PACKET INTER-ARRIVAL TIME ESTIMATION USING NEURAL NETWORK MODELS

Thyda Phit and Kôki Abe
Department of Computer Science, The University of Electro-Communications
1-5-1 Chofugaoka Chofu-shi Tokyo 182-8585
pthyda@cacoa.cs.uec.ac.jp, abe@cs.uec.ac.jp

ABSTRACT

This paper presents the estimation methods of packet inter-arrival times. The approach is based on neural network models for predicting the packet inter-arrival times by learning patterns and characteristics of observed traffics. The proposed methods are compared with the familiar exponentially weighted moving average (EWMA) estimator. Evaluation results demonstrate the effectiveness of the proposed methods. We also investigate the effect of classifying the observed traffics according to their protocol types. The classification helps find patterns of the traffics.

KEY WORDS

Packet inter-arrival time, estimation, EWMA, neural network, backpropagation, classification

1. Introduction

Packet inter-arrival time estimation and measurement are integral parts of many traffic management, monitoring, and control tasks in packet-switched networks. The estimation of the inter-arrival time can be performed off-line or on-line due to the objective.

The packet inter-arrival time estimation is important and has been applied in many scenarios. For example, estimated values of packet inter-arrival time are used to measure the traffic rate for the QoS-enabled Internet [1]. The rate estimation is an essential part of call admission [2], link-sharing [3], and fair scheduling algorithm [4].

In Ref. [5] packet inter-arrival time is used for investigating unsolicited internet traffic or for identifying the abnormal or unexpected network activity. The packet inter-arrival time pattern can be used to identify attacks or network phenomena. Estimation of the end-to-end performance and its improvement are important for web transactions [6].

In the power management of network devices [7], the estimated inter-arrival time of the packets play a significant role for reducing the power consumption of network devices such as LAN Switches. An incoming packet is buffered and wakes up the switch interface. The decision on whether to sleep is based on the estimation of next packet arrival (packet inter-arrival time). If the inter-arrival time is estimated to be long enough, interface goes to sleep, otherwise it stays awake.

This paper proposes methods of estimating the packet inter-arrival times based on neural networks and evaluates the methods in terms of estimation accuracy and computational cost by applying them to real traffic samples.

The remainder of this paper is organized as follows. Section 2 describes the characteristics of packet inter-arrival time. Section 3 proposes the methods of estimating the packet inter-arrival time based on neural networks. Section 4 describes the traffic data used for evaluating the proposed methods. Section 5 gives the evaluation results and discussion. Section 6 closes the paper with conclusions.

2. Characteristics of Packet Inter-Arrival Time

The traffic behaviour in a network has serious implications for the design, control, and analysis of the network. By analyzing collected data, it was demonstrated that the characteristics of packet inter-arrival time series is self-similar and long range dependence [6]. It is said that packet arrival process for internet has heavy tailed distribution characteristic. The fact that distribution of inter-arrival packet times is characterised by self-similarity implies that the future events depend on what happened before. If the distribution of inter-arrival packet times had the Poisson nature, the past behaviour would not affect events following in time. From the observations, it is natural to consider that various machine learning methods are applicable to the prediction of packet inter-arrival time if appropriate history information is taken into account in the learning process. Among machine learning technologies, we employ neural network models in the study because their structures are simple and many examples of successful applications of neural networks have been reported.

Looking into packet flows in networks, some protocols generate periodic traffics and some others generate random traffics which arrive randomly in the networks. The traffic flow is changed during day and night, weekday and holiday etc. The observation that packet streams consist of several types of traffic with respect to time and space implies that machine learning on classified traffics will increase the accuracy of packet inter-arrival estimation.

3. Packet Inter-Arrival Estimation

Let $x[k]$ denote the observed inter-arrival time between k^{th} and $(k-1)^{th}$ packets in a traffic, k=1,…,N. The estimation means to predict $x[k]$ by using history of previous inter-arrival times $x[k-n],\ldots,x[k-1]$.

This paper proposes methods of calculating the estimated value. Our approaches use neural networks as estimation model for predicting inter-arrival time. The approach simply takes advantage of using fixed structure with a well defined analytical model that is able to predict a traffic pattern after learning the pattern dynamics through the use of information available in traffic samples.

In what follows, we first introduce a conventional method which has been used frequently in network applications, and then present our proposed methods.

A. EWMA Method

EWMA (Exponentially Weighted Moving Average) is an ideal maximum likelihood estimator that employs exponential data weighting and is widely used in many different areas and applications. EWMA is a recursive estimator which uses a weighting factor to shape its memory.

Packet inter-arrival time is estimated as:

$$\overline{x[k]} = \alpha \overline{x[k-1]} + (1-\alpha)x[k-1]$$

where $\overline{x[k]}$ and $\overline{x[k-1]}$ are a new estimated inter-arrival time and the prior estimated values respectively. $x[k-1]$ is the current observed value. The term $0 \le \alpha < 1$ is a specific weight constant and its value is obtained from empirical study.

The merit of the method is that it is easy and fast for estimating inter-arrival time.

B. Proposed Method 1: Two-Layer Linear Neural Network Estimator

This method uses two-layer linear neural network (Perceptron) [8] as estimator model to calculate inter-arrival time. The structure of the model is shown in Figure 1. It consists of an input layer with n neurons and an output layer with one neuron.

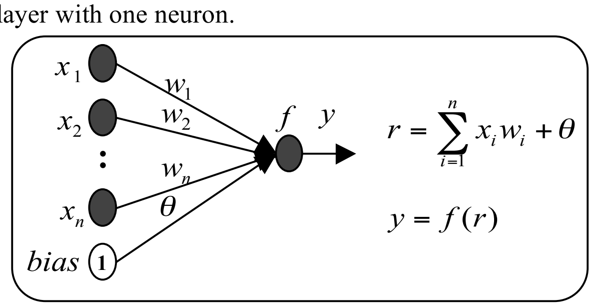

Figure 1 Two-layer neural network.

The inputs $x_1, x_2, .., x_n$ are input samples, $w_1, w_2, .., w_n$ are weights, and θ is a threshold. In this method, estimated inter-arrival time y is given by a linear function $f(r) = r$, where r is a weighted sum of inputs.

The learning algorithm is based on a gradient-descent method in error space. The absolute error function is defined as:

$$E = 1/2(u-y)^2$$

where u is an observed value. In order to decrease the absolute error function, the weights are changed in the opposite direction of the gradient vector defined as:

$$\Delta w_i = -\frac{\partial E}{\partial w_i} = \eta(u-y)x_i \ , \qquad i = 1,\ldots,n$$

$$\Delta\theta = -\frac{\partial E}{\partial \theta} = \eta(u-y)$$

where η denotes a learning rate. Then, the weights and threshold are adjusted as

$$w_i(t) = w_i(t-1) + \Delta w_i \ , \qquad i = 1,\ldots,n$$

$$\theta(t) = \theta(t-1) + \Delta\theta$$

where t denotes an iteration of the learning process. Normally the learning rate is constant and set as a small value. If the learning rate is very small, the algorithm proceeds slowly but accurately follows the path of steepest descent in weight space. On the other hand, the larger rate may cause the system to oscillate and thereby to slow or prevent the network's convergence.

Because this method uses a linear function, it is easy to find or vary the leaning rate according to the input samples which is determined as:

$$\eta = (\sum_{i=1}^{n} x_i^2 + 1)^{-1}$$

With the specific and appropriate learning rate, it is proved that adaptive process is fast and converged quickly.

In an iteration of the estimation, the neural network is provided at time k with samples through $x[k-n]$ to $x[k-1]$ of the observed traffic, and the difference between sample $x[k]$ of the observed traffic and the neural network output is used to adjust the weighting function of the network accordingly. The neural network continues processing more information in consecutive iterations of the learning phase until the absolute error is less than an accepted error bound. When the absolute error is less than an accepted error bound, the training process is terminated. At that time we get a new updated weights and threshold, and then use it for predicting a new inter-arrival time. In the next iteration, sample $x[k-n]$ of the real traffic is discarded, $x[k-n+1]$ through $x[k]$ of the observed traffic pattern are used as new samples.

C. Proposed Method 2: Two-Layer Nonlinear Neural Network Estimator

This approach is to calculate the weight α of EWMA dynamically according to the traffic behaviour, then combine it with EWMA method to estimate inter-arrival time. We use two-layer nonlinear neural network model for obtaining the weight α of EWMA.

The model is illustrated in the same figure as Figure 1 with y replaced by α, i.e., $\alpha = f(r)$, where f is a nonlinear sigmoid function defined as $f = (1 + e^{-r})^{-1}$ with $0 \le f < 1$. As described in Method 1 above, the observed traffic histories are used for input samples. The samples are updated by the most recent data. In this model the weight α is varied between 0 and 1 due to the traffic status.

The training is based on a gradient descent in error space in the same way as Method 1. Because sigmoid function is nonlinear we can not calculate learning rate directly like in Method 1. A simple method of effectively increasing the learning speed is to modify the delta rule by including a momentum term [9]. The adjusted values of weights and threshold are defined as:

$$\Delta w_i(t) = \eta(u - \overline{x[k]})(\overline{x[k-1]} - x[k-1])(1-\alpha)\alpha x_i + \lambda \Delta w_i(t-1) \qquad i = 1,\ldots,n$$

$$\Delta \theta(t) = \eta(u - \overline{x[k]})(\overline{x[k-1]} - x[k-1])(1-\alpha)\alpha + \lambda \Delta \theta(t-1)$$

where λ is a positive constant which is determined experimentally. The learning rate η in the method is a positive constant as well.

D. Proposed Method 3: Multi-Layer Neural Network Estimator

In this method, a three-layer neural network with back-propagation algorithm is applied to inter-arrival time estimation. It consists of an input layer with n neurons, one hidden layer with m neurons, and an output layer with one neuron. Figure 2 illustrates the structure of the neural network. The sigmoid transfer function is utilized to generate the output of each neuron from its compound inputs. The output of each neuron is connected to the input of all the neurons in the succeeding layer after being multiplied by the weights.

The observed inter-arrival time samples are input to the first layer then we obtain the estimated value from the output layer. The training algorithm is based on back-propagation algorithm [9]. It propagates the output error to the preceding layer via the existing connections until the input layer. The neural model also has a bias term in input and hidden layers, but in the following explanation, it is abbreviated for simplicity.

We use the following notation for explaining the learning algorithm.

- $x_1 \ldots x_n$ are the input samples and used as the first layer inputs.
- $y_1 \ldots y_m$ indicate the outputs of the hidden layer
- z represents an output of the network
- v_{ij} denotes a weight of the connection between j^{th} neuron from input layer and the i^{th} neuron of hidden layer
- w_i denotes a weight of the connection between i^{th} neuron from hidden layer and a neuron of output layer

Thus,

$$r_i = \sum_{j=1}^{n} v_{ij} x_j, \quad y_i = f(r_i), \quad i = 1,\ldots,m$$

$$s = \sum_{i=1}^{m} w_i y_i, \quad z = f(s)$$

where f is the sigmoid function.

The adjustment is calculated with leaning rate and momentum term. Weight adjustments are defined as

$$\Delta v_{ij}(t) = \eta\{(u - z) f'(s) w_i . f'(r_i) x_j\} + \lambda \Delta v_{ij}(t-1)$$

$$\Delta w_i(t) = \eta\{(u - z) f'(s) y_i\} + \lambda \Delta w_i(t-1)$$

$$f' = f(1 - f)$$

The input inter-arrival times are normalized within the range [0.1, 0.9] before being passed to the neural network [10]. The estimated value from neural network is denormalized.

$$X_{normalized} = \frac{X_{input} - X_{\min}}{X_{\max} - X_{\min}}(HI - LO) + LO$$

$$X_{denormalized} = \frac{X_{output} - LO}{HI - LO}(X_{\max} - X_{\min}) + X_{\min}$$

where X_{input} is input value, X_{min} and X_{max} are the minimum and maximum of observed inter-arrival times respectively. X_{output} is an output of the neural network. HI and LI are set to 0.1 and 0.9 respectively.

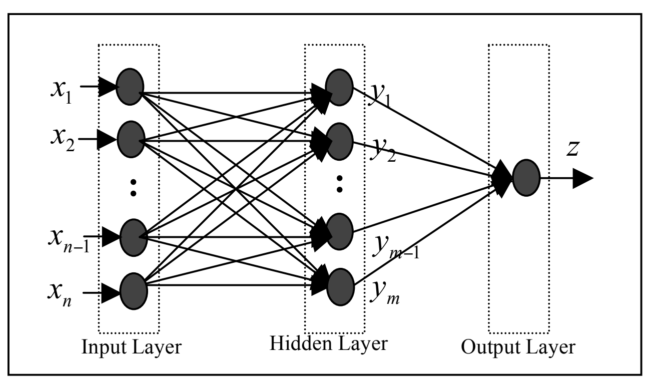

Figure 2 Three-layer neural network.

4. Traffic Data Used for Evaluation

To evaluate the methods proposed above, the traffic data collected at a real network is used, the topology of which is shown in Figure 3 [7]. The traffic data was measured for 2 hours on a regular workday by tcpdump for gathering traces and then the traces were analyzed using the ethereal tool. The entire traffic consists of different protocol traffics. They are IP protocol traffic (which consists of CDP, RPC[1], ICMP, and OSPF[2]), STP[3], CDP[4], and others. We used the traffics of two different connections, one is a connection between Switch FA to Host Y called Host Y and the other is a connection between Switch FA to Switch FB called Switch FB.

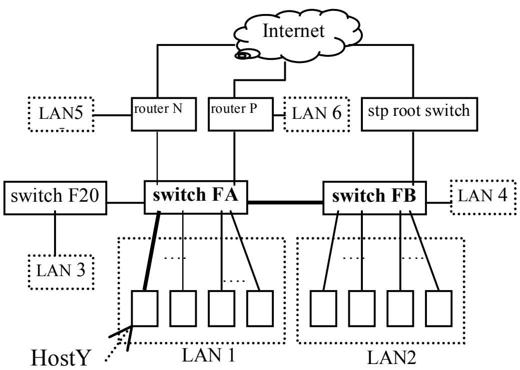

Figure 3 Network topology.

Table 1 Traffic data of Host Y.

Type 0	Type 1	Type 2	Type 3	Total
1421	5657	186	371	7635
IP	STP	CDP	Others	

Table 1 shows the collected traffic of Host Y to Switch FA connection during light activity time which was used in our evaluation. The traffic data was classified into types 0, 1, 2 and 3 which denote IP, STP, CDP, and others traffics, respectively.

Figure 4 illustrates a starting part of distribution of unclassified traffics of Host Y. In the figure, the horizontal axis is the arrival packet sequence number while the vertical axis is an inter-arrival time of the traffics. The inter-arrival time varies from 0 to 2.5 second. Some protocols generate periodic traffics like at number 11, 31 and so on. The entire traffic has similar patterns to the segment shown in the figure.

Type 2 traffic data of Host Y, as shown in Figure 5, is found to be characterized as repetition of sub-patterns. It appears to consist of several periodic traffics with different phases which are generated by CDP protocol of different devices.

[1] Remote Procedure Call
[2] Open Shortest Path First
[3] Spanning-Tree Protocol
[4] Cisco Discovery Protocol

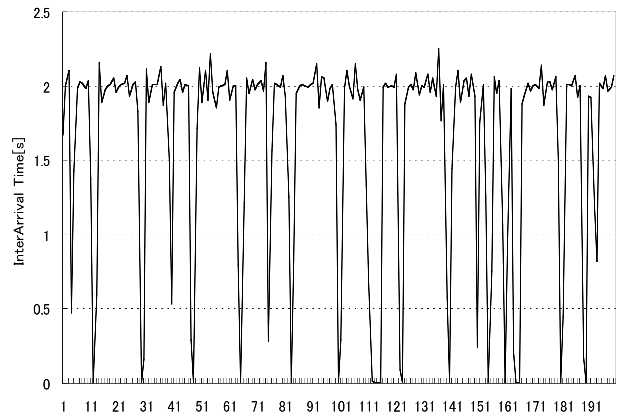

Figure 4 Inter-arrival time distribution of unclassified traffics of Host Y: 200 packets from the beginning.

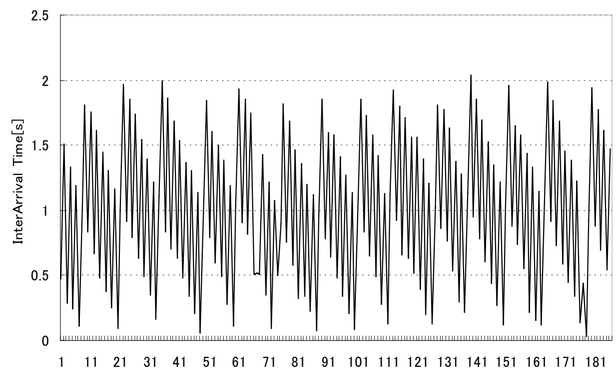

Figure 5 Inter-arrival time distribution of traffic Type 2 of Host Y.

Table 2 Traffic data of Switch FB.

Type 0	Type 4	Total
9980	20	10000
IP	ARP	

Table 2 shows the collected traffic of Switch FB connection during light activity time which is also used in our evaluation. We took 10000 traffics from colleted data then classified into Type 0 and 4 which denote IP and ARP packet protocol, respectively.

Figure 6 illustrates the distribution of Switch FB traffics. 200 traffics are plotted from the traffic number 1000. There are many packets with inter-arrival times ranging from 0.35 to 0.45 millisecond which belong to traffics generated by RPC protocol due to NFS (Network File Service) transactions.

Type 0 traffics of Host Y and Switch FB consist of IP packets generated by UDP, TCP, ICMP and OSPF protocols. Table 3 shows the percentages of those constituents. The UDP traffics of Switch FB mainly come from RPC procedure calls.

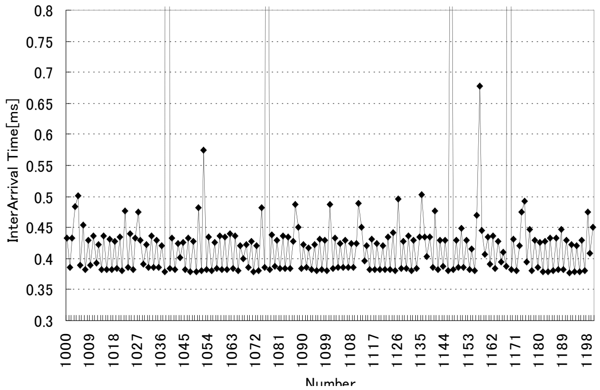

Figure 6 Inter-arrival time distribution of unclassified traffics of Switch FB.

Table 3 Constituents of Type 0 traffics.

	Host Y [%]	Switch FB [%]
UDP	8.89	96.93
(RPC)	(0)	(96.10)
TCP	58.47	2.78
ICMP	32.64	0
OSPF	0	0.29

5. Evaluation Results and Discussions

We evaluate the proposed methods of inter-arrival time estimation by applying them to the traffic described in the previous section. We examine the effect of traffic classification as well. The estimation methods were implemented in C language and executed on a Linux machine of AMD Athlon 64 3000+ operated at 2GHz with 2 GB main memory.

The methods were evaluated in terms of relative error of estimated time that is defined as below.

$$relative\_error = \frac{|real - estimated|}{real} \times 100[\%]$$

The parameters of the methods were determined experimentally so that the average of relative errors be minimum as shown in Table 4.

Table 4 Experimentally determined parameters.

Traffics		Host Y					Switch FB
		Type 0	Type 1	Type 2	Type 3	Total	Type 0
EWMA	α	0.001	0.95	0.95	0.375	0.125	0.4
Method1	n	2	3	13	13	2	3
Method 2	n	3	3	3	3	3	7
	η	0.1					
	λ	0.8					
Method 3	n	8	8	13	8	8	3
	m	20					
	η	0.05					
	λ	0.5					

5.1 Estimation Accuracy of Proposed Methods

Figure 7 shows the average of relative estimation errors on the whole traffic data with different types. Method 3 performs the best estimation except for Type 0 which consists of various kinds of packets including TCP, UDP, ICMP, and OSPF. The estimation accuracy for Type 0 traffic would increase if the packets are classified more finely.

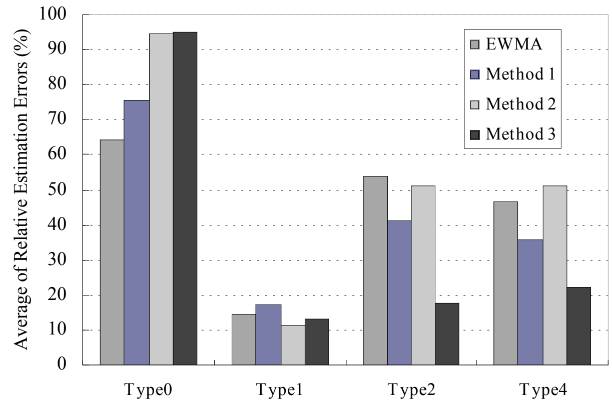

Figure 7 Average of relative estimation errors of classified traffic.

The detail of the estimation results compared with the observed traffics is shown in Figure 8, taking the Type 2 traffic data as an example. It illustrates a segment of traffics estimated by EWMA, Methods 1, 2 and 3. Figure 9 plots their relative errors of estimation. According to the figures, it is found that Method 3 yields the most accurate estimation of inter-arrival time among four estimators.

The estimation accuracy of EWMA and Method 2 were poor; they can not catch the characteristics of the observed traffics. Method 1 can follow the behaviour of the observed traffic except the starting part of the segment where the effect of the previous segment deteriorated the estimation accuracy. Method 3 has good performance over all the segment except the inter-arrival time of number 13 where the behaviour of the observed traffic changes abruptly.

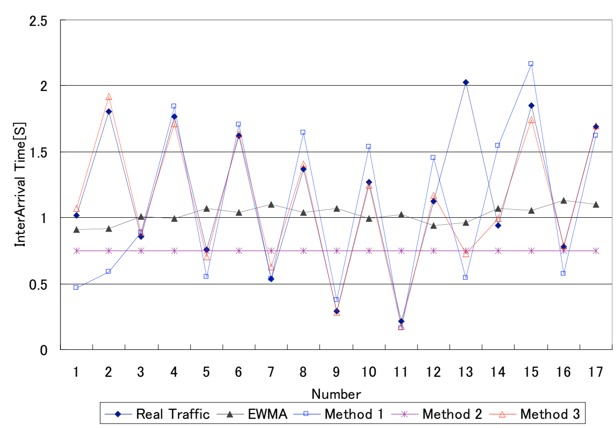

Figure 8 A segment of traffics Type 2 estimated by EWMA, Methods 1, 2 and 3 compared with the real one.

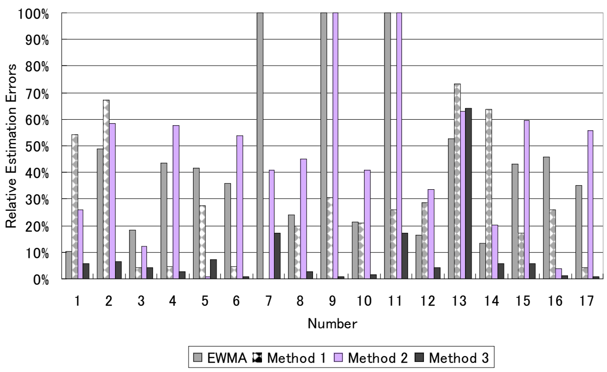

Figure 9 Relative errors of a segment of traffics Type 2 estimated by EWMA, Methods 1, 2 and 3.

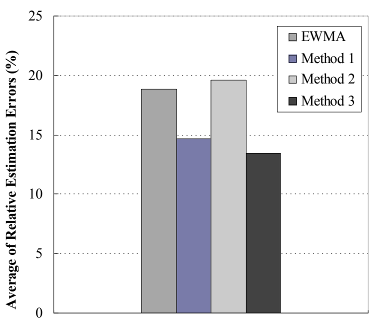

Figure 10 Average of relative estimation errors of Switch FB Type 0 traffics estimated by EWMA, Methods 1, 2 and 3.

The average of relative estimation errors of Switch FB Type 0 traffics are plotted in Figure 10. From the figure, the overall accuracy of estimation errors is good, being different from the case of Host Y. This is attributed to the difference between the behaviours of IP packets of Host Y and Switch FB traffics. Type 0 IP traffics of Switch FB are dominated by UDP packets generated by RPC due to NFS transactions. As shown in Figure 6 the pattern of inter-arrival time is characterized to be somewhat regular, containing a small number of packets with long inter-arrival times. On the other hand Type 0 IP traffics of Host Y consist of various kinds of packets including TCP, UDP, ICMP, and OSPF. As described previously, the estimation accuracy for Type 0 traffics of Host Y would increase if the packets are classified more finely. Method 3 is found to give the lowest errors for estimating Switch FB Type 0 traffics as the case for estimating Host Y traffics.

5.2 Effectiveness of Classification

Figure 11 shows the result of relative estimation errors for unclassified and classified traffics. The gray bars denote the averaged errors of estimation on unclassified traffics while the black bars denote errors averaged over individual estimation on classified traffics except Type 0. According to the figures, classifying the traffics always results in a good performance. It is considered that classification helps the estimators find distinct traffic patterns.

5.3 Static and Dynamic EWMA Methods

Method 2 is a dynamic version of EWMA where the weight α is determined dynamically by a neural network model. According to Figure 11, it is found that varying the weight gives more accurate prediction than conventional EWMA with constant α . However as seen in Figure 5, Method 2 fails to follow the behaviour of the observed traffics. It is found from static EWMA that increasing the weight α leads to better imitating the variation of the observed traffic but with considerable delays, while decreasing the weight makes the prediction insensitive to the current change of observed traffics. Therefore, even if the weight is adjusted dynamically in its magnitude as done in Method 2, improvements of the estimation performance are limited.

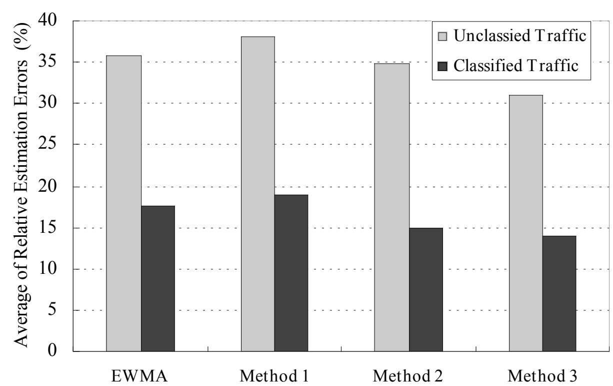

Figure 11 Average of relative estimation errors of unclassified and classified traffics.

5.4 Computational Complexity of Estimators

The time required for each method to estimate the traffics was measured. The result is shown in Table 5. The measurement was carried out for each method on the unclassified traffic with 7636 packets. EWMA with less estimation accuracy takes less time than the others. Method 3 which provides the most accurate prediction takes the longest time among all the methods. Thus, there is a trade-off between estimation accuracy and computational time. For instance Method 3 can be employed for such applications that require accurate estimation for light traffics. (Note that Method 3 takes 15 ms to predict the time when the next packet arrives.)

Table 5 Time of estimation of each method on the unclassified traffic with 7636 packets.

EWMA	Method 1	Method 2	Method 3
0.03[s]	0.04[s]	0.17[s]	116.18[s]

6. Conclusions and Future Work

In this paper, we proposed neural network models for estimating packet inter-arrival time of packet-switched networks. The results show that the models realize more accurate estimation than conventional EWMA method. We also investigated the effectiveness of classifying traffic by protocol types and found that it helps estimators find distinct traffic patterns, resulting in accurate estimation. The accuracy, however, is obtained at the cost of computational complexity. The preferable method to be employed for an application depends on whether it requires estimation accuracy or estimation time.

We need to further evaluate the proposed methods by applying them to the other traffics observed in the real network shown in Figure 3.

Acknowledgments

We would like to thank Dr. M. Gupta with Computer Science Department of Portland State University for kindly providing us with the network traffic data collected at a real network. We also would like to thank the anonymous reviewers for numerous suggestions improving this manuscript. It was partially supported by JSPS Grant-in-Aid for Scientific Research (C) (2) 18500048.

References

[1] F. Agharebparast, and V. C. M. Leung, A new traffic rate estimation and monitoring algorithm for the QoS-enabled Internet, *Global Telecommunications Conference 2003*(*GLOBECOM '03.*), vol. 7, pp. 3883- 3887, Dec. 2003.

[2] M. Grossglauser and D. Tse, A framework for robust measurement based admission control, *IEEE/ACM Trans. Networking*, vol. 7(3), pp.293-309, June 1999.

[3] S. Floyd and V. Jacobson, Link-sharing and resource management models for packet networks, *IEEE/ACM Trans. Networking*, vol. 3 pp.365-386, Aug. 1995.

[4] F. Agharebparast and V.C.M. Leung, Efficient fair queuing with decoupled delay-bandwidth guarantees, in *Proc. IEEE Globecom'01*, San Antonio, TX, pp.2601-2605, Nov. 2001.

[5] J. Zimmermann, A. Clacrk and G. Mohay, The use of packet inter-arrival times for investigating unsolicited internet traffic, *Proc. The first International Workshop on Systematic Approaches to Digital Forensic Engineering* (*SADFE'05*), pp. 89 - 104, Nov. 2005.

[6] T. Kushida and Y. Shibata, Empirical study of inter-arrival packet times and packet losses, *Proc. 22nd International Conference on Distributed Computing Systems Workshops*(*ICDCSW'02*), pp.233 - 238, July 2002.

[7] M. Gupta, S. Glover, and S. Singh, A Feasibility study for power management in LAN switches. Proc. 12th *IEEE International Conference on Network Protocols*, pp. 361-371, Oct. 2004.

[8] M. Minsky and S. Papert, *Perceptrons*, Cambridge, MA, The MIT Press, 1969.

[9] S. Murray, *Neural Networks for Statistical Modeling*, Library of Congress Cataloging-in-Publication Data, 1994.

[10] Y. Sugai, *Learning and Its Algorithms, Chapter 4* (in Japanese), Morikita Publishing Co., 2002.