

자기회귀 웨이블릿 신경 회로망을 이용한 TCP 네트워크 혼잡제어

Congestion Control of TCP Network Using a Self-Recurrent Wavelet Neural Network

김재만*, 박진배**, 최윤호***
Jae Man Kim*, Jin Bae Park**, Yoon Ho Choi***

Abstract - In this paper, we propose the design of active queue management (AQM) control system using the self-recurrent wavelet neural network (SRWNN). By regulating the queue length close to reference value, AQM can control the congestions in TCP network. The SRWNN is designed to perform as a feedback controller for TCP dynamics. The parameters of network are tunes to minimize the difference between the queue length of TCP dynamic model and the output of SRWNN using gradient-descent method. We evaluate the performances of the proposed AQM approach through computer simulations.

Key Words : Self-Recurrent Wavelet Neural Network, AQM, Congestion Control, Direct Adaptive Controller

1. 서 론

네트워크를 통해 이루어지는 인터넷 영역들이 증가하면서 네트워크의 혼잡 제어에 대한 연구가 활발히 수행되어지고 있다. 네트워크에서의 혼잡성은 망에서 사용될 수 있는 패킷의 전송량이나 버퍼의 크기에 비해 더 많은 양을 사용자들이 동시에 요구할 경우에 발생하게 된다. 그로 인해 때로는 사용자가 원하는 정보가 중간에 손실될 경우도 발생하며 또한 원하는 시간에 필요한 정보를 얻을 수 없게 되기도 한다. 이러한 혼잡성의 제어는 TCP (Transmission Control Protocol) 상에서 두가지 측면에서 살펴볼 수 있다. 하나는 송신단측에서 윈도우 사이즈 조절을 통해 혼잡성을 피할 수 있으며 다른 하나는 라우터에서 큐의 길이를 조절하여 혼잡성을 제어할 수 있다. 송신단측에서 이루어지는 윈도우 크기의 조절은 AIMD (Additive Increase Multiplicative Decrease) 방식을 이용하여 송신단에서 ACK (acknowledgement)를 받게 되면 윈도우 사이즈를 증가시키고 만약에 ACK를 못 받게 되면 윈도우 사이즈를 감소시킨다.

AQM은 송신단측이 아닌 라우터에서 혼잡을 제어하는데 사용된다. AQM은 라우터에 들어오는 패킷들이 점점 증가함에 따라 버퍼가 차는 것을 방지하기 위해 나중에 들어오는 패킷들을 미리 폐기시키는 역할을 한다. 그리고 예상되는 혼잡성을 송신단에 미리 알림으로써 패킷들의 전송 속도를 조절하여 다른 패킷들의 손실과 지연을 예방할 수 있다. 이전

까지 능동 큐 관리를 위한 방법으로 네트워크 관점에서는 RED(Random Early Detection), REM(Random Exponential Marking), EWMA(exponentially weighted moving average) 등이 제안되었으나, 이러한 방식들은 확률적으로 패킷들을 임의로 폐기시킴으로써 급격히 변화하는 네트워크 환경 속에서 초기의 혼잡성을 발견하고 제어하는데 어려움이 있다 [1]-[3]. 또한 오직 현재나 과거의 큐의 상태만을 통해 혼잡성을 발견하고 제어하므로 현재나 과거의 혼잡성의 제어에는 민감하지만 예상되는 혼잡성에 대해서는 예측하기가 어렵다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 직접 적응 제어기에 기반을 둔 자기회귀 웨이블릿 신경 회로망(SRWNN) 제어가 설계 방법을 제안한다. 능동 큐 관리 제어 시스템으로부터 나온 결과인 큐의 길이가 직접 웨이블릿 신경 회로망의 입력으로 들어가 시스템의 제어 입력인 패킷의 폐기 확률을 생성하게 되어 신경 회로망의 파라미터 값들은 경사 하강법을 통해 학습된다.

2. 본 론

2.1 능동 큐 관리 제어 시스템

여러 개의 송신단에서 일정한 바이트로 묶인 패킷들을 수신단으로 전송하고자 할 경우, 패킷들은 라우터를 통과하게 된다. 송신단에서 링크를 통해 라우터에 도착한 패킷들은 순서대로 라우터내의 버퍼를 통과하여 수신단쪽으로 가게 되는데 계속 들어오는 패킷들에 비해 라우터의 버퍼는 용량이 제한되어 있으므로 모든 패킷들을 받아들일 수 없게 된다. 따라서 버퍼가 모두 차기 이전에 들어오는 패킷들 중 일부는 폐기시켜야 하는 문제가 발생한다. 능동 큐 관리는 이와 같이 TCP상에서 라우터에 들어오는 패킷들로 인해 버퍼가 차는 것을 방지하기 위하여 라우터로 들어오는 패킷들을 미리 폐기시킴으로써 다른 패킷들의 손실과 지연을

저자 소개

- * 延世大學校 電氣電子學科 碩士課程
- ** 延世大學校 電氣電子學科 教授·工博
- *** 京畿學科校 電子工學部 教授·工博

미리 방지한다. 기본적인 송-수신단간의 연결은 그림 1과 같다. TCP의 동적 모델은 확률적인 미분 방정식 형태로 나타내었다 [4].

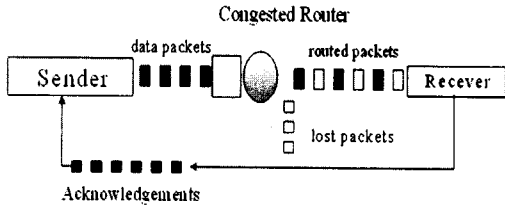


그림 1. 송-수신단간 연결 구조.

모델은 네트워크에서 보이는 중요한 변수들의 평균값과 관련되며 다음과 같이 비선형 방정식으로 나타내어진다.

$$W(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} \quad (1)$$

$$q(t) = \frac{W(t)}{R(t)} M(t) - C \quad (2)$$

여기서,

- W : TCP 윈도우의 평균 크기 (패킷)
- q : 큐의 평균 길이 (패킷)
- R : 송수신단간 왕복 시간 = $\frac{a(t)}{C} + T_p$ (초)
- C : 링크 용량 (패킷/초)
- T_p : 지연 시간 (초)
- N : TCP 세션의 수
- p : 패킷 폐기 확률

식 (1)에 의하면 TCP 윈도우의 크기는 송수신단 간 왕복 시간에 반비례하여 증가하며, 패킷 폐기 확률에 따라 $W/2$ 만큼 감소하게 된다. 큐의 길이는 식 (2)에서 볼 수 있듯이 패킷 수신율과 링크 용량의 차이에 따라 변하는 것을 알 수 있다. 큐의 길이와 윈도우 크기는 모두 0보다 커야 하고 크기 제한이 있다. 즉 $q \in [0, \bar{q}]$, $W \in [0, \bar{W}]$ 을 만족시켜야 하며, \bar{q} 와 \bar{W} 는 각각 버퍼 용량과 최대 윈도우 크기를 나타낸다. 또한 패킷 폐기 확률은 $p \in [0, 1]$ 이다 [5,6].

2.2 자기회귀 웨이블릿 신경회로망

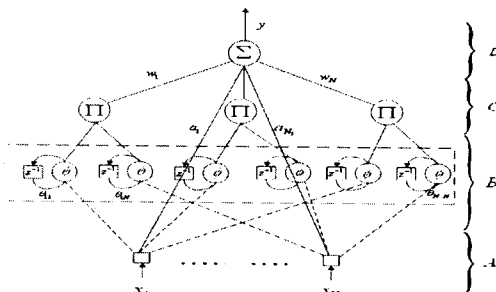


그림 2. SRWNN 구조.

SRWNN의 구조는 그림 2와 같이 N_i 개의 입력과 하나의 출력 그리고 N_h 개의 모체 웨이블릿 노드를 가진다.

A층의 N_i 개의 입력에 대한 B층의 웨이블릿 ϕ_{jk} 는 식 (3)과 같이 정의된다.

$$\phi_{jk} = \phi\left(\frac{u_{jk} - m_{jk}}{d_{jk}}\right), \text{ with } z_{jk} = \frac{u_{jk} - m_{jk}}{d_{jk}} \quad (3)$$

여기서 m_{jk} 와 d_{jk} 는 각각 웨이블릿의 변형 인자와 확장 인자이며 jk 는 j 번째 웨이블릿의 k 번째 입력을 나타낸다. B층의 각 웨이블릿에 대한 입력은 식 (4)같이 정의된다.

$$u_{jk}(n) = x_k(n) + \phi_{jk}(n-1) \cdot \theta_{jk} \quad (4)$$

여기서 θ_{jk} 는 자기 회귀 루프의 가중치를 나타내며, $\theta_{jk}(n-1)$ 은 신경망의 과거 정보를 저장하는 부분으로 WNN과 SRWNN 사이의 차이를 보여준다. C층은 모체 웨이블릿의 곱으로써 식 (5)와 같이 나타낼 수 있다.

$$\begin{aligned} \phi_j(X) &= \prod_{k=1}^{N_i} \phi(z_{jk}) \\ &= \prod_{k=1}^{N_i} \left[-\left(\frac{u_{jk} - m_{jk}}{d_{jk}}\right) \exp\left(-\frac{1}{2} \left(\frac{u_{jk} - m_{jk}}{d_{jk}}\right)^2\right) \right] \end{aligned} \quad (5)$$

D층은 출력층으로 최종 출력은 식 (6)과 같이 나타낼 수 있다.

$$y(n) = \sum_{j=1}^{N_c} w_j \phi_j(X) + \sum_{k=1}^{N_i} a_k x_k \quad (6)$$

여기서 w_j 는 C층과 D층 사이의 연결 가중치이고 a_j 는 A층과 D층 즉, 입력층과 출력층 사이의 가중치이다. SRWNN에서 학습을 통하여 갱신되는 파라미터들의 집합 W 는 다음과 같다.

$$W = [a_k, m_{jk}, d_{jk}, \theta_{jk}, w_j] \quad (7)$$

본 논문에서는 자기회귀 웨이블릿 신경 회로망에 기반을 둔 직접 적응 제어기를 사용한다. 제어기의 전체 구조는 그림 3과 같다.

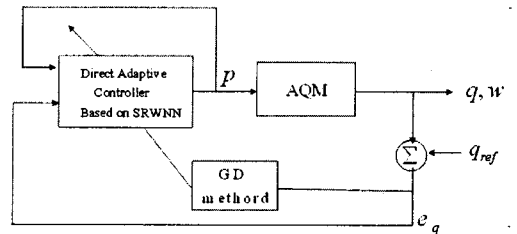


그림 3. 직접 적응제어 구조.

학습 방법으로 식 (8)과 같은 비용함수를 최소화하는 과정을 통해 파라미터들을 갱신한다.

$$J(n) = \frac{1}{2} [y_c(n) - y(n)]^2 = \frac{1}{2} e^2(n) \quad (8)$$

여기서 $y(n)$ 은 능동 큐 관리 제어 시스템을 통해 나온 n 번째 출력이며, $y_c(n)$ 은 기준 출력값이다. 최소화 과정은 반복적인 경사하강법 (gradient-descent method)을 사용하며, 식 (9)와 같은 방법으로 파라미터들을 갱신한다.

$$\begin{aligned} W(n+1) &= W(n) + \Delta W(n) \\ &= W(n) + \eta \left(-\frac{\partial J(n)}{\partial W(n)} \right) \end{aligned} \quad (9)$$

여기서 η 는 SRWNN 파라미터의 학습률을 의미하고, $\frac{\partial K(n)}{\partial W(n)}$ 는 $W(n)$ 에 대한 비용함수의 편미분으로 식(10)과 같이 계산한다.

$$\frac{\partial K(n)}{\partial W(n)} = \epsilon(n) \frac{\partial \epsilon(n)}{\partial W(n)} = -\epsilon(n) \frac{\partial y(n)}{\partial W(n)} \quad (10)$$

식 (9)와 (10)에 의해 식 (7)의 각 파라미터들은 다음과 같이 갱신된다.

$$\frac{\partial x(n)}{\partial a_k(n)} = x_k \quad (11)$$

$$\frac{\partial y(n)}{\partial m_{jk}(n)} = -\frac{w_j}{d_{jk}} \frac{\partial \Phi_j(X)}{\partial z_{jk}} \quad (12)$$

$$\frac{\partial y(n)}{\partial d_{jk}(n)} = \frac{w_j}{d_{jk}} z_{jk} \frac{\partial \Phi_j(X)}{\partial z_{jk}} \quad (13)$$

$$\frac{\partial y(n)}{\partial \theta_{jk}(n)} = \frac{w_j}{d_{jk}} \phi_{jk}(n-1) \frac{\partial \Phi_j(X)}{\partial z_{jk}} \quad (14)$$

$$\frac{\partial y(n)}{\partial w_j(n)} = \Phi_j(X) \quad (15)$$

여기서 $\frac{\partial \Phi_j(X)}{\partial z_{jk}}$ 와 $\phi(z_{jk})$ 는 다음과 같다.

$$\frac{\partial \Phi_j(X)}{\partial z_{jk}} = \phi(z_{j1})\phi(z_{j2}) \cdots \phi(z_{jk})\phi(z_{jN}) \quad (16)$$

$$\phi(z_{jk}) = \frac{\partial \phi_j}{\partial z_{jk}} = (z_{jk}^2 - 1) \exp\left(-\frac{1}{2} z_{jk}^2\right) \quad (17)$$

3. 시뮬레이션

본 논문에서는 자기회귀 웨이블릿 신경 회로망에 의한 능동 큐 관리 제어 시스템의 성능을 확인하기 위해 TCP 세션의 수에 따른 모의 실험을 하였다. 실험의 성능을 확인하기 위해 평균 제곱 오차(MSE)와 평균 큐의 길이 그리고 평균 패킷 폐기 확률 측면에서 살펴보았다. TCP 모델의 파라미터 값들은 다음과 같다 [7].

표 1. TCP 모델 파라미터 값.

샘플링 주파수	160 Hz
링크 용량	1875(패킷/초)
지연 시간	0.005(초)

TCP 세션수의 변화에 대한 큐의 길이와 패킷 폐기확률 실험은 우선 학습률은 0.0001로 정했으며 q_{ref} 는 200으로 잡았다.

표 2. TCP 세션수에 따른 성능 비교.

	N=30	N=60	N=120
MSE	0.4497	0.3108	0.3567
큐의 길이	192.5847	202.3779	209.1436
패킷 폐기 확률	0.0359	0.1222	0.4370

4. 결 론

본 논문에서는 자기회귀 웨이블릿 신경회로망을 기반으로 한 직접적용 제어기를 이용하여 능동 큐 관리 시스템의 혼잡성 제어를 제안하였다. 자기회귀 웨이블릿 신경망은 모체 웨이블릿이 자기 회귀를 하므로 웨이블릿 신경망보다 더 적은 수의 은닉층을 가지고도 동일한 성능을 갖는 직접적용 제어기의 역할을 하였다. 능동 큐 관리 제어는 이전까지는 RED, REM 등 확률적인 방법을 통해

수정되었지만 컴퓨터 모의 실험을 통해 자기 회귀 웨이블릿 신경 회로망과 같은 제어 기법도 라우터에서 큐의 길이를 일정하게 유지시키며 패킷 폐기 확률을 낮추는데 충분히 이용될 수 있음을 확인 할 수 있었다.

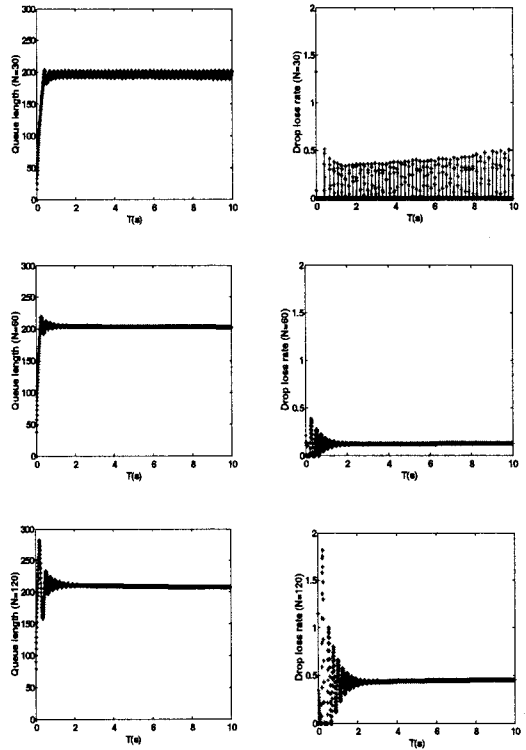


그림 4. 세션 수에 따른 큐의 길이와 패킷 폐기 확률.

참 고 문 헌

- [1] S. Ryu, C. Rump, C. Qiao, "A Predictive and Robust Active Queue Management for Internet Congestion Control," Proceedings of ISCC, vol. 2, pp. 991-998, 2003.
- [2] F. Yanfei, R. Fengyuan, L. Chuang, "Design a PID Controller for Active Queue Management," Proceedings of ISCC, vol. 2, pp. 985-990, 2003.
- [3] S. Ryu, C. Cho, "PI-PD-controller for Robust and Adaptive Queue Management for Supporting TCP Congestion Control," Proceeding of ANSS, pp. 132-139, 2004.
- [4] V. Misra, W. Gong, Don Towsly, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," Proceedings of ACM SIGCOMM, vol. 30, pp. 151-160, 2000.
- [5] C.V. Hollot, V. Misra, "Analysis and Design of Controller for AQM Routers Supporting TCP Flows," IEEE Trans. on Automatic Control, vol. 47, no. 6, pp.945-959, 2002.
- [6] C.V. Hollot, V. Misra, D. Towsley and W. Gong, "A Control Theoretic Analysis of RED," Proc of IEEE INFOCOM, vol. 3, pp. 1510-1519, 2001.
- [7] R. Fengyuan, R. Yong, and S. Xiuming, "Design of a Fuzzy Controller for Active Queue Management", Computer Communications, pp. 874-883, 2002.