# Distributed Virtual Network Embedding

Michael Till Beck, Andreas Fischer, and Hermann de Meer

University of Passau, Innstr. 43, 94032 Passau, Germany
{michael.beck,andreas.fischer,hermann.demeer}@uni-passau.de

## 1 Introduction

The Internet in its current form is considered to be inflexible and difficult to change. Though a number of innovations and improvements (like DiffServ/IntServ, IPv6, ...) have been proposed, the actual architecture of the Internet has experienced a very slow adoption of new ideas. Network virtualization has been recognized as a tool to overcome this perceived ossification [1, 2].

Network virtualization applies virtualization concepts to network resources in order to provide an abstraction from specific hardware. By applying this principle, more flexible virtual resources are created from the underlying physical resources. The added flexibility manifests itself in the possibility to create arbitrarily structured virtual networks that do not necessarily have to reflect the topology or the properties of the underlying physical network. Networks, in this context, are considered to consist of routers and the links that interconnect them. Any number of virtual routers, arbitrarily interconnected, can then be run on a physical topology, provided there are enough physical resources in order to realize the respective virtual resources.

The problem of mapping virtual resources to physical resources in such an environment is commonly known as “Virtual Network Embedding” (VNE). A number of algorithms to solve this problem have been proposed in the literature, so far. Most of these algorithms rely on a centralized entity that will compute an optimal or near-optimal solution for a given situation. However, in realistic scenarios a centralized entity should be avoided in order to not create a single point of failure. Moreover, since the VNE problem tends to be computationally expensive, the exploitation of parallelity is expected to improve performance. Thus, it is necessary to investigate novel approaches for a distributed VNE solution.

## 2 Distributed Virtual Network Embedding

### 2.1 Motivation

Up to now, most algorithms dealing with the VNE problem operate in a centralized manner (a notable exception being the algorithm by Houidi et al. [3]). Virtualized network environments, however, are envisioned to be highly dynamic, with resources being frequently added, moved, and removed on demand. This has to be taken into account by VNE algorithms. It follows, that these algorithms have to operate in an online manner (as opposed to computing an embedding in an arbitrary amount of time beforehand). For centralized algorithms it follows that there is a single point of failure - namely the instance that computes the embedding. Moreover, the scalability of a centralized solution is questionable. Thus, developing new algorithms that operate in a distributed manner is an important goal.

### 2.2 Methodology

A centralized embedding algorithm uses an external resource to calculate the actual result. However, a distributed approach uses the hosts of the physical network itself for calculations. Work is distributed to the physical nodes, and the nodes have to cooperate to find an appropriate mapping. A key requirement of a distributed algorithm is that the physical nodes involved in an embedding operation have to cooperate and exchange notification messages to update their local state of the network. This is similar to peer-to-peer systems where nodes have to send and receive administrative messages, too. Some approaches, like the Kazaa network, try to reduce message and management overhead by partitioning the network into smaller clusters. These clusters maintain their information autonomously. A special node inside the cluster, the supernode, is selected as the leader of the cluster.

Analogous to the Kazaa approach, the substrate network can be clustered into multiple, non-overlapping parts, too. These parts can be used independently to embed virtual network requests in a distributed manner, in parallel. A node inside a cluster that has the most available CPU resources can then be selected as the clusterhead. This node is responsible for computing the actual network embedding for its part of the network. Depending on the size of the virtual network, an appropriate cluster is selected, where the virtual network should be integrated.

We use an heuristic that estimates the potential of a partition to embed a virtual network request. This is done depending on the amount of resources the virtual network request demands. For this reason, two weight values $\alpha$ and $\beta$ are introduced. $\alpha$ is used to weight the demands of CPU resources, and $\beta$ for the weight of bandwidth resources demands. A higher $\alpha$ or $\beta$ value results in a more pessimistic estimation of physical resources. This means that using higher values, the algorithm will try to embed a virtual network into bigger physical partitions, ignoring smaller ones.

### 2.3 Metrics

An embedding algorithm can be evaluated by applying various metrics to the mapping results. Some metrics can be used for all kinds of embedding algorithms. An important example for such a metric is the *revenue-to-cost* relation [4]. It describes how many substrate resources are consumed relative to the amount of virtual demands. The more substrate resources are used, the higher the probability that further networks cannot be embedded: The *virtual network acceptance rate* describes how many virtual networks had to be rejected due to unavailable substrate resources. Another important metric counts the *average length of substrate paths* that are mapped to a virtual link. In general, long paths should be avoided, because forwarding messages lead to transmission delays. Furthermore, additional bandwidth resources of substrate links become unavailable, which also affects the revenue-to-cost value. Depending on the objective of the embedding approach, it might be interesting to discover the number of nodes that are used by a mapping. This includes the nodes that are assigned to virtual nodes, but also those nodes that are part of a path, i.e., nodes that forward messages between two other nodes.

These metrics can be used for evaluating centralized and distributed algorithms. However, in a distributed scenario, additional metrics should be considered, since substrate nodes send and receive coordination messages to perform the embedding. Thus, there are additional expenses that should be taken into account. So the *total number of messages* sent through the network should be measured to estimate this kind of overhead. In this context, the *average number of messages* sent through a link and the *average length of paths* messages are send through might be of interest, too. To measure the degree of load distribution, the number of nodes that were involved into embedding calculations can be evaluated.

### 2.4 Evaluation

Based on the previously described ideas, we implemented a distributed algorithm that partitions the physical network into several parts of different size.

First evaluation results show that, compared to the approach proposed by Houidi, the number of messages sent through the network for coordinating the physical nodes can be reduced drastically. When preferring bigger partitions, the likeliness that virtual networks cannot be embedded by these partitions decreases. This means that the algorithm does not have to send more messages to find other partitions for embedding.

Furthermore, we can show that the algorithm uses a reasonable amount of physical resources for embedding the virtual networks. If we use an estimation method that is pessimistic, it will prefer bigger partitions and the underlying embedding algorithm (which is actually calculating the embedding) uses more resources.

## 3 Conclusion

Network virtualization has been adopted already in a number of different networks (e.g. PlanetLab[1], GENI[2] and G-Lab[3]) . The optimization of resource usage in a virtualized network environment is, therefore, a highly relevant topic. Algorithms that will compute an optimal embedding of virtual resources have

[1] http://www.planet-lab.org
[2] http://groups.geni.net/geni
[3] http://www.german-lab.de

to operate in a distributed manner, however, in order to ensure stability and scalability of the whole system. These algorithms have to be evaluated with new metrics, taking into account the overhead created by synchronizing nodes in the network.

**Acknowledgments.** The research leading to these results has received funding from the EC's Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) in the context of the ResumeNet and EuroNF projects (grant agreement no. 224619 and 216366).

## References

1. Anderson, T., Peterson, L., Shenker, S., Turner, J.: Overcoming the internet impasse through virtualization. Computer **38** (April 2005) 34–41
2. Feamster, N., Gao, L., Rexford, J.: How to lease the internet in your spare time. ACM SIGCOMM Computer Communication Review **37** (2007) 61–64
3. Houidi, I., Louati, W., Zeghlache, D.: A distributed virtual network mapping algorithm. In: ICC, IEEE (2008) 5634–5640
4. Lischka, J., Karl, H.: A virtual network mapping algorithm based on subgraph isomorphism detection. In: VISA. (August 2009) 81–88