Neural Comput & Applic (2003) ■■: ■■■–■■■
DOI 10.1007/s00521-003-0374-z

ORIGINAL ARTICLE

Ng Hock Soon · N. Sundararaja · P. Saratchandran

ATM congestion control using Minimal Resource Allocation Networks (MRAN)

Received: ■■■ / Accepted: ■■■
© Springer-Verlag London Limited 2003

Abstract This paper presents a congestion control scheme for ATM traffic using a minimal radial basis function neural network referred to as Minimal Resource Allocation Network (MRAN). Earlier studies have shown that MRAN is well suited for online adaptive control of nonlinear time varying systems as it can adjust its size by adding and pruning the hidden neurons based on the input data. Since ATM traffic is nonlinear and time varying performance of MRAN as a congestion controller is investigated here. These studies are carried out using OPNET to model the ATM traffic. The ATM traffic model consists of bursty, Variable Bit-Rate (VBR) and custom traffic in a multiplexed form so as to generate a heavily congested traffic situation. For this scenario, the controller has to minimize the congestion episodes and maintain the Quality of Service (QoS) requirements. This paper compares the performance of the MRAN congestion controller with that of a modified Explicit Rate Indication with Congestion Avoidance (ERICA) scheme and a Back-Propagation (BP) neural controller. Simulation results indicate that MRAN controller performs better than the modified ERICA and BP controller in reducing the congestion episodes and maintaining the desirable QoS.

Keywords ■

1 Introduction

The Asynchronous Transfer Mode (ATM) has been recommended by the ITU [1, 2] to be the chosen transfer technique for broadband integrated services digital networks (B-ISDN) to support various classes of multimedia traffic with different bit rates, Quality of Service (QoS) requirements, efficient statistical multiplexing and cell switching. In multimedia technology there is a need to put the different services, like voice, video and data on a single infrastructure. Some of these services are always sending bits at a constant rate – Constant Bit Rate (CBR) services – while some send bits in bursts – Variable Bit Rate (VBR). Thus, ATM has to cater for both CBR and VBR traffic, and this brings the time variability of ATM traffic into focus and makes the traffic unpredictable. Due to this unpredictable fluctuations and burstiness of traffic flows within multimedia networks, congestion can occur frequently. Hence, it is important to design an appropriate congestion control mechanism to ensure that the promised QoS are met. Congestion control is a traffic management mechanism to protect the network and the end-system from congestion, to achieve network performance objectives, while at the same time promoting the efficient use of network resources.

Feedback flow control is one of the solutions that have been extensively studied in the literature [3]. 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 coding rates. The general idea is to adjust traffic flow into the network in such a way as to optimise network performance. One well-developed rate control scheme is the Explicit Rate Indication with Congestion Avoidance (ERICA) scheme [8]. Under this control scheme, the source monitors its load and periodically sends control cells that contain the load information. Instead of using queue threshold detection, each switch periodically computes its input load compared with the target utilisation. The switch target utilisation is determined with a margin of safety required to avoid a congestion. The switch on receiving the control cell uses the information provided in the cell and its current load to determine an appropriate rate adjustment factor, which is inserted into the control cell.

In this paper, a modified ERICA congestion control scheme is used as a conventional controller. The

Ng Hock Soon · N. Sundararaja · P. Saratchandran
School of Electrical and Electronic Engineering,
Block S2 Nanyang Technological University,
Singapore, 639735
E-mail: ensundara@ntu.edu.sg

0 0 5 2 1 3 7 4 B | Journal number | Manuscript number | Dispatch: 3.9.2003 | Journal: NCA | No. of pages: 9 | Author's disk received ☑ | Used ☑ | Corrupted ☐ | Mismatch ☐ | Keyed ☐

modifications made in ERICA are mainly in the values chosen for the decrement and increment of coding rates depending on the congestion. This scheme will decrease the source coding rate by a factor of 0.10 during congested period and increase the source coding rate by a factor of 0.01 when the congestion is over. To ensure that congestion can be avoided effectively, the decreasing rate chosen is 10 times larger than increasing rate. At the same time, by increasing the traffic with 0.01 any immediate congestion situation that occurred can be recovered faster. Several simulations have been carried out with different decreasing and increasing factor. It is found that these are the rates appropriate for the traffic scenario considered in this paper.

Recently, use of Artificial Neural Networks (ANN) in the traffic management of ATM networks has gained momentum [4–7, 15]. An adaptive controller using neural networks for congestion control in ATM multiplexers has been developed [9, 14]. The motivation for using neural networks is to utilise their learning capabilities to adaptively control a nonlinear dynamic system (here, an ATM multiplexer) 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 cannot be described by theoretical models.

In the scheme proposed in Habib et al. [9], 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 overload, the source coding rates will be decreased at the expense of quality, since decreasing the coding rate will decrease the Signal to Noise Ratio (SNR) of the traffic. The sources coding rate for the Adaptive Differential Pulse Code Modulation (ADPCM) scheme considered in this paper lie between 4 and 2 bits/sample, which involves a trade-off. The control law should try to strike a balance between minimising the cell loss rate on the one hand, and maximising the coding rate on the other. To achieve this, a performance index function consisting of two error terms are defined, one represents the difference between the desired and the actual number of cells waiting in the buffer, and the second error term represents the difference between the original uncontrolled coding rates of the coders and the controlled rate after applying the control signal. Maximising the performance index involves in minimising these two error terms, and this is used to adjust the weights of the neural network. The neural network used is to adjust the weights of the neural network. The neural network used is the well known backpropagation feedforward 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 [10, 11], which uses a sequential learning scheme for adding and pruning RBF hidden layer neurons, so as to achieve a minimal network with a better approximation accuracy. When no neuron is added or removed, the algorithm uses an Extended Kalman Filter (EKF) to update the centres, widths and weights of each of the hidden neurons. The performance of MRAN has been evaluated on a number of benchmark problems from the function approximation, pattern classification [11].

This paper presents a detailed study on the application of MRAN for adaptive congestion control in ATM networks. To test the MRAN controller's performance, a heavy traffic scenario has been deliberately created using the network simulation package OPNET [12]. The traffic model consists of three types of input sources (bursty, VBR and custom traffic), and their traffic model parameters are selected and multiplexed to result in a heavy traffic scenario. A preliminary study using two bursty and VBR traffic sources, along with a custom traffic source, has indicated the suitability of MRAN for congestion control [13]. In this paper, the performance of MRAN is evaluated for a multiplexed traffic consisting of four bursty and four VBR sources, along with a custom source to create a heavily congested traffic situation. The performance evaluation is carried out using the following figure of merit parameters, *viz*. cumulative distribution of cell loss rate and queuing delay, traffic quality and average link utilisation.

The evaluation is carried out through a comparison of the performance of the MRAN controller for the above traffic scenario with that of the modified ERICA and BP controller [9].

In comparison with the BP adaptive controller in Habib et al. [9], where the neural network had a fixed structure (i.e. fixed number of neurons and only its weights were adjusted), in the MRAN scheme, the network builds up the hidden neurons from the input data. Also, instead of adjustments of only the weights, as in Habib et al. [9], the MRAN adaptive control scheme provides for adjustments of the centres, width and also the weights.

The paper is organised as follows. Section 2 presents the proposed MRAN adaptive congestion control scheme, and Sect. 3 briefly describes the MRAN algorithm. ATM network simulation using OPNET with a heavy traffic scenario is the main topic covered in Sect. 4. Section 5 presents the performance results of MRAN, and also a comparative evaluation of the MRAN congestion controller with the conventional and BP neural controllers. Conclusions from this study are summarized in Sect. 6.

2 MRAN congestion controller for ATM networks

Figure 1 shows the adaptive congestion control scheme using MRAN, and it is similar to the scheme in Habib et al. [9], except that the neural controller is based on MRAN instead of a BP network. The controlled source coding rate is defined by the equation $C(k) = C_o u(k)$.

The definition of the different variables shown in Fig. 1 is given as below:

$C(k) =$ 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)

$u_d(k+1) =$ 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.

The congestion control system consists of a critic part and a neural networks controller part. The inputs to the control algorithm are the tapped delay values of the number of cells in the multiplexer buffer (which is a measure of potential congestion problem) and the tapped delay of the feedback control signal. The controller's output is a predicted control signal that is fed back to the input sources to alter their coding rates. This will directly control the traffic arrival rate. During the 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, when there is no congestion, 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 minimised. 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 towards the desired objectives, it is reinforced. Otherwise, the weights are changed to generate a corrective control signal. The control signal value will be kept updated to minimise the two error signals over the measurement period. The two error signals consist of 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 trade-off between these two objectives, which means that minimising the former will (at the same time) increase the second error signal term. The performance index function (J) is given as

$$
\begin{aligned}
J &= \sum_{k=1}^{L} R_n S_n(k+1)(n_d(k+1) - n(k+1))^2 \\
&\quad + R_u(u_d(k+1) - u(k+1))^2 \\
&= \sum_{k=1}^{L} R_n S_n(k+1)\varepsilon_n^2(k+1) + R_u \varepsilon_u^2(k+1)) \qquad (1)
\end{aligned}
$$

where

$L =$ 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 on the level of the coding rate performance measure.

$\epsilon_u =$ deviation from voice/video quality from its maximum coding rate.

$\epsilon_n =$ 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 minimises both the cell loss rate and the deviation of the traffic quality from its original uncontrolled value.

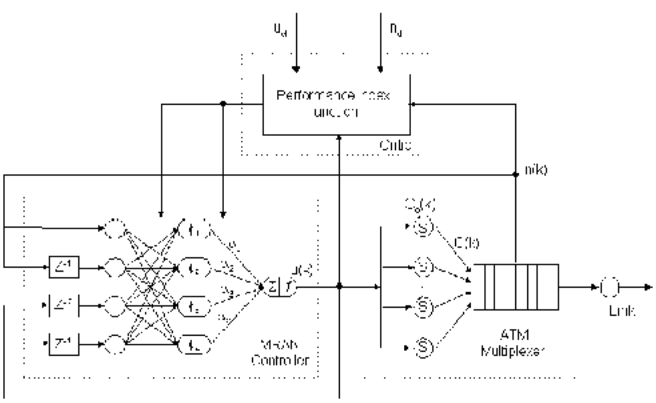

Fig. 1 Adaptive congestion controller using MRAN

3 Minimal Resource Allocation Network (MRAN)

MRAN is a minimal Radial Basis Function Neural Network (RBFNN) recently developed by Yingwei et al. [10, 11].

The output of the MRAN has the form

$$f(x) = \alpha_0 + \sum_{k=1}^{K} \alpha_k \Phi_k(x) \qquad (2)$$

where $\Phi_k(x)$ is the response of the kth hidden neuron to the input x, and α_k is the weight connecting the kth 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.

$$\Phi_k(x) = \exp(-||x - \mu_k||^2/\sigma_k^2) \tag{3}$$

where μ_k is the centre and σ_k is the width of the Gaussian function. || || denotes the Euclidean norm.

In the MRAN algorithm, the network begins with no hidden neuron. 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. A brief outline of the various steps in the MRAN is given below. For details, see Lu et al. [10, 11].

1. Obtain an input and calculate the network output and the corresponding errors e_n, e_{rmsn}.
2. Create a new RBF centre if
   (i) the error e_n exceeds a minimum threshold value,
   (ii) the rms error e_{rmsn} of the network for a series of past data has been above a certain threshold value, and
   (iii) the new input is sufficiently far from the existing centres.
3. If condition (2) is not met, adjust the weights and widths of the existing RBF network using the Kalman Filter Algorithm.

In addition a pruning strategy is adopted:

1. If a centre's normalised contribution to the output for a certain number of consecutive inputs is found to be below a threshold value, that centre is pruned.
2. If two centres are found to be close to each other, as defined by a threshold value, the two centres are combined into a single centre.
3. The dimensions of the EKF are adjusted and the next input is evaluated.

A number of successful applications of MRAN in different areas such as nonlinear system identification, function approximation and time series prediction have been reported [10]. In the present context, some changes have been made to adapt the MRAN for the congestion control scheme. The value of performance index J is used instead of $||y_n - f(x_n)||$ for one of the neuron adding criteria, $||e_i|| > e_{\min}$. Also, the new hidden neuron added will have the following weight parameter: $\alpha_{k+1} = e_n$, with $e_n = (u_d - u_k) + S_n(n_d - n_k)$, where $S_n =$ reward signal.

4 OPNET simulation of MRAN controller under a heavy traffic scenario

4.1 OPNET simulation of heavy traffic scenario

The ATM traffic system is simulated using OPNET Modeler [12]. OPtimized Network Engineering Tools (OPNET) is a comprehensive engineering system capable of simulating communications network with a detailed protocol modelling and performance analysis. Three types of ATM traffic model are simulated in this paper. They consist of Bursty, Variable Bit Rate (VBR) and Custom traffic models. Bursty and VBR traffic represents both audio and video sources. First, a bursty-generator process model sends a burst of certain amount 424-bit (ATM packet = 53-byte) packets to the packet stream. The frequency of the bursts is determined by an attribute of the process model, 'bursts-per-sec'. The generator sends packets to a FIFO queue that can serve at certain service rate (packets/s.). The results show that the bursty traffic generator requires a large FIFO queue. The settings of this traffic model have been configured to emulate the audio traffic sources. Each voice source is simulated using ON/OFF binary-state model. In this case, 29 voice cells are generated during ON period, while no cell is generated during the OFF period. Both periods are exponentially distributed random variables with means $1/\beta = 0.35$ s and $1/\alpha = 0.65$ s. Details of the bursty source are as shown in Table 1.

In the second case, a VBR traffic source process model is designed to emulate a simple Markov chain. The VBR model specifies a variable number of sources described as child processes. These child processes are created according to an exponential distribution. The child processes of a VBR will exist for a certain duration in seconds. All the child processes will generate packets according to an exponential distribution. Each video source is simulated to emulate the first order autoregressive (AR) Markov process. The definition of the AR process is as follows: $X(n) = aX(n-1) + bG(n)$, where $X(n)$ is the bit rate during the nth frame, $G(n)$ is a sequence of independent Gaussian random variables, and a and b are constants. To obtain such a VBR outcome, the OPNET traffic model is configured with parameters as shown in Table 2.

Finally, a Custom traffic model is used, which is a kind of deterministic traffic source that will generate traffic according to a script file included specifically to create a heavy traffic pattern at some intervals. The

Table 1 Bursty source details

Packet size	53 bytes × 8 bits = 424 bits/packet
Burst interarrival	2.86 burst/s = 0.35 ON and 0.65 OFF
Burst size	10 pkts/burst
Average Bit Rate	Burst Interarrival × Burst size × Packet Size = 2.86 burst/s × 10 pkts/ burst × 424 bits/packets = 12126.4 bps (28.6 pkts/s)

Table 2 VBR source details

Packet size	53 bytes × 8 bits = 424 bits/packet
Child processes duration	$0.5\exp^{-0.5x}; E(x) = 2.0, \sigma_x^2 = 4.0$
Child interarrival	$1.54\exp^{-1.54x}; E(x) = 0.65, \sigma_x^2 = 0.4225$
Packet interarrival	$1.0\exp^{-x}; E(x) = 1.0, \sigma_x^2 = 1.0$

Table 3 Custom source details

Start-time	End-time	Generation Rate (pkts/sec)
50.0	100.0	50.0
200.0	230.0	50.0
350.0	400.0	50.0
500.0	550.0	50.0

Custom traffic process is called a scripted generator process, as it reads data from an ASCII file to describe a traffic generation pattern. The data file specifies the generation rate (in packets per second) that will be stochastically generated with an exponential inter-arrival time between a specified simulation time period. The format and details for the traffic generation file is shown in Table 3. The packet generation rate is active between the start and end times. Start and end times must be sequential and must not overlap.

All the above-mentioned traffic types (Bursty, VBR and Custom) will be fed together into an ATM multiplexer, as shown in Fig. 2. The incoming traffic consists of four bursty sources, four VBR sources and one custom source to generate a heavy congestion scenario. This G/D/1/50 ATM multiplexer is modelled as an active, concentrating, packet-oriented First-In First-Out (FIFO) queuing process model, which accepts packets from any number of sources and autonomously forwards them to a single destination module after holding each one for a simulated duration, referred to as the packet's service time. Its performance depends upon three parameters: mean packet arrival rate (λ); mean packet size ($1/\mu$); and service capacity (C). If the combined effect of the average packet arrival rate and the average packet size exceeds the service capacity, the queue size will fill up immediately. For the case of ATM, mean packet size is equal to 424 bits/packet. Enqueued packets wait for the completion of service of the earlier arriving packets before they themselves may begin service and eventually be forwarded. The service capacity is set to 100 packets/s, which will lead to utilisation over 100%. Consequently, severe traffic congestion will occur.

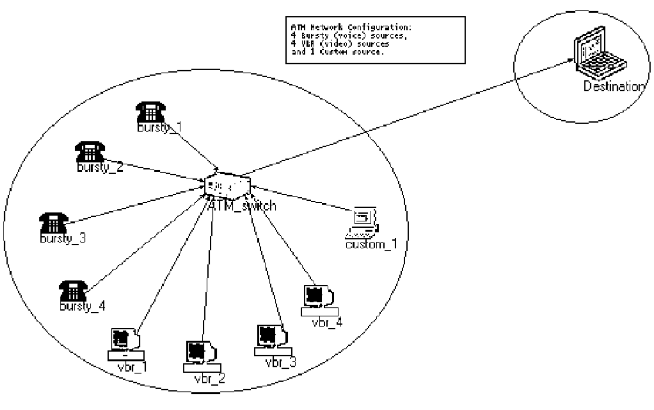

Fig. 2 G/D/1/50 traffic model

The custom traffic is set to randomly add in some heavy traffic loads to overload the network. Here, the custom traffic with constant arrival rate of 50 packet/s was fed into the queue at the period of 50–100 s, 200–230 s, 350–400 s and 500–550 s to make the congestion condition worse. Consequently, severe traffic congestion occurred. The traffic control scheme will handle this problem by decreasing the source-coding rate. As a result, burst size 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.

4.2 MRAN congestion controller's simulation results

Before assessing the performance of the congestion controllers, a simulation is first carried out without any controller to verify the severe congestion. Figure 3 shows the number of cells in the ATM multiplexer versus time. The traffic condition is always congested especially for the period where custom traffic with a constant arrival rate of 50 packets/s was pumped into the queue, as shown in Fig. 3(a). From the heavy traffic scenario discussed above, it may be noted that there is always a blocking of the incoming traffic whenever the buffer is full. This is an important problem in the ATM networks that will affect its Quality of Service (QoS).

After verifying that 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. In the adaptive traffic control system, MRAN will intelligently adjust the coding rate so that it can optimise the traffic quality and congestion. The time interval rate of about 0.01 s is small enough to obtain significant changes in the queuing system. The length of the measurement

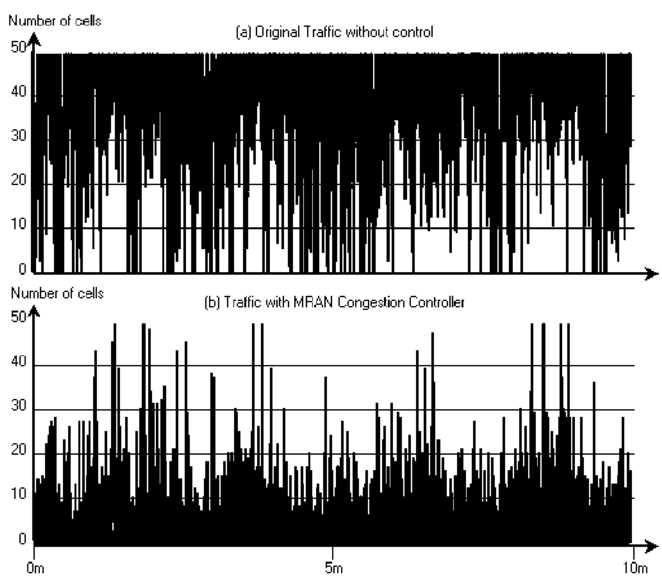

Fig. 3 Number of cells in the ATM multiplexer

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 minimised cell loss rate. The MRAN controller is allowed to operate and its efficiency in removing the congestion is assessed. Figure 3(b) shows the buffer size after applying the MRAN control. As we can observe, there are only a few short periods of congestion.

Figure 4(a) shows the histogram of number of cells in the buffer, which is concentrated at the top of the capacity of the buffer at 50 cells. This leads to a serious congestion problem as shown in Fig. 5(a), where the buffer overflow occurs more than 30,000 times. Thus, the buffer is said to be severely overloaded. Figure 4(b) shows the buffer occurrence histogram after applying MRAN control. The buffer size is always below the maximum buffer capacity, and it is concentrated in the range of 1–30 cells. The distribution is much better as compared with the original case. This can definitely reduce the buffer overflow. Figure 5(b) shows that the occurrence of overflow is being controlled at about 125 times as compared with 30,000 times for the original case.

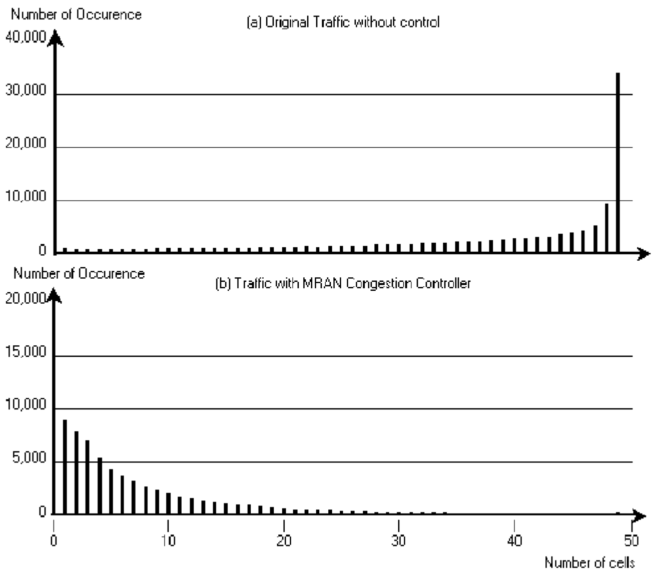

Fig. 4 Histogram of number of cells in the ATM multiplexer

The cost function shown in Fig. 6 indicates the occurrence of performance deviation from its optimal level along the simulation time. The ATM multiplexer without any control is not always working at its optimal level, as shown in Fig. 6(a). Again, the MRAN controller tackles the congestion very well, and tries to minimise the cost function immediately by adaptively changing the source-coding rate through feedback control signal. Figure 6(b) shows that the cost function has been greatly reduced, and most of the congestion has been eliminated.

To observe the performance of MRAN controller, it is worth looking at the evolution of MRAN itself. Figure 7(a) shows the traffic quality after MRAN adaptively adjusts the source-coding rate. Here, the quality is maintained above 70%, although in the case of over 100% utilisation will lead to serious congestion. Figure 7(b) shows the histogram of occurrence of the feedback control signal. Figure 7(c) shows the feedback control signal, which is updated every time interval to overcome the congestion problem. Finally. Fig. 7(d) shows the evolution of MRAN algorithm (i.e. how the neurons build up and get deleted to keep the neural network in a most compact architecture leading to an efficient adaptive controller). This is important, since the

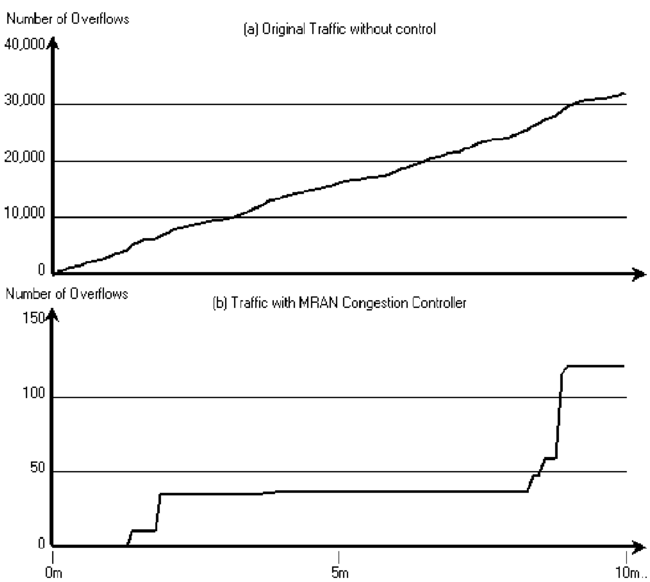

Fig. 5 Buffer overflows

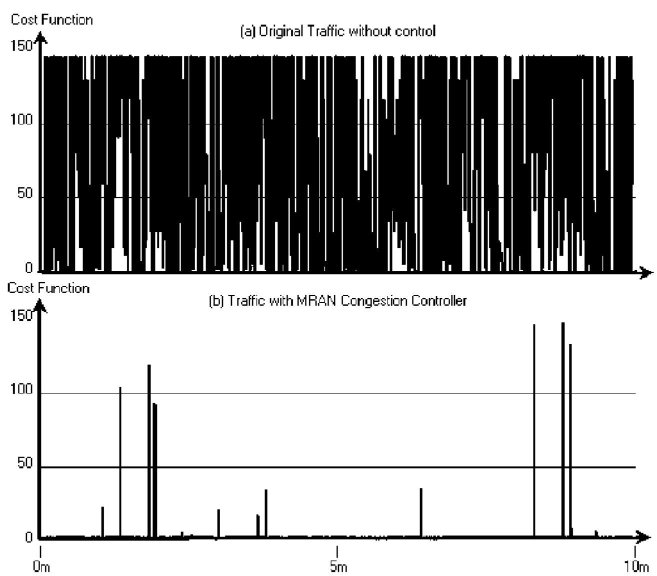

Fig. 6 The ATM cost function

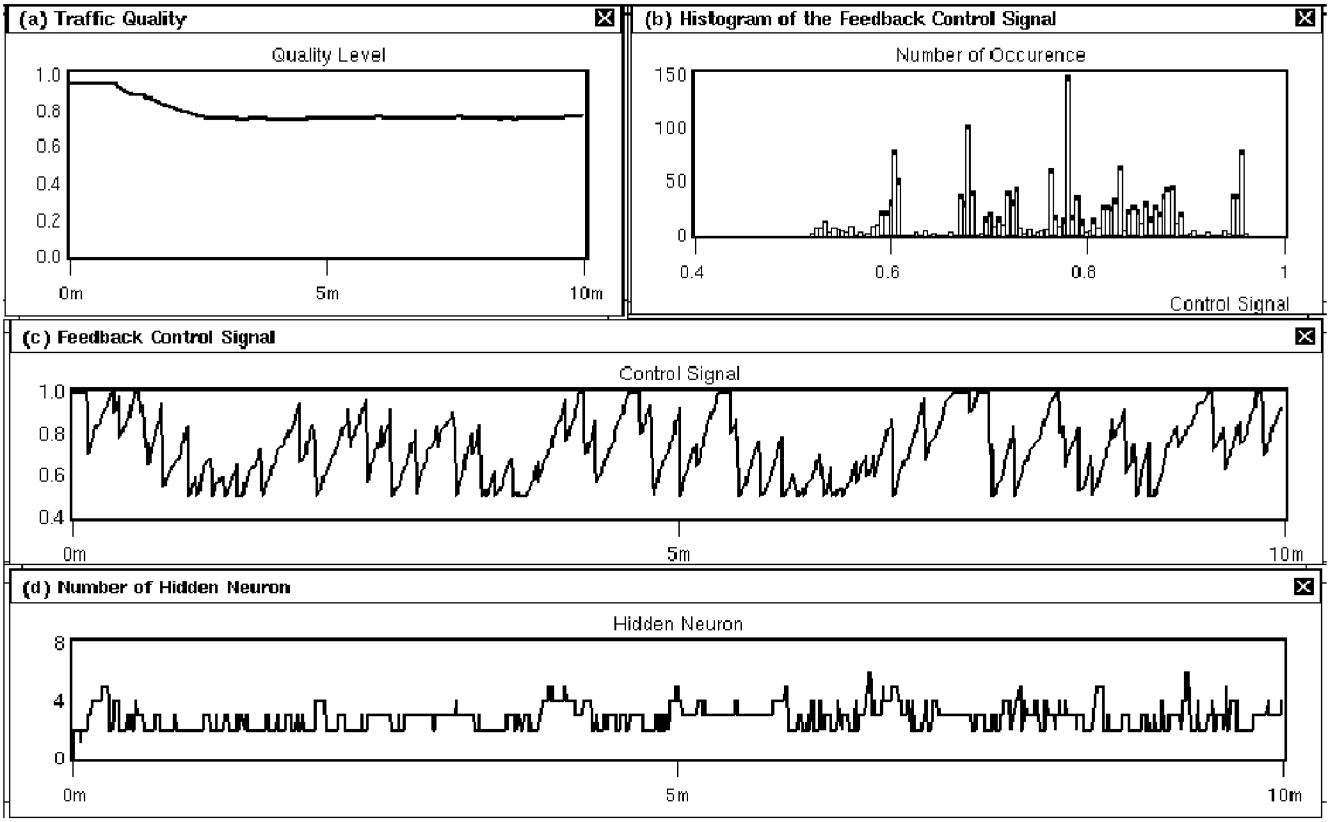

Fig. 7 MRAN network evolution

control scheme has to be integrated into the high-speed communication network where delay must be avoided.

5 Quantitative performance comparison of MRAN, BP and conventional congestion controllers

Section 4 presented typical OPNET simulation results. In this section, we present the performance comparison based on a detailed statistical comparison based on a large number of simulation trials. The performance comparison is done based on the following five important measures/criteria: Cell Loss Rate(CLR), Queuing Delay, Traffic Quality, Link Utilisation; and the total cost function (J). CLR is a QoS parameter that is defined by the ratio of the lost cells to the total number of transmitted cells. Queuing Delay represents the end to end delay the ATM cells to be transferred from its source to its destination over a Virtual Connection (VC). Traffic Quality is taken as the mean value for the traffic source coding rate. Link Utilisation is defined as the percentage of the consumption to date of an available channel bandwidth, where a value of 100.0 would indicate full usage. The total cost function is the accumulated value of the performance index function J over the whole simulation period.

It is interesting to note that all the congestion control schemes work with their best efforts to overcome the severe congestion, especially during the period the custom traffic source is added. The Cumulative Distribution (CDF) of the cell loss rate for MRAN, BP and conventional controllers are shown in Fig. 8(a). From the figure, although all these controllers are trying to overcome the congestion, it is clear that MRAN does the best job by maintaining the lowest CLR, followed by the conventional controller. Here, MRAN keeps the CLR (CDF of 95 percentile) as low as 2.5×10^{-3} while BP controller just manages to reduce it to 9.0×10^{-3}. The conventional controller performs as the poorest one, resulting in a CLR as high as 35×10^{-3}.

Figure 8(b) presents the cumulative distribution of the Queuing Delay. Here also, MRAN performs better with the lowest queuing delay followed by BP and Conventional Controllers. MRAN reacts earlier to maintain the Queuing Delay at about 0.07 s in this context. On the other hand, BP and Conventional controller take more time to settle down. This shows that MRAN can respond to the transient condition faster.

As shown in Fig. 8(c), for the traffic quality. MRAN controller is able to maintain the highest level which is at 76%. Conventional and BP congestion controllers are able to maintain the quality about 65%, 11% lower than the MRAN controller. For the BP controller, the eight-hidden neuron BP works out better for this case, where the quality of service can be maintained quite well while allowing a few short period overflows. However, the improvement is not significant as compared with four or six-hidden neuron network. Figure 8(d) presenting the

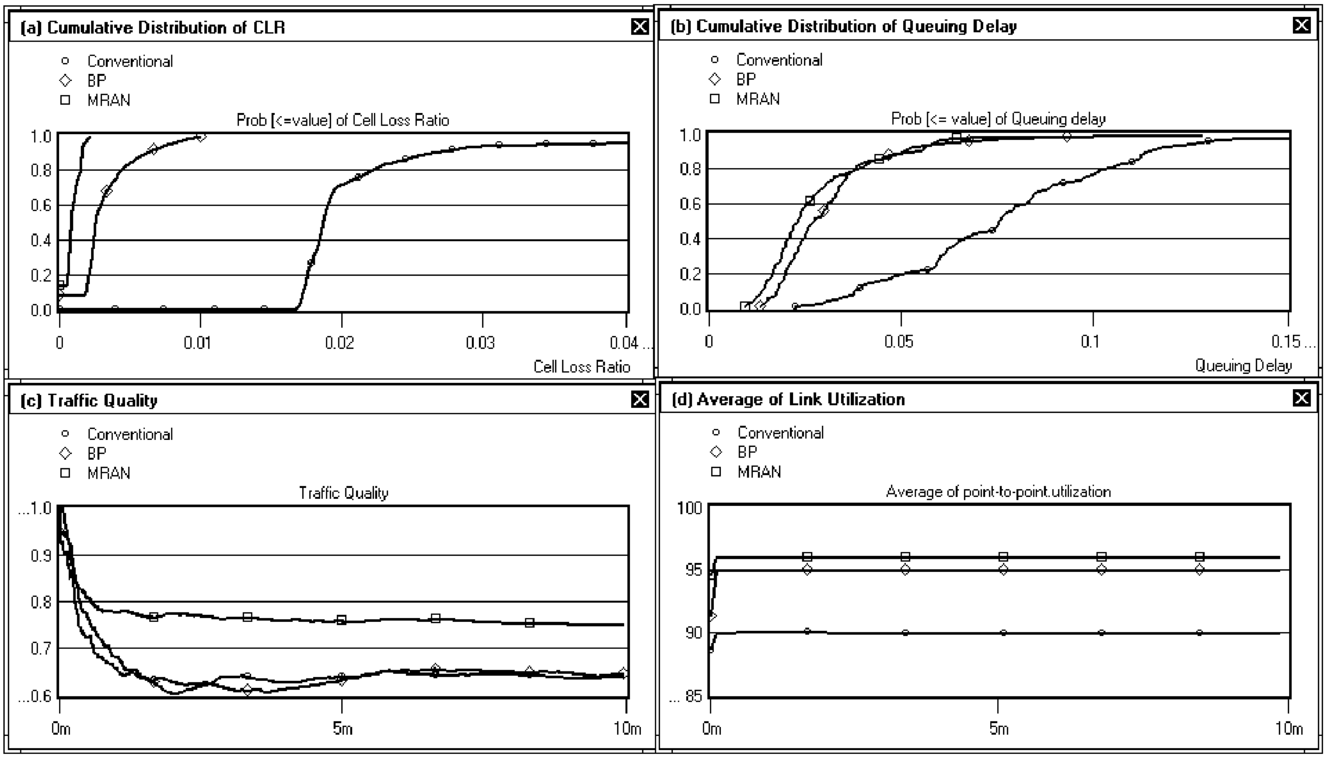

Fig. 8 Multiplexed traffic performance

Average of Link Utilisation shows that MRAN, BP and Conventional controllers all achieved a link utilisation level up to 90%, and there is not much difference among them.

From the performance index point of view, the cumulative cost function is used to compare the overall performance of the controllers, which does the best job for maintaining the traffic quality and keep the CLR at a very low level. This is shown in Fig. 9, where MRAN controller performs much better than BP or the conventional controller during the simulation time. The total accumulated cost function is 8400 for MRAN, 10,000 for BP and 20,000 for the Conventional controller. It is obvious that MRAN does a much better job to reduce and maintain the CLR as low as possible, although there is a lot of dynamic traffic fluctuation conditions. At the same time, it manages to maintain the traffic quality even under severe congestion period when the Custom Traffic source is sending a large amount of packet for certain durations. Furthermore, the queuing delay has been controlled to the lowest level and the link utilisation is at 90%.

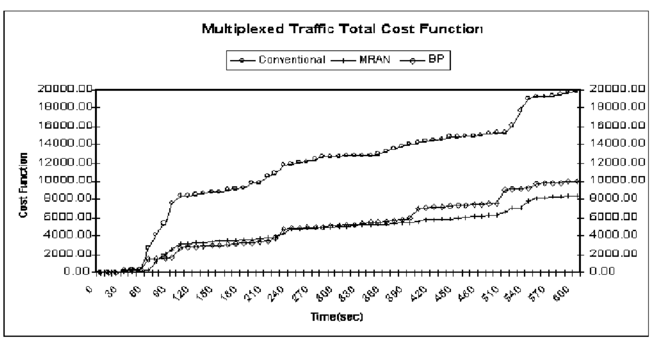

Fig. 9 Multiplexed traffic total cost function.

From the statistical performance figures shown above, MRAN is able to reduce the congestion efficiently. Furthermore, the MRAN controller reacts faster with its most optimised network structure. It is evident that MRAN congestion control scheme can easily handle traffic, which is time varying and unpredictable.

6 Conclusions

In this paper, an adaptive congestion control scheme using MRAN is studied for ATM networks under heavy traffic. For the same traffic, the performance of the modified ERICA and the BP neural congestion controllers is also studied. All these controllers generate feedback control signals in accordance with the traffic congestion situation for reducing the congestion episodes while maintaining the quality of the traffic. Simulation studies using OPNET show that the MRAN controller is able to achieve a significantly lower cell loss rate, and higher traffic quality than the ERICA and BP controllers. MRAN controller is able to respond faster towards traffic changes and maintain the lowest queuing delay. Besides, it uses up the link utilisation efficiently for all the traffic scenarios. Further, in the case of the BP controller, the best structure (i.e. the number of hidden neurons/layers) depends upon the traffic scenario, and has to be fixed *a priori*, whereas MRAN dynamically adjusts its structure to perform well, even when the traffic scenario changes suddenly.

References

1. ITU-T Recommendation I. 37l (1993) Traffic Control and congestion control in B-ISDN. Helsinki
2. ATM Forum Technical Committee Traffic Management Working Group (1996) ATM Forum Traffic Management Specification Version 4.0
3. Liu Y, Douligeris C (1995) Adaptive vs static feedback congestion Controller. Proc IEEE GLOBECOM'95, Singapore, pp 291–295
4. ■ (1995) Neurocomputing in High Speed Networks. IEEE Comm Mag 33(10)
5. Habib I (1996) Applications of neurocomputing in traffic management of ATM networks. Proc IEEE 84(10): 1430–1441
6. Special issue on 'Computational and artificial intelligence in high speed networks'. IEEE J Selected Areas in Comm 15(2) 1997
7. Douligeris C, Develokos G (1997) Neuro-fuzzy control in ATM networks. IEEE Comm Mag 35(5): 154–162
8. Kamolphiwong S, Karbowiak AE, Mehrpour H (1998) Flow control in ATM networks: a survey. Comput Comm 21: 951–968
9. Habib I, Tarraf A, Saadawi T (1997) A neural network controller for congestion control in ATM multiplexers. Comput Networks ISDN Syst 29(3): 325–334
10. Lu Y, Sundararajan N, Saratchandran P (1997) Performance evaluation of a sequential minimal Radial Basis Function (RBF) neural network learning algorithm. IEEE Trans Neural Networks 9(2): 308–318
11. Lu Y, Sundararajan N, Saratchandran P (1997) A sequential learning scheme for function approximation using minimal radial basis function neural networks. Neural Computation 9: 461–478
12. Goyal R, Jain R, Fahmy S, Narayanaswamy S (1998) modelling traffic management in ATM networks with OPNET. Proc OPNETWORK'98, Washington, DC
13. Soon NH, Sundararajan N, Saratchandran (2000) Adaptive neural congestion controller for ATM network with heavy traffic. Broadband Communications – Convergence of Technologies, Kluwer, Boston
14. Yuang MC, Tien PL (2000) Multiple access control with intelligent bandwith allocation for wireless ATM networks. IEEE J Selected Areas Comm 18(9): 1658–1669
15. Al-Hammadi A, Schormans J (1999) Neural network congestion controller in prioritised ATM switch. Electr Lett 34(22): 2097–2098