# STUDY OF THE APPLICATION OF NEURAL NETWORKS IN INTERNET TRAFFIC ENGINEERING

Nelson Piedra, Janneth Chicaiza, Jorge López, Jesús García

**Abstract:** *In this study, we showed various approachs implemented in Artificial Neural Networks for network resources management and Internet congestion control. Through a training process, Neural Networks can determine nonlinear relationships in a data set by associating the corresponding outputs to input patterns. Therefore, the application of these networks to Traffic Engineering can help achieve its general objective: "intelligent" agents or systems capable of adapting dataflow according to available resources. In this article, we analyze the opportunity and feasibility to apply Artificial Neural Networks to a number of tasks related to Traffic Engineering. In previous sections, we present the basics of each one of these disciplines, which are associated to Artificial Intelligence and Computer Networks respectively.*

**Keywords**: *Traffic Engineering, Artificial Neural Networks, Internet Traffic Engineering, Computer Networks.*

**ACM Classification Keywords**: *D.4.4 Communications Management*

**Conference**: *The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008*

## Introduction

The rapid expansion of the Internet regarding services, applications, coverage and users, has changed its traditional approach. A few years ago, the Internet was only a restricted means for data flow. Today, due to the liberalization, flexibility and easy access to Internet, the demand for the requirements of the applications has increased: better quality of service, higher bandwidth, less delay, better transmission quality, amongst others. This involves the research and develpment of more inventive solutions in order to provide a better quality of service for users.

One of the key factors that providers and users must face is the congestion in the service, which what causes undesirable consequences for both parts: loss of money and dissatisfaction for both. A quick solution to this situation is the increase in the capacity of the resources offered by those services. However, this is not an acceptable alternative because the use of the service is not constant and/or static and the budget of resources is limited. Besides, the distribution of data traffic is a stochastic process; therefore, during some periods there are low levels of activity or there is not any activity at all; thus, this capacity is subused.

To improve the performance of networks, we apply the principles, concepts and technologies of Traffic Engineering (TE); consequently, congestion is reduced, and traffic and resources are properly managed. The *Internet Engineering Task Force* (IETF) RFC 3272 describes the supports of *Internet Traffic Engineering* (ITE).

Due to their capacity and characteristics, *Artificial Neural Networks* (ANN) are being applied in various fields in which traditional methods and techniques have not efficiently solved underlying problems.

ANNs appeared with the purpose of emulating some characteristics of human beings, specifically, the capacity for memorizing, relating ideas and perform actions.

Through a training process, ANNs can determine nonlinear relationships in a data set by associating the corresponding outputs to input patterns. Therefore, the application of these networks to Traffic Engineering can help achieve its global objective: "intelligent" agents or systems capable of adapting dataflow according to available resources.

This document consists of three chapters. The first and second chapters deal with the fundamentals of Traffic Engineering and Artificial Neural Networks respectively. Later, in the third chapter, some experimental applications as well as the comparison of results of ANNs with other techniques for the implementation of specific traffic engineering functions are analyzed.

## 1. Traffic Engineering

The general objective of Traffic Engineering is to improve the performance of an operational network [Awduche et al., 2002]; consequently, reducing its congestion and increase the efficiency in using its resources [Delfino et al., 2006].

Trafic Engineering attempts to solve one of the main problems of IP networks: to adjust IP traffic flows to make a better use of bandwidth as well as send specific flows on specific paths too [Alcocer and García].

IETF has proposed several techniques to provide Quality of Service (QoS) on the Internet. Currently, IP networks have three significant characteristics: (1) they provide real-time services, (2) they have become mission critical, and (3) their operating environments are very dynamic [Awduche et al., 2002]. From this perspective, it is complex to model, analyze and solve problems related to maintenance, management and optimization of computers networks.

### 1.1. Concepts of Traffic Engineering

According to García (2002), Traffic Engineering can be defined as the process of controlling data flow through a network, that is, the process of optimizing the use of available resources from various flows and optimizing the global use of resources and benefits of the network [Xio et al., 1999] y [Xio et al., 2000] and [García et al., 2002]. Consequently, TE encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic.

Traffic Engineering deals with planning, control and network optimization with the purpose of achieve its goal: to adapt traffic flow to the physical network resources so that there are no congested resources whereas other resources are subused.

In IETF RFC 3272, the principles of *Internet Traffic Engineering* (ITE) are described, including aspects such as context, model and taxonomy. Moreover, there is a historical review, contemporary TE techniques and recommendations as well as other fundamental aspects.

According to [Awduche et al., 1999] and [Awduche et al., 2002], ITE deals with the management of the capacity of network traffic distribution, considering aspects such as evaluation and performance optimization of operational IP networks.

### 1.2. Causes of network congestion

From what Delfino (2006) [Delfino et al., 2006], network congestion can be caused by:

- Insufficient network resources (for example, link bandwidth or buffer space).
- Inefficient use of resources due to static traffic assignment to certain routes.

The first problem can be solved by increasing the capacity of resources. For the second problem, Traffic Engineering adapts traffic flows to physical network resources; thus, trying to optimally balance the use of these resources, so that there are no subused resources or over-utilized resources that cause bottlenecks. Solving congestion problems at reasonable costs is one of the main objectives of ITE.

When utilizing resources economically and reliably, we must consider requirements and performance metrics: delay, jitter, packet loss and throughput [Awduche et al., 2002]. The application of TE concepts to operational networks helps to identify and structure goals and priorities in terms of enhancing the quality of service. The application of traffic engineering concepts also aids in the measurement and analysis of the achievement of these goals. As a general rule, traffic engineering concepts and mechanisms must be sufficiently specific and well defined to address organizational requirements, but simultaneously flexible and extensible to accommodate unforeseen future demands.

### 1.3. Traffic Engineering Tasks

In [Villén-Altamirano], we can find the four major traffic engineering tasks and their recommendations:

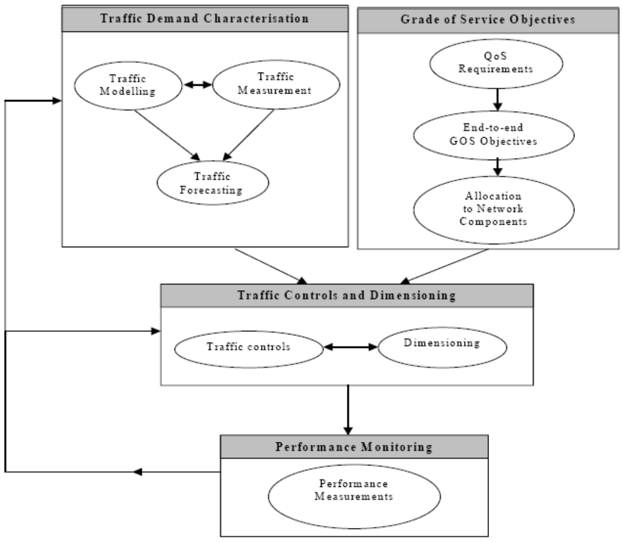


Figure 1. Traffic Engineering Tasks [Villén-Altamirano]

For modeling the complex behavior of the network, **traffic models**, we use the *Traffic Characterization* task. Using these models traffic demand is characterized by a limited set of parameters (mean, variance, index of dispersion of counts, etc). Only those parameters that are relevant to determine the impact of traffic demand on network performance. **Traffic forecasting** is also required for planning and dimensioning purposes. This is necessary to forecast traffic demands for the time period foreseen. In order to validate these models, **traffic measures** are used.

*GoS objectives* are derived from Quality of Service (QoS) requirements. Grade of Service is defined as "a number of TE parameters to provide a measure of adequacy of plant under specified conditions; these GOS parameters may be expressed as probability of blocking, probability of delay, etc".

TE must provide a design and operation of the network that guarantees the support of the traffic demand as well as the achievement of GoS objectives. Thus, **network dimensioning** (of the physical and logical network) assures that the network has enough resources to attend the traffic demand. Among the **traffic controls** we can distinguish: traffic routing, network traffic management controls, service protection methods, packet-level traffic controls, and signaling and intelligent network controls.

Although the *network performance monitoring* it can be correctly dimensioned. GoS monitoring is needed to detect errors or incorrect approximations in the dimensioning and to produce feedback for traffic characterization and network design.

### 1.4. Historical Review and Recent Developments

The first routing algorithms tried to minimize the use of network resources by choosing the shortest path, but this selection criterion can cause congestion in some network links whereas other links could be infra-utilized [García et al., 2002]. When applying TE concepts, some flows could go through other links with less traffic even if they are on a longer route (Figure 2).

Currently, MPLS (Multi Protocol Label Switching), is highly regarded as the proper technology to provide capacity for Traffic Engineering and QoS, -especially for backbone applications- [Sawant and Qaddour]. Among other aspects, MPLS offers: resources reservations, fault tolerance and resource optimization. The combination of

MPLS and DiffServ-TE (Differentiated Services for Traffic Engineering) has advantages to provide QoS while the utilization of network resources is optimized [Minei, 2004]. Among the characteristics of MPLS to provide TE, we have [Roca et al.]:

- Establishing explicit routes (physical path at LSP -Label Switched Path- level).
- Generating statistics regarding the use of LSPs. This information could be used for network planning and optimizing.
- Flexibility in network administration. Constraint-Based Routing can be applied so that routes for certain QoS or special services can be selected.

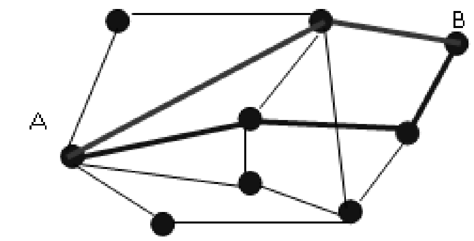


Figure 2. Routing of Packages by means of IGP and MPLS [Roca et al.]

Besides MPLS and DiffServ, other approaches have been proposed or implemented to offer TE. Some routing approaches, used a few years ago, are described in [Awduche et al., 2002].

- It is known that Internet evolved from ARPANET and adopted dynamic routing algorithms with distributed control to determine the routes that packets should take to reach their destination. This type of algorithms are adaptations of shorter path algorithms where costs are besed on link metrics. One of the weaknesses of using link metrics is that unbalanced loads in the network can occur. “In ARPANET, packets were forwarded to their destination along a path for which the total estimated transit time was the smallest”. This approach is known as Adaptive Routing, where routing decisions were based on the current state of the network in terms of delay and connectivity. One inconvenient of this approach is that it can cause congestion in different segments of the network; thus, resulting in network oscillation and instability.
- Type-of-Service (ToS) routing involves different routes going to the same destination with selection dependent upon the ToS field of an IP packet. A separate shortest path tree is computed for each ToS. Classical ToS-based routing is has been updated outdated and the ToS field has been replaced by a Diffserv field. The Diffserv model essentially deals with traffic management on a per hop basis.
- “SPF is modified slightly in ECMP (Equal Cost Multi-Path) so that if two or more equal cost shortest paths exist between two nodes, the traffic between the nodes is distributed among the multiple equal-cost paths”. Thus, it is possible that one of the paths will be more congested than the other.
- Nimrod is “a routing system developed to provide heterogeneous service specific routing in the Internet, while taking multiple constraints into account (RFC, 1992)”. Essentially, Nimrod is a link state routing protocol with mechanisms that allow restriction of the distribution of routing information. “Even though Nimrod did not enjoy deployment in the public Internet, a number of key concepts incorporated into the Nimrod architecture, such as explicit routing which allows selection of paths at originating nodes”.
- The overlay model using IP over ATM requires the management of two separate networks with different technologies (IP and ATM) resulting in increased operational complexity and cost. “The overlay model based on ATM or frame relay enables a network administrator or an automaton to employ traffic engineering concepts to perform path optimization by re-configuring or rearranging the virtual circuits so that a virtual

circuit on a congested or sub-optimal physical link can be re-routed to a less congested or more optimal one".

- In Constrained-Based Routing (CBR), the network administrator can select certain paths for special services with different quality levels (explicit delay guarantees, bandwidth, fluctuation, packet loss, etc). CBR can compute routes subject to the satisfaction of a set of constraints (bandwidth, administrative policies, etc), that is, this procedure considers parameters beyond the network topology in order to compute the most convenient route.

  "Path oriented technologies such as MPLS have made constraint-based routing feasible and attractive in public IP networks". CBR, MPLS and TE in IP networks are defined in RFC 2702.

- As said, MPLS is used to provide TE. Today, there is a wide variety of protocols used for the distribution of labels. MPLS architecture does not specify one of these protocols, but recommends their choice depending on the specific network requirements. The protocoles used can be grouped into two classes: explicit routing protocols and implicit routing protocols. Explicit routing is suitable to offer traffic engineering and allows the creation of tunnels. On the other hand, implicit routing allows establishing LSPs but does not offer traffic engineering characteristics [Sienra, 2003].

Among the most common routing protocols we have; the Constraint-based Routing Label Distribution Protocol (CR-LDP) and the Resource Reservation Protocol-Traffic Engineering (RSVP-TE). CR-LDP is an extension of the LDP, which is an implicit routing protocol, sets up a determined path in advance, that is, LSPs will be established with MPLS Quality of Service. CR-LDP is a solid-state protocol, in other words, after establisshing the connection, this connection remains "open" until it is closed. The operation of RSVP-TE is similar to that of CR-LDP, since it sets up a point-to-point LSP that guarantees an end-to-end service. The difference is that RSVP-TE requires periodic refreshment of the route to remain active (soft state). With these last protocols and the application of various traffic engineering strategies, it is possible to assign different quality of service levels in MPLS networks.

### 1.5. Recommendations for Internet Traffic Engineering

In [Villén-Altamirano] some recommendations for Traffic Engineering are proposed. They are classified according to their major tasks.

RFC3272 [Awduche et al., 2002] describes high-level functional and non-fuctional recommendations for ITE. Functional recommendations are necessary to achieve TE objectives and non-functional recommendations are related to quality attributes or state characteristics of a TE system.

Likewise, in [Feamster et al., 2003], there are some guidelines to provide traffic engineering between domains, more specifically; some approaches of BGP (Border Gateway Protocol) are discussed.This protocol by itself does not facilitate common TE tasks.

## 2. Artificial Neuronal Networks

The idea of Artificial Neuronal Networks (NNA) was conceived originally as a try for modeling the bio—physiology in the human brain; this is, to understand and explain how the brain works. The aim was to create a model capable to emulate the human process for reasoning. Most part of the starting works in neuronal networks was done by physiologists but not by engineers [TRECSoluciones, 1995].

Since Santiago Ramón y Cajal discovered the neuronal structure in the nervous system, many contributions have tried to "reproduce" or at less imitate in a "litte scale" the way the human brain works.; in this context, in 1943, Walter Pitts and Warren McCulloch, proposed a mathematical model of neuron which explains the way that those processing units work.

In 1949, the physiologist Donald Hebb pointed out in his book "*The Organization of Behavior*" the learning rule known as *Rule of Hebb*. His proposal had relation with synapses conductivity, or with neurons connections. Hebb showed that the repeated activation in a neuron for other through a established synapses, increases its conductivity and made it more alike to be active successively, inducing to the formation of a neuronal circuit strongly connected.

In the summer of 1951, Minsky and Edmons made the first neuronal networks machine which consisted basically of 300 empty tubes and an automatic pilot from a B-24. They called their creation “Sharc”; it was a network with 40 artificial neurons which imitated a rat’s brain.

In 1957, Frank Rosenblatt presented the Perceptron, a neuronal network with supervised learning which learning rule was a modification to the Hebb’s proposal.

Almost one decade later, in 1969, Marvin Minsky and Seymour Paper wrote a book called “*Perceptrons*”, in which they probed the limitations of perceptrons in solving problems relatively easy; when they published the book, all the research about perceptrons were suspended and annulled.

In the 60´s other two supervised models were proposed, based in the Perceptron of Rosenblatt called Adaline and Madaline. In those cases, the adaptation of the weights was done taking into account the error, calculated as the difference between the wished output and the one given by the network, similar to the perceptron, nevertheless, the learning rule used is different.

The modern age for ANN surges with the backpropagation learning technique. In 1977, James Anderson developed a lineal model, called Lineal Associator, which consisted of some lineal integrators elements (neurons) which added their inputs. In 1982, John Hopfield showed a work on neuronal networks in the National Sciences Academy; which describes clearly and with mathematical rigor a network which was give his name, and is a variation from the Lineal Associator. Also, in this year, Fujitsu Enterprise started the development of thinking computers for application in robotic.

The 80´s decade was overpowering for spreading the ANN, as some non supervised and hybrid models and more developed kind of networks were proposed. Nowadays, many works show their successful application in different non lineal problems, which have not been modeled using traditional methods such as Statistics, Operations Research and others.

### 2.1. Structure and Functioning

A Biological Neuronal Network (brain) is constituted by a series of interconnected elements, called neurons, which operate in parallel. It has been estimated that in our brain there are around 100 thousand million neurons and more than 100 billion of connections (synapses).

Neurons, as the other cells in the body, work through electric impulses and chemical reactions. The electric impulses that a neuron uses to exchange information with other neurons in a network go through the axon which makes contact with the dendrites in the next neuron through the synapses. The intensity in the signal — synaptic weight- transmitted depends in the efficiency of the synaptic transmission. The signal transmitted to the neuron can be inhibitor or stimulator. The neuron shoots, or sends the impulse through its axon, if the stimulation exceeds its inhibition by a critic value — neuron threshold- [TRECSoluciones, 1995].

### 2.2. Elements of Artificial Neuron

Following, is presented the basic structure of the artificial neuron.

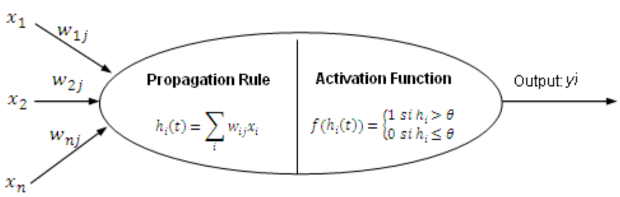


Figure 3. Elements of Artificial Neuron

- Xj, neurons inputs.
- Wij (weights) are coefficients which can be adapted inside the network. They determine the intensity in the input signal registered.
- Propagation function. Allowing obtaining, from the inputs and the weights the value of the post- synaptic potential of the neuron (*hi*). The most common function is the pondered addition of all the inputs (Figure 3). However, the propagation function can be more complex than just products addition.
- Activation or Transference Function. The result of the propagation function is transformed in the real output of the neuron through an algorithmic process known as activation function.

There are some activation functions to determine the neuron's output; for example, when the output value in the neuron is compared with a threshold value; if the addition is higher than the threshold value, the neuron will generate a signal; if the addition is lower than the threshold value, none signal will be generated; this function is called *heaviside*. It also can be used the lineal, sigmoid, hyperbolic tangent and others function. Particularly the sigmoid one works quite well and is normally the most common.

- Yj, neuron output.

A more complete artificial neuron model includes other elements such as: an output function, which is applied after the activation function is calculated; in most cases the identity function is used, therefore, it is not part of the basic figure presented.

## 2.3. Training of artificial neural networks

Every learning process, has two phases; the training one and the testing one; in both cases, we supply the ANN with a series of prototypes or cases, this knowledge is which allows the network to learn from the experience; in the case of supervised models, the network get its errors comparing the calculated value and the desired value. When there is a difference between those two values, the learning rules are applied to modify the weights in the ANN, until minimize the global error or any other cost function. On the other hand, in unsupervised models (or self-organized), the desired output is not known; in this case the network must organize itself to find common characteristics in the training data.

An additional element that must be established in the training phase is the learning rate. The learning rate in the ANN depends of different controllable factors which must be taken into account. Obviously, a low value in the learning rate means more time for training in order to produce a well trained ANN. With higher learning values, the network could not be able to discriminate in the same way that a system that learns slower does. Generally, additional factors -apart of time- must be considered when discussing the training off-line:

- Network Complexity: size, paradigm, architecture.
- Type of learning algorithm
- Error allowed in the final network

If changing any of those factors the training time can increase to an elevated value or obtain an unacceptable error.

## 2.4. ANN Architecture and Topology

The ANN topology is determined for the neurons organization and their disposition in the network. One layer is a inter-connected neurons set, most of connections happen between neurons in adjacent layers.

Therefore, the collection of parameters that define an ANN architecture are: number of layers, generally one input layer and one output layer and 0 or more intermediate (hidden); the number of neurons by layer, one or more; and the connectivity grade between the neurons, which is the number of connections between the neurons in different layers or between neurons in the same layer. In the Figure 4, it is described the architecture of a more used network called Feedward [Pizarro].

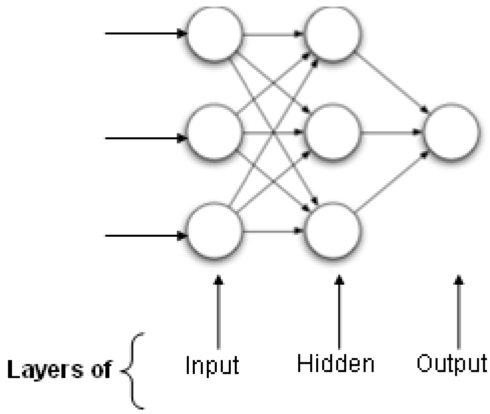


Figure 4. Multilayer Perceptron Estructure

### 2.5. Evaluation of the Neuronal Network to be used

The model of ANN to be used, can be selected according to:

- The number of layers, the ANN can be Monolayer —one input layer and one output layer- or Multilayer, generalization of the last one, which are added intermediate layers (hidden) between the input and the output.
- The connection type, the ANN can be: Feedforward, if the signal propagation is produced in just one way, therefore, they do not have a memory. And Recurrent if they keep feedback links between neurons in different layers, neurons in the same layer or in the same neuron.
- The connection grade, They can be: Totally connected, in the case where all the neurons in a layer are connected with the neurons in the next layer (feedforward networks) or with the neurons in the last layer (recurrent networks); and Partially connected networks, in the case when there is not total connection between neurons from different layers [Soria].
- The learning paradigm, networks can be supervised or unsupervised (or hybrid), which basic functions were described before.

Between the main neuronal models which combine the networks types mentioned before, there are:

- Perceptron, is a supervised network, monolayer, feedforward and is the base for the most of the a architecture of the ANN which interconnect between their selves.
- Backpropagation, as the perceptron, the backpropagation network uses supervised learning; however, this one is multilayer. The importance of this network is its generalization capacity or produce satisfactory outputs for inputs that the system has never seen before during its training phase.
- Self-organized maps, they constitute a practice of unsupervised learning and competitive; it considers that the influence that a neuron exercises on the others is a function of the distance between them. They can be applied to cover two basic functionalities; as classificatory or to represent multidimensional data in less dimension spaces (normally one or two dimensions), preserving the topology from the input.

Once presented the fundaments and models of the most important neuronal networks, following will be presented some successful applications in the Traffic Engineering field.

## 3. Applications of Neural Networks in Internet Traffic Engineering

Below we mention some characteristics of Artificial Neural Networks that can be crucial when applying them in areas such as Internet Traffic Engineering:

- ANNs, through a training process, are capable of determine nonlinear relationships in a data set by associating the corresponding output or outputs to input patterns. Consequently, many ANN models are used for determining forecasts based on a data source. This characteristic can be used for making predictions. For example, to determine the available bandwidth, detect traffic congestion patterns, forecast the use of resources (for instance, links) and even to establish or improve routing algorithms and, in general, to apply it to the tasks related to TE.

- The types of learning available for some models are batch learning (off-line) and on-line learning. They can be used for forecasting and classification depending on the data available and the available processing capacity. On-line learning is usually used in those problems in which there are a lot of training patterns. With these capacities trace files generated by some devices and network applications could be processed (in real-time or off-line); thus, facilitating TE tasks such as traffic modeling, control optimization and network dimensioning.
- Supervised Models such as the Multilayer Perceptron through the backpropagation algorithm or Adaline; or unsupervised models such as Kohonen Maps (due to their capacity for memorizing patterns) can be applied to extract or eliminate noise in signals.
- A neural network considers changes in the environment and can adapt itself to these changes, that is, once the network has been trained and tested, it will be capable of establishing the learned relationships on a new data set.
- An ANN-based approach can learn specific models from each network system and provide acceptable solutions of the underlying real systems.

Now, we will mention some characteristics of the tasks to be performed by Traffic Engineering (associated to the processes in Figure 1). Later, some projects of ANN applications in this area will be described.

- Measurement and network performance forecasting. The use of shared network resources and bandwidth are dynamic [Eswaradass et al., 2005]. Therefore, a bandwidth forecast is a very complicated task for being approached with traditional methods such as Statistics.
- Network systems modeling is a complicated tasks that can be solved trough neural networks (network traffic is nonlinear and very difficult to model and predict). In addition, traffic statistics of various applications show that each type of traffic presents a different traffic patter. By using a neural network, we can characterize the heterogeneous nature of changes in network traffic [Eswaradass et al., 2005].
- Network planning. Since a neural network is capable of establishing patterns that model traffic nature, it will also be capable of establishing mechanisms for network planning by providing guides to adapt traffic flow to physical network resources (so that there are no congested resources where as others are underutilized, this is a Traffic Engineering objective).

### 3.1. Bandwidth Forecasting

There are some methodologies and tools for estimating bandwidth capacity and availability respectively (some of them are mentioned in the Eswaradass', Sun and Wu job). However, they do not provide complete metrics; for instance, they do not predict bandwidth. Due to the heterogeneous and dynamic nature of network traffic, there are a few available works to predict network performance in terms of available bandwidth and lantency [Eswaradass et al., 2005].

In [Eswaradass et al., 2005], an available bandwidth forecasting method is proposed, this approach is based on Artificial Neural Networks. The prediction must consider various network applications (TCP, UDP, ICPM and others). This system has been tested on traditional trace files and compared to a system known as NWS (Network Weather Service, a model that is widely used for prediction). The experimental results showed that the neural networks approach always provides a better prediction (more precision based on the minimum global error) on NWS systems.

Predictions have been made by making an ANN for each type of network traffic, integrating partial results to obtain global predictions. Besides, noise and performance predictions are categorized after noise reduction.

In Table I, some details about the model are shown, according to [Eswaradass et al., 2005]. Although, it is not specified in the document, we can conclude that the network used (due to the description of the solution) is the Multilayer Perceptron, to which the real bandwidth value has been provided and the adjustment of its is based on the network error calculations.

Table I
Description of the Neural Network Model

| Configuration Parameters | | |
|---|---|---|
| Parameter | Description | Value |
| Learning rate | Determines the network learning rate | 0.01 |
| Number of epochs | Indicates the number of times a data set is trained | 700 |
| *Network Architecture* | | |
| Layer type | Description | |
| Input layer | Depends upon the number of selected parameters: timestamp, average packet rate and average bit rate (in this case 3). | |
| Hidden layer | 3 hidden layers and 3 perceptrons in each layer. The nonlinear sigmoid function is used as an activating function. | |
| Output layer | Available bandwidth/minute | |
| *Training Patterns.* As input data for the training process, trace files generated in the University of Auckland have been used. These historic files have been previously pre-processed and contain the record of time and network traffic — of different types: TCP, ICMP and UDP-. Each trace log contains incoming packet (timestamp, packet length, source and destination IP addresses). According to [Eswaradass et al., 2005], the number of packets in each second and the number of bits in each second are sufficient to produce estimates of the consumed bandwidth over time. | | |
| *Cost function.* The metric used for evaluation is the relative prediction error, err.*err = P-redictedValue-ActualValue ActualValue.* PredictedValue is the bandwidth predicted for the next n seconds and ActualValue is the bandwidth measured for the next n seconds. Mean error, which is calculated by averaging all of the relative errors, is used as the cost function to be minimized. | | |
| *Simulation Software.* For the simulation of the network model WEKA has been used, which is a free software package that offers a collection of various algorithms for solving data mining, including ANN. | | |

*1) Implementation in ANNs:* Predictions have been made by making an ANN for each type of network traffic, integrating partial results to obtain global predictions. Besides, noise and performance predictions are categorized after noise reduction.

In Table I, some details about the model are shown, according to [Eswaradass et al., 2005]. Although, it is not specified in the document, we can conclude that the network used (due to the description of the solution) is the Multilayer Perceptron, to which the real bandwidth value has been provided and the adjustment of its is based on the network error calculations.

*2) Discussion on the Problems and Strengths of the ANN-based approach:* Below we discuss some problems, strengths and future works of the neural networks-based approach.

- With more parameters and input data, the accuracy of the results is better. However, the increase of parameters and input data will increase prediction time and network training [Eswaradass et al., 2005]. Therefore, trace files must be analyzed in order to identify small data sets and input parameters.
- The selection of parameters can be done with the technique known as analysis of main components, which is implemented through a unsupervised network model, that is, all the components of an input pattern —or many parameters of trace files- could be provided for the unsupervised ANN; finally, we will get only more important parameters for forecasting.
- One problem to be solved is the selection of a proper training set. According to [Eswaradass et al., 2005], "the prediction performance with an ANN is not satisfactory for short-term trace files, which contain data for a

couple of hours or less than 1 day", or files containing traffic data more than 3 weeks. On the contrary, "network traffic data in 7-10 days is enough for neural network training".

- The construction cost of an ANN in general "is greater than those prediction systems that use linear prediction models".
- The ANN-based prediction mechanism is viable and practical. It can be used as an only prediction component or can be incorporated into the NWS for a better network prediction.
- This approach uses batch learning, that is, considers historic trace files as training patters. The next step is to provide run-time prediction. For this, the on-line processing algorithm should be used.

Table II summarizes the experimental results of the prediction performed with the data recorded at the University of Auckland uplink. AUCKLAND II is a collection of 1-day trace between December 1999 and June 2000. AUCKLAND IV is a single trace that contains data of the traffic reported between February and April 2001 (6 1/2 week trace). As seen, ANN-based prediction is the most accurate in all cases. The most reduced error percentage (5% for daily traces) occurs when a separated prediction for each type of traffic is done. Consequently, if the trace file would contain traffic data of a single application, the prediction could be even more accurate.

Table II

Global error reduction percentage for NWS and ANN

| | Original Prediction* | Before noise reduction** | After noise reduction*** |
|---|---|---|---|
| AUCKLAND IV | 1.39% | 2.33% | 3.14% |
| AUCKLAND II | 2.49% | 3.68% | 5% |

* Prediction performed considering the various traffic flows as a whole

** Prediction performed after separating the various types of traffic

***Results after removing ICMP and UDP, only keeping TCP, which is the dominant constituent of the network traffic (95%).

### 3.2. Classification of Internet Traffic

The classification of Internet traffic can be used for differentiating services or for applying network security schemes. The traditional classification is usually done using the packet header field of 'port number", the layer 4 header (TCP/UDP). However, the use of this number could be unreliable in the classification of Internet traffic given the nature and characteristics of this network: free. Therefore, it is not mandatory that these applications use specific port numbers [Li et al., 2000] in [Trussell et al., 2005].

In [Trussell et al., 2005], a classification and estimation method of traffic intensity in an application is proposed. This method is based on the size distribution of packets registered in a switch (or router) during a short period time, identifying flows with significant quantity of time-sensitive data, such as voice over IP or real-time video. A switch (or router) can give preference to these flows, thus, being a mechanism to increase the Quality of Service (QoS).

As said, packet size distribution, as part of the characteristics of an application, is used as an indicator of application type. The distribution data can be obtained from the IP packet (layer 3) in order to avoid accessing the TCP header, which takes additional time and computation [Trussell et al., 2005].

*1) Comparing MMSE, POCS and ANNs:* In [Trussell et al.,2005], three methods for estimation of the traffic are compared: CLLSQ (Constrained Least SQuares), POCS (Projections Onto Convex Sets) and Neural Networks. According to this document, methods that use ANNs performed best in the tests. The detection of several significant classes can be done reliably. Below we describe some details of the project:

Table III

Details of the Project [Trussell et al., 2005]

| |
|---|
| *Training Data.* The data for the ANN training are collected from the North Carolina University backbone network using a tool for analysis of network traffic, named TCPDUMP. The data was collected continuously for four hours. The recorded parameters are: source port number, destination port number, packet size. “The applications were identified using the source and destination port numbers depending on the port assignments by IANA” (Internet Assigned Numbers Authority). |
| *Histogram Generation.* “In order to reduce the dimensionality of the data, the Ethernet packet sizes range from 60-1514 bytes were considered (some of them divided into a manageable number of bins)”. |
| *Clustering.* “To verify the conjecture that applications could be reliably characterized by their histograms”, the histogram collection using several clustering methods was analyzed, “which all resulted in natural groupings of the histograms of applications”. |
| *Estimation an Detection.* “The total distribution of packet sizes at a particular network node is the mixture of the distribution of the individual applications. Therefore, we can model the total network traffic as the linear combination of major applications”. |

“The architecture used for the neural networks was a simple single hidden layer with a single output neuron”, the activation function for hidden layer is the log-sigmoid. For estimation, the output neuron used a linear function; while for the detection case, the output neuron used a log-sigmoid function”. According to [Trussell et al., 2005], in the case of estimation, it was determined that using six neurons in the hidden layer is appropriate to model the problem “In the case of detection, it was found that two hidden neurons were sufficient to give good results”. In this document, no method is indicated to establish the number of hidden neurons that should be used in every layer. Consequently, a simulation using software tools and the “trial and error” are required to determine this data.

The result of estimation performance is given in Table IV. The RMS (Root Mean Square) error obtained by the ANN is inferior than the other methods. “This result is obtained by training on one set of 24 samples and testing on the other set . If the estimation was limited to the percentage of a single application, all methods improve” and, as in the previous case, the ANN performs best.

Table IV

RMS error [Trussell et al., 2005]

| Application | Average | Error RMS | | |
|---|---|---|---|---|
| | | CLLSQ | POCS | ANN |
| RTP | 0.0119 | 0.0010 | 0.0029 | 0.0004 |
| Napster | 0.0111 | 0.0016 | 0.0013 | 0.0001 |
| eDonkey | 0.0097 | 0.0052 | 0.0010 | 0.0002 |

*2) Estimation of the presence of a single application:* To estimate the probability of a specific application being present in the traffic flow a neural network was used. “Since the original data contains most applications in each data set, to test detection, we created artificial data sets, based on actual data files”. According to [Trussell et al., 2005], the method obtains a very high accuracy of detecting the presence of specific applications, even at low percentages. In some applications, there is a lower detection rate due to the fact that these applications have statistical properties that are similar to other applications (in this case eDonkey).

An strengths of the ANN approach is that “will allow the reduction of the size of the histograms and a corresponding decrease in computation time† Very small weight on a particular bin of input vectors for all neurons indicates that this bin is not needed for estimation or detection”.

### 3.3. Overload Control in Computer Networks

Neural Networks can also be used for controlling overload in Computer Networks. In [Wu and Michael], a supervised network model capable of learning control actions based on historical records. The result, according to this, is a control system that is simple, robust and near-optimal.

Guaranteeing good performances of overload control systems is essential. Therefore, control actions are required to protect network resources from excessive loads. These actions must be based on mechanisms that regulate new arriving requests.

According to [Wu and Michael], there are two kinds of control strategies, namely, local or centralized; according to the amount of information the control decisions are based.

As known, "traffic is stochastic and the mapping from traffic to optimal decisions is complex. To solve this problem, ANN can be used, "bearing in mind its ability of learning unknown functions from a large number of examples and its implementation in real time once being trained". The first step is to "generate examples for the training the network. The second step is to train a group of neurons based on these data. After training, the neurons cooperate to infer the control decisions based on locally available information".

*1) Requirements for Implementing Overload Control*: A network device is "overloaded if its work load averaged over a period exceeds a predefined threshold. Overload control can be implemented by gating new calls. The gate values, i.e., the fraction of admitted calls, are updated periodically. An effective control is to find out the optimal gate values for each period". In [Wu], five requirements are described and than an ideal control algorithm should satisfy.

*2) Solution using Neural Networks:* The network inputs are parameters about requests to a network device and output corresponding control decisions accordding to maximum value allowed. The input-output mapping is reached through learning process using examples generated by CCM (Centralized Control Method). "It is difficult to train the neural networks properly using examples generated for a large range of traffic intensity, but on the other hand, training them at a fixed traffic intensity makes them inflexible to changes"; thus, losing generalization performance. Hence, for each network device, a group of neural networks was built; "each member being a single layer perceptron trained using examples generated at particular background traffic intensity".

In [Wu and Michael], the training of a member of the group of neural networks is explained. This training is similar to that of a back-propagation network in output signals and the calculation of the mean square error. "Each hidden unit is trained at a particular traffic intensity".

Wu's approach compares the CCM, LCM and ANN methods. To obtain results, they performed simulations on part of the Hong Kong metropolitan network. Call attempts (call arrival rates between different nodes) "were generated according to the Poisson process, and accepted with probability given by the corresponding gate values".

Finally, the results prove that ANN "has a throughput higher than CCM, moreover, decreases the time for making decisions (about 10% of the CPU time of CCM); thus, NNM can be implemented in real time".

### 3.4. Fault Diagnosis

In Computer Networks, the proper management of error messages can facilitate fault diagnosis in a system. For example, when occur a network breakdown, a lot of error messages are generated, making it difficult to differentiate the primary sources and secondary consequences of a problem. Thus, it is desirable to have an efficient and reliable error message classifier [Wu and Michael].

Several learning machine-based algorithms can be used for classification tasks, such as, decision rules, nearest neighbor-based, tress, and more; nevertheless, they do not support a high level of "noisy and ambiguous features inherent in many diagnosis tasks". Thus, ANN can approximate highly nonlinear functions with a high precision.

In [Wu and Michael] we can see that the hybrid classifier is composed of an input layer, a hidden layer that contains R nodes representing classification rule vectors and a perceptron output layer. The approach is based on a competitive network model called winner-take-all.

In this case, el training set was a collection of error messages generated from a telephone exchange computer. The training set consists of 442 samples and the test set of 112 samples. As mentioned in previous cases the ANN-based approach yields better results than the other options analyzed.

## Conclusion

The Traffic Engineering in order to reach its aim of improving the performance of an operational networkwork, minimizing the congestion of resources and the effective use of them, must take into account the different requirements and metrics of performance, mechanisms and politics that improve the integrity and reliability of the network [Awduche et al., 2002] covering aspects like: characterization of the traffic demand, planning, control and optimization of the network.

Nowadays, publications, studies, applications and efforts related to NNA are considerable, despite its complexity. There are different simulation tools that can facilitate its comprehension and results verification. According to [Werbos, 1998] y [Brio and Sanz, 2001], can be considered that the application of neuronal networks have reached their maturity.

The application of the ANN in traffic engineering is quite promising. As Del Brio [Brio and Sanz, 2001] points out, the characteristics which make that a specific situation to be an ideal candidate for NNA application are the following, which are massively present in all the traffic engineering problems.

- There is not a method which describes the problem completely; therefore, modelling it becomes a complex task.
- To have an important amount of data, which will serve as examples or patterns for the learning of the network; the data related to the problem is imprecise or include noise; the problem is high dimensionality.
- In changing working conditions, The NNA can adapt their selves perfectly due to its adapting capacity (re-training).

There are different proposes that have shown a potential application of the Artificial Neuronal Networks in the Communication Networks field; in this work, applications in specific tasks of the Traffic Engineering have been shown, such as: prediction, control, monitoring and resources performance. Have been seen some approximations for prediction of bandwidth prediction [Eswaradass et al., 2005] overcharge control [Wu and Michael], traffic classification [Trussell et al., 2005] and diagnosis of error messages [Wu and Michael].

Due to the own characteristics of network traffic, the application of the methods and conventional statistics techniques is not appropriate to provide optimal predictions; On the other hand, the experimental results provided by NNA models demonstrate that those tools offer best predictions - minimal error — in contrast with other systems.

For data prediction tasks, in general, different models can be used: deterministic, statistical, probabilistic, and based on machine learning; each model has its own strengths and weaknesses. Real problems can be disarranged in different modules, each one implemented with different techniques; it implies, depending of the problem characteristic and requirements, the best technique can be selected or use hybrid models to obtain better results.

The described works for NNA application in Traffic Engineering have in common a pre-processing phase, in which, the incoming data are treated, depurated and selected, before being processed by the neurons of the NNA; this phase can be the most extensive and determine extensively the success in the realization of other parts of the project, helping to control risks, to reach a maximum performance and avoid mistaken conclusions.

## Bibliography

[Alcocer and García] Alcocer, F. and García, J. Curso de Teleeducación sobre MPLS.

[Awduche et al., 1999] Awduche, D., Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and McManus, J. Requirements for traffic engineering over MPLS. Technical report, IETF. 1999.

[Awduche et al., 2002] Awduche, D., Chiu, A., Elwalid, A., and Widjaja, I. Overview and principles of Internet Traffic Engineering. Technical Report, IETF. 2002.

[Brio and Sanz, 2001] Brio, M. D. and Sanz, A. 2001. Redes Neuronales y Sistemas Difusos. 2da. edition.

[Delfino et al., 2006] Delfino, A., Rivero, S., and SanMartín, M. Ingeniería de tráfico en Redes MPLS. Technical report, Congreso Regional de Telecomunicaciones. 2006.

[Eswaradass et al., 2005] Eswaradass, A., Sun, X., and Wu, M. A neural network based predictive mechanism for available bandwidth. The ACM Digital Library. 2005

[Feamster et al., 2003] Feamster, N., Borkenhagen, J., and Rexford, J. 2003. Guidelines for Internet Traffic Engineering. ACM SIGCOMM Computer Communication Review.

[García et al., 2002] García, J., Raya, J., and Raya, V. Alta velocidad y calidad de servicio en Redes IP. 2002.

[Li et al., 2000] Li, F., Seddigh, N., Nandy, B., and Malute, D. 2000. An empirical study of today's Internet Traffic for Differentiated Services IP QoS.

[Minei, 2004] Minei, I. MPLS Diffserv-aware Traffic Engineering. Technical report, Juniper Networks Inc. 2004.

[Pizarro] Pizarro, F. El paradigma de las Redes Neuronales Artificiales. Technical report, Departamento de Informática Tributaria de España.

[Roca et al.] Roca, T., Chica, P., and Muñoz, M. Ingeniería de Redes. trabajo sobre mpls.

[Sawant and Qaddour] Sawant, A. and Qaddour, J. Mpls diffserv: a combined approach. Illinois State University.

[Sienra, 2003] Sienra, L. Ofreciendo Calidad de Servicio mediante MPLS. Centro de Investigación e Innovación en Telecomunicaciones (CINIT). 2003.

[Soria]. [TRECSoluciones, 1995] Soria, E. Redes neuronales artificiales. TRECSoluciones 1995. Redes neuronales artificiales.

[Trussell et al. (2005)] Trussell, H., Nilsson, A., Patel, P., and Wang, Y. Characterization, Estimation and Detection of Network Application Traffic. North Carolina State University Raleigh. 2005.

[Villén-Altamirano] Villén-Altamirano, M. Overview of itu recommendations on traffic engineering. Department of Computer Science of University of Cyprus.

[Werbos, 1998] Werbos, P. Neural Networks combating Fragmentation. IEEE Spectrum Magazine. 1998.

[Wu and Michael], Wu, S. and Michael, K. Neural networks: Techniques and applications in Telecommunications Systems. The Honk Kong University of Science and Technology.

[Xio et al., 2000] Xio, X., Hannan, A., Bailey, B., and Ni, L. Traffic Engineering with MPLS in the Internet. IEEE Network Magazine. 2000.

[Xio et al., 1999] Xio, X., Irpan, T., Hannan, A., Tsay, R., and Ni, L. Traffic Enginnering with MPLS, America's Network Magazine. 1999.

## Authors' Information

*Nelson Piedra - Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación - UTPL, Ecuador, nopiedra@utpl.edu.ec*

*Janneth Chicaiza - UTPL - Unidad de Proyectos y Sistemas Informáticos, Ecuador, jachicaiza@utpl.edu.ec*

*Jorge López - UTPL - Unidad de Proyectos y Sistemas Informáticos, Ecuador, jalopez2@utpl.edu.ec*

*Jesús García Tomás - Universidad Politécnica de Madrid - UPM, Facultad de Informática, España, jgarcia@fi.upm.es*