transmit frames keeps a regular size. Through the simulation using NS-2 and JSVM(Joint Scalable Video Model), the generation number of buffer overflow and underflow, and PSNR(Required Peak Signal to Noise Ratio) performance between the existing method and proposed method were compared. As a result, it was found that the proposed method would have better performance than the existing method.

Keywords : Video streaming, Frame receive order, Buffer overflow, Buffer underflow, PSNR

1. 서론

※ Corresponding Author : Dong-Hoi Kim

Received: March 04, 2016

Revised: June 13, 2016

Accepted: June 25, 2016

* Kangwon National University, Electrical and Electronic Engineering

Tel: +82-33-250-6349, Fax: +82-33-259-5676

email: donghk@kangwon.ac.kr

** Kangwon National University, Electrical and Electronic Engineering(Corresponding Author)

현재의 실시간 비디오 스트리밍 서비스 환경은 유비쿼터스 시대에 맞추어 다양한 네트워크 환경 및 장치간의 전송에 대한 필요성이 증가하고 있다[1]. 2015년 TDG Research 조사에 따르면 인터넷에서 차지하는 비디오 트래픽은 전체 소비자 트래픽의 약 51%에 달하고 있으며, 이 비율은 2016년부터는 대략 86%으로까지 증가할 것으로 전망하고 있다. 또한 시스코에서 실시한 연구조사 결과에 의하면 2014년 대비 2019년에는 비디오 트래픽은 연평균 23% 성장률을 가질 것을 예상하고 있으며 트래픽이 3배 증가할 것으로 전망 하고 있다[2][3]. 비디오 스트림에 따른 연구가 진행되면서 시간에 따라 변화하는 네트워크의 특성과 단말의 성능에 적합하게 비디오 스트림을 전송하기 위한 SVC 기반의 비디오 스트리밍 기법이 기존 연구에서 제안되었다[4]. 기존 방법을 사용하여 비디오 스트리밍 기법을 적용할 경우에는 수신버퍼에서 전송 프레임 간에 일정한 패킷 크기 유지 효과가 깨지고 일부 프레임에 큰 패킷들의 군집 또는 성김 현상이 발생한다. 이러한 패킷이 특정 버퍼의 패킷 크기보다 더 큰 경우에는 오버플로우가 발생하고, 재생을 해야 하는 버퍼의 패킷들이 부족한 경우에는 언더플로우가 발생하게 된다. 본 논문에서 제안하는 프레임 수신 순서 재할당 방법에서는 송신버퍼의 프레임 순서와 동일하게 수신버퍼의 프레임 순서를 재배치하는 별도의 재생버퍼를 추가함으로써 송신 단의 송신버퍼에서 프레임 간에 유지되었던 패킷 크기 분산 효과를 수신 단에서도 동일하게 유지하도록 하였다.

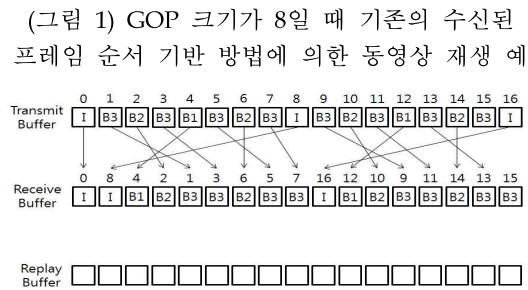
본 논문의 2장에서는 기존 프레임 수신 순서 기반 동영상 재생 방법이 소개되고 3장에서는 제안하는 프레임 수신 순서 재할당에 의한 동영상 재생이 서술되었다. 4장과 5장에서는 2장과 3장의 방법을 시뮬레이션하기 위한 시뮬레이션 환경과 결과를 보여주고 6장에서 본 논문의 결론에 대해 서술한다.

2. 기존의 수신된 프레임 순서 기반 방법

스케일러빌리티 종류에는 시간적 스케일러빌리티(temporal scalability), 공간적 스케일러빌리티(spatial scalability), 화질적(SNR) 스케일러빌리티, 그리고 복합 스케일러빌리티(combined scalability)가 있다. 기존 방법에서는 시간적 스케일러빌리티를 사용한다. H.264에서 이미 부호화된 모든 영상은 참조 영상으로써 다음에 오는 영상의 움직임 보상 예측에 이용될 수 있다.

Memory Management Control Operation(MMCO) 명령어는 복호화된 영상을 저장할 수 있는 Decoded Picture Buffer(DPB)를 적절하게 제어하며, DPB에 저장된 영상들은 Reference Picture List Re-ordering(RPLP) 명령어를 사용하여 임의적으로 참조 영상으로 사용할 수 있다. 이러한 H.264의 특징들은 이전 비디오 부호화 방식들이 지원하지 못했던 임의 순서의 부호화 및 참조 영상의 선택을 가능하게 한다. 이를 기반으로 B 영상의 참조를 허용하는 General B picture로써 시간적 스케일러빌리티를 지원하는 hierarchical B picture가 제안되었다[5]. 비디오 첫 영상은 IDR(Instantaneous Decoder Refresh) 영상으로서 화면 내 부호화 된다.

(그림 1) GOP 크기가 8일 때 기존의 수신된 프레임 순서 기반 방법에 의한 동영상 재생 예



(Figure 1) When GOP size is 8, the example of video replay by the existing video replay method based on frame receiving order

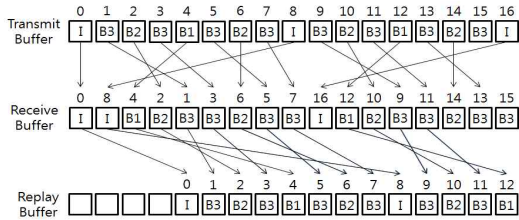
(그림 1)에서 기존의 수신된 프레임 순서 기반 방법을 사용했을 때 송신버퍼에서의 프레임 순서와 수신버퍼에서 프레임의 순서가 달라진 것을 확인할 수 있다. 수신버퍼와 송신버퍼를 비교하면 순서대로 패킷을 재생할 시 송신버퍼 순서가 아닌 불규칙적인 패킷들이 수신버퍼에 입력된다. 시각적 스케일러빌리티 기술을 사용하여 hierarchical B Picture 구조로 GOP 크기를 16으로 설정한다. 비디오 화면에서 손실되면 안 되고 크기가 제일 큰 I 프레임 영상 패킷이 인코딩과

정을 거치면서 전송되기 시작한다. 중요한 패킷을 가지는 I 프레임 영상 패킷에 손실이 일어 날 경우에는 다른 B 프레임 패킷에도 치명적인 손실이 발생하게 된다. (그림 1)에서는 송신 단의 송신버퍼와 다른 순서로 프레임을 수신했다. 이 프레임이 수신버퍼를 통과한 뒤 재생버퍼로 그대로 입력된다면 불규칙한 패킷 크기로 입력된다. 만약 수신되는 패킷 량이 재생 량보다 더 클 경우에는 오버플로우가 발생하고 반대로 수신되는 패킷 량이 재생 량보다 작을 경우에는 언더플로우가 발생한다.

3. 제안하는 프레임 수신 순서 재할당 방법

수신 단에서 송신 단의 송신버퍼의 프레임 순서와 동일한 프레임 순서를 유지하기 위해서는 (그림 2)에서 보여주고 있는 것처럼 수신버퍼와 재생버퍼의 사이에 자동으로 재생시점을 맞추는 방법이 있다. 이 방법으로 인해 재생버퍼가 송신버퍼의 프레임 순서와 동일한 프레임 순서를 가지게 된다. 유무선 네트워크에서는 송신버퍼에서 수신버퍼로 전송되는 인코딩과정에서 프레임의 순서가 달라질 수 있다. GOP 크기에 인코딩 되는 순서에 hierarchical B picture 기술을 이용하면 I 프레임의 위치가 달라지고 B 프레임이 늘어나는 것을 확인할 수 있을 것이다. 따라서 재배치되는 위치가 GOP 마다 달라 질 것이며 GOP 크기가 커질수록 더욱 복잡한 것을 확인할 수 있을 것이다.

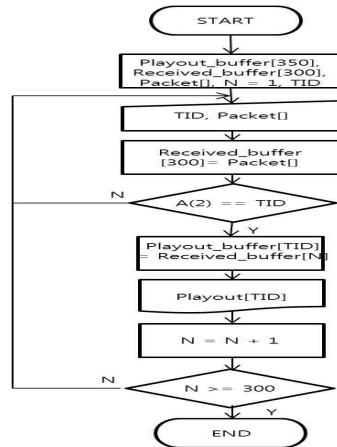
(그림 2) GOP 크기가 8일 때 제안하는 프레임 수신 순서 재할당 방법에 의한 동영상 재생 예



(Figure 2) When GOP size is 8, the example of video replay by the proposed video replay method based on frame receiving order rearrangement

(그림 2)는 GOP 크기가 8일 때 제안하는 프레임 수신 순서 재할당 방법에 의한 동영상 재생 예를 보여주고 있다. GOP 크기에 따라서 수신버퍼에 들어오는 프레임의 순서가 달라지며 재생버퍼의 프레임 순서 또한 달라진다. GOP 크기에 따라서 자동으로 재생시점을 맞추는 방법으로는 (그림 3)과 같은 알고리즘을 사용할 수 있다. 제안하는 알고리즘을 통하여 송신버퍼의 프레임 정보들처럼 순서대로 재생 할 프레임 패킷 정보들이 들어 왔는지 판단을 할 수 있다.

(그림 3) 자동으로 재생시점을 맞추는 방법



(Figure 3) Algorithm to automatically adjust a replay timing according to GOP size

(그림 3)은 수신버퍼와 재생버퍼사이에 자동으로 재생시점을 맞추는 알고리즘이다.

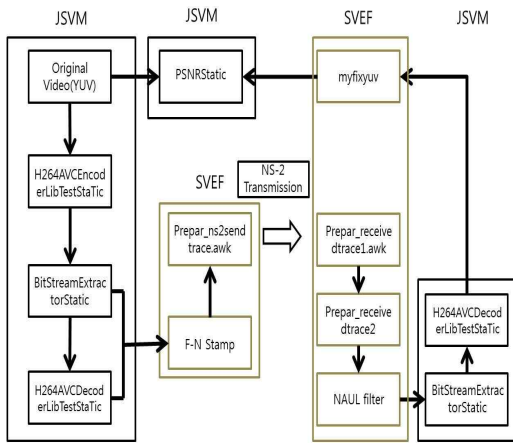
Payout_buffer(재생버퍼)은 Received_buffer(수신버퍼)에서 받은 패킷들을 할당 받는 공간이다. 먼저 실시간으로 유무선 네트워크를 통해 패킷과 프레임(TID)들이 수신버퍼에 수신된다. 이 수신버퍼에 있는 패킷과 프레임들을 송신버퍼의 패킷, 프레임과 동일한 순서로 재구성하는 재생버퍼를 만들기 위해 송신버퍼의 순서대로 프레임 3개가 정상적으로 수신되었는지 판단 한 뒤, 정상적으로 수신되었을 경우에만 해당 프레임(TID)을 재생버퍼에 차례대로 패킷을 넣을 수 있도록 하였다. 예를 들어, 한 yuv 영상에 대해서 전체 프레임 개수가 300인 경우에 동작하는 프레임 율이 30인 경우, 수신버퍼에서 1개의 프레임을 평균 0.3333초마다 수신했을 시 3개의 프

레이프는 0.1초마다 수신버퍼, 재생버퍼, 송신버퍼의 순서로 재할당된다. 따라서 상기 알고리즘처럼 재생버퍼를 사용하여 동영상을 재생한다면, 재생버퍼에서는 패킷들을 송신버퍼의 패킷들과 동일한 순서대로 수신하기 때문에 오버플로우와 언더플로우가 발생하는 것을 방지할 수 있다. 재생버퍼는 출력이 가능하며 해당 동영상이 300프레임을 가지고 있다고 가정하면 N은 300보다 크거나 같을 때까지 반복을 하도록 하였다.

4. 시뮬레이션 환경

본 논문에서는 JSVM 9.19.14, SVEF-1.4, ns-2.35 버전을 사용을 하였다. 상기 버전을 사용하는 이 방법은 일반적으로 비디오 스트리밍 시뮬레이션에 많이 쓰이는 기법이다[6]. (그림 4)의 시뮬레이션 과정에서 비디오 파일의 정보를 임의로 만든 .awk을 이용하여 원하는 트레이스 파일 형태로 만든다. 제공하는 도구를 사용하여 NS-2 성능을 분석할 수 있다[7]. 처음으로 해당 비디오 영상을 이용하여 JSVM 도구를 사용할 수 있다[8]. YUV 오리지널 영상을 H264AVCEncoderLibTestStatic을 사용하여 인코딩 한다.

(그림 4) JSVM과 SVEF를 이용한 전체 시뮬레이션 과정



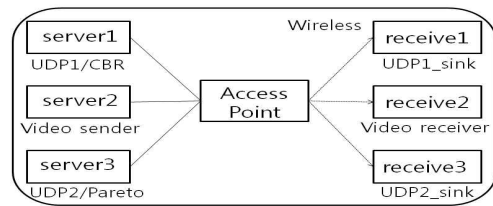
(Figure 4) Total simulation process using JSVM and SVEF

그 이후로 BitStreamExtractorStatic을 사용하여 비트스트림 추출된 트레이스 파일을 만들고

마지막으로 H264AVCDecoderLibTestStatic 사용하여 영상의 디코딩된 트레이스 파일을 만든다. JSVM을 이용하여 2개의 트레이스 파일을 SVEF로 옮겨서 F-N Stamp을 사용하여 하나의 트레이스 파일을 생성한다. SVEF에서 필요한 전송 트레이스 파일을 만들어 NS-2 전송을 수행하기 위해 Prepar_ns2sendtrace.awk을 사용하였다. 전송 할 때 송신 단에서 MyEvalSVC를 참조하여 진행하였으며[9] 수신 단에서는 MyEvalSVC_Sink를 참조하였다. 재생을 하기 위해서 타이머 함수를 이용하여 원하는 트레이스 파일을 얻을 수 있도록 했다.

NS-2로 통하여 시뮬레이션을 최종적으로 수신한 트레이스 파일을 SVEF 파일에 필요한 형식으로 Prepar_receivedtrace1.awk을 사용하여 트레이스 파일을 변환한다. Prepar_receivedtrace2를 이용하여 SVEF에 필요한 수신기 트레이스 파일을 만든다. 수신기 트레이스 파일에 NALU filter를 사용하여 프레임 중속성으로 인해 디코딩 할 수 없는 패킷을 삭제한다. 이후, JSVM에서 디코딩 할 수 있는 트레이스 파일을 옮겨 비트스트림을 추출해 트레이스 파일을 만들고 디코딩을 한다. 동영상을 재생하면 끊기는 부분이 있기 때문에 myfixyuv를 사용하여 패킷이 삭제된 부분은 이전 프레임 그대로 멈춰 있는 영상을 확인 할 수 있다. 본 영상과 시뮬레이션 한 영상 두 개를 이용하여 최대 신호 대 잡음비 도구를 사용해 성능을 확인할 수 있다.

(그림 5) 시뮬레이션 토폴로지



(Figure 5) Simulation topology

시뮬레이션 토폴로지는 (그림 5)와 같다. 각 트래픽 서버와 AP는 유선으로 연결되어 있으며 유선 대역폭은 각 3 Mbps이며 무선 대역폭은 10 Mbps로 설정하였다. 전파 모델은 Two-ray ground 모델을 사용하여 경쟁 트래픽을 생성하

도록 하였다. sever 1 노드는 CBR 트래픽을 생성하여 receive 1에 전송하고 server 3 노드는 파레토 트래픽을 생성하여 receive 3에 전송한다. 이때, 비디오 서버는 초당 30개 프레임의 정보를 receive 2에 전송하게 되며 시뮬레이션 시간 내 특정 구간에서만 전송을 시도하고 종료하여 무선 대역폭을 차지하게 된다. 파레토 트래픽을 발생시키는 On 기간과 트래픽을 발생시키는 않는 Off 기간은 파레토 분포에 따라 결정되는데 본 시뮬레이션에서는 설정한 값은 burst_time이 0.5초, idle_time이 1초로 평균 0.5초의 On 기간과 평균 1초 동안의 Off 기간을 가진다[5].

<표 1> 사용된 시뮬레이션 파라미터들

Used simulation tool	NS-2, JSVM	
Node number	sender 3, receiver 3, AP 1	
Propagation model	Two-Ray ground model	
Wired bandwidth	3Mbps	
Wireless bandwidth	10Mbps	
Simulation video type	soccer.yuv	city.yuv
Video resolution	CIF(352X288)	
Total frame number	300 frame	
GOP size	16	
Frame rate	30	
Total packet size/300 frame	592,791 byte	426,444 byte
Average playout packet size/3 frame	5,928 byte	4,264 byte
Overflow generation packet size	30,000 byte	

<Table 1> The used simulation parameters

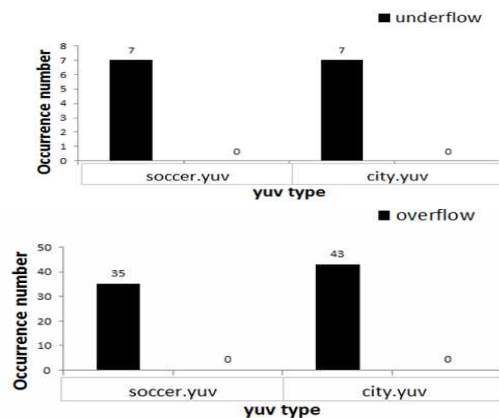
<표 1>는 시뮬레이션에서 사용된 파라미터들을 나타내고 있다. 송신 노드와 수신 노드는 각 1개씩 설정하고 시뮬레이션은 NS-2 도구를 사용하여 수행하였다. 시뮬레이션에서 사용된 비디오는 soccer.yuv와 city.yuv 2개를 사용하였으며 각각 300 프레임을 가지고 있다. 1초에 30 프레임을 재생할 수 있도록 프레임 율을 30으로 설정하고 각 영상은 CIF (352x288) 해상도를 가지

고 있다. 본 시뮬레이션에서는 인코딩 과정에서 사용되는 GOP 크기를 16으로 설정하였다. 각 영상의 재생 패킷 크기는 0.1초로 설정하였으며 수신버퍼에 누적되는 패킷 크기는 soccer.yuv는 5,928 바이트이고 city.yuv는 4,264 바이트 단위로 재생이 되도록 하였다. 오버플로우 발생 패킷 크기는 30,000 바이트로 설정하였다. (그림 3)의 자동으로 재생시점을 맞추는 알고리즘에 의해 재생시점을 설정하여 시뮬레이션을 수행하게 되면 <표 1>과 같은 시뮬레이션 파라미터를 가지게 된다.

5. 시뮬레이션 결과 및 분석

본 장은 기존 방법과 제안한 방법의 성능분석을 위해 시뮬레이션을 수행한 결과를 분석하였다. 재생을 수행하기 위한 가상의 평균 패킷 크기를 사용하여 시뮬레이션을 할 경우 누적되는 패킷크기가 최대 버퍼 크기인 30,000 바이트를 넘어가는 순간 오버플로우가 발생하게 된다. soccer.yuv로 예를 들면, 평균 패킷 크기는 5,928 바이트를 사용하는데 수신버퍼에서 누적 패킷크기가 30,000 바이트보다 커지는 경우에는 오버플로우가 발생되고 누적 패킷크기가 5,928 바이트보다 작아 재생할 수 있는 패킷 량이 부족한 경우에는 언더플로우가 발생된다.

(그림 6) 기존 방법과 제안된 방법에 의한 버퍼 오버플로우와 언더플로우 발생 횟수

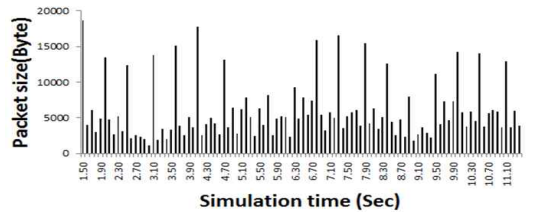


(Figure 6) Generation number of buffer overflow and underflow by the existing method and proposed method

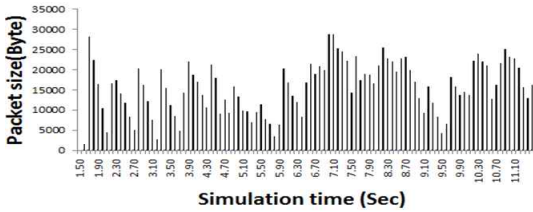
(그림 6)은 기존 방법과 제안된 방법에 의한 버퍼 오버플로우와 버퍼 언더플로우의 발생 횟수를 보여주고 있다. 버퍼 오버플로우의 발생 횟수를 살펴보면 기존 방법에 의해서는 soccer.yuv 영상에서 35회 발생하고 city.yuv 영상에서는 오버플로우가 43회 발생하였다. 반면에 제안하는 방법에 의해서는 2개의 영상에서 모두 0회로 오버플로우가 발생하지 않는 것을 확인할 수 있다.

마찬가지로 버퍼 언더플로우의 발생 횟수를 살펴보면 기존 방법에 의해서는 soccer.yuv와 city.yuv 영상에서 모두 7회 발생하고 제안하는 방법에 의해서는 두 영상 모두 0회로 언더플로우가 발생하지 않는 것을 확인할 수 있다. 제시하는 방법을 이용, 별도의 재생버퍼를 추가해 수신버퍼의 프레임 순서를 송신버퍼와 동일하게 재배치한 결과 오버플로우와 언더플로우가 발생하지 않았다.

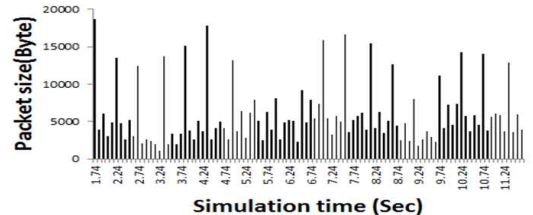
(그림 7) soccer.yuv 영상일 때 전송, 수신, 재생버퍼에서의 패킷 크기



(a) Transmit buffer in transmitter



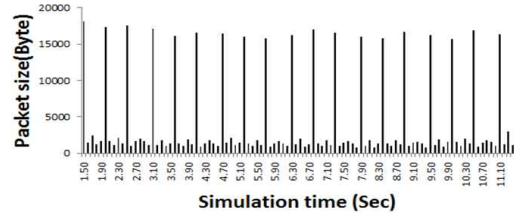
(b) Receive buffer in receiver



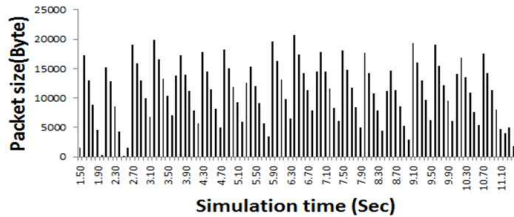
(c) Replay buffer in receiver

(Figure 7) When soccer.yuv is used, packet size in the transmit, receive, and replay buffer

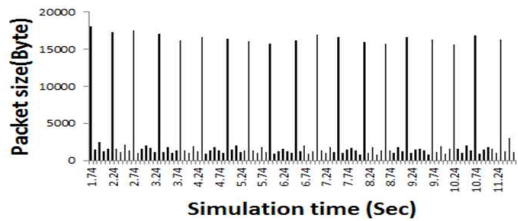
(그림 8) city.yuv 영상일 때 전송, 수신, 재생버퍼에서의 패킷 크기



(a) Transmit buffer in transmitter



(b) Receive buffer in receiver



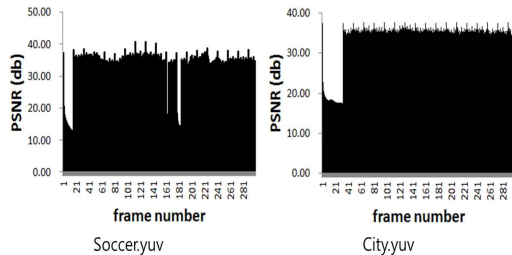
(c) Replay buffer in receiver

(Figure 8) When city.yuv is used, packet size in the transmit, receive, and replay buffer

(그림 7)은 soccer.yuv 영상이, (그림 8)은 city.yuv 영상이 사용될 때 시뮬레이션 시간에 따른 패킷 크기를 나타낸 송신버퍼, 수신버퍼, 재생버퍼 그림이다. (그림 7)과 (그림 8)의 모든 실험 결과에서 y축의 패킷 크기는 3 프레임 동안 패킷들의 누적 합을 의미한다. (그림 7)과 (그림 8)에는 송신 단의 송신버퍼와 제안한 방법에 의해 새롭게 만들어진 수신 단의 재생버퍼에서 x축의 시뮬레이션 시간을 확인할 수 있다. 재생버퍼가 송신버퍼에 3개의 프레임이 제대로 들어왔는지 확인한 이후에 시작하기 때문에 약간의 시작 지연이 발생하지만 y축의 패킷 크기가 동일한 형태를 가지는 것을 확인할 수 있다. 반 기존 방법에서 패킷을 최종적으로 수신하는 수신버퍼의 경우 송신버퍼와 다른 패킷 크기를 가지고, 불규칙하게 수신됨을

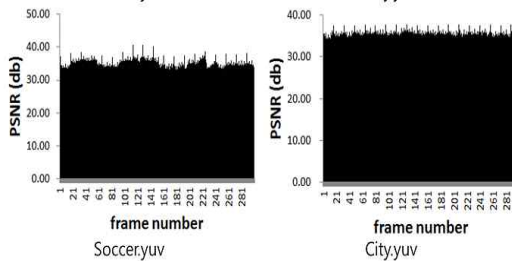
확인할 수 있다. 이러한 불규칙성 때문에 버퍼 오버플로우와 언더플로우가 다수 발생하는 것으로 파악된다.

(그림 9) 기존 방법에 의한 최대 신호 대 잡음비 성능



(Figure 9) PSNR performance by the existing method

(그림 10) 제안하는 방법에 의한 최대 신호 대 잡음비 성능



(Figure 10) PSNR performance by the proposed method

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |s(i,j) - p(i,j)|^2 \quad (1)$$

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (2)$$

$$= 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

식 (1)의 평균 제곱 오차(Mean Square Error)를 이용하여 식 (2)의 최대 신호 대 잡음비(Peak Signal-to-Noise Ratio)를 계산할 수 있다. 식 (1)에서 $s(i, j)$ 은 원 영상, $p(i, j)$ 은 부호화된 영상이다. m 과 n 은 영상의 크기를 의미한다. 일반적으로 8비트 영상을 사용하므로 PSNR의 계산

에서 분자에 들어가는 숫자는 255^2 이다.[10] 식 (1)을 이용하여 식 (2) PSNR을 구할 수 있으며 MAX는 해당 영상의 최대의 값으로써, 해당 채널의 최대의 값에서 최소의 값을 빼서 구할 수 있다. 기존 방법과 제안한 방법에서의 최대 신호 대 잡음비의 결과는 (그림 9)과 (그림 10)에서 볼 수 있다. 먼저 제안한 방법에 의한 최대 신호 대 잡음비를 나타내는 (그림 10)은 모든 프레임들에서 일정한 최대 신호 대 잡음비 값을 가짐을 확인할 수 있다. 기존 방법에 의한 최대 신호 대 잡음비를 나타내는 (그림 9)을 보면 (그림 10)의 제안한 방법에 의한 최대 신호 대 잡음비 값보다 일부 프레임에서 급격히 작은 값들을 가짐을 확인할 수 있다. 결과적으로 (그림 9)에서는 최대 신호 대 잡음비 값이 떨어지는 일부 프레임 구간들이 존재하기 때문에 재생 품질이 우수하지 않을 것이지만 (그림 10)에서는 최대 신호 대 잡음비 값이 떨어지는 프레임 구간이 없기 때문에 비디오 재생 품질이 우수함을 알 수 있다.

6. 결론

기존의 수신된 프레임 순서 기반 방법처럼, 수신버퍼 내의 프레임이 송신버퍼와 다른 순서를 가진 채로 동영상을 재생하게 되면 수신버퍼내의 패킷량과 재생 패킷량이 일치하지 않아 버퍼 오버플로우와 버퍼 언더플로우가 발생하는 것을 확인할 수 있었다. 하지만 본 논문에서 제안하는 프레임 수신 순서 재할당 방법에 의해서 전송 프레임 간에 분산 효과를 유지하는 역할을 수행하는 재생버퍼를 추가하여 시뮬레이션을 수행하면 버퍼 오버플로우와 버퍼 언더플로우가 발생하지 않는다. 시뮬레이션 결과, 버퍼 오버플로우와 버퍼 언더플로우의 발생 횟수, 그리고 최대 신호 대 잡음비 성능 측면에서 제안한 방법이 기존 방법보다 더 우수함을 확인할 수 있었다. 제안하는 방법에서 발생할 수 있는 문제점으로는 ns-2 시뮬레이션 과정에서 비디오 정보가 혼잡 상태에 빠질 경우에 재생버퍼에서 재생해야 할 패킷들의 부족으로 동영상이 멈추는 현상이 발생할 수도 있다. 따라서 향후 연구로써 혼잡이 일어날 경우에도 재생을 원활하게 할

수 있는 새로운 방법에 대한 연구를 진행할 예정이다며 또한 열악한 무선통신 환경의 추가에 따른 성능 열화를 보상하는 새로운 알고리즘을 찾는 연구도 진행할 예정이다.

References

[1] Chang, S.F. and Vetro, A., "Video adaptation concepts, technologies, and open issues," Proceedings of the IEEE, vol. 93, Issue 1, pp. 148-158, Jan. 2005.

[2] Source: TDG Research, <http://www.tdgresearch.com>.

[3] Cisco visual networking index : Forecast and methodology, <http://www.cisco.com/>

[4] M.Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-time system for adaptive video streaming based on SVC," IEEE Transactions on Circuits and System for Video Technology, vol.17, no.9, pp.1227-1237, Sep. 2007.

[5] Haechul Choia, Kyung Il Leea, Jung Won Kanga, Seong-Jun Baea, and Jeong-Ju Yoo . "Overview and Performance Analysis of the Emerging Scalable Video Coding" . The Korean Institute of Broadcast and Media Engineers. vol. 12, no 6, pp.542-554 . No v. 2007

[6] Young-Min Jang, The application of NS-2 Network Simulation, Hongrung Publishing Company, 2008.

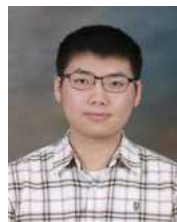
[7] Chin-Heng Ke, How to do H.264 SVC transmission simulations part2 : combined scalability(2012), Retrieved Sep, 2015, from <http://csie.nqu.edu.tw/smallko>

/ns2/svc2.htm

[8] JSVM Software Manual(2006)

[9] C. -H. Ke, "myEvalSVC : An intrgrated simulation framework for evaluation of H.264./SVC transmission," KSII Trans. internet abd Inf. Syst., vol. 6, no. 1, 99. pp. 378-393, Jan. 2012.

[10] Jungdong Seo and Kwanghoon Sohn . "PSNR Estimation of H.264/AVC Bitstream for Hierarchical-B Picture Structure". The Korean Society of Broadcast Engineers. vol. 16, no 6, pp. 996-1008. Nov. 2011



강 동 진

2011년 3월 - 현재 : 강원대학교 IT 대학 전기전자공학부 학사과정

관심분야 : 차세대 이동통신 및 차세대 무선 네트워크



김 동 회

2005년 : 고려대학교 전파공학과 (공학박사)

1989년 1월 ~ 1997년 1월 : 삼성전자 전임연구원
 2000년 8월 ~ 2005년 8월 : 한국전자통신연구원 선임연구원
 2006년 3월 ~ 현재 : 강원대학교 IT대학 전기전자공학부 전자통신학과 교수
 관심분야 : 이동통신 및 무선 네트워크 등