Measure-based Estimation of The Required Capacity on ATM Switches Using Neural Networks

Miguel Franklin
castro@hugo.int-evry.fr

Adriano Nascimento
nascimen@hugo.int-evry.fr

Marcelino Pequeno
marcel@lia.ufc.br

Mauro Oliveira
mauro@etfce.br

Artificial Intelligence Laboratory - LIA
Federal University of Ceará - UFC - Brazil

Multiinstitutional Laboratory of Computer Networks - LAR
Federal Center for Technological Education of Ceará - CEFET-CE - Brazil

Abstract

The Required Capacity is the minimum amount of bandwidth that must be allocated to a traffic source in order to grant the Quality of Service for the system. This value can be used as a parameter for Connection Admission Control (CAC) and Resource Management. The main purpose of this paper is to experiment and vindicate the usage of Artificial Neural Networks to estimate the required capacity on ATM switches. For this purpose, it was developed a specific approach based on parameters that define the overall behavior of the aggregate traffic that reaches the switch, instead of using traffic descriptors on analytical methods.

1 Introduction

The Asynchronous Transfer Mode (ATM) is the technology used to support the Broadband Integrated Services Digital Networks (B-ISDN) and is also used for LAN (Local Area Network), MAN (Metropolitan Area Network) and WAN (Wide Area Network) support. One of the main characteristics of ATM is the possibility to switch distinct natures of connections, maintaining its overall Quality of Service (QoS).

There are two basic behaviors for ATM traffic sources: constant rate (CBR sources) and variable rate (ABR and VBR sources) [8]. CBR traffic sources are just well studied and do not present considerable difficulty to resource allocation and operation. On the other hand, VBR and ABR traffic sources present a greater complexity due to new variables and requirements [6]. For instance, take on conventional telephone service with silence detection. In this case, traffic is generated on a constant rate of 64 Kbps only when one of the interlocutors speaks. On moments of silence, no traffic is generated [6]. Thus, it's found to be much more complex to characterize traffic for sources with variable rate than it is for the constant rate case.

The Required Capacity is the minimum amount of bandwidth that must be allocated to a traffic source in order to keep its QoS requirements. On Statistical Multiplexing, this value for each variable traffic source is supposed to be less than its maximum transfer rate (PCR - Peak Cell Rate). The evaluation of the Required Capacity for an individual traffic source and for the aggregate traffic is considered a complex task, due to the essentially stochastic nature of the ATM traffic and the number of variables involved, *e.g.* buffer size and source traffic descritors. Several papers have presented

analytical methods for estimating this value [11] [12]. Other papers [7] [3] use Neural Networks for estimating the Required Capacity applied on mechanisms such as Connections Admission Control (CAC).

The evaluation of the Required Capacity of a traffic source is generally based on its characterization. However, this characterization can be under- or overestimated because of the fuzziness of the ATM traffic. This can lead to a corrupted value of the Required Capacity. Therefore, it becomes necessary to develop a new method that takes into account the actual behavior of the aggregated traffic, instead of its theoretical description.

This paper describes an experimentation that vindicates the use of Artificial Neural Networks (NNs) on measure-based estimation of the Required Capacity on ATM switches. For this purpose, a specific approach was developed based on parameters that define the overall behavior of the aggregated traffic that reaches the ATM switch with destination to one only output link. For this purpose, a novel architecture named RENATA (Neural Networks applied to ATM Traffic) was implemented.

This document is organized as follows. Section 2 describes some aspects of the ATM traffic, such as statistical multiplexing and required capacity. Section 3 describes RENATA architecture, the prototyped environment for this experiment. On Section 4, an experimentation based on Neural Networks is proposed and modeled. Section 5 discusses the obtained results while the Section 6 presents conclusions about these experiments.

2 ATM Traffic

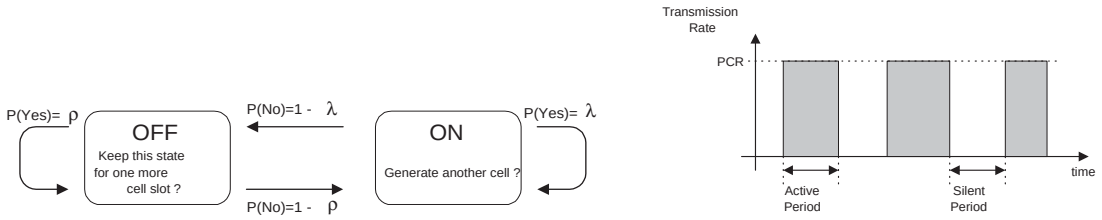
One of the most important factors for ATM traffic control complexity is the great variety of application types that can be supported, especially multimedia applications [5]. Each user may present distinct traffic characteristics and communication services may have different Quality of Service (QoS) requirements [6]. All this variety makes the traffic management task on an ATM network much more complex than on conventional ones.

One example of traffic control operation is the Connection Admission Control (CAC). In order to establish a new connection on ATM networks, a traffic contract must be accorded between application and the CAC mechanism. The parameters dealed with this contract include the required QoS and the traffic descriptor for the application. Usually, all traffic management schemes are based on parameters from traffic contracts [1].

2.1 Traffic Characterization

The transmission rate of a CBR connection remains constant for the time it lasts. This simple type of traffic can easily be characterized by its Sustainable Bit Rate (SBR). A VBR connection might assume several transmission states. A transmission state is given by its transmission rate and mean size. There is also another kind of application that demands for a variable traffic. It is named ABR (Available Bit Rate). The transmission rate in ABR sources is subjected to the availability of resources.

In this work, it was only considered VBR traffic sources characterized as ON–OFF, *i.e.*, it operates on two states: transmitting full peak rate (active period) or transmitting no traffic (silent period). The VBR ON–OFF traffic source behavior can be represented by a two-state Markovian chain, with state diagram shown on Fig. 1(a), where ρ and λ represent the probabilities for turning to active and silent periods, respectively. Traffic sources with state duration modeled according to Negative Exponential Distribution have been frequently studied. This model has been applied for data traffic characterization. Actually, it has been used for characterization of any sort of bursty traffic [9]. Figure 1(b) exemplifies a VBR ON–OFF traffic source.



(a) State Diagram (b) Example

Figura 1: ON–OFF Traffic Sources

To this special type of traffic, the most common parameters to describe its behavior are: Peak Cell Rate (PCR), Mean Active Period Length (t^{on}) and Mean Silent Period Length (t^{off}).

2.2 The Required Capacity

A traffic source may vary its transmission rate during operation. Hence, if the resource allocation for each application is defined by its PCR, there will be moments of wasting of resources, once that the bandwidth left idle by a traffic source could have been used by other applications. Thus, *Statistical Multiplexing* allows provisioning an amount of bandwidth smaller than the peak rate for each traffic source. Consequently, this procedure is more sensitive to congestions. However, there are statistical mechanisms which guarantee the Quality of Service for the whole system, maintaining the probability of cell loss under an arbitrary threshold ε.

The *Required Capacity* is, therefore, the minimum amount of bandwidth that must be provisioned to a traffic source in order to grant the Quality of Service for all traffic sources involved on the system.

The most important restriction for evaluating the Required Capacity for an application is to have an accurate description of its overall traffic behavior. This is not an easy task, due to the fuzziness of the ATM traffic.

2.3 The Equivalent Bandwidth Method

Many approaches were developed to estimate the Required Capacity for statistical multiplexing. One of them is *Equivalent Bandwidth*[1] (EB) which is proposed in [2]. This method estimates the Required Capacity for each individual VBR ON–OFF traffic source based on the following information:

- Traffic Source Information (for each application j):

  Peak Cell Rate (PCR_j), Mean Size of Active Period (t_j^{on}) and Mean Size of Silent Period (t_j^{off}).

- Switch Information:

  Buffer Size (ξ)

Thus, the Required Capacity for a traffic source j according to *Equivalent Bandwidth* method is defined by:

$$EB_j = PCR_j \times \frac{y_j - \xi + \sqrt{(y_j - \xi)^2 + 4\xi\rho_j y_j}}{2y_j}. \quad (1)$$

[1] Also known as *Equivalent Capacity*

where

$$y_j = \alpha t_j^{on}(1-\rho_j)PCR_j. \quad \text{and} \quad \alpha = ln(1/\varepsilon). \quad \text{and} \quad \rho_j = \frac{t_j^{on}}{t_j^{on}+t_j^{off}}.$$

The Equivalent Capacity is scalar, that is, the estimative of the EB for the aggregated traffic is obtained as follows:

$$EB = \sum_{j=1}^{N} EB_j.$$

EB is considered a quick and precise enough method to obtain the Required Capacity for VBR ON–OFF traffic sources. Nevertheless, the usage of wrong traffic descriptors might lead to an undesirable under- or overestimation of the Required Capacity value. So, it becomes necessary a novel method that takes into account mainly the actual traffic behavior, rather than potentially inaccurate traffic descriptors.

3 RENATA Architecture

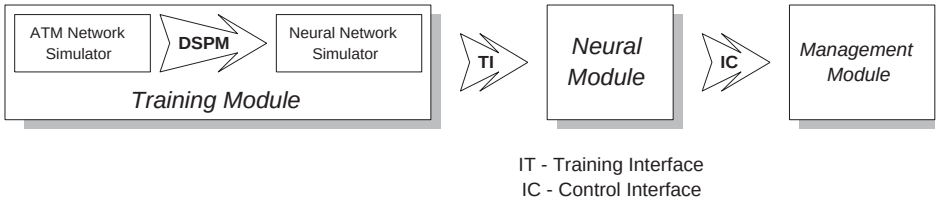


Figura 2: Functional Architecture of RENATA

RENATA (Neural Networks Applied to ATM Traffic) is an architecture applied for proactive management on ATM networks, using Artificial Neural Networks as an intelligent component.

This environment is composed by several modules that work together with the aim of presenting – at the end of the running process – an intelligent agent designed for monitoring and controlling tasks on network resources. This architecture is illustrated on Fig. 2.

The knowledge needed to training the neural network is obtained through simulation, what turns the task of experimentation on critical states much more easy and flexible. Furthermore, the results obtained can be much more precise as it is shown on Section 5.

The sample patterns that will be used for neural training are composed of a set of input and output values. The input values represent the variables that describe the configuration to be classified, *i.e.* the aggregate traffic behavior. The output value represents the expected value (the Required Capacity) that the neural network must estimate.

Thus, the mapping process between input parameters – which should characterize the aggregated traffic behavior – and the value of the required capacity for this traffic is done by an Artificial Neural Network. The main reason for this choice is that there is no analytical method to perform such mapping. Furthermore, ATM traffic control requires real-time processing, which can be offered by a neural network.

Each module is briefly described in the next subsections.

3.1 Training Module

The *Training Module* is composed by an ATM network simulator, a *Data Selection and Preparation Module* and a Neural Network simulator.

The ATM simulator must work on a theoretical model of the queuing system on an ATM switch fed with an arbitrary number of characterized VBR ON-OFF traffic sources. Based on this configuration, the simulator must produce a trace of parameters obtained from the simulated operation along time.

The *Data Selection and Preparation Module* (DSPM) contain information about the simulated environment, and is designed to read and filter the data produced by the ATM simulator. It plays a twofold role: to select the relevant information from the traces and to prepare sample patterns for use in the neural network training process.

The Neural Network simulator takes as input sample patterns produced by the DSPM. These patterns will be used for training and validating the neural network. After the training process, the resulting neural network must be checked out on new examples in order to estimate how good its generalization is.

3.2 Neural Module

The *Neural Module* is supposed to receive the trained neural network and transform it on a standalone working module, *e.g.* a C language code. The product of this process will be attached on the Management Module to some communication facility, in order to interact with actual resources.

3.3 Management Module

The *Management Module* represents the interface between the Neural Module and the actual network resources to be managed. This communication may be through any Manager × Agent protocol, such as SNMP (Simple Network Management Protocol), or through any specific interface to the ATM resources.

4 Neural Network-based Experiments

This section presents how RENATA architecture is modeled and implemented in order to evaluate Required Capacity from the actual traffic behavior of the ATM network. The patterns for training and validating the neural network are composed in a mixed fashion. The input parameters are acquired from the trace of simulated traffic, whereas the desired output is obtained from analytical method (EB) applied to traffic descriptors that were simulated (Fig. 3).

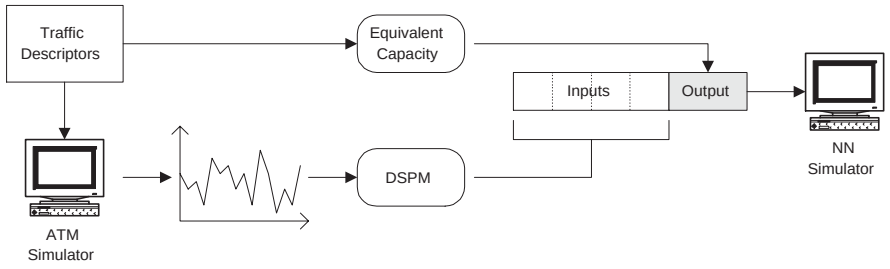


Figura 3: Sample Patterns Generation Scheme

For implementation purpose, the following statements are assumed:

- All applications have the same desirable max. CLP (Cell Loss Probability) value ($\varepsilon = 10^{-5}$);
- Only VBR ON-OFF traffic sources are switched;
- Traffic sources have their state sizes exponentially distributed;

- The inter-arrival time of cells for a traffic source follows Poisson Distribution.

The adaptation process made to RENATA architecture for the proposed problem can be described in two phases:

- Neural Network Design
- Experimentation

Each one of these tasks is described in the following subsections.

4.1 Neural Network Design

One of the most important tasks for designing a neural network is the formal description of input and output parameters. Input parameters must be chosen in the best way to represent faithfully the actual status that must be inferred, just the way output parameters must contain the desired high level semantics that the neural network is required to present.

For the proposed problem, the considered ATM switch equipment is characterized by its maximum number of applications supported N_{max} and buffer's maximum capacity ξ. On a given moment, this equipment multiplexes N traffic sources originated from input links that fill capacity L_{in}. All traffic generated by this applications must be switched to a unique output link with capacity L_{out}. Such applications are defined by VBR ON-OFF traffic. Each traffic source j is characterized by its Peak Cell Rate P_j and by its mean lengths of active and silent periods t_j^{on} and t_j^{off}, respectively, with $j \leq N \leq N_{max}$. The maximum Peak Cell Rate permitted for applications is P_{max}.

The proposed neural network makes use of information obtained from readings made on aggregated traffic's transmission rate at previous moments. By the use of this information, the neural network must be able to estimate the Required Capacity for the aggregated traffic at the present moment.

Measurement Points (MP's) and Check Points (CP's) are defined along time. MP's are points where measures on the instantaneous aggregated transmission rate are done. The time interval between two MP's is represented by $\Delta\tau$. CP's are special MP's where the obtained values from subsequent MP's until present moment are totaled. CP's are equidistant among them on ω intervals $\Delta\tau$ *i.e.* there are $(\omega - 1)$ MP's between two CP's. The present time is represented by T_0, while previous MP's are represented by T_i, with incremental values of i. Therefore, $T_i - T_{i+1} = \Delta\tau$, to any $i \geq 0$. The distribution of MP's and CP's along time is exemplified on Fig. 4.

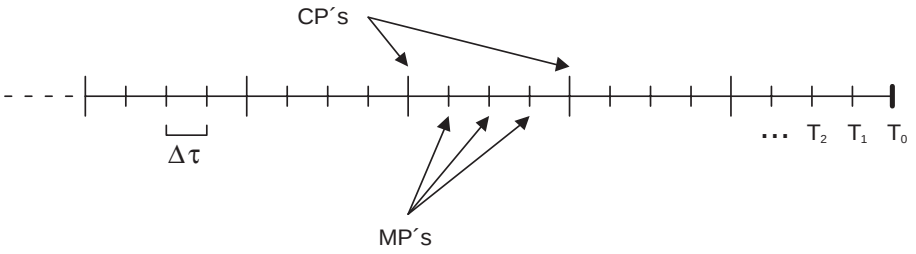


Figura 4: Check and Measure Points Diagram

The term *history* is defined as the number of CP values that must be included on one vector for input pattern. Therefore, if h is the history defined, then the system must be observed for $h \times \omega \times \Delta\tau$ for the aim to compose one input vector.

Let R_t^j be the *instantaneous transmission rate* of application j on the moment T_t. Consequently, if C_t^j is the number of bits transmitted by a traffic source j until instant T_t, then:

$$R_t^j = \lim_{\delta \to 0} \frac{C_{t+\delta}^j - C_t^j}{\delta}.$$

Then, the *instantaneous aggregated transmission rate* that reaches the switch on moment T_t is expressed by:

$$R_t = \sum_{j=1}^{N} R_t^j$$

where N is the number of applications currently switched by the system to the output link.

On each MP the instantaneous aggregated transmission rate is measured and on each CP the ω measured values from subsequent MP's – including the one from the CP itself – are totaled and processed.

Therefore, on instant T_m where $m \; mod \; \omega - 0$, *i.e.*, the moment T_m is a CP, σ_t is defined as being the standard deviation of the instantaneous aggregated transmission rates R_t, with t on the integer interval $[0; m]$.

$$\sigma_m = \sqrt{\frac{\sum_{t=0}^{m} (\bar{R}_m - R_t)^2}{m-1}}.$$

Where $\bar{R}_m$ is the average of aggregated instantaneous transmission rate from moment T_m to present time T_0.

$$\bar{R}_m = \frac{\sum_{t=1}^{m} R_t}{m}$$

Based on experimented values, the following functions were defined in order to normalize values $\bar{R}_t$ and σ_t:

$$f(t) = \frac{2 \times \bar{R}_t}{PCR_{max}} \qquad\qquad g(t) = \frac{4 \times \sigma_t}{PCR_{max}}.$$

Where $PCR_{max} = P_{max} \times N_{max}$.

Therefore, the selected values to compose the input vector $\mathcal{I}$ for neural network are:

$$\mathcal{I} = \left(\frac{N}{N_{max}}, \frac{\sum P_j}{PCR_{max}}, \overbrace{f(1 \times \omega), g(1 \times \omega), \cdots, f(h \times \omega), g(h \times \omega)}^{h \times} \right).$$

The values N_{max} and PCR_{max}, taken as denominators for the input vector, represent the normalization factors that must restrict the input values to the continuous interval $[0; 1]$.

The output vector for the neural network is composed by one only value, which represents the normalized required capacity for the given configuration. This value is obtained from the method *Equivalent Bandwidth* (EB), described by Eq. (1) applied to the traffic descriptors that were fed to the ATM network simulator. Thus, although the input parameters are obtained from simulated traffic behavior, the output value is taken from traffic descriptors. This choice makes the neural network represent the mapping between the overall traffic behavior and the desired value of Required Capacity. Therefore, the output for the neural network is an unidimensional vector given by:

$$\mathcal{O} = \left(\frac{\sum_{j=1}^{N} EB_j}{L_{out}} \right)$$

The neural network variety of choice for this experiment is *feed-forward*, which uses the *Backpropagation Momentum* [10] as training algorithm. This variation of the well-known *Backpropagation* has extra mechanisms that help ignoring local minima. The topology of choice consists of a three-layer neural network (input, hidden and output layers). The number of neurons on the input layer varies according to *history* parameter $n_{in} = 2 \times (h + 1)$. The output layer is composed by one only neuron, which represents the normalized value for the Required Capacity. The number of neurons on the hidden layer was varied looking towards a better precision on results. Figure 5 illustrates the designed neural network.

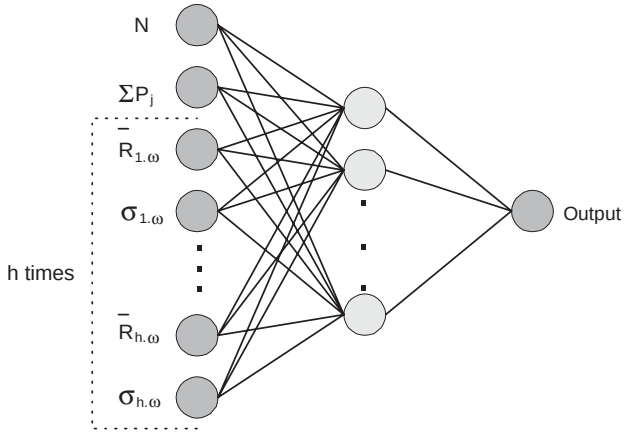


Figura 5: Neural Network Topology

4.2 Experimentation

For implementation purpose, boundary values were chosen for traffic descriptor parameters for traffic sources of hypothetical applications. Thus, PCR values vary from 0.1 to 2.75 $Mbps$, with granularity of 0.01 $Mbps$; and values for t^{on} and t^{off} are found between 0.01 and 2.5 ms, with granularity of 0.01 ms.

Each configuration is randomly composed and represents a situation where N traffic sources reach a switch with input links totaling $L_{in} = 155.52$ $Mbps$ with destination to a single output link with capacity $L_{out} = 51.84$ $Mbps$. A configuration may be composed for at least $N_{min} = 20$ and at most $N_{max} = 80$ applications (traffic sources), each with its values of PCR, t^{on} and t^{off} randomly generated based on the boundaries cited above.

For composition of sample patterns for the neural network training process, 4 000 configurations were generated, where 2 000 of them were used as training patterns and the rest has composed the validation pattern.

Each ATM network configuration is simulated for a virtual period of 1 second. The simulation tool used for this was the NIST (National Institute of Standards and Technology) ATM Network Simulator [4]. The mean processing time for 2 000 configurations is 10 hours on an IBM SP-2 parallel supercomputer with 4 nodes. For each simulated configuration, a log file is produced, which traces the instantaneous aggregated transmission rate with granularity of 10 ns.

The DSPM is fed with the log files produced by the ATM simulator. This module is supposed to randomly select time intervals on log files with the aim to extract values that must compose

input vectors for sample patterns. Each configuration can generate one or more sample vectors. For these experiments, it was defined to extract only one pattern for each simulated configuration. The software tool chosen for designing, training and validating neural network is the SNNS (Stuttgart Neural Network Simulator) [10]. Therefore, the output generated by the DSPM is a sample pattern file, in the shape of a compatible pattern file for the SNNS simulator.

After training and validating the designed neural network, a code written in C programming language is generated, representing the stub for integrating the trained neural network to other software modules.

5 Results and Discussion

In this section, the obtained results from several ATM simulations and neural network validations are presented. Two parameters are used for evaluating the neural network accuracy. The first one is *Sum of Squared Error* (SSE), described by:

$$MSE = \frac{\sum_{i=1}^{k}(w_i - o_i)^2}{k}$$

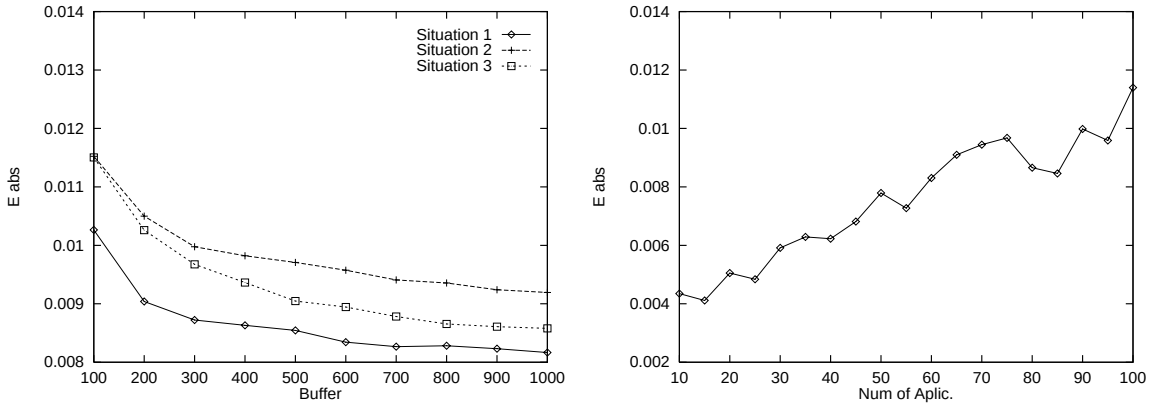
Where k represents the number of patterns in the validation set and w_i and o_i represent the desired and obtained values for neural network, respectively..

The other evaluation criterion is defined as *Absolute Mean Error*, given by:

$$E_{abs} = \frac{\sum |EB_{exp} - EB_{obt}|}{L_{in} \times N}$$

where EB_{exp} and EB_{obt} are values of expected and obtained Equivalent Bandwidth, respectively, for each traffic source evaluated by the neural network, L_{in} is the total capacity of input links and N is the number of active traffic sources on the evaluated configuration.

In general, experiments were performed using neural networks with 22 neurons on input layer (what corresponds to $history = 10$), 6 neurons on the hidden layer and one neuron on output layer. Exceptions are the experiments that generated Fig. 8(a) and 8(b)), where the parameter *history* and the number of neurons on hidden layer, respectively, were varied.



(a) (b)

Figura 6: Plots for Results

The first experiment evaluates the precision of the neural network according to the switch buffer size. Furthermore, different situations were defined and submitted to the original set of patterns.

Each of them corresponding to a filtered subset of patterns that were used for training the neural network. This was made to evaluate the neural network design on different situations of environment. The defined situations are:

- Situation 1:

  Select all patterns on the set;

- Situation 2:

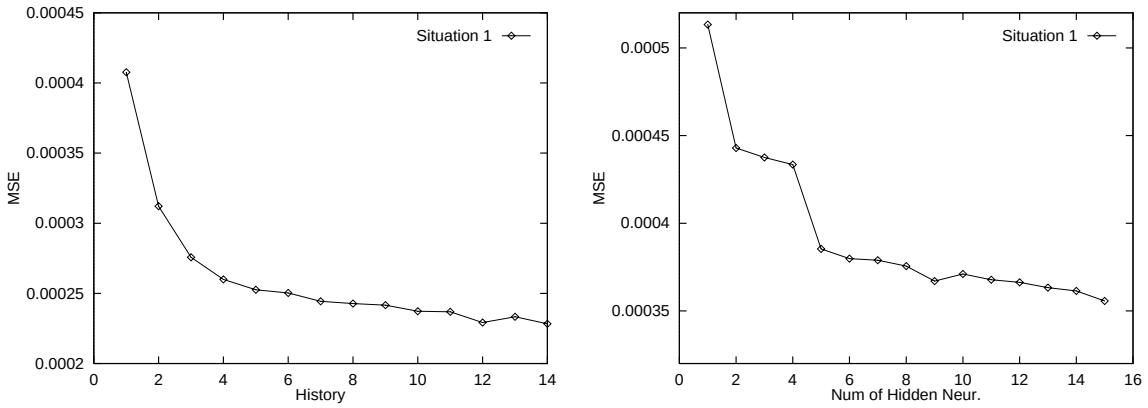
  Select only the patterns where the summation of the peak rates of applications ($\sum PCR$) is greater than or equal to the output link capacity (L_{out});

- Situation 3:

  Select the patterns where the expected value of Required Capacity is at most 40% distant of the output link capacity, *i.e.* $|L_{out} - EB_{exp}| \leq 0.4 \times L_{out}$;

The results for the first experiment is shown on Fig. 6(a). It's shown that Situation 1 reaches a better accuracy, followed by Situations 3 and 2.

A second experiment evaluates how the neural network accuracy is influenced by the characteristic of the number of applications contained on the configuration. It was found out that the precision of the neural network decreases as the number of applications involved on the configuration also increases. The results are ploted on Figure 6(b).

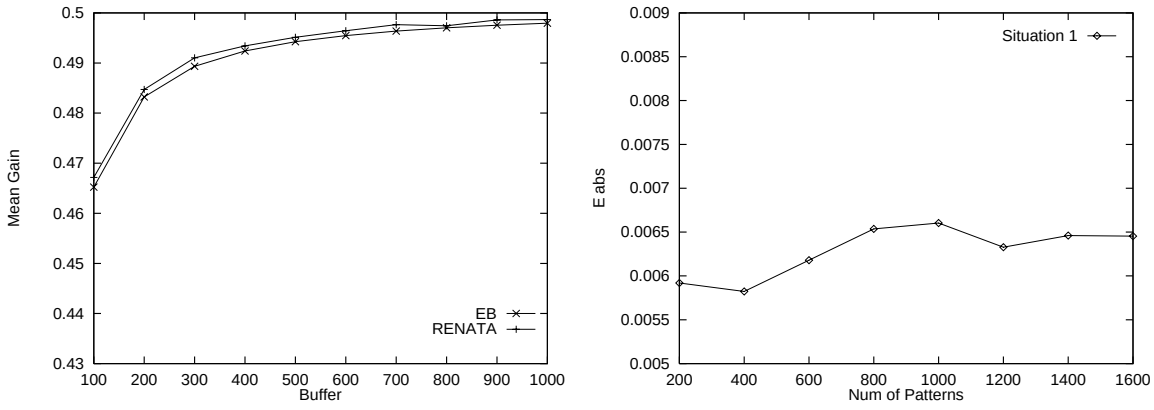


(a) (b)

Figura 7: Plots for Results

The third experiment shows how the neural network accuracy behaves with history variation. It was observed that the bigger the history is, the smaller is the MSE for the neural network. Nevertheless, the more history increases, the greater observation time is necessary for decision making. Therefore, another point to consider is that as history increases, the number of neurons on the input layer of the NN also increases, which can lead to a greater computational effort for training and processing the NN. Results are plotted on Fig. 7(a).

Another experiment tries to reach an optimum value to the number of neurons on hidden layer for buffer size of 1 000 cells. It was observed that the neural network precision degrades as the number of hidden neurons increases. Therefore, it was defined to use $N_h = 6$ neurons on hidden layer for performed experiments. Results are ploted on Figure 7(b)



(a) (b)

Figura 8: Plot for Results

Another experiment tries to show how the number of training patterns can influence on the neural network precision. It's observed on Figure 8(a) that there isn't a direct relationship between the number of training patterns and the neural network accuracy.

Figure 8(b) shows how accurate the neural network approach is according to the Equivalent Bandwidth. To obtain this, the average of statistical gain on validation set for both NN and EB is calculated. The deviation of the mean statistical gain between both methods was maintained by 10^{-3} order of magnitude, with buffer size varying from 100 to 1000 cells. It was also observed that the greater is the buffer size, the more approximated are the results of the NN to desired values.

6 Conclusions

The new approach presented on this paper has shown that it is viable to estimate Required Capacity based on parameters that describe the actual behavior of aggregated traffic, instead of traffic descriptors. The overall result for these experiments is that the NN found estimated values for Required Capacity that are distant from the desired value (EB) within average of 1.2 %. This result validates the proposed approach.

Although the approach presented on this paper depends on previous time that must be observed, the worst case for this task is still acceptable for real-time situations.

Nevertheless, experiments modeled for this experiment are still hypothetical. Actual applications require more observation time, as the values for their traffic descriptors are proportionally higher than experimental ones. Yet, we believe that all this adjustments can be made without any considerable loss of precision on results.

Referências

[1] Martin Bernhardt. Design and Implementation of a Web Based Tool for ATM Connection Management. Master's thesis, International Computer Science Institute, Berkeley University, August 1996.

[2] A. Elwalid and D Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-Speed Networks. *IEEE/ACM Transactions on Networking*, 1(3):329–343, Jun 1993.

[3] Olle Gallmo, Ernst Nordstrom, Mats Gustafsson, and Lars Asplund. Neural Networks for Preventive Traffic Control in Broadband ATM Networks. In *International Workshop on Mechatronical Computer Systems for Perception and Action (MCPA '93)*, pages 139–145, Halmstad, Sweden, Jun 1993.

[4] Nada Golmie, Frederic Mouveaux, Lance Hester, Yves Saintllan, Alfred Koenig, and David Su. *The NIST ATM/HFC Network Simulator: Operation and Programming Guide - Version 4.0.* NIST - National Institute of Standards and Technology - U.S. Department of Commerce, Dec 1998.

[5] Raj Jain. Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey. In *Computer Networks and ISDN Systems*, February 1995.

[6] Daniel Minoli and Thomas Golway. *Planning & Managing ATM Networks.* Ed. Manning, Feb 1997.

[7] E. Nordström, O. Gällmo, L. Asplund, M. Gustafsson, and B. Eriksson. Neural Networks for Admission Control in an ATM Network. In L.F. Niklasson and M.B. Bodén, editors, *Connectionism in a Broad Perspective: Selected Papers from the Swedish Conference on Connectionism - 1992*, pages 239–250. Ellis Horwood, 1994.

[8] Mauro Oliveira, Miguel Franklin, Adriano Nascimento, and Marcelo Vasconcelos. *Introdução à Gerência de Redes ATM.* Editora CEFET-CE, 2nd. edition, 1998.

[9] J. M. Pitts and J. A. Schormans. *Introduction to ATM Design and Performance.* John Wiley & Sons, 1996.

[10] University of Stuttgart - Institute for Parallel and Distributed High Performance Systems (IPVR). *The Stuttgart Neural Network Simulator - Version 4.0*, 1995.

[11] Tao Yang and Danny H. K. Tsang. A Novel Approach to Estimating the Cell Loss Probability in an ATM Multiplexer Loaded with Homogeneous On-Off Sources. *IEEE Transactions on Communications*, 43(1):117–126, Jan 1995.

[12] Tao Yang and Jun Yei. Optimal Solutions for a Dynamic Bandwidth Allocation Scheme in High-speed Networks. *Telecommunication Systems*, 5:389–412, 1996.