AutoEmbed: Automated Multi-Provider Virtual Network Embedding

David Dietrich
Leibniz Universität
Hannover, Germany
david.dietrich
@ikt.uni-hannover.de

Amr Rizk
Leibniz Universität
Hannover, Germany
amr.rizk
@ikt.uni-hannover.de

Panagiotis Papadimitriou
Leibniz Universität
Hannover, Germany
panagiotis.papadimitriou
@ikt.uni-hannover.de

ABSTRACT

We present AutoEmbed, a fully-automated framework for VN embedding across multiple substrate networks. To automate VN embedding, AutoEmbed deploys functions over three layers: (i) Service Providers, (ii) VN Providers, and (iii) Infrastructure Providers (InPs). AutoEmbed enables VN Providers to partition VN requests among multiple substrate networks based on resource and network topology information that is not treated as confidential by InPs. Subsequently, each VN segment is mapped by the corresponding InP onto its substrate network.

AutoEmbed enables the evaluation of various aspects of multi-provider VN embedding, such as the efficiency and scalability of embedding algorithms, the impact of different levels of information disclosure on VN embedding efficiency, and the suitability of VN request specifications.

Categories and Subject Descriptors

C.2.1 [Computer Communication Networks]: Network Architecture and Design

Keywords

Network virtualization, resource assignment, performance evaluation

1. INTRODUCTION

Network virtualization comprises a viable solution for the concurrent deployment of service-tailored network slices on top of shared physical infrastructures. Wide-area virtual network (VN) deployment poses the need for VN embedding across multiple substrate networks. To this end, network virtualization architectures have introduced a layer of indirection, namely VN Provider (VNP), between Physical Infrastructure Providers (InPs) and Service Providers [6, 8]. The VNP discovers and selects resources from multiple InPs, stitches together VN segments acquired from different InPs

and eventually delivers a wide-area VN that can be configured and operated by the Service Provider.

Multi-provider VN embedding entails several challenges. In particular, the VNP will have to embed VN requests with limited visibility on the substrate networks, since InPs will not be willing to disclose detailed (e.g., router-level) topology information and their resource availability.

Given the challenges faced by VNPs and the lack of evaluation environments, we present AutoEmbed, a fully automated framework for VN embedding across multiple substrate networks. AutoEmbed comprises a realistic evaluation environment for VN request partitioning and intra-domain mapping algorithms. Furthermore, AutoEmbed can be used for the investigation of various VN embedding aspects, such as the impact of information disclosure on VN embedding efficiency and the suitability of different VN request descriptions (e.g., topology-based vs. traffic-matrix based VN requests). VN embedding insights gained by AutoEmbed can assist InPs and potential VNPs in configuring their information disclosure policies and adjusting their cost models.

2. FUNCTIONALITY

AutoEmbed functionality is distributed over the Service Provider, the VNP, and the InPs. For each InP, AutoEmbed delegates all control and embedding functions into a centralized management node. AutoEmbed relies on a substrate network model, in which the network topology is represented as a weighted directed graph. Synthetic topologies generated with tools, such as the IGen [2], can be imported. AutoEmbed provides a user interface for the configuration of substrate resource attributes and their associated cost.

Multi-provider VN embedding comprises the following functions:

VN Request Submission. VN requests, submitted by the Service Provider, consist of virtual node specifications and the bandwidth demands between the virtual nodes. AutoEmbed allows the specification of the location for each requested virtual node. AutoEmbed supports the submission and processing of topology-based and traffic matrix based VN requests, providing different levels of VN specification to Service Providers. VN lifetime and the arrival rate of VN requests can be configured using a graphical interface.

VN Request Partitioning. The VNP partitions incoming VN requests among the participating InPs. VN request partitioning is carried out based on the (non-confidential) information disclosed by InPs. AutoEmbed by default uses the

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCOMM'13, August 12–16, 2013, Hong Kong, China.
ACM 978-1-4503-2056-6/13/08.

Figure 1: Visualisation of traffic matrix based VN request, VN request partitioning, VN segment mapping, and reference embedding.

binary integer program presented in [4], which minimizes the cost for the Service Provider. The VN partitioning solver, which is implemented using the GNU Linear Programming Kit (GLPK) [1], provides a set of VN segments, where each virtual node is mapped onto a peering node. Each VN segment is subsequently relayed to the corresponding InP for its mapping. AutoEmbed can integrate other VN partitioning methods (e.g., [5]).

VN Segment Mapping. Each VN segment is mapped by the InP onto its substrate network with respect to the virtual node and link capacity requirements. Since VN segment mapping is decoupled from VN request partitioning, any existing mapping algorithm can be employed. One requirement is that each VN segment mapping should comply with the virtual node to peering node bindings defined by the preceding VN partitioning step. Such bindings are already supported by existing VN mapping algorithms [3, 4]. The subsequent operations required for the VN deployment, such as VN segment stitching and VN setup, are exemplified in previous work [6, 7].

AutoEmbed provides the visualisation of VN request partitioning and VN segment mapping onto substrate networks (Fig. 1). To compare the efficiency of multi-provider VN embedding, AutoEmbed computes and visualizes a reference embedding, in which all substrate resource and topology information is available to the VNP (i.e., “best-case” scenario). The reference embedding essentially corresponds to VN request mapping onto a single substrate network. As such, AutoEmbed uses the same method employed for VN segment mapping. To quantify VN embedding efficiency, AutoEmbed computes and displays the VN embedding cost, the VN acceptance rate, and the average number of hop count for virtual links. Detailed statistics on VN embedding are maintained in local repositories.

3. DEMONSTRATION

In the demonstration, we visualize and assess the efficiency of VN embedding across three InPs. We use separate laptops (i.e., five in total) for the Service Provider, the VNP, and for each one of the InP management nodes. The demo setup is depicted in Fig. 2. For each substrate network, we use pre-configured topologies (illustrated in the second row of Fig. 1) and peerings between the InPs. Informa-

Figure 2: Demo setup.

tion available to the VNP includes the geographic coverage of each InP, one instance of each offered virtual node type (inline with Amazon EC2), the InP peerings, the location of the peering nodes and the bandwidth cost for the links between the peering nodes. VN request partitioning (at the VNP) and VN segment mapping (at the InPs) are carried out using the methods in [4].

The main goal of the demo is to quantify the implications of limited information disclosure (LID) on VN embedding efficiency. Using traffic matrix based VN requests, we compare the VN embeddings under LID with the reference embedding (i.e., where all substrate information is accessible by the VNP) in terms of embedding cost and VN acceptance rate. Based on the VN embedding statistics provided by AutoEmbed, we show that LID incurs additional embedding cost which stems from the increased hop count of virtual links.

4. REFERENCES

- [1] GNU Linear Programming Kit, <http://www.gnu.org/software/glpk/>.
- [2] IGen, <http://informatique.umons.ac.be/networks/igen>.
- [3] N. Chowdhury et al., Virtual Network Embedding with Coordinated Node and Link Mapping, IEEE INFOCOM 2009.
- [4] D. Dietrich, A. Rizk, and P. Papadimitriou, Multi-Domain Virtual Network Embedding with Limited Information Disclosure, IFIP Networking, May 2013.
- [5] I. Houidi et al., Virtual Network Provisioning Across Multiple Substrate Networks, Computer Networks, Vol. 55, No. 4, March 2011.
- [6] G. Schaffrath et al., Network Virtualization Architecture: Proposal and Initial Prototype, ACM SIGCOMM VISA 2009.
- [7] C. Werle et al., Building Virtual Networks Across Multiple Domains, Poster, ACM SIGCOMM 2011.
- [8] Y. Zu et al., Cabernet: Connectivity Architecture for Better Network Services, ACM ReArch 2008.