A SEMI-MARKOV DECISION MODEL-BASED BROKERING MECHANISM FOR MOBILE CLOUD MARKET

by

Elena Degtiareva

MSc in Material Science, Moscow Institute for Electronic Engineering, 1984

A Thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2017

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

A Semi-Markov Decision Model-based Brokering Mechanism for Mobile Cloud Market

Master of Science
in Computer Science
Ryerson University

Abstract

As the multitude and complexity of cloud market increases the evaluation and selection of cloud services becomes a burdensome task for the users. With the increased rise of available services from various Cloud Service Providers (CSP), the role of cloud brokers becomes more and more important. In this thesis, the challenge of optimally allocating multiple cloud system resources to multiple mobile user's requests with different requirements is investigated and an optimal Cloud Broker model is proposed. The cloud brokering mechanism is formulated as a Semi-Markov Decision Process (SMDP) model under the average system cost criteria, taking into consideration the cost of the occupying computing resources, the communication costs, the request traffic, and some security risk degrees and resource requirements from the multiple mobile users. Through minimizing the overall system cost, the optimal resource allocation policy is derived by using the Value Iteration Algorithm. Simulation results are provided, demonstrating the efficiency of the proposed Cloud Broker design.

Acknowledgments

Foremost , I would like to express my immeasurable appreciation and deepest gratitude to my supervisor Dr. Isaac Woungang, for his continuous support throughout my graduate studies. His invaluable guidance helped me throughout my research work, and for the completion of this thesis. I would also like to thank the DABNEL lab team, in particular, Dr. Glaucio Carvalho for his fruitful advice throughout my studies and to the Department of Computer Science at Ryerson University for providing me the useful resources toward the accomplishment of my degree.

I wish to express my love, and sincerity to my mother for her continuous encouragement and support throughout my life. I wish to thank my husband Alexandre for his continuous support and motivation, without whom this goal would have been hard to accomplish. I wish to thank my brother, daughter and friends for their continuous encouragement throughout my studies.

Contents

List of Figures

List of Tables

List of Abbreviations

BS	Base Station
CCC	Cloud Computing Center
CSB	Cloud Service Broker
CSP	Cloud Service Provider
CPU	Central Processing Unit
CTMC	Continuous-Time Markov Chain
HetNet	Heterogeneous Wireless Network
IaaS	Infrastructure-as-a-Service
MDP	Markov Decision Process
MCC	Mobile Cloud Computing
MNO	Mobile Network Operator
MU	Monetary Unit
NIST	National Institute of Standards and Technology
OS	Operating System
PaaS	Platform-as-a-Service
QoS	Quality of Service
RAM	Random Access Memory
SaaS	Software-as-a-Service
SLA	Service Level Agreement
SMI	Service Measurement Index
SMDP	Semi-Markov Decision Process
TTP	Third Trusted Party
VM	Virtual Machine
UE	User Equipment

Chapter 1

Introduction

1.1 Context and Motivation

Cloud computing has become a widely used computing infrastructure based on which many existing cloud vendors (here referred to as cloud service providers (or CSPs for short) are providing various different types of cloud services. In this regard, selecting the best cloud services for specific applications is still a challenge since doing so requires that multiple factors such as technologies, policies, business and resource allocation demands, computing requirements, to name a few, be taken into account.

Toward proposing a resource management framework to address this problem, the work carried in this thesis departs from the idea that smartphones, tablets and cloud computing technologies are converging into the so-called new mobile cloud market [1], where the usage of mobile applications and services is expected to growth rapidly. In this context, it is assumed that we have an application that is supposed to run on a mobile device (such as cell phone) but this device is short of resources (e.g. CPU) to carried out that task. Therefore, the user of the device sends a request (via its mobile cellular network) for connection with a CSP via a cloud Broker, who then finds the necessary resource to run the application at the satisfaction of the mobile user.

In this thesis, a cloud brokering mechanism is proposed that can be used to manage the cloud resources across multiple CSPs for both the cloud system and the mobile users. To specify the settings in which the proposed cloud brokering mechanism has been designed, it is necessary to introduce the following paradigms.

1.1.1 Cloud Service Models

According to the National Institute of Standards and Technology (NIST) [2], a cloud service is mainly characterized by five key attributes as follows:

- *On-demand self-service*: a customer must be able to request or cancel at any time the usage of a cloud service with a CSP without requiring human interaction, and it should be as convenient and straightforward as possible.
- *Broad network access*: the customer can utilize the cloud service as long as he has Wi-Fi, broadband, or landline network connectivity without location dependency and with minimal dependency on the device (e.g. mobile phone, tablet, laptop, and workstation) used for the access.
- *Measured service*: the resource usage can be monitored and reported. This allows a CSP to use pay-as-you-go model and to charge their consumers only for the resources consumed by them.
- *Rapid elasticity*: the cloud service can scale up or down, adjusting automatically to the real-world demands from the end users which occur in real-time.
- *Resource pooling*: the cloud resources are shared across multiple customers using a multi-tenancy model. As a result, there are concerns regarding the data security and privacy of the end users.

According to the considered level of abstraction, existing well-known service models [3] for cloud computing are: Platform as a Service (PaaS), Software as a Service (SaaS) and Infrastructure as a Service (IaaS).

- IaaS: this model (also known as hardware-as-a-service) provides the physical machines to the users for the delivery of storage, processing and network. Examples of CSPs running the IaaS model are Amazon web services, Rackspace, CenturyLink, to name a few.
- PaaS: this model allows the users to host their software in the cloud. It is a service in which the CSP provides computing platforms such as OS, servers, programming languages, databases, application program development and deployment tools, application testing, to name a few. By using PaaS, the users can achieve cost savings, can reduce their risks by using protected technologies, can improve software security, and reduce the burden that new systems' development can incur. Examples of CSPs running the PaaS model are Google, Azure, Force.com, to name a few.
- SaaS: this model allows the users to subscribe and access software from their remote systems. It is a shift from locally installed software to on-line software that offers the same functionality, hence, eliminates the costs of purchasing and installing software as well as the cost of maintenance and support by the IT staff. The user is charged only for the bandwidth, not for the hardware. Examples of CSPs running the SaaS model are Salesforce.com, Gmail, Intuit-QuickBooks, to name a few.

Due to the explosive growth in number of IaaS-based CSPs and the cloud computing market, the users of cloud services are now offered many VM types, cloud interfaces and different competitive pricing models. In this context, the cloud brokering mechanisms become a necessity since they are crucial in transforming the heterogeneous cloud market into commodity-like services by providing the scheduling mechanisms that are needed for the optimization of the VMs placement amongst the CSPs. They also offer a common interface for the deployment, monitoring, and engineering of these VMs, taking into account the independence of the individual CSPs technology. Currently, there is no agreed-upon mechanism for achieving the design of such interface [4], but rather, the CSPs interface with each other using their specific sets of APIs in the form of software-based adapter layers. For the cloud Broker design proposed in this thesis, the IaaS model has been adopted.

1.1.2 Cloud Architectures

Based on the aforementioned deployment models, four types of cloud architectures prevail, which are: public cloud, private cloud, community cloud, and hybrid cloud.

- A private cloud is essentially an internal data center that uses a cloud structure that is not available to the general public since its technology, configuration, resources and services are owned and operated by a user or group of users. Typically, this type of cloud is more expensive than a public one, but is also more secure.
- A public cloud is characterized by pay-as-you-go services available, where the user does not own the core technology resources and services but outsources them.
- A community cloud architecture is provisioned for exclusive use by a specific community of consumers that have some common shared concerns such as objective, security requirements, policy, and compliance considerations. It may be owned, managed, and operated by one or more entities in the same community or by a Third Trusted Party (TTP).
- Hybrid cloud: this type of cloud architecture merges the features of both the private and public clouds to offer the benefits of multiple deployment models. It may allow its users to keep critical, confidential data and information in a private cloud while utilizing the public cloud for non-confidential data or additional tasks that the private cloud cannot accommodate.

In using either of these cloud architecture models, the mobile users can have access to resources in the cloud using mobile handsets and they can run sophisticated applications while being provided significant higher data storage and processing resources. However, these advantages also come with several challenges such as communication and architectural security concerns, including virtualization issues, data storage issues, VMs' identity management and access control issues since the sharing of data over the Internet increases data exposure. Data owners worry that their data could be misused or accessed by unauthorized users. The CSP shares the infrastructure between the clients and it also controls the workloads between dissimilar physical machines. Virtualization alters the relationship between the OS and the underlying hardware and brings unique security concerns for users of a public cloud service. Specific concerns include the possibility of compromising the virtualization or hypervisor software. The distribution of the data centers across national boundaries of multinational CSPs is also an issue. This means that organizations are totally unaware of where their data is located or the way it is protected. The diverse privacy regulation and set of laws that can be applied to the management of the data from one country to another may lead to legal problems in case there are disputes. Surveillance by foreign intelligence agencies is a new threat. Data owners need to have strong assurance of data integrity and availability. The CSP may be dishonest and discard the data which has not been accessed or rarely accessed to save the storage. It can keep fewer replicas than promised or even choose to hide the data loss and claim that the data are still correctly stored in the cloud in order to protect its reputation. Contractual, ethical and legal issues, and latency incurred by the massive data transfer to the cloud, availability, supplier stability and interoperability issues are also of concern [5]. The QoS from the CSP may deteriorate or cannot always be available when needed, and the CSP may cease its operations. The MCC environment addresses some of these limitations and provides numerous benefits to both the CSPs and their users [6].

1.1.3 Mobile Cloud Computing

Mobile cloud computing (MCC) is a paradigm and technology used to augment the mobile device capabilities such as computational resources, battery lifetime, to name a few, in an environment made of several cloud systems by wirelessly transferring the computation burden from a mobile device to multiple data centers, where the computing tasks are to be processed using the resources provided by the VMs on the its behalf. With the success of MCC, more CSPs are encouraged to collaboratively cooperate in order to share their resources. In such an Inter-Cloud environment, despite the fierce competition, CSPs will be expected to cooperatively advertise their services and associated prices to their mobile end users, allowing them to choose the best CSP that can meet their budgetary and technical needs. Despite the fact there are multiple CSPs available to choose from, several issues may arise; for instance, that of selecting a suitable CSP to handle the incoming service request from the users. Two basic delivery models of MCC prevail, namely: the federations model and the multi-clouds model.

- *In the federations* model, the CSPs are in agreement with each other to join a federation, with the goal to offer enhanced services to their consumers. In this case, the CSPs voluntarily collaborate with each other to share their resources. This type of MCC is mostly viable for private CSPs and governmental clouds. From an architectural perspective, this MCC model can further be categorized into (1) peer-to-peer MCC architecture - where no mediator is required between the CSPs and the CSPs negotiate directly and communicate with each other to exchange their resources; and (2) centralized MCC architecture - where the presence of a central entity is mandatory and its role is to facilitate the allocation of resources to the CSPs. This central entity acts as both a market place and a repository for the cloud resources. In this model, application brokering is implemented either by the CSPs or in a centralized entity.
- *In the multi-clouds* model, a Third Trusted Party (TTP) is introduced who is responsible for building a unique entry point for the participant CSPs without a-priori agreement between them. In this case, an application is entitled to use multiple CSPs in aggregation. This type of MCC can also be used to share resources from private CSPs or government clouds. From a development perspective, this type of MCC model can further be categorized into (1) Multi-clouds services - where a service local or external to a CSP is responsible for application provisioning and deployment based on provisioning rules or SLA provided by the application developers; and (2) Multi-clouds libraries - where the scheduling and provisioning of the application components across various clouds are directly taken care of by the custom application brokers using the available inter-cloud libraries. In this model, application brokers can be considered as part of the service that enable access to the multiple clouds.

The main driving forces behind using any of these MCC delivery models have been reported in [7]. For instance, the usage of services from multiple clouds may allow the users to achieve optimized costs, unlimited scalability, energy saving, QoS improvement, flexibility in adjusting to changes in constraints imposed by the CSPs such as vendor lock-in, replicate applications/services consuming services from different clouds. It helps to deal with the peaks in service and resource requests, minimizes the possibility of downtime, improves disaster recovery and geo-presence. Directing traffic to the closest to users CSP ensures low-latency access and minimizes issues over data sovereignty.

1.1.4 Cloud Service Brokering

As cloud computing has become an increasingly mainstream phenomenon using multiple clouds strategy, the need to use multiple clouds in order to achieve a lower cost while maintaining an acceptable performance level has necessitated the introduction of a novel concept, that of cloud service brokering [8]. A cloud Broker is a software-based entity whose role is to work with the CSPs individually in order to manage their identities, access by mobile users, and delivery of requests from the UEs to the clouds [9], and to select a suitable CSP that can handle the user request in a timely and effective fashion. As such, our proposed optimal cloud Broker is meant to provide the best possible match for the mobile users and the CSPs. According to [10], the cloud Broker is the best solution for multiple cloud orchestration, including aggregating, customizing, integrating and governing cloud services.

1.2 Research Problem

Modern mobile devices can support diverse OSs, offer a wide variety of tools and applications, and generate massive growth in mobile data, however, they have limited resources such as battery life, network bandwidth, storage capacity, and processor performance. As stated earlier, MCC is a service model where resource-hungry mobile applications are offloading their requests for services and data from resource-constrained devices to powerful cloud data centers in order to satisfy their immediate need for CPU cycles.

In the traditional MCC design, a mobile device wirelessly offloads its computation task to a resourceful remote cloud data center or a cloudlet attached to a base station (BS). Despite the simplicity in terms of designing, planning, operating, and troubleshooting a MCC system based on a single CSP, this approach might result to a vendor lock-in problem, which constrains the client to access the CSP services and products. Additionally, considering the ever growing the MCC demand, the over-reliance on a single CSP might expose the users to critical functional aspects such as high availability, elasticity, reliability, scalability, affordability, which would make difficult, if not impossible, for a single CSP to satisfy.

To successfully cope with these open issues, mobile users can opportunistically exploit the recent rise of multiple CSPs to offload their computational tasks in a more affordable manner in a mobile cloud market, where the CPU cycles can be competitively and freely traded. The realization of a mobile cloud market will also enables a CSP to efficiently leverage its cloud resources utilization by making the resource allocation more flexible and efficient while maximizing its profitability.

A crucial component of a mobile cloud market is the aforementioned brokering mechanism, which is responsible for performing the matchmaking process between the needs of the mobile users and the services offered by the CSPs. By inter-mediating the negotiation between the demand and the supply, the cloud broker can minimize the consumer purchase time and risk in dealing with distinguishable CSPs specifications while maximizing its experience and profitability. The proposed method for selecting the CSP that can best allocate its resource for the mobile user request can directly affect the system capacity of MCC. In this respect, it is assumed that the cloud resource allocation strategy introduced in [11] is exploited by the system.

In this thesis, a cloud brokering mechanism is proposed that can be used to manage the cloud resources across multiple CSPs for both the cloud system and the mobile users. In this setting, (1) the mobile services and applications are delivered from a data center to a mobile device (such as a smartphone) because the mobile device does not have enough computational resources to store or process them; (2) the mobile user accesses these services on-demand using a browser or thin client on their mobile devices; and (3) our proposed cloud broker mechanism is used to select a CSP among the available ones, which has the best possible QoS and price, and with available resources to fulfill the mobile user request.

1.3 Proposed Approach

In this thesis, the cloud Broker mechanism is modelled by means of a Semi-Markov Decision Process (SMDP) model. The choice of using the SMDP as modelling approach is justified by the fact that this paradigm is appropriate when assessing a dynamic system like the one under study, which evolves on time, requiring that its evolution be controlled based on some criteria. In our case, the cloud market evolves based on the arrivals and departures of clients (mobile users) as well as the CSP occupation.

Typically, an SMDP is a decision process in which it is assumed that the system dynamics are determined by a Markov Decision Process (MDP), but the agent cannot directly observe the underlying state of the system and the times between the decision epochs are not constant but random. The transition time between the decision epochs depends on the current system state, the taken action, and the potential next state. An exact solution to the SMDP yields the optimal action which in turn minimizes the expected cost of the agent over a possibly infinite horizon. The sequence of optimal actions is referred to as the optimal policy. In order to efficiently manage the cloud resources across multiple CSPs for both the cloud system and mobile users, the expected overall system cost is evaluated. In our work, the optimal resource allocation policy, which minimizes the overall system cost by seeking a balance between maximizing the CSP VM's utilization and delivering the best prices and QoS to the mobile users, is determined by means of the Value Iteration Algorithm [12].

1.4 Thesis Contributions

In this thesis, we have proposed a novel an SMDP-based formulation of the cloud Broker problem in an MCC environment considering the minimization of the overall system cost as target objective.

The overall system cost takes into account the expected system expenses such as the cost of the occupying computing resources, the communication costs, and the mobile user's requests. The optimal policy that determines the cloud Broker's decision on selecting of the best possible CSP that satisfies the mobile user request is derived.

1.5 Thesis Outline

The thesis is organized as follows:

- **Chapter 1** introduces the subject, motivation, and contributions of our research.
- **Chapter 2** presents some background information and related works.
- **Chapter 3** describes our proposed SMDP-based cloud Broker design in some depth.
- **Chapter 4** describes the performance analysis of the proposed cloud Broker design
- **Chapter 5** concludes the thesis and describes some directions for future work.

Chapter 2

Background and Related Work

This chapter discusses the current cloud market challenges, as well as representative work on cloud Broker design from a resource allocation and VM management perspectives.

2.1 Cloud Market

Cloud computing is not a buzz word anymore and using cloud services has become a usual practice. However, there are several issues that the nowadays open and competitive cloud markets of interoperable cloud services will have to address. As more CSPs do differentiate their services offering by adding more value-added features, the competitiveness in the market tend to grow. Customers are no longer locked-in by the choice of services offered by a single CSP as they are now able to choose from a huge variety of services offered by multiple clouds and which fit their specific needs. As reported in [13], in order to accommodate this scenario, there is a need of tools that can help achieving the following goals:

- Support the CSPs in creating and advertising their offers in this collaborative/cooperative environment. Indeed, the CSP needs to advertise its cloud products and deliver them as much as possible among potential customers. A typical CSP offer can target a particular application type or it can refer to a specific cloud deployment and service model. The CSP itself may want to choose among several price models and selling strategies. An offer can also be associated with some guarantees and a specific technical support service.

- Help the customers to make fine-tuned searches. Indeed, a customer need is to get its job done as soon as possible and at the lowest possible price. The customer is aware that several types of cloud deployment and service models are available and may want to opt for a specific one. The cloud resource usage pattern plan, the adequate technical support service and the SLA guarantees can be part of the customer's requirements.
- Implement the supply-demand matchmaking between what the CSPs offer and what the customers' applications require, in such a way as to maximize both the CSP's profit and the customer's utility.

In [14], Ludwig and Schmid reported two widely used market models that most cloud applications tend to follow, namely the horizontal market model and the vertical one, and studied of the benefits and beneficiaries of customers specification flexibilities. In this regard, while some applications are network-hungry, latency sensitive, and have deadlines and strict QoS networking guarantees, other applications are delay-tolerant. Thus, depending on the application and the adopted market model, customers may have flexibility in setting the execution time as long as the best prices for services are obtained and the CSP may schedule the applications in a more flexible way, making a better use of its resources and sharing the gains with its customers. In the horizontal market model, the customers directly request the cloud resources on demand from different CSPs whereas in the vertical market model (so-called broker market), the customer's requests are handled by a cloud Broker. The cloud Broker can buy the cloud resources from the different CSPs subject to some volume discounts, however, there is a tradeoff that prevails since buying too much contracts may be wasteful and buying less contracts may yield small discounts. In [15], some mechanisms based on economic principles have been suggested, which can be used to address the issue of cloud resource provisioning in a marketplace in terms of resource allocation and pricing following these principles:

- *Request indivisibility* - this principle advocates that any set of VMs requested in a bundle by a user must be delivered entirely by a single CSP;
- *Demand-side aggregation* - this principle stresses that the cloud market should permit the CSPs to make divisible offers that can be used to satisfy multiple customers' requests in the case where the CSPs deliver their services to a large number of users.

In an MCC environment, the availability of resources and workloads changes dynamically. As a result, achieving simultaneously the QoS constraints imposed by the users while maintaining an acceptable level of system's performance and utilization presents some challenges from a resource allocation perspective. Recent research work in this direction have mainly focused on methods for comparing cloud services based on a set of quantitative/qualitative attributes [16]. Quantitative characteristics can be measured without any uncertainty, e.g., response time, while qualitative characteristics refer mainly to service characteristics and cannot be quantified in an objective manner. There are 7 top level categories that allow the comparison between cloud services. These top level categories are refined by three or more levels of attributes.

- *Accountability* - which measures properties such as audit-ability, compliance, governance, sustainability, which characterize the CSP's organization.
- *Agility* - which measures the impact made by a service upon the client's ability to quickly change its strategy with minimal disruption. Examples of agility measures include elasticity, portability, scalability, and flexibility.
- *Assurance* - which indicates how likely the service offered by the CSP will be available as specified. Examples of assurance metrics include availability, reliability, and recoverability.
- *Financial attributes* - which relate to the amount of money spent by the user on acquiring the service from the CSP. Examples of financial metrics include the cost and the billing process.
- *Performance* - this covers the performance features and functions of the provided services. Examples of performance metrics include accuracy, service response time, and interoperability.

- *Security and privacy* - which indicates the effectiveness of controls of a CSP on accessing the services, data, and physical facilities from which the services are provided. Examples of security and privacy metrics include access control, data integrity, data privacy, data loss, and security management.
- *Usability* - this relates to the ease with which a service can be used by its consumers. Examples of usability metrics include installation cost, transparency, to name a few.

Many attributes such as cost, availability, reliability, service response time, auditability, priority, resource demand, interoperability should be considered in the resource allocation problem. It is important to consider the MCC resource allocation problem from the security prospective since offloading computational tasks to the cloud raises users security concerns [17]. The key in gaining trust in cloud computing is to address privacy and security issues as a matter of high priority.

With the recent growth in mobile cloud computing market and the increasing number of IaaS based CSPs, the users of cloud systems are now offered various pricing schemes and types of VMs. In this context, it is essential that the designed cloud Broker be able to provide a commodity-like service, by fulfilling two essential roles, which are: (1) providing a way to schedule and optimize the placement of VMs amongst the multiple CSPs, and (2) work as an intermediary entity between the mobile users and the CSPs to facilitate the connection between the CSPs and the mobile users, allowing them make appropriate business-critical decisions. The cloud Broker not only helps the mobile users in choosing the best and cost-effective CSP that can fulfill their requests, but it also helps in selecting the best possible solution for multiple CSPs orchestration [18] such as data aggregation, customization, and governance of the offered cloud services. In this sense, it was reported [2] that a cloud Broker must achieve:

- *Service Intermediation* - a cloud Broker enhances a given service by improving managing access to cloud services, identity management, performance reporting, enhanced security, etc. and providing value-added services to cloud consumers.
- *Service Aggregation* - a cloud Broker must combine and integrate the multiple services of-

fered by the CSPs into one or more new services. It must also provide data integration while ensuring a secure data transfer between the users and the CSPs; movement between the cloud consumer and multiple cloud providers.

- *Service Arbitrage* - a cloud Broker must have the flexibility to choose the services from the CSPs, for instance, by using a credit-scoring service.

A comprehensive classification of cloud brokering mechanisms as applied in industry and academia is presented in [18], with focus on how Inter-Cloud environments can alleviate the brokering of various applications across different CSPs considering the non-functional requirements. In our case of interest, i.e. MCC services, depending on the way that each application logic is specified, the cloud Broker decides on behalf of the mobile users on the type of brokering requirements. In this regard, brokering requirements can be (1) SLA-based - i.e. the brokering approach relies on a specified SLA between the CSPs and the cloud Broker; (2) Trigger-action based - i.e. a set of triggers and their associated actions (such as allocation of new VMs) are specified. In this thesis, the trigger-action based approach is adopted, which is described in the form of an SMDP-based model.

2.2 Related Work

Several work in the literature have been devoted to cloud broker design, taking into account both the cloud market and cloud brokering requirements. From a resource allocation (i.e. pricing, services) perspective, representative such work are described as follows.

2.2.1 With Respect to Cloud Market Models

With regard to cloud brokering markets, Samimi et al. [19] proposed a market model for cloud resource allocation (called Combinatorial Double Auction Resource Allocation (CDARA)), which includes single-sided auction designs and double-sided auction designs. A single-sided auction is a mechanism implementing a one-to-many price negotiations whereas in a double-sided auction, both sides submit the bids for multiple items. The double-sided auction design is more efficient than the one-sided auction counterpart and prevents providers from having monopolies. The experimental results clearly showed that the proposed method generated higher revenues for CSP and was cost efficient.

The introduction of complex pricing schemes (spot pricing, reservation pricing, dynamic pricing) presents complexities that need to be addressed. Cloud Service Providers may be willing to sell at lower prices the unused capacity which they are not able to allocate through direct-selling.

Bonacquisto et al. [20] proposed a procurement auction-based mechanism to sell the CSP's residual computing capacity. An adaptive bidding strategy is also devised. On the same line, Weinman [21] introduced some discussions on cloud pricing and markets, reporting that: (1) both the CSPs and the customers can benefit from dynamic pricing, where the offering price varies over time; and (2) based on dynamic pricing, the customers can achieve some cost savings and the CSPs can better utilize their resources, for instance, by lowering their prices at off-peak times.

Zhou et al. [22] proposed an auction design for cloud computing jobs, which consists of a pricing framework along with an approach to manage soft deadline constraints, themselves described by a preferred job completion time and a penalty function associated with different degrees of deadline violation. The auction systematically adapts the prices incurred due to a fluctuation in supplies and demands, then allocates the cloud resources to those jobs who value them the most, the goal being to maximize the overall reward of each participating entity in the cloud system.

Javed et al. [23] presented a cloud marker mechanism which does not only assist the CSP in determining the market price and dynamically adjusting its prices in real-time using a supply demand auction-based model, but also support the customers in making the decision of selecting the most suitable CSP that can fulfill their requirements. Following the same trend, Wang et al. [24] proposed a cloud brokerage service mechanism in which the cloud Broker optimally exploits some pricing models for selecting the appropriate CSP to handle the user's request. It takes advantage of pricing benefits of long-term instance reservations and significant volume discount by aggregating the users demands and purchasing a large pool of instances.

Chard and Bubendorfer [25] proposed a cooperative resource allocation scheme in which the CSPs contribute in terms of computing resources to the federation for the purpose of allocation and other management operations. In this setting, the contributed resources from the CSPs are used to provide the core functionality, thereby creating a reliable globally distributed infrastructure. On the other hand, the Cloud federations allows to improve the responsiveness and overall QoS while avoiding the lock-in problem that may arise when using a single CS. In addition, secure and privacy preserving allocation protocols are proposed to conduct trustworthy allocations on potentially untrusted hosts.

Do et al. [26] investigated the pricing models for both the CSPs and the customers' service selection and formulated the problem of pricing competition between the CSPs as a two-level non-cooperative game. Using $M/M/1$ and $M/M/\propto$ as queuing models, some correlations among the expected tasks finishing times, resource capacity, and requested rates are established.

2.2.2 With Respect to Cloud Broker designs

With regard to cloud broker designs and resource management, in [18], Grozev and Buyya provided a comprehensive survey of cloud brokering mechanisms as applied in industry and academia, with focus on how Inter-Cloud environments can alleviate the brokering of various applications across different clouds taking into account the nonfunctional requirements. A qualitative analysis of trends and challenges for Inter-Cloud applications taking into account the role played by the brokering mechanisms is also proposed, revealing that the focus of most research projects in academia is on investigating the challenges related to the development of Inter-Cloud federations with high focus on SLA-based brokering mechanisms, whereas industry relevant projects are mostly concerned with the development of services for scheduling and provisioning across MCC, with emphasis on actions to achieve direct and flexible brokering management. The work carried in this thesis fits within the above-mentioned reported research challenges.

Talbi and Haqiq [27] proposed an efficient cloud Broker mechanism that can be used to identify the vulnerabilities that may occur in the CSP system, then analyze them in order to rank the CSPs based on measuring the risks of confidentiality, integrity, and availability.

Liang et al. [28] studied the problem of resource management in a multi-domain mobile cloud system and proposed an SMDP-based computing model for inter-domain services in such environment considering the system gain, communication costs, cloud computational resources expenses, and mobile users' quality of experience. In their approach, the inter-domain resource transfer relies on a SMDP model, with the goal to maximize the overall rewards of the mobile users and cloud system based on randomized arrivals and departures of the mobile application services. It should be noted that the SMDP's decision epochs are chosen at the points when random events occur and the optimal policy reveals the optimal resource allocation among mobile cloud service domains. Simulation results have shown that in terms of service rejection probability, the proposed algorithm outperforms the greedy one in which the resources are systematically allocated to the mobile service requests as much as possible. However, no comparison of this model against other benchmarking SMDP-based models is presented. In addition, their proposed model supports the migration of services between MCCs but does not take into account the fact that the VM occupancy may fluctuate over time in neighbouring CSPs.

The model for resource allocation for security services in [29] has two security categories: Critical Security (CS) and Normal Security (NS). CS service implements more complex security features such as longer key size, stronger authentication and encryption algorithms, more strict security access policies and as a result consumes more cloud resources. However this approach cannot modify its policy according to the resource availability, traffic and does not deal with the varied requirements of resources from mobile users.

Liu and Lee [17] proposed a security-aware resource allocation model for which the basic idea consists of classifying the requests within two dimensions: (1) the multiple risk degrees standing - for the security requirement level; and (2) the minimum number of virtual machines (VM) required for the application execution of the mobile user. In their scheme for a request with a higher risk degree, the system allocates additional VMs. Although this model is a promising one, it is reported to be limited to a single offloading site.

Tordsson et al. [30] proposed a cloud Broker architecture that can provide an optimized placement of VMs across multiple CSPs according to user-specified criteria. Their proposed model can also provide an interoperability layer on top of the various CSPs' interfaces. Experimental results are provided, confirming that the multi cloud deployment provides lower costs and better QoS to the users compared to what a single CSP would have provided.

Mehta et al. [31] introduced a two-level cloud Broker system that can be used to search for the desired cloud resources in an Inter-Cloud environment based on standard VM features such as processor speed, RAM, disk space, bandwidth and the CSP's custom features such as cost of resources and trust level of a resource provider attributes. The first level broker (so-called master broker) consists of registration, searching, and controlling modules whereas the second level broker (so-called local broker) acts on each resource provider site and helps the users to control their leased resources.

Li et al. [32] proposed a trust-aware cloud brokering scheme for efficient matching of cloud resources with the user's requests in an Inter-Cloud environment. Their scheme is composed of a sensor-based service monitoring (SSM) module, which dynamically monitors the real-time service of the allocated resources and a resource matching module, which selects the resources for the users from a trusted pool of resources using the so-called Hybrid and Adaptive Trust Computation Model (HATCM) to compute the trust degree of a service resource.

A cloud service broker framework based on dynamic game theory for bilateral SLA negotiation in cloud environment was proposed in [33]. Three models (nonlinear model, transaction period model and exponential function model) with different satisfaction degree were introduced. However in the negotiation process only 2 SLA properties (the price and bandwidth) were considered.

Achar and Thilagam [34] proposed a cloud broker architecture with multiple QoS criteria that can be used for selecting a suitable CSP among available ones in an Inter-Cloud environment. Their proposed scheme uses specific criteria to evaluate, weight, and rank the CSPs based on their so-called Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) approach.

TOPSIS implements an effective multi-attributes ranking mechanism using a service measurement index (SMI).

Felemban et al. [35] proposed an architecture for MCC over wireless networks consisting of several cloudlets integrated into base stations which provides a close-to-the-user proxy system functionality and facilitates the traffic conformance between the wireless networks and the Internet. The proposed architecture uses multiple functional layers to address the resource management challenges and QoS requirements such as end-to-end delay, bandwidth, and buffering. However, there is no analytical model provided, nor simulation studies involving the use of a case study to validate the proposed MCC architecture.

Manikandan and Kousalya [36] presented the design of an intelligent broker system capable of selecting the best CSP among multiples ones based on some functional and non-functional QoS parameters for each CSP, considering a decision-tree based classification analysis. The proposed model is suitable for cross cloud service selection. Simulation experiments using various classification algorithms such as LADTree, BTTree, supporting both nominal and categorical data have been provided to validate the proposed model in terms of accuracy and execution time.

Zhou et al. [37] investigated the cloud market mechanisms for computing jobs with completion deadlines and proposed a design of online auctions for cloud resource provisioning that can explicitly handle the jobs with prescribed deadlines. Their proposed scheme includes a pricing framework for truthful online auctions and a linear programming-based solution for handling soft deadline constraints expressed by cloud users. However, the provision of applying the proposed method for handling soft deadlines in other auction designs is discussed, but no design proposal is presented.

In [8], the utility theory in economics [38] is exploited to investigate two parameters that can affect the cloud brokering roles, namely the response time of VMs and the prices of their instances, with the goal to achieve the best possible QoS for the users and cloud broker profit. A multi-objective genetic based algorithm for cloud brokering (so-called MOGA-CB) is proposed, which dispatches the best possible combination of VM instances in terms of response time and cost using a set of Pareto optimal assignments. The behaviours of the proposed algorithm is analyzed using different various configurations, yielding the relationship that may prevail between the above targeted objectives and the profit/satisfaction results. Simulations results are provided, showing that the assignment of VM requests is effectively handled before the scheduling phase is terminated. However, a better economic model is yet to be incorporated in this approach in order to minimize the response time of the VM instances under realistic scenarios.

The selection of a CSP among various CSPs to handle the user's requests may be dependent on the considered pricing models, VM instance types, and a range of value-added features. This selection may may involve choosing appropriate services based on the user's requirements. Following this trend, Subramanian and Savarimuthu [39] proposed a cloud brokering architecture that optimizes the placement and deployment of virtual resources in an Inter-Cloud environment, with the goal to select the CSP that offers the best possible cloud services at optimal cost considering various attributes and logical constraints associated with the services offered by each CSP. The proposed cloud broker design is formulated as a mixed integer program and solved using the Ben ders decomposition algorithm. Numerical and sensitivity analysis are presented to validate the effectiveness of the proposed model. However, the scenario of dynamic workloads has not been investigated, nor the effect of VM migration.

Wagle et al. [40] proposed an evaluation model in the form of a visual recommender system to assess the real status and quality of cloud services delivered by the CSPs to their users, in terms of service delivery compliance and compliance with the SLA. The proposed SLA-based brokering model selects the best possible CSP based on both the services experienced by the users and the quality of the services delivered by the CSPs in terms of static and dynamic attributes of these services. However, the proposed model shows some limitations in the sense that the feedback of the users with respect to the quality of the offered services is not taken into account in this model, nor the penalty that may incurred for service violation by the CSPs.

Ghosh et al. [41] provided a survey of various trust models in cloud computing environments and proposed a trust-based framework called SelCSP which acts as a cloud broker to accommodate the user's request based on estimating the risk of interaction with the CSPs. This is done through modelling the reputations of the CSPs by combining the concepts of competence - based on transparency of the CSP's SLA guarantees, and trustworthiness - computed from direct interactions with the CSPs as well as their reputations. The proposed risk estimation model quantitatively computes the risk involved with interaction with the CSPs, based on which the ideal CSP is selected to offer its services to the user. A case study validating the effectiveness of the proposed model is provided, showing promising behaviours. However, the extension of the proposed model to cover the case of a secure multi-domain collaboration in cloud environments is yet to be addressed.

Tossi et al. [42] investigated the problem of maximizing the CSP revenue while optimizing the admission control to resources in an environment composed of multiple cloud markets in terms of reservation, on-demand pay-as-you and spot markets. Yangui et al. [43] developed an open source broker (called CompatibleOne) which can supports all three primary roles for a cloud Broker, namely aggregator, integrator, and customizer. The proposed scheme provides a middleware for the description and federation of heterogeneous clouds and different CSPs' resources. CompatibleOne is based on the object-based CompatibleOne Resource Description System (CORDS) description model and the Advanced Capabilities for CORDS (ACCORDS) model. The CORDS model is used for cloud applications, services and resources description whereas the ACCORDS model is used for cloud application provisioning and deployment.

Vaezpour et al. [44] proposed an approach (advocated by telecom companies) to provide cloud services and act as a Mobile Telecom Cloud (MTC) brokerage. Typically, the MTC can be used to manage third-party cloud (TPC) resources, allowing mobile users to maintain their connections to the telecom network infrastructure and their CSPs based on their QoS requirements.

Aazam and E. Huh [45] proposed a framework for advanced resource allocation operated by a cloud Broker. Based on the types of services and customers, prices are determined and the QoS are maintained. The prediction and pre-allocation of resources also depend on the type of customers.

Son et al. [46] proposed a SLA-based framework which can be used to facilitate the resource allocation. In this scheme, the resource capacity of a CSP is assumed to be limited. Empirical results are provided, showing that the proposed workload and location-aware resource allocation scheme (WLARA) performs better than other related benchmarking schemes such as the round robin, greedy, and manual allocation schemes in terms of SLA violations.

Amato and Venticinque [47] proposed a model for measuring the compliance of the SLAs against the users requirements. The SLA template expresses the configuration of resources that are necessary for the user and can be complemented with other related information. On the other hand, the brokering policy sets the constraints and objectives to be enforced by the brokering algorithm. These constraints can be architectural or they can be considered as service-level constraints such as hard and soft constraints.

Wagle [48] proposed a multi-objective optimization method for multi-cloud brokering considering both the SLA offer and the SLA delivered by the CSPs. The proposed technique enables the users to choose guaranteed cloud services from the multi-cloud architecture according to their expectations. The implementation of this method in the cloud brokering architecture enables the CSB to offer the cloud services at minimal cost and it also helps the customers to choose suitable cloud services according to the user's expectations.

Park et al. [49] proposed a cloud Broker design called virtual cloud bank (VCB), which can be used to identify and recommend the cloud services from a consumer-centric perspective. In the VCB scheme, the service recommendation process increases the efficiency of the system while decreasing the cost of the cloud service recommendation. The consumers' requirements and the specifications of the cloud services are collected during the preparation step. The requirement analysis step consists of some functional features such as filtering, pairwise comparison, and importance rate calculation. The calculated importance rate is utilized to quantify the score obtained for the CSP during the service-matching step. The final step is service recommendation, where the selected candidate service lists are displayed in descending order of their calculated scores and the consumer selects a service among the available recommended services.

Qiu et al. [50] addressed the problem of optimizing the energy efficiency in green computing while maintaining a higher service level performance to the tenants. A pricing policy is derived which maximizes the cloud Brokers profit while minimizing the CSP energy cost. The proposed algorithm distributes the tenants demands to multiple CSPs. It is shown that this scheme yields a superior performance compared to some benchmark schemes in terms of energy efficiency and resource utilization.

Patiniotaki et al. [51] proposed a cloud Broker design called preference-based cloud service recommender (PuLSaR), which relies on a holistic multi-criteria decision making (MCDM) approach. In this scheme, the brokerage service takes into account both precise and imprecise metrics while dealing also with their fuzziness.

In this thesis, the approach taken for designing the cloud Broker differs fundamentally from previous approaches by the fact that the optimal cloud Broker problem is formulated as an SMDP model, with the goal to minimize the overall system cost, itself derived by taking into account the cost of computing resources occupancy, the communication costs, and the QoS from the user perspective.

Chapter 3

Methodologies

In this Chapter, the proposed model, its system states, and the actions are described. The cost structure, which is critical for making a decision whether the mobile service request will be accepted or rejected, is presented. Finally, the Value Iteration algorithm [12], used to derive the optimal policy is described.

3.1 System Model

The considered physical system is depicted in Fig.3.1(a) where the mobile users offload their computation tasks through a broadband wireless network to a cloud market, which hosts different CSPs. In our work, we assume that the resources to be negotiated are VMs at a fixed prices. By collecting the prices and the demands of the mobile users for service, the cloud brokering mechanism performs the matchmaking process. Fig.3.1(b) displays a high-level abstraction of the cloud market containing solely its major components.

Let $\mathbb{C} = \{1, \cdots, \mathbf{C}\}$ be the set of CPSs, each supplying V_j VMs to the cloud market, where $1 \leq j \leq \mathbf{C}$. For simplicity, but without loss of generality, it is assumed that the mobile users come into the cloud market looking for secure services. Thus, the service requests vary based on their risk degrees and the corresponding number of VMs needed to properly execute the application. For instance, a low risk degree application features a low security requirement and a few VMs, and

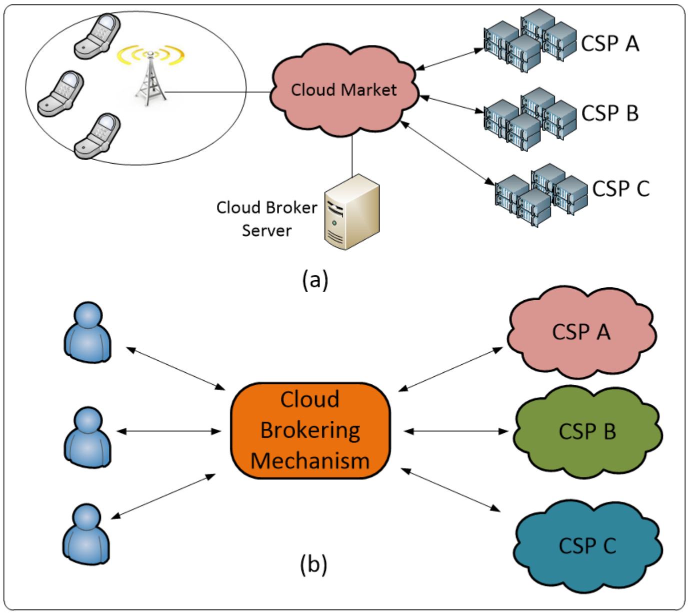

Figure 3.1: System under analysis: (a) Physical system (b) Abstraction.

so on. Let R_i and b_i denote respectively the risk degree and the number of VMs required to secure the i service class application, where $i \in \{1, \cdots, R\}$.

As for the arrival process, it is assumed that service class i arrives into the system according to an independent Poisson process with parameter λ_i and it requires a service time that is exponentially distributed with mean $1/b_i\mu_i$. Other remaining parameters used are listed in Table 3.1. Most of these parameters are inherited from the settings presented in [11].

Parameter	Description
R_i	Risk degree of the i^{th} service class (SC) request
b_i	Number of VMs required by the i^{th} SC request
V_j	Number of VMs in the j^{th} CSP
λ_i	Arrival rate of the i^{th} SC request
μ_i	Departure rate of the i^{th} SC request
δ_j	Task transmitting time to the j^{th} CSP
m_i	Local processing cost of the i^{th} SC
p_{ij}	Price of i^{th} SC occupying the j^{th} CSP per VM
β	Price per mobile unit time
Q_i	Penalty of rejecting a i^{th} SC request

Table 3.1: Notations

3.2 Optimal Control Problem

The optimal control problem makes use of the Semi Markov Decision Process (SMDP). This framework is an outgrowth of Markov models and dynamic programming which has been developed for the study of sequential decision problems [52]. Unlike the MDP model, in a SMDP model, the times between transitions are considered as continuous random variables whose distribution may depend upon the current state, the action taken and possibly the next state [53], [54]. Also, the actions take variable amounts of time as they are intended to model temporally-extended courses of action. In addition, the arrivals and departures determine the dynamics of the system, which are modelled by exponential distributions. In this case, the decision epochs are state transition epochs with random lengths. It should also be noted that to obtain the optimal policy, the continuous time SMDP model is first converted into a discrete time MDP model in such a way that for each stationary policy, the average cost per time unit in the discrete-time Markov model is similar to that used in the Semi-Markov model. After that, the Value Iteration algorithm [12] is used in the transformed model to obtain the optimal policy.

3.2.1 Proposed SMDP Model

The proposed SMDP model is composed of system states, actions, expected time until the next decision epoch, state transitions, and the design of a suitable cost function, as described in the sequel.

3.2.2 State Variables, System State, and State Space

Let x_{ij} denote the number of allocated VMs to the i^{th} service class into the j^{th} CSP. The vector state $\mathbf{x} = [x_{ij}, \nu]$ is given by every combination of x_{ij} that does not surpass the system capacity $\sum_{j=1}^{\mathbf{C}} \sum_{i=1}^{R} b_i x_{ij} \leq V_j, j \in \mathbb{C}$.

Considering the risk degree VM requirement b_i, the maximum number of x_{ij} co-running applications is obtained as $\lfloor \frac{V_j}{b_i} \rfloor$, where $\lfloor g \rfloor$ is the largest integer not greater than g. Furthermore, let ν

be a vector of size $R+1$ whose elements represent the last occurred event. Considering the system under analysis as a queueing system, the events that govern its dynamics are: $\nu(1) = 0$ - which denotes the service departure and $\nu(2) = 1, \cdots, \nu(R+1) = R$ which denote the arrival of the service class $1, 2, \cdots, R$, respectively. Based on these quantities, the state space is given by

$$X = \left\{ \mathbf{x} : \sum_{j=1}^{\mathbf{C}} \sum_{i=1}^{R} b_i x_{ij} \leq V_j, j \in \mathbb{C} \right\}. \tag{3.1}$$

3.2.3 Actions

To control the brokering mechanism towards the optimal behavior, a set of actions $\mathbb{A}$ is defined. An action $a \in \mathbb{A}$, which will determine how the system will dynamically react to external and internal events, is selected for each state. Based on ν values, we get

- $a = ij$, which denotes the acceptance of the i^{th} risk degree request into the j^{th} CSP for $\nu(2), \cdots, \nu(R+1)$;
- $a = 0$, which enotes the rejection of the i^{th} risk degree request into the j^{th} CSP for $\nu(2), \cdots, \nu(R+1)$. In order to minimize the SMDP size, we have adopted the action $a = 0$ for the service departure event $\nu(1) = 0$ as well. Howver, for this case, the optimal control interprets the command as doing nothing.

3.2.4 Expected Time Until the Next Decision Epoch

Considering the compound random processes that dynamically make the system to evolve, the expected time until the next decision epoch $\tau_s(a)$ is a combination of the arrival processes $\Lambda = \sum_{i=1}^{R} \lambda_i$ and departure processes $\Delta = \sum_{j=1}^{\mathbf{C}} \sum_{i=1}^{R} b_i x_{ij} \mu_i$. Thus, if the system is in the vector state $\mathbf{x} \in X$ and the action a is selected, then $\tau_{\mathbf{x}}(a) = \dfrac{1}{\Lambda + \Delta}$.

3.2.5 Transition probabilities

Let $P_{\mathbf{xy}}(a)$ denote the probability that in the next decision epoch, the system will be visiting the vector state $\mathbf{y}$ given that the current state vector matrix is $\mathbf{x}$ and the action a is taken. Considering the random events previously defined, the following possible state transition can be defined:

3.2.5.1 Arrival Events

An arrival of the i^{th} risk degree request takes place with probability $\lambda_i \tau_{\mathbf{x}}(a)$, $i \in \{2, \ldots, R+1\}$. From this point onwards, the service request may be either rejected, i.e., the controller selects the action $a = 0$ or admit into the j^{th} CSP where $j \in \mathbb{C}$, i.e., the controller selects the action $a = ij$. Based on the chosen action, the embedded Markov chain will stay in the same state, i.e., $\mathbf{y} = \mathbf{x}$ for a rejection or transit to $\mathbf{y} = \mathbf{x} + \mathbf{1}_{ij}$ where $\mathbf{1}_{ij}$ is a matrix containing only 0s, except for the (i, j) position, which is 1.

3.2.5.2 Departure Events

A departure event of the i^{th} risk degree ongoing service on the j^{th} CSP takes place with probability $b_i x_{ij} \mu_i \tau_{\mathbf{x}}(a)$. In this case, there is nothing to do, but letting the service departure. Thus, for an event $\nu[1]$ and action $a = 0$, the embedded Markov chain will move to $\mathbf{y} = \mathbf{x} - \mathbf{1}_{ij}$. For any other event, $P_{\mathbf{xy}}(a) = 0$.

Fig. 3.2 depicts the decision making process operated by the proposed optimal cloud Broker and the corresponding system states for the arrival of Service Class 1 requests in a cloud system with two public clouds. By taking into account the traffic, security risk and availability of the resources of the CSP, the optimal cloud Broker (CB) evaluates the overall system cost in order to make a decision as to whether the request should be accepted or not. If the CB decides to reject the incoming service request, this request will be locally processed on the mobile device. Otherwise, the request will be accepted and the CB will decide on the suitable CSP to handle it and allocate the required resources accordingly.

For a state $s = [x_{11}, x_{12}, x_{21}, x_{22}, 1]$, if the request is allocated to CSP1, s changes to

$[x_{11}+b_1, x_{12}, x_{21}, x_{22}, 0]$, and the next possible state can be obtained as follows:

- if $\nu = 1 : [x_{11}+b_1, x_{12}, x_{21}, x_{22}, 1]$
- if $\nu = 2 : [x_{11}+b_1, x_{12}, x_{21}, x_{22}, 2]$
- if $\nu = 0$ and $x_{11} > 0 : [x_{11}+b_1-b_i, x_{12}, x_{21}, x_{22}, 0]$
- if $\nu = 0$ and $x_{12} > 0 : [x_{11}+b_1, x_{12}-b_i, x_{21}, x_{22}, 0]$
- if $\nu = 0$ and $x_{21} > 0 : [x_{11}+b_1, x_{12}, x_{21}-b_i, x_{22}, 0]$
- if $\nu = 0$ and $x_{22} > 0 : [x_{11}+b_1, x_{12}, x_{21}, x_{22}-b_i, 0]$

3.2.6 Cost Structure

Based on the selected action $a \in \mathbb{A}$ and the system state $\mathbf{x} \in X$, a cost $C_{\mathbf{x}}(a)$ is incurred. Following the definition in [55] the cost can be evaluated as the sum of the cost of decision making and the continuous cost of resource usage.

The cost of decision making includes $\delta_j\beta$ the transition expense to transfer the task from the mobile device to the j^{th} CSP, $\beta/b_i\mu_i$ the resource occupation expense, and Q_i/R_i the penalty for rejecting a service request as a way to reinforce the need to accept the incoming service requests.

The continuous cost of resource usage includes b_ip_{ij} the cost of the i^{th} service class occupying the j^{th} CSP as well as m_i the local processing cost of the i^{th} service class on the mobile device if a new service request is rejected. The cost structure used to determine the behavior of the optimal brokering algorithm is defined as

$$C_{\mathbf{x}}(a) = \delta_j\beta + \beta/b_i\mu_i + Q_i/R_i + b_ip_{ij} + m_i, \tag{3.2}$$

where δ_j denotes the time consumed on transmitting the task, β is the price per mobile unit time, μ_i is the departure rate of the requested service task, b_i is the number of VMs required by the i^{th} service class request, Q_i represents the penalty of rejecting the i^{th} service class request with a

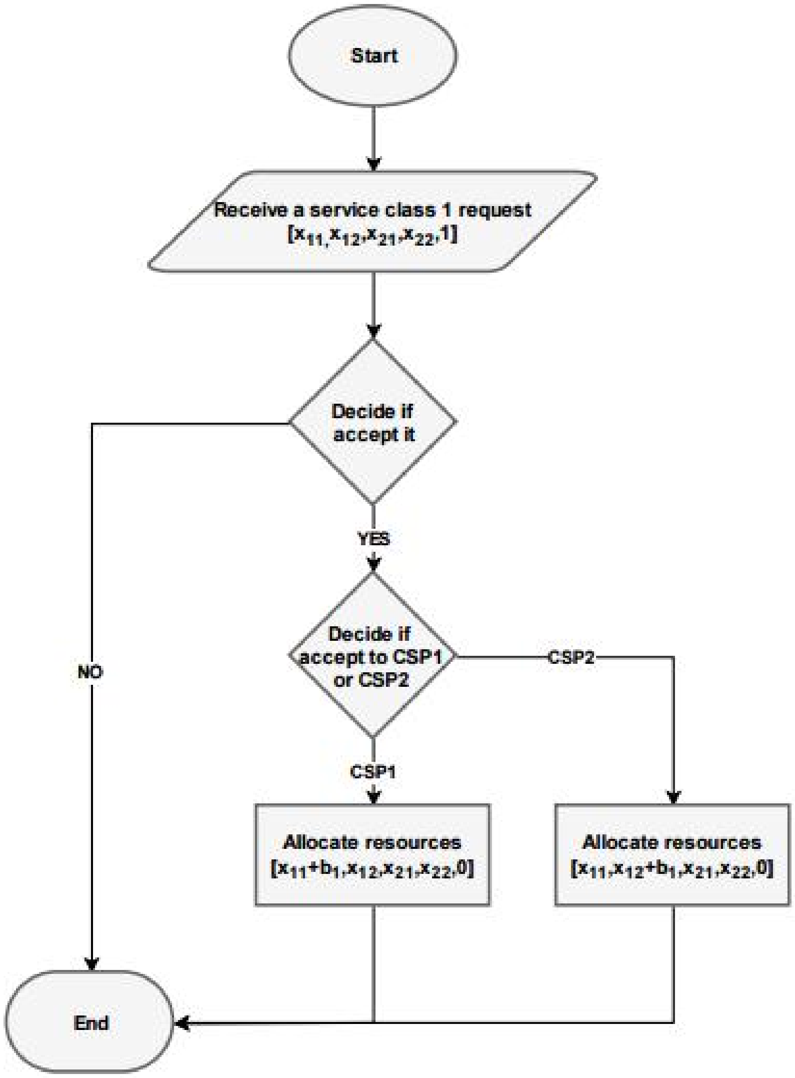
Start
Receive a service class 1 request $[x_{11},x_{12},x_{21},x_{22},1]$
Decide if accept it
YES
NO
Decide if accept to CSP1 or CSP2
CSP2
CSP1
Allocate resources $[x_{11}+b_1,x_{12},x_{21},x_{22},0]$
Allocate resources $[x_{11},x_{12}+b_1,x_{21},x_{22},0]$
End

Figure 3.2: Decision making process considering an Inter-Cloud with two CSPs.

lowest risk degree, p_{ij} is the price of the i^{th} service class occupying the j^{th} CSP per 1 VM and m_i is the local processing cost of the i^{th} service class on the mobile device.

3.2.7 Optimization Problem and Value Iteration Algorithm

Let ζ denote a stationary policy and let $\psi_{\mathbf{x}}(\zeta)$ be its average cost. Let $Z(t)$ be the total cost incurred up to time t, where $t \geq 0$. Let $E_{\mathbf{x},\zeta}$ be the expectation operator when the initial state $\mathbf{x}_0 = \mathbf{x} \in X$ and the policy ζ is used. Then, according to [12], the limit

$$\psi_x(\zeta) = \lim_{t\to\infty} \frac{1}{t} E_{x,\zeta}[Z(t)] \tag{3.3}$$

exists for all $\mathbf{x} \in X$.

Based on (3.3), the optimization problem is that of minimizing $\psi_{\mathbf{x}}(\zeta)$ among all policies, i.e. determining $\psi^* \leq \psi_{\mathbf{x}}(\zeta)$ for all $\mathbf{x} \in X$, which is the minimal average cost whose the optimal policy is ζ^*.

In this thesis, the Value Iteration Algorithm [12] is applied to derive the optimal policy. The principle behind this method is to approximate the minimal average cost through a sequence of value functions $V_n(\mathbf{x})$ for all $\mathbf{x} \in X$. The value functions provide lower and upper bounds on the minimal average cost, which iteratively converge to the minimal average cost. The Value Iteration Algorithm is described as follows [12]:

Step 0: Choose $V_0(\mathbf{x})$ such that $0 \leq V_0(\mathbf{x}) \leq \min_a\{C_{\mathbf{x}}(a)/\tau_{\mathbf{x}}(a)\}$ for all $\mathbf{x} \in X$. Choose a number τ with $0 < \tau < \min_{x,a} \tau_x(a)$. Let $n := 1$.

Step 1: Compute the recursive function $V_n(\mathbf{x}), \mathbf{x} \in X$ as

$$V_n(\mathbf{x}) = \min_{a\in A(x)} \left[\frac{C_x(a)}{\tau_x(a)} + \frac{\tau}{\tau_x(a)} \sum_{y\in X} p_{xy}(a) V_{n-1}(y) \right.$$
$$\left. + \left(1 - \frac{\tau}{\tau_x(a)}\right) V_{n-1}(x) \right].$$

Let $\zeta(n)$ be a stationary policy whose actions minimize the right-hand side of the above recursive function.

Step 2: Compute the bounds

$$m_n = \min_{y\in X}\{V_n(y) - V_{n-1}(y)\} \text{ and}$$

$$M_n = \max_{y\in X}\{V_n(y) - V_{n-1}(y)\}.$$

The algorithm is stopped with policy $\zeta(n)$ when $0 \leq \frac{M_n - m_n}{m_n} \leq \varepsilon$ where ε is a predefined accuracy number. In this thesis, $\varepsilon = 10^{-12}$. Otherwise, go to Step 3.

Step 3: $n := n + 1$ and go to Step 1.

After a finite number of iterations, the algorithm terminates and outputs a policy $\zeta(n)$ whose average cost function $\psi_x(\zeta(n))$ satisfies $0 \leq \frac{\psi_{\mathbf{x}}(\zeta(n))-\psi^*}{\psi^*} \leq \varepsilon$ for all $\mathbf{x} \in X$.

The optimal policy ζ^* is a decision rule $f : X \to A$ that dictates the action $f(\mathbf{x}) \in A(\mathbf{x})$ each time the system is observed in the state $\mathbf{x} \in X$ [12]. Under ζ^*, the underlying Continuous-Time Markov Chain (CTMC) model is solved. To this end, its infinitesimal generator matrix $\boldsymbol{Q}$ is build following the specifications of the optimal policy. From that point on, taking into account the normalization condition $\sum_{\mathbf{x} \in X} \pi(s) = 1$, one can compute the steady-state probability vector $\boldsymbol{\pi}$ by solving the system of linear equations $\boldsymbol{\pi Q} = 0$ using a standard numerical technique. In this thesis, the successive over-relaxation (SOR) method has been considered for this purpose [56].

Chapter 4

Performance Evaluation

This Chapter studies the performance of the proposed SMDP-based cloud Broker design. A customized event-driven simulation system written in Borland C++ which implements the SMDP dynamics, is used for the evaluation. Simulations are conducted using $1.0e^{-12}$ as precision in the Value Iteration Algorithm [12].

4.1 Simulation Setup

The considered system model is depicted in Fig.1(a), where the mobile users offload their computation tasks through a wireless network to a cloud market consisting of two public clouds (Cloud 1 and Cloud 2) with different capacities and access prices. The arriving requests are classified into two service classes, namely service class 1 and service class 2. Service class 2 has the higher security risk degree as well as the number of VMs being requested. In the simulations, parameters such as access price and cloud capacity are varied for performance comparison purpose.

For our numerical computation, the following parameters are considered: local processing cost $m_1 = m_2 =$ 1 monetary units (MU); $Q_1 = Q_2 =$ 1 MU; risk degree $R_1 =$ 1, $R_2 =$ 2; arrival rates $\lambda_1 = \lambda_2 =$ 15 requests/s; departure rates $\mu_1 = \mu_2 =$ 6.6 s^{-1}; $\delta_1 = \delta_2 = 0.1s$; $\beta =$ 0.015 MU; $b_1 = 3$ VMs, $b_2 = 4$ VMs; access prices $p_{11} = p_{21} =$ 0.05 MU, $p_{21} = p_{22} =$ 0.04 MU.

4.2 Performance Metrics

4.2.1 Local Processing Probability

It is the probability that the i^{th} risk degree service request be blocked. To compute this, we have to consider the probability that the optimal controller selects the action $a = 0$ for an arrival event $\nu(2), \cdots, \nu(R+1)$, which implies that the request is locally process. P_{LP} is obtained as

$$P_{LP}(a=0) = \sum_{\mathbf{x} \in X} \pi(\mathbf{x}). \tag{4.1}$$

4.2.2 CSP Utilization

The j^{th} CSP utilization U_j is defined as the ratio between the mean number of busy VMs and the total number of available VMs, i.e.

$$\begin{aligned} U_j(a) =& \frac{1}{V_j}\Bigg(\sum_{x_{1j}=1}^{V_j} \sum_{x_{2j}=1}^{V_j} \cdots \sum_{x_{Rj}=1}^{V_j} (b_1 x_{1j} + b_2 x_{2j} + \\ & \cdots + b_R x_{Rj} \Bigg) \pi(\mathbf{x}). \end{aligned} \tag{4.2}$$

4.3 Numerical Results

4.3.1 Impact of the Access Price

The numerical results are presented for two different scenarios, where the number of total available VMs in the cloud are set to (1) $V1 = 20$ VMs, $V2 = 10$ VMs, and (2) $V1 = 10$ VMs, $V2 = 20$ VMs respectively. Cloud 1 access prices p_{11}, p_{21} are 0.05 MU and remain the same, while Cloud 2 access prices p_{12}, p_{22} are varied from 0.04 to 0.06 MU. The impact of the access price on the local processing probability, cloud utilization and optimal cost of the system are shown in Fig. 4.1, Fig. 4.2 and Fig. 4.3 respectively.

It can be observed that while the Cloud 2 access price is less than the Cloud 1 access price,

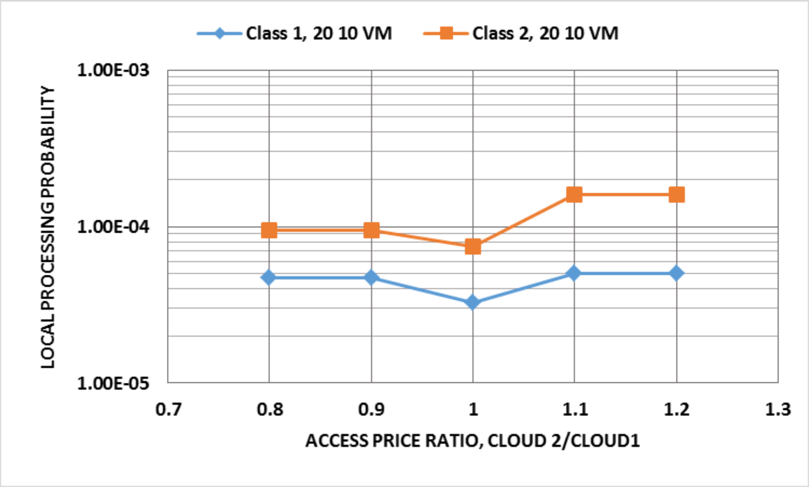

Figure 4.1: The local processing probability under various Cloud 2 access prices.

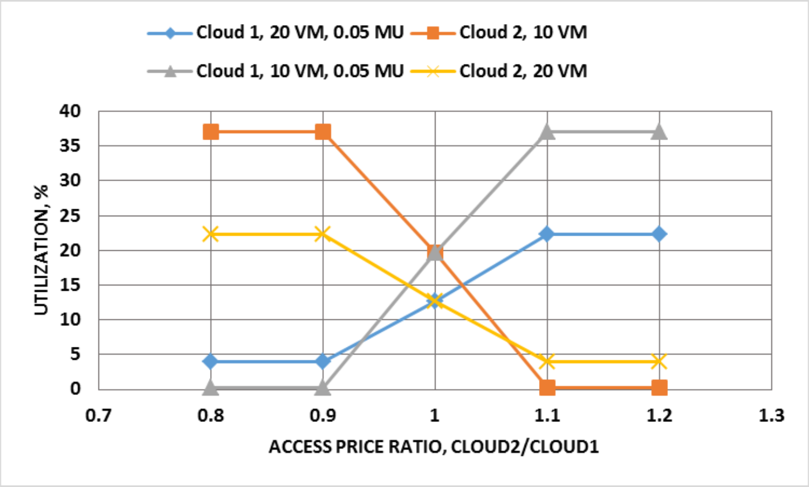

Figure 4.2: The utilization under various Cloud 2 access prices.

the Optimal Broker routes the majority of service requests to Cloud 2. When Cloud 2 access price increases, its utilization starts to decrease offset by the increase in Cloud 1 utilization. The Cloud with the lowest access price at any point gets the most amount of traffic. Fig. 4.1 reveals that a significant drop in the local processing probability for both service classes occurs when Cloud 1 and Cloud 2 access prices are equal. As the Cloud 2 access price exceeds the Cloud 1 access price the Optimal Broker decides to resort to a local processing more often in order to ensure the lowest cost. Since the service class 2 requests require more resources than its counterpart, its local

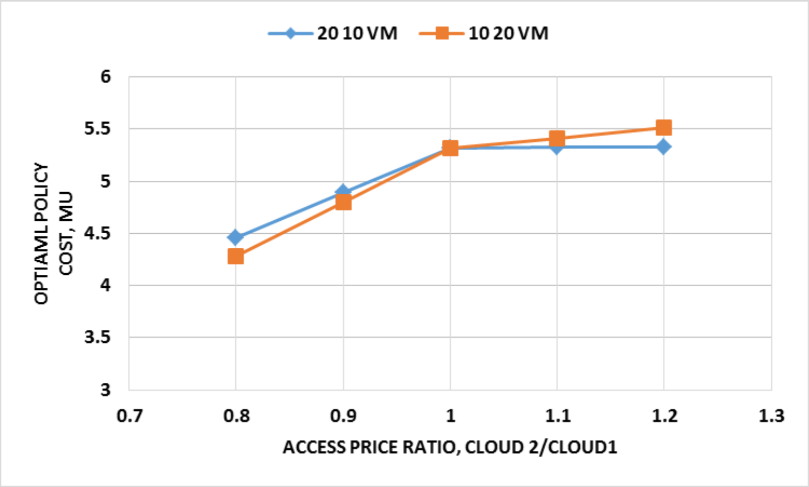

Figure 4.3: The optimal cost under various Cloud 2 access prices.

processing probability increases sharply. The optimal cost goes up, which results in a degraded overall reward of the cloud system. The results show that the CSPs can ensure a better QoS and gain a higher revenue when they cooperate.

4.3.2 Impact of the Number of Requested VMs

In this scenario, the number of requested VMs is varied and the impact of this variation on the local processing probability is investigated. The number of VMs being requested by service class 1 b_1 was varied from 1 to 5, while the number of VM being requested by service class 2 was $b_2 = b_1 + 1$.The results are captured in Fig 4.4. It can be observed that the higher the number of VMs requested, the higher the local processing probability. The local processing probability increases with the arrival rate. It can also be observed that service class 2 generates higher local processing probability compared to service class 1 since it requires more resources.

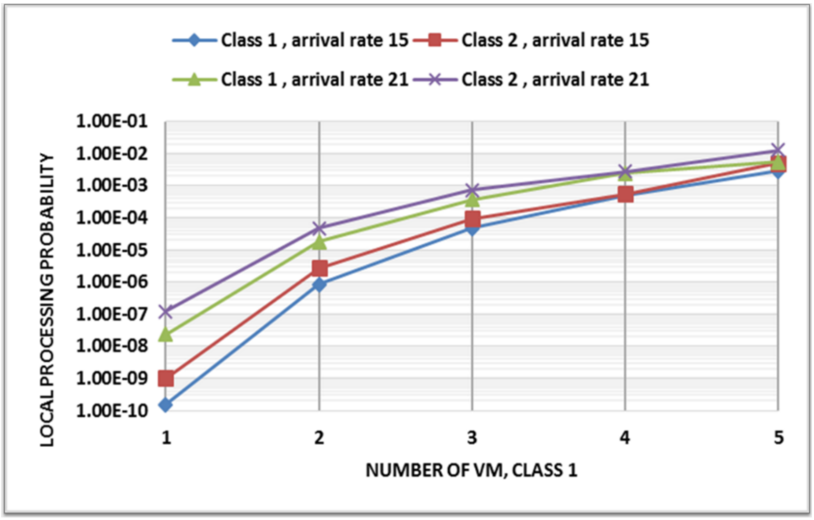

Figure 4.4: The local processing probability under various number of requested VM.

4.3.3 Impact of the Arrival Rate

In this scenario, the arrival rate is varied from 11 to 19 and the impact of this variation on the performance metrics is investigated. The numerical results are presented for two different scenarios, where the number of total available VMs in the cloud are set to (1) $V1$ = 20 VMs, $V2$ = 10 VMs, and (2) $V1$ = 18 VMs, $V2$ = 8 VMs respectively. The results are captured in Fig 4.5 and Fig 4.6 respectively. It can be observed that with an increased service arrival rates, the local processing probability and cloud utilization become higher. Since service class 2 requires more cloud resources than service class 1, the service class 2 requests are more likely to be rejected when the cloud resources are limited due to less VMs available in the cloud system or the increased traffic.

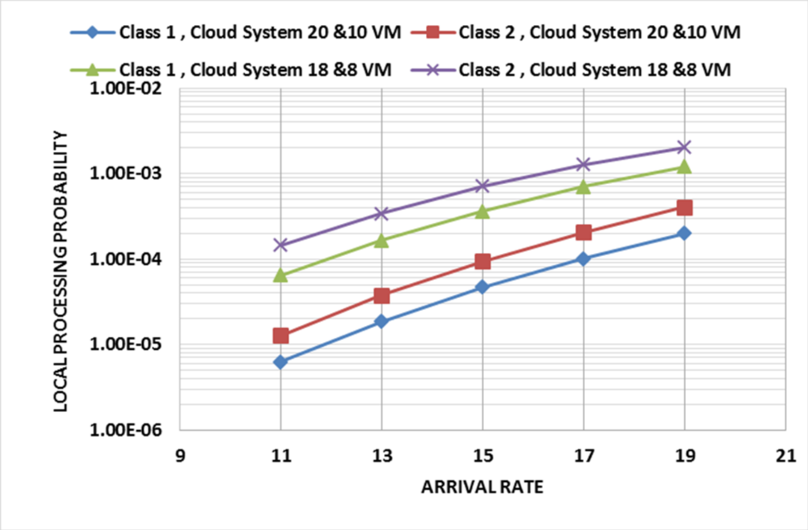

Figure 4.5: The local processing probability under various service arrival rates.

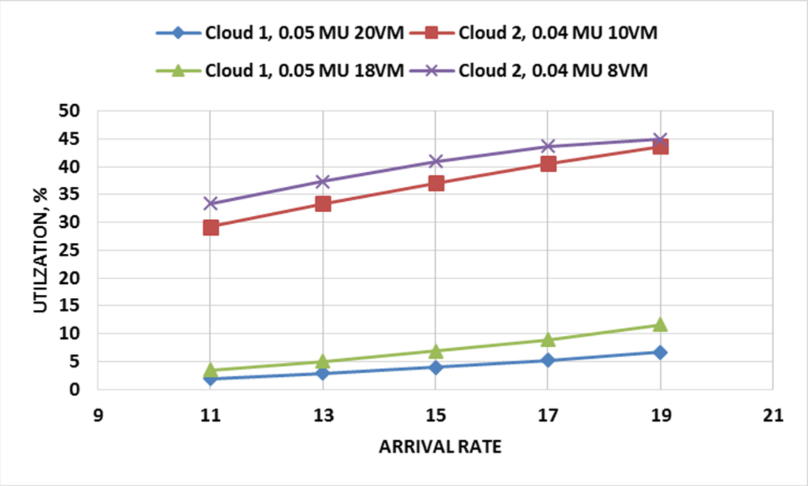

Figure 4.6: The utilization under various service arrival rates.

4.3.4 Impact of the Departure Rate

In this scenario, the departure rate is varied from 0.1 to 0.2 and the impact of this variation on the performance metrics is investigated. The numerical results are presented for two different scenarios, where the number of total available VMs in the cloud are set to (1) $V1$ = 20 VMs, $V2$ = 10 VMs, and (2) $V1$ = 18 VMs, $V2$ = 8 VMs respectively.

The local processing probability and cloud utilization under various service departure rates are shown in Fig 4.7 and Fig 4.8 respectively. With an increased service departure rate the local processing probability and cloud utilization decreases. Since service class 2 requires more cloud resources it is more likely to be rejected. A lower local processing probability is achieved when more cloud resources are available due to higher cloud system capacity or higher departure rate.

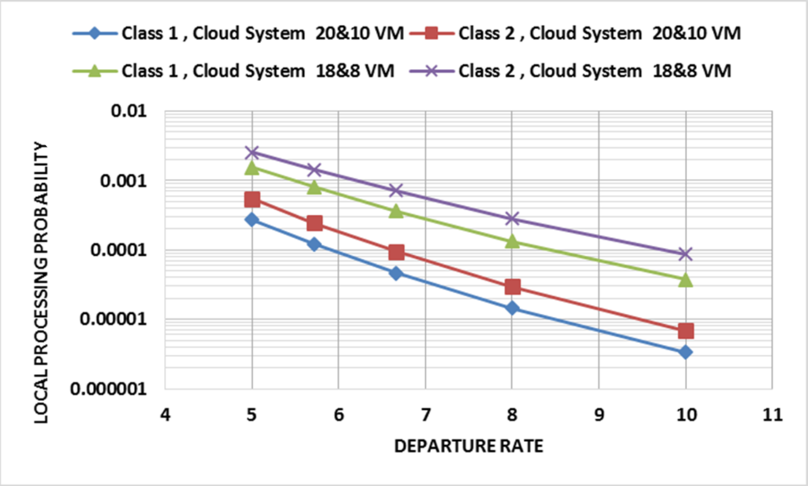

Figure 4.7: The local processing probability under various service departure rates.

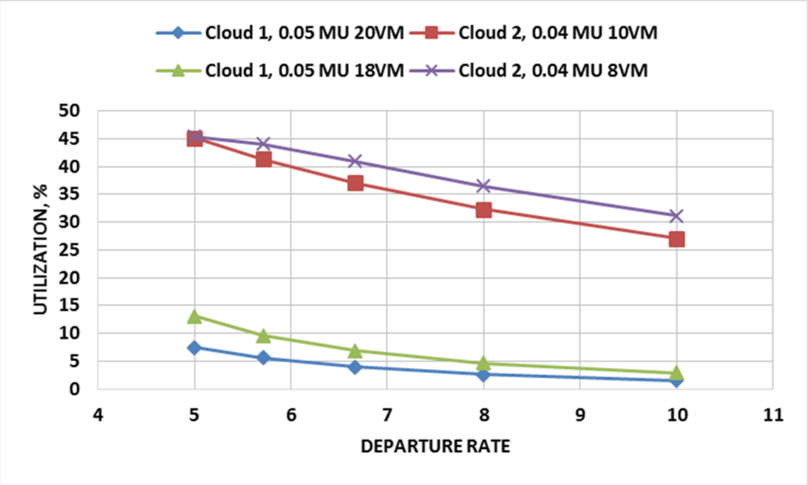

Figure 4.8: The utilization under various service departure rates.

4.3.5 Impact of the Cloud Capacity

In this scenario, the number of VMs in cloud 2 is constant (10 VMs) and the number of VMs in the cloud 1 is varied; and the impact of this variation on the local processing probability is studied. The results are captured in Fig 4.9. It can be observed that the smallest number of VMs yields the highest local processing probability. It can also be observed that with an increase in the system capacity, the local processing probability decreases accordingly.

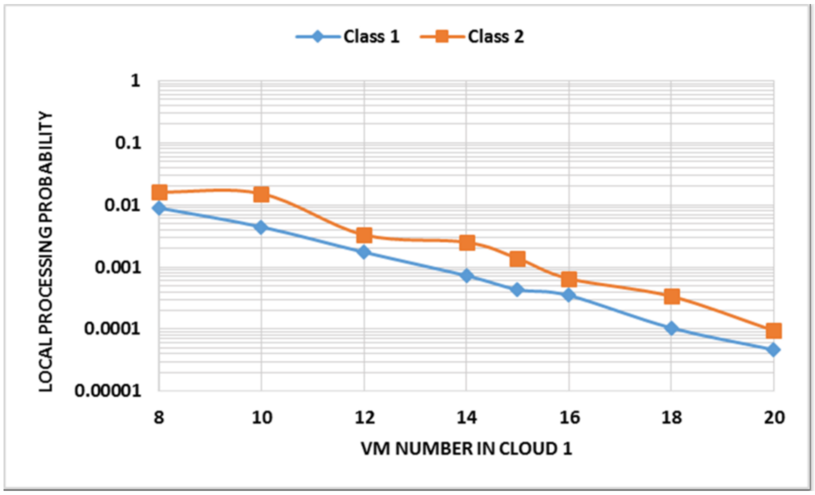

Figure 4.9: The local processing probability under various Cloud 1 capacity.

4.4 Analysis of the Optimal Policy

4.4.1 Impact of the access price on the structure of the optimal policy

The optimal resource allocation policy is presented for 2 different scenarios, where the Cloud 2 access prices are 0.04 MU and 0.06 MU respectively. The number of the total available VMs is set to 10 in Cloud 2 and 20 in Cloud 1, arrival rate $\lambda_1 = \lambda_2 =$ 15 requests/s; departure rate $\mu_1 = \mu_2 =$ 6.6 s^{-1}.

The selected actions for service class 1 and service class 2 requests when the Cloud 2 access price is 0.04 MU are presented in Table 4.1 and Table 4.2 respectively. Table 4.3 and Table 4.4 show the structure of the optimal policy for service class 1 and service class 2 respectively when the Cloud 2 access price is 0.06 MU. In those tables, the following convention has been adopted:

○ denotes the selection in Cloud 1;

☆ denotes the selection in Cloud 2;

◇ denotes that the call is processed locally in the device.

In the above tables, it can be observed that when the resource is sufficient, the cloud Broker allocates less expensive cloud resources to both types of service requests. As the number of available VMs decreases, the system starts to allocate more expensive cloud resources to the incoming service class 2 requests and reserve less expensive cloud resources for the potential incoming service class 1 requests with lower requirements on the VMs. The service class 1 requests will be processed locally only when both Cloud 1 and Cloud 2 are at full capacity.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	☆	☆	☆	○
1	☆	☆	☆	○
2	☆	☆	☆	○
3	☆	☆	○	○
4	☆	☆	○	○
5	☆	☆	○	◇
6	☆	☆	○	◇

Table 4.1: The structure of the optimal policy for service class 1 when p_{12} = p_{22}= 0.4 MU.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	☆	☆	☆	○
1	☆	☆	☆	○
2	☆	☆	○	○
3	☆	☆	○	○
4	☆	☆	○	○
5	☆	☆	◇	◇
6	☆	☆	◇	◇

Table 4.2: The structure of the optimal policy for service class 2 when p_{12} = p_{22}= 0.4 MU.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	○	○	○	○
1	○	○	○	○
2	○	○	○	○
3	○	○	○	○
4	○	○	○	○
5	○	☆	☆	◇
6	☆	☆	☆	◇

Table 4.3: The structure of the optimal policy for service class 1 when p_{12} = p_{22}= 0.6 MU.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	○	○	○	○
1	○	○	○	○
2	○	○	○	○
3	○	○	○	○
4	○	○	○	○
5	☆	☆	◇	◇
6	☆	☆	◇	◇

Table 4.4: The structure of the optimal policy for service class 2 when p_{12} = p_{22}= 0.6 MU.

4.4.2 Impact of the departure rate on the structure of the optimal policy

In this scenario, the impact of the departure rate on the structure of the optimal policy is investigated. Tables 4.5 and 4.6 show the structure of the optimal policy for service class 1 and service class 2 respectively using the service requests departure rate of 5 s^{-1}. It can be observed that with the decreased service requests departure rate, more computing resources are demanded. By comparing Tables 4.1 and 4.5, it can be observed that as the departure rate decreases, the system starts to allocate more less expensive cloud 2 resources and less more expensive cloud 1 resources to the incoming service class 1 requests. By comparing Tables 4.2 and 4.6, it can also be observed that with the decreased service requests departure rate, the system starts to allocate more expensive Cloud 1 resources and less expensive Cloud 2 resources to the incoming service class 2 requests.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	☆	☆	☆	○
1	☆	☆	☆	○
2	☆	☆	☆	○
3	☆	☆	☆	○
4	☆	☆	☆	○
5	☆	☆	☆	◇
6	☆	☆	☆	◇

Table 4.5: The structure of the optimal policy for service class 1 when p_{12} = p_{22}= 0.4 MU, departure rate μ_1 = μ_2=5 s^{-1}.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	☆	☆	○	○
1	☆	☆	○	○
2	☆	☆	○	○
3	☆	☆	○	○
4	☆	☆	○	○
5	☆	☆	◇	◇
6	☆	☆	◇	◇

Table 4.6: The structure of the optimal policy for service class 2 when p_{12} = p_{22} = 0.4 MU, departure rate μ_1 = μ_2=5 s^{-1}.

4.4.3 Impact of the arrival rate on the structure of the optimal policy

In this scenario, the impact of the arrival rate on the structure of the optimal policy is investigated. Tables 4.7 and 4.8 show the structure of the optimal policy for service class 1 and 2 with service requests arrival rate 21 s^{-1}. It can be observed that with the increased service arrival rate, more computing resources are demanded. By comparing Tables 4.1 and 4.7, it can be observed that with the increased service requests arrival rate, the system starts to allocate more less expensive Cloud 2 resources and less more expensive Cloud 2 resources to the incoming service class 1 requests. By comparing Tables 4.2 and 4.8, it can also be observed that as the number of available resources decreases, the system starts to allocate more expensive cloud resources to the incoming service class 1 requests and less expensive cloud resources to the incoming service class 2 requests.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	☆	☆	☆	○
1	☆	☆	☆	○
2	☆	☆	☆	○
3	☆	☆	☆	○
4	☆	☆	☆	○
5	☆	☆	☆	◇
6	☆	☆	◇	◇

Table 4.7: The structure of the optimal policy for Service Class 1 when p_{12} = p_{22}= 0.4 MU, arrival rate $\lambda_1 = \lambda_2 = 21\ s^{-1}$.

Requests accepted by Cloud1/Cloud2	0	1	2	3
0	☆	☆	○	○
1	☆	☆	○	○
2	☆	☆	○	○
3	☆	☆	○	○
4	☆	☆	○	○
5	☆	☆	◇	◇
6	☆	☆	◇	◇

Table 4.8: The structure of the optimal policy for Service Class 2 when p_{12} = p_{22}= 0.4 MU, arrival rate $\lambda_1 = \lambda_2 = 21\ s^{-1}$.

Chapter 5

Conclusion

With the extraordinary rise of available services from various Cloud Service Providers, the role of cloud brokers has become more and more important. This thesis have addressed the challenge of optimally allocating multiple cloud system resources to the mobile user's requests with different requirements by proposing an optimal cloud Broker model.

Among the available stochastic decision making algorithms, we have opted to formulate the optimal cloud Broker design within the framework of the SMDP. As such, the system under analysis is a sequential decision problem where the times between the decision epochs are random and the results of the actions taken are uncertain. The SMDP model has proven to be a natural and powerful mathematical tool to analyze such type of problem, the goal was to determine an optimal policy that minimizes the overall total cost. There are two well-known methods for deriving optimal policy for Semi-Markov decision processes, namely iterative policy improvement algorithms and linear programming. Compared to linear programming models, iterative policy improvement models are considerably more efficient in terms of space and time and space. Considering the curse of dimensionality that often occur when using SMDP models with a multidimensional state space, we have opted to use the Value Iteration algorithm for computing the optimal policy that can help achieving a satisfactory performance in terms of overall system cost. The overall system cost is derived by taking into consideration the expected system expenses such as the cost of occupying computing resources, the communication costs and the security risk degree of mobile requests. The optimal resource allocation policy focuses on maximizing the rewards for both the cloud system and the users by maximizing the cloud utilization and minimizing the number of service requests rejections that degrade the user's QoS.

The effects of the cloud resource capacity, arrival and departure rates of the service requests on the performance metrics and optimal resource allocation policy are studied. In our proposed model, the considered simulation values used for the local processing cost, arrival rates, departure rates, number of VMs, and access prices are inherited from [11]. These were selected to mimic the ratio values that can be used in real systems. Extensive simulations have been performed to validate the effectiveness of the proposed cloud Broker design. Simulation results results reveal that: (1) in the presence of multiple clouds (here 2 clouds), when the access price of any the clouds increases, the optimal Broker is able to route the service requests to the appropriate CSP for handling or to rely on the mobile device's local processing to perform the service request while assuring the lowest possible cost of processing. In addition, the results also show that the CSPs can ensure a better QoS and gain a higher revenue ifn they cooperate; (2) when the number of requested VMs (by the mobile user) is increased, the local processing probability also increases with the arrival rate; (3) when the service arrival rate (respectively departure rate) increases, the local processing probability and cloud utilization increases as expected and the optimal Broker is able to determine which service class requests are more likely to be rejected when the cloud resources are limited; and (4) when the system capacity increases, the local processing probability decreases.

The analysis of the derived optimal policy reveal that: (1) when there are sufficient resources in the system, the optimal Broker tends to allocate less expensive cloud resources to both types of service requests; and (2) the optimal cloud Broker demands for more computing resources to satisfy the service requests from the mobile users when a decrease is observed in the service requests departure rate (respectively arrival rate).

As future work, we plan to: (1) compare our proposed cloud Broker design algorithm against the baseline greedy ”first come, first serve” algorithm. To the best of our knowledge, there has been no optimal cloud Broker design proposed so far in the literature which makes use of the same system model and SMDP model considered in this thesis; and (2) explore other stochastic decision making approaches, with the goal to design cloud Brokers that are comparable in performance with the one presented in this work.

Since most of the current research efforts now focus only on quantitative or quantifiable metrics, we also plan to incorporate qualitative characteristics such as auditability, compliance, sustainability, governance, into our proposed SMDP model and assess the efficiency of the obtained cloud Broker design using simulations.

Finally, to support the mobile user in selecting a suitable CSP to handle his/her request, energy-efficient green cloud services can also be investigated. According to the current research trends, many incentives can be used to maximize the greenness of cloud services, which may lead to new cloud Broker designs; but these are yet to be explored. In addition, improving the energy efficiency may help reducing the CSP operational costs while creating higher demand for green cloud services and possible larger market shares.

Bibliography

[1] Business Wire: http://www.businesswire.com/news/home/20161108005831/en/Global-Mobile-Cloud-Market-Worth-USD-38.48 (Last visited Jan 26, 2017)

[2] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, ”NIST Cloud Computing Reference Architecture Recommendations of the National Institute of Standards” NIST Special Publication 500-292, ISBN:1478168021, 2012.

[3] M. N. O. Sadiku, S. M. Musa, O. D. Momoh, ”Cloud Computing: Opportunities and Challenges,” IEEE Potentials, vol. 33, Feb. 2014, pp.34-36.

[4] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, I. M. Llorente, ”Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers”, Future Generation Computer Systems 28 (2012) 358-367

[5] M. Ali, S.U. Khan, and A.V. Vasilakos, ”Security in cloud computing: Opportunities and challenges,” Information Sciences, Volume 305, June 2015.

[6] A. N. Toosi, R. N. Calheiros, R. Buyya, ”Interconnected cloud computing environments: Challenges, taxonomy, and survey,” ACM Comput. Surv. 47, 1, Article 7, 2014.

[7] D. Petcu ”Consuming Resources and Services from Multiple Clouds. From Terminology to Cloudware Support” Journal of Grid Computing, vol. 12, June 2014, pp.321-345.

[8] Y. Kessaci, N. Melab, E-G. Talbi, ”A Pareto-based Genetic Algorithm for Optimized Assignment of VM Requests on a Cloud Brokering Environment”, Proc. of IEEE Congress on Evolutionary Computation (CEC), Cancun, Mexico, June 20-23, 2013, pp. 2496-2503.

[9] Gartner report about cloud service broker. http://www. gartner.com/it/page.jspid=1064712

[10] S. Murugesan and I. Bojanova, Encyclopedia of Cloud Computing, Chichester, West Sussex, UK: John Wiley & Sons, 2016.

[11] G. H.S. Carvalho, I. Woungang, A. Anpalagan, M. Jaseemuddin, E. Hossain, ”Intercloud and HetNet for Mobile Cloud Computing in 5G Systems: Design Issues and Challenges”, (Accepted Dec 25, 2016), IEEE Network Magazine. In Press.

[12] H. C Tijms, ”A first course in Stochastic models”, John Wiley and Sons Ltd, ISBN: 978-0-471-49880-3, 492 pages, 2003.

[13] G. D. Modica and O.Tomarchio, ”Matching the business perspectives of providers and customers in future cloud markets,” Cluster Computing, vol. 18, Issue 1, 03/2015.

[14] A. Ludwig and S. Schmid, ” Distributed Cloud Market: Who Benefits from Specification Flexibilities,” ACM SIGMETRICS Performance Evaluation Review, vol. 43, Issue 3, 11/2015.

[15] S. Chichin, B. Vo and R. Kowalczyk, ”Towards Efficient and Truthful MarketMechanisms for Double-sided Cloud Markets,” IEEE Transactions on Services Computing, 2016.

[16] CSMIC, Cloud Service Measurement Index Consortium: SMI framework Version 2.1. 2014 (Last visited Dec. 13, 2016)

[17] Liu and M. J. Lee, ”Security-Aware Resource Allocation for Mobile Cloud Computing Systems”, Proceedings of the 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, 3-6 Aug., 2015, pp.1-8.

[18] N. Grozev and R. Buyya, ”Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey,” Software: Practice and Experience, vol. 44, no. 3, pp. 369-390, March 2014.

[19] P. Samimi, Y. Teimouri and M. Mukhtar, ”A combinatorial double auction resource allocation model in cloud computing,” Information Sciences, vol. 357, 08/2016.

[20] P. Bonacquisto, G. D. Modica, G. Petralia and O. Tomarchio, ”A Procurement Auction Market to Trade Residual Cloud Computing Capacity,” IEEE Transactions on Cloud Computing, vol. 3, no. 3, July-Sept., pp. 345 - 357, 2015.

[21] J. Weinman, ”Cloud Pricing and Markets,” IEEE Cloud Computing, vol. 2, no. 01 Jan.-Feb., pp. 10-13, 2015.

[22] R. Zhou, Z. Li, C. Wu and Z. Huang, ”An Efficient Cloud Market Mechanism for Computing Jobs With Soft Deadlines,” IEEE/ACM Transactions on Networking, vol. PP, no. 99, pp. 1-13, 2016.

[23] B. Javed, P. Bloodsworth, R. Ur Rasool and O. Rana ”Cloud Market Maker: An automated dynamic pricing marketplace for cloud users,” Future Generation Computer Systems, vol. 54, no. 1, pp.52-67, 2016

[24] W. Wang, D. Niu, B. Li and B. Liang , ”Dynamic Cloud Resource Reservation via Cloud Brokerage,” Proceedings of 2013 IEEE 33rd International Conference on Distributed Computing Systems, pp.400-409,2013

[25] K. Chard and K. Bubendorfer, ”Co-operative Resource Allocation: Building an Open Cloud Market using Shared Infrastructure,” IEEE Transactions on Cloud Computing, vol. PP, no. 99, 2016.

[26] C.T. Do, N.H. Tran, E. Huh, C.S. Hong, S. Choong, D. Niyato and Z. Han, Dynamics of service selection and provider pricing game in heterogeneous cloud market, Journal of Network and Computer Applications, vol. 69, pp. 152165, July 2016.

[27] J. Talbi and A.Haqiq, ”A broker framework for selecting secure cloud service provider using security risk approach ,” International Journal of Cloud Computing, vol. 5, No. 3, pp. 175-186, 2016

[28] H. Liang, L. X. Cai, D. Huang, X. Shen, D. Peng, ”An SMDP-Based Service Model for Inter-domain Resource Allocation in Mobile Cloud Networks,” IEEE Trans. on Vehicular Technology, vol. 61, no. 5, June 2012.

[29] H. Liang, D. Huang, L. X. Cai, X. Shen and D. Peng, ”Resource Allocation for Security Services” in Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference, Shanghai, 2011, pp. 191-195.

[30] J. Tordsson, R.S. Montero, R. Moreno-Vozmediano and I.M. Llorente, ”Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers,” Future Generation Computer Systems, vol. 28(2), pp. 358367, 2012.

[31] H.M. Mehta, P. Pawar and P.Kanungo, ”A Two Level Broker System for Infrastructure as a Service Cloud,” Wireless Personal Communications, vol. 90, Issue 3, 10/2016.

[32] X. Li, H. Ma; F. Zhou and W. Yao, ”T-Broker: A Trust-Aware Service Brokering Scheme for Multiple Cloud Collaborative Services,” Proceedings of IEEE Transactions on Information Forensics and Security, vol. 10, Issue 7 pp. 1402 - 1415, 2015.

[33] H. Chen, X. Liu, H. Xu and C. Wang, ”A cloud service broker framework Bilateral SLA Negotiation in Cloud Environment,” International Journal of Grid and Distributed Computing, vol. 9, no. 9, pp. 251-268, 2016.

[34] R. Achar and P. S. Thilagam, ”A broker based approach for cloud provider selection,” Proceedings of 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1252 - 1257, 2014.

[35] M. Felemban, S. Basalamah, A. Ghafoor, ”A distributed cloud architecture for mobile multimedia services”, IEEE Network, vol. 27, no. 5, pp.20-27, 2013.

[36] R. Manikandan and G. Kousalya, ”A framework for an intelligent broker model of cloud service selection”, Asian Journal of Information Technology 15 (11): 1776-1784, 2016.

[37] R. Zhou, Z. Li, C. Wu, Z. Huang, ”An Efficient Cloud Market Mechanism for Computing Jobs with Soft Deadlines”, IEEE/ACM Trans. on Networking, vol. PP, Issue 99, Oct. 2016, pp. 1-13.

[38] N. G. Mankiw, ”Principles of economics”, 5th Ed., South Western Educational Publishing, ISBN 0324589972, 2008, 872 pages.

[39] T. Subramanian, N. Savarimuthu, ”Application based brokering algorithm for optimal resource provisioning in multiple heterogeneous clouds”, Vietnam J Comput Sci (2016) 3,pp. 57-70.

[40] S. S. Wagle, M. Guzek, P. Bouvry, R. Bisdorff, ”An Evaluation Model for Selecting Cloud Services from Commercially Available Cloud Providers”, Proc. of IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), Vancouver, BC, Canada, Nov. 30 - Dec 03, 2015, pp. 107-114.

[41] N. Ghosh, S. K. Ghosh, S. K. Das, ”SelCSP: A Framework to Facilitate Selection of Cloud Service Providers”, IEEE Trans. on Cloud Computing, vol. 3, no. 1, Jan-Mar. 2015, pp. 66-79.

[42] A. Toosi, K. Vanmechelen, K. Ramamohanarao and R. Buyya, ”Revenue Maximization with Optimal Capacity Control in Infrastructure as a Service Cloud Markets,” IEEE Transactions on Cloud Computing, pp. 261-274, 2015.

[43] S. Yangui, I.J Marshall, J.P. Laisne and S. Tata, ”CompatibleOne: The open source cloud broker,” Journal of Grid Computing, vol. 12(1), pp. 93109, 2014.

[44] S. Y. Vaezpour, K. Wu and G. C. Shoja, ”Mobile Telecom Cloud brokerage with orchestrated multi-tier resource pooling,” in Cloud Networking (CloudNet), 2015 IEEE 4th International Conference, 5-7 Oct. 2015, Pisa, Italy, 2015.

[45] M. Aazam and E. Huh, ”Advance resource reservation and QoS basedrefunding in cloud federation,” Proceedings of 2014 IEEE Globecom Workshops (GC Wkshps), pp. 139 143, 2014.

[46] S. Son, G. Jung and S. Jun, ”An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider,” The Journal of Supercomputing, 64(2), pp. 606637, 2013.

[47] A. Amato and S. Venticinque, ”Modelling, design and evaluation of multi-objective cloud brokering,” International Journal of Web and Grid Services, vol. 11, Issue 1, 2015.

[48] S. S. Wagle, ”Cloud Service Optimization Method for Multi-cloud Brokering” Proceedings of 2015 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 132-139, 2015.

[49] J. Park, Y. An, T. Kang and K. Yeom, ”Virtual cloud bank: consumer-centric service recommendation process and architectural perspective for cloud service brokers,” Computing, vol. 98, Issue 11, 11/2016.

[50] C. Qiu, H. Shen and L. Chen, ”Towards green cloud computing: Demand allocation and pricing policies for cloud service brokerage,” Proceedings of 2015 IEEE International Conference on Big Data (Big Data), pp. 203 - 212, 2015.

[51] I. Patiniotakis, Y. Verginadis and G. Mentzas, ”PuLSaR: preference-based cloud service selection for cloud service brokers,” Journal of Internet Services and Applications, 6:26, Aug. 2015.

[52] S. J. Russell and P. Norvig, ”Artificial Intelligence: A Modern Approach,” 3rd Ed., ISBN-13: 978-0136042594, Pearson Education; 2014.

[53] S. Stidham Jr.and R. R. Weber, ”A survey of Markov decision models for control of networks of queues,” Queueing Systems, 13, pp. 291-314, 1993.

[54] A. A. Yushkevich and A. R. Kraiman, ”On semi-Markov controlled models with an average reward criterion,” Theory of Probability and Its Applications, 26, pp.796-802, 1981.

[55] M. Puterman, ”Markov Decision Processes: Discrete Stochastic Dynamic Programming”, 1st ed., New York, NY, USA: John Wiley & Sons, Inc, 1994.

[56] 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, 2006.