ISSN 1392 – 1215 ELECTRONICS AND ELECTRICAL ENGINEERING 2007. No. 6(78)
ELEKTRONIKA IR ELEKTROTECHNIKA

T180 TELECOMMUNICATIONS ENGINEERING
TELEKOMUNIKACIJŲ INŽINERIJA

Study on Internet Traffic Prediction Models

G. Rutka, G. Lauks
Faculty of Electronics and Telecommunication, Riga Technical University,
Riga, Azenes st. 12-317, LV-1048, phone:+371 29627600, e-mail: gundegarutka@tvnet.lv

Introduction

The predictability of network traffic is of significant interest in many domains, including adaptive applications [1], congestion control [2], admission control [3], wireless and network management [4]. An accurate traffic prediction model should have the ability to capture the prominent traffic characteristics, e.g. short and long dependence, self similarity in large-time scale and multifractal in small-time scale. For these reasons time series models are introduced in network traffic simulation and prediction.

Accurate traffic prediction may be used to optimally smooth delay sensitive traffic [5] or dynamically allocate bandwidth to traffic streams [6].

The problem of traffic prediction is a standard time series prediction task, the goal of which is to approximate the function that relates the future values of a variable of the previous observations of that variable. The estimation value is given by:

$$\hat{x}(n) = F(x(n-1), x(n-2), ..., x(n-p)) + e(n) . \qquad (1)$$

Neural networks offer interesting alternative solutions to many problems in communications. Useful applications have been designed, built and commercialized and much research continues in hopes of extending current success.

Self- similarity

For a self similar time series:

$$\{X\}=\{X_1, X_2, \ldots, X_k\}. \qquad (2)$$

The m-aggregate $\{X_k^{(m)}\}$ with its k-th term:

$$X_k^{(m)} = \frac{X_{km-m+1} + ... + X_{km}}{m}, \text{ where } k=1,2,3,\ldots \qquad (3)$$

The Hurst parameter H in (2) is in the range $0.5<H<1$ and it characterizes the process in terms of the degree of self-similarity and long time dependence. The degree of self-similarity and long-range dependence increases as $H \rightarrow 1$. In our experiments self-similarity will be estimated by the use of variance-time plot method. This is one of the easiest methods how to estimate Hurst's coefficient. In the process the variance of aggregate the self-similar process is defined:

$$VAR(X^{(m)})= VAR(X)/m^{\beta}. \qquad (4)$$

In the (4) β is calculated from the equation:

$$H=1-\beta/2. \qquad (5)$$

The (4) can be rewritten is the following form:

$$\log\{ VAR (X^{(m)})\} \sim \log\{ VAR(X) \} - \beta \log \{m\}. \qquad (6)$$

If VAR(X) and m are plotted on a log-log graph then by fitting a least square line through the resulting points we can obtain a straight line with the slope of $-\beta$ [7,8,9,10].

Neural networks

Many authors have applied many different neural network (NN) architectures and algorithms to explore traffic modeling task [9,11,12]. The neural network models most widely used in time series prediction problems are based in feedforward NN with backpropagation learning algorithm. Those models can be used as one-step as multi-step prediction. They consist of approximating the function F by a multilayer feedforward neural network. Introducing the vector (x(k),…mx(k-d)) as the k-th network input pattern, the one step predicted value by the neural network model can be written as follows:

$$\tilde{x}(k-1) = \tilde{F}(x(k), ...x(k-d), W1) . \qquad (7)$$

where W_1 is the parameter set of the neural network model, which is obtained using the backpropagation algorithm. The update of the parameter set is based on the local error between the measured and predicted values:

$$e(k+1) = \frac{1}{2} \cdot (x(k+1) - \tilde{x}(k+1))^2 . \qquad (8)$$

Poisson Process

Poisson Process (PP) is one of the most important models in queuing theory.

PP is characterized by a rate parameter λ, also known as intensity. Intensity characterizes the number of events in time interval (t,t+τ). The relation is given as

$$P[(X(t+\tau)-X(t))=k]=\frac{e^{-\lambda\tau}(\lambda\tau)^k}{k!}, k=0,1,... \qquad (9)$$

where X(t+τ)-X(t) describes the number of events in time interval (t,t+τ).

Proposed models

In our research we have adopted a three layer feedforward neural network (FFNN) which consists of an input layer, an output layer and a hidden layer. Each of these layers consists of one or more neurons (processing units) – during the simulation process we modulate different number of neurons in each layer. Fig.1 shows an example of three-layer FFNN with four neurons in the input layer, three neurons in the hidden layer and one neuron in the output layer. The layers of our FFNN predictor is feedforward connected with sigmoid or units in the input and output layers.

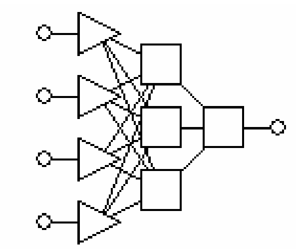

Fig. 1. A three-layer FFNN (4,3,1)

In the prediction process we do the following steps: Firstly, we construct a NN by learning equations - NN has been initialized (this property defines the function used to initialize the network's weight matrices and bias vectors).

Once the network weights and biases have been initialized, the network is ready for training. The training process requires a set of examples of proper network behavior - network inputs p and target outputs t. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function. The training and simulation accuracy is being determined by three types of error: mean squared error (MSE), generalized MSE (MSEREG) and the sum of squared error performance function (SSE)

$$MSE=\frac{1}{N}\sum_{i=1}^{N}(e_i)^2=\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x}_i)^2 . \qquad (10)$$

It is possible to improve generalization if we modify the performance function by adding a term consists of the mean of the squares of the network weights and biases:

$$MSEREG=\gamma MSE+(1-\gamma)MSW , \qquad (11)$$

where γ is the performance ratio and

$$MSW=\frac{1}{n}\sum_{j=1}^{n}w_j^2 . \qquad (12)$$

Using the performance function will cause the network to have smaller weights and biases. This will force the network response to be smoother and less likely to over fit.

For description for the sum of squared differences between the network targets and actual outputs for a given input vector or set of vectors we use the sum squared error performance function (SSE).

Secondly, we predict the future values of given traffic data. The forecasting is evaluated based on the constructed network. We use two choices for adopting prediction data. One is used last observed data just before the forecasting. The conception is called as "one step ahead prediction" or "short-range prediction". Moreover, the quantity is not gotten by the last data, but information from rather past data. The other is used output data calculated only by the neural network. The conception is called as "long- range prediction".

Review of studied cases (results)

Our research is emphasized to self- similar traffic prediction using neural networks. Traffic data is taken from website http://freestats.com/ , collected for one year. Another data trace is collected using website access statistics of local area network users using access to the site www.fotoblog.lv. As the third type of traffic data we analyze simulated PP depending on the parameter λ. In our experiments we analyze traffic sources as follows:

Table 1. Summary of the traffic data used in the study

Name	Observations	Step
Freestats statistics	8760	1 hour
Fotoblog statistics	172800	1 sec
Poisson process with λ=11,9 (PP1)	8760	1 sec
Poisson process with λ=4,35 (PP2)	8760	1 sec
Poisson process with λ=3,5 (PP3)	8760	1 sec
Poisson process with λ=2 (PP4)	8760	1 sec
Poisson process with λ=1 (PP5)	8760	1 sec
Poisson process with λ=11,9 (PP6)	172800	1 sec
Poisson process with λ=4,35 (PP7)	172800	1 sec
Poisson process with λ=3,5 (PP8)	172800	1 sec
Poisson process with λ=2 (PP9)	172800	1 sec
Poisson process with λ=1 (PP10)	172800	1 sec

For statistical analyses and neural network testing we use program package "MATLAB 6.5".

The main points of interest of our research are:

1. Traffic predictability.
2. The accuracy of the predicted traffic.
3. The maximum prediction interval for a prediction error minimum (how far in the future network traffic can be predicted with confidence).

For these reasons, firstly, we have deeply studied the character of the statistical material (traffic data). Accordingly to that we have calculated and proved that traffic data is self-similar.

Secondly, we have estimated the simulation and training error for different prediction models using neural network models (we use MSE, MSEREG and SSE).

Thirdly, we have tested and estimated the prediction interval. Using different step prediction we have verified the prediction accuracy.

Analyzing the self similarity of these traces we have calculated the Hurst parameter. The results are shown below in the Fig.2.

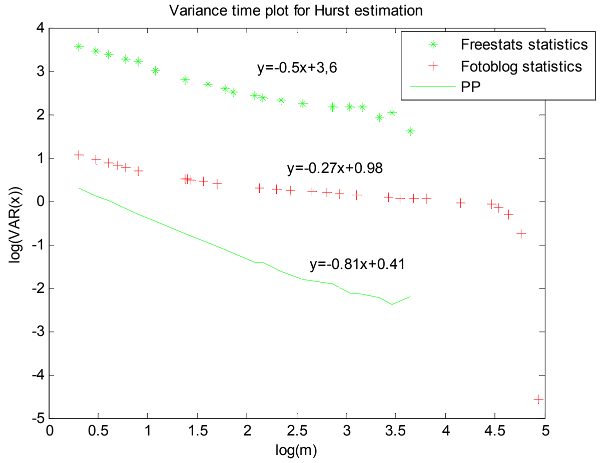

Fig. 2. The Hurst parameter estimation with the variance-time plot

The variance-time curve (Fig. 2) shows an asymptotic slope that is easily estimated to be about –0.50 for Freestats trace, –0.27 for Fotoblog trace and –0,81 for PP2, resulting in a practically identical estimate of the Hurst parameter H of about 0.75 for Freestats trace, 0.865 for Fotoblog trace and 0.60 for PP2.

The results of Hurst parameter estimation for the PP with different λ is presented in Table 2.

Table 2. The Hurst parameter estimation with the variance-time plot.

Data	**H**
PP1	0.57
PP2	0.60
PP3	0.54
PP4	0.56
PP5	0.52
PP6	0.50
PP7	0.45
PP8	0.50
PP9	0.45
PP10	0.50

We must exclude the PP6, PP7, PP9 and PP10 from our experiments because the Hurst parameter of these traces is out of the range 0.5<H<1.

For future value prediction modeling we used FFNN. For this reason, firstly, we estimated the simulation error for NN models using different traffic data. Some results are shown in Table 3.

Table 3. The simulation and training errors

Traffic data	**Simulation and training error**		
	MSE	MSEREG	SSE
Freestats statistics	2,18e^{-31}	0,03	2,00e^{-27}
Fotoblog statistics	3,80e^{-32}	0,06	8,60e^{-27}
PP1	1,09e^{-35}	0,21	1,38e^{-32}
PP2	1,20e^{-35}	0,2	4,22e^{-31}
PP4	1,08e^{-29}	0,21	0

The results in Table 3 show that the MSE and SSE give the smallest value. The NN simulation and training time depends on what kind of error network performs. For this reasons for future experiments we choose only MSE.

The prediction accuracy according to the NN model is as follows in Table 4 - Table 6 according to the K-step ahead prediction:

Table 4. The prediction error of PP4 depending on different parameters

Simulation steps	**Prediction error (MSE)**			
	K=1	K=10	K=25	K=100
30	2.00	1.89	1.85	1.36
300	2.01	1.90	1.79	1.41
1000	2.03	1.93	1.81	1.23
10 000	2.03	1.98	1.91	1.29

Table 5. The prediction error of Freestats trace depending on different parameters.

Simulation steps	**Prediction error (MSE)**			
	K=1	K=10	K=25	K=100
30	5566	5524	5075	5230
300	5103	5385	5408	5316
1000	5120	5096	5420	5320
10 000	5100	5480	5042	5303

Table 6. Summary of the best prediction results

Traffic data	**MSE**	**Input neurons**	**Hidden neurons**	**Simulation steps**	**Time of training**
Freestats	5096	10	10	10000	20min
Fotoblog	16.02	10	1	30	2h
PP4	1.39	10	100	30	1h

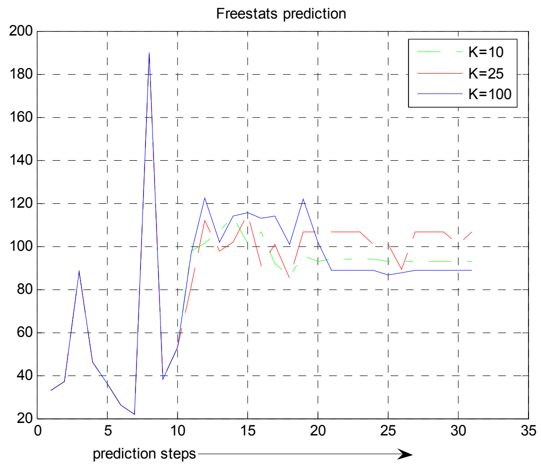

Fig. 3. The prediction results of Freestats trace

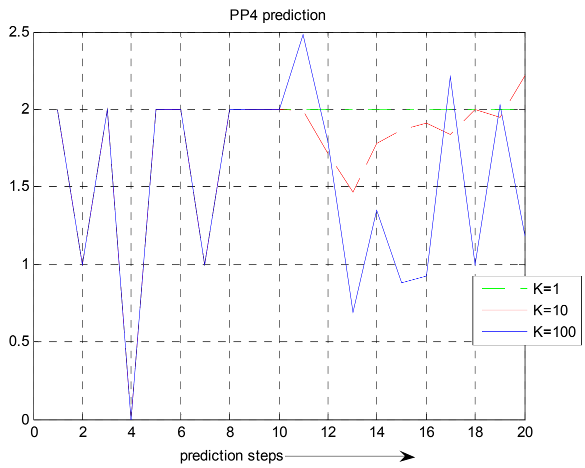

Fig. 4. The prediction results of PP4 trace

For illustrative view we have shown the prediction results in Fig. 3–4. In the prediction process we have used all observations, but for better illustrative view we have shown the last 10 observations and 10-15 future predicted values.

Conclusions

This paper investigates the predictability of network traffic in order to explore the potentials of multi step traffic prediction for network capacity purposes.

Our analysis is based on the PP model and real traffic models (Freestats and Fotoblog) for their theoretically available optimal predictors.

Different traffic statistics play different roles in predictability. Moreover, numerical studies of real traffic traces verify the prediction of real network traffic is not so easy and almost impossible when we speak about long-range prediction. Contrary, the simulated statistics (Poisson Process) gives the average prediction error from 1.3- 2%.

References

1. **Stemm M., Seshan S., Katz R. H.** A network measurement architecture for adaptive applications // Proceedings of INFOCOM 2000. – Vol. 1. – P. 285–294.
2. **Jacobson V.** Congestion avoidance and control // Proceedings of the ACM SIGCOMM 1988, Symposium. – P. 314–329.
3. **Jamin S., Danzing P., Schenker S., Zhang L.** A measurement based admission control algorithm for integrated services packet networks // Proceedings of ACM SIGCOMM. – 1995. – P. 56–70.
4. **Kim M., Noble B.** Mobile network estimation // Proceedings of 7th Annual International Conference on Mobile Computing and Networking. – P. 298–309.
5. **Ott T. J., Lakshman T. V., Tabatabai A.** A scheme for smoothing delay sensitive traffic offered to ATM networks // Proc. IEEE INFOCOM ’92. – 1992. – P. 776–785.
6. **Moh W. M., Chen M. J., Chu N. M., Liao C. D.** Traffic prediction and dynamic bandwidth allocation over ATM: a neural network approach // Computer Communications. – 1995. – Vol.18, No. 8. – P. 563–571.
7. **Jon M. Peha.** Protocols can make traffic appear self-similar // IEEE/AMC/SCS Communication networks and distributed systems modeling and simulation conf. – Jan. 1997. – P. 47–52.
8. **Robert S., Le Boudee J. Y.** A Markov modulated process for self-similar traffic. – Sept 23, 1995.
9. **Jonckheere E. A., Silvester J. A.** Neural network modelling of self-similar teletraffic patterns. – 2003.
10. **Rose O.** Estimation of the Hurst parameter of long range dependent time series. – University of Wurzburg, Research report series. – Feb. 1996.
11. **Tong H., Li C., He J.** Internet traffic prediction based on boosting feedforward neural network. – 2004.
12. **Li H., Kozma R.** A dynamic Neural Network Method for Time Series Prediction Using the KIII Model // Proceedings IJCNN. – Portland. – 2003.

Submitted for publication 2007 02 28

G. Rutka, G. Lauks. Study on Internet Traffic Prediction Models // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 6(78). – P. 47–50.

A view of models used for Internet data traffic prediction using neural network applications is presented. We look at the problem of traffic prediction in the presence of self-similarity. Self-similarity is an important characteristic of traffic in high-speed networks that cannot be captured by traditional traffic models. Our experiments inspect performance and prediction error using feed forward neural networks. Ill. 4, bibl. 12 (in English; summaries in English, Russian and Lithuanian).

Г. Рутка, Г. Лаукс. Модели для предсказания трафика интернета // Электроника и электротехника. – Каунас: Технология, 2007. – № 6(78). – C. 47–50.

Представлены модели, используемые для предсказания нагрузки в интернете применяя нейронные сети. Рассматривается проблема предсказания трафика в случае самопохожести. Самопохожесть это очень важная особенность трафика в высокоскоростных сетях, которую невозможно встретить при обычной модели трафика. Наши эксперименты показывают ошибки предсказания нагрузки. Ил. 4, библ. 12 (на английском языке; рефераты на английском, русском и литовском яз.).

G. Rutka, G. Lauks. Interneto duomenų srauto prognozavimo modelių tyrimas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2007. – Nr. 6(78). – P. 47–50.

Apžvelgti modeliai, taikomi interneto apkrovos prognozei naudojant neuroninius tinklus. Duomenų srauto prognozavimo problema analizuota savaiminio panašumo atveju. Savaiminis panašumas yra svarbi duomenų srautų didelės spartos tinkluose charakteristika. Į ją negalima atsižvelgti taikant tradicinius duomenų srautų modelius. Tirtas neuroniniais tinklais pagrįsto modelio našumas ir prognozavimo paklaida. Il. 4, bibl. 12 (anglų kalba; santraukos lietuvių, anglų ir rusų k.).

DOI: 10.5755/j02.eie.10814