

신경 회로망 예측 모델을 이용한 실시간 멀티미디어 데이터 전송률 제어

정회원 김용석*, 권방현**, 정길도***

A Real-Time Multimedia Data Transmission Rate Control Using Neural Network Prediction Model

Yong-Seok Kim*, Bang-Hyun Kwon**, Kil To Chong*** *Regular Members*

요약

본 논문에서는 멀티미디어 전송 시 QoS(Quality of Service)를 개선하기 위한 유효패킷 전송률을 향상 시키는 방법으로 신경회로망을 이용한 예측 알고리즘을 제안하였다. 신경회로망 모델을 이용하여 왕복지연시간과 패킷손실률을 예측하고 예측된 인자를 이용하여 데이터 전송률을 결정하는 방법이다. 제안한 방법은 과거의 데이터를 기준으로 전송률을 결정하여 전송하는 데이터의 양을 제어하는 기존의 방법보다 향상된 성능을 확보할 수 있게 된다. 제안한 방법의 성능을 확인하기 위하여 실 시스템에 적용하는 실험을 실시하였다. 리눅스 운영 PC를 사용하였으며, UDP 프로토콜을 이용하여 실시간 데이터를 전송하는 실험 장치를 구현하였다. 제안한 방법의 유효패킷 전송률이 기존의 TCP-Friendly 혼잡제어 방법에 비하여 5% 정도 향상된 성능을 보였다.

Key Words : Multimedia Data Transmission; Neural Network; TCP; UDP

ABSTRACT

This paper proposes a neural network prediction model to improve the valid packet transmission rate for the QoS(Quality of Service) of multimedia transmission. The Round Trip Time(RTT) and Packet Loss Rate(PLR) are predicted using a neural network and then the transmission rate is decided based on the predicted RTT and the PLR. The suggested method will improve the transmission rate since it uses the rate control factors corresponding to time of data is being transmitted, while the conventional one uses the transmission rate determined based on the past informations. An experimental set-up has been established using a Linux PC system, and the multimedia data are transmitted using UDP protocol in real time. The valid transmitted packets are about 5% higher than the one in the conventional TCP-Friendly congestion control method when the suggested algorithm was applied.

I. 서론

현재 인터넷을 이용한 데이터의 송수신에는 웹 서버로부터의 문자 정보뿐만 아니라 오디오나 비디

오 등 멀티미디어 정보가 지속적으로 증가하고 있다. 인터넷을 통한 멀티미디어 서비스는 이제 주요한 어플리케이션으로 자리 잡아 가고 있으며 실시간 방송에 대한 수요뿐만 아니라 영상 전화 영상

* 삼성전자 Mobile R&D(yongseok528.kim@samsung.com), ** 전북대학교 네트워크 시스템 제어연구실(iam720@chonbuk.ac.kr)
*** 전북대학교 전자정보공학부 부교수(kitchong@chonbuk.ac.kr)
논문번호 : KICS2004-07-101, 접수일자 : 2004년 7월 16일

회의 형태의 대화용 영상 서비스에 대한 수요도 증가하고 있다.

멀티미디어 서비스에서 핵심 미디어인 비디오의 특징으로는 실시간성과 많은 정보량을 들 수 있다 인터넷을 이용한 실시간 비디오 전송은 채널의 충분한 대역폭, 적은 지연시간, 그리고 적은 패킷 손실 등의 조건이 보장되어야 한다^[1]. 그러나 현재의 인터넷은 비디오 전송에 필요한 QoS(Quality of Service)를 만족시키기 위한 네트워크 계층(Network layer)의 특별한 기능을 제공하지 못하고 있다 따라서 서비스의 품질 보장이 네트워크 계층 상위에서 수행되어야만 한다. 이를 위해 제안된 방법 중 하나가 TCP-friendly 알고리즘이다^{[2][3][4][5]}. TCP-Friendly 알고리즘은 UDP 프로토콜을 이용하면서 TCP의 혼잡 제어를 가능하게 한 방법이다

현재 멀티미디어 전송 시 체증(Congestion)을 줄이기 위해 단대단(End-to-End) 포인터에서 네트워크 특성들이 연구되고 있으며^[6], 실시간으로 네트워크의 시간지연 특성을 분석하여 하나의 컴퓨터에서 단-대-단 시간지연특성을 시뮬레이션 할 수 있는 ENDE^[7]와 같은 유용한 시뮬레이터도 개발되었다. 그리고 NS-2^{[8][9]}와 같은 네트워크 시뮬레이터도 개발되어 많은 연구에 이용되고 있다 이러한 방법들을 바탕으로 실시간 전송률 적응 방법과 손실률 예방에 대한 연구가 진행된 바 있다^{[10][11]}.

본 논문에서는 TCP-Friendly 알고리즘을 이용한 시스템을 구현하고 실제 네트워크 상에서의 실험을 통하여 멀티미디어 전송에서 중요한 파라미터들인 왕복지연시간(RTT : Round Trip Time)과 패킷 손실률(PLR: Packet Loss Rate)의 변화에 따른 전송률의 변화를 살펴보았다. 또한 TCP-Friendly 혼잡 제어 알고리즘과 신경 회로망 예측 알고리즘^{[2][3][4][11][5][12]}을 이용한 실시간 멀티미디어 전송률 예측 알고리즘을 제안하였다 제안된 신경회로망 예측 모델은 TCP-Friendly 알고리즘을 통해 얻은 데이터를 이용하여 학습하였고 검증과정을 통하여 모델의 타당성을 확인하였다. 이러한 학습과 검증 과정 후에 신경회로망 예측 모델을 실제 시스템에 적용하는 실험을 실시하였다. 라운드 단위로 얻어지는 왕복지연시간과 패킷 손실률을 입력으로 하는 예측 모델을 통하여, 한 단계 후의 왕복지연시간과 패킷 손실률을 실시간으로 예측하고 한 라운드 후의 네트워크의 상황이 고려된 전송률을 예측하였다 네트워크의 효율적인 체증 제어를 위하여 시간대별 요일별 실험을 실시하였고, 제안된 모델을 이용한 시스템의

효율성을 검증하였다.

II. TCP-Friendly Transmission Rate Control(TFRC)

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 왕복지연 시간(RTT)을 가진 TCP 연결들이 동일한 채널을 공유한다면 그 TCP 연결들은 가용 대역폭을 균등하게 분배하여 갖게 된다. 과거에는 대부분의 트래픽이 TCP 기반 프로토콜을 사용하여 대역폭 분배가 문제되지 않았으나, IP telephony, 영상회의 등의 실시간 응용 서비스와 음성영상 스트리밍 서비스 등 Non-TCP 트래픽의 사용이 증가함에 따라 가용 대역폭의 분배가 중요한 문제로 대두되었다. 그러나 Non-TCP 트래픽은 TCP와 달리 다른 프로토콜과 양립할 수 있는 혼잡 제어 방법이 없기 때문에 혼잡 발생 시 원래의 전송률로 계속 전송하여 전체 트래픽을 가중시킨다. 따라서 Non-TCP 트래픽도 TCP와 양립할 수 있는 전송률 제어 메커니즘을 필요로 하며, 이러한 메커니즘으로써 Non-TCP 트래픽을 TCP에 Friendly하게 만들어 주어 가용 대역폭을 공정하게 분배하는 TCP-Friendly 혼잡 제어 방법을 사용한다.

TCP-Friendly 혼잡 제어 방법은 TCP 모델[2][8]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 식(1)과 같은 형태로 표현된다.

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

그림 1은 일반적인 TCP Reno 모델^{[3][7]}의 동작 원리를 나타낸 것으로서 패킷 손실이 발생할 경우

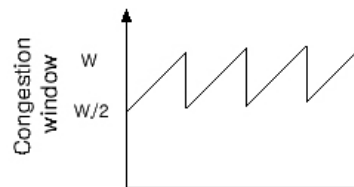


그림 1. TCP Reno 모델의 정상 상태 동작 메커니즘
Fig 1. TCP Reno's congestion window in steady-state

혼잡 윈도우의 크기를 반으로 줄이는 동작을 보여 준다. W 를 혼잡 윈도우 크기, s 를 패킷의 크기로 정의하면 패킷이 손실되기 전까지의 전송률은 $R = \frac{Wk_s}{RTT}$ 이 되고, 패킷을 손실한 후의 윈도우의 크기는 $w/2$ 가 되어 전송률은 $R = 0.5 \times \frac{Wk_s}{RTT}$ 로 된다. 따라서 톱니바퀴 사이클 4개 전구간의 평균 전송률은 $R = 0.75 \times \frac{Wk_s}{RTT}$ 이 된다. 그림1에서 하나의 톱니바퀴 사이클에서의 손실율 PLR 에 대한 식은 $\frac{1}{PLR} = \left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2$ 이며, $W \approx \sqrt{\frac{8}{3PLR}}$ 이 되어 최종적으로 시간 t 에서 유효 전송률 $R(t)$ 는 식(2)과 같이 근사식으로 표현된다⁶⁾.

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

본 논문에서는 식 (2)을 이용한 실험을 통하여 왕복지연시간(RTT)과 패킷 손실률(PLR) 데이터를 수집하였다.

III. 신경회로망

3.1 신경회로망의 구조

본 연구에서 사용된 예측 모델구조는 다층 퍼셉트론(Multi-layer perceptron) 신경회로망으로 입력층(Input layer), 은닉층(Hidden layer) 그리고 출력층(Output layer)의 3개의 층으로 구성되어 있다. 다층퍼셉트론의 출력 \hat{y} 는 식(3)과 같다⁹⁾.

$$\hat{y}_i(t) = g_i[\Phi, \Theta] = F_i \left[\sum_{j=1}^{n_h} w_{i,j} f_j \left(\sum_{l=1}^{n_s} w_{j,l} \phi_l + w_{j,0} \right) + w_{i,0} \right] \quad (3)$$

여기서 ϕ_l 는 입력, Θ 는 신경회로망 구조에서 조정 가능한 모든 매개변수를 포함하는 매개변수벡터이며, $w_{x,y}$, $W_{i,0}$ 은 각각 연결강도와 바이어스를 의미한다. 일반적으로 바이어스는 1을 사용한다. 연결강도를 결정하기 위해서는 출력 \hat{y}_i 와 입력 ϕ_l 의 관계를 내포하는 학습데이터가 필요하며 이 학습데이터를 이용하여 연결강도를 결정하는 과정을 학습이라 부른다.

학습데이터 출력과 신경회로망 출력의 차이를 나타내는 오차함수 E 는 식(4)과 같이 정의된다.

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

여기에서 y_n 는 학습데이터 출력 즉 목표 값이고, o_n 은 신경회로망의 출력을 의미한다.

3.2 LM-BP 학습 방법

가우스 뉴우톤 방법 중의 하나인 LM(Levenberg Marquardt) 알고리즘은 동적으로 최급강화법과 뉴우톤 방법이 가지고 있는 문제를 해결할 수 있다 즉 학습 초기에는 최급강화법을 사용하면서 가중치를 크게 설정하고 어느 정도 학습된 상황에서 수렴 속도가 감소하면 뉴우톤 방법에 가중치를 주어 국부적 최소치로 수렴시킨 후 다시 최급강화법을 사용하여 빠르게 최적해 방향으로 수렴하게 하는 방법이다.

즉, 식 (5)을 사용하여 학습함으로 가중치를 구한다.

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

여기서

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

$$F = \sum_{k=0}^N e_k^2 \text{는 square-sum error, } k \text{는 } k \text{번째 샘플}$$

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

이며 λ 는 동적으로 조절된다.

그러나 실제 LM알고리즘을 사용하는 BP신경회로망(LMBP ; Levenberg Marquardt Back Propagation)에서는 뉴우톤 방법의 H 가 2차 도함수이기 때문에 이것을 1차 도함수로 근사화시켜 사용하는 가우스뉴우톤 방법이 사용된다 즉 뉴우톤 방법에서 H 는 다음과 같이 구할 수 있으며,

$$\begin{aligned} H = [\nabla^2 F(w)]_{ij} &= \frac{\partial^2 F(x)}{\partial w_i \partial w_j} \\ &= 2 \sum_{k=0}^N \left[\frac{\partial e_k(w)}{\partial w_i} \frac{\partial e_k(w)}{\partial w_j} + e_k(w) \frac{\partial^2 e_k(w)}{\partial w_i \partial w_j} \right] \end{aligned} \quad (6)$$

식(6)의 두 번째 항은 무시할 수 있는 항이므로

$$[\nabla^2 F(w)]_{ij} \approx 2 \sum_{k=0}^N \frac{\partial e_k(w)}{\partial w_i} \frac{\partial e_k(w)}{\partial w_j} = 2J^T(w)J(w) \quad (7)$$

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

이러한 근사화를 통하여 2차 도함수의 필요성을 제거할 수 있다. 그리고 식(7)에서의 $\nabla F(w_i)$ 는

$$\nabla F(w_i) = J^T(w_i)e(w_i) \quad (8)$$

으로 정의될 수 있기 때문에 수정된 LMBP 알고리즘은

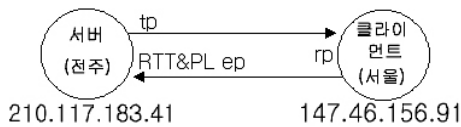
$$w_{m+1} = w_m - [J^T(w_m)J(w_m) + \lambda_m I]^{-1} J^T(w_m)e(w_m) \quad (9)$$

로 최종 정리된다. 식(9)은 m 번째 반복 과정에서 조정되는 가중 파라미터이다.

일반적으로 $\lambda_k = 0.01$ 로 시작되며 만약 Square-some error(SSE)가 충분히 작지 않으면 새로운 $\lambda_k = \lambda_k \cdot \theta$ 로 대체된다. 여기서 θ 는 $1 < \theta < 10$ 의 상수로서 λ 을 증가시키기 위한 일종의 배율기이다.

IV. 데이터 전송 실험

학습과 검증에 사용할 데이터를 수집하기 위하여 그림 2와 같이 실험환경을 구축하였다. 서버와 클라이언트는 리눅스 운영체제를 사용하는 컴퓨터 2대를 사용하였으며, 각각 서울대와 전북대에 설치하였다. ANSI C언어의 소켓프로그램을 사용하여 전송 프로세서(tp), 수신-재전송프로세서(rp), RTT-PLR 측정 프로세서(RTT-PLR ep)를 각각 프로그래밍 하였으며, 전송프로세서와 RTT-PLR 측정 프로세서는 전북대의 서버에, 수신-재전송 프로세서는 서울대의 클라이언트에 설치하였다.



	운영체제	CPU	Memory
Sever	Redhat Linux8.0 (Kernel 2.4.20)	Pentium IV 1.7GHz	512Mbytes
Client	Redhat Linux8.0 (Kernel 2.4.20)	AMD Athlon 1200	512Mbytes

그림 2. 실험 시스템
Fig 2. Experiment set-up

S. N. (tp)	S. N. (rp)	S. N. (ep)	T. S. (tp)	T. S. (rp)	T. S. (ep)	User Data
------------	------------	------------	------------	------------	------------	-----------

S. N. : Sequence Number
T. S. : Time Stamp

그림 3. 프로브헤더
Fig 3. Probe header

전송프로세서는 2절에서 논의한 TCP-Friendly 방법으로 패킷을 전송하게 된다. 패킷 전송 실험은 식(2)을 사용하여 전송하였다. 초기 전송 속도는 100Kb/s이고, 패킷의 크기는 총 625 byte이며, 이중 64 byte는 프로브 헤더에 할당된다. 프로브 헤더는 RTT와 PLR을 측정하기 위해 전송 패킷의 헤더 부분에 삽입되며, 각 프로세서에서 전송된 패킷의 순서를 표시하는 순서번호와 전송된 시각을 저장할 수 있게 구성되어 있다⁶⁾. 그림 3은 프로브 헤더의 구성을 나타낸다. 전송프로세서에서 수신재전송 프로세서로 패킷을 전송하게 되면, 패킷을 수신한 수신-재전송프로세서에서는 프로브 헤더를 분리하여 프로브 헤더에 패킷번호와 현재시간을 입력한 후 RTT-PLR 측정 프로세서로 재전송하게 된다. 프로브 헤더를 수신한 RTT-PLR ep에서는 식(10)과 식(11)을 사용하여 RTT와 PLR을 측정한다.

$$RTT(ms) = time(ep) - time(tp) \quad (10)$$

$$PLR(\%) = 1 - \frac{\text{라운드 } i \text{에서 수신한 패킷의 총합}}{\text{라운드 } j \text{에서 수신한 패킷의 총합}} = 1 - \frac{R_i - R_{i-1}}{S_i - S_{i-1}} \times 100 \quad (11)$$

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(TCP-Friendly Transmission Rate Control) 메커니즘에 직접 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 계산 방법을 따른다. TCP는 저대역 필터를 사용하여 자연스럽게 RTT 측정값과 PLR 측정값을 변경하도록 되어 있다. RTT와 PLR의 측정값은 식(12)

표 1. 트래픽 크기에 따른 RTT와 PLR
Table 1. RTT and PLR according to the traffic size

size	0Mbyte	1Mbyte	2Mbyte	5Mbyte	7Mbyte
Max. RTT(ms)	10.8	17.3	23.9	45.2	79.3
Min. RTT(ms)	10.1	11.5	19.4	29.7	34.9
Max. PLR(%)	1.2	2.4	8.1	12.7	14.4
Min. PLR(%)	0	0.5	3.8	11.3	12.5

과 같은 이동 평균값(moving average)을 이용한다.

$$\begin{aligned}
 RTT^* &= \alpha RTT^* + (1-\alpha) \neq wRTT \\
 PLR^* &= \alpha PLR^* + (1-\alpha) \neq wPLR
 \end{aligned}
 \tag{12}$$

여기서 α 는 권장값 0.9를 사용하였으며, $\neq wRTT$ 와 $\neq wPLR$ 은 새로 측정된 RTT와 PLR이다. 이동 평균값은 트래픽 문제 발생시 RTT와 PLR의 급격한 변화를 완화시켜서 전송률 제어에 사용된다. 전송 실험 시 매 라운드마다 RTT와 PLR을 측정하였으며, 식(12)을 이용하여 moving average RTT와 moving average PLR을 구하였다. 본 연구에서는 RTT, PLR, moving average RTT 그리고 moving average PLR을 각각 학습시켰으며, 예측 결과를 검증하였다.

네트워크의 다양한 상황을 구현하는 방법으로 트래픽 생성기인 IPERF^[11]를 사용하였다. 표 1은 실험을 통하여 수집된 데이터의 RTT와 PLR의 측정값이다. 네트워크에 트래픽을 부과하지 않을 경우 평균 RTT는 약 10.3ms 정도이며, 평균 PLR은 약 0.5% 정도이었다. 트래픽의 부하가 증가할수록 RTT와 PLR은 급격하게 증가하였다. 패킷 전송 실험은 일주일동안 매 시간마다 30분씩 실시하였다.

V. 신경회로망 모델링 및 전송률 제어

왕복지연시간과 패킷 손실률을 예측하는 모델로 신경회로망을 이용하였다. 왕복지연시간과 패킷 손실률의 학습 데이터는 2절에서 설명한 TCP Reno의 혼잡 회피 알고리즘을 응용한 TCP-Friendly 처리율 계산 모델을 실 시스템에 구현하는 전송실험을 통해서 수집하였다.

5.1 학습 데이터

그림 4(a)와 4(b)는 각각 TCP-Friendly 알고리즘

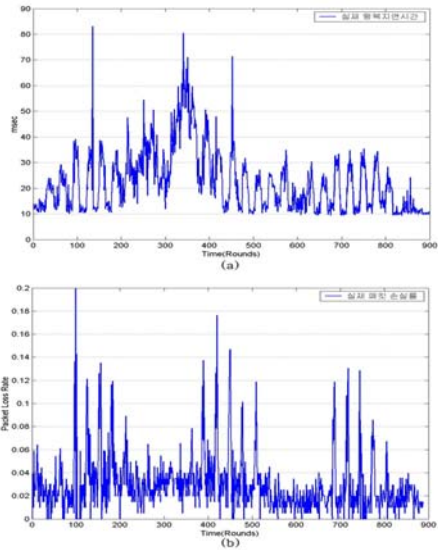


그림 4. 신경회로망 학습데이터
(a) 패킷손실률, (b) 왕복지연시간
Fig 4. Neural Network Training Set
(a) Packet Loss Rate, (b) Round Trip Time

을 적용한 실험을 통해 수집한 왕복지연시간과 패킷 손실률 데이터의 일부분이다. 왕복지연시간은 최소 약 10ms 정도에서 네트워크 혼잡 시 40~80ms 정도까지 증가하고 있음을 볼 수 있으며, 100ms 이상으로 증가하는 경우도 발생한다. 패킷 손실률은 네트워크가 정상상태일 때 0~2% 정도이고, 네트워크가 혼잡할 경우에는 4~5% 정도까지 증가하며, 10~20%까지 증가하는 경우도 발생한다.

5.2 RTT와 PLR 신경회로망 모델링

5.2.1 신경회로망 학습(Training)

학습에 사용된 데이터는 그림 4(a)와 4(b)에서 보인 TCP-Friendly 시스템을 통해 수집한 RTT와 PLR 데이터이며, 원활한 학습을 위해서 0~1사이의 값들로 정규화해서 사용하였다. 전체 데이터의 처음 70%를 학습용으로 사용하였다. 신경회로망 예측 모델 구조는 입력층에 20개의 노드를 은닉층에 8개의 노드를 그리고 출력층에 1개의 노드로 구성하였으며, LMBP 알고리즘을 이용하여 학습하였다. 신경회로망 학습은 수집한 시계열 데이터에서 21개의 데이터를 포함하는 윈도우를 설정하고 윈도우의 처음 20개는 입력으로 21번째 데이터는 출력으로 사용하여 학습을 실시하였다.

그림 5(a)와 5(b)는 각각 batch 방법으로 RTT와 PLR을 1000번 학습한 후의 학습된 데이터 중 200

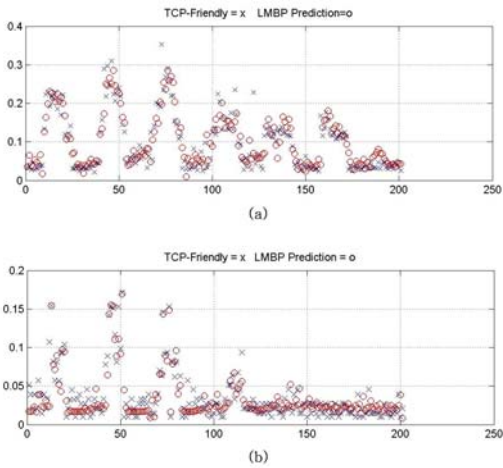


그림 5. 예측 오차; (a) 왕복지연시간, (b) 패킷손실률
Fig 5. Prediction Error; (a) Round Trip Time, (b) Packet Loss Rate

개의 데이터에 대한 예측 결과 이다. 그림에서 ×로 표시된 값들이 TCP-Friendly 알고리즘을 통해 얻은 실제의 데이터들이고 ○로 표시된 데이터가 학습된 신경회로망을 통해 예측된 값들이다. 본 연구를 통해 제안한 신경회로망 예측 모델이 적은 오차범위 안에서 예측함을 알 수 있다

5.2.2 신경회로망 모델 검증(Validation)

그림 6(a)와 6(b)는 각각 RTT와 PLR에 대하여 학습되지 않은 검증 데이터에 대한 신경회로망 예측 알고리즘을 통해 예측한 결과를 나타낸다. 그림에서 ×로 표시된 값들이 TCP-Friendly 알고리즘을

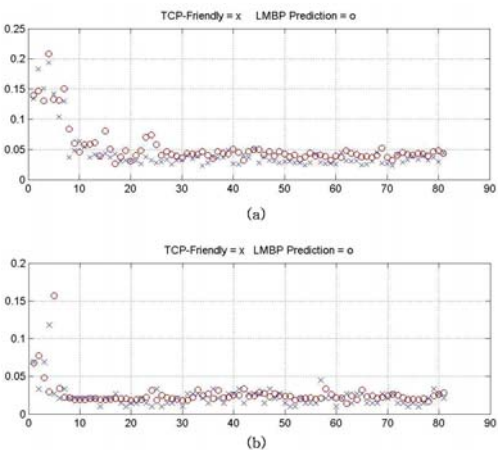


그림 6. 예측 검증과정; (a) RTT, (b) PLR
Fig 6. Validation of Prediction Model; (a) RTT, (b) PLR

통해 얻은 실제의 데이터들이고 ○로 표시된 데이터가 학습된 신경회로망을 통해 예측된 값들이다. 본 연구를 통해 제안한 신경회로망 예측 모델이 적은 오차 범위 안에서 예측함을 알 수 있다. 실 시스템의 실험에서는 전진과정만을 포함하는 신경회로망을 이용하게 된다.

5.3 실 시스템 적용 실험

본 절에서는 5.2 절을 통해서 모델링 한 예측 신경회로망 모델을 이용하여 4장에서 언급한 실 시스템에 적용하는 실험을 실시하였다.

5.4.1 실 시스템의 왕복지연시간

그림 7은 본 연구에서 제안한 신경회로망 모델을 이용하여 예측한 왕복 지연 시간과 실 시스템을 통해 수집한 실제의 왕복 지연 시간을 나타낸 그래프이다. 그래프에서 * 로 표시된 데이터는 실 시스템에서 수집한 데이터이고, . 로 표시된 데이터는 신경회로망을 이용하여 예측한 데이터이다. 실제 시스템의 데이터를 살펴보면 정상 상태에서 10ms 정도의 왕복지연시간을 보이다가 네트워크에 부하가 걸렸을 때 20~40ms 정도의 왕복지연시간이 나타나고 있음을 볼 수 있다. 또한 신경회로망 예측 알고리즘의 예측 결과가 비교적 정확히 네트워크의 부하를 예측하여 왕복 지연 시간을 예측하는 신경회로망 모델의 능력을 확인할 수 있다

5.4.2 신경회로망 모델을 이용한 패킷 손실률 예측

그림 8은 왕복지연시간의 실험 경우와 유사하게, 패킷 손실률에 대한 실험을 실시한 결과의 그래프이다. 그림에서 * 로 표시된 데이터는 실 시스템에

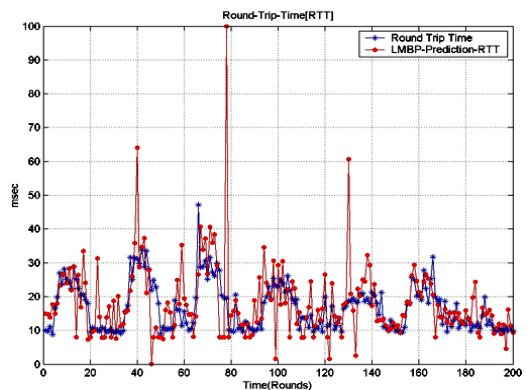


그림 7. 실제 시스템에서의 왕복지연시간의 예측값과 실제값
Fig 7. NN Predicted values and the experimental values of RTT

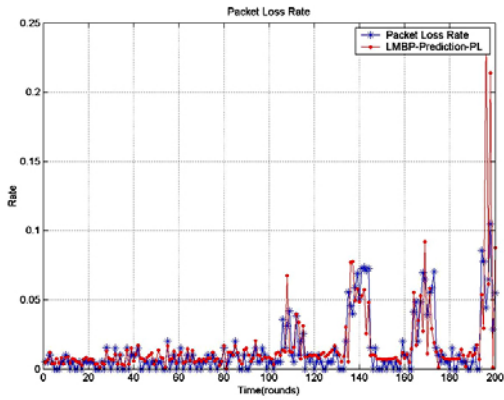


그림 8. 실제 시스템에서의 패킷 손실률 예측값과 실제값
Fig 8. NN Predicted values and the experimental values of PLR

서 수집한 패킷 손실률을 나타내고, .로 표시된 데이터는 신경회로망을 이용하여 예측한 데이터이다. 실제 시스템의 데이터를 살펴보면 정상 상태에서 0~0.01% 정도의 패킷 손실률을 보이고 있고, 네트워크에 부하가 걸렸을 경우 5% 정도의 패킷 손실이 있음을 알 수 있다. 왕복 지연 시간 예측 알고리즘의 경우와 유사하게 신경회로망 예측 알고리즘의 예측 결과가 비교적 정확히 패킷 손실률을 예측하고 있음을 확인할 수 있다

5.4.3 예측 모델을 이용한 시스템의 누적 유효 패킷수

본 절에서는 일정 시간동안 서버에서 전송한 패킷이 클라이언트에 전달된 패킷 즉 유효 패킷수에 대한 실험을 실시하였다. 신경회로망을 이용하여 한 라운드 후의 왕복지연시간과 패킷손실률을 예측하고, 예측된 데이터를 이용하여 패킷 전송률 제어 인자를 결정하고 실 시스템을 이용하는 실험을 실시하였다. 그림 9는 기존의 TCP-Friendly 알고리즘을 이용한 실험 결과와 본 논문에서 제안한 신경회로

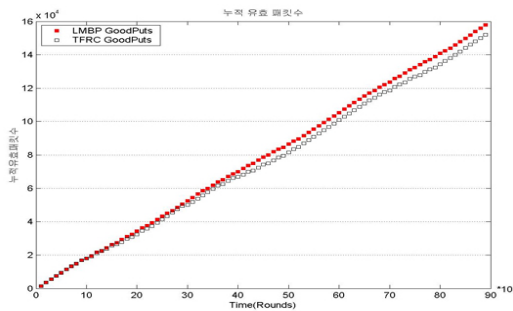


그림 9. 클라이언트 측에서의 누적 유효 패킷수
Fig 9. Accumulated valid packets received in client server

망 예측 알고리즘을 적용한 시스템의 결과를 나타낸 것이다. 제안한 시스템은 네트워크의 부하 상황에 따라 적절하게 전송률을 제어하며, 동일한 시간에 실험한 TCP-Friendly 시스템보다도 많은 유효한 패킷을 보내고 있음을 확인하였다. 본 논문에서 제안한 방법은 기존의 방법보다 5% 정도 성능이 향상되었다.

VI. 결론

인터넷을 이용한 실시간 멀티미디어 전송에 있어서 실시간성과 유효 데이터 전송량은 중요한 요소이며, 채널의 충분한 대역폭, 적은 지연시간, 그리고 적은 패킷 손실 등의 조건에 따라 결정된다. 본 논문에서는 인터넷을 이용한 실시간 멀티미디어 데이터 전송에서 유효 패킷의 향상을 위해 예측 데이터 전송률 제어 시스템을 제안하였다. 예측 알고리즘을 이용하여 한 단계 후의 왕복지연 시간과 패킷 손실률을 예측하고, 예측된 인자를 이용하여 전송률을 결정한다. 제안한 방법은 과거의 데이터를 기준으로 전송률을 결정하고 결정된 전송률에 따라 전송하는 기존의 방식보다 향상된 성능을 확보할 수 있는 방법이다.

제안한 시스템의 성능을 확인하기 위하여 TCP-Friendly 알고리즘을 이용한 실시간 멀티미디어 전송 시스템을 구현하였고, 시스템의 왕복지연시간과 패킷 손실률을 측정하였다. 수집한 데이터를 이용하여 신경회로망 모델링을 실시했으며 검증과정을 통하여 신경회로망 예측 모델의 성능을 확인하였다. 예측 모델을 통해 구한 왕복지연시간과 패킷 손실률을 이용하여 전송률을 결정하였고 이 전송률을 이용하여 실 시스템의 데이터 전송 실험을 실시하였다. 실험을 통해 본 연구에서 제안한 알고리즘이 기존의 TCP Friendly 알고리즘에 비하여 유효전송 패킷의 수를 향상시킴을 확인하였다

참 고 문 헌

[1] A.S Tanenbaum, *Computer Networks(third edition)*, Prentice Hall International, Inc., 1996
[2] V. Jacobson. "Congestion Avoidance and Control" SIGCOMM Symposium on Communications Architectures and Protocols, pages 214-329, 1988

[3] V. Paxson, Automated packet trace analysis of TCP Implementations. IN Proceedings of SIGCOMM 97, 1997

[4] Joerg Widmer, Robert Denda, and Martin Mauve, Parkische Informatic IV, A Survey on TCP-Friendly Congestion Control, IEEE Network, vol. 3, pp.28-37, May/June, 2001

[5] M. Mahdavia and S.Floyd. TCP-friendly unicast rate-based flow control Note sent to end2end-internet mailing list, Jan 1997

[6] V. Paxson, End-To-End Internet packet dynamics. In Proceedings of SIGCOMM 97, 1997

[7] Ik Jun Yeom, "ENDE: An End-To-End Network Delay Emulator", Master Thesis, Texas A&M University, 1998

[8] "http://www.isi.edu/nsnam/ns/ns-build.html" The Network Simulator, version 2

[9] "http://www.isi.edu/nsnam/ns/ucb-tutorial.html"

[10] Q.Zhand, G.Wang, W.Zhu, and Y.Zhang, "Robust scalable video streaming over internet with network-adaptive congestion control and unequal loss protection, "Technical Paper. Microsoft Research, Beijing, China, 2001

[11] W.Zhu, Q.Zhang and Y.-Q. Zhang, "Network-adaptive rate control with unequal loss protection for scalable video over the internet. in Proceeding of IEEE International Symposium on Circuits and Systems 2001, vol. 5, pp. 109-112.

[12] W. S. McCulloch and W. H. Pitts, "A Logical calculus for the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics, vol.5, pp. 115-133, 1943.

[13] F. Rosenblatt, "The perceptron a probabilistic model for information storage and organization in the brain", Psychol. Rev. vol. 65, pp356-408,1958

[14] J. Nie and D.A. Linkens, "Fuzzy-Neural Control : Principles, Algorithms and Applications", pp. 203-220, Prentice Hall, 1995.

[15] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks : Perceptron,

Madaline, and backpropagation", Proceedings of the IEEE, vol. 78 no. 9 pp. 1415-1442, Sept. 1990.

[16] Jacek M. Zurada, "Introduction to Artificial Neural Systems", West Publishing Company, 1992.

[17] D. Wu, Y.T.Hou, W.Zhu, H.J.Lee, T.Chiang, Y.Q.Zhang, and H.J Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," IEEE Trans. Circuits Syst. Video Technology vol. 10, pp 923-941, Sept. 2000

김 용 석(Young-Seok Kim)

정회원



2002년 2월 전북대학교 전자
정보공학부 졸업
2004년 2월 전북대학교 제어
계측공학과 석사
2004년 2월~현재 삼성전자
Mobile R&D 연구원
<관심분야> 인터넷 멀티미디어
데이터 전송

권 방 현(Bang-Hyun Kwon)

정회원



2004년 2월 전북대학교 기계
공학과 졸업
2004년 3월~현재 전북대 제
어계측공학과 석사과정
<관심분야> 인터넷 멀티미디어
데이터 전송, 로봇틱스

정길도(Kil-To Chong)

정회원



1984년 6월 미국 Oregon
State University 기계공학학
사

1986년 12월 미국 Georgia
Institute of Technology 기
계공학석사

1992년 12월 미국 Texas
A&M University 기계공학박사

1993년 9월~1995년 2월 영남대학교 전임강사

1995년 3월~현재 전북대학교 전임강사, 조교수, 부
교수

<관심분야> 통신 네트워크 성능 분석 Time-Delay,
실시간 멀티미디어 전송, Web 기술