A QoS-Provisioning Neural Fuzzy Connection Admission Controller for Multimedia High-Speed Networks

Ray-Guang Cheng, Chung-Ju Chang, *Senior Member, IEEE*, and Li-Fong Lin, *Student Member, IEEE*

***Abstract*— This paper proposes a *neural fuzzy* approach for connection admission control (CAC) with QoS guarantee in multimedia high-speed networks. Fuzzy logic systems have been successfully applied to deal with traffic-control-related problems and have provided a robust mathematical framework for dealing with real-world imprecision. However, there is no clear and general technique to map domain knowledge on traffic control onto the parameters of a fuzzy logic system. Neural networks have learning and adaptive capabilities that can be used to construct intelligent computational algorithms for traffic control. However, the knowledge embodied in conventional methods is difficult to incorporate into the design of neural networks. The proposed *neural fuzzy connection admission control* (NFCAC) scheme is an integrated method that combines the linguistic control capabilities of a fuzzy logic controller and the learning abilities of a neural network. It is an intelligent implementation so that it can provide a robust framework to mimic experts' knowledge embodied in existing traffic control techniques and can construct efficient computational algorithms for traffic control. We properly choose input variables and design the rule structure for the NFCAC controller so that it can have robust operation even under dynamic environments. Simulation results show that compared with a conventional effective-bandwidth-based CAC, a fuzzy-logic-based CAC, and a neural-net-based CAC, the proposed NFCAC can achieve superior system utilization, high learning speed, and simple design procedure, while keeping the QoS contract.**

I. Introduction

HIGH-SPEED network supporting multimedia services have to be capable of handling bursty traffic and satisfying various quality-of-service (QoS) and bandwidth requirements. Therefore, a multimedia high-speed network must have an appropriate connection admission control (CAC) scheme not only to guarantee QoS for existing calls but also to achieve high system utilization. As is known asynchronous transfer mode (ATM) is one of the technologies that can integrate multimedia services for high-speed networks.

Conventional CAC schemes [1]–[5] that utilize either capacity estimation or buffer thresholds suffer from some fundamental limitations. One of the limitations is the difficulty of obtaining complete statistics on input traffic to a network. As a result, it is not easy to accurately determine the equivalent capacity or effective thresholds for multimedia high-speed networks in various bursty traffic flow conditions. Besides, these conventional schemes provide optimal solutions only under a steady state. A control scheme that dynamically regulates traffic flows according to changing network conditions, however, requires understanding of network dynamics. The rationale and principles underlying the nature and choice of thresholds or equivalent capacity under dynamic conditions are unclear [6]. Networks are forced to make decisions based on incomplete information [6] so that the decision process is full of uncertainty. Thus, because of unpredictable statistical fluctuations of the system, these control schemes will always be subject to decision error, which degrades performance.

Fuzzy logic systems have been widely employed to deal with CAC-related problems in ATM networks [7]–[9]. Fuzzy set theory appears to provide a robust mathematical framework for dealing with real-world imprecision, and the fuzzy approach exhibits a soft behavior, which means a greater ability to adapt itself to dynamic, imprecise, and bursty environments [7]–[9]. Bonde and Ghosh [7] used fuzzy mathematics to provide a flexible high-performance solution to queue management in ATM networks. Ndousse [9] proposed a fuzzy logic implementation of the leaky bucket mechanism that used a channel utilization feedback to improve performance. In [8], a fuzzy traffic controller which simultaneously incorporates CAC and congestion control was proposed. It is a fuzzy implementation of the two-threshold congestion control method and the equivalent capacity admission control method extensively studied in the literature. Comparative studies have shown that the proposed fuzzy approaches significantly improve system performance compared with conventional approaches. However, no clear and general technique has been presented to map existing knowledge on traffic control onto the design parameters of the fuzzy logic controller. Self-learning capability should be incorporated into the fuzzy logic controller to simplify the design procedure and obtain better control results.

Manuscript received February 1997; revised April 1998; approved by IEEE/ACM Transactions on Networking Editor T. V. Lakshman. This work was supported by the National Science Council, Taiwan, R.O.C. under Contract NSC 87-2213-E009-117. This paper was presented in part at the IEEE INFOCOM'96 Conference, San Francisco, CA, March 1996.

R.-G. Cheng was with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. He is now with the Advanced Technology Center, Computer and Communications Research Laboratories, Industrial Technology Research Institute, ChuTung, Taiwan, R.O.C. (email: crg@atc.ccl.itri.org.tw).

C.-J. Chang and L.-F. Lin are with the Department of Communication Engineering and Center for Telecommunications Research, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. (email: cjchang@cc.nctu.edu.tw; u8513508@cc.nctu.edu.tw.).

Publisher Item Identifier S 1063-6692(99)02271-2.

The self-learning capability of neural networks has been applied to characterize the relationship between input traffic and system performance [10]–[13]. In [11], Hiramatsu used a neural network as a CAC. In [12], Tran-Gia and Gropp investigated the possible use of a neural network to perform CAC. In [13], Youssef, Habib, and Saadawi proposed a call admission controller for ATM networks. A neural network is trained to compute the effective bandwidth required to support MPEG-1 VBR video calls with different QoS requirements. They showed that the adaptability of the neural network controller to new traffic situations had been achieved by adopting a hierarchical approach to the design. However, in most of the proposed neural-net approaches for CAC, the numbers of users for each kind of service were selected as input parameters. The dimension of neural network and the learning time would increase as the number of traffic types grows. The system complexity would increase for system upgrade. Therefore, the application of neural network to CAC is limited to a simplistic traffic environment, such as limited traffic type, simplified traffic source, etc.

Conventional, fuzzy-logic-based, and neural-net-based CAC schemes all have various benefits in handling CAC. Conventional CAC, based on mathematical analysis, provides robust solutions for different kinds of traffic environments but suffers from estimation error (due to modeling) and approximation error (due to the need to complete calculations in real time), so is not suitable for dynamic environments. Fuzzy-logic-based CAC is excellent in dealing with real-world imprecision and has a greater ability to adapt itself to dynamic, imprecise, and bursty environments, but lacks the learning capability needed to automatically construct its rule structure and membership functions so as to achieve optimal performance. Neural-net-based CAC provides learning and adaptation capabilities which reduce the estimation error of conventional CAC and achieve performance similar to that of a fuzzy logic controller. However, the knowledge embodied in conventional methods is difficult to incorporate into the design of a neural network.

This paper proposes a *neural fuzzy connection admission control* (NFCAC) scheme, which absorbs benefits of the three approaches while minimizing their drawbacks, for multimedia high-speed networks. The NFCAC scheme utilizes the learning capability of the neural network to reduce decision errors of conventional CAC policies resulted from modeling, approximation, and unpredictable traffic fluctuations of the system. It also employs the rule structure of the fuzzy logic controller to prevent operating errors, due to incorrect learning, and to decrease training time. Furthermore, the neural fuzzy network is a simple structured network. Here we properly choose input variables and design the rule structure for the NFCAC scheme so that it not only provides a robust framework to mimic experts' knowledge embodied in existing traffic control techniques but also constructs intelligent computational algorithm for traffic control. Simulation results reveal that the NFCAC scheme achieves superior system utilization and high learning speed while keeping the QoS contract, compared with the effective-bandwidth-based CAC (EBCAC) [3], the fuzzy-logic-based CAC (FLCAC) [8], the neural-net-based CAC (NNCAC) [14], and the radial-basis-function-based CAC (RBFCAC) schemes.

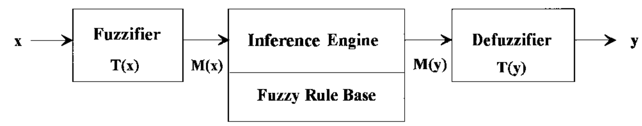


Fig. 1. The basic structure of a fuzzy logic controller.

The rest of this paper is organized as follows. In Section II, the basic concepts behind a neural fuzzy controller are introduced, and an NFCAC scheme is proposed to cope with CAC-related problems in multimedia high-speed networks. Section III presents simulation results comparing the proposed NFCAC scheme with the existing effective bandwidth approach, the fuzzy logic approach, and neural net approach. Finally, some concluding remarks are given in Section IV.

II. NFCAC CONTROLLER

A. Neural Fuzzy Controller

A fuzzy set F in a universe of discourse U is characterized by a membership function which takes values in the interval $[0, 1]$. A linguistic variable x in U is defined by $T(x) = \{T_x^1, T_x^2, \cdots, T_x^k\}$ and $M(x) = \{M_x^1, M_x^2, \cdots, M_x^k\}$, where $T(x)$ is a term set of x, i.e., a set of terms T_x^i with membership function M_x^i defined on U, and $M(x)$ is a semantic rule for associating each term with its meaning.

A *fuzzy logic controller*, as shown in Fig. 1, has three functional blocks: a fuzzifier, a defuzzifier, and an inference engine containing a fuzzy rule base [15]. The fuzzifier is a mapping from an observed m-dim input x_i to fuzzy set $T_{x_i}^{k_i}$ with degree $M_{x_i}^{k_i}$, $i = 1, \cdots, m$. The fuzzy rule base is a control knowledge-base characterized by a set of linguistic statements in the form of "if–then" rules that describe a fuzzy logic relationship between m-dim inputs x_i and n-dim outputs y_j. The inference engine contains the decision-making logic; it acquires the input linguistic terms of $T(x_i)$ from the fuzzifier and uses an inference method to obtain the output linguistic terms of $T(y_j)$. The defuzzifier adopts a defuzzification function to convert $T(y_j)$ into a nonfuzzy value that represents decision y_j. The fuzzy-logic controller can incorporate domain knowledge from existing techniques.

A multilayer feedforward neural network is a layered network that consists of an input layer, an output layer, and at least one hidden layer. Each hidden layer consists of nonlinear processing elements, called nodes. Nodes in two adjacent layers are fully interconnected with variable link weights. The output of a node in one layer multiplied by the link weight becomes the input of a node in the next layer. Each node forms a weighted sum of its inputs and generates an output according to a predefined activation function $a(\cdot)$. Consider a feedforward network $\mathrm{NN}(X, W)$ with input vector X and a set of weight vectors W which will be updated by some learning rules. It needs to train $\mathrm{NN}(X, W)$ (actual output) to approximate a desired output function $g(X)$ as close as possible. The Stone–Weierstrass theorem [16] shows that for any continuous function $g \in C(D)$ with respect to X and a

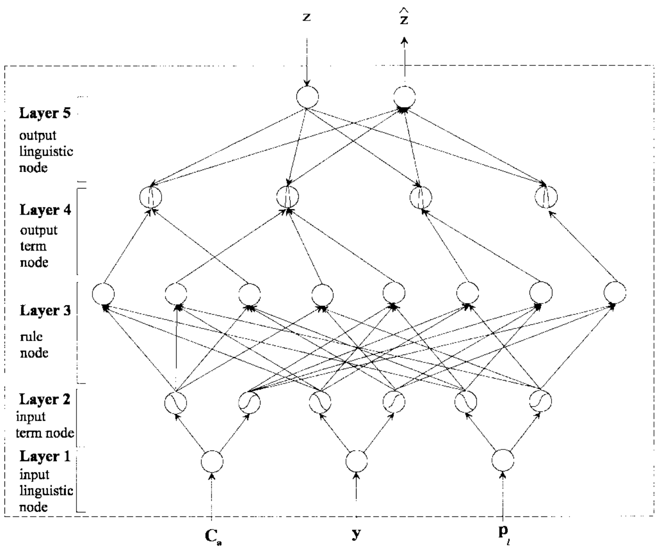


Fig. 2. The architecture of the NFCAC controller.

compact metric space $C(D)$, an $\mathrm{NN}(X, W)$ with appropriate weight W can be found so that $\|\mathrm{NN}(X, W) - g(X)\|_X < \varepsilon$ for an arbitrary $\varepsilon > 0$, where $\|e\|_X = \sum_{X \in D} \|e(X)\|^2$ and $\|\cdot\|$ is a vector norm. The neural network is a nonstructured network, which cannot incorporate knowledge about system.

A neural fuzzy network integrates a fuzzy logic system with a neural network. The integration brings the low-level learning and computational power of the neural network into the fuzzy logic system, and provides the high-level human-like thinking and reasoning of fuzzy logic system for the neural network. The neural fuzzy network generally takes the form of a multilayer network to realize a fuzzy logic system [17]. It is a structured network that can incorporate domain knowledge from conventional policies.

B. NFCAC Controller

We adopt a five-layer neural fuzzy architecture to design the *NFCAC controller*. As shown in Fig. 2, the NFCAC controller has nodes in layer one as input linguistic nodes. It has two kinds of output linguistic nodes used in layer five. One is for feeding training data (desired output) into the net and the other is for pumping decision signals (actual output) out of the net. The nodes in layer two and layer four are term nodes which act as membership functions of the respective linguistic variables. The nodes in layer three are rule nodes; each node represents one fuzzy rule and all nodes form a fuzzy rule base. The links in layer three and layer four function as an inference engine—layer-three links define preconditions of the rule nodes and layer-four links define consequences of the rule nodes. The links in layer two and layer five are fully connected between the linguistic nodes and their corresponding term nodes.

The NFCAC controller adopts three linguistic inputs of an available capacity C_a, a congestion indicator y, and a cell loss ratio p_l and outputs a decision signal $\hat{z}$ to indicate acceptance or rejection of the new call request (shown in Fig. 2). The available capacity C_a is the amount of remaining capacity,

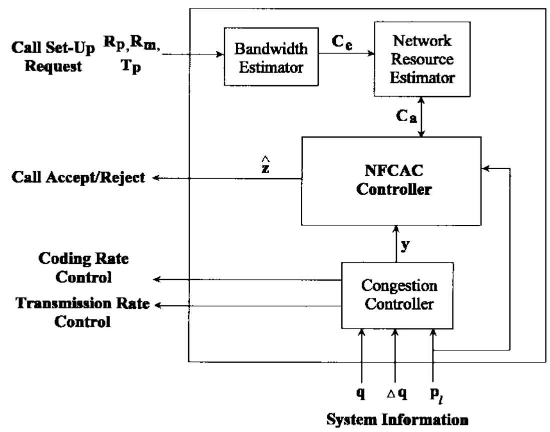


Fig. 3. An NFCAC controller with its peripheral processors.

that is, the total capacity subtracted from those required by the new call and all of the existing calls, and is widely employed in the equivalent-bandwidth-based CAC schemes. The congestion indicator y is the degree of congestion currently in the network that provides more insight information of the system. And the cell loss ratio p_l is the system performance feedback which can be used to provide a closed-loop control system capable of adjusting itself to provide stable and robust operation. In order to generate these input linguistic variables, some peripheral processors are designed for the NFCAC controller.

Fig. 3 shows an NFCAC controller with its peripheral processors for multimedia high-speed networks. The peripheral processors are a congestion controller, a bandwidth estimator, and a network resource estimator. The *congestion controller* generates a congestion indicator y according to the measured system statistics, such as the queue length q, the change rate of the queue length Δq, and the cell loss ratio p_l. Different congestion control algorithms could be employed to implement the congestion controller. For example, rate-based feedback congestion control approaches, which commonly take the queue length and the cell loss ratio into account, could be used. One of the most frequently used congestion control methods is the buffer threshold method, where a congestion alarm occurs whenever the queue length exceeds some predefined thresholds. Here, we adopt a fuzzy congestion controller [8] which is a fuzzy implementation of the two-threshold congestion control scheme proposed in [18]. Network congestion is then averted by regulating the traffic flow of the incoming sources according to the traffic load adjustment parameter generated by the fuzzy congestion controller. The *bandwidth estimator* estimates the required capacity C_e for a new connection from its traffic description parameters such as the peak cell rate, sustainable cell rate, and peak cell rate duration, denoted by R_p, R_m, and T_p, respectively. It employs the equivalent-capacity-based algorithm proposed in the literature. The equivalent capacity method [1, eq. (2)] transforms the traffic characteristics (usually described by three traffic parameters: peak cell rate, sustainable cell rate, and peak cell rate duration) of a new call into a unified metric, called the equivalent bandwidth, to reduce the dependence of the proposed control mechanism on the traffic type. Such a transformation can greatly reduce the number of dimensions of the NFCAC scheme and save a large percentage of learning time. Here, we adopt a fuzzy bandwidth estimator [8], which is a fuzzy implementation of the equivalent capacity method in [1]. The *network resource estimator* does the accounting for system-resource usage. When a new connection with bandwidth C_e is accepted, the value of C_a is updated by subtracting C_e from the original value of C_a. Conversely, when an existing connection with bandwidth C_e is disconnected, the value of C_a is updated by adding C_e to the original value of C_a. C_a is initially set to 1. The *NFCAC controller* takes the available capacity C_a, the congestion indicator y, and the system performance feedback of cell loss ratio p_l as input linguistic variables to handle the CAC procedure and sends a decision signal $\hat{z}$ back to the new connection to indicate acceptance or rejection of the new call request.

In general, the NFCAC controller (shown in Fig. 2) has a net input function $f_i^{(k)}(u_{ij}^{(k)})$ and an activation output function $a_i^{(k)}(f_i^{(k)})$ for node i in layer k, where $u_{ij}^{(k)}$ denotes a possible input to node i in layer k from node j in layer $(k-1)$. The layers are described below.

Layer 1: In this layer, there are three input nodes with respective input linguistic variables C_a, y, and p_l. Define

$$f_i^{(1)}\big(u_{ij}^{(1)}\big) = u_{ii}^{(1)} \quad \text{and} \quad a_i^{(1)} = f_i^{(1)} \tag{1}$$

where $u_{11}^{(1)} = C_a$, $u_{22}^{(1)} = y$, $u_{33}^{(1)} = p_l$, and $1 \le i \le 3$.

Layer 2: The nodes in this layer are used as the fuzzifier. As in the CAC methods in the literature [1], [8], the term used to describe the remaining capacity available for a new connection is either "Enough" or "Not Enough." Thus the term set for the available capacity is defined as $T(C_a) = \mathrm{NotEnough(NE)}, \mathrm{Enough(E)}$. The system is in either a congestion state ("y is Negative") or a congestion-free state ("y is Positive"), so the term set for the traffic load adjustment parameter is defined as $T(y) = \mathrm{Negative(N)}, \mathrm{Positive(P)}$. The term used to describe the cell loss ratio, which is one of the dominant QoS requirements, is either "Satisfied" or "Not Satisfied," and thus the term set for the cell loss ratio is defined as $T(p_l) = \mathrm{Satisfied(S)}, \mathrm{NotSatisfied(NS)}$. In all, we have six nodes in this layer. Each node performs a bell-shaped function, defined as

$$\begin{aligned} f_i^{(2)}\big(u_{ij}^{(2)}\big) &= -\frac{\big(u_{ij}^{(2)} - m_{jn}^{(I)}\big)^2}{\sigma_{jn}^{(I)^2}} \\ a_i^{(2)} &= e^{f_i^{(2)}} \end{aligned} \tag{2}$$

where $u_{ij}^{(2)} = a_j^{(1)}$, $1 \le i \le 6$, $j = \lceil \frac{i+1}{2} \rceil$, and $m_{jn}^{(I)}$ and $\sigma_{jn}^{(I)}$ are the mean and the standard deviation of the nth term of the input linguistic variable from node j in input layer, respectively. $n = 1$ if i is the odd node and $n = 2$ if i is the even node.

Layer 3: The links perform precondition matching of fuzzy control rules. According to fuzzy set theory, the fuzzy rule base forms a fuzzy set with dimensions $|T(C_a)| \times |T(y)| \times |T(p_l)|$ ($|T(x)|$ denotes the number of terms in $T(x)$). Consequently, there are eight rule nodes in this layer. Each rule node performs the fuzzy AND operation defined as

$$\begin{aligned} f_i^{(3)}\big(u_{ij}^{(3)}\big) &= \min\big(u_{ij}^{(3)};\ \forall j \in P_i\big) \\ a_i^{(3)} &= f_i^{(3)} \end{aligned} \tag{3}$$

where $u_{ij}^{(3)} = a_j^{(2)}$ and $P_i = \{j \mid$ all j that are precondition nodes of the i-th rule$\}$, $1 \le i \le 8$.

Layer 4: The nodes in this layer have two operating modes: *down-up* and *up-down*. In the down-up operating mode, the links perform consequence matching of fuzzy control rules. In order to provide a soft admission decision, not only "Accept" (A) and "Reject" (R) but also "Weak Accept" (WA) and "Weak Reject" (WR) are employed to describe the accept/reject decision. Therefore, NFCAC controller may have an alternative choice for calls which fall into the area around the call acceptance/rejection decision boundary. Thus, the term set of the output linguistic variable $\hat{z}$ is defined as $T(\hat{z}) = \mathrm{R, WR, WA, A}$. There are four nodes in this layer. Each node performs a fuzzy OR operation to integrate the fired strength of rules that have the same consequence. Thus, we define

$$\begin{aligned} f_i^{(4)}\big(u_{ij}^{(4)}\big) &= \max\big(u_{ij}^{(4)};\ \forall j \in C_i\big) \\ a_i^{(4)} &= f_i^{(4)} \end{aligned} \tag{4}$$

where $u_{ij}^{(4)} = a_j^{(3)}$ and $C_i = \{j \mid$ all j that have the same consequence of the ith term in the term set of $\hat{z}\}$, $1 \le i \le 4$. The up-down operating mode is used during the training period. The nodes in this layer and the links in layer five have functions similar to those in layer two. Each node performs a bell-shaped function defined as

$$\begin{aligned} f_i^{(4)}\big(u_{ij}^{(4)}\big) &= -\frac{\big(u_{ij}^{(4)} - m_j^{(O)}\big)^2}{\sigma_j^{(O)^2}} \\ a_i^{(4)} &= e^{f_i^{(4)}} \end{aligned} \tag{5}$$

where $u_{ij}^{(4)}$ is set to be $a_j^{(5)}$ obtained from the up-down operating nodes in layer five, and $m_j^{(O)}$ and $\sigma_j^{(O)}$ are the mean and the standard deviation of the j-th term of $\hat{z}$, respectively, $1 \le i \le 4$, $j = 1$.

Layer 5: There are two nodes in this layer. One node performs the down-up operation for the actual decision signal $\hat{z}$. The node and its links act as the defuzzifier. The function used to simulate a center-of-area defuzzification method is approximated by

$$f_i^{(5)}\left(u_{ij}^{(5)}\right) = \sum_{j=1}^{4} m_j^{(O)} \sigma_j^{(O)} u_{ij}^{(5)}$$

$$\text{and} \quad a_i^{(5)} = U\left(\frac{f_i^{(5)}}{\sum_{j=1}^{4} \sigma_j^{(O)} u_{ij}^{(5)}} - z_a\right) \qquad (6)$$

where $u_{ij}^{(5)} = a_j^{(4)}$, $i = 1$, z_a is the decision threshold, and

$$U(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \qquad (7)$$

Clearly, $\hat{z} = a_1^{(5)}$ and a new connection will be accepted only if $\hat{z} = 1$. The other node performs the up-down operation during the training period. It feeds the desired decision signal z into the controller to adjust the link weights optimally. For this kind of node

$$f_i^{(5)}\left(u_{ij}^{(5)}\right) = u_{ii}^{(5)} \quad \text{and} \quad a_i^{(5)} = f_i^{(5)} \qquad (8)$$

where $i = j = 1$ and $u_{11}^{(5)} = z$.

C. Hybrid Learning Algorithm

A hybrid learning algorithm is applied in the design of the NFCAC controller. The algorithm is a two-phase learning method. In phase one, a self-organized learning scheme is used to construct the rules and to locate the initial membership functions. In phase two, a supervised learning scheme is adopted to optimally adjust the membership functions for desired outputs. Training data must be provided for the learning process, in addition to the size of the term set for each input/output linguistic variable and the fuzzy control rules. The procedure for constructing the set of training data is described below.

[*Construction of Training Data:*]
For a new connection request with traffic parameters of R_m, R_p, and T_p
- **Estimate** the required capacity C_e by using fuzzy bandwidth estimator
- **Count** the available capacity C_a by using network resource estimator
- **Generate** a congestion indicator y by using fuzzy congestion controller
- **Get** the cell loss ratio p_l measured from system information statistics
  - If $p_l >$ QoS
  - Then
    - **Reject** the request and set the desired output $z = 0$
  - Else
    - **Accept** the request and set the desired output $z = 1$
    - [*Verification of the Acceptance Decision:*]
    - **Continue** the simulation for a predefined time interval, without accepting any new connection requests
    - **Obtain** the statistics of cell loss ratio p_l'
    - If $p_l' >$ QoS (acceptance decision is failed), then
      - **Set** $z = 0$
    - EndIf
  - EndIf
- **Store** training data of C_a, y, p_l, and z

Using the input training data C_a, y, p_l, the desired output z, the fuzzy partitions $|C_a|$, $|y|$, $|p_l|$, $|\hat{z}|$, and the desired shape of the membership functions, the self-organized training would locate the membership functions and find the fuzzy control rules. If an initial knowledge base is employed to help constructing an initial structure of the fuzzy control rules, a number of possible rule structures can be formed by slight modification of rules. Among all of the possible structures, the one that yields the minimum square error E for the training data is selected. E is defined as

$$E = \frac{1}{2} \sum_{j=1}^{N} [z(t_j) - \hat{z}(t_j)]^2 \qquad (9)$$

where N is the number of training data and $z(t_j)$ and $\hat{z}(t_j)$ are the desired output and the actual output obtained at time t_j, respectively.

If an initial knowledge base is not provided, the initial locations of membership functions are estimated by using Kohonen's self organizing feature-maps algorithm and the *N-nearest-neighbors* scheme [19], and the initial rule structure is constructed via genetic algorithms (GA's) [20].

The procedure to locate the means m_i of the ith membership function for linguistic variable x, $1 \leq i \leq M$, given a set of training data x_j for x, $1 \leq j \leq N$, is described below. It employs the statistical clustering technique of Kohonen's feature-maps algorithm [17].

[*Obtain m_i by using Kohonen's Feature-Maps Algorithm:*]

Step 1: Set initial values of m_i for all membership functions, $1 \leq i \leq M$, such that

$$\min_{1 \leq j \leq N} x_j \leq m_i \leq \max_{1 \leq j \leq N} x_j.$$

Set an initial learning rate α $(0 < \alpha < 1)$.

Step 2: Set $j = 1$.

Step 3: Present training data x_j and compute the distance $d_i = |x_j - m_i|$, $1 \leq i \leq M$.

Step 4: Determine the kth membership function which has the minimum distance d_k $(d_k = \min_{1 \leq i \leq M} d_i)$.
Update m_k by

$$m_k = m_k + \alpha(x_j - m_k).$$

Step 5: If $j < N$, $j = j + 1$, Goto *Step 3*
Else
Decrease α and Goto Step 2.
EndIf

The above procedure will stop until $\alpha \leq 0$. The determination of which d_i is minimum at *Step 4* can quickly be accomplished in constant time via a winner-take-all circuit [17]. The adaptive algorithm can be independently performed to obtain m_i for each input and output linguistic variables.

As for the corresponding standard deviation σ_i of the ith membership function of x, since m_i and σ_i will be finely tuned in the supervised learning phase, we just use a first-nearest-neighbor heuristic to estimate σ_i, which is given by

$$\sigma_i = \frac{|m_i - m^*|}{\gamma} \tag{10}$$

where

$$m^* = \begin{cases} m_{i-1}, & \text{for } |m_i - m_{i-1}| < |m_i - m_{i+1}| \\ m_{i+1}, & \text{otherwise,} \end{cases} \tag{11}$$

and γ is called an overlap parameter used to describe the degree of overlapping for the two membership functions.

GA's are search algorithms based on the mechanics of natural selection and natural genetics [21, pp. 1–22]. They combine the survival of the fittest and some of the innovative flair of human search. According to the fittest values among those randomly selected string structures, a structured but randomized information exchange is defined to form a search algorithm. Although the randomized generating procedure is used, GA's are not simple random walks. They efficiently make use of the historical information to speculate on new search points with expected improved performance [21]. The input/output rule structure is encoded into a gene string $G(t)$ defined as

$$G(t) = [g_1(t), g_2(t), \cdots, g_n(t)] \tag{12}$$

where n is the total number of rule nodes, and $g_i(t)$ $(1 \le i \le n)$ denotes the ith gene in $G(t)$. For example, if the ith rule node in Layer 3 is connected to the jth node $(1 \le j \le |T(\hat{z})|)$ in Layer 4 at time t, then $g_i(t)$ is set to j. Initially, the rules $g_i(0)$ are integers and are randomly assigned within the range of $[1, |T(\hat{z})|]$. $G(t)$ is then updated by genetic operators of *crossover* and *mutation* according to the value of fitness function, which is defined as the inverse of the error E defined in (9). The structure that provides the minimum value of E will be chosen as the optimal structure.

After the self-organized training phase, the NFCAC controller then enters the supervised learning phase. The aim of the supervised learning is to further minimize E for the training data using a back-propagation learning algorithm. Starting at the output node, a backward pass is used to compute $\partial E/\partial w$ for all the hidden nodes in Layer 4 and Layer 2. Assuming that w is an adjustable parameter in a node (i.e., the mean or the standard deviation of the membership function), the general learning rule is

$$w^{\text{new}} = w^{\text{old}} + \eta \frac{\partial E}{\partial w} \tag{13}$$

where η is the learning rate and

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial f}\frac{\partial f}{\partial w} = \frac{\partial E}{\partial a}\frac{\partial a}{\partial f}\frac{\partial f}{\partial w}. \tag{14}$$

f and a were defined in the previous subsection. Here, different values of η could be used in Layer 2 and Layer 4 to provide different learning rates for input and output variables. Different values of η represent different adoption rates for these variables. If the membership function of a specific linguistic variable is not intended to be modified, then $\eta = 0$ is used.

III. Simulation Results

Simulations were performed to test the effectiveness of the proposed NFCAC scheme. Before discussing the results of the simulations, we will first describe the simulation environment.

A. Simulation Environment

Assume that an ATM network is chosen to be the high-speed network supporting multimedia services. The input traffic is categorized into two types: real-time (type-1) and nonreal-time (type-2) traffic. Video and voice services are examples of type-1 traffic, while data services are examples of type-2 traffic. The network system provides two separate finite buffers with size K_i, in order to support different QoS requirements for type-i traffic, $i = 1$ and 2. When the buffer is full, incoming cells are blocked and lost. The system reserves C_r portion of its capacity for type-1 traffic and the remaining $(1 - C_r)$ portion for type-2 traffic. When there is unused type-1 or type-2 capacity, it is used for the other type of traffic. In the simulations described here, $K_1 = K_2 = 100$ cells and $C_r = 0.8$. Also, the QoS requirement for type-1 traffic $QoS_1 = 10^{-5}$ and that for type-2 traffic $QoS_2 = 10^{-6}$.

The cell-generation process for a video coder is assumed to have two motion states: one is the low motion state for the rate of interframe coding and the other is the high motion state for the rate of intraframe coding [22]. The rate of intraframe coding is further divided into two parts: the first part has the same rate as the interframe coding and the second part, called difference coding, is the difference between the rates of intraframe coding and interframe coding. The interframe coding and the difference coding are all modeled as discrete-state Markov-modulated Bernoulli processes (MMBP) with basic rates A_r and A_a. The state-transition diagram is shown in Fig. 4(a) and (b). Let $\lambda_a(t)$, $\lambda_r(t)$, and $\lambda'_a(t)$ denote the cell generation rates for intraframe coding, interframe coding, and difference coding at time t, respectively, from the video coder. Clearly, $\lambda_a(t) = \lambda_r(t) + \lambda'_a(t)$. The process of $\lambda_r(t)$ is an $(M_r + 1)$-state birth–death Markov process. The state-transition diagram for $\lambda_r(t)$ uses the label $m_r A_r$ to indicate the cell generation rate of interframe coding of a state and uses the labels $(M_r - m_r)\gamma$ and $m_r\omega$ to denote the transition probabilities from state $m_r A_r$ to state $(m_r + 1)A_r$ and from state $m_r A_r$ to state $(m_r - 1)A_r$, respectively. Similarly, the process for $\lambda'_a(t)$ is an $(M_a + 1)$-state birth–death Markov process. The state-transition diagram for $\lambda'_a(t)$ uses the label $m_a A_a$ to indicate the additional cell generation rate of a state due to intraframe coding and uses the labels $(M_a - m_a)\phi$ and $m_a\psi$ to denote the transition probability from state $m_a A_a$ to state $(m_a + 1)A_a$ and from state $m_a A_a$ to state $(m_a - 1)A_a$, respectively. One should note that the long-term correlation behavior of a video source is resulted from the process $\lambda_a(t)$. The video source will alternate between interframe and

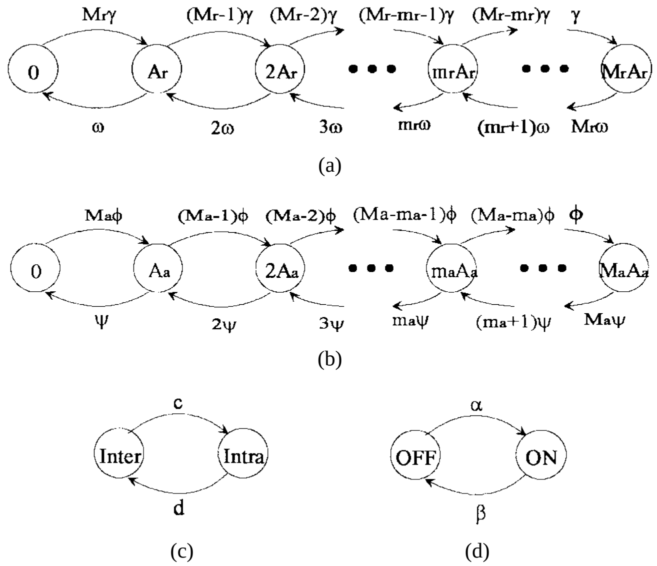


Fig. 4. Level transition diagram for (a) interframe coding $\lambda_r(t)$, (b) difference state $\lambda'_a(t)$, (c) interframe and intraframe alternate model, and (d) voice source.

TABLE I
THE RULE STRUCTURE FOR THE NFCAC

Rule	C_a	y	p_l	$\hat{z}$	Rule	C_a	y	p_l	$\hat{z}$
1	NE	N	NS	R	5	E	N	NS	WR
2	NE	N	S	WR	6	E	N	S	WA
3	NE	P	NS	R	7	E	P	NS	WA
4	NE	P	S	WR	8	E	P	S	A

intraframe, depending on the video source activity factor. As shown in Fig. 4(c), there is a transition rate c in the interframe state and a transition rate d in the intraframe state. The values of γ, ω, M_r, A_r, ϕ, ψ, M_a, A_a, c, and d can be obtained from the traffic variables R_p, R_m, and T_p.

The cell-generation process for a voice call is modeled by an interrupted Bernoulli process (IBP) [18]. As shown in Fig. 4(d), during the ON (talkspurt) state, voice cells are generated with rate A_v; during the OFF (silence) state, no cells are generated. A voice source has a transition rate α in the OFF state and a transition rate β in the ON state.

As for the data source, there are high-bit-rate and low-bit-rate data services. The generation of high-bit-rate and low-bit-rate data cells is characterized by Bernoulli processes with rates θ_1 and θ_2, respectively. Also, the distributions of the holding times for video, voice, high-bit-rate data, and low-bit-rate data are assumed to be exponentially distributed.

In the simulations, for the arrival process of a video source, it is assumed that $R_p = 3.31 \times 10^{-2}$, $R_m = 1.10 \times 10^{-2}$, and $T_p = 0.5$ s, which would give $M_r = M_a = 20$, $A_r = 1.34 \times 10^{-3}$, $A_a = 3.15 \times 10^{-4}$, $\gamma = 3.77 \times 10^{-6}$, $\omega = 5.65 \times 10^{-6}$, $\phi = \psi = 2.83 \times 10^{-5}$, $c = 5.65 \times 10^{-6}$, and $d = 5.09 \times 10^{-5}$; for the arrival process of a voice source, it is assumed that $R_p = 4.71 \times 10^{-4}$, $R_m = 2.12 \times 10^{-4}$, and $T_p = 1.35$ s, which would give $A_v = 4.71 \times 10^{-4}$, $\alpha = 1.71 \times 10^{-6}$, and $\beta = 2.09 \times 10^{-6}$; for high-bit-rate data sources, it is assumed that $R_p = 7.36 \times 10^{-2}$, $R_m = 7.36 \times 10^{-3}$, and $T_p = 3.14 \times 10^{-2}$ s, which would give $\theta_1 = 0.1$, and for low-bit-rate data sources, it is assumed that $R_p = 3.68 \times 10^{-2}$, $R_m = 7.36 \times 10^{-4}$, and $T_p = 2.88 \times 10^{-2}$ s, which give $\theta_2 = 0.02$. The mean holding time is 60 min for a video service, 3 min for a voice service, and 18 s for both high- and low-bit-rate data services. Notice that the values of R_p and R_m have been normalized by the network capacity.

Two kinds of cell loss ratios for type-i traffic are considered: the source loss ratio due to selective discarding at the customer side $p_{s,i}$ and the node loss ratio due to blocking at the network side $p_{n,i}$. The overall cell loss ratio for type-i traffic $p_{l,i}$ is defined as

$$p_{l,i} = \kappa p_{s,i} + p_{n,i}, \quad i = 1, 2 \tag{15}$$

where κ is used to indicate the significance of the node loss ratio over the source loss ratio. $\kappa = 0.8$ is assumed here because selectively discarding cells at the source should have less effect on information retrieval than blocking cells at the node. In the simulations, the cell loss ratio is estimated as the total loss cells divided by the arriving cells during the whole simulation interval.

B. Simulation Results and Discussion

On the basis of prior knowledge concerning CAC, the rule structure and parameters of the NFCAC controller can be initially set and then properly adjusted via the learning algorithm. The membership functions of the linguistic variables for type-1 and type-2 traffic were initially specified in the left-hand side of Fig. 5(a) and 5(b), respectively. As we know, the available capacity C_a, deduced from the equivalent capacity C_e of the existing calls, may possess estimation errors. In order to utilize the network as much as possible, we may employ an idea of "budget deficit" to over-assign the capacity. Thus, the mean value $m_{11}^{(I)}$ of the membership function of *NE* was set to be a negative value and the mean value $m_{12}^{(I)}$ of the membership function of E was set to be a value close to zero.

The behavior of the congestion indicator y could be monitored from the congestion and congestion-free states during a long-term simulation of the network operation. Thus, the membership functions of y could be initially optimized based on the obtained information. The mean value $m_{22}^{(I)}$ of the membership function of P would be set to be the mean value of the queue-length change rate during congestion-free periods, the mean value $m_{21}^{(I)}$ of the membership function of N would be set to be the mean value of the queue-length change rate during congestion periods, and let $\sigma_{21}^{(I)} = \sigma_{22}^{(I)} = m_{22}^{(I)} - m_{21}^{(I)}$. These parameters could be further off-line optimized via GA by simulation.

The initial membership functions of the cell loss ratio p_l were set according to the QoS requirement. The mean value $m_{32}^{(I)}$ of the membership function of *NS* would be set to be the QoS requirement, the mean value $m_{31}^{(I)}$ of the membership function of S would be set to be a fraction of the QoS requirement, and the standard deviations would be set to be $\sigma_{31}^{(I)} = \sigma_{32}^{(I)} = m_{32}^{(I)} - m_{31}^{(I)}$. As a result, there exists a safety margin between the membership functions of terms S and

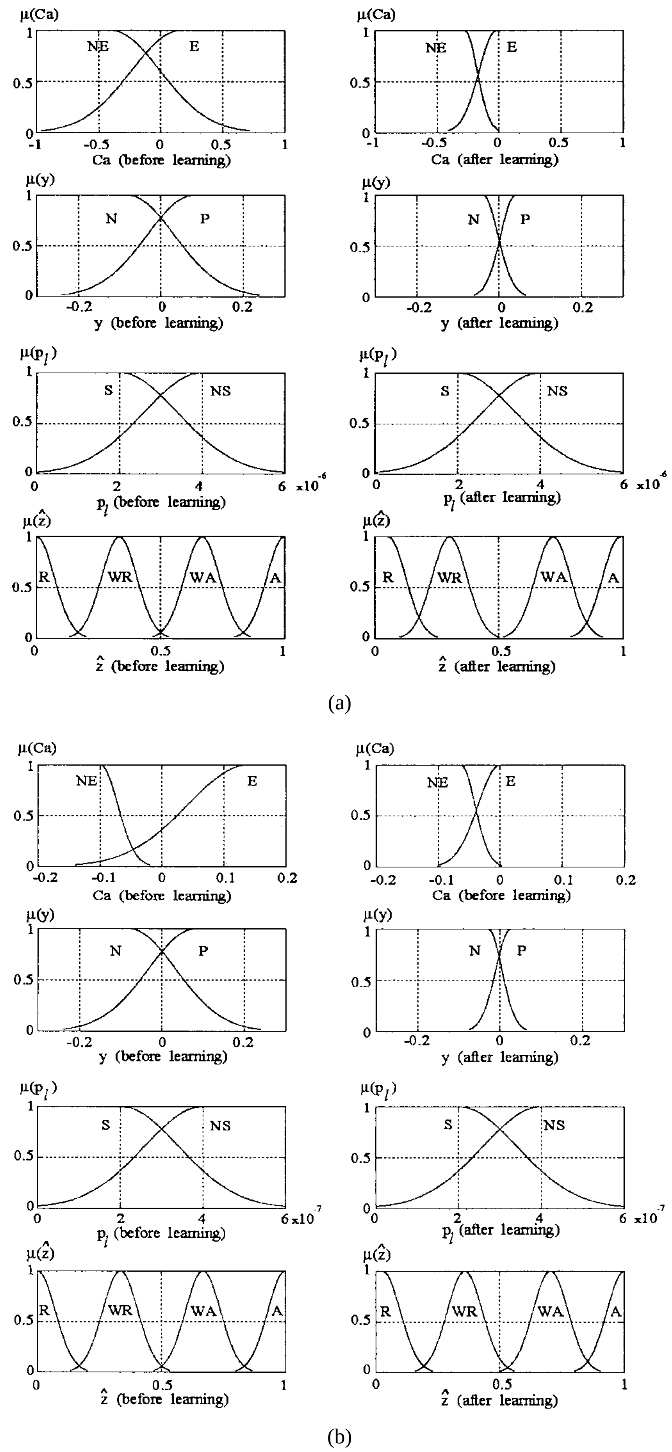


Fig. 5. Membership functions of C_a, y, p_l and $\hat{z}$ for (a) type-1 traffic and (b) type-2 traffic.

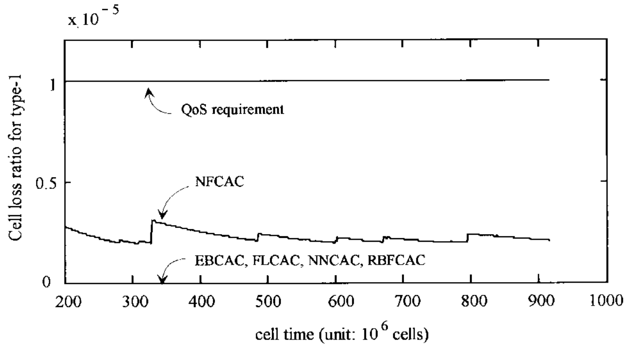
NS provided to tolerate the dynamic behavior of the network operation and insure the QoS requirement.

Here, little information about the setting of initial values for the mean $m_j^{(O)}$ of the term set $T(\hat{z})$ could be employed; therefore, the values of $m_j^{(O)}$ are set to be equally spaced in the range of $[0, 1]$. Based on the initial membership functions, an optimal rule structure shown in Table I was obtained by using GA in the self-organized learning phase. When the fuzzy logic rules were found, the NFCAC controller entered the supervised learning phase, in which the membership functions were adjusted optimally.

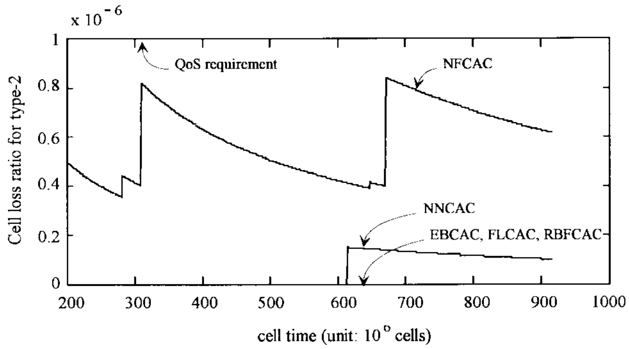
Three different values of η were used for the variables C_a, y, p_l, and $\hat{z}$. η was set to zero for p_l because the membership functions were specified by the QoS constraint and should not be modified. $\eta = 0.001$ was used for y because the membership functions of y were initially optimized. As for C_a and $\hat{z}$, their initial membership functions were heuristically set and required further optimization in the supervised learning phase. Thus, $\eta = 0.01$ was used. The use of different η may drastically reduce the training time required in the supervised learning phase. The learned membership functions of the linguistic variables for type-1 and type-2 traffic were shown in the right-hand side of Fig. 5(a) and Fig. 5(b), respectively.

For type-1 traffic in Fig. 5(a), it can be found that the differences of the membership functions before and after learning are as follows. For the membership functions of C_a, the mean value $m_{11}^{(I)}$ of the membership function of *NE* was properly modified from -0.4 to -0.27. Similarly, the mean value $m_{12}^{(I)}$ of the membership function of E was properly modified from 0.16 to -0.02. There is a drastically change for membership functions of C_a, and the phenomenon can also be found in the membership functions of y. It is because we heuristically set their initial values and we used only two terms to describe C_a or y. The change of the position of one term of C_a and y will squeeze the other term but receive less counteraction from the other one term (compared to $\hat{z}$ described later). Membership functions of p_l are not changed since η for p_l was chosen to be zero. For the membership function of $\hat{z}$, however, the mean $m_1^{(O)}$ of the membership function of R is slightly increased from 0 to 0.05, representing that the effect of "Reject" is decreased. Also, the mean $m_3^{(O)}$ of the membership function of *WA* is slightly increased from 0.67 to 0.72, representing that the effect of "Weak Accept" is increased. The small change is because we used four terms to describe $\hat{z}$. The change of the position of one term of $\hat{z}$ will squeeze the other three terms but receive more counteraction from the three terms. Therefore, the change of position would be confined in a smaller range. The changes of membership functions of $\hat{z}$ imply that the NFCAC controller prefers to accept new calls. This phenomenon demonstrates that the NFCAC controller intends to recover some system bandwidth which the equivalent capacity method wastes due to over-estimation, while keeping the QoS contract. It may be the reason for the utilization improvement of the proposed NFCAC controller, which will be shown below. Similar results could be found for type-2 traffic in Fig. 5(b).

We compare the NFCAC scheme with the effective-bandwidth-based CAC (EBCAC) scheme proposed in [3], the fuzzy-logic-based CAC (FLCAC) scheme proposed in [8], the neural-net-based CAC (NNCAC) scheme proposed in [14], and the radial-basis-function-based CAC (RBFCAC) scheme from the aspects of the cell loss ratio (CLR), the system utilization, and/or the training time under the constraint of QoS guarantee. The EBCAC scheme is a hybrid technique combining the conventional techniques of the Gaussian approximation and the bufferless analysis; it is an improved version of the



(a)



(b)

Fig. 6. Cell loss ratio for (a) type-1 traffic and (b) type-2 traffic.

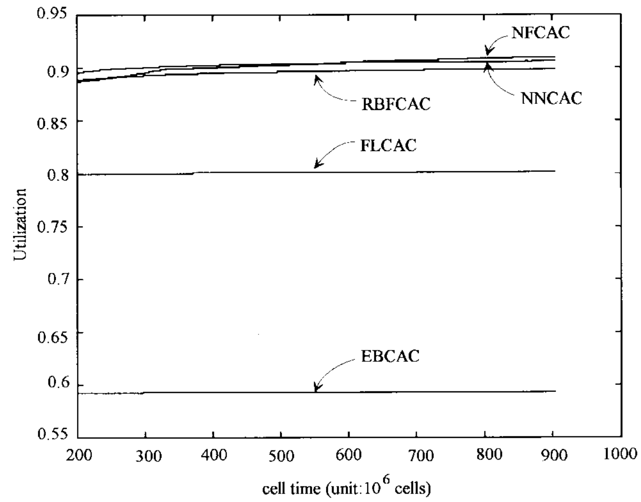
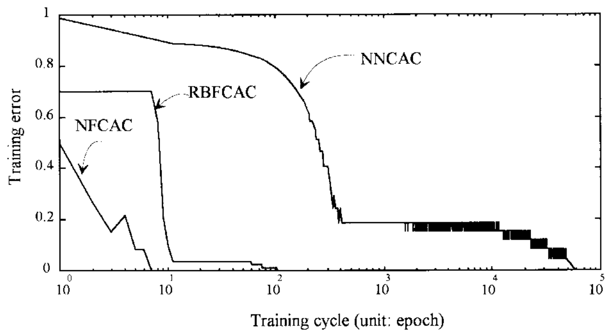


Fig. 7. System utilization.

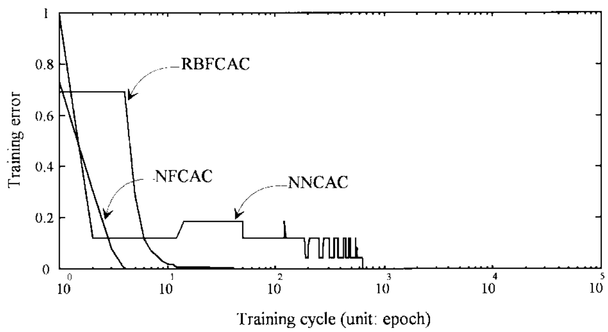
equivalent capacity method [1]. Simulation of the EBCAC scheme is simply to calculate the required bandwidth of a new connection. The new connection request is accepted if the total bandwidth required by the new connection and the existing connection is less than the system capacity. Otherwise, it is rejected. The FLCAC scheme is a fuzzy implementation of the equivalent capacity admission control method; details for the FLCAC scheme can be referred to [8]. The NNCAC and RBFCAC schemes are neural-net implementation of the equivalent capacity admission control method, where the NNCAC adopts the multilayer perceptron (MLP) structure with 30 hidden nodes, while the RBFCAC uses radial basis function network (RBFN) with 30 hidden nodes. Details for the NNCAC scheme can be referred to [14]. In the simulations, the FLCAC, NNCAC, or RBFCAC controller is equipped with the same three peripheral processors as those used in the NFCAC controller shown in Fig. 3. The sizes of training set and test set are all equal to 200, the number of repeated experiments is 20, and the standard deviation is less than 5%.

Fig. 6 shows the CLR's of an ATM traffic controller employing the NFCAC scheme, and the EBCAC, FLCAC, NNCAC, RBFCAC schemes. It is found that the QoSs for both types of traffic are indeed guaranteed for all of these control schemes. Fig. 7 shows that the system utilization of the NFCAC scheme and the four schemes. We can find that the utilization of the NFCAC scheme is slightly greater than that of the NNCAC and the RBFCAC schemes; the system utilizations of NFCAC, NNCAC, and RBFCAC are 91%, 90.5%, and 89%, respectively; and the NFCAC scheme offers about 32% and 11% greater system utilization than the EBCAC scheme and the FLCAC scheme. It is because NFCAC can incorporate the domain knowledge obtained from both the analytical-based method (the equivalent capacity scheme [1] is employed in the bandwidth estimator) and the measurement-based method (the system statistics of the queue length, the change rate of the queue length, and the CLR are considered in the congestion controller). Also, the reason for the performance improvement is that NFCAC possesses the learning capability of the neural network.

Fig. 8 shows the training time required for the NFCAC scheme and the NNCAC, RBFCAC schemes. Here, a widely used back-propagation learning algorithm was employed to adjust the membership functions (i.e. represented in terms of weights) of the multilayer neural fuzzy network and neural network for the NFCAC and NNCAC schemes, while the RBFCAC scheme is basically trained by the hybrid learning rule: unsupervised learning in the input layer and supervised learning in the output layer. It is found that NFCAC has training time of 7 (4) epochs, while RBFCAC and NNCAC have training time of 103 (40) and 5×10^4 (6×10^2), respectively, for type-1 (type-2) traffic. The NFCAC has higher learning speed than the RBFCAC and NNCAC. One reason is that the neural fuzzy network is a structured network, thus the NFCAC controller can easily adopt the domain knowledge of conventional control methods to construct the initial rule structure and the parameters of the membership functions, providing an excellent initial guess in adjusting its weights; on the contrast, the neural network is a nonstructured network, which cannot incorporate domain knowledge about system. The other reason is that the neural fuzzy network has simpler structure than the neural network; the number of tunning parameters used in the neural fuzzy network is quite small, as compared to the neural network such as MLP and RBFN considered here. In this paper, there are only 16 weighting parameters used in NFCAC, while there are 150 and 480 weighting parameters required for the RBFCAC and NNCAC, respectively. It is also noted that the RBFCAC scheme has less learning time than the NNCAC scheme. This is because the RBFCAC scheme can have the proper initial setting of means



(a)



(b)

Fig. 8. Training cycles needed for (a) type-1 traffic and (b) type-2 traffic.

and variances for the Gaussian activation functions during unsupervised learning according to the prior knowledge, and it has only one layer of connection needed to be trained by supervised learning.

As usually noted, RBFCAC can have faster training speed than NNCAC but cannot achieve the same accuracy as the back-propagation NNCAC. In the simulations, we first adopted the same set of data used to train NFCAC and NNCAC for RBFCAC. However, it was found that RBFCAC finally violated the QoS contracts due to its error decision of accepting more users than it should be. In order to provide QoS guarantee for RBFCAC, we have to prepare much more training data, especially those around the acceptance/rejection boundary. This will increase the training time of RBFCAC in each epoch than those required by NFCAC and NNCAC. Moreover, the overall processing time of RBFCAC is greater than that needed by either NFCAC or NNCAC because RBFCAC uses more nodes (compared with NFCAC) and a more complicated activation function (compared with NNCAC). All of these would degrade the performance of RBFCAC in real application.

IV. CONCLUSION

This paper proposes a neural fuzzy approach for connection admission control in high-speed multimedia networks. The NFCAC scheme combines the linguistic control capability of a fuzzy logic controller and the learning ability of a neural network. This type of integrated neural fuzzy system can automatically construct a rule structure by learning from training examples and can self-calibrate parameters of membership functions. It not only provides a robust framework to mimic experts' knowledge embodied in existing traffic control techniques but also constructs intelligent computational algorithms for traffic control. It can be easily trained and enhances system utilization. Simulation results show that the proposed NFCAC scheme provides system utilization about 32% and 11% higher than the EBCAC and FLCAC schemes proposed in [3] and [8], respectively, and the NFCAC scheme requires only a fraction of the 10^3 order and the 10^1 order of training cycles, consumed by the NNCAC scheme proposed in [14] and RBFCAC scheme, respectively. An NFCAC scheme such as the one introduced here may be the answer to the problem of designing a coherent call admission controller for ATM systems.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments, which led to a substantial improvement in the paper.

REFERENCES

[1] R. Guèrin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Area Commun.*, vol. 9, pp. 968–981, Sept. 1991.
[2] H. Saito, "Call admission control in an ATM network using upper bound of cell loss probability," *IEEE Trans. Commun.*, vol. 40, pp. 1512–1521, Sept. 1992.
[3] G. Kesidis, J. Walrand, and C. S. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Trans. Networking*, vol. 1, pp. 424–428, Aug. 1993.
[4] A. Elwalid, D. Heyman, T. V. Lakshman, D. Mitra, and A. Weiss, "Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing," *IEEE J. Select. Area Commun.*, vol. 13, pp. 1004–1016, Aug. 1995.
[5] D. Tse and M. Grossglauser, "Measurement-based call admission control: Analysis and simulation," *INFOCOM'97*, pp. 983–991
[6] M. Murata, Y. Oie, T. Suda, and H. Miyahara, "Analysis of a discrete-time single-server queue with bursty input for traffic control in ATM networks," in *IEEE GLOBECOM'89*, pp. 1781–1787.
[7] A. R. Bonde and S. Ghosh, "A comparative study of fuzzy versus 'fixed' thresholds for robust queue management in cell-switching networks," *IEEE/ACM Trans. Networking*, vol. 2, pp. 337-344, Aug. 1994.
[8] R. G. Cheng and C. J. Chang, "Design of a fuzzy traffic controller for ATM networks," *IEEE/ACM Trans. Networking*, vol. 4, pp. 460–469, June 1996.
[9] T. D. Ndousse, "Fuzzy neural control of voice cells in ATM network," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1488–1494, Dec. 1994.
[10] Ed. I. W. Habib, "Neurocomputing in high-speed networks," *IEEE Commun. Mag.*, vol. 33, pp. 337–344, Oct. 1995.
[11] A. Hiramatsu, "ATM communications network control by neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 122–130, Mar. 1990.
[12] P. Tran-Gia and O. Gropp, "Performance of a neural net used as admission controller in ATM systems," in *IEEE GLOBECOM'92*, pp. 1303–1309.
[13] S. Y. Youssef, I. W. Habib, and T. N. Saadawi, "A neural network control for effective admission control in ATM networks," in *IEEE ICC'96*, pp. 434–438, June 1996.
[14] R. G. Cheng and C. J. Chang, "Neural network connection admission control for ATM networks," *Proc. Inst. Elect. Eng. Commun.*, vol. 144, no. 2, pp. 93–98, Apr. 1997.
[15] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice-Hall, 1992.
[16] N. E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 290–295, Dec. 1990.
[17] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, pp. 1320–1336, Dec. 1991.
[18] N. Yin, S. Q. Li, and T. E. Stern, "Congestion control for packet voice by selective packet discarding," *IEEE Trans. Commun.*, vol. 38, pp. 674–683, May 1990.
[19] S. T. Welstead, *Neural Network and Fuzzy Logic Applications in C++*. New York: Wiley, 1994.

[20] Y. Lin and G. A. Cunningham III, “A new approach to fuzzy-neural system,” *IEEE Trans. Fuzzy Syst.*, pp. 190–198, May 1995.

[21] D. E. Goldberg, *Genetic Algorithms*. Reading, MA: Addison-Wesley, 1989.

[22] D. L. Gall, “MPEG: A video compression standard for multimedia applications,” *Commun. ACM*, vol. 34, no. 4, pp. 46–58, Apr. 1991.

Ray-Guang Cheng received B.E., M.E., and Ph.D. degrees in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1991, 1993, and 1996, respectively.

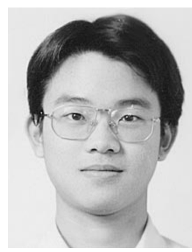
Since 1997, he has been a Researcher in the Advanced Technology Center, Computer and Communications Research Laboratories, Industrial Technology Research Institute, Taiwan, where he was involved in the designing of the medium access control protocol for the third-generation wireless networks. His current research interests include performance analysis, broadband wireless communication networks, and asynchronous transfer mode networks.

Dr. Cheng is a member of Phi Tau Phi.

Chung-Ju Chang (S’81-M’85-SM’94) received the B.E. and the M.E. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1972 and 1976, respectively, and the Ph.D. degree in electrical engineering from National Taiwan University in 1985.

From 1976 to 1988, he was with Telecommunication Laboratories, Directorate General of Telecommunications, Ministry of Communications, Republic of China, as a Design Engineer, Supervisor, Project Manager, and then Division Director. There, he was involved in designing digital switching system, ISDN user–network interface, and ISDN service and technology trials. He has also been a Science and Technical Advisor for Minister of the Ministry of Communications from 1987 to 1989. In August 1988, he joined the faculty of the Department of Communication Engineering and Center for Telecommunications Research, College of Electrical Engineering and Computer Science, National Chiao Tung University, as an Associate Professor. He has been a Professor since 1993. He was Director of the Institute of Communication Engineering from August 1993 to July 1995. He served as an Advisor for the Ministry of Education to promote the education of communication science and technologies for colleges and universities in Taiwan since 1995. He is also a committee member of the Telecommunication Deliberate Body. His research interests include performance evaluation, wireless communication networks, and broadband networks.

Dr. Chang is a member of the Chinese Institute of Engineers (CIE).



Li-Fong Lin (S’98) received the B.E. degree in communication engineering from Nation ChiaoTung University, Hsinchu, Taiwan, in 1996, where he is currently working toward the Ph.D. degree in communication engineering.

His current research interests include performance analysis, connection admission control over asynchronous transfer mode networks, fuzzy systems, and neural networks.