ADAPTIVE NEURAL CONGESTION CONTROLLER FOR ATM NETWORK WITH HEAVY TRAFFIC

Ng Hock Soon, N. Sundararajan, and P. Saratchandran
School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore.
ensundara@ntu.edu.sg

Abstract This paper presents an adaptive control scheme using a newly developed Minimal Resource Allocation Network (MRAN) to solve the traffic congestion problem in ATM networks. MRAN generates a minimal radial basis function neural network by adding and pruning hidden neurons based on the input data and is ideal for on-line adaptive control for fast time varying nonlinear systems. The ATM traffic modeling is carried out using the well-known network simulation software OPNET for multiplexed traffic (combining both speech and video signals). Performance of MRAN controller is compared with conventional method and Back-Propagation (BP) neural network controller with the aim of minimizing the congestion episodes and maintaining the quality. Simulation results indicate that MRAN controller performs better than both conventional and BP controller in reducing the congestion and maintaining a better quality of the traffic.

Keywords: Congestion control, Neural networks, ATM

1. INTRODUCTION

Congestion control is a traffic management mechanism to protect the network and the end-system from congestion in order to achieve network performance objectives, while at the same time promoting the efficient use of network resources. Congestion control refers to the set of actions taken by the network to minimize the intensity, spread and duration of congestion. Feedback flow control is one of the solutions that have been extensively studied in the literature [1]. In feedback control schemes, when possible traffic congestion is detected at any network element, feedback signals are sent to all the sources and the traffic submitted to ATM connections is then regulated by modifying the source

D. H. K. Tsang et al. (eds.), *Broadband Communications*

coding rates. Recently, use of artificial neural networks (ANN) in traffic management of ATM networks is gaining momentum [2]-[3]. ANNs have several valuable properties that are quite useful when implementing ATM traffic control. First, ANNs can implement direct adaptive control tailored to the actual characteristics of the cell and /or call traffic. No explicit model of the traffic is needed as in traditional methods. ANNs can learn the relationships between many inputs and outputs and can explicitly consider propagation delay. Second, the parallel structure of ANNs can be exploited in hardware implementations, which provide short and predictable response times.

Recently, an adaptive controller using neural networks for congestion control in ATM multiplexers has been developed [4]. The motivation to use neural networks is to utilize their learning capabilities to adaptively control a non-linear dynamic system without having to define an accurate analytical model of the system. The neural network learns the dynamics of the system from input/output examples. Another motivation is to use the adaptive capabilities of neural networks to handle unpredictable time varying and statistical fluctuations of ATM traffic, which can not be described by theoretical models.

In this scheme [4], the control signal is generated based on the real time measurement of arrival rate process and queuing processes which are indicative of the congestion episodes. This control signal is then fed back to the traffic sources to dynamically modulate the arrival rates by changing the source coding rates. The number of cells waiting in the multiplexer buffer is used as an indicator of congestion. During periods of buffer overloads, the source coding rates will be decreased at the expense of quality, since decreasing the coding rate will decrease the signal to noise (SNR) ratio of the traffic. The sources coding rate for the Adaptive Differential Pulse Code Modulation (ADPCM) scheme considered lie between 4 bits/sample, 3 bits/sample or 2 bits/sample. This involves a trade-off. The control law should try to strike a balance between minimizing the cell loss rate on one hand and maximizing the coding rate on the other hand. To achieve this, a performance index function which consists of two error terms are defined, one the difference between the desired and actual number of cells waiting in the buffer and the second error term which is the difference between the original uncontrolled coding rates of the coders and the controlled rate after applying the control signal. Maximizing the performance index involves in minimizing these two error terms and this is used to adjust the weights of the neural network. The neural network used is the well known back propagation feed forward network and the results indicate that the proposed neural

adaptive control scheme can reduce the congestion in the network in a significant manner.

Recently, a new minimal radial basis function (RBF) neural network called Minimal Resource Allocation Network (MRAN) has been developed by the authors [5], which uses a sequential learning scheme for adding and pruning RBF hidden layer neurons, so as to achieve a minimal network with better approximation accuracy. When no neurons are added or removed, the algorithm uses an Extended Kalman Filter (EKF) to update the centers, widths and weights of each of the hidden neurons. This paper presents the application of MRAN for adaptive congestion control scheme for ATM networks. In comparison to the adaptive controller in [4] where the neural network had a fixed structure i.e. fixed number of neurons and only its weights were adjusted, in the proposed scheme the network builds up the hidden neurons from the input data and it does this in an efficient manner to realize a compact RBF network with better approximation accuracy. Also, instead of adjustments of only the weights as in [4] the proposed MRAN adaptive control scheme provides for adjustments of the centers, widths and also the weights which result in better approximation for the input -output nonlinear functions.

The paper is organized as follows. Section 2 describes the proposed adaptive control scheme using MRAN for congestion control of ATM traffic, which is similar to that of [4]. Section 3 describes briefly MRAN algorithm. Section 4 describes the adaptive neural control for ATM networks under heavy traffic using OPNET while section 5 show MRAN controller performs to clear the heavy congestion in the network. Conclusions from this study are summarized in Section 6.

2. ADAPTIVE CONGESTION CONTROLLER FOR ATM NETWORKS

Figure 1 shows the adaptive congestion control scheme using MRAN and is similar to the scheme in [4] except that the neural controller is based on MRAN instead of BP network. In Fig.1, the controlled source coding rate is defined by the equation:

$$C(k) = C_o u(k) \tag{1}$$

where
C(k) = controlled coding rate at sample k
$C_o(k)$ = maximum uncontrolled coding rate of the source
u(k) = feedback control signal produced by the controller at sample k
n(k+1) = number of cells in the buffer at sample (k+1)
$n_d(k+1)$ = desired number of cells in the buffer at sample (k+1),

$n_d(k+1) \leq n_{max}$ (maximum length of the buffer)
u(k+1) = feedback control signal at the sample (k+1)
$u_d(k+1)$ = desired value of the feedback control signal which is also the maximum value of the feedback control signal: $u(k+1) \leq u_d(k+1)$
z^{-1} represents a unit delay.

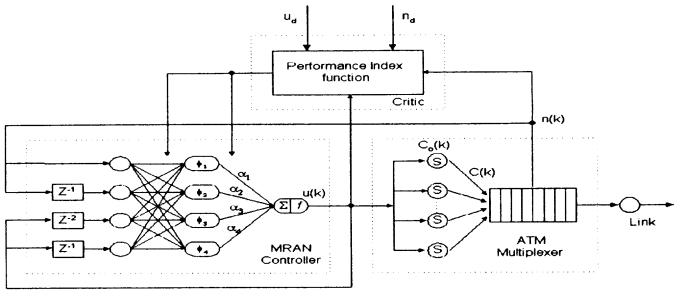

Figure 1 Adaptive Congestion Controller Using MRAN

The congestion control system consists of a critic part and a neural networks controller part. The inputs to the control algorithm are taped delay values of the number of cells in the multiplexer buffer (which is a measure of potential congestion problem) and the taped delay of the feedback control signal. The controller's output is a predicted optimal control signal that is fed back to the input sources to alter their coding rates. This will directly control the traffic arrival rate. During overflow condition, the control signal will reduce the packet arrival rate by decreasing the coding rate of the ADPCM for both bursty and VBR sources. On the other hand, the coding rate is switched back to higher level to maintain the traffic quality.

The critic part involves the performance index of the system (cost function) to be minimized. According to this cost function, the critic part evaluates the system performance and generates an evaluation signal that is a function of the deviation of the system performance from the desired optimal level and is used to change the weights of the neural network controller. Hence, if the control signal is driving the system toward the desired objectives, it is reinforced. Otherwise, the weights are changed to generate a correct control signal. The control signal value will keep updating to minimize two error signals over the measurement period: the difference between the original uncontrolled coding rate of the coders and the controlled rate after applying the control signal; and the difference between the desired and actual number of cell in the buffer. There is a tradeoff between these two objectives, means minimizing the

former will at the same time increase the second error signal term. The performance index function (J) is given as below:

$$J(P) = \sum_{k=1}^{L} R_n S_n(k+1)\varepsilon_n^2(k+1) + R_u\varepsilon_u^2(k+1)) \tag{2}$$

P = trial number
L = length of the measurement period
$S_n(k+1)$ = Reward Signal to reset the control signal as long as the number of cells in the buffer is less than the desired level.
= 0 if $n(k+1) < n_d(k+1)$
= 1 if $n(k+1) \geq n_d(k+1)$
R_n = weight value on the buffer overflow performance index
R_u = weight value the on the the coding rate performance measure.
$\varepsilon_n^2(k+1) = (n_d(k+1) - n(k+1))^2$
$\varepsilon_u^2(k+1) = (u_d(k+1) - u(k+1))^2$
ϵ_u = deviation from voice/video quality from its maximum coding rate.
ϵ_n = cell loss term.

The term ϵ_n represents the cell loss and the term ϵ_u represents the deviation of the traffic quality from its maximum value. Thus, the feedback control signal is determined such that it minimizes both the cell loss rate and the deviation of the traffic quality from its original uncontrolled value.

3. MINIMAL RESOURCE ALLOCATION NETWORK(MRAN)

The MRAN is a minimal Radial Basis Function Neural Network (RBFNN) which is a sequential learning algorithm recently developed by Yingwei et al [5] which combines the growth criterion of RAN with a pruning strategy to realize a minimum RAN. The hidden layer consists of an array of neurons (ϕ_1 to ϕ_n) connected to the output by n connection weights (α_1 to α_n). The output of the hidden layer is the vector $\phi_k(\mathbf{x})$ with m inputs $x(x_1$ to $x_m)$. The second layer of the RBF network is essentially a linear combiner. The overall network response is:

$$f(x) = \alpha_0 + \sum_{k=1}^{K} \alpha_k \phi_k(\mathbf{x}) \tag{3}$$

where $\phi_k(x)$ is the response of the k^{th} hidden neuron to the input x, and α_k is the weight connecting the k^{th} hidden unit to the output unit. α_0 is the bias term. Here, K represents the number of hidden neurons in

the network. $\phi_k(x)$ is a Gaussian function given by,

$$\phi_k(x) = exp(-\|x - \mu_k\|^2/\sigma_k^2) \tag{4}$$

where μ_k is the center and σ_k is the width of the Gaussian function. $\|\ \|$ denotes the Euclidean norm.

In the MRAN algorithm, the network begins with no hidden units. As each input-output training data (x_n, y_n) is received, the network is built up based on certain growth criteria. The algorithm adds hidden units, as well as adjusts the existing network, according to the data received. The criteria that must be met before a new hidden unit is added are :

$$\|x_n - \mu_{nr}\| > \epsilon_n \tag{5}$$

$$e_n = y_n - f(x_n) > e_{min} \tag{6}$$

$$e_{rmsn} = \sqrt{\sum_{i=n-(M-1)}^{n} \frac{e_i^2}{M}} > e_{min1} \tag{7}$$

where μ_{nr} is the center (of the hidden unit) which is closest to x_n, the data that was just received. ϵ_n , e_{min} and e_{min1} are thresholds to be selected appropriately. Equation (5) ensures that the new node to be added is sufficiently far from all the existing nodes. Equation (6) decides if the existing nodes are insufficient to obtain a network output that meets the error specification. Equation (7) checks that the network has not met the required sum squared error specification for the past M outputs of the network. Only when all these criteria are met, is a new hidden node added to the network. Each new hidden unit added to the network will have the following parameters associated with it :$\alpha_{K+1} = e_n$, $\mu_{K+1} = x_n$, $\sigma_{K+1} = \kappa\|x_n - \mu_{nr}\|$.

The overlap of the responses of the hidden units in the input space is determined by κ, the overlap factor. When an input to the network, does not meet the criteria for a new hidden unit to be added, the network parameters $w = [\alpha_0, \alpha_1, \mu_1^T, \sigma_1, ..., \alpha_K, \mu_K^T, \sigma_K]^T$ are adapted using the EKF as follows :

$$w_n = w_{n-1} + e_n k_n \tag{8}$$

where k_n is the Kalman gain vector given by,

$$k_n = [R_n + a_n^T P_{n-1} a_n]^{-1} P_{n-1} a_n \tag{9}$$

where a_n is the gradient vector (for details, see[5]),R_n is the variance of the measurement noise and P_n is the error covariance matrix which is updated by,

$$P_n = [I - k_n a_n^T] P_{n-1} + QI \tag{10}$$

where Q is a scalar that determines the allowed random step in the direction of the gradient vector. If the number of parameters to be adjusted is N, P_n is a $N \times N$ positive definite symmetric matrix.

The algorithm also incorporates a pruning strategy, which is used to prune hidden nodes that do not contribute significantly to the output of the network, or are too close to each other. The former is done by observing the output of each of the hidden nodes for a period of time, and then removing the node that has not been contributing a significant output for that period. Consider the output, o_k of the k^{th} hidden unit :

$$o_k = \alpha_k exp(-\|x - \mu_k\|^2/\sigma_k^2) \tag{11}$$

If α_k or σ_k in the above equation is small, o_k might become small. Also, if $\|x - \mu_k\|$ is large, the output will be small. This would mean that the input is far away from the center of this hidden unit. To reduce inconsistency caused by using the absolute values of the outputs, their values are normalized to that of the highest output. This normalized output of each node is then observed for M consecutive inputs. A node is pruned, if the output of that node falls below a threshold value for M consecutive inputs. The dimensions of the EKF are then reduced to suit the reduced network.

4. OPNET SIMULATION OF ATM WITH HEAVY TRAFFIC

The ATM traffic system is simulated using OPNET Modeler[6]. OPtimized Network Engineering Tools (OPNET) is a comprehensive engineering system capable of simulating communications networks with detailed protocol modeling and performance analysis. OPNET features include graphical specification of models; a dynamic, event-scheduled Simulation Kernel; integrated data analysis tools; and hierarchical, object-based modeling. OPNET's hierarchical modeling structure accommodates special problems such as distributed algorithm development.

The traffic model is shown in Figure 2. There are 3 kind of input sources: bursty, VBR and custom traffic. Two bursty sources with average arrival rate of 29 packets/sec and VBR sources are multiplexed into a FIFO queue. The custom traffic is set to randomly add in some heavy traffic load to overload the network, from there, observations of how the

MRAN and BP controller perform to overcome the congestion under heavy traffic condition can be made. Here, the custom traffic with constant arrival rate of 50 packet/sec was fed into the queue at the period of 50-100 sec, 200-230 sec, 350-400 sec and 500-550 sec. Each bursty source

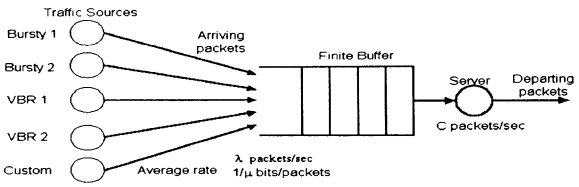

Figure 2 G/D/1/50 Traffic Model

is simulated using ON/OFF binary-state model. In this case, 29 cells are generated during ON period while no cell is generated during the OFF period. Both periods are exponentially distributed random variables with means $1/\beta = 0.35$ sec and $1/\alpha = 0.65$ sec. At the same time, the service capacity is set to 100 packets/sec which will lead to utilization over 100%. Consequently, severe traffic congestion will be occurred. The traffic control scheme will handle this problem by decreasing the source-coding rate. As a result, packet arrival rate will be reduced to avoid the congestion episode. On the other hand, compression made (by reducing the coding rate) will affect the quality of the traffic sources.

This G/D/1/50 queue consists of a first-in-first-out (FIFO) buffer with packets arriving randomly and a server, which retrieves packets from the buffer at a constant service rate. Its performance depends on three parameters: packet arrival rate, packet size, and service capacity. If the combined effect of the average packet arrival rate and the average packet size exceeds the service capacity, the queue size will be fill up immediately. In order to assess the performance of the controller, first the simulation is carried out without any controller and this results in a severe congestion. Figure 3 present the typical simulation results based on OPNET simulation for this case. Figure 3(a) shows the traffic situation without any controller and it can be clearly seen that the traffic condition is heavily congested especially for the period where custom traffic with a constant arrival rate of 50 packets/sec was pumped into the queue. The number of cell in the buffer is concentrated at the top of the capacity of the buffer as shown in Figure 3(b). This leads to a serious congestion problem where as we can observe from Figure 3(c), that overflow occurs more than 15,000 times. Thus, the buffer is said to be severely overloaded. As a result, a very high Cell Loss Rate at about 0.25 occurred all the time according to Figure 3(d). The cost function

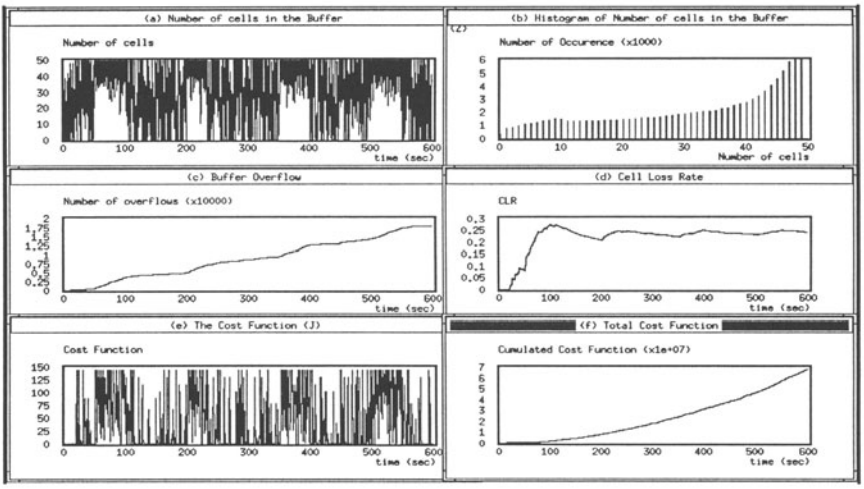

Figure 3 Multiplexed Traffic without Control

and the its accumulated sum are also shown in Figure 3(e) and Figure 3(f) respectively as a deviation measurement from its optimal condition.

5. PERFORMANCE OF MRAN CONGESTION CONTROLLER

After knowing that in the above scenario the congestion problem is severe and noting that it is quite hard to overcome congestion for a dynamically changing multiplexed traffic, the MRAN controller is integrated into the loop and Figure 4 shows the performance of the MRAN controller. In the adaptive traffic control system, MRAN will intelligently adjust the coding rate so that it can optimize the traffic quality and congestion. The time interval rate about 0.01 sec is small enough to obtain the significant changes in the queuing system. The length of the measurement period will affect the sensitivity for the neural network control system to overcome the congestion. However, too frequent updates may result in possible instabilities in the controller. Besides, the weight parameter R_u and R_n give the priority either for achieving good traffic quality or minimized cell loss rate.

The MRAN controller is allowed to operate and its efficiency in removing the congestion is assessed. Figure 4(a) shows the buffer size after applying the MRAN control. As we can observe, there is only a few short period of congestion. Initially, in the case without control, the buffer size is always full which lead to serious traffic congestion. Figure 4(b) shows the buffer occurrence histogram in which the buffer size is

always below the maximum buffer capacity and it is concentrated in the range of 1–30 cells. This can definitely avoid overflow. Figure 4(c) shows the overflow vs. time. Figure 4(d) shows the Cell Loss Rate (CLR) which is the ratio of lost cells to the total number of transmitted cells to the buffer. Again, it is clear that MRAN works to keep the CLR as low as $2.5x10^{-3}$. At the same time, Figure 4(e) shows the cost function which is minimized by the traffic control system. The MRAN control tackles the congestion very well and tries to minimize the cost function immediately by adaptively changing the source-coding rate through feedback control signal. Figure 4(f) is the total cost function cumulated over the time. In the case without control, the cost function will keep increasing exponentially. However, MRAN keeps a control of it although there are a lot of dynamically traffic fluctuation condition.

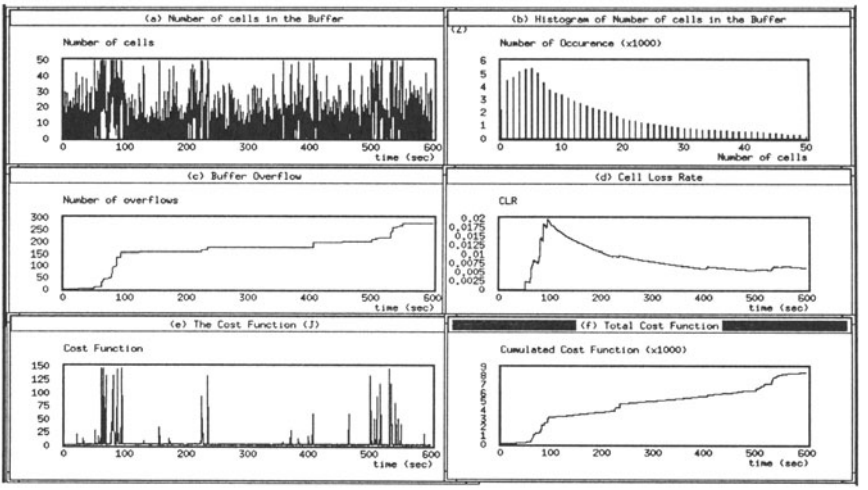

Figure 4 Multiplexed Traffic with MRAN Control

Some performance comparisons between the conventional, BP and MRAN congestion controller have been done. Conventional control which is a modified simple ERICA congestion control scheme is being used to reduce the packet generation rate to the queue when congestion is detected. This conventional control scheme will decrease the source coding rate by a factor of 0.10 during congested period and increased the source coding rate by a factor of 0.01 when the congestion is over. This decreasing rate chosen is 10 times larger than increasing rate, this is to ensure that congestion can be avoided effectively. At the same time, by increasing the traffic with 0.01 any immediate congestion situation that occured can be recovered faster. There are three most important criteria to be highlighted: Traffic Quality, Cell Loss Rate and the total cost

function as a measure of the overall performance of the system. Figure 5 below shows the Cell Loss Rate for the conventional, BP and MRAN controller which have been plotted in the same graph.

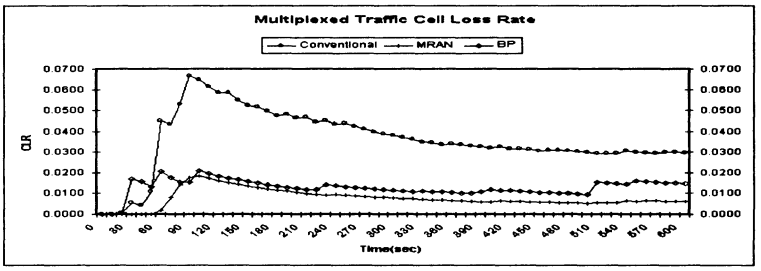

Figure 5 Cell Loss Rate

It is obvious that MRAN does a much better job to reduce and maintain the CLR as low as possible. At the same time, it manages to maintain the traffic quality as shown in Figure 6 although under severe congestion period when the Custom Source sent a large amount of packet for certain durations. On the other hand, the cumulated cost function is used to compare the overall performance of the controllers to do the best optimization job for maintaining the traffic quality and keep the CLR at very low level. This is shown in Figure 6 where MRAN controller is much better than BP along the simulation time. Again, the graph indicates that MRAN tackles the congestion problem faster and more efficiently than conventional or BP.

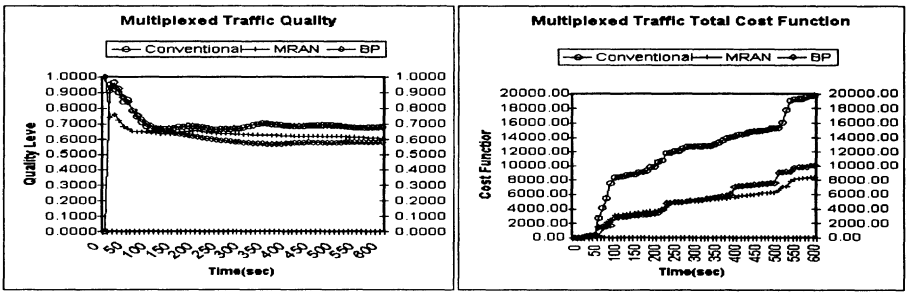

Figure 6 Traffic Quality

From the above simulation results, MRAN performed better where it can reduce CLR four times lower and at the same time maintain the traffic quality 5% higher than BP controller. Furthermore, MRAN can react faster with its most optimized network structure.

6. CONCLUSION

In this paper, three adaptive congestion control schemes using conventional method, BP neural networks and the recently developed MRAN are compared using OPNET simulation of ATM networks with heavy traffic. The neural network controllers generate feedback control signal in accordance to the traffic congestion situation and try to reduce the congestion episodes while maintaining the quality of the traffic. The performance index used as a measure of the traffic performance consists of two parameters with different weights, one concerning the cell loss rate while the other is related to the quality of multiplexed traffic. It is shown that MRAN can adapt and control the system more effectively as compared to conventional or BP controller even under heavy congestion. Based on a detailed comparison based on several simulation studies, it is shown that MRAN controller responds faster and is more efficient than conventional and BP schemes. This is due to the minimal network structure of MRAN which is suitable for fast sequential learning and application to the time-varying nonlinear dynamic system.

References

[1] P.Newman "Backward explicit congestion notification for ATM local area network". *Proc. IEEE GLOBECOM'93*, Houston, TX,1993, pp. 719-723.

[2] "Computational and Artificial Intelligence in High Speed Networks". *IEEE Journal on Selected Areas in Communication*, Vol. 15, No. 2, February, 1997.

[3] C. Douligerisand G. Develokos "Neuro-fuzzy Control in ATM Networks". *IEEE Communication Magazine*,Vol. 35, No. 5, May, 1997, pp. 154-162.

[4] I. Habib, A. Tarraf, and T. Saadawi "A neural network controller for congestion control in ATM multiplexers". *Computer Networks and ISDN Systems*, Vol . 29, No. 3, 1997, pp.325-334.

[5] Y. Lu, N. Sundararajan and P. Saratchandran "A Sequential Learning Scheme for Function Approximation Using Minimal Radial Basis Function Neural Networks". *Neural Computation*,Vol. 9, 1997, pp. 461 - 478.

[6] R. Goyal, R. Jain, S. Fahmy and S. Narayanaswamy "Modeling Traffic Management in ATM Networks with OPNET". *Proc. of OPNETWORK'98*,Washington DC, May 1998.