멀티미디어 인터넷 전송을 위한 전송률 제어 요소의 신경회로망 모델링

Modeling of Multimedia Internet Transmission Rate Control Factors Using Neural Networks

유 성 구*, 정 길 도
(Sung-Goo Yoo and Kil-to Chong)

Abstract : As the Internet real-time multimedia applications increases, the bandwidth available to TCP connections is oppressed by the UDP traffic, result in the performance of overall system is extremely deteriorated. Therefore, developing a new transmission protocol is necessary. The TCP-friendly algorithm is an example satisfying this necessity. The TCP-Friendly Rate Control (TFRC) is an UDP-based protocol that controls the transmission rate that is based on the available round trip time (RTT) and the packet loss rate (PLR). In the data transmission processing, transmission rate is determined based on the conditions of the previous transmission period. If the one-step ahead predicted values of the control factors are available, the performance will be improved significantly. This paper proposes a prediction model of transmission rate control factors that will be used in the transmission rate control, which improves the performance of the networks. The model developed through this research is predicting one-step ahead variables of RTT and PLR. A multiplayer perceptron neural network is used as the prediction model and Levenberg-Marquardt algorithm is used for the training. The values of RTT and PLR were collected using TFRC protocol in the real system. The obtained prediction model is validated using new data set and the results show that the obtained model predicts the factors accurately.

Keywords : multimedia transmission, one-step ahead prediction modeling, neural network

I. 서론

인터넷 트래픽의 대부분은 HTTP(Hypertext Transfer Protocol), SMTP(Simple Mail Transfer Protocol), FTP(File Transfer Protocol) 등과 같은 TCP를 기반으로 하는 프로토콜로 인해 발생한다. 그러나 IP 텔레포니, 인터넷 오디오 플레이어, VOD 등과 같은 실시간 오디오/비디오 스트리밍 어플리케이션이 증가함에 따라 이것들은 인터넷 트래픽의 새로운 중요한 원인이 되고 있다. 일반적으로 실시간 어플리케이션은 복잡한 재전송 알고리즘을 사용하는 TCP[1]를 사용하지 않고 혼잡제어를 고려하지 않는 UDP[1] 알고리즘을 사용한다. 만약, TCP와 UDP가 한 개의 동일한 링크를 공유한 상태에서 혼잡상황이 발생하게 된다면, TCP는 혼잡 문제를 해결하기 위하여 전송률을 감소시키지만, UDP는 원래의 전송률을 계속 유지하기 때문에 유효 대역폭의 대부분을 점유하게 되며 혼잡상황을 가중시킨다. 뿐만 아니라 네트워크의 사용에 있어서 불공정이 발생한다.

이러한 문제를 해결하는 방안으로 TCP의 전송률 수정 메커니즘과 양립할 수 있도록 non-TCP 트래픽에 대하여 전송률 수정규칙을 적용할 수 있다. 이러한 전송률 수정 규칙은 non-TCP 어플리케이션이 TCP-friendly 성질을 갖도록 만들어야 하며 시스템은 균등한 분배를 지원해야 한다. 불균등 분배를 해결하는 여러 TCP-friendly 알고리즘이 제안되었다[2,3,4]. 현재 제안된 주요 TCP friendly 알고리즘에는 RAP(Rate Adaptation Protocol) [5]와 TCP Friendly Rate Control [6] 등이 있다.

TCP-friendly 알고리즘의 중요한 특징의 하나는 현재 인터넷의 혼잡상황을 측정하여 적응적으로 전송률을 제어하는 것이다. 하지만 이것은 TCP flow와의 fairness 측면만을 강조한 부분으로 전송된 영상의 화질에 영향을 미치는 QoS(Quality of Service)에 대한 부분은 고려하지 않았다. 또한 기존의 TCP-friendly는 현재를 기준으로 한단계 이전의 왕복 지연 시간(RTT)과 패킷 손실율(PLR)을 사용하여 전송률을 제어한다. 본 연구에서는 좀더 능동적으로 혼잡상황에 대처하며 보다 향상된 유효 전송율로 데이터를 전송하기 위하여, 데이터가 전송되는 시간의 인터넷 대역폭 즉 RTT 와 PLR를 예측하여 예측된 값에 의하여 전송률을 제어하는 방법을 고려하여, 예측 전송에 중요 요소인 RTT와 PLR을 신경회로망을 이용하여 예측하는 모델링을 실시하였다.

예측 모델링 방법에는 의사 결정 나무(decision tree), 룰(rule), 신경회로망(neural network) 등이 있는데, 본 연구에서는 불확실한 비선형 시스템의 모형화 및 재현이 가능한 신경회로망을 이용하였다[7,8]. 신경회로망 모델로는 비선형 입출력 특성을 추출하는 다층 퍼셉트론(MLP) 구조[9,10]를 사용하였으며, 학습 방법으로는 역전파알고리즘(back-propagation)[11,12]의 단점인 지역 최소 점에 수렴하는 문제를 개선하고 학습능력을 향상시킨 LMBP(Levenberg-Marquardt Back-propagation)[13]을 사용하였다.

모델링에 필요한 데이터를 수집하기 위해 리눅스 OS운영체제를 갖춘 2대의 컴퓨터를 서울대학교와 전북대학교에 각각 설치하였으며, TFRC 알고리즘을 이용하여 상호간에 패킷 전송을 실시하였다. 그리고 다양한 네트워크 혼잡상황을 구성하기 위하여 traffic generator인 IPERF[14]를 사용하였다. 본 논문의 구성은 다음과 같다. 2장에서는 전송 실험에 사용한 TFRC에 대한 설명을, 3장에서는 신경회로망의 구조와 LMBP학습 알고리즘에 대해서 기술하였다. 그리고 4장에서

* 책임저자(Corresponding Author)
논문접수 : 2004. 9. 22., 채택확정 : 2004. 12. 1.
유성구, 정길도 : 전북대학교 전기전자제어공학부
(ding5@chonbuk.ac.kr/kitchong@chonbuk.ac.kr)

는 예측 모델링에 대한 시뮬레이션과 분석 결과에 대해서 기술하였다. 마지막으로 5장에서는 본 연구에 대한 결론과 향후 연구에 대해서 기술하였다.

II. TCP-Friendly 전송률 제어

멀티미디어는 웹 서버로부터 제공받을 때 전체 파일을 모두 내려 받은 후 재생하는 방법과, 전체 파일의 일부를 내려 받은 후 재생과 함께 나머지 파일을 동시에 내려 받는 스트리밍 방법이 있다. 스트리밍 방법이 오디오나 비디오의 실시간 방송에 적합하다.

이러한 인터넷을 이용한 멀티미디어 전송의 경우 패킷 손실은 주로 전송 오류와 체증에 의해서 발생한다. 패킷 손실 발생시 TCP는 자체적인 체증 제어 방법에 의해 전송률을 감소시킨다. 따라서 유사한 왕복 지연 시간을 가진 TCP 연결들이 동일한 채널을 공유하고 있다면 그 TCP 연결들은 가용 대역폭을 균일하게 분배하여 갖게 된다. 과거에는 대부분의 트래픽이 TCP 기반 프로토콜을 사용하여 대역폭 분배가 문제되지 않았으나, IP telephony, 영상회의 등의 실시간 응용 서비스와 음성/영상 스트리밍 서비스 등의 Non-TCP 트래픽의 사용이 증가함에 따라 가용 대역폭의 분배 문제가 중요하게 대두되었다. 하지만 Non-TCP 트래픽은 TCP와 양립할 수 있는 혼잡 제어 방법이 없기 때문에 혼잡 발생시 TCP는 전송률을 줄이지만 Non-TCP 프로토콜은 원래의 전송률로 계속 전송하여 전체 트래픽을 가중시킨다. 따라서 Non-TCP 트래픽도 TCP와 양립할 수 있는 전송률 제어 메커니즘이 필요하며 Non-TCP 트래픽을 TCP-Friendly하게 만들어 주어 가용 대역폭을 공정하게 분배 하므로 문제들을 해결하는 방법이 TCP-Friendly 혼잡 제어 방법이다.

TCP-Friendly 혼잡 제어 방법은 TCP 모델의 전송 메커니즘[15,16]을 근거로 전송률을 계산하는데 이는 TCP의 정상 상태 동작을 고려하여 시간상 평균 전송률을 모델링 한 것이다. TCP의 동작에 따라 여러 가지 형태로 나타낼 수 있으나 기본적으로 (1)과 같은 형태로 나타낼 수 있다.

$$R = f(PLR, RTT) \tag{1}$$

여기서 R 은 전송률, PLR 은 패킷 손실률, RTT 는 왕복지연시간이다.

그림 1은 일반적인 TCP Reno 모델[17,18]의 동작 원리를 나타낸 것으로서 패킷 손실이 발생할 경우 혼잡윈도우의 크기를 반절로 줄이는 동작을 보여준다.

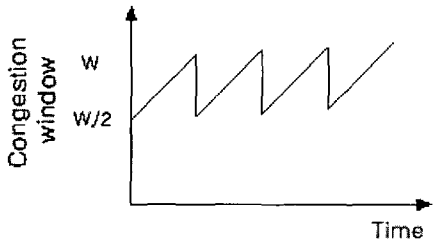


그림 1. TCP Reno 모델의 정상 상태 동작 메커니즘.
Fig. 1. TCP Reno's congestion window in steady-state.

여기서 W 을 혼잡 윈도우 크기, s 를 패킷의 크기라고 한다면 패킷이 손실되기 전까지의 전송률은 $R = \frac{W \times s}{RTT}$ 이고, 패킷이 손실된 경우 윈도우의 크기는 $\frac{w}{2}$ 가 되어 전송률은 $R = 0.5 \times \frac{W \times s}{RTT}$ 로 된다. 따라서, 톱니바퀴 사이클 4개 전구간의 평균 전송률은 $R = 0.75 \times \frac{W \times s}{RTT}$ 이 된다. 그림 1에서 하나의 톱니바퀴 사이클에서의 손실률 p 에 대한 식은 $\frac{1}{p} = \left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2$ 이며, $W \approx \sqrt{\frac{8}{3p}}$ 이 되어 최종적으로 시간 t 에서 유효 전송률 $R(t)$ 는 (2)와 같은 근사식으로부터 얻어진다[19,20].

$$R(t) = \frac{1.22 \times s}{RTT(t) \times \sqrt{p(t)}} \tag{2}$$

본 논문은 (2)를 이용한 실험을 통하여 왕복지연시간(RTT)과 패킷 손실률(PLR) 데이터를 수집하였다.

III. 신경회로망

1. 신경회로망의 구조

본 연구에서 사용된 예측 모델구조는 다층 퍼셉트론 신경회로망으로 그림 2와 같이 입력층(input layer), 은닉층(hidden layer) 그리고 출력층(output layer)의 3개의 층으로 구성되어 있다.

다층퍼셉트론의 출력 $\hat{y}$ 는

$$\begin{aligned}\hat{y}_i(t) &= g_i[\varphi, \theta] \\ &= F_i\left[\sum_{j=1}^{n_h} W_{i,j} f_j\left(\sum_{l=1}^{n_\varphi} w_{j,l}\varphi_l + w_{j,0}\right) + W_{i,0}\right]\end{aligned} \tag{3}$$

와 같다[11]. 여기서 φ 는 입력, θ 는 신경회로망 구조에서 조정이 가능한 모든 매개변수를 포함하는 매개변수벡터이며, $\{w_{j,l}, W_{i,j}\}$ 은 연결강도와 바이어스이다. 보통 바이어스는 1로 사용한다. 연결강도를 결정하기 위해서는 출력 $\hat{y}_i$ 가 어

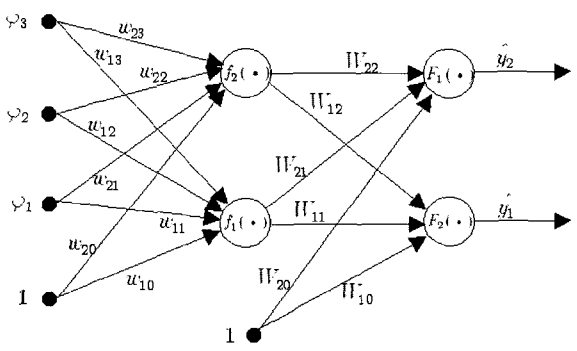


그림 2. 다층 퍼셉트론 신경회로망의 구조.
Fig. 2. Multi-layer perceptron neural network structure.

떻게 입력 φ와 연관되는 지를 보여주는 학습데이터가 필요하며, 이 학습데이터를 이용하여 연결강도를 결정짓는 과정을 학습이라 부른다.

예측 목표값과 신경회로망 출력의 차이를 나타내는 오차함수 E는 (4)와 같이 정의된다.

$$E=\frac{1}{2}\sum_{n-1}^{k}(y_n-o_n)^2 \tag{4}$$

여기서 y_n는 예측하고자 하는 목표값이고, o_n는 신경회로망의 출력을 의미한다.

2. LM-BP 학습방법

신경회로망에서 사용될 수 있는 학습 알고리즘들은 크게 최급강화법(Steepest Descent), 뉴우톤 방법(Newton), 가우스-뉴우톤 방법(Gauss-Newton)[11] 등으로 구분할 수 있다. 이 방법들 중에서 최급강화법은 수렴성에 문제점을 가지고 있다. 더 좋은 수렴성을 가지는 뉴우톤 방법은 2차 도함수를 사용하며, 이것으로 인하여 계산상 어려움이 존재한다. 그래서 일반적으로 실제 신경회로망의 적용에서는 가우스-뉴우톤 방법이 많이 사용되고 있다.

가우스 뉴우톤 방법중의 하나인 LMBP 알고리즘은 동적으로 최급강화법과 뉴우톤 방법의 문제를 해결할 수 있다. 즉 학습 초기에는 최급강화법을 사용하면서 가중치를 크게 설정하여 학습하고, 어느정도 학습된 상황에서 수렴이 느려지게 되면 뉴우톤 방법에 가중치를 주어 국부적 최소치로 수렴시킨 후 다시 최급강화법에 의해 빠르게 최적해 쪽으로 수렴하게 하는 방법이다. 즉, 다음과 같은 (5)에 의해 학습함으로 가중치 w를 구한다.

$$w_{i+1}=w_i-(H+\lambda I)^{-1}\nabla F(w_i) \tag{5}$$

여기서

$$\nabla F(w_i)=\frac{\partial F}{\partial w_i}:gradient\ ,\ w_i\text{는 }i\text{번째 가중치} \tag{6}$$

$$F=\sum_{k=0}^{N}e_k^2\text{ 는 SSE(Square-Sum Error),}$$
$$k\text{ 는 }k\text{ 번째 샘플} \tag{7}$$

$$H=\nabla^2F(w)\text{ 는 Hessian matrix} \tag{8}$$

이며 λ는 동적으로 조절된다.

하지만 실제 LMBP알고리즘을 사용하는 BP신경회로망에서는 뉴우톤 방법의 H가 2차 도함수를 사용하기 때문에 이를 1차 도함수로 근사화시켜 사용하는 가우스-뉴우톤 방법이 사용된다. 즉 뉴우톤 방법에서의 H는 (9)과 같이 구할수있으며,

$$\begin{aligned}H&=\left[\nabla^2F(w)\right]_{ij}=\frac{\partial^2F(x)}{\partial w_i\partial w_j}\\&=2\sum_{k=0}^{N}\left[\frac{\partial e_k(w)}{\partial w_i}\frac{\partial e_k(w)}{\partial w_j}+e_k(w)\frac{\partial^2e_k(w)}{\partial w_i\partial w_j}\right]\end{aligned} \tag{9}$$

(9)에서의 두 번째 항은 충분히 무시할 수 있는 항이므로

$$\begin{aligned}\left[\nabla^2F(w)\right]_{ij}&\cong 2\sum_{k=0}^{N}\frac{\partial e_k(w)}{\partial w_i}\frac{\partial e_k(w)}{\partial w_j}\\&=2J^T(w)J(w)\end{aligned} \tag{10}$$

와 같이 나타낼 수 있다. 여기서, $J_{ki}=\frac{\partial e_k}{\partial w_i}$은 Jacobian matrix이다.

이 근사화를 사용함으로써, 2차 도함수의 필요성을 제거할 수 있다. 그리고 (10)에서의 $\nabla F(w_i)$는

$$\nabla F(w_i)=J^T(w_i)e(w_i) \tag{11}$$

으로 정의될 수 있기 때문에 수정된 LMBP알고리즘은

$$w_{m+1}=w_m-\left[J^T(w_m)J(w_m)+\lambda_mI\right]^{-1}J^T(w_m)e(w_m) \tag{12}$$

로 최종 정리된다. (12)는 m 번째 반복 과정에서 조정되는 가중 파라미터이다.

여기서, $\lambda=0$는 순수한 가우스-뉴우톤 방법이고, $\lambda\to\infty$ 일 경우에는 최급 강화 방법이다. 전형적으로 $\lambda_k=0.01$로 시작되며 만약 SSE(Square-Some Error)가 충분히 작지 않으면 새로운 $\lambda_k=\lambda_k\cdot\theta$로 대치된다. 여기서 θ는 $1<\theta<10$의 상수로서 λ을 증가시키기 위한 일종의 배율기이다.

IV. 예측 신경회로망 모델링

1. 데이터 수집

학습과 검증에 사용할 데이터를 수집하기 위하여 그림 3과 같이 실험환경을 구축하였다. 서버와 클라이언트는 리눅스 운영체제를 사용하는 컴퓨터 2대를 사용하였으며, 각각 서울대와 전북대에 설치하였다.

ANSI C언어의 소켓프로그램을 사용하여 전송프로세서(TP: Transmission Processor), 수신-재전송프로세서(RP: Retransmission Processor), RTT-PLR측정 프로세서(RTT-PLR EP: RTT-PLR Estimation Processor)를 각각 프로그래밍 하였으며, 전송프로세서와 RTT-PLR측정 프로세서는 전북대의 서버에 설치하였으며, 수신-재전송프로세서는 서울대의 클라이언트에 설치하였다.

표 1. 컴퓨터 구성.

Table 1. Composition of computers.

	운영체제	CPU	Memory
서버	Redhat Linux 8.0	Pentium IV 1.7Ghz	512Mbyte
클라이언트	Redhat Linux 8.0	AMD Athlon 1200	512Mbyte

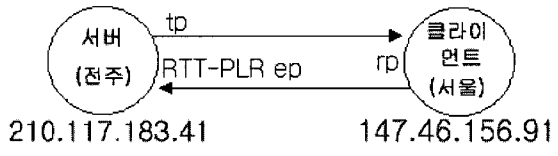


그림 3. 실험 시스템.

Fig. 3. Experiment system.

전송프로세서는 2절에서 논의한 TCP-Friendly 방법으로 패킷을 전송하게 된다. 패킷 전송 실험은 (2)를 사용하여 전송하였다. 초기 전송 속도는 100Kb/s이고, 패킷의 크기는 총 1088byte이며, 이 중 64byte는 프로브 헤더의 크기이다. 프로브 헤더는 RTT와 PLR을 측정하기 위해 전송 패킷의 헤더 부분에 부착되며, 각 프로세서에서 전송된 패킷의 순서를 표시하는 순서번호와 전송된 시각을 저장할 수 있게 구성되어 있다[19]. 그림 4는 프로브 헤더의 구성을 나타낸다. 전송프로세서에서 수신-재전송프로세서로 패킷을 전송하게 되면, 패킷을 수신한 수신-재전송프로세서에서는 프로브 헤더를 분리하여 프로브 헤더에 패킷번호와 현재시간을 입력한 후 RTT-PLR측정 프로세서로 재전송하게 된다. 프로브 헤더를 수신한 RTT-PLR ep에서는 (13)와 (14)을 사용하여 RTT와 PLR을 측정한다.

$$RTT = \text{RTT-PLR ep의 패킷 도착시간} - \text{tp의 패킷 전송시간} \tag{13}$$

$$\begin{aligned} PLR(\%) &= 1 - \frac{\text{라운드 } i\text{에서 수신한 패킷의 총합}}{\text{라운드 } i\text{에서 송신한 패킷의 총합}} \times 100 \\ &= 1 - \frac{R_i - R_{i-1}}{S_i - S_{i-1}} \times 100 \end{aligned} \tag{14}$$

R_i : 라운드 i 에서 마지막으로 수신한 데이터의 순서번호

S_i : 라운드 i 에서 마지막으로 송신한 데이터의 순서번호

R_{i-1} : 라운드 $(i-1)$ 에서 마지막으로 수신한 데이터의 순서번호

S_{i-1} : 라운드 $(i-1)$ 에서 마지막으로 송신한 데이터의 순서번호

여기서, 라운드는 2초 간격을 의미한다.

인터넷에 트래픽 문제가 발생할 경우 RTT와 PLR은 급격하게 변동하게 된다. 급변하는 RTT와 PLR을 TFRC 메커니즘에 바로 적용시켜 전송률을 제어할 경우 좀 더 신속하게 트래픽 문제를 해소할 수 있지만, 이 경우 실시간 어플리케이션의 서비스 상태는 급격히 떨어지게 된다. 이를 방지하기 위해 TFRC 알고리즘의 RTT와 PLR은 TCP 알고리즘의 RTT와 PLR의 계산 방법을 따른다. TCP는 저대역(low-pass) 필터를 사용하여 자연스럽게 RTT 측정(estimation)값과 PLR 측정값을 변경하도록 되어 있다. RTT와 PLR의 측정값은 (15)과 같이 이동 평균값(moving average)을 이용한다.

$$\begin{aligned} RTT^* &= \alpha RTT^* + (1-\alpha) new\_RTT \\ PLR^* &= \alpha PLR^* + (1-\alpha) new\_PLR \end{aligned} \tag{15}$$

여기서 α 는 권장값 0.9를 갖는 모멘트 계수이고, new_RTT 와 new_PLR 은 새로 측정된 RTT와 PLR이다. 이동 평균값은 트래픽 문제 발생시 RTT와 PLR의 급격한 변화를 감소시켜서 전송률 제어에 사용된다. 전송 실험시 매 라운드(2초)마다 RTT와 PLR을 측정하였으며, (15)을 이용하여 moving average RTT와 moving average PLR을 계산하였다. 본 연구에서는 RTT, PLR, moving average RTT 그리고 moving average PLR을 각각 학습시켰으며, 예측 결과를 검증하였다.

네트워크의 다양한 상황을 구현하는 방법으로 트래픽 생성기인 IPERF[14]를 사용하여 전송 실험 과정에서 네트워크의 트래픽을 조절해 보았다.

표 2는 IPERF를 사용하여 생성한 트래픽의 용량과 그에 상응하는 실험의 RTT와 PLR의 측정 결과이다. 네트워크에

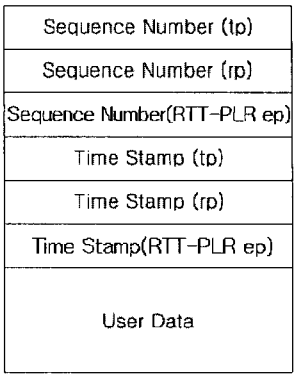


그림 4. 프로브 헤더.

Fig. 4. Probe header.

표 2. 트래픽 크기에 따른 RTT와 PLR.

Table 2. RTT and PLR according to traffic size.

	0 MB	1 MB	2 MB	5 MB	7 MB
Max RTT	10.8ms	17.3ms	23.9ms	45.2ms	79.3ms
Min RTT	10.1ms	11.5ms	19.4ms	29.7ms	34.9ms
Max PLR	1.2%	2.4%	1%	12.7%	14.4%
Min PLR	0%	0.5%	3.8%	11.3%	12.5%

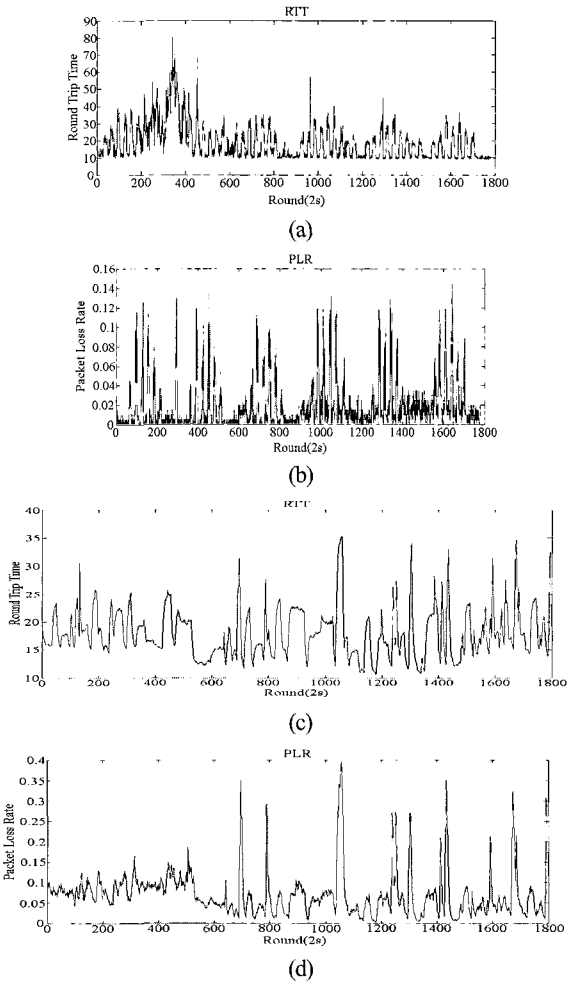


그림 5. 전송 실험 결과 측정된 RTT와 PLR의 값.

Fig. 5. Value of RTT and PLR that is measured according to transmission experiment result.

트래픽을 부과하지 않을 경우의 RTT는 평균 10.3ms 정도 였으며 PLR은 평균 0.5%정도였다. 트래픽의 부하가 증가할수록 RTT와 PLR은 급격하게 증가하였다.

패킷 전송 실험은 일주일동안 매 시간마다 실시 하였으며 30분간 전송하는 실험을 실시하였다. RTT와 PLR의 측정은 편의상 2초를 의미하는 라운드 단위를 사용하였다. 그림 5는 전송 실험을 실시하여 수집한 RTT와 PLR의 값이다.

그림 5의 (a),(b)는 IPERF를 사용하여 매 60초마다 트래픽을 생성했을 경우에 해당하는 RTT와 PLR의 결과 그래프이며, (c)와 (d)는 트래픽을 임의로 생성하여 실험을 실시한 RTT와 PLR의 결과 그래프이다.

2. 모델링

본 논문에서 실험을 통하여 수집된 RTT와 PLR데이터는 각각 약 150,000개이다. 이 중 70%는 매개변수를 측정하기 위한 학습에 사용하였으며, 나머지 30%는 신경회로망의 성능 측정을 위한 검증에 사용되었다.

예측 모델은 그림 2에서 표기된 다층 퍼셉트론 구조를 사용하였으며, 입력층 20개, 은닉층 8개, 출력층이 1개의 구조를 가진 다층 신경회로망 구조를 사용하였다. 그리고 학습 방법은 3장에서 설명한 LMBP를 사용하였다[21].

뉴런에 사용한 활성화 함수는 Hyperbolic Tangent함수를 사용하였으며, (16)과 같다.

$$f(x) = \tanh(x) = 1 - \frac{2}{\exp(2x)+1} \tag{16}$$

학습을 통하여 입력층-은닉층 사이에 168개의 연결강도와 은닉층-출력층 사이에서 9개의 연결강도를 구하였다.

그림 6은 왕복 지연 시간 데이터를 학습시킨 결과의 일부분을 나타낸 그래프이며, 그림 7은 학습에 사용하지 않은 데이터를 본 연구를 통해 구한 신경회로망 모델에 적용시켜 검증한 그래프이다. 그래프 내에서 RTT의 값이 증가하는 부분들은 4.1절에서 언급한 IPERF 트래픽 생성기를 사용하여 네트워크의 부하를 조절한 부분이다. 학습결과를 통하여 네트워크에 부하가 많이 걸렸을 경우, 즉 RTT가 갑자기 증가하는 부분에서도 신경회로망 모델의 예측 오차가 작은 것을 인할 수 있다. 학습에 사용되지 않은 데이터를 사용하여 검증한

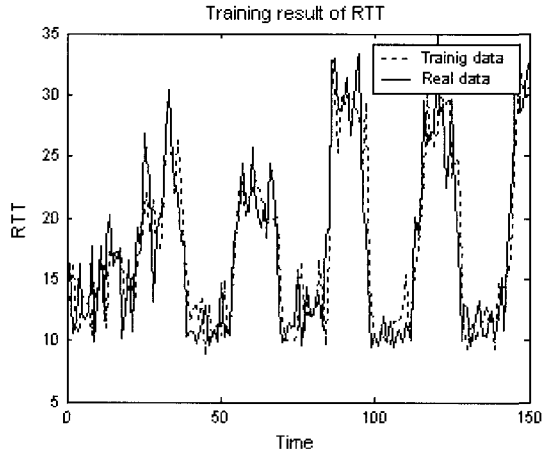


그림 6. 왕복 지연 시간의 예측 신경회로망 모델 학습 결과 그래프.

Fig. 6. Training graph of RTT's prediction NN model.

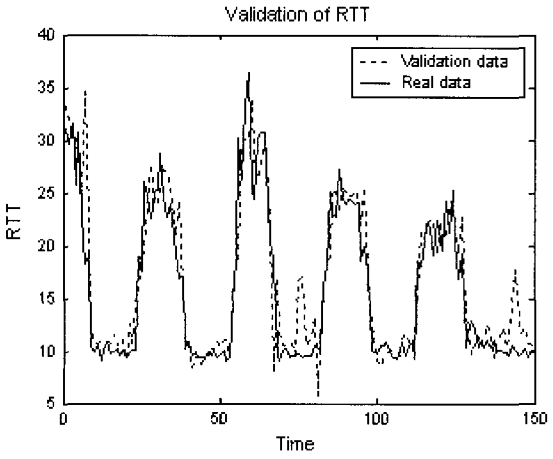


그림 7. 왕복 지연 시간의 예측 신경회로망 모델 검증 그래프.

Fig. 7. Validation graph of RTT's prediction NN model.

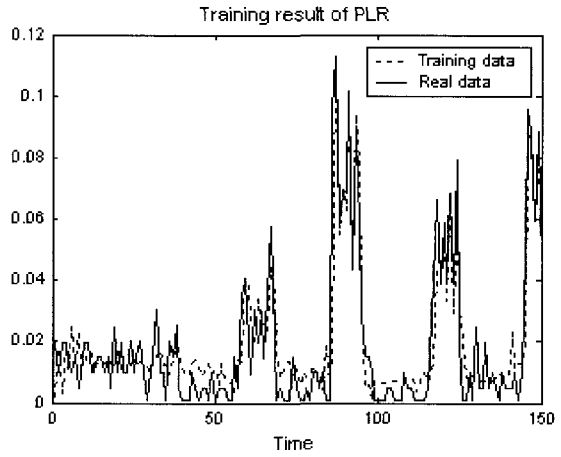


그림 8. 패킷 손실률의 예측 신경회로망 모델 학습 결과 그래프

Fig. 8. Training graph of packet loss rate's prediction NN model.

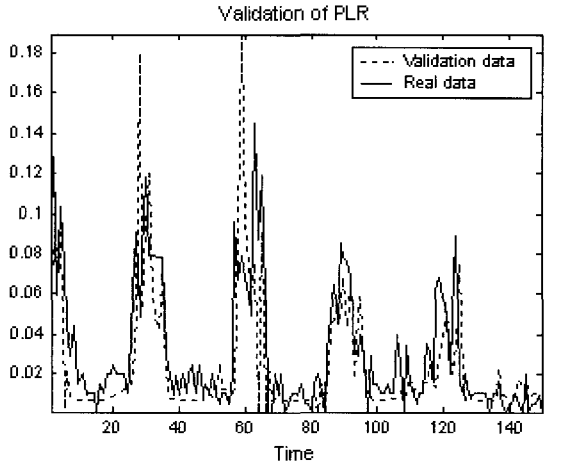


그림 9. 패킷 손실률의 예측 신경회로망 모델 검증 그래프.

Fig. 9. Validation graph of packet loss rate's prediction NN model.

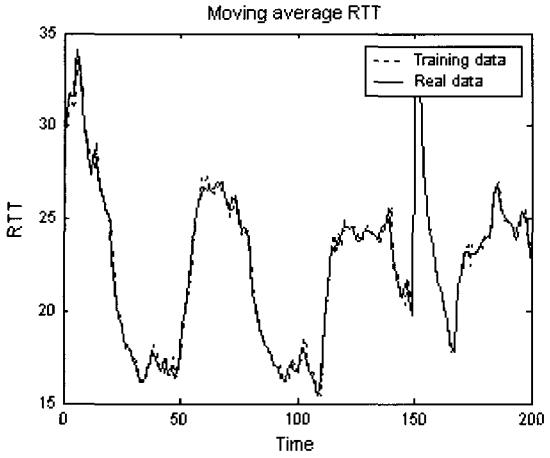


그림 10. Moving average RTT의 예측 신경회로망 모델 학습 결과 그래프.

Fig. 10. Training graph of moving average RTT's prediction NN model.

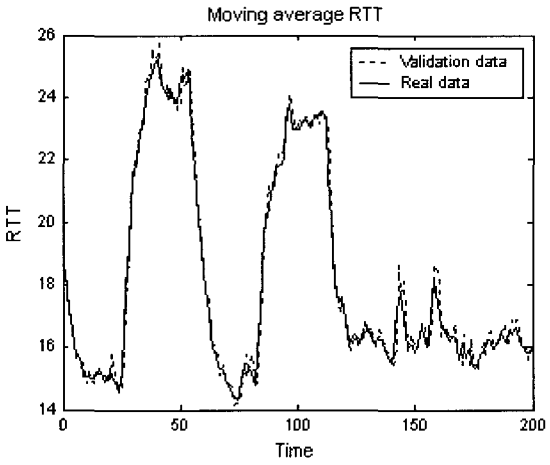


그림 11. Moving average RTT의 예측 신경회로망 모델 검증 그래프.

Fig. 11. Validation graph of moving average RTT's prediction NN model.

결과 그래프에서는 약간의 오차가 발생하는 것을 볼 수 있지만 네트워크 부하시의 예측값은 오차가 적은 것이 확인된다. 예측값의 평균제곱오차는 학습데이터의 경우 0.34ms이며, 검증데이터의 경우 1.004ms로서 RTT의 전체 범위 내에서는 아주 미세한 오차 범위이다.

그림 8과 9는 각각 패킷 손실률에 대해서 신경회로망 모델의 학습과 검증 결과의 일부를 나타낸 그래프이다. 패킷 손실률의 경우 측정된 수치가 매우 작은 값이기 때문에 조그마한 값의 변동에도 그래프가 심하게 변동하는 것을 볼 수 있다. 하지만 그래프의 결과를 보면 네트워크 부하가 걸리는 구간에서 예측값의 오차가 적은 것을 확인할 수 있다. 예측값의 평균 제곱 오차는 학습결과가 1.52(%)이며, 검증결과는 3.82(%)이다.

그림 10과 11은 (15)을 사용하여 측정된 moving average RTT를 제안한 예측 시스템을 통하여 학습과 검증 결과를 나타낸 그래프이다. moving average RTT는 심하게 변동하는 RTT

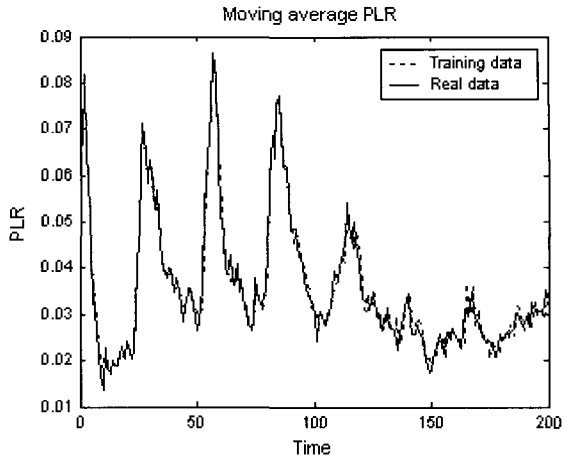


그림 12. Moving average PLR의 예측 신경회로망 모델 학습 결과 그래프.

Fig. 12. Training graph of moving average PLR's prediction NN model.

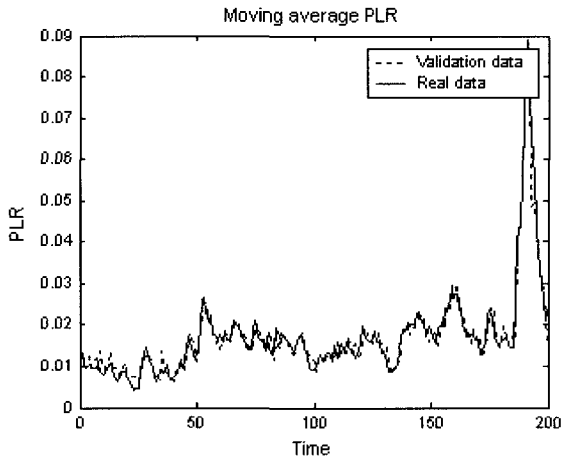


그림 13. Moving average PLR의 예측 신경회로망 모델 검증 그래프.

Fig. 13. Validation graph of moving average PLR's prediction NN model.

에 비해서 완만한 변동폭을 가지기 때문에 학습결과와 예측 성능을 나타내는 검증결과가 그림 9,10에 비해서 향상된 것을 확인할 수 있다. 학습결과의 평균제곱오차는 0.680ms 이며, 예측 성능을 나타내는 검증결과의 오차는 0.7502ms로 예측시스템의 성능을 확인할 수 있다.

그림 12, 13은 moving average PLR에 대해서 학습과 검증결과를 나타낸 그래프이다. moving average PLR도 마찬가지로 급격한 변동을 보이는 PLR에 비해 학습결과와 예측성능이 향상됨을 확인할 수 있다. 학습결과의 평균제곱오차는 0.87(%)이며, 검증결과 예측성능의 평균제곱오차는 0.9(%)이다.

V. 결론

본 논문에서는 인터넷을 통해 데이터를 전송하는데 있어서, 인터넷 대역폭을 고려한 전송률 제어 메커니즘에서 중요한 요소를 미리 예측하는 예측 모델을 설계하였다. 예측 모델링 방법으로는 비선형 시스템의 모델링이 가능한 신경회

로망을 사용하였으며, 지역적 최소 점에 수렴하지 않으며, 빠른 수렴성을 보이는 LMBP 알고리즘을 사용하였다. UDP기반의 적응전송제어 방법인 TFRC 전송 방식을 사용하여 RTT와 PLR의 데이터를 수집하였으며, 수집한 데이터를 신경회로망 예측 모델의 학습 데이터로 이용하였다. 신경회로망 학습을 통한 예측 모델의 설계 결과 예측 모델이 한 단계 이후의 RTT와 PLR을 예측 할 수 있으며, 예측 결과의 오차가 적은 것을 확인할 수 있었다. 이러한 결과는 제안한 예측 모델을 사용하면 인터넷의 혼잡 상황을 예측할 수 있으며, 능동적으로 혼잡 상황에 대처하여 많은 양의 데이터를 전송할 수 있는 메커니즘 구현 가능성을 보여준다. 본 연구를 통하여 구한 신경회로망 예측 모델을 이용하여 실재 멀티미디어 데이터의 전송률을 제어하는 시스템을 구성하고 기존의 방법과 비교함으로 예측알고리즘의 성능을 확인하는 것이 향후 진행될 연구 부분이다.

참고문헌

[1] A. S. Tanenbaum, *Computer networks(third edition),* Prentice Hall International, Inc, 1996.

[2] J. Widmer, R. Denda, and M. Mauve, Parkitsche Informatic IV, "A Survey on TCP-friendly congestion control," *IEEE Network*, vol. 3, pp. 28-37, May/June, 2001.

[3] L. Rizzo, "Pgmcc: A TCP-friendly signle-rate multicast congestion control scheme," *Proc. ACM SIGCOMM,* Stockholm, Sweden, pp 17-28, Aug 2000.

[4] S. Sisalem and A. Wolisz, "MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments," 8th Int'l. Wksp. QoS, June 2000.

[5] D. Rajate, M. Handley, D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism or realtime streams in the Internet," *INFOCOM'99,* pp. 1337-1345 vol. 3, March 1999.

[6] J. Mahadavi and S. Floyd, "TCP-friendly unicast rate-based flow control," *Tech. Rep., Technical note sent to the end2end interest mailing list,* January 1997, Available at: http://www.psc.edu/networking/tcp_friendly.html.

[7] J. G. Lee, J. Y. Choi, "Modeling of nuclear power plant Steam Generator using Neural Networks," *Journal of Control, Automation and System Engineering.* vol. 4, no. 4, August, 1998.

[8] 최진영, 박현주, "신경회로망을 이용한 시스템 모델링 및 제어", 제어 · 자동화 · 시스템공학회지, 제1권 제3호, pp. 62-73, 1995.

[9] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural network," *IEEE Trans. Neural networks,* vol. 1, no. 1, pp. 4-27, March, 1990.

[10] 왕현민, 허경무, 우광준, "개선된 신경망을 이용한 헬리콥터 고도 제어기 설계", 제어 · 자동화 · 시스템공학 논문지, 제7권 제3호, pp. 229-237, 2001.

[11] M. Norgaard, O. Ravn, N. K. Poulsen and L. K. Hansen, "Neural networks for modeling and control of dynamic systems," *A practitioner's Handbook,* Springer.

[12] S. Haykin, *Neural networks,* MacMillan, 1994.

[13] Finschi, "An implementation of the Levenberg-Marquardt algorithm," *clausiusstrasses 45,* CH-8092, Zuerich, 1996.

[14] The IPERF, "http://dast.nlanr.net/Projects/Iperf/".

[15] V. Jacobson, "Congestion avoidance and control," *SIGCOMM Symposium on Communications Architectures and Protocols,* pages 214-329, 1988.

[16] J. Michael, L. Kenneth, *The pocket guide to TCP/IP sockets : C version,* Morgan Kaufmann Publishers, Inc. 2001.

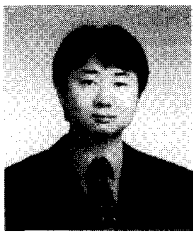
[17] V. Paxson, "Automated packet trace analysis of TCP implementations," *IN Proceedings of SIGCOMM 97,* 1997.

[18] J. Padhye, V. Firoiu, D. Towsley, J. Kurose. "Modeling TCP throughput : A simple model and its empirical validation," *ACM SIGCOMM,* 1998.

[19] I. J Yeom, *ENDE : an end-to-end network delay emulator,* Texas A&M University, 1998.

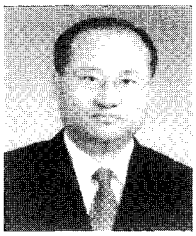
[20] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of TCP congestion avoidance algorithm," *ACM Computer Communication Review,* 27(3):67-82, July 1997.

[21] M. Jacek, "Introduction to artificial neural systems," West *Publishing Company,* 1992.



유 성 구

1979년 8월 9일생. 2003년 전북대학교 제어계측과(공학사). 2005년 전북대학교 제어계측과(공학석사). 2005년 3월~현재 전북대학교 대학원 제어계측과 박사과정 재학중. 관심분야는 멀티미디어 전송, 전송 프로토콜 개발.



정 길 도

1960년 7월 24일생. 1984년 미국 오레곤 주립대학 기계공학(공학사). 1986년 미국 죠지아공대 기계공학과(공학석사). 1993년 미국 택사스 A&M 대학기계공학과(공학박사). 1993년~1995년 영남대학교 기계공학과 전임강사. 1995년~현재 전북대학교 전자정보공학부 부교수. 관심분야는 시스템규명, 멀티미디어전송, 시간지연 시스템제어, 컴퓨터 네트워크 모델링.