Loss-RTT 기반 차등 전송률 조절 알고리즘에 관한 연구

김지언, 정재일
한양대학교 전자통신공학과
전화 : 02-2282-4487 / 핸드폰 : 019-390-1807

Loss-RTT based Differentiated Rate Adaptation Algorithm

Ji-Eon Kim, Jae-Il Jung
Dept. of Electrical Communication Engineering, Hanyang Univ.
E-mail : jekim@mnlab.hanyang.ac.kr

Abstract

TCP is ill-suited to real-time multimedia applications. Its bursty transmission, and abrupt and frequent wide rate fluctuations cause high delay jitters and sudden quality degradation of multimedia applications. Deploying non congestion controlled traffic results in extreme unfairness towards competing TCP traffic. Therefore, they need to be enhanced with congestion control schemes that not only aim at reducing loss ratios and improve bandwidth utilization but also are fair towards competing TCP connections.

This paper proposes a differentiated rate adaptation algorithm based on loss and round trip time. Rate in a sender quickly responds to loss ratio and holds steady state. Additionally, this algorithm reduces loss ratio by loss prediction in a receiver.

Ⅰ. 서론

오늘날 인터넷에서 사용되고 있는 멀티미디어 애플리케이션의 대부분이 UDP 프로토콜을 사용하고 있다. 혼잡제어 기능이 없는 UDP 트래픽은 TCP 트래픽이 상대적으로 적은 대역폭을 할당받도록 한다. 네트워크의 혼잡상태에 대한 고려가 전혀 없는 best-effort 트래픽은 패킷 손실을 가져오기 쉽다. 오늘날 인터넷 트래픽의 95%를 차지하고 [1]있는 TCP는 패킷 손실 시 전송률을 줄인다. 그러나 혼잡제어 기능이 없는 트래픽에서 전송률 감소가 없다면 TCP는 공정한 대역폭을 할당받지 못할 것이다. 그러므로 UDP 소스는 혼잡제어 메커니즘과 loss ratio를 줄일 뿐만 아니라 TCP와 공정하게 대역폭을 할당받도록 개선되어야 한다.

본 논문에서는 송신 측이 네트워크의 상태에 따라 전송률 조절하여 QoS를 제어하는 메커니즘을 제안한다. 즉 수신 측으로부터 오는 피드백 정보에 따라 송신 측은 네트워크가 부하상태이면 전송률을 줄이고, 그렇지 않으면 높인다. 하지만 각 세션마다 요구하는 전송률이 다르기 때문에 이를 고려하여 차등적으로 전송률의 증감량을 결정하여 네트워크의 상황에 대해서 모두 같은 방식으로 대응하지 않고 서로 다른 증감량으로 전송률을 조절하여 QoS를 보장할 수 있다. 또한 패킷 손실 시 수신 측에서 피드백을 보내어 네트워크 상황에 빠르게 반응하며 손실예측을 통해 loss ratio를 줄일 수 있다.

2절에서는 TCP-friendly 알고리즘을 소개하고, 3절에서는 본 논문에서 제안하는 알고리즘을 소개하고, 4절에서는 시뮬레이션 결과를 보여주고, 5절에서는 결론을 맺는다.

Ⅱ. TCP-Friendly algorithms

TCP와 유사한 방식으로 송신 측에서 전송률을 조정하는 많은 방법들이 연구되고 있다. 이 절에서는 이들 중 몇 개의 주요한 특성을 알아보도록 한다.

Jacobs[2]는 TCP의 혼잡제어 메커니즘을 이용하지만 손실된 패킷에 대한 재전송이 없는 메커니즘을 제안했다. 송신 측에서는 수신 측에서 보내오는 ACK에 따라 미리 transmission window를 유지한다. window의 크기에 따라 전송률을 조절한다. 그러므로 같은 loss, delay환경에서 TCP와 같은 대역폭을 할당받게 된다. 하지만 수신 측에서 패킷마다 ACK를 보내게 되므로 네트워크의 트래픽을 증가시키고, 특히 패킷 길이가 작은 음성 같은 경우 전송률

조절의 이득이 줄게 된다.

Floyd[3]은 delay와 loss값을 가지고 TCP의 전송률(R_{tcp})을 모델링 하였다.

$$R_{tcp} = \frac{1.22 \times M}{\tau \times \sqrt{l}}$$

M은 패킷 크기, τ는 RTT, l은 average loss ratio이다. TFRC(TCP-friendly rate Control)[4]에서는 패킷 손실을 네트워크의 혼잡으로 인식하고 loss와 RTT값으로 위 식을 이용하여 TCP의 전송률을 구하고 이를 참조하여 전송률을 구한다.

RAP(Rate Adaptation Protocol)[5]은 송신 측에서 sequence number를 가진 data packet을 보내고 수신 측에서는 각 패킷마다 ACK를 보내어 측정된 RTT와 loss를 값을 통해서 전송률을 결정하게 된다. 패킷 loss가 발생하지 않으면 패킷 전송간격을 조정하여 전송률을 높이고 loss가 발생하면 현재의 전송률의 1/2로 전송하게 된다.

LDA(Loss-Delay based Adjustment Algorithm)[6]는 RTP프로토콜을 이용하였다. 수신 측에서 매 패킷마다 ACK를 보내는 것이 아니라 RTCP report를 이용하여 loss 발생여부를 알린다. RTCP receiver report를 개선하여 수신 측에서 bottleneck bandwidth를 측정하여 송신 측으로 보고하고, loss와 bottleneck bandwidth를 기반으로 하여 전송률을 조절한다.

위에서 제안된 알고리즘들의 공통된 문제점은 TCP와 대역폭공유에 초점을 맞춘 것이다. TCP와의 공존은 중요하지만 서로 다른 특성을 가진 스트림에 같은 알고리즘을 적용하여 대역폭이 많이 필요한 스트림에는 여전히 좋은 QoS를 제공할 수 없다. 그러므로 본 논문에서는 TCP와 대역폭을 공유하면서도 각 스트림의 특성에 따라 차등적으로 전송률을 조절하는 알고리즘을 제안한다.

Ⅲ. 차등 전송률 조절 - LRDA(Loss-RTT based Differentiated rate Adaptation)

본 논문에서 제안하는 전송률 조절 메커니즘의 목표는 다음과 같다.

- 각 스트림이 원하는 전송속도에 따라 차별화 되게 전송률 조절한다.
- loss와 RTT에 따라 전송률이 조절됨에 의해 네트워크의 상황에 반응한다.
- 수신 측으로부터 오는 피드백의 양을 줄여서 네트워크의 트래픽을 줄인다.
- 네트워크의 혼잡상황에 빨리 대처한다.
- loss예측을 통해 loss ratio를 줄인다.

이를 위해 송신 측은 INCREASE, DECREASE, HOLD, HUNT의 상태를 가지고 수신 측은 LOSS, PREDICT, NORMAL, HUNT의 상태를 가진다. INCREASE는 송신 측이 현재 전송률을 증가시키고 있다는 것을 나타내고, DECREASE는 감소시키고 있다는 것을, HOLD는 이전의 전송률을 유지하고 있다는 것을 나타내며 HUNT는 수신 측에 피드백 정보를 보내라는 것을 의미한다. 수신 측의 LOSS는 패킷 손실이 발생했다는 것을, PREDICT는 손실 예측, NORMAL 정상상태, HUNT는 송신 측의 HUNT에 대한 응답을 나타낸다.

3.1 송신 측의 기능

차별 화된 전송률 조절을 위해서 송신 측은 스트림이 요구하는 전송률(TR-Target Rate)을 알고 있다는 가정하에 전송률을 조절한다. 송신 측은 목표 전송속도의 50%에서 전송을 시작하고 최대 TR의 110%까지 전송률을 증가시킬 수 있다. 송신 측은 hunting time(같은 전송률을 유지하는 시간)마다 상태를 갱신하며, 수신 측은 패킷을 받을 때 상태를 갱신한다.

loss가 발생하지 않고 hunting time이 경과하면 TR의 1%를 증가시킨다. 하지만 네트워크의 상황과 수신 측의 상황을 고려하여 증가시킨다. 현재의 전송률이 수신 측이 인식하는 전송률에 증가 분을 더한 값보다 작으면 전송률을 증가시키고, 그렇지 않으면 그대로 유지한다.

loss가 발생한다면 현재 전송속도를 TR의 3%만큼 감소시키고 송신 측의 상태는 DECREASE가 된다.

수신 측으로부터 loss 예측이 발생했을 때 송신 측의 상태가 INCREASE이면 현재의 전송률을 그대로 유지하고, DECREASE이거나 HOLD이면 TR의 1%만큼 감소하고 상태는 DECREASE가 된다. 그림 1은 수신 측의 상태에 대한 송신 측의 상태의 변화를 나타내는 그림이다.

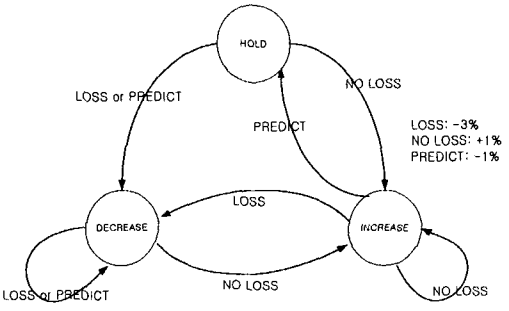

그림 1 송신 측의 상태변이도

송신 측의 각 상태는 hunting time동안 유지되며 hunting time동안에 전송률 조절을 1회로 제한하여 전송률의 잦은 변화를 제한한다. TCP는 RTT마다 윈도우를 1씩 증가시키기 때문에 본 논문에서도 TCP의 상황을 고려하여 hunting time을 RTT 와 loss rate의 합으로 결정한다. loss가 많이 발생할수록 hunting time이 증가하게 되어 그

만큼 전송률이 증가하는 속도가 느려지게 됨으로써 네트워크의 상황에 대해서 전송률 조절과 같은 전송률을 유지하는 시간의 조절을 통해 빠르게 대응할 수 있다.

3.2 수신 측의 기능

수신 측은 패킷을 수신하면서 loss와 loss 예측, 패킷 도착률을 계산하여 송신 측으로 전달한다. 피드백 패킷을 전달하는 시기는 송신 측에서 HUNT상태의 패킷을 보냈을 때, loss가 발생했을 때, loss예측이 발생했을 때이다. loss를 측정하는 방법에는 여러 가지 방법이 있지만 본 논문에서는 네트워크의 상황을 신속하게 인식하기 위해서 [4]에서 사용된 방법을 변형시켜 사용하였다. loss가 발생하는 시간간격을 weighted average로 구한다. 이 방법은 loss rate의 갑작스런 변화를 막는다. 평균 loss 구간은 S(n)은 다음과 같이 구해진다.

$$S(n)=\frac{\sum_{i=1}^{n} w_i\, s_i}{\sum_{i=1}^{n} w_i}+s_0$$

$$w_i=1,1\le i\le n/2,$$

$$w_i=1-\frac{i-n/2}{2/n+1},\ n/2<i<+n$$

n=8이면 weight는 1, 1, 1, 1, 0.8, 0.6, 0.4, 0.2이다 최종적인 loss rate는 1/S(n)이며, 수신 측의 상태를 LOSS로 변경하여 송신 측에 전달한다. 그림 2는 loss rate를 구하기위해 사용된 loss의 weighted interval을 구하는과정을 나타낸다.

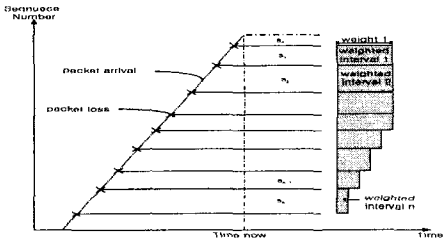

그림 2 loss rate를 계산하기 위해 사용된 loss 사이의 weighted interval

Loss 예측은 수신 측에서 패킷이 도착할 때마다 도착간격을 측정하여 평균도착 간격보다 1.8배 이내에 패킷이 도착하지 않으면 loss가 발생할 것으로 예측하고 피드백 정보를 송신 측으로 보낸다. 수신 측에서 보내는 피드백 정보는 loss rate, arrival rate, 마지막으로 받은 패킷의 timestamp로 구성된다. 마지막 패킷의 time stamp를 송신 측으로 되돌려 보냄으로서 송신 측에서 RTT를 측정하게 한다

Ⅳ. 실험 및 검토

본 논문에서 제안하는 알고리즘을 실험하기 위해 NS-2[7] 시뮬레이터를 이용하였다. 서로 다른 목표 전송률을 가진 stream이 다른 비율로 네트워크 대역폭을 할당받는지를 알아보기 위한 실험과 TCP와 대역폭을 공평하게 할당받는지를 알아보기 위한 실험이 이루어 졌다.

4.1 차별 화된 rate 조절

목표 전송률에 따라 차별 화된 rate 조절이 이루어지는지 알아보기 위해서 4개의 연결이 bottleneck 라우터를 공유한다. 라우터간의 연결은 1.5Mbps의 대역폭과 5msec의 Delay를 가진다. Sender 1은 800Kbps, 2는 600Kbps, 3은 400Kbps, 4는 200Kbps의 목표 전송률을 가지고 전송을 하게 된다. Sender 1은 0초에 시작하고, Sender 2는 5초, 3은 10초 4는 15초에 시작을 하고 100초에 시뮬레이션이 종료된다. 라우터는 Floyd와 Jacobson[8]에 의해 제안된 RED gateway이다.

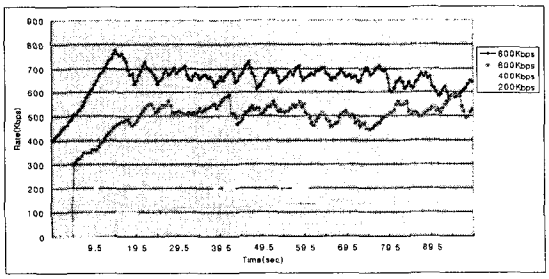

그림 3 IRDA의 전송률

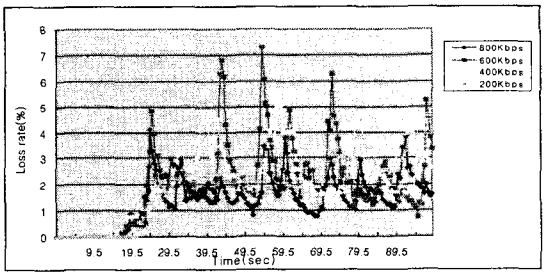

그림 4 LRDA의 loss rate

그림 3은 송신 측에서 측정한 전송률의 변화를 나타낸다. 링크 대역폭이 전체 요구하는 대역폭의 합보다 작기 때문에 요구하는 대역폭을 전송을 할 수는 없지만 차별화된 rate로 전송하고 있음을 알 수 있다. 요구 전송률이 높은 연결은 변화량이 크기 때문에 상대적으로 진동이 많다. 그림 4는 수신 측에서 측정한 loss rate를 나타낸 것이다. 그림 3의 rate와 비교한다면 loss가 발생한 시간에 바

로 전송률이 줄어듦을 알 수 있으며 rate 조절에 의해 loss가 많이 줄어듦을 알 수 있다.

4.2 TCP와 비교

다음 실험에서는 TCP와 LRDA와 대역폭을 공유하는지를 알아보기 위해서 4개의 TCP연결과 1개의 LRDA연결로 네트워크를 구성하였다. 라우터간의 대역폭은 3Mbps이며, delay는 5ms이고, 라우터는 이전 실험에서 사용된 것과 같은 RED gateway이다.

그림 5는 TCP와 LRDA의 전송률을 비교한 것이다. LRDA의 목표 전송률은 1Mbps이다. 50%인 500Kbps에서 전송을 시작한다. TCP의 대역폭 점유 때문에 크게 증가하지는 못하지만 TCP와 공평하게 대역폭을 점유함을 알 수 있다. bottleneck link의 대역폭이 3Mbps이고 총 5개의 연결이 있으므로 1개의 연결당 600Kbps의 대역폭을 점유한다면 공평하다. LRDA의 대역폭은 평균 600Kbps이며 TCP에 비해 전송률의 변화가 훨씬 적어서 스트리밍 전송에 적합함을 알 수 있다. 그림 6은 LRDA의 loss rate를 나타낸다.

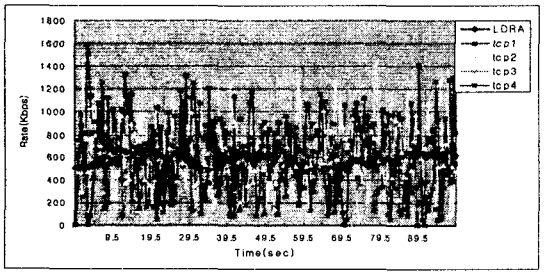

그림 5 TCP와 LRDA의 전송률 비교

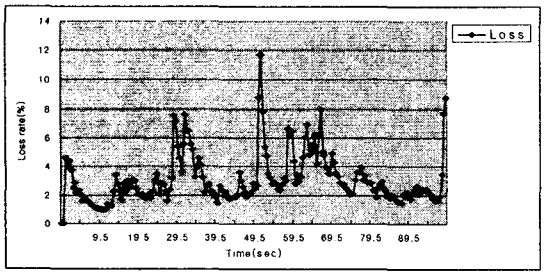

그림 6 LRDA의 loss rate

V. 결론

혼잡제어 기능이 없는 트래픽이 TCP와 같이 네트워크의 대역폭을 공유할 때 TCP가 더 적은 대역폭을 할당받도록 한다. 이를 해결하기 해서 많은 TCP-friendly rate control 알고리즘들이 제기되어 왔다. 하지만 기존에 제안된 TCP-friendly rate control 알고리즘들[2,4,5,6]은 TCP와의 공정한 대역폭 할당에 치중하여 다른 요구를 가지는 스트림에 대해서 차별 화된 서비스를 제공하지 못했다. 본 논문에서 제안된 방식은 서로 다른 대역폭 요구를 가지는 스트림에 대해서 loss와 delay값에 따라 차별 화된 전송률 조절을 제공할 뿐만 아니라 TCP와 공평하게 대역폭을 할당하였음을 시뮬레이션을 통해 알아보았다. 앞으로 실제 네트워크에서의 검증과 Unicast뿐만 아니라 Multicast상황도 고려한 연구가 더 필요할 것이다.

참고문헌

[1] K.Thompson, G.J.Miller, and R.Wilder. "Wide-area internet traffic patterns and characteristics". IEEE Network, 11(6), November/December 1997.

[2] S.Jacobs and A.Eletfheriadis. "Providing video services over networks without quality of service guarantees", RTMW'96, Oct. 1996

[3] S.Floyd and F.Kevin, "Router mechanism to support end-to-end congestion control", 8th Joint European Networking Conference, May. 1997

[4] S. Floyd, and Mark, "Equation-Based Congestion Control for Unicast Applications", SIGCOMM 2000, May 2000

[5] R.Rejaie and M.Handley and D.Estrin, "RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet", Proceedings of the IEEE INFOCOM'99 - Volume 3 March 1999

[6] D.Sisalem and H.Schulzrinne, "The loss-delay based adjustment algorithm : A TCP-friendly adaptation scheme" Proc. NOSSDAV98, July 1998

[7] UCB/LBNL/VINT Network Simulator - NS (version 2), http://www.isi.edu/nsnam/ns/

[8] S.Floyd and V.Jacobson, "Random early detection gateways for congestion avoidance", IEEE/ACM transactions on Networking, 1(4):397-413, Aug. 1993