Prediction-Based Admission Control Schemes for Proportional Differentiated Services Enabled Internet Servers

Chenn-Jung Huang,[1] Yi-Ta Chuang[1] and Rui-Lin Luo[2]

[1]*Institute of Learning Technology, National Dong Hwa University, Hualien Taiwan 970;* [2]*Department of Computer Science, National Tsing Hua University HsingChu, Taiwan 300*

ABSTRACT

A common problem in contemporary web servers is the unpredictability of response time. Researchers have recently considered different admission control algorithms with differentiated service for web servers to complement the Internet differentiated service model, and thereby provide QoS support to users of the World Wide Web. However, most of these admission control mechanisms do not ensure the QoS requirements of all admitted clients under a bursty workload. Although enabled in web servers to improve the QoS guarantee predicament in previous literature, an Internet service model called proportional differentiated service remains impractical and incompatible with current Internet protocols. This work presents two algorithms for admission control and traffic scheduling algorithms of the web server under proportional differentiated service, which embed a time series predictor to estimate the client traffic load in the next time period. The time series predictor is implemented using four different approaches based on their successful prediction rate in the literature. The experimental results demonstrate that the proposed approaches can effectively realize proportional delay differentiation service in multiclass Web servers when the support vector machines algorithm is utilized as the time-series predictor.

Reprint requests to: e-mail: cjhuang@mail.nhlue.edu.tw

KEYWORDS

admission control, quality of service, proportional differentiated service, time series prediction, fuzzy logic systems, self-similarity, support vector machines

1. INTRODUCTION

As the Internet gains global popularity, the use of web servers to advertise and sell merchandise in business is increasing. Traditional web servers apply a first-come-first-served (FCFS) approach to provide service for client requests, introducing unpredictable response times for the clients when the incoming traffic is bursty. Customers who become frustrated by a long response time may terminate the network connection with the web servers without completing the business transaction, causing loss of revenue for the businesses and unsatisfactory quality of service (QoS). Although the study of QoS provision in network transmission, such as Integrated Services (IntServ) and Differentiated Services (DiffServ), has become active in the research communities in recent years, guarantee of network layer QoS alone might not be able to offer clients perceivable services when the servers are overloaded by the unexpectedly significant increase in connections.

Prioritized processing in web servers has been investigated recently (Eggert & Heidemann, 1999; Bhatti & Friedrich, 1999; Vasiliou & Lutfiyya, 2001; Chen et al. 2001; Chen & Mohapatra, 2002; Chen & Mohapatra, 1999; Kant & Mohapatra, 2001; Lee et al. 2002; Kanodia and Knightly, 2003; Ritter et al. 2000; Dovrolis et al. 2002). Eggert and Heidemann attempted to provide QoS service at the application level by grouping client requests into two classes as in the network layer, and restricting the process pool size and response transmission rate for different priority groups. Bhatti and Friedrich developed tiered service levels and overloaded the management model in their work and implement admission control mechanism by blocking low-priority tasks when the number of high priority jobs exceeded a predefined threshold. Both admission control algorithms adopt a conventional fixed-bandwidth leased line approach that satisfies the requirement of bursty workload. Although the service quality of high priority tasks is guaranteed, some bandwidth remains unused. Vasiliou and Lutfiyya proposed a QoS architecture that can adjust the number of requests for different priority groups according the performance of the

high priority group during the runtime. However, their architecture still degrades the service quality of low-priority tasks. Chen et al. presented service differentiating Internet servers to cope with the QoS service provided by the network layer. Their prioritized scheduling and task assignment schemes also significantly improve the service to high-priority tasks. However, low-priority task are still starved of resources due to the monopolization of the resources by high-priority tasks.

To enhance the performance of the low priority tasks, Lee et al. (2002) developed an admission control algorithm that enables proportional delay differentiated service (PDDS) (Leung et al. 2001; Li et al. 2000; Vuong & Shi, 2000; Li & Lai, 2001; Jin & Li, 2001) at application level. Under PDDS, client requests are first classified into different priority groups as in previously discussed approaches, and are then allocated services according to class in proportional to the ratios set in the service contracts. Client can then be charged for usage based on their maximum average waiting time QoS requirement. Kanodia and Knightly (2003) devised the latency-targeted multiclass admission control algorithm, which controls the QoS of each class by measuring requests and service latencies. The works of Lee et al. and Kanodia and Knightly might resolve the problem of starvation of low-priority tasks in other approaches. Unlike the use of HTTP tags and HTML links, however, the algorithm of Lee et al. requires clients to explicitly feed two parameters, maximum arrival rate, and maximum average delay time, into the server to launch the admission control mechanism. This approach is impractical in real life. Furthermore, the simulations conducted in model the aggregate request rate from all clients as a Poisson process, which is inconsistent with findingsthat the traffic from World Wide Web (WWW) transfers has self-similarity (Arlitt & Jin, 2000; Arlitt & Williamson, 1997; Huebner et al. 1998; Deng, 1996; Crovella & Bestavros, 1997; Adas, 1997; Paxson & Floyd, 1995). Therefore, the validity of the performance evaluation reports in (Lee et al. 2002; Kanodia & Knightly, 2003) is disputable.

This work presents admission control algorithms to enable PDDS at application level. The proposed algorithms adopt a prediction mechanism to predict the total maximum arrival rate and maximum average delay time of each priority task group for the next measurement period according to the arrival rate of each class during the current and the last three measurement periods. The admission control mechanisms then utilize the predicted values to determine the next client for service from one of the queues maintained for each priority task group. Significantly, the automatic computation of these two parameters by the system resolves the impracticality of

requiring the clients to specify. Moreover, WWW traffic is self-similar, and therefore has a predictable time series. Support vector machines (SVM) have been successfully employed in many areas, including time series prediction (Cao & Tay, 2003; Gestel et al. 2001; Zhu et al. 2002; Raicharoen et al. 2003), Internet traffic prediction, call classification for AT&T's natural dialog system, multi-user detection and signal recovery for a code division multiple access (CDMA) system. Additionally, many VLSI-chip-based solutions permit the SVM to be hardware-computed, and high-speed low-cost SVM chips have been introduced recently, making hardware implementation of SVM feasible (Anguita & Boni, 2001; Anguita et al. 1999a; Anguita et al. 1999b). This work therefore employs SVM to realize the prediction mechanism, and compares this approach with another well-known machine learning technique, namely the fuzzy logic system, which is renowned for its mathematical framework for handling real world imprecision, and which allows decision-making with estimated values under incomplete or uncertain information (Buckley & Eslami, 2002).

The rest of this paper is organized as follows. The proposed admission control schemes are introduced in Section 2. Section 3 introduces the prediction techniques employed in the admission control schemes, namely, the fuzzy logic system and support vector machines techniques. Section 4 presents the simulation results of comparing the proposed algorithms with the FCFS service model and with two representative time series predictors in the literature. Conclusions are presented in Section 5.

2. ADAPTIVE ADMISSION CONTROL SCHEMES

World Wide Web traffic has been observed to have self-similarity (Arlitt & Jin, 2000; Arlitt & Williamson, 1997; Crovella & Bestavros, 1997). The evidence shown in Box and Jenkins (1970) implies that WWW traffic is predictable because self-similar time-series can be forecast. We thus tend to incorporate a prediction algorithm in our admission control mechanism to estimate the ratio of the expected average delay time between different classes and investigate whether the service contract of each class is violated. We first give a brief review of definition and characteristic of self-similarity as follows.

2.1 Self-similarity

We assume that, $X=(X_t, t=0,1,2\cdots)$ is a stationary stochastic process. If we compute the average of the series X over non-overlapping blocks of size m, we obtain a m-aggregated stationary time series $X^{(m)}=(X_k^{(m)}, k=0,1,2,\cdots)$ as follows:

$$X_k^{(m)}=\frac{\sum_{l=0}^{m-1}X_{km+l}}{m},\ m\in N \tag{1}$$

When the variances and the autocorrelations of $X^{(m)}$ and X satisfy the following relation,

$$Var(X^{(m)})=\frac{Var(X)}{m^{2(1-H)}},\ 0.5<H<1, \tag{2}$$

$$r_{X^{(m)}}(i)=r_X(i),\ i\geq 0, \tag{3}$$

where H denotes the Hurst parameter. It is said that X is exactly self-similar. In addition, X is asymptotically self-similar if the following relation is satisfied,

$$Var(X^{(m)})\sim\frac{Var(X)}{m^{2(1-H)}},\ 0.5<H<1, \tag{4}$$

$$r_{X^{(m)}}(i)\to r_X(i),\ m\to\infty. \tag{5}$$

The autocorrelations given in Eqs. (3) and (5) tell that the degree of variability or burstiness is identical at different time scales for self-similar stochastic process, and the autocorrelation does not drop to zero as $m\to\infty$. This is in contrast to the characteristic of the stochastic processes used in typical data models:

$$r_{X^{(m)}}(i)\to 0,\ m\to\infty. \tag{6}$$

As for the variances given in Eqs. (2) and (4), they decrease more slowly than $1/m$ when $m\to\infty$.

As the study in (Arlitt & Jin, 2000; Arlitt & Williamson, 1997) showed that the self-similar traffic pattern generated by Web browsers fits very well to a Pareto-type distribution, our simulation model will thus assume the packet interracial times for

each priority task group to be independent and identically distributed according to the infinite-variance Pareto distribution with shape parameter α and cut-off parameter k

$$\begin{cases} f(t)=\dfrac{\alpha}{k}\left(\dfrac{k}{t}\right)^{\alpha+1} \\ F(t)=P(T\le t)=1-\left(\dfrac{k}{t}\right)^{\alpha}, \end{cases} \quad \alpha \ge 0,\ t>k, \tag{7}$$

$$f(t)=F(t)=0,\ t\le k \tag{8}$$

2.2 Proportional Delay Differentiated Service

The basic principle of proportional delay differentiated service (PDDS) proposed by Dovrolis et al. (2002) is that the higher-class requests will receive better performance compared with the lower class requests. Specifically, we assume there are $N > 1$ service classes, and the priority of each class is set in a decreasing order, then average delay time is in proportion to the priority inversely for each class:

$$\frac{D_i}{D_j}=\frac{P_j}{P_i},\ 1\le i,j\le N, \tag{9}$$

where D_i and P_i represent the average delay time and priority for class i, respectively. In other words, the average delay time will be shorter for the priority task groups that pay higher usage cost.

Our admission control scheme will predict the average delay time of each class for next measurement period, and Eq. (9) is used to select the class client with the largest gap in the ratio of the average delay time to receive next service from the server. Notably, a new client requesting for service will be place in the class queue that the client belongs to if the average delay time for that class does not exceed some predefined threshold. Each class possesses its own average delay time threshold value, and it is adjustable during operation based on the number of the clients who leave the class queue without service due to long waiting time.

2.3 Parameters Required for Admission Control Mechanism

The client is requested to supply some essential information before admitted into the service. There are two options for each client—the class that the client belongs to and the maximum average delay time that the client can endure, respectively. The specification for the class to which the client belongs is simple and consistent with the approach taken in the differentiated service enabled at the network layer, whereas the provision of the maximum average delay time directly reflects the customer's requirement. Based on the two kinds of parameter specifications, the corresponding admission control mechanisms are developed as follows.

2.3.1 Using client class as the parameter. As shown in Figure 1, each class client waiting for the service is placed in the corresponding class queue, and each queue is managed by the FCFS service model. According to conservation law (Bolch et al. 1998), if the average arrival rate is λ_i for the class i client during next measurement period, then the average delay time for class i, D_i, should be,

$$\sum_{j=1}^{N} \lambda_j \cdot D_j = \sum_{j=1}^{N} \lambda_j \cdot \overline{D}\,, \tag{10}$$

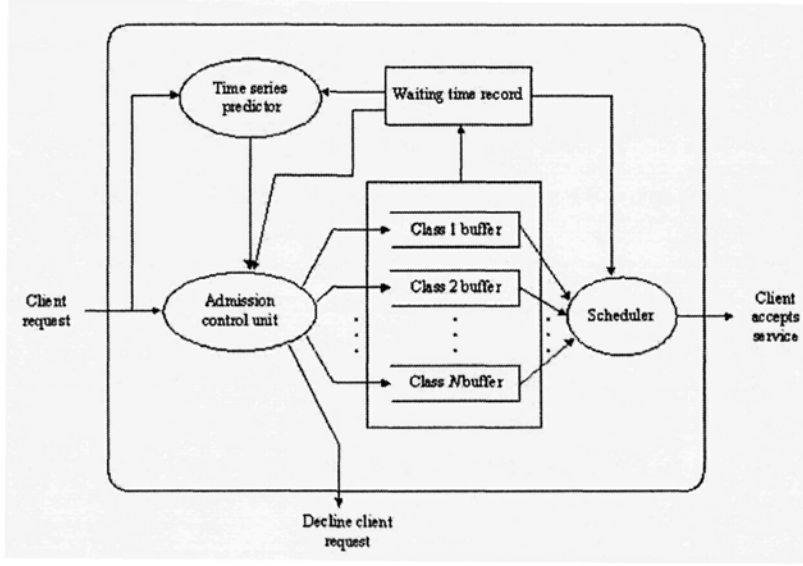

Fig. 1: Admission control scheme with a parameter of each client's class specification.

where $\overline{D}$ denotes the average delay time for the aggregate traffic serviced by a work-conserving FCFS server.

Based on Eq. (9), Eq. (10) can be rewritten as,

$$\sum_{j=1}^{N} \lambda_j \cdot D_i \cdot \frac{P_i}{P_j} = \sum_{j=1}^{N} \lambda_j \cdot \overline{D} . \tag{11}$$

Then the average delay time for class i traffic during next measurement period can be derived as follows:

$$D_i = \frac{\sum_{j=1}^{N} \lambda_j \cdot \overline{D}}{P_i \cdot \sum_{j=1}^{N} \frac{\lambda_j}{P_j}}, \; 1 \le i \le N . \tag{12}$$

As it is difficult to obtain the average arrival rate for each class during next measurement period in Eq. (12), we incorporate a time series predictor into our admission control scheme to foresee the average arrival rate and thus resolve the issue of the unreasonable request for the arrival rate specified by each client as presented in (Lee et al. 2002).

We assume that class 1 clients possess the lowest priority, and the maximum delay time for each class 1 client is D_1^{MAX}. Based on Eq. (9), the maximum delay time that class i client can tolerate is $\frac{D_1^{MAX} \cdot P_1}{P_i}$. Accordingly, the incoming class i client is allowed to use the server if the following relationship is satisfied,

$$\frac{\sum_{j=1}^{N} \hat{\lambda}_j \cdot \overline{D}}{P_i \cdot \sum_{j=1}^{N} \frac{\hat{\lambda}_j}{P_j}} \le \frac{D_1^{MAX} \cdot P_1}{P_i}, \tag{13}$$

where $\hat{\lambda}_j$ denotes the average arrival rate for class i aggregate traffic foreseen by the time series predictor.

When the server is ready to service next client, the scheduler as shown on the right of Figure 1 will use the following equation to determine which class client should be selected for service:

$$k = arg \max_{1 \le i \le N} \frac{W_i \cdot P_i}{W_1 \cdot P_1}, \tag{14}$$

where W_i denotes the waiting time for the client at the front of the class i queue, and P_i represents the priority of class i.

As the clients of some classes can tolerate longer waiting time, such as best-effort traffic, our scheme can pop up an interactive dialogue to ask the clients if they are willing to wait longer when the server is overloaded. The usage cost and the priority for the clients will be lower down if they are willing to wait longer. This approach can reduce the number of the customers in the higher class queues under a bursty workload, and maintain the stringent QoS requirement for higher class clients.

Now we summarize the algorithm for the admission control scheme with the class to which each client belongs as the parameter as follows.

1. When a class i client arrives, use the time series predictor to forecast the average arrival rate of class i aggregate traffic during next measurement period.
2. Use Eq. (12) to compute average delay time of class i during next measurement window.
3. Use Eq. (13) to determine if the incoming client is admitted to use the server.
4. If admitted, the client is placed at the end of the class i queue.
   If Eq. (13) is not satisfied, but the client is willing to wait longer; search for the first lower class that satisfy the requirement of Eq. (13).
   If found, place the client into the corresponding class queue.

2.3.2. Using maximum delay time as the parameter. In our admission control scheme, we also allow the client to specify the maximum delay time as shown in Figure 2. Notably, a classifier is needed in the scheme to compute the class that the incoming client belongs to as illustrated in Figure 2.

Let the maximum delay time requested by the incoming client is ω, and D_1^{MAX} denotes the maximum delay time for class 1 clients that posses the lowest priority. According to Eq. (9), we know that the longest delay time that class$i$ clients can bear is $\frac{D_1^{MAX} \cdot P_1}{P_i}$; the classifier then can use the following equation to derive the class that the incoming client belongs to:

$$l = \arg \min_{1 \le i \le N} \max \left(\frac{\omega \cdot P_i}{D_i^{MAX} \cdot P_i} - 1,0 \right). \quad (15)$$

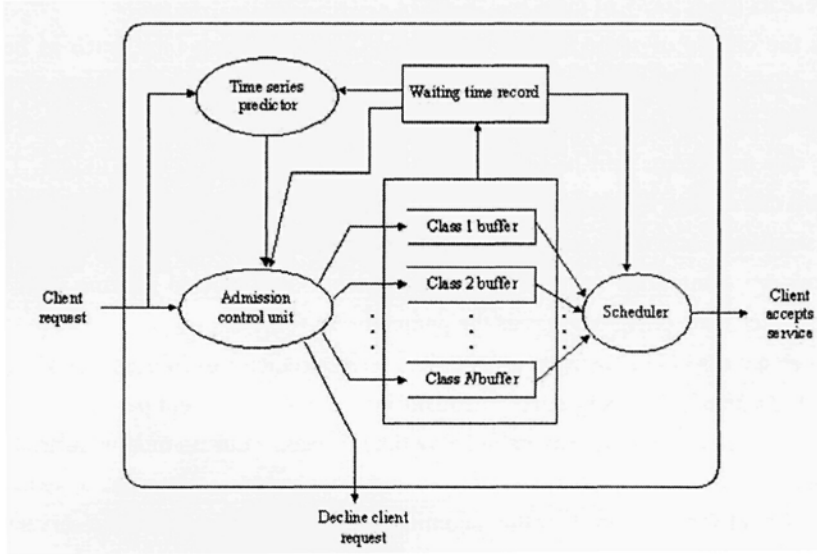

Fig. 2: Admission control scheme with the parameter of the maximum delay time.

Note that Eq. 15 is used to locate the highest priority task group whose requirement of the maximum delay time is longer than the client can stand.

The algorithm for the admission control scheme as shown in Figure 2 can be summarized as follows:

1. Use the classifier to categorize the incoming client based on Eq. (15).
2. Use the time series predictor to forecast the average arrival rate of the class, *i*, that the incoming client belongs to during next measurement period.
3. Use Eq. (12) to compute average delay time of class *i* during next measurement window.
4. Use Eq. (13) to determine if the incoming client is admitted to use the server.
5. If admitted, the client is placed at the end of the class *i* queue.
   If Eq. (13) is not satisfied, but the client is willing to wait longer; search for the first lower class that satisfy the requirement of Eq. (13).
   If found, place the client into the corresponding class queue.

3.0 TIME SERIES PREDICTOR

Time series is a sequence of numerical values indexed by increasing time units. Well-known time series prediction techniques in the literature include the 'average' method, which computes the mean of the past M measurement periods:

$$\hat{\lambda}(t+1) = \overline{\lambda} = \frac{\sum_{j=0}^{M-1} \lambda(t-j)}{M}, \tag{16}$$

and the weighted moving average, which increases the weight of the last measurement period,

$$\hat{\lambda}(t+1) = (1-\rho)\cdot\overline{\lambda} + \rho\cdot\lambda(t), \tag{17}$$

where ρ=1, and $\overline{\lambda}$ is the average computed by Eq. (16).

ARIMA (Box & Jenkins, 1970; Zhang, 2003) is a tool that is often adopted to model a given time-series data set. The flaw of ARIMA is its inability to identify the complex properties of the real world. Machine learning techniques, such as grey prediction theory (Chi et al. 1999; Sun, 2004) and neural networks (Sun, 2004), have also recently been applied in the application of time-series prediction. Neural networks models have been found to outperform grey system in predicting long-term time-series data (Sun, 2004).

Fuzzy logic has recently been employed to solve multi-connection admission control in Asynchronous Transfer Mode (ATM) and wireless networks and time-series prediction problems efficiently (Liang, 2002.; Ren & Ramamurthy, 2000; Bensaou et al. 1997; Ye et al. 2003; Liang & Mendel, 2000; Park & Kwang, 2001). The SVM system is also popular for time-series forecasting, such as forecasting financial markets (Suykens et al. 1999), forecasting electricity prices (Sansom et al. 2002), the estimation of power consumption (Chen et al. 2001), and the reconstruction of chaotic systems (Matterra & Haykin, 1999; Ding et al. 2002). The SVM model has been reported to give a better performance than the neural-network prediction model on time series prediction (Yang et al. 2002).

This work employs the SVM and fuzzy logic systems as the time-series predictor in the admission control scheme based on their high successful prediction rates in previous literature. Simulations were also undertaken for the two approaches

presented in Eqs. (16) and (17) to demonstrate the superiority of the proposed prediction techniques.

3.1 Fuzzy Logic Predictor

In this subsection, we first apply fuzzy logic controller concept to predict the maximum arrival rate and maximum delay time as shown in the scheme presented in the previous section.

As in Ren & Ramamurthy (2000), Bensaou et al. (1997), and Ye et al. (2003), we use the average arrival rate for each class during the current and the last four measurement periods $\lambda(t-3), \lambda(t-2), \lambda(t-1)$, and $\lambda(t)$ to predict the average arrival rate during next measurement period $\hat{\lambda}(t+1)$. Figure 3 shows the corresponding fuzzy logic time series predictor. The basic functions of the components employed in the predictor are described as follows.

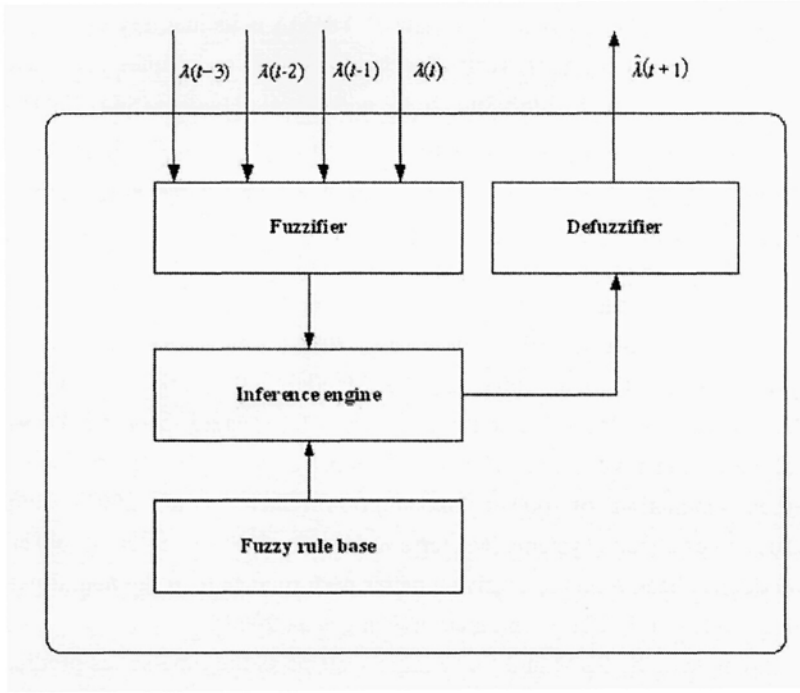

Fig. 3: The fuzzy logic based time series predictor

- **Fuzzifier**: The fuzzifier performs the fuzzification function that converts crisp input data into suitable linguistic values that are needed in the inference engine.
- **Fuzzy rule base**: The fuzzy rule base is composed of a set of linguistic control rules and the attendant control goals.
- **Inference Engine**: The inference engine simulates human decision-making based on the fuzzy control rules and the related input linguistic parameters. The max-min inference method is used to associate the outputs of the inferential rules (Buckley & Eslami, 2002), as described later in this subsection.
- **Defuzzifier**: The defuzzifier acquires the aggregated linguistic values from the inferred fuzzy control action and generates a non-fuzzy control output, the foreseen average arrival rate of each class during next measurement period. The Mamdani defuzzification method is employed in this paper to compute the centroid of membership function for the aggregated output, where the area under the graph of membership function for the aggregated output is divided into two equal subareas (Buckley & Eslami, 2002).

Figure 4 shows the mapping of four inputs of the fuzzifier and the output parameter of the inference engine into some appropriate linguistic or membership values, which are expressed by the values within the range of 0 and 1. Three membership functions for each of four inputs and the output are given in Figure 4, where the linguistic variables "low", "medium", and "high" give the measure of the average arrival rate for each class. Note that the following Gaussian membership function is chosen for the antecedents and the consequent,

$$\mu_i(\lambda(t-i)) = exp\left(-\frac{1}{2}\left(\frac{\lambda(t-i)-m_i}{\sigma_i}\right)^2\right),\ i=0,1,2,3, \tag{18}$$

where m_i denotes the mean, σ_i represents the variance.

The input and output fuzzy sets are correlated to establish the inferential rules of the fuzzy logic time series predictor. Note that three fuzzy sets are used for each antecedent, so the number of fuzzy rules is 3^4=81. By way of illustration, each fuzzy rule can be interpreted as:

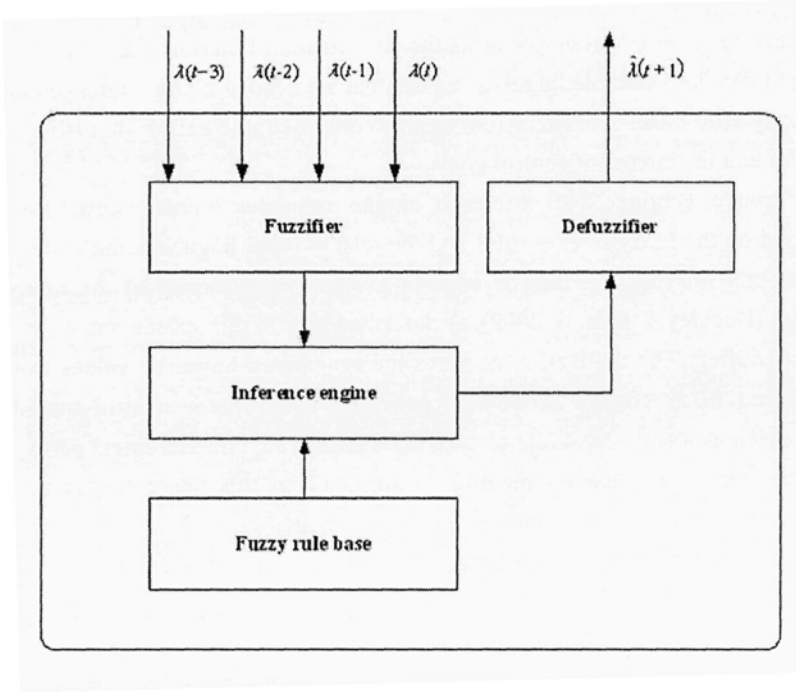

Fig. 4: Membership function for the antecedents and the consequent.

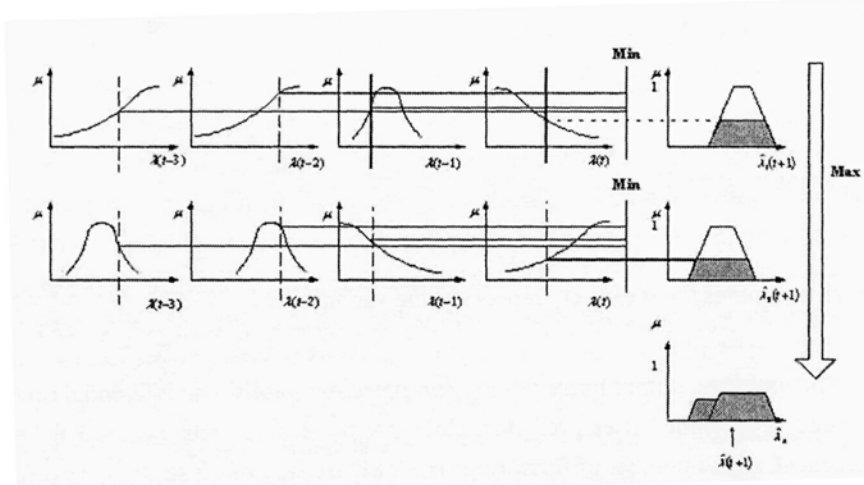

Fig. 5: The reasoning procedure for Mamdani defuzzification method.

Fuzzy rule R_j: IF $\lambda(t-3)$ is A_j and $\lambda(t-2)$ is B_j and $\lambda(t-1)$ is C_j and $\lambda(t)$ is D_j, **THEN** $\hat{\lambda}_j(t+1)$ is E_j. (19)

The inference engine then jumps to the following conclusion for fuzzy rule R_j:

$$E_j' = \left(A_j' \times B_j' \times C_j' \times D_j'\right) \circ \left(A_j \times B_j \times C_j \times D_j \rightarrow E_j\right), \tag{20}$$

where A_j', B_j', C_j' and D_j' stand for the membership grades of four inputs obtained from fuzzy rule R_j, respectively, and the expression inside the second parenthesis denote the simplified representation for Eq. (19).

Figure 5 illustrates the reasoning procedure for a two-rule Mamdani fuzzy inference system. Note that the composition of minimum and maximum operations, which corresponds the ∘ operator in Eq. (20), is employed in the evaluation of the fuzzy rules. The non-fuzzy output of the defuzzifier can then be expressed by the following algebraic expression:

$$\hat{\lambda}(t+1) = \frac{\int\left(\mu_A\left(\hat{\lambda}_A\right) \cdot \hat{\lambda}_A\right) d\hat{\lambda}_A}{\int \mu_A\left(\hat{\lambda}_A\right) d\hat{\lambda}_A}, \tag{21}$$

where $\mu_A\left(\hat{\lambda}_A\right)$ denotes the membership function of the aggregated output $\hat{\lambda}_A$.

3.2 Support Vector Machines Approach

Support vector machines (SVM) have recently been gaining popularity due to their numerous attractive features and eminent empirical performance (Vapnik, 1995; Burges, 1998; Vapnik, 1998; Suykens et al. 1999). The main difference between the SVM and conventional regression techniques is that it adopts the structural risk minimization (SRM) approach, as opposed to the empirical risk minimization (ERM) approach commonly used in statistical learning. The SRM tries to minimize an upper bound on the generalization rather than minimize the training error and is expected to perform better than the traditional ERM approach. Moreover, the SVM is a convex optimization, which ensures that the local minimization is the unique minimization.

To solve a nonlinear regression or functional approximation problem, the SVM nonlinearly map the input space into a high-dimensional feature space via a suitable kernel representation, such as polynomials and radial basis functions with Gaussian

kernels. This approach is expected to construct a linear regression hyperplane in the feature space, which is nonlinear in the original input space. Then the parameters can be found by solving a quadratic programming problem with linear equality and inequality constraints (Burges, 1998).

We assume that a training data set $D=\{(\mathrm{x}_i, y_i)\in \Re^n \times \Re, i=1,...,l\}$, which consists of l pair training data $(\mathrm{x}_i, y_i), i=1,...l$, is given. The inputs $\mathbf{x}_i$'s are n-dimensional vectors, and the system responses y_i's are continuous values. Based on the knowledge of data set D, the SVM attempts to approximate the following function:

$$f(\mathrm{x},\mathrm{w})=\sum_{i=1}^{N} w_i \cdot \varphi_i(\mathrm{x})+b, \tag{22}$$

where b is the bias term, and w_i's are the subjects of learning. Moreover, a mapping $\mathrm{z}=\Phi(\mathrm{x})$ is chosen in advance to map input vectors $\mathbf{x}$ into a higher-dimensional feature space F, which is spanned by a set of fixed functions $\varphi_i(\mathrm{x})$'s.

By defining a linear loss function with the following ε-insensitivity zone as shown in Figure 6,

$$|y_i - f(\mathrm{x}_i,\mathrm{w})|_\varepsilon = \begin{cases} 0 & if\ |y_i - f(\mathrm{x}_i,\mathrm{w})| \le \varepsilon \\ |y_i - f(\mathrm{x}_i,\mathrm{w})| - \varepsilon & otherwise \end{cases}, \tag{23}$$

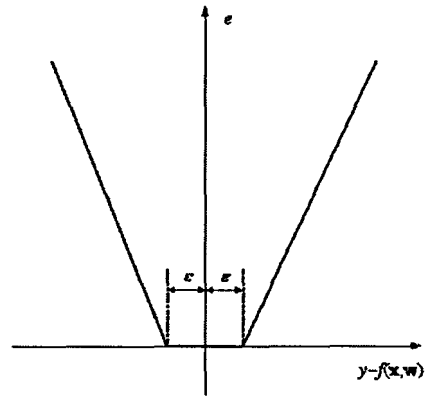

Fig. 6: ε-insensitivity loss function.

w_i's in Eq. (22) can be estimated by minimizing the risk:

$$R=\frac{1}{2}\|\mathrm{w}\|^2+C\left(\sum_{i=1}^{l}|y_i-f(\mathrm{x}_i,\mathrm{w})|_\varepsilon\right), \tag{24}$$

where C is a user-chosen penalty parameter that determines the trade-off between the training error and VC dimension of the SVM model. Note that the VC dimension is a scalar value that measures the capacity of a set of functions(Burges, 1998).

Eq. (24) can be further derived into the following constrained optimization problem:

$$R(\mathrm{w},\xi,\varsigma)=\frac{1}{2}\|\mathrm{w}\|^2+C\left(\sum_{i=1}^{l}\xi+\sum_{i=1}^{l}\xi^*\right), \tag{25}$$

subject to constraints

$$\begin{cases} y_i-\mathrm{w}^T\mathrm{x}_i-b\le\varepsilon+\xi \\ \mathrm{w}^T\mathrm{x}_i+b-y_i\le\varepsilon+\xi^* \\ \xi,\xi^*\ge 0 \end{cases}, \tag{26}$$

where ξ and ξ^* represent the measurements above and below the zone with the radius ε in Vapnik's loss function as given in Eq. (23), respectively.

It can be shown (Burges, 1998) that the above constrained optimization problem is solved by applying the Karush-Kuhn-Tucker (KKT) conditions (Taha, 1997) for regression, and maximizing the following Lagrangian:

$$L(\alpha)=-0.5\alpha^T\mathrm{H}\alpha+\mathrm{f}^T\alpha, \tag{27}$$

under constraints

$$\begin{cases} \sum_{i=1}^{l}\alpha_i^*=\sum_{i=1}^{l}\alpha_i \\ 0\le\alpha_i\le C,\ i=1,\ldots,l \\ 0\le\alpha_i^*\le C,\ i=1,\ldots,l \end{cases}, \tag{28}$$

where $\mathrm{f}=[\varepsilon-y_1\ \ \varepsilon-y_2\ \ \ldots\ \ \varepsilon-y_N\ \ \varepsilon+y_1\ \ \varepsilon+y_2\ \ \ldots\ \ \varepsilon+y_N]$, (α_i,α_i^*) denotes one of l Lagrange multiplier pairs, and the Hessian matrix **H** is given as

$$H=\begin{bmatrix} G & -G \\ -G & G \end{bmatrix}. \tag{29}$$

G denotes the corresponding kernel matrix.

The best nonlinear regression hyperfunction is then given by

$$f(x,w)=Gw_o+b_o, \tag{30}$$

where $\mathbf{w}_o$ and b_o denote the optimal desired weights vector and the optimal bias, respectively.

$\mathbf{w}_o$ and b_o can be derived by

$$w_o=\alpha^*-\alpha, \tag{31}$$

$$b_o=\frac{1}{l}\sum_{i=1}^{l}(y_i-g_i), \tag{32}$$

where $\mathbf{g}=\mathbf{G}\,\mathbf{w}_o$.

The training of $\mathbf{w}_o$ and b_o will be reinitiated whenever the cumulative error measure as given in Eq. (23) for l successive incoming input/output pairs is larger than a predefined threshold. This can assist the SVM in keeping up with the abrupt change in the average arrival rate of the aggregate traffic.

4.0 PERFORMANCE EVALUATI ON

This work conducted a series of simulations to evaluate the performance and the behavioral specifics of the admission control algorithms. The performance metrics that particularly interest us include the achieved delay time ratio for different classes of requests and the percentage of infringement of QoS requirement for admitted clients of each priority task group.

4.1 Simulation Scenario

This work developed an event-driven simulator for the experimental study of the proposed admission control schemes. Although previous works reported that the WWW traffic pattern has self-similar characteristic (Arlitt & Jin, 2000; Arlitt &

Williamson, 1997; Crovella & Bestavros, 1997; Box & Jenkins, 1970), the precise generation of representative self-similar WWW traffic for performance evaluation is still an open problem (Barford & Crovella, 1998; Paxson & Floyd, 1997). The present work thus simulated various offered loads to the Web server, keeping a fixed targeted waiting time ratios by using the real trace based on the logs from the Web server at the National Hualien University of Education. The trace files contain hundred thousands of requests within two-week period. The access logs provide the request timestamp, client ID, object URL, service status, and reply size of each request. The simulation parameters are listed in Table 1.

TABLE 1

Simulation parameters

Parameter	Value
Priority Level	2
Measurement period	1 sec.
Disk seeking overhead	1 ms.
Disk bandwidth	10 Mbps
Maximum server process number	30
Maximum queue length	1000
Delay time differentiations	1.4
Maximum delay time for basic class	11 sec.
Average system service time	18 ms

As the primary concern of the simulations is to examine effectiveness of our admission control schemes, only two priority levels are considered—namely premium clients and basic clients to simplify the study during the experiments. The delay time differentiation of the two class clients is 1.4. The maximum delay times of all clients are drawn uniformly between 2 and 11 seconds for the scheme using maximum delay time as the parameter. The service times of all requests are exponentially distributed with mean equal to 18 ms. The priority of each incoming request is assigned randomly, and the number of high priority tasks is identical to that of low priority tasks. Identical to the parameter settings adopted in (Chen & Mohapatra, 2002), the disk I/O throughput of the server is set to 10 MB per second, and disk seeking overhead of each disk request is set to 1 ms per second.

4.2 Simulation Results

We ran a bunch of simulations for our admission control schemes embedded with the four types of time series predictor, i.e. support vector machines (SVMAC), fuzzy logic system (FLAC), average method (AAC), and the weighted moving average (WMAAC), respectively, and contrasted their results to the first-come-first-serve service model (FCFS). The admission control scheme that utilizes the maximum delay time as the parameter is denoted as scheme I, and the admission control scheme using each client's class specification as the parameter is denoted as scheme II in the following illustrations.

To see the prediction performance of the four time series predictors employed in this work, this work used a time-series data set consisting of the number of sunspots recorded from 1749 to 1983 (Genet & Petrowski, 2003) to compare the prediction capability of these four predictors. Figures 7-10 depicts the prediction outcomes of these four techniques. Obviously, the SVM apparently outperforms other three techniques as expected.

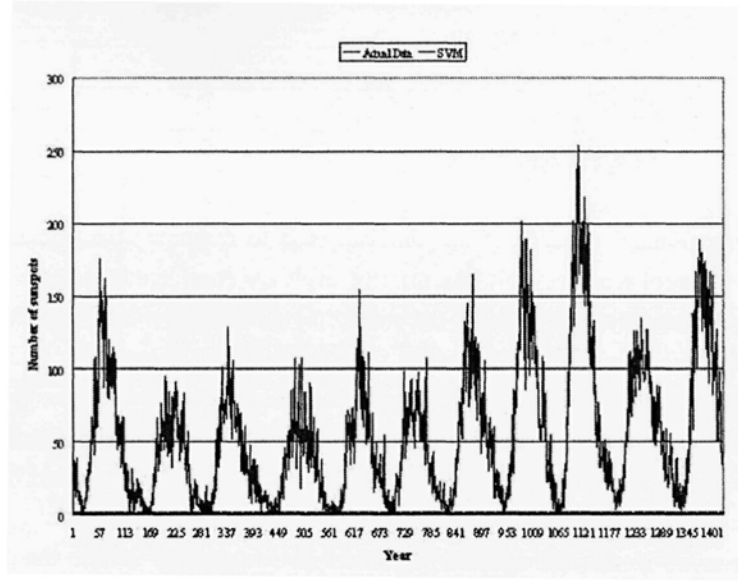

Fig. 7: Time series prediction outcomes of SVM.

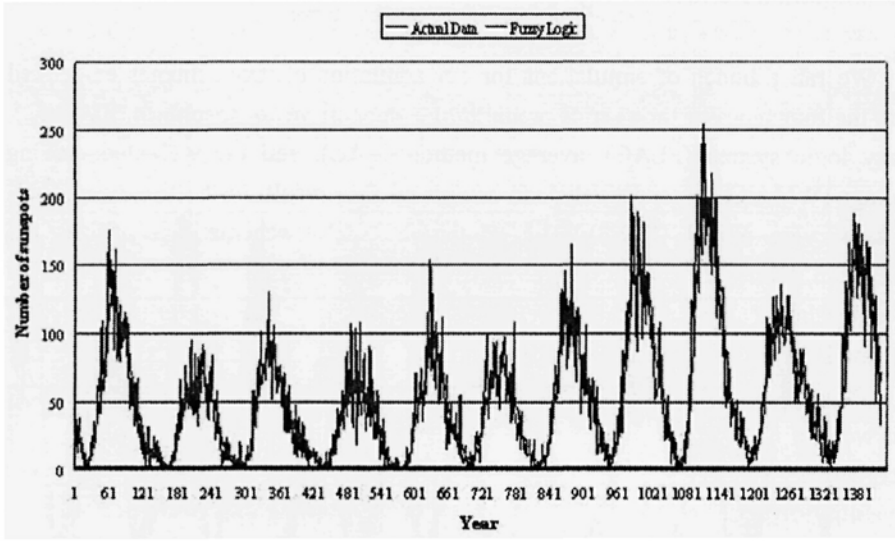

Fig. 8: Time series prediction outcomes of fuzzy logic system.

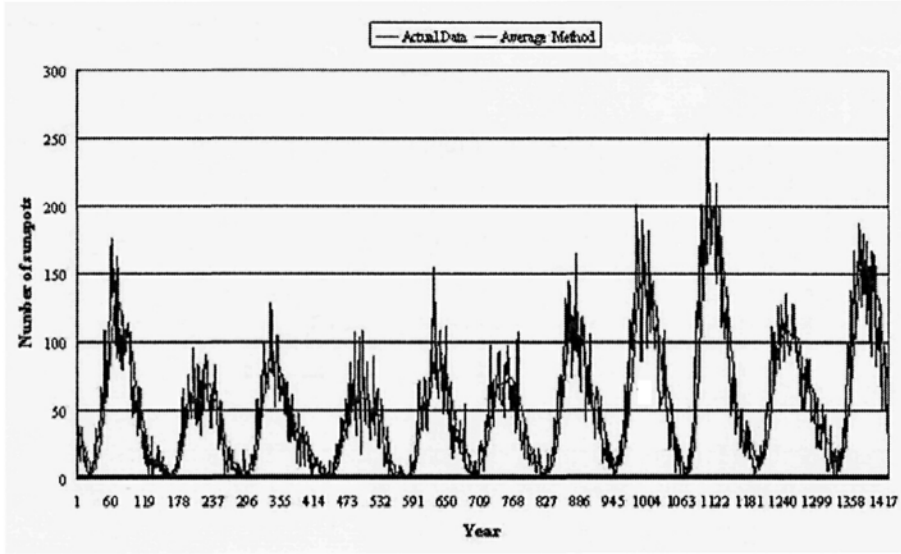

Fig. 9: Time series prediction outcomes of average method

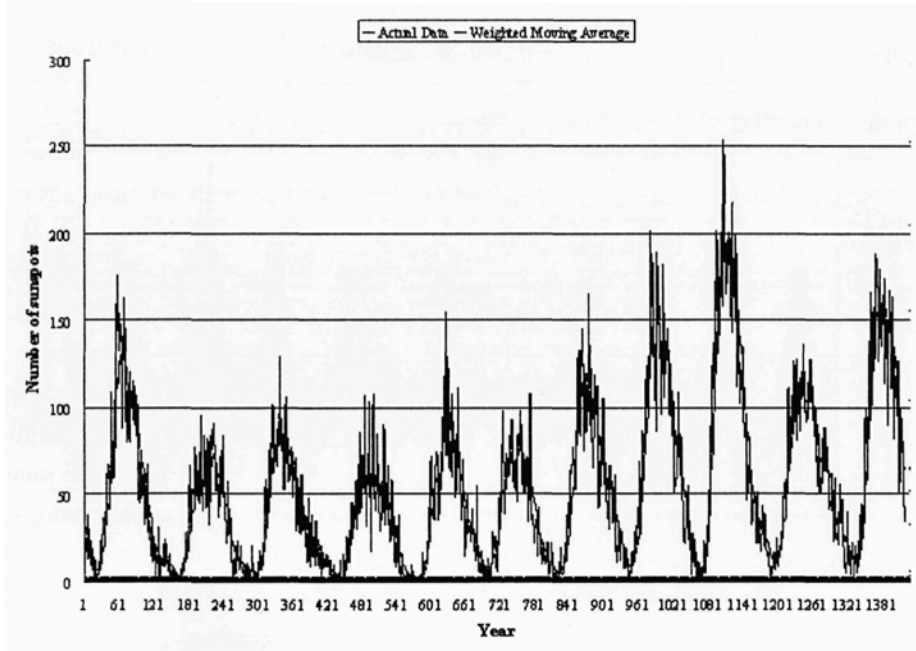

Fig. 10: Time series prediction outcomes of weighted moving average method

TABLE 2

NMSE comparison of four time series predictors

Method	SVR	Fuzzy Logic	Average	Weight Average
NMSE	19.542	28.8	24.37	24.66

Table 2 lists the normalized mean square error (NMSE) of time series prediction outcomes of the four prediction models. Notably, the NMSE is defined as:

$$NMSE = \frac{1}{n\sigma^2}\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2, \tag{33}$$

where $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, and x_i and $\hat{x}_i$ represent the actual and the predicted number of sunspots, respectively.

Table 2 shows that the SVM indeed has the lowest normalized mean square error among the four prediction models.

The comparison of the delay time ratio for the five methods under different offered loads to the Web server is given in Figures 11 and 12. The former shows the comparison for admission control scheme I, and the latter the admission control scheme II. As seen in Figures 11 and 12, the ratio of the SVMAC approach is closer to the expected value, 1.4, as specified in Table 1, than to the other four models for higher traffic loads, despite four time series predictors achieving about the same performance under low workloads. As for the FCFS service model, it does not give any differentiations between the two priority task groups as expected.

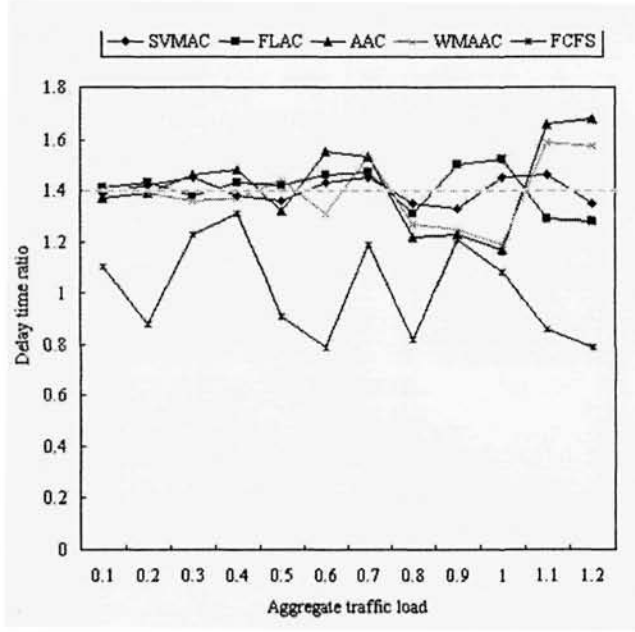

Fig. 11: Delay time ratio for two priority task groups under different workloads for admission control scheme I.

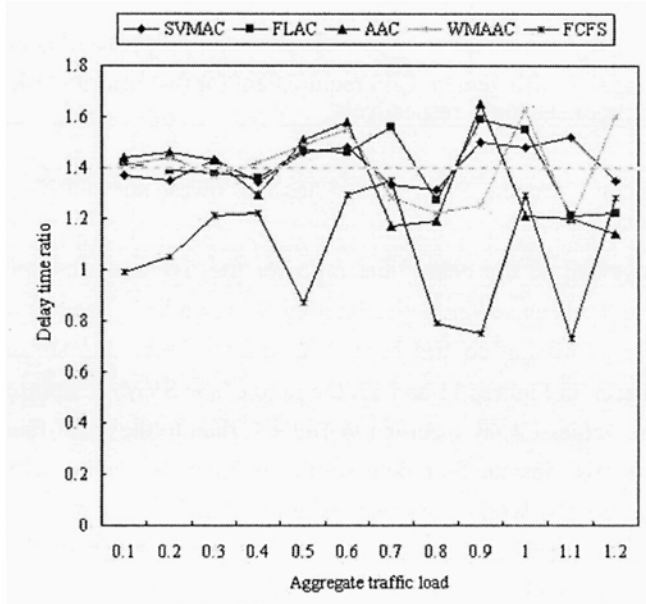

Fig. 12: Delay time ratio for two priority task groups under different workloads for admission control scheme I

Table 3 lists the comparison of the average delay time ratio achieved for two admission control schemes. The ratio for the SVMAC approach is almost identical to the expected value, 1.4, for both schemes, and the FLAC approach achieves the second best. The evidence shown in Figures 11 and 12 and in Table 3 demonstrate the effectiveness of the machine learning techniques, such as support vector machines and fuzzy logic system, used in the prediction of selfsimilar time series.

TABLE 3

Comparison of the average delay time ratio for two priority task groups

	SVMAC	FLAC	AAC	WMAAC	FCFS
Admission control scheme I	1.402	1.423	1.446	1.382	1.015
Admission control scheme II	1.41	1.421	1.363	1.424	1.068

TABLE 4

Percentage of infringement QoS requirement for two priority task groups

Admission control scheme	Client's priority (class)	SVMAC (%)	FLAC (%)	AAC (%)	WMAAC (%)	FCFS (%)
Scheme I	Premium	0.012	0.063	0.218	0.108	19.515
	Basic	0.033	0.029	0.135	0.079	13.289
Scheme II	Premium	0.019	0.074	0.256	0.131	21.642
	Basic	0.031	0.035	0.113	0.094	12.443

Table 4 shows the percentage of the clients whose QoS requirement (maximum delay time) is infringed. We can see that the SVMAC model gives the most satisfactory service for the admitted clients of each class. Even though the other four algorithms might accept more clients for service than the SVMAC scheme, the higher percentage of the violation of QoS requirements for the admitted clients, as shown in Table 4, is highly undesirable for the realization of proportional **differentiated services enabled** Internet servers.

CONCLUSION

In this paper, two adaptive admission control schemes are proposed to provide proportional delay differentiated services from an Internet server. Four different time series predictors are embedded in the admission control schemes to estimate the traffic load of the client in the next measurement period. The prediction is required to determine whether the client can be accepted in the admission control and the forecast is promising since a self-similar time series is predictable. Support vector machines, fuzzy logic, average method, and weighted moving average algorithms are used to implement the time-series prediction module for the admission control schemes in turn. The simulation results show that the implementation of the time-series prediction module with support vector machines is better then the other three methods when the performance metric of proportional delay differentiated service consistency is compared. Although the admission control scheme that utilizes the maximum delay time as the parameter slightly achieves better performance than that of the

other scheme that employs each client's class specification as the parameter, the difference of performance for the two admission control schemes is still insignificant. Subsequent research will incorporate other intelligent tools such as neuro-fuzzy and genetic algorithms into our scheme to improve the accuracy of prediction for the arrival rate of the aggregate traffic.

ACKNOWLEDGMENT

This research was partially supported by National Science Council under grant NSC 92-2213-E-026-001

REFERENCES

Adas, A. 1997. Traffic models in broadband networks, *IEEE Communication Magazine*, 82-9.

Anguita, D. and Boni, A. 2001. Towards analog and digital hardware for support vector machines, *2001 International Joint Conference on Neural Networks*, **4**, 2422-6.

Anguita, D., Boni, A. and Ridella, S. 1999a A VLSI friendly algorithm for support vector machines, *1999 International Joint Conference on Neural Networks*, **2**, 939-42.

Anguita, D., Boni, A. and Ridella, S. 1999b Learning algorithm for nonlinear support vector machines suited for digital VLSI, *Electronics Letters*, **35**, 1349-50.

Arlitt, M.F. and Jin, T. 2000. A workload characterization study of the 1998 world cup web site, *IEEE network*, 30-7.

Arlitt, M.F. and Williamson, C.L. 1997. Internet web servers: workload characterization and performance implications, *IEEE/ACM Trans. Networking*, **5**, 631-45.

Bensaou, B., Lam, S.T.C., Chu, H.-W. and Tsang, D.H.K. 1997. Estimation of the cell loss ratio in ATM networks with a fuzzy system and application to measurement-based call admission control, *IEEE/ACM Trans. Networking*, **5**, 572-84.

Bhatti, N. and Friedrich, R. 1999. Web server support for tiered services, *IEEE Network*, **13**, 64-71.

Bolch, G., Greiner, S., de Meer, H. and Trivedi, K.S. 1998. *Queuing networks and Markov chains: modeling and performance evaluation with computer science*

applications. New York, John Wiley & Sons, Inc.

Box G. and Jenkins, G. 1970. *Time series Analysis: Forecasting and Control*, Holden-Day.

Buckley, J. and Eslami, 2002. An introduction to fuzzy logic and fuzzy sets *Advances In Soft Computing*, Physica Verlag.

Burges, C.J.C. 1998. A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, **2**, 121-67.

Cao, L.J. and Tay, F.E.H. 2003. Support vector machine with adaptive parameters in financial time series forecasting, *IEEE Transactions on Neural Networks*, **14**, 1506-18.

Chen, S.; Samingan, A.K.; Hanzo, L. 2001. Support vector machine multiuser receiver for DS-CDMA signals in multipath channels, *IEEE Transactions on Neural Networks*, **12**, 604-11.

Chen, X. and Mohapatra, P. 1999. Providing differentiated service form an Internet server, *IEEE Inter. Conf. Computer Communications and Networks*, 545-54.

Chen, X. and Mohapatra, P. 2002. Performance evaluation of service differentiating Internet servers, *IEEE Trans. Computers*, **51**, 1368-75.

Chen, X., Mohapatra, P. and Chen, H. 2001. An admission control scheme for predictable server response time for web access, *10 h Inter. World Wide Web Conf.*, 545-54.

Chi, S.C., Chen H.P. and Cheng, C.H. 1999. A forecasting approach for stock index future using grey theory and neural networks, *International Joint Conference on Neural Networks*, **6**, 3850-5,

Crovella, M.E. and Bestavros, A. 1997. Self-similarity in World Wide Web traffic: evidence and possible cause, *IEEE/ACM Trans. Networking*, **5**, 835-46.

Deng, S. 1996. Empirical model of WWW document arrivals at access link, *IEEE Inter. Communications Conference*,1797-802.

Dovrolis, C., Stiliadis, D. and Ramanathan, P. 2002. Proportional differentiated services: delay differentiation and packet scheduling, *IEEE/ACM Trans. Networking*, **10**, 12-26.

Eggert, L. and Heidemann, J. 1999. Application-level differentiated services for web servers, *World Wide Web Journal*, **3**, 133-42.

Gong; X. and Kuh, A. 1999. Support vector machine for multiuser detection in CDMA communications, *The Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, **1**, 680-4.

Haffner, P., Tur, G. and Wright, J.H. 2003. Optimizing SVMs for complex call

classification, *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, **1**, I-632-5.

Hasegawa, M., Wu, G. and Mizuno, M. 2001. Applications of nonlinear prediction methods to the Internet traffic, *The 2001 IEEE International Symposium on Circuits and Systems*, **2**, III-169-72.

Huebner, F., Liu, D. and Fernandez, J.M. 1998. Queuing performance comparison of traffic models for Internet traffic, *IEEE Global Telecommunications Conf.*, 471-6.

Jin, X. and Li, J. 2001. Improve WFQ to implement PDS and its performance analysis, *IEEE Inter. Conf. Info-tech and Info-net*, **2**, 736-40.

Kanodia, V. and Knightly, E.W. 2003. Ensuring latency targets in multiclass Web servers, *IEEE Trans. Parallel and Distributed Systems*, **14**, 84-93.

Kant, K. and Mohapatra, P. 2001. Current research trends in Internet servers, *Workshop on Performance and Architecture of Web Servers*, 5-7.

Kuh, A. 2001. Adaptive kernel methods for CDMA systems, *2001 IEEE International Joint Conference on Neural Networks*, **4**, 2404-9.

Lee, S.C., Lui, J.C. and Yau, D.K. 2002. Admission control and dynamic adaptation for a proportional-delay DiffServ-enabled web server, *ACM SIGMETRICS Inter. Conf. Measurement and Modeling of Computer Systems*, **30**, 172-82.

Leung, M.K., Lui, J.C. and Yau, D.K. 2001. Adaptive proportional delay differentiated services: characterization and performance evaluation, *IEEE/ACM Trans. Networking*, **9**, 801-817.

Li, C.-C., Tsao, S.-L., Chen, M.-C., Sun, Y. and Huang, Y.M. 2000. Proportional delay differentiation services based on weighted fair queuing, *IEEE Inter. Conf. Computer Communications and Networks*, 418-23.

Li, J.-S. and Lai, H.-C. 2001. Providing proportional differentiated services using PLQ, *IEEE Global Telecommunications Conf.*, **4**, 2280-4.

Liang, Q. 2002. "Ad hoc wireless network traffic-self-similarity and forecasting," *IEEE Communication Letters*, **6**, 297-299.

Liang, Q. and Mendel, J.M. 2000. Interval type-2 fuzzy logic systems: theory and design, *IEEE Trans. Fuzzy Systems*, **8**, 535-50.

Park, S. and Lee-Kwang, H. 2001. A designing method for type-2 fuzzy logic systems using genetic algorithms, *IEEE Joint 9th IFSA World Congress and 20th NAFIPS Inter. Conference*, 2567-72.

Paxson, V. and Floyd, S. 1995. Wide area traffic: the failure of Poisson modeling, *IEEE/ACM Trans. Networking*, **3**, 226-44.

Peter Zhang, G. 2003. Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*, **50**, 159-75.

Raicharoen, T., Lursinsap, C. and Sanguanbhokai, P. 2003. Application of critical support vector machine to time series prediction, *2003 International Symposium on Circuits and Systems*, **5**, V-741-4.

Ren, Q. and Ramamurthy, G. 2000. A real-time dynamic connection admission controller based on traffic modeling, measurement, and fuzzy logic control, *IEEE J. Selected Areas in Comm.*, **18**, 184-96.

Ritter, H., Pastoors, T. and Wehrle, K. 2000. DiffServ in the web: different approaches for enabling better services in the World Wide Web, *Joint Conference of Broadband Communications, High Performance Networking and Performance of Communication Networks*, 555-66.

Sun, Chia-Hung 2004. *The application of grey prediction and genetic artificial neural network for taiwan index option*, Master Thesis, Chaoyang University of Technology.

Van Gestel, T., Suykens, J.A.K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B. et al. 2001. Financial time series prediction using least squares support vector machines within the evidence framework, *IEEE Transactions on Neural Networks*, **12**, 809-21.

Vapnik, V. 1995. *The nature of statistical learning theory*, New York, Springer-Verlag.

Vasiliou, N. and Lutfiyya, H. 2001. Managing a differentiated quality of service in a World Wide Web server, *IEEE Inter. Symp. Integrated Network Management*, **VII**, 309-12.

Vuong, S. and Shi, X. 2000. A proportional differentiation service model for the future Internet differentiated services, *IEEE Inter. Conf. Communication Technology*, **1**, 416-23.

Ye, J., Shen, X. and Mark, J.W. 2003. Call admission control in wideband CDMA cellular networks by using fuzzy logic, *2003 IEEE Wireless Communications and Networking*, **3**, 1538-43.

Zhu, J.-Y., Ren, B., Zhang, H.-X. and Deng, Z.-T. 2002. Time series prediction via new support vector machines, *2002 International Conference on Machine Learning and Cybernetics*, 1, 364-6.