Synopsis of the Ph.D. Thesis

WEB APPLICATION CHARACTERIZATION AND EVALUATION USING A BLACK-BOX MODEL

by

N. HARINI

Department of Computer Science and Engineering

Amrita School of Engineering

Coimbatore – 641 112, Tamil Nadu, India.

MATA AMRITANANDAMAYI MATH

श्रद्धावान् लभते ज्ञानम्

AMRITA
VISHWA VIDYAPEETHAM
UNIVERSITY
Established u/s 3 of UGC Act 1956

March, 2016

1. Introduction

Today Internet has become a platform to perform searches, transact business and connect with friends and colleagues. With growing Internet user population, which is expected to be around 3 billion in 2015 [1] and the growing number services like: Social networking (Facebook, hi5.com), online shopping Websites (Amazon, Flipkart), Blog pages (Blogger, Word Press), online news Websites (CNN, NDTV, IBNLive), Internet search engines (Google, Bing), alternatives to desktop software applications (Google Docs, Slide Rocket)) etc., offered through the Internet, the future will likely see more of business Web presence in the form of Web applications. Today, global enterprises are increasingly dependent on Internet-based applications for e-commerce, supply chain management, human resources, and financial accounting; many individuals consider e-mail, chat, and social-networking and Web access to be indispensable lifelines. *This growing number of users and their dependence upon Internet Services underscores the importance of the ability of the service in terms of its responsiveness, availability, scalability, and ability to handle large loads*.

Internet Services are subjected to huge variations in demand, with bursts coinciding with the times that the service has the most value. Apart from these flash crowds, sites are also subject to denial-of-service (DoS) attacks that can knock a service out of commission. Such attacks are becoming increasingly sophisticated, often being launched simultaneously from thousands of sites across the Internet [4]. DoS attacks have had a major impact on the performance of sites such as Buy.com, Yahoo!, and whitehouse.gov [14, 19]. *Such attacks have to be identified and filtered out without hampering the genuine requests or their progress.*

Popular sites are constantly adding new functionality to stay competitive, a trend which is supported by the increasing ease of Internet Service development using simplified Web scripting languages such as PHP [20] and Java Servlets [21]. It is excessively time-consuming to carefully engineer and fine-tune each new service component to deal with real-world loads. An apt approach to this problem is through a generic service platform that manages load in a manner that is cleanly separated from the service-specific logic; this gives services the ability to participate in load-management decisions. The high complexity and cost of maintaining a hosting platform infrastructure has resulted in a growing trend among businesses and institutions to have their applications hosted on platforms managed by another party. Such a hosting platform is a Web hosting provider (Service Provider -SP); an SP sells space on its servers to Website owners (Service Requestor - SR). The SP provides a full-time, high-bandwidth connection to the Internet, so that visitors to the Website can access the sites easily. SPs enable entrepreneurs and emerging organizations to focus on their business rather than technology. They are typically expected to provide performance guarantees to the hosted applications (such as guarantees on response time or throughput) in return for revenue [12]. *Service Level Agreements* (SLAs) *between the SP and the SR incorporate the scope of the Web Service, the performance guarantees to be offered, their quality, and modes of their quantification.*

A survey from 2006 revealed that 62% Web users were willing to wait only 6 seconds or less for a single page load before they leave the Web site [22]. A more recent research indicated that this performance expectation has become more demanding as 83% Web users expected a Web page to load in 3 seconds or less [23]. In addition, this research also found that 79% of online shoppers who visited an underperforming Web site were not likely to buy from that site [10]. Obviously, performance guarantees for Web applications have become business critical. Performance metrics of an Internet Service are influenced by many factors. Several papers have investigated various configurable parameters, and the way in which they affect the performance of Web-based software systems [8, 13, 5]. Statistical methods and hypothesis tests [9] are being used to retrieve factors influencing the performance. The performance metrics can be predicted at early stages with the help of a properly designed performance model. *All these point to a clear need for the crafting and use of a performance model.*

Depending on the resource requirements of the applications and the strictness of the performance or resource guarantees they require, a platform may employ a dedicated or a shared model for hosting them. Designing a hosting platform is made challenging by the following characteristics of Internet applications and their workloads (Complex multi-tier software architecture and Internet workload characteristics like: multi-time-scale variations, extreme overloads, session based, and multiple session classes) [6, 7, 11]. To meet its goal of maximizing revenue given the above challenges, a hosting platform needs to carefully multiplex its resources among the hosted applications. *The performance model should have* *the flexibility to accommodate the variety in software architectures and Internet workload characteristics*.

Performance modeling is an important part of the research area of Web servers. Without a correct model of a Web server it is difficult to give an accurate prediction of performance metrics. This is the basis of Web server capacity planning, where models are used to predict performance in different settings [17]. An important component of Web performance is the architectural design of the Web server software. The biggest challenge in designing an efficient Web server lies in efficiently supporting thousands of connections concurrently. Different Web server architectures therefore vary mainly in how they implement concurrency [17]. The earliest Web servers were all process based [12, 17], i.e., they maintain a separate Operating System (OS) process for each request. The process driven approach simplifies code development, since all responsibility for handling concurrency is relayed to the OS. However, a major drawback of processes is that they are heavy-weight, and creating and deleting them and context-switching between them is expensive. An alternative approach is to use threads instead of processes [12]. Threads are more light-weight since they allow sharing of the same address space. While sharing the address space reduces the context switching cost, it has the disadvantage that a poorly programmed thread can crash the entire server. In the above approaches the Web server software is implemented in user space. Some Web servers such as Tux [3] are entirely implemented in kernel space thereby avoiding overheads associated with systems calls and copying of data between user and kernel space. The downside of the in-kernel implementation is that it is extremely sensitive to programming errors, since a crash of the Web server code translates to a crash of the kernel and therefore the whole machine. *The basic need here is to provide concurrency whether the service is process-based or thread-based.*

2. Motivation

The above overview of Internet Services leads us to the following inferences which constitute the motivation for the present work:

1. The key attributes expected of an Internet Service are its responsiveness, availability, scalability, and ability to handle large loads.
2. DoS attacks – despite their sophistication – have to be identified and filtered out without hampering the genuine requests or their progress.

3. SLAs between the SP and the SR incorporate the scope of the Web Service, the performance guarantees to be offered, their quality, and modes of their quantification. The dynamic nature of present day services should be reflected in SLA formulation and implementation.
4. The course of exposed resources along with the level of privacy, value of the information they hold, and the increase in their usage, has led to the escalation in the number of security threats and violation attempts that existing systems do not appear robust enough to address peak loads.
5. Whether the service is process-based or thread-based, the basic purpose is to provide concurrency.
6. Perhaps one of the most compelling problems of the modern Internet is the lack of a comprehensive model that can aid in performance evaluation, unifying approaches to deal with service concurrency, security, and availability while preserving ease of implementation.
7. Any model crafted and used to analyse an Internet Service should have the flexibility to accommodate the variety in software architectures and Internet workload characteristics.

3. Objective

This research work focuses on providing a comprehensive black-box model that can be used to evaluate the performance of a variety of Web applications. The research objectives are stated as follows:

1. To present a coherent and inclusive way to model Web applications with details and abstractions which are appropriate for SPs in terms of capacity planning and SRs in terms quality assessment.
2. To demonstrate the utility of the proposed model for the purpose of performance prediction and capacity planning with varying workload and service differentiation strategies.
3. To demonstrate the ability of the model to capture the category based service differentiation, improved service availability, and maximize user satisfaction for a real time Web application.

4. To introduce the problems that exist in the currently used authentication schemes and workloads that particularly affect the availability of the Internet Services and subsequently derives an appropriate solution for the same.

The approach leverages models of a service's application-level performance in evaluating the Quality of Service (QoS) of an Internet Service. The proposed model represents the relevant aspects of the system in a simplified and ideally comprehensible manner. The model presented and discussed in this work does not address network characteristics. It allows SPs to query and understand what performance would be like if a particular action was taken. For example, the model answers the following queries:

- What happens to performance if a new functionality is added to the service under the current configuration?
- What if the allocation policy for resource assignment was changed?
- What configurations yield high performance?

The evaluation is based on the implementation, assessment and analysis (based on collected logs and statistical techniques).

The logs collected by the model allow SR to study the behavior patterns of online customers on their site and improve business strategies. It also serves as a guideline for SRs to understand the measures of the service quality that have been offered by the hosting platform services. It aids SP to evaluate the measures of commitments in service quality that could be agreed. These derived measures from the model could enable them in formalizing the SLA. Further the model shields application programmers from many of the details of scheduling and resource management.

4. Methodology of the research work

Several approaches like measurement, simulation and analytical, and black-box modeling are used in Internet Service performance evaluation [6]. In this work, the relevant aspects of the system to be modeled are analyzed first; it is followed by study of Web application characteristics. After having gained insight into the workings of Internet Services, a preliminary working model (**'BBAM-I' –** Black Box Analytical Model Version I) was implemented to understand the nuances of the implementation and make a case for a sophisticated one. The behavior of this prototype in terms of offering service availability was carefully and systematically examined for various workloads.

The preliminary model was enhanced (**‘BBAM-II’–** Black Box Analytical Model Version II) to improve service availability with the addition of a **QUEUE** (‘*Token-Basket*’) data-structure over and above the server architecture. A detailed analysis on the effect of this added data structure on concurrency (number of simultaneous users accessing the service), with varied number of tokens was conducted. The results of the preliminary working model were compared with the results of enhanced version to understand the contribution of the enhanced model in terms of improved service availability. The capability of the enhanced version to serve additional user’s requests with and without sacrifice in service quality was investigated. With detailed simulation it has been demonstrated that the enhanced version has been potentially able to provide improved service availability without performance degradation by utilizing the gap between execution time and response time effectively; any further improvement was possible only with an acceptable reduction in the QoS.

The next step was to explore the possibility of deriving limits for various quality factors that determine the performance of Web applications. Experimentation with different parameter sets, confirmed that a basis to quantify the guarantees to be made for various parameters in SLA can be derived using the prototypes.

Subsequently, the model was tested for its suitability to accommodate real time Internet Services, with e-commerce as a representative Web application for study. It was verified that the prototype is versatile enough to accommodate new classes of e-customers, logs collected were sufficient enough to evaluate performance metrics, and was able to do informed scheduling and resource management decisions, such as reordering, filtering, or aggregation of requests as well as several other characteristic operating scenarios.

Two aspects that could affect the service availability were identified at the end of the study of the modeling the E-Commerce application:

1. Possible delay in system throughput, and increased response time at the home page due to malicious requests in the workload.
2. Possible service unavailability (time out of sessions/delayed response time) at the payment stage because of the dependency on an external mobile network service (Short Message Service – One Time Password --SMS-OTP) [18] for authentication.

The model supports integration of a pre-filtering algorithm to condition the workload (removing malicious requests) before its acceptance into the system to avoid (1) and uses a

Quick Response-Code (QR-Code) based OTP scheme capable of overcoming the delay caused by the prevailing SMS-OTP scheme.

5. Conclusions

The work describes the design, architecture, and a black box implementation model of service architecture suitable for evaluating the performance of Web applications with focus on enhancing the service availability. This platform provides efficient request scheduling and resource-control mechanisms, including '*Tokens-Basket*' sizing and adaptive load shedding. The model can serve as a guideline for Internet application developers and SPs of hosting platforms to understand the performance of a given application design and aid them in formulating commitments/expectations in SLAs. The work also presents a coherent and complete way to model Web applications with details and abstraction that are appropriate for SPs in terms of capacity planning and SR as well as end users in terms quality assessment. An e-commerce application has been used to illustrate the effectiveness of the model. A systematic performance evaluation has been carried out with varied system capacities, bounds on '*Tokens-Basket*', workload characterization, and user behavior patterns. The work also introduces the problems that exist in the currently popular SMS-OTP authentication schemes while making on-line payments; subsequently an appropriate authentication scheme that improves Internet Service availability – particularly at times of peak loads on server – has been evolved.

6. Thesis organization

Employing black box simulation to model Web applications and study their performance, the thesis focuses on analyzing the reasons for service unavailability of Internet Services and proposes ways of improving the same.

Internet Services have become critical both for driving large businesses as well as for personal productivity. Chapter 1 is a brief introduction to them; it highlights the fact that the growing dependence upon Internet Services underscores the importance of their availability, scalability, and ability to handle peak loads as these bursts coincide with the time the service has the most value.

In Chapter 2 the challenges faced by the current Internet Services, and the literature related to modeling Web applications are presented. The aspects analyzed include workload characteristics, user behavior and representation of Web applications. This laid groundwork for developing a preliminary model for performance evaluation of Web applications. A Finite State Machine (FSM) has been used to simulate processing in a server. A sequential set of FSM states were taken in this scheme to accommodate the service architecture. Behaviours of users of Web applications have been modeled using User Behavior Model Graphs (UBMG) [2] which represents the probabilities of transition from one state of FSM to another.

With the service architecture '*thread-per-connection*' in use by APACHE as basis, Chapter 3 develops a generic comprehensive model (**'BBAM-I'**) to represent Web applications. The model is structured as a sequence of stages with equal execution times. At every time slot all active client requests as well as new requests are queued conforming to a *Priority List* (PL) representing the desired service differentiation policy. The threads are allocated to the requests in the queue; left over requests – if any– are denied service (dropped requests). User requests under normal conditions conform to a Gaussian distribution pattern with specified $\{\mu, \pm^2\}$ set. Further they progress conforming to the UBMG.

Extensive simulation over the full meaningful $\{\mu, \pm^2\}$ range was carried out. With each set the simulation duration was long enough to ensure that each possible request value occurred sufficient number of times. The simulation study effectively brought out the following:

- The mean arrival rate should be smaller than the system capacity representing the number of threads for the service to be effective.
- The maximum $\pm^2$ value decides the difference between the above two. For a given value of the difference the drop rate and the deficiency in system utilization increase with $\pm^2$.

The simulation was continued with an impulse load superposed on the regular stationary random load – representing (possibly) the peak load at its extreme. It confirmed that the system accommodates peak loads if the mean arrival rate is sufficiently smaller than the system capacity; the price paid as an increase in response time.

The simplest sequential set of stages considered here directly represents a pipelined process at the highest level of abstraction. Each stage can be further split and represented as sub-stages of execution at the next finer level of granularity. This can be continued down to the finest level of granularity desired.

The model does not call for any detailed decision making for thread allocation. There can be time slots when threads are free and those when requests are dropped though both do not happen simultaneously. The simulation results with the model showed its suitability for services where there is not much of deviation in average workload and those like video streaming (i.e., it is suitable for Internet Services that are yet to become popular and those which require high quality of service guarantees for accepted connections rather than serving many concurrent users). The service becomes unavailable as soon as all system capacity exhausts. Additional increase in average load is managed by rejecting the requests.

The generic model **'BBAM-I'** can represent any architecture with any combination of sequential /parallel stages of execution. After showing this Chapter 4 presents an enhanced version of the model (**'BBAM-II'**) that uses a variation of '*thread-per-request*' model used by servers. In **'BBAM-II'** client requests can be executed concurrently until the number of simultaneous requests exceeds the number of threads in the pool. At this point, additional requests (a decided number in advance) are queued in a '*Tokens-Basket*' until a free thread becomes available. This is used to limit idling of threads and reduce denial of requests (provide enhanced service availability). Defining admission policies on various factors like information generated from user behavior models can contribute to devising cost effective infrastructures. To facilitate this, concept of priority list (PL) has been introduced here. The requests are assigned tokens and serviced conforming to a predetermined service differentiation strategy which is decided by the algorithm/procedure defined by the PL. Thread assignment is done afresh for each request in each time slot for each stage separately. Complexity in assigning threads here is more than that of **'BBAM-I'**. The number of tokens in the '*Tokens-Basket*' is chosen such that it is greater than the number of threads in the thread pool. If used correctly, bounded '*Tokens-Basket*' is an effective means to represent service availability in an Internet Service. The important aspects of the enhanced operational model are:

1. Inclusion of a '*Tokens-Basket*' structure over and above the existing server architecture enables the server to accommodate requests (enhance server availability) that would have been otherwise rejected as in the case of **'BBAM-I'.**
2. The introduction of '*Tokens-Basket*' achieves two purpose:

- Serving additional requests without degraded quality as long as the execution time is lower than the expected response time. (Free threads available in thread pool at different timeslots can exploit this to serve additional requests without sacrifice of response time).
- Serving additional requests with a slight delay in response time. The acceptable limit set by the SR on response time can be used to decide on how many of such additions can be accepted.

3. The PL Rule that is incorporated to schedule the accepted requests enables the SR realize the mission of offering differentiated services to their visitors/customers.
4. Making the '*Tokens-Basket*' explicit allows applications to make informed scheduling and resource management decisions, such as reordering, filtering, or aggregation of requests.
5. The increase in number of tokens poses a corresponding delay in service time. This is obvious from the fact that it is possible for a set of requests to be kept waiting with a token assigned to them due to unavailability of threads in that timeslot. This otherwise means that the introduction of the '*Tokens-Basket*' has reduced the service rate (computed as the ratio of the number of requests completed in that time slot to the total number of requests residing in the system) at each time slot whenever the number of requests in the system exceeds the system capacity.
6. Deployment is made easy, shielding application programmers from many of the details of scheduling and resource management.
7. The model supports Internet Services to analyze the request stream and to adapt behavior to changing load conditions. For example, the system should be able to prioritize and filter requests to support degraded service under heavy load.

Chapter 5 focuses on the utility of the model in terms of support for strong profiling. It has been shown that the collected logs and derived performance metrics allow detailed assessment of the business performance (like studying behaviour patterns of online customers on their site and improving business strategies) for SR and capacity planning (like indicating resource bottlenecks and under utilization) for SP; it also aids them to find if and when SLA re-negotiation is required.

In Chapter 6, '**BBAM-I**' and '**BBAM-II**' are explored for their suitability to accommodate the real time requirements of Web applications. The requirements of an e-commerce application were used to evaluate user satisfaction in terms of service availability and response time. A detailed experimentation with different workload mix and varied request scheduling policies carried out clearly brought out the flexibility of the model '**BBAM-II**' to accommodate service specific requirements. It also helped in identifying two reasons other than request rejection that may affect the availability of the service namely malicious requests accepted in the '*Tokens-Basket*' and dependency on external SP's network for request completion.

Chapter 7 explores ways of improving service availability without degrading QoS. A filter component has been added to the architecture along with an authentication scheme that minimizes the delay in the response time by reducing dependency on Mobile Network Operator for SMS-OTP. Incorporation of these leads to better service availability particularly at times of overloads on the server.

7. References

[1] N. Singh, H. S. Alhorr and B. P. Bartikowski, "Global e-commerce: A Portal Bridging the world markets", *Journal of Electronic Commerce Research: Special Issue: Global B-Commerce*, vol. 11, no. 1, pp. 1-5, 2010.

[2] D. Wang and K. Trivedi, "Modeling user-perceived reliability based on user behavior graphs", *International Journal of Reliability, Quality & Safety Engineering*, vol. 16, no. 4, pp. 303-330, 2009.

[3] Bhuvan Urgaonkar , Giovanni Pacifici , Prashant Shenoy , Mike Spreitzer , Asser Tantawi, "*Analytic modeling of multitier Internet applications*", ACM Transactions on the Web (TWEB), vol.1, no. 1, 2007.

[4] Xinyi Huang, Yang Xiang, Member, IEEE, Ashley Chonka, Jianying Zhou, and Robert H. Deng, "A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems", *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, August 2011.

[5] M. Sopitkamol and D.A. Menasce, "A Method for Evaluating the Impact of Software Configuration Parameters on E-commerce Sites", in *Proceedings of ACM*

5th International Workshop on Software and Performance, Palma, Illes Balears, Spain, pp. 53-64, 2005.

[6] D. Villela, P. Pradhan, and D. Rubenstein, "Provisioning Servers in the Application tier for e-commerce Systems", in *Proceedings of International Workshop on Quality of Service* (IWQOS 2004), pp. 57 - 66, 2004.

[7] A. Robertsson, B. Wittenmark, M. Kihl, and M. Andersson, "Admission control for Web server systems - design and experimental evaluation", in *Proceedings of 43rd IEEE Conference on Decision and Control*, vol. 1, pp. 531– 536, 2004.

[8] A. Iyengar, E. Nahum, A. Shaikh, and R. Tewari, "Improving Web Site Performance", in *Handbook of Internet Computing*, M.P. Singh, Editor, Chapman & Hall/ CRC Press, 2004.

[9] KyoungSoo Park and Vivek S. Pai, "Connection conditioning: architecture-independent support for simple, robust servers", *3rd Symposium on Networked Systems Design & Implementation,* USENIX Association, 2006.

[10] Zakaria Al-Qudah, Michael Rabinovich, Mark Allmanf, "Web Timeouts and Their Implications", *Passive and Active Measurement*, vol. 6032, pp 211-221, 2010.

[11] A. Williams, M. Arlitt, C. Williamson, and K. Barker, "Web Workload Characterization: Ten Years Later", *Lecture Notes in Computer Science,* Springer, 2005.

[12] Jordi Guitart, Jordi Torres and Eduard Ayguade , "A survey on performance management for Internet applications", *Concurrency and Computation: Practice and Experience*, vol. 22, no. 1, pp. 68–106, 2010.

[13] Kriticos et al, "A Survey on Service Quality Description", *ACM Computing Surveys*, vol. 46, no. 1, article no. 1, 2013.

[14] D. MacPherson and C. Labovitz. "Worldwide Infrastructure Security Report", *ACM Computing Surveys*, Technical Reports, Arbor Networks, vol. 2, 2006.

[15] Jianli Pan, Subharthi Paul, and Raj Jain, "A Survey of the Research on Future Internet Architectures", *IEEE Communication magazine*, pp. 26-36, 2011.

[16] Kishor S. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Science Applications" John Wiley and Sons Ltd., 2002.

[17] G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi, "Queueing networks and

Markov chains: Modeling and Performance Evaluation with Computer Science Applications", Wiley-Interscience New York, 2005.

[18] Google Inc. Verifying your account via SMS or Voice Call.
http://support.google.com/mail/bin/answer.py?hl=en&answer=114129

[19] R. Lemos. Web worm targets White House,
CNET *News.com*. http://news.com.com /2100-1001-270272.html.

[20] The PHP Group. PHP: Hypertext Preprocessor.
http://www.php.net

[21] Sun Microsystems Inc. Java Servlet API.
http://java.sun.com/products/servlet/ index.html

[22] "Retail Web site performance: Consumer reaction to a poor online shopping experience". http://www.akamai.com/4seconds.

[23] Forrester Research "e-commerce Web site performance today: An updated look at consumer reaction to a poor online shopping experience". http://www.akamai.com/2seconds.

[24] The Apache software foundation. The Apache Web server.
http://httpd.apache.org.

8. Publications based on the research work

International Journals:

1. N Harini and T R Padmanabhan, "A Secured-Concurrent Available Architecture for Improving Performance of Web Services", *International Journal of Communications in Computer and Information Science*, vol. 292, no. 1, pp. 621-631, Springer, 2012. **SCOPUS**
2. N Harini, Dr. T.R Padmanabhan , "Admission Control and Request scheduling for secured-concurrent-available architecture", *International Journal of Computer Applications*, vol. 63, no. 6, pp. 24-30, February 2013. **Impact Factor 0.8**
3. N Harini, Dr. T.R Padmanabhan, "2CAuth: A New Two Factor Authentication Scheme Using QR-Code", *International Journal of Engineering and Technology*, vol. 5, no.2, pp. 1087-1094, 2013. **SCOPUS ---- 8 CITATIONS**

4. N Harini, Dr. T.R Padmanabhan, "3CAuth: A Novel Authentication Scheme Using QR-Code", *International Journal of Network Security*, 2015. **SCOPUS** (**Accepted for Publication**)
5. N Harini, Dr. T.R Padmanabhan, "SLACalc: A framework to prove IT's value to business", *International Journal of Modeling and Simulation*. **SCOPUS.** (Communicated)
6. N Harini, Dr. T.R Padmanabhan, "A Blackbox Simulation model for performance evaluation of Web applications", *International Journal of Computer Science and Applications*, **SCOPUS.** (Communicated)

International Conferences:

7. Aishwarya Malepati, Sai Vaishnavi.K, Sukanya R, Aiswariya.C, M Senthil Kumar, N Harini, "An Improved Authentication Scheme Using Color QR Codes" ,*International Conference on emerging technologies*, pp. 65-71, Elsevier, 2014.
8. Sikha Radhakrishnan and N Harini ,"Performance Analysis of Multi-Factor Authentication", *International Conference on Advances in Computer Communications and information science*, pp 282-288, Elsevier, 2014.

Book Chapter:

9. N Harini, CK Shyamala, TR Padmanabhan, "Securing Cloud Environment", *Cyber Security, Cyber Crime and Cyber Forensics: Applications and Perspectives*, pp. 115-123, IGI Global publishers, 2011.

Book:

10. CK Shyamala, N Harini, TR Padmanabhan, "Cryptography and Security", Wiley India Pvt Ltd, 2011.

Title of the Thesis	**:** WEB APPLICATION CHARACTERIZATION AND EVALUATION USING A BLACK-BOX MODEL
Name of the Student	**:** N. Harini
Register Number	**:** 0130601111
Year of Registration	**:** 2007
Name of the Department	**:** Computer Science and Engineering
Thesis Advisor	**:** Dr. T.R. Padmanabhan, Thesis Advisor, Professor Emeritus, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore – 641 112, India.
Name of the School	**:** Amrita School of Engineering, Ettimadai.
Name of the University	**:** Amrita Vishwa Vidyapeetham University, Amritanagar Post, Ettimadai, Coimbatore – 641 112.

CONTENTS