

웨이블릿 신경 회로망에 기반한 능동 큐 관리 제어 시스템 설계

*김재만, *박진배, **최윤희
 *연세대학교 전기전자공학과 **경기대학교 전자공학부

Design of Active Queue Management Control System Based on Wavelet Neural Network

*Jae Man Kim, *Jin Bae Park, **Yoon Ho Choi
 *Dept. Electrical & Electronic Eng. Yonsei Univ. **School of Electronic Eng. Kyonggi Univ.

Abstract - 본 논문에서는 웨이블릿 신경 회로망에 기반을 둔 능동 큐 관리(Active Queue Management) 제어 시스템을 설계하는 것을 제안한다. 제안한 제어 시스템에서 웨이블릿 신경 회로망은 능동 큐 관리를 위한 제어기로 사용한다. TCP 동적 모델의 실제 출력, 큐의 길이와 웨이블릿 신경 회로망을 이용한 출력의 오차가 최소화되도록 웨이블릿 신경 회로망의 파라미터 값들을 변화시키며 각각의 파라미터 값들은 경사 하강법을 통해 학습시킨다. 마지막으로 제안한 방법은 모의실험을 통해 패킷 손실률과 큐의 길이의 관점에서 제안한 방법의 향상성을 보이고자 한다.

1. 서 론

네트워크를 통해 이루어지는 인터넷 영역들이 늘어나면서 능동 큐 관리 제어 시스템이 중요한 역할을 하고 있다. 능동 큐 관리 제어 시스템은 TCP(Transmission Control Protocol)상에서 라우터에 들어오는 패킷들이 점점 늘어남에 따라 버퍼가 차는 것을 막기 위해 나중에 들어오는 패킷들을 미리 폐기시키는 역할을 한다. 그리고 예상되어지는 혼잡성을 송신단에 미리 알림으로써 패킷들의 전송 속도를 조절하여 다른 패킷들의 손실과 지연을 예방할 수 있다. 능동 큐 관리를 위한 방법으로 RED(Random Early Detection), REM(Random Exponential Marking), EWMA(exponentially weighted moving average) 등이 제안되었다.

하지만 이러한 방식들은 확률적으로 패킷들을 임의로 폐기시킴으로 급격히 변화하는 네트워크 변화 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. 이러한 문제를 해결하기 위해 본 논문에서는 직접 적응 제어기에 기반을 둔 웨이블릿 신경회로망(WNN)을 제안한다. 능동 큐 관리 제어 시스템으로부터 나온 결과인 큐의 길이가 직접 웨이블릿 신경회로망의 입력으로 들어가 시스템의 제어입력인 패킷의 폐기 확률을 생성하게 되며 신경 회로망의 파라미터 값들은 경사 하강법을 통해 학습시킨다.

2. 본 론

2.1 능동 큐 관리 제어 시스템

여러 개의 송신단에서 일정한 바이트로 묶인 패킷들을 수신단으로 전송하고자 할 경우, 패킷들은 라우터를 거치게 된다. 송신단에서 링크를 통해 라우터에 도착한 패킷들은 순서대로 라우터내의 버퍼를 지나 수신단쪽으로 가게 되는데 계속해서 들어오는 패킷들에 비해 라우터의 버퍼는 용량이 제한되어있기 때문에 모든 패킷들을 받아

들일 수 없게 된다. 따라서 버퍼가 모두 차기 이전에 들어오는 패킷들 중 일부는 폐기시켜야 하는 문제가 생긴다. 능동 큐 관리는 이와 같이 TCP상에서 라우터에 들어오는 패킷들로 인해 버퍼가 차는 것을 막기 위하여 라우터에 들어오는 패킷들을 미리 폐기시키는 것으로써 다른 패킷들의 손실과 지연을 예방한다. 기본적인 송-수신단간의 연결은 그림 1과 같다.

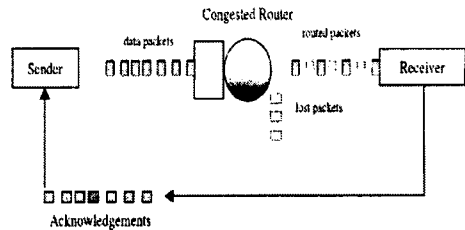


그림 1. 송-수신단간 연결 구조

TCP의 동적 모델은 확률적인 미분 방정식 형태로 나타내었다[4]. 이 모델이 TCP의 동적변화를 정확히 나타내고 있음은 시뮬레이션 결과를 통해 보여진다. 이 모델은 네트워크에서 보이는 중요한 변수들의 평균값과 관련되며 다음과 같이 비선형 방정식으로 나타내어진다.

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

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

W ≡ TCP 윈도우의 평균 크기 (패킷)
 q ≡ 큐의 평균 길이 (패킷)

$$R(t) \equiv \text{송수신단간 왕복 시간} = \frac{d(t)}{C} + T_p(\text{초})$$

C ≡ 링크 용량 (패킷/초)

T_p ≡ 지연 시간 (초)

N ≡ TCP 세션의 수

β ≡ 패킷 폐기 확률

식 1에 의하면 TCP 윈도우의 크기는 송수신단간 왕복 시간에 반비례하여 증가하며 패킷 폐기확률에 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 커야하고 크기 제한이 있다. 즉 $q \in [0, \bar{q}]$, $W \in [0, \bar{W}]$ 를 만족시켜야 하며 \bar{q} 와 \bar{W} 는 각각 버퍼 용량과 최대 윈도우 크기를 나타낸다. 또한 패킷 폐기 확률은 $\beta \in [0, 1]$ 이다[5,6].

2.2 웨이블릿 신경 회로망

웨이블릿 신경회로망은 신경회로망 내부에 웨이블릿 함수를 사용함으로써 다층구조 신경 회로망의 층수를 3 개로 줄였으면서도 성능은 향상시킨 신경 회로망이다[7]. 웨이블릿 함수는 적절한 모체 웨이블릿을 선택하고 이때 결정되는 이동(translation)인자 m 과 확장(dilation)인자 d 를 통해 결정해준다. 여기에서 사용한 모체 웨이블릿은 가우시안(Gaussian)함수의 1차 미분형이다.

$$\phi(x) = -x \exp\left(-\frac{1}{2}x^2\right) \quad (3)$$

N_i 개의 입력, 단일 출력 및 웨이블릿 층으로 구성된 웨이블릿 신경회로망의 구조는 그림 2와 같다.

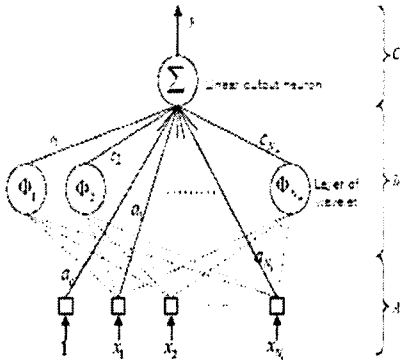


그림 2. 웨이블릿 신경회로망 구조

A층은 웨이블릿 신경 회로망의 입력단이고 B층은 웨이블릿 함수층으로 각 노드에 대한 함수는 다음과 같이 정의된다.

$$\phi_j(z) = \phi\left(\frac{x - m_j}{d_j}\right) \quad (4)$$

여기에서 m_j 는 j 번째 웨이블릿 노드의 이동인자이며 d_j 는 확장인자이다.

B층에서의 각 노드에 대한 출력은 식 5와 같다.

$$\Phi_j(x) = \prod_{k=1}^{N_w} \phi(z_{jk}) \quad (5)$$

여기에서, $z_{jk} = \frac{x_{jk} - m_{jk}}{d_{jk}}$ 이고, $j=1, \dots, N_w$, $k=1, \dots, N_i$ 이며, N_i 는 입력의 수와 웨이블릿의 차원음, N_w 는 웨이블릿 함수의 수이다. 출력인 C층은 다음과 같이 나타낼 수 있다.

$$y = \Psi(x, \theta) = \sum_{j=1}^{N_w} c_j \Phi_j(x) + \sum_{k=1}^{N_i} a_k x_k + a_0 \quad (6)$$

여기에서 a_0 와 a_k 는 입력 노드와 출력노드사이의 가중치이며 c_j 는 웨이블릿 노드와 출력노드 사이의 가중치이다. 그리고 θ 는 학습을 통해 갱신시킬 웨이블릿 신경 회로망 파라미터들의 집합으로 다음과 같다.

$$\theta = \{m_{jk}, d_{jk}, c_j, a_k, a_0\} \quad (7)$$

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

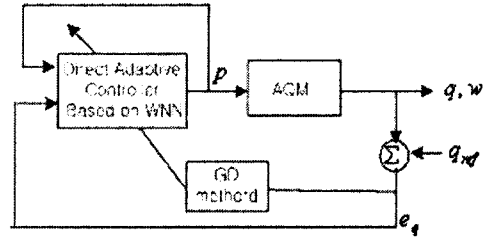


그림 3. 직접 제어 구조

제어 입력 p 를 통해 나온 능동 큐 관리 시스템의 출력인 큐의 길이는 원하는 큐의 길이와의 오차를 계산하여 직접 제어기의 입력으로 들어가게 된다. 이전 패킷 폐기 확률과 같이 두 개가 들어가는 제어기의 입력은 현재 패킷 폐기 확률이 출력으로 나오게 된다. 신경 회로망 학습은 다음과 같은 비용함수를 최소화하는 과정을 통해 파라미터들을 갱신한다.

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

여기에서 $y(n)$ 은 능동 큐 관리 제어 시스템을 통해 나온 n 번째 출력이며, $y_r(n)$ 는 기준 출력값이다. 학습방법은 경사 하강법(gradient descent method)을 이용하여 구해진다.

$$\alpha(n+1) = \alpha(n) - \eta \frac{\partial J(\alpha(n))}{\partial \alpha(n)} \quad (9)$$

여기에서, η 는 웨이블릿 신경회로망의 학습률(learning rate)을 나타내며, $\frac{\partial J(\alpha(n))}{\partial \alpha(n)}$ 는 $\alpha(n)$ 에 대한 비용함수의 기울기로서 다음과 같이 계산된다.

$$\frac{\partial J(\alpha(n))}{\partial \alpha(n)} = -e(n) \frac{\partial y(n)}{\partial \alpha(n)} \quad (10)$$

각 파라미터들은 다음과 같이 갱신된다.

$$\frac{\partial y(n)}{\partial a_0} = 1 \quad (11)$$

$$\frac{\partial y(n)}{\partial a_k} = x_k \quad (12)$$

$$\frac{\partial y(n)}{\partial c_j} = \Phi_j(x) \quad (13)$$

$$\frac{\partial y(n)}{\partial m_{jk}} = -\frac{c_j}{d_{jk}} \frac{\partial \Phi_j}{\partial z_{jk}} \quad (14)$$

$$\frac{\partial y(n)}{\partial d_{jk}} = -\frac{c_j}{d_{jk}^2} z_{jk} \frac{\partial \Phi_j}{\partial z_{jk}} \quad (15)$$

여기에서 $\frac{\partial \Phi_j}{\partial z_{jk}}$ 는 다음과 같다.

$$\frac{\partial \Phi_j}{\partial z_{jk}} = \phi(z_{j1}) \phi(z_{j2}) \phi(z_{j3}) \cdots \phi(z_{jN_i}) \quad (16)$$

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

2.3 모의실험

본 논문에서는 웨이블릿 신경 회로망에 의한 능동 큐 관리 제어 시스템의 성능을 확인하기 위해 TCP 세션의 수와 웨이블릿 신경 회로망의 학습율에 따른 모의실험을 하였다. 실험의 성능을 확인하기 위해 평균제곱오차(MSE)와 평균 큐의 길이 그리고 평균 패킷 폐기 확률측면에서 살펴보았다. TCP 모델의 파라미터 값들은 다음과 같다.[8]

표1. 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 학습율은 0.0001로 정했으며 q_{ref} 는 200으로 잡았다.

표2. TCP 세션수에 따른 성능 비교

	N=30	N=60	N=120
MSE	0.8794	0.6108	0.6430
큐의 길이	189.87	201.42	210.03
패킷 폐기확률	0.036	0.122	0.421

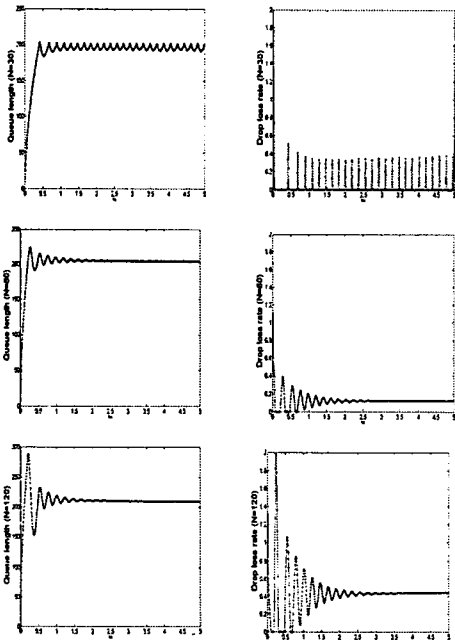


그림 4. 세션수에 따른 큐의 길이와 패킷 폐기 확률

웨이블릿 신경회로망의 학습율의 변화에 대한 큐의 길이와 패킷 폐기확률 실험은 우선 TCP 세션수는 120으로 정했으며 q_{ref} 는 200으로 잡았다

표3. 학습율(mul)에 따른 성능 비교

	mul=0.01	mul=0.001	mul=0.0001
MSE	0.4469	0.4907	0.6430
큐의 길이	196.63	202.67	210.03
패킷 폐기확률	0.5089	0.4987	0.4210

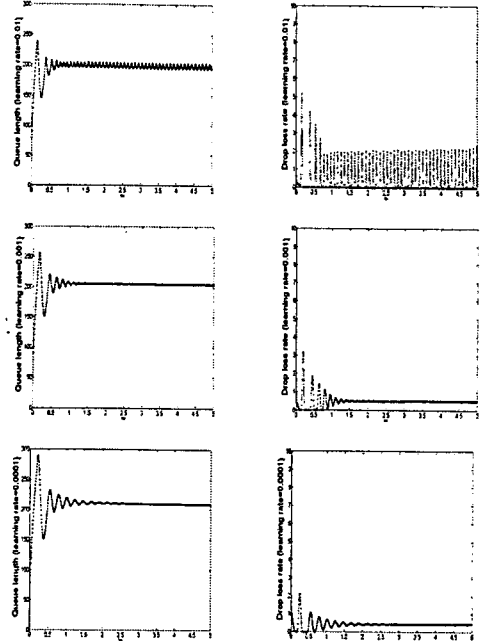


그림 5. 학습율에 따른 큐의 길이와 패킷 폐기 확률

3. 결 론

본 논문에서는 능동 큐 관리 제어 시스템을 위해 웨이블릿 신경회로망을 기반으로 한 직접적용 제어기의 설계를 제안하였다. 능동 큐 관리 제어 시스템이 이전까지는 RED, REM등 네트워크 관점에서 다루어졌지만 모의실험을 통해 웨이블릿 신경 회로망과 같은 제어기법을 통해서도 충분히 다룰 수 있음을 확인할 수 있다. 하지만 여기에서는 서로 성능의 우수성을 판단할 수 없으며 또 다른 제어 기법은 PL PD-controller과의 성능 우수성이 비교되지 않았으므로 앞으로는 이와 관련된 연구가 계속 되어야 할 것이다.

[참 고 문 헌]

- [1] S. Ryu, C.Rump and C.Qiao, "A Predictive and Robust Active Queue Management for Internet Congestion Control", Proceedings of ISCC, Vol. 2, pp. 991-998, 2003.
- [2] FAN Yanfei, REN Fengyuan, and LIN 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] Vishal Misra, Wei-Bo Gong, and 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, Vishal Misra, "Analysis and Design of Controller for AQM Routers Supporting TCP Flows" IEEE Transactions on automatic control, Vol. 47, No. 6, pp.945-959, 2002.
- [6] C.V. Hollot, Vishal Misra, D. Towsley and W.Gong, "A Control Theoretic Analysis of RED", IEEE INFOCOM, Vol. 3, pp. 1510-1519, 2001.
- [7] Q. Zhang and A. Benveniste, "Wavelet Networks," IEEE Transactions on Neural Networks, Vol. 3, No. 6, pp.889-898, 1992.
- [8] Ren Fengyuan, Ren Yong, and Shan Xiuming, "Design of a fuzzy controller for active queue management", Computer communications, pp. 874-883, 2002.