

큐 예측을 통한 인터넷 혼잡 제어*

권성기^U 장봉석
목포대학교 멀티미디어공학과
jang@mokpo.ac.kr

Internet Congestion Control Using Queue Prediction

Sung-Gi Kwon^U Bongseog Jang
Dept. of Multimedia Engineering, Mokpo Nat'l University

요 약

본 논문에서는 인터넷 혼잡제어를 위한 새로운 방법을 제안한다. 라우터 큐에 예측제어함수를 적용하여 미래의 혼잡상황을 예측하고 소스에게 미리 피드백을 수행하여 혼잡제어를 한다. 예측제어함수는 실제 큐와 예측된 큐의 오차를 계산하여 주기적으로 예측함수를 갱신하는 NLMS 방식의 예측제어함수를 적용한다. 피드백 정보의 전송지연으로 인한 혼잡상황 악화가 발생하기 전에 혼잡상황에 대응할 수 있으므로 라우터 버퍼 사용효율을 최적함을 유지할 수 있으며 버퍼 오버플로우로 발생하는 패킷의 손실을 최소화 할 수 있다. 혼잡상황을 야기하도록 과도한 트래픽을 생성하여 라우터에서 예측함수를 적용하는 경우와 단지 혼잡알림제어를 수행하는 경우를 비교하여 시뮬레이션을 수행하였다. 예측함수를 적용하는 경우는 시스템 성능효율을 증가시키며 라우터 버퍼 크기를 최적으로 사용할 뿐만 아니라 오버플로우가 발생하지 않았으나 예측함수를 적용하지 않고 혼잡알림제어를 수행하는 경우는 과도한 큐 크기와 오버플로우가 발생하였음을 시뮬레이션을 통해서 보인다.

1. 서 론

인터넷 혼잡제어는 TCP에 의해서 수행된다. TCP의 혼잡제어 방식은 윈도우 기반의 제어 방식이며 비 실시간 데이터 트래픽을 중단시스템간에 효과적으로 제어할 수 있었다. 그러나 실시간 스트리밍 데이터의 증가와 고속 데이터 처리의 요구와 더불어 중단 시스템간의 혼잡제어는 인터넷 붕괴(collapse)에 해당하는 상황을 초래할 수 있다. 그러므로 TCP 우호적인(friendly) 제어방식들이 제안되고 있다[1]. 그러나 궁극적으로는 라우터에서 인터넷 혼잡상황을 제어하여야 보다 효과적이다. MPLS (Multiple Protocol Label Switching), WFQ(Weighted Fair Queueing) 및 CBQ(Class Based Queueing) 등이 제안되고 있으며 이러한 방식들은 라우터에서 고속의 패킷 처리 및 큐 관리 기능을 수행하여 인터넷 트래픽 관리를 최적화하는 기술들이다[2].

ATM(Asynchronous Transmission Mode) 망은 보다 트래픽 제어 기술들이 잘 정의되어 있다. 그 중에서 ABR (Available Bit Rate) 서비스는 ATM 스위치에서 오는 피드백 정보를 이용하여 소스가 전송률을 제어하며 CN (Congestion Notification)과 ER (Explicit Rate) 방식으로 나뉜다. CN 방식은 혼잡상황을 단지 소스에게 알려주기만 하고 ER 방식은 스위치가 송신자에게 명백하게 전송률을 ATM 스위치의 버퍼링 정도에 따라서 정해준다[4].

기존의 혼잡제어 기술들은 수신 중단 시스템이나 스위치에서의 피드백 정보를 이용한다. 그러나 혼잡상황이 발생함으로써 혼잡상황 피드백을 시작하는 close loop 형태의 제어를 한다. 본 논문에서는 혼잡상황이 발생하

기 전에 미리 혼잡상황을 예측하고 그 정보를 트래픽 소스에게 피드백 해주는 방법을 제안한다. 그러므로 피드백정보의 전송지연으로 발생하는 혼잡상황 악화를 방지할 수 있다. 라우터에서 주기적으로 큐 크기를 측정하고 측정된 큐 길이를 모형화 한다. 그 모형을 이용하여 다음 주기의 큐 크기를 예측을 시도한다. 또한 이 모형은 주기적으로 실제 큐와 예측된 큐의 오차에 의해 갱신된다. 예측된 큐 크기가 라우터 버퍼에 설정된 임계값을 초과하면 트래픽 소스에게 ER 형태의 전송률 정보를 피드백 한다. 그러므로, 라우터 큐 길이의 최적화를 유지할 수 있고 라우터에 설정된 버퍼 임계값을 초과하지 않으므로 버퍼 오버플로우를 최소화 할 수 있다.

본 논문에서는 제안된 방식이 보다 효과적으로 라우터 큐 관리를 할 수 있음을 시뮬레이션을 통하여 보인다. 라우터에 예측방법을 적용하지 않는 CN 방식과 예측방법을 적용한 ER 방식을 비교하였다. ER 방식을 적용한 경우가 보다 최적한 큐 관리 효율을 보이고 버퍼 오버플로우가 발생하지 않는 반면에 CN 방식은 요구되는 큐 크기가 버퍼 오버플로우를 야기 시킴을 알 수 있다. 추후에는 RED(Random Early Detection)[3]와 CN 방식을 조합한 경우와 RED 와 예측방법을 적용한 CN 방식을 비교할 예정이다.

본 논문의 구성은 다음과 같다. 다음 장에서는 예측제어함수를 소개한다. 3장에서는 시뮬레이션 결과를 소개하고, 4장에서는 결론과 향후연구에 대해서 언급한다.

2. 혼잡상황 예측제어 모델

다음 그림은 송수신 시스템과 라우터로 구성된 인터넷의 구조이며, 라우터가 피드백 제어정보 전송을 하는 과정을 보여주고 있다. 그림에서 R 은 라우터를 의미한다. 본 논문의 피드백 제어정보전송은 인터넷 에지(edge) 라

* 본 연구는 한국과학재단 지역우수과학자 연구 (과제번호 R02-2001-00952)지원으로 수행되었음.

우터와 송신시스템간의 범위만 가정한다. 즉, edge 라우터는 큐 변화량에 예측제어함수를 적용하고 예측된 결과를 피드백 패킷으로 소스에게 전송하는 구조이다. ICMP (Internet control message protocol) 의 source quench 메시지를 이용하여 피드백정보를 전송한다고 가정한다. Source quench message는 BSD system에서 옵션으로 실행할 수 있도록 구현되어 있다.

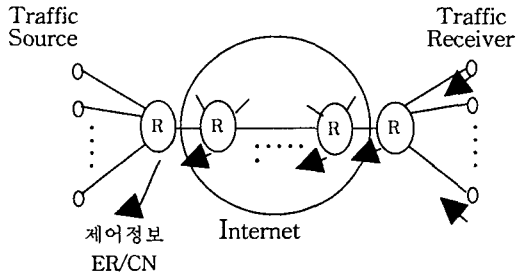


그림1. 라우터 피드백제어

예측제어함수는 라우터 큐의 시간에 따르는 변화량에 반응하여 미래의 큐 길이 값을 계산한다. 만일 미래의 예측된 큐의 크기가 버퍼의 상한 임계치 T_H 를 초과할 경우, 라우터는 폭주 상황이 발생할 것을 고려하여 소스의 전송률을 감소시키기 위해 ICMP 패킷에 소스의 ER 값을 계산하여 보낸다. 그러나, 제어메시지의 발생을 최소화 하기 위하여 버퍼크기가 상한 임계치 T_H 보다 작은 경우로 예측되면 ICMP 패킷을 피드백 전송하지 않는다. 이 경우에는 소스는 지속적으로 소스 전송률을 증가시키게 되거나 이미 피드백 된 ER 값 만큼 지속적으로 보낼 수 있다고 가정한다. 그리고 다시 라우터에서 예측된 큐가 상한 임계치를 초과하면 ICMP 패킷에 새로이 계산 된 ER을 실어서 소스에게 보낸다.

NLMS(normalized least mean square) [5] 방법은 라우터 예측제어함수를 추정하는데 적용한다. 현재 큐 길이와 시간 n 에서의 가중인자(기울기)를 가진 선형함수를 이용하여 다음 k 단계 이후의 버퍼 큐 길이를 예측한다. $Q(n)$ 이 시간 n 에서 큐 길이라 하면, k 단계의 예측제어기는 다음과 같이 주어진 식에서 k 단계의 큐 길이를 $Q(n)$ 로부터 추정한다.

$$Q(n+k) = a^k Q(n) \dots \dots \dots (1)$$

여기서 $a(n)$ 은 n 시간에 추정된 가중인자 (기울기)이고, k 는 예측구간이다. 이 때 시간 n 에서의 예측오차는

$$e(n) = Q(n) - \hat{Q}(n) \dots \dots \dots (2)$$

이고, $\hat{Q}(n)$ 은 예측된 큐길이가 된다. 여기서

$$\hat{Q}(n) = a(n-1)Q(n-1) \dots \dots \dots (3)$$

이다. 위의 선형 예측함수는 각 시간단계에서 가중인자를 수정하기 위해서 발생하는 에러 값(식 (2))을 사용한다. 가중인자 $a(n)$ 은 소스가 추가 또는 제거되거나 소스의 전송률이 변화되는 것에도 영향을 받을 수 있다. 가중인자를 추정하는데 정규화된 최소제곱평균오차 (NLMS) 알고리즘을 사용한다. 초기 값 $a(0) = 0$ 로 주어

지면 가중인자 값은 다음과 같이 수정된다.

$$a(n) = a(n-1) + \frac{\mu e(n) Q(n-1)}{|Q(n-1)|^2} \dots \dots \dots (4)$$

여기서 μ 는 상수이다. $Q(n)$ 이 정정일 경우, $a(n)$ 은 최적해로서 평균제곱에 수렴하고, 정규화된 최소제곱평균오차 NLMS는 인자 μ 에 별로 영향을 받지 않으며 각 시간 단계에서 가중인자 $a(n)$ 은 실제 버퍼의 큐 길이 $Q(n)$ 과 추정된 버퍼의 큐 길이 $\hat{Q}(n)$ 의 차이인 잔차 $e(n)$ 에 의해 버퍼 크기가 증가 또는 감소되었는가에 대한 함수 변화 방향을 지시한다.

그림 2는 본 연구에서 제시하는 예측계획을 나타낸다. 즉, $a > 1$ 이라면, 예측된 큐 길이는 $Q(n+k) = a^k Q(n)$ 식에 의해 증가되며 T_H 에 도달한 시간 k 단계의 예측 값은 시간 n 에서 명시된 $Q(n)$ 과 $a(n)$ 을 사용하여 미리 예측할 수 있다. 그러므로 제시된 선형 예측함수에 의하여 시간 n 에서 미래 큐 길이를 추정함으로써 미래의 폭주 상황 여부를 예측할 수 있다.

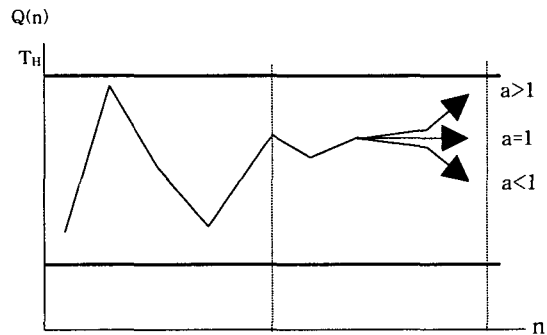


그림2. 예측함수의 큐 예측

라우터는 현재 큐 길이의 모니터링을 주기적으로 하며 설정된 상위와 하위 임계치를 참조하며 예측 제어함수를 이용하여 미래 큐 길이의 예측 값을 계산한다. ER 방식은 예측된 큐 값이 T_H 보다 크면 혼잡상황을 예측하고 계산된 ER 값이 피드백 된다. 반면에 CN 방식은 혼잡상황이 발생했을 때 소스에게 알려주기만을 수행하고 예측 방식을 적용하지 않는다.

3. 시뮬레이션

예측된 큐 값을 이용한 예측제어알고리즘은 다음 3 가지 경우에 대하여 ER 값을 계산한다. 하나는 혼잡상황, 두 번째는 normal한 상황, 마지막은 underload 상황이다. ER 계산 방법은 ATM 망에서 적용한 방식을 응용하였다 [4].

다음 그림은 본 시뮬레이션에서 적용하는 시뮬레이션 변수들을 나타내는 그림이다. 각 소스의 패킷간 간격은 geometric 확률분포를 이용하였다. 소스는 최대 10개로 구성되었으며 각 소스의 geometric 확률분포함수의 평균 값은 서로 다른 값을 사용하였다. 그러므로 ER 값 및 CN 정보들 라우터가 소스들에게 전달하더라도 패킷 간격이 랜덤하므로 소스에서 라우터에 도착하는 시뮬레이션 시간은 소스마다 차이를 갖게 된다. 각 소스에서 라

우터까지의 전송 지연시간 (왕복지연시간은 아님)은 $d_i = \{2, 4, 6, 9, 11, 13, 16, 20, 21, 23\}$ 시뮬레이션 타임을 설정하였다. 라우터의 예측 구간은 10 ($\hat{Q}(n+10) = a^{10}Q(n)$) 시뮬레이션 타임으로 설정하였다. 라우터의 상위 큐 임계값(T_H)은 5000, 그리고 하위 큐 임계값(T_L)은 1000으로 설정하였다. 라우터의 출력률은 200으로 설정하였다. ER 경우는 소스 전송률은 라우터가 결정하는데 반면에 CN 방식은 소스에게 최대 출력률 100까지 혼잡상황이 발생하지 않으면 지속적으로 증가하도록 한다.

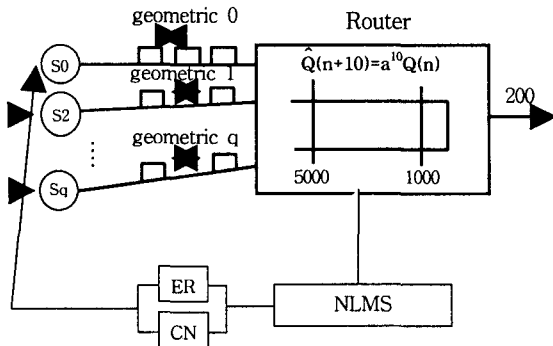


그림3. 라우터와 소스간의 시뮬레이션 Parameters

다음 그림들은 시뮬레이션 결과로서 ER 과 CN 방식의 라우터에서의 큐 길이의 변화정도를 보여 준다. ER 방식 시뮬레이션 결과는 예측을 통해서 피드백 제어정보의 지연에 대한 상쇄함을 볼 수 있다. 즉 오버플로우가 발생하지 않았고 라우터는 최대 시스템 효율을 지속적으로 유지함을 보여준다. 라우터의 큐 길이가 상위 한계치에 근접해서 모든 시뮬레이션 시간동안 유지되기 때문이다.

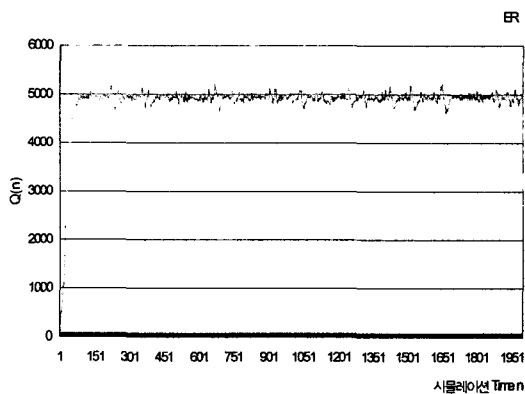


그림4. ER 과 예측제어방식 적용 시 라우터 큐 변화량

반면에 CN 방식은 예측제어알고리즘을 사용하지 않고 혼잡상황이 발생하면 ($Q(n) > 5000$) 혼잡상황을 소스에게 단지 하나의 비트를 사용하여 알려주는 역할만을 한다고 가정하였다. 이때 소스는 소스 자체적으로 새로이 전송률을 계산하여 보내게 되는데 혼잡상황이 알려지면 전송률이 거의 제로가 된다. CN 방식 결과로서, 큐의

변화가 심하게 발생할 뿐만 아니라 fluctuation 도 심하게 발생한다. 피드백되는 제어정보의 지연으로 인하여 이와 같은 변화가 심한 큐가 형성되고 지터도 더 심하게 될 뿐만 아니라 버퍼오버플로우가 심각하게 발생함을 알 수 있다.

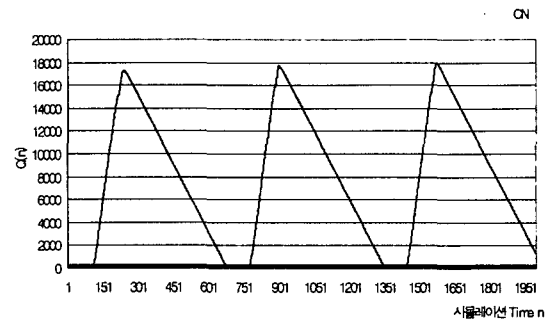


그림5. CN 적용시 라우터의 큐 변화량

4. 결론 및 향후연구

ER 방식과 라우터 큐 예측제어함수를 적용하여 라우터에서 혼잡상황 한계값 이내에서 최대의 전송효율을 보임을 시뮬레이션을 통해서 알 수 있다. CN 방식을 사용하는 경우는 반면에 제어정보전송의 지연으로 인한 급격히 변화하는 큐를 생성하고 혼잡상황이 심각하게 나타남을 볼 수 있다.

향후 지속적으로 제안된 알고리즘을 보완할 계획이며, CN과 RED 방식에 예측함수를 적용한 결과에 대한 비교 분석을 수행할 것이다. 그리고 MPLS, WFQ, CBQ 등 기존의 라우터 제어 및 관리 기술에 예측제어방식을 접목하는 결과에 대한 비교연구를 진행할 것이며 궁극적으로 그러한 기술들과 접목에 대한 연구를 진행할 것이다.

[참고문헌]

- [1] Floyd, S., and Fall, K., "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions on Networking, Aug. 1999.
- [2] Floyd, S., and Jacobson, V., "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, Vol. 3 No. 4, Aug. 1995.
- [3] Mahajan, R., Floyd, S. and Wetherall D., "Controlling High-Bandwidth Flows at the Congested Router", 9th International Conference on Network Protocols (ICNP), Nov. 2001.
- [4] Jang, B., B. G. Kim and G. Pecelli "A Prediction Algorithm for Feedback Control Models with Long Delays", IEEE Broadband Switching Systems (BSS), Dec. 1997.
- [5] Haykin, S., Adaptive Filter Theory, 2nd ed., Prentice Hall, 1991.