UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LEONARDO RICHTER BAYS

Efficient, Online Embedding of Secure Virtual Networks

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Luciano Paschoal Gaspary
Advisor

Porto Alegre, May 2013

CIP – CATALOGING-IN-PUBLICATION

Bays, Leonardo Richter

Efficient, Online Embedding of Secure Virtual Networks / Leonardo Richter Bays. – Porto Alegre: PPGC da UFRGS, 2013.

81 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2013. Advisor: Luciano Paschoal Gaspary.

I. Gaspary, Luciano Paschoal. II. Título.

*“It is not knowledge, but the act of learning,
not possession but the act of getting there,
which grants the greatest enjoyment.”*
— Carl Friedrich Gauss

AGRADECIMENTOS

Agradeço primeiramente à minha família, por toda a ajuda e todo o apoio que me foram dados ao longo da minha vida. Agradeço, também, aos amigos e colegas da UFRGS, em especial aos colegas de laboratório. Nossa convivência tornou o curso de Mestrado muito mais agradável, e a troca de experiências foi fundamental para o desenvolvimento desse trabalho. Por fim, agradeço aos professores Luciano, Marinho e Luciana, cuja orientação garantiu a qualidade desse trabalho, além de gerar bons frutos na forma de produções bibliográficas.

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

AES	Advanced Encryption Standard
CA	Certificate Authority
DDoS	Distributed Denial of Service
DoS	Denial of Service
ILP	Integer Linear Programming
MIP	Mixed Integer Programming
MIPS	Million Instructions per Second
OS	Operating System
SLA	Service Level Agreement
TVD	Trusted Virtual Domain
VLAN	Virtual Local Area Network
VMM	Virtual Machine Monitor
VPN	Virtual Private Network

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Network virtualization has become increasingly prominent in recent years. It enables the creation of network infrastructures that are specifically tailored to the needs of distinct network applications and supports the instantiation of favorable environments for the development and evaluation of new architectures and protocols. Although recent efforts (motivated mainly by the search for mechanisms to evaluate Future Internet proposals) have contributed substantially to materialize this concept, none of them has attempted to combine efficient resource mapping with fulfillment of security requirements (*e.g.*, confidentiality). It is important to note that, in the context of virtual networks, the protection of shared network infrastructures constitutes a fundamental condition to enable its use in large scale.

Considering the negative impact of security provisions in the virtual network embedding process is of paramount importance in order to fully utilize physical resources without underestimating capacity requirements. Therefore, in this thesis we propose both an optimal model and a heuristic algorithm for embedding virtual networks on physical substrates that aim to optimize physical resource usage while meeting security requirements. Both approaches feature precise modeling of overhead costs of security mechanisms used to protect virtual networks, and are able to handle virtual network requests in an online manner. In addition, we present the results of an extensive evaluation we carried out, including a detailed comparison of both the optimal model and the heuristic algorithm. Our experiments show that the Integer Linear Programming (ILP) model is capable of optimally embedding virtual networks on physical infrastructures with up to a hundred routers, while the heuristic algorithm is capable of scaling to larger infrastructures, providing timely, sub-optimal mappings.

Keywords: Network Virtualization, Embedding, Security, Confidentiality, Linear Programming.

Mapeamento Eficiente e *On-Line* de Redes Virtuais Seguras

RESUMO

A virtualização de redes tem se tornado cada vez mais proeminente nos últimos anos. Tal técnica permite a criação de infraestruturas de rede que se adaptam a necessidades específicas de aplicações de rede distintas, além de dar suporte à instanciação de ambientes favoráveis para o desenvolvimento e avaliação de novas arquiteturas e protocolos. Apesar de esforços recentes (motivados principalmente pela busca de mecanismos para avaliar propostas relacionadas à Internet do Futuro) terem contribuído substancialmente para a materialização desse conceito, nenhum preocupou-se em conciliar alocação eficiente de recursos e satisfação de requisitos de segurança (*e.g.*, confidencialidade). É importante ressaltar que, no contexto de redes virtuais, a proteção de infraestruturas de rede compartilhadas constitui condição fundamental para seu uso em larga escala.

É de grande importância que o impacto negativo causado pelo aprovisionamento de segurança seja considerado no processo de mapeamento de redes virtuais, de forma a permitir o uso integral dos recursos físicos sem subestimar requisitos de capacidade. Portanto, nesta dissertação, são propostos um modelo ótimo e um algoritmo heurístico para realizar o mapeamento de redes virtuais em substratos físicos que têm por objetivo otimizar a utilização de recursos físicos garantindo a satisfação de requisitos de segurança. Ambas as abordagens possuem uma modelagem precisa de custos adicionais associados a mecanismos de segurança usados para proteger redes virtuais, e são capazes de atender requisições de redes virtuais de forma *online*. Além disso, são apresentados os resultados de um extensivo processo de avaliação realizado, incluindo uma comparação detalhada entre o modelo ótimo e o algoritmo heurístico. Os experimentos revelam que o modelo baseado em Programação Linear Inteira é capaz de alocar redes virtuais de forma ótima em substratos físicos com até cem roteadores, enquanto que o algoritmo heurístico é capaz de adaptar-se a infraestruturas maiores, provendo mapeamentos sub-ótimos em um curto espaço de tempo.

Palavras-chave: Virtualização de Redes, Mapeamento, Segurança, Confidencialidade, Programação Linear.

1 INTRODUCTION

In recent years, there has been a growing demand for adaptive network services with increasingly distinct requirements. Driven by such demands, and stimulated by the successful employment of virtualization for hosting custom-built servers, researchers have started to explore the use of this technique in network infrastructures. Network virtualization enables the creation of virtual topologies on top of physical substrates. This is made possible by instantiating one or more virtual routers on physical devices and establishing virtual links between these routers, forming topologies that are not limited by the structure of the physical network.

In addition to the ability to create different topological structures, virtual networks are also not bound by other characteristics of the physical network, such as its protocol stack. Thus, it is possible to instantiate virtual network infrastructures that are specifically tailored to the needs of different network applications (FERNANDES et al., 2011). These features also enable the creation of virtual testbeds that are similar to real infrastructures, a valuable asset for evaluating newly developed architectures and protocols without interfering with production traffic. For these reasons, network virtualization has attracted the interest of a number of researchers worldwide, specially in the context of Future Internet research (FARIAS et al., 2011).

Network virtualization has been embraced by the Industry as well. Important companies nowadays offer network devices supporting virtualization, and this new functionality allowed infrastructure providers to offer new services. The support of major Industry players to this kind of initiative can be observed, for example, in the list of members of the Open Networking Foundation[1], which promotes the development and usage of software-defined, virtualized networks.

Despite the wide applicability of network virtualization, both efficient resource allocation and security provisions must be taken into consideration. Regarding resource allocation, on one side there are infrastructure providers, which aim to increase their revenue by hosting the highest possible number of virtual networks while minimizing their costs. On the other, there are a number of clients who request virtual networks with specific resource demands. The resource allocation method needs to guarantee that the requested resources will be available for each of these clients, while attempting to minimize the infrastructure provider's costs. Additionally, the result of the mapping process needs to be delivered in an acceptable time frame.

Security concerns in virtual network environments arise from the shared use of

[1] http://www.opennetworking.org/membership

routing devices and communication channels. Without adequate protection, users from a virtual network may be able to capture data or even tamper with traffic belonging to other virtual networks on the same substrate. Such actions would violate security properties such as confidentiality and integrity. Therefore, it is of great importance that virtualization architectures offer protection against these and other threats that might compromise their security.

The resource allocation problem is known to be NP-hard (ANDERSEN, 2002), and has been commonly approached in the literature with resource embedding algorithms modeled by means of linear programming. Although related work exists in the area of virtual network embedding (YU et al., 2008; CHOWDHURY; RAHMAN; BOUTABA, 2009; FAROOQ BUTT; CHOWDHURY; BOUTABA, 2010; RAHMAN; AIB; BOUTABA, 2010; ALKMIM; BATISTA; FONSECA, 2013; CHENG et al., 2011; DAVY et al., 2011), we are not aware of previous investigations aimed at reconciling efficient resource mapping and satisfaction of security requirements. The motivation to properly tackle this issue is threefold. First, infrastructure providers need to be able to host a large number of virtual networks sharing the same physical substrate while preserving the confidentiality of each network. Second, while physical resources need to be efficiently utilized, the amount of resources needed to offer security provisions must be considered in order to not underestimate the capacity requirements of virtual network requests. Third, adequate mappings that meet the previously mentioned requirements must be generated as promptly as possible.

To cover the aforementioned gap, in this thesis we propose both an optimal model (based on Integer Linear Programming – ILP) and a heuristic algorithm for embedding virtual networks on physical substrates that aim to optimize physical resource usage while meeting security requirements. Both approaches feature precise modeling of overhead costs of security mechanisms used to protect virtual networks (which directly impact the embedding process), and are able to handle virtual network requests in an online manner (*i.e.*, individually as they arrive). In addition, we present the results of an extensive evaluation we carried out, including a detailed comparison of both the optimal model and the heuristic algorithm.

In summary, we introduce as major relevant contributions of this thesis: *(i)* a characterization of the state-of-the-art regarding security in network virtualization; *(ii)* an ILP model capable of optimally embedding virtual network requests in an online manner, featuring precise modeling of overhead costs of security mechanisms; and *(iii)* a heuristic algorithm with the same features as the ILP model, capable of scaling to larger infrastructures by providing timely, sub-optimal mappings.

2 BACKGROUND

Our first effort to understand the area of security in virtual networks resulted in a systematic analysis of the state-of-the-art. In the following sections, we first present a brief overview of existing techniques employed for the instantiation of virtual networks, as well as a taxonomy created to organize existing publications in this area in a comprehensive manner. Subsequently, we describe the main security threats and services found in the literature, categorized according to the aforementioned taxonomy. Should the reader be familiar with these concepts and techniques, reading of this chapter may be omitted.

2.1 Network Virtualization Techniques

Network virtualization consists in sharing resources from physical network devices (routers, switches, etc.) among different virtual networks. It allows the coexistence of multiple, possibly heterogeneous networks, on top of a single physical infrastructure. The basic elements of a network virtualization environment are shown in Figure 2.1. At the physical network level, a number of autonomous systems are represented by interconnected network substrates (e.g., substrates A, B, and C). Physical network devices are represented by nodes supporting virtualization technologies. Virtual network topologies (e.g., virtual networks 1 and 2), in turn, are mapped to a subset of nodes from one or more substrates. These topologies are composed of virtual routers, which use a portion of the resources available in physical routers, and virtual links, which are mapped to physical paths composed of one or more physical links and their respective intermediate routers.

From the point of view of a virtual network, virtual routers and links are seen as dedicated physical devices. However, in practice, they share physical resources with routers and links from other virtual networks. For this reason, the virtualization technology used to create this environment must provide an adequate level of isolation in order to enable the use of network virtualization in real, large scale environments.

Over the years, different methods for instantiating virtual networks have been used. Typical approaches include VLANs (Virtual Local Area Networks) and VPNs (Virtual Private Networks). Recently, Virtual Machine Monitors and programmable networks have been employed to create virtual routers and links over physical devices and communication channels. These approaches are briefly revisited in the following subsections.

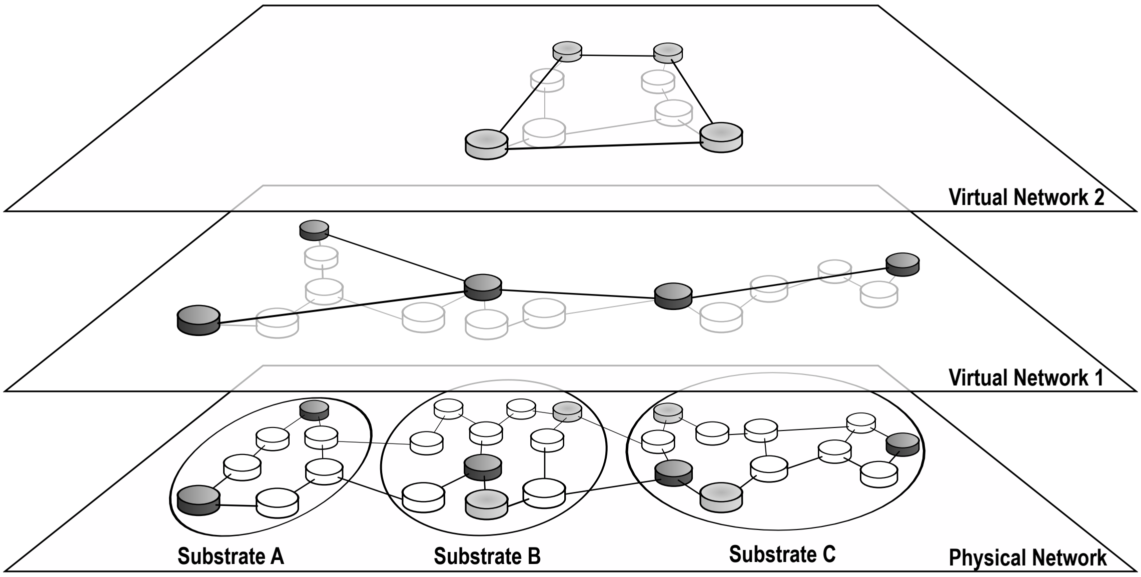


Figure 2.1: Network virtualization model, denoting a scenario with multiple physical substrates and virtual networks.

2.1.1 Protocol-Based Approaches

Protocol-based approaches consist in implementing a protocol that enables identifying and isolating distinct networks. The only requirement of this kind of approach is that physical devices (or a subset of them) support the selected protocol.

One example of protocol-based network virtualization are VLANs. VLANs consist of logical partitions of a single underlying network. Devices in a VLAN communicate with each other as if they were on the same Local Area Network, regardless of physical location or connectivity. All frames sent through a network are tagged with their corresponding VLAN ID, processed by VLAN-enabled routers and forwarded as necessary (LAN/MAN STANDARDS COMMITTEE, 2006).

Another commonly used approach is the creation of Virtual Private Networks. VPNs are typically used to provide a secure communication channel between geographically distributed nodes. Cryptographic tunneling protocols are used in order to provide data confidentiality and user authentication. VPNs can be provided in the physical, data link, or network layers according to the protocols used (ROSEN et al., 2006).

2.1.2 Machine Virtualization-Based Approaches

Machine virtualization-based approaches consist in creating virtual networks by means of groups of interconnected virtual machines. Virtual Machine Monitors are used to create virtual routers, and virtual links are created between them, regardless of physical network topology. Table 2.1 shows different machine virtualization-based techniques that can be used to create virtual networks, as well as a brief explanation and an example of each.

This alternative is remarkably flexible, as it allows the use of customized software,

Technique	Description	Examples
Full Virtualization	The Virtual Machine Monitor emulates a complete machine, based on the underlying hardware architecture. The guest Operating System runs without any modification.	VMware Workstation, VirtualBox
Paravirtualization	The Virtual Machine monitor emulates a machine which is similar to the underlying hardware, with the addition of a hypervisor. The hypervisor allows the guest Operating System to run complex tasks directly on non-virtualized hardware. The guest OS must be modified in order to take advantage of this feature.	VMware ESX, Xen
Container-based Virtualization	Instead of running a full Virtual Machine, this technique provides Operating System-level containers, based on separate userspaces. In each container, the hardware, as well as the Operating System and its kernel, are identical to the underlying ones.	OpenVZ, Linux VServer

Table 2.1: Virtualization techniques.

and relatively cheap, as it does not require the use of specific hardware[1]. However, this approach introduces problems from the area of server virtualization, some of which are mentioned in Sections 2.3 and 2.4. A general study on the security issues that arise from the use of machine virtualization was performed by van Cleeff *et al.* (CLEEFF; PIETERS; WIERINGA, 2009).

2.1.3 Programmable Networks

Programmable routers have been used to enable the creation of virtual networks. Although this is not a new concept, research in this area has been recently stimulated by the inception of Software-Defined Networking (SDN). This technique consists in creating flow rules in order to provide logical partitioning of physical networks. Traffic flows that belong to distinct virtual networks are treated according to their own sets of rules, allowing data plane isolation.

OpenFlow (MCKEOWN et al., 2008), one of the most promising techniques for implementing this technology, consists in using a secure protocol that allows a central controller to create and manage flow rules, which are transmitted to routers that support this protocol. OpenFlow gave rise to the Open Networking Foundation, an organization ran by major companies within the area of computer networks that aims to disseminate this type of technology.

[1] Machine virtualization is available for personal computers, in commonly used operating systems (e.g., Windows, Linux, and Mac OS X).

2.2 A Taxonomy of Security in Network Virtualization

The first step towards a comprehensive analysis of the literature was the selection of a number of publications from quality conferences and journals. Following this, a taxonomy was created in order to aid the organization and discussion of the selected publications. For this purpose, two well known classifications in the area of network security were chosen. Papers are organized according to the *security threats* they aim to mitigate, and afterwards, according to the *security services* they provide. As different authors have different definitions for each of these concepts, these classifications are briefly explained in the following subsections. The direct connection between them and the area of virtual network security is explained in sections 2.3 and 2.4, respectively.

In addition to these broad classifications, subcategories were created in order better organize this body of work. Figure 2.2 presents the full hierarchical organization that will be used in sections 2.3 and 2.4. Dark gray boxes represent broad categories used in the literature (SHIREY, 2000; STALLINGS, 2006), while white boxes denote subdivisions proposed and created by the authors of this thesis.

Security Threats

There are a number of malicious actions, or threats, that may violate security constraints of computational systems. Shirey (SHIREY, 2000) describes and divides the consequences of these threats into four categories, namely *disclosure*, *deception*, *disruption*, and *usurpation.*

Unauthorized *disclosure* is defined as gaining unauthorized access to protected information. Sensitive data may be erroneously exposed to unauthorized entities, or acquired by an attacker that circumvents the system's security provisions.

Deception is characterized by intentionally attempting to mislead other entities. For example, a malicious entity may send false or incorrect information to others, leading them to believe that this information is correct. Fake identities may be used in order to incriminate others or gain illegitimate access.

Disruption means causing failure or degradation of systems, negatively affecting the services they provide. This may be done by directly incapacitating a system component or the channel through which information is delivered, or by inducing the system to deliver corrupted information.

Last, through *usurpation*, an attacker may gain unauthorized control over a system. This unauthorized control may allow the attacker to illegitimately access protected data or services, or tamper with the system itself in order to cause incorrect or malicious behavior.

Security Services

Due to the existence of the previously described threats, computational systems must provide a series of services in order to maintain a desirable level of security. Stallings (STALLINGS, 2006) categorizes these essential services into six subdivisions, namely *access control*, *authentication*, *data confidentiality*, *data integrity*, *nonrepudiation*, and *availability.*

Access control allows a system to administer which entities will be able to access its functions, and what permissions each of these entities will have. In order to grant individual access rights and permissions, entities must be properly authenticated in

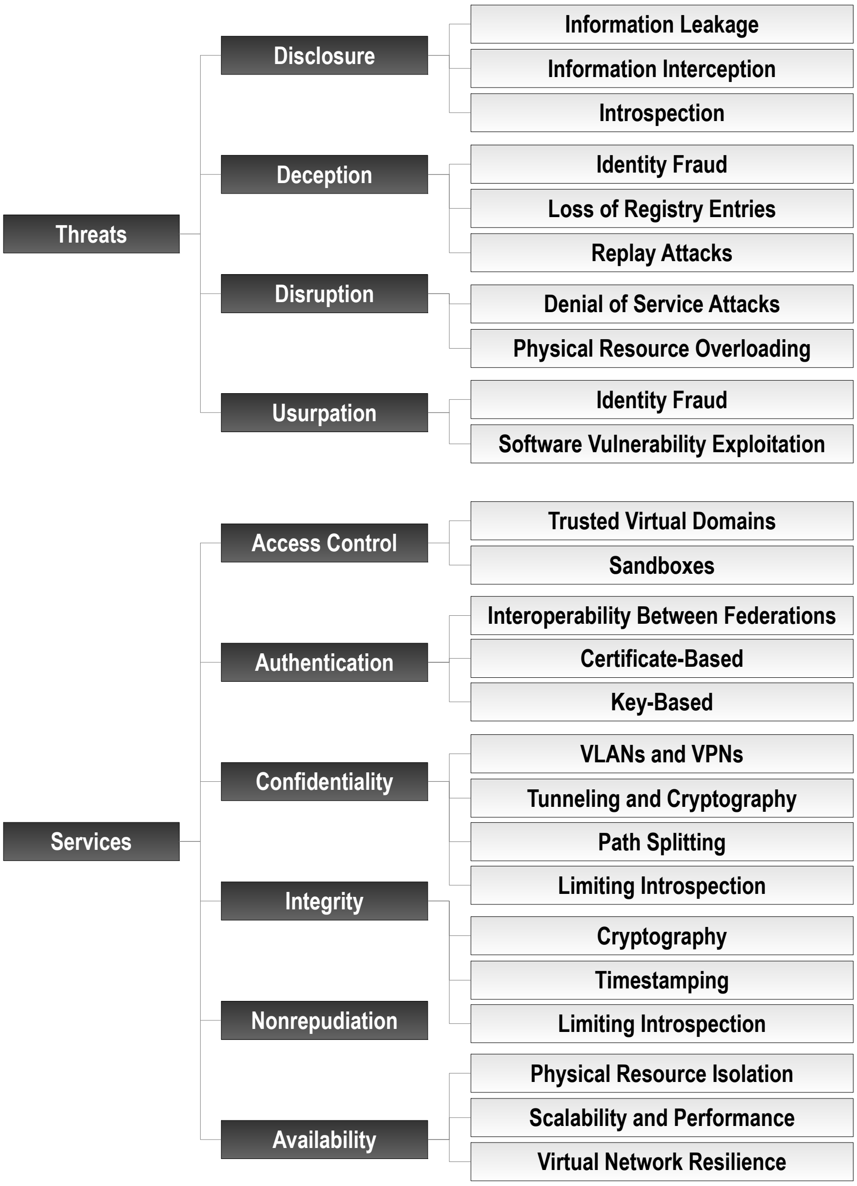


Figure 2.2: Taxonomy used to classify publications in the area of virtual network security.

the system.

The purpose of *authentication* is to ensure that entities communicating with each other are, in fact, the entities they claim to be. The receiver of a message must be able to correctly identify its sender, and an entity must not be able to impersonate another.

Providing adequate *data confidentiality* means ensuring that third parties do not have access to confidential information being transmitted between two entities. Additionally, the system should inhibit attackers from deriving information by analyzing traffic flow characteristics.

The *data integrity* service has the purpose of assuring that data stored by entities or transmitted through a network are not corrupted, adulterated or destroyed. Attacks such as duplication, modification, reordering, and replay of messages must be prevented. Furthermore, mechanisms for recovering from data corruption may also be provided.

In communications between peers, *nonrepudiation* provides a way to settle disputes when an entity denies having performed a certain action. The goal of this service is to prevent entities from falsely denying participation in any (possibly malicious) network-related activity.

The last security service is *availability*. System resources must be available upon request by an authorized entity, and the system must also conform to its performance specifications. In order to maintain availability, countermeasures against attacks such as *denial of service* must be provided.

2.3 Security Threats

In this section, we present a comprehensive list of threats found in network virtualization environments.

2.3.1 Disclosure

In an environment where physical resources are shared between a number of virtual networks, there is a series of behaviors that may result in undesired disclosure of information. Threats related to disclosure of private or sensitive information are explained next.

Information Leakage

Cavalcanti *et al.* (CAVALCANTI et al., 2006) mention the possibility of messages being leaked from one virtual network to another. In this type of attack, an entity may disclose private or sensitive information to members of other virtual networks, who should not have access to such information. Wolinsky *et al.* (WOLINSKY et al., 2006) describe a similar attack, in which virtual nodes send messages to outside the boundaries of a network virtualization environment. This way, it would be possible for messages to reach physical nodes that not only do not belong to any virtual network, but are hosted outside of the virtualized network infrastructure.

Information Interception

Attackers in a virtual network environment may capture messages being exchanged between two entities in order to access their content. This type of attack, described by Cabuk *et al.* (CABUK et al., 2007) as “eavesdropping”, may lead to theft of confidential information. Eavesdropping is a common threat in any networking environment, but the use of shared of physical resources by multiple virtual networks further exacerbates this problem. According to these and other authors, such as Cui *et al.* (CUI; SHI; WANG, 2009), networking solutions provided by virtual machine monitors may not properly isolate data belonging to different virtual networks. This means that members of one virtual network may be able to access data being transfered by other virtual networks sharing the same substrate.

Even if data inside network packets is protected (e.g. through the use of cryptography), entities may be able to derive sensitive information by analyzing them. In traffic-analysis attacks, described by Huang *et al.* (HUANG; ATA; MEDHI, 2010), entities acquire such information by analyzing characteristics of traffic flows between communicating entities in virtual networks.

Introspection

Introspection is a feature present in virtual machine monitors that grants access to data inside virtual machines. According to van Cleeff *et al.* (CLEEFF; PIETERS; WIERINGA, 2009), this feature may be misused or exploited by attackers aiming to disclose sensitive data. This problem is aggravated by the fact that virtual nodes may be moved or copied between multiple virtual machine monitors, as sensitive data may be compromised through the exploitation of this feature on any virtual machine monitor permanently or temporarily hosting such virtual nodes.

2.3.2 Deception

We have identified three subcategories of threats that may lead to deception in virtual network environments. These subdivisions – namely identity fraud, loss of registry entries and replay attacks – are explained next.

Identity Fraud

In addition to dealing with unauthorized disclosure, Cabuk *et al.* (CABUK et al., 2007) also describe threats related to deception in virtual network environments. Specifically, virtual entities may inject malicious messages into a virtual network, and deceive others into believing that such messages came from another entity.

Certain characteristics of virtualized network environments increase the difficulty of handling identity fraud. The aggregation of different virtual networks into one compound network, known as federation, is indicated by Chowdhury *et al.* (CHOWDHURY; ZAHEER; BOUTABA, 2009) as one of such characteristics. Federation raises issues such as the presence of separate roles and possible incompatibility between security provisions or policies from aggregated networks. Another complicating factor mentioned by the authors is the dynamic addition and removal of entities. An attacker may force a malicious node to be removed and re-added in order to obtain a new identity.

Other characteristics that complicate the handling of identity fraud involve operations such as migration and duplication of virtual nodes, as mentioned by van Cleeff *et al.* (CLEEFF; PIETERS; WIERINGA, 2009). The study presented by the authors refers to virtualization environments in general. Therefore, in the context of this study, a virtual node may refer to either a virtual router or a virtual workstation. If a virtual node is migrated from one physical point to another, the identity of the machine that contains this virtual node may change. Moreover, virtual nodes may be copied to one or more physical points in order to provide redundancy, which may lead to multiple entities sharing a single identity. Both of these issues may cause inconsistencies in the process of properly identifying the origin of network messages, which may be exploited in identity fraud attacks.

Loss of Registry Entries

Van Cleeff *et al.* (CLEEFF; PIETERS; WIERINGA, 2009) also mention issues related to logging of operations in virtualization environments. If information regarding which entity was responsible for each operation in the network is stored in logs inside virtual machines, entries may be lost during rollback procedures. Likewise, logs of malicious activities performed by attackers may also be lost.

Replay Attacks

Fernandes and Duarte (FERNANDES; DUARTE, 2011a) mention replay attacks as another form of deception in virtual networks. In this type of attack, a malicious entity captures legitimate packets being transfered through the network and retransmits them, leading other entities to believe that a message was sent multiple times. The authors explain that virtual routers may launch attacks in which they repeat old control messages with the intention of corrupting the data plane of the attacked domain.

2.3.3 Disruption

In a network virtualization environment, proper management of resources is crucial to avoid disruption. Virtual networks may intentionally or unintentionally attempt to use more resources than they are allowed to, causing physical resources to be overloaded. Additionally, physical nodes may fail or need to be interrupted, causing a constant concern during the lifetime of the network.

Denial of Service Attacks

During their lifetime, virtual networks may suffer from attacks that aim to cause disruption. These attacks may come from within the virtual network itself, or from outside sources. The most common threats are Denial of Service (DoS) attacks, as presented by Yu and Zhou (YU; ZHOU, 2008), as well as Roschke *et al.* (ROSCHKE; CHENG; MEINEL, 2009) and Mazzariello *et al.* (MAZZARIELLO; BIFULCO; CANONICO, 2010).

Physical Resource Overloading

Physical resource overloading may lead to failure of virtual nodes, or cause the network performance to degrade below its minimum requirements. This degradation may cause congestion and packet loss in virtual networks, as stated by Zhang *et al.* (ZHANG; GAO; WANG, 2009). In addition to causing disruption in already established networks, overloading may also hinder the deployment of new ones.

Resource requirements themselves can be a point of conflict in virtual network environments. As explained by Marquezan *et al.* (MARQUEZAN et al., 2010), multiple virtual networks may require an excessive amount of resources in the same area of the substrate network. While such prohibitive demands may be unintentional, they may also be due to a coordinated attack. This may not only happen during deployment operations, but also during the lifetime of virtual networks.

It is also possible for one virtual network to disrupt another by using more than its fair share of resources. This concern is explored by a number of authors in their respective publications (GOVINDAN et al., 2009; WU; SHANBHAG; WOLF, 2010; KOKKU et al., 2010; FERNANDES; DUARTE, 2011a,b). Isolation and fair distribution of physical resources among virtual networks are essential to maintain the network virtualization environment operating properly. This includes assuring that the minimum requirements of each network will be fulfilled, as well as prohibiting networks from consuming more resources than they are allowed to.

Last, the process of virtualization itself generates resource overheads, which raises concerns regarding performance and scalability. These concerns are exposed by authors such as Bhatia *et al.* (BHATIA et al., 2008) and El-Darieby *et al.* (EL-DARIEBY; ROLIA, 2006).

2.3.4 Usurpation

In virtual network environments, usurpation attacks may allow an attacker to gain access to privileged information on virtual routers, or to sensitive data stored in them. Such attacks may be a consequence of identity fraud or exploited vulnerabilities, which are explained next.

Identity Fraud

As previously mentioned in Subsection 2.3.2, identity fraud attacks can be used to impersonate other entities within a virtual network. By impersonating entities with high levels of privilege in the network, attackers may be able to perform usurpation attacks. As an example, the injection of messages with fake sources mentioned by Cabuk *et al.* (CABUK et al., 2007) is used for this purpose. By sending a message that appears to have been originated from a privileged entity, attackers may perform actions restricted to such entities, including elevating their own privilege level.

Software Vulnerability Exploitation

Roschke *et al.* (ROSCHKE; CHENG; MEINEL, 2009) mention that virtual machine monitors are susceptible to the exploit of vulnerabilities in their implementation. According to the authors, by gaining control over a virtual machine monitor, attackers can break out of the virtual machine, obtaining access to the hardware layer. In an environment that uses full virtualization or paravirtualization to instantiate virtual routers, exploiting such vulnerabilities may enable an attacker to have full control over physical routers. By gaining access to physical devices, attackers could easily compromise any virtual networks provided by the infrastructure.

2.4 Security Services

In this section, we explore solutions published in the literature that aim to provide security and protect the environment from the aforementioned security threats.

2.4.1 Access Control

Access control aims to enforce distinct privilege levels for virtual network usage. This service is described in two approaches in the literature, namely Trusted Virtual Domains and sandboxes, explained next.

Trusted Virtual Domains

Cabuk *et al.* (CABUK et al., 2007) devised a framework to provide secure networking between groups of virtual machines. Their security goals include providing isolation, confidentiality, integrity, and information flow control in these networks. The framework provides the aforementioned security services through the use of Trusted Virtual Domains (TVDs). Each TVD represents an isolated domain, composed of “virtualization elements” and communication channels between such elements. In Cabuk's proposal, the virtualization elements are virtual workstations. However, the concept of TVDs may be applied to any device supporting virtualization.

Access control is performed when virtual machines join a TVD, ensuring that only machines that satisfy a given set of conditions are able to join. This admission control may be applied continuously in case prerequisites to join a TVD are changed. Additionally, TVDs leverage access policies to prevent unauthorized access.

Sandboxes

Wolinsky *et al.* (WOLINSKY et al., 2006) use virtual machine sandboxes in order to provide security in large scale collaborative environments. Although this work focuses on networked virtual machines hosting virtual workstations, this concept can be extended to virtual networks. Sandboxes are used to limit virtual machine access to physical resources, preventing malicious virtual machines from accessing data within other virtual machines. Moreover, each virtual machine supports IPSec, enabling the creation of secure communication channels, and X.509, providing virtual machine authentication. The authentication process is detailed in Subsection 2.4.2.

2.4.2 Authentication

In this subsection, we describe the approaches found in the literature which aim to provide authentication in network virtualization environments.

Interoperability between Federated Virtual Networks

Although isolation is one of the main security requirements in virtual networking, there are cases in which distinct virtual networks must be able to cooperate. The federation of virtual networks can, for example, enable end-to-end connectivity – through virtual devices of distinct virtual networks – or allow access to distinct services. However, it may not be possible to provide interoperability due to the heterogeneous nature of virtual networks (which may implement different, incompatible protocols). Chowdhury *et al.* (CHOWDHURY; ZAHEER; BOUTABA, 2009) partially tackle this issue with a framework that manages identities in this kind of environment. The main objective of the work is to provide a global identification system that does not restrict the internal identification mechanisms used locally by virtual networks, allowing each virtual network to keep its own internal naming scheme. Additionally, this framework provides interfaces and mechanisms to enable end-to-end connectivity without limiting the internal functionalities of virtual networks. Moreover, identifiers do not restrict the mobility of machines or network devices, as they are not associated with physical location. Last, in order to provide trust and security in the environment, the framework requires global identifiers to be unique and immutable.

Certificate-Based

As previously mentioned, the framework presented by Cabuk *et al.* (CABUK et al., 2007) makes use of Trusted Virtual Domains (TVDs) to provide access control and network isolation. The authentication necessary to support access control is provided by means of digital certificates. These certificates ensure the identity of entities joining the network. Additionally, the system makes use of Virtual Private Networks (VPNs) to authenticate entities in network communications.

Analogously, Wolinsky *et al.* (WOLINSKY et al., 2006) use IPSec with X.509-based authentication for the purpose of access control in their system. In order access the system, joining machines must request a certificate to the Certification Authority (CA). The CA responds by sending back a signed certificate to the node. The IP address of the requesting node is embedded into the certificate in order to prevent other nodes from reusing it.

Key-Based

Fernandes and Duarte (FERNANDES; DUARTE, 2011a,b) present an architecture that aims to provide efficient routing, proper resource isolation and a secure communication channel between routers and the Virtual Machine Monitor (VMM) in a physical router. In order to ensure efficiency, virtual routers copy routing-related information to the VMM – in this case, the hypervisor. This process is performed by a plane separation module, which separates the data plane (which contains routing rules) and the control plane (responsible for creating routing rules). As a result, packets matching rules in the hypervisor routing table do not need to be redirected to virtual routers, resulting in a significant performance speedup. However, the process of copying routing information needs to be authenticated such that a malicious router is not able to compromise the data plane of another router.

In order to prevent identity fraud, the system requires mutual authentication between virtual routers and the VMM. The proposed solution uses asymmetrical cryptography to perform an initial exchange of session keys, enabling the creation of a secure communication channel. Figure 2.3 illustrates the architecture developed by the authors. Each virtual router, upon instantiation, connects to the hypervisor following the client–server paradigm. After the initial key exchange, the secure communication module is used by other system modules in order to allow message exchanges with the hypervisor.

2.4.3 Data Confidentiality

As previously mentioned, confidentiality is an extremely important security service in environments where network devices and links are shared between multiple entities. This subsection explores the approaches presented in the literature aiming to provide this service.

VLANs and VPNs

The security goals approached by Cabuk *et al.* (CABUK et al., 2007) include integrity, data isolation, confidentiality, and information flow control. Other than integrity, the remaining three goals, according to the authors, are directly related, and can be tackled by a data confidentiality service. The framework uses TVDs to control data access. However, virtual machines that belong to different TVDs may be hosted in the same physical machine. Therefore, it is necessary to ensure proper isolation, preventing a TVD from accessing data that belongs to another TVD.

The proposed solution for this challenge employs a combination of Virtual Local Area Networks (VLANs) and Virtual Private Networks (VPNs). VLANs are used to identify packets belonging to different networks, allowing VLAN-enabled devices

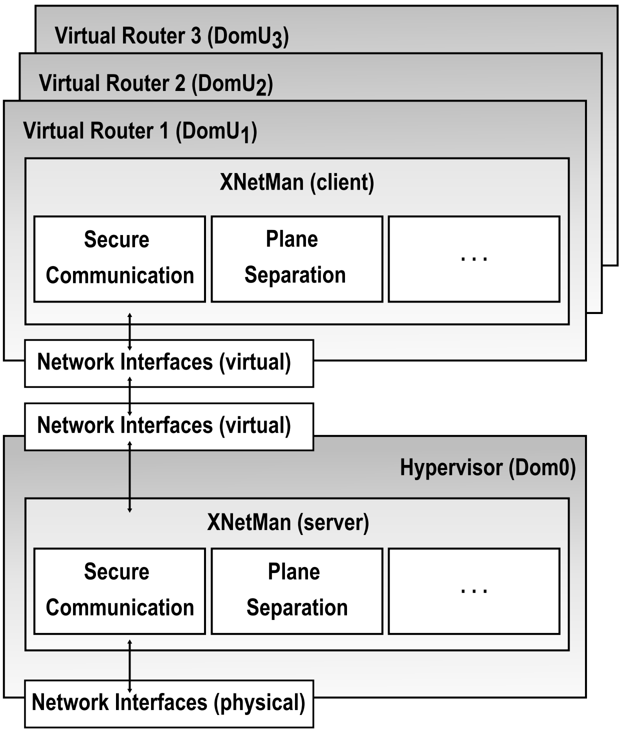


Figure 2.3: Simplified version of the architecture presented by Fernandes et al. (FERNANDES; DUARTE, 2011b,a), showing the secure communication modules.

to route packets to the appropriate network interfaces, thus providing adequate isolation. Untrusted physical channels, however, may require a higher level of security. Therefore, if necesary, VPNs are used to provide data confidentiality by means of end-to-end cryptography.

Tunneling and Cryptography

Wolinsky *et al.* (WOLINSKY et al., 2006) make use of tunelling in order to isolate network traffic between virtual machines (in this case, virtual workstations). Two tunneling approaches are employed. In the first approach, the host system runs a tunneling software that captures packets incoming from physical interfaces and forwards them to virtual machines. In the second approach, the tunneling software runs inside virtual machines, and traffic is restricted within virtual networks through the use of firewall rules. According to the authors, while the second approach is easier to deploy, malicious users may be able to subvert this firewall, compromising the system. Although the focus of Wolinsky's work is isolation between virtual workstations, we believe that the techniques used to achieve such isolation could be extended to virtual routers in network virtualization environments.

Fernandes and Duarte (FERNANDES; DUARTE, 2011a,b) deal with data confidentiality in communications between a virtual router and the Virtual Machine Monitor (VMM) hosting it. After an initial session key exchange, described in Subsection 2.4.2, virtual routers use symmetrical cryptography in order to securely communicate with the VMM.

Huang *et al.* (HUANG; ATA; MEDHI, 2010) present a framework that provides secure routing. In the environment presented by the authors, routing information that is propagated through a virtual network is confidential and needs to be kept secret from unauthorized network entities. Routing information is categorized in groups, and group keys are assigned to virtual routers. Therefore, routing information can be encrypted, ensuring that only routers with the correct key are able to decrypt this information. Thus, routing information relative to a given group is protected against unauthorized access from other groups, other virtual networks or the physical network itself.

Path Splitting

In addition to encryption of routing information, Huang *et al.* (HUANG; ATA; MEDHI, 2010) use variable paths in virtual networks to propagate data flows. According to the authors, this path splitting approach helps mitigate traffic-analysis attacks coming from the physical network.

Limiting Introspection

Finally, van Cleeff *et al.* (CLEEFF; PIETERS; WIERINGA, 2009) present recommendations for safer use of virtualization. One of these recommendations is to limit, or even disable, the introspection feature, which allows virtual machine monitors to access data inside virtual machines. While useful, this functionality may be exploited by attackers, as explained previously.

2.4.4 Data Integrity

Data integrity is a highly important security property of virtual networks, preventing entities from tampering with data that passes through shared physical devices and links. Next, we describe the proposals found in the literature that aim to provide this security service.

Cryptography

In addition to authentication (*i.e.*, source integrity) and confidentiality, the framework developed by Cabuk *et al.* (CABUK et al., 2007) makes use of VPNs to provide data integrity to virtual networks. The use of cryptographic tunneling protocols prevents malicious entities from manipulating messages going through the network. As previously discussed, the authors use IPSec as the tunneling protocol.

Timestamping

As previously discussed, replay attacks are one of the threats to data integrity that may be present in network virtualization environments. The addition of unique identifiers inside encrypted messages makes it possible to detect duplicated messages, and therefore, replay attacks. For this purpose, the architecture proposed by Fernandes and Duarte (FERNANDES; DUARTE, 2011a,b) inserts timestamps inside encrypted messages in order to ensure that messages are non-reproducible.

Limiting Introspection

Besides mitigating information theft, disabling or limiting introspection also prevents data tampering. According to van Cleeff *et al.* (CLEEFF; PIETERS; WIERINGA, 2009), this functionality allows the VMM to modify applications running inside it, which may cause inconsistencies. Another recommendation consists of specifically designing applications that facilitate batch processing and checkpointing. According to the authors, this minimizes security issues associated with rollback and restore operations that may otherwise threaten integrity.

2.4.5 Nonrepudiation

The nonrepudiation service provides evidences regarding which (potentially malicious) actions have been performed by which entities. This service is highly valuable in the context of network virtualization environments, in which a number of physical devices are shared by different users. Nevertheless, we are not aware of any publication that targets this service specifically.

2.4.6 Availability

Last, we present proposals that aim to maintain the availability of network virtualization environments.

Physical Resource Isolation

One of the main concerns regarding availability is the abuse of physical resources by virtual networks. Virtual networks may attempt to use as much resources as possible in order to maximize their performance. If the environment is not adequately protected, this behavior may lead to the exhaustion of physical resources, compromising the availability of other virtual networks on the same substrate. Therefore, physical resources must be shared in a fair manner, and actions performed by a virtual network must not negatively impact others.

According to Wu *et al.* (WU; SHANBHAG; WOLF, 2010), the sharing of physical resources by packet processors is usually only performed at a granularity of entire processor cores. The authors claim that finer-grained processor sharing is required in order to provide scalability for network virtualization environments. Thus, the authors propose a system that allows multiple threads to share processor cores concurrently while maintaining isolation and fair resource sharing. However, typical multithreading approaches consider a cooperative environment, which is not the case in network virtualization environments. The authors devise a fair multithreading mechanism that allows the assignment of different weights to each thread, in order to increase or decrease their priority. Additionally, this mechanism takes into account the history of how much processing has been performed by each thread. Inactivity times are also considered in order to guarantee that threads will not stay idle for too long. The evaluation performed by the authors shows that the proposed mechanism is able to properly distribute processing resources according to the defined weights. Furthermore, while it requires more processing power, it is able to provide better resource utilization in comparison to coarse-grained approaches.

Kokku *et al.* (KOKKU et al., 2010) propose a network virtualization scheme that provides resource isolation while aiming to maximize substrate utilization. It is capable of managing shared resources in order to simultaneously meet bandwidth-based and resource-based reservations. Slices are divided in two groups according to the type of reservation they require, and treated independently by the slice scheduler. The slice scheduler calculates a weight for each slice based on its reservation and its average resource usage rate, and schedules the slice with the maximum calculated weight at each instant. According to the authors, the implemented prototype was capable of ensuring that each slice meets its reservations.

Fernandes and Duarte (FERNANDES; DUARTE, 2011a) present a network monitor that employs plane separation in order to provide resource isolation in network virtualization environments. The system is able to allocate resources based on fixed reservations, as well as to redistribute idle resources between virtual networks that have a higher demand. Additionally, an administrator is able to control the amount of resources to be used by each virtual network, as well as set priorities for using idle resources. The system continuously monitors the consumption of physical resources by each virtual router. If any virtual router exceeds its allowed use of bandwidth, processing power, or memory, it is adequately punished by having packets dropped, or a percentage of its stored routes erased. Harsher punishments are instituted if there are no idle resources available. On the other hand, given punishments are gradually reduced if the router resumes to use no more than its allocated resources. According to the authors, this system is capable of adequately preventing physical resources from being overloaded, and packet drops employed by the punishment mechanism do not cause a major impact on network traffic.

In another publication (FERNANDES; DUARTE, 2011b), the same authors extend the previously described network monitor. This new system is able to control both short term and long term reservations. Short term reservations may be allocated on demand (*i.e.,* only allocated when necessary) or in an exclusive manner (always allocated, even if part of the allocated resources is idle). Long term reservations are only guaranteed over a greater time interval, and only if there is demand for them. The authors also propose an adaptive control scheme in order to improve the probability that long term reservations, if needed, will be met. The system calculates a weight for each virtual network based on a ratio between its unused long term resources and the unused long term resources of all virtual networks. This weight is used to prioritize virtual networks with greater long term reservations. The presented evaluation shows the improvement of this system over the original (FERNANDES; DUARTE, 2011a) in terms of guaranteeing that the demands of each virtual network will be met, as well as reducing resource load on the physical substrate.

According to Govindan *et al.* (GOVINDAN et al., 2009), it is common for virtual networks to experience delays while they wait for resources to become available. The authors propose a CPU scheduler that aims to reduce the aggregate delay of hosted virtual machines, while simultaneously providing guarantees on CPU allocations. While this solution is presented with a focus on server virtualization, the fact that this scheduler is oriented mainly towards input and output operations makes it highly applicable to network virtualization. The presented algorithm selects the virtual machine that, if scheduled, will result in reduced delay for the greatest number of packets, as long as this selection does not violate any resource reservations. This prioritization of communication-sensitive virtual machines results in a potentially unfair division of resources in the short term. However, this unfairness is controlled, and the scheduler guarantees that the reservations of each virtual machine will be met in the long term. Additionally, as virtual networks may either consume resources directly or through the Virtual Machine Monitor, both types of resource consumption are accounted by the system. Experiments performed by the authors show that the system meets its goal of long term fair scheduling, while increasing overall performance in most cases.

Scalability and Performance

El-Darieby and Rolia (EL-DARIEBY; ROLIA, 2006) present a scalable protocol for creation of virtual networks in scenarios of increasing demands and network sizes. In order to maintain scalability, the authors propose a hierarchical process for the creation of virtual networks, in which nodes are divided into hierarchically organized domains. Increasing hierarchy levels represent higher levels of abstraction in network topologies. For example, elements within a level "n" are organized into domains that are abstracted, represented, and managed by a single element in level "n+1". This scheme of hierarchical abstractions is represented in Figure 2.4, in which domains 1, 2, and 3 in the first level are represented by three elements in the second level, and these three elements, on their turn, are grouped and represented by a single element in the highest level. The interconnections between these domains are preserved in their higher level abstractions. According to the evaluation presented by the authors, this approach is scalable in terms of communication overhead generated by the setup process, as well as total setup time. However, due to its high cost, the authors deem it more suitable for creating long-lived, high-bandwidth virtual networks.

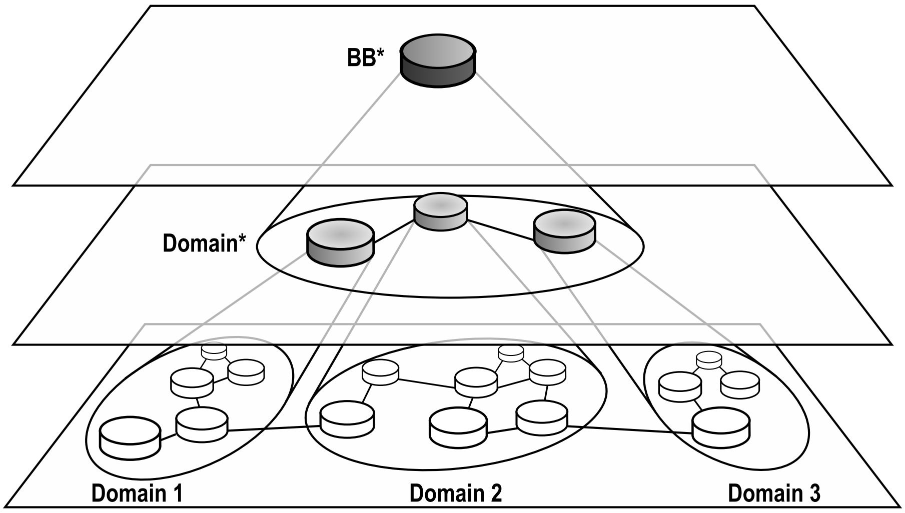


Figure 2.4: Graphical representation of the hierarchical system proposed by El-Darieby and Rolia (EL-DARIEBY; ROLIA, 2006).

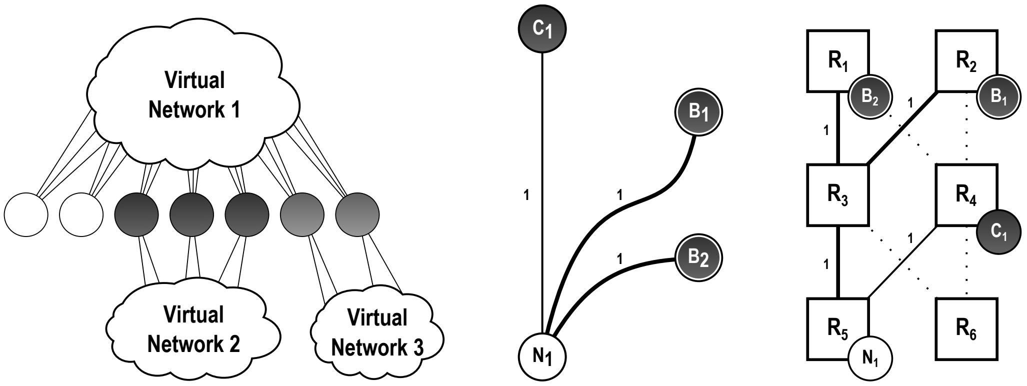
Virtual network performance is the main goal of the work presented by Bhatia *et al.* (BHATIA et al., 2008). The authors propose a platform that enables the creation of flexible, high performance virtual networks. This platform employs container-based virtualization, a method in which virtual containers are created at an operating system level through the use of isolated namespaces. This type of virtualization provides high performance at the cost of flexibility, since data structures are shared between containers. However, the proposed solution compensates this limitation by providing separate network namespaces, allowing each container to customize aspects of their network stack. Additionally, a tunneling mechanism based on Generic Routing Encapsulation is used in order to provide transparent link virtualization. The use of this tunneling module allows virtual routers hosted on the same physical network to use overlapping address spaces. Presented experiments show the performance advantage of this solution in comparison to full virtualization, in which each virtual instance is a complete emulated machine.

Virtual Network Resilience

Even with proper physical resource isolation, maintaining availability remains a challenge in network virtualization. The virtualization layer must be resilient, maintaining its performance and mitigating attacks in order to sustain its availability.

The solution presented by Yeow *et al.* (YEOW; WESTPHAL; KOZAT, 2011) aims to provide network infrastructures that are resilient to physical router failures. This objective is achieved through the use of backups (*i.e.,* redundant routers and links). However, redundant resources remain idle, reducing the utilization of the physical substrate. To minimize this problem, the authors propose a scheme that dynamically creates and manages shared backup resources. This mechanism minimizes the number of necessary backup instances needed to achieve a certain level of reliability. While backup resources are shared, each physical router is restricted to hosting a maximum number of backup instances in order to not sacrifice reliability.

Figure 2.5.a shows a simple representation of how backup nodes may be shared among different virtual networks. Figure 2.5.b depicts, in greater detail, how backups are allocated to virtual routers. A virtual router C_1 has virtual routers B_1 and B_2 as backups, and these backups preserve the connectivity of the original router with virtual router N_1 in terms of number of links and bandwidth. Further, Figure 2.5.c demonstrates how this topology may be allocated on the physical network. The aforementioned virtual routers, represented by circles, are instantiated in different physical routers, represented by squares. Physical links being used by virtual links are represented by solid lines, while unused links are represented by dashed lines.



(a) Backup resources being shared among virtual networks

(b) Virtual router and its backup instances

(c) Mapping of backup nodes

Figure 2.5: Examples of sharing and mapping of backup instances, used by Yeow *et al.* (YEOW; WESTPHAL; KOZAT, 2011) to provide resilient virtual networks.

The system presented by Zhang *et al.* (ZHANG; GAO; WANG, 2009) uses redundant virtual networks in order to provide reliable live streaming services. It is able to detect path failures and traffic congestion, dynamically redirecting data flows. Initially, the data flow is distributed equally among available virtual networks. Figure 2.6 depicts the distribution of the data flow among virtual networks, using multiple paths between a server and a client. Gradually, the number of packets routed through each virtual network is adapted according to their relative bandwidth capacities. Additionally, an active probing mechanism is used to detect failures in the physical network or routing problems (changes in routing tables, for example, may have a significant impact in live streaming applications). If an issue is detected, the system is able to redirect data flows away from problematic networks and redistribute it among the remaining ones. Experiments performed by the authors demonstrate advantages in using multiple networks instead of a single one, with increasing gains when using up to four virtual networks. Additionally, the authors claim that the bandwidth cost of the probing mechanism is neglectable.

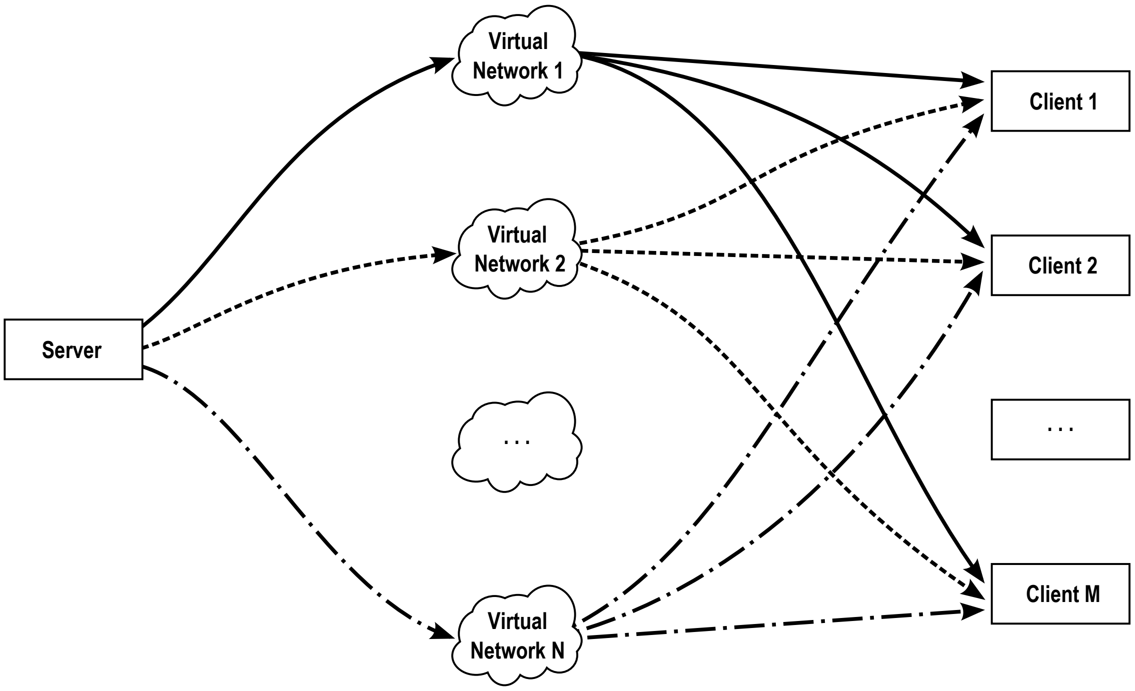


Figure 2.6: A live streaming data flow is distributed among different virtual networks, a mechanism used by Zhang *et al.* (ZHANG; GAO; WANG, 2009).

Distributed Denial of Service (DDoS) attacks are a common threat to the availability of network services. The system proposed by Yu and Zhou (YU; ZHOU, 2008) aims to detect such attacks on community networks (federated virtual networks that belong to cooperating entities). The devised solution takes advantage of this collaborative environment to detect possible attacks at an early stage. In this approach, edge routers monitor traffic passing through them and calculate the entropy of its flows. Traffic surges in any of these flows will cause the entropy to drop, indicating a possible attack. In this case, edge routers notify their respective downstream routers to calculate the entropy rate of this suspected flow. Calculated values are compared, and if they are similar, a DDoS attack is confirmed.

2.5 Discussion

A number of insights can be obtained from the extensive investigation of the state-of-the-art reported in this chapter. First, it is possible to observe that the publications in the area are not equally distributed between the main security cate-

Publication	Threats			
	DI	DE	DR	US
(WOLINSKY et al., 2006)	×			
(CUI; SHI; WANG, 2009)	×			
(HUANG; ATA; MEDHI, 2010)	×			
(CLEEFF; PIETERS; WIERINGA, 2009)	×	×		
(CABUK et al., 2007)	×	×		×
(CHOWDHURY; ZAHEER; BOUTABA, 2009)		×		
(FERNANDES; DUARTE, 2011a)		×	×	
(ROSCHKE; CHENG; MEINEL, 2009)			×	×
(YU; ZHOU, 2008)			×	
(BHATIA et al., 2008)			×	
(EL-DARIEBY; ROLIA, 2006)			×	
(ZHANG; GAO; WANG, 2009)			×	
(MARQUEZAN et al., 2010)			×	
(GOVINDAN et al., 2009)			×	
(WU; SHANBHAG; WOLF, 2010)			×	
(KOKKU et al., 2010)			×	
(FERNANDES; DUARTE, 2011b)			×	
(YEOW; WESTPHAL; KOZAT, 2011)			×	
(MAZZARIELLO; BIFULCO; CANONICO, 2010)			×	

Table 2.2: Security threats mentioned in the studied publications. From left to right: Disclosure, Deception, Disruption, Usurpation.

gories. Certain security threats and services are approached by a higher number of publications than others. Tables 2.2 and 2.3 show, respectively, the security threats and security services approached in these publications. In both tables, publications have been grouped together according to the security elements they approach, whenever possible.

It is noticeable that disruption and availability, a security threat and a security service that are directly correlated, are approached in the majority of these publications. Further, only a small number of publications approach more than one threat or service simultaneously. No single publication has dealt with threats in more than two of the four categories, or presented solutions that provide more than four security services, out of a total of six. Additionally, one security service in particular – nonrepudiation – was not approached by any of the publications. The combination of authentication and integrity, which exists in some publications, can be considered as the basis for the provision of nonrepudiation, but this specific service is not targeted.

As expected, many of the security issues seen in these publications arise from the shared use of physical resources. Infrastructure providers must impose limits on the actions of virtual networks in order to prevent intentional or unintentional abuse. Another consequence of resource sharing is the fact that any failure on a single point of the physical layer may harm a significant number of virtual infrastructures.

In the publications presented in this chapter, we can observe the use of a number of distinct virtualization techniques. These techniques range from full virtualization platforms (e.g., Xen and VMware), container-based virtualization (e.g., VServer and

Publication	Services					
	AC	AU	CO	IN	NR	AV
(HUANG; ATA; MEDHI, 2010)			×			
(WOLINSKY et al., 2006)	×	×	×			
(CABUK et al., 2007)	×	×	×	×		
(FERNANDES; DUARTE, 2011a)		×	×	×		×
(FERNANDES; DUARTE, 2011b)		×	×	×		×
(CLEEFF; PIETERS; WIERINGA, 2009)			×	×		
(CHOWDHURY; ZAHEER; BOUTABA, 2009)		×				
(YU; ZHOU, 2008)						×
(BHATIA et al., 2008)						×
(EL-DARIEBY; ROLIA, 2006)						×
(ZHANG; GAO; WANG, 2009)						×
(GOVINDAN et al., 2009)						×
(WU; SHANBHAG; WOLF, 2010)						×
(KOKKU et al., 2010)						×
(YEOW; WESTPHAL; KOZAT, 2011)						×

Table 2.3: Security services provided by the studied publications. From left to right: Access Control, Authentication, Confidentiality, Integrity, Nonrepudiation, Availability.

OpenVZ), or even programmable network routers (e.g., OpenFlow). Each of these platforms have their own sets of advantages, as well as security concerns, which have to be taken into consideration.

Last, to the best of our knowledge, there have been no previous attempts to consider the fulfillment of security requirements in the process of virtual network embedding. The approaches proposed in this thesis are an effort to cover this gap, reconciling these two areas. Our approaches – one based on Integer Linear Programming, and the other, on metaheuristics – allow virtual network requesters to choose among three levels of confidentiality, aiming to optimize resource usage while considering precise overhead costs of mechanisms used to provide confidentiality in virtual network environments.

3 RELATED WORK

In this chapter, we discuss previous work in the area of virtual network embedding, focusing on the distinctive features of each approach.

Yu et al. (YU et al., 2008) present a heuristic-based approach to solve the problem of virtual network embedding. The authors devise a greedy node mapping algorithm that prioritizes virtual networks with largest revenue value. Link mapping is performed by selecting the shortest path with enough bandwidth capacity or, if the request accepts path splitting, by solving the multicommodity flow problem. Furthermore, the algorithm is able to reoptimize the physical substrate by adjusting splitting ratios or remapping virtual links to different paths. The model considers that virtual network requests are not known in advance, and takes into account CPU and bandwidth requirements, as well as the maximum amount of time a request can wait before being served.

Chowdhury et al. (CHOWDHURY; RAHMAN; BOUTABA, 2009) introduce two Mixed Integer Programming (MIP) formulations, the second being a relaxed version of the first. Both models use location constraints from virtual nodes to preselect node mappings, which, according to the authors, facilitates the mapping of virtual links. As the relaxed version does not return integer values, it employs rounding techniques to select definitive node mappings and solves the multicommodity flow problem to map virtual links. CPU and bandwidth requirements are considered, and both splittable or unsplittable paths are allowed. Virtual network requests are received and embedded in an online manner, and therefore not known in advance.

Butt et al. (FAROOQ BUTT; CHOWDHURY; BOUTABA, 2010) devise a mechanism for considering different characteristics of substrate nodes in virtual network embeddings. Weights are assigned to substrate nodes according to how "critical" and "popular" they are. A node is considered critical if the failure of this node has the potential of partitioning the substrate network, whereas the popularity of a node is measured as the number of different virtual networks that would be affected by its failure. The authors also present a mechanism that reoptimizes virtual network embeddings by identifying and rearranging virtual networks that contribute to physical resource fragmentation.

Rahman et al. (RAHMAN; AIB; BOUTABA, 2010) develop a heuristic-based approach that considers single substrate link failures in the virtual network embedding process. This approach preemptively calculates alternate paths that are used to reroute virtual links in the event of a physical link failure. A portion of the total available bandwidth of each physical link is used as a pre-reserved quota for backup paths.

Two virtual network embedding approaches are proposed by Alkmim et al.

(ALKMIM; BATISTA; FONSECA, 2013). These approaches combine allocation requirements (such as the ones in previously mentioned formulations) with constraints related to virtual router images. Binary images need to be transferred from a repository to the physical router in which a virtual router will be instantiated. Therefore, the model tries to minimize the time needed to transfer virtual router images while considering CPU, memory, bandwidth and location requirements. Based on the same ILP formulation, the difference between them is the method used to solve the optimization problem. The first employs the traditional branch and cut method provided by CPLEX to traverse a search tree until an optimal solution is found. In contrast, the other limits the search to the root of the search tree, often resulting in a sub-optimal solution but reducing solution time. Virtual network requests are received and handled in an online manner.

Cheng et al. (CHENG et al., 2011) present two algorithms that take advantage of node ranking to select router mappings. Virtual and physical nodes are ranked according to their own capacity and the capacities of their neighbors. For example, the ranking of physical routers is affected not only by its available capacity, and may be increased or decreased according to the available capacities of neighbor routers. Similarly, the ranking of virtual routers and links also takes into consideration the requirements of the neighborhood. The first algorithm sorts virtual and physical routers in non-decreasing order according to their ranks and matches these sorted lists to map them. Links are mapped in a separate stage using either the k-shortest path or the multicommodity flow algorithm, depending on whether path splitting is allowed or not. The second algorithm, in turn, maps routers and links in a single stage. This algorithm builds a breadth-first search tree of virtual nodes sorted by their ranks in non-increasing order. It then attempts to map each virtual node to a physical node that meets all capacity constraints. In case of failure, it is able to backtrack and remap the previous virtual node, in an attempt to solve the issue.

Another ILP formulation is proposed by Davy et al. (DAVY et al., 2011). Unlike previous proposals, this model does not receive a complete network topology as a request. Instead, a request contains the end points that must be interconnected (*i.e.*, a source and one or more destinations). The model builds a virtual network in the form of a tree, spanning from the source to the target locations. Besides location restrictions, this model also takes into consideration the preference of the requester for either lower hosting costs or lower delay (the latter incurring in higher costs). The virtual network is then instantiated, obeying the aforementioned requirements and seeking to minimize costs for the infrastructure provider.

4 PROPOSED SOLUTION

Next, we explain the assumptions behind our proposed solution, and introduce our ILP formulation and heuristic algorithm. In order to represent the scenario of virtual network embedding with a desired level of accuracy, several details were taken into consideration. We envision a scenario in which an infrastructure provider supplies virtual networks to a number of clients. In order to request the creation of a virtual network, these clients sign a Service Level Agreement (SLA) with the infrastructure provider. This SLA describes the characteristics of the requested virtual network and its security requirements, which must be honored by the provider.

We assume that the infrastructure provider will receive a series of virtual network requests over time. Therefore, these requests must be handled in an online manner, *i.e.*, individually as they arrive. If the substrate has sufficient free resources to embed a request, the output of the model indicates the optimal mapping in terms of resource usage, maximizing the amount of free resources available for future requests. If the substrate is not capable of embedding a virtual network due to lack of resources, the request is denied. In practice, we envision that our proposed solution may be used either to automatically handle virtual network requests received by an infrastructure provider (communicating directly with a preexisting virtual network embedding platform) or as an “advisor” (providing candidate mappings to a human operator that may approve, deny, or change such mappings as desired).

4.1 ILP Model

Before presenting our model, we introduce the syntax for our formulation. Capital letters represent sets or variables, and superscripts denote whether a given set or variable refers to physical (P) or virtual (V) entities, or to routers (R) or links (L). Also, each subscript represents an index associated to a variable or path.

Topologies

Virtual network requests must specify the desired topology, *i.e.*, the number of virtual routers in the network and the interconnections between these routers. Physical and virtual network topologies are represented as directed graphs $N = (R, L)$. Each vertex in R denotes a router, and each edge in L denotes a unidirectional link. Bidirectional links are represented as a pair of edges in opposite directions. Each virtual router is mapped to a single physical router, while virtual links may be mapped to either a physical link or a substrate path.

Physical and Virtual Capacities

Physical routers have limited throughput capacity. In other words, routers are able to handle a limited amount of traffic in terms of bits per second. The throughput capacity of a physical router i is expressed by T_i^P. Likewise, $T_{r,i}^V$ denotes the throughput required by virtual router i from virtual network r. As for physical and virtual links, bandwidth limits are represented by $B_{i,j}^P$ and $B_{r,i,j}^V$ respectively, where (i, j) denotes a link, and r, a virtual network. Virtual router and link requirements represent the portion of physical resources that must be allocated for their consumption. We assume that the virtualization architecture is capable of adequately isolating physical resources, enforcing these limits.

Locations

We assume that a majority of clients will request virtual networks that require one or more of its routers to be hosted in specific geographical locations. Therefore, physical routers are associated with location identifiers, stored in set S^P, and virtual network requests may contain location requirements for any number of its routers. Virtual routers that demand to be mapped to a physical router in a specific location are stored in set S^V.

Security

The model allows each virtual network to require varying levels of security. The provision of confidentiality services aims to deal with security concerns related to the shared use of physical routers and links, which may lead to unwanted exposure of sensitive data. Virtual network requesters are able to choose among three distinct confidentiality levels. The first and second levels relate to cryptographic techniques. In the first level, the substrate provides end-to-end cryptography – packets must be encrypted and decrypted at the edges of the network. The second level provides point-to-point cryptography, which requires decrypting and re-encrypting every packet on each hop. Set K_i^P enables the substrate network to indicate which routers support protocol suites that allow cryptographic operations, such as IPSec (KENT; SEO, 2005). Likewise, set $K_{r,i}^V$ indicates whether particular virtual routers require this feature.

In addition to support for cryptographic protocols, our model also considers additional processing and bandwidth costs that arise from the use of cryptographic techniques. Set $W_{r,j}^R$ represents the additional processing cost a virtual router j from network r will demand from the physical router hosting it. This processing cost is modeled as a ratio based on the normal processing cost for a packet that does not require encryption or decryption. As such, the cost is 1.0 if a virtual router does not require cryptographic operations. As the model considers processing overheads for individual virtual routers, this allows virtual networks to request varying cryptographic algorithms and key sizes. It also enables the consideration of different costs depending on the number of cryptographic operations that need to be performed on each packet, as: *(i)* no such operations may be required; *(ii)* only one operation – either encryption or decryption – is needed; or *(iii)* two operations – decryption and re-encryption – may be necessary.

As previously stated, in addition to processing overheads, the model considers bandwidth overheads. Set W_r^L represents the additional bandwidth necessary to encapsulate packets in a network r that requires cryptography (in relation to packets that do not require such encapsulation). As end-to-end and point-to-point cryptography impose different bandwidth overheads, W^L may be set to different values for each network. If no cryptography is required by a given network, this cost is 1.0.

The third and last security level concerns isolation and allows virtual network requesters to indicate other virtual networks that must not share physical resources with their own. Sets of conflicting virtual networks, which are not allowed to share physical routers and links, are stored in set X.

Previous Mappings

As the model handles virtual network requests in an online manner, it is necessary to consider the mappings of virtual routers and links already embedded on the substrate when a new request is received. Sets $E^R_{i,r,j}$ and $E^L_{i,j,r,k,l}$ denote the mappings of previously embedded virtual routers and links, respectively.

The variables of our model indicate where virtual routers and links are mapped on the substrate.

- $A^R_{i,r,j} \in \{0,1\}$ – Router allocation, indicates whether the physical router i is hosting virtual router j from virtual network r.
- $A^L_{i,j,r,k,l} \in \{0,1\}$ – Link allocation, indicates whether the physical link (i,j) is hosting virtual link (k,l) from virtual network r.

Next, we present the objective function (Formula 4.1) and its constraints (C1–C11). The objective function aims at minimizing the bandwidth consumed by embedded virtual networks, while considering overheads introduced by security provisions.

Objective:

$$min \sum_{(i,j)\in L^P} \sum_{r\in N^V,(k,l)\in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l} \tag{4.1}$$

Subject to:

$$\sum_{r\in N^V,j\in R^V} T^V_{r,j} W^R_{r,j} A^R_{i,r,j} \leq T^P_i \qquad \forall i \in R^P \quad \text{(C1)}$$

$$\sum_{j\in R^V} A^R_{i,r,j} \leq 1 \qquad \forall i \in R^P, r \in N^V \quad \text{(C2)}$$

$$\sum_{r\in N^V,(k,l)\in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l} \leq B^P_{i,j} \qquad \forall (i,j) \in L^P \quad \text{(C3)}$$

$$K^V_{r,j} A^R_{i,r,j} \leq K^P_i \qquad \forall i \in R^P, r \in N^V, j \in R^V \quad \text{(C4)}$$

$$\sum_{i \in R^P} A^R_{i,r,j} = 1 \qquad \forall r \in N^V, j \in R^V \quad \text{(C5)}$$

$$\sum_{j \in R^P} A^L_{i,j,r,k,l} - \sum_{j \in R^P} A^L_{j,i,r,k,l} = A^R_{i,r,k} - A^R_{i,r,l}$$
$$\forall r \in N^V, (k,l) \in L^V, i \in R^P \quad \text{(C6)}$$

$$\sum_{q \in N^V, k \in R^V} A^R_{i,q,k} + \sum_{r \in N^V, l \in R^V} A^R_{i,r,l} \leq 1$$
$$\forall q, r \in X, i \in R^P \quad \text{(C7)}$$

$$\left\lceil \frac{\sum_{q \in N^V, (k,l) \in L^V} A^L_{i,j,q,k,l}}{|L^P|} \right\rceil + \left\lceil \frac{\sum_{r \in N^V, (o,p) \in L^V} A^L_{i,j,r,o,p}}{|L^P|} \right\rceil \leq 1$$
$$\forall q, r \in X, (i,j) \in L^P \quad \text{(C8)}$$

$$j A^R_{i,r,k} = l A^R_{i,r,k} \qquad \forall (i,j) \in S^P, r \in N^V, (k,l) \in S^V \quad \text{(C9)}$$

$$A^R_{i,r,j} = E^R_{i,r,j} \qquad \forall (i,r,j) \in E^R \quad \text{(C10)}$$

$$A^L_{i,j,r,k,l} = E^L_{i,j,r,k,l} \qquad \forall (i,j,r,k,l) \in E^L \quad \text{(C11)}$$

Constraint C1 ensures that the maximum throughput capacity of each physical router is not exceeded, considering the throughput requested by virtual routers as well as any overhead costs. Constraint C2 prevents multiple virtual routers from a single virtual network from sharing a physical router. Constraint C3 ensures that bandwidth capacities of physical links will be respected, considering bandwidth overheads in a similar way to constraint C1. Constraint C4 does not allow virtual routers that require cryptographic operations to be mapped to physical routers that do not support such features. Constraint C5 guarantees that each virtual router is mapped to a physical router. Constraint C6 ensures that each virtual link is mapped to a physical path between the routers hosting its source and destination.

Constraint C7 prevents virtual routers from conflicting virtual networks from sharing physical routers. Constraint C8 applies the same restriction to virtual links from conflicting virtual networks, while allowing internal links from each of these networks to share physical routers. C8 is nonlinear, and it was presented in this manner for the sake of comprehension. In practice, C8 was linearized by replacing it with the following 3 constraints (using auxiliary variables $Y, Z \in \{0,1\}$):

$$Y_{q,r,i,j} \geq \frac{\sum_{q \in N^V, (k,l) \in L^V} A^L_{i,j,q,k,l}}{|L^P|}$$
$$\forall q, r \in X, (i,j) \in L^P \quad \text{(C8.1)}$$

$$Z_{q,r,i,j} \geq \frac{\sum_{r \in N^V,(o,p) \in L^V} A^L_{i,j,r,o,p}}{|L^P|} \qquad \forall q,r \in X,(i,j) \in L^P \tag{C8.2}$$

$$Y_{q,r,i,j} + Z_{q,r,i,j} \leq 1 \qquad \forall q,r \in X,(i,j) \in L^P \tag{C8.3}$$

Constraint C9 forces virtual routers with location requirements to be mapped to physical routers in the specified location. Last, constraints C10 and C11 guarantee that the mapping of previously embedded virtual routers and links, respectively, will be maintained.

4.2 Heuristic Algorithm

The proposed heuristic algorithm receives the same inputs and produces the same outputs as the previously described ILP model. Furthermore, generated solutions are bound by the same constraints. However, instead of exploring the entire solution space searching for the optimal mapping, this algorithm employs Simulated Annealing to iteratively generate possible mappings, stopping after a maximum number of cycles is reached or when the solution is close enough to optimality. The adoption of Simulated Annealing was based on the fact that it is a classic metaheuristic method for solving combinatorial optimization problems. Additionally, it is considerably flexible, as its iterative search can be fine-tuned in order to favor higher quality solutions or faster solution times.

Simulated Annealing works by first generating an initial solution, and afterwards generating a similar solution (called a neighbor) in each iteration. If the generated neighbor is better than the current solution according to an evaluation function, the algorithm moves to it. Otherwise, a probability function is used to decide whether the algorithm should move to the new solution. This possibility of moving to a worse solution aims to prevent the method from getting stalled at a local optimum and potentially missing the global optimum. Regardless of the current solution, the best found solution is always stored separately. Algorithm 1 presents a simplified pseudocode version of our annealing-based solution, and its details are explained next.

Function *generateInitialSolution* (line 1) places virtual routers semi-randomly on the substrate, and allocates physical paths between these routers for each virtual link. The details of this function will be explained after the main algorithm is described. The initial solution is then evaluated by function *evaluateSolution* (line 2). This function first checks solution s against the same set of constraints as our ILP model. If s satisfies all constraints, it is evaluated according to the total bandwidth it consumes (Formula 4.2).

$$e \leftarrow \sum_{(i,j) \in L^P} \sum_{r \in N^V,(k,l) \in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l} \tag{4.2}$$

If any constraints are not satisfied, *evaluateSolution* applies a penalty to the evaluation, as shown in Formula 4.3. The penalty is calculated as a function of the number of unsatisfied constraints. If the current solution has unsatisfied constraints, this penalty induces the algorithm to move to solutions with less or no unsatisfied

Algorithm 1 Simulated Annealing

```
1: $s \leftarrow generateInitialSolution$
2: $e \leftarrow evaluateSolution(s)$
3: $sbest \leftarrow s; ebest \leftarrow e$
4: $k \leftarrow 0$
5: while $k < kmax$ and $e > emax$ do
6:    $snew \leftarrow generateNeighbor(s)$
7:    $enew \leftarrow evaluateSolution(snew)$
8:    $t \leftarrow temperature(k, kmax)$
9:    if $probability(e, enew, t) > random[0, 1)$ then
10:       $s \leftarrow snew; e \leftarrow enew$
11:   end if
12:   if $e < ebest$ and $isFeasible(s)$ then
13:       $sbest \leftarrow s; ebest \leftarrow e$
14:   end if
15:   $k \leftarrow k + 1$
16: end while
```

constraints. It also discourages moving to solutions that have more unsatisfied constraints than the current. In this representation, γ is a constant that defines the severity of the applied penalty, and κ is the number of unsatisfied constraints. γ should always be at least greater than 1.0, as otherwise there will be no penalty if a single constraint is not satisfied.

$$e \leftarrow \gamma \ \kappa \sum_{(i,j) \in L^P} \sum_{r \in N^V, (k,l) \in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l} \tag{4.3}$$

The initial solution and its evaluation are then stored as the current best (in *sbest* and *ebest*, respectively – line 3), and k, which represents the current iteration, is initialized as 0 (line 4).

Next, the iterative search for solutions is started (line 5). This iterative process continues until the maximum number of iterations is reached ($k = kmax$) or the evaluation of the best found solution is equal to or better than a desired maximum ($e \leq emax$). In our algorithm, $emax$ represents the maximum desired bandwidth a solution may consume in order to be accepted immediately, and is calculated as shown in Formula 4.4. The formula computes the total bandwidth required by virtual network requests and multiplies it by a constant β, which represents the maximum bandwidth overhead allowed in the mapping process. As any virtual link can be mapped to a path composed of two or more physical links, this overhead is commonly present (and also occurs in the ILP model). β may be adjusted according to the interests of the infrastructure provider, varying from a more conservative scenario in which no exceeding resource consumption is tolerated ($\beta = 1.0$) to more relaxed cases where a certain percentage of overhead is allowed ($\beta > 1.0$). If $emax$ is never reached, the iterative algorithm will continue until $k = kmax$.

$$emax \leftarrow \beta \sum_{r \in N^V, (k,l) \in L^V} B^V_{r,k,l} W^L_r \tag{4.4}$$

In each iteration, a neighbor solution *snew* is generated by applying a small change to the current solution (line 6). Similarly to *generateInitialSolution*, the details of function *generateNeighbor* will be subsequently described. After being generated, the neighbor is then evaluated by function *evaluateSolution*, explained previously (line 7).

After the neighbor solution has been generated and evaluated, the temperature of the current iteration is calculated (line 8). This temperature influences the probability of moving from the current solution to the generated neighbor. As shown in Formula 4.5, it is calculated as a function of k (the current iteration) and $kmax$ (the maximum allowed number of iterations). The value returned by this function starts at 1.0 in the first iteration, and linearly decreases towards zero.

$$temperature(k, kmax) \leftarrow 1 - \frac{k}{kmax} \tag{4.5}$$

Next, the algorithm calculates the probability of moving from current solution s to neighbor $snew$ (line 9). If the evaluation of the neighboring solution is better than the current one, the probability function returns 1.0. In other words, if a newly generated neighbor is better than the current solution in terms of consumed bandwidth, the algorithm will always move to it. Otherwise, the probability is calculated as a function of the ratio between the bandwidth consumed by solutions s and $snew$ (e and $enew$, respectively) and the current temperature t. Our probability calculation, presented in Formula 4.6, is a slightly modified version of the standard calculation used in Simulated Annealing. Lower $e/enew$ ratios influence the probability function positively, as in this case $e/enew - 1$ tends to 0. In other words, there is a greater probability of moving to a worse solution if it is only slightly worse than the current. And, as $e/enew - 1$ tends to -1 if the new solution is significantly worse, the probability will be lower. In a similar way, the probability function is also affected by the current temperature. The temperature is closer to 1 in the beginning of the algorithm, causing a higher chance of moving to worse solutions. As the temperature tends to 0 towards the end of the iterative process, the chances of moving to worse solutions are smaller. This means that the algorithm assumes a riskier behavior at first, gradually becoming more conservative.

$$probability(e, enew, t) \leftarrow exp(\frac{e/enew - 1}{t}) \tag{4.6}$$

As the calculated probability always results in a value between 0 and 1, it is compared with a randomly generated number within the interval $[0, 1)$ in order to decide if the algorithm should move to the new solution. If $probability(e, enew, t) > random[0, 1)$, the move will be made, and current solution s and its evaluation e will be replaced by $snew$ and $enew$, respectively (line 10). Next, if the current solution is feasible and has a better evaluation than the current best, it is stored in $sbest$, and $ebest$ is updated (lines 12 and 13). Finally, in the last step of each iteration, the iteration counter k is incremented by 1 (line 15).

As previously explained, the iterative process will continue until either $kmax$ or $emax$ is reached. At the end of the execution, if the best found solution is feasible (*i.e.*, it satisfies all constraints), it is used to embed the current virtual network. If no feasible solution is found, the virtual network request is denied.

Last, we present in further detail the mechanism of our initial solution and neighbor generating functions. Function *generateInitialSolution*, shown in Algorithm 2, first initializes A^R and A^L with the values from E^R and E^L, respectively (lines 1–6).

This ensures that routers and links from previously embedded networks will remain mapped to the same physical network elements. Next, all routers from the virtual network currently being embedded are mapped to physical routers (lines 7–15). If a virtual router has a location requirement, it is mapped to a random physical router in the desired location (line 10). Otherwise, it is mapped to any randomly selected physical router (line 12). Subsequently, each link (k, l) from the current virtual network is mapped to a physical path between the physical routers where virtual routers k and l are mapped (lines 16–22). The physical path is created using the Dijkstra's algorithm (function *dijkstraShortestPath* in line 19). When running this algorithm, the weight of each physical link is set to the amount of virtual links previously mapped to it plus one, as shown in Formula 4.7. This weight calculation aims to favor the selection of physical paths with greater amounts of free resources.

Algorithm 2 Function generateInitialSolution

```
 1: for $(i, r, j) \in E^R$ do
 2:     $A^R_{i,r,j} \leftarrow E^R_{i,r,j}$
 3: end for
 4: for $(i, j, r, k, l) \in E^L$ do
 5:     $A^L_{i,j,r,k,l} \leftarrow E^L_{i,j,r,k,l}$
 6: end for
 7: $n \leftarrow N^V$ currently being embedded
 8: for $j \in R^V_n$ do
 9:     if $R^V_{n,j} \in S^V$ then
10:         $i \leftarrow$ random $R^P$ in the desired location
11:     else
12:         $i \leftarrow$ random $R^P$
13:     end if
14:     $A^R_{i,n,j} \leftarrow 1$
15: end for
16: for $(k, l) \in L^V_n$ do
17:     $i \leftarrow R^P$ where $R^V_{n,k}$ is mapped
18:     $j \leftarrow R^P$ where $R^V_{n,l}$ is mapped
19:     for $(p, q) \in dijkstraShortestPath(i, j)$ do
20:         $A^L_{p,q,n,k,l} \leftarrow 1$
21:     end for
22: end for
```

$$weight(i, j) \leftarrow 1 + \sum_{r \in N^V, (k,l) \in L^V} E^L_{i,j,r,k,l} \tag{4.7}$$

Function *generateNeighbor*, depicted in Algorithm 3, receives the current solution as a parameter, and generates a neighbor by moving a virtual router in it to a different physical router. First, a router from the virtual network currently being embedded is selected in a random manner, and the mapping of this virtual router is removed (lines 1–4). After this step, all virtual links associated with this router are deallocated from physical links (lines 5–12). The virtual router is then allocated to another randomly selected physical router (respecting location constraints, if present

– lines 13–18), and links associated with it are reconstructed using Dijkstra's algorithm (in order to point to the new physical location of the virtual router – lines 19–27).

Algorithm 3 Function generateNeighbor

```
 1: $n \leftarrow N^V$ currently being embedded
 2: $j \leftarrow$ random $R^V$ from $N^V$ currently being embedded
 3: $i \leftarrow R^P$ where $R^V_{n,j}$ is mapped
 4: $A^R_{i,n,j} \leftarrow 0$
 5: for $(k, l) \in L^V_n$ do
 6:     if $k = j$ or $l = j$ then
 7:         $pLinks \leftarrow$ set of $L^P$ hosting $(k, l)$
 8:         for $(p, q) \in pLinks$ do
 9:             $A^L_{p,q,n,k,l} \leftarrow 0$
10:         end for
11:     end if
12: end for
13: if $R^V_{n,j} \in S^V$ then
14:     $i \leftarrow$ random $R^P$ in the desired location
15: else
16:     $i \leftarrow$ random $R^P$
17: end if
18: $A^R_{i,n,j} \leftarrow 1$
19: for $(k, l) \in L^V_n$ do
20:     if $k = j$ or $l = j$ then
21:         $r \leftarrow R^P$ where $R^V_{n,k}$ is mapped
22:         $s \leftarrow R^P$ where $R^V_{n,l}$ is mapped
23:         for $(p, q) \in dijkstraShortestPath(r, s)$ do
24:             $A^L_{p,q,n,k,l} \leftarrow 1$
25:         end for
26:     end if
27: end for
```

5 EVALUATION

In this chapter, we describe the workloads used for the performance evaluation, and present a detailed comparison between the heuristic-based approach and the optimal model. All experiments were performed in a machine with four AMD Opteron 6276 processors, 64 GB of RAM and Operating System Ubuntu GNU/Linux Server 11.10 x86_64. The heuristic algorithm was implemented in Java, while the ILP model was implemented and run in the CPLEX Optimization Studio (version 12.3).

5.1 Workloads

The workload for each experiment is generated by a simulator developed by the authors, which randomly creates virtual network requests according to a series of parameters. The simulator is run for 250 time slots and generates, in average, 5 requests per slot, following a Poisson distribution. If accepted, requests remain embedded for, in average, 5 time slots before being deallocated, following an exponential distribution. The segmentation of each experiment in time slots, as well as the distributions used for the arrival and duration of requests, are concepts and values adopted from important publications in the area (most notably, from the work accomplished by Yu et al. (YU et al., 2008)).

In all experiments, physical routers have a throughput capacity of 10 Gbps, and link bandwidth is uniformly distributed between 1 and 10 Gbps. 95% of the physical routers support protocols that enable the provision of cryptographic operations. All physical routers are equally distributed among 16 locations.

In virtual network requests, link bandwidth is uniformly distributed between 1 and 5 Gbps. The throughput requested by each virtual router is equal to the bandwidth required by the link with the largest capacity connected to this router. 35% of the requests do not demand any type of cryptography, while 35% require end-to-end cryptography, and the remaining 30%, point-to-point cryptography. Further, 5% of all requests demand that the mapping of its virtual network must not overlap with another network, randomly chosen among currently embedded networks. Each virtual network has two edge routers with randomly generated location requirements. Physical and virtual topologies are generated with BRITE using the Barabási-Albert (BA-2) model (ALBERT; BARABÁSI, 2000).

In addition to the aforementioned fixed parameters, experiments have a number of varying parameters. Experiments were performed with physical networks containing 100 and 500 physical routers. In experiments that use a physical network of size 100, virtual networks range from 2 to 5 virtual routers. For experiments with physical networks of size 500, virtual network sizes vary between 2 and 10.

Another varying parameter is the key size used for cryptographic operations. In the experiments, we considered the AES cryptographic algorithm with 128 and 256-bit key sizes. In addition to being widely used in real environments, AES-128 and AES-256 are considered to provide a substantially high level of security. The algorithm and the key sizes influence the throughput and bandwidth overheads in networks that require end-to-end or point-to-point cryptography. The values for these overheads were set based on the characterization presented by Xenakis et al. (XENAKIS et al., 2006), and will be explained next.

Bandwidth overheads vary depending on whether virtual networks request end-to-end or point-to-point cryptography. Considering packet sizes of 1,536 bytes, the overhead imposed by IPSec encapsulation is of 10.8% for transport mode (end-to-end) and 12.5% for tunnel mode (point-to-point). Therefore, input W^L was set to 1.108 in networks that require end-to-end cryptography, and 1.125 in networks that require point-to-point cryptography.

Throughput overheads depend on the cryptographic algorithm used, key size, router CPU performance, and the number of operations a router needs to perform on each packet. As mentioned previously, the selected combinations of algorithm and key size were AES-128 and AES-256. Throughput overheads for networks requiring end-to-end or point-to-point cryptography were set in line with benchmark results presented by Xenakis et al. (XENAKIS et al., 2006) with routers capable of performing 100 million instructions per second (MIPS). These values are described next.

In networks that require end-to-end cryptography, edge routers need to perform one cryptographic operation on each packet – either encryption or decryption. For such routers, in networks that require AES-128, W^R was set to 1.222, while in networks that require AES-256 it was set to 1.375 (XENAKIS et al., 2006). This is the overhead generated by the decryption operation, which has a higher cost in relation to the encryption operation. We consider the operation which has the highest cost in order to not underestimate this overhead. The remaining routers in such networks do not need to perform any cryptographic operations.

Edge routers in networks that require point-to-point cryptography also need to perform only one cryptographic operation per packet. Therefore, the same overhead values were used as for end-to-end cryptography. However, in order to provide point-to-point cryptography, core routers need to decrypt and reencrypt each packet. For this reason, overhead costs for these routers were set as the aggregated costs of both operations. Namely, W^R was set to 1.222 in networks that require AES-128, as the overhead cost of the encryption operation is negligible, and for the decryption operation it is 22.2%. Further, it was set to 1.532 in networks that require AES-256, as the overhead costs of encryption and decryption are respectively 15.7% and 37.5% (XENAKIS et al., 2006).

All experiments were performed on both the ILP model and the heuristic algorithm. However, when attempting to use the ILP model for experiments with physical networks of size 500, individual requests took several hours to be processed. Therefore, such experiments were canceled after the first time slot, as the ILP model was deemed unfeasible for these workloads. Additionally, the heuristic algorithm was executed 5 times for each workload, and configured with the following parameters:

- $kmax$ = 5,000 (maximum number of iterations);

- $\beta = 3.0$ (maximum bandwidth overhead tolerated in a given solution in order to terminate the iterative process before $kmax$ is reached);

- $\gamma = 100$ (penalty for unsatisfied constraints).

5.2 Results

First, we analyze the average time needed by each approach to reach a solution. Figure 5.1 depicts the aggregated average solution time in each experiment, which encompasses solution times observed from the beginning of the experiments until each time slot. For physical networks with 100 routers (represented in the legend as 100r), the heuristic algorithm takes on average 1.77 seconds considering the AES algorithm with key size of 128 bits, and 1.83 seconds considering AES-256. In contrast, the ILP model takes approximately 2.6 seconds for both cases. Although CPLEX is able to find the optimal solution using the ILP model in a short time for networks of this size, it takes approximately 44.4% longer per request.

Further analysis of Figure 5.1 reveals that, for physical networks with 500 routers, the heuristic algorithm is able to find a solution after approximately 9.06 seconds on average considering AES-128, and 10.57 seconds considering AES-256. In other words, as the physical network size was multiplied by 5 and the maximum virtual network size was multiplied by 2, solution times remained in the order of seconds. This shows that the heuristic algorithm is able to scale to larger physical networks while exhibiting excellent performance. Furthermore, average solution times observed when considering AES-256 with the heuristic algorithm are marginally higher than those obtained when considering AES-128 (3.4% and 16.7% higher for physical networks with 100 and 500 routers, respectively). This is likely due to slightly elevated resource usage, increasing the difficulty of generating valid mappings. The influence of different key sizes on solution times using the ILP model is negligible. Additionally, all aforementioned experiments exhibit lower solution times towards the beginning, as substrate resources are initially 100% free. After a number of time slots, which varies for each experiment, solution times become stable.

As previously stated, the ILP model was considered unfeasible for physical networks with 500 routers, taking several hours to produce an optimal solution after receiving a virtual network request. For this reason, the corresponding experiments were terminated after the average solution time was calculated at the end of the first time slot. While the graph shown in Figure 5.1 was not scaled to accommodate solution times found in such experiments (as doing so would significantly compress the curves related to other experiments), the results obtained in the first time slot are represented in this particular graph. The average solution time in these scenarios was approximately 3 hours and 46 minutes, producing two overlapping vertical lines next to the Y axis.

Next, we analyze the average acceptance rate achieved in each experiment, shown in Figure 5.2. With physical networks of size 100, the ILP model achieves average acceptance rates of 91% and 89.4% considering virtual networks requiring AES-128 and AES-256, respectively. Using the same physical network and virtual network requests generated for the aforementioned experiments, the heuristic algorithm is able to achieve acceptance rates of 66.7% and 66.6%, respectively. As the ILP model produces optimal results in terms of minimal bandwidth usage, it is able to preserve

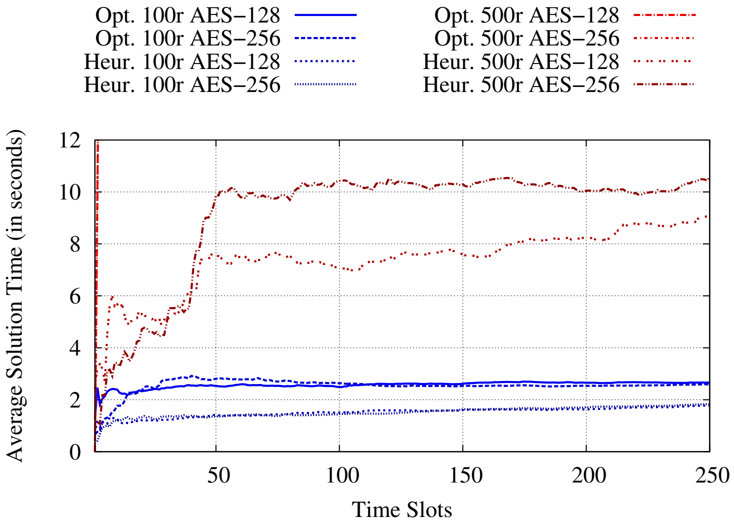


Figure 5.1: Time needed to find the accepted solution in each experiment.

the maximum amount of resources for subsequent allocations, therefore leading to higher acceptance rates. Lower acceptance rates achieved by the heuristic algorithm can also be explained by the fact that its parameterization in the performed experiments is permissive in terms of bandwidth overhead ($\beta = 3.0$). Better results may be achieved by decreasing β or increasing the maximum number of iterations ($kmax$), at the cost of possibly increasing solution time.

The acceptance rate in experiments using physical networks with 500 routers was of 46.2% considering AES-128 and 43.4% considering AES-256. These acceptance rates, which are lower than those observed in other experiments, can be explained by the increased complexity in these scenarios. Although the physical network in these experiments is larger, the maximum size of virtual network requests was also increased from 5 to 10 virtual routers. This significantly increases the amount of resources demanded by virtual networks, causing a lower acceptance rate. Moreover, the number of possible mappings for each network is also significantly increased, likely demanding more iterations in order to find feasible solutions. The use of different key sizes only produces a significant influence in terms of acceptance rate in experiments with 500 physical routers. The use of AES-128 leads to a 6.5% higher acceptance rate in relation to AES-256. Additionally, all depicted experiments exhibit greater variations towards the beginning. The acceptance rate is initially high as the substrate has enough free resources to embed all requests, and starts decreasing once resources become scarce. Eventually, acceptance rates stabilize as previously embedded networks expire, being removed from the substrate and freeing resources for new requests.

In Figure 5.3 we present the ratio between the bandwidth needed to embed each virtual network and the bandwidth requested by such network. A ratio of 1.0 indicates no overhead, which is only observed when each virtual link is mapped to a single physical link (*i.e.*, no virtual links are mapped to paths composed of multiple physical links). As expected, virtual networks with no security requirements generate the least amount of overhead. In experiments with a physical network composed of 100 routers performed with the ILP model (Figures 5.3.a and 5.3.b),

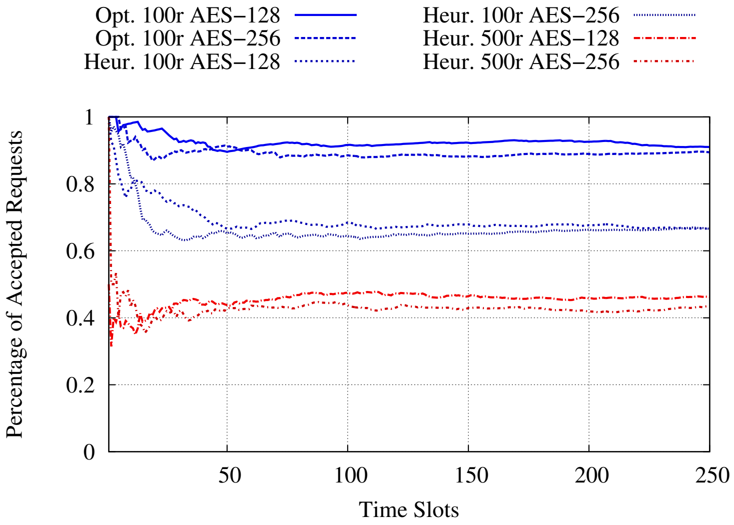


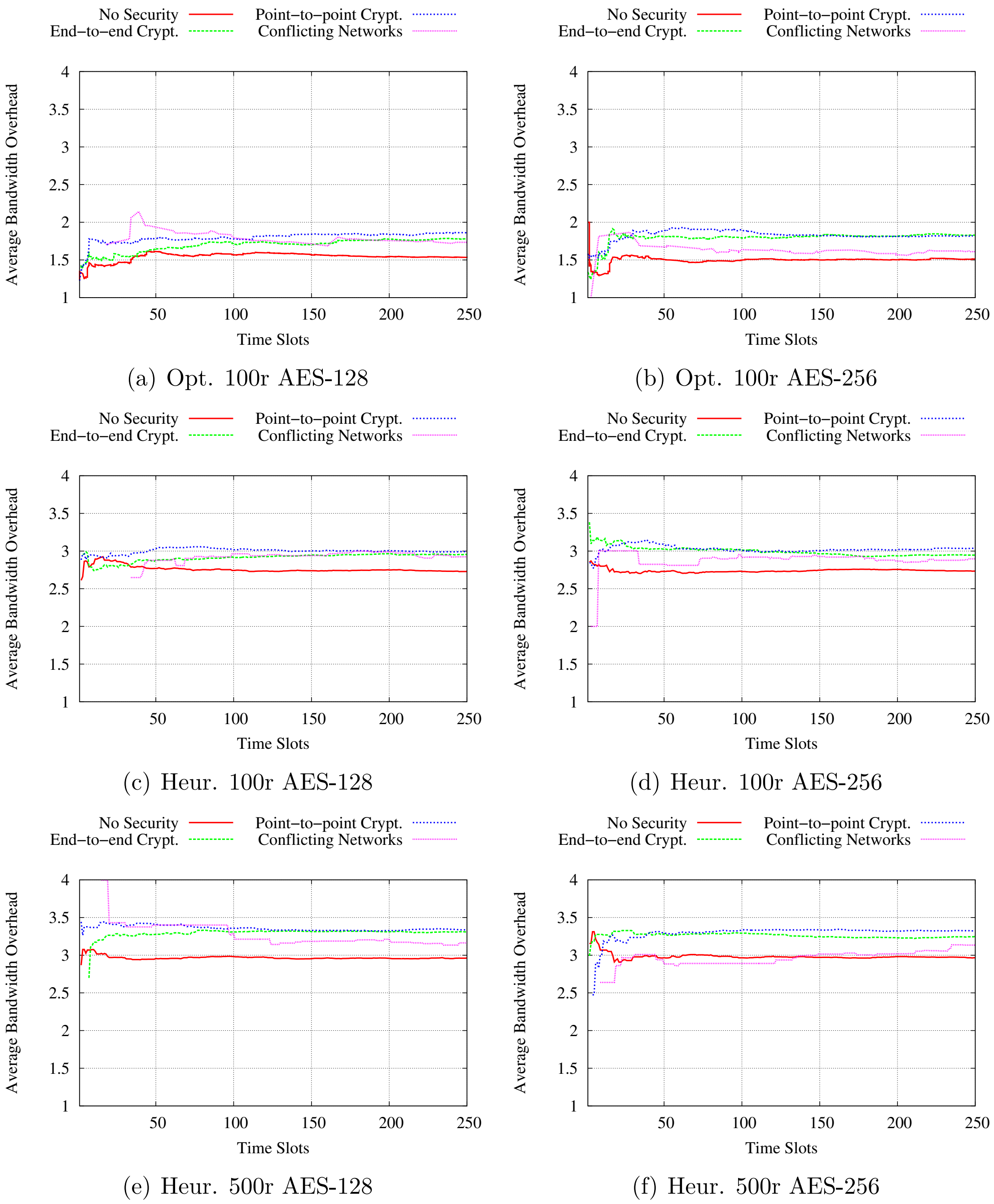
Figure 5.2: Acceptance rate in all completed experiments.

the average ratio is approximately 1.5. Using the same workload with the heuristic algorithm (Figures 5.3.c and 5.3.d), the average ratio is approximately 2.7. In the remaining experiments (Figures 5.3.e and 5.3.f), which use a physical network of size 500 and virtual networks with a maximum of 10 routers, this ratio is approximately 2.96.

The higher bandwidth overhead ratios observed when using the heuristic algorithm can be attributed to the tolerance introduced by constant β, since in our experiments the algorithm stops searching for better solutions when the ratio is less than or equal to 3.0. If an infrastructure provider is able to tolerate solution times in the order of minutes (in contrast to seconds, as observed in our evaluation), β may be set to lower values, potentially leading to ratios which are close to optimality.

The overhead generated by virtual networks requiring end-to-end cryptography considering a physical network of size 100 is approximately 1.80 and 2.95 in experiments performed with the ILP model (Figures 5.3.a and 5.3.b) and the heuristic algorithm (Figures 5.3.c and 5.3.d), respectively. In the remaining experiments (Figures 5.3.e and 5.3.f), the ratio is approximately 3.28. Bandwidth overhead ratios for virtual networks requiring point-to-point cryptography in the aforementioned experiments are 1.84, 3.01, and 3.32, respectively. This shows that the use of stricter security mechanisms generates slightly higher overheads. Meanwhile, the difference in terms of bandwidth overhead generated by different key sizes has not been mentioned, as it was negligible.

As for conflicting networks, despite requiring their virtual routers and links to be mapped on different physical devices, the overhead caused by this constraint is less significant than that caused by other security constraints. In experiments with physical networks of size 100 performed on the ILP model, conflicting networks caused average bandwidth overheads of 1.73 and 1.61. In experiments performed with the heuristic algorithm and 100 physical routers, the averages were 2.92 and 2.89, while with 500 physical routers, averages were 3.17 and 3.13. In all cases, the higher overhead values refer to experiments that consider AES-128 rather than AES-256. While it may seem counterintuitive that overheads were higher in experiments



(a) Opt. 100r AES-128

(b) Opt. 100r AES-256

(c) Heur. 100r AES-128

(d) Heur. 100r AES-256

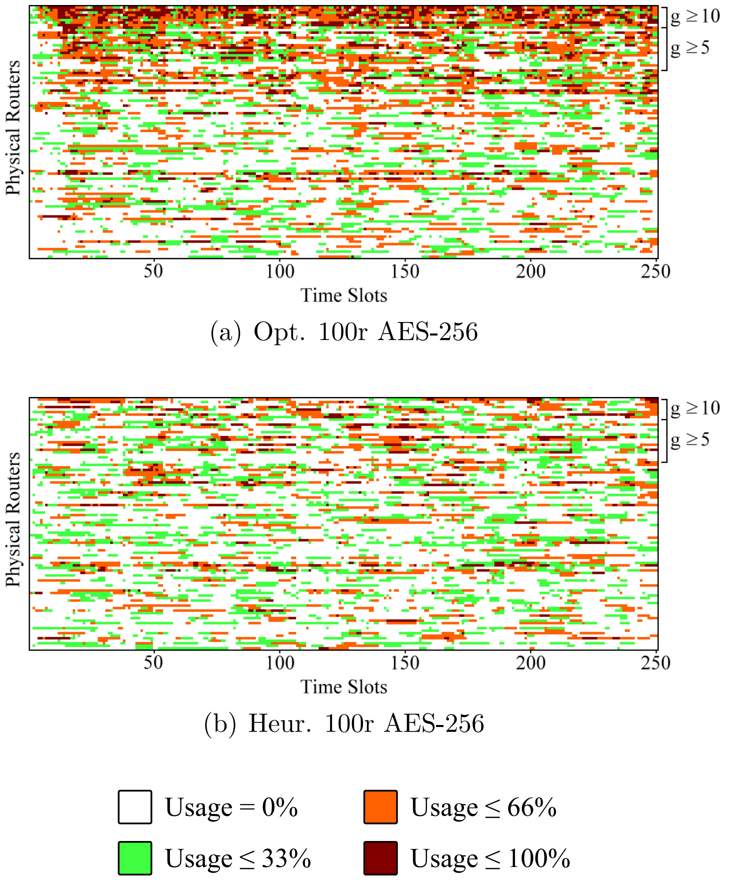
(e) Heur. 500r AES-128

(f) Heur. 500r AES-256

Figure 5.3: Ratio between the bandwidth allocated by the substrate network in order to embed each network and its requested bandwidth. Each graph shows this ratio for each type of request in one experiment.

using the smaller key size, the authors would like to emphasize that this parameter only affects router throughput, and therefore does not affect bandwidth overheads directly. Instead, this is likely related to the fact that experiments performed with AES-128 have a higher acceptance rate, resulting in more virtual networks sharing the same substrate. Additionally, it is noticeable that overheads for this type of request oscillate more during the experiment than others. This happens because networks with this requirement may also demand end-to-end or point-to-point cryptography, which implies security provisions that affect bandwidth overhead in a more significant manner.

Next, we analyze in further detail the characteristics of virtual network mappings generated by the optimal and heuristic methods. Figure 5.4 shows the throughput usage of each router at the end of each time slot in experiments employing a physical network of size 100 and the AES-256 protocol. Routers are represented along the vertical axis, ordered by decreasing connectivity degree (*i.e.*, the amount of bidirectional links connected to each router – represented as g on right side of the figure), from top to bottom. Time slots are represented along the horizontal axis, in increasing order from left to right. When analyzing the results of the ILP model, there is a noticeable trend of higher usage of routers with higher connectivity degree throughout the entire duration of the experiment. However, the same trend is not observed when using the heuristic model. This is due to the fact that the heuristic algorithm employs a load balancing technique when mapping links (by favoring the selection of physical paths with lower usage), whereas the ILP model does not.



(a) Opt. 100r AES-256

(b) Heur. 100r AES-256

Figure 5.4: Throughput usage of each router at the end of each time slot. The vertical axis represents all routers in the network, while the horizontal axis represents time slots.

Last, in order to further analyze the relationship between router usage and connectivity, Figure 5.5 depicts the average throughput usage of physical routers grouped by their connectivity degree. This graph emphasizes the correlation between higher resource usage and connectivity degree observed in results generated by the ILP model. In this experiment, the average throughput usage of routers with connectivity degree between 2 and 8 varies between 12.8% and 32.3%. Meanwhile, the average usage of routers with connectivity degree between 10 and 20 ranges from 35% to 64.2%. In contrast, when analyzing the experiment performed using the heuristic method, these values range from 11.4% to 17.7% and from 12.7% to 24.4%, respectively. While throughput usage still increases with higher connectivity degrees, this growth is not nearly as significant as the one observed with the ILP model. This highlights the efficacy of the heuristic algorithm in distributing load in a more efficient manner, a desirable feature as the overloading of highly connected physical routers may hinder subsequent virtual network mappings. Furthermore, these results also emphasize the impact of topological aspects in the mapping process, as the presence of highly connected routers on the physical infrastructure may favor the instantiation of a greater number of virtual networks.

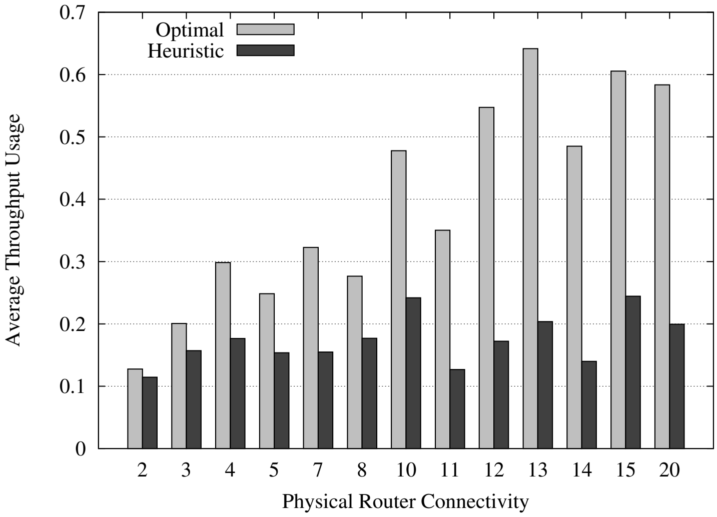


Figure 5.5: Average throughput usage of routers with different connectivity degrees in experiments using a physical network of size 100 and AES-256.

6 CONCLUSIONS

Network virtualization enables the subdivision of a single network infrastructure into multiple virtual architectures. The benefits of this technique apply to a wide range of applications, including the creation of virtual testbeds, community networks, and cloud computing infrastructures. Furthermore, network virtualization has been proposed by researchers as the basis for the creation of a new architecture for the Internet, allowing pluralist network environments that support a number of different network protocols simultaneously.

In spite of its benefits, network virtualization demands a careful balance between optimality and timeliness regarding resource mapping. Additionally, as physical network devices and communication channels are shared among a number of different entities, it is necessary to preserve the confidentiality of each virtual network hosted in such an environment. Nevertheless, we are not aware of previous investigations aimed at reconciling both of these areas, which is of paramount importance in order to fully utilize physical resources without underestimating the capacity requirements of virtual networks.

To tackle the aforementioned issues, we first devised an ILP model capable of optimally embedding virtual networks to physical substrates in an online manner while ensuring that both capacity and security requirements are met, whenever possible. Subsequently, we developed a heuristic algorithm based on Simulated Annealing in order to provide timely, sub-optimal mappings with the ability to scale to larger infrastructures. Both approaches feature precise modeling of overhead costs of security mechanisms, which are taken into consideration if such features are required by virtual network requesters.

Our experiments have shown that the ILP model is able to find optimal solutions in the order of seconds when considering physical networks with up to a hundred routers. However, as it is modeled to solve an NP Hard problem, it does not scale to larger network sizes. Experiments performed with this model revealed that after increasing the physical network size to 500 routers, several hours were needed to map individual virtual network requests. Conversely, the proposed heuristic algorithm is able to find feasible mappings for environments using such large networks while remaining in the order of seconds. Additionally, the heuristic algorithm is flexible, allowing parameterizations that lead to more precise mappings if so desired, at the cost of elevating solution times to the order of minutes. To summarize, while the ILP model is capable of optimally embedding virtual networks on smaller physical infrastructures, the heuristic algorithm is better suited for larger substrate networks, being able to map virtual network requests in a timely manner.

Through this research, we have produced three publications in national and international conferences, two of which are included as appendices to this thesis. The literature search we conducted was used as the basis for a short course published at the 30th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC 2012) (BAYS et al., 2012). This short course explains the fundamentals of virtual networking, describes the main security threats in network virtualization environments as well as state-of-the-art approaches to provide security, and presents some of the main active research projects in this area.

Our first iteration towards the development of an optimal model was published as a paper in the 8th International Conference on Network and Service Manager (CNSM 2012), in the Mini-Conference track (BAYS et al., 2012a) (Appendix A). This paper introduces an ILP model based on an offline version of the virtual network embedding problem, and presents an evaluation that demonstrates the effectiveness of this model, as well as the impact of considering security provisions in the mapping process. A second iteration was published in the 12th Brazilian Symposium on Information and Computer System Security (SBSeg 2012), in the main track (BAYS et al., 2012b) (Appendix B). This paper presents a new ILP model based on the online version of the embedding problem, as well as an evaluation comparing its performance under different workloads and analyzing the impact of security provisions in this version of the problem.

The main contributions of this thesis are threefold. First, through an extensive analysis of the literature, we characterized the state-of-the-art regarding security in network virtualization. Second, we developed an ILP model capable of optimally embedding virtual network requests in an online manner while considering precise overhead costs of security requirements. Last, we devised a heuristic algorithm based on simulated annealing, capable of scaling to larger network infrastructures by providing timely, sub-optimal mappings. The heuristic algorithm has the same features as the ILP model regarding online handling of virtual network requests and modeling of security-related overheads.

We envision three main perspectives for future work. The first is to explore the trade-off between parameters used by the heuristic algorithm (such as the maximum bandwidth overhead allowed and the maximum number of iterations) and performance metrics such as solution time and acceptance rate. Better understanding the influence of these factors has the potential to allow us to improve our method and obtain better results. Another possibility is the analysis of the impact of different types of topologies on the process of virtual network embedding. Certain topological features in physical or virtual networks may improve the utilization of physical resources, potentially increasing profits obtained by infrastructure providers as well as reducing costs for virtual network requesters. The third and last perspective is the inclusion of a reoptimization mechanism. As virtual network requests are handled in an online manner, available physical resources may become increasingly fragmented over time. By migrating previously embedded virtual routers and links in a manner that reduces fragmentation it is possible to increase acceptance rates in the long term, which directly benefits infrastructure providers.

REFERENCES

ALBERT, R.; BARABÁSI, A.-L. Topology of Evolving Networks: local events and universality. **Physical Review Letters**, [S.l.], v.85, p.5234–5237, Dec 2000.

ALKMIM, G. P.; BATISTA, D. M.; FONSECA, N. L. S. Mapping virtual networks onto substrate networks. **Journal of Internet Services and Applications**, [S.l.], v.3, n.4, p.1–15, 2013.

ANDERSEN, D. **Theoretical Approaches to Node Assignment**. Unpublished manuscript. 2002. Disponível em: <http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps>. Acesso em: 30 apr. 2013.

BAYS, L. R. et al. Segurança de Redes Virtuais: fundamentos, tecnologias e tendências. In: **SBRC 2012 – Minicursos Livro Texto**. Ouro Preto, Brasil: SBC, 2012. p.59–98.

BAYS, L. R. et al. Security-aware Optimal Resource Allocation for Virtual Network Embedding. In: INTERNATIONAL CONFERENCE ON NETWORK AND SERVICE MANAGEMENT, 8., Las Vegas, USA. **Proceedings. . .** [S.l.: s.n.], 2012. p.378–384.

BAYS, L. R. et al. Um Modelo para Mapeamento Ótimo de Redes Virtuais com Requisitos de Segurança. In: SIMPÓSIO BRASILEIRO DE SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS, 12., Curitiba, Brasil. **Anais. . .** [S.l.: s.n.], 2012. p.249–262.

BHATIA, S. et al. Trellis: a platform for building flexible, fast virtual networks on commodity hardware. In: ACM CONEXT CONFERENCE, 4., New York, USA. **Proceedings. . .** ACM, 2008. p.72:1–72:6.

CABUK, S. et al. Towards automated provisioning of secure virtualized networks. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 14., New York, USA. **Proceedings. . .** ACM, 2007. p.235–245.

CAVALCANTI, E. et al. Sandboxing for a free-to-join grid with support for secure site-wide storage area. In: INTERNATIONAL WORKSHOP ON VIRTUALIZATION TECHNOLOGY IN DISTRIBUTED COMPUTING, 2., Washington, USA. **Proceedings. . .** IEEE Computer Society, 2006. p.11–.

CHENG, X. et al. Virtual network embedding through topology-aware node ranking. **SIGCOMM Computer Communication Review**, New York, NY, USA, v.41, n.2, p.38–47, Apr. 2011.

CHOWDHURY, N. M. M. K.; ZAHEER, F.-E.; BOUTABA, R. iMark: an identity management framework for network virtualization environment. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 11., Piscataway, USA. **Proceedings...** IEEE Press, 2009. p.335–342.

CHOWDHURY, N.; RAHMAN, M.; BOUTABA, R. Virtual Network Embedding with Coordinated Node and Link Mapping. In: IEEE CONFERENCE ON COMPUTER COMMUNICATIONS, 28., Rio de Janeiro, Brasil. **Proceedings...** [S.l.: s.n.], 2009. p.783 –791.

CLEEFF, A. van; PIETERS, W.; WIERINGA, R. J. Security Implications of Virtualization: a literature study. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND ENGINEERING, 3., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2009. p.353–358.

CUI, Q.; SHI, W.; WANG, Y. Design and Implementation of a Network Supporting Environment for Virtual Experimental Platforms. In: WRI INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND MOBILE COMPUTING, 3., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2009. p.406–412.

DAVY, S. et al. Policy-assisted planning and deployment of virtual networks. In: INTERNATIONAL CONFERENCE ON NETWORK AND SERVICE MANAGEMENT, 7., Paris, France. **Proceedings...** [S.l.: s.n.], 2011. p.1–8.

EL-DARIEBY, M.; ROLIA, J. Hierarchical Creation of Virtual Networks. In: IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, 10., Vancouver, Canada. **Proceedings...** [S.l.: s.n.], 2006. p.220–229.

FARIAS, F. N. N. et al. Pesquisa Experimental para a Internet do Futuro: uma proposta utilizando virtualização e o framework openflow. In: **SBRC 2011 – Minicursos Livro Texto**. Campo Grande, Brasil: SBC, 2011. p.1–61.

FAROOQ BUTT, N.; CHOWDHURY, M.; BOUTABA, R. Topology-Awareness and Reoptimization Mechanism for Virtual Network Embedding. In: INTERNATIONAL IFIP TC 6 NETWORKING CONFERENCE, 9., Chennai, India. **Proceedings...** Springer Berlin Heidelberg, 2010. v.6091, p.27–39.

FERNANDES, N. C.; DUARTE, O. C. M. B. XNetMon: a network monitor for securing virtual networks. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 10., Kyoto, Japan. **Proceedings...** IEEE, 2011.

FERNANDES, N. C.; DUARTE, O. C. M. B. Provendo Isolamento e Qualidade de Serviço em Redes Virtuais. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 29., Campo Grande, Brazil. **Anais...** [S.l.: s.n.], 2011.

FERNANDES, N. et al. Virtual networks: isolation, performance, and trends. **Annals of Telecommunications**, [S.l.], v.66, n.5-6, p.339–355, 2011.

GOVINDAN, S. et al. Xen and Co.: communication-aware cpu management in consolidated xen-based hosting platforms. **IEEE Transactions on Computers**, [S.l.], v.58, n.8, p.1111–1125, aug. 2009.

HUANG, D.; ATA, S.; MEDHI, D. Establishing Secure Virtual Trust Routing and Provisioning Domains for Future Internet. In: IEEE CONFERENCE ON GLOBAL TELECOMMUNICATIONS, 29., Miami, USA. **Proceedings. . .** [S.l.: s.n.], 2010. p.1–6.

KENT, S.; SEO, K. **RFC 4301**: security architecture for the internet protocol. 2005. Disponível em: <http://www.ietf.org/rfc/rfc4301.txt>. Acesso em: 30 apr. 2013.

KOKKU, R. et al. NVS: a virtualization substrate for wimax networks. In: ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 16., New York, USA. **Proceedings. . .** ACM, 2010. p.233–244.

LAN/MAN STANDARDS COMMITTEE. **IEEE Standard for Local and metropolitan area networks – Virtual Bridged Local Area Networks**. 2006.

MARQUEZAN, C. et al. Distributed autonomic resource management for network virtualization. In: IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, 12., Osaka, Japan. **Proceedings. . .** [S.l.: s.n.], 2010. p.463–470.

MAZZARIELLO, C.; BIFULCO, R.; CANONICO, R. Integrating a network IDS into an open source Cloud Computing environment. In: INTERNATIONAL CONFERENCE ON INFORMATION ASSURANCE AND SECURITY, 6., Atlanta, USA. **Proceedings. . .** [S.l.: s.n.], 2010. p.265–270.

MCKEOWN, N. et al. OpenFlow: enabling innovation in campus networks. **SIGCOMM Computer Communication Review**, New York, USA, v.38, p.69–74, March 2008.

RAHMAN, M.; AIB, I.; BOUTABA, R. Survivable Virtual Network Embedding. In: INTERNATIONAL IFIP TC 6 NETWORKING CONFERENCE, 9., Chennai, India. **Proceedings. . .** Springer Berlin Heidelberg, 2010. v.6091, p.40–52.

ROSCHKE, S.; CHENG, F.; MEINEL, C. Intrusion Detection in the Cloud. In: IEEE INTERNATIONAL CONFERENCE ON DEPENDABLE, AUTONOMIC AND SECURE COMPUTING, 8., Washington, DC, USA. **Proceedings. . .** IEEE Computer Society, 2009. p.729–734.

ROSEN, E. et al. **RFC 4364**: bgp/mpls ip virtual private networks (vpns). 2006. Disponível em: <http://www.ietf.org/rfc/rfc4364.txt>. Acesso em: 30 apr. 2013.

SHIREY, R. **RFC 2828**: internet security glossary. 2000. Disponível em: <http://www.ietf.org/rfc/rfc2828.txt>. Acesso em: 30 apr. 2013.

STALLINGS, W. **Cryptography and network security**: principles and practice. 4th.ed. [S.l.]: Pearson/Prentice Hall, 2006.

WOLINSKY, D. I. et al. On the Design of Virtual Machine Sandboxes for Distributed Computing in Wide-area Overlays of Virtual Workstations. In: INTERNATIONAL WORKSHOP ON VIRTUALIZATION TECHNOLOGY IN DISTRIBUTED COMPUTING, 2., Washington, DC, USA. **Proceedings. . .** IEEE Computer Society, 2006. p.8–.

WU, Q.; SHANBHAG, S.; WOLF, T. Fair multithreading on packet processors for scalable network virtualization. In: ACM/IEEE SYMPOSIUM ON ARCHITECTURES FOR NETWORKING AND COMMUNICATIONS SYSTEMS, 6., New York, USA. **Proceedings...** ACM, 2010. p.1:1–1:11.

XENAKIS, C. et al. A generic characterization of the overheads imposed by IPsec and associated cryptographic algorithms. **Computer Networks**, New York, USA, v.50, n.17, p.3225–3241, Dec. 2006.

YEOW, W.-L.; WESTPHAL, C.; KOZAT, U. C. Designing and embedding reliable virtual infrastructures. **SIGCOMM Computer Communication Review**, New York, NY, USA, v.41, n.2, p.57–64, Apr. 2011.

YU, M. et al. Rethinking virtual network embedding: substrate support for path splitting and migration. **SIGCOMM Computer Communication Review**, New York, USA, v.38, n.2, p.17–29, Mar. 2008.

YU, S.; ZHOU, W. Entropy-Based Collaborative Detection of DDOS Attacks on Community Networks. In: ANNUAL IEEE INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS, 6., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2008. p.566–571.

ZHANG, Y.; GAO, L.; WANG, C. MultiNet: multiple virtual networks for a reliable live streaming service. In: IEEE CONFERENCE ON GLOBAL TELECOMMUNICATIONS, 28., Piscataway, USA. **Proceedings...** IEEE Press, 2009. p.2254–2259.

APPENDIX A – PUBLISHED PAPER – CNSM 2012

- Title: Security-aware Optimal Resource Allocation for Virtual Network Embedding
- Conference: 8th International Conference on Network and Service Management (CNSM 2012) – Mini-Conference
- URL: http://www.cnsm-conf.org/2012/
- Date: 22–26 October 2012
- Venue: Las Vegas, Nevada, USA

Security-aware Optimal Resource Allocation for Virtual Network Embedding

Leonardo Richter Bays, Rodrigo Ruas Oliveira, Luciana Salete Buriol,
Marinho Pilla Barcellos, Luciano Paschoal Gaspary
Institute of Informatics
Federal University of Rio Grande do Sul (UFRGS)
{lrbays,ruas.oliveira,buriol,marinho,paschoal}@inf.ufrgs.br

***Abstract*—Network virtualization enables the creation of multiple instances of virtual networks on top of a single physical infrastructure. Given its wide applicability, this technique has attracted a lot of interest both from academic researchers and major companies within the segment of computer networks. Although recent efforts (motivated mainly by the search for mechanisms to evaluate Future Internet proposals) have contributed substantially to materialize this concept, none of them has attempted to combine efficient resource allocation with fulfillment of security requirements (e.g., confidentiality). It is important to note that, in the context of virtual networks, the protection of shared network infrastructures constitutes a fundamental condition to enable its use in large scale. To address this problem, in this paper we propose a virtual network embedding model that satisfies security requirements and, at the same time, optimizes physical resource usage. The results obtained demonstrate that the model is able to correctly and optimally map virtual networks to a physical substrate, minimizing bandwidth costs for infrastructure providers.**

I. Introduction

In recent years, there has been a growing demand for adaptive network services with increasingly distinct requirements. Driven by such demands, and stimulated by the successful employment of virtualization for hosting custom-built servers, researchers have started to explore the use of this technique in network infrastructures. Network virtualization enables the creation of virtual topologies on top of physical substrates. This is made possible by instantiating one or more virtual routers on physical devices and establishing virtual links between these routers, forming topologies that are not limited by the structure of the physical network.

Virtual networks allow the creation of infrastructures that are specifically tailored to the needs of distinct network applications [1]. Furthermore, virtual networks can be used as testbeds, creating favorable environments for the development and evaluation of new architectures and protocols [2]. Network virtualization has been embraced by the Industry as well. Important companies nowadays offer network devices supporting virtualization, and this new functionality allowed infrastructure providers to offer new services.

Despite its wide applicability, maintaining a network virtualization environment requires adequate resource allocation. On one side there are infrastructure providers, which aim to increase their revenue by hosting the highest possible number of virtual networks while minimizing their costs. On the other, there are a number of clients who request virtual networks with specific resource demands. The resource allocation method needs to guarantee that the requested resources will be available for each of these clients, while attempting to minimize the infrastructure provider's costs. Additionally, the result of the mapping process needs to be delivered in an acceptable time frame.

A second major concern that arises from the shared use of routing devices and communication channels is security. Without adequate protection, users from a virtual network may be able to capture data or even tamper with traffic belonging to other virtual networks on the same substrate. Such actions would violate security properties such as confidentiality and integrity. Therefore, it is of great importance that virtualization architectures offer protection against these and other threats that might compromise their security.

In order to enable the use of virtualization in real environments, both efficient resource allocation and security provisions must be taken into consideration. The resource allocation problem is known to be NP-hard [3], and has commonly been approached in the literature with resource embedding algorithms modeled by means of linear programming. There exists a body of work on the optimal allocation of resources in the network embedding problem [4]–[8]. However, to the best of our knowledge, previous work has not taken into consideration security requirements. To cover this gap, in the present paper we propose a virtual network embedding model that optimizes physical resource usage while meeting security requirements whenever feasible. In addition to capacity and location constraints, clients requesting virtual networks are able to specify security requirements for their networks, which must be honored by the infrastructure provider. The proposed model determines the best possible mapping in terms of resource usage, while taking all security requirements into consideration.

The remainder of this paper is organized as follows. Section II presents related work from the areas of resource mapping and virtual network security. In Section III we introduce our proposed solution, explaining the theory behind it and presenting its formulation. Section IV outlines the performed evaluation and presents the obtained results. Last, Section V presents final remarks and perspectives for future work.

II. Related Work

In this section, we present the related work focusing on virtual network embedding, as well as some of the main proposals for securing virtual networks. We present a brief summary of each proposal, highlighting its distinctive features.

Yu et al. [4] devise a virtual network embedding model with support for path splitting and migration. The algorithm proposed to accomplish this objective takes advantage of the flexibility gained by splitting virtual links over multiple substrate paths in order to reduce the time needed to complete the mapping process. Additionally, the substrate is able to periodically re-optimize its resource usage by migrating already established virtual routers and links. The model considers that virtual network requests are not known in advance, and takes into account CPU and bandwidth requirements, as well as the maximum amount of time a request can wait before being served.

Another model, formulated by Chowdhury et al. [5], aims to provide better coordination between router and link allocation, which are performed in two separate steps. Router mappings are preselected in a way that assists the subsequent stage of link mapping. As the previous model, it allows path splitting and considers CPU and bandwidth requirements. Virtual network requests are received and allocated *online*, and are able to specify explicit locations in which certain virtual routers must be mapped.

The model designed by Alkmim et al. [6] extends previous work by combining allocation requirements with constraints related to virtual router images. Images need to be transferred from a repository to the physical router in which a virtual router will be instantiated. Therefore, the model tries to minimize the time needed to transfer virtual router images while considering CPU, memory, bandwidth, and location requirements. This model also receives and handles virtual network requests *online*.

Cheng et al. [7] propose a *node ranking*-based approach that considers not only the capacity of routers and links, but also the capacities of those in its immediate neighborhood. For example, the ranking of physical routers is affected not only by its available capacity, and may be increased or decreased according to the available capacities of neighbor routers. Similarly, the ranking of virtual routers and links also takes into consideration the requirements of the neighborhood. The mapping process allocates virtual routers and links to physical elements with similar rankings. According to the authors, this strategy tends to reduce potential bottlenecks.

Unlike previous proposals, the model presented by Davy et al. [8] does not receive a complete network topology as a request. Instead, a request contains the end points that must be interconnected (a source and one or more destinations), and the solution builds a virtual network that satisfies the demand. This virtual network is built in the form of a tree, spanning from the source to the target locations. Besides location restrictions, this model also takes into consideration the requester's preference for either a lower-cost network or a higher-cost, lower delay network.

Aside from resource allocation, the network virtualization environment needs to provide correct data isolation. Cabuk et al. [9] devise a framework to provide secure virtual networks. This framework employs the use of Trusted Virtual Domains (TVDs) in order to offer access control, confidentiality, and integrity to network communications. Each TVD represents an isolated domain, composed of virtual entities and the links between them. Digital certificates are used in order to assure that only entities that satisfy a given set of conditions are able to join a TVD. The authors use VLANs to isolate the traffic within a trusted network, and VPNs to interconnect such networks.

Huang et al. [10], on their turn, propose a scheme that uses cryptography to protect routing information and variable paths to mitigate traffic analysis attacks. The scheme classifies routers into groups and distribute group keys for each of these routers. This way, only routers within a certain group can access the corresponding information. Furthermore, each virtual link is mapped onto a set of physical paths. Before sending traffic, routers select an arbitrary path to hinder traffic analysis.

To the best of our knowledge, existing approaches on the problem of virtual network embedding do not consider security requirements. Meanwhile, there are a number of publications that focus on offering network virtualization environments with specific security provisions. Both of these aspects are major factors in enabling the use of virtual networks in real environments. Therefore, our proposed solution aims at optimizing the mapping of virtual networks on physical resources while guaranteeing the fulfillment of security requirements.

III. Proposed Solution

In an attempt to address the problem of optimizing resource usage while fulfilling security requirements, we have modeled our solution by means of Integer Linear Programming (ILP). In order to create a mathematical model that represents the scenario of virtual network embedding with a desired level of accuracy, several details were taken into consideration. We envision a scenario in which an infrastructure provider supplies virtual networks to a number of clients. In order to request the creation of a virtual network, these clients sign a Service Level Agreement (SLA) with the infrastructure provider. This SLA describes the characteristics of the requested virtual network and its security requirements, which must be honored by the provider.

Before presenting our model, we introduce the syntax for our formulation. We use capital letters to represent sets or variables. Each superscript denotes if a given set is virtual (V) or physical (P). Also, each subscript represents an index associated to a variable or path.

Virtual network requests must specify the desired topology, *i.e.*, the number of virtual routers in the network and the interconnections between these routers. We represent each network topology, physical or virtual, as a directed graph $N = (R, L)$, where each vertex set R denotes the routers in the network. Similarly, each edge set L denotes the links on this network. Additionally, a link between two routers a and b is represented by a pair of symmetrical edges with opposite directions (a,b) and (b,a). Moreover, while a virtual router will be mapped to exactly one physical router, virtual links can be mapped to either a single physical link, or to a path composed of a series of physical links.

Virtual routers and links, when mapped, consume a portion of the available resources on the physical substrate. Therefore, each element in the physical network has a set of capacities associated with it, representing physical limitations. Physical routers have limited CPU and memory capacities, expressed by C_i^P and M_i^P respectively (where i is the index of the router). In addition, each link has limited bandwidth capacity $B_{i,j}^P$, where the pair i, j represents a physical link between i and j. Similarly, $C_{n,i}^V$, $M_{n,i}^V$ and $B_{n,i,j}^V$ represent the CPU, memory, and bandwidth requirements for each virtual network n. For virtual routers, these requirements indicate how much CPU and memory will be consumed by it, while for virtual links, they indicate how much bandwidth must be allocated in the physical paths to which they will be mapped.

We believe that clients will likely request virtual networks to provide connectivity between two or more geographical locations. For this reason, each physical router is also associated with a location identifier S^P (S represents *site* – this notation was chosen to avoid confusion with L, the set of links). Virtual network requests may or may not require that a number of its routers be mapped to physical routers on certain locations. Such requirements are represented by the set S^V.

Our model also allows each virtual network request to have a set of security requirements associated with it. These security requirements, if present, aim to provide one of three distinct levels of confidentiality to communications within these networks:

- End-to-end cryptography: If this level of confidentiality is requested, the *end points* of a virtual network must be mapped to physical routers that are able to provide this feature. In practice, this means that these end points must support protocol suites such as IPSec [11], which provides end-to-end cryptography when used in *transport mode*.
- Point-to-point cryptography: In this level of confidentiality, packets are encrypted in their entirety, protecting not only their payload but also the header. This means that packets need to be decrypted and reencrypted on each hop in order to be properly routed. Therefore, every router in a virtual network that requests this level of confidentiality must be mapped to a physical router which is capable of supporting such operations. This level corresponds to the *tunnel mode* in IPSec, meaning that physical routers that support this protocol are able to provide this feature.
- Non-overlapping networks: A virtual network request may also demand that its virtual routers and links do not share any physical routers or paths with one or more other virtual networks. This is an extreme case that may be used, for example, to protect highly sensitive information from competitors.

In order to provide the first two levels of security, virtual network requests must be able to indicate which, if any, of its routers must be able to encrypt and decrypt network packets. Therefore, the model also incorporates sets K_i^P and $K_{n,i}^V$, which indicate whether a physical router is capable of providing this feature, and whether virtual routers demand it.

As for the third level, requests must be able to specify other virtual networks which are not allowed to share the same substrate routers and links. To provide this level, we use the X set. This set is composed of pairs of virtual networks that must not share the same substrate resources (*i.e.*, if $(i, j) \in X$, then virtual networks i and j must not share resources).

Next, we present the output variables of our proposed solution. The values returned by these variables indicate the allocation of virtual elements on the physical substrate, representing the solution to the problem. After the problem is solved, each virtual router will be mapped to a single physical router, and each virtual link will be mapped to a path on the physical substrate. This path may be equivalent to a single physical link, or to a series of sequential physical links.

- $A_{i,n,j}^R \in \{0, 1\}$ – Router Allocation: Indicates whether the physical router *i* is hosting virtual router *j* from virtual network *n*.
- $A_{i,j,n,k,l}^L \in \{0, 1\}$ – Link Allocation: Indicates whether the physical link *(i,j)* is hosting virtual link *(k,l)* from virtual network *n*.

Last, we present the objective function of our model and its constraints. The objective function aims to minimize the physical bandwidth consumed by virtual links in virtual network requests, thus minimizing cost and preserving bandwidth for future allocations. Meanwhile, the constraints ensure that all requirements will be met, and that physical capacities will not be exceeded.

Objective:

$$min \sum_{(i,j)\in L^P} \sum_{n\in N^V,(k,l)\in L^V} A_{i,j,n,k,l}^L B_{n,k,l}^V$$

Subject to:

$$\sum_{n\in N^V, j\in R^V} C_{n,j}^V A_{i,n,j}^R \le C_i^P \qquad \forall i \in R^P \quad \text{(C1)}$$

$$\sum_{n\in N^V, j\in R^V} M_{n,j}^V A_{i,n,j}^R \le M_i^P \qquad \forall i \in R^P \quad \text{(C2)}$$

$$\sum_{n\in N^V,(k,l)\in L^V} B_{n,k,l}^V A_{i,j,n,k,l}^L \le B_{i,j}^P \qquad \forall (i,j) \in L^P \quad \text{(C3)}$$

$$K_{n,j}^V A_{i,n,j}^R \le K_i^P \qquad \forall i \in R^P, n \in N^V, j \in R^V \quad \text{(C4)}$$

$$\sum_{i\in R^P} A_{i,n,j}^R = 1 \qquad \forall n \in N^V, j \in R^V \quad \text{(C5)}$$

$$\sum_{j\in R^P} A_{i,j,n,k,l}^L - \sum_{j\in R^P} A_{j,i,n,k,l}^L = A_{i,n,k}^R - A_{i,n,l}^R$$
$$\forall n \in N^V, (k,l) \in L^V, i \in R^P \quad \text{(C6)}$$

$$\sum_{m\in N^V,k\in R^V} A_{i,m,k}^R + \sum_{n\in N^V,l\in R^V} A_{i,n,l}^R \le 1$$
$$\forall m, n \in X, i \in R^P \quad \text{(C7)}$$

$$\sum_{m\in N^V,(k,l)\in L^V} A_{i,j,m,k,l}^L + \sum_{n\in N^V,(o,p)\in L^V} A_{i,j,n,o,p}^L \le 1$$
$$\forall m, n \in X, (i,j) \in L^P \quad \text{(C8)}$$

$$jA_{i,n,k}^R = lA_{i,n,k}^R \qquad \forall (i,j) \in S^P, n \in N^V, (k,l) \in S^V \quad \text{(C9)}$$

The first three constraints ensure that the capacity requirements of virtual routers and links will be met. Constraint C1 ensures that the CPU usage required by virtual routers mapped to a physical router will not exceed its maximum CPU capacity. Constraint C2 applies the same restriction to the memory capacity of physical routers, and constraint C3, to the bandwidth capacity of physical links.

Constraint C4 ensures that all virtual routers that must perform encryption and decryption of packets will be mapped to physical routers that support these operations. This is the case for edge routers in virtual networks that request end-to-end cryptography, or all routers in virtual networks that require point-to-point cryptography.

Constraint C5 guarantees that each virtual router will be mapped to a physical router. In a complementary way, constraint C6 ensures that the path formed by the set of physical links hosting a virtual link will be valid. For any virtual link *(a,b)*, C6 guarantees the creation of a path between *a* and *b* on the physical substrate. This happens because for a link *(a,b)*, the right side of the equation will be 1 and -1 for *a* and *b*, respectively. That is, *a* will have an outgoing link and *b* will have an incoming link. Since for all other nodes the right side of the equation is 0, arcs will be inserted in the solution completing a path between *a* and *b*.

Constraints C7 and C8 refer to pairs of conflicting virtual networks – *i.e.*, virtual networks that must not share any physical resources. Constraint C7 does not allow virtual routers that belong to conflicting virtual networks to be mapped to the same physical routers. Likewise, constraint C8 guarantees that virtual links belonging to these conflicting networks will not share any physical paths[1]. Finally, constraint C9 ensures that each virtual router that has a location requirement will be mapped to a physical router at that specific location.

IV. Performance Evaluation

In order to evaluate the performance of our proposed solution, our model was implemented and run in the CPLEX Optimization Studio. Using a number of varying workloads as inputs, we were able to measure the time needed to solve the problem under a series of different conditions.

All experiments were performed in a machine with four AMD Opteron 6276 processors running at 2.3 GHz, using a maximum of four threads. The machine is also equipped with 64 GB of RAM, and its operating system is Ubuntu GNU/Linux Server 11.10 x86_64.

A. Workloads

Similarly to previous work [4]–[7], physical and virtual topologies were randomly generated. In order to create these topologies, we used the BRITE topology generator [12] with the Barabási-Albert (BA-2) model [13].

Table I summarizes the 24 experiments that were performed. In the experiments, four different factors were used: virtual router capacity requirements (*i.e.*, CPU and memory), virtual link bandwidth requirements, physical network size, and the total number of virtual routers in virtual networks requests. Physical routers initially have 100% free CPU and 256 MB of memory. Physical links have available bandwidth uniformly distributed between 1 and 10 Gbps. Experiments were designed as a full factorial, exploring all possible combinations between the aforementioned factors and their levels. For ease of reference, these 24 experiments were divided into four groups (1–4) in which we vary the CPU, memory, and bandwidth requirements, each with six experiments (A–F), in which we vary the size of the physical network and the aggregated number of virtual routers in virtual network requests.

In addition to the aforementioned characteristics of the physical network (CPU and memory capacities of physical routers and physical link bandwidth), 95% of all routers in physical networks support protocols that allow the encryption and decryption of packets. Furthermore, physical routers are equally distributed among 16 geographical locations.

Virtual network requests contain 2 to 5 virtual routers connected by virtual links following the previously mentioned BA-2 topology model. The resource requirements of virtual routers and links are uniformly distributed with the values presented in Table I (for example, in experiment 1C, virtual routers have CPU requirements of either 10, 20, or 30%). With respect to location requirements, all virtual network requests have two virtual edge routers. These routers must be mapped to physical routers in specific geographical locations (chosen at random). Finally, security requirements present four possibilities:

- No security: a number of virtual network requests, adding up to 35% of the virtual routers to be mapped in each experiment, have no security requirements.
- End-to-end cryptography: a number of virtual network requests, adding up to another 35% of all virtual routers, require that their edge routers must support encryption and decryption of packets.
- Point-to-point cryptography: virtual network requests that require this level of confidentiality, where every router must support encryption and decryption, add up to 20% of virtual routers in each experiment.
- Non-overlapping networks: A smaller number of virtual network requests, adding up to the last 10% of virtual routers, require that their entire network do not share physical routers and links with other two virtual networks (chosen at random).

In each scenario, all virtual network requests are known in advance. Therefore, all requests are mapped to the physical substrate simultaneously.

B. Results

To quantify the effectiveness of the proposed model, we measure the overall resource consumption, the resource load on physical routers and links, the impact of security requirements, and the time needed to find optimal mappings. For ease of comprehension, consider that all experiments achieve optimal results. We will discuss such consideration when evaluating running times.

In Figure 1, we present the total bandwidth consumed by virtual networks in each experiment. Results obtained for CPU

[1] As a side effect, C8 also does not allow virtual links from any network in the conflicts set to share physical links. We intend to improve this constraint in future work in order to eliminate this behavior.

Experiments	1A	1B	1C	1D	1E	1F	2A	2B	2C	2D	2E	2F	3A	3B	3C	3D	3E	3F	4A	4B	4C	4D	4E	4F
Bandwidth Req.	Uniformly distributed between 100 Mbps and 3 Gbps												Uniformly distributed between 100 Mbps and 5 Gbps											
CPU Req.	10, 20, or 30%						10, 20, 30, 40, or 50%						10, 20, or 30%						10, 20, 30, 40, or 50%					
Memory Req.	32, 64, or 80 MB						32, 64, 80, 96, or 128 MB						32, 64, or 80 MB						32, 64, 80, 96, or 128 MB					
Phys. Routers	50			100			50			100			50			100			50			100		
Virt. Routers	17	25	33	33	50	66	17	25	33	33	50	66	17	25	33	33	50	66	17	25	33	33	50	66

TABLE I

Workload used in each experiment performed for the evaluation of our model.

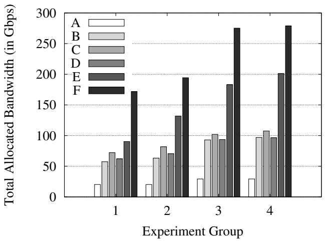


Fig. 1. Total amount of bandwidth consumed by the optimal allocation in each scenario.

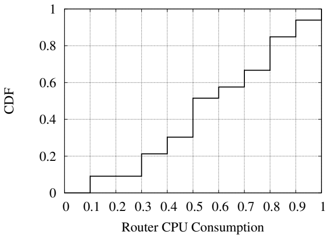


Fig. 2. Cumulative Distribution Function of CPU usage on physical routers hosting virtual routers in experiment 4F.

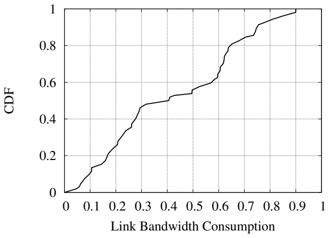


Fig. 3. Cumulative Distribution Function of bandwidth usage on physical links hosting virtual links in experiment 4F.

and memory resources were similar, and thus omitted due to space constraints. Within each experiment group, bandwidth usage grows proportionally to the number of virtual network requests, increasing from experiment A to F. Raising resource limits on each request also causes a growth in bandwidth consumption. However, the effect is notably less significant. This is verified by comparing, for example, experiments 2A, 2B, and 3A. Increasing the number of virtual network requests from 17 to 25 between 2A and 2B raises bandwidth consumption by approximately 212% (from 20.26 Gbps to 63.25 Gbps). By comparison, increasing bandwidth limits from 3 Gbps to 5 Gbps between 2A and 3A causes a smaller growth of 45% (from 20.26 Gbps to 29.30 Gbps).

Bandwidth consumption is also indirectly affected by varying CPU and memory requirements in virtual network requests. For example, experiments 1 and 2 have the same bandwidth limit in each virtual link (*i.e.*, 3 Gbps) and the same number of virtual network requests, but the total amount of bandwidth consumption increases (*e.g.*, in 1F it is 171.80 Gbps whereas in 2F it is 194.42 Gbps). This can be explained by the fact that raising resource usage in virtual routers causes our algorithm to select more physical routers to allocate them. Thus, the number of selected physical links also increases in order to create valid end-to-end paths. It is also worth noting that there is a slight decrease in bandwidth consumption in experiments D of each group. This can be explained by the fact that the physical network increases from 50 to 100 routers, thus resulting in less substrate saturation. In general, reducing substrate saturation tends to increase the amount of possible solutions, which may lead to better results.

Figures 2 and 3 depict the Cumulative Distribution Functions (CDFs) for resource consumption on physical routers and links that are used to embed requested virtual networks. Figure 2 shows that 57% of physical routers use, at most, 60% of their resources. Also, only about 15% of physical routers have resource consumption higher than 80%. This shows that the proposed method tends to not overload physical routers, as it avoids wasting resources unnecessarily. Similar results are obtained when analyzing bandwidth. Figure 3 shows that approximately 60% of physical links have less than 60% of bandwidth consumed. Additionally, no more than 6% of physical links have over 80% of bandwidth usage. Avoiding the overload of physical resources is desired in virtual network environments since it may increase the acceptance of future requests [5]. Furthermore, overloading physical devices may decrease performance and increase the occurrence of failures.

In order to measure the impact of considering security requirements during the allocation process, all security related constraints were disabled, and all security requirements in virtual network requests were removed. This resulted in a second algorithm that only considers CPU, memory, and bandwidth requirements when trying to allocate virtual networks. Figure 4 shows the difference in bandwidth consumption between the results obtained from both algorithms (with and without secu-

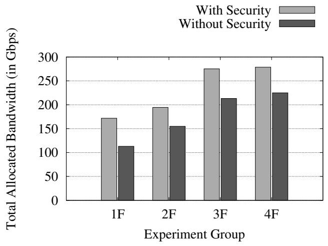


Fig. 4. Comparison between the bandwidth consumed by the optimal allocation in scenarios 1–4F with and without security requirements.

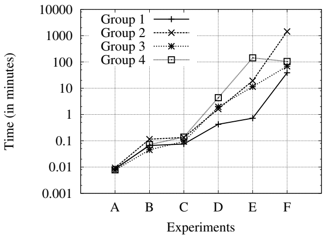


Fig. 5. Time needed to find the optimal solution in each scenario.

rity requirements, respectively). As can be observed, providing security in this environment causes a significant overhead. Disabling security requirements in experiment 1F reduced bandwidth consumption by approximately 34% (from 171.80 Gbps to 113.08 Gbps), representing the greatest reduction among these four comparisons. The less significant reduction, of approximately 19%, was observed in experiment 4F, which still represents a high overhead.

The main reasons for this overhead are: *i*) the set of routers that support encryption/decryption protocols is a subset of all possible routers, thus resulting in a more constrained solution space; and *ii*) the non-overlapping requirement forces the allocation algorithm to select detour paths in the substrate network, which results in higher resource consumption. These reasons also indicate that, in the best case scenario, the bandwidth consumption considering security-related constraints will be only as good as without considering them. Thus, results obtained in this analysis also evidence that minimizing bandwidth consumption is a desirable optimization objective when allocating virtual network requests with security-related constraints.

Last, we analyze the time needed to solve the virtual network embedding problem. Figure 5 presents the total duration of each experiment. The time axis is represented in logarithmic scale, as running times differ significantly among results. We consider that the time needed to execute most experiments would be acceptable in real environments. Experiments in groups A to C finished in less than a minute, while all but one experiment in groups D and E finished in less than 20 minutes. As results are optimal, infrastructure providers may find these times acceptable since the benefit of decreased cost may outweigh waiting times.

As for the remaining experiments, it becomes clear that there is a trade-off between running time and optimality. With the exception of experiment 2F, all other experiments finished in less than 3 hours and presented optimal solutions. Experiment 2F was aborted after 24 hours, but despite achieving a sub-optimal solution, the gap to optimality was less than 1%. By exploiting this gap, it is possible to obtain better performance at the cost of obtaining a sub-optimal solution. In this experiment, the gap to optimality after 20 minutes was of 10.72%. Further, after 3 hours of execution, the gap was reduced to 5.81%. Therefore, an infrastructure provider could find acceptable to stop the execution after a given time or gap threshold (possibly a combination of both).

As previously stated, we considered all obtained results to be optimal. This is true for all experiments except 2F, since it was aborted after running for 24 hours. Nevertheless, as the gap to optimality was significantly small, the authors of this paper considered this experiment to be fit for analysis.

V. Conclusions

During our research, we observed a number of existing approaches on the problem of virtual network embedding. We have also observed the existence of proposals that aim at providing security to virtual network environments. However, to the best of our knowledge, there have been no previous attempts to combine these two areas, providing security-aware optimal virtual network embedding.

Considering both optimal mapping and security to be equally important, we devised a model that combines CPU, memory, bandwidth, and location constraints with security requirements. Virtual networks may require end-to-end or point-to-point cryptography between their routers, or may demand that their virtual routers and links do not share physical devices and paths with other specific virtual networks.

In most of our experiments, our solution was able to find the optimal mapping in a reasonable time frame. However, some of our tests indicate that it may be necessary to use alternative methods (possibly suboptimal) in order to find a solution for more complex scenarios in a shorter time frame. We intend to enhance our model by using metaheuristics, which would deliver an approximate solution in a shorter amount of time.

Other perspectives for future work include the *online* handling of virtual network requests, as well as allowing the migration of previously embedded virtual networks. Despite the increased complexity, such features would render our solution more appropriate for real life scenarios, in which requests are typically not known in advance, and embedding virtual networks as they arrive may lead to resource fragmentation.

Acknowledgments

This work has been supported by the European Commission's Seventh Framework Programme and CNPq Research Agency (Project FP7-ICT-2011-EU-Brazil – SecFuNet), as well as RNP Academic Network's CTIC program (Project ReVir).

References

[1] N. Fernandes, M. Moreira, I. Moraes, L. Ferraz, R. Couto, H. Carvalho, M. Campista, L. Costa, and O. Duarte, "Virtual networks: Isolation, performance, and trends," in *Annals of Telecommunications*, 2010.

[2] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, Apr. 2005. [Online]. Available: http://dx.doi.org/10.1109/MC.2005.136

[3] D. Andersen, "Theoretical approaches to node assignment," 2002, unpublished manuscript. [Online]. Available: http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps

[4] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355737

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

[6] G. P. Alkmim, D. M. Batista, and N. L. S. Fonseca, "Optimal mapping of virtual networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, December 2011, pp. 1–6.

[7] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," in *SIGCOMM Computer Communication Review*, vol. 41. New York, NY, USA: ACM, April 2011, pp. 38–47.

[8] S. Davy, J. Serrat, A. Astorga, B. Jennings, and J. Rubio-Loyola, "Policy-assisted planning and deployment of virtual networks," in *Network and Service Management (CNSM), 2011 7th International Conference on*, oct. 2011, pp. 1 –8.

[9] S. Cabuk, C. I. Dalton, H. Ramasamy, and M. Schunter, "Towards automated provisioning of secure virtualized networks," in *Proceedings of the 14th ACM conference on Computer and communications security*, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 235–245.

[10] D. Huang, S. Ata, and D. Medhi, "Establishing secure virtual trust routing and provisioning domains for future internet," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, dec. 2010, pp. 1–6.

[11] S. Kent and K. Seo. (2005, December) Rfc 4301: Security architecture for the internet protocol. [Online]. Available: http://tools.ietf.org/rfc/rfc4301.txt

[12] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: Boston university representative internet topology generator. [Online]. Available: http://www.cs.bu.edu/brite

[13] R. Albert and A.-L. Barabási, "Topology of evolving networks: Local events and universality," *Phys. Rev. Lett.*, vol. 85, pp. 5234–5237, Dec 2000. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.85.5234

APPENDIX B – PUBLISHED PAPER – SBSEG 2012

- Title: Um Modelo para Mapeamento Ótimo de Redes Virtuais com Requisitos de Segurança
- Conference: 12th Brazilian Symposium on Information and Computer System Security (SBSeg 2012)
- URL: http://sbseg2012.ppgia.pucpr.br/
- Date: 19–22 November 2012
- Venue: Curitiba, Paraná, Brazil

Um Modelo para Mapeamento Ótimo de Redes Virtuais com Requisitos de Segurança

Leonardo Richter Bays[1], Rodrigo Ruas Oliveira[1], Luciana Salete Buriol[1], Marinho Pilla Barcellos[1], Luciano Paschoal Gaspary[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{lrbays,ruas.oliveira,buriol,marinho,paschoal}@inf.ufrgs.br

Abstract. *Network virtualization enables the creation of multiple instances of virtual networks on top of a single physical infrastructure. Given its wide applicability, this technique has attracted a lot of interest both from academic researchers and major companies within the segment of computer networks. Although recent efforts (motivated mainly by the search for mechanisms to enable the evaluation of Future Internet proposals) have contributed substantially to materialize this concept, none of them has attempted to combine efficient resource allocation with fulfillment of security requirements (e.g., confidentiality). It is important to note that, in the context of virtual networks, the protection of shared network infrastructures constitutes a fundamental condition to enable its use in large scale. To address this problem, in this paper we propose a virtual network embedding model that aims to provide the desired level of security while optimizing physical resource usage. The results obtained demonstrate that the model is able to correctly and optimally map virtual networks to a physical substrate, minimizing bandwidth costs for infrastructure providers.*

Resumo. *A virtualização de redes permite a criação de múltiplas instâncias de redes virtuais sobre uma única infraestrutura física. Devido à sua ampla aplicabilidade, tal técnica tem atraído grande interesse tanto de pesquisadores quanto de empresas importantes do segmento de redes de computadores. Apesar de esforços recentes (motivados principalmente pela busca de mecanismos para viabilizar a avaliação de propostas na temática Internet do Futuro) terem contribuído substancialmente para a materialização do conceito, nenhum preocupou-se em conciliar alocação eficiente de recursos e satisfação de requisitos de segurança (ex: confidencialidade). Ressalta-se que, no contexto de redes virtuais, a proteção de infraestruturas de rede compartilhadas constitui condição fundamental para seu uso em larga escala. Para abordar o referido problema, neste artigo propõe-se um modelo de alocação de redes virtuais que busca satisfazer o nível especificado de segurança e, ao mesmo tempo, otimizar a utilização dos recursos físicos. Os resultados obtidos demonstram que o modelo é capaz de alocar redes virtuais a um substrato físico de forma correta e ótima, minimizando custos de largura de banda para provedores de infraestrutura.*

1. Introdução

Nos últimos anos, têm surgido demandas cada vez maiores por serviços de rede específicos, com requisitos peculiares e distintos. Motivados por tais demandas, e estimulados pelo sucesso no emprego de virtualização para hospedagem de servidores personalizados, pesquisadores passaram a explorar o uso dessa técnica em infraestruturas de rede. A virtualização de redes permite a criação de múltiplas topologias virtuais sobre um mesmo substrato físico. Isso é possível por meio da instanciação de um ou mais roteadores virtuais em dispositivos físicos e do estabelecimento de enlaces virtuais entre esses roteadores, formando topologias arbitrárias.

Entre outras vantagens, o uso de virtualização de redes permite a um provedor de infraestrutura acomodar simultaneamente múltiplas pilhas de protocolo no mesmo substrato. Isso possibilita a criação de infraestruturas de rede adaptadas às necessidades de aplicações de rede específicas [Fernandes et al. 2010]. Ademais, essa técnica pode ser usada para a execução de experimentos sem interferir com tráfego de produção, em escala e com um alto grau de similaridade com infraestruturas reais. Dessa forma, é possível criar ambientes favoráveis ao desenvolvimento e avaliação de novas arquiteturas e protocolos, o que pode contribuir para o avanço de pesquisas relacionadas à Internet do Futuro [Anderson et al. 2005].

A virtualização de redes também tem recebido grande apoio no mercado. Empresas importantes passaram a oferecer dispositivos com suporte nativo à virtualização. Essa nova funcionalidade permite que provedores de infraestrutura passem também a oferecer novos serviços. O suporte de grandes nomes da indústria a esse tipo de iniciativa pode ser observado, por exemplo, na lista de membros da *Open Networking Foundation*[1], que promove o desenvolvimento e o uso de redes virtualizadas definidas por software.

Apesar de sua ampla aplicabilidade, manter um ambiente de virtualização de redes requer uma distribuição adequada dos recursos. Por um lado, há provedores de infraestrutura, que desejam obter o máximo de lucro hospedando a maior quantidade possível de redes virtuais, minimizando seus custos. Por outro, há uma série de clientes que solicitam redes virtuais com demandas de recursos específicas. O método de alocação deve garantir que os recursos requisitados estarão disponíveis para esses clientes e, ao mesmo tempo, minimizar os custos do provedor de infraestrutura. Além disso, o resultado do processo de mapeamento deve ser entregue em um tempo aceitável.

Outra grande preocupação que surge com o uso compartilhado de dispositivos de roteamento e canais de comunicação é a segurança. Sem a proteção adequada, é possível que usuários de uma rede virtual capturem ou até mesmo adulterem dados de outras redes virtuais no mesmo substrato. Tais ações violariam propriedades de segurança tais como confidencialidade e integridade. Portanto, é de grande importância que arquiteturas de virtualização ofereçam proteção contra essas e outras ameaças que possam comprometer sua segurança.

Para viabilizar o uso de virtualização em ambientes reais, tanto alocação eficiente de recursos quanto segurança devem ser levados em consideração. O problema da alocação de recursos é considerado *NP-hard* devido a sua similaridade com o *multi-way separator problem* [Andersen 2002], e tem sido geralmente abordado na literatura com algoritmos de alocação modelados por meio de programação linear. Há uma série de trabalhos focando no problema da alocação de recursos de redes virtuais [Yu et al. 2008, Chowdhury et al. 2009, Alkmim et al. 2011, Cheng et al. 2011, Davy et al. 2011]. No entanto, os autores deste artigo desconhecem propostas que levam em consideração requisitos de segurança. Para preencher essa lacuna, no presente artigo é proposto um modelo de alocação de redes virtuais que otimiza a utilização de recursos físicos ao mesmo tempo em que atende requisitos de segurança. Além de requisitos de capacidade e

[1] http://www.opennetworking.org/membership

localização, solicitantes de redes virtuais podem especificar requisitos de segurança para suas redes, que devem ser atendidos pelo provedor de infraestrutura. O modelo proposto recebe requisições de forma *on-line* e determina a melhor alocação possível em termos de utilização de recursos, considerando todos os requisitos de segurança dos solicitantes.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados às áreas de mapeamento de recursos e segurança em redes virtuais. Na seção 3 é descrito o modelo proposto e sua formulação. A Seção 4 relata a avaliação realizada e apresenta os resultados obtidos. Por fim, na Seção 5 são apresentadas as considerações finais e perspectivas para trabalhos futuros.

2. Trabalhos Relacionados

Nessa seção, serão apresentados os trabalhos relacionados focando no mapeamento de redes virtuais, bem como algumas das principais propostas visando prover segurança a redes virtuais. Cada trabalho será descrito brevemente, ressaltando suas principais características.

Yu et al. [Yu et al. 2008] propõem um modelo de alocação de redes virtuais com suporte à separação de caminhos (fragmentação de enlaces virtuais por múltiplos caminhos do substrato físico) e migração. O modelo proposto aproveita-se do ganho em flexibilitade obtido pela separação de enlaces virtuais em múltiplos caminhos (caso tal separação seja permitida pelo solicitante), reduzindo o tempo necessário para completar o processo de mapeamento. Além disso, o substrato é capaz de reotimizar sua utilização de recursos periodicamente por meio da migração de roteadores e enlaces virtuais previamente alocados. O modelo considera que requisições de redes virtuais não são conhecidas *a priori*, e leva em consideração requisitos de CPU e largura de banda, bem como o tempo máximo que uma requisição pode aguardar antes de ser atendida.

Outro modelo, formulado por Chowdhury et al. [Chowdhury et al. 2009], visa aprimorar a coordenação entre a alocação de roteadores e enlaces, que é realizada em duas fases separadas. Isso é feito por meio da pré-seleção de alocações de roteadores de forma a auxiliar o estágio de mapeamento de enlaces. Assim como o modelo previamente mencionado, esse também permite a separação de caminhos, e considera requisitos de CPU e largura de banda. Requisições de redes virtuais são alocadas de forma *on-line*, e podem especificar as localizações físicas em que certos roteadores virtuais devem ser mapeados.

O modelo desenvolvido por Alkmim et al. [Alkmim et al. 2011] estende os trabalhos anteriores combinando requisitos de capacidade com restrições relacionadas à transferência de imagens utilizadas nos roteadores virtuais. O modelo visa minimizar o tempo necessário para transferir tais imagens, ao mesmo tempo que considera requisitos de CPU, memória, banda e localização. Como nos trabalhos anteriores, requisições são recebidas e alocadas de forma *on-line*.

Cheng et al. [Cheng et al. 2011] apresentam uma abordagem baseada na classificação de nós (*node ranking*), que considera tanto a capacidade de roteadores e enlaces quanto as capacidades de seus vizinhos. Por exemplo, a classificação de roteadores físicos é afetada não só por sua própria capacidade, mas também pela capacidade disponível em outros roteadores conectados ao mesmo. De forma análoga, a classificação de roteadores e enlaces virtuais também leva em consideração as características de seus vizinhos. O processo de mapeamento aloca roteadores e enlaces virtuais a elementos da rede física com classificações similares. Tal estratégia, segundo os autores, tende a reduzir possíveis gargalos criados pela alocação de redes virtuais.

Ao contrário das propostas apresentadas anteriormente, o modelo proposto por Davy et al. [Davy et al. 2011] não recebe como entrada uma topologia de rede completa. Em vez disso, requisições contêm apenas os *end points* que devem ser interconectados (uma origem e um ou mais destinos). A solução constrói redes virtuais em forma de árvore, partindo da localização de origem até os destinos requisitados. Além de restrições de localização, o modelo também considera a preferência dos solicitantes por redes de baixo custo, ou redes com menor atraso e custo maior. As redes virtuais são então instanciadas, obedecendo os requisitos previamente mencionados e visando minimizar os custos do provedor de infraestrutura.

Apresentados e discutidos os principais trabalhos relacionados à alocação de recursos, passa-se, agora, a uma síntese das investigações que buscam oferecer segurança em redes virtualizadas. Cabuk et al. [Cabuk et al. 2007] apresentam um arcabouço para criação de redes virtuais seguras. Os autores utilizam "Domínios Virtuais Confiáveis" (*Trusted Virtual Domains* – TVDs) para prover controle de acesso, confidencialidade e integridade a comunicações de rede. Cada TVD representa um domínio isolado, composto de entidades virtuais e de enlaces entre as mesmas. Certificados digitais são usados para assegurar que somente entidades que satisfaçam um determinado conjunto de condições sejam capazes de participar de um TVD. Os autores usam VLANs para isolar o tráfego dentro de cada rede confiável, e VPNs para interconectar tais redes.

Huang et al. [Huang et al. 2010] propõem um método que lança mão de criptografia para proteger informações de roteamento, bem como caminhos variáveis para mitigar ataques de análise de tráfego. Tal método classifica roteadores em diferentes grupos, e distribui chaves de grupo para cada roteador. Dessa forma, somente roteadores pertencentes a um determinado grupo são capazes de acessar as informações protegidas de tal grupo. Além disso, cada enlace virtual é mapeado a múltiplos caminhos físicos. Fluxos transmitidos por tais enlaces são divididos aleatoriamente entre os caminhos físicos disponíveis, visando evitar análise de tráfego.

Ao mesmo tempo em que há trabalhos focando no problema de alocação de redes virtuais, há outros visando oferecer serviços de segurança a ambientes de redes virtuais. No entanto, os autores deste artigo desconhecem trabalhos que abordem ambas as áreas simultaneamente. Ao não considerarem segurança, um aspecto essencial em ambientes de virtualização de redes devido ao compartilhamento de recursos físicos, os trabalhos anteriores na área de alocação de redes virtuais acabam por subestimar a quantidade de recursos necessária para acomodar tais redes. Nesse contexto, busca-se, neste artigo, suprir tal lacuna, ao propor uma solução que concilia alocação eficiente de recursos com satisfação de requisitos de segurança, fatores fundamentais para ampla adoção de redes virtuais em ambientes de produção.

3. Modelo Proposto

Para abordar o problema da otimização do uso de recursos considerando requisitos de segurança, foi desenvolvido um modelo por meio de Programação Linear Inteira. Para criar um modelo que represente o cenário de alocação de redes virtuais com um nível desejável de fidelidade, diversos detalhes foram levados em consideração. Vislumbra-se um cenário em que um provedor de infraestrutura fornece redes virtuais a diversos clientes. Para solicitar a criação de uma rede virtual, clientes devem firmar um Acordo de

Nível de Serviço (*Service Level Agreement* – SLA) com o provedor de infraestrutura. Tal SLA descreve as características da rede virtual solicitada e seus requisitos de segurança, que devem ser atendidos pelo provedor.

Assume-se que um provedor de infraestrutura receberá uma série de requisições de redes virtuais ao longo do tempo. Portanto, tais requisições devem ser tratadas de forma *on-line*, isto é, uma a uma conforme são recebidas. Caso haja recursos suficientes no substrato para que a alocação seja possível, a saída do modelo deve indicar a melhor alocação em termos de utilização de recursos, maximizando os recursos disponíveis para futuras requisições. Caso não haja recursos suficientes para alocar uma rede virtual, a requisição é negada.

Antes de apresentar o modelo proposto, será descrita a sintaxe usada na sua formulação. Letras maiúsculas são usadas para representar conjuntos ou variáveis. Letras sobrescritas indicam se um conjunto refere-se a recursos virtuais (V) ou físicos (P), ou se o mesmo se refere a roteadores (R) ou enlaces (L). Ademais, letras subscritas representam índices associados a variáveis.

Topologias. Requisições de redes virtuais devem especificar a topologia desejada, isto é, o número de roteadores virtuais na rede e as interconexões entre os mesmos. Cada topologia de rede virtual, bem como a topologia da rede física, é representada como um grafo direcionado $N = (R, L)$, no qual os vértices R e as arestas L representam, respectivamente, roteadores e enlaces. Além disso, um enlace entre dois roteadores a e b é representado por um par de arestas simétricas com direções opostas, (a, b) e (b, a). Roteadores virtuais são mapeados a exatamente um roteador físico, enquanto que enlaces virtuais podem ser mapeados a um único enlace físico ou a um caminho composto por dois ou mais enlaces físicos.

Capacidades físicas e virtuais. Roteadores físicos possuem capacidades limitadas de CPU e memória, expressas por C_i^P e M_i^P, respectivamente (em que i é o índice do roteador). Por sua vez, enlaces possuem capacidade de banda limitada $B_{i,j}^P$, em que o par (i, j) representa um enlace físico entre i e j. De forma similar, $C_{r,i}^V$ e $M_{r,i}^V$ representam os requisitos de CPU e memória de um roteador virtual de uma rede r. Além disso, $B_{r,i,j}^V$ representa o requisito de largura de banda de um enlace virtual entre os roteadores virtuais i e j de uma rede r. Os requisitos desses elementos virtuais definem a parcela dos recursos físicos que deve ser alocada para seu consumo. Presume-se que a arquitetura de virtualização é capaz de isolar adequadamente os recursos físicos, garantindo o cumprimento desses limites.

Localidades. Assume-se que a maioria dos clientes requisitará redes virtuais fixando um ou mais pontos onde roteadores virtuais deverão ser hospedados. Portanto, cada roteador físico está associado a um identificador de localização S^P, e requisições de redes virtuais podem ou não requerer que alguns de seus roteadores sejam mapeados a roteadores físicos em localidades específicas. Roteadores virtuais com requisitos de localidade são armazenados no conjunto S^V.

Segurança. O modelo também permite que cada requisição de rede virtual possua requisitos de segurança associados. O oferecimento de serviços de confidencialidade visa tratar preocupações relacionadas ao uso compartilhado de roteadores físicos e canais de comunicação, o que pode fazer com que dados sensíveis sejam expostos a terceiros. Tais requisitos, se presentes, indicam um de três níveis distintos de confidencialidade que devem ser fornecidos às comunicações dessas redes:

- Criptografia fim-a-fim: caso esse nível de confidencialidade seja solicitado, os roteadores de borda da rede virtual devem ser hospedados em roteadores físicos que suportam tal característica. Na prática isso significa que esses roteadores físicos devem dar suporte a suítes de protocolos tais como IPSec [Kent and Seo 2005], que fornece criptografia fim-a-fim quando usado em *modo de transporte*.
- Criptografia ponto-a-ponto: nesse nível de confidencialidade, pacotes inteiros são criptografados, protegendo não só os dados contidos nos mesmos mas também seu cabeçalho. Isso significa que pacotes precisam ser decriptografados e recriptografados em cada salto para serem roteados adequadamente. Portanto, cada roteador em uma rede virtual que requer esse nível de confidencialidade deve ser mapeado a um roteador físico capaz de dar suporte a tais operações. Esse nível corresponde ao *modo de túnel* do IPSec, o que significa que roteadores físicos com suporte a esse protocolo são capazes de prover tal característica.
- Não-sobreposição de redes: uma requisição de rede virtual pode também exigir que seus roteadores e enlaces virtuais não compartilhem roteadores nem caminhos físicos com uma ou mais redes virtuais. Tal caso extremo pode ser usado, por exemplo, para proteger informações altamente sigilosas de empresas concorrentes.

Para prover os dois primeiros níveis de segurança, requisições de redes virtuais devem ser capazes de indicar quais de seus roteadores devem ser capazes de criptografar e decriptografar pacotes de rede, caso desejado. Portanto, o modelo também incorpora os conjuntos K^P_i e $K^V_{r,i}$, que indicam se um roteador físico é capaz de oferecer tal característica, e se um roteador virtual a requer.

Já quanto ao terceiro nível, requisições devem ser capazes de especificar o conjunto de redes virtuais com o qual roteadores e enlaces físicos não serão compartilhados. Para oferecer tal nível, é usado o conjunto X. Esse conjunto é composto por pares de redes virtuais que não devem compartilhar recursos do substrato (por exemplo, se $(i, j) \in X$, não é permitido que as redes virtuais i e j compartilhem recursos).

Alocação prévia. Por fim, os conjuntos $E^R_{i,r,j}$ e $E^L_{i,j,r,k,l}$ indicam onde encontram-se alocados, respectivamente, os roteadores e enlaces das redes virtuais já alocadas no substrato. Caso não haja nenhuma rede virtual alocada no momento em que uma requisição é recebida, tais conjuntos estarão vazios. A seguir, para maior clareza, apresenta-se um sumário das entradas do modelo proposto.

- $N^P = \{R^P, L^P\}$ – Representa a rede física, composta por um conjunto de roteadores físicos R^P e um conjunto de enlaces físicos L^P.
- $N^V = \{R^V, L^V\}$ – Representa uma requisição de rede virtual, composta por um conjunto de roteadores virtuais R^V e um conjunto de enlaces virtuais L^V.
- $X \in N^V \times N^V$ – Conjunto de redes virtuais conflitantes. Representa redes virtuais que não devem ser mapeadas aos mesmos elementos do substrato físico.
- $S \in \mathbb{N}$ – Conjunto de todas as possíveis localidades físicas onde roteadores físicos podem residir, representadas por números naturais.
- $S^P \in R^P \times S$ – Indica a localização de roteadores da rede física.
- $S^V \in R^V \times S$ – Indica requisitos de localização de roteadores em requisições de redes virtuais.
- $C^P_i \in \mathbb{N}$ – Indica a capacidade total de CPU de um roteador físico i.
- $M^P_i \in \mathbb{N}$ – Indica a capacidade total de memória de um roteador físico i.
- $B^P_{i,j} \in \mathbb{N}$ – Indica a largura de banda de um enlace físico (i, j).

- $K^P_i \in \{0, 1\}$ – Indica se um roteador físico i suporta protocolos que o permitam criptografar e decriptografar pacotes de rede. Se um roteador físico é capaz de dar suporte a tais protocolos, o valor é definido como 1; caso contrário, é definido como 0.
- $C^V_{r,i} \in \mathbb{N}$ – Indica a capacidade de CPU exigida por um roteador virtual i de uma rede virtual r.
- $M^V_{r,i} \in \mathbb{N}$ – Indica a capacidade de memória necessária a um roteador virtual i de uma rede virtual r.
- $B^V_{r,i,j} \in \mathbb{N}$ – Indica a largura de banda exigida por um enlace virtual (i, j) de uma rede virtual r.
- $K^V_{r,i} \in \{0, 1\}$ – Indica se um roteador virtual i de uma rede virtual r deve ser capaz de criptografar e decriptografar pacotes de rede. Se um roteador virtual requer tal característica, o valor é definido como 1; caso contrário, é definido como 0.
- $E^R_{i,r,j} \in \{0, 1\}$ – Indica se um roteador virtual j de uma rede virtual r previamente recebida encontra-se alocado no roteador físico i. Em caso positivo, assume o valor 1; caso contrário, assume o valor 0.
- $E^L_{i,j,r,k,l} \in \{0, 1\}$ – Indica se um enlace virtual (k, l) de uma rede virtual r previamente recebida encontra-se alocado no enlace físico (i, j). Em caso positivo, assume o valor 1; caso contrário, assume o valor 0.

De forma similar, as variáveis de saída do modelo proposto são apresentadas a seguir. Os valores retornados por tais variáveis indicam a alocação de elementos virtuais no substrato físico, representando a solução do problema. Uma vez que o problema é solucionado, cada roteador virtual estará mapeado a um único roteador físico, e cada enlace virtual estará mapeado a um caminho no substrato físico. Tal caminho pode ser um único enlace físico, ou uma série de enlaces físicos consecutivos.

- $A^R_{i,r,j} \in \{0, 1\}$ – Alocação de roteadores, indica se o roteador físico i está hospedando o roteador virtual j da rede virtual r.
- $A^L_{i,j,r,k,l} \in \{0, 1\}$ – Alocação de enlaces, indica se o enlace físico (i, j) está hospedando o enlace virtual (k, l) da rede virtual r.

Por fim, é apresentada a função objetivo do modelo e suas restrições. A função objetivo visa minimizar a largura de banda física consumida pelos enlaces virtuais nas redes solicitadas, dessa forma minimizando custos e preservando largura de banda para alocações futuras. Por sua vez, as restrições garantem que os requisitos serão atendidos, e que as capacidades físicas não serão excedidas.

Objetivo:

$$min \sum_{(i,j)\in L^P} \sum_{r\in N^V,(k,l)\in L^V} A^L_{i,j,r,k,l} B^V_{r,k,l}$$

Sujeito a:

$$\sum_{r\in N^V, j\in R^V} C^V_{r,j} A^R_{i,r,j} \leq C^P_i \qquad \forall i \in R^P \quad \text{(R1)}$$

$$\sum_{r\in N^V, j\in R^V} M^V_{r,j} A^R_{i,r,j} \leq M^P_i \qquad \forall i \in R^P \quad \text{(R2)}$$

$$\sum_{r\in N^V,(k,l)\in L^V} B^V_{r,k,l}A^L_{i,j,r,k,l} \leq B^P_{i,j} \qquad \forall (i,j) \in L^P \quad \text{(R3)}$$

$$K^V_{r,j}A^R_{i,r,j} \leq K^P_i \qquad \forall i \in R^P, r \in N^V, j \in R^V \quad \text{(R4)}$$

$$\sum_{i\in R^P} A^R_{i,r,j} = 1 \qquad \forall r \in N^V, j \in R^V \quad \text{(R5)}$$

$$\sum_{j\in R^P} A^L_{i,j,r,k,l} - \sum_{j\in R^P} A^L_{j,i,r,k,l} = A^R_{i,r,k} - A^R_{i,r,l} \qquad \forall r \in N^V, (k,l) \in L^V, i \in R^P \quad \text{(R6)}$$

$$\sum_{q\in N^V,k\in R^V} A^R_{i,q,k} + \sum_{r\in N^V,l\in R^V} A^R_{i,r,l} \leq 1 \qquad \forall q, r \in X, i \in R^P \quad \text{(R7)}$$

$$\sum_{q\in N^V,(k,l)\in L^V} A^L_{i,j,q,k,l} + \sum_{r\in N^V,(o,p)\in L^V} A^L_{i,j,r,o,p} \leq 1 \qquad \forall q, r \in X, (i,j) \in L^P \quad \text{(R8)}$$

$$jA^R_{i,r,k} = lA^R_{i,r,k} \qquad \forall (i,j) \in S^P, r \in N^V, (k,l) \in S^V \quad \text{(R9)}$$

$$A^R_{i,r,j} = E^R_{i,r,j} \qquad \forall (i,r,j) \in E^R \quad \text{(R10)}$$

$$A^L_{i,j,r,k,l} = E^L_{i,j,r,k,l} \qquad \forall (i,j,r,k,l) \in E^L \quad \text{(R11)}$$

$$\sum_{j\in R^V} A^R_{i,r,j} \leq 1 \qquad \forall i \in R^P, r \in N^V \quad \text{(R12)}$$

As primeiras três restrições garantem que os requisitos de capacidade dos roteadores e enlaces virtuais serão atendidos. A restrição R1 garante que a quantidade de CPU requisitada por roteadores virtuais mapeados a um roteador físico não excederá sua capacidade máxima. A restrição R2 aplica o mesmo controle à capacidade de memória dos roteadores físicos, e a restrição R3, à largura de banda dos enlaces físicos.

A restrição R4 garante que todos os roteadores virtuais que devem realizar criptografia e decriptografia de pacotes serão mapeados a roteadores físicos que suportam tais operações. Tais roteadores virtuais são os roteadores de borda no caso de redes virtuais que solicitam criptografia fim-a-fim, e todos os roteadores no caso de redes virtuais que requerem criptografia ponto-a-ponto.

A restrição R5 garante que cada roteador virtual será mapeado a um roteador físico. De forma complementar, a restrição R6 garante que o caminho formado por um conjunto de enlaces físicos hospedando um enlace virtual será válido. Em outras palavras, o caminho físico hospedando um enlace virtual (a, b) deve ser um caminho válido entre o roteador físico hospedando o roteador virtual *a* e o roteador físico hospedando o roteador virtual *b*.

As restrições R7 e R8 referem-se a pares de redes virtuais conflitantes – isto é, redes virtuais que não podem compartilhar recursos físicos. A restrição R7 não permite que roteadores virtuais que pertencem a redes virtuais conflitantes sejam mapeados aos mesmos roteadores físicos. De forma análoga, a restrição R8 garante que enlaces virtuais dessas redes conflitantes não compartilharão quaisquer caminhos físicos.

A restrição R9 garante que todo roteador virtual que possua um requisito de localidade será mapeado a um roteador físico na localidade solicitada. As restrições R10 e R11 garantem que os elementos das redes virtuais previamente alocadas continuarão alocados aos mesmos elementos físicos. A alocação dos roteadores será mantida pela restrição R10, enquanto que a alocação dos enlaces, pela restrição R11. Por fim, a restrição R12 impede que múltiplos roteadores virtuais de uma mesma rede virtual sejam hospedados no mesmo roteador físico.

4. Avaliação

Para avaliar o modelo em Programação Linear Inteira apresentado na seção anterior, o mesmo foi implementado e executado no *CPLEX Optimization Studio*[2] versão 12.3. Os experimentos foram realizados em uma máquina com quatro processadores AMD Opteron 6276, usando no máximo quatro *threads* simultâneas. A máquina possui 64 GB de RAM e usa o sistema operacional Ubuntu GNU/Linux Server 11.10 x86_64.

4.1. Cenários

Para realizar os experimentos, foi desenvolvido um simulador capaz de gerar requisições de redes virtuais de forma aleatória. O simulador é executado por 500 janelas de tempo, e são geradas em média cinco requisições em cada uma, seguindo uma distribuição de Poisson. Cada requisição permanece alocada por, em média, cinco janelas de tempo, seguindo uma distribuição exponencial. Ressalta-se que essa forma de instanciação, isto é, o emprego de janelas de tempo e os modelos de chegada de requisições e de duração de redes virtuais na infraestrutura, é empregada em trabalhos importantes da área, com destaque para o realizado por Yu et al. [Yu et al. 2008].

A topologia da rede física e de cada rede virtual é gerada por meio da ferramenta BRITE[3], usando o modelo Barabási-Albert (BA-2) [Albert and Barabási 2000]. A rede física possui 100 roteadores, cada um com capacidade total de CPU definida como 100%, e 256 MB de memória. Além disso, os roteadores são distribuídos uniformemente entre 16 localidades, e 95% suportam protocolos que os permitem oferecer serviços de criptografia. A largura de banda dos enlaces físicos é distribuída uniformemente entre 1 e 10 Gbps.

As requisições de redes virtuais possuem entre 2 e 5 roteadores cada. Em cada rede virtual, dois roteadores (os *end points* dessa rede) possuem requisitos de localidade, gerados aleatoriamente entre as 16 localidades existentes. 35% das requisições geradas não possuem requisitos de criptografia, enquanto que 35% requerem criptografia fim-a-fim, e as demais 30%, criptografia ponto-a-ponto. De forma independente, 5% das requisições possuem conflito com uma rede já alocada no substrato.

[2] http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/

[3] http://www.cs.bu.edu/brite/

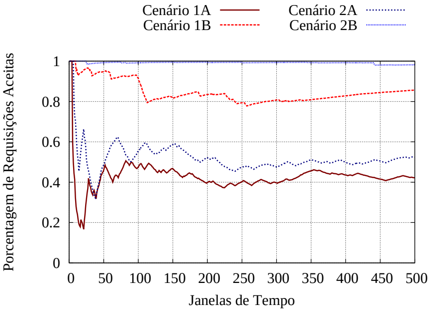


Figura 1. Porcentagem média de requisições aceitas nos experimentos realizados.

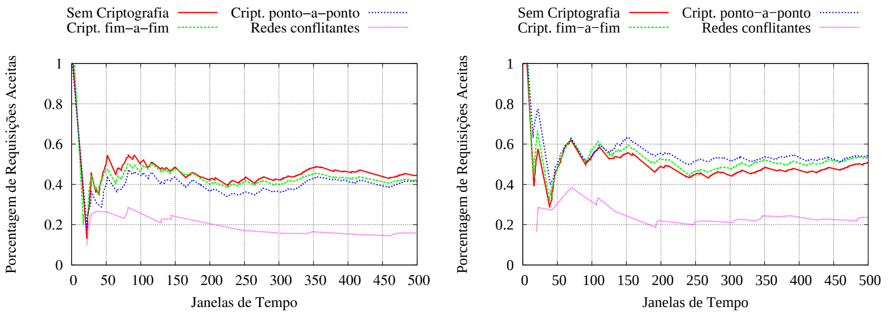
Foram criados dois cenários para avaliar o modelo, os quais se diferenciam pelos requisitos de capacidade dos elementos das redes virtuais. No primeiro, denominado *cenário 1*, cada roteador virtual requer entre 10 e 50% de CPU, e entre 32 e 128 MB de memória. Os enlaces das redes virtuais nesse cenário requerem entre 1 e 5 Gbps. Já no *cenário 2*, roteadores requerem entre 10 e 25% de CPU e entre 32 e 64 MB de memória, e enlaces requerem entre 1 e 2.5 Gbps. Os limites superiores dos requisitos do cenário 1 equivalem a 50% da capacidade disponível em roteadores e enlaces físicos, enquanto que no cenário 2, tais limites equivalem a 25% da capacidade dos elementos físicos. Todos os parâmetros previamente descritos seguem uma distribuição uniforme.

4.2. Resultados

Inicialmente, foram realizados experimentos seguindo os dois cenários descritos na subseção anterior. Em seguida, os experimentos foram repetidos, utilizando as mesmas redes físicas e as mesmas requisições geradas em cada janela de tempo, porém com uma versão modificada do modelo que desconsidera requisitos de segurança. As versões dos cenários 1 e 2 em que são considerados os requisitos de segurança são denominadas *1A* e *2A*. Já as versões modificadas para ignorar tais requisitos são denominadas *1B* e *2B*. Os resultados dos experimentos realizados com essas diferentes versões foram comparados para caracterizar o impacto causado pelo emprego de serviços relacionados à confidencialidade.

Analisou-se a taxa de aceitação de requisições de redes virtuais nos experimentos realizados. Ressalta-se que requisições somente são negadas caso não seja possível acomodar a rede virtual solicitada no substrato atendendo todos os seus requisitos. A Figura 1 ilustra a taxa média de aceitação obtida em cada cenário. Cada ponto no gráfico denota a taxa média de aceitação obtida desde o início do experimento até a janela de tempo em questão.

Analisando o gráfico, percebe-se, de forma clara, o impacto causado pelo fornecimento de serviços relacionados à confidencialidade. Nos cenários 1A e 2A, em que os requisitos de segurança são considerados, a taxa média ao fim dos experimentos é de, respectivamente, 42,2% e 52,5%. Já nos demais cenários, 1B e 2B, as taxas são de respectivamente 85,6% e 98,2%. Observa-se, ainda, uma taxa de aceitação maior nas variantes do cenário 2 em relação às do cenário 1, devido ao fato de que nos cenários 2A e 2B as requisições possuem requisitos de capacidade mais baixos. Além disso, é possível perceber que, em todos os casos, a taxa de aceitação inicial é de 100%, visto que o substrato



(a) Cenário 1A

(b) Cenário 2A

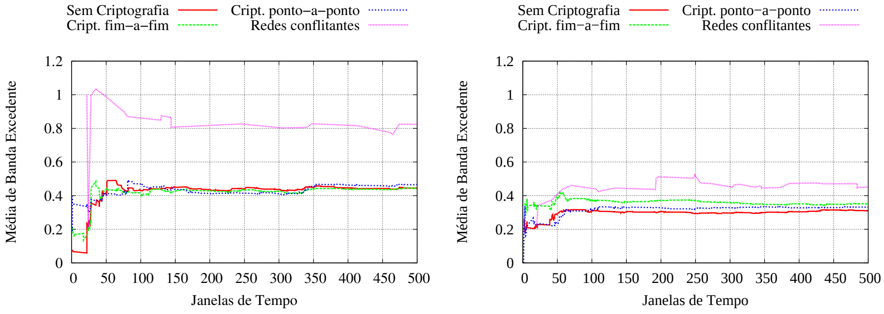
Figura 2. Porcentagem média de requisições aceitas nos cenários 1A e 2A, divididas por tipo de requisição.

encontra-se desocupado no início do experimento. Após algum tempo, devido à saturação dos elementos físicos, requisições passam a ser negadas, causando quedas na taxa média. A mesma, então, volta a subir conforme algumas das redes virtuais alocadas alcançam sua duração máxima e são removidas do substrato. Esse comportamento se repete ao longo dos experimentos, gradualmente convergindo para um valor médio.

A seguir, a Figura 2 apresenta, em maiores detalhes, a taxa de aceitação dos diferentes tipos de requisições presentes nos cenários 1A e 2A. Os gráficos exibem um comportamento em grande parte similar ao anterior, porém nota-se que em ambos a taxa de aceitação de requisições com conflitos é significativamente mais baixa do que a média geral. Isso se deve à dificuldade de alocar redes virtuais conflitantes sem que nenhum de seus roteadores e enlaces se sobreponham. A taxa média de aceitação de redes virtuais com conflitos é de 15,9% no cenário 1A, e de 23,6% no cenário 2A.

Ainda observando o gráfico ilustrado na Figura 2, percebe-se que não há uma grande diferença entre a porcentagem de redes aceitas sem criptografia, com criptografia fim-a-fim e com criptografia ponto-a-ponto. As taxas médias ao longo do cenário 1A são de, respectivamente, 44,4%, 42,1% e 41,6%. Já no cenário 2A, as médias são de, respectivamente, 50,8%, 53,3% e 54,2%. É importante salientar que, nos experimentos realizados, 95% dos roteadores da rede física oferecem suporte a protocolos que permitem a criptografia e decriptografia de pacotes, e que o modelo não considera custos adicionais de processamento e memória necessários para tais operações. Considerando-se uma taxa menor de equipamentos com suporte a tais operações, ou os custos adicionais associados às mesmas, julga-se que haveria uma diferença mais expressiva entre os diferentes tipos de requisições. Contudo, acredita-se conseguir, com o experimento realizado, oferecer uma boa visão global do custo associado para satisfazer requisitos de segurança no contexto investigado.

A próxima análise foca na largura de banda total necessária para alocar cada requisição em relação à largura de banda requisitada. A largura de banda ocupada por redes virtuais alocadas no substrato tende a ser mais alta do que a requisitada, visto que um único enlace virtual pode ser alocado em um caminho composto por uma série de enlaces físicos. A Figura 3 apresenta a média de largura de banda excedente das requisições aceitas nos cenários 1A e 2A, separadas pelo nível de confidencialidade solicitado. Ressalta-



(a) Cenário 1A

(b) Cenário 2A

Figura 3. Média de largura de banda excedente necessária para acomodar as requisições de diferentes tipos aceitas nos cenários 1A e 2A.

se que o modelo visa minimizar a largura de banda consumida pelas redes virtuais alocadas, e que todas as soluções resultantes dos experimentos realizados são ótimas. Nota-se que requisições com conflitos tendem a consumir uma banda significativamente maior em relação às demais. No cenário 1A, tais requisições consomem em média 82,5% de largura de banda além do solicitado, e no cenário 2A, 45,1%. Em outras palavras, o custo da alocação de redes conflitantes para provedores de infraestrutura torna-se muito maior. Isso ocorre devido à necessidade de se utilizar caminhos mais longos para evitar a sobreposição dos elementos dessas redes.

Quanto à largura de banda excedente dos demais tipos de requisições, no início do experimento 1A há uma diferença visível entre as mesmas. Até a vigésima janela de tempo, requisições sem criptografia possuem banda média excedente de 6,7%, enquanto que requisições com criptografia fim-a-fim, 13,4%, e as com criptografia ponto-a-ponto, 35,1%. Ou seja, até esse momento, requisições com níveis mais altos de criptografia exigem uma quantidade maior de recursos para serem alocadas. No entanto, de forma similar aos gráficos da Figura 2, ao longo da execução tais valores convergem para porcentagens muito próximas. Ao término da execução, as médias de largura de banda excedente situam-se entre 44,3% e 46,5%. No cenário 2A, há sobreposições entre tais médias desde o início do experimento, e ao término do mesmo, novamente as médias encontram-se muito próximas, entre 31,1% e 35,1%.

Por fim, é apresentado o tempo médio necessário para encontrar a alocação ótima de cada requisição aceita nos experimentos realizados. Em todos os cenários, a média permanece abaixo dos 4,5 segundos do início ao fim dos experimentos. Apesar de pequena, é possível notar uma diferença na média de tempo entre os diferentes cenários. Os cenários em que os requisitos de segurança são considerados possuem média de tempo mais baixa do que os cenários nos quais os mesmos são ignorados. De forma similar, os cenários com requisitos de capacidade mais altos são resolvidos mais rapidamente do que os cenários em que tais requisitos são mais baixos. Tais diferenças podem ser explicadas pela diminuição no espaço de busca causada tanto por restrições relacionadas à segurança quanto pelos requisitos de capacidade mais altos. A presença de um número maior de restrições ou requisitos de capacidade tendem a diminuir o espaço de soluções factíveis, o que pode a tornar a busca pela solução ótima mais rápida. Nos cenários 1A e 1B, o tempo médio é de respectivamente 2,29 e 2,75 segundos, enquanto que nos cenários 2A e 2B os

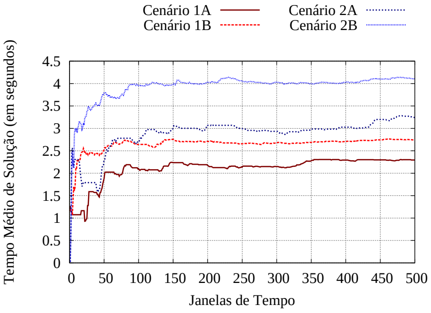


Figura 4. Tempo médio necessário para encontrar a alocação ótima nos experimentos realizados.

mesmos são de 3,24 e 4,1 segundos.

Em resumo, os resultados apresentados mostram que há um impacto significativo em termos de aceitação de requisições de redes virtuais e consumo de banda por parte das requisições aceitas ao se considerar o emprego de serviços de segurança. Esses fatores afetam negativamente a possibilidade de lucro que um provedor de infraestrutura pode obter, visto que haverá menos redes alocadas no substrato, e o custo das mesmas tenderá a ser mais alto. Portanto, julga-se importante considerar o custo necessário para prover segurança às redes virtuais no momento da alocação. Além disso, a avaliação do tempo de resolução do problema mostra que a implementação do modelo proposto é capaz de produzir resultados ótimos em um tempo adequado para uso em ambientes de produção.

5. Conclusões

Virtualização de redes é um tópico importante e que tem recebido atenção da comunidade científica e da indústria, resultando na proposta de uma série de abordagens de alocação. Mais recentemente, surgiram propostas visando prover segurança a ambientes de redes virtuais. No entanto, os autores desconhecem tentativas anteriores de combinar ambas as áreas, provendo alocação ótima e orientada a segurança de recursos de redes virtuais.

Considerando que a alocação de recursos e a segurança são igualmente importantes, desenvolveu-se um modelo que combina restrições de CPU, memória, largura de banda e localidade com requisitos de segurança. Redes virtuais podem solicitar diferentes níveis de criptografia em comunicações entre seus roteadores, visando prover confidencialidade às mesmas, ou podem exigir que seus roteadores e enlaces virtuais não compartilhem dispositivos e caminhos físicos com outras redes virtuais específicas.

Os resultados obtidos demonstram o impacto significativo causado pelo aprovisionamento de serviços de segurança na alocação de redes virtuais, salientando a importância de considerá-los no processo de mapeamento. Além disso, o modelo proposto mostra-se capaz de produzir resultados ótimos em um tempo adequado. Ainda que não sejam considerados custos adicionais de processamento e memória associados aos processos de criptografia e decriptografia, os resultados são capazes de prover uma boa visão global desse impacto. Pretende-se realizar uma revisão mais profunda de trabalhos relacionados à segurança, visando obter medidas reais de tais custos para incorporá-los no modelo. Acredita-se que isso permitirá analisar as consequências do fornecimento de serviços de segurança com uma granularidade mais fina.

Outra perspectiva para trabalhos futuros é permitir a reotimização de redes virtuais já alocadas, migrando recursos virtuais entre roteadores e enlaces do substrato. O processamento de requisições em tempo real pode levar à fragmentação dos recursos físicos, visto que as requisições não são conhecidas *a priori*. Por esse motivo, a reotimização periódica pode beneficiar o provedor de infraestrutura, diminuindo custos e permitindo que uma quantidade maior de requisições sejam atendidas. No entanto, o tempo necessário para avaliar possíveis realocações pode tornar proibitiva a obtenção de soluções ótimas. Por esse motivo, pretende-se criar um algoritmo baseado em metaheurísticas, produzindo soluções sub-ótimas porém minimizando o tempo necessário para obtê-las.

Referências

Albert, R. and Barabási, A.-L. (2000). Topology of evolving networks: Local events and universality. http://link.aps.org/doi/10.1103/PhysRevLett.85.5234. *Phys. Rev. Lett.*, 85:5234–5237.

Alkmim, G. P., Batista, D. M., and Fonseca, N. L. S. (2011). Optimal mapping of virtual networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6.

Andersen, D. (2002). Theoretical approaches to node assignment. http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps. Unpublished manuscript.

Anderson, T., Peterson, L., Shenker, S., and Turner, J. (2005). Overcoming the internet impasse through virtualization. *Computer*, 38(4):34–41.

Cabuk, S., Dalton, C. I., Ramasamy, H., and Schunter, M. (2007). Towards automated provisioning of secure virtualized networks. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 235–245, New York, NY, USA. ACM.

Cheng, X., Su, S., Zhang, Z., Wang, H., Yang, F., Luo, Y., and Wang, J. (2011). Virtual network embedding through topology-aware node ranking. In *SIGCOMM Computer Communication Review*, volume 41, pages 38–47, New York, NY, USA. ACM.

Chowdhury, N., Rahman, M., and Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783 –791.

Davy, S., Serrat, J., Astorga, A., Jennings, B., and Rubio-Loyola, J. (2011). Policy-assisted planning and deployment of virtual networks. In *Network and Service Management (CNSM), 2011 7th International Conference on*, pages 1 –8.

Fernandes, N., Moreira, M., Moraes, I., Ferraz, L., Couto, R., Carvalho, H., Campista, M., Costa, L., and Duarte, O. (2010). Virtual networks: Isolation, performance, and trends. In *Annals of Telecommunications*.

Huang, D., Ata, S., and Medhi, D. (2010). Establishing secure virtual trust routing and provisioning domains for future internet. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pages 1–6.

Kent, S. and Seo, K. (2005). Rfc 4301: Security architecture for the internet protocol. http://tools.ietf.org/rfc/rfc4301.txt.

Yu, M., Yi, Y., Rexford, J., and Chiang, M. (2008). Rethinking virtual network embedding: substrate support for path splitting and migration. *SIGCOMM Comput. Commun. Rev.*, 38(2):17–29.