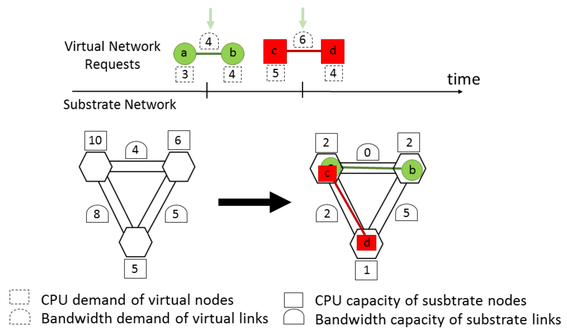
Secure and Dependable Virtual Network Embedding

{Luís Ferrolho, Max Alaluna, Nuno Neves, Fernando M. V. Ramos}
lferrolho,malaluna@lasige.di.fc.ul.pt, nuno@di.fc.ul.pt, fvramos@ciencias.ulisboa.pt
LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal

Introduction & Motivation

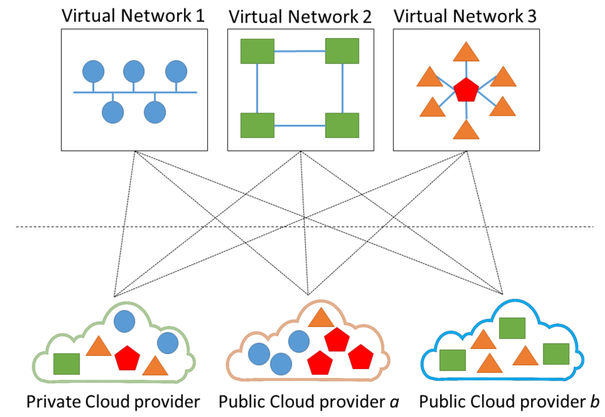


Embedding of virtual networks

In network virtualization, the Virtual Network Embedding (VNE) problem deals with finding an effective mapping of virtual nodes and links onto a substrate network. This problem is traditionally formulated with the objective of maximizing network provider revenue by efficiently embedding incoming virtual network (VN) requests. A mostly unexplored subject on this problem is security [1]. With the advent of **network virtualization platforms** [2], cloud operators now have the ability to extend their cloud computing offerings with VNs. To shift their workloads to the **cloud**, tenants trust their cloud providers to guarantee that their workloads are secure and available. Unfortunately, there is an **increasing number of evidence that problems occur** [3]. Thus, **security and dependability** are becoming critical factors that should be considered by **VNE** algorithms.

Contribution

We propose a VN embedding solution that considers **security and dependability** as first class citizens. For this purpose, we introduce specific security and dependability constraints. To further extend the resiliency properties of our solution, we **assume a multiple cloud provider model** (e.g., one based on nested virtualization). By not relying on a single cloud provider we avoid internet-scale single points of failures, avoiding cloud outages by replicating workloads across clouds. In addition, we can enhance security by leaving sensitive workloads in the tenant's private clouds.



In a multi-cloud environment, different resources from different clouds can be used by the VNs

MIP Formulation

We formulate the problem as a Mixed Integer Program (MIP). The objective is to minimize the cost of embedding VN requests, taking into account the VN lifetime and the resource's security levels:

$$min \quad Dur_k^V [\sum_{i,j \epsilon N_k^V} \sum_{u,v \epsilon N^S} \lambda \quad fw_{u,v}^{i,j} \quad lev^S(u,v)$$

(Reduce the costs of allocated bandwidth)

$$+ \sum_{u,v \epsilon N^S} r_{u,v} \quad lev^S(u,v))$$

(Reduce the costs of allocated bandwidth for backup)

$$+ \sum_{p \epsilon N_k^V} \sum_{w \epsilon N^S} R_N(w) \quad \Theta w_{w,p} \quad lev^S(w)$$

(Reduce the costs of allocated CPU)

$$+ \sum_{w \epsilon N^S} \gamma_w \quad lev^S(w)]$$

(Reduce the costs of allocated CPU for backup)

We consider the typical flow conservation and resource capacity constraints. We further define security and dependability constraints.

Security constraints:

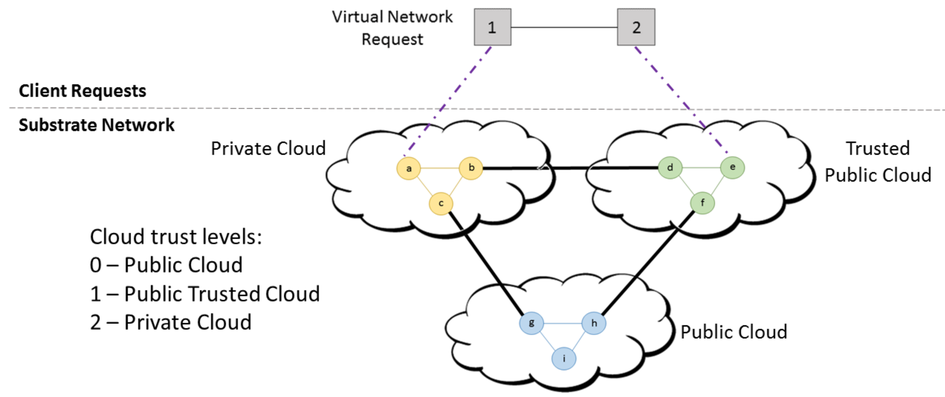
- Security level of a physical node $\geq$ security level required by the virtual node;
- Security level of a physical path $\geq$ security level required by the virtual link;
- Sensitive virtual resources should not be hosted in public clouds.

Dependability Constraints:

- A physical resource should guarantee at least the replication level required by the virtual resource;
- The physical path should guarantee at least the replication level required by the virtual link.

Solution

Security Levels for Virtual Nodes:	Security Levels for Virtual Links:	Dependability Levels for Virtual Nodes:	Dependability Levels for Virtual Links:
0 – Normal container 1 – Normal VM 2 – Secure VM	0 – No security 1 – Authenticity & Integrity 2 – Authenticity , Integrity & Confidentiality	0 – No replication 1 – Replication on the same cloud 2 – Replication in other cloud	0 – No redundancy 1 – Path redudancy



Security and dependability options for the virtual networks

Different clients/tenants have different security and dependability requirements. As such, in our solution clients specify their requirements as security and dependability levels for each virtual resource (nodes and links). In addition, the client can choose to which specific cloud its virtual resources may be mapped.

Work in progress: heuristic

Due to the complexity inherent to the embedding problem and the considerable size of the problem space in our MIP formulation, this solution is not efficient. Thus, we are currently investigating possible heuristics to this problem.

Our initial idea is to design a two-phase algorithm: the first phase for node mapping, and the second for link mapping. We also consider techniques to correlate the two phases, with node mapping taking into account the subsequent link mapping.

For node mapping we plan to explore a greedy approach that chooses a substrate node according to an utility function, while fulfilling the security and dependability constraints. The utility function takes into account the capacity of each node, namely in terms of CPU resources and availability. The proposed function includes a multiplier factor that gives more value to nodes with more links, to increase the correlation between node mapping and link mapping. For the second phase we will consider both shortest path and multi-commodity flow algorithms (the latter for when the substrate network supports path splitting/multi-path).

As security constraints we include the type of clouds where the resource can be mapped and the security level provided by the substrate nodes and links. For dependability, we consider the use of backup nodes (including replication in different clouds), path redundancy, and an availability factor as part of the utility function.

References

[1] A. Fischer et al. Virtual network embedding: A survey. 2013.

[2] T. Koponen et al. Network virtualization in multitenant datacenters. 2014.

[3] Cloud Security Alliance. The notorious nine cloud computing top threats in 2013.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No H2020-643964 (SUPERCLOUD), and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/00408/2013 (LaSIGE).