Analysis and Prediction of Real Network Traffic

Mohamed Faten Zhani and Halima Elbiaze
Department of Computer Sciences, University of Québec in Montréal, Canada
{zhani.mohamed_faten, elbiaze.halima}@courrier.uqam.ca
Farouk Kamoun
National School of Computer Sciences, Manouba 4010, Tunisia
frk.kamoun@planet.tn

***Abstract*— Short period prediction is a relevant task for many network applications. Tuning the parameters of the prediction model is very crucial to achieve accurate prediction. This work focuses on the design, the empirical evaluation and the analysis of the behavior of training-based models for predicting the throughput of a single link i.e. the incoming input rate in Megabit per second. In this work, a neurofuzzy model (α_SNF), the AutoRegressive Moving Average (ARMA) model and the Integrated AutoRegressive Moving Average (ARIMA) model are used for predicting. Via experimentation on real network traffic of different links, we study the effect of some parameters on the prediction performance in terms of error. These parameters are the amount of data needed to identify the model (i.e. training set), the number of last observations of the throughput (i.e. lag) needed as inputs for the model, the data granularity, variance and packet size distribution. We also investigate the use of the number of packets or sampled data as inputs for the prediction model. Experimental results show that training-based models, identified with small training set and using only one lag, can provide accurate prediction. We show that counts of packets and especially large packets can be used to efficiently predict the throughput.**

***Index Terms*— Traffic modeling, traffic measurements, traffic prediction, neurofuzzy models, ARMA model, self-similarity**

I. INTRODUCTION

The predictability of network traffic is of significant interest in many domains [1]–[3]. We can distinguish two categories of prediction: long and short period predictions. Traffic prediction for long periods provides a detailed forecasting of the workload and traffic patterns to assess future capacity requirements, and therefore allows for more accurate planning and better decisions. Short period prediction (milli-seconds to minutes) is relevant for dynamic resource allocation. It can be used to improve the Quality of Service (QoS) mechanisms as well as congestion and resource control by adapting the network parameters to traffic characteristics. It can also be used for routing packets.

Traffic prediction has been extensively investigated since the acceptance of the self-similar and the long-range dependence nature of networks traffic [4]–[7]. While these peculiar characteristics cause dramatic effects on network performance in terms of loss and delay, several studies have shown that the self-similarity can be exploited to characterize or to predict the traffic in order to control the network resources assignment [8]–[14].

From the extensive work done in the field of prediction methods for network traffic, we draw the following conclusions. First, generalization about the predictability of network traffic is difficult to make since network traffic can change considerably over time and space. Traffic is self-similar and has a non-linear nature, and this makes it highly difficult to perform accurate prediction especially for linear models [15]–[20]. Thereby, the prediction model should be adaptive. Second, aggregation and smoothing appear to improve predictability [15]. Third, there are clearly differences in the performance of the various predictive models [19].

We can also classify prediction techniques into two categories: *Training-Based* (TB) techniques and *Non-Training-Based* (NTB) techniques. Specifically, the TB techniques need a training phase. The training phase consists of identifying model parameters based the history of the throughput measurements called *the training data set*. The TB model is then fed by the last observations of the throughput called *lags* in order to predict the future value. Generally, the complexity of the training phase is not crucial since it is performed once. Contrarily, NTB techniques do not need a training phase and calculate the predicted value using only the last lags. The number of lags is usually chosen lower than 10 for TB and NTB models [10], [12]–[17], [19], [21]–[23] in order to reduce the complexity of the model.

The most used training-based models are *the AutoRegressive* (AR) model and *the AutoRegressive Moving Average* (ARMA) model [10], [15]–[17], [19], [21] where as *the Linear Minimum Mean Square Error* (LMMSE) is widely used as an NTB technique [12]–[14], [22].

We choose to investigate TB models since they achieve better performance than NTB models [23], [24]. Thus, three TB prediction models are used namely the ARMA model [10], [15]–[17], [19], the ARIMA model [10], [15]–[17], [19] and the α_SNF model [23], [25].

The α_SNF model combines fuzzy logic which presents the model as a set of simple rules (if *event* then *action*) and neural networks which are the basic learner to capture the non-linear characteristic of network traffic.

Neurofuzzy networks have been used to predict video traffic [26] and Web server traffic [27]. However, model parameters and their effects were not sufficiently investi-

Corresponding author : Mohamed Faten Zhani

doi:10.4304/jnw.4.9.855-865

gated.

The work focuses on the short period throughput prediction (i.e. the incoming input rate in *Megabit per second* (Mbps)). All the performed experiments are using real traffic collected from two links having different characteristics (type of the link, traffic load etc.)

The work investigates the following issues:

- how much data are needed for the training phase,
- how many lags are needed to be used as an input for the prediction model to improve its accuracy,
- what is the effect of the considered *traffic granularity* i.e. the interval of time separating two measurements of the traffic throughput,
- how to use exogenous variables as an input for the prediction model. *Exogenous variables* are variables which are different from the lags such as the number of packets or sampled data.

The aim of the paper is to study the influence of a set of parameters in a prediction model, trying to find out the best options for real applications. Thus, we tried only to find the best parameters which provide the best possible accuracy (lowest error).

The decision if the error is good enough will depend on the application using the prediction i.e. only the application can decide if the error is tolerable or not. For instance, recent work showed that when using prediction to improve active queue management, the prediction error is not crucial [12], [23]. That is the prediction improved the performance although there is a prediction error.

The remainder of this paper is organized as follows. Section II introduces selected related work on traffic prediction. Section III describes our prediction methodology as well as the prediction models used in our study. In Section IV, the real network traces used for experiments are described . Section V discusses the experimental results about the choice of input variables, the effect of the granularity, the variance and the size of the training data set. It also investigates using the exogenous variables as inputs for the model. The conclusions and future work are presented in Section VI.

II. Related Work

Wolski has developed the first network measurement system that integrated prediction [28]. He found that running multiple predictors (mean, median, and AR models) simultaneously and forecasting with the one currently exhibiting the smallest prediction error gives the best results on his measurements.

Yang et al. have attempted to improve *the least-mean square* (LMS) predictor so-called *Error-adjusted LMS* (EaLMS) [20]. The main idea of EaLMS is using previous prediction errors to adjust the LMS prediction value, so that the prediction delay could be decreased. The authors have used traffic obtained by smoothing real traffic assuming that it preserves the main characteristic of original traffic. Compared to LMS predictor, EaLMS significantly reduces prediction delay and avoids the problem of augmenting prediction error at the same time especially for short period prediction.

Other related works on traffic prediction have exploited the self-similarity characteristic of network traffic in TCP congestion control [12]–[14], [22]. He et al. have shown that the correlation structure present in self-similar traffic can be detected on-line and used to predict future traffic [14]. Hence, they define a scheme, called *TCP with traffic prediction* (TCP/TP) that uses the prediction results to infer the optimal point at which a TCP connection should operate.

Tong et al. have proposed a framework which adopts *Principal Component Analysis* (PCA) as an optional step to take advantage of self-similar nature of traffic while avoiding its disadvantages [18]. Neural network is used as the basic regressor to capture the non-linear relationship within the traffic. Experimental results on real network traffic validate the effectiveness of the framework.

Sang et al. have proposed an approach to make predictability analysis of network traffic [15]. The approach assesses the predictability of network traffic by considering two metrics: (1) how far into the future a traffic rate process can be predicted with bounded error; (2) what is the minimum prediction error over a specified prediction time interval. The authors have used two stationary traffic models: the ARMA model and *the Markov-Modulated Poisson Process* (MMPP). Making the assumption that such models are appropriate, they have developed an analytic expressions for how far into the future prediction was possible before errors would exceed a bound. The authors have shown that this bound was affected by traffic aggregation and smoothing of measurements. They have argued that the two models, though both short-range dependent, can capture statistics of self-similar traffic quite accurately, for the limited considered time scales.

He et al. have focused on predictability of large transfer TCP throughput [24]. TCP prediction techniques have been classified into two categories: *Formula-based* (FB) and *History-Based* (HB). FB prediction relies on mathematical models that express the TCP throughput as a function of the characteristics of the underlying network path (round trip time, number of flows etc.). HB techniques predict the throughput measurements on the same path, when such a history is available. It has been shown that HB predictors are quite accurate but are highly path-dependent; whereas, FB predictors are accurate only if the TCP transfer is not saturating the underlying path [24].

Other works in the domain of Internet traffic forecasting addresses long period predictions that are important for IP network capacity planning [10], [17].

Our work is different from the above works in that it focuses on the parameters of the training-based models used for short period prediction of the traffic throughput. To the best of our knowledge, none has analyzed yet the effect of the model parameters on the prediction performance: (i) how many lags are needed to have an accurate model? (ii) how much data are needed for the

training phase to identify an accurate model? and (iii) what is the effect of the granularity of the data on the model performance? (iv) how to use of exogenous variables as an input for the prediction model ?

III. TRAINING-BASED PREDICTION MODELS

In what follows, we introduce the prediction methodology considered in this work. We also present the training-based prediction models considered in this study. The first model is the α_SNF which is a neurofuzzy model used by [23], [25]. It combines fuzzy logic and neural networks. The second model is the ARMA model which is a linear model widely used in literature [10], [15]–[17], [19], [21].

In what follows, we note by $y(t, bps)$ and $y(t, pps)$ the throughput at the time t respectively expressed in Mbps or in packets number per second (pps). In order to simplify, we note $y(t, bps)$ by $y(t)$.

A. Prediction Methodology

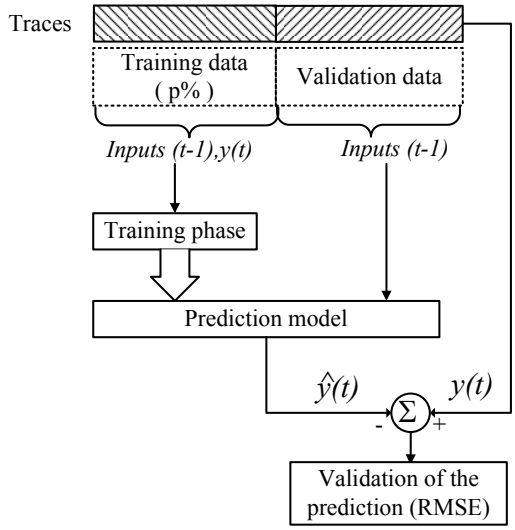


Figure 1. Prediction Methodology

In what follows, we describe the different steps of our prediction methodology illustrated in Fig. 1.

- Available data (e.g. the throughput values) are divided into two sets. The first set called *the training data set* constitutes $p\%$ (usually $p = 50$) of the available data. It is used to identify the prediction model parameters. The second set is called *the validation data set* used to compare the prediction results with the real data in order to evaluate the performance of the predictor.
- **The prediction model**: For each input at time $t-1$ ($inputs(t-1)$), the model calculates $\hat{y}(t)$ the prediction for $y(t)$. The inputs can be a previous observations i.e. lags $y(t-i)$ or any exogenous variable measured at time $t-1$.
- **The training phase:** It is the phase of identifying the model parameters. Thus, we inject the training data set into the training algorithm. The training data set is composed of the inputs ($inputs(t-1)$) and the outputs ($y(t)$). The training algorithm estimates model parameters which give the minimum error between the model outputs $\hat{y}(t)$ and the real traffic values $y(t)$.
- **The prediction phase**: during this phase, only $inputs(t-1)$ from the validation data set are injected to the model in order to estimate $\hat{y}(t)$.
- **Validation of the prediction:** The performance criterion used to evaluate the accuracy of the prediction is *the Root Mean Square Error* (RMSE):

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} \left[y(t) - \hat{y}(t)\right]^2}{n}} \qquad (1)$$

where $y(t)$ is the real output, $\hat{y}(t)$ is the calculated output and n is the number of the input data. Thus, the RMSE measures of the error between the predicted values by the prediction model and the real values actually observed.

In this work, the prediction model can be either the α_SNF model or the ARMA model. In what follows, we present these two models.

B. The Neurofuzzy model (α_SNF)

The α_SNF is a model which combines fuzzy logic and neural networks [25]. The flexibility of fuzzy logic in dealing with uncertainty and the learnability of neural networks make the model more adaptive to the traffic characteristics.

The fuzzy system is described as a non-linear relation between inputs $x_1, ..., x_n$ and an output $Y = f(x_1, ..., x_n)$, where n is the number of inputs x_i. This relation is described by a collection of fuzzy rules. Let c be the number of rules in the fuzzy system.

We note by R_k the k^{th} rule where $1 \leq k \leq c$. A fuzzy rule R_k is given as the following :

$$R_k : if\ (x_1, \ldots, x_n)\ is\ A_k\ then\ y_k\ is\ b_k, \qquad (2)$$

Where A_k is called a cluster and y_k is the output of the rule calculated using a real noted b_k.

In fuzzy logic, every point x belongs to a cluster A with a membership degree that has a value between 0 and 1 given by a membership function $\mu_A(x)$. Thus, each rule R^k evaluates the membership of each element $(x_1, ..., x_n)$ to each cluster A_k noted $\mu_{A_k}(x_1, ..., x_n)$. Then y_k is calculated as:

$$y_k = \mu_{A_k}(x_1, ..., x_n).b_k, \qquad (3)$$

The rule R_k can be written as:

$$R_k : x_1\ is\ A_{k1}\ and\ x_j\ is\ A_{kj}, \ldots, and\ x_n\ is\ A_{kn}\ \textit{then}\ y_k\ is\ b_k. \qquad (4)$$

where the cluster A_{ki} is the projection of A_k in the i^{th} dimension. We note $\mu_{A_{ik}}(x_i)$ the membership function of x_i to the cluster A_{ki}. Then, $\mu_{A_k}(x_1, ..., x_n)$ is given by:

$$\mu_{A_k}(x_1, ..., x_n) = \prod_{i=1}^{n} \mu_{A_{ik}}(x_i),$$

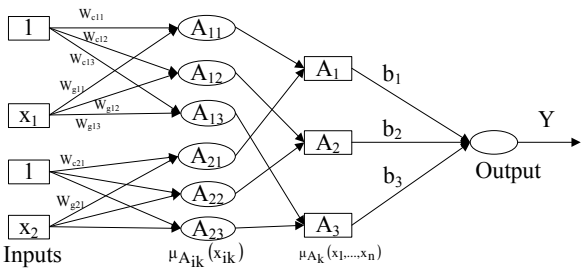


Figure 2. Example of equivalent neural network α_SNF(2 inputs, 3 rules).

Hence, the output of the system Y is given by:

$$Y = \frac{\sum_{k=1}^{c} y_k}{\sum_{k=1}^{c} \mu_{A_k}(x_1, ..., x_n)} \quad (5)$$

We used the same membership function considered by [25] which is:

$$\mu_{A_{ik}}(x_i) = exp(-|w_{gik}x_i + w_{cik}|^{l_{ik}}), \quad (6)$$

where w_{gik},w_{cik} and l_{ik} parameters are used to adjust the general form of the function.

The fuzzy model is then incorporated into an equivalent neural network. Fig. 2 shows an example of α_SNF (2 inputs,3 rules) where x_1 and x_2 are inputs. Each node A_{ik} calculates the used membership function $\mu_{A_{ik}}(x_i)$ using Eq.6 and each node A_k calculates the result of the rule k using Eq.3. The output node calculates the output Y using Eq.5.

The training algorithm used for the α_SNF model is the back-propagation algorithm [25]. Training the neural network aims at changing the parameters w_{gik}, w_{cik} and l_{ik} of each rule in order to reduce the error between the calculated output and the real output.

For selecting the proper number of rules, we have found via experiments that using more than 3 rules does not significantly improve the prediction performance.

C. AutoRegressive Moving Average Model (ARMA)

The most well-known linear forecasting models are the *Autoregressive* (AR), *Moving Average* (MA) and the *the AutoRegressive Moving Average* (ARMA). A time series $y(t)$ is an ARMA(p,q) process if it is stationary and if for every t:

$$y(t) = \phi_1 y(t-1) + .. + \phi_p y(t-p) + \epsilon(t) + \theta_1 \epsilon(t-1) + .. + \theta_q \epsilon(t-q), \quad (7)$$

where ϕ_i and θ_j are the parameters of the model, and $\epsilon(t)$ are error terms. The error terms $\epsilon(t)$are assumed independent, identically distributed sampled from a normal distribution with zero mean and finite variance σ^2.

The equation can also be written in a more concise form as:

$$y(t) = \sum_{i=1}^{p} \phi_i L^i y(t) + (1 + \sum_{i=1}^{q} \theta_i L^i)\epsilon(t), \quad (8)$$

where L is the backward shift operator defined as follows: $L^i y(t) = y(t-i)$. We notice that AR and MA are special cases when $q = 0$ or $p = 0$.

D. Integrated AutoRegressive Moving Average Model (ARIMA)

The ARMA model fitting procedure assumes the data to be stationary. If the time series exhibits variations that violate the stationary assumption, then there are specific approaches to make the time series stationary. The most common one is what is often called the "differencing operation". It is defined by $(1-L)y(t) = y(t) - y(t-1)$. It can be shown that a polynomial trend of degree k is reduced to a constant by differencing k times, that is, by applying the operator $(1-L)^k y(t)$. We could therefore proceed by differencing repeatedly until the resulting series can plausibly be modeled as a realization of a stationary process.

An ARIMA(p,d,q) model is an ARMA(p,q) model that has been differenced d times. Thus, the ARIMA(p,d,q) can be given by:

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1-L)^d y(t) = (1 + \sum_{i=1}^{q} \theta_i L^i)\epsilon(t). \quad (9)$$

For training the ARMA and ARIMA models, *Powell's function minimization routine* is used to choose the coefficients to minimize the sum of squared prediction errors [29].

IV. Analysis of the Used Data

This section presents the real data sets used for experiments, the preprocessing performed on the traces before prediction experiments. Besides, we present an analysis of the traffic characteristics which could have an effect on the prediction performance.

A. Trace and Preprocessing

The first set of data is the *"Auckland-VIII data Set"*[1]. It is a two-week GPS-synchronized IP header trace captured with an Endace DAG3.5E tap Ethernet network measurement card in a link of 100 $Mbps$ in December 2003 at the University of Auckland Internet.

The second set of data is the *"CESCA-I data set"*. It is a three-hour GPS-synchronized IP header trace captured in a 1 $Gbps$ link with an Endace DAG4.2GE dual Gigabit Ethernet network measurement card in February 2004 at the Anella Cientifica (Scientific Ring), the Catalan R&D network.

We analyzed the traces with the libtrace tools[2] in oder to extract the throughput expressed in bbp and pps, the number of connections and their *Round Trip Time* RTT. We applied our expriments to the collected data

[1]Data are available from *the National Science Foundation* (NSF) and the NLANR Measurement and Network Analysis Group (http://pma.nlanr.net/Special/)

[2]*Wand Network Research Group*, The libtrace trace-processing library. http://research.wand.net.nz/software/libtrace.php

for different hours of the day, the results are very close. In what follows, we present the results found using 60 minutes of data from both traces measured at 10 am. We note the 60-minute data Auckland-VIII and CESCA-I. We also present the results for 5 days of data which we note WAuckland VIII. We used 50% of the data as the training data set and 50% as the validation set.

B. Trace Analysis

TABLE I.
TRAFFIC CHARACTERISTICS: AUCKLAND-VIII, CESCA-I AND WAUCKLAND-VIII.

Statistics	Auckland-VIII link	CESCA-I link	WAuckland-VIII link
Link Capacity	100 Mbps	1 Gbps	100 Mbps
Duration	1 hour	1 hour	5 days
Trace Size	60 Mbyte	20.5 Gbyte	12.1 Gbyte
Rate (Mbps)	2.83	487.16	7.61
Packets rate	972.73 pps	100 036.14 pps	1947.14 pps
Throughput Variance	1.280 Mbps	797.781 Mbps	2.36 Mbps
Average Connection Duration	8.23sec	8.14sec	8.10sec

A comparison between Auckland-VIII and CESCA-I traces is presented in Table I. It shows that for the same duration (1 $hour$), the 100Mbps link (Auckland) has much less load than the 1Gbps Ethernet link (CESCA) in terms of received data (size) and throughput mean expressed in Mbps or pps. We notice the high throughput variance of the Gigabit link which could degrade the prediction performance as it will be shown in the next section. However, the mean connection duration time is the same for both links (≈ 8 $seconds$).

Figure 3 shows the duration for existing connections over the Auckland link. The x-axis shows the starting time of the connection while the y-axis shows its corresponding duration. We note that most connections last very short periods (less than 1 $second$). The mean of the connections duration is 8.23 $seconds$. The longest connection lasts for 36 $minutes$.

Figure 3 shows the RTT for all the TCP connections of the Auckland link. The x-axis shows the connection number while the y-axis shows its corresponding RTT. Most of connections have RTT between 20 ms and 200 ms. The average RTT time is about 72 ms. The mean RTT and the mean connection duration could be interesting to identify the granularity for the data analysis. We note that a packet is received approximately every $1ms$. Thus using $1ms$ as granularity for traffic prediction leads to predict the size of the coming packet in the next $1ms$.

Figure 4(b) illustrates the packet size distribution for the Auckland traffic. The smallest packets, 40 bytes in length, represent 27% of packets number. They are mainly TCP packets with ACK, SYN, FIN, or RST flags. They are many 1,500-byte packets (17.67%) which result from the maximum packet size of an IP packet used over an Ethernet link (Ethernet maximum transfer unit is 1,500 bytes). They are also many 1,420-byte packets (2.10%). The large presence of 576-byte packets (5.14%) reflects TCP implementations without path MTU discovery, which use packets of 536 bytes (plus 40-byte header) as the default Maximum Segment Size (RFC 879) [30].

Figure 4(a) shows the packet size distribution considering the generated traffic. *The generated traffic* is defined as the amount of data in MegaBit generated by a particular size. The figure shows that 1,500-byte packets constitutes 57% of the traffic but only 17.67% of the total number of packets. Besides, although more than 27% of the packets are from small packets , they constitute less than 5% of the generated traffic. This observation concurs with the findings of Shao et Al. [21] that the traffic pattern is bimodal: most traffic is carried by a small number of packets and most packets carry smaller number of bytes. The same observation is done in the case of the CESCA traffic (Figure 5).

V. RESULTS

A. How to Choose Input Variables?

In order to choose the candidate input variables for the prediction model, two popular metrics are used: the correlation coefficient and mutual information. These metrics are computed using a candidate input variables (or a set of inputs) and the desired output. In our case, we calculate the mutual information or the correlation coefficient for each input variable (e.g. a lag $y(t-i)$) with the output $y(t)$.

The mutual information expresses the importance of each variable [31]. It measures the relationship between the input and the output. It is expressed by:

$$U(X,Y) = 2\left[\frac{H(Y)+H(X)-H(X,Y)}{H(X)+H(Y)}\right], \quad (10)$$

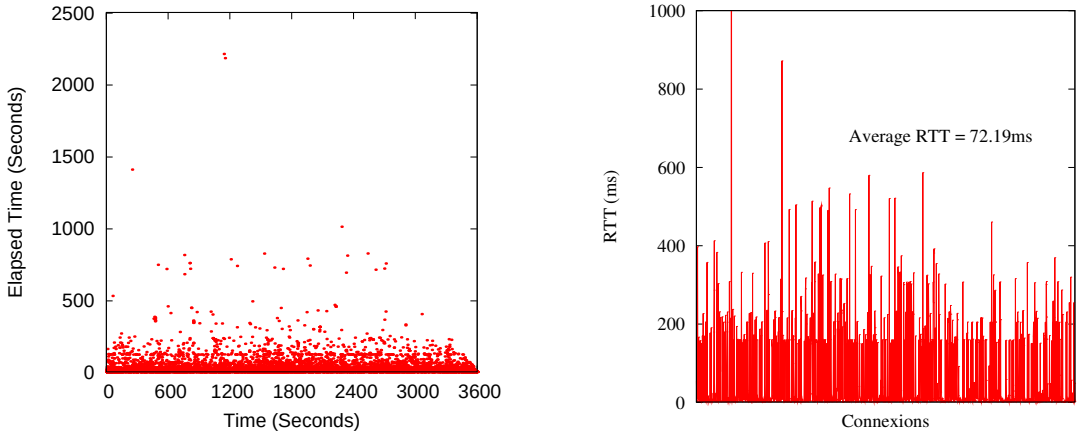
where

$$H(X) = H(p(x)/x \in X) = -\sum_{x\in X} p(x) \;\; log_2 p(x),$$

where $p(x)/x \in X$ is a probability distribution on finite set X, and $H(X,Y) = -\sum_{[(x,y)\in X\times Y]} p(x,y) log_2 p(x,y)$ is the joint entropy defined in terms of the joint probability distribution on $X \times Y$. Thus, the pertinence of the variable increases when its mutual information is more important.

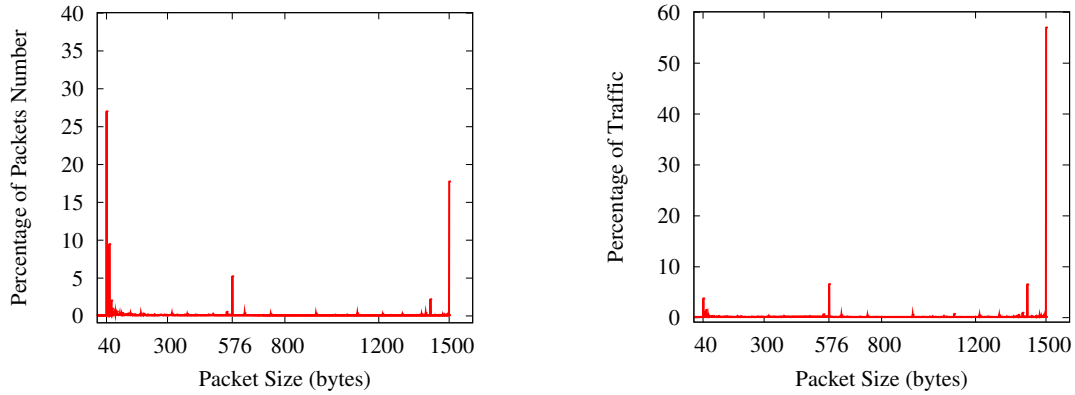
In probability theory and statistics, correlation coefficients indicate the strength and direction of a linear relationship between two random variables. The best known one is the Pearson product-moment correlation coefficient, which is obtained by dividing the covariance of the two variables by the product of their standard deviations [32]. The correlation $\rho_{X,Y}$ between two random variables X and Y with expected values (mean) μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{(X-\mu_X)(Y-\mu_Y)}{\sigma_X \sigma_Y}. \quad (11)$$



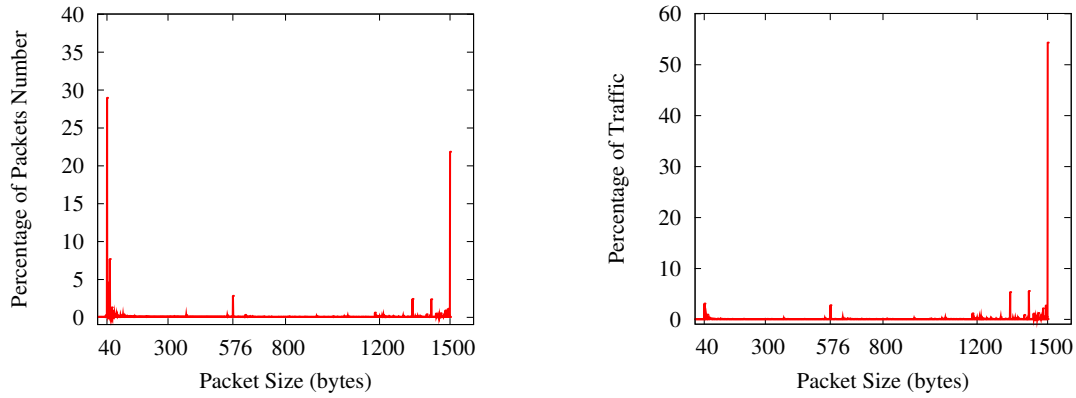
(a) (b)

Figure 3. Analysis of the Auckland traffic: (a) Duration of connections and (b) RTT for existing connections.



(a) Distribution considering the number of packets. (b) Distribution considering the generated traffic.

Figure 4. Packet Size Distribution for the Auckland traffic.



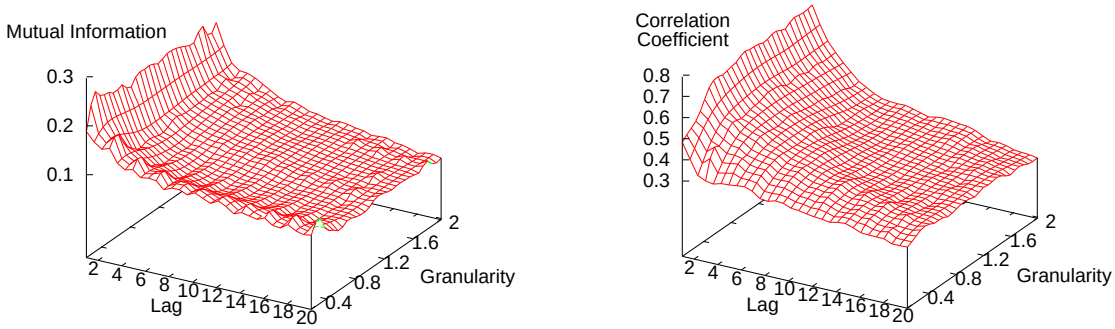
(a) Distribution considering the number of packets. (b) Distribution considering the generated traffic.

Figure 5. Packet Size Distribution for the Cesca traffic.

Fig. 6 shows the mutual information calculated for 20 lags for various granularities. It is obviously shown that pertinence of lags is decreasing when using older lags. Therefore, the mutual information of the last lag $y(t-1)$ is the most significative to predict $y(t)$. Similarly, the correlation coefficient decreases when older lags are used (Fig. 6). In other words, $y(t-1)$ and $y(t-2)$ are the most correlated to $y(t)$.

Fig. 7 shows prediction performance (RMSE) obtained from the ARMA model using input variables from 1 to 20 lags. It is clear that, whatever the granularity, using more than one lag as an input does not really improve the prediction performance.

These findings indicates that training-based model can



(a) (b)

Figure 6. (a) Mutual information and (b) Correlation coefficient for different granularities. (Auckland-VIII data set).

capture, during the training phase, the strong correlation of the traffic with long-range dependence over a range of time-scales and, hence, for the prediction phase, the model needs only the last lag to predict the incoming traffic.

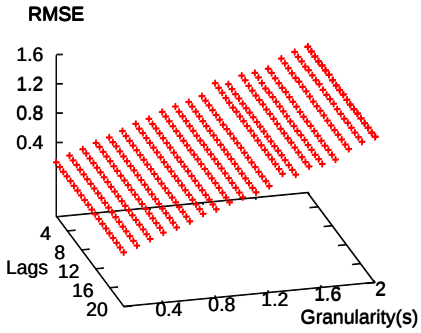


Figure 7. Prediction error using the ARMA model (Auckland-VIII data set).

We note that the mutual information has given more accurate estimation of the correlation of the traffic with the various lags because it shows that the traffic is correlated only with the last lag $y(t-1)$ for all the granularities. This result is validated by the prediction experiments (Fig. 7).

B. How to Choose Traffic Granularity?

In this section, we discuss the effect of the traffic granularity on the prediction performance.

We performed a set of predictions using data with granularity varying from $100\ ms$ to $2000\ ms$. We consider granularities which are multiple of the RTT ($\approx 100\ ms$) because if the prediction is used to improve TCP for example, the protocol reacts in one or in multiple RTT ahead.

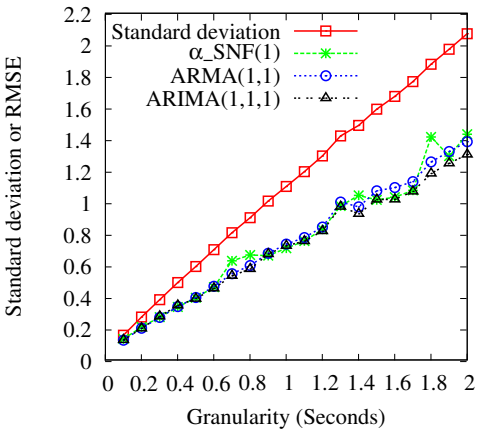
Fig. 8 depicts the obtained RMSE for the performed predictions for both models using different granularities. It shows that the prediction error (RMSE) increases when increasing the granularity. Thus, ideally, the granularity should be chosen based on the RMSE error which can be tolerated by the application of the prediction. For example to enhance TCP, we could need the prediction for the traffic one or multiple RTT ahead. The experimental results show that we can perform prediction for 6 RTT ahead ($\approx 600\ ms$) with an RMSE less than 0.4 for the Auckland traffic. However, the prediction error for the CESCA link is more important (RMSE > 1) and increases as the granularity is increased. This could be explained by the difference between the traffic characteristics of the two links especially the variance (Table I).

Fig. 8 also shows the standard deviation of the data (the square root of the variance) with respect to the granularity for both traces. We notice that when the granularity increases, the standard deviation increases and likewise for the prediction error. The figure shows that the prediction error is correlated to the the standard deviation (variance) of the data.

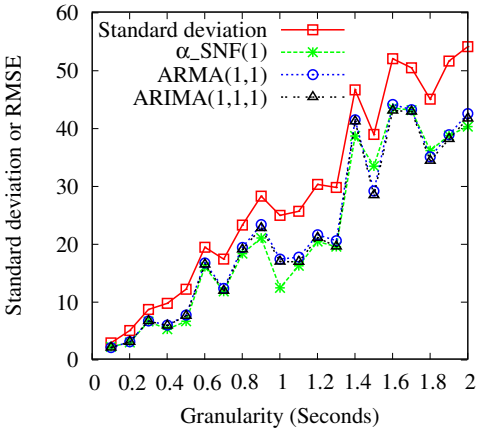
C. How Much Data are Needed for The Training Phase to Have an "accurate" Prediction?

Since the used models need training phase, we focus on how much data are needed by the training phase in order to have a "good" performance. Thus, we have varied the percentage of the training data $p\%$ from 10 to 50%. This means that for each time we use $p\%$ of the data to predict the last 50%.

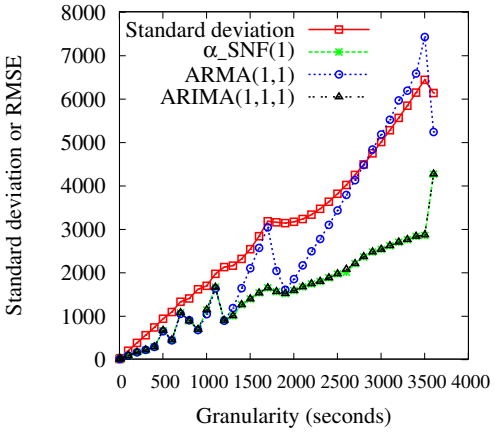
Fig. 9 shows the obtained prediction error for different percentage of training data. We observe that small training dataset (less than 5%) provides high prediction error. That is the data are not enough to adapt the model parameters to the data characteristics. We observe that enlarging training data (25%) improves the prediction accuracy. However if the dataset becomes large the prediction error increases slightly. Thus, enlarging the training does not really improve the prediction error because the model is over-trained i.e. the information contained in the data exceeds the model capacity.



(a) Auckland-VIII data set.



(b) CESCA-I data set.



(c) WAuckland-VIII data set.

Figure 8. Standard deviation and prediction error (RMSE) vs granularity.

It is to be outlined that this result is important because it shows that only a small set of data is needed to identify accurate parameters for the prediction model. Besides, using small training set allows reducing the processing time for the training phase.

We also note that when the traffic nature and characteristics change, the model should be trained to take into account these new characteristics. The longetivity and the robustness of the model depend on the variability of the data characteristics. However, simulations done to predict the traffic for 30 minutes show that the model still efficient for this time interval.

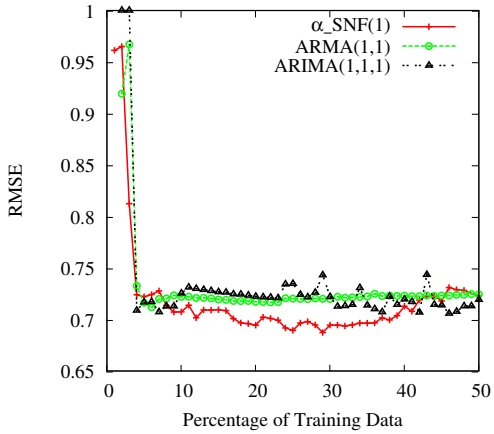


Figure 9. Prediction error for the Auckland-VIII data set using different size of training data (granularity 1 second).

D. Prediction Using Exogenous Variables

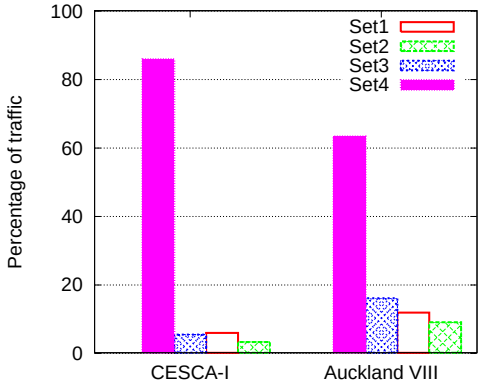
In this section, we focus on the use of sampled data for predicting the throughput $y(t, bps)$. Based on the analysis of the packet size distribution presented in Section IV-B, we divided the traffic into 4 sets:

- Set1 : packets which size belongs to [0:100 bytes[
- Set2 : packets which size belongs to [100:500 bytes[
- Set3 : packets which size belongs to [500:800 bytes[
- Set4 : packets which size belongs to [800:1500 bytes]

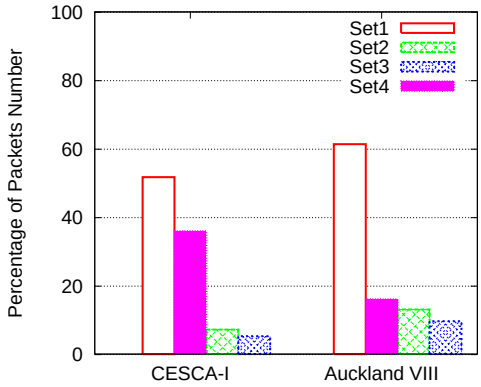
We note by $seti(t-1, bps)$ and $seti(t-1, pps)$ the throughput of the traffic belonging to Set i at the time $t-1$ respectively expressed in Mbps or packets per second (pps).

Fig. 10 depicts the percentage of different sets as generated traffic in Megabit and as packets number. It shows that for the CESCA traffic, Set4 constitutes 85% of the traffic but only 35% of the total number of packets. Similarly, large packets (Set4) in the Auckland link constitute only 16% of the packets but generates more than 60% of the traffic. Besides, more than 60% of the packets are from small packets (Set1), they constitute less than 15% of the generated traffic in the Auckland link and less than 8% of the CESCA traffic. This observation concurs with the findings of Shao et Al. [21] that the traffic pattern is bimodal: most traffic is carried by a small number of packets and most packets carry smaller number of bytes.

Fig. 11(a) shows the mutual information (Eq.10) calculated between the throughput $y(t, bps)$ and each candidate input variable which could be the last lag ($y(t-1, bps)$ or an exogenous variable like $y(t-1, pps)$, $set4(t-1, bps)$, $set4(t-1, pps)$, $set3(t-1, bps)$, $set2(t-1, bps)$, $set1(t-1, bps)$). It shows that the throughput $y(t, bps)$

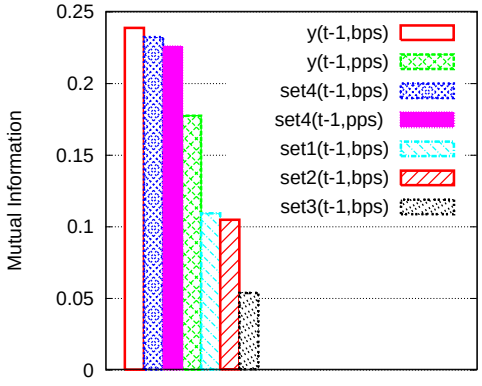


(a) Repartition considering the generated traffic.

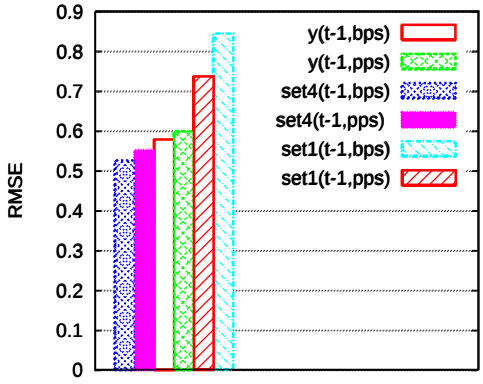


(b) Repartition considering the number of packets.

Figure 10. Percentage of different sets.



(a) Mutual information of the exogenous variables.



(b) Prediction error using exogenous variables.

Figure 11. Experimental results using exogenous variables (using Auckland-VIII data set).

is very correlated with the last lag $y(t-1,bps)$, $set4(t-1,bps)$, and $set4(t-1,pps)$. The other variables are less correlated with $y(t,bps)$. Hence, it suggests using these variables as input for the α_SNF model. The figure also shows that Set1, Set2 and Set3 are less correlated to the throughput $y(t,bps)$.

In order to validate this observation, the α_SNF model was applied using the exogenous variables. We used the same methodology (Fig. 1) but we replaced the $inputs(t-1)$ by the value of the exogenous variable at the time $t-1$ in order to predict the throughput $y(t,bps)$ at the time t.

Fig. 11(b) shows the RMSE obtained for predicting the throughput using a candidate variable for each experiment. We observe that using $set4(t-1,bps)$ or $set4(t-1,pps)$ as an input variable gives the best performance in terms of prediction error. The prediction using $set1(t-1,bps)$ or $set1(t-1,pps)$ provides high prediction error. That is, using Set1 (packets which size ≤ 100 bytes) does not characterize the traffic even though Set1 constitutes more than 50% of packets number.

It is clear that we can characterize the whole traffic by using only the large packets (set4) which constitute 16% of the total number of packets. We prove that sampled traffic on the basis of the packets size - expressed either in pps or as generated traffic (in Mbps) - could be used to predict more efficiently the throughput than last lag $y(t-1,bps)$.

VI. Conclusion and Future Work

In this paper, we present an analysis of prediction performance of training-based models using extensive sets of real Internet measurements. We show that enlarging training data set does not really improve traffic predictability. We find that with only 10% of the measurements are quite sufficient to obtain the same prediction error. We show the training-based model can capture, during the training phase, the strong correlation of the traffic with long-range dependence. Then, one lag is practically sufficient to perform quite accurate prediction regardless of the used granularity. Finally, we investigate using exogenous variables as inputs for the α_SNF model in order to predict the throughput. When the throughput history could not be available, the use of exogenous variables is very interesting especially when those parameters are easier to measure. The considered exogenous variables are number of packets (pps), the sampled traffic based on packets size expressed in pps or Mbps. We have found that traffic behavior depends on large packets (size ≥ 800bytes).

Thus, the performance of throughput prediction is improved when using the throughput of large packets expressed in pps or Mbps as input for the prediction model.

The paper analyzes only some aspects of Internet traffic prediction and many issues require further study, including the effects of some other parameters like the number of flows, packet loss, cross traffic nature etc. on traffic predictability.

ACKNOWLEDGMENT

We thank the editor and the reviewers for their valuable comments.

REFERENCES

[1] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on global Internet," *In IEEE Journal of Selected Areas in Communications*, vol. 13, pp. 1465–1480, 1995.

[2] S. F. Bush, "Active virtual network management prediction," *In Proc. of the* 13^{th} *Workshop on Parallel and Distributed simulation*, pp. 182–192, May 1999.

[3] C. Casetti, J. F. Kurose, and D. F. Towsley, "A new algorithm for measurement-based admission control in integrated service packet networks," *In Proc. of the International Workshop on Protocols for High-Speed Networks*, pp. 13–28, October 1996.

[4] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226–244, 1995.

[5] K. Park, G. Kim, and M. Crovella, "On the effect of traffic self-similarity on network performance," *In Proc. SPIE Int. Conf. Performance and Control of Network Systems*, Nov. 1997.

[6] W. Leland, M. S. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, 1994.

[7] P. Abry and D. Veitch, "Wavelet analysis of long-range-dependent traffic," *IEEE Trans. Info. Theory*, vol. Vol.44, pp. 2–15, Jan. 1998.

[8] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "Modeling Internet backbone traffic at the flow level," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, Aug. 2003.

[9] P. Owezarski and N. Larrieu, "Internet traffic characterization - an analysis of traffic oscillations," 7^{th} *IEEE International Conference on High Speed Networks and Multimedia Communications*, 2004.

[10] D. Papagiannaki, N. Taft, Z. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic: Observations and initial models," *In Proc. of IEEE Infocom*, April 2003.

[11] A. Scherrer, N. Larrieu, P. Borgnat, P. Owezarski, and P. Abry, "Non-gaussian and long-memory statistical modeling of Internet traffic," 4^{th} *International Workshop on Internet Performance, Simulation, Monitoring and Measurements*, February 2006.

[12] Y. Gao, G. He, and J. C. Hou, "On leveraging traffic predictability in active queue management," *In Proc. IEEE INFOCOM*, June 2002.

[13] G. He and J. C. Hou, "On exploiting long-range-dependency in measuring cross traffic," *In Proc. IEEE INFOCOM*, April 2003.

[14] G. He, Y. Gao, J. C. Hou, and K. Park, "A case for exploiting self-similarity of Internet traffic in TCP congestion control," *In Proc. IEEE ICNP*, Nov. 2002.

[15] A. Sang and S. qi Li, "A predictability analysis of network traffic," *In Proc. of IEEE INFOCOM*, March 2000.

[16] P. A. Dinda and D. R. O'Hallaron, "An evaluation of linear models for host load prediction," *The* 8^{th} *IEEE International Symposium on High Performance Distributed Computing*, 1999.

[17] N. K. Groschwitz and G. C. Polyzos, "A time series model of long-term NSFNET backbone traffic," *In Proc. of the IEEE International conference on Communications*, May 1994.

[18] H. Tong, C. Li, and J. He, "A boosting-based framework for self-similar and non-linear Internet traffic prediction," *In Proc. of International Symposium on Neural Networks (ISNN)*, August 2004.

[19] Y. Qiao, J. Skicewicz, and P. Dinda, "An empirical study of the multiscale predictability of network traffic," *In IEEE Proc. of HPDC*, 2003.

[20] Y. Xinyu, Z. Ming, Z. Rui, and S. Yi, "A novel LMS method for real-time network traffic prediction," *In Proc. of Computational Science and Its Applications (ICCSA)*, May 2004.

[21] Q. Shao and L. Trajkovic, "Measurement and analysis of traffic in a hybrid satellite-terrestrial network," *In Proc. SPECTS 2004*, pp. 329–336, July 2004.

[22] G. He and J. C. Hou, "An in-depth analytical study of sampling techniques for self similar Internet traffic," *In Proc. of the* 25^{th} *IEEE International Conference On Distributed Computing Systems*, September 1993.

[23] M. F. Zhani, H. Elbiaze, and F. Kamoun, "α_SNFAQM: An active queue management mechanism using neurofuzzy prediction," *In Proc. of the* 12^{th} *IEEE Symposium on Computers and Communications (ISCC)*, July 2007.

[24] Q. He, C. Dovrolis, and M. Ammar, "On the predictability of large transfer TCP," *SIGCOMM'05,Philadelphia*, August 2005.

[25] F. Rouai and M. Ahmed, "A new approach for fuzzy neural network weight initialization," *INNS-IEEE International Joint Conference on Neural Network (IJCNN)*, vol. 2, 2001.

[26] N. Sadek and A. Khotanzad, "Dynamic bandwidth allocation using a two-stage fuzzy neural network based traffic predictor," *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 3, pp. 2407–2412 vol.3, July 2004.

[27] A.-M. Yang, X.-M. Sun, C.-Y. Li, and P. Liu, "A neuro-fuzzy method of forecasting the network traffic of accessing web server," 2^{nd} *International Conference on Fuzzy Systems and Knowledge Discovery, Lecture Notes in Computer Science*, vol. 3613, August 2005.

[28] R. Wolski, "Forecasting network performance to support dynamic scheduling using the network weather service," *In Proc. of the* 6^{th} *High-Performance Distributed Computing Conference (HPDC)*, August 1997.

[29] S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in fortran," *Cambridge University Press*, 1986.

[30] "RFC 879 - TCP maximum segment size and related topics."

[31] C. Shannon, "A mathematical theory of communication," *Bell System Tech*, vol. 27, pp. 379–423,623–659, 1948.

[32] J. Cohen, "Statistical power analysis for the behavioral sciences (2nd ed.)," *Hillsdale, NJ: Lawrence Erlbaum Associates.*, 1988.

Mohamed Faten Zhani was born in Kairouan, Tunisia on January 25, 1980. He received the Engineering and the M.S. degrees from the National school of computer sciences in Tunis in 2003 and 2005, respectively. He is currently a Research Assistant in the department of Computer Science in the University of Quebec in Montreal and he is working toward the Ph.D. degree. His research interests include network traffic modeling, analysis and prediction, optical networks, congestion control and active queue management.

Halima Elbiaze received the BS degree in Applied Mathematics from the University of Rabat, Morocco, in 1996. She received the MS and Ph.D. degrees in Computer Science from the University of Versailles, France, in 1998 and 2002, respectively. She is currently an Assistant Professor in the Department of Computer Science, University of Quebec in Montreal, Canada. Her research interests are in the area of Quality of Service, performance evaluation and traffic engineering for high speed networks (IP/WDM, TCP/IP, ATM, FR, etc.).

Farouk Kamoun was born in Sfax, Tunisia on October 20, 1946. He received the Engineering Degree from Ecole Suprieur dElectricit, Paris, France in 1970 and the Ph.D. degree in computer science, in 1976, from UCLA, where he participated in the ARPA Network Project and did his research on design considerations for large computer communication networks. He is currently Professor and Director of CRISTAL Research Laboratory at the Ecole Nationale des Sciences de l'Informatique, Tunisia. His research areas of interest deal with protocols for ad-hoc networks, quality of service and network measurements.