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

978-3-901882-48-7

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. Similarly to the model previously mentioned, 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 in an online manner.

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:

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

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

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

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

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

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

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

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

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

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. Furthermore, each group contains 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 and memory resources were similar, and thus omitted due to

[1] As a side effect, C8 does not also 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.

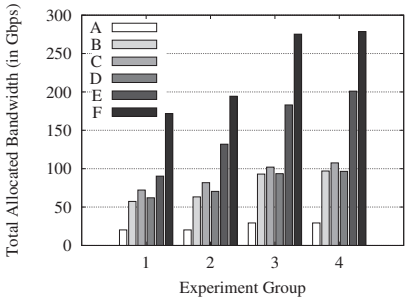

Fig. 1. Total amount of bandwidth consumed by the optimal allocation in each scenario.

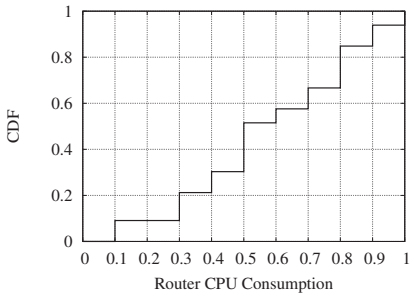

Fig. 2. Cumulative Distribution Function of CPU usage on physical routers hosting virtual routers in experiment 4F.

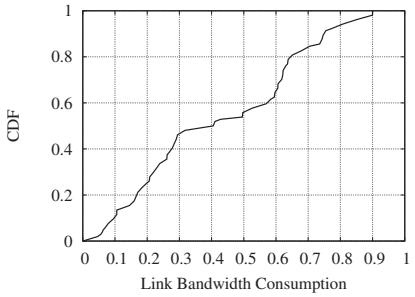

Fig. 3. Cumulative Distribution Function of bandwidth usage on physical links hosting virtual links in experiment 4F.

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 security requirements, respectively). As can be observed, providing

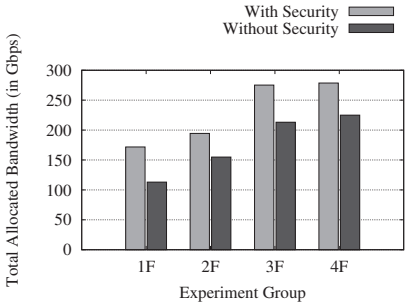

Fig. 4. Comparison between the bandwidth consumed by the optimal allocation in scenarios 1–4F with and without security requirements.

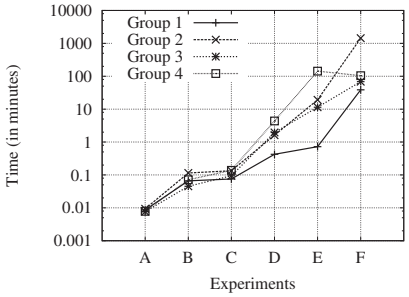

Fig. 5. Time needed to find the optimal solution in each scenario.

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