# NEURAL NETWORKS IN TELECOMMUNICATIONS

*EDITED BY*

**Ben YUHAS**
*Bellcore*
*Morristown, NJ 07960*

**Nirwan ANSARI**
*Electrical and Computer Engineering Department*
*New Jersey Institute of Technology*
*Newark, NJ 07102*

KLUWER ACADEMIC PUBLISHERS
Boston/London/Dordrecht

**Distributors for North America:**
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA

**Distributors for all other countries:**
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

**Library of Congress Cataloging-in-Publication Data**

Neural networks in telecommunications / edited by Ben Yuhas, Nirwan Ansari.
p. cm.
Includes bibliographical references and index.
ISBN 0-7923-9417-8 (alk. paper)
1. Telecommunications systems--Data processing. 2. Neural networks (Computer Science) I. Yuhas, Ben, 1959- . II. Ansari, Nirwan, 1958- .
TK5102.5.N47 1994 93-33665
621.382'0285'63--dc20 CIP

# 7

# STRUCTURE AND PERFORMANCE OF NEURAL NETS IN BROADBAND SYSTEM ADMISSION CONTROL

**Phuoc Tran-Gia and Oliver Gropp**

*Institute of Computer Science*
*University of Würzburg*

## 1 INTRODUCTION

This chapter[1] is dedicated to the use of neural networks for the connection admission control (CAC) in Asynchronous Transfer Mode (ATM) networks. An overview of ATM is provided by Hiramatsu in a preceding chapter. The major aim is to present and to compare possible neural net structures which can be applied to CAC and to show the performance of a basic neural net under various stationary and non-stationary load conditions. In Section 2 basic principles of the use of feed-forward neural networks with back-propagation learning in connection admission control are discussed and different alternative neural net structures are compared. A simple neural net is selected as an example in Section 3 and Section 4 to show the acceptance control performance and to discuss numerical aspects of the neural net under consideration.

## 2 NEURAL NETWORKS FOR CONNECTION ADMISSION CONTROL

Depending on the information available to the CAC function and its location in the communication network, different neural net structures can be developed. In this section we will briefly present these alternatives and discuss in particular the basic function and learning procedure of a back-propagation neural network used as admission controller.

[1]This chapter is an extended and updated version of [1].

## 2.1 Admission control in broadband networks

The connection admission control plays an important role during the resource allocation procedure of an ATM network (cf. [2]). According to CCITT [3, 4] the CAC function is defined as: " ... the set of actions taken by the network at the call set-up phase (or during the call re-negotiation phase) in order to establish whether a (virtual channel or virtual path) connection can be accepted or rejected."

Given that the CAC function is able to estimate the quality of service (QOS) before and after having accepted the requested connection, it can make the acceptance decision, i.e. the request will be rejected if the required QOS cannot be maintained.

In the following, we consider a number $M$ of different connection types to be served by the network.

We distinguish two cases:

i) CAC based on network state : In this case, we assume that the entire information about the number of all connections being multiplexed is available.

The system state seen by the network is denoted by $X = \{n_1, n_2, .., n_M\}$, where $n_i$ is the number of active connections of type $i$ being in the system. The main CAC function can now be represented by a mapping of the system state $X$ to a decision vector $Z$ defined by $Z = \{z_1, z_2, .., z_M\}$, where $z_i = 1$ denotes the acceptance of a connection establishment request of type $i$ and $z_i = 0$ its rejection. The CAC is thus reduced to the implementation of a mapping $f : X \rightarrow Z = f(X)$ according to the predefined quality of service of the network.

The mapping $f$ can further be simplified by using the state $X^* = \{n_1, n_2, \ldots, n_i + 1, \ldots, n_M\}$, i.e. the system state just after accepting the connection request of type $i$. The decision vector is reduced to $Z^* = \{z_i\}$ and the CAC mapping to $f^* : X^* \rightarrow Z^* = f^*(X^*)$.

ii) CAC based on bit-rate process : In this case, only the superimposed bit-rate process is available to the connection admission control function. This is the case if an intermediate ATM switching node does not have the whole system information, but only knows the bit-rate processes to be transferred.



Denoting the observed total bit-rate function during the time interval $(t, t+\Delta t)$ by $Y$, the CAC function can again be represented by the mapping $g : Y \rightarrow Z = g(Y)$.

## 2.2 Neural network as admission controller

As discussed in the previous subsection, the connection admission control function can be interpreted as a mapping of the state vector $X$ into the acceptance decision vector $Z$. This functional mapping divides the $M$-dimensional state space into two regions: the acceptance region and the rejection region. In other words, the CAC problem can be formulated like a pattern recognition problem: upon recognition of the load pattern $X$, a yes/no decision has to be made to accept/reject the connection request. This property in conjunction with the use of a neural net for connection control purposes in ATM systems is thus quite obvious. In this chapter we will use the class of feed-forward neural nets with back-propagation learning algorithm as described in Chapter 1 to solve the CAC problem.

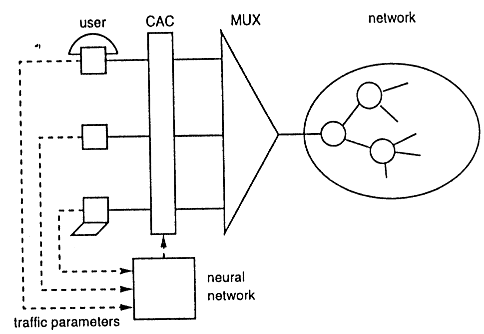



**Figure 1** *Neural net for admission control*

The use of a neural network to control connection acceptance is illustrated in Fig. 1. This basic structure has been proposed in [5] and further developed in [6]. Traffic streams offered by different types of sources are multiplexed at the entry node of the high-speed communication network. In the proposed neural net structure in [5] the bit-rate function is used as input to the neural net. As a quality of service indicator, e.g., the cell blocking probability at the multiplexer can be used. During the learning phase of the neural net, the input/output patterns are as follows. Inputs are formed by the bit-rate pattern including the bit-rate process generated by the actual connection request. The resulting QOS will be observed and compared to the target QOS. If the target QOS is still held, the output of the current input/output pair is $Z = 1$, i.e. the connection can be accepted, the bit-rate pattern is a "good" pattern. If the resulting quality of service is lower than the target one the output is then $Z = 0$, i.e. the connection should not be accepted in the current load situation, the bit-rate pattern is a "bad" pattern. After learning input/output pairs have been presented, the neural net can be used in a recall mode to perform the CAC function. One of the disadvantages of this mechanism is the difficulty to generate a significant number of good- and bad-patterns for the neural net to learn. The CAC performance of the neural net is thus strongly dependent on the statistical significancy of load patterns during the learning phase. Therefore we decided to design a modified learning process for the neural net.

In the current study we devote our attention to the neural net structure depicted in Fig. 2. The neural net is designed to perform the mapping depicted in Section 2.1 case i). We consider a number $M$ of different classes of connections, each with different known bit-rate characteristic. The pairs of input/output patterns for the neural net to be learned is computed as indicated in Fig. 2. Starting with a state vector $X = \{n_1, n_2, .., n_M\}$ as the input part of a pattern the multiplexed bit-rate function is determined. Having this bit-rate function as traffic stream, the cell blocking probability can be estimated giving the actual quality of service. Upon a comparison of this measure with the target QOS, the acceptance decision $Z$ can be made. This can be interpreted as the decision to be made to accept/reject a connection request of type $i$ if actual system state is $\{n_1, n_2, .., n_i - 1, .., n_M\}$. The working mode of the neural net during the recall phase is as shown in Fig. 2, where the net will answer with an accept/reject decision $Z^*$ for a connection request of type $i$ when the input vector $X = \{n_1, n_2, .., n_i + 1, .., n_M\}$ is presented.

Thus, after the learning phase, the neural net performs the CAC by separating the M-dimensional input state space in two regions corresponding to a $(M - 1)$-dimensional decision surface. The decision surface, which separates the "accept" region from the "reject" region in the state space, can be thought of as stored in the weight vectors of the neural net.



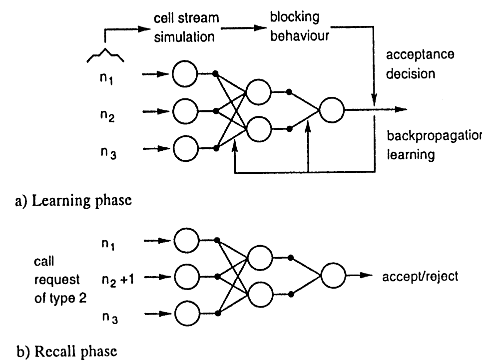



**Figure 2** *Working modes of a neural net based admission controller*

## 2.3 Alternative neural net structures for admission control

In the previous subsection we introduced the two cases: i) CAC based on network state, where information about the number of all connection being multiplexed is available, i.e. the detail state vector X of the system is known by the connection admission control function and ii) CAC based on bit-rate process, i.e. only the superimposed bit-rate process is available or measurable to the connection admission control function. According to these two cases different neural net structures can be developed, as shown in Fig. 2 and 3.

i) CAC based on network state

   A simple backpropagation neural net with only one output neuron is depicted in Fig. 2 b). The same functionality can be obtained using the neural net structure shown in Fig. 3 a).

ii) CAC based on bit-rate process

   Fig. 3 b) depicts a feedforward neural net for the mapping of a bit-rate pattern to an accept/reject decision. This reflects a communication network architecture with less signaling efforts involved, where only the superimposed bit-rate process is available or measurable to the connection admission control function.

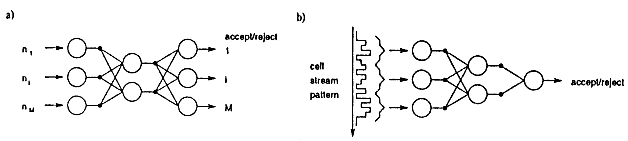



**Figure 3** *Alternative neural net structures for admission control*

## 3 EXAMPLE OF A NEURAL NETWORK FOR ADMISSION CONTROL

In this section we illustrate the performance of the neural net depicted in Fig. 2 b) in connection admission control.

### 3.1 Traffic assumptions and configuration parameters

The parameters of the ATM multiplexer and the connection types are as follows. The output of the multiplexer has a capacity of 600 Mbps, the buffer space is 0.5 Mb large. To model approximately the VBR (variable bit-rate) sources we consider sources with first-order Markovian bit-rate processes, where the two basic types are used as shown in Fig. 4: a) on/off-sources and b) binomial sources with two parameters: mean bit-rate $m$ and peak bit-rate $h$. The time axis is discretized by $\Delta t = 100\, msec$. The bit-rate $R$ will be expressed in number of basic units $\Delta B = 1\, Mbps$. In each $\Delta t$ we assume each source to have an independent bit-rate following the distribution:

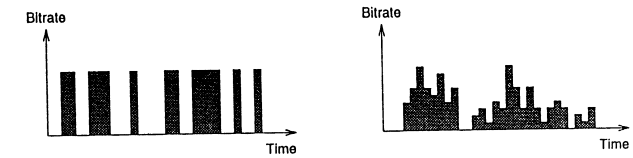
a) on/off-sources:
$p_{ON} = P\{R = \frac{h}{\Delta B}\} = \frac{m}{h}$ ;
$p_{OFF} = P\{R = 0\} = 1 - \frac{m}{h}$ ;

b) binomial sources:
$p_i = P\{R = i\} = \binom{\frac{h}{\Delta B}}{i}(\frac{m}{h})^i(1 - \frac{m}{h})^{\frac{h}{\Delta B} - i}, \quad i = 0, 1, ..., h.$

We consider three connection types:

Type 1: on-off, m = 10 Mbps, h = 40 Mbps, $c_R$ = 1.73.






a) On/off bitrate b) Binomial bitrate

**Figure 4** *Basic traffic source models*

Type 2: binomial, m = 5 Mbps, h = 40 Mbps, $c_R$ = 0.42.

Type 3: binomial, m = 5 Mbps, h = 80 Mbps, $c_R$ = 0.19.

where $c_R$ denotes the coefficient of variation of the bit-rate $R$.

On connection traffic level, the arrival process of connection requests is assumed to be Poisson with a mean interarrival time chosen according to the simulated load scenario. To obtain patterns for the neural net learning process, the cell stream traffic is simulated. During the simulation time the amount of lost cells is estimated by a fluid flow model (cf. Fig. 5). The connection duration is assumed to be exponentially distributed with mean 20 sec. This mean value is intentionally chosen to be short to enable simulation runs without loosing the qualitative significancy of the results obtained.

We simulate the traffic on burst level and the cell loss depending only on the actual sum of the bit-rates of the sources of active connections, the capacity of the output line and the buffer space of the multiplexer. Fig. 5 shows how the buffer occupancy b) depends on the bit-rate a). Cells are only stored in the buffer if the bit-rate exceeds 600 Mbps. The lost period is shaded dark in Fig. 5 b).

### 3.2 Alternative CAC methods for performance comparison

Since an agreement on CAC mechanisms for ATM system is not yet available, we will select a few methods proposed in the literature (cf. [7, 8, 9, 10, 11, 12]) to compare with the neural net CAC approach. The parameters taken into account for CAC purpose are the numbers of active sources with given connection types, and for each connection the mean bit-rate $m$ and the peak bit-rate $h$. The aim

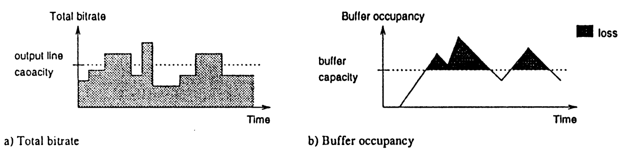



**Figure 5** *Estimation of cell blocking probability*

of the CAC is to keep the QOS, i.e. the cell loss rate below a given value, say $10^{-5}$.

### *Peak reservation method (PR)*

The most simple and robust method to limit the cell loss probability is to reserve the peak bit-rate for each accepted connection. New connections are only admitted if the sum of the peak bit-rates of the active connections and the new connection is smaller than the capacity of the output line. Thus no loss will ever appear. This method reduces ATM rather to STM (Synchronous Transfer Mode). Obviously, for more bursty bit-rate traffic the output channel is used in an inefficient way and the multiplexer utilization may be intolerably low. This peak bit-rate reservation method is considered here only as a lower bound for admission control methods aiming to high multiplexer utilization. In this section the improvement of utilization achieved by more sophisticated algorithms in comparison to this simple method will be shown.

### *Equivalent bandwidth method (EB)*

The expression "equivalent bandwidth" is introduced in [2]. Each source of type $i$ has its equivalent bandwidth $k_i$, which depends on its mean bit-rate $m_i$, its peak bit-rate $h_i$ and the capacity of the multiplexer output line:

$$k_i = C_1 m_i + C_2 \frac{m_i(h_i - m_i)}{c} \tag{1}$$

The constants $C_1$ and $C_2$ depend on the buffer space of the multiplexer and the maximum cell loss rate and have to be determined empirically. If a connection request of type $i$ arrives the following inequality is checked:

$$K + k_i \leq c \tag{2}$$



where $K$ denotes the sum of the equivalent bandwidths of the actual active connections. If it holds, the new connection is accepted, otherwise rejected.

### *Weighted Variance method (WV)*

The original method proposed in [13] has to be modified in the context of this study due to simulation reasons. The original method only works sufficiently well if the peak bit-rates of the subscribers is less than one percent of the output line capacity (cf. [11]). The modified algorithm works as follows, where $m_j$ represents the mean bit-rate of connection $j$, $h_j$ its peak bit-rate and $c$ the capacity of the output line. Connection $k$ is the new connection to be admitted, connections 1 to $k-1$ are already admitted. If

$$\sum_{j=1}^{k} h_j \leq c \tag{3}$$

holds, connection k is accepted. If this inequality does not hold, the following one is employed:

$$\alpha \sqrt{\sum_{j=1}^{k} m_j(h_j - m_j)} + \sum_{j=1}^{k} m_j + \max_{1 \leq j \leq k} h_j \leq c \tag{4}$$

If this inequality holds, connection $k$ is accepted, otherwise it is rejected.

The term $m_j(h_j - m_j)$ is an estimate of the variance of the bit-rate of connection $j$. Thus the constant $\alpha$ determines the influence of the variances of the source s on the CAC process. The term $\alpha$ has to be found in advance by simulation.

### *Neural network CAC (NN)*

We use a three layered feed-forward neural net to evaluate the CAC function. The neural net structure is the one depicted in Fig. 2 b).

The input consists of the vector $X^*$ of the numbers of active sources of each class where the component of the class of the arriving request is incremented by one. The result of the feed-forward computations at the output unit is a real number between 0 and 1. If the output value is less than a threshold (say 0.5) the new connection is accepted, otherwise rejected. The decision of the neural net depends on its internal set of weight matrices which have to be determined in advance during the learning phase as discussed in the previous section.

### 3.3 Neural net convergence and numerical issues

As mentioned before, the neural net needs a learning process to fix its weight vectors. This process uses a set of patterns to be learned. Each pattern consists of an input vector $X^*$ and the corresponding output value $Z^*$, which have to be chosen that the network has the capability to work as a mapping function $f^*$ for CAC (cf. section 2.1). We obtain this pattern using a simulation of the multiplexer state process, i.e. we fix the number of active connection of each traffic class at certain values and determine the corresponding loss rate at the multiplexer buffer. If this loss rate is less than a predefined value (in this study $10^{-5}$) this set of connection can be accepted, otherwise it should be rejected. We perform this simulation for the vectors

$$X_i^* = \{i_1 k, i_2 k, .., i_M k\} \quad \text{with} \quad 0 \leq i_j \leq \frac{600 Mbps}{mean\ bit-rate\ of\ type\ j} \tag{5}$$

and step size $k$. Thus we get an equally spaced $M$ dimensional grid whose nodes are named with A(ccept) or R(eject). This grid can be separated by an $M-1$ dimensional decision surface in an "accept" and a "reject" region. As an example Fig. 6 shows this grid for $M = 2$, where $n_1$ and $n_2$ denote the numbers of active connections of class 1 and 2. In this case the decision surface is just a line.

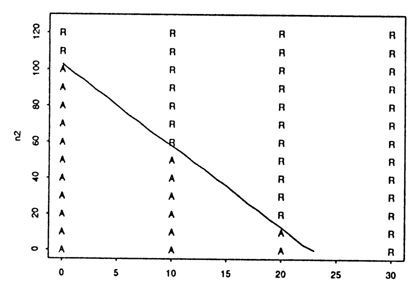



Figure 6 *Learning patterns of neural net*

To complete the pattern each $X_i^*$ gets a $Z_i^* = 0.2$ if the node name is "A" and a $Z_i^* = 0.8$ if the node name is "R." The values 0.2 and 0.8 instead of 0 and 1 are used to obtain a more appropriate learning algorithm.



In the literature a number of different algorithms to adapt the weights of the neural net can be found. In this chapter we use the BFGS algorithm of [14], well known from the theory of unconstrained optimization, aiming for a better convergence speed and a better numerical stability (cf. [15]). First we have to transform the learning process into a function minimization problem. Given the set of learning patterns we define an error function

$$E(W) = \frac{1}{2N} \sum_{i=1}^{N} (Z_i^* - F(X_i^*, W))^2 \tag{6}$$

The term $W$ denotes the vector of weights and $F(X_i^*, W)$ the output of the neural net upon $X_i^*$ presented to the input layer. Using the BFGS algorithm $E(W)$ is minimized in the weight space:

1. Initialize $W$ with random values ranging from $-0.5$ to $0.5$.
2. Calculate the search direction in the weight space and perform a line search in this direction to get the next $W$ with a smaller $E(W)$.
3. Check the stop condition ($E(W)$ small enough or local minimum of $E(W)$ is reached). If the condition is not true continue with step 2.

If the final $E(W)$ is small enough, the learning process is terminated. For the recall phase the internal weights of the neural network are now fixed to their final values. The neural network is thought of to has learned the functional mapping $f^*(X^*) = Z^*$ correctly only for the training patterns. Then it is able to perform this mapping also for all other input patterns with the help of the learned decision surface. This property is often referred to as "learning by examples."

### 3.4 Performance results and discussion

The load control performance of the neural net will be discussed in this subsection, taking into account stationary and non-stationary load conditions.

The neural net operates in the recall mode. Results are obtained by means of simulations with different mixtures of the three connection types described in Section 3.1. For the "Equivalent Bandwidth" and the "Weighted Variance" methods the parameters $C_1$ and $C_2$ or respectively, $\alpha$, had to be determined to guarantee a cell loss rate smaller than the threshold $10^{-5}$. Table 1 shows the multiplexer utilization for the CAC methods considered. The column 'Mix' indicates the mixture of the connection types used. Without any admission

| Mix | PR | EB | WV | NN |
|---|---|---|---|---|
| 1/2 | 20.2 % | 47.9 % | 47.4 % | 47.5 % |
| 1/3 | 17.9 % | 45.4 % | 43.6 % | 48.0 % |
| 2/3 | 10.3 % | 69.1 % | 67.0 % | 66.7 % |
| 1/2/3 | 15.8 % | 49.5 % | 50.6 % | 55.4 % |

**Table 1** Multiplexer utilization

control the utilization of the multiplexer would be about 91 %, without maintaining the desired QOS. As expected, PR is the most restrictive method and has a bad performance, whereas the other methods perform almost on the same level. Only in the case with all the three connection types involved a slight advantage of the NN control can be observed. The reason for this fact is the difference in rejection behavior of the EB and WV method on the one hand and NN on the other hand as shown in Table 2. As shown in Table 2 the two

| Type | PR | EB | WV | NN |
|---|---|---|---|---|
| 1 | 78.2 % | 51.2 % | 50.9 % | 55.0 % |
| 2 | 78.4 % | 36.0 % | 33.0 % | 22.0 % |
| 3 | 94.7 % | 42.5 % | 38.2 % | 21.6 % |

**Table 2** *Call request rejection rates*

methods EB and WV have almost the same connection blocking probabilities. The conclusion of the comparison of their performance with the NN solution is that the NN method rejection decision depends mainly on the mean bit-rate of the connection type while the decision of EB and WV depend on mean bit-rate and variance.

Fig. 7 shows the decision surface of the considered connection admission control methods, which separates the accept and reject regions. The accept region lies on the left hand side of the decision surface. The two methods EB and WV have almost the same decision line, which again indicates the similarity of their performances. The NN decision surface is extremely different. It can be observed that the NN algorithm accepts much more sources with small mean bit-rate (type 2) and less sources with high bit-rate (type 1) than the EB and



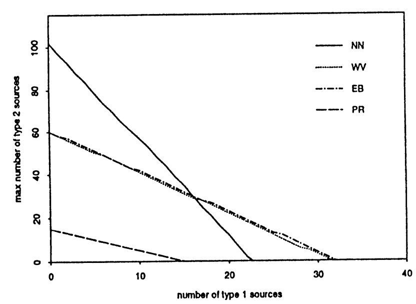



**Figure 7** *Decision surface for CAC*

WV methods. From communication network point of view, this results in the same multiplexer utilization (Table 1, row '1/2'), whereas from user viewpoint the differences for the user groups using connection type 1 or 2 are significant.

To show the overload control performance of the CAC methods under consideration, it is necessary to study the CAC response on a non-stationary overload pattern. In the diagrams to follow we use as overload pattern a rectangular overload pulse as illustrated in Fig. 8 and observe the time-dependent CAC reaction in terms of cell and connection blocking probabilities. Clearly, a better CAC mechanism should react to the overload phase with smaller connection blocking probability while keeping the cell loss rate on the same level as under normal load conditions.

Results of the non-stationary load cases are shown in Fig. 8 and Fig. 9. It should be noted that the coefficients of variation of the source bit-rate processes of the three connection types are 1.73, 0.42 and 0.19 respectively.

Fig. 8 shows a comparison of the non-stationary connection blocking probabilities of connection type 3 of the four CAC control methods. In this case it can be seen that the overload performance of the neural net solution is the most efficient.

In Fig. 9 the overload control performance of the neural net is shown where the three connection types are taken as input. As mentioned above, the connection blocking probability is more sensitive to the mean bit-rate than to the variation of the bit-rate process.

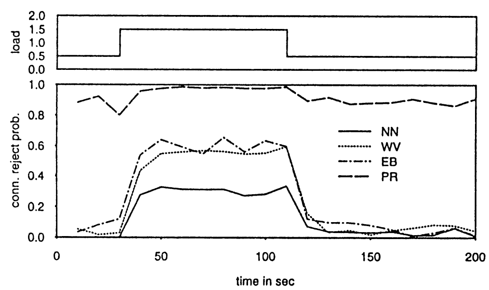



**Figure 8** *Comparison of overload performances*

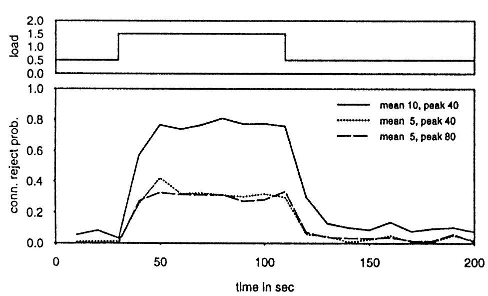



**Figure 9** *Impact of traffic variation on overload control performance*



## 4 CONCLUSIONS

In this chapter different aspects concerning possible use of neural nets to perform connection admission control (CAC) in broadband integrated services networks have been discussed. The formulation of CAC problem as functional mapping and in consequence, the use of learning algorithms to represent the required mapping were shown and architecture alternatives for the CAC neural nets using the class of feed-forward structures in conjunction with back-propagation learning are depicted. In order to discuss performance aspects a basic net example has been investigated. The neural net performance has been compared with other connection admission control mechanisms like the peak bit-rate, the equivalent bandwidth and the weighted variance method. Numerical results for stationary and non-stationary pulse-form overload patterns have been obtained to illustrate the capability of neural nets used as connection admission controller in ATM environments.

In most of load scenarios under consideration the CAC performance of the investigated neural net structure is comparable with and in some cases better than the CAC methods mentioned above, even by using a very small and simple neural net.

To improve the performance of CAC by neural nets, other neural net structures or other input representations can be developed. One promising candidate is a combined solution of an adaptive neural net with learning patterns, which contain more information about the past of the observable load situation.

## REFERENCES

[1] P. Tran-Gia and O. Gropp, "Performance of a Neural Net used as Admission Controller in ATM Systems," Proc. Globecom 92, Orlando, FL, pp. 1303-1309.

[2] COST 224 Committee (ed. J. Roberts), COST 224 Final report, Performance evaluation and design of multiservice networks, Paris, October 1991.

[3] CCITT, Draft Recommendation I.311, B-ISDN General network aspects, COM XVIII-R 34-E, Geneva 1990.

[4] CCITT, Temporary Document 43/XVIII, On networking and resource management, Study group XVIII, Matsuyama 1990.

[5] A. Hiramatsu, "ATM communications network control by neural networks," Proc. IEEE Int. Conf. on Neural Networks 1989, Vol. I, pp. 259-266.

[6] A. Hiramatsu, "ATM communications network control by neural networks," IEEE Transactions on Neural Networks, (1990)1, pp. 122-130.

[7] P. Castelli, E. Cavallero, and A. Tonietti, "Policing and call admission problems in ATM networks," Proc. 13th Int. Teletraffic Congress, Copenhagen 1991, pp. 847-858.

[8] A. E. Eckberg, D. T. Luan, and D. M. Lucantoni, "Meeting the challenge: Congestion control strategies for broadband information transport," Proc. Globecom 1989, Dallas, pp. 1769-1773 (49.3).

[9] G. Galassi, G. Rigolio, and L. Verri, "Resource management and dimensioning in ATM networks," IEEE Network Magazine, (1990)5, pp. 8-17.

[10] S. B. Jacobson, K. Moth, and L. Dittmann, "Load control in ATM networks," Proc. XIII Int. Switching Symposium, Stockholm 1990, pp. 131-138.

[11] A. Lombardo, S. Palazzo, and D. Panno, "Admission control over mixed traffic in ATM networks," Int. Journal of Digital and Analog Communication Systems, (1990)3, 155-159.

[12] C. Rasmussen, J. Sørensen, K. S. Kvols, and S. B. Jacobsen, "Source-independent acceptance procedures in ATM networks," Journal on Selected Areas of Communications, JSAC-(1991)4, pp. 351-358.

[13] E. Wallmeier, "A connection acceptance algorithm for ATM networks based on mean and peak bit rates," Int. Journal of Digital and Analog Communication Systems, (1990)3, 143-153.

[14] R. Fletcher, Practical methods of optimization, Wiley 1980.

[15] R. L. Watrous, "Learning algorithms for connectionist networks: Applied gradient methods of nonlinear optimization," Proc. IEEE Int. Conf. on Neural Networks 1987, pp. 619-627.