

# 영상 스트리밍 서비스를 위한 우선순위 버퍼 혼잡제어 알고리즘

준회원 김 승 훈\*, 최 재 원\*, 정회원 최 승 식\*

## Congestion Control of a Priority-Ordered Buffer for Video Streaming Services

Seung-Hun Kim\*, Jae-Won Choi\* *Associate Members*, Seung Sik Choi\* *Regular Member*

### 요 약

최근 네트워크 기술이 발전하면서 사용자들의 요구가 다양화되고 대용량의 멀티미디어 데이터에 대한 필요성이 증가하고 있다. 실시간성과 동시성이 중요시 되는 멀티미디어 데이터의 전송에는 UDP(User Datagram Protocol) 트래픽이 사용되고 있는데, UDP 트래픽은 현재의 인터넷 기반인 TCP(Transmission Control Protocol) 트래픽과 경쟁 관계에 있어 네트워크의 혼잡을 초래하는 경우가 많았다. 반면 TCP 트래픽은 네트워크의 혼잡제어를 수행하지만 수신자의 관점을 고려하지 않은 점이 있어서 스트리밍 전송에는 적합하지 않은 단점을 가지고 있다. 본 논문에서는 네트워크의 형평성 문제를 해결하고 수신자의 관점에서 RTP/RTCP(Real-time Transport Protocol/Realtime Transport Control Protocol)를 기반으로 하는 네트워크 알고리즘을 제안한다. 네트워크에 우선순위를 적용하여 수신자의 버퍼를 고려한 혼잡제어 알고리즘인 POBA(Priority-Ordered Buffer Algorithm)는 시뮬레이션 결과를 통해 우선 순위가 없는 기존 네트워크에 비해 패킷 손실률과 버퍼 점유도 면에서 스트리밍 전송에 적합한 환경을 제공하고 있음을 확인할 수 있었다.

**Key Words** : Congestion, Buffer, RTP/RTCP, Streaming services

### ABSTRACT

According to the recent development of network technology, the demands of users are diversified and the needs of multimedia traffic are increasing. In general, UDP(User Datagram Protocol) traffic is used to transport multimedia data, which satisfied the real-time and isochronous characteristics. UDP traffic competes with TCP traffic and incur the network congestion. However, TCP traffic performs network congestion control but does not consider the receiver's status. Thus, it is not appropriate in case of streaming services. In this paper, we solve a fairness problems and proposed a network algorithm based on RTP/RTCP(Real-time Transport Protocol/Realtime Transport Control Protocol) in view of receiver status. The POBA(Priority Ordered Buffer Algorithm), which applies priorities in the receiver's buffer and networks, shows that it provides the appropriate environment for streaming services in view of packet loss ratio and buffer utilization of receiver's buffer compared with the previous method.

※ 본 연구는 2006년 인천대학교 교내연구비 지원으로 수행되었습니다.

\* 인천대학교 컴퓨터공학과 무선정보 네트워크 연구실 (winlab.incheon.ac.kr)

논문번호 : KICS2006-08-354, 접수일자 : 2006년 8월 25일, 최종논문접수일자 : 2007년 4월 5일

I. 서론

인터넷이 널리 보급되면서, 인터넷에 대한 일반인의 관심이 증가하고 있다. 그리고 Fast Ethernet, Gigabit Ethernet, ATM (Asynchronous Transfer Mode) 등의 보급을 통해서 그 동안 문제가 되었던 네트워크의 속도가 고속화되고 대역폭이 빠르게 증가함에 따라 컴퓨팅 환경은 과거에 비해서 빠르게 증가함에 따라 컴퓨팅 환경은 과거에 비해서 놀랄 정도로 향상되었다.

네트워크의 성능이 향상되면서 일반 텍스트(Text)나 정지 화상(JPEG) 데이터 뿐만 아니라 대용량의 실시간 동영상 데이터의 요구도 증가되게 되었는데, 최근에는 이런 동영상 데이터 파일을 실시간으로 네트워크에서 전송 받아 출력시키는 스트리밍 기술이 널리 보급되고 있다. 스트리밍 시에 전송하는 멀티미디어 데이터(오디오, 비디오 등)의 경우 대용량이고, 높은 전송 대역폭을 요구하며 실시간으로 재생성을 필요로 하는 특징을 갖고 있어서, 패킷 손실이나 지연, 대역폭 상태에 크게 좌우된다. 하지만 현재의 인터넷의 경우 전송의 신뢰성을 기본으로 하는 TCP(Transmission Control Protocol) 기반으로 하는 네트워크이기 때문에 네트워크 상황에 따른 전송률의 급격한 변화와 높은 손실률에 따른 재전송의 시간 지연으로 스트리밍 서비스에 부적합한 환경을 제공한다. 이에 실시간으로 전송할 수 있는 프로토콜을 대신하여 사용하게 되었는데, UDP(User Datagram Protocol)와 RTP(Real-time Transport Protocol)가 바로 그것이다.<sup>[1-2]</sup>

이런 UDP나 RTP를 이용하는 트래픽들의 경우 TCP에서와 같은 혼잡 기피기법을 사용하지 않기 때문에 같은 경로를 통해 동시에 TCP 트래픽과 경쟁하게 될 때 불공정한 전송결과를 가져올 수 있다. 이런 문제점을 해결하고자 TCP와 유사한 혼잡제어 기법을 이용하여, 네트워크 혼잡을 피하는 많은 연구들이 활발히 진행 중 이다<sup>[3-8]</sup>.

본 논문에서는 II장에서 이런 연구들의 소개와 함께 실시간 스트리밍 프로토콜을 간략히 설명하고 III장에서 네트워크에 접속한 단말간 우선 순위에 따라 혼잡 제어를 달리하는 알고리즘을 제안한다. 그리고 IV장에서 시뮬레이션 결과와 함께 향후 앞으로의 연구 방향을 끝으로 마무리 하겠다.

II. 관련 연구

2.1 혼잡 제어 메커니즘

혼잡 제어 기법으로는 대표적으로 AIMD(Additive Increase Multiplicative Decrease) 알고리즘과 TCP 모델링 방식이 있다<sup>[9-10]</sup>. 먼저, TCP 모델링 방식으로 멀티미디어 트래픽의 혼잡제어를 위해 TCP-friendly 혼잡제어의 일종인 TFRC(TCP-Friendly Rate Control)가 제안되었다(TCP-friendly 전송률 제어 기법은 TCP 트래픽의 혼잡 제어 방식과 유사한 기법을 사용하여 UDP나 RTP트래픽의 전송률을 제어하는 방법이다). TFRC는 TCP의 윈도우 기반 전송 알고리즘과는 대조적으로 수식을 이용한 전송률 제어 알고리즘을 사용한다. TFRC는 Padhye의 TCP 모델링 공식을 전송률 제어 수식을 사용함으로써 TCP와 공정한 대역폭 분배가 가능하도록 한다. Padhye의 TCP 모델링 공식은 아래 (1)과 같다.

$$T = \frac{s}{t_m \sqrt{\frac{2p}{3}} + t_{RTO} (3\sqrt{\frac{3p}{8}}) p(1 \pm 32p^2)} \quad (1)$$

여기서  $t_m$ 는 RTT(Round Trip Time)를, S는 패킷 크기, p는 패킷 손실률,  $t_{RTO}$ 는 RTO(Retransmission Time-Out)를 의미한다.

시간 t일때 혼잡윈도우를  $W_t$ , 시간 t+1일때 혼잡윈도우를  $W_{t+1}$ 라고 할 때 AIMD의 전송률 증가와 감소는 다음과 같이 표현할 수 있다.

- Increase :  $W_{t+1} = W_t + a$   
(일반적으로  $a = 1.0$ )
- Decrease :  $W_{t+1} = W_t * b$   
(일반적으로  $b = 0.5$ )

여기서 a는 증가 혼잡 윈도우 기본 크기(1MSS)이다.

AIMD는 공정성과 안정성을 보장하는 알고리즘이며, 실험을 통해 여러 개의 TCP 연결에서 공평하고 안전한 최적점으로 근접함을 확인할 수 있다. TCP의 AIMD를 그대로 사용하거나 수정해서 사용하는 전송기법에는 TCP Vegas의 혼잡 제어 알고리즘을 사용하는 SCP(Streaming Control Protocol)<sup>[11]</sup>, 데이터 패킷마다 데드라인을 부여하는 TLTCP

(Time-Line TCP)<sup>[12]</sup>, 기존 AIMD 방식의 틀 위에서 TCP의 응답함수를 수정한 HSTCP (High Speed TCP)<sup>[13]</sup>등 여러 전송기법이 제안되었다.

이 혼잡제어 메커니즘 기법들은 네트워크 상태에 따라 전송률을 조절함으로써 패킷 손실 발생을 감소시키는 장점을 가진다. 하지만 AIMD 혼잡제어 조절의 기준이 수학적 알고리즘이 아닌 많은 실험값을 참고로 설정되었기 때문에 고속 네트워크 환경에서 효율적인 동작을 보장할 수 없다는 단점이 있다, 또한 TCP 모델링 방식은 전송률의 변화가 심하게 발생하는 문제와 함께 이런 혼잡제어 기법들은 비디오 품질에 대한 조절을 하지 않을 경우, 멀티미디어 데이터 전송에 지연을 가질 수 있는 문제점을 갖고 있다.

## 2.2 RTP & RTCP

RTP(Real-time Transport Protocol)는 멀티캐스트 또는 유니캐스트상에서 음성, 화상, 또는 모의 데이터와 같은 실시간 데이터를 전송하는 응용에 적합한 단대 단 트랜스포트 기능을 제공한다. 그러나 RTP는 자원 예약에 대한 내용은 다루지는 않으며, 특히 적시 데이터 전송(timely delivery), QoS 보장, 뒤바뀐 순서의 전송 방지와 같은 기능을 제공하지 않는다. 따라서 트랜스포트의 의미는 실시간 데이터의 특성에 중점을 두어 제정한 표준이라고 할 수 있다. RTP패킷은 UDP를 이용하여 전달된다.

V	P	C	CC	M	PT	sequence number
timestamp						
synchronization source(SSRC) identifier						
contributing source(CSRC) identifiers						

그림 1. RTP 패킷 양식

(그림 1)은 RTP 패킷 형태이다. 헤더는 고정 크기를 가지며 멀티미디어 정보에 따라서 헤더 뒤에 특정 정보 및 데이터가 붙게 된다. V는 버전 필드이며 최근 버전은 2.0이다. P는 32비트 단위로 패킷을 구성하기 위해서 사용된다. X비트가 세팅되면 정확하게 한 개의 확장 헤더가 고정 헤더 다음에 온다는 것을 가리킨다. CC는 고정 헤더에서 CSRC (Contributing Source 식별자)의 개수를 가리킨다. M은 멀티미디어 정보에 대한 프레임 영역을 나타

낸다. PT 필드는 RFC 1890에서 정의된 프로파일의 RTP payload 양식을 지칭하고 응용에 의해서 해석된다. 프로파일은 payload type code를 payload format으로 지정되고 고정된 대응을 시킨 것이다. sequence number는 RTP 패킷이 송신될 때마다 1씩 증가한다. 수신측은 이 필드를 이용하여 패킷분실을 감지하고 패킷 순서를 재 저장한다. timestamp 필드는 RTP 패킷의 첫 번째 옥텟이 샘플링된 시점을 나타낸다. 이것은 실시간 데이터의 동기화와 지터 계산에 이용된다. SSRC(Synchronous Source 식별자) 필드는 카메라 또는 마이크 등의 데이터 원천지의 식별자를 가리킨다. CSRC 필드는 RTP 패킷이 중간 시스템에서 혼합되는 경우에 그 소스들을 구별할 수 있는 식별자들을 가리킨다. 그러나 다중화(multiplexing)와 체크섬은 UDP를 이용한다. 또한 여러 목적지로의 데이터전송은 하위 계층에서 제공해야 한다.

RTCP(RTP Control Protocol)는 세션의 모든 참가자들에게 분실된 패킷 수, 지터 간격, 앞의 패킷과의 지연시간 등의 QoS 정보를 교환하여 응용이 적당한 QoS를 평가하여 adaptive encoding을 제공하도록 한다<sup>[14]</sup>. 또한 RTCP는 많은 참여자들의 스케일을 위해서 패킷 손실률을 계산 하고 사용자 인터페이스의 참여자 ID를 지칭하는 최소한의 세션 제어 정보를 나른다.

또한 데이터의 원천지 식별자가 충돌이 되거나 다시 만들 경우에 변경되어야 하므로 CNAME (Cano-nial NAME)이라 부르는 영구 트랜스포트 식별자를 나른다. 마지막으로 RTCP는 제어 패킷을 주기적으로 모든 참여자에게 전송한다. 이를 수행하기 위한 RTCP 메시지들은 다음과 같다(그림 2).

- SR(sender report) : active sender들은 자신의 송수신에 대한 통계 정보를알리는 데 사용한다.
- RR(receiver report) : active sender가 아닌 참여자들이 자신의 송수신에 대한 통계 정보를 알리는 데 사용한다.
- SDES(Source DEscription) : CNAME을 포함 하여 소스 이름을 기술하는 데 사용한다.
- BYE(BYE) : RTP session을 빠져나올 때 사용한다.
- APP(APPLication) : 새로운 응용 또는 새로운 기능을 시험할 때 그 응용에 한정된 기능을 지정하는데 사용한다.

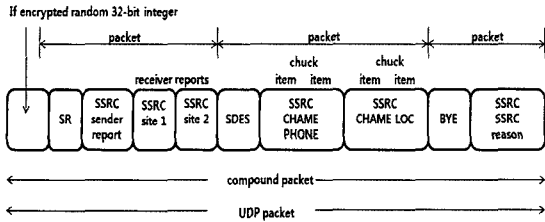


그림 2. RTCP 복합 패킷 양식

RTCP 패킷은 고정된 헤더 필드와 패킷 타입에 따라서 가변의 길이를 갖는 필드가 붙는다. 효과적인 정보 전송을 위해서 여러 RTCP 패킷들이 한 개의 UDP 패킷으로 전송될 수 있다.

### III. POBA(Priority Ordered Buffer-driven A-Algorithm)

본 장에서 제안하는 버퍼 상태를 고려한 우선순위 네트워크 전송 기법은 네트워크의 혼잡 상태에 따라 전송률을 변화하여 멀티미디어 데이터의 전송시 보장되어야 하는 실시간성과 연속성을 고려하였다.

#### 3.1 POBA 개요

POBA는 네트워크 접속시 각 단말들마다 우선순위가 있는 것을 전제로 한다. 그리고 네트워크 관점(혼잡제어)과 사용자의 관점(버퍼상태)을 동시에 고려하여 스트리밍을 하게 된다. POBA는 기본적으로 표준 스트리밍 프로토콜인 RTP/RTCP 모델을 사용, 실시간으로 RTCP 정보를 이용함으로써 네트워크 혼잡 제어를 위해 패킷 손실률을 산출하고 스트리밍 서비스의 전송률을 조절하게 된다.

#### 3.2 우선 순위의 부여

POBA에서는 처음 네트워크 접속 시 우선 순위를 부여받는 과정을 거친다. 네트워크의 혼잡 상태 판단은 RTCP의 Receiver Report(RR) 정보를 이용하여 수식 (2)를 얻을 수 있다.

$$l = \frac{N_{receive}}{N_{max} - N_{first}} \quad (2)$$

$N_{receive}$  는 받은 패킷 수이고  $N_{max}$  는 수신된 패킷의 최대 sequence 넘버,  $N_{first}$  는 최초 패킷의 sequence 넘버이다. 따라서 이 값이 일반적인 링크 에러율 보

다 크면 High-Level, 작으면 Low-Level 값을 갖는다.

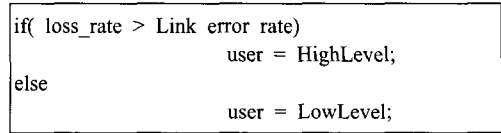


그림 3. 우선 순위 부여

#### 3.3 수신단의 버퍼 상태를 고려한 POBA의 적용

각 단말이 네트워크에서 자신의 우선 순위를 부여받으면 사용자의 멀티미디어 데이터 재생을 고려하여 버퍼상태를 수식 (3)과 같이 확인한다. 그리고 버퍼의 상태의 각 레벨에 맞는 전송률을 조절한다.

$$B(t) = B(t-1) + N(t-1,t) - Output(t-1,t) \quad (3)$$

$B(t)$ 는 현재 버퍼상태,  $B(t-1)$ 은 이전 버퍼상태,  $N(t-1,t)$ 은  $t-1 \sim t$  사이에 받은 패킷 수,  $Output(t-1,t)$ 은 멀티미디어 재생으로 인해  $t-1 \sim t$  동안 소비되는 패킷 수로서 현재의 버퍼 상태는 수신된 RTCP의 RR정보에서  $N(t)$ 값을 얻어와 버퍼의 상태를 예측한다. 현재 버퍼 상태를 파악했으면 연속적인 재생을 위한 최소한의 버퍼 레벨과 Overflow로 패킷 손실이 발생할 수 있는 버퍼 레벨의 값을 정하게 되는데, 이 값들은 수신단의 우선 순위(HighLevel, Low-Level)에 맞는 알고리즘을 적용한다. 알고리즘은 (그림 4)와 같다.

우선 Low-Level에서  $\alpha$ 를 버퍼의 단계를 구분하는 임계값이라 하면 시간  $t$ 에서 버퍼 상태  $B(t)$ 는 다음과 같이 3단계로 분류한다.

- UnderFlow단계 :  $0 < B(t) < 2\alpha\%$
- Stable단계 :  $2\alpha\% < B(t) < 100-\alpha\%$
- OverFlow단계 :  $100-\alpha\% < B(t) < 100\%$

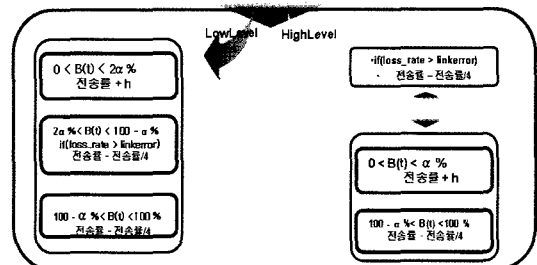


그림 4. POBA 구조

그리고 각 단계에 맞는 전송률을 제어한다. S를 송신측의 전송률, h를 전송률을 올리기 위한 기본 단위 상수 값이라 하면, 각 단계별 전송률 제어는 다음과 같다.

- UnderFlow 단계 :  $S = S + h$
- Stable 단계 :  $\text{if}(\text{loss\_rate} > \text{Link\_error})$   
 $S = S - S/4$
- OverFlow 단계 :  $S = S - S/4$

이와는 다르게 High-Level은 네트워크 상태를 먼저 파악한다.

$\text{if}(\text{loss\_rate} > \text{Link\_error})$   
 $S = S - S/4$

그 후 네트워크 상태가 안정적인 상태가 되면 버퍼 상태를 Low-Level과 다르게 다음과 같이 2단계로 파악한다.

- UnderFlow 단계 :  $0\% < B(t) < \alpha\%$
- OverFlow 단계 :  $100 - \alpha\% < B(t) < 100\%$

각 단계별 전송률 제어는 다음과 같다.

- UnderFlow 단계 :  $S = S + h$
- OverFlow 단계 :  $S = S - S/4$

$\alpha$ 값은 비디오 재생의 질과 각 레벨 간의 버퍼 유지를 위해 전송률 증가로 인한 네트워크 안에서 경쟁을 피하기 위해 몇 번의 실험을 통해 결정하였다. 그리고 전송률 조절 값은 A-IMD 알고리즘에서 패킷 손실 발생시 급격한 전송률 감소로 TCP 친화적으로 동작하게 한 반면 수신측의 버퍼 점유도를 급격히 낮아지는 것을 고려하여 역시 실험을 통하여 결정하였다.

POBA는 Low-level이 High-level보다 버퍼 상태에 대한 우선권을 갖게 된다. 따라서 안정적인 스트리밍 재생을 위해 버퍼의 상태를 우선적으로 고려하여 전송률을 제어한다. 반면 High-level은 낮은 우선권을 갖기 때문에 네트워크의 혼잡도를 먼저 고려하고, 그 후 버퍼 상태를 고려하게 되는데 버퍼 상태를 고려하는 과정에서 High-level이 Low-level보다 더 넓은 범위에서 네트워크의 혼잡도를 고려하게 된다.

#### IV. 실험 및 평가

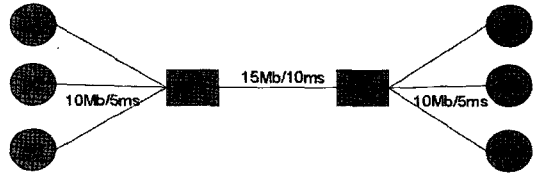


그림 5. 실험 환경

##### 4.1 실험환경

LBNL(Lawrence Berkeley National Laboratory)의 NS2(Network Simulator)를 사용하여 그림 5와 같은 실험 환경을 구성하여 성능 실험을 수행 하였다. 각 노드와 라우터까지는 10Mbps, 라우터 사이는 15Mbps의 대역폭으로 연결되어 있고, 기본적으로 s1은 15Mbps의 전송률과 High-Level, s2는 15Mbps의 전송률과 Low-Level로 동작하는 설정을 두었다. s3은 일반 TCP의 ftp트래픽으로 전송한다.

##### 4.2 POBA 성능 실험

그림 6는 POBA를 적용하지 않고 UDP로 데이터를 전송했을 때 송신노드1과 송신노드2에 나타난 패킷 손실률이고 그림 7는 POBA를 적용했을 때 송신노드1과 송신노드2의 패킷 손실률을 나타낸다. 그림 6에서는 UDP의 공격적인 특성과 같이 서로 경쟁적으로 패킷 손실률이 대칭을 이루는 것을 볼 수 있다. 그림 7에서는 High-Level인 node1이 버퍼 점유도를 높이기 위해 처음에 높은 패킷 손실률을 보이다가 점차적으로 패킷 손실률이 줄어들며 안정적으로 들어가는 것을 볼 수 있고 Low-Level인 node2는 우선 네트워크의 혼잡을 고려한 후 중간에

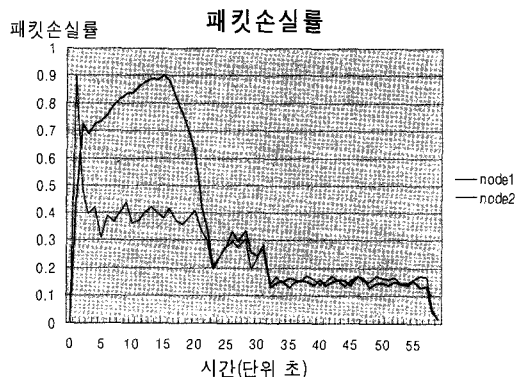


그림 6. 미적용시 패킷 손실률

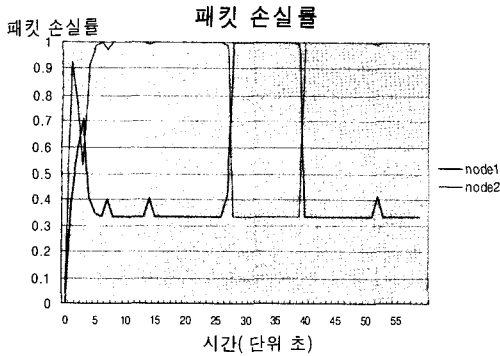


그림 7. 적용시 패킷 손실률

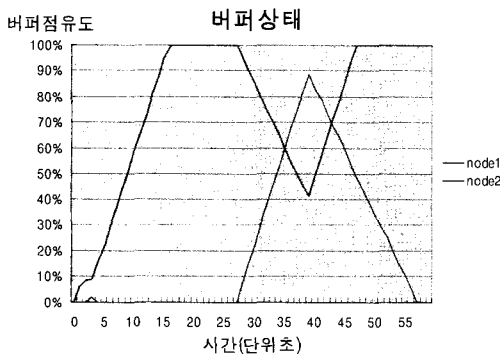


그림 8. 미적용시 버퍼상태

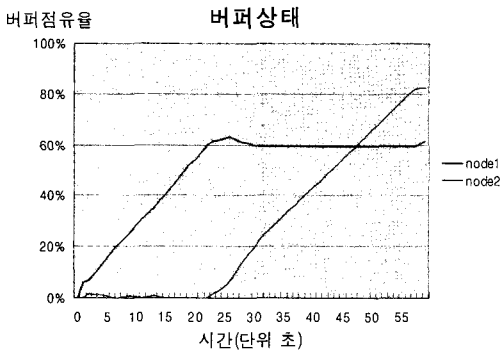


그림 9. 적용시 버퍼상태

버퍼 점유도를 높이기 위해 공격적인 성향을 띠는 결과를 보이지만 시간이 지날수록 안정적으로 들어가는 것을 볼 수 있다. (그림 8)는 POBA를 적용하지 않았을 경우에 node1과 node2의 버퍼 점유도를 나타낸 것이고 (그림 9)는 POBA를 적용한 후 node1과 node2의 버퍼 점유도를 나타낸 것이다. 이 결과에서도 알 수 있듯이 (그림 8)에서는 UDP의 특성처럼 네트워크를 점유한 node가 버퍼 점유도가 높고 특히 overflow나 underflow가 장기간 발생하

는 것을 볼 수 있다. 이와 반대로 (그림 9)를 보면 High-Level인 node1은 처음부터 버퍼 점유도를 안정적인상태를 유지하고 Low-Level인 node2는 처음에는 네트워크 상태를 고려하여 underflow가 발생하나 점차적으로 버퍼 점유도를 높이는 것을 볼 수 있다.

### V. 결론 및 향후 과제

네트워크의 발달에 따라 고속화 되고 대역폭이 늘어나면서 많은 텍스트나 정지화상(JPEG)등의 이용에서 VOD나 실시간 동영상 화상회의 같은 스트리밍 서비스가 늘어나게 되면서 네트워크와의 형평성 문제에 대한 논의가 많이 진행되고 있다. 본 논문에서는 네트워크의 형평성 문제와 사용자측면의 버퍼 상태를 고려한 POBA(Priority Order Buffer Algorithm)를 제시하였다. 스트리밍 서비스는 패킷 손실보다는 서비스 지연에 민감한 것을 착안하면 실험결과에서 보이듯이 높은 혼잡 상태에서 낮은 패킷 손실률과 안정적인 버퍼의 점유율이 나오는 것으로 스트리밍서비스에 적합한 환경을 제공할 것 이라는 것을 예상할 수 있었다.

향후 과제로서 좀 더 확장된 실험 환경과 다양한 대역폭 조건으로 성능 측정을 해야 할 것이고, 좀 더 현재 인터넷인 TCP와 친화적으로 동작할 수 있는 방법에 대한 연구가 지속되어야 할 것이다.

### 참고 문헌

- [1] RFC Recommendation 1889 RTP, "A Transport Protocol for Real-Time Application," Jan. 1996.
- [2] J. Postel, "User Datagram Protocol Sciences Institute," RFC 768, ISI 28 August 1980.
- [3] B. Melander, M. Bjorkman, and P. Gunningberg. "A New End-to-End Probing and Analysis Method for Estimation Bandwidth Bottlenecks," *In Global Internet Symposium*, 2000.
- [4] B. Song, K. Chung, and Y. Shin, "SRTP: TCP-friendly congestion control for multimedia streaming," *16th International Conference on Information Networking*, January 2002.
- [5] S. Na and J. Ahn, "TCP-like flow control algorithm for real-time applications," *IEEE International Conference*, September 2000.
- [6] S. Floyd et al., "Equation-Based Congestion Co-

- ontrol for Unicast Applications," Proc. *ACM SIGCOMM 2000*, pp.43-56, Aug. 2000.
- [7] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE ACM Trans on Networking*, Aug. 1999.
- [8] W. Tan and A. Zakhor, "Real-Time Internet Video Using Error Resilient Scalable compression and TCP-Friendly Transport Protocol," *IEEE Trans Multimedia*, vol. 1, no. 2, pp.172-86, June 1999.
- [9] M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control," RFC 2581, 1999.
- [10] J. Padhye, J. Kurose, D. Towsley and R. Koodli, "A model based TCP-friendly rate control protocol," *NOSSDAV'99*, 1999.
- [11] S. Cen, C. Pu and J. Walpole, "Flow and congestion control for internet streaming application," *Multimedia Computing and Networking*, January 1998.
- [12] B. Mukherjee and T. Brecht, "Time-line TCP for the TCP-friendly delivery of streaming media," *International Conference on Network Protocols*, 2000.
- [13] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC3649, December 2003.
- [14] RFC Recommendation 1890 RTCP, "A Transport Control Protocol for Real-Time Application," Jan. 1996.

김 승 훈 (Seung-Hun Kim)

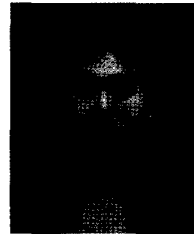
준회원



2007년 2월 : 인천대학교 컴퓨터 공학과 졸업  
2007년 2월~현재 : 인천대학교 교육대학원 석사과정  
<관심분야> Sensor Network, Multimedia Streaming

최 재 원 (Jae-Won Choi)

준회원



2006년 2월 : 인천대학교 컴퓨터 공학과 졸업  
2006년 3월~현재: 인천대학교 일반대학원 석사과정  
<관심분야> WLAN Streaming, Sensor Network

최 승 식 (Seung Sik Choi)

정회원



1988년 2월 연세대학교 전자공학과 학사  
1990년 2월 KAIST 전기 및 전자공학과 석사  
2002년 2월 KAIST 전기 및 전자공학과 박사  
1990년 3월~1998년2월 KT 통신

망연구소 전임연구원

2002년~2004년 KT서비스개발연구소 선임연구원

2004년~현재 인천대학교 컴퓨터공학과 조교수

<관심분야> 무선액세스제어, 무선자원관리, 무선인터넷 프로토콜