

인터넷에서 실시간 스트리밍 제어를 위한 피드백 정보 특성분석

신상철, 노병희
아주대학교 정보통신전문대학원 인터넷 연구실
e-mail : msxfan@nownuri.net

Analysis on the characteristics of feedback information for control real-time streaming in the internet

Sang-Chol Shin and Byeong-hee Roh
Internet Lab, Graduate School of Information and Communication, Ajou University

요 약

실시간 멀티미디어의 전송시 네트워크의 상황에 따라서 서비스의 품질이 급변하게 된다. 이 경우 QoS 를 보장하기 위해서 일반적으로 피드백에 의한 제어방법을 사용하게 된다. 본 논문에서는 UDP 패킷 전송실험을 통하여 실시간 멀티미디어 전송에서의 QoS 를 보장하기 위한 피드백 정보의 이용가능성에 대해 살펴보고 실험 데이터를 여러가지 각도로 분석해보았다. 피드백 정보의 문제점을 살펴보고 개선방안을 제시하였다.

1. 서론

현재 초고속인터넷이 발달됨에 따라 AOD, VOD 와 같은 실시간 멀티미디어 전송 서비스가 많이 사용되고 있다. 실시간 전송 서비스는 일반적인 자료 전송과 같은 비 실시간 전송과는 달리 지연(delay)이나 지터(jitter)에 민감하다. 그러나 현재의 인터넷은 효율성을 우선으로 비 실시간 전송에 적합하게 만들어져 있기 때문에 사용 대역폭(bandwidth)을 안정적으로 보장받을 수 없다. 네트워크를 여러 사용자가 함께 사용하기 때문에 한 사용자가 이용할 수 있는 대역폭은 계속적으로 변하고, 네트워크의 혼잡(congestion)이 발생하면 패킷의 손실(loss)과 지연이 급격히 증가하여 서비스의 품질이 급격히 저하된다[1]. 이러한 네트워크의 혼잡을 피하고 일정한 서비스질을 유지하기 위해서는 어플리케이션 차원에서의 제어가 필요하다.

실시간 전송의 QoS(Quality of Service)를 향상시키기 위한 방법으로는 여러가지 방법이 있으나 일반적인 유선환경하에서는 피드백(feedback) 정보를 바탕으로 한 전송율제어(rate control) 방법을 주로 사용한다. 송신자는 피드백 정보를 통하여 지연시간과 손실율과 같은 네트워크 상황에 대한 정보를 얻고 그에 맞춰

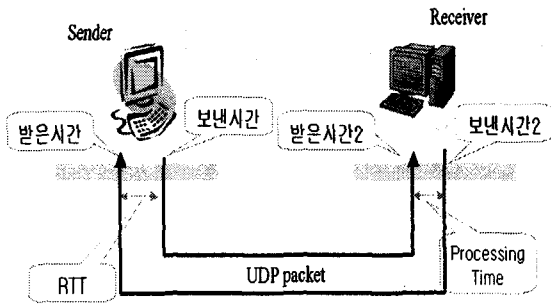
전송율을 조정하여, 네트워크의 혼잡을 피하고 대역폭을 효율적으로 사용할 수 있게 한다. 피드백정보를 이용한 제어방법에는 rate control, rate adaptive video encoding, rate shaping, delay-constrained retransmission 방법등이 있다. [2]

그러나 이와 같은 피드백정보는 이미 전송된 패킷들을 통해서 얻은 정보이기 때문에 과거의 네트워크에 대한 정보이다. 본 논문에서는 실험을 통해서 새로운 패킷을 전송할 때 이와 같은 정보를 그대로 이용하는 것이 타당한지 살펴보고, 그것의 문제점과 개선방향에 대해 제시하고자 한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 트래픽을 측정하기 위한 실험방법에 대해서 설명하였다. 3 장에서는 실험 결과 얻어진 트래픽을 분석하였다. 4 장에서는 실험결과를 정리하고 앞으로의 연구방안을 제시하였다.

2. 실험방법

실제 네트워크환경에서 패킷의 지연 특성을 알아보기 위하여 UDP 패킷을 전송하고 지연시간을 측정하



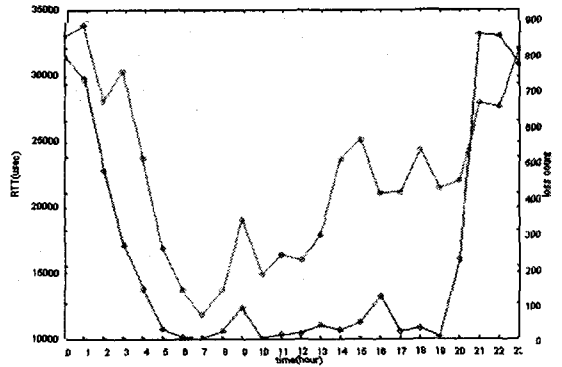
<그림 1> 실험방법

는 실험을 하였다. 윈도우 환경하에서 시간값을 측정하기 위해 WinPcap[3]라이브러리를 사용하여 송신자와 수신자 프로그램을 만들었다. 송신자는 수원 아주대학교 연구실에 설치하고, 수신자는 서울 역삼동에 설치하여 UDP 패킷을 전송하였다.

그림 1 에서와 같이 송신자에서는 100 bytes 크기의 UDP 패킷을 만들어 100ms 간격으로 일정하게 전송하고, 수신자에서는 이 패킷을 받아서 다시 송신자에게 그대로 회신한다. 이를 통해 송신자에서는 보낸시간과 받은시간의 차이를 구해서 RTT(Round Trip Time)값을 계산한다. 수신자에서는 패킷을 받은시간과 보낸시간을 측정하고, 다음번 패킷을 송신자에게 전송할 때 이 값을 기록하여 전송한다. 양쪽 모두 MS Windows 2000 환경의 PC 를 사용하였다. 실험기간은 2003년 2월 21일 금요일 20시 43분부터 2003년 2월 22일 토요일 20시 43분까지 24시간동안이다.

<표 1> 시간대별 RTT 와 손실율

Time (Hour)	RTT (μs)	Loss packet
0	33056	776 (2.15%)
1	34038	714 (1.99%)
2	28135	464 (1.29%)
3	30437	260 (0.72%)
4	23756	138 (0.38%)
5	16888	29 (0.08%)
6	13706	7 (0.02%)
7	11763	4 (0.01%)
8	13677	23 (0.06%)
9	19067	88 (0.24%)
10	14857	3 (0.01%)
11	16402	15 (0.04%)
12	15969	17 (0.05%)
13	17885	40 (0.11%)
14	23711	26 (0.07%)
15	25224	48 (0.13%)
16	21103	119 (0.33%)
17	21141	22 (0.06%)
18	24400	31 (0.86%)
19	21454	8 (0.02%)
20	22029	220 (0.61%)
21	28022	837 (2.33%)
22	27746	833 (2.32%)
23	32149	755 (2.10%)
Total	22315	5477 (0.63%)



<그림 2> 시간대별 RTT 및 손실율 변화

3. 실험결과

24 시간동안 총 879620 개의 UDP 패킷을 전송하였고, 평균 RTT 값은 22315 μs 였다. 시간대별 평균 RTT 값과 손실율을 표 1 에 나타내었다. 심야 시간대에 RTT 값과 손실율이 증가하고, 새벽에는 이용자가 적기 때문에 작게 나오는 것을 볼수있다. 여기서 손실율과 RTT 값은 비례관계에 있음을 확인할 수 있다.

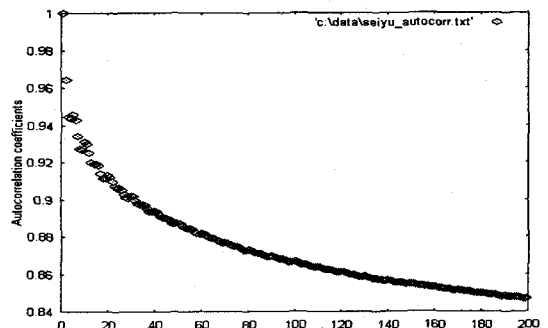
3.1 샘플간의 자기상관도(Autocorrelation)

먼저, 피드백 정보의 이용가능성을 살펴보기 위해 RTT 값의 자기상관도를 구하였다. 이는 하나의 샘플이 있을 때, 그 뒤로 오는 샘플들의 RTT 값이 샘플 간격(lag)이 벌어짐에 따라 얼마만큼 유사함을 보이는 지 나타낸다. 이것은 다음과 같은 식에 의해 구해진다.

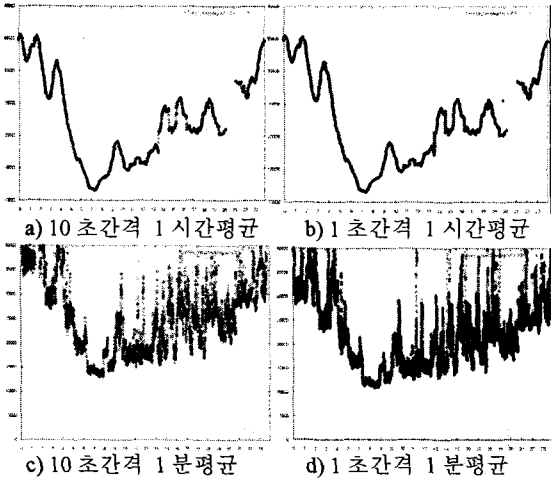
$$r_k = \frac{\sum_{i=1}^{N-k} (Y_i - Y_{avg})(Y_{i+k} - Y_{avg})}{\sum_{i=1}^N (Y_i - Y_{avg})^2} \quad [4]$$

여기서 r은 자기상관도, Y는 RTT 값, Y_{avg}는 RTT의 평균값, N은 전체 샘플 개수, k는 샘플간격이다.

실험결과, 상관도가 높게 나오는 것을 볼 수 있었다. 약 30 개 즉 0.3 초의 간격에서는 0.9의 상관도를 보였고, 200 개 즉 2 초의 간격에서는 0.85의 상관도를 보였다. 이를 통해 어느정도 시간이 지난 뒤에도 RTT 값의 변화가 크지 않음을 알수있고, 피드백 정보를 사용함에 큰 문제가 없음을 보여준다.



<그림 3> RTT 값의 자기상관도

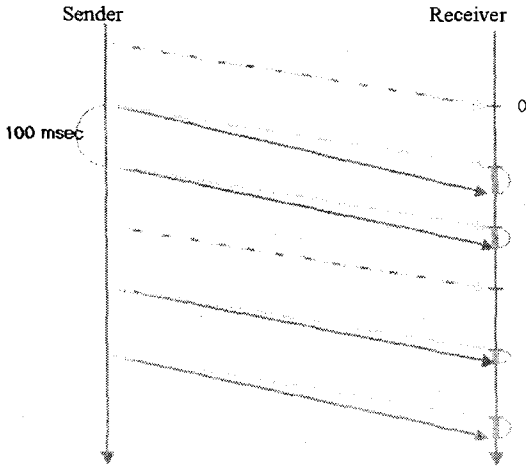


<그림 4> 평균 RTT 추적

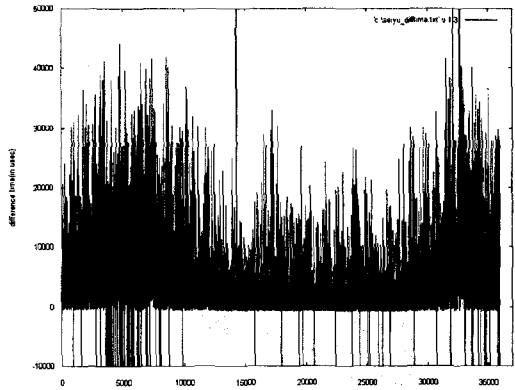
3.2 평균 RTT 추적(trace)

그림 4 는 RTT 의 평균값을 일정한 시간 간격으로 구해서 그래프로 나타낸 것이다. 예를들어 a) 그래프는 1 시간동안의 RTT 평균을 구하고, 10 초뒤에 다시 1 시간동안의 RTT 평균을 구하고 이를 반복한다. 1 분 평균의 이것은 전체적인 네트워크 상황의 변화를 살펴보고, 그에 대해 피드백 정보의 활용성을 알아보기 위해서이다. 그래프를 보면 1 시간평균을 봤을때는 보이지 않던 스파이크들이 돌출하는 것을 볼 수 있다. d) 그래프를 보면 이러한 스파이크 상황에서도 RTT의 급변함을 잘 추적하고 있음을 보인다.

그러므로 앞서 설명한 자기상관도와 이 RTT 추적 그래프를 통해서, 피드백 정보가 비록 과거의 네트워크에 대한 정보이지만 그 간격이 크지 않다면 적용시킴에 무리가 없음을 알 수 있다.



<그림 5> 수신자의 패킷수신 차이시간



<그림 6> 차이시간 그래프 (16 시)

3.3 수신자의 패킷수신 차이시간

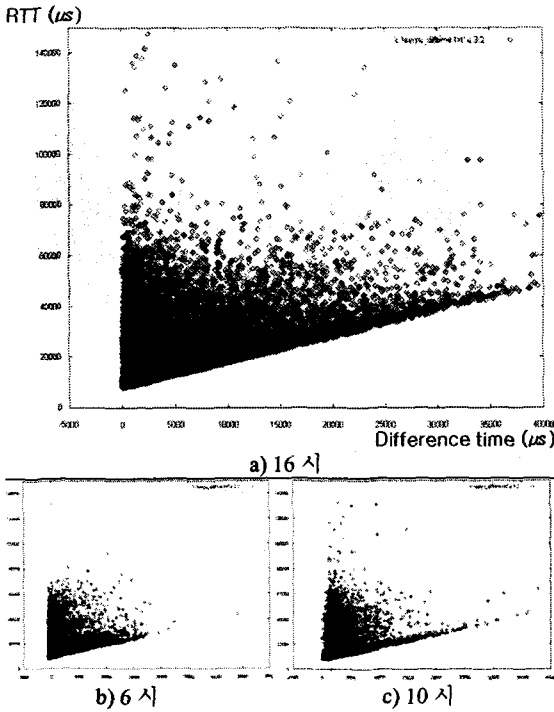
여기서 패킷수신 차이시간(difference time)이란 수신자가 패킷을 수신할 것으로 기대하는 시간과 실제로 도착한 시간의 차이로 정의한다. 이 값을 통해 수신측의 버퍼크기나 재생지연시간을 결정할 수 있다. 또한 손실율과의 비교를 통해 네트워크가 어떤 특성을 가지는지 알 수 있다. 만약 차이시간과 손실율이 비례한다면, 네트워크 중간단의 라우터들에서 버퍼 오버플로우(buffer overflow)로 인한 손실이 발생하는 것을 나타낸다. 이 경우 전송율을 낮게하여 손실을 줄이는 방법이 요구된다. 반면 둘 사이에 비례관계가 없다면 라우터의 버퍼와는 관련이 없으므로 수신측의 버퍼를 크게 잡고 재생지연시간을 길게 가져가야 한다.

패킷수신 차이시간을 구하는 방법은 다음과 같다. 수신자가 최초의 패킷을 받은 시간을 0 으로 두고, 그림 5 에서와 같이 그 이후 패킷들의 도착시간과 송신자의 패킷전송 간격과의 차이를 계산해 차이시간을 구한다. 마지막으로, 결과값중 음수의 최소값을 모든 값에 더하여 최소값을 0 으로 만든다. 여기서 한가지 고려해야할 중요한 점이 있는데 서로간의 clock 의 속도가 다르다는 것이다. 이것은 뒤에서 설명하겠다.

그림 6 은 16 시 한 시간동안의 차이시간과 패킷손실을 나타낸다. 여기서 0 아래로 떨어지는 선들은 패킷손실이 일어났음을 표시하기 위해 그려놓은 것이다. 여기서 패킷손실율과 차이시간간에 비례관계가 있음을 볼 수 있다.

다음으로 RTT 와 차이시간을 비교하였다. RTT 값이 커질수록 수신자가 받는 시간도 늦어질 것이기 때문에, 패킷수신 차이시간도 길어질 것이고 둘 사이에 비례관계가 있을것으로 예상하였다. 그러나 예상과는 달리 V자형의 그래프를 볼 수 있었다.

그림 7 의 a)는 같은 시간인 16 시의 차이시간과 RTT 를 비교한 것이다. 이 그래프는 크게 3 가지 부분으로 나누어 볼 수 있다. 대각선으로 길게 뻗어나가는 부분과 수직으로 올라가는 부분, 그리고 그 사이의 부분으로 나누어 볼 수 있다. 여기서 한가지 생각해야 할 것은 RTT 는 왕복시간이기 때문에, 순방향 지연시간과 역방향 지연시간의 합이라는 것이다.



<그림 7> 차이시간 vs RTT

패킷 전송시 순방향 링크와 역방향링크는 반드시 같으라는 보장이 없기 때문에 각 링크의 네트워크 상황이 서로 다를 수가 있다. 그림에서 대각선 부분은 순방향 지연시간이 증가하지만 역방향 지연시간은 변화가 없는 경우이고, 수직부분은 반대로 역방향 지연시간은 증가하지만 역방향 지연시간은 차이가 없는 경우이며, 그 중간부분들은 양방향 모두 지연시간이 증가하는 경우이다. b)와 c)의 그림들은 각각 6 시와 10 시의 경우인데 역시 비슷한 양상을 보여주고 있다.

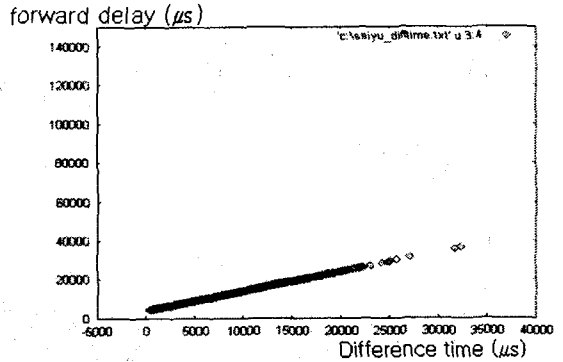
이와 같은 사실은 피드백 정보를 이용할 때 단순히 RTT 값을 사용하는 것은 네트워크 상황을 잘못 반영할 수 있다는 것을 말해준다. 실시간 멀티미디어 전송에 있어서 중요한 것은 서버에서 클라이언트까지의 단방향 링크뿐이다. 역방향의 경우는 피드백을 위해서 이용될 뿐, 실제 데이터전송과는 상관이 없다. 예를 들어, RTT 값이 증가하는 경우 순방향링크의 상황은 좋지만 역방향링크의 네트워크 상황이 안좋아서 지연이 길어지는 경우일 수도 있다. 따라서, 피드백 정보를 이용할 때 RTT 값을 사용하는 것 보다 순방향 지연시간을 구해서 적용하는 것이 보다 올바른 방법이라 하겠다. 순방향 지연시간을 구하는 방법은 다음과 같다.

$$[X] = [\text{받은시간2}] - [\text{보낸시간1}] + \text{clock싱크수정} - \text{RTT}/2$$

$$\text{순방향지연} = [\text{받은시간2}] - [\text{보낸시간1}] + \text{clock싱크수정} - [X]$$

여기서 X는 송신자와 수신자간의 시간차를 뜻한다. Clock 싱크수정이 필요한 이유는 송신자와 수신자간에 PC의 시간이 흐르는 속도가 다르기 때문이다. 이

본 논문은 한국과학재단(R05-2002-000-00829-0)의 지원하에 수행되었음



<그림 8> 차이시간 vs 순방향 지연시간

실험에서는 1 시간당 약 200ms, 하루동안 약 5 초의 시간차이가 있었다. 앞서 설명한 차이시간과 순방향 지연시간을 계산하기 위해서는 이러한 차이를 보정해 주어야 한다. 이 방법으로는 순방향 지연시간의 절대값은 알수 없지만, 절대값에서 일정하게 가감된 값은 구할 수 있다. 즉, 한 순간의 정확한 값은 알수 없지만 일정기간의 변화양상은 구할 수 있다. 이것은 좀 더 개선해야 할 내용이다.

4. 결론

본 논문에서는 UDP 패킷 전송실험을 통하여 전송 지연을 측정하고 네트워크 상황에 따른 피드백 정보의 이용에 대해 알아보았다. RTT 값의 자기상관도와 평균 RTT 추적을 통해서, 피드백 정보를 사용시 네트워크 상황에 대한 정보를 큰 오차없이 얻을 수 있고 전송을 제어에 적용할 수 있음을 알았다. 수신자측의 패킷수신 차이시간과 손실을 비교를 통해 네트워크 상황에 대한 판단을 할 수 있었다. 또한 차이시간과 RTT 값의 비교를 통해 단순히 RTT 값을 사용하는 대신 순방향 지연시간을 적용시켜야 올바른 네트워크 상황을 반영할 수 있음을 알았다.

앞으로 이러한 결론을 바탕으로 실시간 멀티미디어 데이터를 전송시 피드백 제어를 위한 방법을 찾기 위한 추가 연구가 필요하다.

참고문헌

- [1] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413.
- [2] DAPENG WU, "Transporting Real-Time Video over the Internet: Challenges and Approaches", Proceeding of IEEE vol 88, Dec. 2000.
- [3] <http://winpcap.polito.it>
- [4] Roy D. Yates, "Probability and Stochastic Processes", JOHN WILEY & SONS, pp. 215-219.