

영상 스트리밍 서비스를 위한 우선순위 버퍼 혼잡제어 알고리즘

김승훈*^o 최재원* 최승식*

* 인천대학교 컴퓨터공학과

Priority Ordered Buffer and Congestion Control Algorithm for Video Streaming Service

Seung Hun Kim^o Jae Won Choi Seung Sik Choi

Department of Computer Engineering, University of Incheon

winlab@incheon.ac.kr

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)가 사용되고 있는데, UDP는 현재의 인터넷 기반인 TCP(Transmission Control Protocol)와 경쟁관계에 있어 네트워크의 혼잡을 초래하는 경우가 많았다. 또한 TCP 트래픽은 네트워크의 혼잡제어를 수행하지만 수신자의 관점을 고려하지 않은 점이 있어서 스트리밍 전송에는 적합하지 않은 단점을 가지고 있다. 본 논문에서는 UDP(User Datagram Protocol) 트래픽의 네트워크 형평성 문제를 해결하고 수신자의 관점을 고려하기 위해 네트워크에 우선순위를 적용하여 RTP/RTCP를 기반으로 수신자의 버퍼를 고려한 혼잡제어 알고리즘을 제안하였다. 시뮬레이션 결과를 통해 우선 순위가 있는 네트워크가 우선 순위가 없는 네트워크에 비해 패킷 손실률과 버퍼 점유도 면에서 스트리밍 전송에 적합한 환경을 제공하고 있음을 확인할 수 있었다.

1. 서론

네트워크의 성능이 향상되면서 스트리밍 기술이 널리 보급되고 있다. 스트리밍 시에 전송하는 멀티미디어 데이터(오디오, 비디오 등)의 경우 대용량이고, 높은 전송 대역폭을 요구하며 실시간 전송을 필요로 하는 특징을 갖고 있어서, 멀티미디어 데이터들은 패킷 손실이나 지연, 대역폭 상태에 크게 좌우된다. 여기에 현재의 인터넷의 경우 전송의 신뢰성을 기본으로 하는 TCP(Transmission Control Protocol) 기반의 네트워크이기 때문에 네트워크 상황에 따른 전송률의 급격한 변화와 높은 손실률에 따른 재전송의 시간 지연으로 스트리밍 서비스에 부적합한 환경을 제공한다. 이에 실시간으로 전송할 수 있는 프로토콜로서 UDP(User Datagram Protocol)와 RTP(Real-time Transport Protocol)가 사용되고 있다[1-2]. 이런 UDP나 RTP를 이용하는 데이터들의 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 중이다 [3-5].

본 논문에서는 2장에서 이런 연구들의 소개와 함께 실시간 스트리밍 프로토콜을 간략히 설명하고 3장에서 네트워크에 우선 순위를 적용하여 우선 순위에 따라 스트리밍 서비스를 위한 사용자의

버퍼와 동시에 네트워크의 혼잡을 다르게 제어하는 하이브리드 방식의 알고리즘을 제안한다. 그리고 4장에서 시뮬레이션 결과를 제시한 후 앞으로의 연구 방향으로 끝을 맺도록 하겠다.

2. 관련 연구

2.1 혼잡 제어 메커니즘

혼잡 제어 기법으로는 대표적으로 AIMD(Additive Increase Multiplicative Decrease) 알고리즘과 TCP 모델링 방식이 있다[6-7].

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 RTP 트래픽의 전송률을 제어하는 방법이다.) TFRC는 TCP의 윈도우 기반 전송 알고리즘과는 대조적으로 율제어 등식 기반 전송 알고리즘을 사용한다. TFRC는 Padhye의 TCP 모델링 공식을 율제어 등식으로 사용함으로써 TCP와 공정한 대역폭 분배가 가능하도록 한다. Padhye의 TCP 모델링 공식은 아래 (1)과 같다..

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

여기서 t_m 는 RTT(Round Trip Time)를, S는 패킷 크기, p는 패킷 손실률, t_{RTO} 는 RTO(Retransmiss-

ion Time-Out)를 의미한다.

TCP 의 Congestion Avoidance algorithm 인 AIMD 의 제어 알고리즘은 다음과 같다.

AIMD 정상 상태 제어 방법.

Increase : $Wt+R = Wt + a$ (일반적으로 $a = 1.0$)

Decrease : $Wt+R = Wt * b$ (일반적으로 $b = 0.5$)

TCP 의 AIMD 를 그대로 사용하거나 수정해서 사용하는 전송기법에는 TCP Vegas 의 혼잡 제어 알고리즘을 사용하는 SCP(Streaming Control Protocol)[8], 데이터 패킷마다 데드라인을 부여하는 TLTCP(Time-Line TCP)[9], 기존 AIMD 방식의 틀 위에서 TCP 의 응답함수를 수정한 HSTCP(High Speed TCP)[10]등 여러 전송기법이 제안되었다.

이 혼잡제어 메커니즘 기법은 네트워크 상태에 따라 전송률을 조절함으로써 패킷 손실 발생을 감소시키는 장점을 가진다. 하지만 AIMD 방식의 혼잡제어 조절의 기준이 수학적 알고리즘이 아닌 많은 실험값을 참고로 설정되었기 때문에 고속 네트워크 환경에서 효율적인 동작을 보장할 수 없고 TCP 모델링 방식 또한 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 실시간 전송 및 제어 프로토콜 (RTP/RTCP)

RTP(Real-time Transport Protocol)

RTP 는 기본적으로 UDP 를 기반으로 패킷 순서화와 흐름제어를 수행하며 종점간 실시간 데이터의 전송을 담당한다. RTP 는 멀티캐스트 또는 유니캐스트상에서 음성, 화상, 또는 모의 데이터와 같은 실시간 데이터를 전송하는 응용에 적합한 단대단 트랜스포트 기능을 제공한다. 그러나 RTP 는 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.

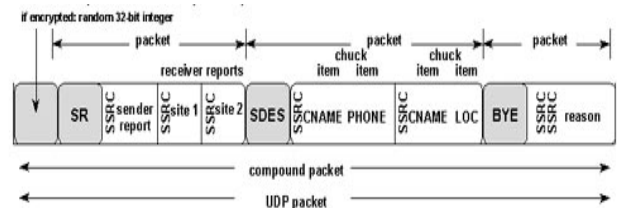
RTCP(RTP Control Protocol)

RTCP 는 세션의 모든 참가자들에게 분실된 패킷 수, 지터 간격, 앞의 패킷과의 지연시간 등의 QoS 정보를 교환하여 응용이 적당한 QoS 를 평가하여 adaptive encoding 을 제공 하도록 한다. 또한 RTCP 는 많은 참여자들의 스케일을 위해서 패킷 손실율을 계산 하고 사용자 인터페이스의 참여자 ID 를 지칭하는 최소한의 세션 제어 정보를 나른다.

또한 데이터의 원천지 식별자가 충돌이 되거나 다시 만들 경우에 변경되어야 하므로 CNAME (Canonical Name)이라 부르는 영구 트랜스포트 식별자를 나른다. 마지막으로 RTCP 는 제어 패킷을 주기적으로

모든 참여자에게 전송한다. 이를 수행하기 위한 RTCP 메시지들은 다음과 같다(그림 1).

- ◆ SR (sender report) : active sender 들은 자신의 송수신에 대한 통계 정보를알리는 데 사용한다.
- ◆ RR (receiver report) : active sender 가 아닌 참여자들이 자신의 송수신에 대한 통계 정보를 알리는 데 사용한다.
- ◆ SDES (Source Description) : CNAME 을 포함하여 소스 이름을 기술하는 데 사용한다.
- ◆ BYE (BYE) : RTP session 을 빠져나올 때 사용한다.
- ◆ APP (Application) : 새로운 응용 또는 새로운 기능을 시험할 때 그 응용에 한정된 기능을 지정하는데 사용한다.



(그림 1) 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.

3. POBA (Priority Ordered Buffer-driven Algo-rithm)

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

우선 POBA 는 네트워크 접속시 각 단말들 마다 우선 순위가 있는 것을 전제로 한다. 그리고 네트워크 관점(혼잡제어)과 사용자의 관점(버퍼상태)을 동시에 고려하여 스트리밍을 하게 된다. POBA 는 기본적으로 표준 스트리밍 프로토콜인 RTP/RTCP 모델을 사용, 실시간으로 RTP/RTCP 정보를 이용함으로써 네트워크에 적응하는 전송률 및 패킷 손실률을 산출하여 스트리밍 서비스의 전송률을 조절하게 된다.

3.2 우선 순위의 부여

POBA 에서는 우선 네트워크 접속 시 우선 순위를 부여받는 일련의 과정을 거친다. 먼저, 네트워크의 혼잡 상태 판단은 RTCP 를 이용하여 수식 (2)로 얻을 수 있다.

$$I = \frac{N_{recieve}}{N_{max} - N_{first}} \quad (2)$$

$N_{recieve}$ 는 받은 패킷 수이고 N_{max} 는 수신된 패킷의

최대 시퀀스 넘버, N_{first} 는 최초 패킷의 시퀀스 넘버이다. 따라서 이 값이 일반적인 링크 에러율 보다 크면 Low-Level, 작으면 High-Level 값을 갖는다

```

if( loss_rate > Link error rate)
    user = LowLevel;
else
    user = HighLevel;
    
```

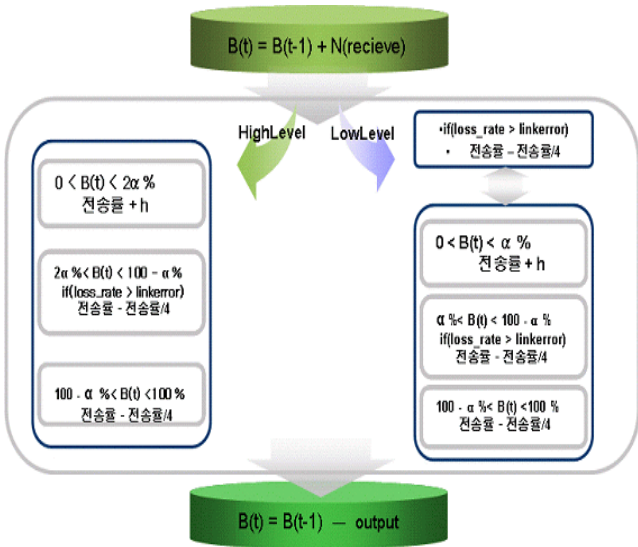
(그림 2) 우선 순위 부여

3.3 수신단의 버퍼상태를 고려한 POBA 의 적용

각 단말이 네트워크에서 자신의 레벨을 부여받으면 POBA 는 사용자의 멀티미디어 데이터 재생의 질을 고려하여 버퍼상태를 수식 (3)과 같이 확인한다. 그리고 버퍼의 상태에 따라 각 레벨에 맞는 전송률을 조절한다.

$$B(t) = B(t-1) + N(\text{receive}) - \text{Output} \quad (3)$$

$B(t)$ 는 현재 버퍼상태, $B(t-1)$ 은 이전 버퍼상태, $N(\text{receive})$ 는 받은 패킷 수, Output 은 스트리밍 재생으로 인해 소비되는 패킷 수로서 RTCP 정보로 $N(\text{receive})$ 값을 얻어와 버퍼의 상태를 예측한다. 현재 버퍼 상태를 파악했으면 각 레벨에 맞는 알고리즘을 적용한다. 알고리즘은 (그림 3)과 같다.



(그림 3) POBA 구조

우선 High-Level 부터 살펴보면, 버퍼의 점유도를 측정할 후 버퍼 점유도가 기준치 이하 값(High-Level : $0 \sim 2\alpha$, Low-Level : $0 \sim \alpha$)으로 내려가면 underflow 방지를 위하여 전송률을 각 레벨의 특정 값(High-Level : h , Low-Level : h)만큼 점차적으로 높게 된다. 이와 반대로 버퍼 상태가 각 레벨의 기준치이상 값(High-Level : $100 - \alpha \sim 100$, Low-Level : $100 - \alpha \sim 100$)으로 올라가면 overflow 를 방지하기

위해 전송률의 1/4 만큼 씩 내리게 된다. 이 값은 AIMD 에서 패킷 손실 발생시 급격한 전송률 감소로 TCP 친화적으로 동작하게 한 반면 수신측의 버퍼 점유도를 급격히 낮아지는 것을 고려하여, 값을 조절하였다. 버퍼 상태가 적정 구간 (High-Level : $2\alpha \sim 100 - \alpha$, Low-Level : $\alpha \sim 100 - \alpha$) 에 있게 되면 네트워크의 혼잡 상태를 파악하게 되는데 네트워크가 혼잡 상태 이면 전송률을 낮추고 그렇지 않으면 전송률을 유지하게 된다. α 값은 비디오 재생의 질과 각 레벨 간의 버퍼 유지를 위해 전송률 증가로 인해 네트워크 안에서 경쟁을 피하기 위해 몇 번의 실험을 통해 적용하였다.

Low-Level 에서는 High-Level 과 경쟁하지 않고 버퍼 유지를 위해 우선, 네트워크 상태가 혼잡상태일 경우 전송률을 1/4 씩 줄이고 그렇지 않을 경우에 High-Level 의 알고리즘을 적용하였다. 단, High-Level 과의 Underflow 기준치 값은 다르게 적용하였다

4 실험 및 평가

4.1 실험환경

LBNL (Lawrence Berkeley National Laboratory)의 NS2 (Network Simulator)를 사용하여 (그림 4)와 같은 실험 환경을 구성하여 성능 실험을 수행 하였다



(그림 4) 실험 환경.

각 노드와 라우터까지는 10Mb, 라우터 사이는 15 Mbps 의 대역폭으로 연결되어 있고, 기본적으로 s1 은 15Mbps 의 전송률과 High-Level, s2 는 15Mbps 의 전송률과 Low-Level 로 동작하는 설정을 두었다. s3 은 일반 TCP 의 ftp 트래픽으로 전송한다.

4.2 POBA 성능 실험

(그림 5)는 POBA 를 적용하지 않고 UDP 로 데이터를 전송했을 때 node1 과 node2 에 나타난 패킷 손실률이고 (그림 6)는 POBA 를 적용했을 때 node1 과 node2 의 패킷 손실률을 나타낸다. (그림 5)에서는 UDP 의 공격적인 특성과 같이 서로 경쟁적으로 패킷 손실률이 대칭을 이루는 것을 볼 수 있다

(그림 6)에서는 High-Level 인 node1 이 버퍼 점유도를 높이기 위해 처음에 높은 패킷 손실률을 보이다가 점차적으로 패킷 손실률이 줄어들며 안정적으로 들어가는 것을 볼 수 있고 Low-Level 인 node2 는 우선 네트워크의 혼잡을 고려한 후 중간에 버퍼 점유도를 높이기 위해 공격적인 성향을 띠는 결과를 보이지만 시간이 지날수록 안정적으로 들어가는 것을 볼 수 있다. (그림 7)는 POBA 를 적용하지 않았을 경우에 node1 과 node2 의 버퍼 점유도를 나타낸 것이고 (그림 8)는 POBA 를 적용한 후 node1 과 node2 의 버퍼 점유도를 나타낸 것이다. 이 결과에서도 알 수 있듯이 (그림 7)에서는 UDP 의

특성처럼 네트워크를 점유한 node 가 버퍼 점유도가 높고 특히 over-flow 나 underflow 가 장기간 발생하는 것을 볼 수 있다. 이와 반대로 (그림 8)를 보면 High-Level 인 node1 은 처음부터 버퍼 점유도를 안정적인 상태를 유지하고 Low-Level 인 node2 는 처음에는 네트워크 상태를 고려하여 underflow 가 발생하나 점차적으로 버퍼 점유도를 높이는 것을 볼 수 있다.

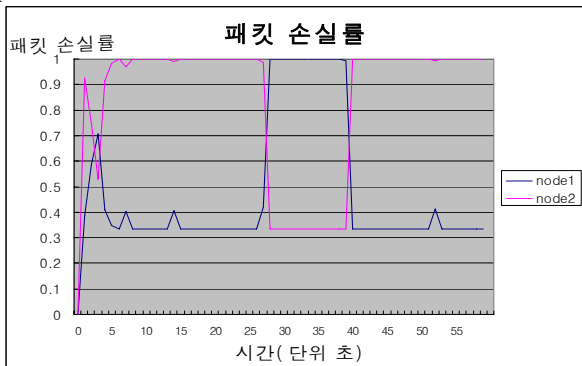
5. 결론 및 향후 과제

본 논문에서는 네트워크의 형평성 문제와 사용자측면의 버퍼 상태를 고려한 POBA(Priority Order Buffer Algorithm)를 제시하였다. 실험결과를 통해 높은 혼잡 상태에서 낮은 패킷 손실률과 안정적인 버퍼의 점유율이 나오는 것으로 스트리밍 서비스에 적합한 환경을 제공할 것이라는 것을 예상할 수 있었다. 향후 과제로서 좀 더 확장된 실험 환경과 다양한 대역폭 조건으로 성능 측정을 해야 할 것이고, 좀 더 현재 인터넷인 TCP 와 친화적으로 동작할 수 있는 방법에 대한 연구가 지속되어야 할 것이다.

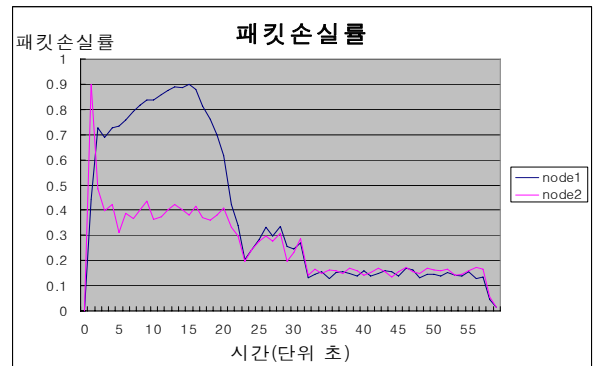
[2] Postel, J., "User Datagram Protocol Sciences Institute,," RFC 768, ISI 28 August 1980.
 [3] B. Melander, M. Bjorkman, and P. Gunning berg. "A New End-to-End Probing and Analysis Method for Estimation Bandwidth Bottlenecks", In Grobal Internet Symposium, 2000.
 [4] B. Song, K. Chung and Y. Shin, "SRTP:TCP-friendly congestion control for multimedia streaming," 16th Intentional Conference on Information Networking, Junaury 2002.
 [5] S. Na and J. Ahn, "TCP-like flow control algorithm for real-time applications," IEEE International Conference, September 2000.
 [6] M. Allman, V. Paxon, and W. Stevens, "'TCP Congestion Control,'" RFC 2581, 1999.
 [7] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," NOSSDAV'99, 1999.
 [8] S. Cen, C. Pu, and J. Walpole, "Flow and congestion control for internet streaming application," Multimedia Computing and Networking, January 1998.
 [9] B. Mukherjee and T. Brecht, "Time-line TCP for the TCP-friendly delivery of streaming media," International Conference on Network Protocols, 2000.
 [10] S. Floyd, "HighSpeed TCP for Large Congestion Windows", RFC3649, December 2003.

[참고 문헌]

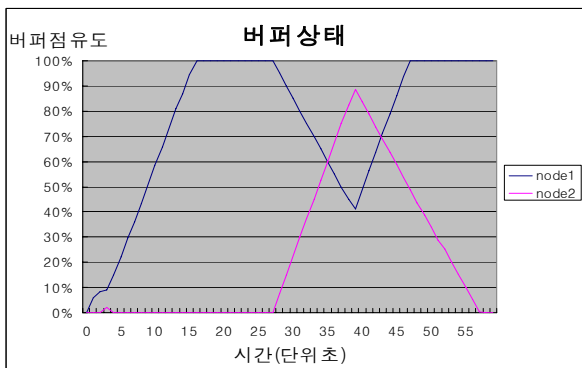
[1] RFC 1889, RTP - A Transport Protocol for Real-time Applications.



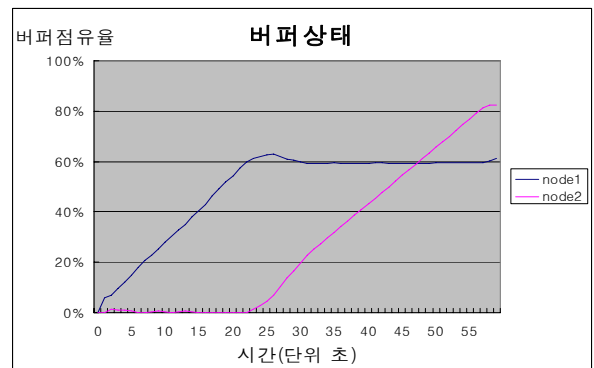
(그림 5) 미적용시 패킷 손실률



(그림 6) 적용시 패킷 손실률



(그림 7) 미적용시 버퍼상태



(그림 8) 적용시 버퍼상태