

# Internet Router를 위한 Active Queue Management의 성능 개선

## Performance Improvement of Active Queue Management for Internet Routers

°임 혁, 박경준, 박은찬, 최중호\*

\* 서울대학교 전기컴퓨터공학부 (Tel: 82-2-880-7313, E-mail: chchoi@csl.snu.ac.kr)

**Abstract:** In this paper, we propose a control scheme for improving the performance of a conventional Proportional-Integral (PI) controller for Active Queue Management (AQM) supporting TCP flows. When the PI controller is used for AQM, the windup phenomenon of the integral action causes the performance degradation. Therefore we model AQM as a system with a saturator and apply anti-windup methods to the PI controller for AQM. We compare the performances of anti-windup algorithm with the conventional PI controller through ns simulations. The simulation results show that the PI controller with anti-windup method performs better than the conventional PI controller.

**Keywords :** Active queue management, proportional-integral control, anti-windup method

### 1. 서론

Drop-tail을 사용하고 있는 현재의 TCP 혼잡 (congestion) 제어는 여러 문제를 가지고 있다. Drop-tail 큐가 사용된 네트워크에서 TCP 소스는 라우터의 큐 오버플로우에 의한 패킷 손실이 발생한 후에야 자신의 패킷 전송 속도를 줄인다. 패킷 손실과 이의 발견 사이에는 상당한 시간이 걸린다. 그러나 이 시간동안에도 TCP 소스는 네트워크가 감당할 수 없는 전송속도로 계속 패킷을 보내기 때문에 라우터에서는 상당한 양의 패킷이 손실된다. 이러한 drop-tail 큐에서의 패킷 손실은 TCP 소스들의 global synchronization을 발생시킬 수도 있다 [1].

Drop-tail 큐가 가지고 있는 이러한 문제를 줄이기 위해서 Random Early Drop (Red)가 Active Queue Management (AQM)의 방법으로서 제안되었다 [2]. 그러나 RED 알고리즘은 다음의 몇 가지 단점을 가지고 있다. 첫째 RED의 파라미터 설정에 관한 정해진 방법이 아직 없다 [3][4]. 라우터의 버퍼가 작은 경우에는 RED의 사용이 적절치 않다는 연구 결과도 발표되었다 [3]. 이러한 RED의 문제들을 해결하기 위해 RED의 변형들이 많이 제안되고 있다 [4][5][6]. 최근에는 제어 이론에 바탕을 둔 방법들이 제안되고 있다 [7][8][9][10]. [7]과 [8]에서는 RED에 대한 해석과 성능 개선을 위해 비례 적분 (PI) 제어기를 설계하였다. [9]과 [10]에서는 큐의 안정화에 초점을 맞추었으며 적분 제어기를 제안하였다.

그러나 비례 적분 제어기가 AQM에 사용되면 그 제어 출력이 0과 1사이에 제한되기 때문에 “windup” 현상이 발생한다. 제어기는 패킷을 drop하기 위해서 출력을 양의 값을 갖도록 할 수 있지만 트래픽을 늘리기 위해 음의 값을 갖도록 할 수는 없다. 이러한 제한 때문에 TCP 트래픽이 적은 상태로 상당히 오래 지나면 적분 제어기의 출력이 매우 커지는 현상이 발생한다. 이를 “windup” 현상이라고 한다. 이 현상은 saturator를 갖는 시스템에 적분제어기가 사용될 때 나타난다. 이 현상이 발생하

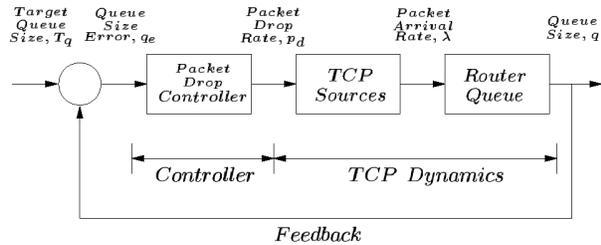


그림 1. TCP의 혼잡 제어의 되먹임 시스템 모델

면 적분기에 누적된 값 때문에 제어기의 성능이 저하된다.

본 논문에서는 이 문제를 해결하기 위해서 AQM을 saturator를 갖는 시스템으로 모델링하고 anti-windup 방법을 적용한다 [11]. 시뮬레이션 결과를 통해서 anti-windup을 적용한 제어기가 기존의 비례 적분 제어기 보다 트래픽의 변화가 심한 실제 네트워크 환경에서 좋은 성능을 가짐을 보인다.

논문은 다음과 같이 구성된다. 2절에서는 saturation을 고려한 AQM 시스템의 모델을 제시한다. 3절은 anti-windup 알고리즘을 설명한다. 4절에서는 ns-2 시뮬레이터[12]를 사용한 실험을 통하여 anti-windup 방법을 적용한 비례 적분 제어기의 성능을 일반적인 비례 적분 제어기의 성능과 비교한다. 마지막으로 5절에서는 결론을 내린다.

### 2. Active queue management의 되먹임 제어

RED는 패킷을 drop하거나 ECN(Explicit Congestion Notification) mark를 함으로써 네트워크의 혼잡을 제어하는 AQM 알고리즘이다. TCP 소스는 자신의 패킷이 drop된 것을 발견하면 자신의 패킷 전송 속도를 조절함으로써 라우터의 큐 크기를 줄이게 된다. 이러한 과정은 그림 1에서 보여지는 것 처럼 폐루프 되먹임 제어 시스템을

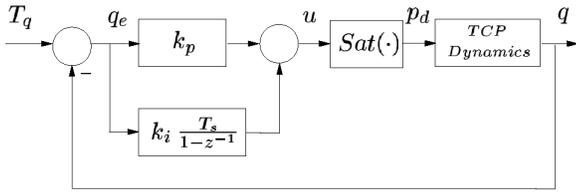


그림 2. 일반적인 비례 적분 제어기

이룬다. 이 시스템은 TCP 소스, 라우터 큐 그리고 혼잡 제어기로 구성된다. 비례 적분 제어기는 정상 상태 오차가 0 이 되게 하기 때문에 RED 보다 성능이 더 좋다. 그림 2는 일반적인 비례 적분 제어기를 보여 준다. 제어기의 입력은 큐 크기 오차  $q_e$ 이다. 이는 기준 큐 크기와 라우터 큐 크기의 차이이다.

$$q_e = e(n) - T_q \quad (1)$$

단,  $T_q$ 는 기준 큐 크기이며  $n$ 은 이산 시간 인덱스이다. Drop 확률은 제어기 출력을  $[0, p_{max}]$ 사이로 제한함으로써 얻어진다.

$$p_d(n) = sat(u(n)) \quad (2)$$

이며

$$sat(x) = \begin{cases} p_{max} & x > p_{max} \\ 0 & x < 0 \\ x & otherwise \end{cases} \quad (3)$$

단,  $p_{max}$ 는 최대 drop 확률이다. 이 drop 확률에 따라 패킷들이 drop 되거나 mark 된다. 이렇게 되면 TCP 소스들은 자신의 윈도우 크기를 줄여 전송 속도를 늦추게 되고 따라서 큐의 크기는 감소한다.

비례 적분 제어기는 큐의 크기가 기준 큐 크기 근처에 있을 때 성능이 우수하다. 그러나 실제 네트워크 트래픽은 네트워크의 burst한 특성에 의해 매우 빨리 변화한다. 때로는 패킷량이 기준 큐 크기를 유지하기에 너무 적을 수도 있다. 큐의 크기가 기준 큐 크기보다 상당시간 동안 작으면 적분 제어기에 누적된 오차는 상당히 큰 음수 값이 된다. 물론 이때에도 drop 확률은 0이다. 한번 이런 현상이 발생하면 트래픽이 증가하여도 적분 제어기의 출력이 음수에서 양수로 증가하는데 시간이 걸리므로 패킷을 drop하지 못하고 overflow를 일으키게 된다. 이런 경우 시스템의 성능은 저하된다.

### 3. Anti-windup 방법을 적용한 비례 적분 제어

비례 적분 제어는 큐 크기 오차에 대한 비례와 적분항을 사용하여 큐 크기를 제어한다.

$$u(n) = k_p q_e(n) + k_i T_s \sum_i q_e(i) \quad (4)$$

단,  $k_p$ 와  $k_i$ 는 각각 비례, 적분 이득이며  $T_s$ 는 제어기의 샘플링 간격이다. 위의 수식에서 첫 번째 항은 큐 크기 오차에 비례하며 두 번째 항은 오차의 적분항이다. 제어기의 구현은 다음과 같다.

$$u(n) = u(n-1) + k_p(q(n) - q(n-1)) + k_i T_s q_e(n) \quad (5)$$

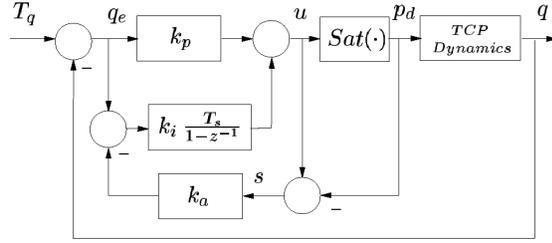


그림 3. 선형 되먹임 anti-windup 구조를 가진 비례 적분 제어

단,  $n=0,1,2,\dots$ 이며  $u(0) = k_p(q(0) - T_q)$ 이다. Drop 확률은 (3)의 saturator에 의해서 구해진다. 따라서 그 크기는  $0 \leq p_d \leq p_{max}$ 로 제한된다.

비례 적분기가 saturation된 동안 적분항의 크기가 지나치게 커지는 현상을 막기 위해 anti-windup 알고리즘을 적용한다. 선형 되먹임 anti-windup 알고리즘은 그림 3 에서와 같이  $p_d(n) - u(n)$ 을 상수 이득  $k_a$ 를 거쳐 적분항에 되먹임한다. 적분기의 출력  $u(n)$ 이 saturator에 의해 제한되면 제어기 출력과 drop 확률의 차를 적분항에 되먹임 함으로써 windup의 영향을 줄인다. 이 알고리즘은 다음과 같이 표현된다.

$$\begin{aligned} s(n) &= u(n-1) - sat(u(n-1)) \\ u(n) &= k_p q_e(n) + k_i T_s \sum_i (q_e(i) - k_a s(i)) \end{aligned} \quad (6)$$

단,  $s(n)$ 은 제어기 출력과 drop 확률의 차이이다. 이 제어기는  $k_a s(n)$ 을 제외하면 비례적분 제어기와 동일하다. 구현을 위해 제어기는 다음과 같이 표현될 수 있다.

$$\begin{aligned} s(n) &= u(n-1) - sat(u(n-1)) \\ u(n) &= u(n-1) + k_p(q(n) - q(n-1)) \\ &\quad + k_i T_s (q_e(n) - k_a s(n)) \end{aligned} \quad (7)$$

단,  $n=0,1,2,\dots$ 이며  $u(0) = k_p(q(0) - T_q)$ 이다.

선형 되먹임 anti-windup에서  $k_a$ 를 선정하는 것은 매우 중요하다.  $k_a$ 는 되먹임 이득으로서 anti-windup의 성능에 직접적인 영향을 미친다. 만약 이 이득이 작게 설정 되면 적은 양의  $s(n)$ 이 적분기로 되먹임 된다. 이득이 크면 windup의 과보상이 이뤄진다. 따라서 다음과 같이 conditioning technique를 사용한다 [11].

$$k_a = \frac{1}{k_p} \quad (8)$$

그러나 경우에 따라 conditioning technique를 사용하는 대신에  $k_a$ 를 적절히 tuning 함으로써 TCP 혼잡 제어의 성능을 개선할 수 있다.

### 4. 실험

Anti-windup의 성능을 평가하기 위해 두 개의 라우터와 여러개의 TCP 연결을 가진 간단한 병목 네트워크를 구성하여 ns-2 시뮬레이터를 사용하여 실험하였다. 라우터는 T3 (45Mbps) 링크로 연결되어 있다. 다른 링크는 모두 같은 대역폭을 가지며 병목 현상을 일으키지 않도록 구성하였다. 할당된 버퍼의 크기는 400 Kbyte로서 평균

표 1. 실험에 사용된 파라미터

Symbol	Parameter	Value
$k_p$	비례 이득	0.002
$k_i$	적분 이득	0.001
$k_a$	선형 되먹임 anti-windup 이득	1/0.002
$p_{max}$	최대 drop 확률	1.0
$T_s$	샘플 간격	0.5 ms
$T_q$	기준 큐 크기	400 packet

패킷 사이즈가 500 byte일 때 800 패킷에 해당한다. 각 링크의 전송 속도 지연은 10 ms이며 round-trip 시간 (RTT)는 60 ms이다. 실험에서 사용된 파라미터 값은 표 1에 나타나 있다. 비례 적분 제어기의 이득은 [8]의 (6)과 (7)에서 구하였다. 선형 되먹임 anti-windup 이득  $k_a$ 는 (8)에서 구하였다. 샘플링 간격은 출력 링크의 대역폭에 대하여 5개의 패킷이 전송되는 시간으로 정하였다.

4.1. 실험 1

실험 1은 빈 상태의 버퍼에 패킷이 쌓일 때의 성능을 평가한다. TCP의 burst한 특성으로 트래픽이 없다가 갑자기 증가할 수 있다. 이러한 현상을 구형파로 모델링하였다. 500개의 TCP 연결을 각각 40초와 20초 동안 ON/OFF한다.

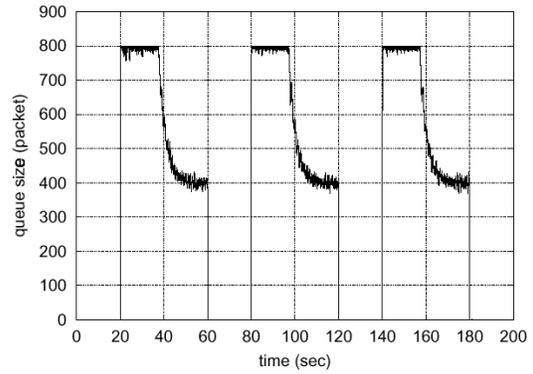
그림 4는 비례 적분 제어기의 결과이다. TCP 소스가 20초에 패킷 전송을 시작하였기 때문에 첫 20초 동안은 라우터에 들어오는 패킷이 없다. 이런 경우 기준 큐 크기는 400패킷이지만 실제 큐의 크기는 0이 된다. 이 경우 큰 오차가 발생되고 적분기의 출력 값이 음수가 되고 그 크기가 시간에 따라 계속 증가하게 된다. 그림 4(b)에서 제어기의 출력 값은 20초에서 -9정도이다. 물론 drop 확률은 0으로 제한된다. 20초에 TCP 소스가 패킷을 전송하지만 적분기의 출력값이 상당한 시간동안 음수로 머물러 있기 때문에 제어기는 큐 크기가 기준 큐 크기를 초과한 후에 도달한 패킷에 대하여서도 전혀 drop을 하지 못하였다. 결국 큐의 overflow가 되었다. 그림 5는 anti-windup 방법을 적용한 경우의 결과이다. 제어기의 출력이 saturation 되어 있는 동안 제어기의 출력값이 지속적으로 커지지 않는다. 이 상태에서 제어기의 출력값은 최대 -0.6 정도이다. Anti-windup 방법은 TCP 트래픽이 없거나 상당히 적은 상태가 끝났을 때 제어기가 선형 상태에서 동작하는 하도록 한다.

실험 1은 비례 적분 제어기가 AQM에 사용될 때 anti-windup 방법과 병행되지 않으면 심각한 성능의 저하를 가져 올 수 있음을 보여준다.

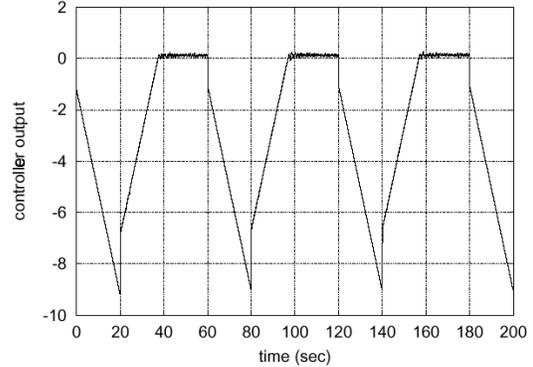
4.2. 실험 2

실험 2는 TCP 연결을 10초 간격으로 100개씩 증가시키면서 혼잡 제어의 성능을 살펴본다. TCP 연결 개수는 100초 동안 100에서 1000으로 증가된다.

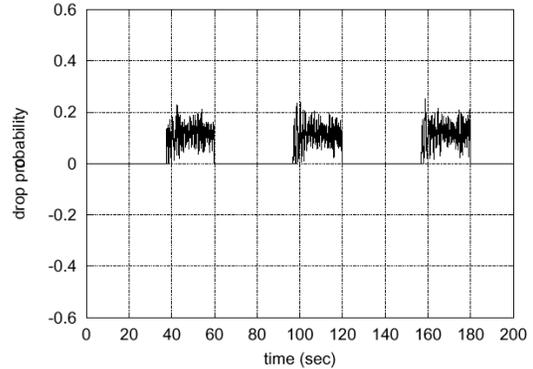
그림 6은 일반적인 비례 적분 제어기의 결과이다. 초기의 under-utilization 이후에 큐의 크기가 650정도에 이르는 overshoot이 발생하였다. 이는 큐의 크기가 0에서 기준 큐 크기에 이르는 동안 적분기에 누적된 오차 때문이다. 큐의 크기가 기준 큐 크기보다 커진 시점에서 적분



(a) Queue size



(b) Controller output  $u$



(c) Drop probability  $p_d$

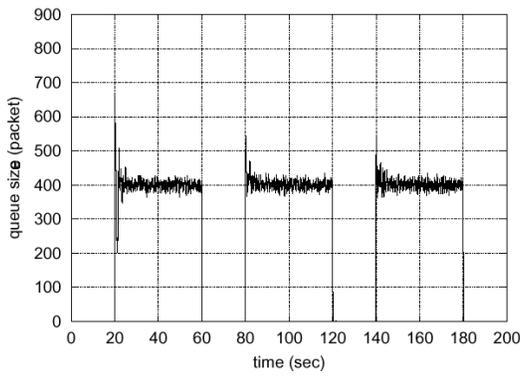
그림 4. 실험1; 비례 적분 제어

기에 누적된 오차에 의해 적절히 drop하지 못하여 발생하였다. 그림 7은 anti-windup을 적용한 비례 적분 제어기의 결과이다. 초기 under-utilization 이후의 overshoot이 줄어들었음을 알 수 있다.

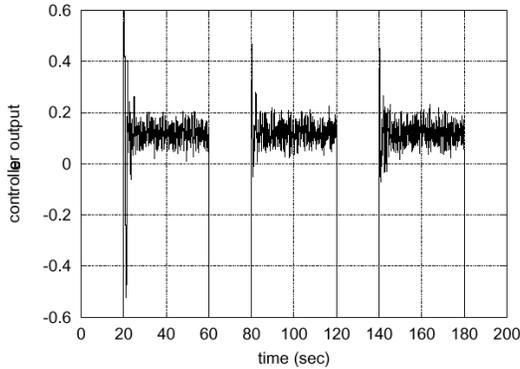
실험 2는 네트워크의 트래픽이 동적으로 변화하고 있을 때 비례 적분 제어기가 정상 상태에 이르는 동안 누적된 오차에 의해 비교적 큰 overshoot가 발생할 수 있음을 보여준다. 또한 anti-windup 방법을 적용함으로써 이러한 영향을 줄일 수 있음을 알 수 있다.

5. 결론

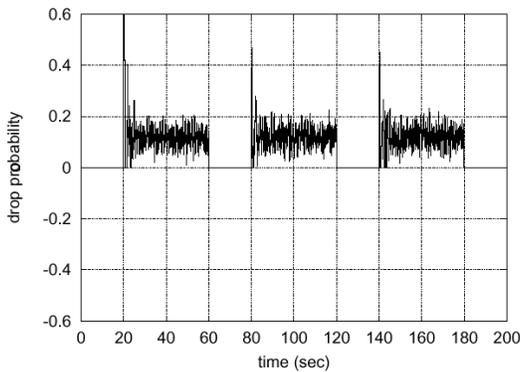
Active queue management에 비례 적분 제어기를 사용하면 RED 보다 성능 향상을 가져올 수 있다. 그러나 drop 확률이 0부터 1사이 제한되기 때문에 성능의 저하가 생긴다. 이러한 문제를 해결하기 위해 AQM을



(a) Queue size



(b) Controller output  $u$



(c) Drop probability  $p_d$

그림 5. 실험1; Anti-windup 방법을 적용한 비례 적분 제어

saturator를 가진 시스템으로 모델링하고 anti-windup 방법을 적용하였다. 실험을 통하여 anti-windup 방법을 적용하면 비례 적분 제어기의 성능이 개선됨을 보였다.

참고문헌

[1] S. Floyd and V. Jacobson, On traffic phase effects in packet switched gateways, *Internetworking: Research and Experience*, vol. 3, pp. 115-156, 1992.  
 [2] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Trans. on Networking*, vol. 1, pp. 397-413, 1993.  
 [3] M. May, J. Bolot, C. Diot, and B. Lyles, Reasons not to deploy RED, in *Proceedings of IWQoS*, 1999, pp. 260-262.  
 [4] W. Feng, D. Kandlur, D. Saha, K. Shin, "A self configuring RED gateway, in *Proceedings of IEEE/INFOCOM*, 1999, vol. 3, pp. 1320-1328.

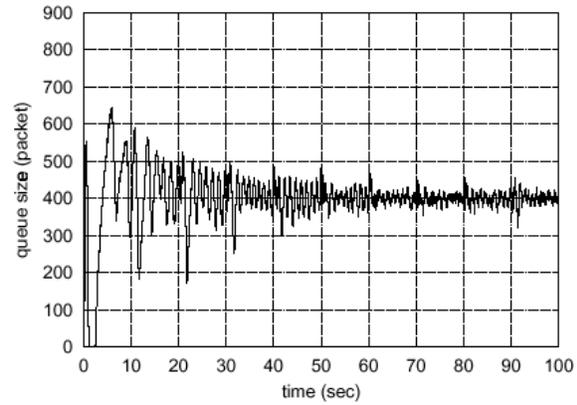


그림 6. 실험2; 비례 적분 제어

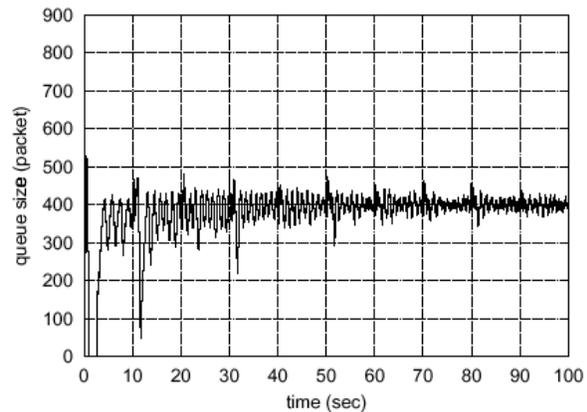


그림 7. 실험2; Anti-windup 방법을 적용한 비례 적분 제어

[5] T. J. Ott, T. V. Lakshman, and L. H. Wong, SRED: Stabilized RED, in *Proceedings of IEEE/INFOCOM*, 1999, vol. 3, pp. 1346-1355.  
 [6] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, REM: Active queue management, *IEEE Network*, vol. 15, pp. 48-53, 2001.  
 [7] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, A control theoretic analysis of RED, in *Proceedings of IEEE/INFOCOM*, 2001, vol. 3, pp. 1510-1519.  
 [8] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, On designing improved controllers for AQM routers supporting TCP flows, in *Proceedings of IEEE/INFOCOM*, 2001, vol. 3, pp. 1726-1734.  
 [9] J. Aweya, M. Ouellette, and D. Y. Montuno, A control theoretic approach to active queue management, *Computer Networks*, vol. 36, pp. 203-235, 2001.  
 [10] J. Aweya, M. Ouellette, and D. Y. Montuno, An optimization-oriented view of random early detection, *Computer Communications*, vol. 24, pp. 1170-1187, 2001.  
 [11] Y. Peng, D. Vrancic, and R. Hanus, Anti-windup, bumpless and conditioned transfer techniques for PID controllers, *IEEE Control Systems*, vol. 16, pp. 48-57, 1996.  
 [12] ns-2 network simulator, Obtain via <http://www.isi.edu/nsnam/ns>.