IP Bandwidth Allocation Management using Agents and Neural Network Approach

Danielo G. Gomes*, Nazim Agoulmine*, and José N. de Souza#1

* Laboratory of Complex Systems – LSC, University of Evry, France
#Department of Electrical Engineering, University of Ceará, Brazil

1 Danielo G. Gomes and Nazim Agoulmine are with the Laboratory of Complex Systems – LSC, University of Evry, 40, Rue du Pelvoux – CE 1455 Courcouronnes 91020 Evry Cedex, France, e-mail: {dgomes,nazim}@iup.univ-evry.fr, José N. de Souza is with the Department of Electrical Engineering, University of Ceará – UFC, Campus do Pici – Bloco 705, 60455-760, Fortaleza, CE, Brazil, e-mail: neuman@ufc.br
Abstract— Video and LAN traffic can be modeled as self-similar processes, whereas Internet traffic can be modeled by multifractal processes. The Hurst parameter is a measure of the self-similarity of a process. The objective of this work is to use this characteristic of Internet traffic in order to allow future Video on Demand Service Providers (VDSP) to optimize their bandwidth utilization and consequently their communication cost. The work addresses one aspect of a global project that specify an intelligent agent architecture to manage the relationship between VDSP, ISPs and end customers. In this paper, we address the egress traffic aspect of the VDSP and propose a neuronal network approach to allow a VDSP agent to estimate the nature of the VDSP egress traffic using the Hurst parameter. This approach is evaluated against statistical estimators.

I. INTRODUCTION

With the explosive growth of the Internet and of private networks attached to the Internet, a large number of new demands arises. Low-volume Telnet conversations are replaced by high-volume Web traffic, which is increasingly graphics intensive. At the same time, real-time applications demand an even quicker network response. In order to deal with these demands, it is not enough to increase internet capacity and accuracy methods for managing the traffic are needed. Several studies have claimed that different types of network can be accurately modeled using a self-similar process. A self-similar process is able to capture the long-range dependence (LRD) phenomenon exhibited by such traffic. Moreover, studies have demonstrated that the long range dependencies may have a pervasive effect on queuing performance. In fact, there is clear evidence that it can potentially cause massive cell losses. Furthermore, such a queuing system suffers from the buffer inefficacy phenomenon. Increasing the buffer size is not effective for decreasing the buffer overflow probability significantly [1].

The case study treated in this paper concerns Future Value Added Service Providers and particularly Video on Demand VASP. The objective of this work is part of the work undertaken at the LSC Lab to define a global architecture based on intelligent agents. Part of the intelligence is on neural networks. The aim of the architecture is to allow VDSP to improve the use of the bandwidth while reducing the risk of Quality of Service Degradation. VDSP are connected to a particular ISP that provide the necessary bandwidth that permit to end customers to access video service. The egress traffic from the VDSP domain is composed by the sum of individual video streams that are sent to the end customers. The classical approach for VDSP to allow this communications is to establish a contract with an ISP that provide it with a certain bandwidth. However, the main problem with this approach is that usually the bandwidth is provided as leased line. Consequently, if the sum of traffic is inferior to the available bandwidth, the VDSP will pay for bandwidth that is not used. In the reverse, if the number of customer increase, it is possible that the allocated bandwidth is not sufficient and consequently the quality of service of the communication will be violated or the VDSP will have to refuse new connection. Thus, its is important for the VDSP to be able to control dynamically the ISP allocated bandwidth and have the possibility to increase it or reduce dependently on forecasting traffic. Because the Internet exhibit a long-range dependence, one possible approach to predict the traffic is to calculate in real time the nature of the VDSP egress traffic and detect whether it is possible or not to predict what will the future traffic look like. If the traffic is decreasing significantly than it is the advantage of the VDSP to negotiate online a decrease of its bandwidth reservation. In contrarily, if the traffic is going to increase over the maximum bandwidth reservation, the VDSP will negotiate an increase of its bandwidth. In the other cases, the bandwidth allocation is optimal or it is not possible to have a clear view about the future.

The Hurst parameter or parameter H characterizes the degree of self-similarity of a process degree. The calculation in real time of this parameter will give a good view about the type of traffic that flooding on a particular link. Thus the objective in this paper is identify the nature of the VDSP egress traffic by calculating the Hurst Parameter based on online monitoring of traffic at the VDSP access router. There are several statistical estimators for the Hurst parameter [2] but they need several samples for an accurate analysis and thus are not able to provide a fast an accurate calculation of the Hurst parameter. The objective of this work is to evaluate the possibility to use an Artificial Neural Networks (ANNs) to calculate in real time an estimation of the Hurst parameter. This neuronal network is managed by a agent that is placed in or near the VDSP access router and capable to take a decision regarding forecasting bandwidth reservation. The traffic for this analysis is generated by an algorithm and the results are compared with three other analytical estimators in order to evaluate the accuracy of the proposal. Results indicate that the neurocomputation approach provides reasonably accurate results and is proper for real time implementation.

The remainder of this paper is organized as follows. Section II provides some notions about Internet traffic, Hurst parameter, agent technology and neural networks. The proposed architecture is presented in Section III, the statistical estimators in Section IV and the neural estimator in Section V. Numerical results are shown in Section VI and the Section VII concludes the paper.

II. BACKGROUND

A. Internet Traffic

The Internet Protocol (IP) is part of the TCP (Transmission Control Protocol)/IP protocol suite and is the most widely-used internetworking protocol. The IP function is to transfer data blocks, namely datagrams, transported from the source host to the destination host. IP is a connectionless network protocol. Every IP datagram is seen as an independent unit. The communication is unreliable, i.e., there is no end-to-end recognition nor between intermediate nodes. In addition, there is no mechanism of error control of the data transmitted and no flow control is used. In the old days, most of the data carried on networks was textural data. Today, with the rise of multimedia applications and network technologies, multimedia has become an indispensable feature on the Internet. Real-time voice and video applications become more and more popular on the Internet.

However, multimedia networking is not a simple task and, so far, the Internet has been following the best-effort delivery model. There is no admission control and the network does its best to transmit information as quickly as possible but there is no assurance about the delivery of the packets. Most of the applications on the Internet were elastic [3] in nature, in that they tolerated packet delays and packet losses and they could be relatively served by the best-effort model. Nevertheless, the emerging multimedia traffic, such as voice and video, do not
perform well under extreme variations in the delay and excessive dropping of packets.

In the current Internet model, the inelastic traffic does not perform adequately and interferes with the elastic traffic, leaving it with less bandwidth. Hence, to accommodate inelastic traffic, it is necessary to extend the Internet model in order to support the newer applications by providing QoS control [3][4][5].

B. The Hurst Parameter

Let \( x(t) \), with \( t = 0, 1, 2, \ldots \), a stationary stochastic process [6]. For each \( m = 1,2,\ldots \), let \( x^{(m)}(k), k = 1,2,3,\ldots \), denote a new series obtained by averaging the original series \( x(t) \) over non-overlapping blocks of size \( m \).

A process \( X \) is called exactly second-order self-similar with parameter \( H = 1 - \beta/2 \), \( 0 < \beta < 1 \), if its autocorrelation function is [6]:

\[
r^{(m)}(k) = \frac{1}{2}[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}] = q(k), \quad 0 < \beta < 1, \quad k = 1,2,\ldots
\]  

and \( X \) is called asymptotically second-order self-similar with parameter \( H = 1 - \beta/2 \), \( 0 < \beta < 1 \), if for all \( k = 1,2,\ldots \),

\[
\lim_{m \to \infty} r^{(m)}(k) = \frac{1}{2}[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}] = y(k)
\]  

In self-similar processes, the autocorrelations decay hyperbolically implying in a non-summable autocorrelation function \( \Sigma_r(k) = \infty \) (long-range dependences),

The Hurst parameter \( (H) \) gives the degree of self-similarity of a process, and, consequently, expresses the pattern of dependencies of a process. If \( 0.5 < H < 1 \), the process is a long-range Dependent (LRD) process. If \( 0 < H < 0.5 \) it is an anti-persistence process, and if \( H = 0.5 \) it is a short-range dependent (SRD) process.

Figure 1 illustrates the auto-correlation decay for different values of the Hurst parameter.

C. Agent Technology

The agent concept has been widely proposed and adopted within both the telecommunications and Internet communities is a key tool in the creation of an open, heterogeneous and programmable network environment. This trend is motivated by the desire to use the agents to solve some of the problems encountered in large-scale distributed and real-time systems such as the volume and complexity of the tasks, latency, delays, and others. Generally, an agent can be regarded as an assistant or helper, which performs routine and complicated tasks on the user's behalf. In the context of distributed computing, an agent is an autonomous software component that acts asynchronously on the user's behalf. Agent types can be broadly categorized as static or mobile.

The majority of current communication system architectures employ the client-server method that requires multiple transactions before a given task can be accomplished. This can lead to increased signaling traffic throughout the network. This problem can rapidly escalate in an open network environment that spans multiple domains. As an alternative solution, mobile agents can migrate the computations or interactions to the remote host by moving the execution there. For example, mobile agents can be delegated to complete specific tasks on their own, providing that a certain set of constraints or rules have been defined for them [7]. They can then be dispatched across the network in the form of mobile program or mobile code that can be recompiled and executed in the remote host.

The main motivation of the use of agent technology in this work is driven by the desire to automate the control and management processes by allowing for more programmability of the network to rapidly customize the provision of new information and telecommunication services. Hence, agents can also be used to implement Service Level Agreements (SLAs) between different actors of the network and service era. Agent can then be used as brokers or mediators between end users and a service provider in order to implement the SLA. In this way, complicated QoS metrics (from end user’s point of view) can be communicated in a simplified manner. Service provider and network provider agents can then negotiate with users’ agents in order to meet the required service [8].

D. Neural Networks

A neural network is a system composed by a high number of simple processors (neurons or nodes), highly interconnected and based on a simplified model of a neuron. Neural networks can adapt the computation taking into account previous knowledge of solutions for the problem under investigation (training) [9] [10].

The capacity to learn information from models is the one of most important features of NNs. They learn where are inserted and modify their acting depend on this learning.

In fact, the weights represent the knowledge of the NN at the end of the training process and the learning is the result of all the process. Therefore, the learning is a process where the synaptic connections of the neural network are adapted by a
continuous stimulus process from the environment where the network is inserted [11].

III. PROPOSED ARCHITECTURE

The target architecture we attend to deploy is composed by a set of agents. In order to be operational, we suppose that all actors (VDSP, ISP and the end customer) will deploy an agent platform. At the ISP boundary, the platform will support four types of agents: a Policy agent which is responsible for the management of global ISP policies that define high level specification of the system behavior (pricing rules, allocation rules, etc.), a Reservation agent that is responsible to interacting with physical equipments (routers) in order to configure them to provide the bandwidth agreed at the business level to the ISP customers (end customer and VDSP), an Accounting and Billing agent that is responsible to accounting and billing network resource usage and allocating them to ISP customers and finally the Bandwidth Broker that is responsible to interacting with ISP customer for the purpose of negotiating bandwidth allocation (Figure 2).

![Fig. 2. Multi-agents environment](image)

Et the VASP VoD premise, the VASP intelligent agent is responsible for the monitoring of the egress traffic and the prediction of the forecasting traffic. At the end customers premise, a negotiation agent allows the end customer to register for the VoD service. The VDSP is responsible for the end customer bandwidth availability.

The starting point of the process is a VASP VoD willing to offer on line a VoD service to a large number of end users. The VASP has contracted a line with a certain bandwidth to a particular ISP. We consider that initially end customers who have a basic IP service with the same ISP using DSL technology. Hence, we suppose that the ISP is able to control the bandwidth allocation of the end user. The customer register online to the VoD VASP in order to receive a particular movie stream at a certain date/time. Because of the busy nature of the traffic that is sent by the servers when flooding videos, the interest of the VoD VASP is to adapt the bandwidth reservation with the ISP network to a level that is compatible with what is globally sent, thus to avoid underprovisioning or overprovisioning and then extra cost or quality of service degradation. This can be possible only if it is possible for the VADS to predict what the traffic will look like in the near future. This task is affected to the Intelligent VASP agent.

Once the agent detect an important change in the traffic, it informs the BB agent of the ISP management system in order to request or release resources to adapt to the new traffic profile.

Therefore, the Intelligent VASP Agent is at the heart of the suggested architecture.

Thus the objective of this work is to show whether it is possible to predict the nature of the traffic rapidly so that to permit to renegotiate the allocated bandwidth.

In the architecture the renegotiation dialog is realized between the intelligent VDSP agent and the ISP BB agent.

This aspect of the architecture and the internal interactions between the agent are not addressed in this paper. Only the intelligent VDSP behavior is addressed.

The objective is to use a neuronal network that permit the VDSP agent to take a decision regarding what to do in the near future. The idea is to train the neuronal network with existing traffic profile and determine whether it is capable to calculate the Hurst parameter that gives information about the nature of the traffic in a relatively short period. In the real situation, it is not false to say that we can train the neuronal network with the existing video streams as the VDSP have them already. Every time a new movie is included in the movie portfolio, the agent is trained with its traffic.

To validate the usability of the approach, we have adopted the following methodology. We have taken some traffic examples for which we have the exact value of the Hurst parameters. Then we calculate the Hurst parameters using two methods the statistical approach and the neuronal approach and finally we make a comparison of the different approaches:

![Fig. 3. Hurst Parameter Calculation](image)

IV. STATISTICAL ESTIMATORS

Three statistical estimators were used to validate results given by the neuro estimator: the R/S statistical, the Higuchi method and the Abry-Veitch estimator.

A. The R/S Statistical Estimator

The R/S estimator, defined by Hurst (1951), is one of the most well-known and simplest methods for estimation of dependence level of a series of samples [3].

For a stochastic process $x(t)$ defined at discrete-time intervals $\{x_t, t = 0, 1, 2, \ldots\}$, the rescaled range of $x(t)$ over a time interval $N$ is defined as the ratio $R/S$: \[
R/S = \frac{S}{N^{\alpha}},
\]

where $S$ is the range of the series, $N$ is the number of samples, and $\alpha$ is the Hurst parameter. The estimation process involves selecting a series of non-overlapping sub-series of length $N$, calculating the range of each sub-series, and averaging the ranges to obtain $S$. The Hurst parameter $\alpha$ is then estimated by fitting the following linear regression model to the log-log plot of $R/S$ vs. $N$: \[
\log(R/S) = a \log(N) + b
\]

The slope $a$ of this line is estimated using least squares or other regression methods. The value of $\alpha$ is then calculated as $\alpha = -a$.
Fractal Brownian Motion samples [9] [10] were generated to train the neural network and to generate the statistical estimator inputs. The three statistical estimators described before were used for the verification of the precision of the neural network. Results were derived using the Stuttgart Neural Network Simulator was used in the experiment [11]. Sequences of 10 samples have characteristics similar to sequences of 100, 1000 or 10000 samples; (ii). training time is short with little neurons.

There is no established procedure of choice for the optimum number of neuron. Then, experiments with 2, 5, 10 15 and 20 neurons were tried and the better results were the ones with 15 hidden neurons (Table 2). The patterns were chosen in the following way:

- 200 traces sequences with $H=0.5$;
- 100 traces sequences with $H=0.6$, $H=0.7$, $H=0.8$ and $H=0.9$

Each sequence has 10 bursts of samples. For the trace with $H=0.5$, it was necessary to have twice the number of samples due to short range dependences. After the learning phase the performance of the neural network was checked. These patterns should not have been presented to the network before.

For the definition of the number of neurons and of the activation function, the first training used a stopping criterion of 1000 epochs. Each epochs contains one complete cycle (sweeping) through all training pattern set. This training allows for the selection of the neural network topology which is more adapted to the problem. The second training considered another stopping criterion: a small error.

After the training phase, 600 sequences of the test set were applied to NN. These sequences are not known by the network since it is necessary to verify the generalization capacity of the neural network.

VI. RESULTS AND DISCUSSION

Estimations of $H$ and respective errors of the three statistical methods are shown in Table 1.

<table>
<thead>
<tr>
<th>Hidden neurons number</th>
<th>Hidden layer activation function</th>
<th>Logistic Mean-Square Error (1000 epochs)</th>
<th>Hyperbolic tangent Mean-Square Error (1000 epochs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Hyperbolic tangent</td>
<td>0.3371</td>
<td>0.2191</td>
</tr>
<tr>
<td>5</td>
<td>Hyperbolic tangent</td>
<td>0.3219</td>
<td>0.1225</td>
</tr>
<tr>
<td>10</td>
<td>Hyperbolic tangent</td>
<td>0.3190</td>
<td>0.0197</td>
</tr>
<tr>
<td>15</td>
<td>Hyperbolic tangent</td>
<td>0.1355</td>
<td>0.0060</td>
</tr>
<tr>
<td>20</td>
<td>Hyperbolic tangent</td>
<td>0.2008</td>
<td>0.0131</td>
</tr>
</tbody>
</table>

Table 2 presents different scenarios used in the experiments. The difference of magnitude of the mean-square error given by the hyperbolic tangent and the logistics activation function is evident.

**TABLE 2**

| MEAN-SQUARE ERROR OF THE INITIAL TRAINING |
For both in logistic and hyperbolic tangent functions, the worst and the best cases happened with 2 and 15 neurons, respectively. Table 2 reveals that the neural network with a hidden layer of 15 neurons and with hyperbolic tangent activation function is the best option.

After choosing the neural network topology chosen (10 input neurons, 15 hidden neurons and 1 output neuron), the real training was persued. The stopping criterion used for the learning process was an error less than 0.0010 (0.1%). The NN was trained with 2616 epochs (Fig. 4).

Six hundred 600 sequences of the test set were applied to neural network in the execution phase. These sequences had never been known by the network. The neural estimator error for different values of the parameter $H$ can be seen in the Fig. 5. Notice that the error has an initial increase, from $H=0.5$ to $H=0.6$, and then decreases. The three statistical estimators have used the complete traces (10000 samples) to make their calculations.

The neural network needed only one tenth of the total trace, 1000 samples, for each LRD trace ($H=0.6$, $H=0.7$, $H=0.8$ and $H=0.9$). When the input samples were doubled duplicated for the SRD trace ($H=0.5$), i.e., 2000 samples, there was considerable improvement in the error (Fig. 3). In other words, the neural networks enhanced the precision of the estimations given that sample size increase needed by the statistical estimators.

It can be seen in Figure 4 that for values of the Hurst parameter close to 0.5 (SRD), the neural network estimator produces the least accurate results. However, for the range of interest for network traffic ($H > 0.7$), the neural network estimator gives more precise results than the R/S and the Higuchi estimators.

For the range of interest of the Hurst parameter, the neural network produces values of $H$ which differ at most 0.02 from the results produced by the Abry-Veitch estimator which is the most precise one. The difference between the results by these two estimators decreases as $H$ increases.

Moreover, the neural networks demanded half of the number of the samples required by the statistical estimators, i.e., the neural network estimator can generate quite accurate results much faster than the statistical estimators which is specially advantageous to quickly detect variations of the Hurst parameter in real time. The drawback of the neural networks is the delay in learning which impact, however, decreases for large traces.

VI. CONCLUSION

Long-range dependences have a significant impact on both network dimensioning and traffic management. The Hurst parameter is a measure of self-similarity of a process, and, thus, the intensity of the long-range dependences of a process.
Small variations in the Hurst parameter value may lead to considerable changes on traffic control. Therefore, an accurate and quick evaluation of the Hurst parameter is of paramount importance. The statistical estimators requires large sample size. Consequently, it presents limitations for real time evaluation of $H$. The present work investigated the effectiveness of a neural network estimator for the Hurst parameter. Neural networks, even demanding a significant time for training, represent an accurate and fast estimation of the parameter $H$. The presented approach aims to use this neuronal network as the intelligent part of an agent that permit a Video on Demand Service Provider to predict its traffic. The global architecture will permit in a competitive telecommunication market to an ISP to provide a new type of on-demand bandwidth provisioning. This will help future VASP to reduce their communication cost while preserving the required QoS. The future work is to introduce this neuronal network in the agent and to realize the overall loop provisioning i.e. allocation of bandwidth, monitoring of traffic, predicting future traffic, renegotiating bandwidth allocation, reallocation resources. Then the deployed architecture will be evaluated in term of accuracy of the decision algorithm as well as the impact of the these reconfiguration on the final QoS visible to the end customers.

REFERENCES