國 立 清 華 大 學

碩士論文

電子商務使用者行為導向第七層內容交換器測試

E-Commerce User Behavior Oriented Layer Seven Content Switch Testing

系 所 別 ： 通訊工程研究所碩士班

學號姓名 ： 895627 李其晃 (Chi-Huang Lee)

指導教授 ： 黃能富 博士 (Prof.Nen-Fu Huang)

中華民國九十一年六月

中文摘要

隨著網路的蓬勃發展，近年來第七層內容交換器(layer seven content switches)已急需於世界各地的網站中，為了提昇網站效率這是不得不為的選擇之一。但現今無法提供一套可接受與具公信力的標準來精準測試各類的第七層內容交換器，以便讓網站架設者輕易挑選所需特性或品質較佳的交換器，更進一步地說這些測試流程中並不能提供 stateful 處理、功能性驗證與模擬真實使用者行為的 HTTP 請求(requests)。

在此，我們提出一兼具電子商務使用者行為與整合 cookie 運作機制的模型，進而滿足完整的第七層內容交換器的測試。以此模型為基礎，進一步發展電子商務使用者行為內容交換器測試系統模組(E-Commerce User Behavior Content Switch Testing System Module)來達成實際完整的測試。此模組具同質擴充性(self-scaling)與同質配裝性(self-configuring)，其分別為可調整任何 HTTP 請求的強度規模與可依所需之流量特性來設定使用者行為，主要提供依原則的功能性與正確性驗證來測試第七層內容交換器的各種特性。

最後，論文中定義四種測試第七層內容交換器的單位，包含服務準確率(Hit Rate)、連線失敗率(Connection Failure Rate)、網頁失敗率(Page Failure Rate)與交易成功率(Deal Successful Rate)，希望能成為具公信的標準。並使用電子商務使用者行為內容交換器測試系統模組來做測試工具、簡述測試環境與結果分析。此結果將證明測試系統模組具有精確性與實用性，可產生實際網際網路的特性來測試第七層內容交換器。

Abstract

In recent years, layer seven content switches (L7-CS) have become demand extremely on the Web sites around the world. Currently, it is a problem to accurately measure the performances of various layer seven content switches. Importantly, there is no widely accepted criterion for such tasks. Furthermore, the available tests don't yield satisfactory stateful transactions, functional verification and emulation of real user behavior HTTP requests.

This thesis proposes a model based on E-Commerce user behavior and incorporates cookie maintenance mechanism to address the insufficiencies in the testing of layer seven content switches. Based on this model, we further develop E-Commerce User Behavior Content Switch Testing System Module for realistic tests. This module has both self-scaling and self-configuring, meaning that it can scale the HTTP requests to any strengths and configure user behaviors to specific traffic characteristics. This module principally provides functional and accurate verification in principled ways to measure the entire characteristic of layer seven content switches.

Finally, we define four standard units in the testing of layer seven content switches. We will also present tests using E-Commerce User Behavior Content Switch Testing System Modules and examine the testing environments procedures, and the results. The results show that the module is accurate and useful to evaluate the layer seven content switches under real Internet situations.

Contents

List of Figures

Chapter One Introduction

A typical corporate Web site has become a complex grouping of specialized technology. These sites could easily include firewalls, routers, Layer 2/Layer 3 switches, load-balancing devices, cache servers and Web servers. All these devices are put together directly impacts Web site performance. But for business consideration especially E-Commerce merchants, it must have a high service rate on this Web site and better be quick: End users lose patience when Web sites take longer than more seconds to load.

Furthermore, it must have rich content (voice, data, music, image, and video) and broadband access on next generation network. Except the performance consideration, content services seamlessly provided are important issues. These content services problems imply that less page failures, short response time, commerce transactions online correctness and rich information. These problems would have been resolved in the further and become good issues of network technology.

In general Web hosting, layer 4 switching devices are a comparative choice. Layer 4 switches can help implement QoS within a network because they can distinguish between different types of applications by gleaning information beyond the network layer (layer 3), such as the TCP port ID field. These capabilities are often referred to as Layer 4 filtering and forwarding. With Layer 4 switches, network management can set application priorities, thus maintaining a high service level for selected, more critical end users and applications. However, they cannot distinguish between different types of Web applications because of the common use of TCP port 80 as the TCP port for HTTP traffic.

Another way to improve performance is to control how traffic enters your environment. Web content switches--also known as URL switches or layer seven switches--can provide you with the highest level of control over your incoming Web traffic. Web content switches look all the way into the HTTP header (URLs and cookies) to make load-balancing decisions, rather than stopping at the TCP port number. By examining the HTTP header, these switches can make decisions on how individual Web pages and images get served from your site. This level of traffic control can be helpful if your Web servers are optimized for specific functions, such as image serving, SSL (Secure Sockets Layer) sessions or cookie transactions.

Recent examinations of layer seven testing tools are used usually to measure Web server performance, so these tools have lacks of testing mechanisms for layer seven content switches. We summarize the features of testing layer seven content switches as follows:

- Persistence: This verifies accurately user's connection redirection correctness.
- Stateful operation: The testing generator can generate the particular request which relies on different responses from remote Web server.
- Analytic workload generation: That generates HTTP requests with mathematical model for different workload characteristics [23].
- Flexibility: In comparison with using prerecorded logs of past workloads, the testing methods must be emulated various workload characteristics.
- Variable demands: They have been shown to exhibit high variability, such as HTTP request interval times, types and sizes.
- Real world emulation: The layer seven content switch testing tools must emulate network real world, especially E-Commerce environment is needed.

In this thesis we will propose a model based on E-Commerce user behavior in chapter four and incorporates cookie maintenance mechanism in chapter five to address the insufficiencies in the testing of layer seven content switches. E-Commerce user behavior oriented model having Browsing-&-Searching behavior state, Shopping behavior state, Payment-Procedure behavior state represent a complete commerce procedure online. cookie maintenance mechanism is a critical technology and essentially demands at E-Commerce Web sites. Based on this model and mechanism, we further have developed E-Commerce User Behavior Content Switch Testing System Module for realistic tests. It has both self-scaling and self-configuring, meaning that it can scale the HTTP requests to any strengths and configure user behaviors to specific traffic characteristics. Those address analytic workload and flexibility of testing methods. This module principally provides functional, stateful, emulating real environment and accurate verification to measure the entire characteristic of layer seven content switches.

We further define four standard units in the testing of layer seven content switches and these units have different kinds of indicating criterion. Tests using E-Commerce User Behavior Content Switch Testing System Modules including testing environment, procedures as well as results have presented in this thesis. The results show that the module is accuracy and useful to measure the layer seven content switch under real Internet situations.

Chapter Two Layer Seven Content Switch Feature

The features of layer seven content switches include Server Load Balancing, Filtering, Application Redirection, Firewall Load Balancing, Health Checking, Global Server Load Balancing, Content Intelligent Switching, Persistence and Bandwidth Management [22]. The layer seven content switches provide good solution for E-Commerce environment because of allowing re-establishing a user's connection to a particular server. That is so called Persistence mechanism and briefly shown below.

2.1 Persistence

Why should we need the persistence in the E-Commerce Web sites? Typical Web sites are constructed as a server farm with multiple servers. If an Internet user connects to one server of this server farm, it may be desirable for the user to reconnect to that same server in the future. This is necessary when a server has data stored with the customer, yet the data is not dynamically shared with the other servers. Each particular data or content separately is segregated in different Web server or server farms, such as the relevant set of Images and shopping-basket are segregated in individual server farm. Because that constructed Web sites are more efficiency than setting all content in the all Web servers, the persistence is needed. For example, let's say a customer builds a shopping-basket of goods at a Web site, and then leaves the Web site before completing the transaction online. If the customer is returning to the Web site, the switch directs the customer's request to a different server, that server

may not know about the customer and his selected items. Under this condition, content switch must facilitate session persistence seamlessly to redirect to the previous Web server.

2.2.1 Cookie Persistence

In generally, the content switch provides three types of cookie persistence that uses an HTTP cookie stored on the customer's computer to allow the customer to reconnect the same server previously visited at a Web site: Insert mode, Rewrite mode, and Passive mode. Please referring to Alteon WebSystems, Inc. [11], Cisco Systems, Inc. [12] and F5 Networks, Inc. [13]. Below are brief descriptions of each mode.

- **Insert mode**:

In the Insert mode of operation, the content switch will create the cookie and send an HTTP "set-cookie" to the client. After completion of the TCP connection to the server, the connection is established. Since the cookie contents are derived from the real id and the cookie's expiration time, there is no need to store a sticky association. Inbound packets are scanned for the cookie, and routed to real servers accordingly. If no cookie is present in the packets, the content switch initiates a connection to the server as the client, and generates a cookie. Again, the rest of the transaction passes without the content switch scanning data.

- **Rewrite mode**:

If Rewrite mode is specified, the content switch intercepts the response with cookie which is sent from the server to the customer, then rewrites the name of the cookie specified server. When the content switch rewrites the cookie of the response, the server information and time-out values are reset. Rewrite mode requires setting up of the cookie created by the server, and requires a blank cookie coming from the Web server for the content switch to rewrite.

- **Passive mode (Learning mode)**:

In the Passive mode of operation, this feature looks for HTTP "Get"s in inbound packets. The content switch will learn the cookie that is set by the real server by its response to this request. On the first request without a cookie, the "set-cookie" response from the server creates a cookie association. In order for Passive mode to work, there needs to be a cookie response coming from the Web server with the appropriate server information in the cookie.

When the error of cookie persistence procedure occurs (that is customer's request dispatched different server by content switch), it would terminate the E-Commerce transaction between the Web merchant and the visiting customer. This condition is shown in Figure 2.1. A new customer at client computer initiates a HTTP request to the remote Web site and first, content switch establishes a TCP connection to server1 by load-balancing algorithm. Then the request is directed to server1 through the content switch seamlessly and server1 responses an HTTP data with cookie backward to client computer at the same time. If the customer is returning to the Web site with cookie, the content switch dispatches the request to wrong side server2. This condition must disorder the procedure of each server and challenge the customer's patience. We hope it wouldn't happen always in Web merchants.

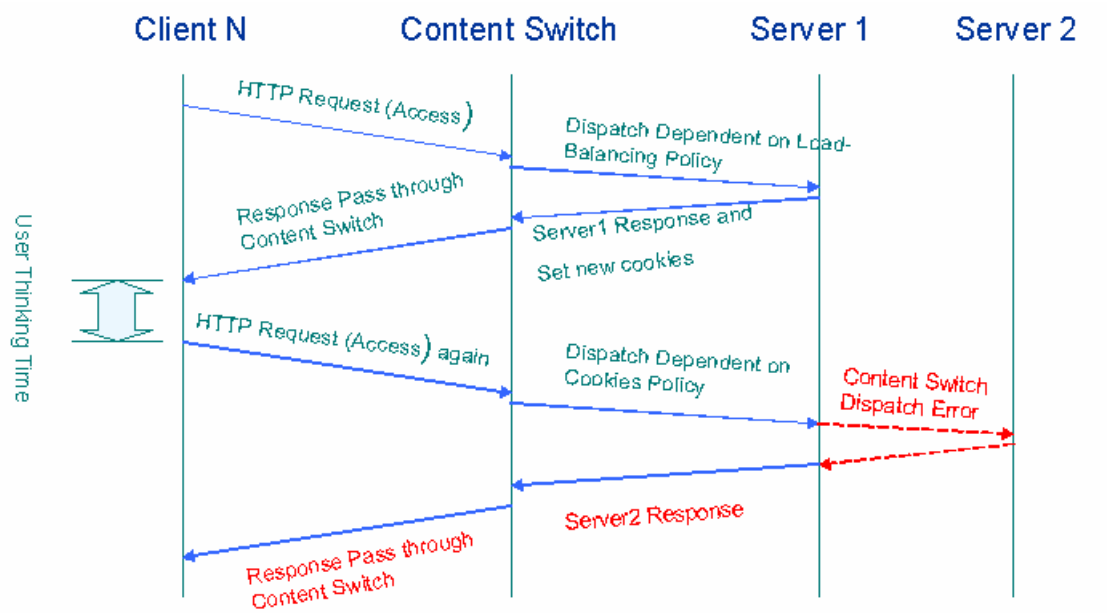


FIGURE 2.1 COOKIE PERSISTENCE ERROR OCCURRENCE

Chapter three Layer Seven Content Switch Testing

3.1 Testing Tools

In this chapter, this thesis briefly describes layer seven content switch testing tools and its disadvantages for high service level testing.

- **WebBench** [14]

WebBench measures Web server software performance by running different Web server packages on the same server hardware or by running a given Web server package on different hardware platforms. WebBench's standard test suites produce two overall scores for the server: requests per second and throughput as measured in bytes per second. WebBench provides both static standard test suites and dynamic standard test suites. No matter which test suites you use, your PC clients must be running either Windows 95/98 or Windows NT/2000 and the controller must be running Windows NT or Windows 2000. WebBench workloads design for Web server software performance tests lack of stateful transactions between Web servers and client computers. These statefulless transactions workloads design can't efficiently measure layer seven content switches overall aspects including performance and functional verification of high service level network devices.

- **Sniffer** [15]

Sniffer is used to capture traditional LAN based network packets and support monitoring and analysis of various layer protocols. These protocols involve from layer 2 to layer 7, especially HTTP header and contents analysis we concerned. But

using Sniffer for layer seven content switches testing, it wouldn't achieve real time or uninterrupted tests and further cause the testing inexact result.

In order to achieve the real environment and stateful workload designs of layer seven content switch testing, we must focus on user behaviors and cookie maintenance mechanism developments. Furthermore layer seven content switches usually are used at E-Commerce Web sites, so E-Commerce user behaviors design must be considered importantly.

3.2 User Behaviors and HTTP Generator Design

Introduction

To make better user behaviors, there are several types of behavior models on E-Commerce. One model of them in [16] [18] analyzes the overall trends of customers or measures Web audiences at E-Commerce Web sites. Another model is analysis of access logs of Web server and then characterizes an probability model based on the characterization in [17][19]. For realistic E-Commerce user behavior, we construct entire behavior models in each category. These categories include the Browsing-&-Searching behavior state, the Shopping behavior state and the Payment-Procedure behavior state as well as achieve completely to emulate user actions and any behavior practices.

The traffic generator development, which is generally using common Web benchmark set to generate HTTP traffics. But this is not a better generator such as WebBench because of flexibility and scalability. It can scale the traffic to any intensity and configure the arrival characteristics to conform any specific Web sites, which is in [20]. The self-scaling and self-configuring aspects are undoubted approach to design

E-Commerce user behavior for HTTP generator of layer seven device testing. By means of these excellent points, we already have designed E-Commerce User Behavior Content Switch Testing System Module in according with these characteristics.

The E-Commerce user behavior, cookie maintenance mechanism and E-Commerce User Behavior Content Switch Testing System Module development describe in details in chapter four and five.

Chapter Four E-Commerce User Behavior Oriented Module

4.1 Introduction to E-Commerce User Behavior

In general, the E-Commerce (E-C) transaction procedure probably could separate into three parts. First, Internet user tough pages briefly without any intention, or objective browsing and searching concern pages. We call those actions user behavior state. Next, Internet user actions at E-Commerce procedure are shopping on the shopping Web sites and click particular items through the business Web pages. This behavior about buying a trade article or change order can group into a shopping state. After ordering, Internet shopping user proceeds to Checkout the order form at Web commerce procedure and we call that Payment-Procedure behavior state. In this Payment-Procedure behavior state, the Internet shopping user (registered customer) must login its identity number and password before, and then enter settling account in according with Web commerce procedure. In this payment procedure included selecting payment methods like MasterCard、Visa or Credit Account、entering personal information (like E-mail address、shipping address、country、city and address) 、selecting Shipping Method and other information applied for completing whole Web transaction. Some more excellent Web merchants offer safely billing procedure or fastest and easiest way to orders, like Amazon merchants 1-Click® [21] service. 1-Click is turned on immediately after customer place his first order, provided you've paid for that order with a credit card. The customer can place an order to any address he is shipped to before by clicking just one button. Amazon transaction

procedure will automatically reference your account for shipping and billing information.

An Online Shop may provide various kinds of features and services. In [2] lists these possible features and services of the major service type groups are enumerated. For layer seven content switch testing, we would be possible to refine the elements and then consider important parameters can influence the performance and accuracy of L7-CS operation. These parameters include search function, One-to-One production list, order and order tracking. Since this proposal, we can further propose a model for an Internet user behavior whether he is browsing Web pages, shopping on line or doing any Web accesses action. By means of Internet user behavior model provide E-C traffic characterization of WWW requests.

4.2 Traffic Characterization and Modeling for WWW E-C User Behavior Requests

4.2.1 An Overview of WWW E-C Requests Based On User Behavior

WWW E-C user behaviors can be classified into three different behaviors according to previous mention about Internet user activities on line. When an Internet user browse Web pages and search particular information can be categorized into Browsing-&-Searching behavior state (B&S behavior state). If that user want shopping on line thought Internet shopping mall, those products selected located at Shopping behavior state (S behavior state). And then the shopping customer must enter payment procedure after completing significance transactions that is called Payment-Procedure behavior state (P-P behavior state). The Actions between states

transition as show by Figure 4.1. These Three behavior states are using probability distribution to model and describe more details in the remainder of this section. Note that WWW E-C user behaviors of multiple Internet users usually overlap with each other.

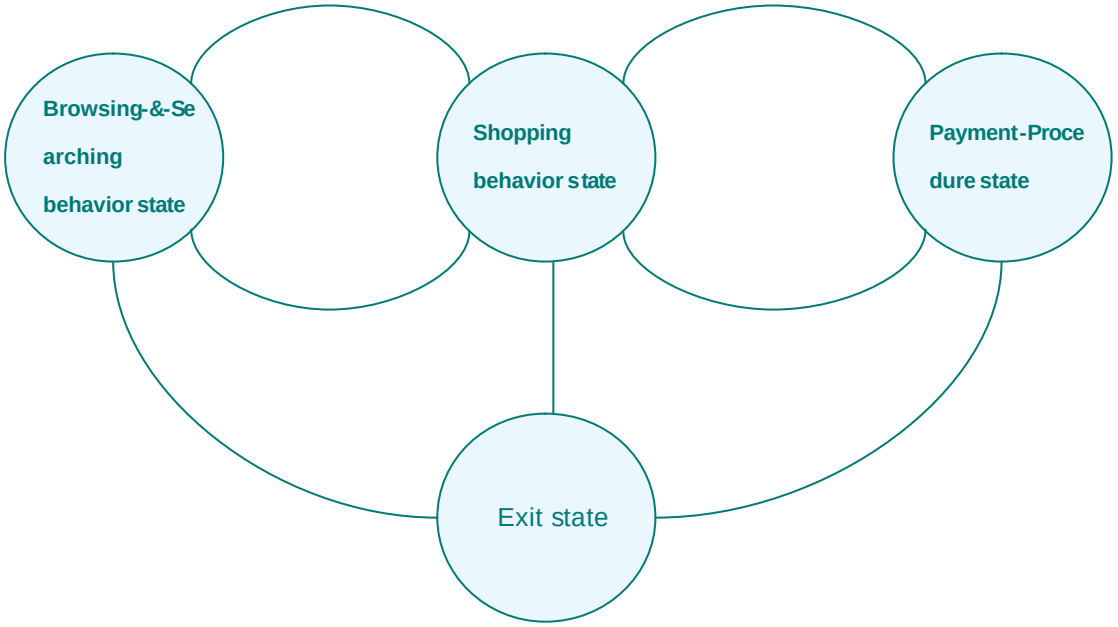


FIGURE 4.1 BEHAVIOR STATES TRANSITION

4.2.2 Discussion on Browsing-&-Searching behavior state

A Browsing-&-Searching behavior state starts when an Internet user initiates the first page request to remote server and ends when the last page request made by the Internet user is ready to exit or enter next state. Browsing-&-Searching behavior state sub-states transit diagram as shown in Figure 4.2. User behaviors represent two sub-states transition that are Browsing state and Searching state. In real Internet, the HTTP requests of Browsing state and Searching state has similar character although individual state request processes are differently way served by corresponding to

server. By means of this feature, Browsing state and Searching state can be considered in the same state for simply constructing unsophisticated HTTP pattern of requests, depicted by the diagram in Figure 4.3. When an Internet user clicks a Web page, Brower (such as Netscape or Internet Explore) automatically generates multiple HTTP requests to remote Web server. Each page request includes multiple HTTP requests and different kinds of request categories such as request sizes and file types. Presently more detail describes request categories which involve static pages, dynamic pages, PHP, ASP (Active Server Pages), HTML, and IMAGE files.

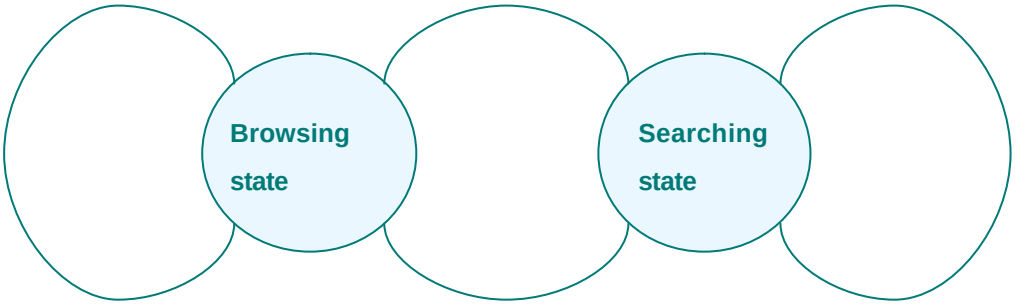


FIGURE 4.2 BROWSING-&-SEARCHING BEHAVIOR STATE TRANSITIONS

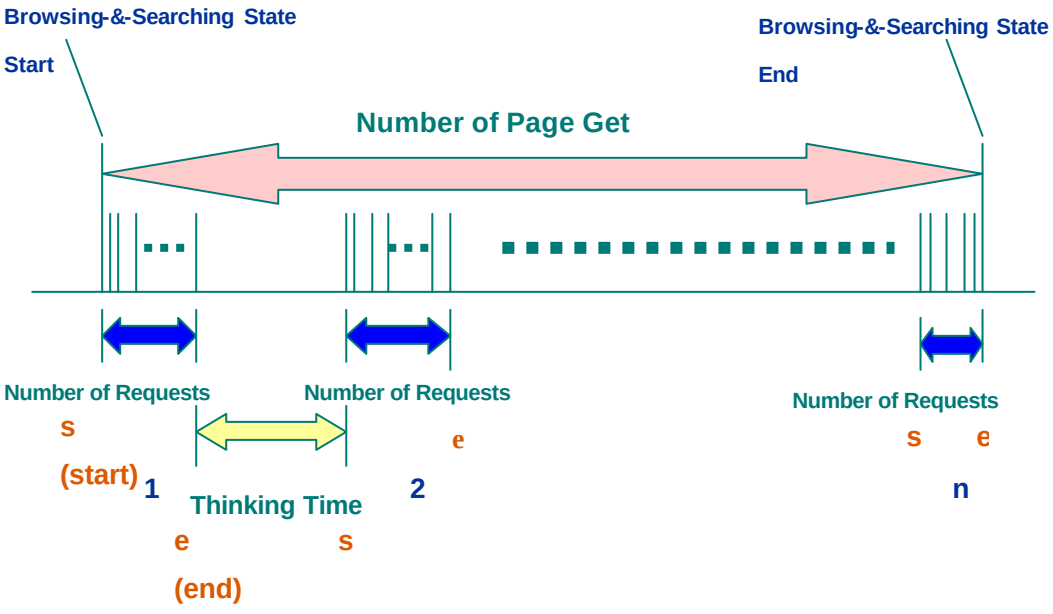


FIGURE 4.3 BROWSING-&-SEARCHING BEHAVIOR STATE HTTP PATTERN OF REQUESTS

Referring to HTTP pattern of requests in Figure 4.3, the number of pages represents that user initiating first page to last page in this state. The delay time between individual pages induced by the user is called Real Thinking Time (RTT). For layer seven content switch testing consideration, Real Thinking Time extend the life time of each page. If any request with cookie occurs, the life time of that page imply that survival time of its cookie. Each page can be considered as composing multiple relevant requests generated by Browser. Similarly, one page transmission time is equal to superpose multiple relevant requests and that time we abbreviate TST. The Browsing-&-Searching behavior state HTTP pattern of requests is examined and they are more details defined as follows:

Definition Number of Pages per B&S behavior state: An Internet user usually browses more than one page on remote Web server. When each B&S behavior state occurs, ***n*** is the number of pages got by user in average situation. Each B&S behavior state is hold by different Internet users which behave independently, and each Internet user in B&S behavior state visits total number of pages that follows the geometric distribution. When the k-th page has been got by particular Internet user, this denotes n^k . The major consideration of total number of pages should be generated by random process. ***N*** is total number of pages random variable in each B&S behavior state. The ***p*** is the probability of browsing one page in one B&S behavior state. The probability mass function of geometric distribution [3] is given by

$$P[N = n] = (1 - p)^{n-1} p \qquad n = 1, 2, ..., \qquad (4.1)$$

Definition Number of Requests per Page: Generating how many inline html

files and image files requests must be took into account when the Internet user had clicked single Web page. There is at least more than one request per Web page, and this assume that the number of requests accord to random process. In other word the number of files per Web page in remote Web server is arbitrarily arranged for particular advertisement. The probability of single request in one page that has clicked by the user is p. R is number of requests random variable located in one page. Therefore that probability distribution is similar to browsing number of pages and is given by

$$P[R = r] = (1- p)^{r-1} p \quad r = 1, 2, ..., \quad (4.2)$$

Definition Computed Thinking Time: The Computed Thinking Time (CTT) defines inter-arrival time between pages in each B&S behavior state. Precisely CTT estimates the time unit between the first requests of two pages and superimpose transmission time and Real Thinking Time among pages. This denotes $CTT = n_s^k - n_s^{k-1}$ *for* $TST \ll CTT$ **.** Note that this assume the transmission time is extremely small than Computed Thinking Time. In general the transmission time of one page is Exponentially distributed and we assume the Real Thinking Time also follows Exponential distribution, so Computed Thinking Time is Exponential distribution. However, the Real Thinking Time model of WWW arrivals belongs to Weibull distribution in [4]. This thesis quotes the latter because of providing better model of lift-length process. The Weibull distribution is given by cumulative probability function

$$F(x) = Prob.(t \le x) = 1- e^{-\left(\frac{x}{q}\right)^k} \quad x > 0 \quad (4.3\text{-}1)$$

And probability density function is

$$p(x) = \left(\frac{k}{q}\right)\left(\frac{x}{q}\right)^{k-1} e^{-\left(\frac{x}{q}\right)^{k}} \quad x > 0 \qquad (4.3\text{-}2)$$

Definition Page Size: For achieving accurate at user behavior in WWW world, the page size must be considered to model this HTTP pattern. There has location characteristic at boundary window of sizes by observing currently Amazon's Web pages. In this thesis we choose the most common files that are html files and image files, both of those files separately locates at 40~70k boundary window and 0~5k boundary window. By means of this characteristic observing, we assume the html and image file sizes pursuant follows Binomial distribution and the random variable of the size is ***S*** and ***z*** is fixed amount of html and image files that specify size variation. The Binomial distribution is given by probability mass function

$$P[S = k] = \binom{z}{k} p^{k} (1-p)^{z-k} \; for \quad k = 0,1.....,z \qquad (4.4)$$

As we shall see next chapter in our statistic result and probability parameters be gained the best value.

4.2.3 Discussion on Shopping behavior state

A Shopping behavior state is the sub-states of WWW E-C user behavior. This state starts when the Internet customer selects the desirable products such as

magazines, books or mobile phone at Web shopping-cart. These actions in this state includes gathering or removing product items from that visited Web site and next checks those products selected. The Shopping behavior state is more sophisticated than the Browsing-&-Searching behavior state, Figure 4.4 is the Shopping behavior state transitions diagram. If the Add Item(s) state temporarily exit to transit forward to Remove Item(s) state, the Remove Item(s) state could be transit backward to Add Item(s) state latter or terminated forever by the customer next time. In other words, an Internet customer can round backward and forward repeatedly to buy desirable articles, and further the customer might temporarily exit to search or browse Web pages or will permanently terminate this transaction to enter Payment-Procedure. Whatever hopes to model the WWW E-C user behavior more realistic and flexibility.

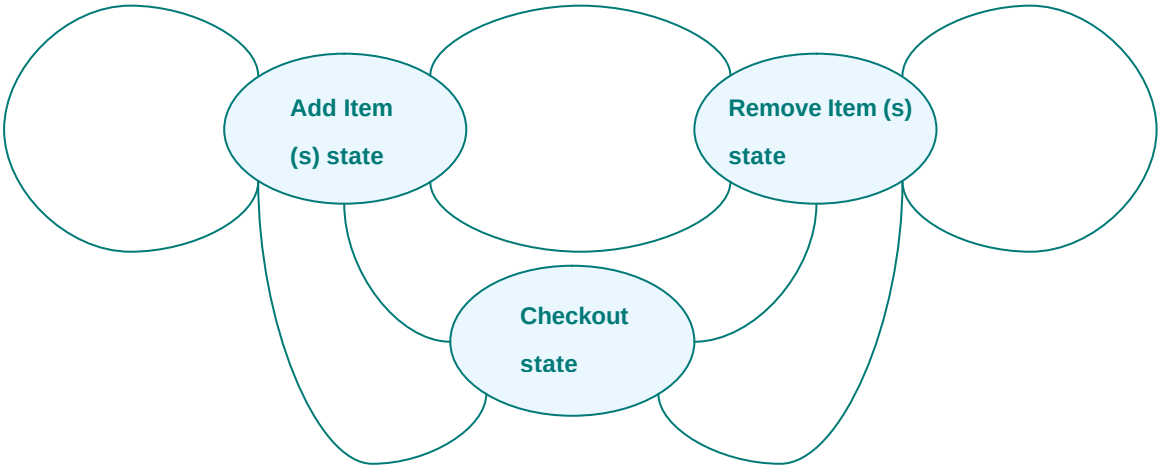


FIGURE 4.4 SHOPPING BEHAVIOR STATE TRANSITIONS

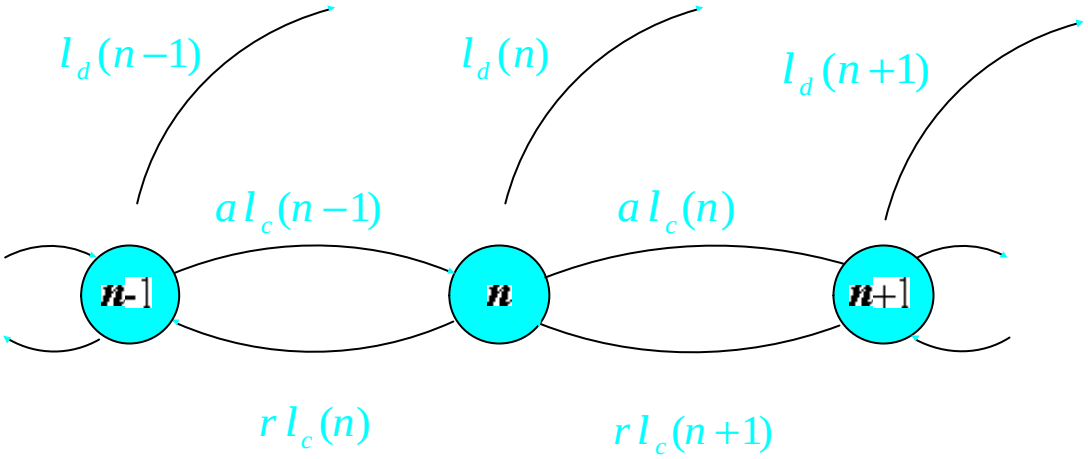


FIGURE 4.5 SHOPPING BEHAVIOR STATE TRANSITION SHOPPING-PROCEEDING CHAIN

We can transform the Shopping behavior state into precisely shopping-proceeding train with Internet user practice, and as shown in Figure 4.5. The number of elected items that denote n represents the customer purchase condition, and note that n is less or equal in total items that are provided by Web merchants. Don't confuse the number of elected items n with the number of pages n that had described above. The parameter l represents a rate of action occurrences that are the Web page clicked by a customer. The action occurrences with mean lt are made by a customer according to a Possion process. The number of action occurrences A(t) in the interval[0,t] with mean lt has

$$P\left[A(t)=a\right]=\frac{(lt)^{a}}{a!}e^{-lt} \quad for \quad a=0,1,..... \quad (4.5)$$

Exactly explaining these actions represent that are picking up items, removing the selected items or terminated temporarily by customers. First, a customer at shopping

state is continuing to purchase or remove merchandises must be corrected with correction parameter C_n that is at ***n***-th Shopping behavior state. Notice that the correction parameter C_n variations are dependent on ***n*** and the curves of C_n as shown in Figure 4.6. "C_n Curve a" represents strength of shopping to have power increasingly at the front region and decrease quickly at the back region. "C_n Curve b" is the strength of shopping decreasing exponentially at entire region.

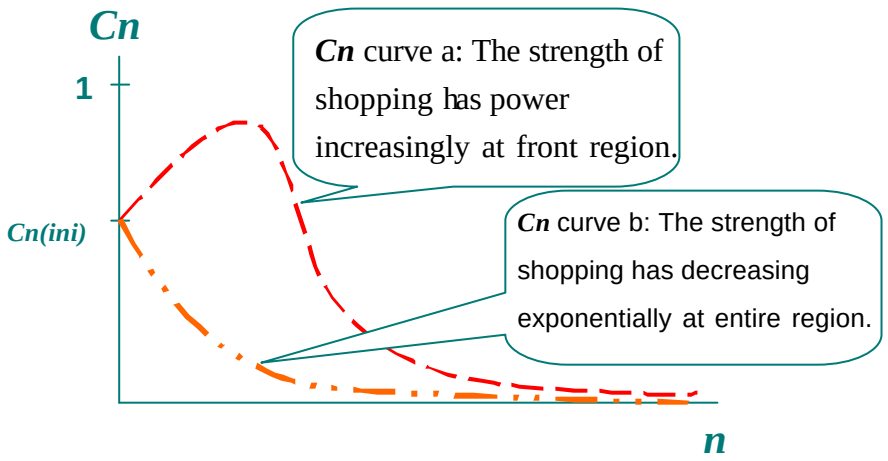


FIGURE 4.6 THE CURVES OF C_n EXAMPLES

The correction of l at ***n***-th Shopping behavior state denotes $l_c(n)$ that is l multiplied by C_n. The deviation of l at ***n***-th Shopping behavior state $l_d(n)$ is the error between l and $l_c(n)$, and that represents a customer going to terminate temporarily or exit forever. We assume the correction parameter C_n is Bernoulli process and depends on ***n***. The probability parameters imply that following properties:

$$\begin{cases} l_c(n) / l_d(n) > 1 & (4.6\text{-}1) \\ l_c(n) / l_d(n) = 1 & (4.6\text{-}2) \\ l_c(n) / l_d(n) < 1 & (4.6\text{-}3) \end{cases}$$

The equations (4.6-1), (4.6-2) and (4.6-3) separately imply the strength of staying at shopping-cart, fair action and going out of shopping-cart. And next factor of shopping state behaviors is the merchandises selection; this is of course dependent on different habitual behaviors of customers. In this thesis it simply characterizes the behaviors as a Bernoulli process. The strength of adding a new item and removing a selected item at ***n***-th Shopping behavior state is

$$\begin{cases} a\, l_c(n) \quad for\ a + r = 1 & (4.7\text{-}1) \\ r\, l_c(n) = l_c(n) - a l_c(n) & (4.7\text{-}2) \end{cases}$$

We have the relationship of ***a*** and ***r*** implied that

$$\begin{cases} a > r & (4.8\text{-}1) \\ a \le r & (4.8\text{-}2) \end{cases}$$

In equation (4.8-1) implies that the strength of addition items and the customers intend to buy particular merchandises at this shopping period. Inversely, equation (4.8-2) the browsers have poor interests to buy any products or change his mind about purchasing in this shopping period. We have a and r followed Bernoulli strengths.

Finally, what is going to after finishing this state? It depends on p_{term} whether the customers change his mind to continue shopping, decide to browse and search the particular pages or enter next Payment-Procedure states. We call this sub-operation is Checkout state that is show in previous figure. Also, p_{term} is Bernoulli definitely

distributed and let p_{term} guide the customers to enter into Browsing-&-Searching behavior state or Payment-Procedure behavior state. Payment-Procedure behavior state is discussed in more details latter.

4.2.4 Discussion on Payment-Procedure behavior state

The Payment-Procedure behavior state represents the final step of E-Commerce transactions as well as the unsafe step is started. The consideration factor is how to protect the sensitive information of purchasers (e.g., customer name, payment information, shipment information and a credit card number). This is an important issue to construct a safe Payment environment.

In this behavior model, the purchaser repeatedly fills out the forms state by state after login correctly. Please refer to Figure 4.7 and now we consider two factors about this state that is procedure number and thinking time. The procedure number determined by Web designers is repeat periods of the Payment-Procedure states and typically about 4 to 5. The thinking time is also determined using Weibull distribution discussed or Exponential distribution above. After making rounds of this procedure, an E-Commerce transaction is finished and a deal between the purchaser and corresponding vendor is successful. At this state we must consider another aspect that is Payment-Procedure state itself. Different Payment-Procedure states has dissimilar procedure processes and as shown in Figure 4.8.

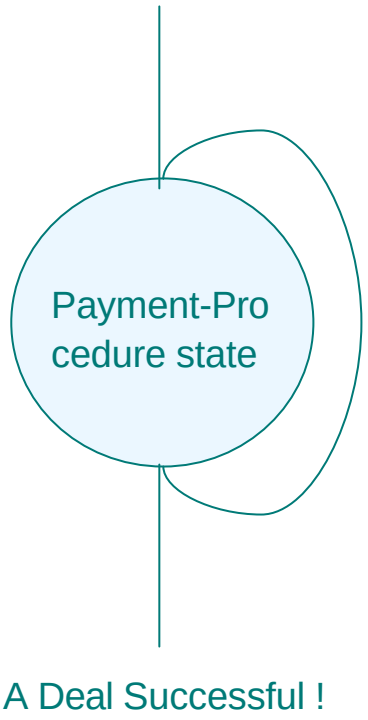


FIGURE 4.7 PAYMENT-PROCEDURE BEHAVIOR STATE

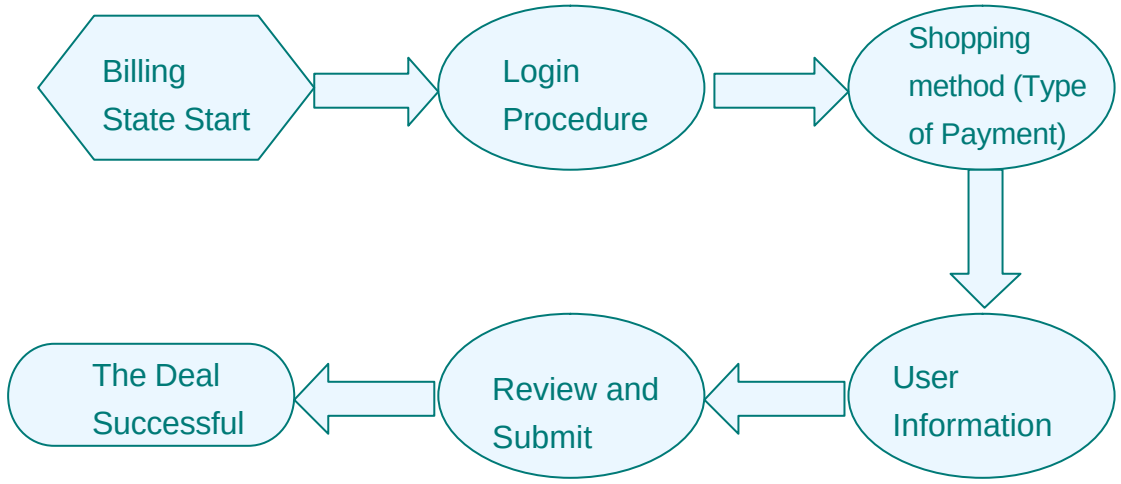


FIGURE 4.8 PAYMENT-PROCEDURE BEHAVIOR STATE CHARACTERISTIC FLOW CHART

When a purchaser enters this Payment-Procedure behavior state, corresponding Web server must identify him as a certified user first. So that purchaser send ID

(Identification) and Password at login or sign-in certification procedure for initiating Payment-Procedure. The Web server receives ID and Password to certify the purchaser in accordance with the user database at remote Web server. If the purchaser is a certified user, the Payment-Procedure proceed to next sub-procedure and if isn't, the new user must apply for a certified user. After login, the purchaser separately sends his payment information, shipment information, the sensitive information and other information with this transaction demand. Lastly, the purchaser confirms the order and personal information and submits to remote Web server. After these complexity procedures and state operations, a whole business deal in line is finished and success.

We achieve real E-Commerce user behavior oriented layer seven content switch testing, the previous discussion is examined. Furthermore, E-Commerce user behavior model not only for content switch testing but also can measure E-Commerce Web server or other field refer to Internet user behaviors.

Chapter Five E-Commerce User Behavior Content Switch Testing System Module

5.1 Module Development

In this Chapter we discuss the E-Commerce User Behavior Content Switch Testing System Module (E-CUB Module) has been developed by using the C++ language in this thesis. The E-CUB Module consists of six main components: the Parameter Input and Central Controller (PI-CC), the E-Commence User Behavior Generator (E-CUBG), the Probability Distribution Generator (PDG), the Cookies Maintenance Generator/Verifier (CMG/V), the HTTP Request Generator (HTTP-RG) and the Real Time Data Monitor (RTDM). As shown in Figure 5.1, the components of E-CUB Module will be described separately more details latter section. Except the E-CUB Module, this test must include the server workloads design and will also interpret the details latter in this thesis. The testing user can operate these components through a Graphical User Interface (GUI) that is embedded in the Parameter Input and Center Controller and control the E-CUB Module by the same component PI-CC.

5.1.1 The Parameter Input and Central Controller

The PI-CC composes Graphical User Interface, Parameter Input Mode (PIM) and Central Controller. The E-CUB Module generates HTTP requests of the E-Commerce user behavior that is according to the particular probability model discussed previous chapter. And Real Thinking Time follows Exponential distribution

for easy to use. These parameters of probability model include:

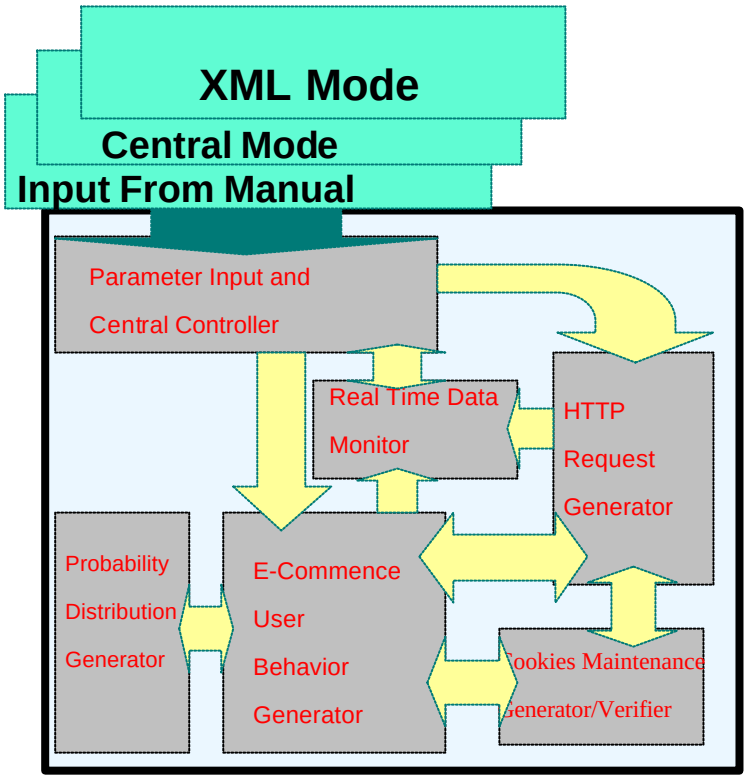


FIGURE 5.1 E-CUB MODULE COMPONENTS

- **B&S behavior state:**

  1. Page Size (Binomial distribution)
  2. Number of Requests per Page (Geometric distribution)
  3. Number of Pages (Geometric distribution)
  4. Thinking Time (Exponential distribution)

- **S behavior state:**

  1. Rate of action occurrences (Possion process)
  2. Correction parameter $C_{n(ini)}$ (Bernoulli process)
  3. Shopping strength $\boldsymbol{a}$(Bernoulli strength)

4. State Transition Parameter p_{term} (Bernoulli process)

- **P-P behavior state:**

1. Procedure number
2. Thinking Time (Exponential distribution)

- **Testing Method:**

1. Entire users generation per test
2. Number of clients generation per computer
3. Server URL

The parameters of behavior states are described in chapter four besides testing method. The testing method is the last setting of the E-CUB Module, which includes the testing quantity, the number of threads and Web server address. The testing quantity is entire users generated per test. The number of threads is equal to number of clients which is generated per testing computer. The Web server address specifies the URL (Uniform Resource Locator) of corresponding remote Web server.

The PIM includes three mode of setting parameter:

- **Input From Manual (Default setting)** :

  This mode is default setting and convenient for slightly adjusting the parameters of user behaviors during a test. There illustrates an example as shown in figure 5.2.

- **XML Mode**:

  The Extensible Markup Language (XML) [5] is the universal format for structured documents and data on the Web. XML is a set of rules for designing text formats that let you structure your data. XML makes it easy for the computer to generate parameter, read parameter, and ensure that the

parameter structure is unambiguous. And it is extensible, platform-independent, and it supports internationalization and localization. XML makes use of *tags* (such as <price>) and *attributes* (of the form name="test"). XML uses the tags only to delimit sections of parameter, and leaves the interpretation of the parameter completely to the testing system that reads it.

The following is E-CUB Module XML parameter format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<InputParameter>
    <BrowsingandSearching>
        <BSDS> Page Size HTML Files</BSDS>
        <BSDSI> Page Size Image Files</BSDSI>
        <BSNA> Number of Requests per Page </BSNA>
        <BSND> Number of Pages </BSND>
        <BSTT> Thinking Time </BSTT>
    </BrowsingandSearching>

    <Shoppingcart>
        <SR> Rate of action occurrences </SR>
        <SCDCP> Correction parameter  C_n(ini) </SCDCP>
        <SARP> Shopping strength α </SARP>
        <SSTP> State Transition Parameter  p_term </SSTP>
    </Shoppingcart>

    <Accounting>
        <APN> Procedure number </APN>
        <ATT> Thinking Time </ATT>
    </Accounting>

    <TestingMethod>
        <TMTUTI>NULL</TMTUTI>
        <TMTUUN> Entire users generation per test </TMTUUN>
```

<TMNCPP> Number of clients generation per computer </TMNCPP>

<TMHTTPURL> Server URL </TMHTTPURL>

</TestingMethod>

</InputParameter>

The XML Mode operation of E-CUB Module illustrates an example as shown in figure 5.3.

- **Central Mode** :

This mode provides entire solution of Content Switch Testing System that is controlled from remote controller located within another testing computer. When this mode is selected by testing user, the remote controller manipulates the E-CUB Module completely and testing user can operate facility the E-CUB Modules distributed in all client computers including initial E-CUB Modules, XML documents transmission, start testing and data report.

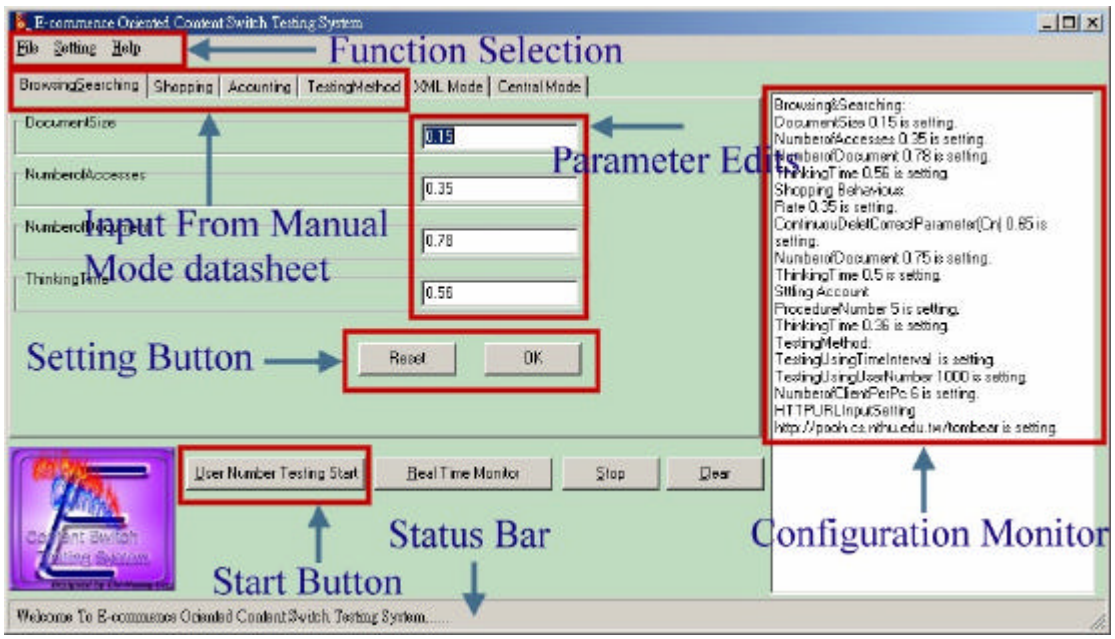


FIGURE 5.2 THE E-CUB MODULE INPUT FROM MANUAL MODE

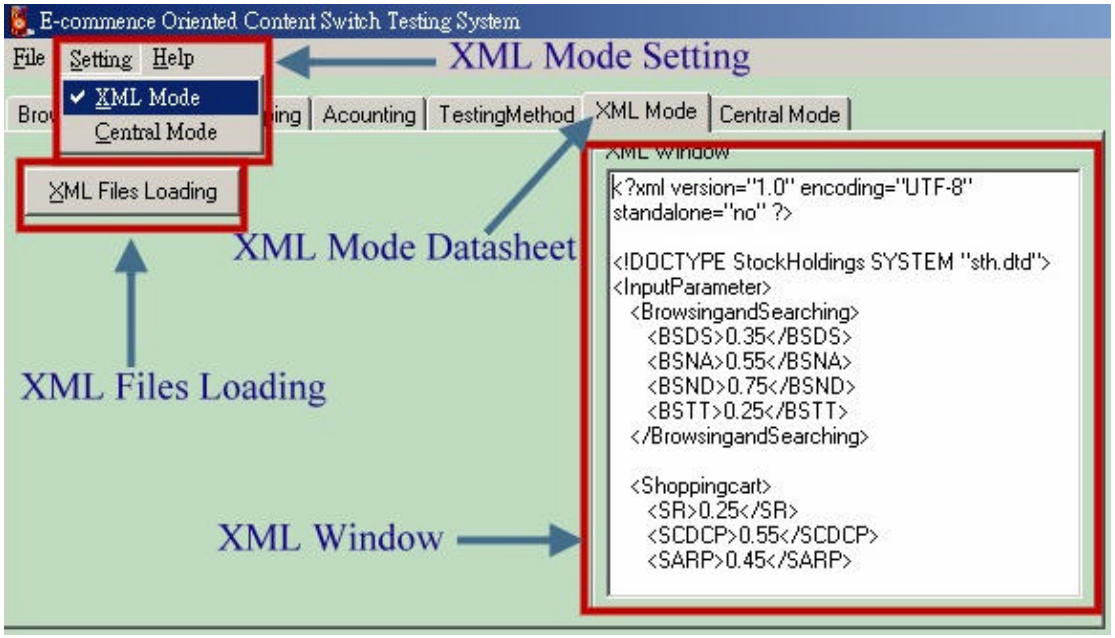


FIGURE 5.3 THE E-CUB MODULE XML MODE

5.1.2 The E-Commence User Behavior Generator

This component is used to arrange the user behavior procedure based on the probability parameters generated by the Probability Distribution Generator and cookies arrangement through the Cookies Maintenance Generator. The user behavior procedure makes use of the Browsing-&-Searching behavior, the Shopping behavior and the Payment-Procedure behavior in the foregoing chapter. Note that each generated user has own thread and the users are supposed to work independently to each other.

5.1.3 The Probability Distribution Generator

This component is responsible to generate user behavior probability parameters in accordance with respectively specifying probability distribution and parameters

setting by testing user. This probability distribution composes of the Binomial process, the Geometric process, the Weibull process, Bernoulli process and the Possion process. These user behavior probability parameters provide the E-Commence User Behavior Generator to generate HTTP requests with E-Commerce user behaviors.

5.1.4 The Cookies Maintenance Generator/Verifier

This component is like an agent to maintain cookies generation and verification within each user behavior HTTP request. When a new user or customer is created by the E-Commence User Behavior Generator, a new HTTP request without cookie also is ready to send to remote Web site at the same time. If Web server receives that request, it would set-cookie to corresponding client computer. The cookie format is "server="cookie number"; expires=week, date GMT time". And then this component must simulate the real user request another page with the same cookie, furthermore even the user has entered shopping period. If the layer seven content switches dispatch the request with cookie to the wrong Web server, this Verifier must analysis and detect this cookie persistence error.

5.1.5 The HTTP Request Generator

If the E-Commence User Behavior Generator has already done, the HTTP Request Generator is responsible for creating a characteristic request. Before a characteristic request is initiated by this component, that Generator must open an individual TCP connection [6] for a new HTTP request or use the connected one for proceeding request. The HTTP requests and TCP connections in this thesis are using Keep-Alive Connection that is according to HTTP 1.1[7].

5.1.6 The Real Time Data Monitor

This component will handle two data processes that are monitoring real time data and reporting the final testing results. The real time data includes {Requests per Second; Connection Failure Count}, {Deal Successful Count; Deal Error Count}, {Cookies successful Count; Cookies Number Count} and {Pages Successful Count; Pages Error Count}. This Real Time Data Monitor is designed for testing user convenience to observe each E-CUB Module status as shown in figure 5.4. The final testing results report following statistics list, for example:

XXX request per second

XXX request account

XXX successful requests per second

XXX connection error

XXX connection error rates

XXX page request number

XXX page error

XXX page request number per second

XXX page error rates

XXX cookies account

XXX cookies error

XXX cookies account per second

XXX cookies error rates

XXX deal account

XXX deal error

XXX deal account per second

XXX deal error rates

These results are examined in more details in the next section of this chapter.

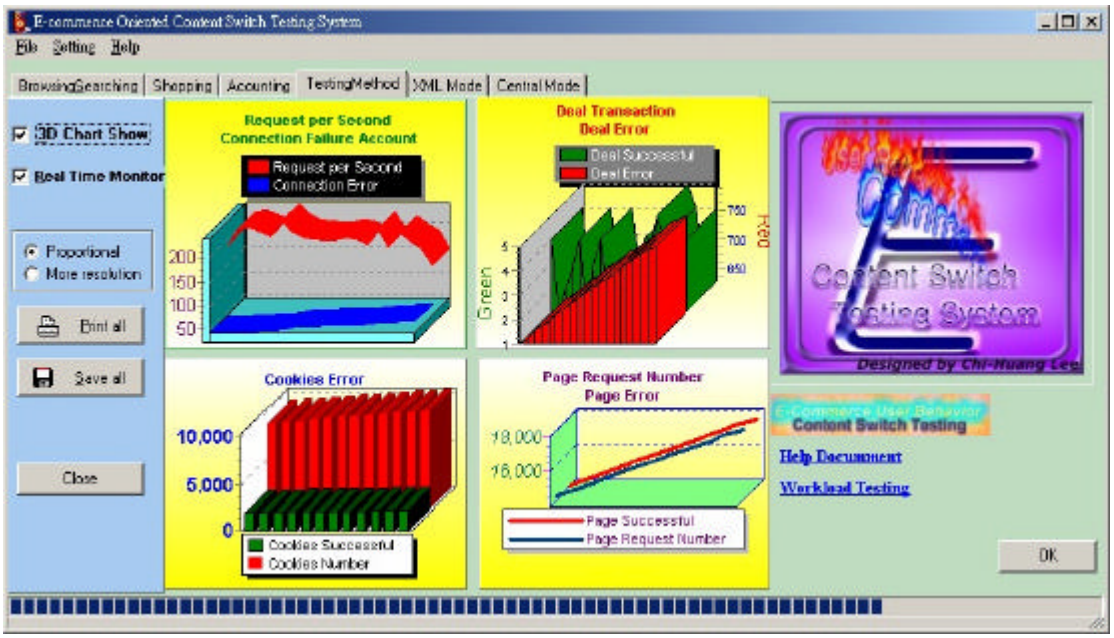


FIGURE 5.4 E-CUB MODULE REAL TIME MONITOR

5.2 Server Workload development

5.2.1 Page Sizes Design

This thesis analyzes the greater part Web pages of the famous Web merchants Amazon.com, Inc. [1] and gathers the particular statistics about page sizes. The

statistics are grouped into two catalogs: HTML documents as well as Image files that are totally based on static pages. In figure 5.5 and 5.6, represent the statistics of static page sizes. The HTML documents approximately are located around 40k~70k, especial half of that is existing work at 50k~60k. The Image file statistics differ from the HTML documents; this is because that a great deal of using small icons, the segment pictures or tiny logotypes. The 98 percentages of Image file statistics are less than 5k image sizes. This statistic data mixes JPEG image files and GIF (Graphics Interchange Format) image files [8] that are usually used to behave in the Web pages. The heterologies of image types don't affect transmission characteristics through Internet.

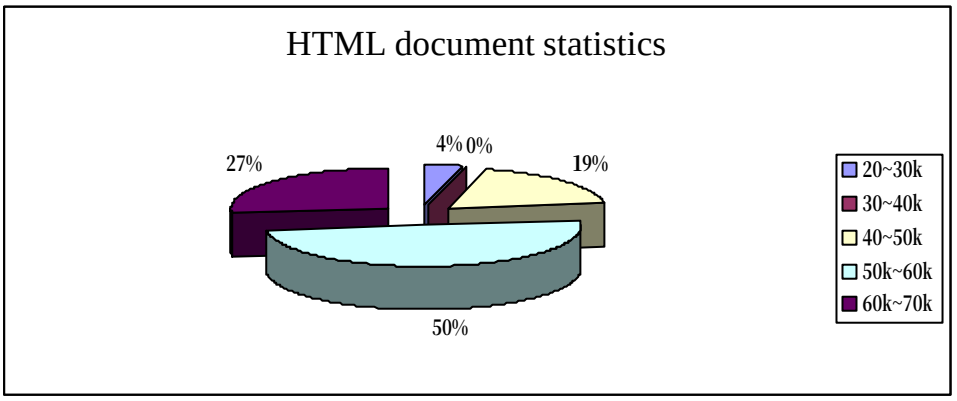


FIGURE 5.5 THE HTML DOCUMENT STATISTICS

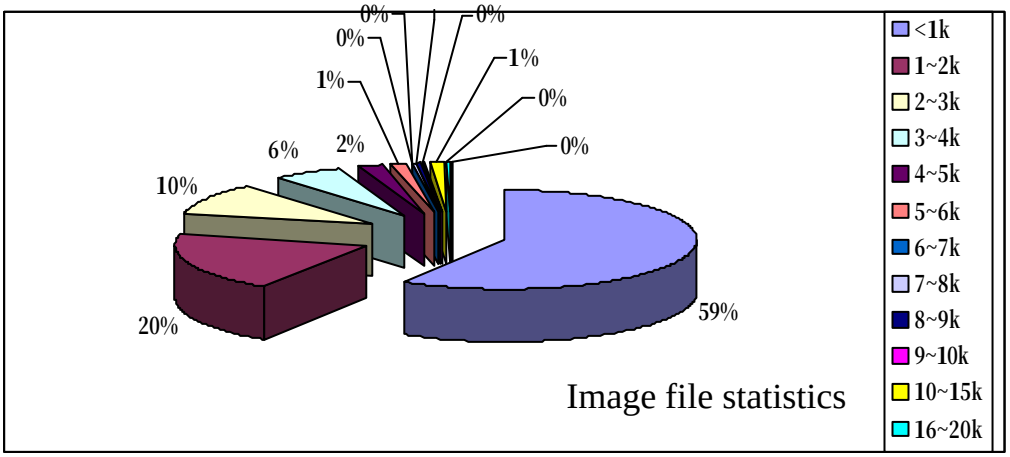


FIGURE 5.6 THE IMAGE FILE STATISTICS

5.2.2 Page Types Design

Except HTML documents and Image files previous mention, there also have dynamic pages behaved and exist work in the Internet simultaneously. Dynamic pages do not exist until a user asks for information e.g., submits a search request, places an order, signs up for a service. In the present days of the Web the server-side scripting language use PHP [9] (which stands for PHP Hypertext Preprocessor), Microsoft Active Server Pages [10] (ASP/VBScript) and Allaire Cold Fusion.

PHP is an open-source server-side scripting language intended specifically for use on the Web. Like C++ and Java, it has a heavily object-oriented structure and it's free. PHP can be used in both main Web server environments, On Windows Web servers PHP can be added to Microsoft IIS, Personal Web Server and Apache as well as On Unix and Linux Web servers it is most commonly added as an Apache module.

ASP is a technology developed by Microsoft to take text scripts in an HTML context and run them on the Web server to create dynamic and interactive pages. Until very recently, ASP was limited to Web servers running IIS and Microsoft Personal Web Server. But then Chili!Soft developed an ASP server for Unix and Linux.

Because PHP provided free development environments and was easy to use, we finally chose PHP to create dynamic pages for our testing server end design. These dynamic pages in this testing system developed include Shopping-Basket, Payment-Procedure, and Cookies Operation and will describe more details in latter sections.

5.2.3 Shopping-Basket Design

This thesis isn't emphasis of visual emotions and user interfaces, we focuses on the function integrity of all Web side development instead. The user interface of Shopping-Basket is pure text format show and flow chart of transaction process is described in figure 5.7.

5.2.4 Payment-Procedure Design

In final procedure of E-Commerce transaction is more important than other action in merchant Web server. Except the related steps have been mentioned in chapter four, there have some corresponding work handled by the merchant Web server and flow chart shown in figure 5.8.

5.2.5 Cookies Operation Design

The implement of cookie operations are divided into two parts: the E-CUB Module and server workload design. Both of them implement are based on the general cookies operation in the Web server, and programming using PHP function and C++ language. We have discussed these operations in details foregoing sections and previous chapter four that don't mention verbosity here.

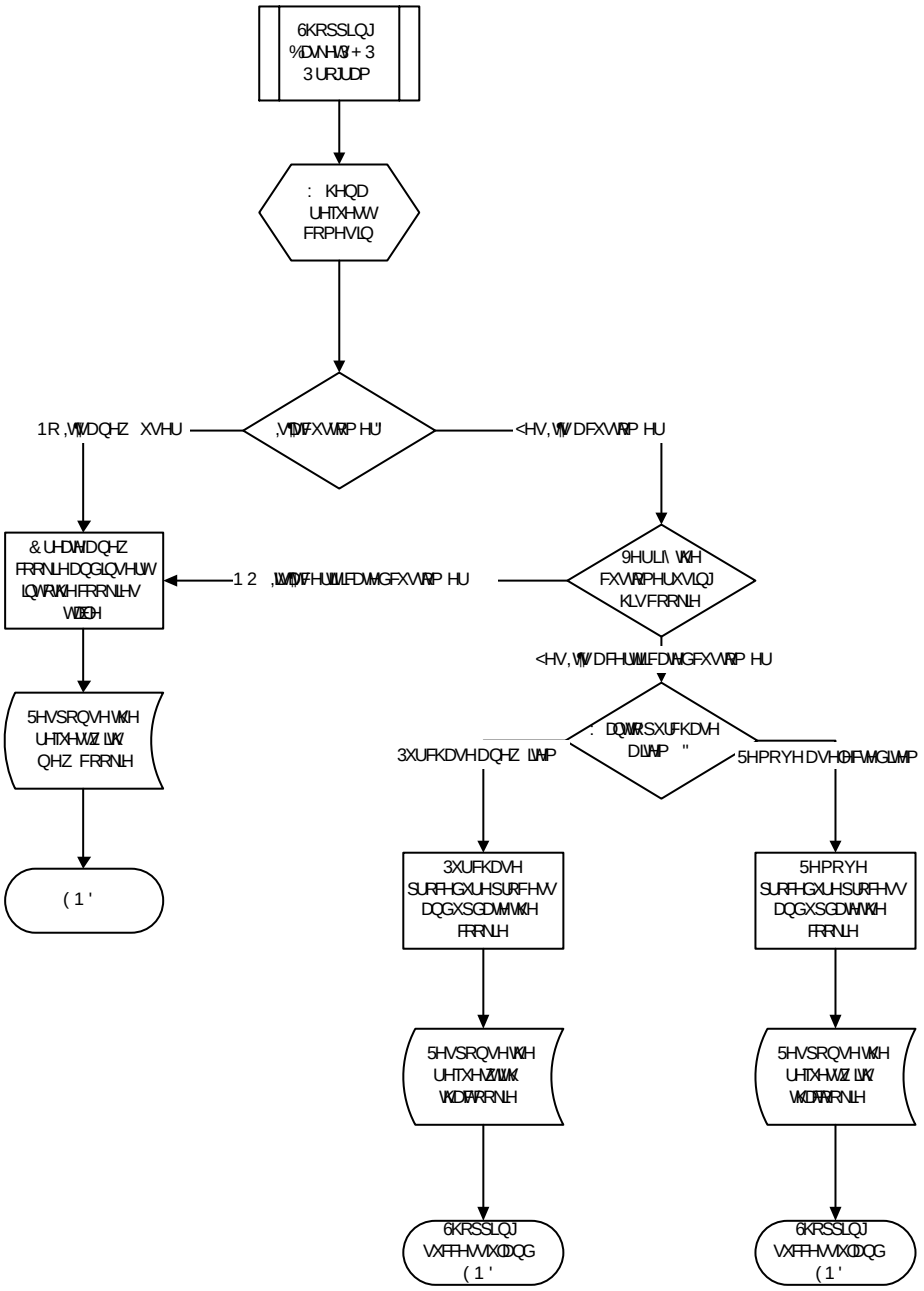
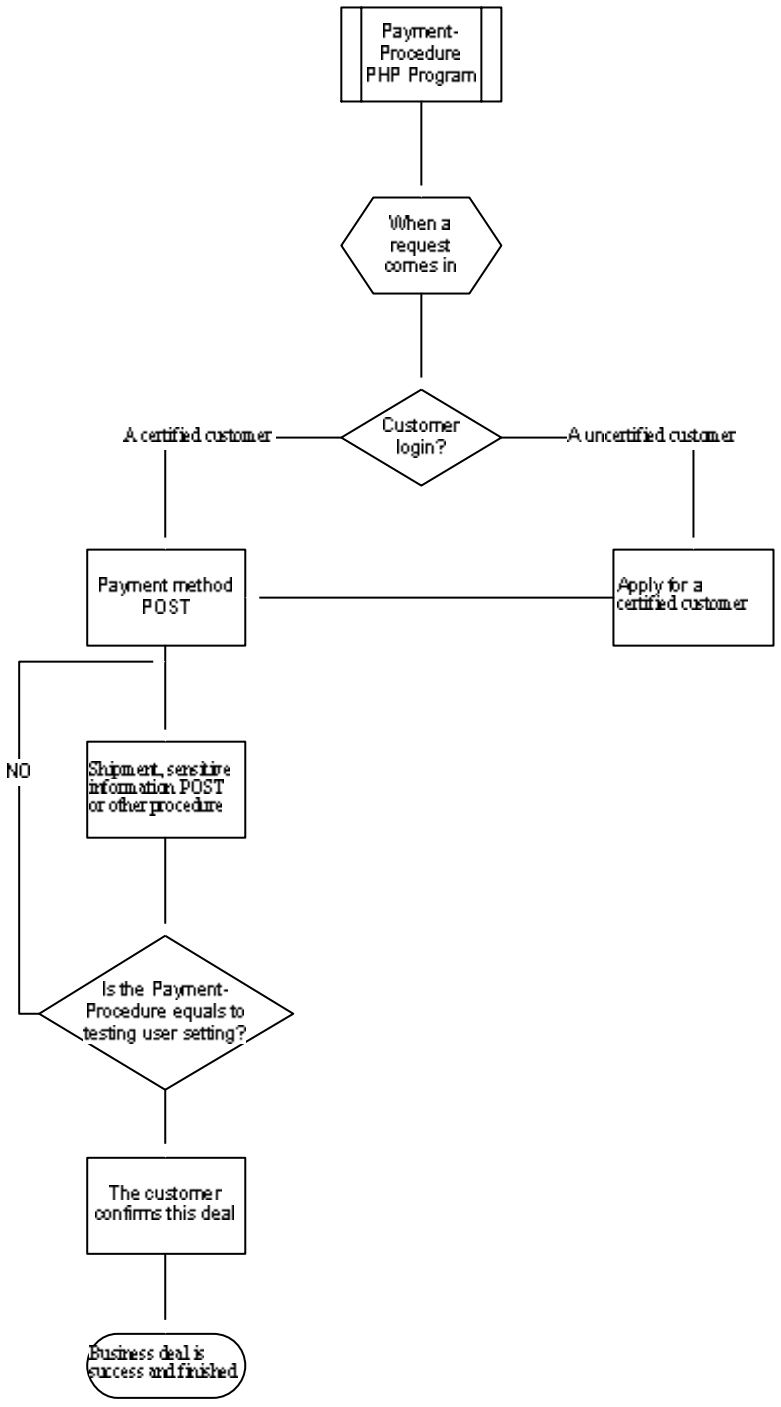


FIGURE 5.7 THE SHOPPING-BASKET FLOW CHART


Payment-Procedure PHP Program
When a request comes in
Customer login?
A certified customer
A uncertified customer
Payment method POST
Apply for a certified customer
NO
Shipment, sensitive information POST or other procedure
Is the Payment-Procedure equals to testing user setting?
The customer confirms this deal
Business deal is success and finished

5.3 Result Definition

In this thesis, we define the four important estimating units to observe deliberately accurate in testing Layer seven content switches. These definitions are:

Hit Rate (?):

The unit of Hit Rate is specific to the cookie persistence accurate verification. A hit represents the request with cookie to re-establish a user's connection to a particular server that may have data associated with a specific user. If the request of that user is dynamically redirect to other server at the same site, this records a cookie error or hit failure in each cookie service period. The formula is

$$Hit\ Rate = \frac{Requests\ With\ Cookies - Hit\ Failures}{Requests\ WithCookies} \times 1000 \qquad (5.1)$$

Connection Failure Rate (?):

The Connection failures show the health condition of networks and the performance of tested devices. In another word, this depends on network bandwidth and utility of the tested content switch. We define

$$Connection\ FailureRate = \frac{Connection\ Failures}{Total\ Requests} \times 1000 \qquad (5.2)$$

Page Failure Rate (?):

We define page failures in accordance to the HTTP Error Messages that follow [7]

HTTP Error 400: **Bad Request**

HTTP Error 404: **File Not Found**

HTTP Error 500: **Internal Server Error**

The formula is

$$PageFailureRate = \frac{PageFailures}{Total\ Page\ Requests} \times 1000 \qquad (5.3)$$

Deal Successful Rate (?):

For the Layer seven content switches testing, we hope to promote a standard estimate unit that fully consists of the accuracy and performance of the tested device. The “Deal Success” represents an E-Commerce transaction on the Web which completely behaves at browsing pages, searching particular contents, shopping and payment procedure. Once error occurs whether awareness or not in each unfinished deal, that must be strictly recorded into a deal error. We define

$$Deal\ Successful\ Rate = \frac{Total\ Deals - Deal\ Failures}{Total\ Deals} \times 1000 \qquad (5.4)$$

5.4 Testing Environment

Finally, we briefly illustrate the test of Layer seven content switches using the E-CUB Module. The specifications of devices and apparatuses and the testing environment are separately shown in figure 5.9 and figure 5.10.

Devices/apparatuses	***Specifications***	***Notes***
Content Switch	Alteon 184	Gigabit Ethernet Interface
HTTP Servers	● Intel Pentium III 800MHz ● RAM 128M ● Intel 10/100 Ethernet Card ● Apache HTTP Server 2.0	HTTP Server * 8
HTTP Clients	● Intel Pentium III	HTTP Client * 12

	800MHz ● RAM 128M ● Intel 10/100 Ethernet Card ● E-CUB Module	
Central Controller	● Intel Pentium III 800MHz ● RAM 128M ● Intel 10/100 Ethernet Card ● E-Commerce Oriented Content Switch Testing System	Central Controller * 1
Layer Two Switch	Accton CheetahSwitch Workgroup-3526C	24 10BASE-T/100BASE-TX (RJ-45) Ports, and 2 Slots for Gigabit Uplink Modules

FIGURE 5.9 DEVICE SPECIFICATIONS

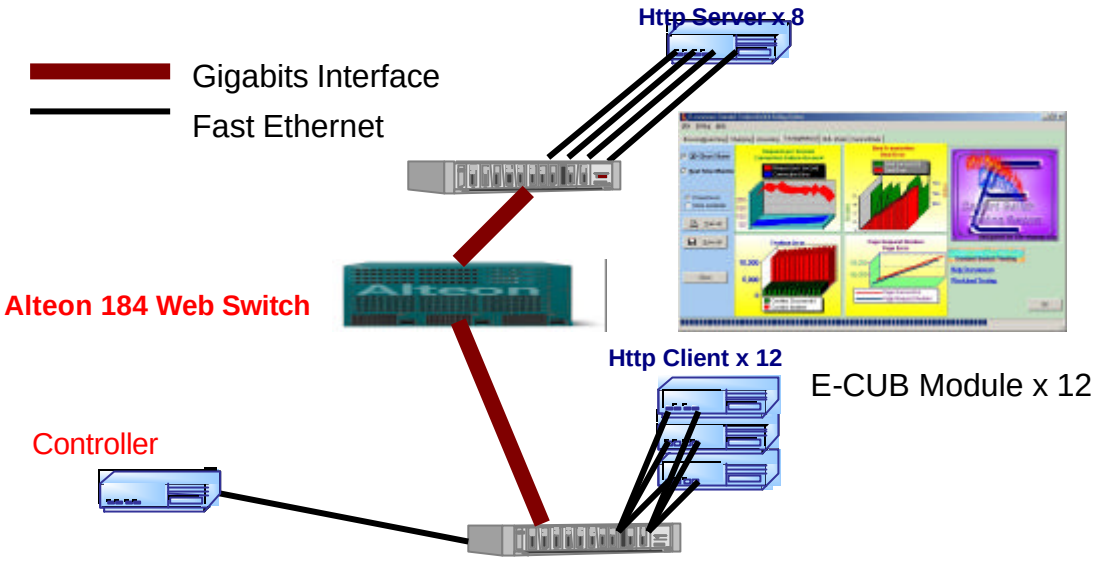


FIGURE 5.10 THE TESTING ENVIRONMENT

5.5 Testing Results

The parameters setting follows below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<InputParameter>
    <BrowsingandSearching>
        <BSDS>0.001</BSDS>
        <BSDSI>0.001</BSDSI>
        <BSNA>0.5</BSNA>
        <BSND>0.5</BSND>
        <BSTT>0.99</BSTT>
    </BrowsingandSearching>

    <Shoppingcart>
        <SR>0.99</SR>
        <SCDCP>0.5</SCDCP>
        <SARP>0.5</SARP>
        <SSTP>0.5</SSTP>
    </Shoppingcart>
```

```
    <Accounting>
        <APN>3</APN>
        <ATT>0.99</ATT>
    </Accounting>

    <TestingMethod>
        <TMTUTI></TMTUTI>
        <TMTUUN>1500</TMTUUN>
        <TMNCPP>5</TMNCPP>
        <TMHTTPURL>http://192.168.2.201/server</TMHTTPURL>
    </TestingMethod>
</InputParameter>
```

We set the small page size and the short thinking time of this tests and 192.168.2.201 is an address of eight servers through Alteon Content Switch. About 360 thousands requests have occurred and one thousand deal transactions are applied of this test. We have

- The volume generated by the E-CUB Modules and the performances of the Alteon 184 are shown in figure 5.11. The performance of layer 4 configuration is more excellent than layer 7.
- In Figure 5.12, the health of testing network is fine which refers to **lower Connection Failure Rate** and **Page Failure Rate**. Obviously the Passive mode of Alteon Content Switch has superior accuracy and functionality of this test.

Finally we briefly summarize this test using E-CUB Module

1. Layer 4 has excellent performance.
2. Layer 7 using Passive mode has cookie operation extreme powerful efficacy.
3. Layer 7 using Passive mode has seamless services of E-Commerce in Web sites.

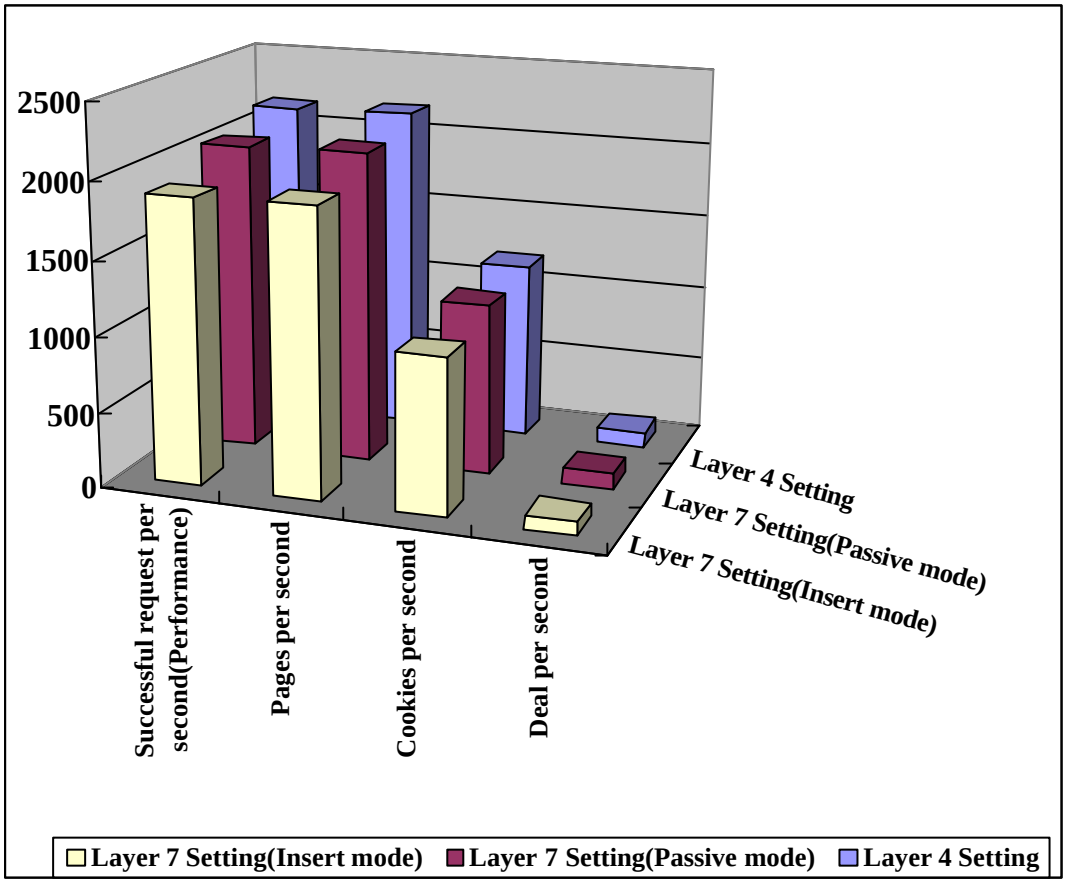


FIGURE 5.11 PERFORMANCES AND VOLUME TESTING

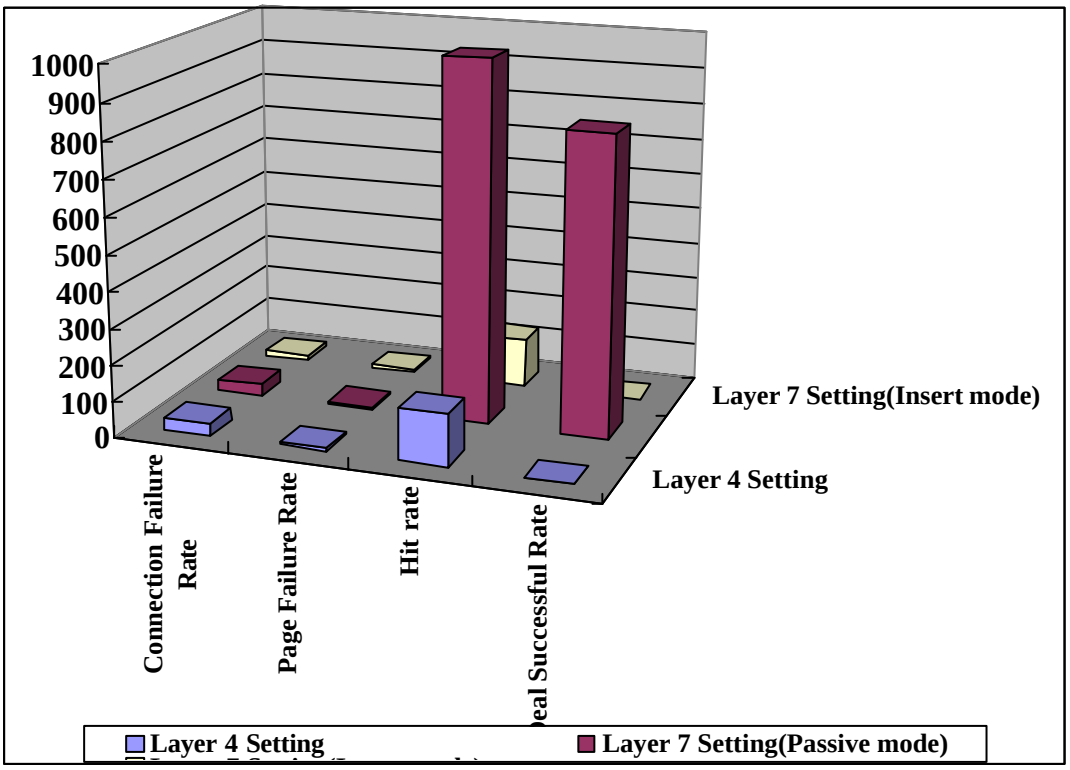


FIGURE 5.12 THE FUNCTIONALITY TESTING RESULTS

Chapter Six Conclusion

This thesis have proposed E-Commerce user behavior oriented model and incorporate cookie maintenance mechanism to account for the insufficiencies in Layer seven content switch tests. E-Commerce user behavior oriented model including Browsing-&-Searching behavior state, Shopping behavior state, Payment-Procedure behavior state represent a complete commerce procedure online. cookie maintenance mechanism is a critical technology and essentially demands at E-Commerce Web sites. Based on this model and mechanism, we further have developed E-CUB Module for realistic tests. It has both the properties of self-scaling and self-configuring, meaning that it can scale the HTTP requests to any strength and configure user behaviors to specific traffic characteristics. This E-CUB Module principally provides functional, stateful, emulating real environment and accurate verification to measure the entire characteristic of Layer seven content switches.

Lastly, we have defined already four standard units of layer seven content switch testing and these units have different kinds of indicating criterion. Tests using E-CUB Modules including test environment, procedures as well as results have presented in this thesis. The results show that these E-CUB Module are both accuracy and useful to measure the layer seven content switch under real Internet situations.

About Layer seven content switch testing, we have observed that the testing results are difference dependent on the manufacturer of switches. For the Web site Corporation, there are no confident criterions to purchase layer seven content switches. So public standard testing procedure and results are future important work as well as measure other applications used in particular field. To achieving more diversification testing tool development that we didn't consider in this thesis.

References

[1] Amazon.com, Inc.

http://www.amazon.com, May, 2002.

[2] Andreas Bartelt and Jochen Meyer, "A Practical Guideline to the Implementation of Online Shops" Proceedings of the 18th IEEE Symposium , pp. 348 -353, 1999.

[3] Alberto Leon-Garcia, "Probability and Random Processes for Electrical Engineering", 1994.

[4] Shuang Deng, "Empirical model of WWW document arrivals at access links", Proc. 1996 IEEE Int. Conf. Commun., pp. 1797-1802, Jun. 1996.

[5] W3C. Extensible Markup Language (XML). WWW, http://www.w3.org/XML/, May, 2002.

[6] Addison-Wesley, "TCP/IP Illustrated, Volume 1: The Protocols", ISBN 0-201-63346-9, 1994.

[7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T.Berners-Lee, Hypertext Transfer Protocol -- HTTP/1.1,"RFC 2616, June 1999.

[8] http://www.widearea.co.uk/designer/compress.html, May, 2002.

[9] The PHP Group, www.php.net, May, 2002.

[10] Microsoft Corporation, http://www.microsoft.com, May, 2002.

[11] Alteon WebSystems, Inc. Is Now A Part Of Nortel Networks, http://www.nortelnetworks.com/corporate/acquisitions/alteon/index.html, May, 2002.

[12] Cisco Systems, Inc., http://www.cisco.com, May, 2002.

[13] F5 Networks, Inc., http://www.f5.com, May, 2002.

[14] Ziff Davis Media Inc.,

http://www.etestinglabs.com/benchmarks/webbench/webbench.asp, May, 2002.

[15] Sniffer Technologies is a business unit of Network Associates, Inc. http://www.sniffer.com, May, 2002.

[16] Seki, Y.; Nishikawa, N.; Tsuji, H., "User's Behavior and URL Analysis at EC Sites" IEEE SMC '99 Conference Proceedings, IEEE International Conference on , Volume: 4 , 1999.

[17] Yeung, K.H.; Szeto, C.W., "On the Modeling of WWW Request Arrivals", International Workshops on, pp. 248 -253, 1999.

[18] Kallepalli, C.; Tian, J., "Measuring and Modeling Usage and Reliability for Statistical Web Testing", Software Engineering, IEEE Transactions , Volume: 27, pp. 1023 -1036, 11 , Nov. 200.

[19] Mah, B.A., " An Empirical Model of HTTP Network Traffic", INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE, Volume: 2 , pp. 592 -600 vol.2, 1997.

[20] Leung, K.Y.; Yeung, K.H.," the Design and Implementation of a WWW Traffic Generator", Parallel and Distributed Systems, 2000. Proceedings. Seventh International Conference on, pp. 509 -514, 2000.

[21] Hartman; Peri, Seattle, WA, Bezos; Jeffrey P., Seattle, WA, Kaphan; Shel, Seattle, WA, Spiegel; Joel, Seattle, WA, Amazon .com Inc, "Method and system for placing a purchase order via a communication network", United States Patent, Patent Number:5,690,411, Sept. 28, 1999.

[22] Alteon WebSystems, Inc. WebOS Switch Software, "8.0 Application Guide", Part Number: 050087, Revision B, August 2000.

[23] Paul Barford and Mark Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation" in *ACM SigMetrics' 98*, Madison, WI,

1998, pp. 151–160.