

TCP-friendly RTP Rate Control

*하상석, 정선태

숭실대학교 정보통신전자공학부

e-mail : nessus75@syscon.ssu.ac.kr

TCP-friendly RTP Rate Control

*Sang-Seok Ha, Sun-Tae Chung

School of electronic engineering, college of engineering
Soongsil university

Abstract

TCP is taking over 95% among the Internet traffics. Recently the demands of multimedia services in the Internet has been increasing. These multimedia services mostly need real-time delivery, and then RTP has been a de facto to transmission protocol for these real-time multimedia services. RTP uses UDP as its underlying transport protocol, and thus it does not support any rate and congestion control. Thus, for fair use of the Internet bandwidth with TCP traffics. RTP also needs a rate control. One constraint of RTP is that the feedback information(delivered by, RTP's twin protocol, RTCP) is recommended to be sent no less than 5 seconds.

In this paper, we propose a TCP-friendly RTP rate control which use only RTCP feedback information at every 5 seconds. The experiment results show that our proposed algorithm works. But, it is found that we need more time to test the effects of parameters and policies of the algorithms, which will be reported later.

I. 서론

현재 인터넷 서비스의 다변화로 인하여 인터넷에서 오디오 및 비디오 서비스(화상 및 음성전화, 음악방송 등)와 같은 멀티미디어 서비스의 수요가 증가하고 있다. 이러한 실시간 서비스에는 TCP[5]의 적용이 힘들다. 첫 번째로 연결지향형 서비스 이기 때문에 멀티캐스팅에 문제가 있다. 두 번째로 패킷 손실시 재전송을 하기 때문에 문제가 있다. 세 번째로 세그먼트가 만들어질 때 시간정보를 가지고 있지 않다. 이러한 이유로 UDP[5]를 이용해서 실시간 데이터를 보내게 되지만 UDP의 경우 TCP와 달리 Congestion control을 하지 않기 때문에 네트워크의 95% 이상을 TCP가 점유하고 있는 현재 네트워크 사용에 있어서 불공정한 결과를 초래하게 된다. 이러한 문제점을 해결하기 위해 TCP와 유사한 Congestion control[5] 방법을 이용하여 TCP 트래픽과 대역폭을 공정하게 이용하면서 네트워크의 체증을 방지한다. RTP[1]를 이용하여 실시간 데

이터를 전송하고 RTCP[1]를 통하여 Congestion control 하는것의 가장 큰 문제점은 그 전송간격이 5초 이상이라는 것이다. 수신측에서 긴 전송시간 동안의 패킷 손실 정보를 종합하여 송신측으로 보내주게 되고 이 결과 정보를 이용하여 앞으로의 전송간격 동안 전송률을 조절 함으로써 문제점을 해결하였다.

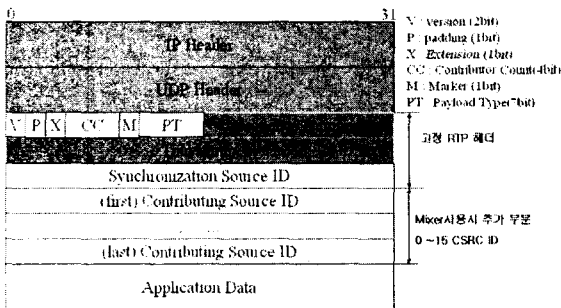
2장에서는 RTP[1] 및 RTCP에 대해서 소개하고, 3장에서는 TFRC(TCP-friendly Rate Control)[3], 수신측의 기능과 송신측의 기능에 대해서 설명하고, 이후 4장에서는 실험 및 고찰, 5장에서는 결론 및 향후 연구 방향에 대해서 설명한다.

이 논문의 첫 번째 목적은 현재 네트워크의 주된 프로토콜인 TCP[5]와 공정한 대역폭을 사용하면서 실시간 데이터 전송에 맞는 Rate Control을 하는 것이고, 두 번째 목적은 RTP에 알맞도록 알고리즘 및 예러의 패턴을 분석하여 앞으로 수초동안의 네트워크 상태를 예측 함으로써 피드백 트래픽을 줄이는데 그 목적이 있다.

II. RTP 및 RTCP

2.1 RTP

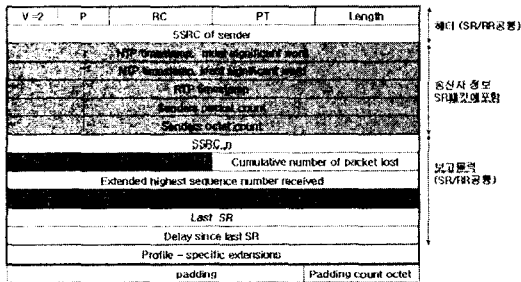
RTP[1]는 오디오, 비디오 및 시뮬레이션 데이터와 같은 실시간 데이터를 멀티캐스트 또는 유니캐스트 네트워크를 이용해서 전송하는 응용 서비스에 알맞은 단말-대-단말 네트워크 전송 기능을 제공한다. RTP는 자원 예약을 수행하지 않으며, 따라서 적시 전달, 순차 전달과 같은 서비스 품질도 보장하지 않는다. 하지만 실시간 데이터를 전송할 수 있도록 헤더에 Sequence Number 와 Timestamp를 넣어서 전송하게 되고, [그림 1]은 RTP header의 그림이다[1].



[그림 1] RTP header

2.2 RTCP

RTCP[1] 패킷은 SR 패킷, RR 패킷, SDES 패킷, BYE 패킷, APP 패킷으로 구분할수 있다[1].



[그림 2] RTCP RR / SR packet header

RTCP패킷은 세션의 참가자들에게 주기적으로 전송하기 때문에 각 참가자는 세션의 전체 참가자의 수를 파악할수 있고, 전송간격이 조정된다. [그림-2]는 RTCP 패킷중 자신이 송신자일 경우에 보내는 SR 패킷과, 수신자 일 경우에 보내는 RR 패킷의 헤더를 보여주고 있다. 최대 31개의 수신 보고 블록을 포함할수 있으며, 수신자는 이 보고패킷을 통하여 현재의 상태를 송신자에게 보내준다.

III. 제안된 Rate Control

3.1 TFRC (TCP-friendly Rate Control)

TFRC는 TCP 트래픽의 Congestion Control과 비슷한 방법을 사용하여 UDP 또는 RTP 트래픽을 컨트롤하는 방법이다. [3]으로부터 TCP 응답 함수를 보면 다음과 같다[2].

$$T = \frac{packetsize}{RTT \sqrt{\frac{2p}{3} + t_{RTO} \left(3 \sqrt{\frac{3p}{8}} \right) p (1 + 32p^2)}} \quad (1)$$

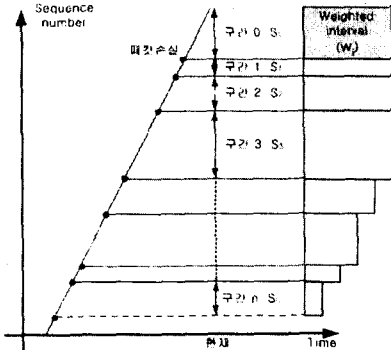
이것은 UDP 및 RTP를 이용해서 데이터를 전송할 때 최대로 전송할수 있는 기준을 제시해준다.

(p:패킷손실확률, t_{RTO}:재전송 타임아웃)

3.2 Receiver 기능

TCP 및 기존의 UDP[5]를 이용해서 데이터를 전송시 각 패킷에 대한 결과를 ACK 신호를 통하여 받게 되고 네트워크의 상태를 체크하게 된다. 하지만 RTP의 경우에는 각 패킷에 대한 패드백 정보는 세션에 참가하고 있는 사용자의 수와 대역폭이 고려되어 전송간격이 계산된다. RTP를 이용해서 Rate control을 하게 될 경우 고려하여야 할 사항중에 가장 큰 문제는 현재 네트워크의 상태를 보고 받을수 있는 RTCP 패킷이 최소 5초 단위로 전송이 된다는 것이다. 이 하나의 RTCP 패킷으로 네트워크의 상태를 파악하여 앞으로의 전송률을 결정하여야 한다. 이러한 전송간격을 보완하기 위하여 [그림 3]과 같은 방법[2]을 사용한다. 수신측에서는 각 패킷을 관찰하고 손실된 패킷의 시퀀스 넘버를 저장하게 된다. n개의 패킷손실 구간에 대하여 시간의 가중치를 주어 최근값에 더욱 충실되되 과거의 값도 반영을 할수 있도록 하는 것이 그 목적이 된다. 가중치는 (2)와 같이 구하게 되고, 각 손실구간 S_i에 곱한다. 데이터 전송후 제일 처음 발생하는 손실에 대하여 덜 민감하게 반응하게 하기 위하여 각 손실구간에 초기값을 넣어주었다. 손실구간은 서클큐로 이용하여 구현 하였으며 손실이 발생할 때마다 그 값을 큐에 넣어서 (3)과 같이 마지막 에러가 난 구간까지의 값을 이용하여 S값을 계산한다. 또한 마지막 에러가 발생한 구간부터 현재 데이터를 받은 순간까지의 값을 (4)와 같이 계산을 한다. 이후 MAX(S(1,n), S(0,n))를 구하여 값을 선택한다. 이렇게 계산된 S값은 평균적으로 패킷손실이 발생하는 값이 되므로 역수를 취함으로써 패킷손실확률을 구할수 있다[2].(p = 1/S) RTP에 적용시 알고리즘 특성상 네트워크의 상태가 좋음에도 불구하고 순간의 버스트 에러 발생시 전송률이 급격히 감소하게 되고, 느린속도로 증가하게되는데, 수신측에

서 마지막 에러발생부터 현재까지의 값을(S_0) 계산할 때 a 크기 만큼의 가중치를 주어서 문제를 해결하였다. 이후 계산된 값을 RTCP를 통하여 송신측으로 보내주게 됨으로써 수신측의 역할은 끝난다.



[그림 3] 가중치 부여된 손실구간

$$w_i = 1 - \frac{i - \frac{n}{2}}{\frac{n}{2} + 1}, \left(\frac{n}{2} \leq i \leq n\right)$$

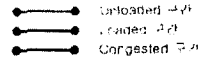
$$w_i = 1, (1 \leq i \leq \frac{n}{2}) \quad (2)$$

$$S(1, n) = \frac{\sum_{i=1}^n w_i s_i}{\sum_{i=1}^n w_i} \quad (3)$$

$$S(0, n) = \frac{\sum_{i=0}^n w_i s_i}{\sum_{i=1}^n w_i} \quad (4)$$

3.3 Sender 기능

RTCP를 통하여 현재의 네트워크 상태를 최소 5초 간격으로 보고를 받는다. RTCP에 포함되어있는 S 값을 이용하여 (1)을 통하여 최대전송률을 구하게 되고, 이 값을 넘지 않는 범위에서 Rate Control을 하게 된다. RTCP를 통하여 RTCP전송간격 동안의 패킷손실에 대한 정보를 얻을수 있고, 그 값을 [그림 4]와 같이 세 구간으로 나누어 컨트롤 하게 된다. 구간 1은 Unload구간, 구간 2는 Loaded구간, 마지막으로 구간 3은 Congested구간으로 분류된다. 구간 1은 손실이 없는 경우, 구간 2는 손실이 4개 이하인 경우, 구간 3은 손실이 5개 이상인 경우일때 실험상으로 좋은 결과를 얻을수 있었다.



[그림 4] 컨트롤 구간

각 구간에 대하여 [표 1]에서 보이는 방법으로 전송률을 조절한 후 수신측으로 데이터를 전송함으로써 Congestion control 한다.

(r_i : congested 구간에서의 sending rate)

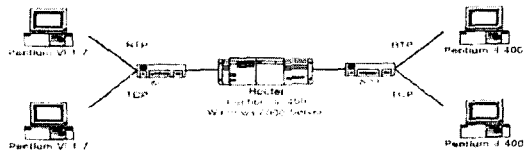
네트워크 상태가 좋음에도 불구하고 발생한 순간의 버스트 에러 문제를 sender측에서는 $T(1)$ 가 $2 \times r_1$ 보다 작은 경우 T값을 $k \times T$ 값으로 적용함으로써 해결하였다. ($0.6 < K < 1$)

구분	Sending Rate
구간 1(unloaded)	if Sending rate < Maximum rate $r_i = r_{i-1} + \frac{r_{i-1}}{\delta \times RTT}$ if Sending rate > Maximum rate $r_i = r_{i-1} - \frac{r_{i-1}}{\gamma}$
구간 2(loaded)	$r_i = r_{i-1} - \frac{r_i}{\gamma}$
구간 3(congested)	$r_i = \frac{r_{i-1}}{2}$

[표 1] 구간에 따른 전송률

IV. 실험 및 검토

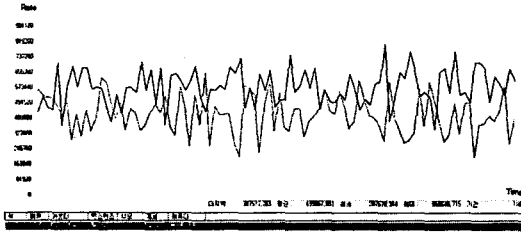
[그림-5]는 RTP rate control을 실험하기 위하여 구성한 실험환경을 보여준다. Windows 2000 Server를 이용하여 라우터를 구성하였으며, TCP 및 RTP 전송을 위한 펜티엄 컴퓨터 4대, 허브 2개로 구성하였다.[4]



[그림 5] 실험환경

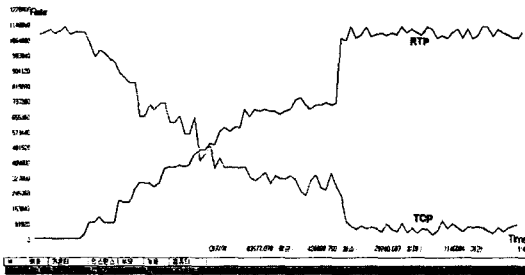
먼저 기존의 TCP의 Congestion control[5] 동작에 대한 그림을 보면 [그림 6]과 같다. 그림에서 보면 TCP간의 공정한 대역폭 사용을 볼 수 있다.

[그림 7]은 네트워크의 상태를 고려하지 않고 흐름제어를 하지 않는 RTP를 통하여 실시간 데이터의 양을 점차적으로 증가 시켰을 경우에 TCP와 RTP의 그림이



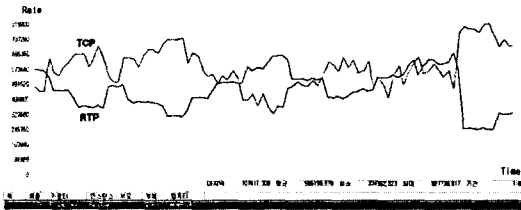
[그림 6] TCP간의 실험결과

다. 시간이 지날수록 TCP 프로토콜 특성상 적은 대역폭을 사용하게 되고 RTP는 많은 대역폭을 사용하게 되는 불공정한 대역폭 사용을 보여주고 있다.



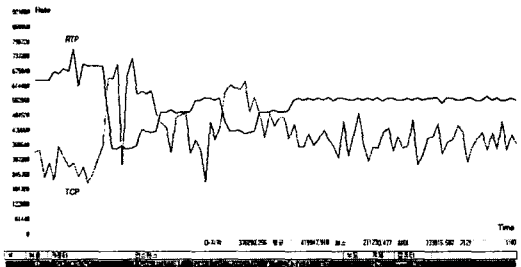
[그림 7] 제어안하는 RTP와 TCP

[그림 8]에서는 제안하는 Rate control을 사용하여 RTP를 통하여 데이터를 보내는 경우의 대역폭 사용을 나타내는 그림이다. [그림 7]과는 달리 TCP와 RTP가 대역폭 사용에 있어서 공정함을 알수 있다.



[그림 8] 제안하는 방법을 이용한 RTP

[그림 9]는 버스트 에러가 발생했을 경우 적은폭으로 전송률이 증가되는 것을 제안된 방법을 이용하여 향상 시킨 결과를 보여주고 있다.



[그림 9] RTP에 알맞도록 개선 제안하는 Rate control 방법을 사용하였을 경우 기준

의 방법인 패킷마다 ACK신호를 받아서 Rate control 하는 경우와 비슷한 결과를 얻을수 있었다. 그러나 RTCP 보고 패킷은 최소 5초 마다 전송이 되어지기 때문에 그 사용자가 많아 질 경우 네트워크에서 무시 못 하는 부분을 차지하는 피드백 트래픽을 줄일 수 있는 효과를 내포하고 있다.

V. 결론

앞으로 시간이 흐를수록 멀티미디어 데이터와 같은 실시간 데이터의 사용량이 점점 더 많아질 것이며, 이러한 데이터를 보내기 위하여 RTP/RTCP가 사용되어 질 것이다. 하지만 프로토콜 특성상 기존의 TCP와 불공정한 대역폭 사용이 예상되므로 실시간 데이터의 특성을 살리면서 네트워크 변화에 신속하게 대응할 수 있는 RTP Rate control 방법을 제시하였다. 본 논문에서 사용되어진 파라미터 값들은 실제 네트워크 망에서 최적화 된 값들이 아니기 때문에 향후 실제 망의 특성을 적용하여 최적화된 값을 설정해야 하는 연구와 무선망에서의 손실과 유선망에서의 손실을 구분하여 RTCP를 통하여 송신측으로 전송한 후 각각의 환경에 맞는 Rate control 하는 연구가 남아있다.

참고문헌

- [1] H. Schulzrinne, S. Casner, R. Frederick, Van Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, Jan. 1996
- [2] Sally Floyd, Mark Handley, Jitendra Padhye and Jörg Widmer "Equation-Based Congestion Control for Unicast Applications", ACM SIGCOMM, pp.43-56, August 2000
- [3] Pages: J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. SIGCOMM Symposium on Communications Architectures and Protocols, pp43 - 56, Aug. 1998.
- [4] MaSaki Miyabayashi, Naoki Wakamiya, Masayuki Murata and Hideo Miyahara "Implementation of Video Transfer with TCP-friendly Rate Control Protocol", Proceedings of International Technical Conference on Circuits / Systems, Computers and Communications 2000 1, pp. 117-120, July 2000
- [5] W.Stallings, "High-speed networks: TCP/IP and ATM design principles", Prentice Hall, 1998