Modeling Web Session for Detecting Pseudo HTTP Traffic

Y. Xie*, S. Tang†, X. Huang ‡ and C. Tang§
* School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, P.R. China.
† Department of Engineering Technology, Missouri Western State University St. Joseph, MO 64507, USA
‡ Network and Information Technology Center, Sun Yat-Sen University, Guangzhou 510275, P.R. China.
§ School of Computer Science and Eng., Guilin University of Electronic Tech.,Guilin 541004, China
Email: *xieyi5@mail.sysu.edu.cn,

***Abstract*—More and more Internet services and applications are transferred by the HTTP protocol due to its openness. This brings new challenges to the security management of network boundary. In this paper, a new approach is proposed to detect the pseudo Web behavior which abuses the general HTTP protocol to pass through the network boundary. A new parameter is defined to extract the features of Web-session based on the inter-arrival time of HTTP requests. A nonlinear mapping function is introduced to protect the weak signals from the interference of the infrequent large values. An hidden Markov model with state duration is applied to describe the normal access behavior of Web sessions. The proposed model is dynamic, and does not rely on presupposed threshold and client- or server-side data which are widely used in traditional session detection approaches. An objective function is derived for predicting the near future behavior of a user's Web-session. The deviation between the prediction result and the real observation is used for detecting the pseudo Web behavior. Experiments based on real HTTP traces from large-scale Web proxies are implemented to valid the proposal.**

***Index Terms*—Web session, modeling, detection**

I. INTRODUCTION

Web-based attacks and abuses are a more serious issue now than they have ever been, due to the rapid expansion of the HTTP-based World Wide Web. For example, the well known SQL injection, Cross Site Scripting (XSS), HTTP-based worm, Trojan and botnet. In recent years, more and more illegal applications have been discovered to utilize the openness of HTTP to pass through border firewall by mimicking general HTTP traffic, e.g., HTTP tunnel [1] [2], HTTP-based botnet [3], and application-layer distribution denial of service [4]. These new illegal technologies may seriously affect the information security strategy of inside network and may spread the malicious traffic from internal network to outside network, even to the backbone of the Internet. Thus, during the past decade, many studies were done on Internet traffic measure and identification. These research can be roughly divided into two categories: network layer-based approaches [5] and application layer-based approaches [6] [1] [2].

The former usually uses IP header, TCP connection and machine learning algorithms to profile the traffic characteristics and to detect the abnormality. The latter is generally based on modeling the traffic characteristics of the application-layer from the perspective of a higher level or extracting the extended client behavior information from the observed application-layer traffic. Some of the more complex schemes use semantic analysis or payload pattern recognition.

Although the network layer-approaches are easy for implementation because of using the non-discriminatory IP traffic or TCP connection, their main disadvantage is that these methods are unable to collect enough offensive signals for detecting the Web-based attacks and abuses because they belong to different layers respectively. Compared with the network layer-approaches, the application layer-based approaches are more closer to the natural behavior of a human who is the actual source of launching all Web transactions and HTTP traffic. Therefore, in recent years the research gradually shifts to the application layer and client behavior. Literatures collected in this paper do indicate that the detection results of application layer-based approaches are better than the network layer-based.

For those application layer approaches which are based on the Web behavior of clients, the key issue is how to accurately describe the Web-session process of a user via the general observed HTTP traffic with high real time capability and low computational complexity. However, existing schemes' performance on modeling client behavior is not really good. The reasons mainly lie in the following aspects: (i) the concept of "HTTP traffic" is not in accord with a human's natural Web access process which has a remarkable characteristic of multi-stage and at least includes three sub-processes, i.e., "interaction","browsing" and "off-lining" [7]; (ii) since the real Web access process includes multiple stages, the exiting schemes using a single model to describe the long-term behavior of clients is unsuitable, because they may mask (or smooth) a lot of details of different stages; (iii) normal HTTP traffic is very easy to be forged by a general HTTP generator [8] with universal open statistical parameters for concealing the malicious behavior.

Due to these reasons, Web session-based approaches become a new direction to detect the anomaly HTTP traffic, e.g., [9] [10] [11]. However, it is impossible for

doi:10.4304/jcp.8.2.341-348

them to distinguish the Web access processes of users accurately, because detection systems are usually unable to obtain the client-side data. Hence, different methods are designed to estimate the Web-session processes, e.g., threshold for detecting the session boundary [12] [13], clustering algorithms for session patterns identification [14] [15]. Unfortunately, these schemes cannot realize their goals as expected.

The main problem of identifying Web access processes is due to the difficulties in the definition of the Web access process (or said "session") itself, which depends on the application used: applications such as *Telnet* or *SSH* tend to generate a single TCP connection per user session, whereas application layer protocols such as HTTP, IMAP/SMTP and X11 usually generate multiple connections per user session. Also, the generally accepted conjecture that such sessions would follow a Poisson arrival process might have reduced the interest in the session process analysis.

Here, we try to go a step further in this area by a new method which is designed to identify Web-sessions and to detect the sham Web behavior for the boundary network nodes (e.g., boundary proxy of an autonomous system). The main contributions in this work are: (i) a new model is proposed to automatically describe and scout the Web-session processes of users; and (ii) a probability function is introduced to predict and detect the abnormality of Web-session processes.

The rest of the paper is organized as follows. The next section covers the major related work in the Web-session identification. Section III gives an overview of a physical model for Web-session processes. Section IV describes the modeling methodology. Section V presents evaluation results, and Section VI outlines the assumptions and limitations in formulating the model. Finally, Section VII concludes and suggests future research directions.

II. Related work

The definition of a Web-session is not straightforward. Traditionally, a Web-session is defined as a period of time when a sequence of requests from the same user are generated during a single visit to a special Web site, and then followed by a "long" inactivity period. For example, placing an order through the e-commerce site involves requests related to selecting a product (activity period), thinking time used for comparing items or making a decision (inactivity period), providing shipping information, arranging payment, and receiving confirmation (activity period).

Identifying Web-session plays an important role both in Internet traffic characterization and in the proper dimensioning of network resources. Unfortunately, the identification of active and inactivity periods is not trivial.

Traditional approaches (e.g., [13] [12]) rely on the adoption of a threshold η : HTTP requests (or IP packets/TCP connections) are aggregated in the same session if the inter-arrival time between two consecutive HTTP requests (or IP packets/TCP connections) is smaller than

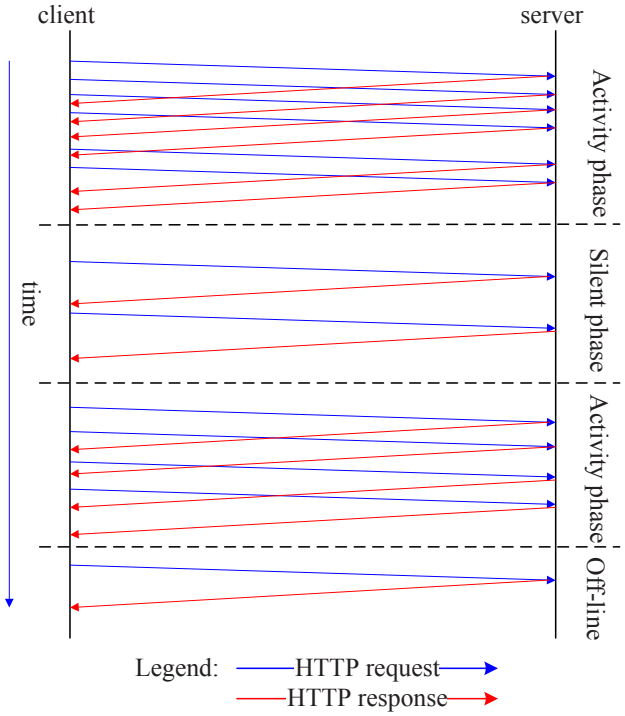


Fig. 1. Definition of Web-session

the given threshold value η; otherwise, a new session is identified. This approach works well if the threshold value is correctly matched to the average value of connection and session inter-arrival time. However, to know these values in advance is unrealistic in practice. If the threshold value is not correctly matched to the user session statistical behavior, the threshold based mechanisms are highly error prone in the session identification. Furthermore, this type of approach lacks universality because the threshold values are quite different when this method is applied to different experiment data. For example, in [13], threshold is set to be 100s, while in [12], the value is 1s. Recently, in [16] FARIMA is used to model the session level arrivals. However, they did not provide an automatic method for identifying the session boundary. The session data used for model training were preprocessed based on the predefined threshold.

Clustering is another type mainstream technique which is applied to characterizing Web-sessions, e.g., [14] [15] [17]. The main advantage of using this approach is that there is no need to predefine any threshold value. However, this method is very easy to be affected by the noise requests launched by some routines, e.g., on-line update programs, browser plug-ins running automatically, auto-refresh function of Web pages. Moreover, clustering is only a static method, it cannot describe the dynamic characteristics of a series of consecutive sessions. Hence, it is merely applied to static analysis of the observed session data instead of modeling and predicting the Web-sessions.

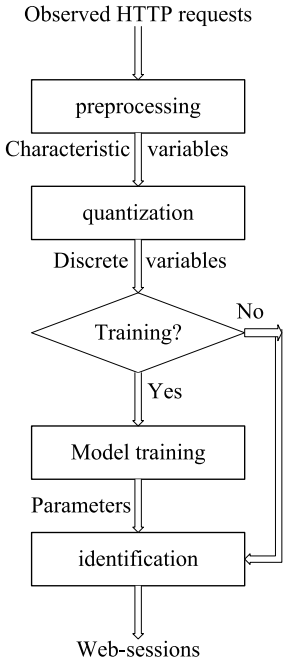


Fig. 2. Flow chart of Web-session identification

III. WEB-SESSION MODEL

Different from traditional Web-session models, in this paper a Web-session of a special user is divided into three phases (as shown in Fig.1): activity phase, silent phase and off-lining phase. Activity phase means that the user is surfing the Internet, which causes the frequent interactions between the user and different remote servers. Silent phase indicates that network connection is enable, but the user does nothing. During this period, the HTTP requests are mainly launched by those programs which reside in the memory of a client's computer, instead of the user's own actions. Thus, the number of requests in this period is far less than that of the activity phase. The last phase means that the network connection is unworkable or the user has left.

IV. MODELING METHODOLOGY

As shown in Fig.2, the proposed scheme includes four modules: *Preprocessing*, *Quantization*, *Model training* and *Identification*. *Preprocessing* is used to extract variables from the observed data. *Model training* is used to construct the non-parametric state model for the identification of Web-session processes after *Quantization* is applied to discretize the input data. When the model's parameters are obtained, the *Identification* module can be used to recognize the Web-session processes from the continuous HTTP flows.

A. Logarithm-based quantizer

In this paper a discrete state stochastic model is proposed to describe the Web-session process. Thus, the first step is quantization. In order to protect the information embedded in the small observed data which occur relatively frequently, a logarithm-based nonuniform quantizer

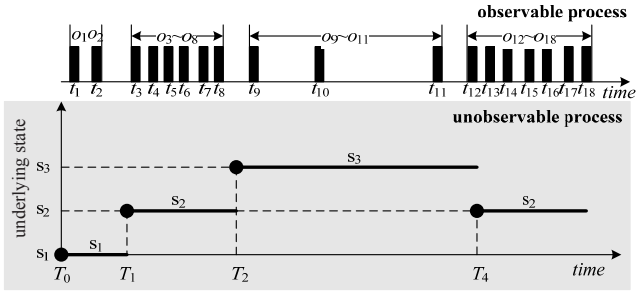


Fig. 3. A HTTP sequence controlled by a hidden semi-Markov process

is applied to our proposal. The form of this quantizer is defined in the following which has been widely used in communication systems:

$$y = \begin{cases} \dfrac{Ax}{1+\log A+\log x_{\max}}, & 0 \le x \le \dfrac{1}{A} \\ \dfrac{1+\log A+\log x}{1+\log A+\log x_{\max}}, & \dfrac{1}{A} \le x \le x_{\max} \end{cases} \quad (1)$$

where x and y are the input and output positive values, and A is a positive constant. $x_{\max}$ is the maximal value of input data. This quantizer is equivalent to passing the observed signal x through a logarithmic *compressor* and then applying the compressed signal y to a uniform quantizer.

B. Hidden Markov model with state duration

Since introduced by Baum and Petrie [18], the hidden Markov models (HMMs) have become very popular in a wide range of applications like biology, speech recognition, image processing and text recognition. A serious disadvantage of HMMs is the fact that the sojourn times of the hidden process are geometrically or exponentially distributed. Such a model was found to be inappropriate for some applications. To solve this problem, Ferguson [19] proposed a model that allows arbitrary sojourn time distributions for the hidden process which is known as an explicit-duration HMM or variable-duration HMM. As the hidden process becomes semi-Markovian, this model is also called hidden semi-Markov model (HsMM).

An HsMM consists of a pair of discrete-time stochastic processes S_t and O_t, $t \in \{0, ..., \tau - 1\}$. The observed process O_t is linked to the hidden, i.e., unobserved state process S_t by the conditional distribution depending on the state process. The state process of an HsMM is a finite-state semi-Markov chain. Inheriting the property that the conditional distributions usually overlaps, a specific observation can arise from more than one state. Thus the state process S_t is not directly observable through the observation process O_t, but can be estimated. Figure 3 visualizes the concept of the HsMM. As shown in the figure, there are two layers of the hidden semi-Markov process: the top level is an observable process O_t and the low one is an unobservable process S_t.

In the discrete case the output process O_t is related to the semi-Markov chain S_t by the observation probabili-

ties:

$$b_i(o_t) = P(O_t = o_t|S_t = i), \tag{2}$$

where $\sum_{o_t} b_i(o_t) = 1$. The observation process is characterized by the conditional independence property,

$$\begin{aligned} &P(O_t = o_t|O_0^{\tau-1} = o_0^{\tau-1}, S_0^{\tau-1} = s_0^{\tau-1}) \\ &= P(O_t = o_t|S_t = s_t) \end{aligned} \tag{3}$$

where $O_0^{\tau-1} = (O_0, ..., O_{\tau-1})$ and $\{O_0^{\tau-1} = o_0^{\tau-1}\} = \{O_0 = o_0, ..., O_{\tau-1} = o_{\tau-1}\}$. This process implies the fact that the output at time t depends only on the state of the underlying semi-Markov chain at time t.

The transition probabilities of the underlying semi-Markov chain for the state i is

$$a_{ij} = P(S_{t+1} = j|S_{t+1} \neq i, S_t = i), j \neq i \tag{4}$$

where $\sum_{j\neq i} a_{ij} = 1$, $a_{ii} = 0$. The sojourn time distributions $p_i(u)$ of the unobserved semi-Markov chain of length u from $t+1$ until $t+u$ in the state i is denoted by

$$\begin{aligned} p_i(u) = P(&S_{t+u+1} \neq i, S_{t+u-v} = i, \\ &v = 0, ..., u-2|S_{t+1} = i, S_t \neq i) \end{aligned} \tag{5}$$

where $u \in \{1, ..., U_i\}$ and U_i the upper bound of the time spent in state i. Before and after this sojourn in state i, the process has to be in some different state.

In this paper, it is assumed that the state occupancy distribution is concentrated on the finite set of time points $1, ..., U_i$, where U_i may also increase up to the entire length of the observed sequence. For the particular case of the last visited state i, the sojourn time function $P_i(u)$ is defined as:

$$P_i(u) = \sum\nolimits_{v\geq u} p_i(v) \tag{6}$$

The survivor function sums up the individual probability masses of all possible sojourns of length $v \geq u$. With the definition of the initial state probabilities $\pi_i = P(S_0 = i)$ where $\sum_i \pi_i = 1$, the following relation can be verified if the process starts in state i at time $t = 0$ and remains its initial state i till it transits to state j ($j \neq i$) at time $t = \tau$:

$$\begin{aligned} P(O_0^{\tau-1} = o_0^{\tau-1}, S_0^{\tau-1} = i, S_\tau = j) = \\ \pi_i p_i(\tau) \left\{ \prod\nolimits_{k=0}^{\tau-1} b_i(o_k) \right\} a_{ij} \end{aligned} \tag{7}$$

More details on HsMMs can be found in the recent work of the state-of-the-art survey [20].

C. Mapping the Web behavior to an HsMM

Different from most current work based on inter-arrival time, the proposed model for Web-session clustering and prediction is based on a new variable called average inter-arrival time which is defined by

$$\bar{x}_t = \frac{\sum_{m=1}^{N_t} \Delta_t^m}{N_t - 1} \tag{8}$$

where N_t is the HTTP request number during the t^{th} time unit, Δ_t^m the inter-arrival time between the $(m-1)^{th}$ request and the m^{th} request launched by the same client during the t^{th} time unit. Following the flow chart introduced above, $\bar{x}_t (t = 1, ...)$ are quantized by the logarithmic quantizer after they are extracted in the preprocess module. And then, the quantizer's output data are used to train the non-parametric HsMM.

Let O_t denote the stochastic variable expressing the observed value at the t^{th} time unit, e.g., the quantized average inter-arrival time ($\bar{x}_t$) in this paper, $\mathbb{O}$ be the value space of O_t, i.e., $(O_t = o_t)$ where $o_t \in \mathbb{O}$. Let the underlying semi-Markov process $\{S_t\}$ present the unique Web-session type of a given client during the t^{th} time unit, $\mathbb{S}$ be the state space of $\{S_t\}$, i.e., $(S_t = s_t)$ where $s_t \in \mathbb{S}$. In this paper, the $\mathbb{S}$ includes three states which represent three types of Web-sessions: Activity, Silence and off-lining, respectively. Let U_i denote the maximal duration time of state i (or said the i^{th} Web-session type). Thus, the state transition probability matrix presents the transfer relation between different Web-session types. The duration of a state presents the sojourn time of a Web-session when a user is surfing the Web. Since the actual type of a Web-session is usually unobservable, the state chain is hidden. Thus, the Web access behavior can be mapped to the HsMM.

In order to realize the Web-session prediction, let $\mathcal{P}_t(i, u)$ stand for the probability function, given the observed o_1^t and the model parameter set Ω, the current underlying state (Web-session type) s_i will not transit to other state within u time units:

$$\mathcal{P}_t(i, u) = P(S_t^{t+u} = s_i|O_1^t = o_1^t, \Omega) \tag{9}$$

for the current state $S_t = s_i$, it takes place either from the transition of $S_{t-1} = s_j$ for $j \neq i$ or from state s_i when it starts at the $(t-v)^{th}$ time unit for all $v \leq U_i - u$. Then, the equation (9) can be rewritten as:

$$\begin{aligned} \mathcal{P}_t(i, u) = &\sum\nolimits_{j\neq i} P(s_j \text{ ends at time } t, S_t^{t+u} = s_i \\ &|O_1^t = o_1^t, \Omega) + \sum\nolimits_v P(s_i \text{ starts at time} \\ &t - v, S_{t-v}^{t-1} = s_i, S_t^{t+u} = s_i|O_1^t = o_1^t, \Omega) \end{aligned} \tag{10}$$

In order to compute the prediction probability function $\mathcal{P}_t(i, u)$, a forward variable $\alpha_t(i)$ is defined as the probability of the observation sequence for all state sequences where state s_i ends at time t. Now, we readily obtain the following forward recursion formula:

$$\begin{aligned} &\alpha_t(i) = P[O_1^t = o_1^t, s_t = i \text{ ends at } t|\Omega] \\ &= \sum_j \sum_{u=1}^{\min(U_j,t)} \alpha_{t-u}(j) a_{ji} p_j(u) \prod_{m=0}^{u-1} b_i(o_{t-m}) \end{aligned} \tag{11}$$

Then, the observed sequence likelihood can be computed by:

$$P(O_1^t = o_1^t|\Omega) = \sum\nolimits_i \alpha_t(i) \tag{12}$$

Using the recursion formula of forward variable $\alpha_t(i)$, we obtain the formula for prediction function $\mathcal{P}_t(i,u)$:

$$\mathcal{P}_t(i,u) = \frac{1}{\sum_i \alpha_t(i)} \left(\sum_{i \neq j} \alpha_{t-1}(j) a_{ji} p_i(u) b_i(o_t) + \sum_{i \neq j} \sum_{v \leq U_i - u} \alpha_{t-v-1}(j) a_{ji} p_i(u+v) \prod_{\tau=t-v}^{t} b_i(o_\tau) \right) \quad (13)$$

The prediction probability function of $\mathcal{P}_t(i,u)$ can be used to detect the sham Web behavior. Assuming the real semi-Markov state and its duration time of time t are i and u, respectively, the abnormality of time t can be evaluated by $\mathcal{P}_t(i,u)$. If the values of $\mathcal{P}_t(i,u)$ fall into the credible area, the HTTP behavior is normal, otherwise it is abnormal and a warning may be raised.

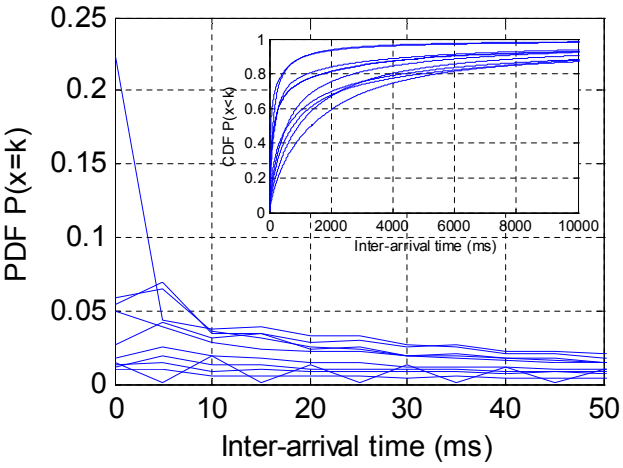
V. Experiments

In this section, the proposed approach is implemented to scout and predict the Web-sessions based on the real traces of proxies. All the data are collected during 2007 to 2009 from the logs of ten large-scale bound proxies which are located in different areas. Time resolution of these data is millisecond, which can provide more information for Web-session processes scout and prediction than general logs.

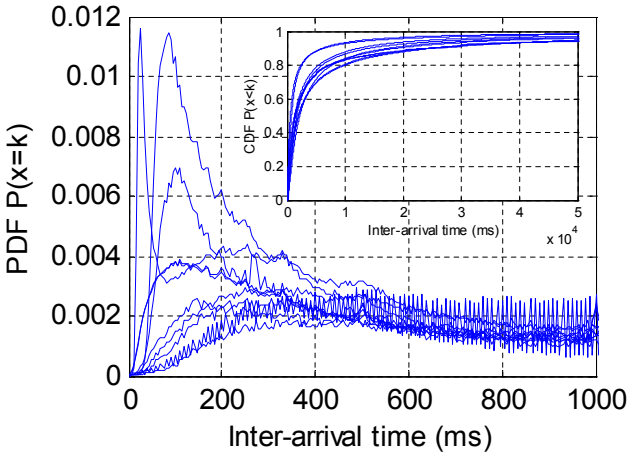
Fig.4 compares the Probability Distribution Function (PDF) and Cumulative Density Function (CDF) between the raw inter-arrival time and the average inter-arrival time per second. Although these proxies are located in different places, Fig.4(a) shows their common characteristics: more than 80% of observed inter-arrival time is less than $4000ms$. This result is quite different from most previous work. It also implies that the Web-session structure is changing with the evolution of the modern network and Website. Thus, the traditional Web-session identification methods based on inter-arrival threshold become inaccurate. Fig. 4(b) shows an interesting phenomenon: each PDF curve has a peak near the point of $100ms$. This result gives us an edification that the average inter-arrival time may be better than the raw inter-arrival time for describing the Web-session processes.

Fig.5 shows the relationship between the input and output of the quantizer. The continuous average inter-arrival time is mapped (or compressed) into a new space based on Equation (1). And then, those compressed values are quantized by uniform quantizing. As showed in the figure, over 40% of average inter-arrival time is less than $20s$, which can keep their information by our quantization algorithm.

Fig.6 shows a short sample sequence of the Web-session scout and prediction based on our model. The figure plots the quantized average inter-arrival time by the grey bar and the Web-session scout and prediction (i.e., the underlying state) by red line. This result shows that the model can automatically scout the different session types. Three underlying Markov states are defined in this



(a) inter-arrival time



(b) average inter-arrival time per second

Fig. 4. PDF (and CDF in the inset) of inter-arrival time and average

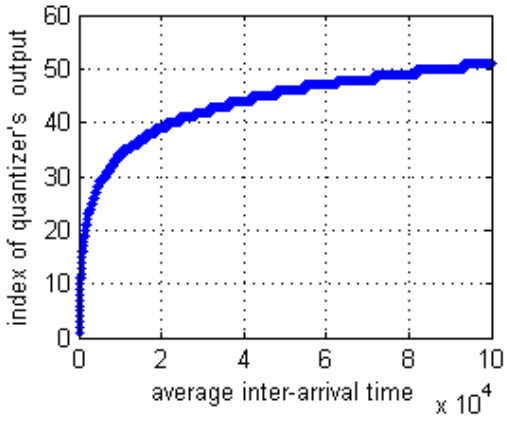


Fig. 5. average inter-arrival time vs. quantized value

experiment. As shown in the figure, state 1 corresponds to the small values of average inter-arrival time while the state 3 is for those large values. Hence, state 1 can be used to describe the interactive process when user is surfing; state 2 can be considered as the reading (or thinking) process when user is browsing the Web; state 3 can be considered that the use is not online. The reason that there

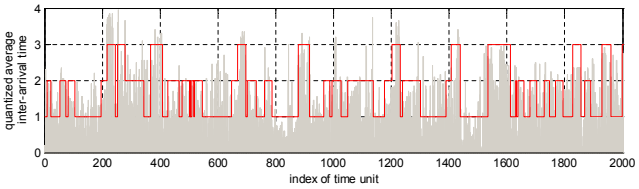


Fig. 6. scouting and predicting the Web-session processes

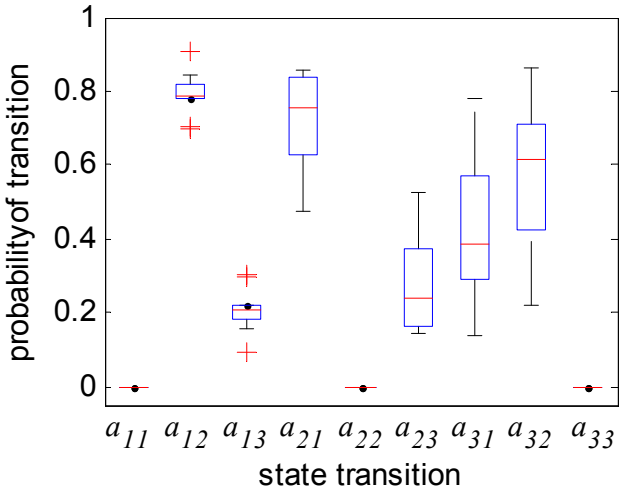


Fig. 7. probability distribution of state transition

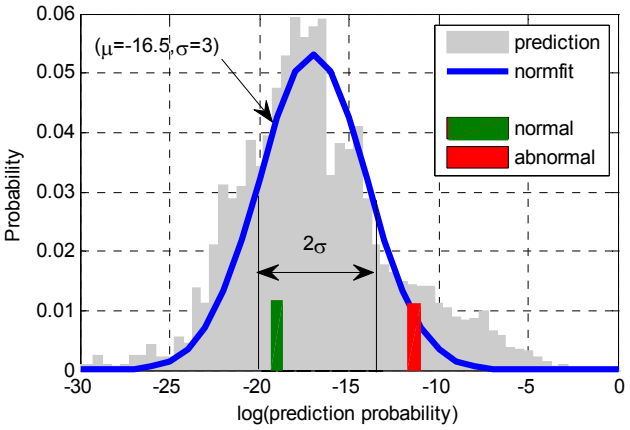


Fig. 10. probability distribution of prediction

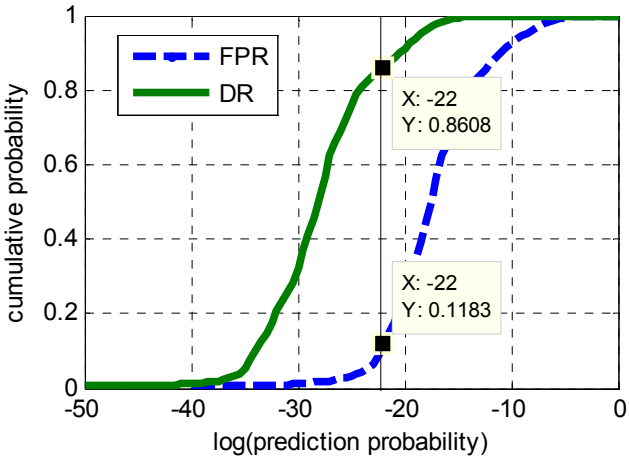


Fig.11. ROC curve for detection

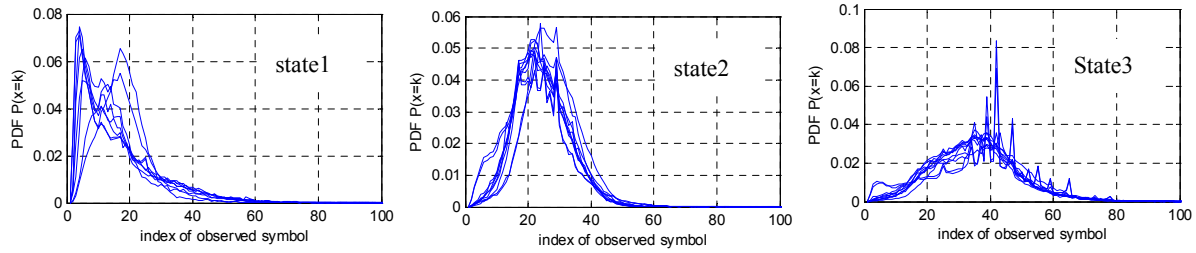
are still some requests during the duration of state 2 and 3 is that many modern programs may automatically and periodicity connect to their remote servers by HTTP form, e.g., software update and Web page refresh.

Different session models are built for ten proxies' clients, whose model characteristics are shown in Fig. 7 and Fig.8. In Fig. 7, box plot is used to present the transition relations between different states. The box has lines at the lower quartile, median, and upper quartile values. Whiskers extend from each end of the box to the adjacent values in the data. Data with values beyond the ends of the whiskers are displayed with a red "+" sign. The result shows the top two active states are state 1 and 2 which correspond to the "interactive phase" and "reading phase", respectively. Assuming the initial entry is s_3, we can draw a typical state diagram for Web-session process as $(s_3 \rightarrow s_2 \rightarrow s_1 \rightarrow \cdots \rightarrow s_2 \rightarrow s_3)$ based on Fig. 7. This state diagram matches the natural Web access behavior of human beings: after a long time of offline (s_3), a user starts his/her computer system before Web access; during the booting, some softwares may connect to their remote servers by launching a small amount of requests (s_2); when the system is ready,the user will start Web surfing, thus, s_1 and s_2 may alternately appear till the user stop the Web access actions (s_3). Fig.8(a) and Fig.8(b) respectively show the probability distributions of state vs. output and state vs. duration. In each sub-figure, the curves from the models of different proxies are centralized. Moreover, each state respectively represents a typical probability distribution of output symbols and duration time. These results indicate that the proposed nonparametric model does have the ability to automatically scout and describe the observed values. Furthermore, our algorithm is self-adaptive and convergent.

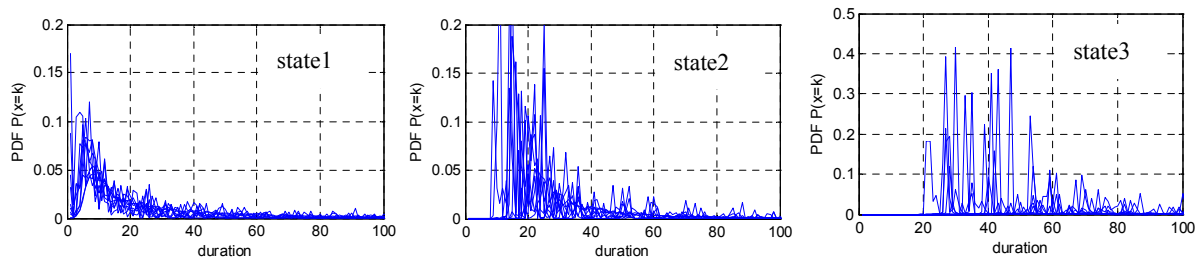
In Fig.9, the characteristics of different types of Web-session process are shown. The main sojourn time of state 1 is near $10s$ while the center duration time of the other two states is $100s$ and $800s$, respectively. Most request numbers of state 1 are less than 10 while the other two states' request numbers are near 20 and 50, respectively.

Then, the Web-session model is applied to anomaly HTTP traffic detection. The anomaly HTTP traffic used in this experiment is generated by the OICQ traffic which is encapsulated by the open HTTPtunnel tool for passing through firewall. HTTPtunnel creates a bidirectional virtual data connection tunnelled in HTTP requests. The HTTP requests can be sent via an HTTP proxy if so desired. For users behind restrictive firewalls, if the WWW access is allowed through a HTTP proxy, it's possible to use HTTP tunnel to send/receive any types of data to/from a computer outside the firewall, e.g., telnet, FTP, or Peer-to-Peer.

Fig.10 shows that the proposed Web-session model does work for the anomaly HTTP traffic detection. In Fig.10, the shadow is the prediction probability distribution for some time (said time t) based on its his history session model. The blue curve is the fitted Gauss distri-

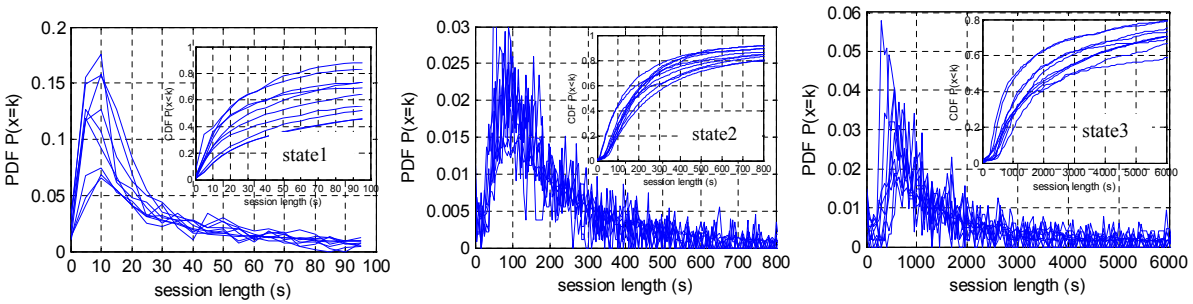


(a) output probabilities of states

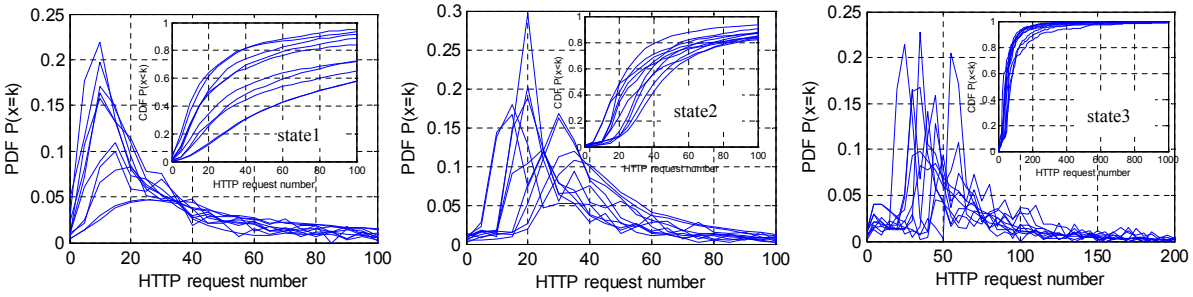


(b) duration times of states

Fig. 8. characteristics of model parameters



(a) session length (second)



(b) requests per session

Fig. 9. characteristics of Web-session

bution for prediction probability. The area $[\mu - \sigma, \mu + \sigma]$ is used as the confidence interval in this experiment, where μ and σ are the mean and variance of prediction probability distribution. Thus, all the real probabilities of time t (e.g., the green one) falling into the confidence interval are considered as normal, otherwise, they are abnormal HTTP traffic (e.g., the red one). Fig.11 shows the Receiver operating characteristic (ROC) curve of the detection. When -20 is used as the anomaly detection threshold, the detection rate (DR) is over 85% with the false positives rate (FPR) is about 11%.

Fig.12 shows that the model parameters can be obtained within 5 iterations during the training phase. This result illustrates that the proposed method is suitable for realtime applications.

VI. Discussion

A. Non-uniform quantization

Traditional, the quantization process may follow a uniform law, however, it is preferable to use a variable separation between the representation levels. For example, the range of inter-arrival of HTTP requests, from the peaks to the weak, is on the order of 10^5 to 0. By using a nonuniform quantizer with the feature that the step-size increases as the separation from the origin of the input-output amplitude characteristic is increased, the large end

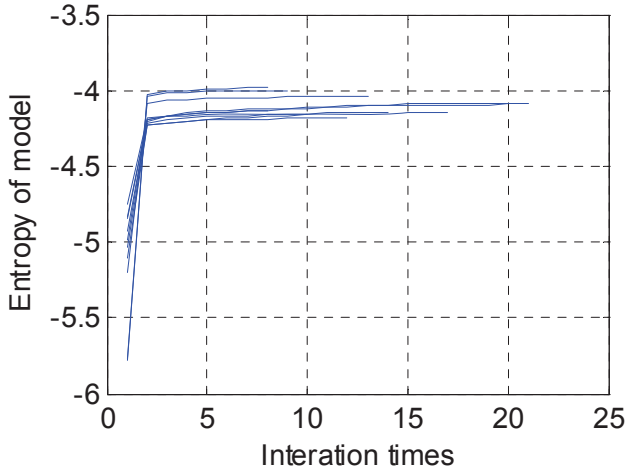


Fig. 12. Interation times of model training

steps of the quantizer can take care of possible excursions of the observed signal into the large amplitude ranges that occur relatively infrequently. In other wards, the weak values, which need more protection, are favored at the expense of the large values. In this way, a nearly uniform percentage precision is achieved throughout the greater part of the amplitude range of the input signal, with the result that fewer steps are needed than would be the case if a uniform quantizer were used.

B. Non-parametric model

Many previous work on the HsMM models the duration times by a predefined parametric family of continuous distributions, e.g, exponential family of distributions or negative binomial distribution. One disadvantage of parametric model is that an initial distribution has to be assumed. Without this priori knowledge, the model may cause serious deviation. For the non-parametric method used in this paper, although its computation is more than the parametric one, it needn't the priori assumption. Thus, this method is more suitable for dynamic describing the Web-session processes than the parametric model.

VII. Conclusion

A novel model based on hidden Markov model with explicit state duration is applied to describe and scout the Web-session processes. Nonlinear algorithm is introduced for improving the quantification precision and reducing the computational complexity. A probability function is derived to predict Web-session processes. Numerical results of experiments show that the proposed method is practical in the Web-session processes' scouting and prediction.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities (Grant No.11lgpy38), the National Natural Science Foundation of China (Grant No. 60970146); Doctoral Fund of Ministry of Education of China (Grant No.20090171120001).

References

[1] T. Jackson, "Anomaly-based HTTP covert tunnel detection using hidden Markov models," in *Masters Abstracts International*, vol. 46, no. 4, 2007.

[2] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel hunter: Detecting application-layer tunnels with statistical fingerprinting," *Computer Networks*, vol. 53, no. 1, pp. 81–97, 2009.

[3] J. Lee, H. Jeong, J. Park, M. Kim, and B. Noh, "The activity analysis of malicious http-based botnets using degree of periodic repeatability," in *Security Technology, 2008. SECTECH'08. International Conference on*. IEEE, 2008, pp. 83–86.

[4] Y. Xie and S. Yu, "Monitoring the application-layer DDoS attacks for popular websites," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 1, pp. 15–25, 2009.

[5] G. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning. Communications Surveys and Tutorials," *IEEE*, vol. 10, no. 4, pp. 56–76, 2008.

[6] Y. Song, A. Keromytis, and S. Stolfo, "Spectrogram: A mixture-of-markov-chains model for anomaly detection in web traffic," in *Proc of the 16th Annual Network and Distributed System Security Symposium (NDSS)*, 2009.

[7] Y. Xie and S. Yu, "Anomaly Detection Based on Web Users' Browsing Behaviors," *Journal of Software*, vol. 18, no. 4, pp. 967–977, 2006.

[8] J. Cao, W. Clevel, Y. Gao, K. Jeffay, F. Smith, and M. Weigle, "HTTP SOURCE TRAFFIC MODELING 1 Stochastic Models for Generating Synthetic HTTP Source Traffic," 2008.

[9] S. Cho and S. Cha, "SAD: web session anomaly detection based on parameter estimation," *Computers & Security*, vol. 23, no. 4, pp. 312–319, 2004.

[10] M. Mizutani, S. Shirahata, M. Minami, and J. Murai, "ROOK: Multi-session Based Network Security Event Detector," in *Proceedings of the 2008 International Symposium on Applications and the Internet-Volume 00*. IEEE Computer Society Washington, DC, USA, 2008, pp. 48–54.

[11] M. Meiss, J. Duncan, B. Gonçalves, J. Ramasco, and F. Menczer, "What's in a session: tracking individual behavior on the web," in *Proceedings of the 20th ACM conference on Hypertext and hypermedia*. ACM, 2009, pp. 173–182.

[12] F. Smith, F. Campos, K. Jeffay, and D. Ott, "What TCP/IP protocol headers can tell us about the web," *ACM SIGMETRICS Performance Evaluation Review*, vol. 29, no. 1, pp. 245–256, 2001.

[13] C. Nuzman, I. Saniee, W. Sweldens, and A. Weiss, "A compound model for TCP connection arrivals for LAN and WAN applications," *Computer Networks*, vol. 40, no. 3, pp. 319–337, 2002.

[14] A. Bianco, G. Mardente, M. Mellia, M. Munafo, and L. Muscariello, "Web user session characterization via clustering techniques," in *Proc. IEEE GLOBECOM 2005*. Citeseer, pp. 1102–1107.

[15] L. Chaofeng, "Research on Web Session Clustering," *JOURNAL OF SOFTWARE*, vol. 4, no. 5, p. 461, 2009.

[16] X. Wang and K. Goseva-Popstojanova, "Modeling Web Request and Session Level Arrivals," in *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications-Volume 00*. IEEE Computer Society, 2009, pp. 24–32.

[17] A. Bianco, G. Mardente, M. Mellia, M. Munafò, and L. Muscariello, "Web user-session inference by means of clustering techniques," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 2, pp. 405–416, 2009.

[18] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, pp. 1554–1563, 1966.

[19] J. Ferguson, "Variable duration models for speech," in *Proceedings of the Symposium on the Application of hidden Markov models to Text and Speech*, vol. 1, 1980, pp. 143–179.

[20] S.-Z. Yu, "Hidden semi-markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215 – 243, 2010.