



UNIVERSITAT POLITECNICA DE CATALUNYA
MASTER OF SCIENCE IN INFORMATION AND
COMMUNICATION TECHNOLOGIES

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Made by: Alejandro Gutierrez Anato

Directors: Xavier Hesselbach
Jose Roberto Amazonas

November 29th, 2014

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

One of the main challenges of network virtualization is the virtual network-embedding problem (VNE). The objective of the VNE is to map a set of Virtual Network Request (VNR) to a physical node and link. VNR is composed by a set of virtual nodes and links with several demands (Processing power, Bandwidth.), which they need to be assigned into a set of paths in the substrate network with sufficient resources to accomplish their demands. Furthermore, these embedding can be optimized with regard to several parameters, such as: embedding cost, link bandwidth, energy- efficiency, packet loss rate, throughput, etc. [3].

To solve the VNE program a mathematical tools was proposed, called “Paths algebra”[4]. This framework can helped solve the multiple multi- constraint routing problems of VNE using linear metrics as bandwidth, number of hops and delay, or non-linear metrics as availability and package loss rate

Most of the existing VNE proposals treat the single-path virtual link-mapping problem as a mono-constraint, that is, their objective is to map the virtual links in substrate paths that minimize/maximize the usage of one resource (typically bandwidth). This paper introduces a virtual link mapping approach supporting multiple constraints using the “paths algebra” routing framework.

Keywords: Network Virtualization, Paths-Algebra, Virtual Network Embedding/Mapping, Nonlinear metrics, linear metrics.

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

Acknowledgements

First and foremost I would like to mention my parents, my brother and my girlfriend; I want to thank them for all their support, patience encouraging words throughout this stage of my life, without them none of this would have been possible.

To Dr. Xavier Hesselbach and Dr. Jose Roberto Amazonas, who had been the tutors of this project guiding me every step of the way as well as encouraging me in moments of difficulty. Their support has been a key element during the development of this Final Master Project.

Many were those who accompanied me in my academic career, and in my Master studies at this final stage of preparing my thesis. Very important people that supported and helped me, and managed to always keep me focus on what truly was useful and beneficial. My eternal gratitude to all of them.

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

Table of Contents

Acknowledgements.....ii

Table of Contents iii

List of Illustrationsvi

List of Tables viii

CHAPTER I 1

Introduction1

 I.1. Scenario.....2

 I.2. Motivation.....3

 I.3. Justification.....3

 I.4. General Objective.....4

 I.5. Specific Objectives.....4

 I.6. Limitation and Viability5

 I.7. Organization of the Master Thesis.....5

CHAPTER II 7

Literature Review7

 I.1. Virtualization.....7

 I.1.2. Types of Virtualization.....9

 I.1.2.3. Application virtualization9

 I.1.2.4. Processing virtualization.....9

 I.1.2.5. Storage virtualization10

 I.1.2.6. Network virtualization10

 II.2 Network virtualization.....10

 II.2.1 Network Virtualization Components11

 II.2.2. Business roles.....12

 II.2.3. Network Virtualization Challenges.....13

 II.2.4. Resource allocation in Virtual Networks.....14

 II.3. Network as a Service.....15

 II.3.1. Functionality.....16

 II.3.1.1. Network visibility.....16

 II.3.1.2. Custom forwarding.....16

 II.3.1.3. In-network processing.....17

 II.3.2. Requirements17

 II.3.2.1.Integration with current hardware.....17

 II.3.2.2. High-level programming model.....17

 II.3.2.3. Scalability and multi-tenant isolation.....17

 II.3.2.4. Application performance17

 II.3.2.5. Topology-dependent complexity18

 II.3.2.6. Location dependency.....18

 II.4. Infrastructure as a Service18

 II.4.1. Roles in IaaS20

 II.4.1.1. End User’s View of IaaS:21

 II.4.1.2. Provider’s View of IaaS.....21

 II.5. Virtual network mapping/embedding.....21

 II.5.1 Virtual Network embedding problem formulation.....22

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

II.5.2. Virtual Network previous proposed solutions	23
II.6. Path algebra framework.....	25
II.6.1. Paths Formulation.....	26
II.6.2. Path Ordering.....	27
II.6.3. Node and links Metric.....	28
II.6.3.1 Node Metric	29
II.6.3.1.1. CPU consumption	29
II.6.3.1.2. Energy.....	31
II.6.3.2. Link Metrics	31
II.6.3.2.1 Bandwidth	31
II.6.3.2.2. Throughput	32
II.6.3.2.3. Packet Loss Ratio	32
II.6.4. Evaluation Metric.....	32
II.6.5. Path and Virtual Network ordering.....	33
II.6.5.1. Path Ordering.....	33
II.6.5.2. Virtual Network Ordering	33
II.6.6. Node and Link Mapping.....	34
II.6.7. Example using default metric.....	35
CHAPTER III	41
Alevin Framework.....	41
III.1. Installation of the simulating environment.....	42
III.1.1. For MacOS.....	42
III.2. Running the simulation.....	45
III.3. MATLAB files	49
III.3.2. Files generated by ALEVIN	51
III.4 Evaluation metrics generation.....	52
CHAPTER IV	55
Scenarios	55
IV.1. Topology Creation	55
IV.1.1. Substrate Network in MATLAB	56
IV.2. Resource and demand deployment	56
IV.3. Linear and Non Linear Metrics	58
IV.3.1. Packet Loss Ratio.....	59
IV.3.1.1. Inclusion into the simulation environment.....	59
IV.3.1.1.1 Metric Creation.....	59
IV.3.1.1.2. Synthesis	60
IV.3.2 Availability	61
IV.3.2.1 Inclusion into the simulation environment.....	61
IV.3.2.1.1. Metric Creation.....	61
IV.3.2.1.2. Synthesis	62
IV.3.3. Energy	63
IV.3.3.1. Inclusion into the simulation environment.....	63
IV.3.3.1.1. Metric Creation.....	63
IV.3.3.1.2. Synthesis	64
IV. 4. Metric Priority.....	65
IV.5. Simulation Parameters.....	66
CHAPTER V	67
Results	67
V.1.Simulation Results	67
V.1.1. Packet Loss Ratio	67

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

V.1.2. Availability	71
V.1.3. Hops	74
V.1.4. Energy	76
V.1.5. Cost/Revenue	80
V.1.6. Percentage Accepted VNR.....	82
CHAPTER VI	85
Conclusion and Recommendation	85
VI.1. Conclusion.....	85
VI.2. Recommendation.....	88
REFERENCES	90
List of Acronyms.....	94
Annexes A	95
Annexes B	111

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

List of Illustrations

Figure 1. Virtualization types source.....	8
Figure 2. Virtual network.....	12
Figure 3. Network Virtualization Business Roles source	14
Figure 4. Two VN mapped in a SN.....	15
Figure 5. Infrastructure as a Service source	20
Figure 6. Allocation of VN. source	23
Figure 7. Representation of a Network.....	27
Figure 8. Path ordering example.....	28
Figure 10. New restringing condition.....	30
Figure 11. Bandwidth transformation	31
Figure 12. Example of two VNR and one SN.	37
Figure 13. SN after first mapping	38
Figure 14. SN after first VN.....	40
Figure 16. Processor and memory	44
Figure 17. Setting windows OS.....	44
Figure 18. Processor and memory Windows OS.....	45
Figure 19. Eclipse	46
Figure 20. "PathsAlgebraAR" java Class	46
Figure 21. Scenario Parameters	47
Figure 22. Scenario Parameters	48
Figure 23. Running the Test.....	48
Figure 24. Matlab File creator	49
Figure 25. Evaluation Metric Parameters	53
Figure 26. Evaluation files	54
Figure 27. Diagram of the ALEVIN online simulation.....	54
Figure 28. Substrate network example	56
Figure 29. Packet Loss Ratio in SN.....	59
Figure 30. Path1 Packet Loss Rate	60
Figure 31. Path 2 Packet Loss Ratio.....	60
Figure 32. Availability in SN.....	61
Figure 33. Path 1 Availability	62
Figure 34. Path 2 Availability	62
Figure 35. Energy consumption in SN	63
Figure 36. SN example.....	64
Figure 37. New metric combination.....	64
Figure 38. Path 1 Energy Consumption	65
Figure 39. Path 2 Energy Consumption	65
Figure 40. Packet Loss Ratio value.....	68
Figure 41. Virtual Request 1	68
Figure 42. Virtual Request 2	69
Figure 43. Average Packet Loss Rate	70
Figure 44 Availability of SN.....	72
Figure 45.Average Availability	73

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

Figure 46. Substrate Network Hops	74
Figure 47. Hops Average	76
Figure 48. Substrate Network Energy	77
Figure 49. Average Energy	78
Figure 50. Energy Path 1	79
Figure 51. Energy Path 2	79
Figure 52. Cost/Revenue.....	81
Figure 53. Percentage Accepted VNR	83

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

List of Tables

Table 1. Path Ordering.....	28
Table 2. Mapping (c,d).....	38
Table 3. Mapping path (a,c).....	39
Table 4.....	40
Table 5.....	40
Table 6. Substrate Network file.....	50
Table 7. Bandwidth metric file.....	50
Table 8. CPU metric file.....	51
Table 9. Substrate Network adjacency matrix.....	56
Table 10. Packet Loss Rate adjacency matrix.....	60
Table 11. Availability Metric.....	62
Table 12. Energy consumption metric.....	63
Table 13. Scenarios and metric priority.....	66
Table 14. Simulation Parameters.....	66
Table 15. Virtual Network 1.....	69
Table 16. Virtual Network 1.....	69
Table 17. Average PLR vs Loads. Scenario 1.....	70
Table 18. Virtual Network 1 Availability.....	72
Table 19. Virtual Network 1 Availability.....	72
Table 20. Average Availability vs Loads. Scenario 1.....	73
Table 21. Virtual Network 1 Hops.....	75
Table 22. Virtual Network 2 Hops.....	75
Table 23. Hops vs Loads. Scenario 1.....	76
Table 24. Virtual Network 1 Energy.....	77
Table 25. Virtual Network 2 Energy.....	78
Table 26. Energy vs Loads. Scenario 1.....	80
Table 27. C/R vs Loads. Scenario 1.....	81
Table 28. Percentage Accepted VNR vs Loads. Scenario 1.....	83

Chapter I

Introduction

Nowadays, the new Internet architecture needs to provide different services and application to users and at the same time effectively manage the consumption of resources. Furthermore, hardware within these networks has also become exceedingly complex because they implement an ever-increasing number of standardized distributed protocols and proprietary interfaces. [1]

In this environment, service providers, researchers, and even vendors continue to innovate. Service providers, continue to look for techniques to customize and optimize networks for their use cases, including the application set relevant to their business. Researchers seek techniques to conduct research with their projects at scale.

However, the current IP-based Internet protocol along with the huge amount of investment in the Internet infrastructure make any disruptive innovation in the Internet architecture very difficult [2]. Nevertheless, one technology stands out among others, Infrastructure as a Service (IaaS) [3]. IaaS is defined as a service in which the providers owns and host computer, network and storage resources and there are offered to the user on-demand. IaaS is divided in two roles. The Infrastructure provider (InP) who owns, deploys and maintains the network infrastructure and the Service Provider (SP) responsible for deploying network protocols and offer end-to-end services [3]. The job of the SP is to perform the optimal allocation of the demanded services over the physical network, which is owned by the infrastructure providers.

Network virtualization will be a key enabler for IaaS that will allow virtual networks to coexist in a same physical network and isolated from each other. In these virtual networks, service can be provided by effectively sharing and using underlying network resources provided by infrastructure providers [2]. This can have a significant impact on the IaaS and can provide flexibility, promotes diversity, and promises increased manageability and stimulate the development and deployment of new Internet services.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Network virtualization has different elements; being the main one the Virtual Network (VN), composed of Virtual Nodes hosted on substrate nodes and interconnected by physical links [4]. In addition, multiple VN can share the same physical resources. This mapping has to be done in an effective and efficient way in order to increase the utilization of the substrate network (SN) resources and the revenue. This problem is commonly known as the virtual network embedding/mapping (VNE).

The objective of the VNE is to map a set of Virtual Network Request (VNR) to a physical node and link. VNR is composed by a set of virtual nodes and links with several demands (Processing power, Bandwidth.), which they need to be assigned into a set of paths in the substrate network with sufficient resources to accomplish their demands. Furthermore, these embedding can be optimized with regard to several parameters, such as: embedding cost, link bandwidth, energy- efficiency, packet loss rate, throughput, etc. [3].

To solve the VNE program a mathematical tools was proposed, called “Paths algebra”[4]. This framework can helped solve the multiple multi- constraint routing problems of VNE using linear metrics as bandwidth, number of hops and delay, or non-linear metrics as availability and package loss rate

I.1. Scenario

In today’s world most of the companies wants to provide several services and applications to the client in ways effective and less costly. To achieve this, the IaaS’s business model was created. The IaaS can provide customized services to a specific group of users and perform optimal allocation of them in the physical network and at the same time deliver and end-to-end Quality of Service.

Network virtualization will be a fundamental technology in the IaaS by allocating several virtual networks on top of a substrate network. The effective allocation of these resources is the objective of the VNE.

Most of the existing VNE proposals treat the single-path virtual link-mapping problem as a mono-constraint, that is, their objective is to map the virtual links in

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

substrate paths that minimize/maximize the usage of one resource (typically bandwidth). This paper introduces a virtual link mapping approach supporting multiple constraints using the “paths algebra” routing framework.

VNE link mapping corresponds to finding the best route(s) on the substrate network for each virtual link in the VN, where best implies the adoption of some optimization criteria.

In previous work [3], it was proposed the utilization of a mathematical multi-constraint routing framework called “paths algebra” to solve the virtual link mapping stage of the Virtual Network Embedding. Paths algebra provides the flexibility to introduce an unlimited number of linear and non-linear constraints and metrics to the problem while finding all the eligible paths in the physical network resulting in better and more flexible embedding.

For now, this mathematical tool has been used to solve Virtual Network Embedding with linear metric, such as CPU, number of hops and Bandwidth. This Master Thesis focus in the introduction of three new non-linear and linear metrics. The metrics that are going to be included are Availability, Packet Loss Ratio (PLR) and Energy Consumption. The main idea is to study and analyzed how “Paths Algebra” framework can help solves the VNE problem considering multi-constraint mapping. In this work several scenario are going to be considering based on metric priority and analyze if “Paths Algebra” allocated the different Virtual Network based on those metrics.

I.2. Motivation

This master thesis main motivation is to contribute to the development of similar researched related to Virtual Network and Virtual Network embedding/mapping such as the work of Boteros[3],] A; Perelló[18] or Chowdhury [26]. In addition, this work will also provide an important step for further investigation of the Network virtualization as a technology for the future Internet architecture and as enabler to provide end-to-end QoS in the Infrastructure as a Service paradigm

I.3. Justification

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

This research provides an important step for further investigations regarding the optimal allocation of virtual network based on multi constraint routing. The impact on knowledge obtained from this investigation is the main justification.

As services demand more bandwidth and network links need to support numerous users, the future indicates that the answer to these needs requires not only increasing link speeds but more likely, taking better advantage of the available resources. This proposal allows that different virtual networks can be shared in one physical environments based on multi constraint routing; this supposes an efficient way to provide QoS and improve the use of resources between networks.

I.4. General Objective

The implementation of different optimization criteria to solve the virtual link mapping stage of the Virtual Network embedding problem using the “Paths Algebra” framework

I.5. Specific Objectives

- Study and analyze previous works and method related to solving the Virtual Network embedding/mapping problem
- Study and analyze the mathematical framework of “Path Algebra” and the simulation software that will be used for the tests through predefined metrics and examples
- Analyze the generation of the VNR and substrate network and study how the mapping is achieved.
- Study how the physical constraint and default metric affect the mapping of the VNR in “Paths Algebra”.
- Study of the new metrics synthesis and formula use in order to include it into the simulation environment
- Inclusion of new non linear and linear metric in the simulation environment and “path algebra” framework
- Simulation of the different scenarios and comparison of the different results regarding metric priority.

I.6. Limitation and Viability

This Final Master Project includes:

- The explanation and analysis of Infrastructure as a service, virtual network and the virtual network-embedding problem. Also an explanation of why the selected tool is the “paths algebra” framework instead of others mathematical framework is stated.
- The design of a new strategy to include non linear metric which add improvements in the virtual network link mapping stage in terms of multi constraint routing, based on the knowledge obtained from previous studies
- The implementations of tests that will help demonstrate the adequate performance of this model.
- The Analysis of the obtained results, based on those found with different metric priority.

The limitations of this project:

- Much of the needed information during the development of this Special Master Project was hard to find because it’s a topic that is being investigated recently. Therefore many of the related works used to make the research have not been published yet.
- Given the complexity of the ALEVIN environment and the time that took to learn how to use it, more metric could not be included for comparison with those already implemented in this Master Thesis.

I.7. Organization of the Master Thesis

This Master Thesis is divided in six chapters. In Chapter I, introduction is presented. The scenario, motivation and justification regarding the thesis are stated. Also the general objective, specifics, viability and also limitations of this work are formulated

The second chapter summarizes the theoretical fundamentals that were used through the investigation, which facilitates the understanding of Infrastructure as a Service, Virtual Network, Virtual Network embedding problem and Path Algebra.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

In the third chapter, the configuration and installation of the simulation environment is explained. In the fourth chapter the scenarios are explained, also how the SN and resources and demands are deployed, an explanation of the procedure use to include the new non-linear metric into the ALEVIN framework is explained. In addition the synthesis and formula used for every metric is stated, and the parameters for every simulation.

In the fifth chapter, the results obtained in the investigation are presented and analyzed. Finally, the sixth chapter includes the conclusions reached during the project development. Additionally, it outlines some recommendations to be followed by further research, to strengthen the settlement proposal, which are points of interest to this area of research.

Chapter II

Literature Review

This chapter presents the main theoretical aspects, which are related to the scope of the thesis, emphasizing concepts that served as the basis of its development. First of all, an explanation of the Virtualization concept, the network virtualization, Network as a Service and Infrastructure as a Service. In second place, the formulation of the Virtual Network embedding problem is presented and some proposed solutions. Finally the Path Algebra framework, features and procedure are explained

Within this perspective, the following is the outline of the Literature Review that supports this investigation.

I.1. Virtualization

IBM introduced the concept of virtual machine in 1960's[5]; the idea was to introduce a virtual layer between the software and the hardware layer. Furthermore, with the virtualization technology an isolated environment between the applications executed can be done and shield the dynamics and heterogeneous of hardware platform, and support share and reuse of hardware resources. [5]

Virtualization is a way to abstract applications and their underlying components away from the hardware supporting them and present a logical or virtual view of these resources. This logical view may be strikingly different from the physical view. The goal of virtualization is usually one of the following: higher levels of performance, scalability, reliability/availability, agility, or to create a unified security and management domain. [6]

This virtual view is constructed using excess processing power, memory, storage, or network link, nodes or equipment.

Virtualization can create the artificial view that many computers are a single computing resource or that a single machine that is really many individual computers. It can make a single large storage resource appear to be many smaller ones or make many smaller storage devices appear to be a single device. [6]

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

One of the advantages of virtualization is cost reduction, simplicity of management and administration. Not surprisingly, there are dozens of virtualization products, and a number of small and large companies that make them. Some examples in the operating systems and software applications are VMware, Xen, and Microsoft virtualization products, to mention a few. Major IT players have also shown a renewed interest in the technology (e.g., IBM, Hewlett-Packard, Intel, Sun, RedHat). [7]

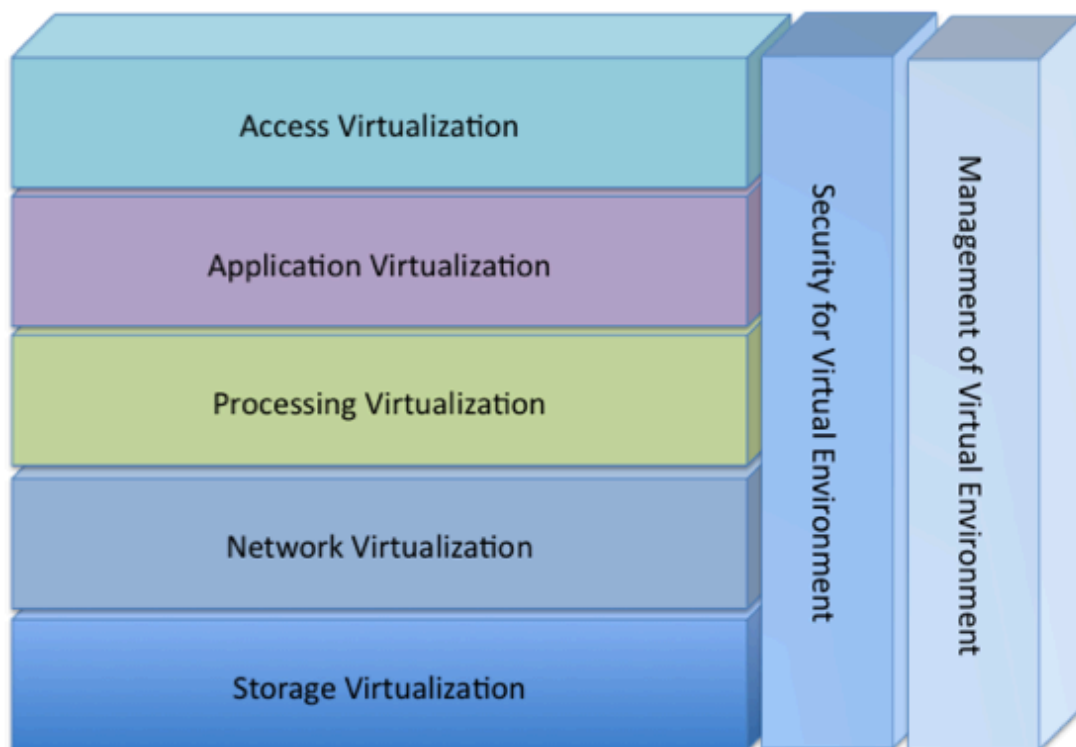


Figure 1. Virtualization types source (*Virtualization: A Manager's Guide.* "O'Reilly Media, Inc.", 2011.)

I.1.1. Goals of Virtualization

Organization are using more the virtualization technology to reach their goals, which can be the following:

- Allowing any network-enabled device to access any network-accessible application over any network, even if that application was never designed to work with that type of device
- Isolation of one workload or application from another to enhance security or manageability of the environment
- Isolation of an application from the operating system, allowing an application to continue to function even though it was designed for a different version of

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

the operating system

- Isolation of an application from the operating system, allowing an application to function on a foreign operating system
- Increasing the number of people that an application can support, by allowing multiple instances to run on different machines simultaneously
- Decreasing the time it takes for an application to run, by segmenting either the data or the application itself and spreading the work over many systems
- Optimizing the use of a single system, allowing it to work harder and more intelligently (that is, reducing the amount of time the processor sits idle)
- Increasing the reliability or availability of an application or workload through redundancy (if any single component fails, this virtualization technology either moves the application to a surviving system or restarts a function on a surviving system). [7]

I.1.2. Types of Virtualization

There are many layers of technology that can be virtualized; each layer will correspond to a type of virtualization:

I.1.2.1 Access virtualization

Hardware and software technology that allows nearly any device to access any application without either having to know too much about the other [6]

I.1.2.3. Application virtualization

This type of virtualization allowed that different application could run on many different operating system and hardware platform.

I.1.2.4. Processing virtualization

This type of virtualization technology can make one system appear to be many or many systems appear to be a single computing resource, to achieve goals ranging from raw performance, high levels of scalability, reliability/availability, agility, or consolidation of multiple environments into a single system [6].

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

I.1.2.5. Storage virtualization

This technology allows many systems to share the same storage devices without knowing that others are also accessing them. This technology also makes it possible to take a snapshot of a live system so that it can be backed up without hindering online or transactional applications [6].

I.1.2.6. Network virtualization

This type of virtualization allowed to isolated the computational and network resources through virtualization to allocate them to a logical (virtual) network for accommodating multiple independent and programmable virtual networks. On the next chapter a more in dept. explanation of the network virtualization will be stated.

II.2 Network virtualization

As explained before, the concept of virtualization was introduced in the Information Technology (IT) realm in the 1960s with the virtualization of the Virtual Machines (VM). The idea was to place a virtual layer between hardware and software layer, so by doing this you can create an abstraction of the physical resources, so the user believe that is interacting with the entire resources but you are only allowing to use an instance of this resource. As a result, it was possible to run multiple autonomous and isolated VMs on top of a single shared physical machine. [8]

The idea was to extend this concept into the network environment, so instead of the resources being CPU, servers or computers, we substitute it by nodes, links and data communication resulting in Virtual Private Networks. VPNs where initially thought as a way to provision logically separated private networks over a public infrastructure by tunneling data traffic between geographically distant sites, offering connectivity services between these sites. The network operator owning the physical infrastructure establishes the connections composing the VPN and provides them together as a service to the client. Note that in VPNs all the connections are established and managed by the network owner, resulting in a client-server relationship.

An evolution of the VPN is the Virtual Network (VN) paradigm, which results in one step ahead on the evolutionary path towards the Future Internet. The VN paradigm pleads for a more disruptive approach, where network owners can offer parts

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

of their infrastructures as services to external service providers, so that they are completely free to manage them to offer end-to-end services to final users. It aims at introducing the essentials of virtualization to network environments [8].

Network virtualization is defined by decoupling traditional Internet Service Providers (ISPs) into two independent entities: Infrastructure Provider (InP), who manages the physical infrastructure, and Service Provider (SP), who creates virtual networks by aggregating resources from multiple infrastructure providers and offers end-to-end services. [3]

Each service provider leases resources from one or more infrastructure providers to create virtual networks and deploys customized protocols and services, taking into account performance, topology, and cost of each infrastructure. [3]

II.2.1 Network Virtualization Components

Network virtualization basic element is the virtual network (VN), a virtual network is a collection of virtual nodes connected together by a set of virtual links to form a virtual topology, which is essentially a subset of the underlying physical topology or substrate node.

Another component is called Virtual Network Request (VNR). The SP on the analysis of the user service demands will generate this request. Each VNR contains a set of demands of networking and no networking parameters required by the demanded service.

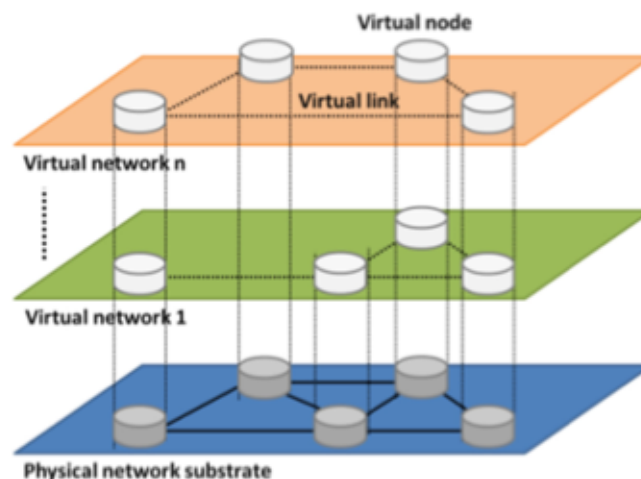


Figure 2. Virtual network

Each virtual Network is going to have virtual node that could be hosted on a particular physical node or could be a logical abstraction of a networking system [10]. Network virtualization is composed of two main components link virtualization and node virtualization. [9]

Link virtualization allows the transport of multiple separate virtual links over a shared physical link. The basic elements of network virtualization are shown in Figure 2. [10] At the substrate level, the substrate node is network equipment capable of supporting virtual nodes by means of any virtualization technology. A single substrate node typically contains a number of virtual nodes.

II.2.2. Business roles

The introduction of network virtualization separates the management and business roles of the SP by identifying three main players (figure 3[13]).

- **The Service Provider (SP)**, SPs lease resources from multiple InP's to create and deploy VNs by programming allocated network resources to offer end-to-end services to end-users. An SP can also provide network services to other SPs. It can also create child VNs by partitioning its resources and act as a virtual InP by leasing those child networks to other SPs [12]
- **Infrastructure Provider (InP)**: InPs deploy and actually manage the underlying physical network resources. They offer their resources through programmable interfaces to different SPs. InPs distinguish themselves through the quality of resources they provide, the freedom they delegate to their customers, and the tools they provide to exploit that freedom. [12]
- **The virtual network provider (VNP)** is responsible for finding and composing the adequate set of virtual resources from one or more infrastructure providers, in order to fulfill the virtual network operator request. The VNP leases slices of the virtualized infrastructure from one or more InPs and puts them together. Strictly speaking, what the VNP provides is not a network yet, but just an empty container where the virtual network operator builds the protocols that make up the VN. [9]

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

- **The virtual network operator (VNO)** deploys any protocol stack and network architecture over a VN, independently of the underlying physical network technologies. The VNO operates, maintains, controls and manages the virtual network. Ideally, the fact that resources are virtual, rather than physical, should not imply any major impact from an operational point of view. VNOs have a unified view of the network, regardless of the multiple infrastructure domains on which it is built [9].

II.2.3. Network Virtualization Challenges

In a network virtualization scenario, an InP must fulfill the network infrastructure provider requirements, as well as new requirements specifically related with virtualization of network resources [9]:

Robustness: The network should continue to operate in the event of node or link failure. [9]

Manageability: The InP's must have a view of the underlying physical topology, operational state, provisioning status, and other parameters associated with the equipment providing the virtual network. [9]

Traffic and network resource control: Traffic engineering and management techniques performed by the InP must not constrain the basic operation of a VN in any way.

Isolation: Mechanisms to guarantee deterministic degrees of isolation between virtual networks must be provided [9]. These means that if one VN is damaged or behaving erratically, this would not affect the performance of other VNs sharing the same resources

Scalability: These means that if the number of VNs increases to the order of thousand or more, these shall not affect the isolation of the different virtual networks. So any technical solution must be scalable to cope with any number of virtual networks

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Inter-domain interoperability: In the case of VNs spanning multiple infrastructure provider domains, seamless interoperability must be guaranteed [9].

On-demand provisioning: It must be possible to create, modify and remove VNs dynamically at request, with a minimal impact on the underlying infrastructure [11].

Network technology agnosticism: Network virtualization should not rely or depend on any specific network technology. In particular, multi-domain scenarios across dissimilar infrastructure domains should be possible [9].

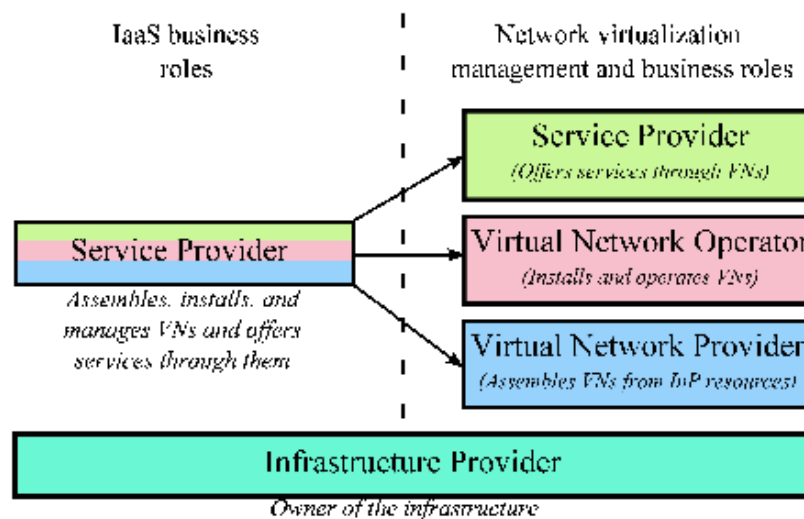


Figure 3. Network Virtualization Business Roles source ("Virtual Network Embedding: A Survey," *Communications Surveys & Tutorials, IEEE* , vol.15)

II.2.4. Resource allocation in Virtual Networks

One of the most important issues in network virtualization is an efficient utilization of SN resources. It will help to improve the resource utilization as well as avoiding congestion in the SN [11].

Typically, a substrate resource is partitioned to host several virtual resources. For example, any available substrate node can in principal, host a virtual node. Moreover, a single substrate node can host several virtual nodes. In some cases, substrate resources can also be combined to create new virtual resources. This is the

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

case for a virtual link, which spans several links (i.e. a path) in the substrate Network [13].

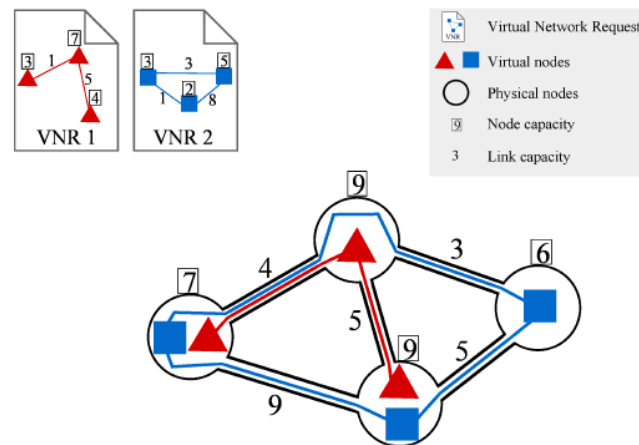


Figure 4. Two VN mapped in a SN

In figure 4 it can be observe that a single substrate node can contain several VNR, in this case there are two VNR with 4 nodes each one. So the substrate node can host several virtual nodes. Likewise, substrate links can host more than one virtual link. Moreover, one of the virtual links spans two substrate links, thus representing a virtual resource combined from several substrate resources [13].

Typically, there are some restrictions to be considered during the mapping. Most obviously, the candidate substrate resources for a mapping have to be able to support the performance requirements of the virtual resources. For example, a 1000 MBit/s virtual link cannot be mapped to a path containing a 100 MBit/s substrate link. Likewise, the CPU power requested by a virtual node has to be less than (or equal to) the CPU power actually provided by a substrate node. If redundancy is required, even more substrate resources may have to be reserved. Nevertheless, substrate resources should be spent economically [13]. Therefore, the mapping has to be optimized. This problem of mapping virtual resources to substrate resources in an optimal way is commonly known as the Virtual Network Embedding.

II.3. Network as a Service

In Network as a Service (NaaS) paradigm, networking resources such as transmission bandwidth, nodes CPU and control functions are demanded by users as

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

services.

NaaS enables abstraction of networking system into network services, which can be composed into composite network services by a network orchestration module to offer end-to-end services. NaaS supports a high-level abstraction of network infrastructure in form of services and allows flexible collaborating among autonomous systems via loose-coupling interactions between the services. Therefore, NaaS may greatly facilitate end-to-end QoS provision in the future Internet [14].

II.3.1. Functionality

For simplicity, we present the three functions of NaaS separately, although in practice they are used together.

II.3.1.1. Network visibility.

Most of the applications that can be use in NaaS are built on top of substrate network. A great effort must be made to optimize the mapping between the logical and physical. Because NaaS will need to handle a great amount of users, the network visibility is important. Several solutions for inferring network locality have been proposed. For example, Orchestra [14] uses a sophisticated clustering protocol to discover the network topology

While black-box approaches are necessary in an open environment such as the Internet, the providers has accurate knowledge of the topology and could make this information available to tenants at no extra cost. This would allow tenants to efficiently allocate virtual network to physical one, without requiring expensive and often inaccurate probing solutions [15].

II.3.1.2. Custom forwarding.

Network visibility would yield a significant performance improvement for providing different applications. However, there are some fundamental limits to the performance achievable using overlay networks. Since servers have usually only one NIC, even a simple multicast tree with a fan out greater than one cannot be optimally mapped to the physical network [15].

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Therefore, the second functionality that NaaS should provide the ability to control packet forwarding at switches. This would allow the implementation of custom routing protocols [15].

II.3.1.3. In-network processing.

The main benefits of NaaS come from providing an in-network packet processing capabilities. By performing in-network aggregation, it is possible to significant reduce the overall traffic sent over the network, thereby greatly reducing execution times. Note that these aggregation functions are application-specific and, hence, could not be provided as a traditional network service [15].

II.3.2. Requirements

In the previous section the function of the NaaS model were presented, but in order to use NaaS in a efficient way the following requirement most be satisfied:

II.3.2.1. Integration with current hardware.

Existing networks and data centers constitute a significant investment. The use of networking equipment, which typically lacks programmability features, reduces the cost of large deployments. For NaaS to become successful, it must not require expensive, non-commodity, networking hardware [15].

II.3.2.2. High-level programming model.

NaaS should expose a programming model that is natural for software developers to use, hiding low-level details of network packet processing and not exposing the full complexity of the physical network topology [15].

II.3.2.3. Scalability and multi-tenant isolation.

Compared to existing software-based router solutions NaaS must be able to support a multitude of different applications, written by different organizations and running concurrently, unaware of each other. Therefore, to be successful, a NaaS model requires strong isolation of the different network resources offered to users [15].

II.3.2.4. Application performance

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Most of the application deliver by the SP require some guaranteed bandwidth between server instances to satisfy user transactions within an acceptable time frame and meet predefined service level agreements (SLAs). Insufficient bandwidth between these servers will impose significant latency on user interactions. Therefore, without explicit control, variations in workloads and oversubscription can cause delay and drift of response time beyond acceptable limits, leading to SLA violations for the hosted applications [10].

II.3.2.5. Topology-dependent complexity

In NaaS, the topology of the network will try to match a predefined traffic requirement. For instance, a network topology optimized for the traffic among server in the center is not the same for the case of traffic from or to the Internet. The topology design also depends on how L2 and/or L3 are utilizing the effective network capacity [10].

II.3.2.6. Location dependency

Network equipment and servers have a statically configured physical network. Which means that they have a location dependency constraint .For instance, the IP address of a server is typically determined based on the VLAN or the subnet to which it belongs. VLANs and subnets are based on physical switch port configuration [6]. Therefore, a VM migration cannot be made smoothly without handling this problem. Furthermore, a constrained VM migration decreases the level of resource utilization and flexibility [10].

II.4. Infrastructure as a Service

According to [15], IaaS is defined as follows:

“Infrastructure as a Service (IaaS) is the delivery of hardware (server, storage and network), and associated software (operating systems virtualization technology, file system), as a service. It is an evolution of traditional hosting that does not require any long-term commitment and allows users to provision resources on demand. Unlike Platform as a Service, the IaaS provider does very little management other than keep the data center operational and users must deploy and manage the software services himself or herself just the way they would in their own data center. Amazon Web

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Services Elastic Compute Cloud (EC2) and Secure Storage Service (S3) are examples of IaaS offerings.”

Infrastructure as a Service (IaaS) is considered a prominent model for IP based service delivery [1]. Nowadays the demands of providing services to the users are increasing, for commercial development and also for the enterprises. However, enabling IaaS is a challenge for traditional Internet Service Providers, because it requires a high abstraction level of network architectures, protocols, and devices. Network control plane architecture plays therefore an essential role in this transition, particularly with respect to new requirements of scalability, reliability, and flexibility. [1]

The growing utilization of real-time services such as network telephony and video conferencing has resulted in a higher need for constant connectivity, which requires more scalable management. Infrastructure as a Service (IaaS) has been introduced to meet this new management, where complex underlying services remain hidden inside the infrastructure provider. Resources are allocated according to user need; hence the highest utilization and optimization levels can be achieved. During the duration of the service, the user owns and controls the infrastructure as if he was the owner. [1]

Infrastructure as a Service is a form of hosting. It includes network services, computing and storage [16] Figure 5. In the case of the network system of IaaS. The operator allocates and de-allocates network resources at close to real-time as required by the application/services. This ensures that consumer will only pay for their actual usage of the resources, rather than a fixed amount [17]. This will not only benefit the customers but also going to give higher revenues to the companies who are delivering the different services.

The IaaS provider will generally provide the hardware and administrative services needed to store applications and a platform for running applications. Scaling of bandwidth, memory and storage are generally included, and vendors compete on the performance [16].

Characteristics and components of IaaS include [16]:

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

- Utility computing service and billing model.
- Automation of administrative tasks.
- Dynamic scaling.
- Desktop virtualization.
- Policy-based services.
- Internet connectivity.

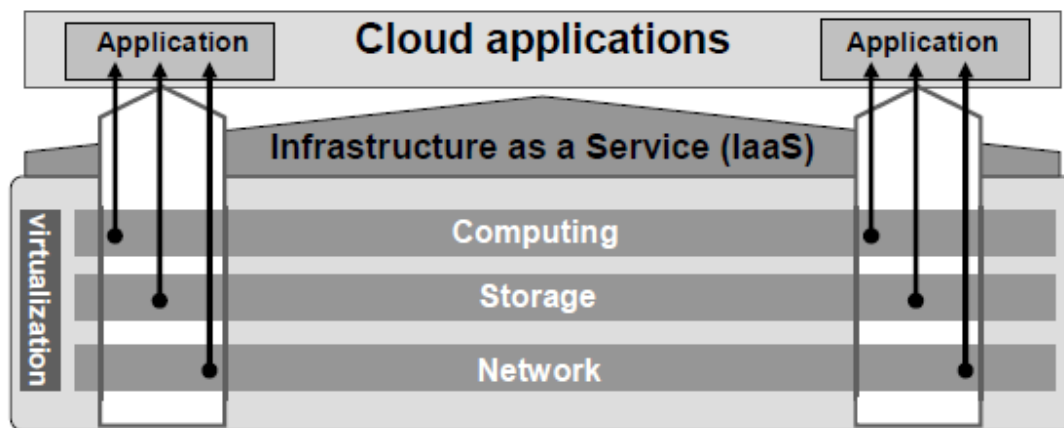


Figure 5. Infrastructure as a Service source (The network aspect of Infrastructure-as-a-Service," *Intelligence in Next Generation Networks (ICIN)*)

II.4.1. Roles in IaaS

IaaS is separated in three roles according to [1]:

- **Infrastructure Provider (InP):** the infrastructure owner who deploys and maintains the network equipment [1].
- **Service Provider (SP):** harvests network instances from one or more InPs and integrate them into a management domain. SP provides various services to end users (e.g. an IP Network service) [1].
- **End User:** uses the services offered by SP. InPs are able to partition their physical infrastructures, based on user/application requirements into virtual networks, and offer them as network services to service providers. A virtual infrastructure is a set of virtual resources interconnected together and managed

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

by a single administrative entity. The logically independent virtualized infrastructure provides a flexible pay as you go method that allows SP manage their resources efficiently on the dynamic demand from various end users [1].

II.4.1.1. End User's View of IaaS:

- Enable users to access applications from anywhere
- A modular system, which is flexible, scalable, virtualized and automated.
- Resilient and always available
- Enable to put applications and data on platform provisioning & maintenance by provider
- Own the hardware & nuances about provisioning & maintaining the OS & hygiene facts like space and power etc. [16]

II.4.1.2. Provider's View of IaaS

- Provide virtual infrastructure (server, storage and Network virtualization).
- Responsible for provisioning of space, power & cooling.
- Deploy web-based applications to easily provision infrastructure for customer on demand.
- Responsible to provide load balancing services.
- Eases the process of cloning apps on additional infrastructure instances.
- Service level agreements with customers on “availability of infrastructure services”
- In a dense, shared, and pooled environment, the security of CPUs, data, and network is paramount.
- Account Management & Provisioning. [16]

II.5. Virtual network mapping/embedding

The previous concepts were related to the Network and Infrastructure as a Service and how the network virtualization can be an enabler for these two services. Nevertheless in order to achieve the objective and been able to mapped the different virtual network into a physical one, the following concept is explained.

II.5.1 Virtual Network embedding problem formulation

As it was stated before there is a challenge in trying to allocate the different resources of the substrate network to the VNR, this is usually referred as the Virtual Network Embedding (VNE) problem. Through dynamic mapping of virtual resources onto a physical hardware, the benefit gained from existing hardware can be maximized. Optimal dynamic resource allocation, leading to the self-configuration and organization of future networks, will be necessary to provide customized end-to-end guaranteed services to end users [13].

This optimal allocation can be done to reach certain objectives, for instance Quality of Service or economical profit, energy-efficiency, etc. As we can see in figure 6 network virtualization make use of the embedding algorithm in order to allocate virtual resources on a physical infrastructure in an optimal way. The VNO uses embedding algorithms to decide which virtual resources to request from the VNP, who, in turn, instantiates them by using the InPs substrate resources [13].

The virtual network embedding can be solved in two stages: the first one is the virtual node mapping [13]. Which is responsible of allocating Virtual Nodes of the VNR in the physical nodes with enough capacity to accomplish the virtual node resource demand, and the Virtual Link Mapping [13] where the virtual link connecting these virtual nodes are mapped to a directed path in the substrate network with enough resource capacity to meet the virtual link demand.

Two main approaches exist in the context of VN embedding depending on the considered scenario. In the offline or static approach, all the VN demands are known in advance and the objective is to maximize the number of correctly mapped demands over the physical substrate given the scarcity of resources. In the online or dynamic approach, demands are not known in advance but they arrive dynamically at the network [18].

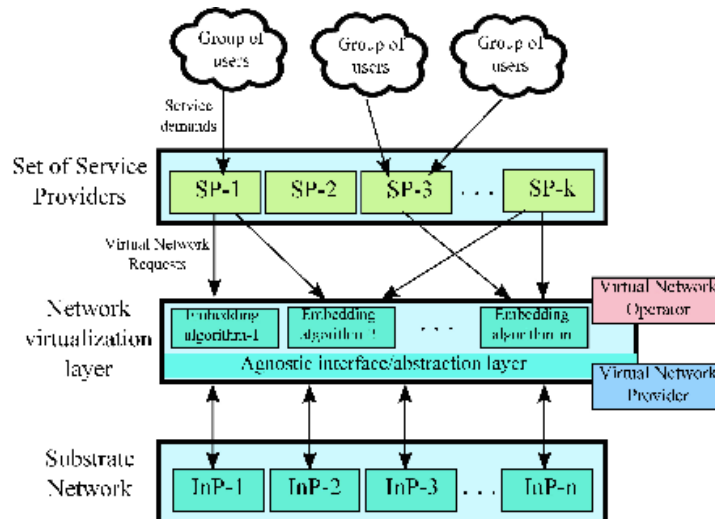
Formally the VNE can be described as following. A SN is represented by a directed graph, with nodes (N) and a links (L), $S = (N, L)$, then on top of the substrate network a set of virtual network request (VNR), $VNR^k = (VN^k, VL^k)$, where k are

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

the set of virtual request $k = 1,2,3 \dots$, each, with virtual nodes(VN) and virtual links(VL).

Then we define two functions, a node mapping and a link mapping, for the node mapping we define the following $X:VN^K \rightarrow N$. This function assigns virtual nodes to substrate nodes. Likewise the function $Y:VL^K \rightarrow 2^P$ where 2^P consists of all sets of directed paths in the SN. If Y is able to assign a virtual link to a set with more than one element, the VNE problem will allow a virtual link mapping with the use of multi-path routing [18].

For both the node mapping and link mapping function we have a limited resources of the substrate node, so is important that neither exceed the resources of either a node or a link in the SN. An optimal VNE is then the result of the node and link mapping functions that satisfies all of the above restrictions and, additionally, reaches a given optimization objective [18].



**Figure 6. Allocation of VN. source ("Virtual Network Embedding: A Survey,"
Communications Surveys & Tutorials, IEEE)**

II.5.2. Virtual Network previous proposed solutions

There are several solutions regarding Virtual Network mapping, two algorithm was proposed by Zhu and Ammar in [19], the first one was called “Basic VN assignment algorithm”, which idea was to do a minimization of the node and link stress sum, and

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

the second algorithm called “Subdividing Algorithm” that splits each VNR into a set of connected sub-VNs, each with a star topology.

The second approach was proposed by J. Lu, J. Turner [20] in which they only consider the bandwidth constraints and is evaluated in an offline scenario. In this approach, VN topologies are limited to be “backbone-star” topologies. These topologies are composed of “access nodes”; nodes representing the SN location at which traffic enters to the VN, and “backbone nodes” connecting the access nodes. An iterative algorithm changing the mapping of backbone nodes is performed to embed the different VNRs.

Other approved by Yu et al. in [21] looks for the maximization of long average revenue i.e. the weighted sum of VNR’s bandwidth and CPU demands. CPU in nodes and BW in links are the resources considered in this proposal

Chowdhury and Boutaba propose an approach for coordinated node and link mapping in [22]. Their main goal is to minimize VN embedding cost. VNE is performed in just one stage in Lischka and Karl [23] proposal. Node and link mapping are performed in the same stage by reducing the VNE to the well-known NP- complete Sub graph Isomorphism Detection (SID) problem.

Also is important to know the concept of hidden Hops. This concept was first introduced in previous work [24]. A substrate node is a hidden hop if it is part of a SN path used to map a virtual link (i.e. it is an intermediate node in the SN path). Besides the demands of some virtual nodes, each hidden hop will have a demand of resources because it has to be configured to forward packets passing through the SN path.

As it was pointed out before, the best solution for the virtual mapping problem is to divide in to stages, node mapping and link mapping. An optimal solution was proposed by Houidi et al. [25]. The solution is found by solving a MIP optimal problem formulation. Although optimal embedding can be performed in this way, as MIPs are complex problems (NP-complete), this solution is not scalable and it will take unaffordable running times for large networks. However, it can be used as a baseline for new VNE heuristics.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

All these previous proposals address the VNE problem in the single infrastructure provider (InP) scenario (Intra-domain VNE).

However, there are proposals to perform the VNE in an inter-provider scenario. For instance Chowdhury et al. in [26] propose a policy-based end-to-end VNE framework (PolyViNE) to embed VNs across multiple InPs. Inside any InP, any of the previous intra-domain proposals can be used, but across domains a distributed protocol that coordinated the participating InPs and ensures competitive pricing is introduced.

II.6. Path algebra framework

To accomplish a Virtual Network Mapping with a QoS control and also QoS parameters (availability, distance, flow, etc.), several network routing solutions can be proposed such as: exact and approximated routing algorithms, algorithms based on backward-forward heuristics, linear composition, hybrid, random, routing computation from the origin or destination, reservation and resource allocation protocols etc. [27,28,29,30].

The problems with these solutions is to their validity is often defined either by the size of the networks or by their topology, and their intuitive portability does not work for other applications. In other words, the use of a heuristic originally designed for distance metrics such as Dijkstra does not converge with another kind of metrics such as flow

Analyzing this problem under the perspective of protocols design, it has been necessary to conceive an heuristic or an algorithm to ensure the routing convergence for different types of QoS metrics or QoS metrics composition, in which this problem could be addressed from an integrated and generic manner by means of a mathematical framework which allows validating the proposed solutions independently from network topology or implementation details [7].

The framework use in which the user can define their own path searching policy that can be closer to the existing traffic profile of their networks is “Path-Algebra”. Such mathematical framework allows for systematically comparing different mono-constraint and multi-constraint routing heuristics concerning their convergence guarantees, best path convergence and loop avoidance, and validating a generic and

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

homogeneous solution that can be integrated into a single mathematical. In this case, QoS is a function of physical parameters associated to the network's performance [10].

Considering Network Virtualization mainly as a business strategy, the main metrics to be use would be cost and revenue. Nevertheless, "Paths algebra" may also use these metrics and, for example, associate them with the QoS. Such flexibility associated to its computing efficiency makes the "Paths Algebra" a suitable tool to perform the virtual link mapping stage inside the VNE [17].

Using paths algebra, the virtual link mapping can be performed in a multi-constraint basis, that is, virtual links mapped to substrate paths that are characterized by an unlimited number of constraints (or combination of constraints). In previous work this framework was tested using two metrics, the CPU of each node and the bandwidth of the links, our contribution for this thesis is the inclusion of non-linear metrics and analyze the behavior of the Path Algebra framework using several data and values such as cost, revenue, ratio mapped revenue, and also compare each of the metric according to the priority.

As we talk in previous chapter the virtual link stage of the VNE problem can be approached as a multiple multi-constraint routing problems. The idea of the Virtual Network embedding link mapping is to find the best route or routes in the substrate network for each of the virtual links of the Virtual network. The selection of the best route is base of the adoption of some optimization criteria.

In the next section the mathematical concept of the "Paths Algebra" is explained using some examples.

II.6.1. Paths Formulation

A network can be represented with the following function $G=(V,A)$, where V are the vertices or nodes and A as a set of arc or links. Considering the following network. The set of vertices are given by $V=(1,2,3,4,5)$ and the set of arc is $A=(a,b,c,d)$. So for instance the path of the source node 1 and destination 4, is (a,b,c) , in other word this path is represented as a successions of vertices or arcs $p_{a,d}$.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

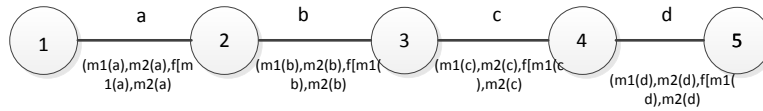


Figure 7. Representation of a Network

Each links in the network can be characterizes by the following formula $(m1(x), m2(x), f[m1(x), m2(x)])$, where $m1(x)$ and $m2(x)$ are the value of the metric of each link and $f[m1(x), m2(x)]$ as a combination of both metrics.

For this example the following matrix can represent the combination of all of the metric in the network.

$$C(p_{a,d}) = \begin{pmatrix} C_a \\ C_b \\ C_c \\ C_d \end{pmatrix} = \begin{pmatrix} m_1(a) & m_2(a) & f[m_1(a) + m_2(a)] \\ m_1(b) & m_2(b) & f[m_1(b) + m_2(b)] \\ m_1(c) & m_2(c) & f[m_1(c) + m_2(c)] \\ m_1(d) & m_2(d) & f[m_1(d) + m_2(d)] \end{pmatrix}$$

The previous formula represented, that for each metric in a path, “Paths Algebra” would have a set of combination function $C(p_{a,d})$.

Another characterization is also use in “Paths Algebra”. The synthesis $S[]$ are a set of operation applied on the values of the links metric combination. By using this formula a characterization of the path based on the constraint is done. The synthesis use in this work are the following: {add(), mult(), max(), min()}.

If the routing has only one constraint, only one value would be obtained. On the other hand is the routing is multi-constraint then, with k-constraint, k values are obtained. So, the synthesis can be formulated like the following, $S[] = [S_1 S_2 S_3]$, where the first letter corresponds to resulting value of the synthesis applied to the whole path; the second letter corresponds to resulting value of the synthesis applied to the sub path obtained by dropping out the last arc; the last letter corresponds to the resulting value of the synthesis applied to the sub path made of only the first arc

II.6.2. Path Ordering

Consider the network represented in figure 8, there are two different paths between the nodes 1, and 4. The first path is the one with the nodes (1,2,3,4) and the

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

links (a,b,c). The second path is with the nodes (1,5,4) and the links (d,e). Each link in the path is characterized by the triple formulation explained before $m_1(x), m_2(x), f[m_1(x), m_2(x)]$, where $f[m_1(x), m_2(x)] = m_1(x) * m_2(x)$. And the synthesis used will be the following: $S[] = [\min() \max() \text{add}()]^t$.

Once the synthesis is calculating, the ordering of the path is done. To analyze which path is worse or less optimized the multidimensional lexical ordering is used. This means if $S[Path1] \preceq S[Path2]$ then path 1 will be less optimized than path 2. For this case, $ML \preceq = \{\geq, \leq, \geq\}$. Table 1, summarizes the result of using this ordering relation and the synthesis

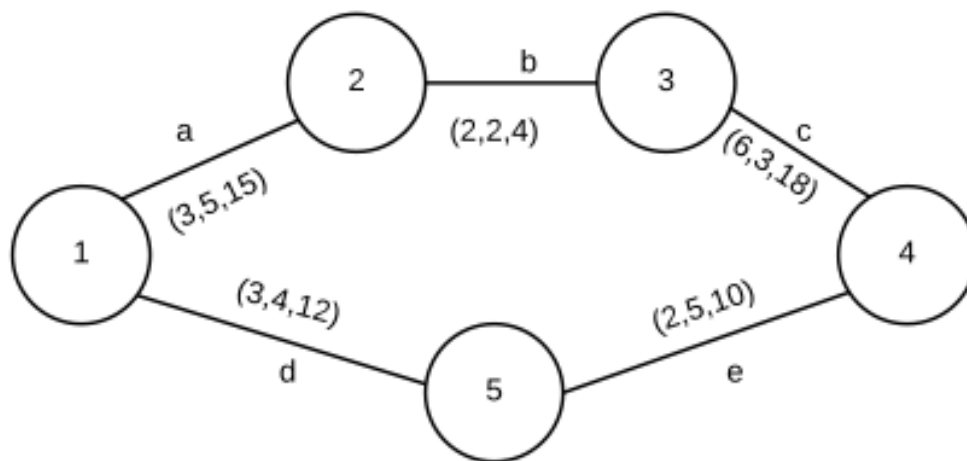


Figure 8. Path ordering example

Path	S1 Min	S2 Max	S3 Add
(1,2,3,4)	6,2,3	3,2,5	37,19,15
(1,5,4)	2,3	5,4	22,12

Table 1. Path Ordering

II.6.3. Node and links Metric

The resources that need to be consumed by the virtual network and the optimal allocation in the VNE problem are attributed with metric. On one hand, substrate resources have individual capacities and qualities. On the other hand, virtual resources

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

each have their respective requirements. For example, a substrate node can provide a certain computing capacity relating to the CPU available to it. Per contra, a virtual node will require a certain computing capacity in order to properly compute routing information. These metric are of paramount importance in order to achieve a valid embedding. Here, the different kinds of metric are discussed [13].

As a first step, one can distinguish between node and link metric. Node metric are attributes that refer to nodes, like computing power. Link metrics are attributes that refer to links, like bandwidth. However, there is a problem arising with such a strict model. When a virtual link is mapped to a path in the substrate network, the computing power of substrate nodes on the path may have an impact on the bandwidth of the virtual link [26]. As such, virtual nodes are somewhat easier to map than virtual links, since the latter consist of a combination of substrate node and links, whereas the former takes only substrate nodes into account. In the next section there will be an explanation regarding the nonlinear and linear metrics use in this work.

II.6.3.1 Node Metric

For the nodes, two metrics will be considered:

- The CPU consumption
- The Energy Consumption

The first one has already been discussed in previous work. Nevertheless in this section there will be an explanation on how to deal with the CPU metric in the “Path Algebra”. In the case of the energy, there will be small description of the metric and in the following chapters there will be a wide explanation on how to include this new metric in the path algebra framework as one of the new constraint for the VNE problem.

II.6.3.1.1. CPU consumption

The first node metric or resources of the virtual and substrate network is the CPU consumption of each nodes, the idea is to maximize the spare CPU after the mapping is done. So an optimal allocation of this resource needs to be made. In order

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

to deal with the metrics associated to the digraph's nodes it is necessary to perform the following transformation. Consider, for example, the part of a digraph shown in Fig. 9. In this figure, the weights are associated both to the nodes and links. Nodes A and B spare CPU capacity are given by CPU (A) and CPU (B), respectively.

The spare bandwidth of the link connecting nodes A and B is the link weight $BW(A-B)$. When a packet traverses the network going from node A to node B it will be processed by both CPUs and flow through the link connecting both nodes. This being so, we can artificially assign all metrics to the arc connecting both nodes as depicted in Fig. 3.b.

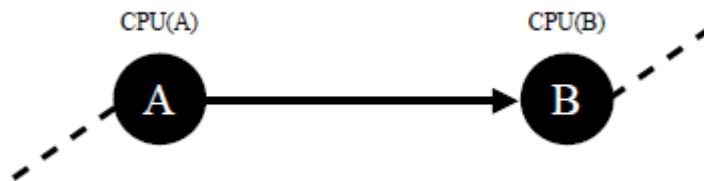


Figure 9. Two nodes

As one possible objective could be to maximize the spare CPU capacity, the most restricting condition is given by $\min [CPU (A), CPU (B)]$. So, we can use the metrics combination function provided by the paths algebra and reduce the metrics to $(\min [CPU(A), CPU(B)])$ as shown in Figure 10. [3]

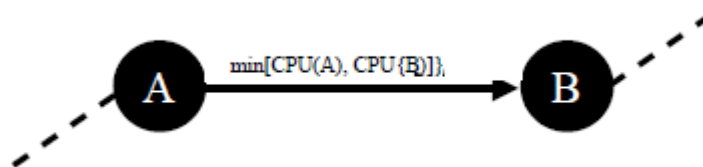


Figure 10. New restricting condition

It is important to realize that independently of a specific objective, any VNE problem has to assign the VNRs to SN nodes and links, under the constraint of available physical resources. The possibility of the paths algebra to deal with both node and CPU metrics as well makes it a complete framework to solve the VNE problem. It can be even envisaged the development of new algorithms in which nodes and link

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

assignments can be done in a coordinated way instead of in separate stages as is presently the case [3].

II.6.3.1.2. Energy

For the case of the energy is important to point out that for the sake of simplicity, we consider that network resources are homogeneous with regard to their energy consumption, this can be the case of substrate networks reduced to just one ISP segment (access, transport or core) where network equipment shares similar characteristics. Our model is proposed based on previous cost-based VNE formulations.

In order to create the energy metric, the next model will be use $ENERGY = A + B * BW$, BW will be the maximum inflow and outflow bandwidth of every node in the network and “A” and “B” would be chose between 0 and 1. In the next chapter there will be an explanation of how to include this metric into the “Path Algebra” framework. Is important to point out that this metric will only be generated for the substrate network unlike the CPU consumption in which there is a resource and demands

II.6.3.2. Link Metrics

II.6.3.2.1 Bandwidth

For this work there several link metric that are going to be use, the first one will be the Bandwidth, the idea is to use the same transformation that for the case of the CPU consumption. We have the next link (figure 11) constituted of two nodes with a spare BW; this means that the arc weight will be the $BW(A-B)$.

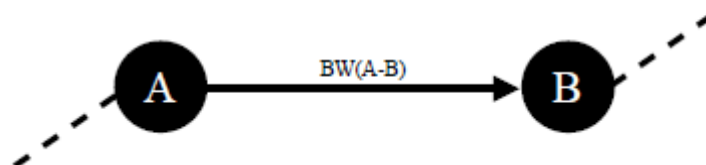


Figure 11. Bandwidth transformation

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Like the CPU consumption in the previous section, the Bandwidth will be a resource of the substrate network, but also a demand of the VN.

II.6.3.2.2. Throughput

This will be a new metric that would be included into the “Path Algebra” framework.

Throughput is the rates of successful packet that can be deliver between a pair of nodes. In our case we are using percentage for this metric for all the links in the substrate network. This constraint will measure in percentage between 90 and 95%. Like the energy for the metric node, this parameters will only be generated for the substrate node, because there will be no demand of the Virtual Networks

II.6.3.2.3. Packet Loss Ratio

This is also part of our contribution for this work, like throughput, this new metric is going to be included in the “Path Algebra” framework.

The Packet Loss Rate is the percentage of packets that were not successfully delivers between a pair of nodes. This metric will also be measured in percentage between 0 and 5 %. And also like the energy and throughput, this parameter will only be generated for the substrate node.

II.6.4. Evaluation Metric

For this master thesis three parameters are chosen and can be used to compare the results obtained by the different metrics priority:

- **Cost:** Refers to the amount of substrate resources that were used for embedding virtual networks [13]. Cost is usually determined by summing up all CPU and bandwidth resources of the substrate network that have been reserved for VNR's [19]. Cost is directly related to the length of substrate paths: This means the longer the path (more hops) the cost will be higher.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

- **Revenue:** Revenue refers to the sum of virtual resources that were actually requested by the virtual entities. This value is usually computed by applying the same scheme that was used to determine Cost [13].
- **Cost/Revenue:** To have a efficient comparison of the different metric priority, and the embedding result, both evaluation metric need to be use because depending on the random topologies Cost may vary By dividing Cost by Revenue, varying Cost values are balanced [13]. The higher the value, the more resources were needed to embed the VNs.
- **Acceptance ratio:** The acceptance ratio metric measures the number of virtual network requests that could be completely embedded by the embedding algorithm, divided by the total number of virtual network [13] demands.

II.6.5. Path and Virtual Network ordering

II.6.5.1. Path Ordering

Now that the metric use in the “Path Algebra” framework are explain is important to describe how the mapping and ordering is done using this mathematical tool. The first step taken by the “Paths Algebra” algorithm is to find all paths connecting all pairs of nodes in the SN. It may seem that this a too expensive procedure but the idea is that this procedure be done only once and can be performed offline (before the embedding process starts).

II.6.5.2. Virtual Network Ordering

The optimization of the Virtual Mapping must be done taking into account the number of virtual network request instead of the amount of virtual link mapped. Furthermore is important to realize that the virtual request consume physical resources, like the ones that we talk in previous sections. Such physical resources are finite and represent the limit of VN assignments. The way they are allocated affects the number

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

of VN requests that can be satisfied. There are different ways to allocate the virtual requests [3]:

- Allocate them in the order they arrive as if it were a FIFO (First In First Out) procedure;
- Allocate them in a non-decreasing, or non-increasing, BW order;
- Allocate them in a non-decreasing, or non-increasing, CPU order;
- Allocate them in a non-decreasing, or non-increasing, (BW+CPU) order.

Suppose, for example, a substrate network that has a total BW of 10 and there are virtual requests of $BW = 1, 2, 3, 4, 5$. In principle, if we use the non-decreasing order, we will be able to allocate 1, 2, 3 and 4. On the contrary, in a non-increasing order we will be able to allocate only 5, 4 and 1. So, the order in which the VN requests are processed may affect the final result [3]. A priori it is not possible to say what is the best policy. If, for example, the number of satisfied VNRs is the optimization criterion, then the non-decreasing order allocation is better. However, if the total amount of assigned BW represents revenue, both policies are equivalent.

For the purpose of our work we are using the non-increasing (BW+CPU). This will decrease the chance to have to perform backtrack. i.e., to re-assign a previous virtual link in order to accommodate another one [3].

II.6.6. Node and Link Mapping

Paths algebra provides a great flexibility to the mapping of transport network requirements. Any set of metrics and combination of metrics can be employed, different optimization criteria can be examined simultaneously and the quality of different solutions can be evaluated.

Now this individual metrics that was talked before can be combines into a cost function. There is no restriction neither about how the metrics are combined nor about the number of metrics plus cost functions to be used in a specific VNE problem [3].

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

The values of all the metric along a certain path are evaluated using the Synthesis operation. Four syntheses may be used: i) minimization - $\min()$; ii) maximization - $\max()$; iii) addition - $\text{add}()$ and iv) multiplication - $\text{mult}()$. The synthesis to be used is metric dependent. For example, to evaluate the path delay we have to add the links delay, while to evaluate the path spare bandwidth capacity we need to find the minimum of the links spare bandwidth [3]. . Is also important to understand the concept of hidden hops [31], which makes reference to the intermediate nodes of a directed path in the SN that is mapping a specific virtual link of a VNR. We claim that a hidden hop entails a resource demand because it has to perform packet forwarding of the traffic that will pass through this virtual link [3].

Now, how this mapping is done in path algebra? First any assignment of a Virtual Request will consume SN's resources and also the packet will be affected by the state of the paths or links, for instance if the link have a high rate of packet loss or throughput. The SN's links weigh are modify by the values they would have if the SN's resources were consumed according to the corresponding VN request. Using the "Paths Algebra" the paths are ordered taking into account all specified metrics. The virtual link is assigned to the best path and the SN resources are updated [3]. This procedure is repeated until all virtual links have been assigned to the SN. In case a virtual assignment is not found because there aren't enough resources to accommodate the request or if there are no available resources to map a virtual link it is necessary to define if the VN request will be simply dropped out or if it will be accommodated with less resources than originally demanded. It is important to point out that the decision is related to the adopted business model and it is not affected by the paths algebra.

II.6.7. Example using default metric.

In this example we have two VNR each of them has the 3 nodes and 2 links (Figure 12). For SN, weights (in bold) over links indicate links' bandwidth and the nodes' weight (in parenthesis) are the nodes' capacity. For the VNs, the links weights indicate the requested bandwidth, nodes' weight are the total CPU capacity requested for terminal nodes, and the links weights shown between parentheses are the total CPU capacity requested for hidden hops. Total CPU requested is to be understood as the

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

capacity demanded by the whole VN request, independently of the number of times the node is used (as terminal or hidden) by the SN assignment [3].

As it was stated in previous section in order to do an optimal mapping a prioritization of the metric need to be made, for this example we are using three metrics (HOPS, BW, CPU) and three different synthesis (add,min,min). It is important to note that there are different types of demands in the VN requests. Some resources lessen when virtual link requests are being assigned, and some others do not [3].

Bandwidth and CPU demands are subtracted from the SN resources each time a virtual link is mapped.

The first step is to implement the algorithm called the Loop Avoidance by the Destination (LADN) algorithm. Using this algorithm the loop created when each nodes decides about the next hop in a hop-by-hop routing. For instance when a node x to reach a destination d decides that the packet has to be sent to node y and node y to send the packet to the same destination d decides that the best next hop is node x . The packet will travel forever between x and y . The LADN algorithm takes care of this situation and enforces the coherence property that avoids loops [3].

The second step to solve the VNE problem is to enumerate all paths between pair of nodes of the SN. Such enumeration is completely independent of the VN requests. The pair of nodes of the SN is then used as terminal nodes by the VN requests. This sequence is also followed by the LADN algorithm. The obtained paths are:

- For the virtual path (C,D) we have the following paths : (C, D), (C, A, D), (C, B, A, D)
- For the virtual path (A,C) we have the following paths : (A, C), (A, B, C), (A, D, C);
- For the virtual path (A,E) we have the following paths : (A, C, E), (A, B, C, E), (A, D, C, E)
- For the virtual path (B,E) we have the following paths: (B, C, E), (B, A, C, E), (B, A, D, C, E)

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

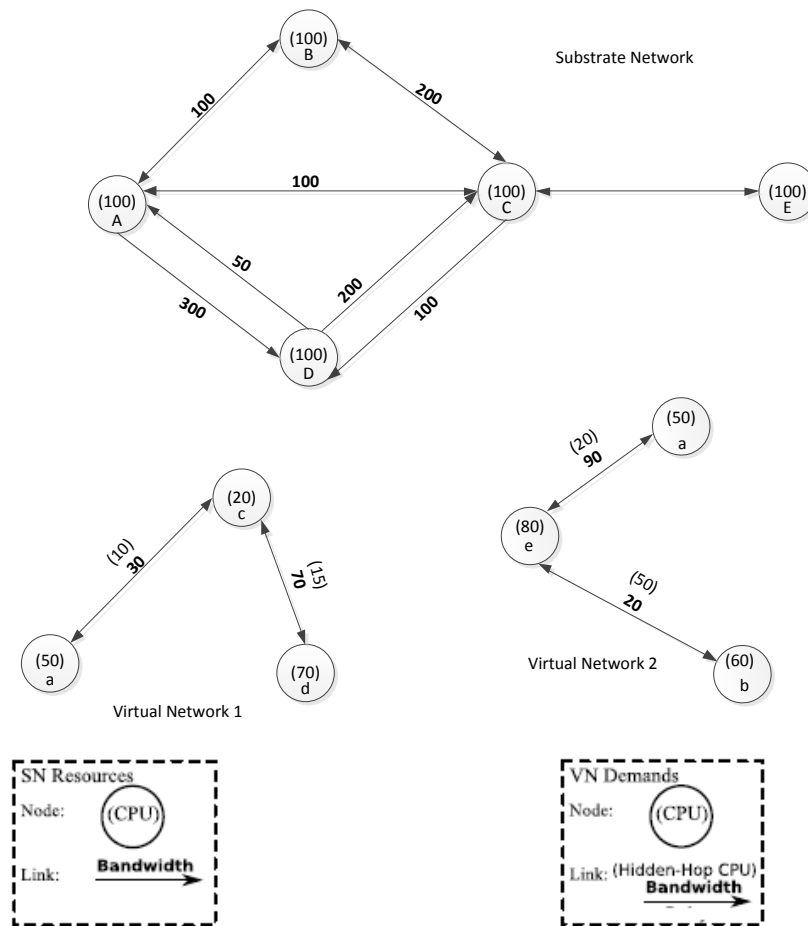


Figure 12. Example of two VNR and one SN.

Next, the VN requests have been sorted in an increasing order of total demand, and within a VN request the virtual links to be assigned to the SN have been sorted in a decreasing order. That means that we are attending the Virtual Network with the high resource, in this case is the Virtual Network 1.

For this examples we are using 4 parameters to choose the best path, first is the number of hops in the path, second will be the minimum spare CPU and the second the BW. After that we are calculating the Cost and the revenue and compare different solution

For the cost the following formula is use:

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

$$COST = CPU(A) + CPU(B) + (1 + K)BW$$

The nodes “A” and “B” are of the substrate network, and K will be the number of hidden nodes, so the cost will be defined as the sum of the BW and CPU spent, by the SN, to map a VNR. [7]

In the case of the revenue the next formula will be use: $COST = CPU(A) + CPU(B) + BW$. In this case A and B are the pair of nodes of the virtual network

The first link that the system is mapping is the virtual link (C, D), in the following table are the result of the mapping.

Path(c-->d)	Hops	Bandwidth	CPU	COST	C/R
(C, D)	1	30	30	300	1.25
(C, A, D)	2	30	30	600	2.5
(C, B, A, D)	3	30	30	800	3.33333333

Table 2. Mapping (c,d)

As it shows in the table, the BW and CPU are the same, but because the highest priority is the hops, the first path is selected. After mapping this link the resources of the substrate node are updates as it shows in the next figure (figure 13).

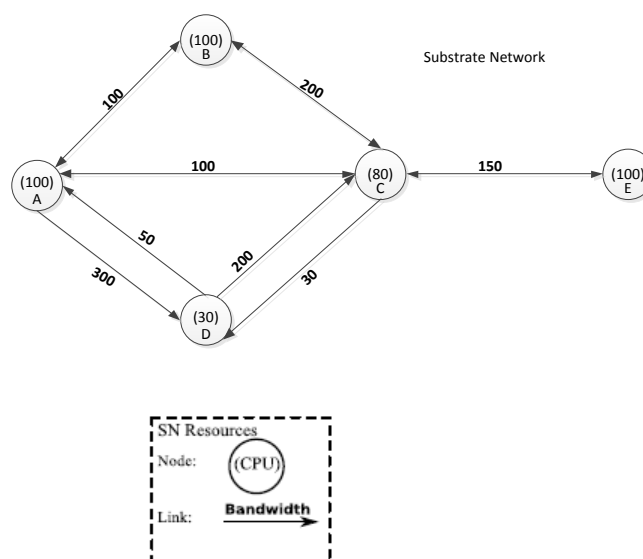


Figure 13. SN after first mapping

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Next, the second link (a,c) is mapped into the substrate network, as we can see in table 3. In this case the selected path is A, C.

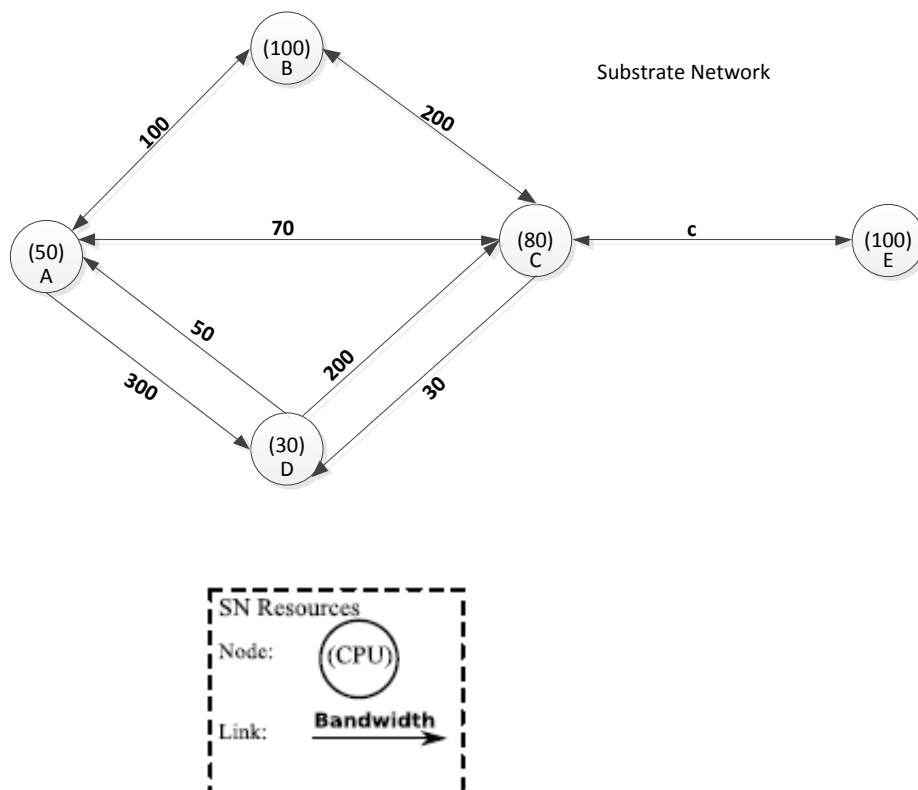
Path(a->c)	Hops	Bandwidth	CPU	COST	C/R
(A, C)	1	70	30	280	1.16666667
(A, B, C)	2	70	30	480	2
(A, D, C)	2	130	30	680	2.83333333

Table 3. Mapping path (a,c)

After that the resources of the SN are updated again (figure 14), and the second Virtual Network request is attended.

For the next two links, (a,e) and (b,e) the same procedure is done, as we can see in table 4, and table 5. For the first case the selected path was (A, B, C, E), because even though that both paths have the same number of hop the third path has a lowest spare minimum BW.

And finally for the last case only one path could be chosen because the remaining one could not be attended because the demanded BW and CPU we highest that the resources of the substrate network.



Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Figure 14. SN after first VN

Path(a-->e)	Hops	Bandwidth	CPU	COST	C/R
(A,C,E)	2	-20	0	0	0
(A, B, C,E)	3	10	0	600	2
(A, D, C,E)	3	60	0	700	2.333333333

Table 4

Path(b-->e)	Hops	Bandwidth	CPU	COST	C/R
(B, C,E)	2	40	20	270	0.9
(B, A, C,E)	3	-10	-50	0	0
(B, A, D, C,E)	4	-10	-50	0	0

Table 5

Chapter III

For this Master Thesis two programs are used. The first one is ALEVIN, which will generate the different substrate network, virtual networks, resources and demands and it will handle the node mapping stage of VNE. The second module is MATLAB, which will be in charge of ordering the different VNR based on priority and the link mapping stage of the VNR. In this chapter the configuration and the installation of the environment is stated and the files and data transmitted between both programs are explained.

Alevin Framework

The simulation environment used for this Master Thesis is the Algorithms for Embedding of Virtual Networks (ALEVIN) [33, 34]. ALEVIN is a modular framework, which can handle several types of virtual embedding algorithms and arbitrary parameters for resources and demands. It also provides the ability to illustrate the deployment of resources in the SN and demands in an arbitrary number of VNs as well as the mapping of demands on resources calculated by a VNE algorithm. Moreover, ALEVIN can be used to create VNE scenarios as well as import and export them using an XML-based exchange format [31]. As already mentioned, the algorithm used to implement paths algebra, called LADN [27], was developed as a simulation tool for testing multi-constraint hop-by-hop routing algorithms in which the optimization strategy is user-defined [3].

ALEVIN is completely modular regarding the addition of new parameters to the VNE model. Using the visitor design pattern in a sophisticated way, we are able to avoid any casts to concrete demand/resource classes. Thus, the number of parameters is not performance-relevant and a convenient implementation of arbitrary parameters is possible. To increase ALEVIN's modularity and to make it a flexible and extensible platform to compare existing and upcoming algorithms, the implementation of algorithms is kept independent of the resource/demand implementation [32].

ALEVIN also is our basis for the evaluation and comparison of VNE algorithms. Therefore, ALEVIN provides a simple interface to add evaluation metrics, which are independent of the implemented parameters and algorithms. This further

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

emphasizes the modularity of ALEVIN. Several evaluation metrics can be applied and be combined in an optimization strategy for a whole VNE scenario. For instance, an optimization objective for a VNE scenario might be to minimize the packet loss rate, maximize the residual bandwidth capacity, minimize the total cost for the substrate, or minimize the overall energy consumption. Due to the ability to combine arbitrary evaluation metrics and optimization objectives in combination with arbitrary parameters and a simple interface for VNE algorithms, ALEVIN is a powerful framework for the comparison and analysis of VNE algorithms regarding different evaluation metrics and optimization objectives [3].

III.1. Installation of the simulating environment

In order to use the Linux virtual Machine in which the ALEVIN software environment is running is necessary to have the VMWare Fusion, software, which can be downloads from the following website

<http://www.vmware.com/products/fusion/overview.htm>

In case the operating system is Windows, the VmWare player can be downloaded from this website. <http://www.vmware.com/products/player/>

III.1.1. For MacOS

Once the VmWare player is installed a decrease of the number of CPU cores is done. To do that it is necessary to go to setting and change the CPU cores to “2” and decrease the memory to 1024 mb. This will affect the performance of the virtual machine.

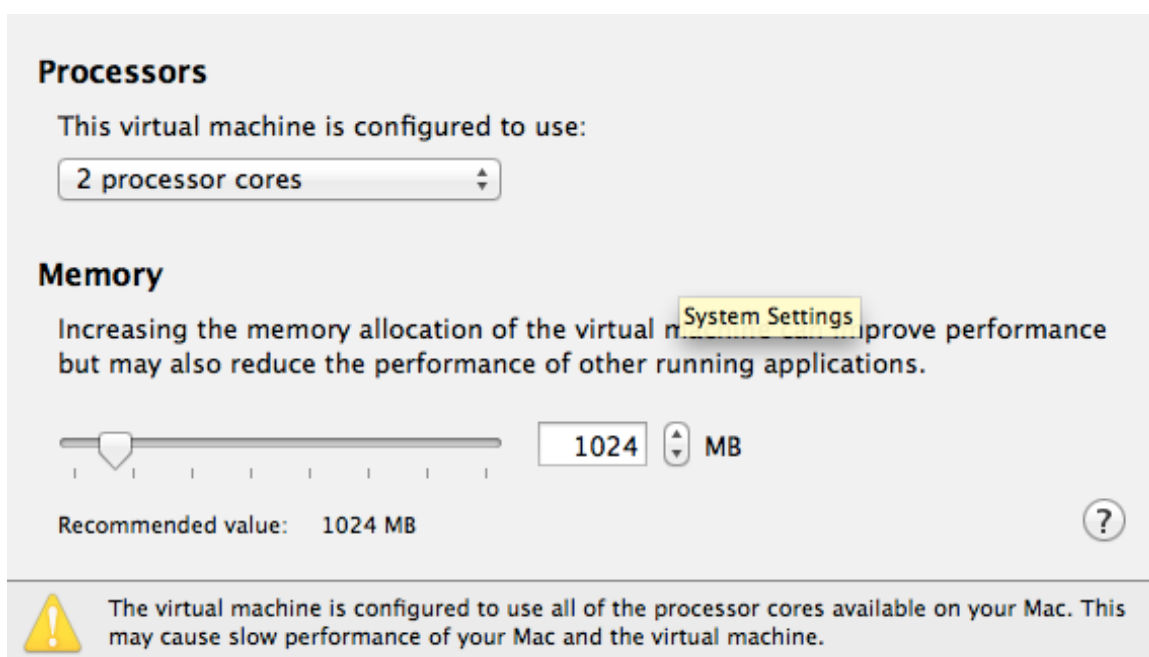
III.1.2. For Windows OS

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service



Figure 15. Setting of the virtual machine

Once the VmWare player is installed, a change in setting of the Linux Virtual machine need to be made like in the previous case. First the decrease of the number of CPU cores is made. To do that first the virtual machine that is going to be modify is selected (Figure 15)



Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Figure 16. Processor and memory

After selecting the virtual machine, go to “Edit”, to modify the virtual machine setting, and decrease the number of CPU core to 2. (Figure 16)

And finally also is necessary to decrease the amount of memory use in the virtual machine. Is important to point out that the decrease of memory and CPU would affect the performance of the virtual machine, so the simulation must be done using small networks. (Figure 17)

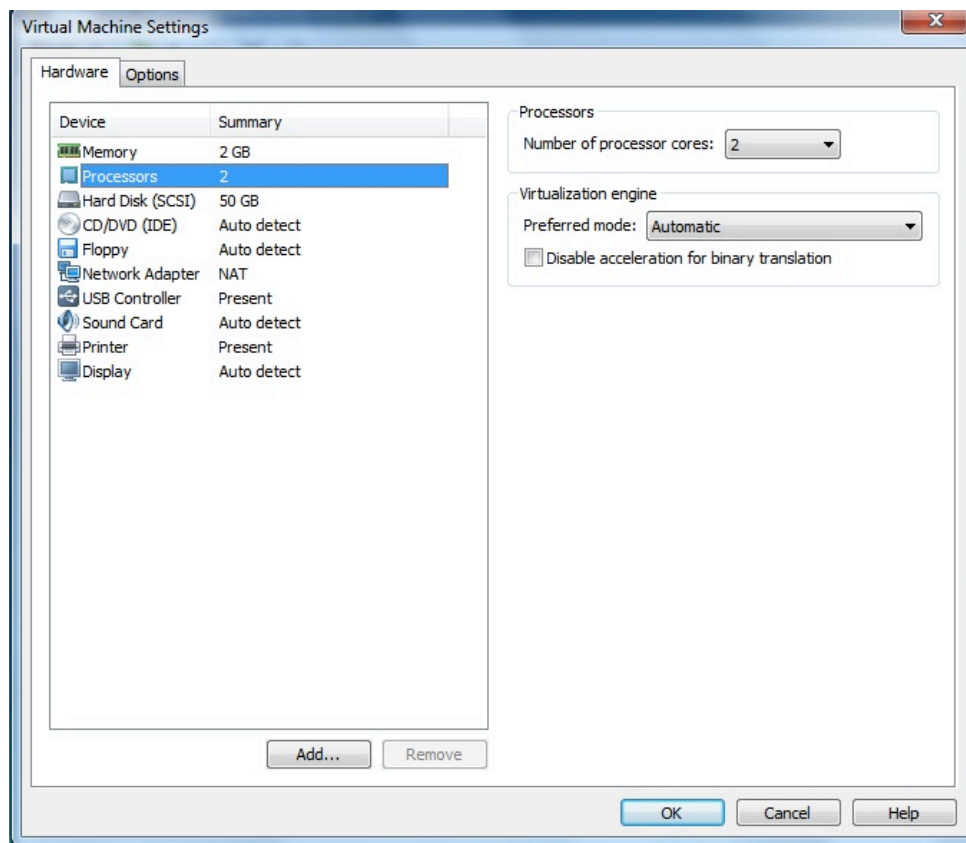


Figure 17. Setting windows OS

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

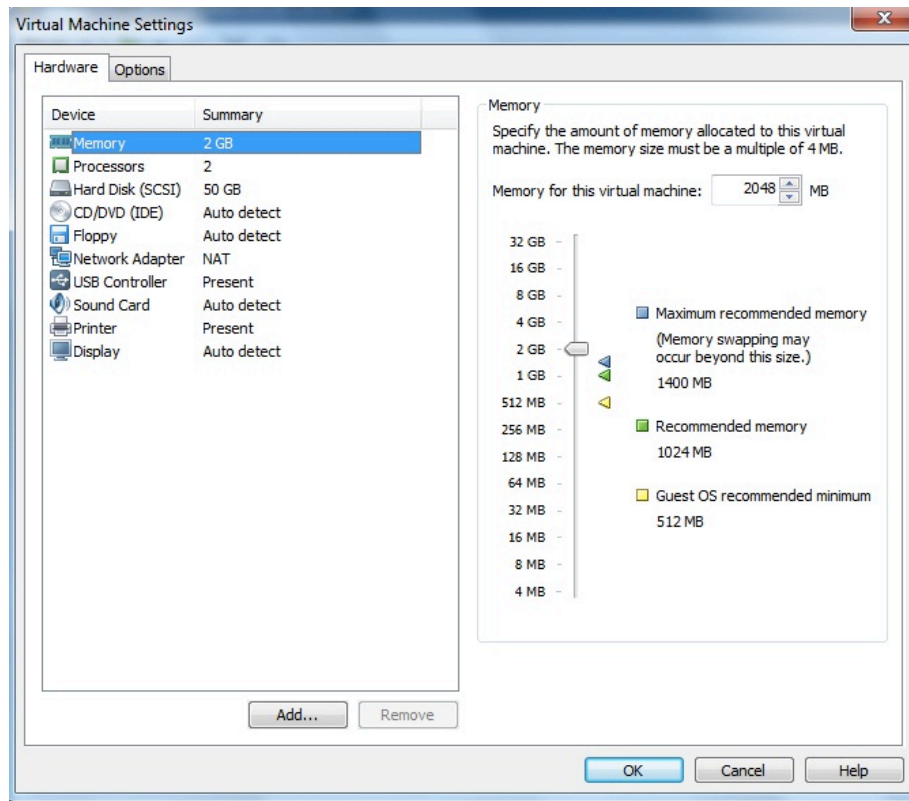
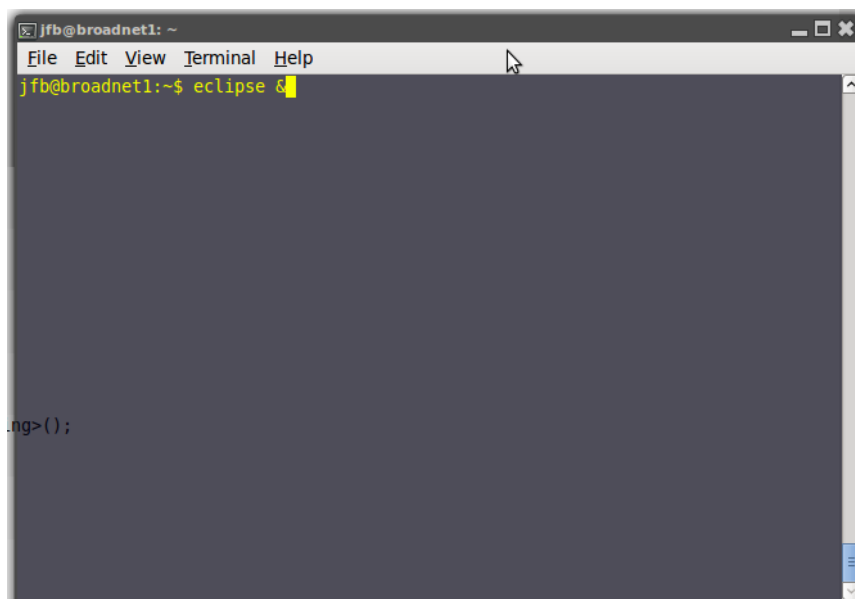


Figure 18. Processor and memory Windows OS

III.2. Running the simulation

- OPEN ECLIPSE(figure 19):
- The project is situated at /home/jfb/tempWorkSpace



Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Figure 19. Eclipse

Go to the pathsAlgebra package (tests.scenarios.pathsAlgebra).

There are there two classes of paths algebra algorithms called “PathsAlgebraAR.java” (that means that the node mapping algorithm is the greedy available resources) and the “PathsAlgebraCoordinated” (meaning that the node mapping algorithm is the coordinated node and link mapping algorithm). Each of these algorithms has the following parameters that can be tuned (Figure 20,21):

- **hiddenHopsFactor** = This value will determine the percentage of the virtual link demand (BW) that will be considered as virtual node demand (CPU) for the hidden hops. *policy* = value or preordering of the network (the value of the virtualStart.dat file)
- **pathOrder** = value of the metric (the value of the virtualStart.dat file)

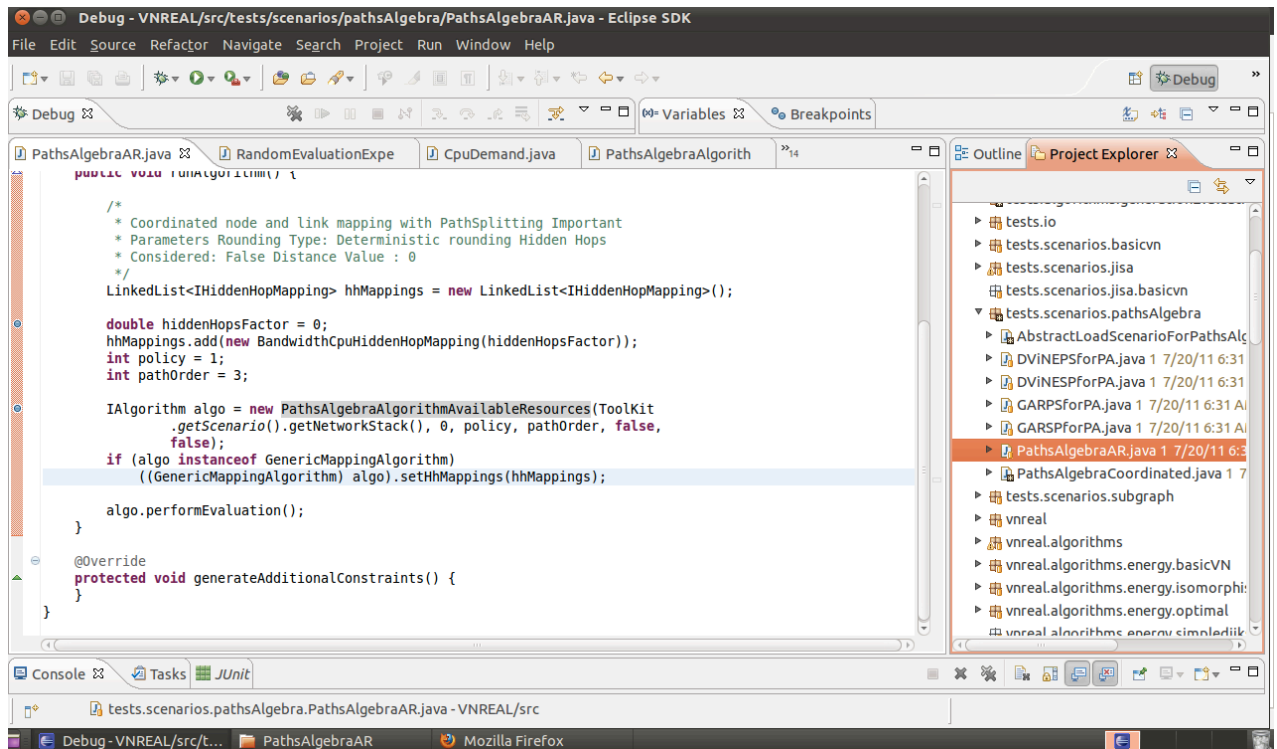


Figure 20. “PathsAlgebraAR” java Class

The scenario parameters are in the “AbstractLoadScenarioForPathsAlgebra.java” class (figure 20). Now because the virtual machine is running on 2 CPU’s cores a decrease of the different parameters needs to be made. The following parameters are

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

the ideal one.

- SubstrateNetworkSize = 20
- VirtualNetworkSize = 10
- NumberofVirtualNetworks = 10
- DifferentLoads: 0.2, 0.3, 0.4, 0.5, 0.6, 0.7
- MaxCPUResource = 100
- MaxBWResource = 100
- Probability of link between any node pair = 0.5

```
@Override
public void runAlgorithm() {

    /*
     * Coordinated node and link mapping with PathSplitting Important
     * Parameters Rounding Type: Deterministic rounding Hidden Hops
     * Considered: False Distance Value : 0
     */
    LinkedList<IHiddenHopMapping> hhMappings = new LinkedList<IHiddenHopMapping>();

    double hiddenHopsFactor = 0;
    hhMappings.add(new BandwidthCpuHiddenHopMapping(hiddenHopsFactor));
    int policy = 1;
    int pathOrder = 3;

    IAlgorithm algo = new PathsAlgebraAlgorithmAvailableResources(Toolkit
        .getScenario().getNetworkStack(), 0, policy, pathOrder, false,
        false);
    if (algo instanceof GenericMappingAlgorithm)
        ((GenericMappingAlgorithm) algo).setHhMappings(hhMappings);
}
```

Figure 21. Scenario Parameters

```
private final static int numScenarios = 2;

/** # of runs with different random seeds if algorithm is non-deterministic */
/* MM protected int numRunsPerScenario = 10; */
protected int numRunsPerScenario = 1;

/* MM private static final int[] numSNodesArray = { 20 }; */
private static final int[] numSNodesArray = { 5 };

/* MM private static final int[] numVNetsArray = { 10 }; */
private static final int[] numVNetsArray = { 2 };

/* MM protected static final int[] numVNodesPerVNetArray = { 10 }; */
protected static final int[] numVNodesPerVNetArray = { 3 };

// private static final double[] rhoArray = { 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 };
// private static final double[] rhoArray = { 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 };

private static final double[] rhoArray = { 0.4 };

private static final double[] maxCPUresArray = { 100.0 };
private static final double[] maxBWresArray = { 100.0 };

private static final double[] alphaArray = { 0.5 };
private static final double[] betaArray = { 0.5 };
```


Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Figure 22. Scenario Parameters

To execute the tests just right click in the name of the class (PathsAlgebraAR.java or PathsAlgebraCoordinated) and run as → junitTest.

Now the simulation will generate two folders in the home/directory files, but it will also generate some files in order to be used in the MATLAB, scripts, those files cannot be seen but it they will explain in this section. The creation of the MATLAB files is found in the “MatlabFilesCreator” class.

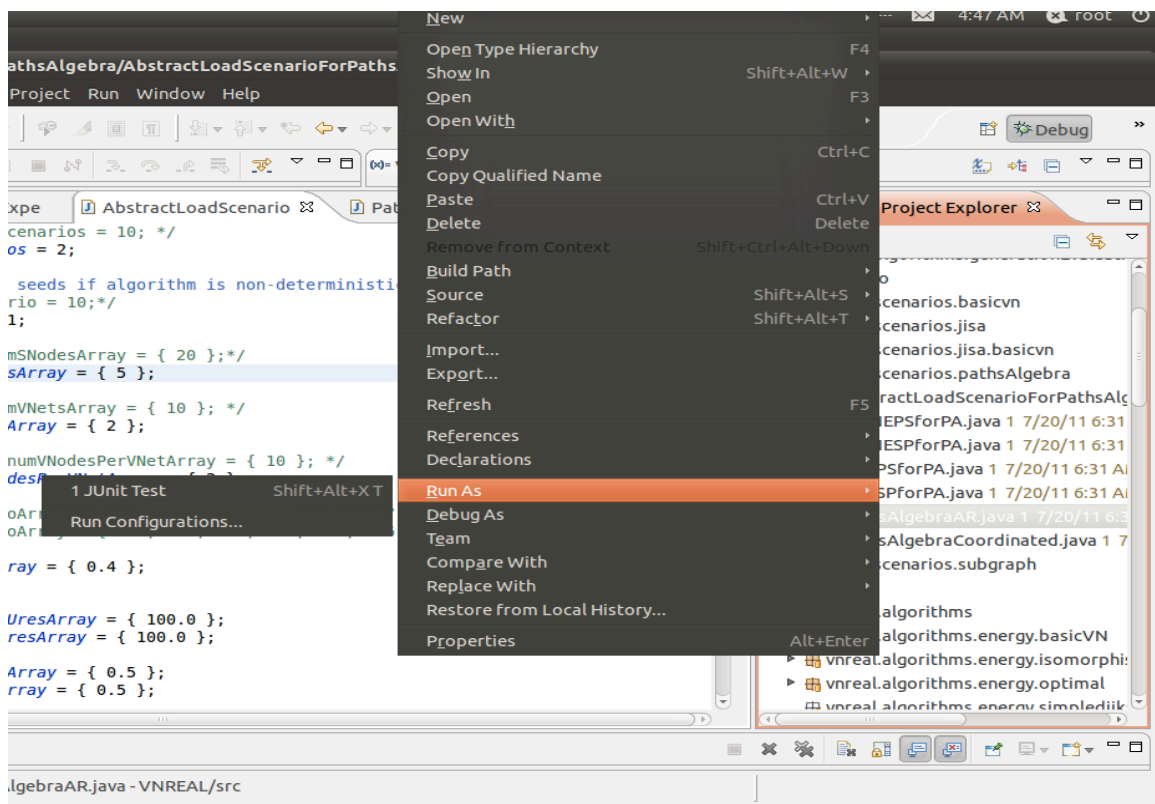


Figure 23. Running the Test

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service



```
PathsAlgebraAR.java PathsAlgebraLinkMapp *AbstractLoadScenari *MatlabFilesCreator. »14
*/
public class MatlabFilesCreator {
    // PRIVATE //
    private static FileWriter fstream, fstream1, fstream2, fstream3, fstream4;
    private static final String TOPOLOGY_FILE_NAME = "substrateNetwork.dat";
    private static final String BANDWIDTH_RESOURCE_FILE_NAME = "substrateMetric1.dat";
    private static final String CPU_RESOURCE_FILE_NAME = "substrateMetric2.dat";
    private static final String PATHS_ORDER_CRITERIA_FILE_NAME = "virtualStart.dat";
    private static final String VIRTUAL_NETWORK_REQUEST = "virtualRequest_";
    private static final String NODE_MAPPING = "virtual_to_Real_";
    private static final String HIDDEN_HOPS_DEMAND = "virtualHidden_";
    private static final String BANDWIDTH_DEMAND_FILE_NAME = "virtualMetric1_";
    private static final String CPU_DEMAND_FILE_NAME = "virtualMetric2_";
    private static final String FINISHING_FILE_NAME = "virtualFinish.dat";
    private static final String DELETESCRIPPT = "script3.sh";
    private static final String VIRTUAL_NODE_MAPPING_SUCCESS = "virtualnodemapping_";

    private static final String MATLAB_PATH = "/home/jfb/PathsAlgebra/Files/";

    // Empty constructor
    public MatlabFilesCreator() {

    }%
}
```

Figure 24. Matlab File creator

III.3. MATLAB files

In this section there is an explanation of the different files uses and generated by the framework ALEVIN, before and after the simulation and also how this files are interchange between the Java and the MATLAB software.

III.3.1. Files shared between MATLAB and Java

The following are the different files that the framework ALEVIN will share between the MATLAB and Java software

- **substrateNetwork.dat**: In this file the topology of the substrate network is shown. This file is a matrix with different columns and rows. In the following table there is an example of this files

In this example there is a substrate network with 10 nodes. For each pair column and row, if the value is "1", then there is a link between those two nodes. If the value is zero there is no connection.

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	1	1	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0
3	0	1	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0	0	1
5	0	1	1	1	0	1	1	1	0	0
6	0	1	0	1	0	0	0	0	1	1
7	1	0	0	0	0	1	0	0	0	0
8	0	0	0	0	1	0	1	0	1	0
9	0	0	0	0	0	0	1	0	0	0
10	0	1	1	0	1	0	0	1	0	0

Table 6. Substrate Network file

- **substrateMetric1.dat:** in this file the bandwidth values of the substrate network are shown. Like the previous case this file is a matrix composed of different values of the bandwidth between the pair of nodes of the substrate network (Table 7).

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	87.852	85.495	13.146	0	0	0	0
2	93.403	0	52.235	0	0	0	0	0	0	0
3	0	53.586	0	45.317	27.407	82.961	0	0	61.757	0
4	0	0	0	0	82.153	0	0	0	0	46.616
5	0	2.054	3.34	83.277	0	74.114	23.153	30.246	0	0
6	0	95.856	0	26.892	0	0	0	0	61.737	29.375
7	69.941	0	0	0	0	23.187	0	0	0	0
8	0	0	0	0	66.705	0	99.333	0	67.028	0
9	0	0	0	0	0	0	8.531	0	0	0
10	0	7.101	83.425	0	60.513	0	0	95.211	0	0

Table 7. Bandwidth metric file

- **substrateMetric2.dat:** Lastly in this file the CPU values of each nodes of the SN are shown. For this case the file will be a vector composed of several columns that represent each node of the SN (Table 8).

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

8.125	90.359	14.688	63.815	78.397	5.543	39.175	96.97	9.249	15
-------	--------	--------	--------	--------	-------	--------	-------	-------	----

Table 8. CPU metric file

- **virtualStart.dat:** Is a vector with three columns. The first column explain the different politics of attendant (Most Consuming First (MCF), Least Consuming First (LCF) etc). In the second columns the metric priority. And finally the third column is the number of request.
- **virtualRequest:** The different virtual request. A file similar to the SN but for each of the Virtual Networks.
- **virtualHidden:** The hidden hops demand
- **virtualMetric1_:**Bandwidth demand of the virtual request similar to the files of the substrate network
- **virtualMetric2_:**CPU demands of the virtual requests similar to the files of the substrate network
- **virtualFinish.dat:** Finishing file name

III.3.2. Files generated by ALEVIN

In the folder home/directory there are several files generated by ALEVIN. The first one has the following identifier.

- a: The number of the virtual request
- x:The origin virtual node
- y:The destination virtual node

The files have the following meaning:

- **virtual_to_Real_a:** The node mapping files of a Virtual request “a”. In this file the virtual nodes mapped into the physical one are shown
- **virtualnodemapping_a:** Virtual node mapping success.
- **VirtualAttend_a_x_y:** In this file the physical path between the virtual pair of nodes(x,y) is shown

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Also other types of files are created, in the same directory: README files, Scenario-template files, and Scenario-mapped files. They have also a identifier `a_b_c_d_e` meaning the following:

- a: Substrate Network Size
- b: Number of virtual networks
- c: Number of nodes per virtual network
- d: run number (each scenario set will have different runs with different random resources and demands but the same topology)
- e: load: Ratio of substrate network resources distributed among the set of virtual networks

The files have the following meaning:

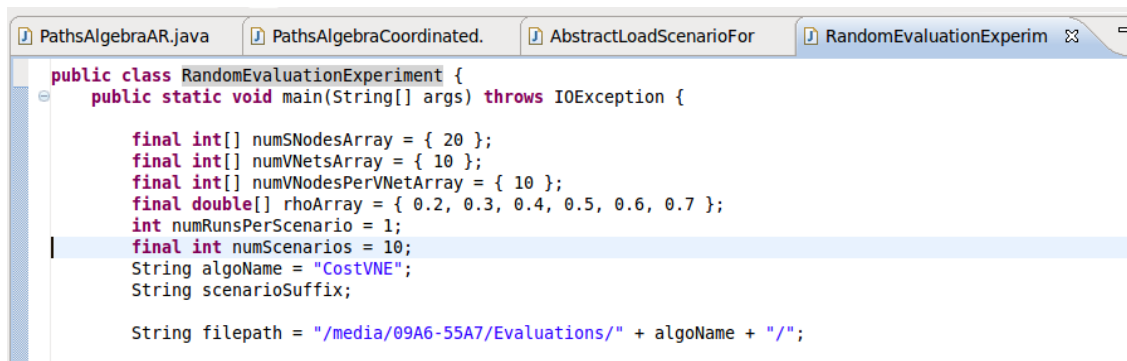
- **README:** File with information about the resources of the scenario (not important)
- **Scenario-template:** This xml file contains the scenario that is going to be mapped
- **Scenario-mapped:** This xml file contains the scenario after the mapping (from here the results are extracted).

III.4 Evaluation metrics generation

To evaluate the performance of the algorithms, there are some evaluation metrics implemented in ALEVIN, p.e.: Cost, Revenue, NodeStress, LinkStress, Virtual Network Acceptance Ratio, etc.

These metrics are generated from the scenario-mapped files. There is a java class to generate the files with the evaluation metric results. This class is located in the `tests.algorithms.generationEvaluation` package (figure 26). The class is called `RandomEvaluationExperiment.java`, as it is shown in the picture it has a field called `algoName` where the name of the algorithm must be put. Then, there is a `filepath` field. This field should be filled with the path where the Scenario-mapped files are.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service



```
public class RandomEvaluationExperiment {
    public static void main(String[] args) throws IOException {

        final int[] numSNodesArray = { 20 };
        final int[] numVNetsArray = { 10 };
        final int[] numVNodesPerVNetArray = { 10 };
        final double[] rhoArray = { 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 };
        int numRunsPerScenario = 1;
        final int numScenarios = 10;
        String algoName = "CostVNE";
        String scenarioSuffix;

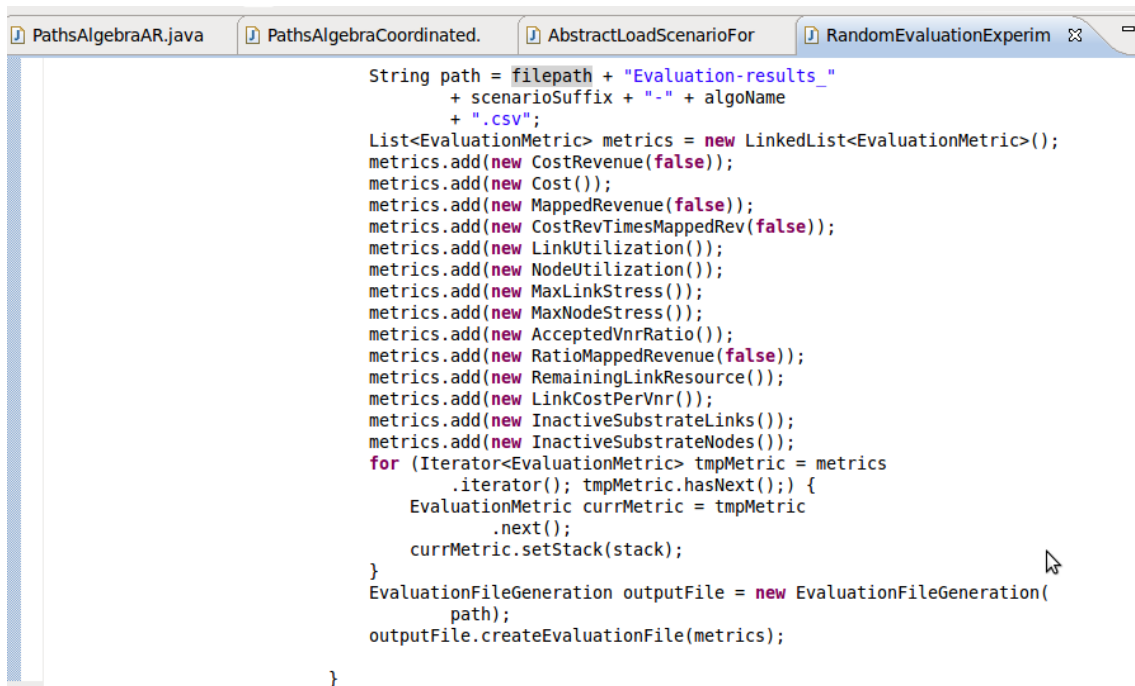
        String filepath = "/media/09A6-55A7/Evaluations/" + algoName + "/";
```

Figure 25. Evaluation Metric Parameters

This class also has the set of metrics to calculate, you can delete the metrics you think are inconvenient.

After the class is modified following your criteria. You can execute it by pressing right click over it and selecting **Run As** → **Java Application**. When the execution finishes, in the path of the scenarioMapped you will find some files called EvaluationResults with the same identificator a_b_c_d_e. These files have the final results and you can use them to realize graphs or whatever you want.

There are two types of simulation the online simulation and offline, for this thesis the offline simulation is use, as it shows in figure 27.



```
String path = filepath + "Evaluation-results_"
    + scenarioSuffix + "-" + algoName
    + ".csv";
List<EvaluationMetric> metrics = new LinkedList<EvaluationMetric>();
metrics.add(new CostRevenue(false));
metrics.add(new Cost());
metrics.add(new MappedRevenue(false));
metrics.add(new CostRevTimesMappedRev(false));
metrics.add(new LinkUtilization());
metrics.add(new NodeUtilization());
metrics.add(new MaxLinkStress());
metrics.add(new MaxNodeStress());
metrics.add(new AcceptedVnrRatio());
metrics.add(new RatioMappedRevenue(false));
metrics.add(new RemainingLinkResource());
metrics.add(new LinkCostPerVnr());
metrics.add(new InactiveSubstrateLinks());
metrics.add(new InactiveSubstrateNodes());
for (Iterator<EvaluationMetric> tmpMetric = metrics
    .iterator(); tmpMetric.hasNext();) {
    EvaluationMetric currMetric = tmpMetric
        .next();
    currMetric.setStack(stack);
}
EvaluationFileGeneration outputFile = new EvaluationFileGeneration(
    path);
outputFile.createEvaluationFile(metrics);
}
```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Figure 26. Evaluation files

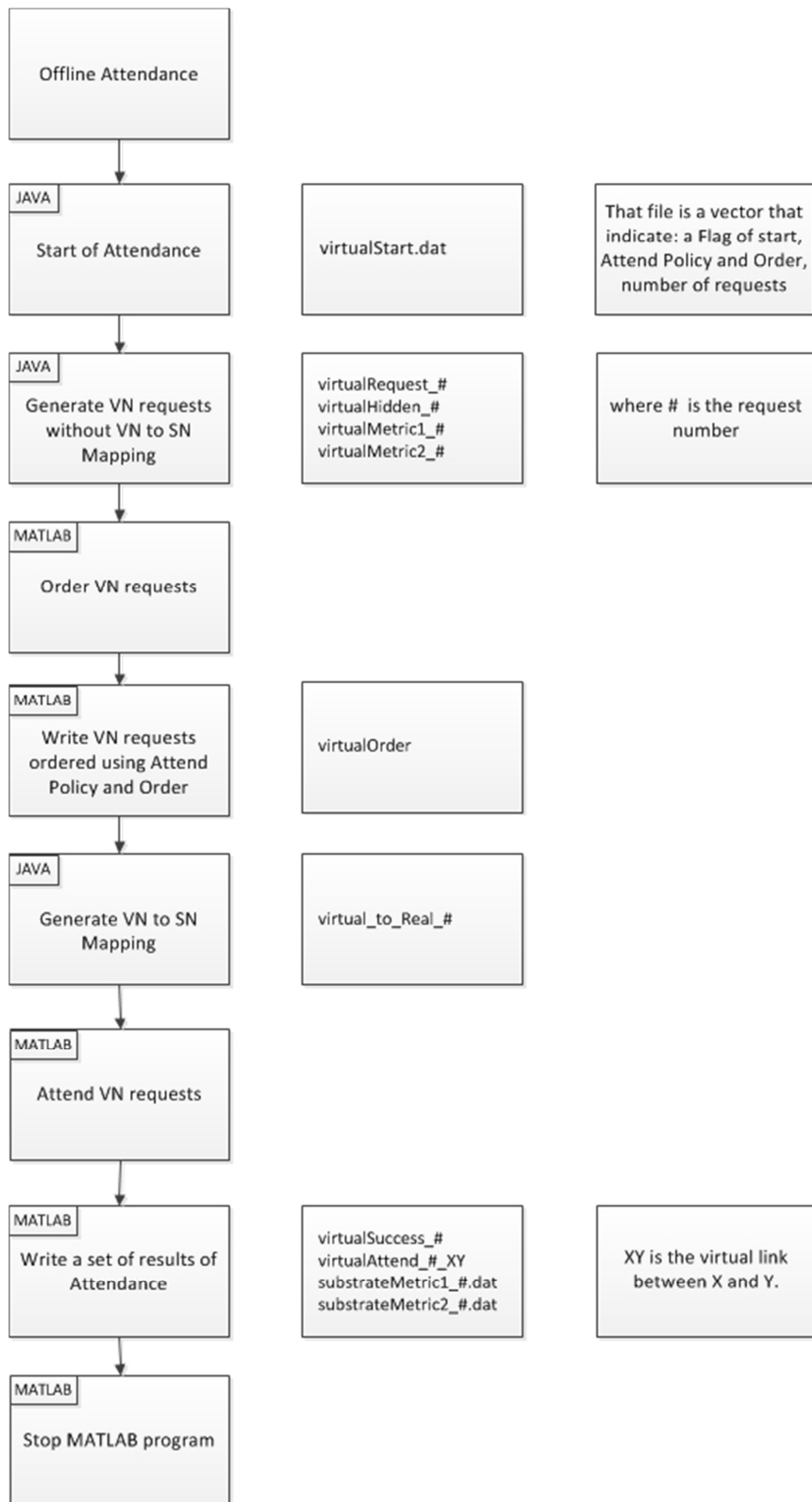


Figure 27. Diagram of the ALEVIN online simulation

Chapter IV

Scenarios

This Master thesis proposes the use of new nonlinear and linear parameters in the network embedding strategy. This work analyzes and compares results with relevant parameters based strategies.

Consequently, to evaluate the inclusion of new parameters into the paths algebra-based approach, we compare four types of scenarios. Each scenario is constituted by:

- The topology of the network
- Resource and demand deployment
- Nonlinear or Linear metrics
- Metric priority

IV.1. Topology Creation

For each of the scenarios, the Waxman topology was used. As it was explained in the previous chapter, the ALEVIN software will be in charge of generating the substrate network topology, using the following formula:

$$P(u, v) = \alpha e^{-\left(\frac{d(u,v)}{\beta * L}\right)} \quad (1)$$

In which, $0 < \alpha, \beta \leq 1$. Larger values of β result in graphs with higher link densities, while small values of α increase the density of short links relative to longer ones. “d” is the Euclidean distance, between the nodes v and u, and L is the maximum Euclidean distance between any pair of nodes. An increase in α will increase the probability of edges between any nodes in the graph. This topology generation procedure is used for the substrate network as well as for all virtual networks.

For the four scenarios the value of α and β is equal to 0.5. These values are chosen because it provides a network with relative medium links density. Furthermore as the coordinates of the nodes are going to be uniformly in an 1x1 square area, the average distance between two nodes will be 0.5 and the maximum distance $\sqrt{2}$. Thus according to the formula explained before (equation 1), the average probability for

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

creating a link between any two nodes is $\frac{1}{4}$.

IV.1.1. Substrate Network in MATLAB

Once the ALEVIN software generated the SN, the topology must be included in the MATLAB program. In order to achieve this, the SN is introduced into a Matrix “M” called the Adjacency Matrix. An example of a SN with 5 nodes is shown in the following figure

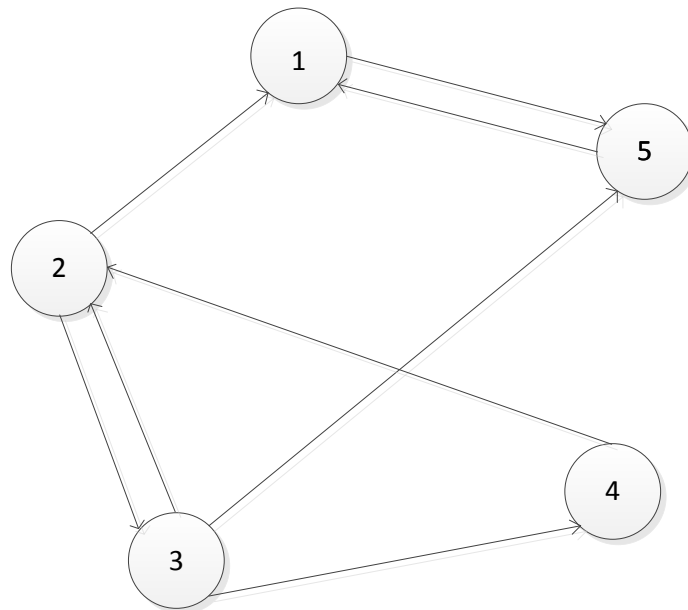


Figure 28. Substrate network example

The adjacency matrix M is constituted of elements $[i, j]$. If this value is equal to 1 then there will be a link connecting the source node i to the destination node j . So, for instance the SN in figure 28 can be modeled using the adjacency matrix shown in table 9.

	1	2	3	4	5
1	0	0	0	0	1
2	1	0	1	0	0
3	0	1	0	1	1
4	0	1	0	0	0
5	1	0	0	0	0

Table. 9. Substrate Network adjacency matrix

IV.2. Resource and demand deployment

A load targeted resource and demand deployment is available in ALEVIN and

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

consists on two steps:

In the first step, the substrate network will be equipped with resources by uniformly distributing each node resource in a given interval $(0, NR^{\max})$ for every substrate node (the interval for CPU resources is $[0, 100]$) as well as each link resource in a given interval $(0, NR^{\max})$ for every substrate link (the interval for BW resources is $[0, 100]$), as usually done in literature.

For the second step, the goal is to achieve a certain average load of every resource. The creation of node demands and link demands that fulfill these load requirements comprises different challenges. Because in the Waxman Topology the amount of nodes is fixed, the calculation of the CPU resources in the node, can be given by the following formula:

$$E[NR^{MAX}] = \frac{NR^{MAX}}{2} \quad (2)$$

The formula to calculate the overall demands of the virtual network is the following:

$$E[DM] = \rho * E[NR] * \left(\frac{|V|}{|V|^k * k^{MAX}} \right) \quad (3)$$

Where

- DM: Demands of the virtual nodes
- V: Number of nodes in the substrate network
- k: Number of virtual network
- $|V^k|$: Number of nodes per virtual Network
- ρ : The amount of demands of the Virtual Request, this means if the loads are higher then the VNR will requested more resources.

The average of the links resources is calculated similar to the nodes resources using the following formula:

$$E[LR] = \frac{LR^{MAX}}{2} \quad (4)$$

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

To calculate the average of demands of the links is necessary to understand that the number of links in a SN and a VNR is not fixed. Nevertheless the average probability of creating a link between a pair of nodes is known, for this thesis it will be $\frac{1}{4}$. Thus, the average number of edges $E[|A|]$ in a directed graph is given by:

$$E[|A|] = \frac{1}{4} * |V| * (|V| - 1) \quad (5)$$

Therefore, the average of the links demands is calculated by:

$$E[LD] = \rho * E[LR] * \left(\frac{E[|A|]}{E[|A^k|] * k^{MAX}} \right) \quad (6)$$

Which result in:

$$LD^{MAX} = 2 * E[LD] \quad (7)$$

However in order to use the above formula to generate both the links and node resources and demands, it is necessary to ensure that the $LD^{MAX} < LR^{MAX}$. In particular it is important to ensure that $E[|A|] < E[|A^k|] * k^{MAX}$ in Eq (6) holds for $0 < \rho < 1$. To achieve that the following constraint needs to be enforced:

$$|V|^2 < k^{MAX} * |V^k|^2 \quad (8)$$

If the above formula is not satisfied the approximation provided above will achieve a higher load value than the targeted load ρ for the link resources.

IV.3. Linear and Non Linear Metrics

Once the topology and the resources and demands are created, the following metrics are going to be included into the “Paths-Algebra” framework:

- Packet Loss Ratio (PLR)
- Availability
- Energy Consumption

There are two types of metric, the linear and the non-linear. Previous work only analyzes the “Paths Algebra” using linear metric such as CPU, Hops or Bandwidth. In the case of this Master Thesis non-linear metrics are used.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

In this part of the chapter, an explanation on how this metrics were generated in the ALEVIN environment and the MATLAB program is stated, also the formula and synthesis used for the “Paths Algebra” framework is explained.

IV.3.1. Packet Loss Ratio

The first metric to be included in the simulation environment is the Packet Loss Ratio; this metric measures the percentage of packet loss in a network. Furthermore this value is a non-linear metric, which means that is not follows a linear tendency.

IV.3.1.1. Inclusion into the simulation environment

IV.3.1.1.1 Metric Creation

Because these metric measures the ratio of loss of packet in a network, the range generated will be between 0% and 2%, it will be a uniformly distributed number and it will be created for the links of the substrate networks. In order to include the metric into the ALEVIN environment, the next procedure is follows:

Using the same adjacency matrix of the substrate network, a new matrix is created for the Packet Loss Ratio. So when the element $a[i, j]$ of the adjacency matrix M is equal to 1, the link with the source node i and destination j will have a PLR metric. Take for instance the SN with the PLR value in each link (figure 29).

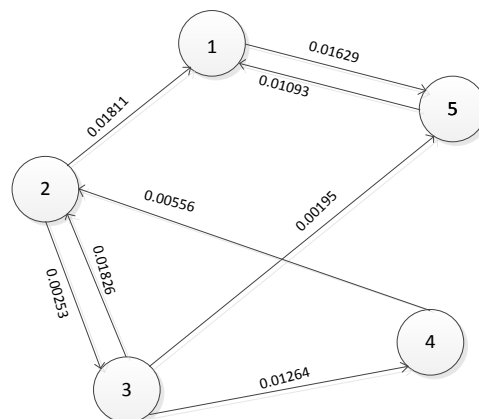


Figure 29. Packet Loss Ratio in SN

With the above figure, a matrix called P is created with the element $PLR[i, j]$ that represent the value of the Packet Loss Ratio of each link in the SN, as it shown in table 10.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

	1	2	3	4	5
1	0	0	0	0	0.01629
2	0.01811	0	0.00253	0	0
3	0	0.01826	0	0.01264	0.00195
4	0	0.00556	0	0	0
5	0.01093	0	0	0	0

Table 10. Packet Loss Rate adjacency matrix

IV.3.1.1.2. Synthesis

Once the PLR is created, a synthesis must be use in order to have a value that can characterizes the different paths as far as the constraint imposed. For the PLR the formula use will be “1-PLR”, the synthesis associated is multiplication-mult () and the ordering relation will be \leq .

To explain better the use of the formula and the synthesis the following example is presented:

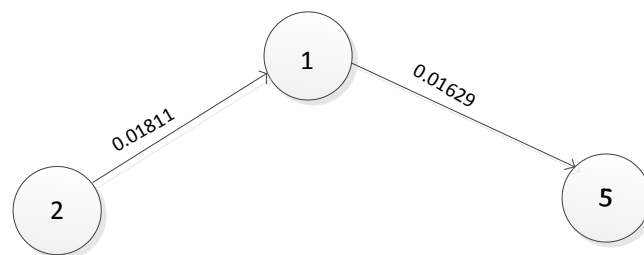


Figure 30. Path1 Packet Loss Rate

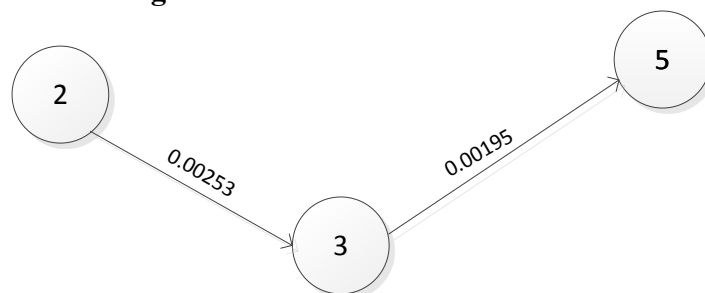


Figure 31. Path 2 Packet Loss Ratio

Considering two paths between the origin nodes “2” and the destination node “5” with different Packet Loss Ratio. Path 1 in figure 30 is composed of the nodes 2, 1, 5 and Path 2 in figure 31 is composed of the nodes, 2, 3, and 5. Using the above

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

formula and synthesis the corresponding value for each path is the following.

- **Path1 PLR:** 0.034115124 or 3,4%
- **Path 2 PLR:** 0.00448559 or 0.4%

IV.3.2 Availability

IV.3.2.1 Inclusion into the simulation environment

IV.3.2.1.1. Metric Creation

This metric will have uniformly distributed values in a range between 95% and 100% corresponding to each link's availability in the substrate network. For the inclusion of this metric into the MATLAB program the same procedure as before was follows.

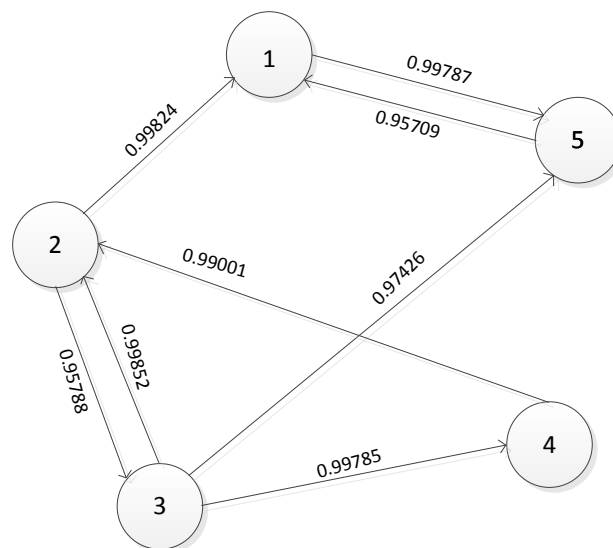


Figure 32. Availability in SN

First, using the same adjacency matrix of the substrate network, a new matrix is created for the Availability. When the element $a[i, j]$ of the matrix M is equal to 1, the link with the source node i and destination j will have an Availability value.

For instance the values of links in the substrate network in figure 32, can be characterize in a matrix called A , where the element $Aval[i, j]$ represents the value of the Availability of each link in the SN, as it shown in table 11.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

	1	2	3	4	5
1	0	0	0	0	0.99787
2	0.99824	0	0.95788	0	0
3	0	0.99852	0	0.99785	0.97426
4	0	0.99001	0	0	0
5	0.95709	0	0	0	0

Table 11. Availability Metric

IV.3.2.1.2. Synthesis

Like in the case of PLR, once the metric is created, a synthesis must be use in order to have a value that can characterizes the different paths. For the Availability the synthesis associated is multiplication-mult (\cdot) and the ordering relation will be \geq . To illustrate better this explanation, the following example is presented:

Considering two paths between the origin node “2” and the destination node “5” with different availability (figure 33 and figure 34). Path 1 is composed of the nodes 2,1,5 and the Path 2 is composed of the nodes, 2,3,5.

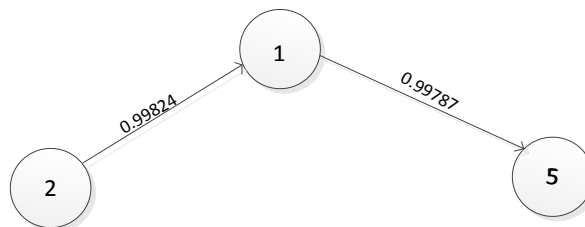


Figure 33. Path 1 Availability

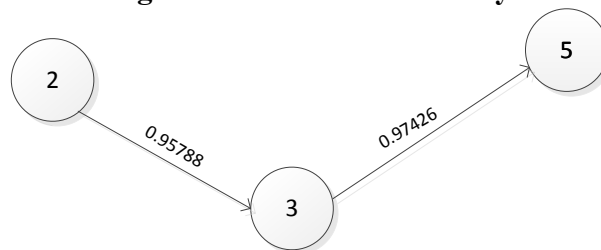


Figure 34. Path 2 Availability

Using the synthesis and ordering relations, the corresponding value for each path is the following.

- **Path1 Availability**=0.933233219 or 93,3,%

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

- **Path 2 Availability**=0.996123499 or 99,6 %

IV.3.3. Energy

The third and last metric to be included in the simulation environment is the energy. This metric correspond to the energy consumption of each nodes in the substrate network

IV.3.3.1. Inclusion into the simulation environment

IV.3.3.1.1. Metric Creation

As explained in previous sections, in order to create the energy metric the next model will be use $ENERGY = A + B * BW$ eqt(9) , BW will be the maximum inflow and outflow bandwidth of every node in the network and “A” and “B” would be chose between 0 and 1.

1	2	3	4	5
0.807113781	1	0.983663866	0.666762733	0.787317637

Table 12. Energy consumption metric

As in previous cases in order to use this metric in the ALEVIN environment, a vector is generated in MATLAB. Each element in the vector will correspond to the value of the energy of each node. For instance the energy consumption of each node in figure 35 can be represented in table 12.

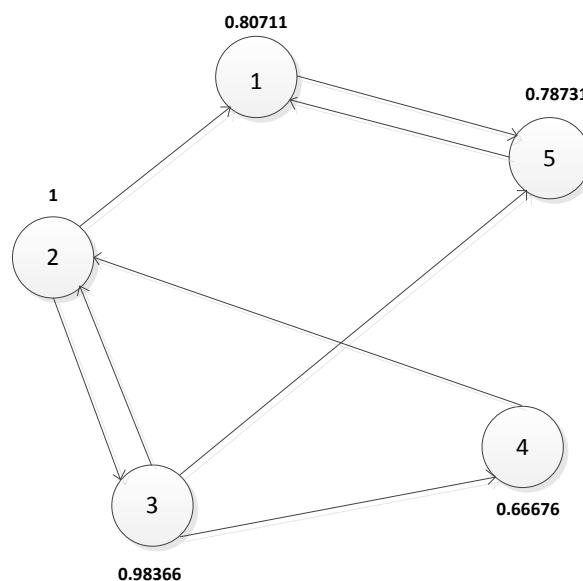


Figure 35. Energy consumption in SN

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Once the energy vector is generated using equation (9), its need to be included in the LADN algorithm, the problem is that paths algebra has been developed associating values of the link of the substrate network instead of the nodes. In order to deal with the metrics associated to the network's nodes it is necessary to perform the following transformation.

Consider, for example, the part of a network shown in Fig. 36. In this figure, the weights are associated to the nodes. Nodes A, B and C energy consumption are given by Energy (A), Energy (B) and Energy (C) respectively.

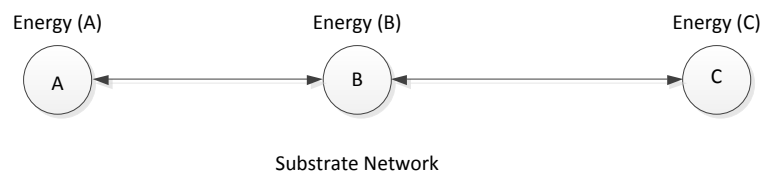


Figure 36. SN example

As one possible objective could be to minimize Energy consumption CPU. The new energy metric can use the metrics combination function provided by the paths algebra and reduce the metrics to $(\text{sum} [\text{Energy}(A), \text{Energy}(B)], \text{Energy}(C))$ as shown in Figure 37.

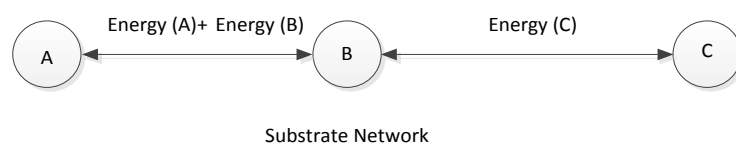


Figure 37. New metric combination

IV.3.3.1.2. Synthesis

For the Energy metric the synthesis associated will be adding -add () and the ordering relation will be \leq . An example is shown in figure 3, with a SN with 5 nodes, each node will have an energy consumption value. Considering two paths between the origin nodes "2" and the destination node "5" with different energy consumption (Figure 38 and figure 39).

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Using the above procedure, the value of each of the nodes consumption was transformed as a function of the links as it shows in the above figures. After using the synthesis the value of each path is the following:

- **Path1 Energy Consumption:** 2.594431419
- **Path 2 Energy Consumption:** 2.770981503

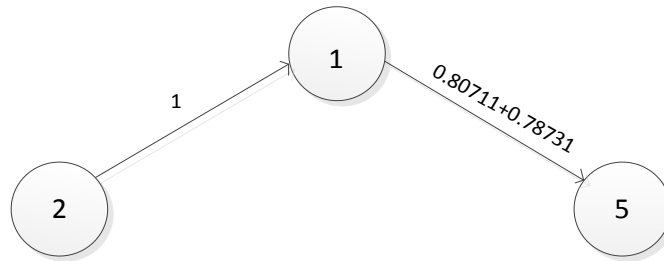


Figure 38. Path 1 Energy Consumption

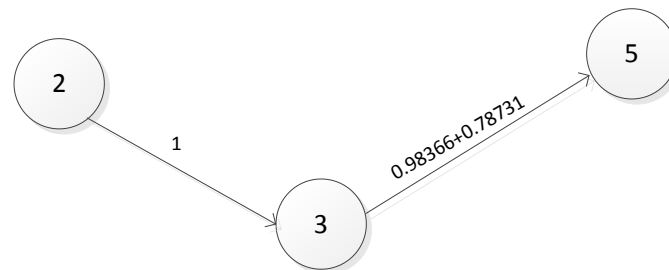


Figure 39. Path 2 Energy Consumption

IV. 4. Metric Priority

For this master thesis four scenarios are going to be compared using the new nonlinear and liner metric included in the ALEVIN environment, each scenario will have different metric priority. Table 13 summarizes the metric ordering in each scenario.

Scenario	Optimization Criteria	Metric Ordering	Metric Priority	Physical Constraint
1	Minimize Cost	Hops, Bandwidth, CPU	Hops	BW, CPU
2	Maximize Throughput	PLR, Hops, Bandwidth, CPU	PLR	BW, CPU
3	Maximize Availability in the Substrate network	Availability, Hops, Bandwidth, CPU	Availability	BW, CPU
4	Minimize Energy consumption	Energy, Availability, Hops, Bandwidth, CPU	Energy	BW, CPU

Table 13. Scenarios and metric priority

IV.5. Simulation Parameters.

For each scenario the same parameters are used. Table 14 summarizes the parameters used in the simulations. The values chosen for the number of substrate nodes and the number of virtual nodes per virtual network is used because it provides the effective runtime of the algorithm and it is a more realistic scenario. The loads' range from 0.1 through 0.99, that means that it will covers situations from lightly loaded to heavily loaded situation. This increase will be for the demands of resources; nevertheless the amount of VNR will be the same for every scenario

As Waxman the topology generation is probabilistic, it was perform $N = 20$ runs for each value of load.

In the next chapter the results from the above simulation are shown and explained with the data summarizes in the beginning of the chapter. The main idea is to analyze if “paths algebra” mapped the different links of the VNR into the SN using the new metrics priority. The total number of times that each algorithm has been exercised is given by $N * k * \rho = 20 * 10 * 11 = 2,200$, and as we are comparing 4 different scenarios, so the total number of VNR assignment attempts is 8,800.

This dimension of example will be large enough to provide quality comparison between the scenarios and help understand the behaviors of each one

Parameters Descriptions	Value
Nodes in the substrate network	20
Nodes in the Virtual Networks	10
Number of Virtual Networks (k)	10
Loads (ρ)	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.95,0.99
Number of Runs per load (N)	20

Table 14. Simulation Parameters

Chapter V

Results

In this chapter the different result and parameter used to compare the scenarios are shown. The idea is to see if the ordering of the new metrics included into the ALEVIN environments can affect the way that “Paths Algebra” chooses the paths to be mapped into the substrate network.

Among the several parameters that can be used to compare the results obtained by the different scenarios, the following were chosen:

- **Average value of Packet Loss Ratio:** The value the Packet Loss Ratio of each path mapped divided by the total number of the path being mapped
- **Average Value of Availability:** The value of the Availability of each path mapped divided by the total number of the paths being mapped.
- **Average Value of Energy:** The value of the energy consumption of each path mapped divided by the total number of the paths being
- **Average Value of hops:** The value of the hops of each path mapped divided by the total number of the mapped being mapped
- **Cost/Revenue(C/R):** The lower this relationship is, the better the result.
- **Percentage Accepted VNR:** the percentage of virtual networks successfully mapped.
- **Rate Mapped Revenue:** the percentage of mapped revenue over the total revenue that could be mapped

Is important to point that that the above parameters were calculated for every load (ρ) of each of the scenarios.

V.1.Simulation Results

V.1.1. Packet Loss Ratio

The first parameter is the Average value of the Packet Loss Ratio. This is the first parameter used to compare the fours scenarios explained in chapter IV.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Using the value of the Packet Loss Ratio of each path mapped, an Average was calculated for every load. To better illustrate this, the following example is presented:

- One substrate network with 5 nodes in figure 40
- Value of PLR between 0% a 2% in each link
- Two Virtual Network Request with 5 nodes each (figure 40 and figure 41)
- Load(ρ): 0.4

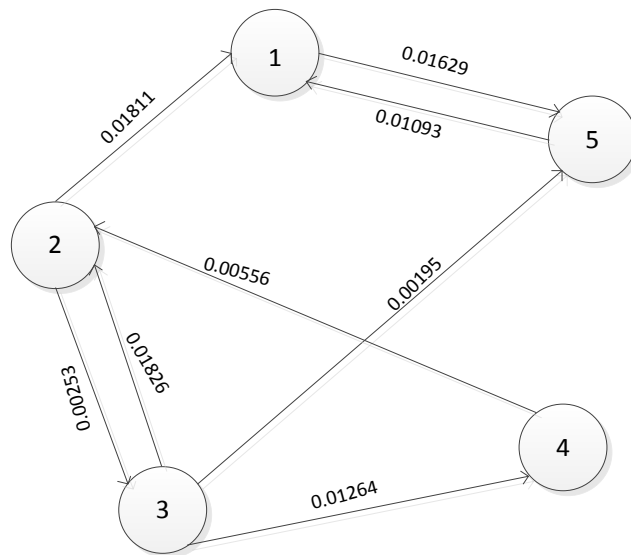


Figure 40. Packet Loss Ratio value

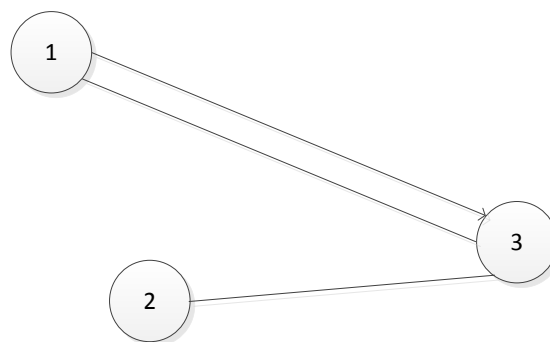


Figure 41. Virtual Request 1

Table 15 and 16 shows the result of mapping the VNR's into the substrate network. The first column corresponds to the Virtual Links. The second one is the path of the SN in which the virtual links are mapped. And finally the last column will correspond to the value of the Packet Loss Ratio of the path.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

In order to calculate the average of the Packet Loss Ratio of this example is necessary to add the value of the PLR each of paths that are mapped. For this example the result of the additive operation will be **0.063824**. Once this value is calculated it must be divided between the numbers of paths mapped, for this case the value this value will be **6**. Finally the resulting average of PLR for this example will be **0.01063**.

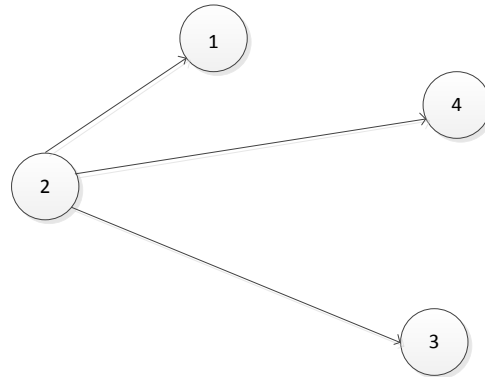


Figure 42. Virtual Request 2

VNR 1		
Virtual Link	Physical Path	Packet Loss Ratio
1,3	2,3,4	0.015154801
3,1	4,2	0.005569964
3,2	4,2,3,5	0.01003057

Table 15. Virtual Network 1

VNR 2		
Virtual Link	Physical Path	Packet Loss Ratio
2,1	2,3	0.002539736
2,4	2,3,4	0.015154801
2,3	2,3,5,1	0.015374159

Table 16. Virtual Network 1

The procedure explained before was done for every loads (ρ) of each scenario. The idea is to graph the value of the average PLR of the fours scenarios in function of the loads. This means that the horizontal value will be the value of the PLR and the vertical the different loads.

Table 17 represents the value of the PLR for each load for only one scenario; this same table is done for the remaining ones.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Loads	Average PLR
0.1	1.986919
0.2	2.053837
0.3	2.023622
0.4	2.079379
0.5	2.077542
0.6	2.132567
0.7	2.11E+00
0.8	2.165483
0.9	2.144338
0.95	2.200522
0.99	2.225807

Table 17. Average PLR vs Loads. Scenario 1

Using the tables for the four scenarios (Annex B), the average PLR was graph as it shows in figure 43.

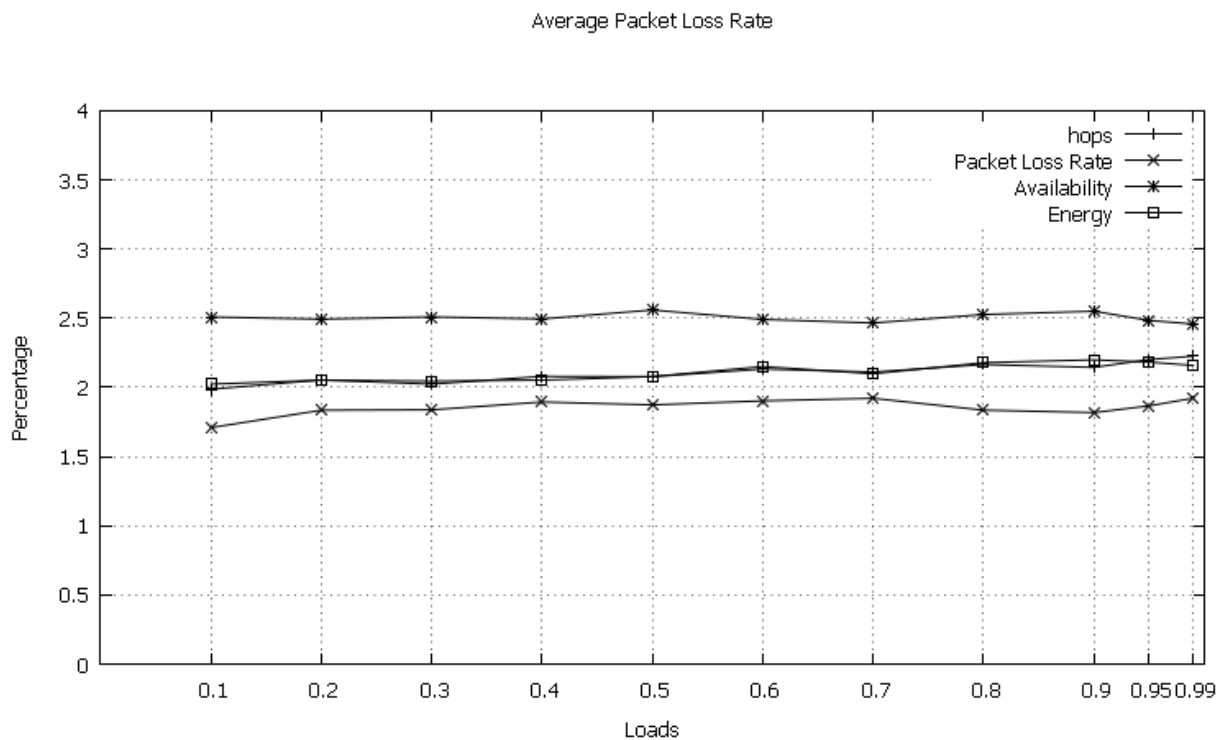


Figure 43. Average Packet Loss Rate

This first graph shows that the lowest average of PLR is when the Packet Loss Ratio has the priority (scenario 2). On the other hand in scenario 3 the value of PLR is

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

the highest, and for the case of the scenario 1(hops) and 4(energy) the values of PLR are very similar but higher than scenario.2

The observed behavior can be understood if we considered that in the case of the scenario 2, the PLR has priority. Consequently, the “Paths Algebra” will try to choose the paths of the SN that has the lowest Packet Loss Rate. So, calculating the value of the average of PLR of the paths mapped for scenario 1, the value will be the lowest, and so it can be concluded that “Paths Algebra” is choosing the best option.

For scenario 2 in which the priority is the Availability “Paths Algebra” did not consider the value of the PLR. So the paths were chosen based on the availability of the links. In consequence the PLR was the highest.

Lastly for scenario 1 and 4, PLR was also not considered, but as the priority are hops and energy, the amount of hops of the paths mapped are shorter than in the case of scenario 3, so the value of PLR will be smaller.

V.1.2. Availability

In the case of this metric, the same procedure as before was used to calculate the average availability of each scenario. The following example explained how this value was calculated.

This example is characterized by:

- One substrate network with 5 nodes (figure 44).
- Value of Availability between 95% and 100% in each link
- Two virtual requests with 5 nodes (figure 41 and 42).
- Load(ρ): 0.4

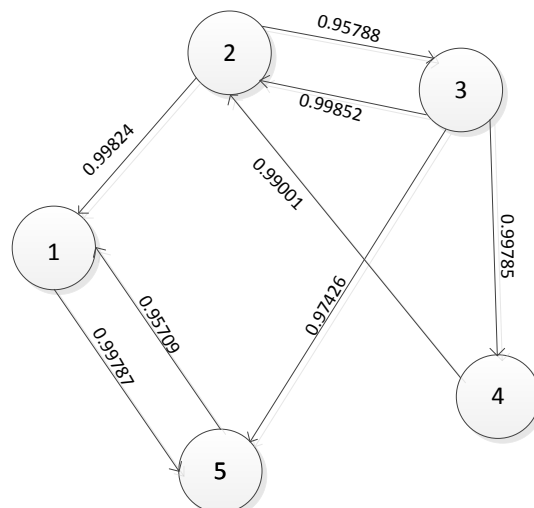


Figure 44 Availability of SN

Table 18 and 19, shows the value of the availability of the paths that were mapped for each Virtual Network.

In order to calculate the average, the availability of each path is added. For this example the result will be **5.84397**. Once is calculated it must be divided between the numbers of paths mapped, which is **6**. So, for this particular case the average of the availability will be **0.97399**.

VNR 1		
Virtual Link	Physical Path	Availability
1,3	2,3,4	0.955829207
3,1	4,2	0.990014023
3,2	4,2,1,5	0.986176233

Table 18. Virtual Network 1 Availability

VNR 2		
Virtual Link	Physical Path	Availability
2,1	2,3	0.957880654
2,4	2,3,4	0.955829207
2,3	2,1	0.998244427

Table 19. Virtual Network 1 Availability

This average was calculated for each of the four simulations. Once calculated, a table of the different value in function of the loads was done. As in PLR, the following table represents only the value of the average availability for one scenario. In the Annexes the remaining tables are represented.

Figure 45 shows that in the case of the scenario 3, the average of the availability is the highest. On the other hand for the other three scenarios (Hops, PLR and energy) the value is lower.

This means that when Availability has the priority (scenario 1), “Paths Algebra” will mapped the paths in the SN that has the highest availability, hence, the average

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

value of the availability for this scenario will be the highest, as it shows in figure 45. On the contrary for the other scenarios the ordering is different, they not considered availability as the metric with the biggest priority, so the chosen paths will be based only on the priority of each scenario, so consequently the availability will be lower.

Loads	Average Availability
0.1	95.162778
0.2	95.029028
0.3	94.983103
0.4	94.881892
0.5	94.924477
0.6	94.803908
0.7	94.740129
0.8	94.64432
0.9	94.972658
0.95	94.743325
0.99	94.865986

Table 20. Average Availability vs Loads. Scenario 1

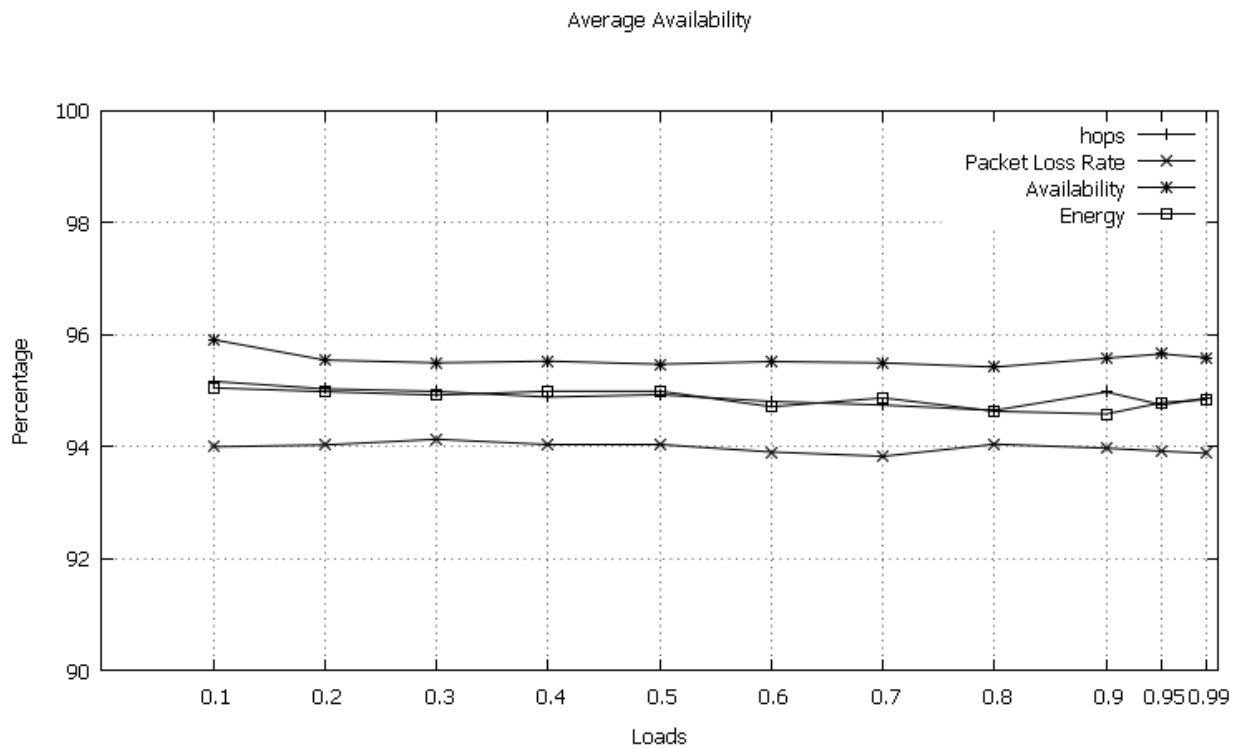


Figure 45. Average Availability

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Is important to point out that for both the availability and the packet loss rate, the increase of the load does not affect the value of the average. This means that even though the Virtual Requests are demanding more resources, the packet loss rate and availability remain almost the same

V.1.3. Hops

For the case of the hops metric the same procedure as before was used to calculate the value of the average. Considered the following example:

- One substrate network with 5 nodes (figure 46).
- Two virtual requests with 5 nodes (figure 41 and 42).
- Load(ρ): 0.4

The amount of hops for each Virtual Network mapped is shown in table 21 and 22. To calculate the average; first the accumulative value of hops needs to be calculated, for this case it will be **10**.

This value is then divided by **6**, which is the total amount paths mapped in the SN. The final result of the average value of the hops for this example is **1.666**.

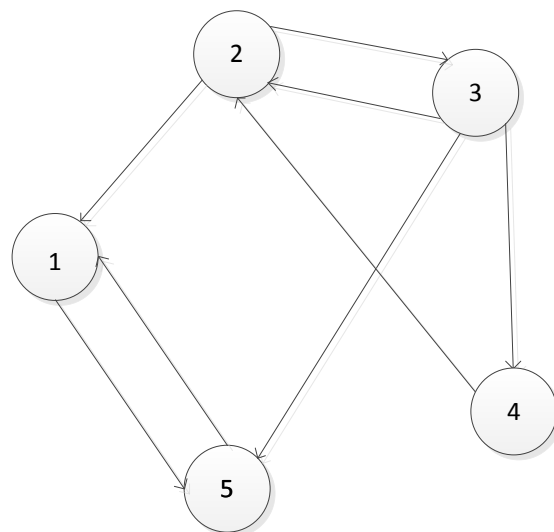


Figure 46. Substrate Network Hops

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

VNR 1		
Virtual Link	Physical Path	Hops
1,3	2,3,4	2
3,1	4,2	1
3,2	4,2,3,5	3

Table 21. Virtual Network 1 Hops

VNR 2		
Virtual Link	Physical Path	Hops
2,1	2,3	1
2,4	2,3,4	2
2,3	2,1	1

Table 22. Virtual Network 2 Hops

This value could indicate which simulation tried to find the shortest path. In table 10 an example on how it was graphs is shown.

The result in figure 47 shows that the averages of the hops are the lowest for the first scenario. On the other hand for the second and third scenario the average of hops are the worst. And finally for the fourth scenario the value of hops very similar to the scenario 1.

In the case of scenario 2 as it was explained before the metric priority is the Packet Loss Ratio, so when paths are chosen, the amount of hops would not be considered, only the value of the PLR of the path. Thus the average of hops is going to be high. This is similar for scenario 3, which is when availability has the highest priority.

Loads	Hops Average
0.1	1.871233
0.2	1.9318718
0.3	1.9436485
0.4	1.9691689
0.5	1.9692833
0.6	2.0123818
0.7	2.0385164
0.8	2.0434783
0.9	2.0648968
0.95	2.0648968
0.99	2.0616438

Table 23. Hops vs Loads. Scenario 1

Comparing the results for the scenario 1 and 4, it shows that the values are very similar, this can be explained because for scenario 1, the hops between the pair of nodes has the priority, hence the paths found will have a less amount of hops. In the case of scenario 4, the energy of the nodes has the highest priority so “Paths Algebra” will tried to find the paths with the less energy consumption but if the value of energy consumption is small, consequently the amount of hops will also be smaller, so the values of hops between the scenarios will be very similar

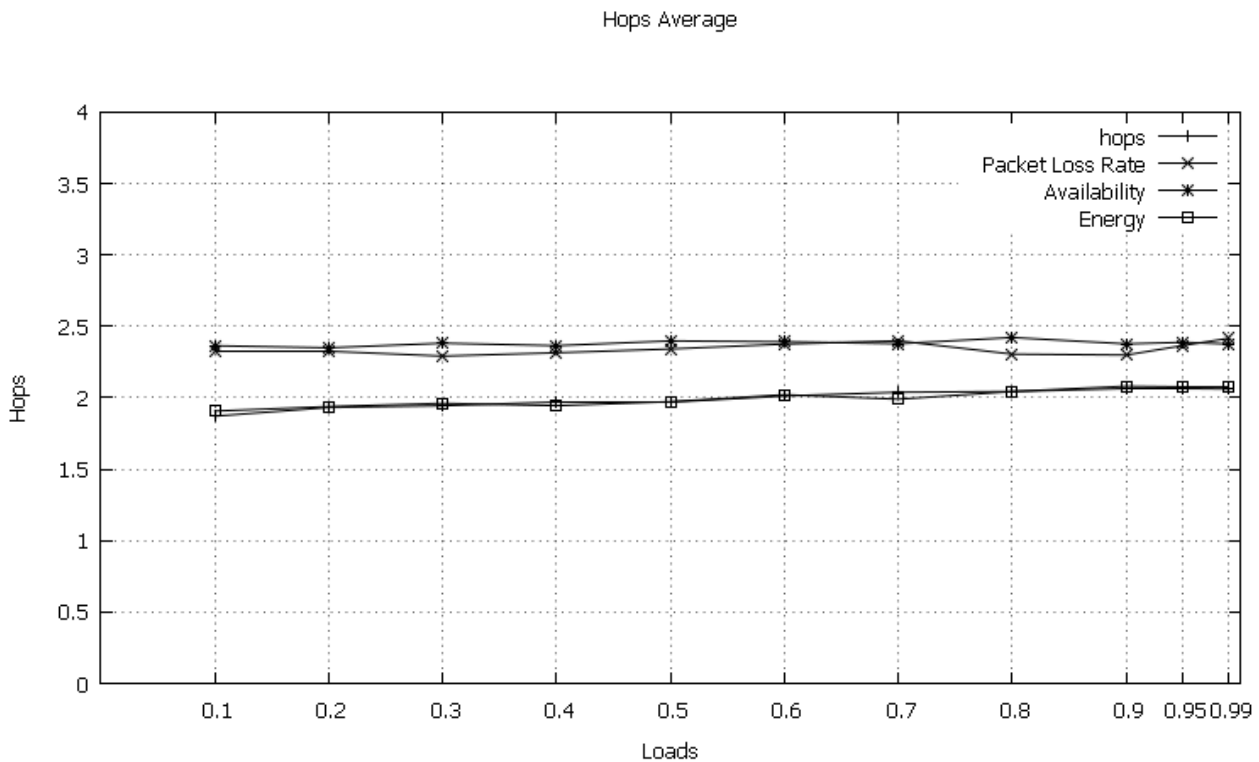


Figure 47. Hops Average

V.1.4. Energy

This metric corresponds to the value of the energy consumption of each node in the Substrate Network. As in previous case to explain the calculation of the average value of the energy, the following example is presented:

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

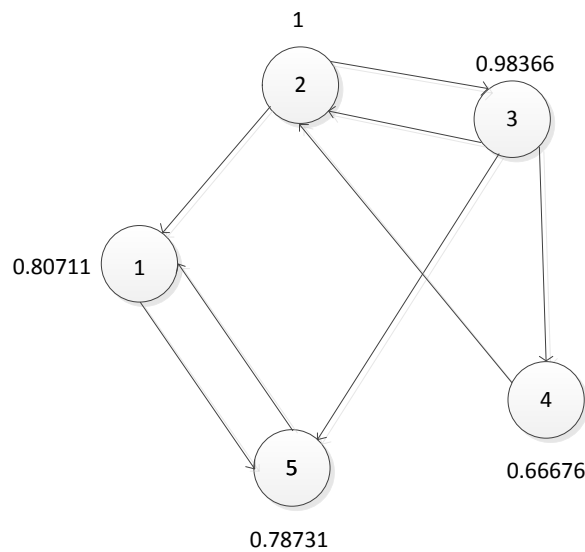


Figure 48. Substrate Network Energy

Considering a substrate network with 5 nodes (figure 48), two Virtual Network request with 3 nodes each (figure 41 and figure 42) and a load of demands of 0.4. In order to calculate the energy average is essential to know the value of the energy of each paths mapped of the SN (table 24 and table 25).

Using this data from the table, the accumulative energy is calculated. Adding the metric for each path, results in a value of **14.01958773**. Then the number of paths mapped in the SN is divided by this accumulative result. For this example the number of paths is **6**. Finally the average of the energy consumption for this example will be **2.33659**.

Lastly a table was created for this metric (Table 26); each scenario is going to have its own table, in which the value of the average energy is shown in function of the different load. The remaining tables for the other scenarios are shown in the Annex B

VNR 1		
Virtual Link	Physical Path	Energy
1,3	2,3,4	2.650426599
3,1	4,2	1.666762733
3,2	4,2,1,5	3.261194151

Table 24. Virtual Network 1 Energy

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

VNR 2		
Virtual Link	Physical Path	Energy
2,1	2,3	1.983663866
2,4	2,3,4	2.650426599
2,3	2,1	1.807113781

Table 25. Virtual Network 2 Energy

In this graph the value of the average energy is very similar for scenario 1 and 4. This happens because while trying to reduce the energy consumption of the paths, the amount of hops is also optimizes. To better explained this situation considered the following example:

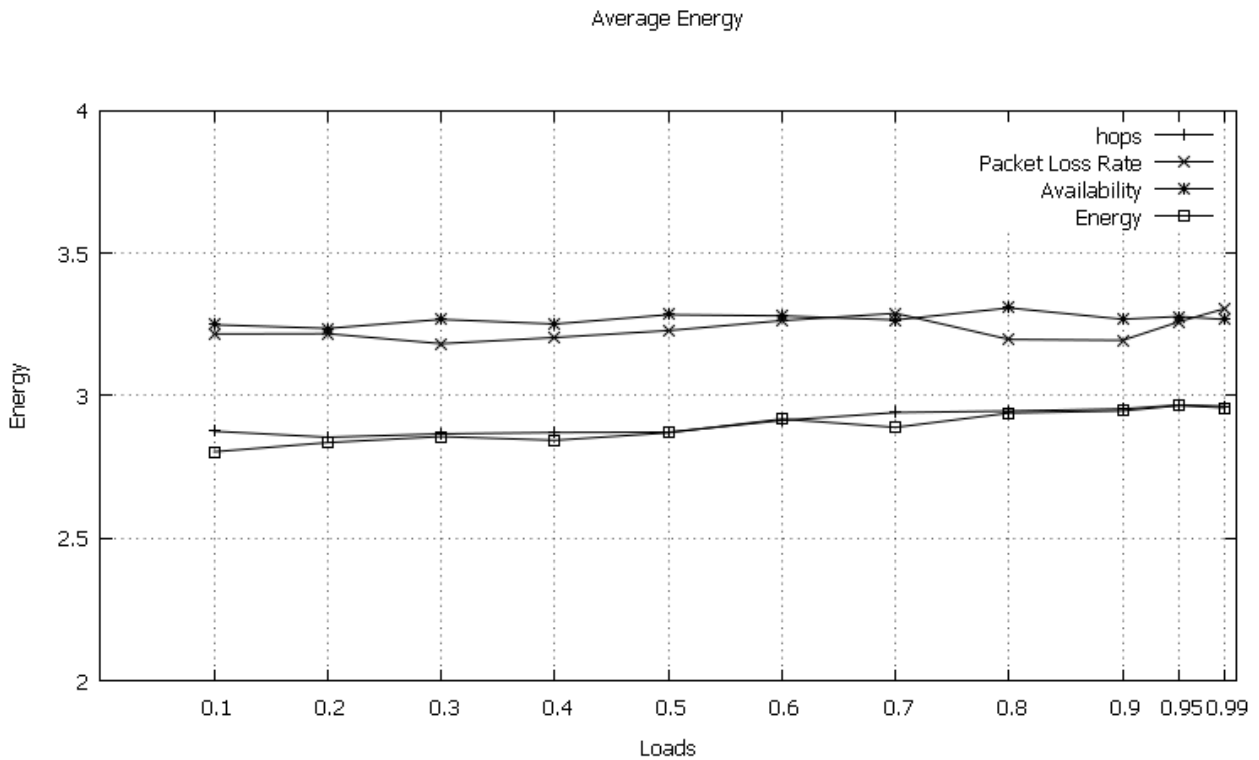


Figure 49. Average Energy

Considered two different paths (Figure 50, and 51). The first path consists of three nodes, 1,3,4. And the second path of the nodes 1,2, 5 and 4.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

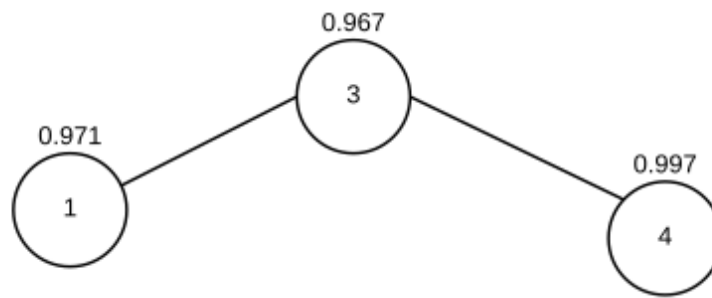


Figure 50. Energy Path 1

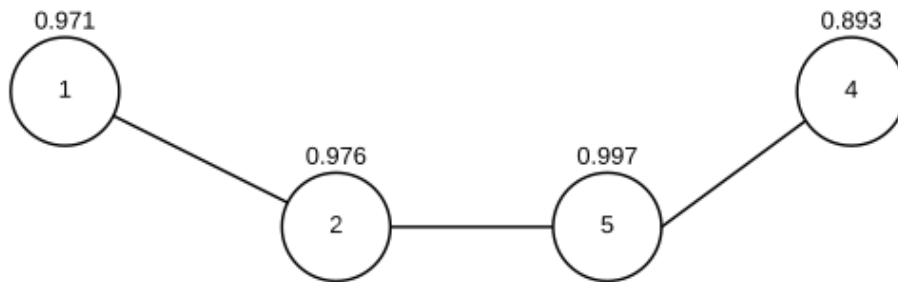


Figure 51. Energy Path 2

In the case of scenario 1, the path selected will be the first one, because the number of hops between the origin node and destination is the smaller.

For the scenario 4, the accumulative value of the energy for both paths is the following:

- **Path1:** 2.935
- **Path 2:** 3,837

So, the selected path will be the same as for scenario 1. This happens, because while trying to optimize the value of the energy, the hops in the path selected will also be optimized. Although this is only a specific example, it explained why the values of the energy average are similar.

Lastly, for the scenario 2(Packet Loss Rate) and scenario 3(Availability) the value of the energy consumption is the highest. This is explained, like the previous cases, because for both scenarios, energy is not the priority, so “Paths-Algebra” will map the path no based on energy, so consequently the energy consumption could be higher.

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

Loads	Energy
0.1	2.8752938
0.2	2.854264
0.3	2.8661631
0.4	2.8707414
0.5	2.8719321
0.6	2.9133867
0.7	2.9409162
0.8	2.9463025
0.9	2.9534533
0.95	2.9675486
0.99	2.9629701

Table 26. Energy vs Loads. Scenario 1

V.1.5. Cost/Revenue

For this evaluation parameter the following formula is used in order to calculate the average of the C/R of each scenario. Is important to remember that each of the fours scenarios are going to be simulated with loads from 0.1 to 0.99. Each load will have 20 runs of the simulation. The value of the C/R is generated in ALEVIN for each run. So the following formula needs to be use in order to calculate the average of Cost/Revenue for each loads.

$$\frac{c}{R} = \frac{\left(\sum_1^N \left(\frac{C}{R}\right)_N\right)}{20} \quad (10)$$

- C/R=Total C/R of the load
- N=Number of runs per load

After using the equation (10) for each of the fours scenario. A Graph showing the value for the C/R in function of the load is done. Table 27 shows an example of one scenario. To see the remaining tables for each scenario refer to the annex B.

Using the previous table, the graph in figure 52 was generated. This graph shows that for the scenario 1(hops) the value of the Cost/Revenue is the smallest. This can be understood as the baseline strategies (hops) will tried to map the VNR's using the shortest available path. If the paths are shorter, following the formula explained in chapter III, the cost will be smaller.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Loads	Cost /Revenue
0.1	1.751453573
0.2	1.792836031
0.3	1.821100723
0.4	1.852392129
0.5	1.864113504
0.6	1.920037621
0.7	1.951711116
0.8	1.930003333
0.9	1.925830556
0.95	1.929358291
0.99	1.937100977

Table 27. C/R vs Loads. Scenario 1
Cost-Revenue

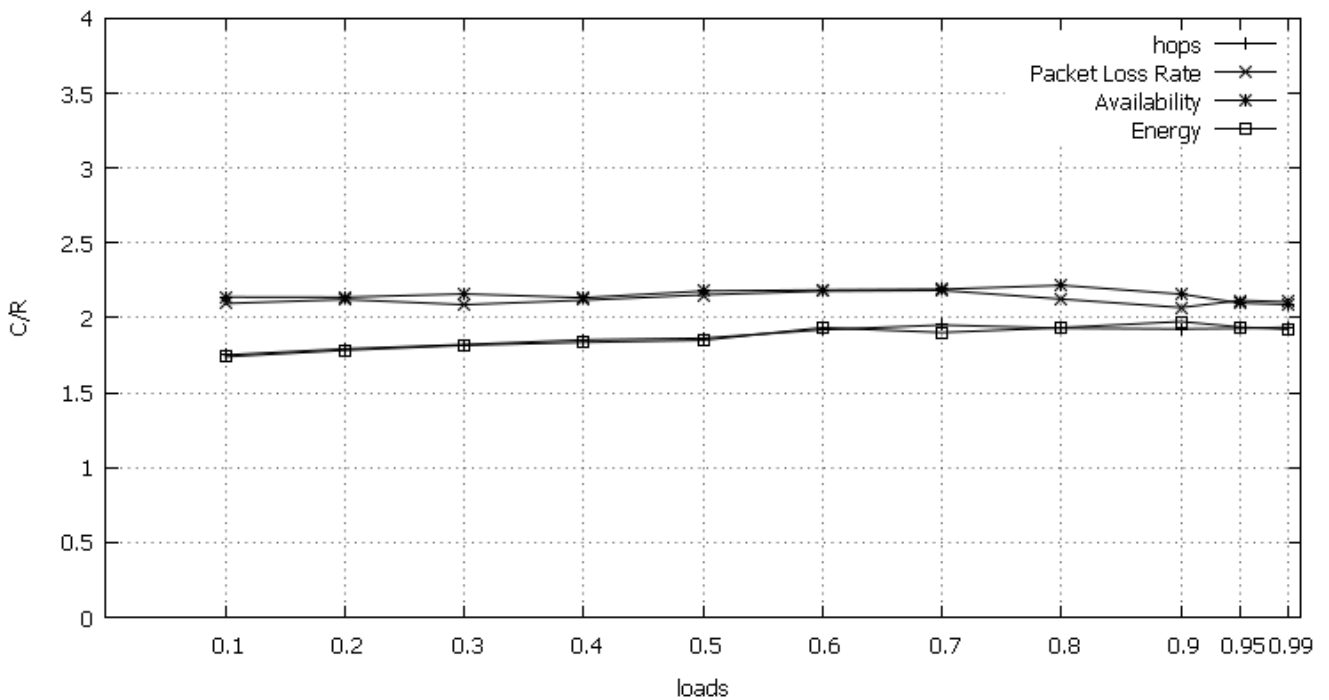


Figure 52. Cost/Revenue

This is also the case for the second scenario, given that the “Paths Algebra” tried to find the paths with the less amount of energy, this could also means that the paths are shorter. When the load increases, in order to satisfy the VNRs, longer paths have to be used and the cost will increase, so for the case of energy metric and hops there will be a slowly increase of the C/R.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

On the other hand when availability and PLR has the highest priority, the cost is bigger, this means that the paths chosen are larger. This also can be seen, looking at figure 47. That graph shows that for scenarios 2 and 3 the average of hops is larger. That is because in the case when both metrics has the priority, they do not try to find the shorter path, but the paths that have the less packet loss rate or the bigger availability.

Furthermore these paths chosen by “Paths Algebra” for both metric can be shorter or larger, so the C/R could remain almost the same for all the loads.

Another thing that can be extracted from this figure is that when the load are higher that 0.9, the value of the C/R for the four simulations converge. This can be explained that when the loads are bigger, the strategies will try to choose whichever paths it can, not matter what the metric priority is.

V.1.6. Percentage Accepted VNR

The same procedure that it was use in the Cost/Revenue was follows to calculate the average value of the Percentage Accepted VNR. So, the following formula was use.

$$PAV = \frac{(\sum_{i=0}^N PAV_i)}{20} \quad (11)$$

- PAV: Percentage Accepted VNR
- N: Number of runs

Once calculated, tables were created for each scenario like the previous case. Table 28 is an example of the Percentage Accepted VNR for only one scenario.

Loads	Percentage Accepted VNR
0.1	85
0.2	60.5
0.3	42.5
0.4	30
0.5	23.5
0.6	17.5
0.7	14
0.8	11
0.9	8.5
0.95	7
0.99	6.5

Table 28. Percentage Accepted VNR vs Loads. Scenario 1

In figure 53 the Percentage of accepted VNR is show for the four simulations. As it can be seen in for the fours scenarios the Percentage of accepted VNR are the same when the loads are the smaller, and as the load increases the value decreases. This happens because if we conspired that a SN offers an amount of resources to be used. When the load is smaller this means that are enough resources to absorb the demand and all VNRs are successfully mapped. Increasing the load means that each time the VNR’s are demanding more resources. As the “Paths Algebra” mapped the first VNR, the SN will have fewer resources to offer, so the excess load cannot be mapped and the VNR will be dropped and the percentage of accepted VNR will be smaller.

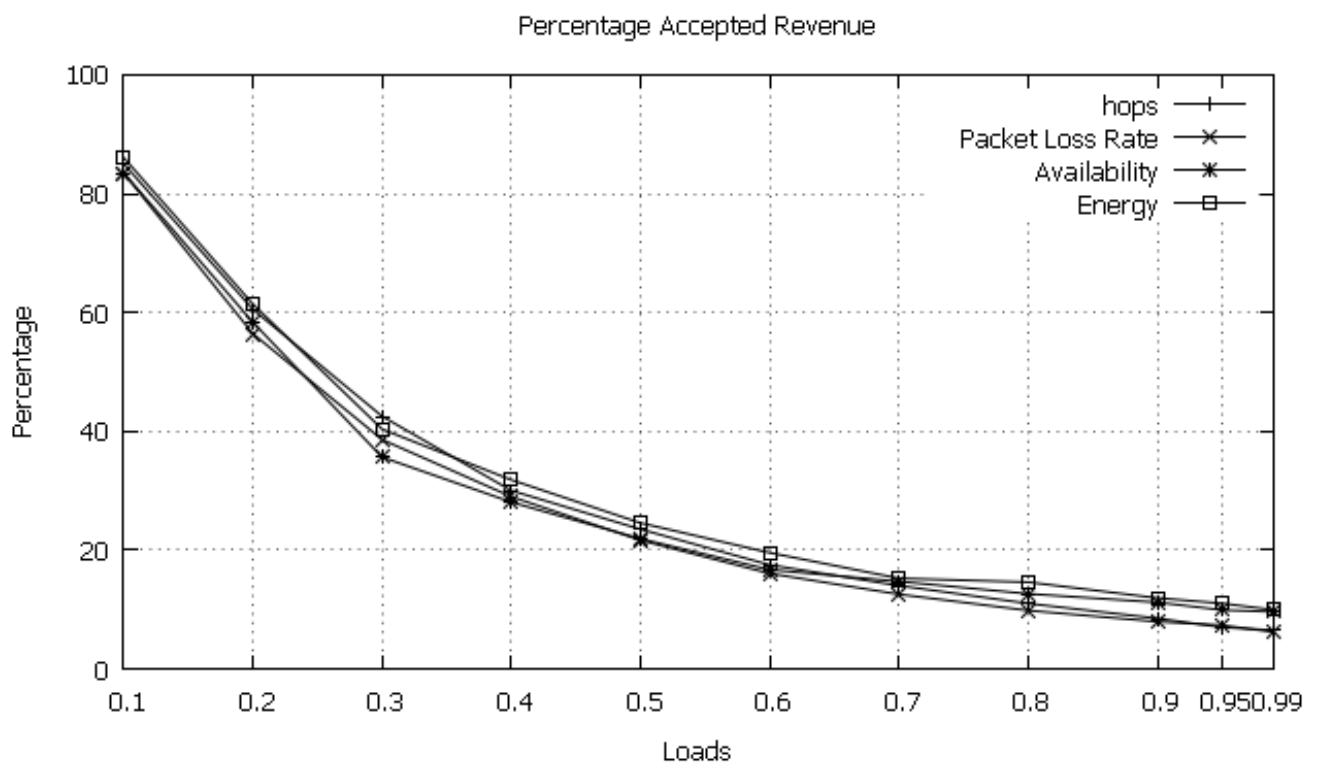


Figure 53. Percentage Accepted VNR.

V.1.7. Ratio Mapped Revenue

As the same calculation of the Cost/Revenue and the Percentage accepted VNR was done for the ratio mapped revenue and the value is proportional, the ratio-mapped revenue has the same behavior as the percentage of accepted VNRs (figure 54)

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

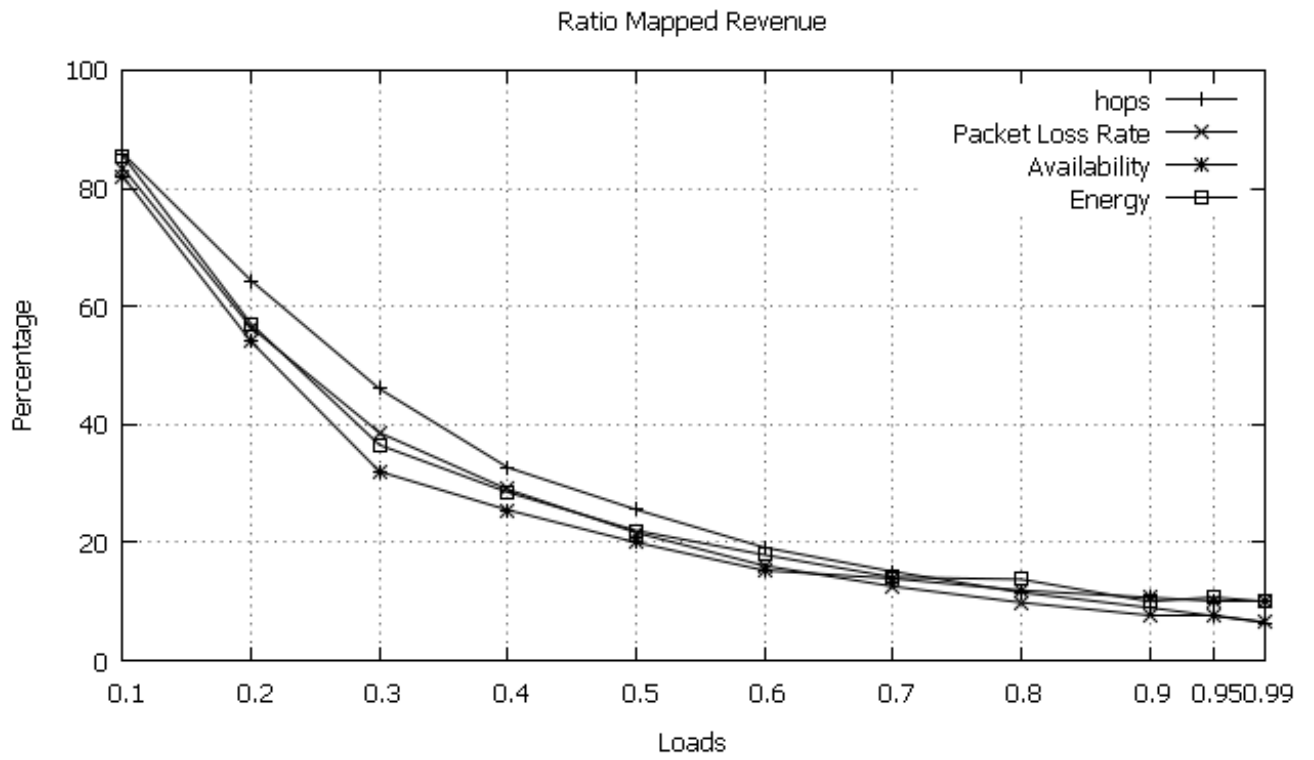


Figure 54. Ratio Mapped Revenue

Chapter VI

Conclusion and Recommendation

In this chapter conclusions are derived from those objectives that were previously established and from the results that were obtained in this project. Also, some recommendations are given for future studies and development of similar strategies.

VI.1. Conclusion

IaaS technology can change the way that Internet work nowadays, with the decoupled of the traditional Internet services into two entities, the Infrastructure Provider and the Service Provider. The main objective of the Service Provider is to perform the optimal allocation of the services demanded by the users.

Another service that could help the development of Internet services is Network as a Service. In NaaS paradigm, networking resources such as transmission bandwidth, nodes CPU and control functions are demanded by users as services. NaaS also enables abstraction of the network systems into network services; hence by doing this it can provide a flexible collaboration and facilitate end-to-end QoS provision in the future Internet

Network virtualization can be an enabler that could provide end-to-end QoS guaranteed to IaaS. Nevertheless one of the challenges of network virtualization will be the efficient allocation of virtual network elements on top of SN elements, this problem is commonly known as the virtual network embedding/mapping (VNE) problem.

The objective of the Virtual Network Embedding/mapping is to allocate or mapped the different links and Virtual Network into the SN optimizing the resources of the physical network.

For this master thesis the “Paths Algebra” mathematical tools is use in order to analyze if the implementation of different optimization criteria can solve the virtual link mapping stage of the Virtual Network embedding problem.

The proposed solution was to include three new metrics into the “Paths

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Algebra” framework. Previous studied only focus in 3 physical constraints, the hops in the path of the SN, the bandwidth resource and the CPU of each node.

In the case of this work the metrics included were: Packet Loss Ratio, Availability and the energy consumption. Then, four scenarios were compared with different metrics priority. To study the results obtained for each scenario the following parameters were used: Average of the availability, packet loss ratio, energy and hops metric. Also the VNR acceptance ratio, Mapped revenue ratio and Cost/Revenue (C/R) relationship.

Because the ALEVIN environment requires enormous amount of computational resources and because the increasing complexities for big network, the parameters were limited. For instance the length of the paths to be considered. By doing this, it provides and control over the number of operations performed by the algorithm and enable the possibility to work with large networks.

Once the metrics were included, the simulation begins, with loads from 0.1 to 0.99. These increases of loads represent the amount of demands that the VNR requested when the mapping is done. This increase was done for every VNR in the simulation. The idea of using these intervals was that the extreme cases were investigated, to see how “Paths Algebra” behaves with the new metrics, when the VNR requested a big amount of resources or in the other hand when they did not request a lot of resources.

The metric priority for each scenario was the following: for scenario 1 Hops were the highest priority, for the second scenario the Packet Loss Ratio, the third scenario the availability and the last scenario the energy consumption of the nodes. Because the computational time was exponential, the amount of SN uses was 20 nodes, and the numbers of VNR was 10 and 10 nodes per VNR. However the time required for having all the result of the metric was approximately two months.

The first set of results showed that the average of the metrics are consequence of each scenario priority. So for instance for the average of PLR, the smallest value was for the second scenario, which it has the PLR as the metric with the highest priority, so consequently the paths chosen were those who had the lowest PLR. In the other hand

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

for the remaining scenario that had different priority the result were worst. This is the same case for availability, which value was the highest for scenario 3 and also for the hops and energy which value were the lowest for scenario 1 and 3 respectively.

For the case of the C/R the best value occur for the first and fourth scenario. This happens because for the metric priority in scenario 1 and 4, “Paths Algebra” will tried to map the VNR using the shortest available path. If the paths are shorter, then the cost will be smaller and consequently the C/R will also be smaller.

On the other hand the worst values are for the second and third scenario, that it when availability and PLR has the priority. This happens because the paths chosen are larger.

Lastly for the graph regarding the Percentage of Accepted VNR and the ratio mapped revenue, the values are the same for the fours scenario when the loads are small, but as the value decreases the amount of accepted VNR also decreases. This happens because if we increase the demands of resources of the VNR, the SN won't have enough resources to accommodate all the VNR, so, some will be drooped and the percentage of accepted VNR will be smaller.

Finally after analyzing the different result for the four scenarios, it shows that each mapping decision was based on the metric that were prioritize, so the allocation of resources were based on the ordering of the metric set by the users, so it can be concluded that “Paths Algebra” is a powerful and flexible tool, because it provides several ways to include different metric with priorities and test Network with different parameters of nodes, loads or amount of VNRs.

With these results, it shows that “Paths Algebra” can perform a virtual link mapping performed in a multi- constraint basis, that is, virtual links can be mapped to substrate paths that are characterized by constraint .For instance, if the application that will run in a VN demands low energy consumption and also low packet loss rate (PLR), our approach is able to rapidly provide all the SN paths ordered in first place by the energy and then by PLR, so that, the virtual link mapping can choose the best suited path for each virtual link.

Also, it can be concluded that “Paths Algebra” does not impose any restriction

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

in the amount of metrics use. For each scenario simulated in this master Thesis, “Paths Algebra” was able mapped the best paths or route based on the priority set by the user. So linear and non-linear metrics can be used together; any combination of metrics can also be employed.

In addition, it shows that “Paths Algebra” does not mapped the VNR based only on the Cost or Revenue, but also on the availability of the network, PLR o the energy consumption of the nodes. Based on the result, for instance if the strategies is a business one, the scenario chosen will be the number 1 because it has the lowest C/R. In the other hand if the strategy is to map the path that has the lower losses, then the scenario has to be the second one. So in conclusion, “Paths Algebra” not only offers the best solution, but the entire eligible path based on different priorities. On the IaaS paradigm it can provide the SP with a powerful tool to offer more services to user with numerous constraint. Nevertheless under heavy loaded scenario “Paths Algebra” can not prioritize the different target, so it would chose any path that is available without considering the metrics ordering, and the C/R of each scenario will converged.

Finally the main objective of implementing new metric and criteria to solve the virtual link mapping stage of the VNE was achieve and the behavior of the “Paths Algebra” under multi-constraint scenario was study and analyzed.

VI.2. Recommendation

Here are some recommendations that emerge from the previous conclusions:

To achieve a more accurate result and to analyze how “Paths Algebra” behave under condition when it have multi-constraint, the recommendation is to include even more linear and non-linear metric not considered for this master Thesis, like for instance Delay or jitter.

For future work a backtracking strategy is also recommended. The idea behind this method is that when a bad mapping decision is detected, a back-track to the previous valid mapping decision is made, avoiding a costly re-map. This will optimize the amount of Virtual Network attended and also will help in allocating the resources in an effective way.

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Another recommendation is the inclusion of an energy model in which the path chosen can be done based on the equipment or nodes that are not used, this means that are turned-off. With this solution the energy consumption could be lower and the model will be more closed to the reality.

Another investigation for future development is the creation of a QoS function to include it in the “Paths Algebra” based on different linear or linear metrics

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

REFERENCES

- [1] Kim-Khoa Nguyen; Cheriet, M.; Lemay, M., "Enabling infrastructure as a service (IaaS) on IP networks: from distributed to virtualized control plane," *Communications Magazine, IEEE* , vol.51, no.1, pp.136,144, January 2013.
- [2] YAZICI, Volkan; SUNAY, M. Oguz; ERCAN, Ali O. Controlling a Software-Defined Network via Distributed Controllers. *arXiv preprint arXiv:1401.7651*, 2014.
- [3] J. F. Botero, X. Hesselbach, M. Molina, J.R. Amazonas, A novel paths algebra-based strategy to flexibly solve the link mapping stage of VNE problems, January 2013
- [4] W. P. Herman, Formulação algébrica para a modelagem de algoritmos de roteamento multi-restritivo hop-by-hop, Phd thesis, Escola Politécnica da USP, São Paulo (2008).
- [5] Jian, Zhang, Li Xiaoyong, and Guan Haibing. "The optimization of Xen network virtualization." *Computer Science and Software Engineering, 2008 International Conference on*. Vol. 3. IEEE, 2008.
- [6] Kusnetzky, Dan. *Virtualization: A Manager's Guide*. " O'Reilly Media, Inc.", 2011.
- [7] Rhoton, John. *Cloud computing explained*. Vol. 10. Breinigsville, PA USA: Recursive Press, 2010.
- [8]. Pages, A.; Perelló, J.; Spadaro, S., "Virtual network embedding in optical infrastructures," *Transparent Optical Networks (ICTON), 2012 14th International Conference on*, vol., no., pp.1,4, 2-5 July 2012
- [9] Jorge Carapinha and Javier Jimenez. 2009. Network virtualization: a view from the bottom. In *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures (VISA '09)*. ACM, New York, NY, USA, 73-80. DOI=10.1145/1592648.1592660 <http://doi.acm.org/10.1145/1592648.1592660>

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

[10] Azodolmolky, Siamak, Philipp Wieder, and Ramin Yahyapour. "Cloud computing networking: challenges and opportunities for innovations." *Communications Magazine, IEEE* 51.7 (2013).

[11] Haider, Aun, Richard Potter, and Akihiro Nakao. "Challenges in resource allocation in network virtualization." *20th ITC Specialist Seminar*. Vol. 18. 2009.

[12] Chowdhury, N.M.M.K.; Boutaba, R., "Network virtualization: state of the art and research challenges," *Communications Magazine, IEEE* , vol.47, no.7, pp.20,26, July 2009 doi: 10.1109/MCOM.2009.5183468

[13] Fischer, A; Botero, J.F.; Till Beck, M.; de Meer, H.; Hesselbach, X., "Virtual Network Embedding: A Survey," *Communications Surveys & Tutorials, IEEE* , vol.15, no.4, pp.1888,1906, Fourth Quarter 2013 doi: 10.1109/SURV.2013.013013.00155

[14] Qiang Duan, "Network-as-a-Service in Software-Defined Networks for end-to-end QoS provisioning," *Wireless and Optical Communication Conference (WOCC), 2014 23rd* , vol., no., pp.1,5, 9-10 May 2014 doi: 10.1109/WOCC.2014.6839919

[15] J. Inführ and G. R. Raidl, "Introducing the virtual network mapping problem with delay, routing and location constraints," in *Proceedings of the 5th international conference on Network optimization*, ser. INOC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 105–117. Pages,

[16] Bhardwaj, Sushil, Leena Jain, and Sandeep Jain. "Cloud computing: A study of infrastructure as a service (IAAS)." *International Journal of engineering and information Technology* 2.1 (2010): 60-63.

[17] Ali, Mufajjul. "Green cloud on the horizon." In *Cloud Computing*, pp. 451-459. Springer Berlin Heidelberg, 2009.

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

[18] A; Perelló, J.; Spadaro, S., "Virtual network embedding in optical infrastructures," *Transparent Optical Networks (ICTON), 2012 14th International Conference on*, vol., no., pp.1,4, 2-5 July 2012
doi: 10.1109/ICTON.2012.6253704

[19] Y. Zhu, M. Ammar, Algorithms for assigning substrate network resources to virtual network components, in: Proc. IEEE INFOCOM, 2006, pp. 2812–2823.

[20] J. Lu, J. Turner, Efficient mapping of virtual networks onto a shared substrate, Washington University in St. Louis, Tech. Rep.

[21] M. Yu, Y. Yi, J. Rexford, M. Chiang, Rethinking virtual network embedding: Substrate support for path splitting and migration, ACM SIGCOMM CCR 38 (2008) 17–29.

[22] N. Chowdhury, M. Rahman, R. Boutaba, Virtual network embedding with coordinated node and link mapping, in: INFOCOM 2009, IEEE, Brazil, 2009.

[23] J. Lischka, H. Karl, A virtual network mapping algorithm based on subgraph isomorphism detection, in: VISA '09: Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures, Spain, 2009.

[24] J. F. Botero, X. Hesselbach, A. Fischer, H. De Meer, Optimal mapping of virtual networks with hidden hops, *Telecommunication Systems* 52 (2013) 1–10, the original publication is available at www.springerlink.com (since March 2011).

[25] I. Houidi, W. Louati, W. B. Ameer, D. Zeglache, Virtual network provisioning across multiple substrate networks, *Computer Networks* 55 (2011) 1011 – 1023, special issue on Architectures and Protocols for the Future Internet.

[26] M. Chowdhury, F. Samuel, R. Boutaba, Polyvine: policy-based virtual network embedding across multiple domains, in: Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures, VISA '10, India, 2010.

[27] Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi, A. Sprintson, Algorithms for

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

computing qos paths with restoration, *Networking, IEEE/ACM Transactions on* 13
(2005) 648–661.

[28] N. Fujita, A. Iwata, Adaptive and efficient multiple path pre-computation for qos routing protocols, in: *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE, USA, 2001.*

[29] F. Kuipers, P. Van Mieghem, T. Korkmaz, M. Krunz, An overview of constraint-based path selection algorithms for qos routing, *Communications Magazine, IEEE* 40 (2002) 50–55.

[30] L. Shen, M. Xu, K. Xu, Y. Cui, Y. Zhao, Simple quality-of-service path first protocol and modeling analysis, in: *Communications, IEEE International Conference on, China, 2004.*

[31] de Paula Herman, Walmará, and José Roberto de Almeida Amazonas. "Hop-by-hop routing convergence analysis based on paths algebra." *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007. IEEE, 2007.*

[32] Botero, Juan Felipe, Xavier Hesselbach, Andreas Fischer, and Hermann De Meer. "Optimal mapping of virtual networks with hidden hops." *Telecommunication Systems* 51, no. 4 (2012): 273-282.

[33] A. Fischer, J. F. Botero, M. Duelli, D. Schlosser, X. Hesselbach, H. De Meer, ALEVIN - a framework to develop, compare, and analyze virtual network embedding algorithms, *Electronic Communications of the EASST* 37 (2011) 1–12.

[34] VNREAL, ALEVIN – Algorithms for Embedding Virtual Networks, <http://alevin.sf.net> (May 2011).

Network Virtualization strategy based on Paths Algebra to implement the concept of
Infrastructure as a Service

List of Acronyms

IaaS: *Infrastructure as a Service*

InP: *The Infrastructure provider*

SP: *Service Provider*

VN: *Virtual network*

SN: *substrate network*

VNE: *Virtual network embedding/mapping (VNE)*

VNR: *Virtual Network Request*

QoS: *Quality of Service*

PLR: *Packet Loss Ratio (PLR)*

IT: *Information Technology (IT)*

VM: *Virtual Machines (VM)*

VPN: *Virtual Private Network*

ISP: *Internet Service Providers (ISPs)*

VNP: *Virtual network provider*

VNO: *Virtual network operator*

NaaS: *Network as a Service*

SLA: *Service Level Agreements*

NIC: *Network Interface Card*

VLAN: *Virtual Local Area Network*

SID: *Sub graph Isomorphism Detection*

FIFO: *First In First Out*

LADN: *Loop Avoidance by the Destination*

ALEVIN: *Algorithms for Embedding of Virtual Networks*

Annexes A

Source Code

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

In this section will be presented the main program source code used to include the different metric into the LADN and ALEVIN simulation environment

The source code of the main program is shown next, , which was programmed in MATLAB.

- **First scrip used**

```
% Atendimento de Requisitos de Rede Virtual Junho script 3
clear all
echo off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Matriz de Substrato
%
%   Matriz T

load '/home/jfb/PathsAlgebra/Files/substrateNetwork.dat'

T=substrateNetwork;
M{1}=T;

% Metrica 1 - BW

load '/home/jfb/PathsAlgebra/Files/substrateMetric1.dat'

maximum=0;

% Metrica 2 - CPU

load '/home/jfb/PathsAlgebra/Files/substrateMetric2.dat'

Rcpu = substrateMetric2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Cor=0; %
Eo=1; %
% quantas iteratoes no máximo foram necessarias para impor a coerencia
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rotina Principal
% main_miguel_fev3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Inicio
%
graph=1; % grafo do teste em questao
[Nn,I] = buscapath_filtro(T,graph);
%[Nn,I] = buscapath(T);
[P,w] = separapath(M,I,Nn);

%%% Leitura de virtualStart.dat

load '/home/jfb/PathsAlgebra/Files/virtualStart.dat'

AttendOrder = virtualStart(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Calculate Maximum BW of the paths
j=1;
i=1;
max=0;
```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```

maxall=0;
x=1;
for o=1:Nn
for d=1:Nn
while (j <= size(P{1,o}{1,d},2))
while (x < size(P{1,o}{1,d}{1,j},2))
x=i+1;
if (substrateMetric1(P{1,o}{1,d}{1,j}(1,i),P{1,o}{1,d}{1,j}(1,x)) > max)
max=substrateMetric1(P{1,o}{1,d}{1,j}(1,i),P{1,o}{1,d}{1,j}(1,x));
end
i=i+1;
end
maxBW{1,o}{1,d}{1,j}=max;
j=j+1;
i=1;
x=1;
max=0;

end
j=1;
end
end
j=1;
for o=1:Nn
for d=1:Nn
while (j <= size(P{1,o}{1,d},2))
if (maxBW{1,o}{1,d}{1,j} > maxall)
maxall=maxBW{1,o}{1,d}{1,j};
end
j=j+1;
end
maxBWall{1,o}{1,d}=maxall;
maxall=0;
j=1;
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Creation of the PLR(packet loss rate)

for i=1:size(T,1)
for j=1:size(T,2)

if T(i,j)==1
a=0;
b=2;
PLR(i,j)=(a + (b-a)*rand)/100;
else
PLR(i,j)=0;
end

end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%Creation of the Avaibility

for i=1:size(T,1)
for j=1:size(T,2)

if T(i,j)==1
a=95;
b=100;
Aval(i,j)=(a + (b-a)*rand)/100;

```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```

else
Aval(i,j)=0;
end

end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Creation of the metric energy%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sum=0;

for i=1:Nn
for j=1:Nn

sum=substrateMetric1(i,j)+sum;
end

bandwidth1(1,i)=sum;
sum=0;
end
sum=0;
for j=1:Nn
for i=1:Nn

sum=substrateMetric1(i,j)+sum;
end

bandwidth2(1,j)=sum;
sum=0;
end

for j=1:Nn

bandwidthTotal(1,j)=bandwidth1(1,j)+bandwidth2(1,j);
end

for j=1:Nn

if ( bandwidthTotal(1,j)>maximum)
maximum=bandwidthTotal(1,j);
end
end

B=0.5/maximum;

for j=1:Nn
energy(1,j)=0.5+B*bandwidthTotal(1,j);
end

Renergy=energy2matrix(T,energy);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Modulo para identificatio da ordem do atendimento

%%% Comentarios de Maio de 2011

%%% AttendOrder=readAttendOrder2;

%%% Comentarios de Maio de 2011

if AttendOrder == 1      % Most Consuming First

AttendOrder = 'MCF';

elseif AttendOrder == 2 % Least Consuming First

AttendOrder = 'LCF';

elseif AttendOrder ==3  % FIFO

AttendOrder = 'FIFO';

elseif AttendOrder ==4  % Most Consuming First (USP Proposed)

AttendOrder = 'MCFUSP';

```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```
else
AttendOrder = 'LCFUSP'; % Least Consuming First (USP Proposed)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Modulo de Definitio de Metricas e Prioridades
%
% Leitura das Metricas da Matriz de Substrato

N{1}=T;
N{2}=substrateMetric1;
N{3}=cpu2matrix(T,Rcpu);
N{4}=1-PLR;
N{5}=Aval;
N{6}=energy2matrix(T,energy);
N{7}=N{2}+N{3}+N{4}+N{5}+N{6}; % metrica da soma de banda e cpu

% N{1} - hop
% N{2} - bw
% N{3} - cpu
% N{4} - PLR
% N{5} - Avalibility
% N{6} - bw+cpu+PLR+Avalibility
% N{7} - energy consumption
% N{8} - PLR (se tiver)

% G{1} - sintese e ordenacao de hop
% G{2} - sintese e ordenacao de bw
% G{3} - sintese e ordenacao de cpu
% G{4} - sintese e ordenacao de PLR
% G{5} - sintese e ordenacao de Avalibility
% G{6} - sintese e ordenacao de bw Total

% wword{1} - word de hop
% wword{2} - word de bw
% ...

[G wword]=readSubstrateMetricsSeptember;

% readMetricsOrder
%
% 1- bw, cpu
% 2- cpu, bw
% 3- hop, bw, cpu
% 4- hop, cpu, bw
% 5- plr,hop, bw, cpu
% 6- delay, cpu, bw
% 7- availability, bw, cpu
% 8- availability, cpu, bw
% 9- do your choice: avalability, delay, jitter, hop, bw, cpu
% 10 - bw+cpu,hop
% 11 - hop,bw+cpu
% 12 - hop, bw Total

% no momento 1 a 4

% Comentarios de Maio de 2011 - M0=readMetricsOrder2;

M0= virtualStart(2);

if M0 == 1
[M F word iType]=metricsOrder1(N,G,wword);

elseif M0 == 2
[M F word iType]=metricsOrder2(N,G,wword);

elseif M0 == 3
```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```

[M F word iType]=metricsOrder3(N,G,wword);

elseif MO == 4
[M F word iType]=metricsOrder4(N,G,wword);

elseif MO == 5
[M F word iType]=metricsOrder5(N,G,wword);
elseif MO == 6
[M F word iType]=metricsOrder6(N,G,wword);
elseif MO == 7
[M F word iType]=metricsOrder7(N,G,wword);

elseif MO == 11
[M F word iType]=metricsOrder11(N,G,wword);

elseif MO == 12
[M F word iType]=metricsOrder12(N,G,wword);

else
%
% [M F word iType]=metricsOrder5(N,G,wword);
fprintf('\n0thers is not Available\n');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
if MO==7
[w,z] = updateNewMetrics2(P,M,Nn);
else
[w] = updateNewMetrics(P,M,Nn);
end

orderNewPaths1;

% Ordenacao dos Caminhos

for i=1:virtualStart(3)

% leitura das VN's
arq1 = strcat('/home/jfb/PathsAlgebra/Files/', 'virtualRequest_',int2str(i));
VN{i}=load(arq1);

% leitura da Metrica 1 - MBW
arq2 = strcat('/home/jfb/PathsAlgebra/Files/', 'virtualMetric1_',int2str(i));
Mbw{i}=load(arq2);

% leitura da Metrica 2 - Vcpu
arq3 = strcat('/home/jfb/PathsAlgebra/Files/', 'virtualMetric2_',int2str(i));
Vcpu{i}=load(arq3);

% leitura da Hidden Matrix
arq4 = strcat('/home/jfb/PathsAlgebra/Files/', 'virtualHidden_',int2str(i));
HiddenMatrix{i}=load(arq4);

end

% inicializatio de variavel
Attend=[];
RV={};

R0=zeros(1,size(VN,2)); % vetor de ordenacao dos caminhos

fid=fopen('/home/jfb/PathsAlgebra/Files/requestOrder','wt+'); % arquivo ordenado

%

if strcmp(AttendOrder,'FIFO')

for i=1:size(VN,2)

R0(i)=i;

```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```
fprintf(fid,'%d\n', R0(i));
end
elseif strcmp(AttendOrder,'MCFUSP')
request = computeTotalRequests2May(Mbw,Vcpu,N{2},Rcpu);
request = bubblesort(request);
k=1;
for j=size(request,2):-1:1
i=request{j}(2);
R0(k)=i;
fprintf(fid,'%d\n', R0(k));
k=k+1;
end
elseif strcmp (AttendOrder,'LCFUSP')
request = computeTotalRequests2May(Mbw,Vcpu,N{2},Rcpu);
request = bubblesort(request);
for j=1:size(request,2)
i=request{j}(2);
R0(j)=i;
fprintf(fid,'%d\n', R0(j));
end
end
elseif strcmp(AttendOrder,'MCF')
request = computeTotalRequests(Mbw,Vcpu);
request = bubblesort(request);
k=1;
for j=size(request,2):-1:1
i=request{j}(2);
R0(k)=i;
fprintf(fid,'%d\n', R0(k));
k=k+1;
end
end
else
request = computeTotalRequests(Mbw,Vcpu);
request = bubblesort(request);
for j=1:size(request,2)
i=request{j}(2);
R0(j)=i;
fprintf(fid,'%d\n', R0(j));
end
end
end
fclose(fid);
% Modulo Principal de Atendimento de Requisitos
%% %% Vou ter que editar esse modulo -----> main_attendVNsMay2;
%% %%
%% %%
save '/home/jfb/PathsAlgebra/Files/searchPath.mat'
```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Resultados:
% Po{m}{o}{d}{:}
% So{m}{o}{d}{:} array das sinteses correspondentes do Po
% Pfo {1}{o}{d}{:} array de caminhos ordenados
% Sfo {m}{o}{d}{:} array das sinteses correspondentes do Pfo
%
exit
```

- **Second script**

```
%%% Script 4D - Atendimento Online

clear all

load '/home/jfb/PathsAlgebra/Files/searchPath.mat'

% main_attendVNsMay2

% Leitura de Requisitos Virtuais - VN requests
%
%

% Inicializacao de Matrices

VN={};
Vcpu={};
Mbw={};
VN2T={};
path={};
PLRpath=[];
sum=0;hop=0;
hoptotal=0;
sumAval=0;
energySum=0;
energySum2=0;
sumEnergy=0;

arq32=strcat('/home/jfb/PathsAlgebra/Files/', 'Counter');
count2=load(arq32);
arq30=strcat('/home/jfb/PathsAlgebra/Files/', 'MinPLRTotal');
minPLR=load(arq30);
arq31=strcat('/home/jfb/PathsAlgebra/Files/', 'MaxPLRTotal');
maxPLR=load(arq31);
arq33=strcat('/home/jfb/PathsAlgebra/Files/', 'Average');
sumTotal=load(arq33);
arq40=strcat('/home/jfb/PathsAlgebra/Files/', 'HopAverage');
HopAverage=load(arq40);
arq41 = strcat('/home/jfb/PathsAlgebra/Files/', 'CounterTotal');
count=load(arq41);
arq43=strcat('/home/jfb/PathsAlgebra/Files/', 'AverageAval');
sumTotalAval=load(arq43);
arq80=strcat('/home/jfb/PathsAlgebra/Files/', 'MinAval');
minAval=load(arq80);
arq81=strcat('/home/jfb/PathsAlgebra/Files/', 'MaxAval');
maxAval=load(arq81);
arq500=strcat('/home/jfb/PathsAlgebra/Files/', 'MaxEnergy');
maxEnergy=load(arq500);
arq501=strcat('/home/jfb/PathsAlgebra/Files/', 'minEnergy');
minEnergy=load(arq501);
arq502=strcat('/home/jfb/PathsAlgebra/Files/', 'Energy');
energyAverage=load(arq502);

% Inicializacao de Variavel

i=1;
```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```
%%% Preparacao para leitura da primeira requisicao
while i <= virtualStart(3)

    % leitura do arquivo virtualnodemapping_x

    arq6 =
strcat('/home/jfb/PathsAlgebra/Files/', 'virtualnodemapping_', int2str(R0(i)), '.dat');

    if ((exist(arq6) == 2))

        arq6 =
strcat('/home/jfb/PathsAlgebra/Files/', 'virtualnodemapping_', int2str(R0(i)), '.dat');
        node = load(arq6);
        a=node;

        if (int2str(a) == '1')

            % leitura do Mapeamento de Nous Virtual para Real
            arq5 =
strcat('/home/jfb/PathsAlgebra/Files/', 'virtual_to_Real_', int2str(R0(i)));

            if ((exist(arq5) == 2))

                % Leitura das VNs, Metricas, Hidden, Mapeamento

                % leitura das VNs
                arq1 =
strcat('/home/jfb/PathsAlgebra/Files/', 'virtualRequest_', int2str(R0(i)));
                VN{R0(i)}=load(arq1);

                % leitura da Metrica 1 - MBW
                arq2 =
strcat('/home/jfb/PathsAlgebra/Files/', 'virtualMetric1_', int2str(R0(i)));
                Mbw{R0(i)}=load(arq2);

                % leitura da Metrica 2 - Vcpu
                arq3 =
strcat('/home/jfb/PathsAlgebra/Files/', 'virtualMetric2_', int2str(R0(i)));
                Vcpu{R0(i)}=load(arq3);

                % leitura da Hidden Matrix
                arq4 =
strcat('/home/jfb/PathsAlgebra/Files/', 'virtualHidden_', int2str(R0(i)));
                HiddenMatrix{R0(i)}=load(arq4);

                % virtual to Real - node mapping
                VN2T{R0(i)}=load(arq5);

                % lembrete - arq1=strcat('discoverTimemarco_teste_', int2str(numero));
                % VNrequest7
                % inicializatio de variavel
                Attend=[];
                RV={};

                % Rotina de Atendimento de Requisitos

                [M,Rcpu,Attend,RV] =
attendVNsHiddenSeptember2(Pfo,M, iType,Mbw{R0(i)},Rcpu,Vcpu{R0(i)},VN{R0(i)},VN2T{R0(i)}
,HiddenMatrix{R0(i)},R0(i),Attend,T,RV);

                % Atualizacao de Indices de Metricas e Reordenacao

                % if i < virtualStart (3)
                if M0==7
                    [w,z] = updateNewMetrics2(P,M,Nn);
                else
                    [w] = updateNewMetrics(P,M,Nn);
                end
            end
        end
    end
end
```


Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```

        orderNewPaths1;

        % end

        % Atualizacao do contador de requisicoes
        i=i+1;
    end

    else
        % Esperar pela proxima requisicao
        % Atualizacao do contador de requisicoes
        i=i+1;
    end

end

end

i=1;
c=1;
while i <= virtualStart(3)
    %-----nuevo-----
    -----

    arq8=strcat('/home/jfb/PathsAlgebra/Files/', 'virtualSuccess_', int2str(R0(i)));
        bool=load(arq8);

        if (bool==1)
            for a=1:size(VN{1,i},1)
                for b=1:size(VN{1,i},1)
                    if (VN{1,R0(i)}(a,b)==1)

arq10=strcat('/home/jfb/PathsAlgebra/Files/', 'virtualAttend_', int2str(R0(i)), '_', int2str(a), '_', int2str(b));
                            path{1,i}{c}=load(arq10);
                            c=c+1;
                            end
                            end
                            end

                            end

                            c=1;
                            i=i+1;
end

i=1;
c=1;
PLRmult=1;
avalMult=1;
b=1;
min=10000000;
mincp=1000000;
BW=substrateMetric1;
CPU=substrateMetric2;

while i<= virtualStart(3)

    arq8=strcat('/home/jfb/PathsAlgebra/Files/', 'virtualSuccess_', int2str(R0(i)));
    bool=load(arq8);

    if (bool==1)
        [x sizes]=size(path{1,i});
        for a=1:sizes
            [row col]=size(path{1,i}{1,a});
            while b+1 <= col

                %-----Calcular el PLR del path-----
                -----
                PLRmult=PLRmult*(1-

```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```

PLR(path{1,i}{1,a}(1,b),path{1,i}{1,a}(1,b+1));
-----
%-----Calcular la availability del path-----
-----
avalMult=avalMult*(Aval(path{1,i}{1,a}(1,b),path{1,i}{1,a}(1,b+1)));

%cCalcular la energia del camino

energySum2=energySum2+energy(1,path{1,i}{1,a}(1,b));

%-----calcular el min BW del path elegido-----
-----

    if BW(path{1,i}{1,a}(1,b),path{1,i}{1,a}(1,b+1)) < min
        min=BW(path{1,i}{1,a}(1,b),path{1,i}{1,a}(1,b+1));
    end
%-----calcular el min CPU de los nodos
del path-----

    if CPU(1,path{1,i}{1,a}(1,b)) < mincp
        mincp=CPU(1,path{1,i}{1,a}(1,b));
    end

%-----calcular el hop-----
hop=hop+1;

    b=b+1;
        end

    PLRpath{1,i}{1,a}=PLRmult;
    Energypath{1,i}{1,a}=energySum2+energy(1,path{1,i}{1,a}(1,b));
    avalPath{1,i}{1,a}=avalMult;
    minBW{1,i}{1,a}=min;
    minCPU{1,i}{1,a}=mincp;

if (bool==1)

    if PLRpath{1,i}{1,a} < minPLR
        minPLR=PLRpath{1,i}{1,a};    %Calculate the minimum PLR
    end

    if PLRpath{1,i}{1,a} > maxPLR
        maxPLR=PLRpath{1,i}{1,a};    %Calculate the maximum PLR
    end

    if avalPath{1,i}{1,a} < minAval
        minAval=avalPath{1,i}{1,a};    %Calculate the minimum
Availability
    end

    if avalPath{1,i}{1,a} > maxAval
        maxAval=avalPath{1,i}{1,a};    %Calculate the maximum
availability
    end

    if Energypath{1,i}{1,a} > maxEnergy
        maxEnergy=Energypath{1,i}{1,a};
    end
    %Calculate the maximum energy

    if Energypath{1,i}{1,a} < minEnergy

```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```

        minEnergy=EnergyPath{1,i}{1,a};
    end

    sumEnergy=(sumEnergy+EnergyPath{1,i}{1,a}); %Calculate the sum of
energy
    sum=(sum+PLRPath{1,i}{1,a}); % Calculate the sum of the PLR
    sumAval=(sumAval+avalPath{1,i}{1,a});%Calculate the sum of
availability
    hoptotal=hoptotal+hop;
end

    %saving the minimum BW in a
    %file-----
    %arq14 =
    strcat('/home/jfb/PathsAlgebra/Files/', 'BWTotPath_', int2str(R0(i)), '_', int2str(i), '_'
, int2str(a));

        % AuxBW = minBW{1,i}{1,a};
        % save (arq14, 'AuxBW', '-ASCII');

    %Saving the minimum CPU in a file-----
    -----
    % arq15 =
    strcat('/home/jfb/PathsAlgebra/Files/', 'CPUTotPath_', int2str(R0(i)), '_', int2str(i), '_'
, int2str(a));

        %AuxCPU = minCPU{1,i}{1,a};
        %save (arq15, 'AuxBW', '-ASCII');

    PLRmult=1;
    avalMult=1;
    min=10000000;
    mincp=1000000;
    c=c+1;
    b=1;
    hop=0;
    count=count+1; %use to calculate the average
    energySum2=0;

    end
end

    i=i+1;
end
count2=count2+1;

arq60 = strcat('/home/jfb/PathsAlgebra/Files/', 'Sum_', int2str(count2));
Auxsum = sum;
save (arq60, 'Auxsum', '-ASCII');
arq60 = strcat('/home/jfb/PathsAlgebra/Files/', 'SumAval_', int2str(count2));
AuxsumAval = sumAval;
save (arq60, 'AuxsumAval', '-ASCII');
sumTotal=sum+sumTotal;
SumTotalAval=sumAval+sumTotalAval;
HopAverage=HopAverage+hoptotal;
energyAverage=energyAverage+sumEnergy;

if (count2 > 1)
    sumTotal=sumTotal/count;
    HopAverage=HopAverage/count;

```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```
SumTotalAval=SumTotalAval/count;  
energyAverage=energyAverage/count;  
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
save '/home/jfb/PathsAlgebra/Files/searchPath.mat'
```

```
arq13 = strcat('/home/jfb/PathsAlgebra/Files/', 'MinPLRTotal');
```

```
    AuxPLR = minPLR;
```

```
    save (arq13, 'AuxPLR', '-ASCII');
```

```
arq61 = strcat('/home/jfb/PathsAlgebra/Files/', 'CounterTotal');
```

```
    AuxCounterTotal = count;
```

```
    save (arq61, 'AuxCounterTotal', '-ASCII');
```

```
arq14 = strcat('/home/jfb/PathsAlgebra/Files/', 'MaxPLRTotal');
```

```
    Aux2PLR = maxPLR;
```

```
    save (arq14, 'Aux2PLR', '-ASCII');
```

```
arq15 = strcat('/home/jfb/PathsAlgebra/Files/', 'Counter');
```

```
    AuxCounter = count2;
```

```
    save (arq15, 'AuxCounter', '-ASCII');
```

```
arq16 = strcat('/home/jfb/PathsAlgebra/Files/', 'Average');
```

```
    AuxSum = sumTotal;
```

```
    save (arq16, 'AuxSum', '-ASCII');
```

```
arq50 = strcat('/home/jfb/PathsAlgebra/Files/', 'HopAverage');
```

```
    AuxHop= HopAverage;
```

```
    save(arq50, 'AuxHop', '-ASCII');
```

```
arq51 = strcat('/home/jfb/PathsAlgebra/Files/', 'AverageAval');
```

```
    AuxSumAval2=SumTotalAval;
```

```
    save(arq51, 'AuxSumAval2', '-ASCII');
```

```
arq52 = strcat('/home/jfb/PathsAlgebra/Files/', 'MinAval');
```

```
    AuxAval = minAval;
```

```
    save (arq52, 'AuxAval', '-ASCII');
```

```
arq53 = strcat('/home/jfb/PathsAlgebra/Files/', 'MaxAval');
```

```
    AuxAval2 = maxAval;
```

```
    save (arq53, 'AuxAval2', '-ASCII');
```

```
arq510 =strcat('/home/jfb/PathsAlgebra/Files/', 'minEnergy');
```

```
    AuxEnergy = minEnergy;
```

```
    save (arq510, 'AuxEnergy', '-ASCII');
```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```

arq511 = strcat('/home/jfb/PathsAlgebra/Files/', 'MaxEnergy')
        AuxEnergy2 = maxEnergy;

        save (arq511, 'AuxEnergy2', '-ASCII');

arq512 = strcat('/home/jfb/PathsAlgebra/Files/', 'Energy');

        AuxEnergy3 = energyAverage;

        save (arq512, 'AuxEnergy3', '-ASCII');

save '/home/jfb/PathsAlgebra/Files/searchPath.mat'

%-----
%-----

%%% Final do Script

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Cor;
% Estado=Eo-1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Resultados:
% Po{m}{o}{d}{:}
% So{m}{o}{d}{:} array das sinteses correspondentes do Po
% Pfo {1}{o}{d}{:} array de caminhos ordenados
% Sfo {m}{o}{d}{:} array das sinteses correspondentes do Pfo
%

exit

```

- **Metric order Script 1**

```

function [M F word iType]=metricsOrder6(N,G,wword)
%
% function [M F word iType]=metricsOrder4(N,G,wword)
%
% Ordenatio por 1) Availability 3) HOP 4) BW and 5)CPU
%
% Leitura das Metricas da Matriz de Substrato
%
%
% N{1} - hop
% N{2} - bw
% N{3} - cpu
% N{4} - PLR
% N{5} - Availability
% N{6} - reliability
% N{7} - energy consumption
%

% Inicializatio de Variaveis de Retorno

M={};
F={};
word={};
iType={};

% Atribuitio de Variaveis

```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```
% Availability
M{1}=N{5};
F{1}=G{5};
word{1}=wword{5};

% hop
M{2}=N{1};
F{2}=G{1};
word{2}=wword{1};

% bw
M{3}=N{2};
F{3}=G{2};
word{3}=wword{2};

% cpu
M{4}=N{3};
F{4}=G{3};
word{4}=wword{3};
```

```
% Artificio para o attendVNs
```

```
iType{1}=[5];
iType{2}=[1];
iType{3}=[2];
iType{4}=[3];
```

• Metric Order scrip 2

```
function [M F word iType]=metricsOrder5(N,G,wword)
%
% function [M F word iType]=metricsOrder4(N,G,wword)
%
% Ordenatio por 1) PLR 2) HOP 3) BW and 4)CPU
%
% Leitura das Metricas da Matriz de Substrato
% Conventio
%
% N{1} - hop
% N{2} - bw
% N{3} - cpu
% N{4} - PLR
% N{5} - PLR (se tiver)
% N{6} - reliability
% N{7} - energy consumption
%

% Inicializatio de Variaveis de Retorno

M={};
F={};
word={};
iType={};

% Atribuitio de Variaveis

% PLR
M{1}=N{4};
F{1}=G{4};
word{1}=wword{4};

% hop
M{2}=N{1};
F{2}=G{1};
word{2}=wword{1};

% cpu
M{4}=N{3};
F{4}=G{3};
word{4}=wword{3};
```

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

```
% bw
M{3}=N{2};
F{3}=G{2};
word{3}=wword{2};
```

```
%
```

```
iType{1}=[4];
iType{2}=[1];
iType{3}=[2];
iType{4}=[3];
```

• Metric Order script 3

```
function [M F word iType]=metricsOrder7(N,G,wword)
%
% function [M F word iType]=metricsOrder4(N,G,wword)
%
% 1) Energy 3) HOP 4) BW and 5)CPU
%
%
%
% N{1} - hop
% N{2} - bw
% N{3} - cpu
% N{4} - PLR
% N{5} - Availability
% N{6} - Energy
%

% Inicializatio de Variaveis de Retorno

M={};
F={};
word={};
iType={};

% Atribuitio de Variaveis

% Energy
M{1}=N{6};
F{1}=G{6};
word{1}=wword{6};

% hop
M{2}=N{1};
F{2}=G{1};
word{2}=wword{1};

% bw
M{3}=N{2};
F{3}=G{2};
word{3}=wword{2};

% cpu
M{4}=N{3};
F{4}=G{3};
word{4}=wword{3};

iType{1}=[4];
iType{2}=[1];
iType{3}=[2];
iType{4}=[3];
```

Annexes B

Results Tables

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

In this part all the tables from the simulation used to graph the results of the four scenarios are shown.

- **Scenario 1 (Hops, BW, CPU)**

Loads	Hops Average
0.1	1.871233
0.2	1.9318718
0.3	1.9436485
0.4	1.9691689
0.5	1.9692833
0.6	2.0123818
0.7	2.0385164
0.8	2.0434783
0.9	2.0648968
0.95	2.0648968
0.99	2.0616438

Loads	PLR Average
0.1	1.986919
0.2	2.053837
0.3	2.023622
0.4	2.079379
0.5	2.077542
0.6	2.132567
0.7	2.110575
0.8	2.165483
0.9	2.144338
0.95	2.200522
0.99	2.225807

Loads	Availability Average (%)
0.1	95.162778
0.2	95.029028
0.3	94.983103
0.4	94.881892
0.5	94.924477
0.6	94.803908
0.7	94.740129
0.8	94.64432
0.9	94.972658
0.95	94.743325
0.99	94.865986

Loads	Energy Average
0.1	2.8752938
0.2	2.854264
0.3	2.8661631
0.4	2.8707414
0.5	2.8719321
0.6	2.9133867
0.7	2.9409162
0.8	2.9463025
0.9	2.9534533
0.95	2.9675486
0.99	2.9629701

Loads	Cost Revenue
0.1	1.751453573
0.2	1.792836031
0.3	1.821100723
0.4	1.852392129
0.5	1.864113504
0.6	1.920037621
0.7	1.951711116
0.8	1.930003333
0.9	1.925830556
0.95	1.929358291
0.99	1.937100977

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Loads	Percentage Accepted VNR
0.1	85
0.2	60.5
0.3	42.5
0.4	30
0.5	23.5
0.6	17.5
0.7	14
0.8	11
0.9	8.5
0.95	7
0.99	6.5

Loads	Ratio Mapped Revenue
0.1	85.84915071
0.2	64.33407198
0.3	46.01347203
0.4	32.68840182
0.5	25.57122977
0.6	19.13467707
0.7	15.0695432
0.8	11.52086717
0.9	8.9539626
0.95	7.557196581
0.99	6.429833729

- Scenario 2 (Packet Loss Ratio, Hops, BW, CPU)

Loads	Hops Average
0.1	2.3291861
0.2	2.3292165
0.3	2.2915952
0.4	2.3148289
0.5	2.3411996
0.6	2.3755102
0.7	2.3982609
0.8	2.3050109
0.9	2.2988827
0.95	2.3654971
0.99	2.4161074

Loads	PLR Average
0.1	1.708837
0.2	1.834948
0.3	1.837508
0.4	1.893824
0.5	1.873091
0.6	1.90236
0.7	1.920965
0.8	1.835115
0.9	1.817538
0.95	1.866046
0.99	1.921151

Loads	Availability Average(%)
0.1	93.99428
0.2	94.025984
0.3	94.127806
0.4	94.039416
0.5	94.039416
0.6	93.901006
0.7	93.82329
0.8	94.039509
0.9	93.969234
0.95	93.914531
0.99	93.876225

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Loads	Energy Availability
0.1	3.216621
0.2	3.2174199
0.3	3.1826762
0.4	3.2043257
0.5	3.228544
0.6	3.2634137
0.7	3.2890322
0.8	3.1975138
0.9	3.1944702
0.95	3.2598688
0.99	3.3056464

Loads	Cost Revenue
0.1	2.09609343
0.2	2.122162498
0.3	2.087415615
0.4	2.117984443
0.5	2.153496237
0.6	2.178009637
0.7	2.183015312
0.8	2.124702232
0.9	2.068437605
0.95	2.116073654
0.99	2.106652169

Loads	Percentage Accepted VNR
0.1	83.43364568
0.2	56.39655593
0.3	38.57208009
0.4	28.99029447
0.5	21.6376753
0.6	16.04938302
0.7	12.5537823
0.8	9.819216481
0.9	7.5
0.95	7.5
0.99	6.5

Loads	Ratio Mapped Revenue
0.1	83.43364568
0.2	56.39655593
0.3	38.57208009
0.4	28.99029447
0.5	21.6376753
0.6	16.04938302
0.7	12.5537823
0.8	9.819216481
0.9	7.94499052
0.95	7.340683615
0.99	6.202389872

- Scenario 3(Availability, hops, BW, CPU)

Loads	Hops Average
0.1	2.36373
0.2	2.3494244
0.3	2.3818828
0.4	2.3644933
0.5	2.3977035
0.6	2.3916667
0.7	2.3780291
0.8	2.4215054
0.9	2.3770053
0.95	2.3875
0.99	2.3766234

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Loads	Average Availability
0.1	95.908403
0.2	95.543444
0.3	95.494536
0.4	95.524214
0.5	95.469786
0.6	95.516439
0.7	95.493361
0.8	95.418289
0.9	95.577339
0.95	95.652256
0.99	95.588032

Loads	Average PLR
0.1	2.508653
0.2	2.493174
0.3	2.509288
0.4	2.494712
0.5	2.558962
0.6	2.49084
0.7	2.465398
0.8	2.526397
0.9	2.549969
0.95	2.483604
0.99	2.457403

Loads	Energy Average
0.1	3.2490058
0.2	3.2362263
0.3	3.2676898
0.4	3.2517837
0.5	3.2842529
0.6	3.2802864
0.7	3.2659008
0.8	3.3079095
0.9	3.2682207
0.95	3.2781901
0.99	3.2673243

Loads	Cost Revenue
0.1	2.138600477
0.2	2.137374637
0.3	2.159334275
0.4	2.133900501
0.5	2.179070763
0.6	2.186038241
0.7	2.19072071
0.8	2.216865463
0.9	2.158296982
0.95	2.098291504
0.99	2.088397331

Loads	Percentage Accepted VNR
0.1	82
0.2	54
0.3	32
0.4	25.5
0.5	20
0.6	15.26315789
0.7	13.88888889
0.8	11.875
0.9	10.66666667
0.95	10
0.99	10

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Loads	Ratio Mapped Revenue
0.1	86.03076236
0.2	61.39353076
0.3	40.32752366
0.4	31.81322278
0.5	24.55555714
0.6	19.54047268
0.7	15.23540358
0.8	14.5982905
0.9	11.93025384
0.95	11.02608927
0.99	9.919784424

- Scenario 4 (Energy consumption, hops, BW, CPU)

Loads	Hops Average
0.1	1.9066333
0.2	1.9378772
0.3	1.9605263
0.4	1.944367
0.5	1.9722463
0.6	2.0212515
0.7	1.989882
0.8	2.0437262
0.9	2.0801034
0.95	2.070028
0.99	2.0701143

Loads	PLR Average
0.1	2.023504
0.2	2.051867
0.3	2.0451
0.4	2.052273
0.5	2.077719
0.6	2.150345
0.7	2.098268
0.8	2.178062
0.9	2.198995
0.95	2.182288
0.99	2.15868

Loads	Average Availability
0.1	95.047258
0.2	94.979629
0.3	94.91837
0.4	94.985197
0.5	94.986816
0.6	94.710784
0.7	94.86762
0.8	94.627959
0.9	94.579912
0.95	94.783325
0.99	94.837661

Network Virtualization strategy based on Paths Algebra to implement the concept of Infrastructure as a Service

Loads	Energy Average
0.1	2.8031472
0.2	2.8355082
0.3	2.8564588
0.4	2.8435474
0.5	2.8699272
0.6	2.9180188
0.7	2.8888794
0.8	2.9390152
0.9	2.9468559
0.95	2.9653921
0.99	2.9578611

Loads	Cost Revenue
0.1	1.739593934
0.2	1.782910451
0.3	1.816658917
0.4	1.835896606
0.5	1.850068973
0.6	1.93467927
0.7	1.901673647
0.8	1.93404587
0.9	1.974756979
0.95	1.937400325
0.99	1.923026572

Loads	Percentage Accepted VNR
0.1	85.5
0.2	57
0.3	36.5
0.4	28.5
0.5	22
0.6	17.89473684
0.7	14.11764706
0.8	13.75
0.9	10
0.95	10.71428571
0.99	10

Loads	Ratio Mapped Revenue
0.1	86.03076236
0.2	61.39353076
0.3	40.32752366
0.4	31.81322278
0.5	24.55555714
0.6	19.54047268
0.7	15.23540358
0.8	14.5982905
0.9	11.93025384
0.95	11.02608927
0.99	9.919784424