

العنوان:	المجدول ذو خاصية أمان وجودة الخدمة في الأنظمة الموزعة متعددة الخوادم في الشبكات محولة الطرود
المؤلف الرئيسي:	عواودة، سراء محمد
مؤلفين آخرين:	الجراح، محمد(مشرف)
التاريخ الميلادي:	2016
موقع:	إربد
الصفحات:	1 - 65
رقم MD:	871190
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	جامعة اليرموك
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	أمن الشبكات، جدولة المعلومات، الأنظمة متعددة الخوادم
رابط:	http://search.mandumah.com/Record/871190



SECURITY &QoS AWARENESS SCHEDULER FOR DISTRIBUTED MULTI-
SERVER SYSTEM IN PACKET SWITCHED NETWORKS

A Thesis Submitted to the Faculty of the Graduate College in Partial Fulfillment of
the Requirements for the Master Degree in Computer Engineering/Industrial
Automation

By:

Saraa M. Awawdeh

Advisor

Mohammad Al-Jarrah, Ph.D.

Yarmouk University
Irbid, Jordan
Dec. 2016


WE, THE UNDERSIGNED MEMBERS OF THE COMMITTEE,
HAVE APPROVED THIS THESIS

**SECURITY & QoS AWARENESS SCHEDULER FOR DISTRIBUTED
MULTI-SERVER SYSTEM IN PACKET SWITCHED NETWORKS**


By

Saraa M Awawdeh

COMMITTEE MEMBERS

Dr. Mohammad Al-Jarrah  Chairman
Associate Professor, Computer Engineering, Yarmouk University

Dr. Ahmad S. Musa  External Member
Associate Professor, Communication Engineering, Yarmouk University

Dr. Osameh M. Al-Kofahi  Member
Assistant Professor, Computer Engineering, Yarmouk University

December 2016

Declaration

I am Saraa Awawdeh, I am hereby declare that this thesis entitled "SECURITY &QoS AWARENESS SCHEDULER FOR DISTRIBUTED MULTI-SERVER SYSTEM IN PACKET SWITCHED NETWORKS", submitted for "Department of Computer Engineering at Hijjawi Faculty for Engineering Technology" is my original work and it has not been presented for any degree in any other university, and that any additional sources of information have been properly cited.

Name: Saraa M. Awawdeh.

ID Number: 2013970013

Date: Dec 15th 2016

Signature:

ACKNOWLEDGEMENTS

My first and many thanks would be for God, whose blesses were the key behind my success in whole life. I would like to thank my research's supervisor for his guidance and support. Dr. Mohammad Al-Jarrah and the cooperated committee members, thank you all. Dad, mom, brothers, sisters and whole family, big thanks for your prayers, help and encouragement. My dear friends and colleagues, thanks for your continuous support. My profound gratitude for my best sister-in-law, Fatimah, thanks for your nice heart and support. My sincere thanks also go to my supporter uncle Ahmad Awawdeh. Finally, I would like to dedicate this thesis to my kids, Salma, Renad, Hashem and the man whose love, support, and patience allows me to complete this thesis. To my husband, Ma'en, thanks for being in my life.

TABLE OF CONTENTS

DECLARATION	II
ACKNOWLEDGMENT	III
LIST OF FIGURES	VI
LIST OF TABLES	VIII
ABSTRACT	IX
CHAPTER	
1. INTRODUCTION	1
1.1 General Overview.....	1
1.2 Research Motivation.....	2
1.3 Research Contributions	3
1.4 Thesis Organization.....	4
2. BACKGROUND AND LITERATURE REVIEW	6
2.1 Client/Server Model	6
2.2 Scheduling in Real-time Systems.....	8
2.3 Scheduling Algorithms for Cloud Servers	10
2.4 Security Services for Cloud Servers	12
3. PROPOSED SYSTEM DESIGN & METHODOLOGY	16
3.1 Introduction	16
3.2 System Model using Multi-Agent System	17
3.2.1 Client Agent	18

3.2.2 Scheduler Agent	19
3.2.3 Server Agent.....	20
3.2.4 Resource Estimator Agent.....	21
3.2.5 Controller Agent.....	21
3.3 Server's Security Services	22
3.4 Single vs. Multi-Server Modes.....	24
3.4.1 Single Server Mode	24
3.4.2 Multi-Server Mode	25
3.5 Research Methodology	26
3.5.1 Single Server Mode	28
3.5.2 Multi-Server Mode	29
4. PROPOSED SYSTEM IMPLEMENTATION	33
5. SYSTEM SIMULATION & RESULTS	42
5.1 Introduction	42
5.2 System Parameters	43
5.3 The QoS Metrics at the Scheduler Sub-Agent	44
5.4 System's Performance at the Server Sub-Agent.....	51
6. CONCLUSION & FUTURE WORK.....	58
6.1 Conclusion.....	58
6.2 Future Work	60
REFERENCES	61

LIST OF FIGURES

Figure 3.1: Multi-agent system process	17
Figure 3.2: Network topology.....	19
Figure 3.3: Agent-based communication design	20
Figure 3.4: Single server mode	25
Figure 3.5: Multi server mode	26
Figure 4.1: System parameters.....	34
Figure 4.2: QoS system results	35
Figure 4.3: Request's class.....	36
Figure 4.4: Generating successive uniform hits.....	37
Figure 4.5: Creating successive uniform hits.....	37
Figure 4.6: Generation of real-time requests	38
Figure 4.7: FCFS queuing algorithm	39
Figure 4.8: EDF queuing algorithm	40
Figure 4.9: Server's class	40
Figure 4.10: Generating servers' permutations	41
Figure 5.1: Miss ratio metric at the scheduler/ single-server mode.....	45
Figure 5.2: Average delay metric at the scheduler/ single-server mode.....	46
Figure 5.3: Miss ratio at the scheduler/ multi-server (2-permutations)	47
Figure 5.4: Miss ratio at the scheduler/ multi-server (3-permutations)	48
Figure 5.5: Average delay at the scheduler/ multi-server (2-permutations)	49
Figure 5.6: Average delay at the scheduler/ multi-server (3-permutations)	49
Figure 5.7: Miss ratio at the scheduler.....	50
Figure 5.8: Average delay at the scheduler.....	51

Figure 5.9: Average server utilization/ single-server mode.....	52
Figure 5.10: Average server utilization/ multi-server mode (2-Permutations).....	53
Figure 5.11: Average server utilization/ multi-server mode (3-Permutations).....	54
Figure 5.12: Average server utilization	55
Figure 5.13: Effect of server's CPU power on miss rate	56
Figure 5.14: Effect of server's CPU power on average delay	57

LIST OF TABLES

Table 3.1: Cryptographic security algorithms	23
--	----

SECURITY & QoS AWARENESS SCHEDULER FOR DISTRIBUTED MULTI-SERVER SYSTEM IN PACKET SWITCHED NETWORKS

ABSTRACT

Saraa M. Awawdeh, Security & QoS Awareness Scheduler for Distributed Multi-Server System in Packet Switched Networks, Master of Science in Computer Engineering/Industrial Automation, Department of Computer Engineering, Yarmouk University, 2016, (Advisor: Dr. Mohammad Al-Jarrah)

The huge and rapid revolution in the telecommunication field makes an evolution in the type of services provided by the network's technology. Early, the network was providing best effort services to its customers, where delivery was the only guarantee provided by such network. Nowadays, networks become commercial-based entities, where different classes of services with different requirements should be guaranteed for different types of clients.

In order to serve clients' requests, the network technology adopts the client/server model. Such model is a distributed structure that provides a communication scheme between two main entities: (1) Client: the one who requests the service; (2) Server: the one who provide the service. Besides providing quality of service (QoS) to its clients, network technologies should be capable of providing them with the required security requirements, and thus protecting the traffic streams from being hacked by different levels of network-security threats.

In this thesis, we proposed a security-awareness scheduler for distributed multi-server system. According to the distributed multi-server systems, each server node has its own security guarantees, memory resources, and power capabilities. The

scheduler serves the clients requests by choosing the appropriate server(s), such that both QoS and security requirements are guaranteed.

The scheduler implements an online monitoring mechanism for the distributed system through a feed-back message passing technique. Such mechanism is the key behind the scheduling algorithm, where estimation for the server's resources is based on such feedback. Accordingly, the scheduler selects the appropriate server that guarantees the real-time client's requests. As a result, our proposed algorithm will protect the system from security threats and prevent the whole system from being congested by those heavy traffic flows. In this thesis, extensive simulations were carried out for two systems: single-server and multi-server systems. The results show that the EDF-based multi-server system outperforms the FCFS based multi-server in terms of miss-ratio, average delay, and utilization. The results also show that the multi-server system outperforms the single-server for the same performance metrics, and thus it's more efficient in protecting the network from being congested by heavy traffic load.

CHAPTER I

INTRODUCTION

1.1 General Overview

The type of services provided by the telecommunication technologies should be compatible with the huge and rapid revolution in the IT field. Such revolution makes a transition from those non-real time services (best effort) to real-time services with complex and strict quality-of-service (QoS) and security guarantees [1]. The transition into internet-of-things (IOT) increases the number of clients that are connecting to the internet exponentially [2], and thus higher computation demands are needed. In order to accommodate such requested services, powerful and efficient servers were deployed by service providers.

A server is a machine that accepts requests from a set of machines (clients) and serves them within the required QoS guarantees in a well-defined client/server model such as mail server model, web server model, file server model, database server model, and print server model [3]. According to the process of resource provisioning, two main servers are implemented: dedicated servers and cloud servers. Dedicated servers are those that are fully controlled by an organization without sharing them with any other one. On the other hand, cloud servers are shared among different clients, where each client rents its needs of resources from the server system [2]. Regardless the type of the implemented server, different performance metrics were

defined such as average load (requests/sec), error rates, average response time, peak response time, uptime, CPU utilization, memory utilization, and power consumption[4].

In order to guarantee the client's requests within the QoS requirements, different scheduling algorithms were implemented at the server side [5]. Such scheduling algorithms assign the server's resources to a set of tasks, such that the best schedulability test of the server's resources is achieved and the tasks are served within their timing and QoS constraints. According to the IOT, the servers are susceptible to different types of security threats. Such threats degraded the type of service provided by the server's system[6]. They also may lead into a catastrophe, especially for those real-time security-critical applications. Servers implement different security services on the requested data as a layer of defense against such security threats.

1.2 Research Motivation

In this research, a security-awareness scheduler for distributed multi-server system was proposed to guarantee both QoS and security requirements for a set of real-time clients' flows in a packet switched network. The scheduler unit implements an online monitoring mechanism for the distributed system through a feed-back message passing technique. Such mechanism will be the key behind the scheduling algorithm, where estimation for the server's resources will be based on such feedback. Since each server node has its own security services, our proposed scheduler assigns the

tasks such that the security requirements are guaranteed. The whole process was modeled based on agent-based mechanism, where the system was decomposed into a set of cooperative sub-agents and controlled by a well-defined set of rules (protocol). According to the number of servers that are used to serve the client's request, two main modes were defined for the distributed server system: single server mode and multi-server mode. As a result, both security and QoS requirements are guaranteed for different data flows and the overall performance of the system is protected from being congested by heavy traffic load.

1.3 Research Contributions

While passing through the different phases of our proposed system, the following contributions were achieved:

- 1) The proposed system was modeled and designed using agent-based methodology, where the overall system was decomposed into a set of cooperative agents controlled by a well-defined protocol.
- 2) Our security-aware scheduling algorithm integrates the scheduling unit that is based on the earliest deadline first (EDF) algorithm with a security awareness unit to provide both QoS and security requirements to a set of real-time data streams.

- 3) The proposed algorithm keeps monitoring the overall system through a feedback mechanism implemented by the resource estimation sub-agents, and thus protecting the entire system from being congested by heavy traffic load.
- 4) The system was modeled using two main modes: (1) single server mode, where one server can be used to accommodate the client's request; (2) multi-server mode, where more than one server can cooperate to provide both security and QoS requirement of the system, and thus more reliability, less delay, and less miss rate.

1.4 Thesis Organization

This thesis is organized in a structure of six chapters including the introduction chapter. In this section, we present the overall outline of the proposed security-awareness scheduler for distributed multi-server systems.

An exhaustive literature review covering the recent related work to our proposed system was presented in chapter two. It provides a description of the client/server model: properties and types, scheduling algorithms for cloud servers, and security services for those cloud servers.

The process of designing our proposed system according to the real-time agent-based methodology was presented in chapter three. The modeling process of each sub-agent was extensively viewed in this chapter. The design process of two

main server modes was presented in this chapter: single-server and multi-server modes. The security service model for each mode was also designed in this chapter. Finally, the research methodology that describes the communication protocol for the proposed multi-agent system was provided in this chapter for the two server's modes: single and multi-server. In chapter four, we provide the process of employing the .Net platform and its real-time capabilities in implementing each sub-agent of the agent based system. It also provides the real-time implementation of the communication protocol that governs the overall functionalities of such agent-based model.

Chapter five provides the overall extensive simulations and system results. According to this chapter, system parameters were initialized to carry out the required simulations to show the performance of our proposed systems. The performance of the proposed system was viewed in this chapter from different agent's perspectives. This thesis ends with chapter six that provides the main research conclusions. It also provides some suggested ideas that may direct the researchers in the future for some novel ideas.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

2.1 Client/Server Model

Nowadays, the world is internet connected, where a huge number of clients from different classes are requesting for different types of services. Client/server model was the adopted design to accommodate such requests, where a network-based protocol governs the communication over a distributed structure of client (service requester) and server (service provider) nodes [7].

The earliest client/server models adopt the dedicated servers in their designs, where an organization has a full control over the providers (servers) without any type of sharing. Such servers are used for those companies that need high computation and security requirements. Dedicated servers suffer from different limitations such as high power consumption, limited memory resources, frequent system failures (unreliable) with high repairing cost, low utilization, and no remote access [2].

In order to accommodate such limitations, virtual private server (VPS) was proposed. In such server, the actual physical server will be partitioned virtually to appear as multi-server system. Such partitioning will eliminate the process of halting the dedicated server completely, where client need not to pay for a full server's power (system resources are distributed over clients) [2]. Accordingly, the VPS solves only

those limitations of the dedicated server that are related to power consumption and maintenance. Other limitations were solved by proposing the cloud server.

Cloud computing depends on distributing the processing capabilities, resources, applications, and systems among different computing entities. As a result, the local computation power of a single entity could be virtually expanded into infinite processing power (internet power) [8]. Accordingly, dedicated servers will not be sufficient for such design.

To handle such model, cloud servers were proposed, where each client rents its needs of resources from the server system, and thus resources are dynamically scaled either ways (up and down) with efficient and easy backup for the data. The first cloud server was proposed by the European Organization for Nuclear Research (CERN) in 1991 and was called CERN httpd that is running on NEXTSTEP [9]. It was proposed to provide a communication scheme between a set of researchers' computation stations using hypertext system. With more and more clients connecting to the internet and requesting for services, the power capabilities were enhanced to accommodate such request. The W3C server proposed in 1994 was the key behind enhancing the cloud computing, where an independent platform and portable server was proposed to be able for being forward and backward compatibility [10].

Nowadays, cloud servers provide three main services for the clients: storage, processing, and software as a service (SaaS). Accordingly, research emphasizes on the performance metrics of the servers that make them capable of handling the huge

amount of incoming requests such as average load (requests/sec), error rates, average response time, peak response time, uptime, CPU utilization, memory utilization, and power consumption[11]. As a result, high computation and powerful models of servers were proposed to offer the previous services to the clients such as application server, communication server, database server, fax server, mail server, web server, computing server, proxy server, game server, etc.

2.2 Scheduling in Real-Time Systems

A real-time system is any information processing system which has to respond to externally generated input stimuli within a finite and specified period. A successful response depends on the logical result within the time it was delivered, and failure to respond is as bad as the wrong response. The real-time systems are classified as hard and soft real-time systems [36]. A hard RT system is an overrun in response time leads to potential loss of life and/or big financial damage, many of these systems are considered to be safety critical, in general there is a cost function associated with the system. Soft Real-Time systems are systems where deadline overruns are accepted, but not desired. There are no catastrophic effects of missing one or more deadlines but there is a cost associated to overrunning that is often connected to Quality-of-Service (QoS) [37]. Tasks are divided into three models, periodic, aperiodic and sporadic tasks [35].

Periodic real-time tasks are time-driven tasks that are activated regularly at fixed periods. Aperiodic real-time tasks are event-driven tasks that are characterized by the computation time, the deadline and some probabilistic forms for arrival time like Poisson model. Sporadic real-time tasks are tasks with known minimum inter-arrival time [38].

In some real-time scheduling algorithms, tasks could be preemptive or non-preemptive. A task can be preempted if another task of higher priority becomes ready. In contrast, the execution of a non-preemptive task should be completed without interruption once it is started [39].

In priority driven scheduling, a priority is assigned to each task. Assigning the priorities can be done statically or dynamically while the system is running. The real-time scheduling algorithms are categorized, based on their priority assignment method, into fixed and dynamic priority scheduling algorithms.

Earliest deadline scheduling is an optimal dynamic priority scheduler, whereevery process tells the scheduler its deadline. The scheduling algorithm simply allows the process that is in the greatest danger of missing its deadline to run first [40]. Generally, this means that one process will run to completion if it has an earlier

deadline than another. The only time a process would be preempted would be when a new process with an even shorter deadline becomes ready to run.

2.3 Scheduling Algorithms for Cloud Servers

In order to serve the clients' requests within the required QoS requirements, different scheduling algorithms were implemented at the server side. The scheduling algorithm decides which task will be executed next on the server among a set of arrived tasks to the scheduling pool of the single server system. In a multi-server system, the scheduling algorithm will decide which task will be executed next and the server node that will be used to serve such task [12]. The proposed scheduling algorithm depends on the type of services requested by the customers. Accordingly, a set of QoS metrics will be the key of implementing each scheduling algorithm.

Different scheduling algorithms were proposed for cloud computing. In [13], a multi-level scheduling algorithm was proposed to serve tasks over a homogeneous multi-server system. The tasks were categorized into two main priorities with a group of server nodes serving each priority level. Dedicated server scheduling algorithm (DSS) was proposed in [14] to serve a set of tasks in homogeneous environment. The DSS algorithm was an enhancement to the work performed in [13], where the QoS metrics are achieved with minimal number of computational nodes.

In [15], a dynamic priority-based scheduling algorithm was proposed to guarantee the QoS requirements for a pre-prioritized set of arrived tasks to the scheduler in a

homogeneous multi-server system. Such scheduling algorithm was implemented based on an improvement of the DSS and was named as Dynamic-DSS (DDSS) that shows higher performance in serving the clients' requests according to their pre-defined priorities.

Heterogeneous Dynamic Dedicated Server Scheduling (h-DDSS) was proposed in [5] to serve a set of pre-prioritized set of tasks over a heterogonous multi-server system. According to the extensive simulations, h-DDSS shows higher performance over DDSS and DSS in serving the arrived task in terms of throughput, utilization, and miss rate QoS metrics.

Besides guaranteeing the QoS requirements of the arrived tasks, some scheduling algorithms preserve the overall performance of the multi-server system in terms of energy reduction, power consumption and management, and pollution control. Such scheduling algorithms are known as green task scheduling algorithms [16].

In [17], a green scheduling algorithm was proposed to minimize the energy consumption in a multi-server system. The algorithm integrates a neural network predictor for turning of unused servers and restarting them again, and thus minimizing the number of running servers, which leads to minimizing the server's energy consumption. A linear combination of dynamic voltage frequency scaling (DVFS) was proposed in [18] to decrease the energy consumption of the multi-server system. Extensive simulations over a set of 1500 task graphs from three different categories (LU decomposition, random, and Gauss-Jordan) show an efficient power

saving. Two novel scheduling algorithms for power consumption were proposed in [19]. The algorithms are based on predicting the unused time by the task and use it in reducing the execution speed of upcoming tasks. As a result, the overall energy consumption for the multi-server system will be reduced.

In modeling and analyzing the performance of the scheduling algorithms for cloud computing, a lot of research was carried out. A fault-recovery scheduling algorithm was proposed in [15] to analyze the overall performance of the system under such reliability condition. According to the queuing theory, simulation results shows that such recovery option degraded the overall performance of the system through increasing the average response time.

In [20], six green task scheduling algorithms were implemented and analyzed for a multi-server cloud system. According to the energy consumption and miss ratio metrics, simulation results prove that the shortest task first was the most efficient one.

2.4 Security Services for Cloud Servers

Cloud computing is an approach that provides a convenient, universal, and geographical-independent access to a shared set of computing resources. According to such model of centralized data, network technologies should be able to provide a level of security for cloud system, and thus making it robust against different cloud security threats [21]. Such security services follow different categories such as data security, identity and access management, trust, and assurance [22].

Different security models were proposed for cloud computing. According to the networking entity that coordinates and controls the transactions in the security model, three main security models were proposed. In [23], Software as a Service security model (SaaS) was proposed. In such model the network provider is the coordinator of the security model, where all transactions are performed through such end provider. Platform as a Service (PaaS) was proposed in [24] to provide security services that are coordinated by the clients (customers & system developers). Accordingly, such model is what we called it on-demand security model. In [25], Infrastructure as a service security model (IaaS) was proposed to provide security services that are coordinated by both system provider and the clients. Accordingly, a level of negotiation on the security service will occur.

According to data security in cloud computing, confidentiality security service was the key behind protecting the database storage from the spoofing security threat. Such security service could be achieved through cryptography. Encryption is the most common cryptographic technique used for clouds. Different encryption-based security models were proposed for securing data in clouds [26].

In [27], a security model that combines the federated identity management (FIM) with the hierarchical identity-based cryptography (HIBC) was proposed to provide a confidentiality security service in cloud computing. A security model that provides confidentiality security service for data in clouds was proposed in [28]. The model provides a method for cipher text retrieval based on efficient encryption mechanism.