Computer Communications 82 (2016) 13–27

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A toolset for efficient privacy-oriented virtual network embedding and its instantiation on SDN/OpenFlow-based substrates

Leonardo Richter Bays*, Rodrigo Ruas Oliveira, Luciana S. Buriol, Marinho Barcellos, Luciano Paschoal Gaspary*

Institute of Informatics, Federal University of Rio Grande do Sul (UFRGS), Caixa Postal 15064, 91501-970 Porto Alegre, Brazil

ARTICLE INFO

Article history:
Received 29 July 2015
Revised 1 December 2015
Accepted 13 February 2016
Available online 23 February 2016

Keywords:
Network virtualization
Virtual network embedding
Privacy
Software-defined networking

ABSTRACT

Network virtualization has become increasingly popular in recent years. It has the potential to allow timely handling of network infrastructure requests and, after instantiated, their lifecycle. In addition, it enables improved physical resource utilization. However, the use of network virtualization in large-scale, real environments depends on the ability to adequately map virtual routers and links to physical resources, as well as to protect virtual networks against security threats. With respect to security, mechanisms supporting confidentiality and privacy have become essential in light of recent discoveries related to pervasive electronic surveillance. In this paper we propose a set of tools to efficiently embed virtual networks with privacy support and to allow their real instantiation on top of SDN/OpenFlow-based substrates. This toolset unfolds into three main contributions: (a) an exact VNE model suitable for smaller networks, which also serves the purpose of determining an optimality baseline; (b) a heuristic VNE algorithm, which features precise modeling of overhead costs of security mechanisms and handles incoming requests in an online manner; and (c) a VNE to SDN/OpenFlow translation mechanism, which takes as input the outcome of the heuristic VNE algorithm and produces a set of coherent OpenFlow rules to guide the real instantiation of the mapped virtual networks. We present a detailed performance comparison between the proposed heuristic and the optimization model. The obtained results demonstrate that the heuristic algorithm is able to find feasible mappings in the order of seconds even when dealing with large network infrastructures. Finally, we demonstrate how mappings generated by our heuristic VNE algorithm may be implemented in practice as well as assess the technical feasibility of this process.

1. Introduction

Network virtualization provides a higher level of dynamicity for Infrastructure Providers (InPs), by simplifying the process of network instantiation and removal. This enables InPs to promptly create networks that are specifically tailored to the needs of distinct customers, as well as to improve physical resource utilization. Such features have rendered network virtualization increasingly popular in recent years.

Although the benefits that may be achieved through the employment of network virtualization are clear, its use in large-scale, real environments depends largely on two factors. Firstly, resource allocation must be handled in a timely and optimal manner. This is considered an NP Hard problem, due to its similarity to the multi-way separator problem [1]. Secondly, the shared use of routing devices and communication channels by a number of different entities introduces security-related concerns. Without adequate protection, users from a virtual network might be able to gain unauthorized access to data being transmitted through other virtual networks, violating the privacy of the entities that own those networks [2].

The recent discovery of pervasive electronic surveillance has highlighted privacy concerns in network infrastructures. In the context of network virtualization, which relies on the sharing of physical resources and is used in key contexts such as data centers and network providers, these concerns are even more exacerbated. For this reason, it is extremely important to consider the provision of security mechanisms in order to maintain a desired level of privacy in virtual network environments. Even though security can be implemented on a per-application basis, we deem important to provision it on the link or network layer so that it applies to all traffic and is somewhat native and transparent to users.

* Corresponding authors. Tel.: +55 51 3308 9450.
E-mail addresses: lrbays@inf.ufrgs.br (L. Richter Bays), ruas.oliveira@inf.ufrgs.br (R. Ruas Oliveira), buriol@inf.ufrgs.br (L. S. Buriol), marinho@inf.ufrgs.br (M. Barcellos), paschoal@inf.ufrgs.br (L. Paschoal Gaspary).

http://dx.doi.org/10.1016/j.comcom.2016.02.007
0140-3664/

Although related work exists in the area of virtual network embedding, little has been done towards reconciling efficient resource mapping and satisfaction of security requirements. Early work in the area focuses on features such as path splitting, migration, and improved coordination between router and link mapping [3,4]. Subsequently, some authors have focused on taking into account a wider array of operational constraints related to the instantiation of virtual networks [5] or formulating more advanced strategies for improved resource distribution [6,7]. Last, there have also been efforts towards ensuring different degrees of survivability for embedded networks [8–10]. However, to the best of our knowledge, the issue of taking into account precise overhead costs of privacy-enabling mechanisms in the embedding process has yet to be approached.

The motivation to properly tackle the aforementioned issue is threefold. First, InPs need to be able to host a large number of virtual networks sharing the same physical substrate while preserving the confidentiality of each network, leading to an increased level of privacy. 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.

In this paper we consolidate a toolset designed to efficiently embed virtual networks with privacy support and to allow their real instantiation on top of SDN/OpenFlow-based substrates. The toolset includes three main components: an exact privacy-aware VNE model; a heuristic algorithm that is capable of sub-optimally embedding large virtual networks with privacy support in a time frame limited to the order of seconds; and a mechanism that takes as input the outcome of the VNE algorithm and translates it into a set of coherent OpenFlow rules to guide the real instantiation of the mapped virtual networks.

In comparison to our previous seminal work on this subject [11], in this paper we present a substantially improved version, in terms of the quality of the computed mappings, of our simulated annealing-based VNE algorithm. This is accomplished by employing a logarithmic cooling schedule (rather than a linear one), a first-improvement-based local search (as opposed to a random walk), and running multiple local searches per temperature change. As demonstrated in our evaluation, these changes have led to significant progress, rendering the results obtained through the heuristic method notably closer to those produced by the optimal model. Additionally, in this paper we introduce a VNE to SDN/OpenFlow translation mechanism, which is supported by a software tool that enables InP administrators to easily visualize and modify mappings produced by our virtual network embedding approaches (*i.e.*, exact model and heuristic algorithm). More importantly, the mechanism/tool ultimately generates the set of SDN/OpenFlow rules needed to instantiate the desired virtual networks taking into account the proposed mappings and any administrator-made modifications. We demonstrate how mappings generated by our heuristic VNE algorithm may be implemented in practice as well as assess the technical feasibility of this process.

The remainder of this paper is organized as follows. In Section 2 we present our heuristic method, as well as a baseline ILP model. In Section 3 we report the results of an extensive evaluation we carried out, including a detailed comparison of both the heuristic method and the optimal model. In Section 4 we propose and evaluate a mechanism to deploy the computed virtual network mappings on top of real physical networks, using an SDN-based substrate as case study. In Section 5 we discuss virtual network embedding approaches proposed in previous investigations. Last, in Section 6 we present final remarks and perspectives for future work.

2. Privacy-oriented virtual network embedding

In this section, we explain the assumptions behind our proposed privacy-oriented embedding 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 consider a scenario in which an infrastructure provider supplies virtual networks to a number of customers. In order to request the creation of a virtual network, these customers 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 our model or algorithm indicates the optimal or near-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). Moreover, we consider a scenario in which virtual networks are instantiated by means of full virtualization or paravirtualization techniques. Therefore, we do not consider costs of operations associated with other types of network virtualization (such as VLANs).

With regards to security, the main threat considered in this paper is the theft of information through the interception of packets being exchanged between virtual routers. This type of attack is commonly known as “sniffing” or “eavesdropping”. While any network is susceptible to this type of attack, it is an even more significant concern in network virtualization environments. As physical resources are shared among different virtual networks, a single compromised router or link may be used to obtain confidential information from multiple customers. Security mechanisms such as Virtual Private Networks (VPNs) secured by IPSec [12] are able to mitigate this threat through the use of encryption. Therefore, in this paper, we assume that data confidentiality is achieved through this technology.

2.1. 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. For ease of reference, the symbols used in this section are listed in Table A.1 (Appendix A).

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.