On Delay-Aware Embedding of Virtual Networks

Michael Till Beck, Claudia Linnhoff-Popien
Ludwig Maximilian University of Munich
{michael.beck,linnhoff}@ifi.lmu.de

***Abstract*—Network Virtualization is seen as a key technology for the Future Internet. In fact, today, Network Virtualization is actively used by telecommunication providers. One of the key challenges in this context is the embedding of virtual networks into the substrate network topology: Virtual Network Embedding algorithms aim to assign substrate nodes and substrate paths to virtual nodes and links in an optimal way. Several embedding algorithms have been proposed in literature, pursuing various optimization goals. Most of them strive to increase cost-efficiency of the embedding. This paper discusses communication delay in the context of the Virtual Network Embedding problem. While embedding cost has already been extensively discussed, queueing delay has only sparsely been analyzed in this context. This paper introduces a delay model that is based on queueing theory and considers demands of virtual network requests. Based on this model, optimization objectives for delay-aware embedding algorithms are presented. Furthermore, delay related evaluation metrics are introduced for analyzing the effectiveness of delay-aware virtual network embedding approaches.**

***Index Terms*—Virtual Network Embedding, Delay**

I. Introduction

Network Virtualization is a promising approach to overcome the ossification of the current Internet infrastructure. Core protocols of the Internet infrastructure are perceived as being difficult to change. This is widely known as the ”IP-waist”. Network Virtualization enables infrastructure providers to separate network capacities into several isolated networks, each capable of running its own communication protocols. First, Network Virtualization has been actively used in the context of Internet testbeds like G-Lab [1] and 4WARD [2]. Today, Network Virtualization is seen as one of the most promising technologies to overcome the resistance of the Internet infrastructure towards novel core protocols. In fact, Network Virtualization is actively used by today's telecommunication providers as a tool to enhance the flexibility of their network infrastructures.

In Network Virtualiation, several virtual networks are deployed on top of a substrate network topology [3]. From an abstract point of view, substrate networks are a collection of network nodes (representing, e.g., physical servers), connected by network links (representing physical communication links, e.g., Ethernet cables). Similarly, virtual networks consist of virtual nodes and virtual links, both demanding network resources (like CPU and bandwidth capacities) provided by the substrate network.

This leads to the Virtual Network Embedding problem: The objective of an embedding algorithm is to embed virtual networks on top of a shared substrate network in an optimal (or near-optimal) way.

Several Virtual Network Embedding approaches have been discussed in literature so far. Many aim to reduce embedding cost, i.e., the amount of substrate network resources that are needed in order to embed the virtual networks. By keeping embedding cost low, the infrastructure provider is able to allocate additional virtual network requests. Besides embedding cost, another key objective in the context of telecommunication networks is to keep network delay low. Despite of the fact that several embedding approaches have been presented in literature so far, delay is only sparsely discussed in this context.

Therefore, this paper presents a delay model for future virtual network embedding approaches that considers the dynamic components of communication delay. The model is based on queueing theory and takes into account both transmission delay and queuing delay. Furthermore, delay-aware optimization objectives are discussed in this context. As discussed in this paper, embedding virtual networks in a delay-sensitive way comes with additional embedding cost. On the one hand, infrastructure providers usually aim to assign network resources in a cost-efficient way in order to increase the amount of resources that are available for future virtual network requests. On the other hand, a delay aware embedding tends to consume additional network resources. Thus, there is a tradeoff between delay-awareness and cost-efficiency. This paper motivates why both aspects should be taken into account when embedding virtual networks. Finally, evaluation metrics are discussed for measuring the effectiveness of embedding results.

II. Background and Related Work

This section shortly motivates the virtual network embedding problem and presents the network model used in this work. Furthermore, related work is discussed.

A. The Virtual Network Embedding Problem

Figure 1 depicts the virtual network embedding problem. Several virtual network requests (VNR) have to be assigned to a shared substrate network. Substrate resources are limited: E.g., substrate nodes provide CPU resources and substrate links offer bandwidth resources that can be assigned to VNRs. Virtual networks demand those resources. Virtual nodes demand CPU resources and virtual links demand bandwidth

 ISBN: 978-1-61208-377-3

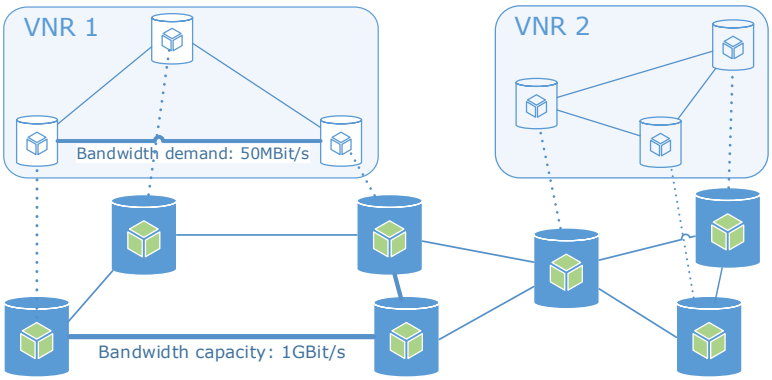

Fig. 1. The Virtual Network Embedding Problem

resources. For the embedding, virtual nodes and links have to be assigned to substrate resources offering sufficient resources. While a virtual node is assigned to just one single substrate node, a virtual link can be embedded to multiple substrate links, i.e., to a substrate path. The embedding algorithm has to ensure that each segment of that substrate path provides sufficient bandwidth resources.

The problem of optimally embedding virtual networks to the substrate network is NP-hard. Therefore, to reduce computational complexity, several heuristical embedding approaches have been introduced, aiming to solve the embedding in a nearly-optimal way [3].

B. Network Model

This subsection shortly describes the network model used in this work. It is based on the one presented in [3].

Both the substrate network and virtual networks are modeled as network graphs: A substrate network SN is represented by a set of substrate nodes N connected by substrate links L. A substrate node $n \in N$ provides CPU resources $\mathrm{res}_{\mathrm{cpu}}(n)$; a substrate link $l \in L$ offers bandwidth resources $\mathrm{res}_{\mathrm{bw}}(l)$. Similarly, the i-th virtual network request VNR^i is a set of virtual nodes N^i and links L^i. CPU demand of a virtual node $n^i \in N^i$ is modeled as $\mathrm{dem}_{\mathrm{cpu}}(n^i)$, and bandwidth demand of a virtual link $l^i \in L^i$ is modeled as $\mathrm{dem}_{\mathrm{bw}}(l^i)$. The virtual network embedding problem is composed of the node mapping problem and the link mapping problem. The node mapping step is described as a function $m_{\mathrm{node}} : N^i \rightarrow N$, and the link mapping step as a function $m_{\mathrm{link}} : L^i \rightarrow L' \subseteq L$. For a more readable representation, we will refer to $R_{\mathrm{total}}(l)$ as being the total bandwidth of a substrate link l; $R_{\mathrm{occupied}}(l)$ represents the amount of bandwidth resources that are allocated to virtual links assigned to a substrate link l, and $R_{\mathrm{available}}(l)$ refers to available bandwidth resources of that link that were not allocated so far.

C. Related Work

Many embedding approaches have been discussed in literature so far [4]. Most of them aim to reduce embedding cost in order to increase the number of networks that can be embedded onto the substrate topology. Besides cost-efficient algorithms, several other embedding approaches aiming for other optimization objectives have been proposed, focussing on energy efficiency [5], workload distribution [4], [6], resilience, etc. An extensive survey on virtual network embedding parameters and optimization objectives is given in [3].

However, only few related work on delay-aware virtual network embedding is available so far. To the best of our knowledge, there are only two publications in that direction:

Karthikeswar et al. introduce an embedding algorithm that considers the tradeoff between end-to-end delay and substrate utilization [7]. The problem is formulated as a mixed integer programming formulation and is based on a static delay model. It is assumed that communication delay is a static property of the communication links and does not depend on the utilization of the link.

Liao et al present a multi-agent approach to solve the virtual network embedding problem by considering link delay [8]. This approach aims to minimize bandwidth cost while keeping delay constraints of communication paths within defined limits. The approach considers a linear delay model.

In contrast to related work, this paper introduces, based on queueing theory, a non-linear delay model. Based on this model, delay-aware optimization objectives are formulated and several evaluation metrics are discussed.

III. Delay-Aware Virtual Network Embedding

In this section, delay-awareness is discussed in the context of the virtual network embedding problem. To this end, a delay model for substrate links is presented. Delay is modeled as a function that heavily depends on the utilization of network links. Based on this model, optimization objectives are introduced that aim to reduce network delay by avoiding highly utilized communication paths.

First, the delay model is introduced. Then, optimization objectives for future virtual network embedding algorithms are formulated. Finally, evaluation metrics are presented for measuring the effectiveness of delay-aware embedding approaches.

 ISBN: 978-1-61208-377-3

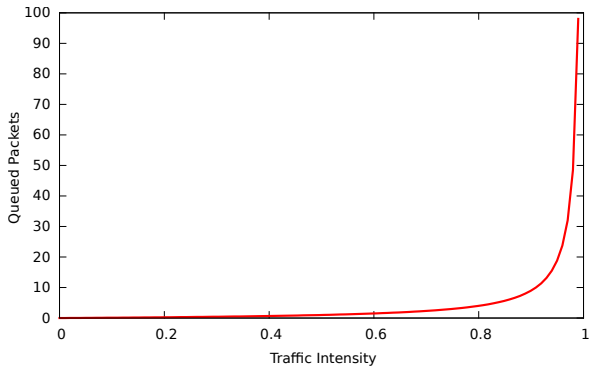

Fig. 2. The average number of queued packets rises if traffic intensity increases

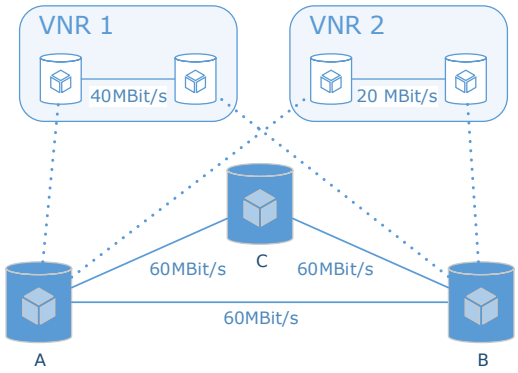

Fig. 3. Delay-Aware Virtual Network Embedding

A. Utilization-Aware Delay Model

In the following, various types of communication delay are being discussed; then, a model is derived that is applicable in the context of the virtual network embedding problem.

In telecommunication networks, there are four types of delay influencing network speed [9]:

- Processing Delay d_{proc}
  Refers to the time needed for processing a packet, e.g., extracting header information, examining how a packet has to be routed and checking whether bit-level errors occurred during transmission. Processing delay is usually in the bounds of nanoseconds and does not depend on the utilization of the routers (unlike queueing delay).
- Transmission Delay d_{trans}
  Time needed in order to transmit a packet of length L from router A to router B. Depends on the transmission rate R of the link. Transmission delay refers to the time that is needed to transmit all bits of a packet from A to B, thus, transmission delay is L/R.
- Propagation Delay d_{prop}
  Time needed to propagate a bit through the communication link. Propagation speed depends on the physical medium and is, in general, nearly equal to the speed of light.
- Queuing Delay d_{queue}
  The time a packet remains in the router's queue before it can be processed. Queuing delay of a packet depends on the number of other packets that arrived before. Queuing delay can vary significantly: If the queue is empty, the router will handle the arriving packet immediately, i.e., queuing delay is zero; if, however, many packets are waiting for transmission, queuing delay contributes heavily to the total end-to-end delay. While previous work focuses on static delay models, this paper also considers the utilization of routers in the embedding process. To this end, transmission delay is considered for the as a non-linear function.

In the following, a delay model that is applicable in the virtual network embedding domain is discussed. As mentioned before, transmission delay depends on bandwidth resources of the substrate network. Furthermore, queuing delay depends heavily on the utilization of the network link. The delay model presented here is based on a well-known utilization-aware queuing delay model: The model considers that queuing delay increases significantly when traffic intensity i of a communication link l increases. As shown in Figure 2, the number of packets that have to be queued increases non-linearly.

Figure 3 motivates delay-awareness for the virtual network embedding problem: In this scenario, two virtual networks are being embedded. It is assumed that, in this scenario, only substrate nodes A and B offer sufficient resources for hosting the virtual nodes. If the virtual link of VNR 1 is assigned to the substrate link between A and B, utilization increases and, as such, also does queuing delay. Still, this link offers sufficient bandwidth resources to the virtual link of VNR2. In fact, current virtual network embedding approaches that do not consider delay as part of their optimization strategy, tend to embed also the second virtual link to this substrate link. However, as utilization is, in this case, 100%, this would significantly increase delay if traffic intensity in virtual networks increases.

Traffic intensity is usually defined as follows:

$$i = \frac{La}{R}$$

with packet length L, arrival rate a and bandwidth R. Consistent with several other work presented in this area of research, it is assumed that a fixed amount of bandwidth resources is assigned to virtual links. Thus, traffic intensity is calculated as

$$i(l) = \frac{R_{\text{occupied}}(l)}{R_{\text{total}}(l)}$$

with R_{total} denoting bandwidth resources of a link l and R_{occupied} bandwidth resources that are allocated to virtual links.

The average number of packets waiting for transmission is calculated based on queuing theory; a substrate link is modeled

 ISBN: 978-1-61208-377-3

as a M/M/1 queuing system; then, the average number of packets waiting in the link's queue is calculated as follows (graph shown in Figure 2) [10]:

$$p(l) = \frac{i(l)}{1 - i(l)}$$

Considering that queue size qs in real systems is limited, queuing delay is then defined as

$$d_{\text{queue}}(l) = \frac{\min\big(p(l), qs\big) \cdot L}{R_{\text{total}}(l)}$$

Thus, total delay is defined as

$$d_{\text{total}}(l) = d_{\text{proc}} + d_{\text{prop}} + \frac{L}{R_{\text{total}}(l)} + d_{\text{queue}}(l)$$

Taking utilization into account, this model describes the delay-behavior of real-world substrate networks more accurately than linear models.

B. Optimization Objectives

Based on the model presented in the previous section, this section discusses optimization objectives for future delay-aware virtual network embedding approaches. Several optimization objectives are applicable in this context:

Minimizing delay is the most obvious objective. This can be done in various ways, depending on the optimization objective of the infrastructure provider: First, one option is to aim for minimum average delay. More precisely, the embedding algorithm aims to minimize average delay of all substrate paths assigned to virtual links. In this case, the infrastructure provider guarantees that on average, delay is below a certain limit. However, in worst case, some communication links do have worse delay properties. Thus, another option is to minimize maximum delay (instead of the average delay). Now, the infrastructure provider is able to guarantee that none of the embedded virtual links suffer from worse communication delay. In general, traffic intensity should not exceed 80-90%, as depicted in Fig. 2.

As a more straight-forward alternative, instead of *optimizing* the embedding towards delay-effectiveness, the embedding can be performed by just *considering* delay constraints: I.e., instead of minimizing delay, the embedding algorithm just assures that communication delay never exceeds a pre-defined limit. For example, virtual links are never assigned to substrate paths if traffic intensity on these links would get too high. This simplified concept can be extended with respect to delay constraints of individual virtual links: In this case, virtual network operators are able to specify delay constraints for individual links (instead of for the whole virtual network). The infrastructure provider assigns these networks in a way that none of these constraints are violated. The traditional embedding problem can be easily extended with respect to both alternatives; in both cases, embedding algorithms just have to validate that none of the constraints has been violated before assigning substrate nodes and links. This works equivalently

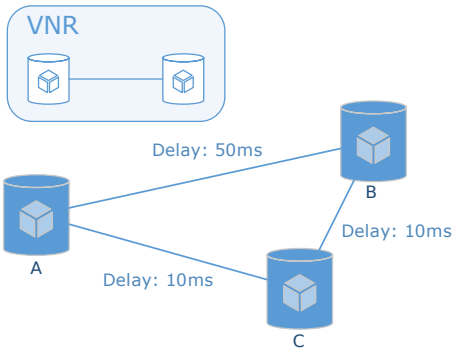

Fig. 4. Delay-Awareness vs. Cost-Efficiency

to the validation of whether substrate resources fulfill CPU and bandwidth demands of the virtual nodes and links. Here, the actual optimization objective of the embedding approaches does not have to be altered.

As mentioned before, the optimization objective of most embedding algorithms is cost-efficiency. Many embedding approaches aim to keep embedding cost low, in order to increase the amount of available network resources, leaving space for future virtual network requests. A delay-aware embedding, however, often comes with higher embedding cost. The tradeoff between delay-awareness and cost-efficiency is depicted in Figure 3. A cost-optimal embedding algorithm assigns the virtual link to the substrate link directly connecting A and B. In this case, the virtual network operator suffers from high delay. A delay-optimal algorithm, however, choses the indirect path A↔B↔C. In this case, the infrastructure provider suffers from high embedding cost. A suitable solution to this dilemma would be to combine delay-awareness and embedding cost by balancing both objectives accordingly.

C. Evaluation Metrics

Several evaluation metrics have been discussed in the context of the virtual network embedding problem so far. A survey on embedding approaches has been presented by Fischer et al. [3], also including an extensive discussion on evaluation metrics. Furthermore, an open source simulation framework implementing many of these metrics has been published [11]. This section presents several new metrics with reference to delay-awareness. Of course, novel embedding algorithms should always be thoroughly analyzed also with regard to other metrics that are not directly related to communication delay, most notably one, the embedding cost metric. Since cost is one of the most notable and well-known metrics in the context of virtual network embedding, it is shortly discussed here.

Embedding Cost: Embedding cost is one of the most common evaluation metrics in the context of virtual network embedding. Cost is defined as follows:

$$\text{Cost}(\text{VNR}^i) = \sum_{n^i \in N^i} \text{res}_{\text{cpu}}(n^i) + \sum_{l^i \in L^i} \sum_{l \in L} \text{res}_{\text{bw}}(l^i, l)$$

Performance of embedding algorithms is usually evaluated through extensive simulations in randomly generated network topologies. Embedding cost significantly depends on the kind of virtual networks that are being embedded: Randomly generated network requests demanding few network resources tend to be much easier to embed than those demanding many. Therefore, the revenue metric is used in order to quantify virtual network requests. Similar to cost, revenue is computed as follows for a virtual network request VNR^i:

$$\mathrm{Revenue}(\mathrm{VNR}^i) = \sum_{n^i \in N^i} \mathrm{dem}_{\mathrm{cpu}}(n^i) + \sum_{l^i \in l^i} \mathrm{dem}_{\mathrm{bw}}(l^i)$$

Thus, to put cost in relation to network requests, the revenue/-cost metric is introduced:

$$\text{Revenue-Cost}(\mathrm{VNR}^i) = \frac{\mathrm{Revenue}(\mathrm{VNR}^i)}{\mathrm{Cost}(\mathrm{VNR}^i)}$$

Path Length: The path length metric reflects how many substrate links were (on average/maximum) assigned to the virtual links. Despite of the fact that communication delay of a substrate path is the sum of the delay on each substrate link segment of that path, long path lengths do not necessarily reflect large delays. As an example, in the scenario depicted in Figure 4, a delay-aware embedding algorithm would chose a longer, but less delay-intense path.

Link Delay: Link delay is defined as the average/maximum delay of all substrate links. This metric is of interest to the infrastructure provider, as it reflect the average/maximum utilization of the substrate network. High link delay indicates that either the infrastructure is not optimally used (e.g., as a result of several non-optimal embeddings of virtual networks) or additional hardware components should be integrated into the substrate network in order to improve network performance and to keep up with increasing virtual network demands. Link delay is therefore also a good indicator for estimating whether sufficient substrate resources are available in order to embed further virtual network requests.

Traffic Intensity: Traffic intensity reflects how many packets (on average/maximum) are being queued by the substrate nodes. Similarly to the link delay metric, traffic intensity indicates which parts of the substrate network suffer from high load and, thus, need to be reconfigured.

Path Delay: Path delay is defined as the sum of the delay of all substrate links that are part of a communication path. I.e., average/maximum path delay is the average/maximum delay of all paths that were assigned to virtual links. This metric is a key indicator to the virtual network operator, as it reflects how well the virtual network has been embedded into the substrate network and how the network performs with respect to communication delay.

Concluding, embedding cost, path length, link delay, and traffic intensity are metrics that are related to the performance of the substrate network. Path delay is a key metric indicating how well a virtual network performs after it has been embedded into the substrate infrastructure.

IV. Conclusion and Future Work

As discussed in this paper, link utilization significantly influences queueing delay of routers; Delay-aware embedding approaches should consider that delay increases non-linearly; the embedding has to be performed in a way such that communication delay is kept within reasonable bounds. In general, infrastructure providers should avoid embedding virtual links to paths such that traffic intensity exceeds 80%. For delay-sensitive applications, new virtual network embedding approaches are needed with delay-aware optimization objectives. As discussed in this paper, there is a tradeoff between delay-awareness and cost-efficiency. Therefore, these algorithms should also consider embedding cost as part of their optimization strategy.

We are currently in the process of implementing a delay-aware embedding algorithm based on the model presented here. Furthermore, evaluation metrics discussed in this paper are in the process of being integrated into the Alevin simulation framework [11], [12] and will, as such, be published as open source.

References

[1] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, "German-lab experimental facility," *Future Internet Symposium (FIS) 2010*, 9 2010.

[2] J. Carapinha and J. Jiménez, "Network virtualization: a view from the bottom," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, VISA '09, (New York, NY, USA), pp. 73–80, ACM, 2009.

[3] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 1888–1906, 2013.

[4] M. T. Beck, J. F. Botero, A. Fischer, H. De Meer, and X. Hesselbach, "A distributed, parallel, and generic virtual network embedding framework," in *IEEE Int'l Conf. on Communications (ICC 2013)*, IEEE, 2013.

[5] A. Fischer, M. T. Beck, and H. De Meer, "An approach to energy-efficient virtual network embeddings," in *Proc. of the 5th Int'l Workshop on Management of the Future Internet (ManFI 2013)*, IFIP, IEEE, 2013.

[6] M. T. Beck, A. Fischer, and H. De Meer, "Distributed virtual network embedding," in *Proc. of the 7th GI/ITG KuVS Workshop on Future Internet*, University of Kaiserslautern, 2012.

[7] K. Ivaturi and T. Wolf, "Mapping of delay-sensitive virtual networks," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pp. 341–347, IEEE, 2014.

[8] L. Shengquan, W. Chunming, Z. Min, and J. Ming, "An efficient virtual network embedding algorithm with delay constraints," in *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*, pp. 1–6, June 2013.

[9] J. Kurose and K. Ross, "Computer networks: A top down approach featuring the internet," *Peorsoim Addison Wesley*, 2006.

[10] T. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Telecommunication networks and computer systems, Springer New York, 2000.

[11] M. T. Beck, A. Fischer, F. Kokot, C. Linnhoff-Popien, and H. De Meer, "A simulation framework for virtual network embedding algorithms," in *Proc. of the 16th International Telecommunications Network Strategy and Planning Symposium*, Networks, 2014.

[12] VNREAL, "ALEVIN2 – ALgorithms for Embedding VIrtual Networks." `http://alevin.sf.net`, May 2014.

 ISBN: 978-1-61208-377-3