



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년07월05일
 (11) 등록번호 10-1636198
 (24) 등록일자 2016년06월28일

(51) 국제특허분류(Int. Cl.)
 H04N 21/24 (2011.01) H04N 21/236 (2011.01)
 H04N 21/433 (2011.01)
 (52) CPC특허분류
 H04N 21/2404 (2013.01)
 H04N 21/236 (2013.01)
 (21) 출원번호 10-2015-0032436
 (22) 출원일자 2015년03월09일
 심사청구일자 2015년03월09일
 (56) 선행기술조사문헌
 KR1020030055701 A
 KR1020070053256 A
 JP2004529528 A
 Kyeong Soo Kim, "The Effect of ISP Traffic Shaping on User-Perceived Performance in Broadband Shared Access Networks", Preprint submitted to Computer Networks, June 4, 2014.*
 *는 심사관에 의하여 인용된 문헌

(73) 특허권자
 강원대학교산학협력단
 강원도 춘천시 강원대학길 1 (효자동)
 (72) 발명자
 김동희
 강원 춘천시 강원대학길 1, IT대학 전기전자공학부 (효자동, 강원대학교)
 이현노
 강원도 춘천시 동내면 거두택지길 7, 103동 606호 (초록지붕아파트)
 (74) 대리인
 김정현

전체 청구항 수 : 총 10 항

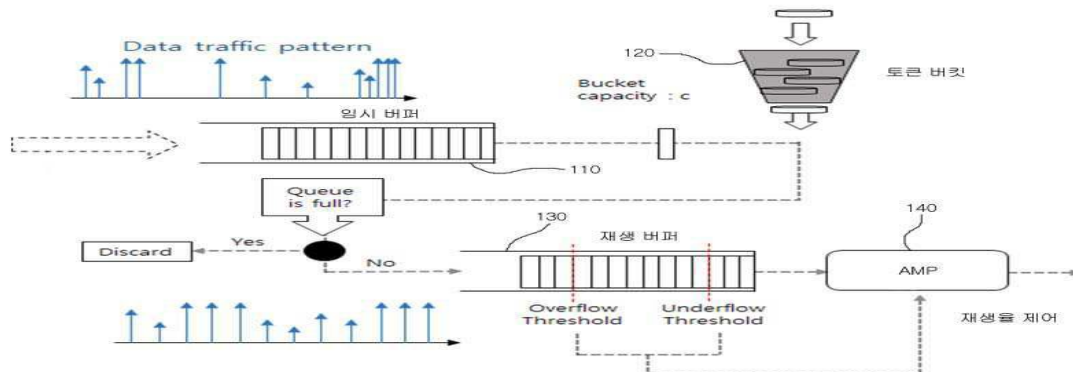
심사관 : 진민숙

(54) 발명의 명칭 **무선 통신 환경에서 비디오 스트리밍의 트래픽 제어 방법 및 이를 구현하는 수신기 장치**

(57) 요약

본 발명의 무선 통신 환경에서 비디오 스트리밍의 트래픽 제어 방법에서, 송신측으로부터 비디오 데이터 트래픽이 송신되면, 임시 버퍼에 임시로 저장하는 단계, 상기 임시 버퍼에 저장된 비디오 데이터의 패킷에 대해 토큰 버킷(Token bucket) 기법을 적용하여 하나의 패킷에 대해 하나의 토큰을 소비하는 방식으로, 미리 정해진 토큰의 개수 내에서만 패킷을 재생 버퍼로 허용하는 단계 및 상기 재생 버퍼의 오버플로우(overflow) 또는 언더플로우(underflow)를 방지하기 위하여 AMP(Adaptive Media Playout) 기법을 적용하여 동적으로 재생 속도를 조절하는 단계를 포함한다. 본 발명에 의하면, 무선통신 환경에서 비디오 스트리밍의 트래픽을 제어하기 위하여 토큰 버킷 기법과 AMP 기법을 결합하여 적용함으로써, 버퍼에서의 언더플로우와 오버플로우 문제를 해소하고, 비디오 스트리밍 성능을 향상시킬 수 있는 효과가 있다.

대표도



(52) CPC특허분류

H04N 21/4331 (2013.01)

H04N 21/64723 (2013.01)

명세서

청구범위

청구항 1

무선 통신 환경에서 비디오 스트리밍의 트래픽 제어 방법에서,

송신측으로부터 비디오 데이터 트래픽이 송신되면, 임시 버퍼에 임시로 저장하는 단계;

상기 임시 버퍼에 저장된 비디오 데이터의 패킷에 대해 토큰 버킷(Token bucket) 기법을 적용하여 하나의 패킷에 대해 하나의 토큰을 소비하는 방식으로, 미리 정해진 토큰의 개수 내에서만 패킷을 재생 버퍼로 허용하는 단계; 및

상기 재생 버퍼의 오버플로우(overflow) 또는 언더플로우(underflow)를 방지하기 위하여 AMP(Adaptive Media Playout) 기법을 적용하여 동적으로 재생 속도를 조절하는 단계를 포함하되,

토큰이 저장되어 있는 버킷의 용량을 c , 초당 생성되는 토큰의 수를 r 이라고 할 때, 임의의 시간 간격 Δt 동안에 상기 재생 버퍼로 들어올 수 있는 비디오 패킷의 최대 수 (N_{max})를,

$$N_{max} = r \times \Delta t + c \quad (\text{수학식 1}) \text{로 나타내고,}$$

상기 토큰 버킷의 최대 평균 속도 (AV_{max})를,

$$AV_{max} = \frac{N_{max}}{\Delta t} \quad (\text{수학식 2}) \text{로 나타낼 수 있는 것을 특징으로 하는 비디오 스트리밍의 트래픽 제어 방법.}$$

청구항 2

삭제

청구항 3

청구항 1에 있어서,

전송하기에 충분한 패킷이 상기 임시 버퍼에 저장되어 있다고 가정하고,

전송되는 패킷의 최대 크기(P_{txmax})에 따라, 토큰 생성에 따른 유입 가능한 최대 패킷량($P_{tokenmax}$)을,

$$P_{tokenmax} = P_{txmax} \times r \quad (\text{수학식 3}) \text{으로 나타내고,}$$

버킷 용량에 따른 유입 가능한 최대 패킷량($P_{bucketmax}$)을,

$$P_{bucketmax} = P_{txmax} \times c \quad (\text{수학식 4}) \text{로 나타내고,}$$

재생량($L_{playout}$)은 단위 시간당 상기 재생 버퍼에서 재생되어 나가는 패킷의 양(ΔP)으로 계산되며,

$$L_{playout} = \Delta P \times \Delta t \quad (\text{수학식 5}) \text{로 나타낼 수 있는 것을 특징으로 하는 비디오 스트리밍의 트래픽 제어 방법.}$$

청구항 4

청구항 3에 있어서,

상기 재생 버퍼가 비어있다고 가정하고,

단위시간 Δt 동안에 상기 재생 버퍼로 유입되는 패킷의 양은, 상기 재생 버퍼의 최대 용량(B_{max})에 재생량 (L_{payout})을 합한 값보다 적거나 같아야 버퍼의 포화상태를 방지할 수 있으며, 이를

$(P_{tokenmax} \times \Delta t) + P_{bucketmax} \leq B_{max} + L_{payout}$ (수학식 6)으로 나타낼 수 있는 것을 특징으로 하는 비디오 스트리밍의 트래픽 제어 방법.

청구항 5

청구항 4에 있어서,

상기 재생 버퍼가 고갈되거나 포화되는 것을 방지하기 위해 언더플로우 임계값(Underflow Threshold)과 오버플로우 임계값(Overflow Threshold)을 정하고, 상기 재생 버퍼의 패킷량을 측정하여 해당 임계값에 도달했을 때, 상기 AMP 기법에서는 프레임 재생의 속도를 적응적으로 변화시키는 것을 특징으로 하는 비디오 스트리밍의 트래픽 제어 방법.

청구항 6

청구항 5에 있어서,

C_{payout} 는 재생 속도 변경율을 의미한다고 할 때,

상기 AMP 기법에서 언더플로우일 경우에 변경하는 재생 속도($V_{underflow}$)는,

$V_{underflow} = V_{normal} - V_{normal} \times C_{payout}$ (수학식 9)로 표현되고,

오버플로우일 경우에 변경하는 재생 속도($V_{overflow}$)는,

$V_{overflow} = V_{normal} + V_{normal} \times C_{payout}$ (수학식 10)으로 표현되는 것을 특징으로 하는 비디오 스트리밍의 트래픽 제어 방법.

청구항 7

무선 통신 환경에서 비디오 스트리밍의 트래픽을 수신하는 수신기 장치에서,

송신측으로부터 비디오 데이터 트래픽이 송신되면, 임시로 저장하기 위한 임시 버퍼;

상기 임시 버퍼에 저장된 비디오 데이터의 패킷에 대해 토큰 버킷(Token bucket) 기법을 적용하여 하나의 패킷에 대해 하나의 토큰을 소비하는 방식으로, 미리 정해진 토큰의 개수 내에서만 패킷을 허용하기 위한 토큰 버킷부;

상기 토큰 버킷부에서 허용된 패킷을 저장하기 위한 재생 버퍼; 및

상기 재생 버퍼의 오버플로우(overflow) 또는 언더플로우(underflow)를 방지하기 위하여 AMP(Adaptive Media Playout) 기법을 적용하여 동적으로 재생 속도를 조절하는 AMP 부를 포함하되,

토큰이 저장되어 있는 버킷의 용량을 c , 초당 생성되는 토큰의 수를 r 이라고 할 때, 임의의 시간 간격 Δt 동안에 상기 재생 버퍼로 들어올 수 있는 비디오 패킷의 최대 수 (N_{max})를,

$N_{max} = r \times \Delta t + c$ (수학식 1)로 나타내고,

상기 토큰 버킷의 최대 평균 속도 (AV_{max})를,

$$AV_{max} = \frac{N_{max}}{\Delta t}$$

(수학식 2)로 나타낼 수 있는 것을 특징으로 하는 수신기 장치.

청구항 8

삭제

청구항 9

청구항 7에 있어서,

전송하기에 충분한 패킷이 상기 임시 버퍼에 저장되어 있다고 가정하고,

전송되는 패킷의 최대 크기(P_{txmax})에 따라, 토큰 생성에 따른 유입 가능한 최대 패킷량($P_{tokenmax}$)을,

$$P_{tokenmax} = P_{txmax} \times r$$

(수학식 3)으로 나타내고,

버킷 용량에 따른 유입 가능한 최대 패킷량($P_{bucketmax}$)을,

$$P_{bucketmax} = P_{txmax} \times c$$

(수학식 4)로 나타내고,

재생량(L_{payout})은 단위 시간당 상기 재생 버퍼에서 재생되어 나가는 패킷의 양(ΔP)으로 계산되며,

$$L_{payout} = \Delta P \times \Delta t$$

(수학식 5)로 나타낼 수 있는 것을 특징으로 하는 수신기 장치.

청구항 10

청구항 9에 있어서,

상기 재생 버퍼가 비어있다고 가정하고,

단위시간 Δt 동안에 상기 재생 버퍼로 유입되는 패킷의 양은, 상기 재생 버퍼의 최대 용량(B_{max})에 재생량(L_{payout})을 합한 값보다 적거나 같아야 버퍼의 포화상태를 방지할 수 있으며, 이를

$$(P_{tokenmax} \times \Delta t) + P_{bucketmax} \leq B_{max} + L_{payout}$$

(수학식 6)으로 나타낼 수 있는 것을 특징으로 하는 수신기 장치.

청구항 11

청구항 10에 있어서,

상기 재생 버퍼가 고갈되거나 포화되는 것을 방지하기 위해 언더플로우 임계값(Underflow Threshold)과 오버플로우 임계값(Overflow Threshold)을 정하고, 상기 재생 버퍼의 패킷량을 측정하여 해당 임계값에 도달했을 때, 상기 AMP 부에서는 프레임 재생의 속도를 적응적으로 변화시키는 것을 특징으로 하는 수신기 장치.

청구항 12

청구항 11에 있어서,

C_{payout} 는 재생 속도 변경율을 의미한다고 할 때,

상기 AMP 부에서 언더플로우일 경우에 변경하는 재생 속도($V_{underflow}$)는,

$V_{underflow} = V_{normal} - V_{normal} \times C_{playout}$ (수학식 9)로 표현되고,

오버플로우일 경우에 변경하는 재생 속도($V_{overflow}$)는,

$V_{overflow} = V_{normal} + V_{normal} \times C_{playout}$ (수학식 10)으로 표현되는 것을 특징으로 하는 수신기 장치.

발명의 설명

기술 분야

[0001] 본 발명은 무선통신 환경에서 비디오 스트리밍의 버퍼 언더플로우와 오버플로우 문제를 개선하기 위한 기술에 관한 것이다.

배경 기술

[0002] 무선통신 환경에서 실시간 비디오 스트리밍 제공에 있어 만족스러운 품질의 서비스가 이루어져야 하며, 중단 간 품질에 있어 중요한 고려사항 중 하나가 재생 품질 보장이다.

[0003] 수신측 재생 버퍼에 채워져 있는 패킷의 양은 네트워크 상황에 따라 시간적으로 가변하며 네트워크 혼잡에 의한 재생의 멈춤 또는 스킵과 같은 불연속적인 프레임의 재생뿐만 아니라 특정 패킷의 손실로 인한 화질 저하 등과 같은 재생 품질 손상들이 발생한다.

[0004] 이러한 문제를 해결하기 위해 무선통신 환경에서 재생 품질 보장하기 위한 기법에는 여러 가지가 있는데, 사용자 체감 품질을 만족시키기 위하여 최소 재생 버퍼 레벨을 보존하여 가변적인 네트워크 환경에서도 재생의 끊김 현상, 재생 중 비디오 품질의 급격한 변화등을 효율적으로 방지하는 비트율 적응 알고리즘도 그 중의 하나이고, 무선 네트워크에서 교차 계층에 기반한 네트워크 적응형 비디오 스트리밍 기법에 대해 제안된 바 있다.

[0005] 비디오 전송자는 MAC과 물리계층의 상태를 측정하여 시간적으로 변하는 무선랜의 채널 상태를 예측한다. 예측된 채널 상태 정보는 네트워크 적응형 비디오 전송을 제어하기 위해 사용된다. 또한 단지 네트워크 지터에 대해 보상하기 위하여 재생 지연 또는 버퍼 사이즈를 동적으로 조절하는 지터 버퍼 컨트롤 메커니즘이 있으며, 이 경우에 오디오 애플리케이션에 대해서는 효과적이거나, 실시간 비디오 스트리밍의 품질을 향상시키기에는 충분하지 않다.

[0006] 콘텐츠 인지형 AMP(Adaptive Media Playout) 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. 하지만 이러한 메커니즘의 경우 매우 높은 복잡성을 가지며 움직임이 많은 스포츠나 액션 영화 같은 경우에는 가능하지 않다.

[0007] AMP(Adaptive Media Playout) 기법은 버퍼에 있는 패킷 또는 버퍼의 변동 횟수에 따라 동적으로 재생 속도를 조절하게 된다. 버퍼에 임계값을 설정하고 언더플로우(underflow) 임계값이나 오버플로우(overflow) 임계값에 현재 측정된 버퍼량이 도달했을 경우, 재생 속도를 원래의 정상 재생 속도에서 감소시키거나 또는 증가시켜서 버퍼의 고갈 및 포화 상태를 방지한다. 즉, 버퍼의 포화상태에 가까워져 오버플로우의 위험이 생기면 재생 속도를 증가시켜서 단위 시간동안 더 많은 데이터를 출력하고, 버퍼의 고갈상태에 가까워져 언더플로우의 위험이 생기면 재생 속도를 감소시켜서 버퍼의 안정 상태를 유지한다.

[0008] 또한 수신측 재생 버퍼로 불규칙하게 들어오는 패킷을 제어하기 위한 토큰 버킷(Token Bucket)기법이 있다. 토큰 버킷 기법은 단위 시간 동안 생성되는 토큰의 수와 버킷의 용량을 파라미터로 설정함으로써, 토큰에 따라 버퍼로 유입되는 트래픽을 제어할 수 있다. 이러한 토큰 버킷 기법은 흐름 제어와 데이터 트래픽의 트래픽 성형을 위한 통신시스템에 널리 적용되고 있다. 토큰 버킷의 파라미터인 초당 토큰 재생률과 버킷의 최대 용량들을 조절하여 패킷 손실률을 감소시키기 위한 연구가 있으며, 멀티미디어 서비스에서 서비스 품질을 보장하기 위해 토큰 버킷을 이용한 대역폭 할당 방식을 사용되고 있다.

[0009] 무선통신 환경에서 수신측의 재생 버퍼에 있는 비디오 스트리밍의 패킷 정보량은 네트워크 상황에 따라 가변한

다. 패킷 정보량이 특정 버퍼량보다 작아지면 버퍼 언더플로우(buffer underflow) 문제가 발생하고 반대로 패킷 정보량이 특정 버퍼량보다 많아지면 버퍼 오버플로우(buffer overflow) 문제가 발생한다. 목적지 수신 버퍼에서 비디오 스트리밍을 재생할 때 이러한 버퍼 언더플로우와 오버플로우 문제는 멈춤 현상이나 스킵 현상을 유발하여 재생의 불연속성을 발생시키는 문제점이 있다.

선행기술문헌

특허문헌

[0010] (특허문헌 0001) 대한민국 공개특허 10-2011-0086642

발명의 내용

해결하려는 과제

[0011] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, 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 비디오 스트리밍의 트래픽을 제어하는 기술을 제공하는데 그 목적이 있다.

[0012] 본 발명의 목적은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0013] 이와 같은 목적을 달성하기 위한 본 발명의 무선 통신 환경에서 비디오 스트리밍의 트래픽 제어 방법에서, 송신측으로부터 비디오 데이터 트래픽이 송신되면, 임시 버퍼에 임시로 저장하는 단계, 상기 임시 버퍼에 저장된 비디오 데이터의 패킷에 대해 토큰 버킷(Token bucket) 기법을 적용하여 하나의 패킷에 대해 하나의 토큰을 소비하는 방식으로, 미리 정해진 토큰의 개수 내에서만 패킷을 재생 버퍼로 허용하는 단계 및 상기 재생 버퍼의 오버플로우(overflow) 또는 언더플로우(underflow)를 방지하기 위하여 AMP(Adaptive Media Playout) 기법을 적용하여 동적으로 재생 속도를 조절하는 단계를 포함한다.

[0014] 토큰이 저장되어 있는 버킷의 용량을 c, 초당 생성되는 토큰의 수를 r이라고 할 때, 임의의 시간 간격 Δt 동안에 상기 재생 버퍼로 들어올 수 있는 비디오 패킷의 최대 수 (Nmax)를, $N_{max} = r \times \Delta t + c$ (수학식 1)로

나타내고, 상기 토큰 버킷의 최대 평균 속도 (AVmax)를, $AV_{max} = \frac{N_{max}}{\Delta t}$ (수학식 2)로 나타낼 수 있다.

[0015] 전송하기에 충분한 패킷이 상기 임시 버퍼에 저장되어 있다고 가정하고, 전송되는 패킷의 최대 크기(Ptxmax)에 따라, 토큰 생성에 따른 유입 가능한 최대 패킷량(Ptokenmax)을, $P_{tokenmax} = P_{txmax} \times r$ (수학식 3)으로 나타내고,

[0016] 버킷 용량에 따른 유입 가능한 최대 패킷량(Pbucketmax)을, $P_{bucketmax} = P_{txmax} \times c$ (수학식 4)로 나타내고, 재생량(Lplayout)은 단위 시간당 상기 재생 버퍼에서 재생되어 나가는 패킷의 양(ΔP)으로 계산되며, $L_{playout} = \Delta P \times \Delta t$ (수학식 5)로 나타낼 수 있다.

[0017] 상기 재생 버퍼가 비어있다고 가정하고, 단위시간 Δt동안에 상기 재생 버퍼로 유입되는 패킷의 양은, 상기 재생 버퍼의 최대 용량(Bmax)에 재생량(Lplayout)을 합한 값보다 적거나 같아야 버퍼의 포화상태를 방지할 수 있

으며, 이를 $(P_{tokenmax} \times \Delta t) + P_{bucketmax} \leq B_{max} + L_{payout}$ (수학식 6)으로 나타낼 수 있다.

[0018] 상기 재생 버퍼가 고갈되거나 포화되는 것을 방지하기 위해 언더플로우 임계값(Underflow Threshold)과 오버플로우 임계값(Overflow Threshold)을 정하고, 상기 재생 버퍼의 패킷량을 측정하여 해당 임계값에 도달했을 때, 상기 AMP 기법에서는 프레임 재생의 속도를 적응적으로 변화시킬 수 있다.

[0019] Cpayout는 재생 속도 변경율을 의미한다고 할 때, 상기 AMP 기법에서 언더플로우일 경우에 변경하는 재생 속도(Vunderflow)는, $V_{underflow} = V_{normal} - V_{normal} \times C_{payout}$ (수학식 9)로 표현되고, 오버플로우일 경우에 변경하는 재생 속도(Voverflow)는, $V_{overflow} = V_{normal} + V_{normal} \times C_{payout}$ (수학식 10)으로 표현될 수 있다.

[0020] 본 발명에서 무선 통신 환경에서 비디오 스트리밍의 트래픽을 수신하는 수신기 장치에서, 송신측으로부터 비디오 데이터 트래픽이 송신되면, 임시로 저장하기 위한 임시 버퍼, 상기 임시 버퍼에 저장된 비디오 데이터의 패킷에 대해 토큰 버킷(Token bucket) 기법을 적용하여 하나의 패킷에 대해 하나의 토큰을 소비하는 방식으로, 미리 정해진 토큰의 개수 내에서만 패킷을 허용하기 위한 토큰 버킷부, 상기 토큰 버킷부에서 허용된 패킷을 저장하기 위한 재생 버퍼 및 상기 재생 버퍼의 오버플로우(overflow) 또는 언더플로우(underflow)를 방지하기 위하여 AMP(Adaptive Media Payout) 기법을 적용하여 동적으로 재생 속도를 조절하는 AMP 부를 포함한다.

[0021] 토큰이 저장되어 있는 버킷의 용량을 c, 초당 생성되는 토큰의 수를 r이라고 할 때, 임의의 시간 간격 Δt 동안에 상기 재생 버퍼로 들어올 수 있는 비디오 패킷의 최대 수 (N_{max})를, $N_{max} = r \times \Delta t + c$ (수학식 1)로

나타내고, 상기 토큰 버킷의 최대 평균 속도 (AV_{max})를, $AV_{max} = \frac{N_{max}}{\Delta t}$ (수학식 2)로 나타낼 수 있다.

[0022] 전송하기에 충분한 패킷이 상기 임시 버퍼에 저장되어 있다고 가정하고, 전송되는 패킷의 최대 크기(P_{txmax})에 따라, 토큰 생성에 따른 유입 가능한 최대 패킷량($P_{tokenmax}$)을, $P_{tokenmax} = P_{txmax} \times r$ (수학식 3)으로 나타내고, 버킷 용량에 따른 유입 가능한 최대 패킷량($P_{bucketmax}$)을, $P_{bucketmax} = P_{txmax} \times c$ (수학식 4)로 나타내고, 재생량(L_{payout})은 단위 시간당 상기 재생 버퍼에서 재생되어 나가는 패킷의 양(ΔP)으로 계산되며, $L_{payout} = \Delta P \times \Delta t$ (수학식 5)로 나타낼 수 있다.

[0023] 상기 재생 버퍼가 비어있다고 가정하고, 단위시간 Δt 동안에 상기 재생 버퍼로 유입되는 패킷의 양은, 상기 재생 버퍼의 최대 용량(B_{max})에 재생량(L_{payout})을 합한 값보다 적거나 같아야 버퍼의 포화상태를 방지할 수 있다. 으며, 이를 $(P_{tokenmax} \times \Delta t) + P_{bucketmax} \leq B_{max} + L_{payout}$ (수학식 6)으로 나타낼 수 있다.

[0024] 상기 재생 버퍼가 고갈되거나 포화되는 것을 방지하기 위해 언더플로우 임계값(Underflow Threshold)과 오버플로우 임계값(Overflow Threshold)을 정하고, 상기 재생 버퍼의 패킷량을 측정하여 해당 임계값에 도달했을 때, 상기 AMP 부에서는 프레임 재생의 속도를 적응적으로 변화시킬 수 있다.

[0025] Cpayout는 재생 속도 변경율을 의미한다고 할 때, 상기 AMP 부에서 언더플로우일 경우에 변경하는 재생 속도(Vunderflow)는, $V_{underflow} = V_{normal} - V_{normal} \times C_{payout}$ (수학식 9)로 표현되고, 오버플로우일 경우에 변경하는 재생 속도(Voverflow)는, $V_{overflow} = V_{normal} + V_{normal} \times C_{payout}$ (수학식 10)으로 표현될 수 있다.

발명의 효과

[0026] 본 발명에 의하면, 무선통신 환경에서 비디오 스트리밍의 트래픽을 제어하기 위하여 토큰 버킷 기법과 AMP 기법을 결합하여 적용함으로써, 버퍼에서의 언더플로우와 오버플로우 문제를 해소하고, 비디오 스트리밍 성능을 향상시킬 수 있는 효과가 있다.

도면의 간단한 설명

[0027] 도 1은 본 발명의 일 실시예에 따른 토큰 버킷 기법과 AMP 기법을 결합한 수신기의 구조를 나타내는 도면이다.
 도 2는 본 발명의 일 실시예에 따른 시뮬레이션 절차를 보여주는 도면이다.
 도 3은 본 발명의 일 실시예에 따른 시뮬레이션에 사용된 YUV 파일을 나타낸 표이다.
 도 4는 본 발명의 일 실시예에 따른 시뮬레이션 파라미터를 나타낸 표이다.
 도 5는 본 발명의 일 실시예에 따른 시뮬레이션에서의 토큰 버킷 파라미터에 따른 성능을 나타낸 표이다.
 도 6은 본 발명의 일 실시예에 따른 비디오 스트리밍 시뮬레이션에서 네 가지의 시나리오를 나타낸 표이다.
 도 7은 수신측 버퍼에 유입되는 비디오 스트리밍 패킷 크기의 패턴에 대해 도시한 그래프이다.
 도 8은 각각의 시뮬레이션 대상 비디오 데이터에 대해 케이스 1과 케이스 2에서 언더플로우와 오버플로우의 발생 횟수를 비교한 그래프이다.
 도 9는 각각의 시뮬레이션 대상 비디오 데이터에 대해 케이스 3과 케이스 4에서 언더플로우와 오버플로우의 발생 횟수를 비교한 그래프이다.
 도 10은 시뮬레이션 대상이 된 각 비디오에 대해 케이스 별로 언더플로우 및 오버플로우의 지속 시간을 나타낸 표이다.
 도 11은 케이스 3과 케이스 4에서 원래의 정상 재생 속도로 재생된 프레임의 비율 결과를 도시한 그래프이다.
 도 12는 각 시뮬레이션 결과에서 패킷 손실율을 도시한 그래프이다.
 도 13 내지 도 15는 각 영상에 대해 모든 케이스에서 수신된 비디오 스트리밍 패킷에 대해 PSNR을 측정하여 도시한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0028] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0029] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0030] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 갖고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 갖는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0031] 또한, 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조부호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다. 본 발명을 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다.

[0032] 본 발명에서는 AMP(Adaptive Media Playout) 기법과 토큰 버킷(Token Bucket) 기법을 결합하는 방식을 제안한다. 무선통신 환경에서 트래픽의 혼잡으로 인해 불규칙하게 들어오는 경우 그 정도가 심해지면 AMP만을 사용할 경우, 재생 속도 변경에 있어서 매우 많은 변화를 줘야 하거나, 재생속도 변경만으로는 버퍼(buffer)의 언더플로우(underflow)나 오버플로우(overflow) 현상을 해소하기 어려운 부분이 있을 수 있다. 따라서 재생 버퍼로 불규칙하게 들어오는 비디오 데이터를 토큰 버킷을 이용하여 어느 정도 규칙적으로 성형해주게 되면 AMP 제어 시 우수한 성능을 나타낼 수 있다.

[0033] 도 1은 본 발명의 일 실시예에 따른 토큰 버킷 기법과 AMP 기법을 결합한 수신기의 구조를 나타내는 도면이다.

[0034] 도 1을 참조하면, 토큰 버킷(120) 기법의 경우 네트워크의 영향으로 인해 수신측으로 불규칙하게 들어오는 트래픽에 대해서 보정해 주는 역할을 한다. 예를 들어, 군집 패킷이 수신측으로 유입된다고 가정했을 때, 먼저 임시 버퍼(Temporary buffer)(110)에 저장되고, 그 후에 토큰 버킷(120) 기법을 통해 한 패킷에 대해 하나의 토큰을 소비함으로써 제한된 토큰의 개수 내에서의 패킷만을 허용하게 된다. 이와 같은 상황에서는 군집 패킷들로 인한 수신측 재생 버퍼(130)의 포화상태를 방지할 수 있을 것이다. 여기에 추가적으로 재생버퍼(Playout buffer)(130)의 오버플로우(Overflow)나 언더플로우(Underflow)를 방지하기 위해 AMP(Adaptive Media Playout) 기법(140)을 적용하여 재생 버퍼(130)에 있는 패킷량에 따라 재생 속도를 변경하게 된다.

[0035] 보다 상세히 설명하면, 수신측으로 들어오는 비디오 데이터 트래픽 패턴(Data traffic pattern)은 네트워크의 혼잡 및 지연 지터에 의해 불규칙하게 들어오는 형태라고 가정한다. 이러한 불규칙한 데이터의 유입은 먼저 임시 버퍼(110)에 저장된다. 그 후 토큰 버킷(120) 기법을 적용하여 재생 버퍼(130)로 들어가는 비디오 패킷의 양을 제어하게 된다.

[0036] 토큰 버킷(120)은 비디오 패킷이 전송될 때마다 하나의 토큰을 제거하게 된다. 그러므로 토큰 버킷(120)의 토큰 카운터가 0이 되면 수신측 재생 버퍼(130)에서는 비디오 패킷을 받을 수 없다. 또한 초당 생성되어 버킷에 추가되는 토큰은 버킷의 최대 용량을 넘을 수 없다. 토큰이 저장되어 있는 버킷의 용량을 c , 초당 생성되는 토큰의 수를 r 이라고 할 때, 임의의 시간 간격 Δt 동안에 재생 버퍼(130)로 들어올 수 있는 비디오 패킷의 최대 수 (N_{max})와 토큰 버킷의 최대 평균 속도 (AV_{max})는 다음 수학적 식 1과 수학적 식 2와 같이 표현된다.

수학적 식 1

[0037]
$$N_{max} = r \times \Delta t + c$$

수학적 식 2

[0038]
$$AV_{max} = \frac{N_{max}}{\Delta t}$$

[0039] 수학적 식 1과 수학적 식 2를 통하여 토큰 버킷(120)을 사용함으로써, 네트워크의 평균 패킷 속도를 제한할 수 있음을 알 수 있다.

[0040] 전송하기에 충분한 패킷이 임시 버퍼(110)에 저장되어 있다고 가정할 경우에 전송되는 패킷의 최대 크기 (P_{txmax})에 따라, 토큰 생성에 따른 유입 가능한 최대 패킷량($P_{tokenmax}$)와 버킷 용량에 따른 유입 가능한 최대 패킷량($P_{bucketmax}$)은 다음 수학적 식 3과 수학적 식 4와 표현될 수 있다.

[0041] 또한 재생량($L_{playout}$)은 단위 시간당 재생 버퍼(130)에서 재생되어 나가는 패킷의 양(ΔP)으로 계산되며, 다음 수학적 식 5와 같다.

수학적 식 3

[0042]
$$P_{tokenmax} = P_{txmax} \times r$$

수학식 4

[0043] $P_{bucketmax} = P_{tmax} \times c$

수학식 5

[0044] $L_{payout} = \Delta P \times \Delta t$

[0045] 재생 버퍼(130)가 비어있다고 가정할 경우, 다음 수학식 6과 같이 단위시간 Δt 동안에 수신측 재생 버퍼로 유입되는 패킷의 양은, 재생 버퍼의 최대 용량(Bmax)에 재생량(Lpayout)을 합한 값보다 적거나 같아야 버퍼의 포화 상태를 방지할 수 있다.

수학식 6

[0046] $(P_{tokenmax} \times \Delta t) + P_{bucketmax} \leq B_{max} + L_{payout}$

[0047] 또한 다음 수학식 7과 같이 단위 시간 동안 토큰 생성에 따른 유입 가능한 패킷이 재생량보다 적을 경우, 재생 버퍼(130) 안에 존재하는 패킷량이 감소함으로써 언더플로우 발생의 가능성이 있으며, 반대로 다음 수학식 8과 같이 큰 생성에 따른 유입 가능한 패킷이 재생량보다 클 경우, 오버플로우 발생의 가능성이 있다.

[0048] 재생 버퍼(130)에서 오버플로우가 발생할 경우 토큰에 의해 유입되는 패킷 또한 폐기될 수 있다. 따라서 적절한 r값을 찾고 그에 따른 버킷의 용량 파라미터를 설정하게 된다.

수학식 7

[0049] $(P_{tokenmax} \times \Delta t) \leq L_{payout}$

수학식 8

[0050] $(P_{tokenmax} \times \Delta t) \geq L_{payout}$

[0051] 이 토큰 버킷 기법을 사용하여 수신측 재생 버퍼(130)로 들어가는 비디오 패킷의 양을 제어하기 위해 두 개의 파라미터(r, c)는 전송되는 패킷의 최대 크기(Prxmax)와 재생 버퍼(130)의 최대 용량(Bmax), 그리고 재생량(Lpayout) 등을 고려하여 설정해야 한다.

[0052] 토큰 버킷을 거쳐 재생 버퍼(130)로 들어온 패킷들에 대해서는 2차적으로 언더플로우 및 오버플로우를 방지하기 위해 AMP 기법(140)을 적용한다.

[0053] 재생 버퍼(130)가 고갈되거나 포화되는 것을 방지하기 위해 언더플로우 임계값(Underflow Threshold)과 오버플로우 임계값(Overflow Threshold)을 정하고, 재생 버퍼(130)의 패킷량을 측정하여 해당 임계값에 도달했을 때, AMP 기법(140)에서는 프레임 재생의 속도를 적응적으로 변화시키게 된다(Playout rate control).

[0054] AMP 기법(140)에서 언더플로우일 경우에 변경하는 재생 속도(Vunderflow)는 다음 수학식 9와 같이 표현되며, 오버플로우일 경우에 변경하는 재생 속도(Voverflow)는 다음 수학식 10과 같이 표현된다.

수학식 9

[0055]
$$V_{underflow} = V_{normal} - V_{normal} \times C_{payout}$$

수학식 10

[0056]
$$V_{overflow} = V_{normal} + V_{normal} \times C_{payout}$$

[0057] 여기에서, C_{payout}는 재생 속도 변경율을 의미한다. 예를 들어, 원래의 재생 속도를 30 fps로 설정하고, AMP 기법 적용 시 재생 속도 변경율을 ±20%로 설정한 경우, 실시간으로 재생 버퍼의 패킷량을 측정하여 언더플로우 임계값 이하로 내려간 경우에는 언더플로우 발생의 위험이 있으므로 재생 속도를 30fps - (30fps × 20/100) = 24 fps로 감소시켜 재생하여, 재생 버퍼(130)안에 머무는 패킷량을 증가시켜 언더플로우의 위험을 피한다. 따라서, 재생 버퍼(130) 내의 비디오 패킷의 부족으로 인한 재생의 멈춤 현상을 방지할 수 있다.

[0058] 반대로 버퍼의 패킷량이 오버플로우 임계값 이상인 경우, 오버플로우 발생의 위험이 있으므로 재생 속도를 30fps + (30fps × 20/100) = 36fps로 증가시켜 재생하여 재생 버퍼(130)에 머무는 비디오 패킷을 더 많이 꺼냄으로써 버퍼의 패킷량을 감소시켜 포화 상태에 따른 오버플로우 위험을 피한다. 이와 같은 조치에 따라 패킷의 손실을 막을 수 있으며 정상적인 복호화로 화질의 열화 및 재생의 스킵(skip) 현상을 방지할 수 있다.

[0059] 본 발명의 시뮬레이션을 수행하기 위하여, 송신측에서 전송되어 수신측 재생 버퍼(130)에 들어오는 비디오 패킷이 네트워크의 혼잡 및 예상치 못한 지터에 의해 불규칙적인 패턴으로 유입된다고 가정한다. 이러한 환경에서 토큰 버킷 기법과 AMP 기법을 적용하고, 언더플로우와 오버플로우의 발생 빈도, 정상 재생 속도로 재생된 프레임의 비율, 패킷 손실률 및 수신측 영상의 PSNR(Peak Signal to Noise Ratio)에 대해 NS-2와 JSVM을 사용하여 성능분석을 진행하기로 한다.

[0060] 여기에서 PSNR이란 동영상의 화질 정보를 수치적으로 표현한 값으로 두 영상에 대해 평균 수치를 확인함으로써 대략적인 화질의 차이를 알 수 있는 수치이다. PSNR을 계산하는 식은 수학식 11과 같다.

수학식 11

[0061]
$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

[0062] 여기서 MAX_I는 해당 영상의 최대 신호값이다. MSE(Mean squared error)는 오차 제곱의 평균이다. 원영상과 수신된 영상에 대해 PSNR을 구할 경우, 두 영상에 대한 오차가 크면 클수록 PSNR값은 낮아진다. 일반적으로 PSNR값이 30dB이 넘으면 두 영상의 차이를 눈으로 구분하기 어렵다.

[0063] 도 2는 본 발명의 일 실시예에 따른 시뮬레이션 절차를 보여주는 도면이다.

[0064] 도 2의 시뮬레이션은 NS-2에서의 트레이스(trace) 기반의 시뮬레이션으로 진행된다. 여기서 비디오 트레이스(trace)는 실제 비디오 스트림으로부터 중요한 특징들을 추출하여 텍스트 파일의 형식으로 저장한 후, 그 데이터를 바탕으로 NS-2를 통해 전송하여 시뮬레이션 한다. 이 비디오 트레이스 파일은 패킷의 전송시간 및 도착시간, 프레임의 길이 및 프레임의 부호화 유형 등 실제 비디오 스트림으로부터 추출되어 저장된 형태이다.

[0065] 이러한 일련의 과정들은 MyEvalSVC를 참조하여 진행하였다. 이 시뮬레이션 방법은 SVEF(Scalable Video-Streaming Evaluation Framework)에 기반을 두어 NS-2 시뮬레이션 환경으로 프레임워크를 확장한 것이며, 제안하는 네트워크 구조 또는 프로토콜을 SVC, H.264에서 평가할 수 있는 방법이다.

[0066] 시뮬레이션의 대상이 될 원시 YUV 비디오를 JSVM을 이용하여 파라미터를 조정하여 인코딩을 하게 된다. 인코딩

후 JSVM에서 제공하는 BitStreamExtractor와 SVEF에서 제공하는 F-N Stamp를 사용하여 비디오 트레이스(Video trace) 파일을 생성하게 된다. NS-2에서는 myEvalSVC 에이전트(Agent)를 사용하여 이 전송 비디오 트레이스(Video trace) 파일을 기반으로 하여 자동으로 해당되는 패킷들을 생성해주게 된다.

- [0067] 수신단 측에서는 myEvalSVC_Sink 에이전트(Agent)를 구축해야 하며, 이 에이전트(Agent)는 전송된 패킷들을 바탕으로 수신 정보를 트레이스(trace) 파일로 다시 저장하게 된다(Received Video trace file). 이 수신단에서 기록된 트레이스(trace) 파일에는 수신 시간, 패킷 사이즈, 프레임 넘버, 프레임 유형 등이 기록되어 있다. 이 기록된 수신 트레이스(trace) 파일을 바탕으로 재생의 불연속성 및 패킷 손실을 등을 계산할 수 있으며, 이 데이터를 바탕으로 SVEF에서 제공하는 NALU 필터(filter)를 거치고 JSVM에서 다시 디코딩을 하게 되면 PSNR 스테틱(Static) 툴(tool)을 이용한 PSNR측정 및 플레이어를 이용한 화질을 볼 수 있다.
- [0068] NALU 필터(filter)에서는 프레임 중속성으로 인하여 디코딩할 수 없는 프레임과 너무 늦게 도착한 프레임들을 폐기하는 역할을 하게 되며, 이 필터링된 데이터를 디코딩하는 것을 실질적으로 네트워크를 거쳐 최종적으로 수신된 비디오 파일이라고 할 수 있다.
- [0069] NS-2에서는 전송자 하나와 수신자 하나를 만들어 시뮬레이션하게 된다. 송신측에서는 위에서 설명한 myEvalSVC를 이용하여 비디오 트레이스(Video trace) 파일을 바탕으로 한 패킷을 생성하고, 네트워크의 혼잡과 불규칙한 지터의 영향을 가정하여 비디오 데이터 패킷을 불규칙하게 전송한다. 수신단에는 토큰 버킷 기법을 수행하는 시스템과 2개의 버퍼를 구현하고, 재생 버퍼에는 AMP 기법을 적용 시키게 된다.
- [0070] 시뮬레이션에서 언더플로우 시 수신 트레이스 기록 방법은 다음과 같다.
- [0071] 수신측 재생 버퍼에서 재생이 이루어졌을 경우, 각 패킷들에 대한 재생시간을 수신측 트레이스(trace) 파일에 추가적으로 기록되도록 한다. 시뮬레이션 진행 중 만약 언더플로우 현상이 일어난다고 하면, 버퍼에 재생하기에 충분한 양의 데이터가 없기 때문에 재생의 멈춤 현상이 일어나게 된다. 따라서 수신측 에이전트(Agent)에서 기록되는 트레이스(trace) 파일에는, 언더플로우 발생 시점에 해당하는 패킷에 대해서 해당 프레임의 원 재생시간이 아닌 다음번 재생시간에 대해서 기록될 것이고, 이것을 바탕으로 언더플로우에 의한 재생의 불연속성을 측정한다.
- [0072] 시뮬레이션에서 오버플로우 시 수신 트레이스(trace) 기록 방법은 다음과 같다.
- [0073] 오버플로우 현상이 해소되지 않고 버퍼가 포화상태일 때, 그 이후로 들어온 패킷에 대해서는 손실 처리를 하게 된다. 따라서 해당 패킷에 대해서는 수신 트레이스(trace) 파일 기록 시 해당 패킷을 기록하지 않으며, NALU 필터(filter)를 거쳤을 때 폐기를 시킬 수 있도록 한다. 즉, 오버플로우 현상을 겪은 패킷은 정상적으로 디코딩이 되지 않으며 그에 중속 특성을 가진 다른 프레임 또한 정상적인 디코딩이 이루어지지 않는다.
- [0074] 시뮬레이션에서 NALU 필터(filter)의 활용법은 다음과 같다.
- [0075] myEvalSVC를 이용한 시뮬레이션 기법에서 NALU 필터(filter)는 NS-2에서 수신된 트레이스 파일을 입력 파일로 하여 필터링을 하게 된다. 필터(filter)에서는 재생 버퍼의 크기와 처리되는 프레임의 값(fps)을 두 개의 파라미터로 받게 된다. 즉, 재생 버퍼를 가정하며 너무 늦게 도착한 프레임이나 네트워크 전송 시 손실로 인해 프레임 중속성으로 디코딩할 수 없는 프레임을 폐기시킨다. 그러나 본 시뮬레이션에서 이러한 기능 부분은 NS-2에 직접적으로 수신측 재생버퍼를 구현하여 시뮬레이션하고 있으므로, 여기서 NALU 필터(filter)의 파라미터는 NS-2에 구현된 수신측 재생 버퍼의 크기, 재생 속도와 같은 값을 주게 되며, 활용법은 JSVM 디코더를 거치기 전 수신 트레이스 파일의 형식을 바꾸는 역할과, 수신측 재생 버퍼 포화 상태 이후에 들어온 패킷에 대해 폐기를 시키는 용도로만 사용하게 된다.
- [0076] 도 3은 본 발명의 일 실시예에 따른 시뮬레이션에 사용된 YUV 파일을 나타낸 표이다. 본 발명에서의 시뮬레이션은 NS-2와 JSVM을 사용하여 진행하고, 시뮬레이션 대상이 된 비디오 파일은 YUV 형식으로 JSVM 인코더를 사용하며, 도 3과 같이 구성되어 있다.
- [0077] foreman 영상 중 하나는 352X288 크기의 foreman_cif 영상이 있으며 704X288의 크기를 가지는 보다 하이 레졸루션(high resolution) 영상인 foreman_2cif 영상이 있다. 또한 foreman과는 영상 내 오브젝트(object) 들의 움직임이 다른 soccer_cif영상을 가지고 시뮬레이션을 진행한다.
- [0078] 시뮬레이션에서 직접적으로 추가적인 노드를 생성하여 혼잡의 영향을 주지 않고, 단지 네트워크의 혼잡이나 지

터의 영향을 가정하여 송신자에서 트래픽 발생을 불규칙적으로 스케줄링하도록 설정하였다. 시뮬레이션 파라미터는 도 4와 같다.

- [0079] 도 4는 본 발명의 일 실시예에 따른 시뮬레이션 파라미터를 나타낸 표이다.
- [0080] 한 패킷의 최대 사이즈는 1500 바이트(byte)이며, 전송된 패킷들을 수신측 재생버퍼에 저장하였다가 30fps로 재생하게 된다. 토큰 버킷의 파라미터 중 초당 생성되는 토큰의 수와 버킷이 저장할 수 있는 토큰의 최대 개수(버킷의 용량)는 재생 버퍼의 크기(30000 바이트)를 고려하여 수학적 식 6을 사용하여 설정하였다. 수학적 식 6에 대입하여 계산하면 r 과 c 를 더한 토큰의 개수를 정할 수 있다.
- [0081] 토큰 버킷 기법 사용시 초당 토큰에 의해 재생 버퍼로 유입되는 트래픽의 크기가 초당 재생되는 비디오 데이터의 크기보다 작으면 언더플로우의 위험이 커지며, 반대로 초당 재생되는 양보다 많이 들어오게 되면 언더플로우의 위험성이 커진다. 따라서 토큰버킷에 의해 초당 제어되는 유입 트래픽 크기는 초당 재생되는 양과 비슷한 것이 이상적이다. 또한, 송신자에서 생성되어 전송되는 비디오 데이터패킷의 양은 각각 다른 크기를 가지고 있으며, 불규칙적이기 때문에 테스트 시뮬레이션을 통하여 초당 생성되는 토큰의 개수 r 을 정하였으며, 이는 도 5와 같다.
- [0082] 도 5는 본 발명의 일 실시예에 따른 시뮬레이션에서의 토큰 버킷 파라미터에 따른 성능을 나타낸 표이다.
- [0083] 시뮬레이션에서 AMP 기법 적용 시 적응적으로 변경하는 재생속도 변경을 파라미터는 총 세 가지로 $\pm 10\%$, $\pm 20\%$, $\pm 30\%$ 로 변경해 가며 진행하였다. 수신 트레이스 파일에서는 각 패킷별로 송신 시간, 수신 시간, 재생 시간 및 해당 프레임의 정보를 수신 트레이스(trace) 파일에 기록하게 되고, 이 수신 트레이스 파일을 바탕으로 성능 분석을 실시한다.
- [0084] 도 6은 본 발명의 일 실시예에 따른 비디오 스트리밍 시뮬레이션에서 네 가지의 시나리오를 나타낸 표이다.
- [0085] 도 6에서 보는 바와 같이, 본 발명에서는 성능분석 시 토큰 버킷 기법과 AMP 기법의 결합을 통한 비디오 스트리밍의 성능 분석을, 총 4가지의 시나리오를 통하여 진행하였다. 케이스1의 경우 아무 기법도 사용하지 않은 경우이며 케이스2의 경우 토큰 버킷 기법만 적용한 경우이고, 케이스3은 토큰 버킷 기법은 사용하지 않고 AMP 기법만 적용한 경우이고, 케이스4의 경우 본 발명에서 제안하는 토큰 버킷 기법과 AMP 기법을 모두 적용한 경우이다.
- [0086] 도 7은 수신측 버퍼에 유입되는 비디오 스트리밍 패킷 크기의 패턴에 대해 도시한 그래프이다. 도 7은 비디오 중 foreman_cif의 비디오 데이터를 송신했을 때, 수신측으로 유입되는 비디오 스트리밍 패킷 크기의 패턴에 대해 나타낸 그래프이다. 특히, 버퍼의 오버플로우가 일어나는 구간 중 1초 동안(1.5 ~ 2.5 초)의 구간에 대해 토큰 버킷을 사용하지 않았을 경우(케이스1, 케이스3)와 토큰버킷을 사용하여 유입되는 트래픽에 대해 성형을 거친 경우(케이스2, 케이스4)에 대해 나타내었다.
- [0087] 도 7에서 토큰 버킷을 사용하지 않은 경우에는 비교적 패킷들이 불규칙하게 유입되며, 군집 트래픽 또한 그대로 버퍼로 유입되는 것을 볼 수 있다.
- [0088] 토큰 버킷을 사용했을 경우에는 패킷들이 일정한 간격을 두고 토큰에 따라 규칙적으로 버퍼에 들어온다. 다른 비디오인 foreman_2cif와 soccer_cif의 경우도 도 7과 마찬가지로 토큰 버킷기법을 사용하였을 때 패킷들이 비교적 일정한 간격을 두고 수신측 재생 버퍼로 성형되어 들어오는 결과를 나타낸다. 시뮬레이션을 통한 성능분석은 도 6에서 소개된 네 가지의 케이스 들에 대하여 수행하였다. 각각의 케이스에서 언더플로우와 오버플로우 발생 횟수, 패킷 손실 비율, 정상 재생 속도에 의한 프레임율 그리고 전송 완료 시 결과 영상에 대한 PSNR값을 비교해 보면 다음과 같다.
- [0089] 도 8은 각각의 시뮬레이션 대상 비디오 데이터에 대해 케이스 1과 케이스 2에서 언더플로우와 오버플로우의 발생 횟수를 비교한 그래프이다.
- [0090] 도 8은 각각의 시뮬레이션 대상 비디오 데이터에 대해 케이스 1과 케이스 2에서 언더플로우와 오버플로우의 발생 횟수를 비교한 그래프이다.
- [0091] 도 8으로부터 foreman_cif 영상의 경우 토큰 버킷을 사용하지 않았을 경우에는 언더플로우가 5회, 오버플로우가 159회 일어났으며, 토큰 버킷을 사용한 경우에는 언더플로우가 5회, 오버플로우가 0회로 많은 성능 차이를 보인다. 다른 영상들인 foreman_2cif와 soccer_cif의 경우도 마찬가지로 케이스1보다 케이스2에서 더 좋은 성능을 나타낸다는 것을 확인할 수 있다. 결국 AMP 기법을 사용하지 않고도 단지 토큰 버킷만을 사용함으로써 불규칙적

으로 들어오는 패킷들에 대해 보정하고 불연속 횡수를 많이 낮출 수 있는 것을 알 수 있다. 특히 토큰 버킷 기법을 사용했을 경우에는 오버플로우 현상 해소에 대해 큰 효과를 나타낸다는 것을 볼 수 있다.

- [0092] 도 9는 각각의 시뮬레이션 대상 비디오 데이터에 대해 케이스 3과 케이스 4에서 언더플로우와 오버플로우의 발생 횡수를 비교한 그래프이다. 도 9는 케이스 3과 케이스 4에서 언더플로우와 오버플로우의 발생 횡수를 비교한 도면이다. 도 9에서 아래쪽 10%, 20%, 30%는 AMP 동작시 재생 속도를 원래 재생 속도보다 얼마만큼 더 빠르거나 느리게 재생했는지에 대한 재생 속도 변경율을 의미한다.
- [0093] 도 9를 참조하면, 케이스 3은 토큰 버킷 기법을 적용시키지 않고 AMP 기법만 적용한 경우이다. 케이스3의 경우에 대체적으로 재생 속도 변경율을 10%정도만 조절했을 경우엔 높은 언더플로우 횡수와 오버플로우 발생 횡수를 나타내고 있고, 재생 속도 변경율을 20%와 30%로 각각 더 높은 변경율을 적용하였을 때에 10%로 설정했을 경우보다 좀 더 좋은 성능을 내는 것을 확인할 수 있다. 하지만 케이스3의 경우 토큰기법의 결합 없이 AMP기법만 사용한 경우로서 soccer_cif 영상 같은 경우는 30% 변경율을 적용하였음에도 불구하고 7회의 언더플로우와 69회의 오버플로우로 안 좋은 환경을 보인다.
- [0094] 반면에 케이스 4 경우는 토큰 버킷 기법과 AMP 기법을 둘 다 적용한 경우로서, 토큰 버킷을 사용한 경우에는 오버플로우를 해소할 수 있었기 때문에 모든 경우에서 오버플로우는 나타나지 않았으며, 재생 속도 변경율을 20%까지 조절했을 경우 soccer_cif의 영상은 언더플로우와 오버플로우 모두 해소되며, 30%까지 조절했을 경우에는 foreman_cif의 영상 또한 재생의 불연속성을 야기시키는 현상들을 모두 해소할 수 있는 것을 확인할 수 있다. 즉, 미디어 재생에 연속성이 보장된다는 것을 확인할 수 있다. foreman_2cif 영상 또한 케이스4에서 제일 좋은 성능을 보이고 있다.
- [0095] 도 10은 시뮬레이션 대상이 된 각 비디오에 대해 케이스 별로 언더플로우 및 오버플로우의 지속 시간을 나타낸 표이다. 도 10은 시뮬레이션 대상이 된 각 비디오에 대해 케이스1부터 케이스4까지 언더플로우와 오버플로우 성능을 분석했을 때, 각 현상들이 얼마나 지속되었는지에 대해 시간에 따라 나타낸 표이다.
- [0096] 도 8 및 도 9에서 본 바와 같이, 토큰 버킷 기법으로 인해 오버플로우를 해결한 경우, 오버플로우에 대한 지속 시간은 0msec로 표기되며, 오버플로우 지속시간과 그에 따른 횡수는 비례하는 모습을 보인다. 또한 언더플로우의 지속시간으로 보아 수신된 각 비디오들이 재생이 되었을 경우 얼마 정도의 영상의 멈춤이 일어나는지를 다른 영상들과 비교하여 비율적으로 알 수 있다.
- [0097] 도 11은 케이스 3과 케이스 4에서 원래의 정상 재생 속도로 재생된 프레임의 비율 결과를 도시한 그래프이다.
- [0098] 도 11은 케이스 3과 케이스 4에서 원래의 정상 재생 속도로 재생된 프레임의 비율 결과를 보여주고 있으며, 계산식은 수학적 식 12와 같다.

수학적 식 12

$$F_{normal} = \frac{N_{normal}}{N_{total}}$$

- [0099] 여기서 F_{normal} 는 원래의 정상 재생 속도에 의한 재생 프레임 비율을 의미하고, N_{normal} 은 원래의 정상 재생 속도에 의한 재생 프레임의 수를 의미하고, N_{total} 은 원래의 정상 재생 속도와 변경된 재생 속도에 의한 전체 재생 프레임의 수를 나타낸다.
- [0100] 도 11을 참조하면, foreman_cif와 foreman_2cif에서 Case 4의 경우가 Case 3보다 원래의 정상속도로 재생되는 프레임의 비율이 각각의 재생 속도 변경율에서 더 높은 것을 알 수 있다. soccer_cif의 경우 10%와 20%일 때 Case4의 정상 재생 속도에 의한 재생 프레임 비율이 Case3보다 적으나 언더플로우 해소에서 높은 성능을 나타내며, 30% 일 때는 정상 속도 재생 비율이 더 높아지는 것을 볼 수 있다. 결과적으로, 토큰 버킷 기법과 AMP 기법 적용을 둘 다 사용하는 경우가 토큰 버킷 기법 없이 AMP 기법만 사용하는 경우보다 버퍼 내의 데이터 량이 더 안정적이며 정상적인 속도로 재생된 프레임의 비율이 대체적으로 높음을 알 수 있다.
- [0101] 도 12는 각 시뮬레이션 결과에서 패킷 손실율을 도시한 그래프이다. 도 12에서는 각 시뮬레이션 결과에서 패킷 손실이 얼마나 일어났는지에 대해 나타낸 그래프이다.
- [0102] 도 12에서 Case2와 Case4의 경우 모든 비디오에 대해서 토큰 버킷을 사용하여 오버플로우로 인한 패킷의 손실을

방지했기 때문에 실질적인 패킷의 손실은 Case1과 Case3에서 나타났다. 패킷의 손실은 버퍼가 포화상태에 이르러 오버플로우 현상이 일어났을 경우에 발생할 수 있으며 프레임의 복호화에 심각한 영향을 주게 된다.

[0104] 도 12에서 아무 기법도 사용하지 않은 Case1의 경우가 제일 높은 패킷 손실율을 보이고 있으며, AMP기법을 사용한 Case3의 경우는 재생 속도 변경률을 높게 조절할수록 한번에 재생되는 데이터의 양을 크게 조절할 수 있으므로, 더 낮은 손실율을 보이는 것을 알 수 있다. 패킷손실의 결과는 영상의 프레임이 손상될 수 있다는 것을 의미하기 때문에 AMP 기법뿐만 아니라 토큰 버킷 기법을 함께 사용함으로써 패킷 손실 문제는 해결되어야 한다.

[0105] 도 13 내지 도 15는 각 영상에 대해 모든 케이스에서 수신된 비디오 스트리밍 패킷에 대해 PSNR을 측정한 그래프이다.

[0106] 도 13은 forman_cif 영상에 대한 PSNR을 도시한 그래프이고, 도 14는 foreman_2cif 영상에 대한 PSNR을 도시한 그래프이고, 도 15는 soccer_cif 영상에 대한 PSNR을 도시한 그래프이다.

[0107] 도 13은 3개의 영상에 대해 모든 케이스에서 수신된 비디오 스트리밍 패킷에 대해 PSNR을 측정한 그래프이다.

[0108] 케이스 2와 케이스 4에서는 토큰 버킷 기법을 적용하여 오버플로우가 발생하지 않아 패킷의 손실이 없고 모든 프레임에 대해 정상적인 복호화가 이루어진 경우이며, 케이스 1과 케이스 3에서는 버퍼의 포화 현상으로 인하여 몇몇 프레임들의 손상이 일어난 경우이다. 물론 케이스 2의 경우는 언더플로우가 발생하기 때문에 영상 재생에 있어 멈춤 현상이 일어날 수 있다. 두 기법 모두 적용한 경우에는 300프레임 모두 정상적으로 재생되었지만, 그렇지 못한 경우에는 프레임의 손실 및 중속적인 특징으로 인해 다른 상위 프레임까지 복호화가 제대로 이루어지지 않아, 특정 개수의 프레임만이 재생되는 영상의 손실 현상을 보이며, 화질의 열화가 일어난다는 것을 알 수 있다.

[0109] 이상에서 본 발명의 무선 통신 환경에서 비디오 스트리밍의 버퍼 언더플로우와 오버플로우 문제를 해결하기 위한 여러 기법 중 하나인 AMP 기법을 적용하고, 추가로 토큰 버킷 기법을 적용하여 비디오 스트리밍에 대한 성능 분석을 하였다. 본 발명에서는 불규칙적으로 들어오는 트래픽들에 대해 제어하고, 오버플로우를 방지하기 위하여 토큰 버킷 기법을 사용하였다. 또한 재생 버퍼에는 AMP 기법을 적용시켰으며, 버퍼의 패킷량에 따른 비디오 패킷의 포화나 고갈을 방지하기 위해 재생 버퍼량을 기준으로 두 개의 임계치를 적용하고 AMP 동작시 재생 속도 변경률을 여러 가지로 바꾸어 연속성에 대한 성능분석을 수행하였다. 결과적으로 AMP 기법과 토큰 버킷 기법을 함께 사용했을 경우에는 다른 경우들 보다 우수한 성능을 나타내었으며 30%의 재생 속도 변경률을 적용하였을 때에는 재생의 멈춤이나 스킵을 대부분 방지할 수 있다는 것을 알 수 있다.

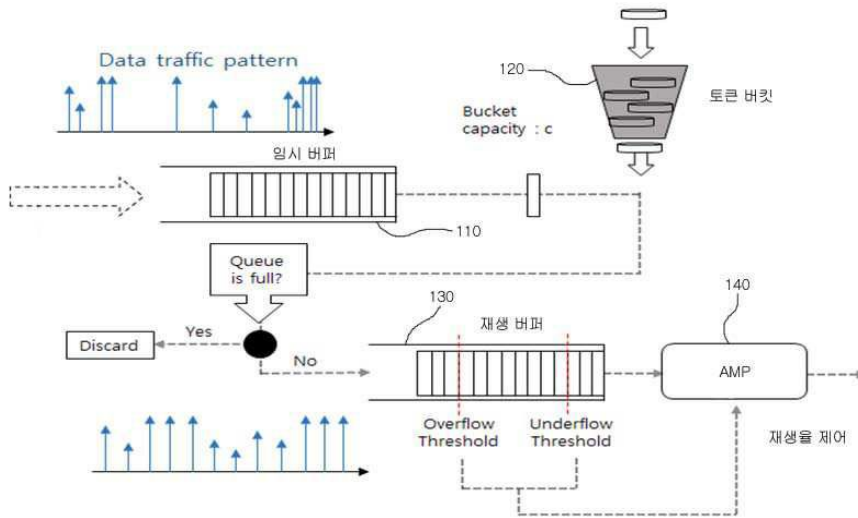
[0110] 이상 본 발명을 몇 가지 바람직한 실시예를 사용하여 설명하였으나, 이들 실시예는 예시적인 것이며 한정적인 것이 아니다. 본 발명이 속하는 기술분야에서 통상의 지식을 지닌 자라면 본 발명의 사상과 첨부된 특허청구범위에 제시된 권리범위에서 벗어나지 않으면서 다양한 변화와 수정을 가할 수 있음을 이해할 것이다.

부호의 설명

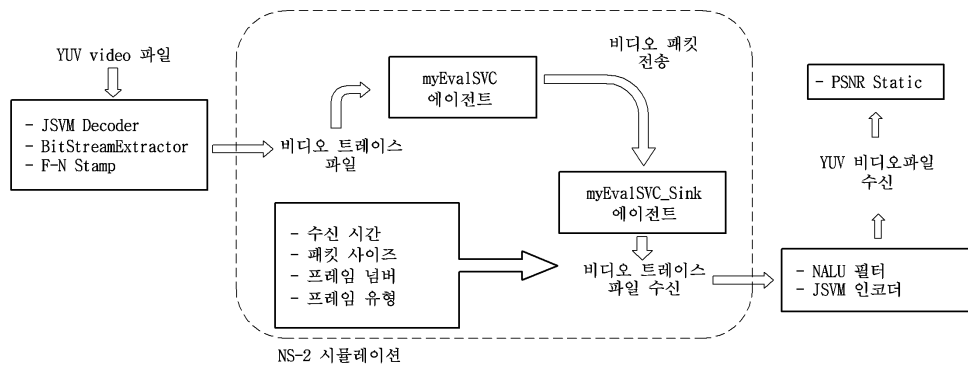
[0111]	110 임시 버퍼	120 토큰 버킷 기법
	130 재생 버퍼	140 AMP 기법

도면

도면1



도면2



도면3

	foreman_cif	foreman_2cif	soccer_cif
Frame number	300	300	300
GOP size	8	8	8
Frame rate	30	30	30
Video resolution	CIF(352 X 288)	2CIF(704 X 288)	CIF(352 X 288)
Total video packet size	518146byte	670675byte	643181byte

도면4

Used simulation tool	NS-2, JSVM		
Node number	Video sender 1, Receiver 1		
Token generation rate (r) for Token Bucket	foreman_cif	foreman_2cif	soccer_cif
	5 / sec	6 / sec	5 / sec
Bucket capacity (c) for Token Bucket	18	18	19
Maximum packet size (Ptxmax)	1,500 Byte		
Buffer size (Bmax)	30,000 Byte		
Normal Playout speed (Vnormal)	30 fps	30 fps	30 fps
	(5182 Byte/0.1s)	(6707 Byte/0.1s)	(6432 Byte/0.1s)
Underflow threshold for AMP	30 %		
Overflow threshold for AMP	70 %		
Playout speed change rate for AMP (Cplayout)	±10 %, ±20 %, ±30 %		

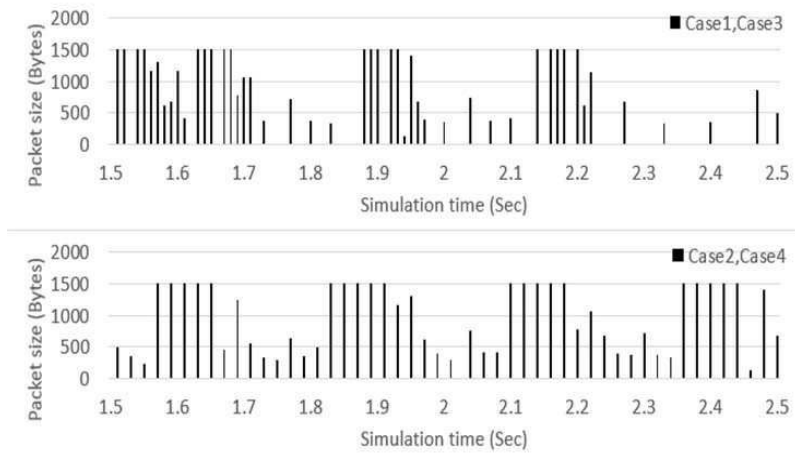
도면5

	Total token num (r과 c를 더한 값)	r / c	Underflow num	Overflow num
foreman_cif	23	r = 5, c = 18	5	0
		r = 4, c = 19	30	0
foreman_2cif	24	r = 5, c = 19	21	0
		r = 6, c = 18	4	0
soccer_cif	24	r = 5, c = 19	10	0
		r = 6, c = 18	0	169

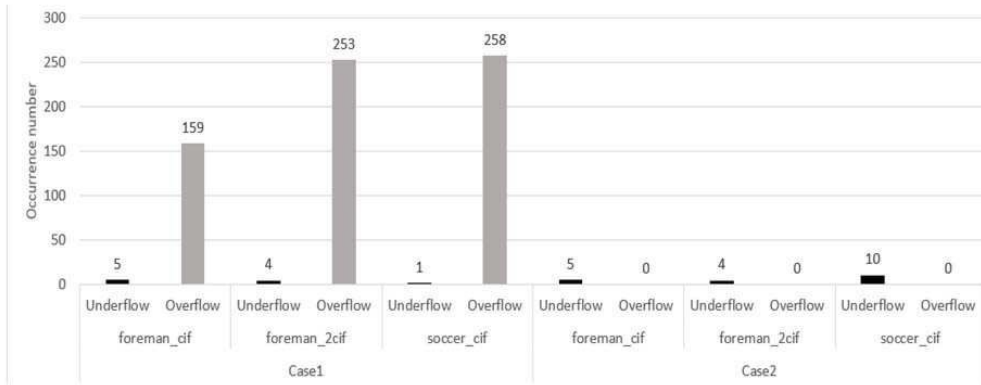
도면6

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

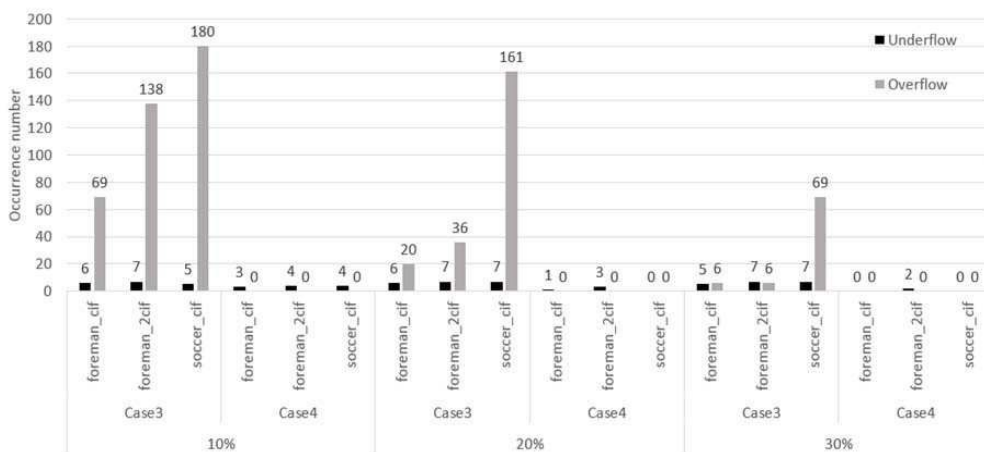
도면7



도면8



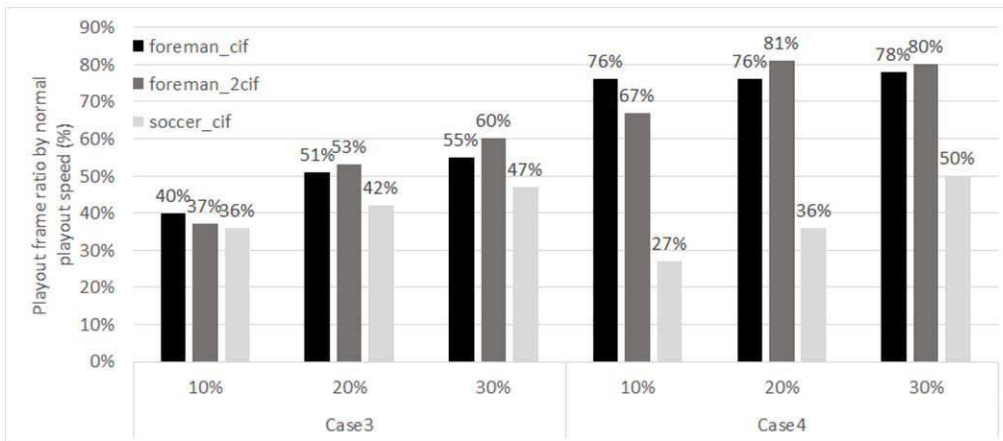
도면9



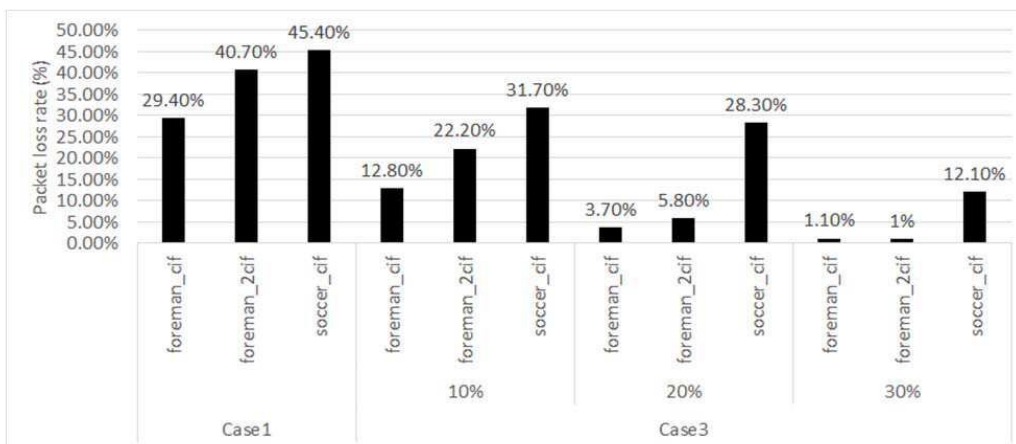
도면10

단위 : msec	foreman_cif		foreman_2cif		soccer_cif		
	Under time	Over time	Under time	Over time	Under time	Over time	
Case1	500	2441	400	3530	100	4211	
Case2	500	0	400	0	1000	0	
Case3	10%	600	998	700	1557	500	2723
	20%	600	233	700	372	700	2195
	30%	500	31	700	42	700	812
Case4	10%	300	0	400	0	400	0
	20%	100	0	400	0	0	0
	30%	0	0	200	0	0	0

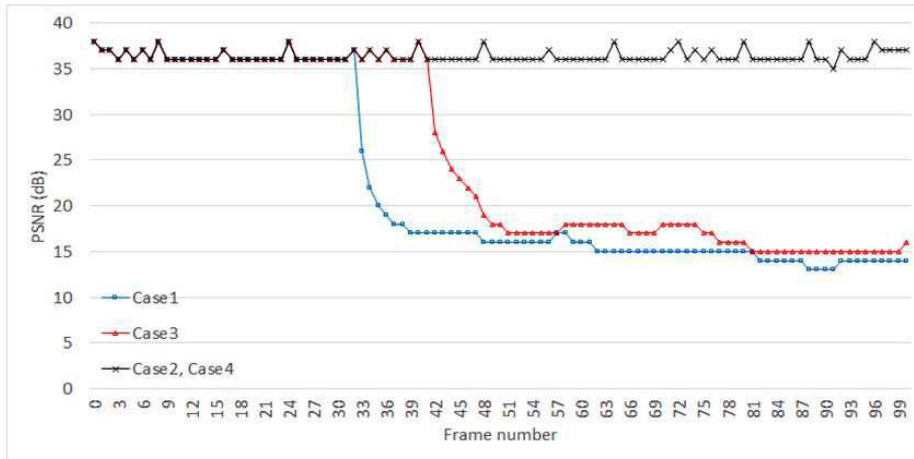
도면11



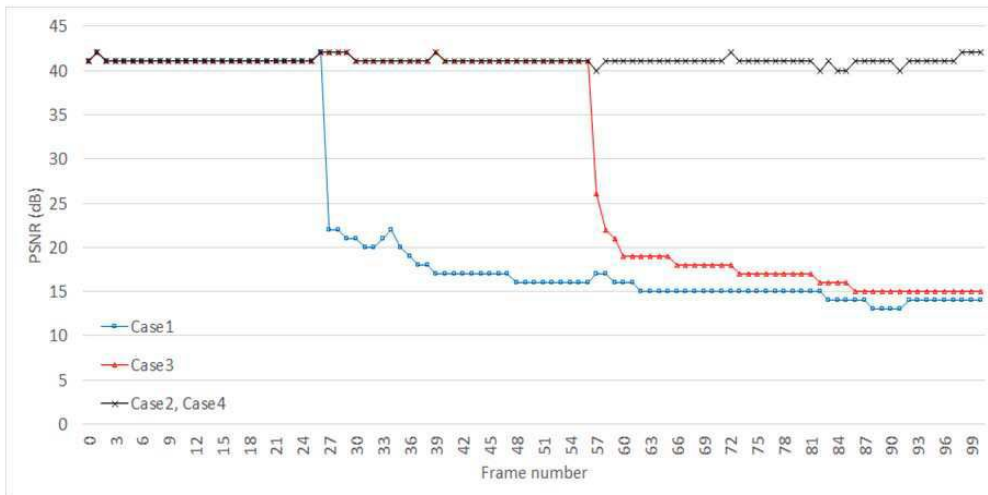
도면12



도면13



도면14



도면15

