8-1-2014 12:00 AM

Online Virtual Network Provisioning in Distributed Cloud Computing Data Centers

Khaled Mefarh Alhazmi, *The University of Western Ontario*

Supervisor: Abdallah Shami, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

ONLINE VIRTUAL NETWORK PROVISIONING IN DISTRIBUTED CLOUD COMPUTING DATA CENTERS

by

Khaled Mefarh Alhazmi

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Engineering Science

The School of Graduate and Postdoctoral Studies
Western University
London, Ontario, Canada

Abstract

Efficient virtualization methodologies constitute the core of cloud computing data center implementation. Clients are attracted to the cloud model by the ability to scale available resources dynamically and the flexibility in payment options. However, performance hiccups can push them to return to the buy-and-maintain model. Virtualization plays a key role in the synchronous management of the thousands of servers along with clients' data residing on them. To achieve seamless virtualization, cloud providers require a system that performs the function of virtual network mapping. This includes receiving the cloud client requests and allocating computational and network resources in a way that guarantees the quality of service conditions for clients while maximizing the data center resource utilization and providers' revenue. In this thesis, we introduce a comprehensive system to solve the problem of virtual network mapping for a set of connection requests sent by cloud clients. Connections are collected in time intervals called windows. Subsequently, node mapping and link mapping are performed. Different window size selection schemes are introduced and evaluated. Three schemes to prioritize connections are used and their effect is assessed. Moreover, a technique dealing with connections spanning over more than a window is introduced. Simulation results show that the dynamic window size algorithm achieves cloud service providers objectives in terms of generated revenue, served connections ratio, resource utilization and computational overhead. In addition, experimental results show that handling spanning connections independently improves the results for the performance metrics measured.

Moreover, in a cloud infrastructure, handling all resources efficiently in their usage, management and energy consumption is challenging. We propose an energy efficient technique for embedding online virtual network requests in cloud data centers. The core focus of this study is to manage energy efficiently in cloud environment. A fixed windowing technique with spanning connections is used. Our algorithm, and a technique for randomly embedding nodes and links are also explained. The results clearly show that the algorithm used in this study generated better results in terms of energy consumption, served connections and revenue generation.

Keywords: Cloud computing, cloud data centers, virtual network requests, node and link mapping, window size decision, energy efficiency, power consumption

Acknowlegements

Alhamdulillah, all praise goes to Allah for the blessing of completing this thesis. It is with pleasure that I write this page to express my joy and to acknowledge all those who have supported me in successfully completing this thesis. It has been a journey of two years to complete this work and I have no doubt that I could not have been able to do it without the love and support of my family, friends, and colleagues.

First, I would like to express my appreciation to my supervisor Professor Abdallah Shami for all the encouragement and precious guidance that I have received from him at every stage of the research. The valuable discussions we had, both inside and outside campus, were always a source of inspiration to me.

It is a wonderful feeling to express my thanks to my colleagues for their help in conducting the research. I am grateful for their valuable advice which I implemented in my experiments. I am also glad to extend my thanks to the Department of Electrical and Computer Engineering at Western University for their assistance while conducting my studies in the lab.

I owe a special word of indebtedness to my father, mother, and brothers and sisters. My heartfelt thanks go out to my beloved wife for her constant inspiration, encouragement, sacrifice, and tolerance during my research work. I am extremely thankful to my wife for taking care of our sweet daughters during this journey. My family and wife's prayers during this period, I am sure, have helped tremendously in completing the research work successfully.

Last and not least, I would like to thank King Abdulaziz City for Science and Technology (KACST) - Saudi Arabia - for providing me with the required support and financial means to complete the program.

Contents

List of Figures

List of Tables

Citations to Publications

Large portions of this thesis appeared or will appear in the following publications:

- K. Alhazmi, M. Abusharkh, D. Ban, and A. Shami, "A map of the clouds: Virtual network mapping in cloud computing data centers," in *IEEE Canadian Conference on Electrical and Computer Engineering 2014 (CCECE 2014)*, (Published: May, 2014).
- K. Alhazmi, M. Abusharkh, D. Ban, and A. Shami, "Drawing the Cloud Map: Virtual network embedding in cloud computing Environment," in *IEEE Systems Journal*, (Submitted: June, 2014).
- K. Alhazmi, and A. Shami, "A Greener Cloud: Energy Efficient Provisioning for Online Virtual Network Requests in Cloud Data Centers," in *IEEE International Conference on Communications (ICC)*, (To be submitted: September, 2014).

Chapter 1

Introduction

1.1 Introduction

Cloud clients are attracted to the cloud model by the ability to scale the number and capabilities of their rented machines dynamically as their business demands require. The flexibility in payment options offered by providers is another attractive feature. However, performance hiccups may force clients to either switch to other cloud providers or even return to the buy-and-maintain model [3]. To maintain the performance in data centers at the level required by clients, cloud providers need a thorough data center management system. This management process includes synchronously handling the diverse resources of thousands of servers located in the data center along with clients' data residing on them [4]. Virtualization's role is a key in this process. Through virtualization, cloud service providers can offer computing power, storage, platforms, and services in a commodity based design without the clients needing to worry about low level implementation details. Clients can compute and connect without the overhead of resource management or network routing and control [5].

To achieve seamless virtualization, cloud providers require an efficient framework to comprehensively perform the functions of resource virtualization, allocation and scheduling in their geographically distributed network of data centers (public clouds). Cloud data centers contain a large number of servers that store, process and exchange clients' data. These systems will receive clients' network and computational resource reservation requests and perform the mapping and scheduling of these requests. In the virtual network model adopted by many providers, clients are able to reserve Virtual Machines (VMs) of multiple types that have different resource configurations [6][7][8]. Clients can also make connection requests to connect to their VMs residing in data centers. These connections will facilitate data exchange between the client VMs or between a client VM and the client's private cloud. The VMs can represent the vertices of a virtual network where each client expects to maintain the agreed upon Quality of Service (QoS) conditions, regardless of how many other clients are sharing the data center resources at the same time. A key condition here is the system's ability to allocate network resources to VMs dynamically at any moment. This virtual network mapping scenario raises certain questions: What is the optimal VM placement policy/method to serve client requests? Which network connection assignment/mapping and scheduling policy should be used? How often are arriving requests processed and mapped? How are these requests prioritized?

In this thesis, we tackle the problem of virtual network mapping in a cloud computing data center environment. Virtual network mapping in this context means finding the optimal technique/policy to serve/handle requests received continuously from clients by constructing virtual networks that contain multiple VM instances running on servers in multiple geographically distributed data centers. Virtual machine instances can be connected through virtual network links or edges that are mapped onto physical (substrate) network paths.

1.2 Contributions

In this thesis, we introduce a new virtual network mapping methodology for cloud computing data centers. Our contribution can be summarized as follows:

- Introduce a comprehensive model that covers computational and network resource requests and supports performing node mapping and link mapping.
- Aggregate the connection requests into virtual network requests and process these requests in a time window based manner.
- Investigate the effect of fixed and dynamic window sizes and the aggregation factor combined with virtual network mapping in a networked cloud environment. The objective is to achieve the optimal window size for a specific virtual network mapping problem.

- Investigate the effect of ordering connections on the performance of the system by testing multiple methods of prioritizing connections before processing.

- Investigate the effect of adding the spanning connections technique and show its effect on revenue and performance.

In addition, in this thesis, we introduces an energy-aware provisioning of online virtual network requests in a cloud computing environment. Our contribution can be summarized as follows:

- Investigate the effect of energy efficient node and link mapping strategies combined with window technique and spanning connections in energy consumption in a cloud environment.

1.3 Thesis Organization

The remainder of this thesis can be outlined as follows.

Chapter 2 provides a conceptual overview of virtualization and network virtualization. Other virtualization platforms and virtualization in data centers are also presented in this chapter. Chapter 3 presents a review of cloud computing, including definition, architecture, advantages, and challenges. Resource allocation in cloud environments is also presented in this chapter. Related past research in virtual network mapping in cloud environment are presented. Chapter 4 addresses the most prevalent resource allocation problem in the network virtualization environment, virtual network embedding, and presents a comprehensive system to solve the problem of virtual network mapping. Chapter 5 proposes an energy efficient technique for embedding virtual network requests in cloud data centers. Finally, Chapter 6 concludes this thesis with possible future work.

Chapter 2

Virtualization and Virtualization platforms

2.1 Virtualization

As the use of computers grew, a number of operating systems have emerged. In addition, large numbers of applications have been developed on these operating systems (OS). Processors have become very powerful and their power is increasing with time [9]. The cost of IT hardware has remained almost constant, even though the processing power and the size of other resources such as memory, and hard disk, etc. have increased. In many cases the processing power of computers remain unutilized as the architecture of X86 computers allows only one operating system to run at a time [10]. In order to increase the resource utilization of computers, the concept of virtualization was introduced. Virtualization allows multiple OSs to run on a single machine by removing the traditional bond between hardware and software [10]. This is done by using a software called hypervisor [11]. Each OS runs on a separate virtual machine created by virtualization. The hypervisor, also called Virtual Machine Monitor (VMM), distributes the various computing resources available such as computing power, memory, storage and networking dynamically among the various virtual machines [12].

Network virtualization allows physically dispersed resources to be run and managed as a single logical resource. The virtual networking software provides an abstraction layer for deployment and management of network resources. The virtualization software partitions the network into a number of isolated logical networks called Virtual Networks (VNs) with unique attributes, including switching, routing, bandwidth, and security. This creates a logical partition for users or applications across the network. These logical partitions then can be managed with common attributes for each logical partition [13][14]. The most important challenge in network virtualization is virtual network embedding that is inserting virtual network in both physical side of nodes such as CPU, memory, and storage and links such as bandwidth on basic network of infrastructure provider [15][16]. The main goal of this chapter is to introduce the concept of network virtualization and discus other virtualization platforms.

The key benefits provided by virtualization are [17][18]:

- Allowing optimal utilization of computing resources by using capacities that may be wasted while using a single operating system.
- Minimizing the administration of infrastructure due to reduction in number of resources used.
- Creating scalable and flexible IT infrastructure.
- Allowing the virtual machines portability of data from one virtual machine to another in case of any problems in one machine.
- Sharing the resources to reduce operating costs as well as capital cost.
- Reducing the electrical power requirement of IT infrastructure by reducing the hardware resources.
- Increasing the availability of services by improving system maintenance and allowing quick recovery in case of disasters.

2.2 Virtualization Platforms

Virtualization solutions basically simulate the hardware platform with virtualization or hypervisor acting as a middle layer between the operating system and the hardware. The virtualization solutions can be categorised as follows:

2.2.1 Desktop Virtualization

The desktop virtualization brings the desktop environment onto the user's device and makes him/her feel as if he/she is solely working on the desktop system. The system works like a client-server computing model with virtualization software running on the centralized server instead of the user's computer [19]. The user can login to the system from any place but will see the environment of normal desktop on his device. For providing desktop virtualization, the hypervisor's software is run on the base hardware. The hypervisor allows the deployment and management of virtual machines [11]. Desktop virtualization has some limitations. It dose not provide the flexibility of some OS like MS Windows that requires fast network connection with servers [20].

2.2.2 Server Virtualization

The server virtualization enables the server to run different operating systems as well as different applications at the same time on a single server machine [12]. The advantage in this case is that the client does not know that a virtual machine is being used for serving user requirement. This scheme allows a number of different applications running on different OSs to run on single server thereby saving on resources and providing optimum server resource utilization [21]. The comparison between traditional computing model and virtual machine model is depicted in Figures 2.1 and 2.2 [18].

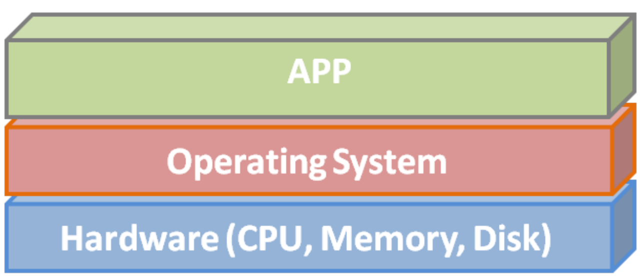

Figure 2.1: Traditional computing model.

The server virtualization software allows load balancing by moving virtual machines from one physical machine to another without causing noticeable downtime. The use of server virtualization provides efficient utilization of IT resources, space and power savings [18].

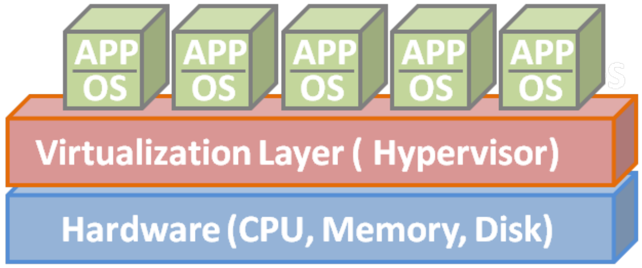

Figure 2.2: Virtual machine model using hypervisor.

2.2.3 Storage Virtualization

The storage virtualization allows the system to see multiple storage media devices as a single device [22]. As per Storage Networking Industry Association (SNIA), the storage virtualization is defined as "The act of abstracting, hiding, or isolating the internal functions of storage (sub) systems or service from applications, and host computers. Or general network resources, for the purpose of enabling applications and network-independent management of storage data" [23]. Thus virtualization provides simple interface for complex storage functions associated with storage.

The storage virtualization provides following benefits [22][23]:

- Removes the single point failures in storage systems in a cost effective manner
- Improves utilization of storage systems
- Provides guaranteed QoS that can also be measured
- Simplifies data archival and disaster recovery

2.2.4 Application Virtualization

Traditionally, applications are run on dedicated OS and hardware. The application virtualization allows the applications to run on any platform. However, the application feels that it is being run on the supported platform and gets access to all the resources required to function. Virtual applications do not require installation in the traditional sense as the applications are isolated from the operating system used and the runtime environment of OS is replaced by an application virtualization layer. However, they are executed on the virtual machine as if they are installed in a traditional manner [18]. Microsoft's Application Virtualization (App-V) is one software that coverts the applications into central management of services [24][25].

2.2.5 Network Virtualization

Network virtualization allows physically dispersed resources to be run and managed as a single logical resource. The virtual networking software provides an abstraction layer for deployment and management of network resources. The virtualization software partitions the network into a number of isolated logical networks with unique attributes, including switching, routing, bandwidth, and security. This creates a logical partition for users or applications across the network. These logical partitions then can be managed with common attributes for each logical partition [13][14].

The virtualized network architecture consists of three main components: network access control, path isolation and virtual services. The network access control grants the access rights to the users for accessing the logical partitions. The path isolation provides routing of traffic across the network for every logical partition. The virtual services provide network services such as DNS, DHCP, and IP telephony management for each logical partition [13].

The network virtualization is aimed to decouple all the network services including configuration, and provisioning from physical layer to virtualization software layer so that they can be managed centrally. The virtualization of a large network allows a network to be managed centrally with less resources [26][27].

IT organizations have started realizing the advantages, the significance and the efficacy of virtual networks over the traditional setup and are contemplating measures to migrate to this new paradigm [28]. Although researchers argue over the significance of virtual networks, they seem to agree on the advantages of virtual network over the rigidity of traditional setup [14].

Virtualization of the network involves separation or decoupling the role played by the current network service providers into two separate segments. While one segment, called the infrastructure providers, takes care of the network hardware, the other segment, called the service providers, manages the virtual networks spread across the network hardware [29].

The infrastructure providers are the managers of the underlying physical hardware. They follow a cost-effective business model where the hardware resources are offered to their clients. The popularity of the infrastructure provider depends on the quality of resources they offer and the flexibility they provide to their clients in terms of customization. The infrastructure providers offer a highly intuitive and user friendly front-end or interface that can be used by their clients to acquire a very holistic view of the available resources and manage them properly.

The service providers rent infrastructure from the infrastructure providers, create and manage the virtual network and provide service to the end-users. For example, in a mobile communication paradigm, various physical hardware utilities such as the network servers, network towers, switches, hubs, and routers. are provided by the infrastructure providers. The service providers make use of the hardware utilities and implement the virtual networks on them and provide mobile communication service to end-users. The end users pay a rent on a pay-per-use basis to the service providers and the service providers pay their rents to the infrastructure providers. The biggest incentive of this architecture comprising of multiple service providers is flexibility. The end-users now have the opportunity to select a service provider that best meet their demands and requirements.

The virtual network environment: The first major constituents of the virtual network environment are the virtual networks managed by the service providers. Each virtual network managed by individual service provider is made up of virtual network nodes that are closely interconnected by the virtual connections. The virtual networks are implemented on the underlying physical hardware obtained from either one or multiple infrastructure providers. This flexibility was absent in the traditional rigid network architecture. Figure 2.3 graphically demonstrates the virtual network environment comprising of the virtual networks managed by service providers and the network hardware managed by the infrastructure providers.

As shown in Figure 2.3, individual virtual networks VN1 and VN2 are managed by individual service providers SP1 and SP2 and they span over physical resources that are being offered by the infrastructure providers InP1 and InP2. The end users depicted as U1, U2 and U3 can subscribe to any of the service providers that best fit into their requirement [1].

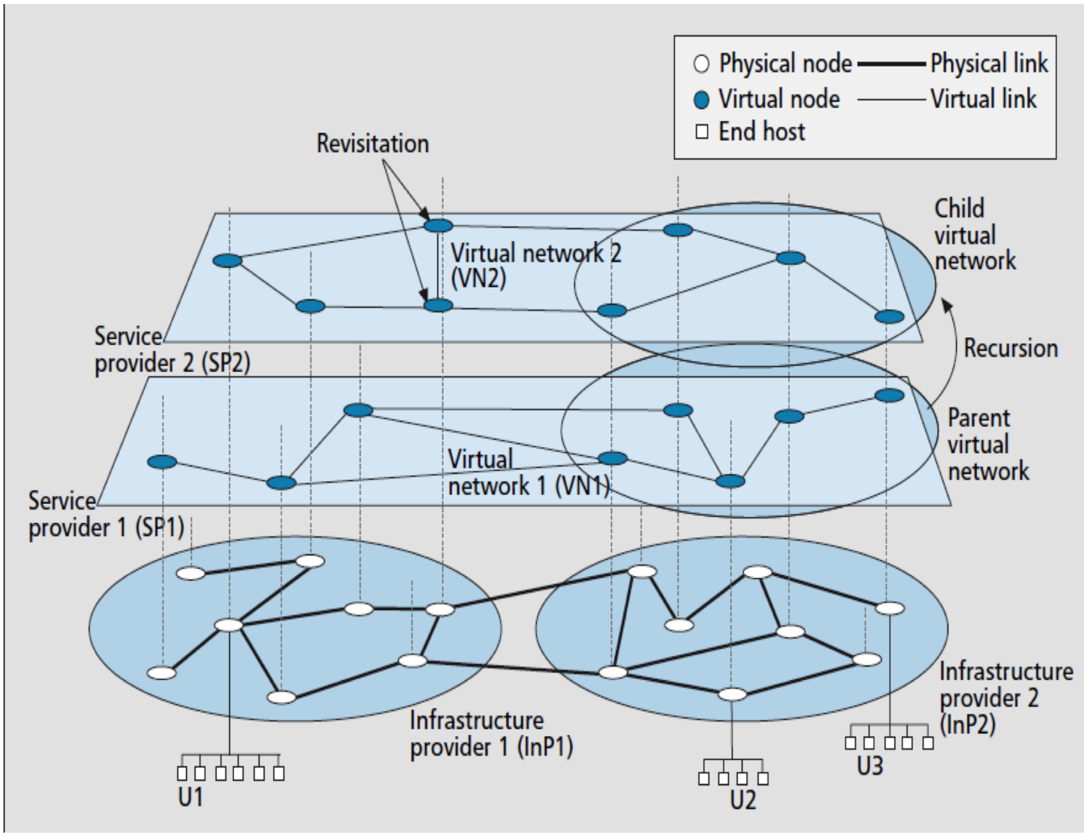

Figure 2.3: The virtual network environment [1].

The Key Principles of Network Virtualization:

As Chowdhury & Boutaba in [1] states, network virtualization stands on 4 major architectural principles: coexistence, recursion, inheritance and revisitation. Coexistence is well depicted in Figure 2.3 where the two virtual networks coexist on the common infrastructure provided by the two infrastructure providers. Recursion occurs when a child parent hierarchy is created between virtual networks. In Figure 2.3, virtual network 1 acts as a parent to virtual network 2 by allocating some of its resources to it. As a result of this recursion an inheritance is established between the parent and the child. In Figure 2.3, SP2 inherits all the characteristics from SP1 along withe the constraints that SP1 might have acquired from its infrastructure providers InP1 and InP2. Revisitation happens when a single physical node of the infrastructure providers hosts and manages more than one virtual node within a single virtual network. This possibility helps the service providers to manage highly complex tasks and streamline virtual network management. In Figure 2.3, a single node of the infrastructure provider 1 hosts two nodes in the virtual network 2.

The Promising Characteristics of Virtual Networks:

Virtualization of the networks was conceptualized with certain objectives in mind. Different researchers and study groups have studied the characteristics of an ideal virtual network. Chowdhury & Boutaba in [1] highlights the following traits that must be present in an ideal virtual network.

- Flexibility: The ablity of the service providers to implement diverse network topologies that best suits their business needs is an absolute requirement for any virtual network framework. The service providers must have the capability to define their network characteristics without having to face any hardware constraints. The arrangement should be such that the service providers are independent to offer highly customized products and services to their end users.

- Manageability: Virtual networks adopt a segmented architecture where the physical layer is separated from the network layer. This set up is much easier to maintain. In ideal virtual network, the service providers should have complete power to manage their virtual networks.

- Scalability: A virtual network environment must be scalable. Hardware infrastructure provided by the infrastructure providers should be competent and robust enough to accommodate multiple virtual networks with any trade-offs in performance and quality of service [30].

- Isolation and Stability: The multiple virtual networks' operation on the hardware layers should be able to operate independently. A standalone virtual network should be able to avoid interference from or dependence on the neighboring networks. This will ensure clients' privacy and will prevent complete network collapse due to any breakdown in a single network. However, if there is a breakdown in the hardware layer hosting multiple virtual networks, the virtual networks will be affected. An ideal virtual environment should be such that in the given scenario, the individual virtual networks should be able to recover on their own and get back to the last ideal state.

- Support to legacy networks setup: While implementing a virtual network, it may be necessary to integrate the established traditional setup with the virtual architecture. This could be critical from different perspectives such as stability, and disruption prevention.

2.3 Data Centers

A data center is a place where large number of servers are installed along with associated networking equipment to provide computing services to large number of clients. With the growth of Internet and the use of Internet for e-commerce applications, the demand for large scale and reliable hosting of web pages has brought out the large scale Internet data centers. The Internet data center traditionally provides hosting services to large number of clients. These centers are equipped with high speed Internet connectivity and reliable hardware and software. Google, Facebook, Yahoo, and Amazon store large amount of data and provide services from data centers [31][32][33]. Data centers may be owned and operated by large businesses such as Facebook, yahoo, Google or by service providers. The data center operator's business model revolves around efficient sharing of data center resources among large number of clients or applications[34]. The infrastructure facilities required for operating large scale data centers are quite large and require substantial investment as well as operating expenditure. The use of 1000's of servers in large scale data center requires large amount of space as well as electric power and air conditioning to cool down the servers. For example, a data center having 1,000 racks requires 10MW of power [35]. In order to bring down the cost of power and air-conditioning, it is necessary to optimize the resource requirement in data centers [36][37][38]. This requires optimizing the use of equipment, space utilization, power management, thermal management. The concept of "green computing" has evolved and is being researched heavily to improve power and cooling system in large scale data centers [39][40].

The key requirements of a data center are efficient utilization of resources, high availability with operations running on 24/7 basis, fault tolerance, and security. In order to provide these requirements, multiple data centers are located in geographically dispersed locations to provide geographical redundancy. The technologies such as data replication, and disaster recovery. provide necessary redundancies for mission critical applications [41].

2.3.1 Virtualization in Data Centers

Data center houses thousands of servers to provide services to clients. Each client has applications that are built on various OS platforms. In order to utilize servers efficiently, servers should be able to run applications on various platforms. In order to achieve this, server virtualization is the key technology used in data centers. Virtualization provides multiple platforms on single powerful machine to optimize resources. In addition, a data center needs to optimize all available resources to provide the customers the services required using various platforms. The data center virtualization basically involves the use of all virtualization technologies such as server virtualization, storage virtualization, network virtualization and application virtualization [42]. The advent and popularity of cloud computing has made the number of data centers grow exponentially. The use of virtualization technologies is the key to efficient utilization of resources in cloud environment.

Chapter 3

Resource Allocation in Cloud Computing Environment

3.1 Introduction

The 21^{st} century has witnessed tremendous growth in data generation, data sharing and information technology in general. Computer hardware and software have become extremely affordable and with the advent of the World Wide Web, information dissemination and knowledge sharing have become virtually seamless. This technological advancement in the IT sector has been extremely instrumental in the development of the modern cloud computing paradigm, where the typical IT infrastructure, such as the central processing unit or the data storage utility, is maintained at data centers in distributed locations. Its services are commercially leased to different organizations and service providers over the Internet, on an on-demand basis. In essence, cloud computing relieves many organizations from investing in expensive IT infrastructure, virtually making computational resources available for use in different geographical locations. This may make the business models of small and medium size corporates highly cost effective [43]. In today's fast paced working environment, there is a growing expectation for the availability of resources on the go. Instead of continuing to use a fixed computer system with enough processing capabilities, modern day corporate executives would prefer light systems with Internet capabilities through which they could access the same or even more powerful computing resources hosted in the cloud. Cloud, in a true sense, provides independence and assures support in the face of changing demands. If demands are high, more computational power could be acquired and if demands are low, operational costs could be reduced by releasing computational resources back to the cloud. Therefore, the very philosophy of cloud computing epitomizes mobility, flexibility and adaptability, traits that are extremely relevant to modern day businesses.

The cloud computing model adopts a two tier structure where the first tier is made up of organizations who manage the cloud computing infrastructure. The second tier is made up of the service providers who rent the cloud computing infrastructure to provide service to the target customers [43][44]. The infrastructure providers (IPs), who rent out computing resources to the service providers, give them the flexibility and independence to operate without having to invest in expensive computational resources. The infrastructure providers rent out their IT resources to the service providers in different ways and scenarios as enumerated below:

- *Infrastructure as a Service scenario (IaaS).*

  The computational resources maintained by the infrastructure providers are virtually allocated into different segments or virtual systems, which are then rented out to the service providers. The software utilities that are required to provide the services to the end user are being managed by the service providers in the allocated spaces.

- *Platform as a Service scenario (PaaS).*

  In this scenario, the infrastructure providers lease the software platform to service providers instead of the virtual infrastructure.

- *Software as a service scenario (SaaS).*

  In this scenario, the end users are offered various software services that are based on the cloud infrastructure. The end-users can make full use of different software utilities without having to install them locally in their machines. A good example of this scenario is Google Docs [45].This offering is a shift from the traditional system of purchasing computer softwares and installing them in a system, and then maintaining a never ending process of updating and licensing. SaaS offers on demand pay-as-you-go service to the client and there is absolutely no need to worry about licenses and version releases. It is highly cost effective; clients no longer have to invest in expensive hardware to fulfill specific software requirements [46].

In essence, the cloud computing paradigm is about the migration of computing architecture and hardware resources to the virtual space, maintained by the centralized unit and its services made available to clients and end users over the network. This may reduce capital costs associated with acquiring expensive hardware and software and make the business models of numerous small and medium sized enterprises highly cost effective.

Cloud computing has been well received in the 21^{st} century IT industry in general and has found appeal even amongst the non-technical businesses and organizations. This clearly demonstrates the significance and usability of the technology for a highly varied range of requirements.

3.2 Resource Allocation in Cloud Computing

One of the biggest advantages of cloud computing architecture is that it is highly scalable. In other words, computational resources could be added or released based on the changing market demands. Consequently, it seems evident that the way the computational resources are allocated by cloud providers determines their profitability in the business and completion of tasks within specified deadlines. In the present market scenario, the demand for cloud computing solutions is high and efficient resource allocation is extremely crucial and challenging to profitably fulfill the requirement of clients. Resource allocation becomes even more critical when the client enters into a Service Level Agreement (SLA) with the cloud provider, and it becomes imperative for the provider to honor the agreement in totality.

There have been numerous studies to define efficient resource allocation algorithms. The major cloud providers operating today give many options to the clients to acquire resource allocation for their tasks. Previous studies on resource allocation have focused on capabilities of cloud infrastructure or virtual machines to efficiently complete real time jobs and ideal virtual machines for optimal energy usage in the cloud facilities. However, there is not much information or guidance to help the clients choose a scalable cloud computing package that is ideal for them in terms of cost effectiveness and efficiency [47]. From the perspective of the

Infrastructure as a Service scenario in the cloud computing setup, Table 3.1 below illustrates the different virtual machine renting options offered by Amazon EC2 cloud services [31].

Table 3.1: Client and service provider input providers.

Virtual Machines type	*Compute Units*	*Memory*	*Storage*	*Cost per hour*
High Memory	6.5	17.1 GB	420 GB	$ 0.69
Standard	8.0	15.0 GB	1690 GB	$ 1.04
High CPU	20.0	7.0 GB	1690 GB	$ 1.24

3.3 Cloud Computing: Its different components, advantages, weakness and types

3.3.1 Cloud Computing Architectures

The general architecture of a typical cloud computing setup is composed of four primary layers. Figure 3.1 is a schematic representation of the different layers of the cloud computing setup [43].

- *The hardware or data centre layer.*

  As the name suggests, this layer is composed of hardware resources such as servers, hubs, routers and server cooling utilities.

- *The infrastructure layer.*

  The infrastructure layer is composed of the virtual segments of storage and computational resources that are offered to service providers. The infrastructure layer is created by virtually partitioning the physical hardware resources.

- *The platform layer.*

  The platform layer is where the operating system resides and houses different application framework utilities such as databases and .NET frameworks.

- *The application layer.*

  The application layer resides at the top of the cloud computing architecture and offers the clients with the actual software and storage utilities that can be used for processing, disseminating and storing data.

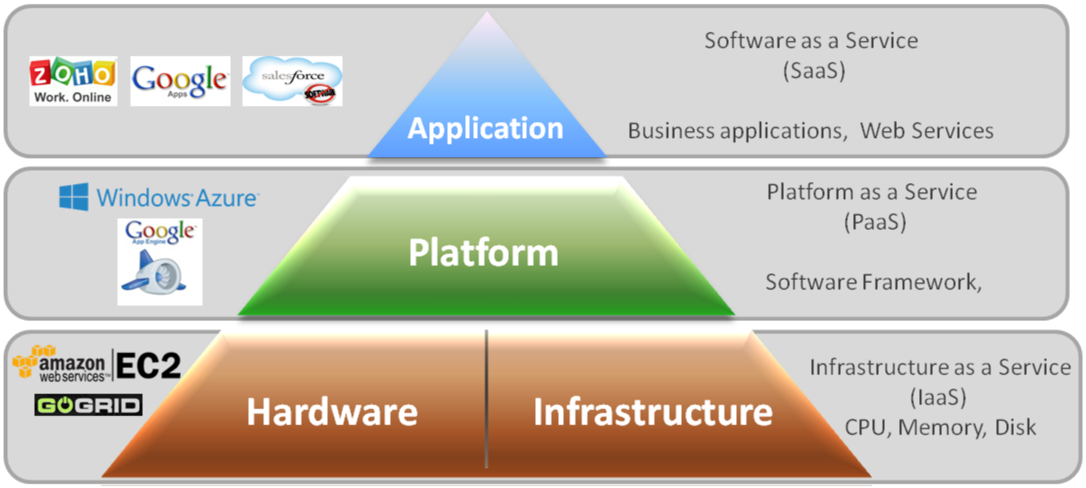

Figure 3.1: Layers of cloud computing architecture.

3.3.2 The Evolution of the Cloud set up

The concept of the cloud first appeared in the form of TCP/IP enabled networks, which were set up in institutions and laboratories to exchange information and resources in-house. As science progressed and collaborations increased, networks of different organizations started communicating between themselves, and this marked the emergence of the World Wide Web. Hyper Text Markup Language (HTML) became the new paradigm to facilitate sharing of information over the network. As the Internet became the new order of information sharing in the 21^{st} century, the concept of grid computing slowly emanated from the increasing needs of modern businesses. Grid computing is all about sharing high performance computing resources and services across the network to perform specialized tasks. The paradigm of cloud computing is the latest offering of this development trend and is being hailed as a revolutionary concept, which might change the way people utilize modern IT resources. Figure 3.2 illustrates the evolution of cloud computing [2].

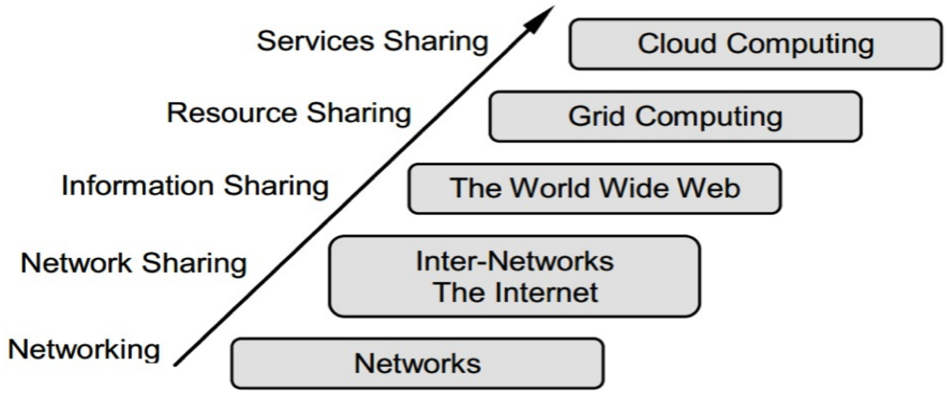

Figure 3.2: Evolution of cloud computing [2].

3.3.3 Technical and Economic Advantages of Cloud Computing Paradigm

Even though cloud computing developed late compared to the emergence of Information Technology in general, it has attained significant heights and revolutionized the business models of major corporates such as Google and Amazon [43].

According to [43], cloud computing solutions not only reduce business capital costs, but helps to reduce the issues experienced by clients when using computational processing services. The critical benefits and advantages associated with cloud computing are shown below [43]:

- There is no capital investment in IT hardware and software. Services offered by the cloud can be procured instantly with an upfront payment when there is an urgent need.

- Cloud services and utilities are easily accessible because they are available on the Internet and can be utilized from different geographical locations 24/7. The cloud services are also highly compatible with different electronic devices such as laptops, notebooks, PDAs and even mobile smart phones.

- Since the service providers and clients using cloud services do not have to maintain hardware resources, they do not face the risk of hardware failures, and they do not need not spend resources on cite staff training.

3.3.4 Challenges

However, there are some potential drawbacks that enterprises might have to deal with when they shift from the traditional IT setup to the cloud. These drawbacks are described below:

- *Data Integrity, Privacy and Security issues.*

  Data associated with strategic business vision and business logic are highly sensitive for any organization. Similarly, there is a huge responsibility for the business organizations to maintain privacy and the integrity of their customer data. When an organization maintains its own IT system, they can be more certain about the security of their data as they are able to monitor and update their security protocols from a close proximity. On the other hand, when they use cloud services they are delegating the job of maintaining the security and privacy of their data to a third party organization, which they cannot monitor themselves. That is why data security is a major issue, especially with large organizations who deal with highly sensitive data such as customer information, business secrets, and patent details. It has been observed that present cloud computing solutions do not have all the means to comply with all the security and privacy requirements associated with such type of data [48].

- *Compatibility issues.*

  Software and data compatibility are common in modern organizations when they try to upgrade their legacy systems. However, in a cloud environment, the scale of the problem gets highly inflated. This is mainly because of the fact that service providers and clients have no control whatsoever to resolve the compatibility issues, and they are required to rely on the third party cloud vendors. Every prominent cloud service provider has its unique format of storing and managing data, and in the event of a client wishing to migrate from one cloud system to another, data compatibility is a major issue. The problem is further aggravated by the data transfer limitations with regards to speed [48].

- *Lack of robust contractual agreements.*

  In the present market scenario, there are not any effective service level agreements offered by the cloud providers to their clients. In other words, there are very few safeguards to ensure that the clients exactly get what they are promised. Even when a service level agreement exists, it is not very effective to guarantee the service that is being promised. Furthermore, given the commercialization of the cloud services, it has been observed that cloud service providers fail to properly appreciate the vital requirements such as continued high computing performance, responsiveness and pro-activeness that are paramount for any organization. It has also been observed that some well-established cloud providers promise a certain level of performance to attract clients, and subsequently offer discounts in their services to compensate for their inability to fulfill the promised level of service. If the promised level of performance is actually a vital requirement for the business model of the client, they are bound to lose revenue and market credibility, which might even lead to complete collapse of their business model [48].

- Cloud performance instability: Even though cloud computing services are being hailed by small and medium sized enterprises for meeting their changing IT demand with agility, issues related to the stability and performance of the cloud systems during peak demand hours vary considerably. Studies carried out in Australia have demonstrated that performance and stability of the cloud systems offered by software giants such as Google and Microsoft show significant variation during peak hours. If the cloud systems fail to handle the peak load and goes down, many enterprises who are completely dependent on the cloud for their entire IT needs will suffer considerably [48].

3.3.5 Inherent Characteristics of Cloud Computing

Some aspects of cloud computing that sets it apart from traditional computing services are discussed below [43][44]. These features are highly instrumental in imparting the technical advantage to the cloud services.

- *Multi-tenancy.*

  A cloud provider rents computational or storage services to multiple clients. These services can be maintained in a single data center. The different services are managed together by the cloud infrastructure providers and clients, and given the layered architecture of a typical cloud set up, the management of services becomes relatively easy. The infrastructure providers and the clients only need to concentrate on the layer where the service is located.

- *Shared resource pooling.*

  The cloud services utilize a pool of computational resources that are dynamically allocated to different clients based on specific requirements. That way the cloud infrastructure providers are able to efficiently utilize their resources in a cost-effective manner and curtail management and operational costs.

- *Wide geo-distribution of data centers, and network access.*

  Data centers that offer cloud services are widely distributed across different geographical locations. Through such a diversified distribution, cloud service providers are able to offer agile and efficient services to the clients. For example, a client located in Seattle would be best served by a data center located in Seattle or even California instead of a center located in Europe. Furthermore, modern cloud services are almost entirely accessible through the Internet and can be utilized by any electronic device that can go online.

- *Dynamic resource provisioning or allocation.*

  Modern cloud computing solutions allow instant dynamic allocation and withdrawal of computational resources to the clients based on the changing requirements. This also makes the cloud model highly cost effective and cuts down operational costs. In essence, a modern cloud computing environment is extremely dynamic and flexible and is highly suitable for modern businesses in today's volatile market spaces.

3.3.6 Current Trends and some Prominent Cloud Players

Cloud computing has widely been implemented and adopted by big corporates such as Microsoft and Google. Cloud computing has also improved the way computational resources and services are managed, utilized and offered to clients over the network. It has given a brand new shape to the business models of thousands of enterprises across the globe and also enabled cloud infrastructure providers to maximize the utility of their hardware and software resources and reduce operational costs. However, there is still much scope for progress, and some pertaining issues associated with data privacy, data security, computational and network resources management and power management.

Some currently available commercial cloud services are illustrated in Table 3.2 below [43].

Table 3.2: Commercial cloud utilities.

Cloud Provider	Amazon EC2	Windows Azure Google App Engine	Google Apps Saleforce.com
Type of service provide	Infrastructure service	Platform service	Software service

3.3.7 Different Types of Clouds

The decision of an organization to migrate to a cloud setup is influenced by different factors. Every organization has different reasons to move their IT infrastructure to a cloud environment. While some clients might be looking to cut down on their day to day IT operational costs, others might like to project high quality and security as their unique selling points. To cater to these diverse demands and requirements, there exist four different kinds of cloud setup as described below [49][43][50][51]:

- *Public clouds.*

  In this type of clouds, the client or the service provider does not need to invest anything to create a capital infrastructure base and the responsibility to deal with the risk factors associated with any modern IT setup is entirely transferred to the infrastructure provider. The service providers offer their resources as service to the general public over the Internet. However, this type of cloud is too general and is incapable of catering to niche requirements. The data security in this type of cloud is not very robust as well.

- *Private clouds.*

  This type of cloud architecture, as the name suggests, caters to an exclusive group of clients mostly within a single organization. The qualities associated with this type of cloud include safety, reliability and high performance. However this type of cloud might require investments from the users.

- *Hybrid clouds.*

  This type of cloud brings together the best of public and private clouds. While they promise robust data security like private clouds, they also retain the high level of flexibility and adaptability of public clouds. However, defining the architecture of a hybrid cloud is quite tricky with regards to the maintenance of an ideal ratio between public and private virtues.

- *Community cloud.*

  This type of cloud caters to a group of organizations that share similar values and visions. Community clouds are constructed and used by a community of organizations and the cloud service provider could be either a commercial one or even a member of the community.

3.4 Resource Allocation Algorithms in Cloud Computing

When an organization migrates to a cloud environment, its entire IT setup is transferred to the cloud. On one hand, it significantly reduces the time, effort and cost involved with maintaining a complete IT infrastructure, but on the other hand the organization becomes entirely dependent on a third party vendor.

Efficient allocation of cloud resources (computational and network) shape the profitability and cost-effectiveness of the business models of both the clients and the cloud service providers alike. While proper resource allocation and management at the cloud infrastructure with regards to issues such as power management and ventilation is highly beneficial for them, appropriate acquisition of cloud computational and network resources by a client makes all the difference between profit and loss.

3.4.1 Resource Allocation in Public Cloud

Earlier studies that focused on the resource allocation in multi-core or multi-processor CPUs did not correlate well with the issues of resource allocation in highly scalable cloud systems [47]. While multi-processor CPU units have a fixed number of processors, cloud computing services could provide numerous instances of virtual machines to deal with a particular task, depending on the need.

In the absence of proper selection guidance, if a user purchases more virtual machine instances than required and the task is completed earlier than expected, the idle virtual machines could have been used to do other tasks for the remaining period of rent time. As a result, it is very obvious in this situation that proper resource allocation will make all the difference between success and failure. To help clients choose the most profitable combination of virtual machine instances, authors in [47] proposed a resource allocation algorithm applicable in the Infrastructure as a Service scenario of cloud setup, where clients get the opportunity to make the resource selection.

The proposed polynomial time algorithm first identifies future tasks that might overlap with the current task before it is over and then allocates resources in such a manner that all the tasks can be finished on time through the use of adequate virtual machine instances. The algorithm starts by allocating a virtual machine with the lowest speed for a particular task and then determines if a virtual machine with a higher processing speed could benefit future overlapping tasks. While high speed virtual machines are being considered, the overlap time between the current task and future tasks is computed. This overlap period is defined as the time period for which the virtual machine assigned to the current task is available for use to the future tasks. This algorithm is designed from the perspective of the cloud user. It seeks to assist them in making the most ideal or appropriate resource allocation, for not just their current real-time task, but also the future tasks that might overlap during the period for which cloud resources are being rented.

On similar lines, Kim et al. [52] studied the scheduling of jobs in the cloud that have real time constraints. In their study, the real time constraints are mentioned in the service level agreements between the cloud providers and the clients. In this kind of scenario, clients do not get the opportunity to decide on the number of virtual machines that needs to be selected for the given tasks, but the cloud service providers who makes the virtual machine selection. The study focused on the aspect of efficient power management while the virtual machines are being allocated to different tasks as per service level agreements. Consequently, it could be observed that the study was cloud-provider-focused rather than client-focused.

3.4.2 Virtual Network Mapping in Cloud Environment

Massive data centers where the IT infrastructure is located, forms the heart of modern day cloud computing services. Furthermore, the data center being centrally located the cost of services in terms of money and time is high for those clients who are located far away. This led to the development of the concept of distributed data centers that are spread across different geographical locations. These distributed data centers are interconnected with the use of a physical network to create infrastructure of cloud. The physical network is used in order to get high speed and bandwidth resources [53]. This model makes full use of its diversity, with regards to the distribution of data centers across different regions, to offer highly cost effective services to its clients.

Virtual network mapping is the main resources allocation problem in the domain of network virtualization. In network virtualization, the inherent resource allocation challenge involves cost effective allocation of a virtual network over the physical network in such a manner that all the client requirements are met with minimum resource usage.

To realize the potential of network virtualization it is important that virtual network embedding techniques are exploited to accurately map every virtual network to the physical nodes and links.

Endo et al. [54] proposed the concept of distributed cloud where the data centers are located in diverse geographical locations and cloud providers hire the services of dedicated communication providers to facilitate communication between the data centers and the clients. The researchers anticipate that the diversely located data centers will immensely benefit Network Virtualization, which essentially is a system that supports "multiple coexisting heterogeneous network architecture from different service providers, sharing a common physical substrate." [54].

The authors addressed the issue of resource allocation in distributed cloud as follows [54]:

- *Resource modeling.*

  Cloud resource modeling (resource description) enables cloud providers to efficiently manage the cloud IT infrastructure. Different optimization algorithms and operations that use a cloud set up to increase efficiency and cost effectiveness depend on proper resource modeling.

- *Resource offering and treatment.*

  Once resource modeling is complete and the cloud infrastructures are properly managed, the cloud services providers can provide an interface in the form of a resource allocation system to the user. It is the resource allocation system that ensures that all the client requirements are properly addressed as per the conditions defined in the service level agreements between clients and cloud service providers.

- *Resource discovery and monitoring.*

  Resource discovery and monitoring constitutes the third pillar through which efficient resource allocation is achieved in a distributed cloud environment. Resource discovery involves identification and localization of ideal cloud resources from different available sources through the use of a discovery framework to meet the demands of the clients within a specified time constraint. Resource monitoring, on the other hand, is instrumental in ensuring that cloud resources are continuously available to the clients on demand anytime anywhere. This necessitates resource monitoring to run as a continuous process, and assist in the overall resource optimization process of the cloud setup through intelligent decisions with regards to allocation and de-allocation of available resources. Resource monitoring also helps identify the advantages and the disadvantages of a discovered resource and its applicability to fulfill specific client requirements.

- *Resource selection.*

  Once appropriate and cost effective resources are identified, the most ideal resource may be selected. Resource selection endeavours to find a resource that completely fulfils all the client requirements in a timely manner. However, the selection of the most ideal resource from a list of discovered resources is very complex and is influenced by different aspects of a cloud computing setup. Therefore, service providers should use optimization algorithms to select the resources that will provide maximum optimization of the cloud infrastructure functioning.

The steps described above help cloud providers allocate resources in a distributed cloud system in an efficient manner. The segmentation of the resource allocation process increases efficiency and allows cloud providers to monitor and manage allocation in a more effective manner.

Cloud computing is attracting more and more attention recently, resulting in a large amount of research on techniques of network virtualization for virtual network embeddings. The larger part of the research deals with efficient resource embedding of virtual network requests on resources that are physically available. The problem of efficient virtual network embedding is currently modeled as a question of mathematical optimization, dealing with minimizing resource cost during the implementation of virtual network embedding [53].

In [55], optimal networked cloud mapping is formulated as a Mixed Integer Programming (MIP) problem, with the objective focusing on cost efficiency, whereas the constraints focus on maintaining the quality of service conditions. A method is subsequently proposed for the efficient mapping of resource requests onto a shared substrate network connecting various islands of computing resources. A heuristic algorithm is adopted to address the problem. A Java-based simulator is also introduced to evaluate the performance of the solution. Moreover, a proof-of-concept realization of the proposed schema, is deployed over the European future Internet test-bed FEDERICA.

Two algorithms are proposed in [9] that offer better correlation between the phases of node mapping and link mapping. These algorithms are D-ViNE, or Deterministic Virtual Network Embedding, and R-ViNE, or Randomized Virtual Network Embedding. In these algorithms, the virtual nodes are mapped to substrate nodes, allowing virtual link mapping to physical paths in the succeeding phase. In the physical network graph, a corresponding meta-node is extended for every virtual node, while each meta-node is also connected to a specific physical node subset [56].

Mixed-integer programming formulation is used for solving embedding problems with constraints on meta-edges, as well as for linear constraints on substrate network edges. Using mixed-integer programming for optimal virtual network embedding, it also becomes an NP hard problem. This approach is used to relax the integer program in order to obtain a linear programming formulation, solved in polynomial time. Rounding techniques, deterministic and randomized on the solution of linear program, are used to approximate the binary variable values in original mixed integer programming. While virtual nodes are only mapped once, a multi-commodity flow algorithm is used for mapping virtual links on substrate networks among mapped virtual nodes. This process is solved in polynomial time, assuming that path splitting is already supported by the substrate network [56]. Along with bandwidth constraints, each virtual link is treated as a commodity, which consists of a pair of meta-nodes. This mapping results in an optimal flow for commodities that is equal to virtual link mapping in an optimal way [56].

Virtual network embedding is a key issue in network virtualization, and can be divided into two categories: intra-domain virtual network embedding, and inter-domain virtual network embedding. An efficient algorithm for intra-domain virtual network embedding is proposed in [15], that increases the utilization of resources through permission for path splitting and immigration in the substrate network.

Some topology-aware designs for node ranking, as well as meta-heuristic based algorithms of virtual network embedding, were presented for revenue-maximization of large numbers of virtual network requests. In inter-domain virtual network embedding, used for enabling virtual network provisioning over a large area, a framework for inter-domain virtual network embedding in a decentralized manner was presented. In order to accommodate a high range of virtual network requests, an approach named Max-flow/Min-cut was used to split the virtual network requests into sub-requests, with these used in turn exact mapping with processes [15].

Vinothina et al. [57] proposed the following set of factors that should be avoided by an efficient resource allocation strategy:

- *Resource contention.*

  There is resource contention when a single resource or cloud utility is being accessed by two different applications at the same time. This can occur when there is no proper scheduling or allocation of resources and can lead to the slowing down of cloud services.

- *Scarcity of resources.*

  There is scarcity of resources when the cloud setup dose not have enough resources to deal with the workload.

- *Resource fragmentation.*

  There is resource fragmentation when valuable resources lie around in a highly disorganized manner. If there is resource fragmentation, a task may be deprived of a resource even when it is available for use. This might affect the cost effectiveness of the business model of the cloud provider and as well as profitability.

- *Over-provisioning.*

  There is over-provisioning when more resources are being assigned to a task than its requirement. This will once again affect the profit margins of the cloud service provider.

- *Under-provisioning.*

  There is under-provisioning when a task does not get adequate resources necessary for its completion. This may lead to dishonoring of the service level agreements between clients and cloud service providers.

A proper resource allocation strategy takes input from both the clients and the cloud service providers to use the most optimal resource allocation. It is often possible that requirements determined by clients may lead to over-provisioning of resources and requirements proposed by cloud providers may lead to under-provisioning. Table 3.3 shows the standard inputs from the cloud service providers and clients [57].

Table 3.3: Client and service provider input providers.

Parameter	*Provider*	*Customer*
Provider offerings	X	-
Resource status	X	-
Available resources	X	-
Application requirements	-	X
Agreed contract between customer and provider	X	X

Chapter 4

A Map of the Clouds: Virtual Network Mapping in Cloud Computing Data Center

4.1 Introduction

Cloud clients are attracted to the cloud model by the ability to scale the number and abilities of their rented machines dynamically as their business demands require. The flexibility in payment options offered by providers is another attractive feature. However, performance hiccups can soon push clients to either switch to other cloud providers or even return to the buy-and-maintain model [3]. To maintain the performance in the data center at the level required by clients, cloud providers need a thorough data center management system. This management process includes synchronously handling the diverse resources of thousands of servers located in the data center along with clients' data residing on them [4]. Virtualization's role is key in this process. Through virtualization, cloud service providers can offer computing power, storage, platforms, and services in a commodity based design without the clients needing to worry about low level implementation details. Clients can compute and connect without the overhead of resource management or network routing and control [5].

To achieve seamless virtualization, cloud providers require an efficient system to comprehensively perform the functions of resource virtualization, allocation and scheduling in their geographically distributed network of data centers (public clouds). Cloud data centers contain thousands of servers that store, process and exchange clients' data. These systems will receive clients' network and computational resource reservation requests and perform the mapping and scheduling of these requests. In the virtual network model adopted by many providers, clients are able to reserve Virtual Machines (VMs) of multiple types that have different resource configurations [6][7][8]. Clients can also make connection requests. These connections will facilitate data exchange between the client VMs or between a client VM and that client's private cloud. The VMs can represent the vertices of a virtual network where each client expects to maintain the agreed upon Quality of Service (QoS) conditions, regardless of how many other clients are sharing the data center resources at the same time. A key condition here is the system's ability to allocate network resources to VMs dynamically at any moment. This virtual network mapping scenario raises certin questions: What is the optimal VM placement policy/method to serve client requests? Which connection assignment/mapping and scheduling policy should be used? How often are arriving requests processed and mapped? How are these requests prioritized?

In our work, we tackle the problem of virtual network mapping in a cloud computing data center environment. Virtual network mapping in this context means finding the optimal technique/policy to serve/handle requests received continuously from clients by constructing virtual networks that contain multiple VM instances running on servers in multiple geographically distributed data centers. Virtual machine instances can be connected through virtual network links or edges that are mapped onto physical (substrate) network paths.

The problem of virtual network mapping can be seen in the literature in different forms, each with the goal of developing the most efficient mapping (embedding or assignment in some sources) technique.

In [58], a resource scheduling model for cloud computing data centers is presented. In this model, requests arrive from clients either to reserve a VM, connect two VMs together or connect a VM to a private cloud. VM placement techniques and connection request scheduling techniques are evaluated. Both computational and network resources are independently considered. Connection requests are processed one by one without employing any aggregation policy.

In [59], the authors introduce another virtual network embedding methodology. Virtual network embedding is divided into node mapping and link mapping. The proposed virtual network embedding algorithm called ViNEYard coordinates node and link mapping stages so as to enhance the embedding efficiency.

The authors also propose a window-based virtual network algorithm called the WiNE. The simulation results show that combining the Virtual Network Requests(VNRs) and processing them in groups at the end of the time interval, called a window, is cost effective in terms of resources. As virtual network requests come in, the WiNE algorithm collects them in batches for a given time period (window) and calculates the potential revenue of each request, and then assigns higher priority to requests with higher potential revenue. Every virtual network request is active during a limited time frame. If this time frame (request lifetime) is over before the end of the window, the request is dropped or ignored. The optimal window size analysis for a set of requests was not addressed in this work.

The authors of [60] propose an embedding algorithm that accumulates multiple virtual network requests during a particular active window then processes them according to their specific requirements. The virtual network requests that cannot be addressed in a particular time window are inserted in a queue and then assigned accordingly in the subsequent windows. The request is dropped only when its maximum waiting time in the queue has passed without processing the request. Path splitting and migration features are also discussed.

In [55], optimal networked cloud mapping is formulated as a Mixed Integer Programming (MIP) problem, with the objective focusing on cost efficiency, whereas the constraints focus on maintaining the quality of service conditions. A method is subsequently proposed for the efficient mapping of resource requests onto a shared substrate network connecting various islands of computing resources. A heuristic algorithm is adopted to address the problem. A Java-based simulator is also introduced to evaluate the performance of the solution. Moreover, a proof-of-concept realization of the proposed schema, is deployed over the European future Internet test-bed FEDERICA.

In [61], the authors use the difference in bandwidth requirements of applications and the cost of dividing flows into different electronic channels as a motive to come up with a new design. Wavelength Division Multiplexing (WDM) networks are implemented because these networks "Employ Optical Add/Drop Multiplexers (OADMs) which allow a wavelength to either be dropped at a node or optically bypass the node's electronics." The WDM mechanism proposed in the paper is traffic grooming. This mechanism aggregates applications that have less bandwidth requirements on shared wavelength channels in order to maximize network resource utilization. A sliding traffic scheduling model is also proposed. Scheduling does not depend on the connection lifetime in this model. Moreover, the authors present a time window based technique in which the network bandwidth requests are divided into multiple time windows. Spanning requests (or requests that are long enough to span over two time windows or more) can be scheduled in an alternative window if no network resources are available to serve these request in the current window.

In cloud computing environments, central network controllers have to deal with the task of mapping numerous virtual network mapping requests within short periods [62] [63]. To guarantee QoS of network connections in a networked cloud computing environment, there is a need for a centralized network controller.* An example of this scenario can be seen with the increasingly popular software defined networking (SDN) technology [64]. SDN controllers are responsible for centrally mapping clients' connection requests/flows. To put this to perspective, a typical SDN controller can support up to 10^5 flows/s in the optimal case [65]. The sheer volume of requests arriving at the SDN controller can constitute a substantial performance handicap. The computational overhead will increase significantly if these requests are processed individually as they come. Therefore, the need arises for connection request aggregation. Considering aggregation as a solution to computational issues brings to the forefront many design questions. A complete methodology needs to be constructed. The most important decisions include aggregation technique, aggregation factor, window size, and request prioritizing. We endeavor to answer these questions and others in this work.

In this chapter, we introduce a new virtual network mapping methodology for cloud computing data centers. Our contribution can be summarized as follows:

- Introduce a comprehensive model that covers computational and network resource requests and supports performing node mapping and link mapping.

- Aggregate the connection requests into virtual network requests and process these requests in a time window based manner. This decreases the computational load on the central controller.

- Investigate the effect of fixed and dynamic window sizes and the aggregation factor combined with virtual network mapping in a networked cloud environment. The objective is to arrive at the optimal window size for a specific virtual network mapping problem.

- Investigate the effect of connections' order on the performance of the system by testing multiple methods of prioritizing connections before processing.

- Investigate the effect of adding the Spanning connections technique and show its effect on revenue and performance.

The following sections are organized as follows: A detailed problem description is given in Section 4.2. Section 4.3 presents the different time window selection techniques, and the spanning connection technique. The simulation environment and results are discussed in Sections 4.4 and 4.5.

4.2 Model Description

When creating an efficient resource allocation methodology, it is critical that the resources involved in the cloud infrastructure are accurately modeled. Another key factor is that any cloud management system should be continuously aware of infrastructure operational status. In the scenario we are investigating, clients will reserve VMs for a fixed or an open amount of time. The specifications of connection requests, including the source, destination, and lifetime are not known in advance. The substrate network consists of data centers and client nodes. Each data center has a number of servers that can host multiple virtual machines without exceeding the server capacity. Multiple types of VMs with different resource configurations are available. After reserving the VM, a client may request connections between the VM and a client node. In a connection request, the client typically defines the source, the destination, the requested (preferred) start time, connection lifetime (duration), requested capacity units (VM specifications), and the required bandwidth. Table 4.1 shows a sample of the input data for the problem in the form of connection requests. To start solving this problem, client requests are aggregated into Virtual Network Requests (VNRs) based on a configurable aggregation factor. Next, serving connection requests is abstracted as a virtual network mapping problem where nodes represent sources and destinations and edges are virtual links between these nodes. Each virtual link represents a physical network path from the source to the destination. Moreover, each VNR will be assigned to a time window based on the requested start time. Therefore, we can abstract a single window as a set of virtual network mapping requests during the time period this window represents. In the case of a VNR lifetime that is long enough to span over more than a window, the VNR will be assigned to all of these windows. We call these requests spanning requests. Subsequently, the system performs node and link mapping for virtual network requests in a specific window on the substrate network.

A key decision here is choosing the window size that yields the best performance and revenue values. In the following section we discuss the techniques we use to decide the window size. The window size selection will affect performance measures like request acceptance ratio, allocation computational overhead, resource utilization and cloud provider revenue.

Table 4.1: An example of a set of connection requests

Connection requests	Source	Destination	CPU	Memory	Storage	Bandwidth	Requested start time	Duration
C1	Client node 2	VM79	12	15	8	26	1	2
C2	Client node 12	VM170	2	19	8	33	1	5
C3	Client node 12	VM68	11	13	4	29	1	6
C4	Client node 0	VM163	1	6	11	22	3	5
C5	Client node 7	VM146	3	15	19	17	3	7
C6	Client node 13	VM28	7	3	11	13	5	3
C7	Client node 7	VM92	11	13	17	38	5	1
C8	Client node 13	VM143	9	15	9	9	7	15
C9	Client node 11	VM37	14	8	8	39	9	10
C10	Client node 9	VM5	5	6	10	31	11	4

Table 4.2: Intersection table for connections in table 4.1 graph

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	0	1	1	0	0	0	0	0	0	0
C2	1	0	1	1	1	1	1	0	0	0
C3	1	1	0	1	1	1	1	0	0	0
C4	0	1	1	0	1	1	1	0	0	0
C5	0	1	1	1	0	1	1	1	1	0
C6	0	1	1	1	1	0	1	1	0	0
C7	0	1	1	1	1	1	0	0	0	0
C8	0	0	0	1	1	1	0	0	1	1
C9	0	0	0	0	1	0	0	1	0	1
C10	0	0	0	0	0	0	0	1	1	0

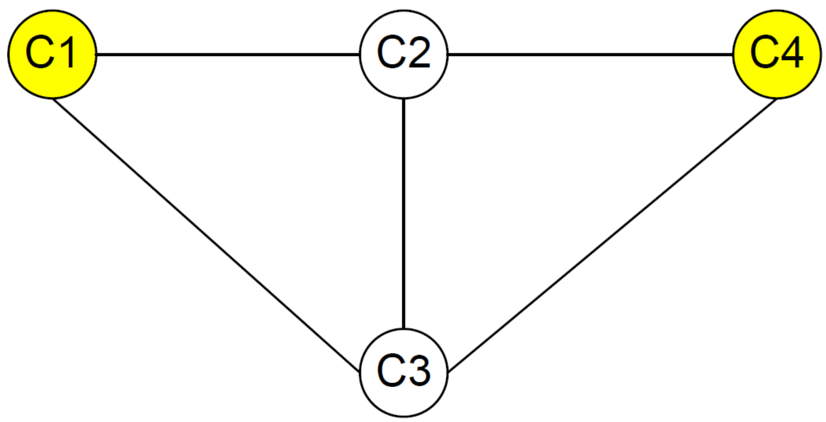

Figure 4.1: Interval graph for 4 connections.

4.3 The Virtual Network Mapping Solution

4.3.1 Virtual Network Mapping

Node Mapping

Before mapping the VNR on the substrate network, the VM needs to be allocated the required computational resources. The VM will be placed on a server with sufficient resources. The node mapping algorithm used is a variation of the node distance algorithm used in [58]. The algorithm adds the advantage of insuring the VMs are distributed widely and this leads to less connection request collision.

Link Mapping

The next step is to map the virtual link onto a physical network path. This path has to be a valid path between the source VM and the client node or private cloud. In addition, the mapping has to satisfy the bandwidth requirements by the virtual link on all the physical links that construct the path. The algorithm used is a greedy algorithm that maps the virtual link onto the shortest path, provided that path has the requested bandwidth available. The advantage of using this algorithm is that it takes less time to calculate the path which decreases the computational overhead.

4.3.2 Time Window Selection Techniques

Fixed Window Technique

In this technique fixed time periods are chosen and windows are defined based on them. The connection requests are aggregated based on a predefined aggregation factor. This factor basically specifies how many connection requests are in one VNR. For any given fixed window, the connection requests are analyzed and those with the highest revenue potential are prioritized. The processed requests are then aggregated and mapped to the substrate network. Requests that cannot be mapped are rejected. A maximum waiting time for every request is also considered. Along with the connection request details, the client defines the maximum tardiness allowed per connection. If the connection was not served within this period, it is considered blocked [66].

Dynamic Window Technique

In this technique, client requests are still distributed over multiple time windows. However, the sizes of these windows might differ. The connections are divided into sets and the time window size is specified based on the number and lifetimes of the requests in this set. We implement this step using the maximum independent set algorithm [67][61]. A variation of this algorithm is used in the context of optical networks in [61]. The input of this algorithm is the set of connections as shown in table 4.1. Afterwards, an intersection table, as shown in table 4.2, is constructed containing binary fields indicating weather two nodes intersect (conflict) in time or not. Next, the interval graph shown in Figure 4.1 is constructed. Each connection request is represented by a node, and if two connections conflict in time, a link will be drawn to connect them. Figure 4.1 represents an interval graph of four connection requests (C1 to C4). The algorithm then divides these connections into independent sets so that if two connections are in the same set then they are in the same dynamic time window. Given the interval graph, the algorithm finds the largest set of connection requests such that no two nodes in the set are connected by a link. This set of nodes is called a maximum independent set of this interval graph. The requested start times for connections in each set form the boundaries of the dynamic time windows. In Figure 4.2, the client connection requests are shown divided into different time windows in which they all overlap in time. Within a single dynamic window, the time requirements for different client requests could also overlap. In other words, two different requests accommodated within a single time window might request resources in the same period of time [66].

As mentioned earlier, virtual network mapping involves node mapping and link mapping where connection requests are mapped to the substrate layer. This process is shown broadly in Figure 4.3. Once the number of windows and window sizes have been decided, the requests are allocated to their respective assigned windows and mapping is performed. The steps taken are detailed in Algorithm I. After deciding the number of windows (NW) and each window size (WS), these values are used to assign connections to windows.

Next, connections that expire before the end of their assigned window are filtered and removed. Then, connections from each window are aggregated into virtual network requests (VNR) and sorted based on the potential generated revenue. The system checks if there is enough computational and network resources to map these requests. Finally, node mapping and link mapping are performed. A summary of the whole process is provided in Figure 4.4.

4.3.3 Spanning Connection Technique

After the time window division is done, some connection requests might be long enough to span over two or more windows. These connections are termed spanning connections. Handling spanning connections can go down one of two roads. Either we assign the spanning connection

Algorithm 1 Virtual network mapping using dynamic window technique

```
INPUTS: Connection requests set CR, Server set S, Virtual Machines VMs, Substrate network G(N,L,P) Node set N, Link set L, Path set P where Pijk is the path number k between nodes i and j, NP number of paths between i and j
OUTPUT: Mapping of virtual network requests from all the dynamic windows on the substrate network
Interval graph for CR is generated
ConnectionsInWindow[w]: Set of connections assigned to window w
VN_request[w]: Set of virtual network requests in window w
NW= Number of windows calculated by running maximum independent graphs algorithm
WS = Array of window sizes calculated by running maximum independent graphs algorithm
w=0
for w<NW do
    ConnectionsInWindow[w] = AssignConToWindow(WS[w])
    w++
end for
w=0
for w <NW do
    for all Con ∈ connectionsInWindow[w] do
        if Con.StartTime + Con.duration + Con.AllowedTardiness >= window[w].size then
            Con.StartTime = windowSizeSet[w]
        else NumBlockedConnections = NumBlockedConnections + 1
        end if
    end for
    w++
end for
w=0
for all w < NW do
    Sort ConnectionsInWindow[w] in descending order based on the revenue
    w++
end for
w=0
k=0
for w < NW do
    while k < connInWindow[w].size do
        addNewVNR(k, aggregationFactor,VN_Request[w])
        k++
    end while
    w++
end for
w=0
y=0
for w < NW do
    for y < VN_requests[w].size do // number of VNR in the window
        if CheckNodeMappingGreedy(VN_requesti[w][y]) then
            if CheckPathBW(VN_request[w][y]) then
                NodeMappingGreedy(VN_request[w][y])
                Linkmapping(VN_request[w][y])
                Accepted_VN_requests + +
                NumServedConnections = NumServedConnections + VN_requests[w][y].size
            end if
        end if
        y + +
    end for
    w + +
end for
```

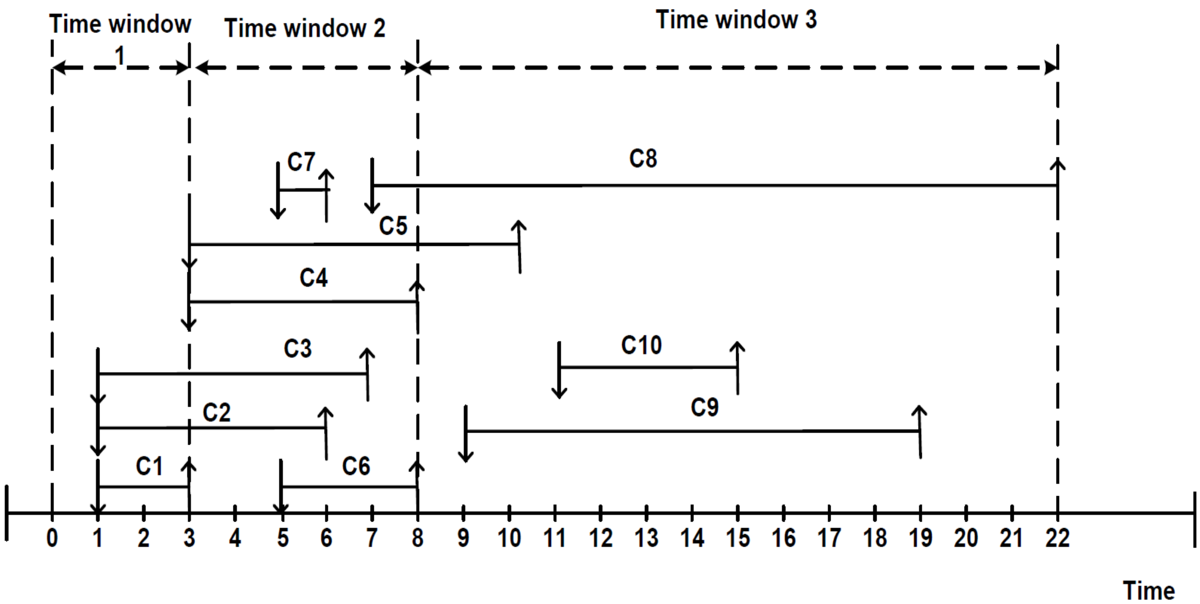

Figure 4.2: Dynamic Windows.

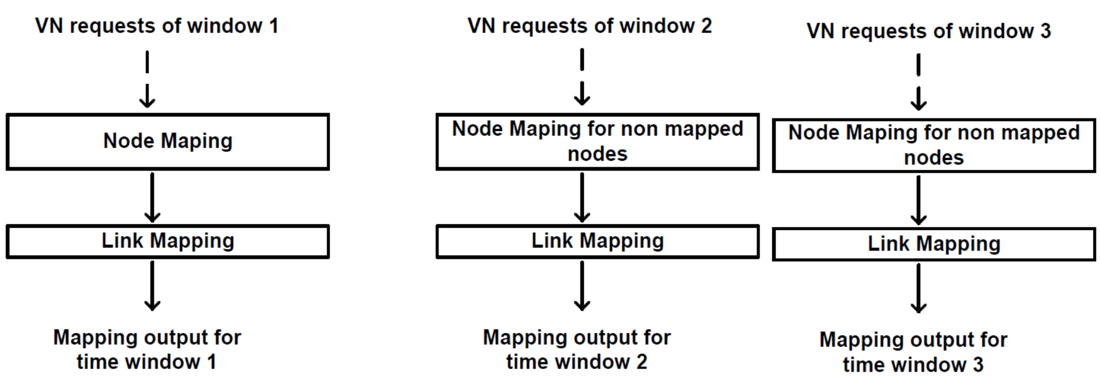

Figure 4.3: Node and Link mapping.

request to one of the windows it spans over, or assign the connection request to more than one (or even all of the covered windows). We start our experimentation by choosing the first option then we investigate the second option and show its effect on performance.

Analysis and aggregation of connections will be performed and this mainly depends on the start window and the duration of connections, which span over single or multiple windows. If the connections have the same start window, then it will be prioritized based on the generated revenue. The aggregation process will be performed only for connections that span over the same number of windows. The process will be repeated until the all connections with the same start window served.

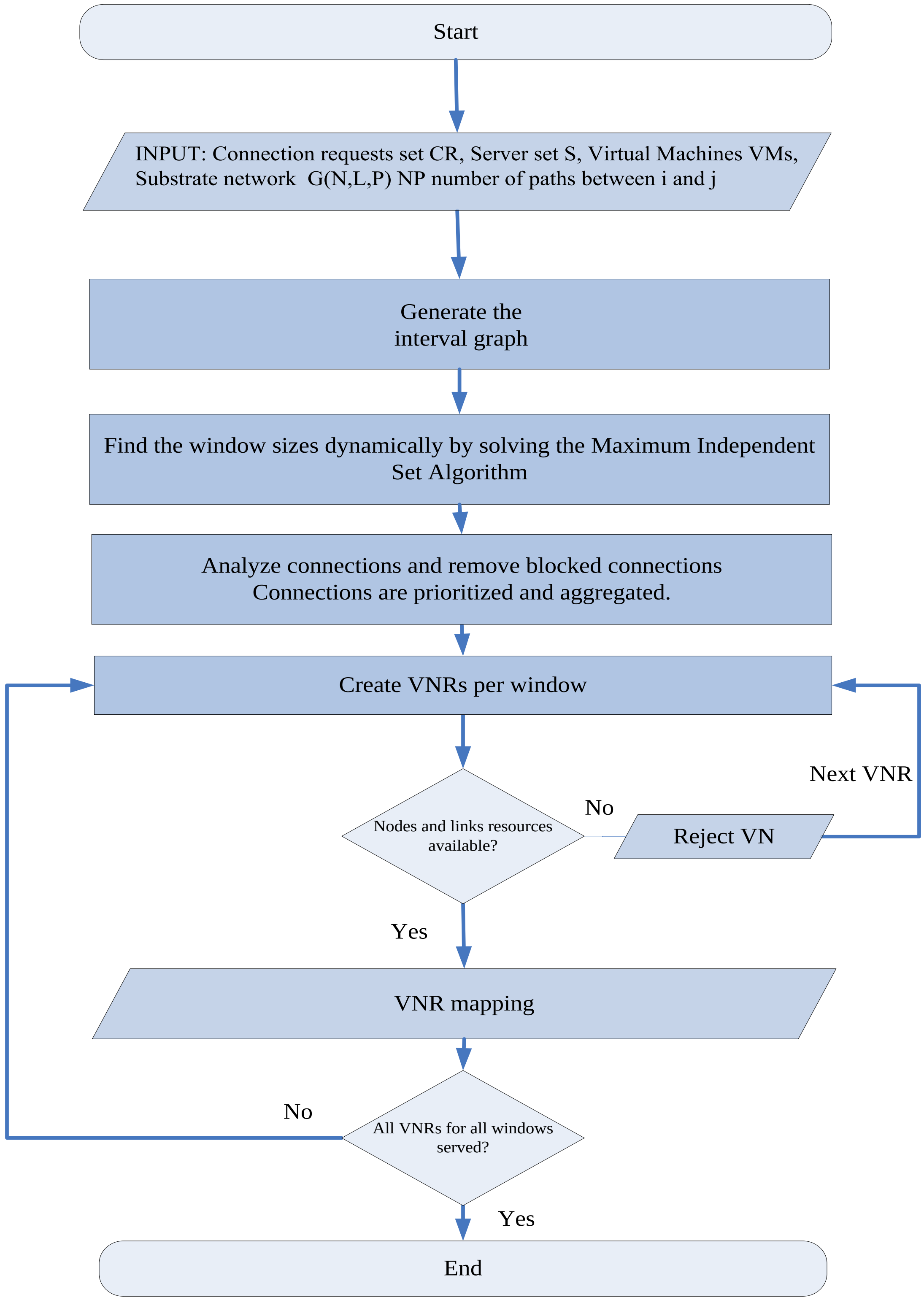

Figure 4.4: The virtual network mapping process.

4.3.4 Revenue Objective Function

One of the objectives cloud providers focus on when designing their virtual network mapping systems is revenue. Revenue mainly depends on the number of accepted VNRs. Revenue is defined in virtual network mapping context as follows [60][68][59] [69] : define the revenue of a single accepted VNR as:

$$R(VNR_i) = \sum_{c \epsilon VNR_i} (\alpha \sum_{t=0}^{T_c} CPU_c + \beta \sum_{t=0}^{T_c} Memory_c + \gamma \sum_{t=0}^{T_c} Storage_c + \delta \sum_{t=0}^{T_c} BW_c) \quad (1)$$

As shown in the equation, the revenue of *VNRi* is the total of the revenue amounts coming from connections aggregated to form this VNR. Consequently, the revenue from a single request is calculated linearly based on the amount of resources it requests. We chose to design the revenue calculation in a generic way that adapts to any cloud provider with any set of resource offerings. The variables α , β, and γ refer to the prices of per unit for computational resources while the variable δ refers to the unit price for bandwidth(BW). CPU, memory, and storage are the main computational resources. Bandwidth is the only network resource used in this work.

4.4 Performance Evaluation

To evaluate the proposed techniques, a discrete event simulator was developed using C++.

With regards to the substrate network, the National Science Foundation Network (NSFNET), as in Figure 5.2, is used in the simulation. Following similar setups to the ones in [58], the network is composed of 14 nodes of which 3 are data center nodes and the rest are client nodes. 132 servers were used in the simulation with 44 servers in each data center. 21 links were set up to connect the nodes with 546 paths defined. Three alternate routing paths were defined for each pair of nodes. The input contained data corresponding to 200 virtual machine instances.

For the connection requests coming from the clients, as in [59] [60] [55][68][70][71], their arrival rates were set according to the Poisson process varying from 1 to 5 steps of 0.5 per 100 time units. The connection request lifetime follows an exponential distribution with an average of 1000 time units. We run each experiment using 3000 connection requests. The

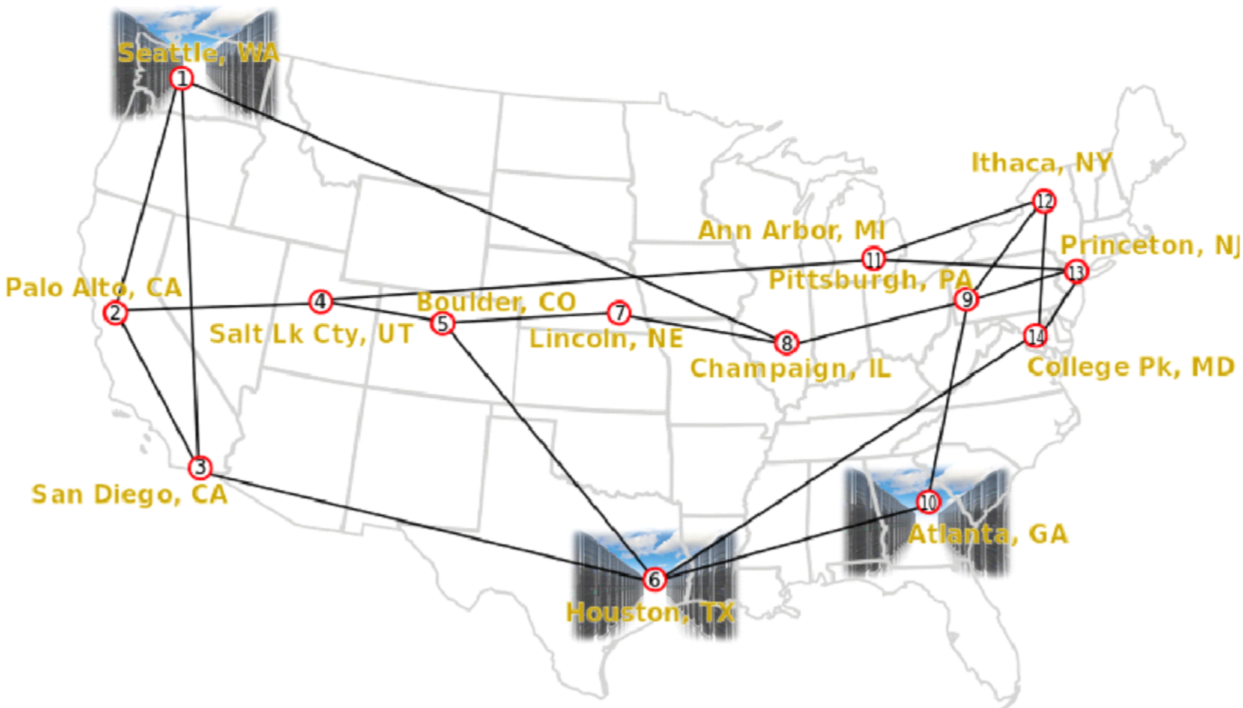

Figure 4.5: NSFNET network of 14 nodes.

maximum waiting time (maximum allowed tardiness) for each request was set to half of its lifetime. A connection request input line includes source, destination, start time, duration, virtual machine specifications and requested bandwidth information. The source nodes are uniformly distributed with a client ID ranging from 0-10, and the destination nodes that represent a virtual machine number follow a uniform distribution of 1-200, given the fact that 200 virtual machine instances were used in the simulation.

As for resource configuration, the CPU resources are uniformly distributed in the range of 50-100, and memory and storage resources are uniformly distributed in a range of 50-100 of their respective units. The available BW is set at 200 for all the links. When looking at the requested virtual machine capacities, CPU resources are uniformly distributed for every request for a virtual machine instance in the range of 0-20. For memory and storage, a uniform distribution is defined with a range of 0-200. Similarly with regards to BW, a uniform distribution within a range of 0-50 is defined.

4.5 Results and Analysis

4.5.1 Effect of Window Size Decision

Multiple metrics were measured during the experiments. Our main objective was to evaluate the window size decision effect over the different performance metrics. We have evaluated a no window scheme along with the dynamic window scheme based on the maximum independent set algorithm. In addition, we have evaluated fixed window scheme with multiple choices for the window size ranging from a small size (50 time units) up to a very large window size (2000 time units).

Figure 4.6 shows the average number of Virtual Network Requests (VNRs) per window. This number becomes very large as the window size chosen grows larger. This has a direct effect on the ratio of served connections. As the number of virtual network requests in a specific window grows, it becomes harder to find enough resources to establish a virtual network by mapping the requested nodes and links. When using dynamic window sizes although it does not produce the lowest number of virtual network requests per window, it still yields a low number close to the numbers yielded by the very small fixed windows size. Therefore, as shown in Figure 4.7 that the served connections' ratio for the dynamic window stands among the best. It was surpassed only by the results from fixed windows of a very small size (50 to 150 time units).

Figure 4.8 shows the average number of blocked connections. As we explained in the previous sections, all connections are aggregated with a factor of three, collected during the window and then processed at the end of the window. This means that if a connection's lifetime expires before its respective window is over, it will be blocked or rejected before the beginning of the mapping process regardless of the availability of the resources. The main factor that affects the ratio of blocked connections, apart from connections' lifetime, is the window size. As the figure shows, the number of blocked connections is very high for large window sizes while it stays at an acceptable level for small and dynamic window sizes. Again, the dynamic window size scheme performs acceptably for this metric.

We now examine the final metric which relates to the computational overhead expected by the network controller that performs the mapping process. The amount of computational

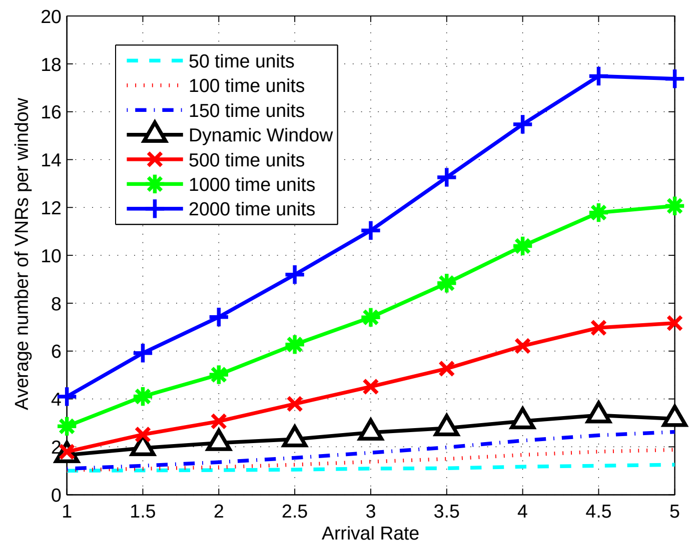

Figure 4.6: Results showing average number of VNRs per window for different window decision techniques.

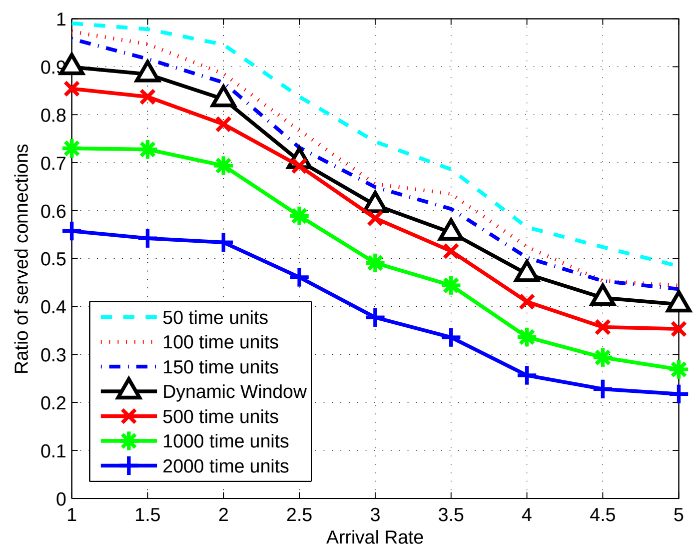

Figure 4.7: Results showing ratio of served connections for different window decision techniques.

overhead needed to map a certain amount of requests is affected by two factors in this problem. The first factor is the number of windows. This is based on the window size when using the fixed windows size scheme. As the windows become smaller, the number of windows needed is larger and the total computational overhead grows. The second factor is the total number of

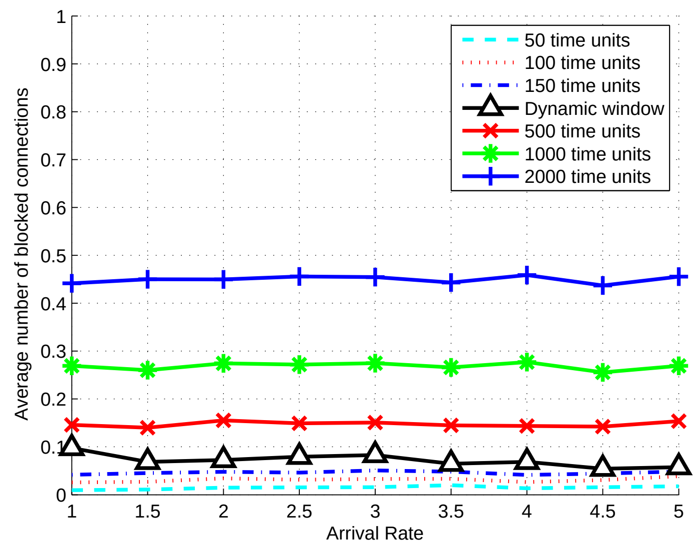

Figure 4.8: Results showing ratio of blocked connections for different window decision techniques.

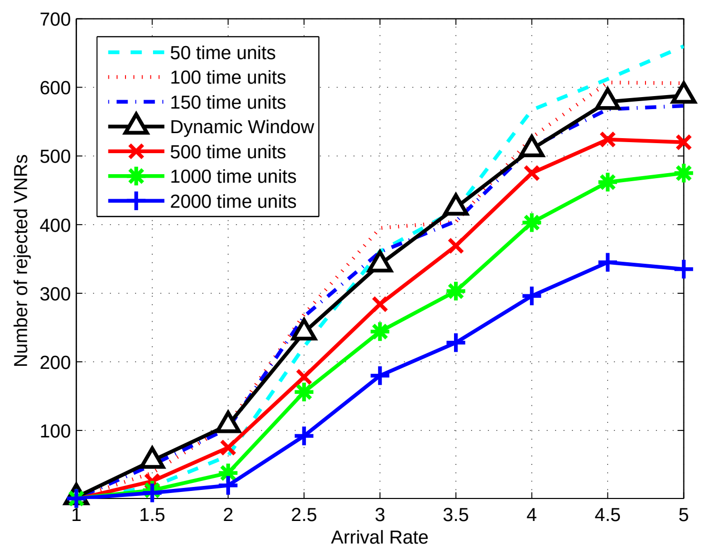

Figure 4.9: Results showing the number of rejected VNRs for different window decision techniques.

VNRs in the problem. Figure 4.10 shows that for a specific amount of requested connections, using a small fixed window size tends to produce a much higher number of VNRs than using the large fixed windows sizes and dynamic windows. This corresponds with a higher overhead that of course is not favorable to the cloud service provider. Comparing the performance metrics

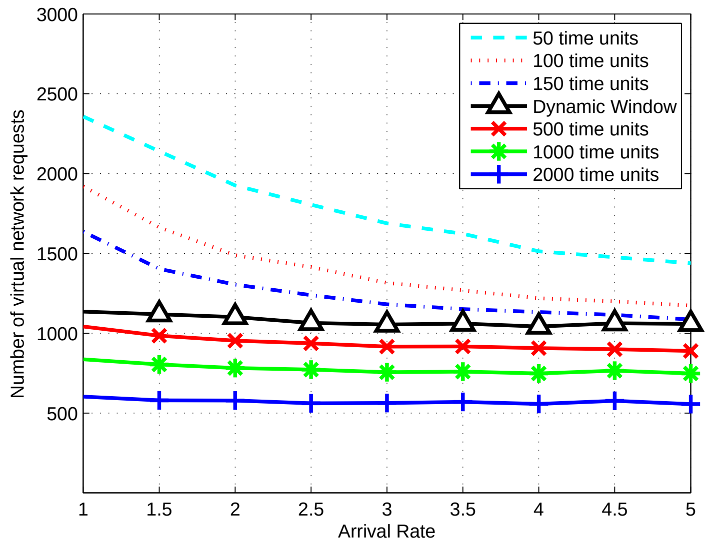

Figure 4.10: Results showing number of virtual network requests for different window decision techniques.

from acceptance, expired connections, and computational overhead, we find that using the dynamic window size scheme is the technique that produces the best performances for all these metrics. Hence, using this scheme would be the best option for the cloud service providers when performing virtual network mapping.

4.5.2 Effect of Connections' Order

After dividing the connections into sets and assigning them to time windows, a decision needs to be made of the order these connections will be processes or served. There are multiple methods to prioritize connection requests over each other. We have evaluated the effect of multiple connections' order schemes on performance metrics. For this set of experiments, multiple choices for the arrival rate ranging from 1 to 5 with a step of 0.5 have been used. CPU, Memory and Link utilization, and generated revenue have been added as metrics. Three prioritizing methods were evaluated for both fixed and dynamic time window schemes. These schemes were used in ordering the requests: high-to-low, low-to-high and same-order. In high-to-low, analyzed connection requests with higher potential revenue are prioritized for aggregation mapping. The low-to-high method gives priority to the other side of the scale while same-order method deals with connections based on their original arrival order. For this

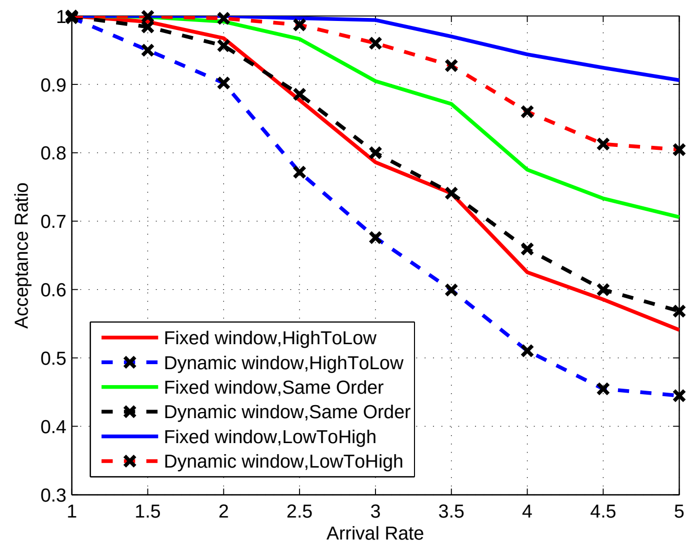

Figure 4.11: Results showing VNR acceptance ratio for different connections order schemes.

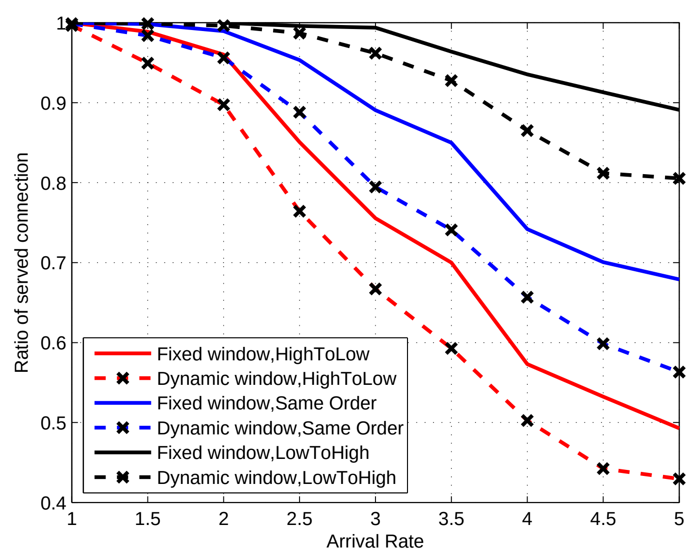

Figure 4.12: Results showing ratio of served connection for different connections order schemes.

round of experiments, fixed window size was set to 50 time units.

Figure 4.11 and Figure 4.12 show the behavior of the three connections' order schemes regarding the metric of acceptance ratio and ratio of served connections as a function of increasing request arrival rate. As we can observe in these figures, the low-to-high method surpasses

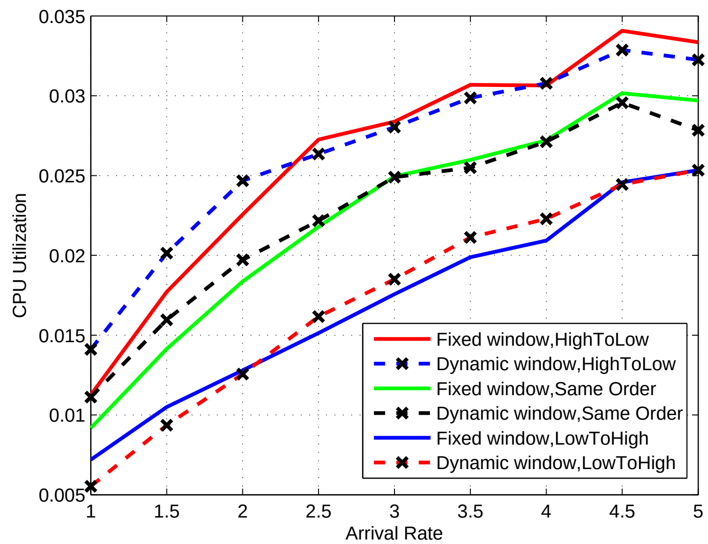

Figure 4.13: Results showing CPU utilization for different connections order schemes.

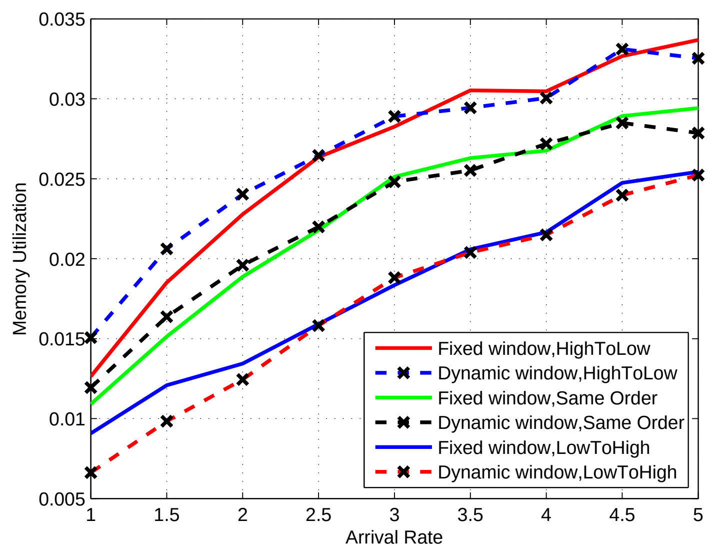

Figure 4.14: Results showing memory utilization for different connections order schemes.

the other two schemes with regards to the acceptance ratio and the ratio of successfully served connections. This can be explained by the close dependency between the amount of revenue generated from a request and the lifetime of a request. Low revenue requests tend to end very quickly, which provides an opportunity to use the network resource to schedule more requests. This reflects positively on the ratio of served connections and the acceptance ratio metrics.

For resource utilization metrics, the picture looks different. Figures 4.13 - 4.16 compare

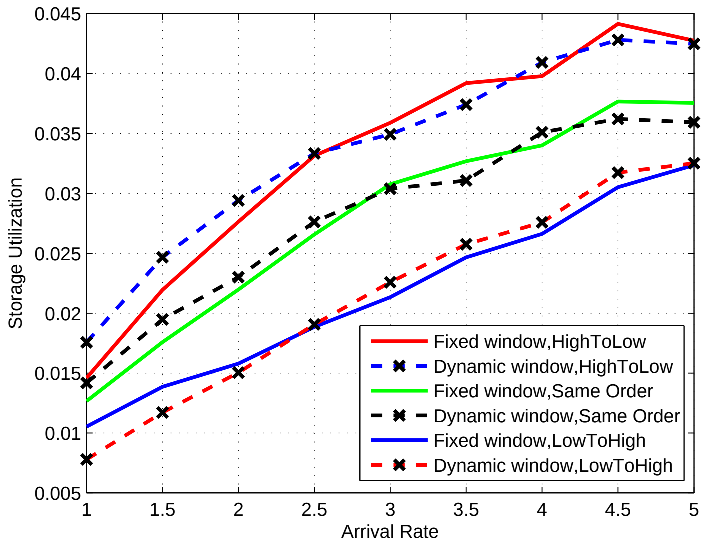

Figure 4.15: Results showing storage utilization for different connections order schemes.

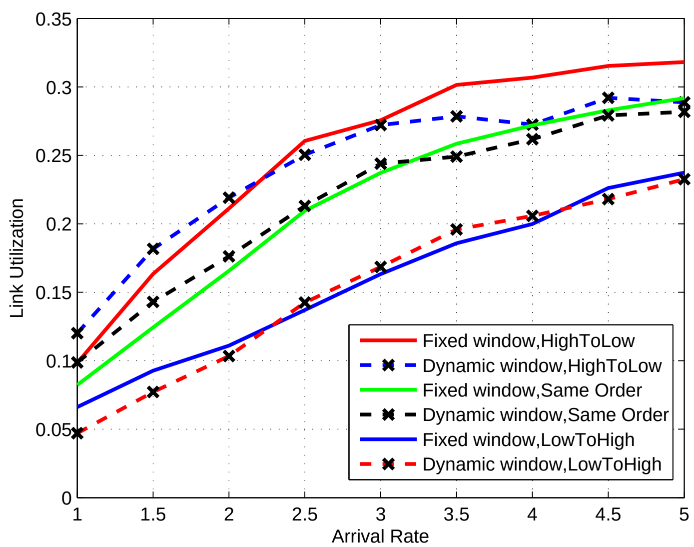

Figure 4.16: Results showing link utilization for different connections order schemes.

the resource utilization for the three methods and covers fixed and dynamic window sizes for each. It is noted that the high-to-low method shows the best resource utilization readings for computational and network resources involving CPU power, memory, storage, and bandwidth. This can be referred to the effect of the method on the served connections. The high-to-low method tends to prioritize a fewer requests but these requests reserve resources for long periods of time. This leads to less resource fragmentation and higher utilization ratios for these

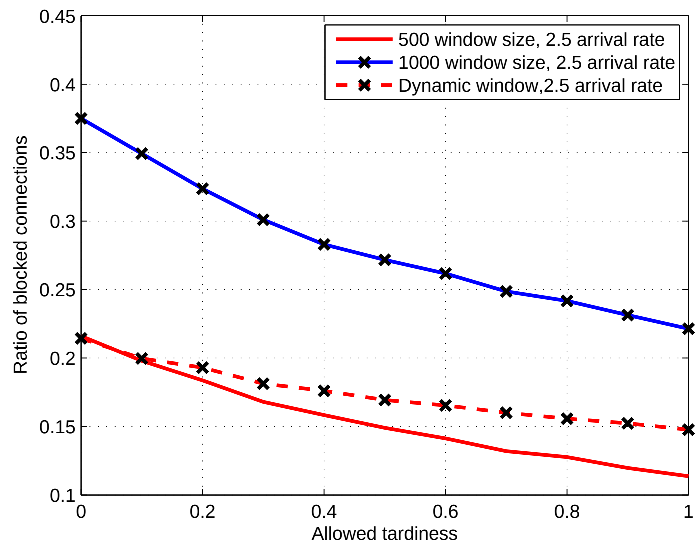

Figure 4.17: Results showing ratio of blocked connections for different allowed tardiness.

resources.

4.5.3 Effect of Maximum Allowed Tardiness on the VN Mapping Performance

The main goal here is to assess the effect of the connection's maximum allowed tardiness on performance metrics. For this experiment, the arrival rate was set to 2.5. We kept the window size either fixed at 500 and 1000 or dynamic, while varying the maximum allowed tardiness of a connection request starting from 10% of the connection's lifetime up 100%.

Figures 4.17 and 4.18 show that choosing an optimal allowed tardiness value is a tradeoff. Figure 4.17 shows the effect of gradually increasing the allowed tardiness limit per connection on the ratio of blocked connections (connections rejected in the phase before the mapping starts). This is shown for three different window size techniques. It is clear that the ratio of blocked connections decreases as the allowed tardiness per connection becomes higher. This is due to the fact that when connections have higher tolerance for tardiness, they are able to wait until the end of the window when the scheduling happens; therefore, fewer connections are blocked.

Figure 4.18 shows the effect of increasing the allowed tardiness level on the ratio of served

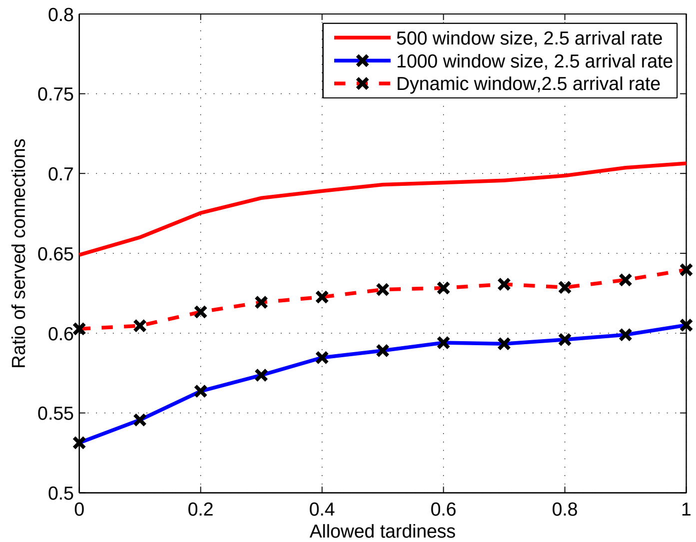

Figure 4.18: Results showing ratio of served connections for different allowed tardiness.

connections. Regardless of the window size technique used, high allowed tardiness levels lead to improvements in the ratio of served connections. Higher tardiness tolerance gives the scheduler more options in terms of connection mapping and leads to more connections being scheduled. These improvements range from around 3% in the case of dynamic window sizing to 8% in the case of large window sizes. The dynamic window size technique shows stable performance regardless of the degree of tardiness allowed. Hence, this technique is more suitable for highly demanding environments where connection allowed tardiness level either keeps fluctuating or is very tight to begin with.

4.5.4 Effect of Spanning Connection Technique

To conclude the set of experiments, an analysis was conducted to evaluate the result of implementing the spanning connection technique on the performance metrics. We have used arrival rates ranging from 1 to 5 with a step of 0.5. We focused in this analysis on generated revenue as the main metric.

Analysis of connection requests was carried out using both fixed window and dynamic window size techniques. Requests are then prioritized based on the highest revenue (using high-to-low method) . Aggregation is performed for the prioritized requests and the mapping

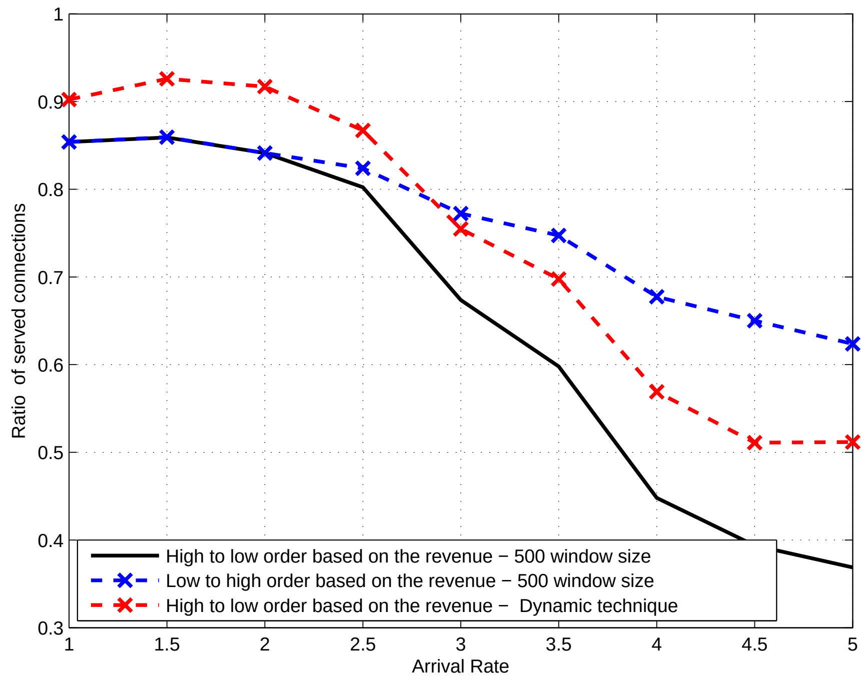

Figure 4.19: Results showing the ratio of served connections for different order schemes using spanning connection technique.

is done onto the substrate network. Requests are rejected if mapping cannot be embedded on the substrate network.

Figure 4.19 shows the effect of enabling the spanning connections on the ratio of served connections when using dynamic window and fixed window size of 500 time units along either high-to-low or low-to-high ordering methods. The figure shows interesting results. The comparison between the three methods is not as straight forward as it was before using the spanning connections. Although fixed window size with high-low method still yields the lowest served connection ratio compared to the other two methods, low-to-high does not guarantee the best performance in all cases like before. Instead, using fixed window size with low-to-high method performs better in the case of high connection load while using the high-to-low method along with the dynamic window size technique method produces better results when the load is low or regular. This is due to the fact that this method prioritized the spanning connections inherited from earlier windows to be served first. This allows for more connections from this category to be served compared to the low-to-high method which will order the spanning con-

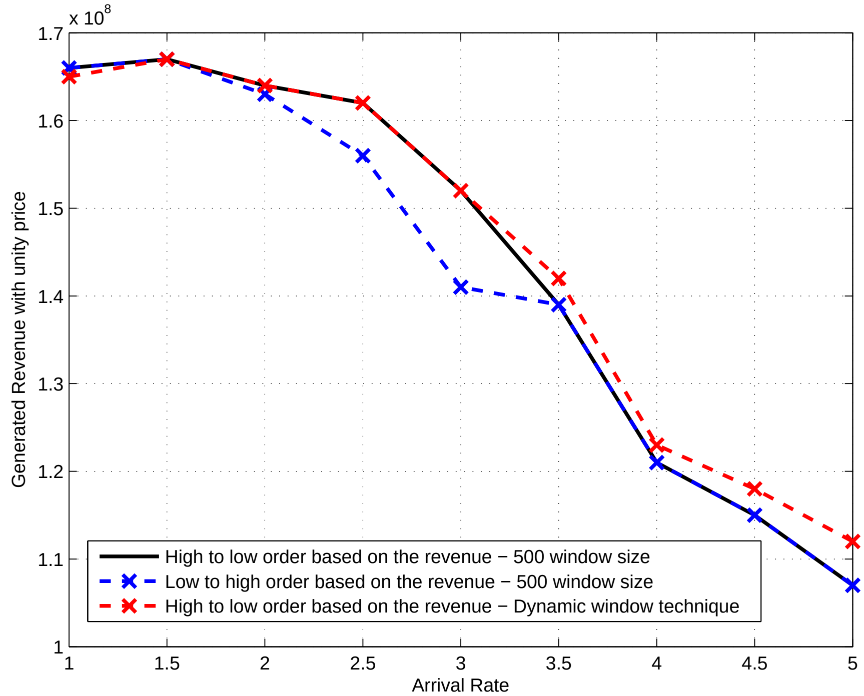

Figure 4.20: Results showing the generated revenue for different mapping schemes using spanning connection technique.

nections along with the current window connections based on their potential revenue. As the connections that arrived in the current window will have more tolerance for tardiness, this in total will lead to more connections served. However , as the load increases, the effect of the spike in the service ratio starts to go down and the original trend starts to show again. This is also supported by the fact that the spanning connections percentage gets marginalized when the number of connections per window becomes higher.

In the experiment shown in Figure 4.20 we try to find the combination of techniques that achieves highest revenue when using spanning connection technique. Using Dynamic window size selection along with high-to-low connection ordering technique achieves the highest revenue for high and low loads. This is despite the number of connections served through dynamic window size selection at high arrival rates were less in high-to-low than when using low-to-high ordering methods. This is because of the focus on higher revenue requests which guarantee more revenue per request.

Figure 4.21 presents the ratio of blocked connection in the case where the maximum al-

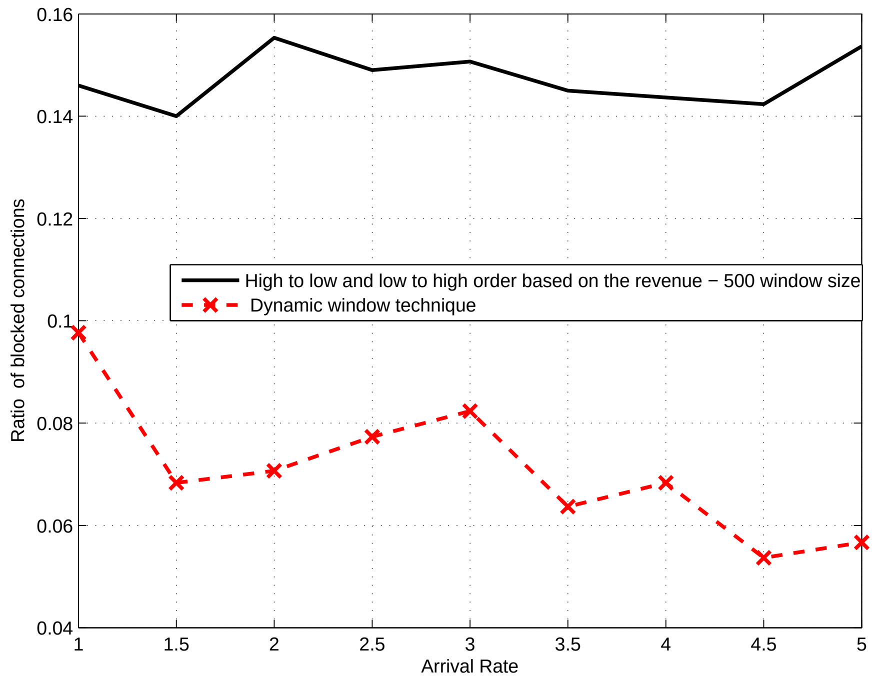

Figure 4.21: Results showing the ratio of blocked connections for different order schemes using spanning connection technique.

lowed tardiness has been reached. The blocking numbers for that case are compared for fixed window seize selection and dynamic window size selection. As discussed earlier, the dynamic window size selection tailors the windows to the shape of the connections' data set. That causes less blocking. Moreover, when using the spanning connection technique, it leads to less blocking caused by the maximum tardiness reached as the problem grows bigger. This is mainly because, as the load grows blocking happen for other reasons like resource congestion.

The evaluation result demonstrates the benefits of proposed spanning connections technique and proves that the results are best drawn in the case of dynamic windowing.

Chapter 5

A Greener Cloud: Energy Efficient Provisioning for Online Virtual Network Requests in Cloud Data Centers

5.1 Introduction

Cloud computing, over the last few years, has emerged as a new computing model, allowing utility based delivery of IT services to end users. Cloud computing relies on virtualization technologies to provide on-demand resources according to the end user's needs, but problems with resource allocation pose an issue for the management of several data centers. There are a number of constraints on the choice of allocation strategies, such as server capacity, fault tolerance or the network bandwidth capacity, just to name a few [72].

An important challenge in network virtualization is virtual network embedding. This involves constructing a virtual network using both the physical side of a network (meaning the nodes) such as hosts, as well as the links, including allocating appropriate CPU, memory and storage [15][16]. Most studies of network virtualization focus on accommodating the maximum number of virtual network requests in order to maximize revenue but fail to consider the electricity costs of the infrastructure provider's network [59][60]. As shown in another study, energy related costs represent 50% of operating costs for a data center, and such costs are increasing faster in relation to other related expenses such as network devices or servers [15].

Data centers consumed on average the energy equivalent of 25,000 households [73][53]. Such high power consumption also leads to other serious issues at data centers, such as shortening the lifetime of different devices, wasting energy, and releasing excessive amounts of CO2, a major cause of global warming [53]. Thus the consideration of power consumption in the design of data centers is important for overcoming these issues [72].

Optimizing energy use in cloud computing is an issue that has been drawing greater attention from multiple levels in recent years. The literature contains a number of different solutions related to energy-awareness, such as (a) using more energy efficient hardware, (b) server consolidation, (c) application of energy efficient network protocols, and (d) adoption of temperature aware workload applications [72].

A typical cloud computing data center consists of large number of geographically dispersed servers or clusters of servers, which are interconnected through the use of a physical network to create the cloud's infrastructure. The different types of devices used in the physical network are the main cause of power consumption. Power consumption in the network can be grouped into two types: that associated with the communication of the network devices, and that associated with the processing of the servers being used, representing roughly one-third and two-thirds, respectively, of the total power consumed by the network. The large number of applications in use in a typical data center makes minimizing power consumption a highly challenging but critically important task [53].

Network virtualization can be used to increase energy efficiency in substrate networks, allowing consolidation through the virtual hosting of different resources on the same substrate resource. Migrating resources virtually allows the network to balance its overall energy load and reduce the total power consumption of the network [74].

Cloud computing environments require a number of VN embedding requests to be handled by central network controllers in short periods of time. If all these requests are fulfilled individually at the time of their arrival then it is possible to significantly increase the computational overhead. Along with high levels of computations, cloud data centers also face energy management problems. Most studies of cloud computing data centers focus on managing computation problems, while small algorithm modifications can lead to energy savings through appropriate network device management. Managing energy use efficiently can also help increase the lifetime of network devices and reduce CO2 emissions, a leading cause of global warming [53].

This chapter introduces an energy-aware provisioning of online virtual network requests in a cloud computing environment. Our contribution can be summarized as follows:

- Investigate the effect of energy efficient node and link mapping strategies combined with window technique and spanning connections in energy consumption in a cloud environment.

5.2 Related Work

5.2.1 Power Efficient Provisioning for VN Embedding

Virtualization or network virtualization is considered a key enabler in cloud computing. Requests for applications from users are treated as virtual machines to differentiate between the nature of the applications and hardware. The cloud infrastructure consists of physical servers and equipment of the WDM network, which all consume some quantity of power. The power consumed by this infrastructure can be divided into two parts, workload dependent power, and workload independent power. Workload independent power is considered idle power, and as such it is subject to reduction through a scheme of energy-efficient virtual network provisioning. The basic plan behind this scheme is to switch off lightly-loaded equipment in the cloud infrastructure, made possible by merging virtual machines with a small number of physical servers and routing communication demands through a small number of fiber links [75].

Continuous monitoring of data traffic and powering off unused switches are highly useful in managing energy efficiency in data center networks. Monitoring network traffic and power usage is also advisable for reducing total energy use in an enterprise network. Another technique involves shutting off cables selectively in bundles or links during light traffic periods. Another study uses resource scheduling to reduce energy consumption in virtual network embedding. Consolidation of these techniques produces an energy consumption model along with an energy-aware virtual network embedding algorithm [76].

5.2.2 Energy Aware Data Centers

The main technology currently used for energy-efficiency programs in data center servers is virtualization. By enhancing management decisions, it is possible to create, move, copy and delete virtual machines that can summarize virtualized services. Consolidation of hardware and the reduction of redundancy are useful in improving energy efficiency. Hibernating or turning off unused servers is also helpful in saving energy. Hardware able to manage higher loads is also available for reducing physical servers. Self-management, however, remains inadequate in managing energy efficiency in data centers. The services on a data center's site can be moved to other sites if necessary, as virtualization and management are not the only solutions.

In addition to considerations of load during the migration of services, there is also the issue of heat generated during these operations, as every time a node in a data center is used it also produces heat. Services with a high operation load and a high generation of energy can be moved to data centers which are low in operational load and energy generation [16].

For handling power use efficiently, a resource allocation energy-aware joint node and link approach is presented in [16]. A generalized power consumption model is presented which captures four different states: fixed power consumption of an active physical node, variable power consumption of an active physical node, fixed power consumption of network equipment and variable power consumption of network equipment [16].

Fixed power consumption of an active physical node deals with idle power consumption, which is also known as baseline power consumption. Variable power consumption of an active physical node deals with CPU power consumption, which is the main contribution to power consumption variations in the physical nodes. Fixed power consumption of network equipment deals with baseline power consumption during periods of non-connectivity of the network ports, while variable power consumption of network equipment deals with the active link's power consumption. The network port's power consumption is independent of the link's actual load, so the number of connected links is directly proportional with the upper power consumption limit [16].

5.2.3 Energy Conservation in Cloud Data Centers

A model for energy conservation in a cloud data center environment is presented in [76]. This study focuses on the problem of energy-aware virtual embedding with the help of consolidation of the least number of substrate nodes and the powering off of unused substrate nodes. There are two challenges for the use of consolidation technique in virtual network embedding, energy consumption modeling and energy-aware embedding algorithm design [76].

The challenge of energy consumption modeling emerged because of the various roles substrate nodes play in order to satisfy CPU and virtual network requests, as well as bandwidth requirements. Because of this, substrate nodes have very different levels of energy consumption, making them difficult to model accurately. The second challenge, that of energy-aware embedding algorithm design, is also known as an NP hard problem, which make it more complex when developing an algorithm that simultaneously considers energy efficiency [76].

In the first challenge, the active substrate nodes are identified and differentiated in two categories, working and intermediate nodes. The working nodes are those nodes which provide satisfactory computing capacity as well as communication bandwidth used for sending and receiving of packets. The intermediate nodes are those which only forward packets. The energy consumption model is designed for both these nodes, as well as a quantitative analysis for all energy consumed including that used by virtual nodes and virtual links in virtual network requests. An integer linear programming problem is formulated for minimizing active substrate nodes [76].

For the second challenge, an algorithm called EA-VNE is designed which uses an energy-aware virtual network embedding heuristic algorithm, i.e. a two stage virtual network embedding algorithm. In the node mapping stage of the algorithm, the best-fit and worst-fit schemes are used for ranking nodes in node mapping. The objective of using the best-fit scheme is to satisfy node requirements in virtual network requests to minimize the number of active working nodes, while the objective of the worst-fit scheme is to satisfy connectivity constraints, which also benefits the succeeding link mapping stage and maximizes the probability of virtual network requests acceptance. A special link-mapping mechanism is also developed in the link mapping stage which is used to minimize number of active intermediate nodes [76].

5.2.4 Energy Aware Virtaul Network Mapping

An energy-aware virtual network embedding approach is proposed in [77], which presents a trade-off between increasing the number of virtual networks accommodated by an ISP and decreasing the energy costs of whole system. An ISP maps virtual networks to physical nodes and links in order to accommodate virtual network requests in its network, along with minimizing the additional energy costs of these requests [77].

There are two important observations generated by this approach. The first is that substrate nodes are normally dispersed geographically to deploy and deliver end-user services, so the energy costs in different areas might be different and fluctuate over time. An ISP needs to efficiently map virtual nodes of a virtual network to physical nodes with lower energy rates while satisfying virtual network location constraints [77]. Secondly, server power consumption and CPU utilization are approximately linear with a large offset that is up to 50% of power at peak. For this reason, an ISP needs to map the virtual network's virtual nodes to its active physical nodes so the maximum number of nodes without any load can be put into sleep mode to save energy [77]. Modeling and computing the energy cost of the complicated physical infrastructure of an ISP network is a technical challenge, as is implementing such a solution, which is also addressed in this study. A further technical challenge that is also considered is designing a virtual network embedding algorithm which is energy-aware [77].

5.3 System Model

It is very critical to model the resources involved in cloud infrastructure accurately, while creating a competent energy-efficient resource allocation methodology. It is also important for cloud management systems to be continuously aware of the status of their infrastructural operations. In the scenario we are investigating, clients will reserve VMs for a fixed amount of time. There are both data centers, client nodes, and routing nodes in a substrate network, with each data center able to host a number of virtual machines. The number of virtual machines a data center can host cannot exceed its server capacity. With different types of resource configurations there are multiple virtual machines available. When a client reserves a virtual machine, it is possible for client to request a connection between the client node and the virtual machine. Establishing such a request requires the client to define some aspects such as source, destination, the preferred (requested) starting time, duration or connection lifetime, bandwidth required, and virtual machine specifications or preferred capacity units (requested). Table 5.1 shows a sample of connection request input data. As a solution, clients' connection requests are aggregated based on a preset aggregation into Virtual Network Requests (VNRs). Each VNR will be assigned to a fixed time window based on the requested start time. Consequently, a set of VNRs within each window will be abstracted as virtual network embedding requests. Subsequently, the system performs energy efficient node and link embedding for VNRs. In the following section, we discuss the energy consumption model based on this scenario for embedding virtual nodes and virtual links into the substrate network.

Table 5.1: An example of a set of connection requests

Connection requests	Source	Destination	CPU	Memory	Storage	Bandwidth	Requested start time	Duration
C1	Client node 12	VM68	12	15	8	26	1	1
C2	Client node 12	VM170	2	19	8	33	1	4
C3	Client node 2	VM79	11	13	4	29	1	5
C4	Client node 0	VM163	1	6	11	22	3	5
C5	Client node 7	VM146	3	15	19	17	3	9
C6	Client node 13	VM28	7	3	11	13	5	3
C7	Client node 7	VM92	11	13	17	38	5	1
C8	Client node 13	VM143	9	15	9	9	7	15
C9	Client node 11	VM37	14	8	8	39	9	10
C10	Client node 9	VM5	5	6	10	31	11	4

5.4 Power Efficient Virtual Network Embedding

This section demonstrates the network model or cloud infrastructure which describes the structure of the network. It further proceeds with virtual network requests, virtual network embedding and power efficient virtual network provisioning.

5.4.1 Network Model / Cloud Infrastructure

The network model in this study is a substrate network. The substrate network has elements such as nodes and links. Every node and link is associated with a set of capacities. CPU, memory and storage are used for the nodes while the capacity of available bandwidth is used for links.

The virtual network has elements such virtual nodes and virtual links. The requested capacities of virtual nodes are CPU, memory, and storage. In our case, bandwidth is the only requested capacity of virtual links . A requested start time, duration, allowed tardiness are associated with each virtual network request.

5.4.2 Energy Aware Virtual Network Embedding

Virtual network embedding is composed of node embedding and link embedding. Both embeddings are described in this section below.

Energy Aware Virtual Network Embedding: Node Embedding Stage

Virtual network embedding is required to allocate necessary computational resources prior to embedding the virtual network request on the substrate network. Having an energy-aware virtual network embedding sets a specific goal for the node embedding stage. This goal is embedding virtual nodes into the least possible number of substrate nodes while fulfilling the virtual nodes' CPU, memory and storage requirements.

In this technique we calculate the difference between the CPU capacity required by the virtual node and the CPU capacity available on the substrate node. Obviously, only substrate nodes with enough resources that have more than 15% of their capacity available as a safeguard are considered as candidate nodes. The virtual node is then embedded to the substrate node that it best fits. This method has some benefits such as saving energy by reducing the working nodes.

Energy Aware Virtual Network Embedding: Link Embedding Stage

In virtual network embedding, there is a need for embedding virtual links to a physical network path. The embedded network path should be a valid path between the source of the request (a embedded virtual node) and the private cloud or client node. The link embedding also requires fulfilling the bandwidth requirements for the virtual links on each physical link used for path construction. A greedy embedding algorithm for the selection of the most energy efficient shortest path is used. The algorithm finds the three shortest paths between the given two nodes.

Following this, the path with the minimum number of bridging nodes needing to be switched on will be selected. We anticipate that this will lead to energy efficiency.

5.4.3 Used Techniques

Fixed Window Technique

In a fixed window technique, fixed time periods are created to define windows. On the basis of a pre-defined aggregated factor, the requests of connections are aggregated. Connection requests are analyzed for any given fixed given window as well as prioritized on the basis of their revenue potential. The prioritized connection requests are then aggregated and embedded towards the substrate network. The remaining connection requests which are neither embedded nor aggregated are rejected. The maximum amount of waiting time required for every connection request is also considered. The maximum value of tardiness which can be allowed for each connection is also defined by the client through their connection details. The connections which cannot be served within this time period are also blocked.

Spanning Connection Technique

After finalizing the time-window division there might be some connection requests which are too long and need to span in two or more windows. These connection requests are termed spanning connections. There are two possible methods to handling such connections: the spanning connection can be assigned to spanned windows, or it might be assigned to more than one connection request. Our experimentation will start with first option, and after will investigate the second option to show the effect of each on performance.

The analysis and aggregation will be performed on connections which mostly depend on starting window and connection duration. This might span over multiple or single windows. For connections having same start windows the prioritization will depend on generated revenue. Connections having the same number of windows are required to perform an aggregation process. This process will be performed repeatedly until all same start window connections are served. A summary of the whole process is provided in Figure 5.1.

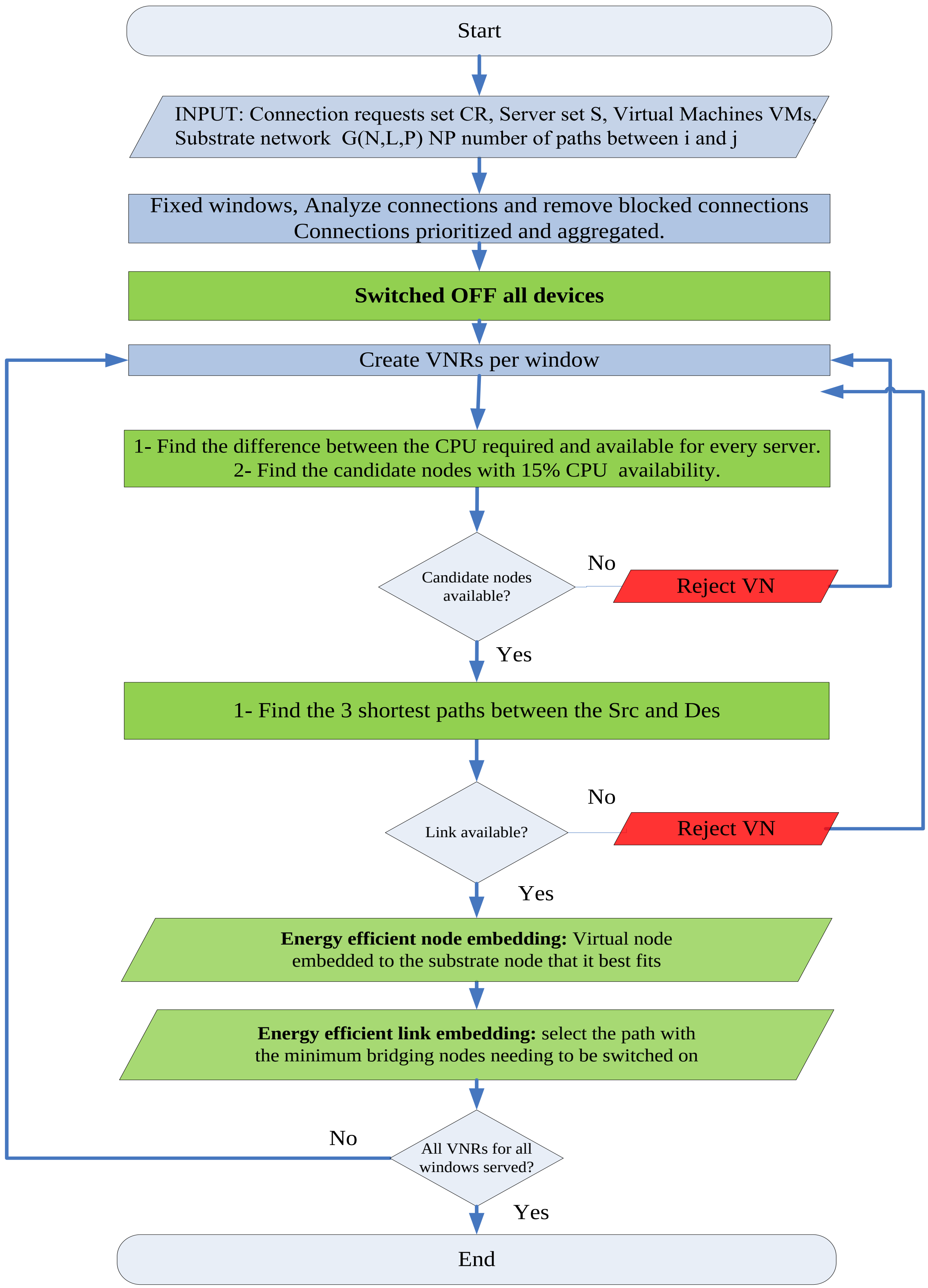

Figure 5.1: The energy efficient virtual network embedding process.

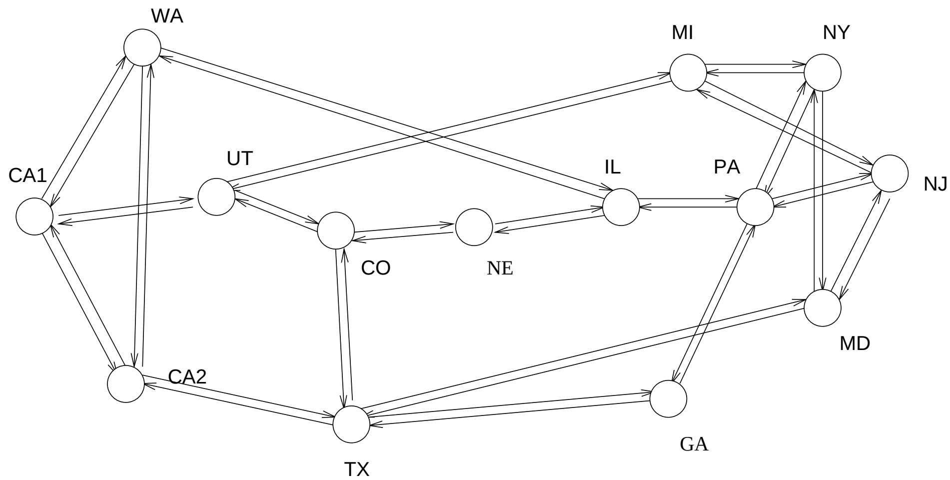

Figure 5.2: NSF Network of 14 nodes.

5.5 Performance Evaluation

5.5.1 Evaluation Setup and Metrics

A discrete event simulator has been designed and developed with the use of C++ for evaluating the proposed technique.

The NFS network with regards to the substrate network is used in the developed simulator. The designed network, as shown in Figure 5.2, has 14 total nodes, where 3 nodes are data center nodes and 11 are client nodes. The simulation uses 132 servers, with each data center containing 44 servers. 546 paths are defined to connect the nodes with 21 links altogether. For each pair of nodes, there are 3 alternate routing paths, as shown in Table 5.2.

Table 5.2: Substrate Network.

Substrate Network	
Network used in the simulation	NSF
Number of nodes	14 nodes of which 3 are,DCs
Number of links	21
Paths	546
Alternate routing paths	3
Available BW	200
Number of servers	132 servers, 44 servers per DC

The simulator's input data parallels 200 virtual machine instances. The arrival rate for the connection requests of clients is set using the Poisson process to 2.5 per 100 time units. An exponential distribution of connection request lifetime is followed with average of 1000 time units.

For each set the maximum allowed tardiness, or waiting time, was set to half of the lifetime. The input line of a connection request includes the source, destination, starting time, lifetime, specifications of the virtual machine, allowed tardiness and requested bandwidth information.

Each experiment is run for different number of connection requests 500-3000. For each set the maximum allowed tardiness, or waiting time, was set to half of the lifetime. The input line of a connection request includes the source, destination, starting time, lifetime, specifications of the virtual machine, allowed tardiness and requested bandwidth information. In the simulation 200 virtual machines are used.

As shown in Table 5.3, for the server configuration, the CPU resources are also distributed uniformly over a range of 50 to 100, while memory and storage resources are distributed uniformly over a range of 50 to 100 with respective units.

Table 5.3: Server configuration

Servers specifications	
Available resources per server	CPU, Memory, and Storage
CPU resources	Uniformly distributed in a range of 50-100.
Memory and storage	Uniformly distributed in a range of 50-100.

For all links, the BW is a set of 200 units. As shown in Table 5.4, the virtual machine capacities, which are requested the CPU resources for each request, are also distributed uniformly from 0 to 20 for each virtual machine instance. Memory and storage distribution is designed with a range of 0 to 200, while the BW range is from 0 to 50, designed for uniform distribution.

Table 5.4: Capacities of the requested virtual machine

Requested virtual machine,capacities	
CPU	Uniformly distributed for every request for a virtual machine instance in the range of 0-20
Memory and Storage	A uniform distribution is defined with a range of 0-200
Bandwidth	Uniformly distributed in a range of 50-100.

Initially, all the nodes are switched off. As in [76], servers and bridging nodes in an idle state consume 150W . For full CPU utilization, the servers' power consumption is set to 300W. The power needed for routing a connection request is set to 15W.

5.5.2 Numerical Results and Comparison

The results are discussed in this section with the help of figures. Figure 5.3 is showing the energy consumption for our algorithm used in simulator i.e. energy efficient node and link mapping as discussed in previous sections. The proposed algorithm is compared with a random technique which is used to map nodes and links. The physical node is embedded with virtual node randomly as well as physical link is embedded with virtual link in the same way. It is clearly shown in the figure that our technique outperforms the random technique in terms of energy consumption. This graph also proves that our algorithm is more efficient in terms of energy efficiency. It can be clearly seen that our algorithm has much better results than the other random technique. This also shows that our algorithm is more efficient when used in larger networks and enhance the energy consumption of the network. Figure 5.4 shows the amount of energy gained using BestFit technique with windowing in cloud environment.

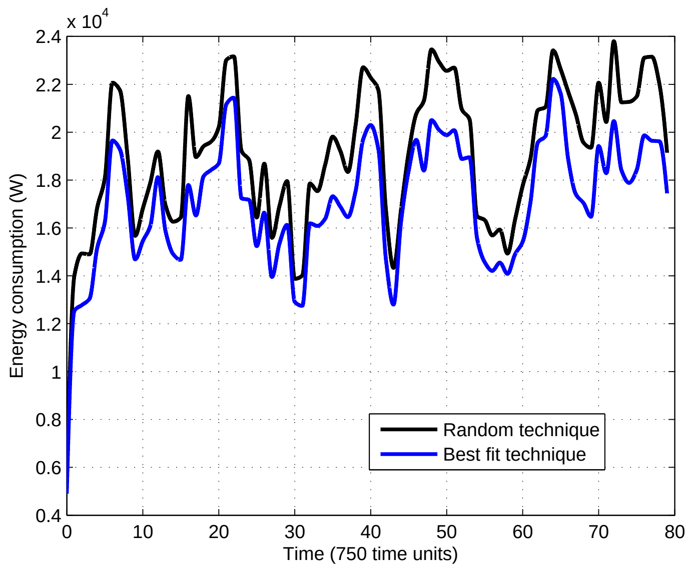

Figure 5.3: Real time energy consumption.

The number of served connections used for the both techniques are shown in Figure 5.5. Our algorithm has a better acceptance ratio as compare to the random technique. In the figure

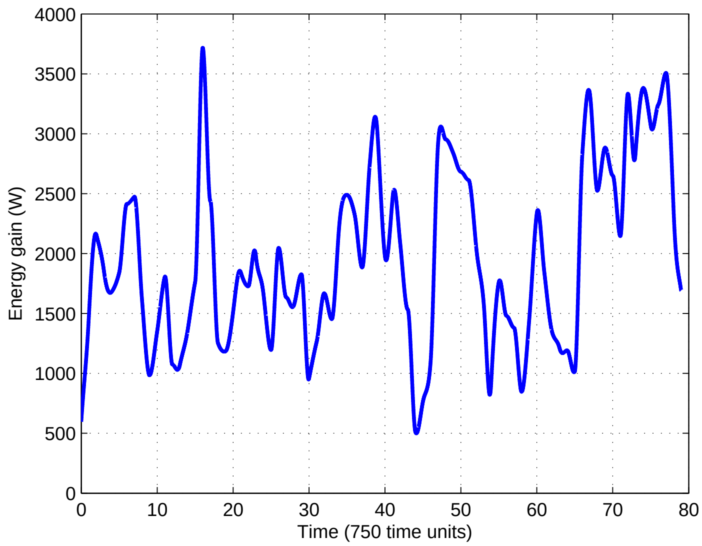

Figure 5.4: Energy gain using BestFit with windowing in cloud environment.

5.5, the served connection decreases (e.g at 1500 connections) that means that the algorithm prioritized the connections and tend to serve the connections with highest revenue even the number is less.

The objective of this work is to make energy consumption more efficient and improvement of revenue. In Figure 5.6, it is clearly shown that our algorithm generated better revenue than random technique. The figure is showing that the revenue is increasing all the time with a good ratio which has its own importance.

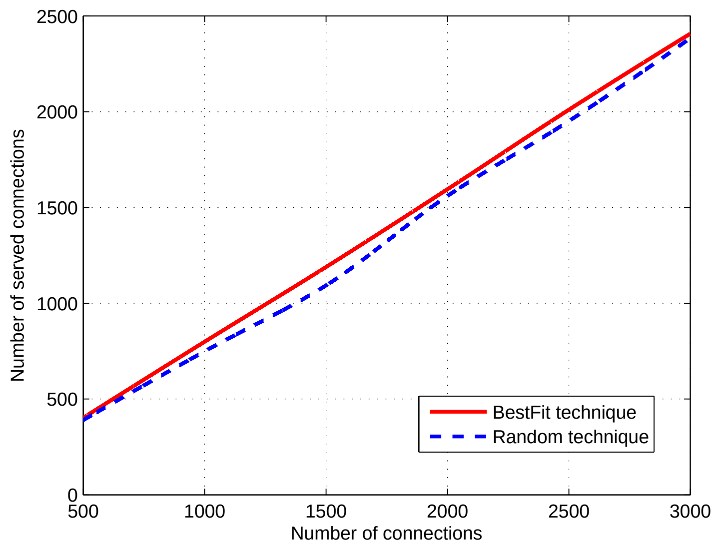

Figure 5.5: Number of served connections.

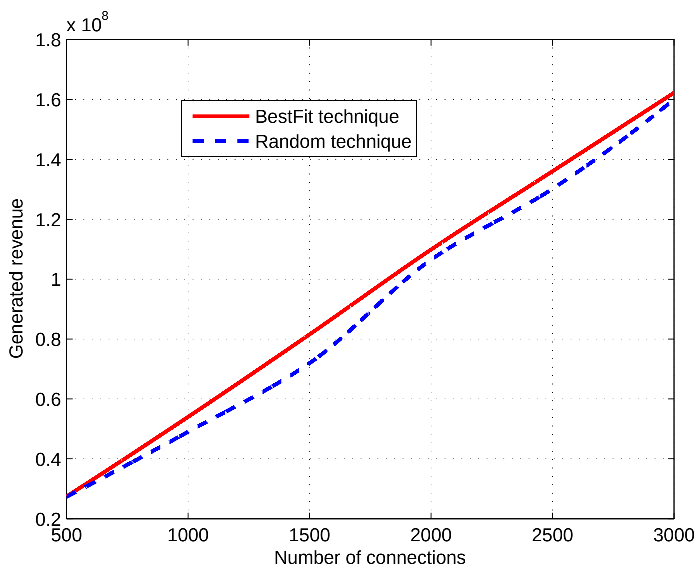

Figure 5.6: Generated Revenue.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Implementing virtualization in a smooth and cost effective way is crucial to the cloud service acceptance and market penetration. A challenge faced by cloud service providers is designing the resource allocation techniques that will tackle the problem of virtual network mapping. Clients send numerous requests to reserve computational and network resources and expect the Quality of Service conditions they specify to be maintained through the request lifetime. One of the main features that define a virtual network mapping policy is the windows size selection scheme. Multiple window size selection schemes were presented and evaluated in chapter 4. A dynamic window selection scheme was introduced in the context of virtual network mapping for cloud computing data centers. After evaluating the possible window size selection techniques, simulation results showed that the Dynamic Windows Size scheme achieved all cloud service providers objectives in terms of served connection ratio, resource utilization and computational overhead. Moreover, three connection ordering schemes were investigated. Low-to-High technique achieved the best performance in terms of the ratio of served connections while high-to-low had the advantage in terms of resource utilization. In addition, the effect of adding features like maximum allowed tardiness and the spanning connection technique was studied.

An energy efficient technique for embedding online virtual network requests in cloud environment was proposed in chapter 5. The proposed algorithm produces better results in terms of energy consumption and revenue generation. Results demonstrate that the proposed algorithm is clearly managing energy consumption more efficiently than the random technique. The algorithm also generates better revenue and has a better acceptance ratio as the number of connections increases, as it prioritized the connections and tended to serve the connections with the highest revenue even when the number was less.

6.2 Future Research

There are several possible research area that directly follow from our work. Below are some of the most important ones:

As a future step, we will further investigate the impact of a different aggregation factor, different pricing options in a network virtualization environment and the distributed cloud network topology on performance, energy consumption, and revenue of the virtual network mapping requests. Design and implementation of a comprehensive prototype of the network virtualization environment is of utmost importance. Moreover, instead of the simple revenue calculation used in this work and in the existing literature, an advanced model can be proposed. The virtual network mapping problem in cloud environment needs to be formulated, this formulation can be used to develop other virtual network provisioning algorithms. Moreover, a VM migration method that captures the network topology, the available resources of the hosting servers and real time communication between multiple virtual machines needs to be developed in order to minimize the total migration time and maintain QoS of the hosted services.

Bibliography

[1] N. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *Communications Magazine, IEEE*, vol. 47, pp. 20–26, July 2009.

[2] I. Bojanova and A. Samba, "Analysis of cloud computing delivery architecture models," in *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*, pp. 453–458, March 2011.

[3] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, p. 13, 2009.

[4] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, pp. 44–51, Aug 2009.

[5] M. Sharkh, M. Jammal, A. Shami, and A. Ouda, "Resource allocation in a network-based cloud computing environment: design challenges," *Communications Magazine, IEEE*, vol. 51, pp. 46–52, November 2013.

[6] " Amazon Web Services, Amazon elastic compute cloud (Amazon EC2) ." `http://aws.amazon.com/fr/ec2/`. [Online; accessed June-2013].

[7] " Microsoft Azure ." `http://azure.microsoft.com`. [Online; accessed June-2014].

[8] " Google Cloud Platform Home Page ." `https://cloud.google.com/products/app-engine`. [Online; accessed June-2014].

[9] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, (New York, NY, USA), pp. 164–177, ACM, 2003.

[10] Vmware, "Virtualization Basics,." `https://www.vmware.com/in/virtualization/virtualization-basics/what-is-virtualization`, 2014. [Online; accessed May-2014].

[11] K. Eric, "What desktop virtualization really means,." http://www.infoworld.com/d/virtualization/what-desktop-virtualization-really-means-352, 2013. [Online; accessed Feb-2014].

[12] N. Ruest and D. Ruest, *Virtualization, A Beginner's Guide*. McGraw-Hill, Inc., 2009.

[13] L. Nicholas, "Network virtualization: A new building block of network design.," 2007.

[14] R. Chowdhury, M.and Boutaba, "Network virtualization: state of the art and research challenges," *Communications Magazine, IEEE*, vol. 47, no. 7, pp. 20–26, 2009.

[15] Z. Zhang, S. Su, X. Niu, J. Ma, X. Cheng, and K. Shuang, "Minimizing electricity cost in geographical virtual network embedding," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 2609–2614, Dec 2012.

[16] B. Wang, X. Chang, J. Liu, and J. Muppala, "Reducing power consumption in embedding virtual infrastructures," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, pp. 714–718, Dec 2012.

[17] IBM, "Virtualization Basics,." http://pic.dhe.ibm.com/infocenter/tivihelp/v28r1/index.jsp?topic=%2Fcom.ibm.tivoli.tpm.scenario.doc%2Fvirtual%2Fccom_basics.html. [Online; accessed May-2014].

[18] S. Lowe, "Back to basics (Part 1): Virtualization 101,." http://www.virtualizationadmin.com/articles-tutorials/general-virtualization-articles/back-to-basics-part1.html, September 2011. [Online; accessed May-2014].

[19] L. Yan, "Development and application of desktop virtualization technology," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pp. 326–329, May 2011.

[20] E. Khnaser, "Desktop Virtualization Challenges - Introduction to Desktop Virtualization,." http://www.petri.co.il/desktop-virtualization-xendesktop-5-part-3.htm, Oct.

[21] J. Daniels, "Server virtualization architecture and implementation," *Crossroads*, vol. 16, pp. 8–12, Sept. 2009.

[22] T. Clark, *Storage virtualization: technologies for simplifying data storage and management*. Addison-Wesley Professional, 2005.

[23] F. Bunn, N. Simpson, R. Peglar, and G. Nagle, "Storage Virtualization,." http://www.snia.org/sites/default/files/sniavirt.pdf, OCtober.

[24] Microsoft, "Application Virtualization (App-V),." http://www.microsoft.com/en-in/windows/enterprise/products-and-technologies/mdop/app-v.aspx, 2014. [Online; accessed May-2014].

[25] G. Somani and S. Chaudhary, "Application performance isolation in virtualization," in *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on*, pp. 41–48, Sept 2009.

[26] B. Hedlund, "What is Network Virtualization?,." http://bradhedlund.com/2013/05/28/what-is-network-virtualization/, May 2013. [Online; accessed May-2014].

[27] J. Carapinha and J. Jiménez, "Network virtualization: A view from the bottom," in *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, (New York, NY, USA), pp. 73–80, ACM, 2009.

[28] J. Carapinha, P. Feil, P. Weissmann, S. E. Thorsteinsson, Ç. Etemoğlu, Ó. Ingþórsson, S. Çiftçi, and M. Melo, "Network virtualization-opportunities and challenges for operators," in *Future Internet-FIS 2010*, pp. 138–147, Springer, 2010.

[29] H. Kim and N. Feamster, "Improving network management with software defined networking," *Communications Magazine, IEEE*, vol. 51, pp. 114–119, February 2013.

[30] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *Communications Magazine, IEEE*, vol. 51, pp. 136–141, February 2013.

[31] " Amazon Web Services, Amazon elastic compute cloud (Amazon EC2) ." http://aws.amazon.com/en/ec2/. [Online; accessed June-2013].

[32] " Amazon Simple Storage Service, ." http://aws.amazon.com/ec2/. [Online; accessed June-2013].

[33] D. Carr, " How Google Works, ." http://www.baselinemag.com/crm/How-Google-Works/. [Online; accessed May-2014].

[34] A. Chandra, W. Gong, and P. Shenoy, "Dynamic resource allocation for shared data centers using online measurements," in *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, (New York, NY, USA), pp. 300–301, ACM, 2003.

[35] C. Patel, R. Sharma, C. Bash, and A. Beitelmal, "Thermal considerations in cooling large scale high compute density data centers," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2002. ITHERM 2002. The Eighth Intersociety Conference on*, pp. 767–776, 2002.

[36] S. Greenberg, E. Mills, and B. Tschudi, "Best practices for data centers: Lessons learned from benchmarking 22 data centers,"

[37] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, pp. 826–831, May 2010.

[38] C. Patel, C. Bash, R. Sharma, M. Beitelmal, and R. Friedrich, “Smart cooling of data centers,” in *ASME 2003 International Electronic Packaging Technical Conference and Exhibition*, pp. 129–137, American Society of Mechanical Engineers, 2003.

[39] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle, “Managing energy and server resources in hosting centers,” in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, SOSP ’01, (New York, NY, USA), pp. 103–116, ACM, 2001.

[40] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, “No "power" struggles: Coordinated multi-level power management for the data center,” in *Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS XIII, (New York, NY, USA), pp. 48–59, ACM, 2008.

[41] E. Geisa, “ DATA CENTER VIRTUALIZATION Q&A, .” `http://www.f5.com/pdf/white-papers/dc-virtualization-wp.pdf`. [Online; accessed June-2013].

[42] Vmware, “ Data Center Virtualization and Cloud Infrastructure Products, .” `http://www.vmware.com/in/products/datacenter-virtualization`. [Online; accessed June-2014].

[43] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[44] S. Zhang, S. Zhang, X. Chen, and X. Huo, “Cloud computing research and development trend,” in *Future Networks, 2010. ICFN ’10. Second International Conference on*, pp. 93–97, Jan 2010.

[45] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break in the clouds: Towards a cloud definition,” *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 50–55, Dec. 2008.

[46] L. Wu, S. Garg, and R. Buyya, “Sla-based resource allocation for software as a service provider (saas) in cloud computing environments,” in *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on*, pp. 195–204, May 2011.

[47] K. Kumar, J. Feng, Y. Nimmagadda, and Y. Lu, “Resource allocation for real-time tasks using cloud computing,” in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pp. 1–7, July 2011.

[48] P. Hofmann and D. Woods, “Cloud computing: The limits of public clouds for business applications,” *Internet Computing, IEEE*, vol. 14, pp. 90–93, Nov 2010.

[49] Y. Jadeja and K. Modi, “Cloud computing - concepts, architecture and challenges,” in *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, pp. 877–880, March 2012.

[50] J. Yang and Z. Chen, “Cloud computing research and security issues,” in *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pp. 1–3, Dec 2010.

[51] S. Rajan and A. Jairath, “Cloud computing: The fifth generation of computing,” in *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pp. 665–667, June 2011.

[52] K. Kim, A. Beloglazov, and R. Buyya, “Power-aware provisioning of cloud resources for real-time services,” in *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*, MGC ’09, (New York, NY, USA), pp. 1:1–1:6, ACM, 2009.

[53] G. Sun, H. Yu, V. Anand, D. Liao, and L. Li, “Exploring power-efficient provisioning for online virtual network requests,” in *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on*, pp. 51–55, Oct 2012.

[54] P. Endo, A. de Almeida Palhares, N. Pereira, G. Goncalves, D. Sadok, J. Kelner, B. Melander, and J.-E. Mangs, “Resource allocation for distributed cloud: concepts and research challenges,” *Network, IEEE*, vol. 25, pp. 42–46, July 2011.

[55] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervello-Pastor, and A. Monje, “On the optimal allocation of virtual resources in cloud computing networks,” *Computers, IEEE Transactions on*, vol. 62, pp. 1060–1071, June 2013.

[56] N. Chowdhury, M. Rahman, and R. Boutaba, “Virtual network embedding with coordinated node and link mapping,” in *INFOCOM 2009, IEEE*, pp. 783–791, April 2009.

[57] V. Vinothina, D. Sridaran, and D. PadmavathiGanapathi, “A survey on resource allocation strategies in cloud computing,” *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 6, 2012.

[58] M. Abu Sharkh, A. Ouda, and A. Shami, “A resource scheduling model for cloud computing data centers,” in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp. 213–218, July 2013.

[59] M. Chowdhury, M. Rahman, and R. Boutaba, “Vineyard: Virtual network embedding algorithms with coordinated node and link mapping,” *Networking, IEEE/ACM Transactions on*, vol. 20, pp. 206–219, Feb 2012.

[60] M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding: Substrate support for path splitting and migration,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 17–29, Mar. 2008.

[61] B. Wang, T. Li, X. Luo, and Y. Fan, “Traffic grooming under a sliding scheduled traffic model in wdm optical networks,” in *IEEE Workshop on Traffic Grooming in WDM Networks*, 2004.

[62] Z. Wang, J. Wu, Y. Wang, N. Qi, and J. Lan, “Survivable virtual network mapping using optimal backup topology in virtualized sdn,” *Communications, China*, vol. 11, pp. 26–37, Feb 2014.

[63] P. Lin, J. Bi, and H. Hu, "Vcp: A virtualization cloud platform for sdn intra-domain production network," in *Network Protocols (ICNP), 2012 20th IEEE International Conference on*, pp. 1–2, Oct 2012.

[64] C. Papagianni, G. Androulidakis, and S. Papavassiliou, "Virtual topology mapping in sdn-enabled clouds," in *Network Cloud Computing and Applications (NCCA), 2014 IEEE 3rd Symposium on*, pp. 62–67, Feb 2014.

[65] T. Sunay, "Of Controllers and Why Nicira Had to Do a Deal, Part III: SDN and Openflow Enabling Network Virtualization in the Cloud." `http://pluribusnetworks.com/blog`, 2012. [Online; accessed June-2014].

[66] K. Alhazmi, M. Abusharkh, D. Ban, and A. Shami, "A map of the clouds: Virtual network mapping in cloud computing data centers," in *IEEE Canadian Conference on Electrical and Computer Engineering 2014 (CCECE 2014)*, May 2014.

[67] A. Dharwadker, *The independent set algorithm*. CreateSpace Independent Publishing Platform, October 2011.

[68] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–12, April 2006.

[69] C. Wang, S. Shanbhag, and T. Wolf, "Virtual network mapping with traffic matrices," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 2717–2722, June 2012.

[70] Sun.G, V. Anand, H. Yu, D. Liao, and L. Li, "Optimal provisioning for elastic service oriented virtual network request in cloud computing," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 2517–2522, Dec 2012.

[71] M. Rahman and R. Boutaba, "Svne: Survivable virtual network embedding algorithms for network virtualization," *Network and Service Management, IEEE Transactions on*, vol. 10, pp. 105–118, June 2013.

[72] A. Leivadeas, C. Papagianni, and S. Papavassiliou, "Energy aware networked cloud mapping," in *Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on*, pp. 195–202, Aug 2013.

[73] J. M. Kaplan, W. Forrest, and N. Kindler, "Revolutionizing data center energy efficiency," tech. rep., Technical report, McKinsey & Company, 2008.

[74] A. Fischer, M. Beck, and H. de Meer, "An approach to energy-efficient virtual network embeddings," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pp. 1142–1147, May 2013.

[75] G. Sun, V. Anand, D. Liao, C. Lu, X. Zhang, and N.-H. Bao, "Power-efficient provisioning for online virtual network requests in cloud-based data centers," 2013.

[76] S. Su, Z. Zhang, X. Cheng, Y. Wang, Y. Luo, and J. Wang, "Energy-aware virtual network embedding through consolidation," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 127–132, March 2012.

[77] S. Su, Z. Zhang, A. Liu, X. Cheng, Y. Wang, and X. Zhao, "Energy-aware virtual network embedding," 2014.

Curriculum Vitae

Name: Khaled Alhazmi

Objective To participate with well-planned research group and to be able to utilize my skills, experience, and education in the field of Electrical and Computer engineering.

Education Ph.D. - Electrical and Computer Engineering
Western University - London, Ontario, Canada
2014 - Present

Master of Engineering Science - Electrical and Computer Engineering
With 95% cumulative average
Western University - London, Ontario, Canada
2012 - 2014

Bachelor Degree in Computer Engineering - College of Computer and Information Science
GPA 4.36 / 5.00 with second class Honour
King Saud University - Riyadh, Saudi Arabia
2002 - 2007

Accomplishments: Master of Engineering Science thesis:

- Thesis:
  - Virtual Network Provisioning in Distributed Cloud Computing Data Centers
- Projects completed:
  - Resource Allocation in Cloud Computing Environment, A literature survey
  - Optimal Resource Allocation for Multimedia Cloud in Priority Service Scheme
  - Software Defined Networking, A literature Survey
  - Novel Techniques for Multiple Antenna Transmission and Detection

Bachelor Graduation Project:

- Network Intrusion Detection System using Hidden Markove Model Tool-Kit HTK

Researcher of industry projects:

- King Abdulaziz City For Science and Technology - Electronic and Computer research Institute as a researcher and project manager:
  - Interactive Voice Response System project (One year)
  - Automatic Speech Recognition ASR project (One year)
  - Text To Speech TTS project (One year)
  - Optical Character recognition project using Hidden Markove Model Tool kit HTK (2 Years)

Project team member:

- "King Abdullah Initiative for Arabic Content". Supported by KACST (2007-2009).

Committee member:

- Organizing the Second International Symposium on Computer and Arabic Language: Content Industry (10-12 October 2009) Riyadh.
- Organizing the Science and Technology National Week. September 2009.

Main researcher:

- The Image Processing and Signal Analysis and Recognition (IPSAR) research group.

Patent and Publications:

Patents:

- M S Khorsheed, H Al-Omari, K Alfaifi and K Alhazmi, "System and Methods for Arabic Text Recognition Based on Effective Arabic Text Feature Extraction", US 12/430,773, FILED April 27,2009, PUBLISHED October 28, 2010.

- H Al-Omari and M S Khorsheed ,K Alfaifi, K Alhazmi, "Method and System for Preprocessing an Image for Optical Character Recognition", US 12/430,773, FILED April 27, 2009; PUBLISHED October 28, 2010.

Publications:

- K. Alhazmi, M. Abusharkh, D. Ban, and A. Shami, "A map of the clouds: Virtual network mapping in cloud computing data centers," in *IEEE Canadian Conference on Electrical and Computer Engineering 2014 (CCECE 2014)*, (Published: May, 2014).
- K. Alhazmi, M. Abusharkh, D. Ban, and A. Shami, "Drawing the Cloud Map: Virtual network embedding in cloud computing Environment," in *IEEE Systems Journal*, (Submitted: June, 2014).
- K. Alhazmi, and A. Shami, "A Greener Cloud: Energy Efficient Provisioning for Online Virtual Network Requests in Cloud Data Centers," in *IEEE International Conference on Communications (ICC)*, (To be submitted: September, 2014).
- M S Khorsheed, K Alhazmi and A Asiri, "Developing Typewritten Arabic Corpus with Multi-fonts (TRACOM)", MOCR '09: Proceedings of the International Workshop on Multilingual OCR , Catalonia - Spain 25th July, pp. 1-6 , 2009.

Training:

- Lean Six Sigma, king Abdullah University for Science and Technology KAUST, Saudi Arabia - 2013
- Advanced Java Training Course - TriOS College, London, ON, 1 Month, May 2012.
- Cisco Certified Network Associate (CCNA) 25 Feb 2009.
- Intel Software College, Intel Multi-Core Programming, 16-18 February 2008.
- KACST, Advanced course in JAVA language, 13-27 October 2008.
- RDI Company (Egypt), Speech Processing Course, 2-13 November 2008.
- RDI Company (Egypt), Automatic Speech Recognition ASR and Text to Speech TTS using Hidden MArkove Model course, 3 Aug - 4 Sep 2010.
- Emcanat Training Center, project Management Basic Course, 19-26 November 2007 (Total 35 Hours).
- Emcanat Training Center, Microsoft Project In Project Management, 27-28 November 2007 (Total 35 Hours).
- Microsoft Arabia, Microsoft Approved Course "Managing Projects Using MS Project 2007", 12-23 January 2008.
- Microsoft Arabia, Microsoft Approved Course "5060: Implementing Windows Share Point Services 3.0", 01-05 March 2008.
- Bahrain Society of Engineers Training Center, Technical Report Writing, 16-19 March 2008.
- Matlab and Simulink course, Singapore, One week.
- Diploma in the art and science of Neuro Linguistics Programming, 20 The FEB in year 2003
- Certificate from Global NLP Training and Consulting Community (G.T.C), 20 th FEB 2003.
- Nomination to represent the Kingdom by the ministry of Communications and Information Technology at the ITU (International Telecommunication Union) Telecom World 2006 in China.

Employment History:

- Researcher and project manager: King Abdulaziz City For Science and Technology - Computer and Electronic Research Institute, 2007-present,
- Main researcher: The Image Processing and Signal Analysis and Recognition (IPSAR) Group, 2007-2010
- Network administrator: King Saud University 2004-2006
- Teaching assistant:
  - Western University, Faculty of Engineering, 2012 -2014
  - King Saud University, College of Computer and Information science, 2005- 2007.
- Wireless Network Administrator: Summer Training, Saudi Aramco, 2005