Modeling Security Requirements for VNE algorithms

Andreas Fischer, Ramona Kühn, Waseem Mandarawi, Hermann de Meer
University of Passau
Faculty for Computer Science and Mathematics
Innstraße 43, 94032 Passau, Germany
{andreas.fischer,ramona.kuehn,waseem.mandarawi,hermann.demeer}@uni.passau.de

ABSTRACT

Public and private Infrastructure as a Service (IaaS) clouds are widely used by individuals and organizations to provision flexible virtual computing resources on demand. Virtual Network Embedding (VNE) algorithms are employed in this context to provide an automated resource assignment. With multiple involved parties, security-aware Virtual Machine (VM) placement becomes highly relevant for production environments. Moreover, VNE algorithms should also consider the security requirements of the interconnections between VMs, thereby extending the problem to networks. This paper discusses security requirements of Virtual Networks (VNs) and shows how they can be modeled in VNE to map them to the provided security mechanisms in the physical network. The paper also presents an implementation of this security-aware VNE model in the public simulation platform ALEVIN, demonstrating the applicability with a realistic use case of such a model.

CCS Concepts

•Networks → Network domains; Cloud computing; Data center networks; Overlay and other logical network structures; •Computing methodologies → Modeling and simulation; Simulation tools; •Security and privacy → *Security requirements; Formal security models; Security services; Firewalls;*

Keywords

Virtual network embedding; Cloud computing; Security; Virtual network, Substrate network, Network virtualization

1. INTRODUCTION

Network virtualization is the primary enabling technology to overcome ossification effects in todays networks. It allows network administrators to deploy multiple VNs on a single Substrate Network (SN). The respective resource assignment problem is called VNE. It describes how a VN can be embedded or mapped on the given SN. The networks can be represented as a graph with nodes connected by links, where the virtual nodes or links pose demands for certain resources. Then, they have to be mapped on appropriate hardware components offering these resources.

So far, VNE approaches focus mostly on optimizing the performance of the embedding. Approaches to make the embedding more secure remain mostly abstract and do not easily lend themselves to practical application. Nevertheless, security is a major request nowadays, either to meet legal requirements, to protect own data, or for a network provider to satisfy the needs of the customers. In contrast to other approaches, this paper focuses on concrete security mechanisms like firewalls, Network Intrusion Detection Systems (NIDSs), and Trusted Hardware (TH), and discusses how they can be taken into account during the embedding process. It considers VNE problems with unsplittable links and focuses on offline evaluation of VNE algorithms.

In this setting, it is analyzed to what extent common security mechanisms require new constraints that have to be considered for the embedding process. The paper demonstrates how these constraints can be incorporated into the common algorithm evaluation process with minimal changes to the embedding algorithms themselves. This enables researchers to easily extend their evaluations to include new problems including security requirements, thereby speeding up research in this area. This paper shows the implementation and a proof-of-concept embedding that uses a common VNE simulator tool called ALEVIN [9, 3]. The paper presents the concepts of VNE and an overview of ALEVIN, then an extension to the tool to support security requirements of VNs. An overview of typical security requirements and a use case are discussed to show the usability of the tool to support security in virtual environments.

The remainder of this paper is structured as follows: In Section 2, background information about the VNE problem is provided and the related work on the modeling of security mechanisms for VNE is discussed. Section 3 describes the problem of modeling security mechanisms for VNE. An example use case is presented, highlighting the relevance of security-aware embedding algorithms. The modeling of the respective security requirements with appropriate constraints is analyzed in Section 4. Different types of requirements are identified and the formulation as embedding constraints is presented. The concrete implementation of the security mechanisms and constraints in a common VNE simulation framework is described in Section 5. In addition, the necessary changes to the simulation framework and the em-

ValueTools '16 October 26–28, 2016, Taormina, Italy

DOI: 10.1145/1235

VALUETOOLS 2016, October 25-28, Taormina, Italy

DOI 10.4108/eai.25-10-2016.2266673

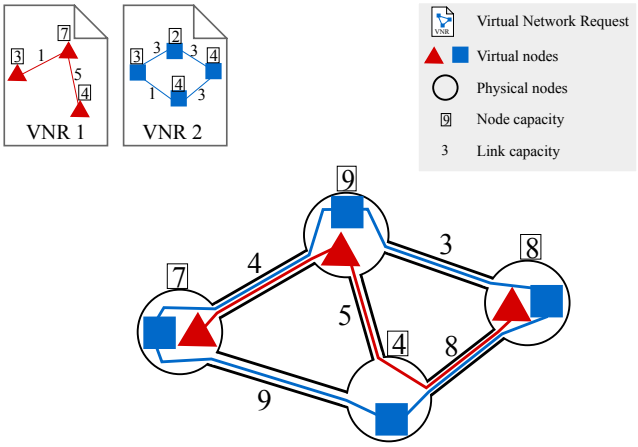

Figure 1: Example for a VNE scenario (adapted from Fischer et al. [8])

bedding process are presented. Finally, Section 6 provides a conclusion and a discussion of next steps.

2. BACKGROUND AND RELATED WORK

The VNE problem deals with the question how nodes and links of a VN should be mapped to the nodes and links of a SN. Both nodes and links are considered to provide resources in the case of substrate elements, and pose respective demands in the case of virtual elements.

VNs come in the form of Virtual Network Requests (VNRs): requests of users for instantiation of a particular network. It is the task of a VNE algorithm to decide whether all requests can be supported by the given SN, and if so, how the individual elements should be mapped.

Fig. 1 shows an example for this problem with two VNRs to be mapped onto a SN with four nodes. A possible embedding is already depicted. Each virtual element poses a demand on its respective substrate element, and the sum of the demands of all virtual elements hosted on a substrate element may not exceed the available resources. If not enough resources are left to support a VNR, it is rejected and only a partial solution can be found. Solutions (partial or complete) are not necessarily unique.

The VNE problem is known to be $\mathcal{NP}$-hard [2, 1]. Efficient approaches, therefore, require the application of heuristics. Many VNE algorithms have been proposed up to now in the literature [8]. Most VNE algorithms are performance-oriented, optimizing for cost or for the number of VNRs to be mapped onto a SN. In contrast, security considerations have been a relatively new addition to that topic [10, 12]. Often, these approaches model only abstract security requirements in a sense of different security levels or classes. One application of this is data protection: It is shown that the concept of such security classes can help to define a control flow for different kinds of data (business or personal) and define a location-based resource allocation to fulfill legal requirements [7]. This means that the virtual resources are mapped on hardware resources that comply with an adequate level of protection. However, the definition of such security classes often remains abstract. In contrast, the concrete definition of common security mechanisms and requirements such as firewalls, NIDS, or specific TH are considered in this paper. Respectively, their modeling for use in common VNE simulation software is described.

Multiple simulation frameworks are available for VNE evaluation. Yu et al. propose the VNE Simulator [15]. Chowdhury et al. describe Vineyard [6, 5]. Papagianni et al. propose CVI-Sim [14]. In this paper, the ALEVIN simulator [9, 3] is used, due to its flexibility and extensibility. It provides in particular a flexible resource/demand model that can be adapted to model security requirements. ALEVIN also offers a broad variety of functions such as developing new algorithms and defining new metrics for the comparison and evaluation of VNE algorithms. A wide set of algorithms and metrics are already implemented. Furthermore, ALEVIN is in principle scalable for large networks. This scalability depends, however, on the embedding algorithms being used. In addition, ALEVIN provides different methods for creating physical and virtual networks and assigning resources and demands to network entities. These methods are Graphical User Interface (GUI), input data files (XML), and random network generators with the required parameters.

3. PROBLEM DESCRIPTION

The implementation of security mechanisms and constraints is highly relevant for experimentation with VNE algorithms. This requires a proper formulation of these requirements for VNE simulation. In this section, first an overview is provided, specifying the necessity to formulate concrete security features. A motivational example helps to delineate the problem. The respective requirements are extracted and classified as node-based, link-based, and topology-based requirements.

3.1 Overview

Network security requirements are substantially different from conventional embedding constraints such as bandwidth or CPU time. They typically do not refer to a consumable resource, but rather to a specific set of features that need to be available. For example, a customer might require one of his virtual nodes to be executed in a particularly safe environment, requiring specific protection from the underlying substrate node.

An abstract approach to this problem is to define security levels for substrate and virtual nodes, requiring the embedding algorithm to match these levels appropriately (cf. [10, 12, 13]). However, in practice the concept of strictly hierarchical levels proves to be too abstract. In an environment with multiple involved parties (as it is common in cloud computing), it is difficult to find a comprehensive definition of levels that can satisfy each party.

Instead, it is more likely that customers specify a particular set of security requirements, defining for example that a certain node should be protected by intrusion detection software, or that a set of nodes should be protected by a firewall. A cloud provider, on the other hand, can label its equipment such that the customer's requests can be mapped appropriately. However, a customer cannot easily prove that the provider actually implements the required mechanisms. In this paper, we assume a trusted provides and that the provider conforms to the security aspects of the Service-level-agreement (SLA).

3.2 Motivational example

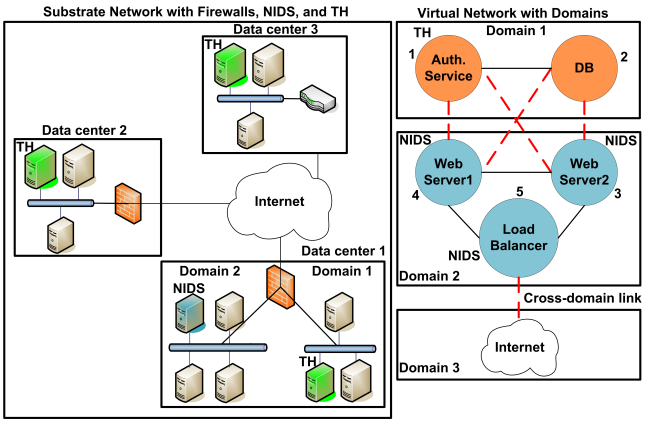

Figure 2: Cloud provider infrastructure and a VN for a web service

A motivational example is presented here to illustrate the application of security requirements in VNE. This example is depicted in Fig. 2. A cloud provider offers computing resources distributed over three data centers. Two of those data centers are protected by a firewall. One of those data centers offers two separated subnets. A client wants to implement a web service which consists of a load balancer, two web servers, a data base, and an authentication service. These components are interconnected and should be deployed in the cloud infrastructure.

Each of the components has its specific demands which have to be adhered to by the cloud provider. Some of these demands reflect the security requirements of the underlying software. For example, the web servers have to be protected from the internet by a firewall. Since a firewall cannot prevent all attacks, a NIDS should provide information about potential malicious actions. The authentication service requires TH, as it is highly security critical. Both the authentication service and the data base should be protected from the web servers by a firewall.

VNE algorithms can help to identify how the virtual infrastructure can be mapped while adhering to these requirements. However, a suitable model applicable for VNE algorithms must be found. This paper discusses how these requirements can be modeled in a public VNE simulator for experimentation with VNE algorithms.

3.3 Classification of requirements

The VNE constraints commonly investigated up to now focus mostly on quantitative values—node and link resources and demands. Popular examples are bandwidth for links and CPU capacity for nodes. Security mechanisms, in contrast, are typically qualitative in nature: A particular feature or mechanism is required from the SN. This feature or mechanism is not consumed by a virtual entity, but remains available for further nodes or links.

Security mechanisms, such as demanded by the discussed web service, can be roughly classified into three different types. There are requirements that are specific to a single node and requirements that are specific to a single link—similar to conventional constraints. However, in addition there are also more complex requirements that refer to a part of the topology. In the following, these three types are discussed in detail.

3.3.1 *Node requirements*

When a virtual node demands a security mechanism, the physical node has to offer this mechanism to be a possible candidate for mapping. Examples for this node-to-node mapping are TH, encrypted data storage and Virtual Machine Introspection (VMI). For the purpose of this paper, TH is used as an exemplary mechanism in the motivational example.

This means, if a virtual node demands TH, it can only be mapped on a physical node that offers TH. It is, however, possible to map an arbitrary number of virtual nodes on the same substrate node, as long as other resources are not exceeded. Moreover, if a virtual node does not demand the mechanism, a mapping on a physical node offering it is still possible. The virtual node does not have to use the property of the physical node. However, it has to be ensured that for all further virtual nodes demanding TH there are still enough physical nodes offering it, so that a mapping of the VN on the SN is possible.

3.3.2 *Link requirements*

There are also security mechanisms that affect the links between two nodes. A virtual link might demand that it can only be mapped on a physical link that offers a specific security mechanism. As an example, a requirement of a virtual link can be that it is only mapped on a physical link that provides data encryption.

3.3.3 *Topological requirements*

A new kind of requirement describes security mechanisms that affect not only an individual network entity, but a part of the topology. Both nodes and links are affected and the topological structure has to be taken into account. Firewalls and NIDSs are examples for these kinds of security mechanisms.

In the motivational example, several components demand the protection by a firewall. They have to be grouped into domains that identify the parts of the affected topology. For example, it is not allowed that one node protected by the firewall is connected to another node in a different domain of the network via a link that does not pass the firewall. If this were the case, the protection of the firewall would be obsolete. Therefore, this has to be prevented during embedding.

To handle this constraint, the network has to be separated explicitly into different network domains. On the one hand, there are domains which have to be protected by a firewall or a NIDS. On the other hand, there may also be domains where such a security mechanism is not needed. The traffic between those domains has to be exclusively routed through the firewall for example.

A security-aware VNE takes the requirements described above into account. Therefore, it has to recognize the topology of the network to be able to divide it such that the firewall offers full protection and is not circumvented. The way the presented requirements have to be translated so that VNE simulation frameworks are able to understand and implement them is discussed in the next section.

4. MODELING SECURITY REQUIREMENTS WITH RESOURCE-DEMAND PAIRS

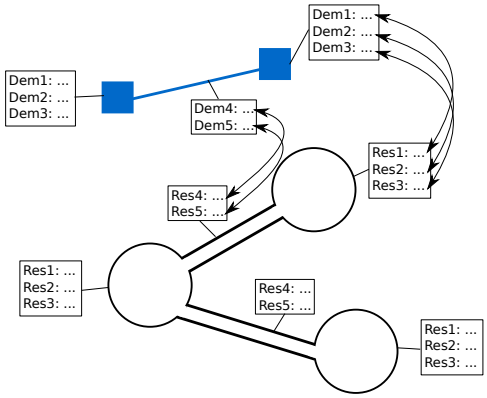

Figure 3: Resource and demand pairs

A VNE model encompasses the SN and the VNs that are mapped on it. They are modeled as graphs where nodes represent active network elements (e.g., routers) and edges represent the connectivity among them. In VNE, the consumable capabilities of the physical network entities are represented as resources that are attached to physical nodes and links, such as CPU, memory, and bandwidth. The basic model presented in Section 2 can be easily extended to contain multiple types of node and link resources.

This approach can be extended to also model specific properties of the physical network elements such as security mechanisms available in physical nodes. Likewise, the requirements of the VNs can be modeled as demands attached to virtual entities in the VNR.

In this paper, a demand is formulated to request a certain capacity of a consumable resource or request a certain property in the physical network entity. The resource/demand model is used to represent the relationship between a virtual demand and a physical resource. Resources and demands form pairs: Each demand corresponds to its respective resource.

The relationship between arbitrary resources and demands is depicted in Fig. 3. The figure shows a VN with two nodes and a link between them, and a SN with three physical nodes and two links. Each virtual node and link has demands for certain resources and each physical node and link offers certain resources. The demands and resources in the figure are depicted as Dem[ID] and Res[ID] identifiers for simplicity. The embedding algorithm has to make sure for nodes and links to pair any demand $DemN$ with the corresponding resource $ResN$.

This concept can be further extended to provide a base for modeling security capabilities and their corresponding demands. This allows to model security requirements of VNs. Here, the same resource/demand methodology is used to build a prototype that represents the security requirements. Mapping of these demands does not occupy capacity. The mapping of security requirements of virtual nodes rather depends on available properties in the corresponding physical nodes. Likewise, the mapping of the security requirements of the virtual links depends on the type of the requirements and has to check for certain properties along the physical path that maps a particular virtual link.

Here, the concepts described in the motivational example are discussed and it is shown how the resource/demand model can be adapted to implement them. The concept of resources is re-interpreted to create "pseudo-resources" that are not consumed by their corresponding demand. This allows to model THs, NIDSs, and firewalls:

- TH: A TH provides a trusted computing base to the hosted VMs such as a virtualized Trusted Platform Module (TPM) (Berger et al. [4]), for example. This is a simple node-based requirement that can be modeled by creating a special resource/demand pair "trusted hardware" in which the demand does not consume the resource.
- NIDS: A NIDS, as discussed above, represents a topological requirement. However, when the requirement is reformulated from "A NIDS is present" to "The node is protected by a NIDS", the requirement can actually be reformulated as a node-based requirement. It is then modeled similar to a TH node.
- Firewall: The firewall is more complex to model. It is not defined explicitly in the VNR. Instead, the VNR has to specify the respective network domains that should be protected and separated from each other. Using this information, Cross-domain Links (CDLs) can be identified by the embedding algorithm. As such, the demands in the VNR actually refer to domains.

  The SN, on the other hand, provides firewall nodes. This can be simply modeled with a "firewall" resource. Any CDLs is required to cross such a firewall node to ensure that nodes are properly separated. Intra-domain links, on the other hand, are preferably mapped to the same subnet.

The resource/demand model can be used to model these requirements. While TH and NIDS are straightforward to implement, the concept of firewalls requires more work. The disparate resource/demand pair has to be combined properly. Appropriate checks for CDLs have to be performed. An implementation of these mechanisms in a public VNE simulation framework is described in the following section.

5. IMPLEMENTATION OF SECURITY REQUIREMENTS CHECKS FOR VNE SIMULATION

The evaluation of VNE algorithms under security constraints requires the implementation of security requirements checks in the employed simulation framework. Here, the implementation in the ALEVIN framework is discussed and the realization of the use case presented in Subsection 3.2 is demonstrated.

5.1 Implementation of resource/demand pairs

The resource/demand model is implemented in ALEVIN using the visitor pattern to represent the occupying relationship between a virtual demand and a physical resource. ALEVIN has a generic structure that makes it easy to add new resource/demand pairs. Here, a set of resource/demand pairs that represent the basic security requirements needed for the motivational example is added.

As discussed in Section 4, security requirements that are specific to individual nodes and links can be modeled by

adapting the concept of resource/demand pair. Qualitative security requirements such as "*needs protection by a NIDS*" are matched with respective security features such as "*offers protection by a NIDS*". For the presented scenario, a NIDS demand and a TH demand are implemented.

Topological requirements prove to be more involved, though. Here, the simple resource/demand model has to be extended. The simulator has to check the validity of a particular mapping between a virtual link and its respective path in the SN. The mapping is considered valid only if the path can satisfy the security requirements of the virtual link. Firewall demands are implemented through the definition of different domains. These domains are represented as identifiers that are attached to the nodes. Firewall resources are attached to the respective substrate nodes.

However, in addition to this, a check for CDLs is necessary. As an example, a check for firewall constraints is presented in Fig. 4. The check is performed during the link mapping stage and forces all CDLs to go through a firewall. First, the algorithm filters the VNR to find CDLs by comparing the domain identifiers of the source and destination nodes of the link. Then, for each link, a set of possible physical paths is selected according to the link mapping method. The possible paths are then checked to assert if at least one of the nodes along the path provides a firewall service.

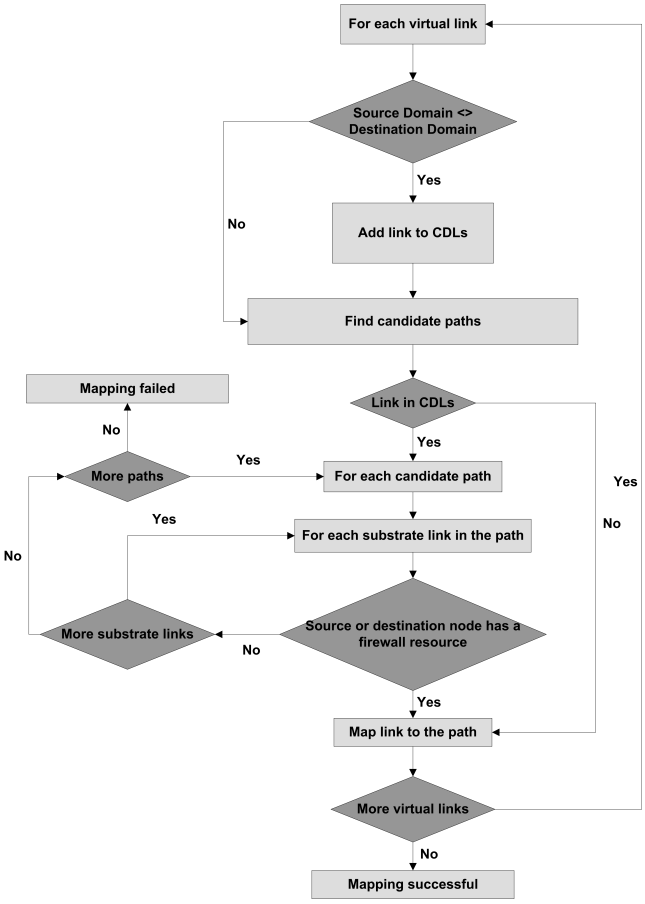

Figure 4: Embedding algorithm for CDLs

The generic embedding algorithm enforces in particular the following embedding constraints: The virtual domain is not split by a firewall and is mapped in one physical domain. CDLs are forced to go through a firewall. Virtual nodes that require a TH are mapped only to substrate nodes that offer it. Virtual nodes that require a NIDS are mapped only in domains in which at least one substrate node offers NIDS.

The implementation in the simulator itself allows to stay agnostic of the employed VNE algorithm. It is, thus, possible to evaluate scenarios with security constraints with any VNE algorithm[1].

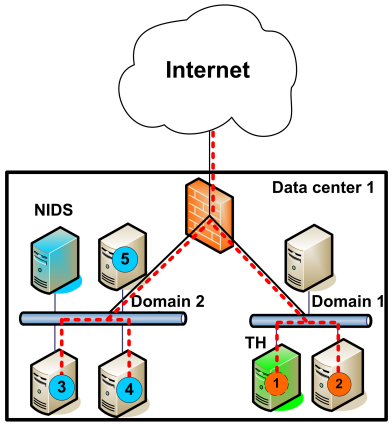

Figure 5: Motivational scenario for security-aware VNE

5.2 Realization of the motivational scenario

Fig. 5 shows the results of the mapping when implementing the motivational scenario for security-aware VNE showed in Fig. 2. In the original figure, the SN includes three different data centers. Domains are defined to identify the subnets that are protected by a firewall. Two nodes in the SN are labeled as firewall nodes and three other nodes are labeled as providing TH. The virtual network represents the specified web service with three different domains: Authentication and database, load balancer and web servers, and the general internet. The load balancer should connect to the internet to receive the web requests from users of the service. Intra- and inter-domain virtual links are depicted. The authentication server demands a node with TH. The mapping results of the VN on the SN are depicted in Fig. 5. To ensure readability, only mapped CDLs are represented.

The depicted scenario is realized in ALEVIN to test the functionality of the new security-aware VNE structure and algorithm. For demonstration, a commonly known mapping algorithm is used to perform the actual embedding. Here, the vnmFlib algorithm by Lischka and Karl [11] has been chosen. When the original topology does not contain firewall resources, the mapping procedure will not succeed since CDLs can only be mapped over nodes containing a firewall. However, when a firewall is added to the node that connects the first data center to the internet, the mapping is successful. Fig. 6 depicts the ALEVIN GUI with the constructed example topology and the procedure of adding the property of a firewall to a substrate node. For simplicity, only CDLs are shown in the VNR, and only links to the firewall in the

[1]It should be noted, though, that algorithms that do not optimize for security constraints will likely produce suboptimal results in many cases

data centers and internet links are shown in the SN. The CPU and bandwidth capacities and demands are included in the topology and allow the mapping, but are neglected from the figure for clarity.

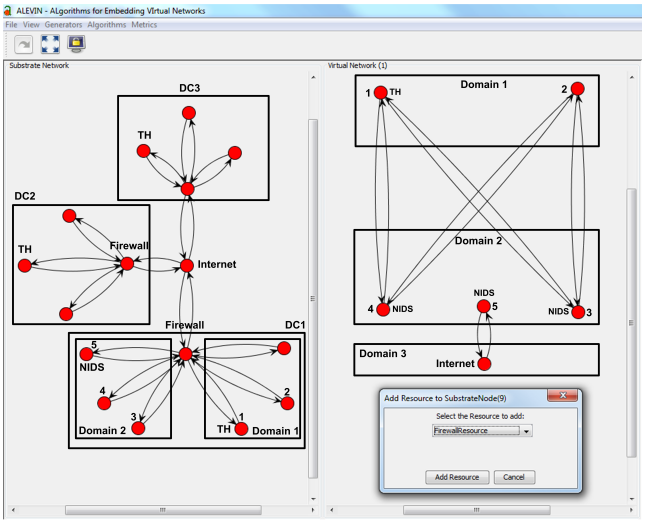

Figure 6: Adding a firewall-resource to ALEVIN

6. CONCLUSION AND FUTURE WORK

Security requirements are highly relevant for VNs deployed in public environments. It is necessary to extend VNE to incorporate these requirements and the respective security capabilities of the SN so that the embedding algorithms can satisfy them. Preferably, this is done without having to change the embedding algorithms, themselves. This paper introduced some security requirements of VNs and presented a generic methodology for modeling them. Topological constraints were identified as a new type of constraint that requires additional support by the simulation framework. The paper demonstrated a proof-of-concept implementation of a security-aware VNE model in the ALEVIN simulator, showing that, with some modifications, the existing resource/demand model can be adapted to implement security requirements. A motivational scenario that represents a web service has been discussed and implemented to demonstrate the applicability of the concept.

Future work will focus on generalizing the security constraint model to be able to adapt to more types of security requirements. Moreover, now that a generic implementation is available, multiple VNE algorithms can be evaluated in a security-aware environment.

7. ACKNOWLEDGMENTS

The research leading to these results was supported by the "Bavarian State Ministry of Education, Science and the Arts" as part of the FORSEC research association.

8. REFERENCES

[1] E. Amaldi, S. Coniglio, A. M. Koster, and M. Tieves. On the computational complexity of the virtual network embedding problem. *Electronic Notes in Discrete Mathematics*, 52:213–220, 2016. {INOC} 2015 – 7th International Network Optimization Conference.

[2] D. G. Andersen. Theoretical approaches to node assignment. Unpublished Manuscript, Dec. 2002.

[3] M. T. Beck, A. Fischer, F. Kokot, C. Linnhoff-Popien, and H. De Meer. A simulation framework for virtual network embedding algorithms. In *6th International Telecommunications Network Strategy and Planning Symposium (Networks 2014)*, pages 1–6. IEEE, Sept. 2014.

[4] S. Berger, R. Cáceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn. vtpm: Virtualizing the trusted platform module. In *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, USENIX-SS'06, Berkeley, CA, USA, 2006. USENIX Association.

[5] M. Chowdhury, M. Rahman, and R. Boutaba. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. *Networking, IEEE/ACM Transactions on*, 20(1):206–219, Feb 2012.

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

[7] B. Doll, D. Emmerich, R. Herkenhöner, R. Kühn, and H. de Meer. *On Location-determined Cloud Management for Legally Compliant Outsourcing*, pages 61–73. Springer Fachmedien Wiesbaden, Wiesbaden, 2015.

[8] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach. Virtual network embedding: A survey. *IEEE Communications Surveys and Tutorials*, 15(4):1888–1906, 2013.

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

[10] A. Fischer and H. De Meer. Position paper: Secure virtual network embedding. *Praxis der Informationsverarbeitung und Kommunikation*, 34(4):190–193, 2011.

[11] J. Lischka and H. Karl. A virtual network mapping algorithm based on subgraph isomorphism detection. In *VISA '09: Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pages 81–88, New York, NY, USA, 2009. ACM.

[12] S. Liu, Z. Cai, H. Xu, and M. Xu. Security-aware virtual network embedding. In *Communications (ICC), 2014 IEEE International Conference on*, pages 834–840, June 2014.

[13] S. Liu, Z. Cai, H. Xu, and M. Xu. Towards security-aware virtual network embedding. *Computer Networks*, 91:151 – 163, 2015.

[14] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervello-Pastor, and A. Monje. On the optimal allocation of virtual resources in cloud computing networks. *Computers, IEEE Transactions on*, 62(6):1060–1071, June 2013.

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