The Price of Specificity in the Age of Network Virtualization

Arne Ludwig, Stefan Schmid, Anja Feldmann
Telekom Innovation Laboratories & TU Berlin, Germany
`{arne,stefan,anja}@net.t-labs.tu-berlin.de`

***Abstract*—The virtualization trend in today's Internet decouples services from the constraints of the underlying physical infrastructure. This decoupling has the potential to facilitate more flexible and efficient resource allocations: the service can be realized at *any* place in the substrate network which fulfills the service specification requirements.**

This paper studies such flexibilities in the context of virtual network (VNet) embeddings. The network virtualization paradigm envisions an Internet where users can request arbitrary VNets from a substrate provider (e.g., an ISP). A VNet describes a set of virtual nodes which are connected by virtual links; both nodes and links provide certain resource guarantees. While some parts of the VNet may be fully specified (e.g., the node and link locations or technologies), other parts may be flexible or left open entirely. For example, it may be irrelevant for some users on which vendor hardware the VNet is realized, or a user only requests that the VNet runs in some *European* cloud provider.

We study how flexible specifications can be exploited to improve the embedding of virtual networks. We introduce the notion of the *Price of Specificity* which captures the resource cost of the embedding under a given specification. We analyze on which parameters the Price of Specificity depends, and evaluate its magnitude in different scenarios.

I. INTRODUCTION

Virtualization is a central design principle in the Internet today and a motor for innovation. While end-system virtualization revamped the server business (e.g., via the Xen or VMware technology), also link virtualization is gaining momentum: router vendors such as Cisco and Juniper offer router virtualization, and recently Google announced to have adopted a software-defined networking (SDN) approach to manage its massive internal network using OpenFlow.

The *network virtualization paradigm* [5] goes one step further and heralds a networking era where entire virtual networks (VNets) can be requested on demand and for a desired duration only (e.g., for a telephone or video conference, or for a scientific experiment). These VNets are *embedded* on a shared physical (or again virtual!) network, a.k.a. the *substrate network*, which facilitates cheap realizations. This shared substrate is not necessarily a data center, but it may also be a wide area or backbone network of one or multiple ISPs. In spite of the shared infrastructure, with appropriate isolation (resource multiplexing / traffic shaping) policies in place, a VNet can appear as a "dedicated network" to the user.

One of the main features of virtualization technology is the decoupling of services from the physical constraints of the underlying infrastructure network where the service is embedded. As virtual networks are often not fully specified (e.g., VNets may require hardware from certain vendors or certain network interfaces, but may be flexible with respect to the embedding location), this decoupling can be exploited to optimize the embedding within the constraints of the VNet specifications: the more flexible the specification, the more resource efficient the VNet's realization.

A central theme in network virtualization is hence the question of how to algorithmically *exploit* the placement flexibility. Surprisingly, however, not much is known about the important relationship of specification flexibility and resource costs. This paper attends to this tradeoff which we believe has an interesting economical dimension.

VNet Embeddings. Let us briefly review the VNet embedding problem and introduce some formalism before we discuss our contributions. We represent the substrate network as a graph $G_S = (V_S, E_S)$ where V_S represents the substrate nodes and E_S represents the substrate links. Also a VNet request comes in the form of a graph, represented as $G_V = (V_V, E_V)$ (V_V are the virtual nodes, E_V are the virtual links). This VNet needs to be *embedded* on G_S: each virtual node of G_V is mapped to a substrate node, and each virtual link is mapped to a *path* (or a *set of paths*). Figure 1 illustrates an example.

For simplicity and to focus on the specificity, throughout this paper, we will study undirected and unweighted VNets only, i.e., we assume that each node $v \in V_V$ and each link $e \in E_V$ has a unit capacity. Our approach can easily be extended to weighted VNets.

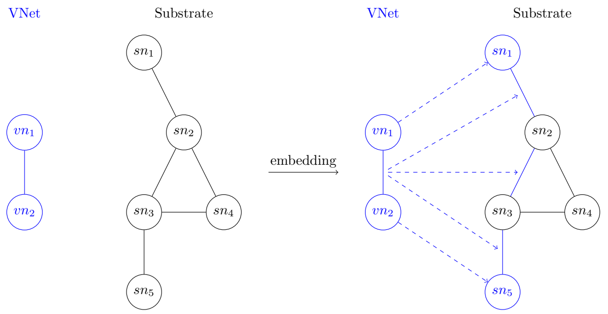

Fig. 1. Visualization of a VNet embedding: the 2-node VNet on the left is mapped to the 5-node substrate network. Each virtual node of the VNet maps to a substrate node, and for the realization of the virtual link resources are allocated along a path.

We will consider the following embedding cost model.

Definition I.1 (Embedding Costs). *Let* $\Pi(e) = \{\pi_1, \pi_2, \ldots\}$ *for some* $e \in E_V$ *denote the set of substrate paths over which* e *is realized (i.e., embedded). Let* $\omega(\pi)$ *for some* $\pi \in \Pi(e)$ *denote the fraction of flow over path* π*, and let* $\lambda(e)$ *denote the length of* e *in terms of number of hops. The cost of embedding a VNet* $G_V = (V_V, E_V)$ *on a substrate* G_S *is defined as* $Cost = \sum_{e \in E_V} \sum_{\pi \in \Pi(e)} \omega(\pi) \cdot \lambda(e)$.

In other words, the allocation cost is simply the weighted distance of the different paths used by the virtual edges.

Our Contribution. This paper initiates the study of the tradeoff of specification flexibility and allocation cost by introducing the notion of the *Price of Specificity (PoS)*. We identify different types of specifications and analyze their influence on the VNet allocation cost.

We believe that our evaluation not only sheds light onto the resource costs of a VNet in different scenarios (and hence in some sense, the real "value" of a resource) but can also provide insights on how to structure a substrate network in order to increase the number of embeddable networks (and reduce the Price of Specificity) at minimal cost.

In this sense, we regard our work as a first step towards a better understanding of the economical dimension of the VNet embedding problem, and we provide a short discussion of its limitations and further directions.

II. THE PRICE OF SPECIFICITY

To study the Price of Specificity, we consider the following model. We assume that each substrate node $v_S \in V_S$ of $G_S = (V_S, E_S)$ can be described by a set of k *properties* $P = \{p_1, \ldots, p_k\}$, e.g., the geographical *location* (e.g., `data center in Berlin, Germany`), the hardware *architecture* (e.g., `64-bit SPARC`), the *operating system* (e.g., `Mac OS X`), the virtualization *technology* (e.g., `Xen`), and so on. The specific property $p \in P$ of v_S can be realized as a specific *base type* $t_{v_S}(p)$. For example, the set $T(p)$ of base types for an operating system property $p \in P$ may be $T(p) = \{$`Mac OS X`, `RedHat 7.3`, `Windows XP`$\}$. (Note that if not every substrate node features each property, a dummy type *not available* can be used.)

Similarly, the VNet $G_V = (V_V, E_V)$ comes with a certain *specification* $\sigma(G_V)$ of allowed types. While the substrate nodes V_S naturally are of specific base types, VNet specifications can be more vague. For example, the types $T(p)$ can often be described *hierarchically*: the location `Berlin` can more generally be described by `Germany`, `Europe`, or `?` (*don't care*); or instead of specifying the operating system `Mac OS X`, a VNet may simply require a `Mac`.

Concretely, we assume that each virtual node v_V comes with a specification $\sigma(v_V) \subseteq T(p_1) \times \ldots \times T(p_k)$ of allowed type combinations for the different properties. The substrate node v_S to which v_V is embedded must fulfill at least one such type combination.

Definition II.1 (Valid Embedding). *Let* $t(v_S) = \times_{p \in P} t_{v_S}(p)$ *denote the vector of types of substrate node* v_S*. A VNet* $G = (V_V, E_V)$ *embedding is valid iff for each virtual node* $v_V \in V_V$*, it holds that* v_V *is mapped to a node* v_S *with* $t(v_S) \in \sigma(v_V)$*. In addition, node and link capacity constraints are respected.*

Of course, a user must not specify $t(v_S)$ explicitly by enumerating all allowed combinations: this set only serves for formal presentation. Rather, a user can specify the types of VNet nodes with an arbitrary resource description language, and use *white lists* (e.g., only `Mac`) or *black lists* (not on `Sparc`), or more complex logical formulas.

The question studied in this paper revolves around the tradeoff of the VNet specificity and the embedding cost.

Definition II.2 (Price of Specificity (PoS) ρ). *Given a VNet* G_V*, let* $Cost_0$ *denote the embedding cost (cf Definition I.1) of* G_V *in the absence of any specification constraints, and let* $Cost_\sigma$ *denote the embedding cost under a given specification* $\sigma(G_V)$*. Then, the* Price of Specificity $\rho(G_V)$ *(or just* ρ*) is defined as* $\rho = Cost_\sigma / Cost_0$.

Note that the Price of Specificity ρ depends on the specific embedding algorithm ALG. In this paper, we do not propose any specific embedding algorithm, but just use the placeholder ALG to denote an arbitrary state-of-the-art VNet embedding algorithm. (In the related work section, Section V, we will review some candidates from the literature.) However, in the simulations, we will use an optimal algorithm ALG that minimizes resources.

Although our definition of the Price of Specificity is generic and does not depend on a particular definition of specificity, for our evaluation, we will use the following metric.

Definition II.3 (Specificity σ). *The* specificity $\sigma(v_V)$ *of a virtual node* v_V *captures how many alternative type configurations are still allowed by a specification compared to a scenario where all configurations are allowed. Formally, we define* $\sigma(v_V)$ *as the percentage of lost alternatives:* $\sigma(v_V) = 1 - (|t(v_S)| - 1)/(|T(p_1) \times \ldots \times T(p_k)| - 1)$*. The* specificity $\sigma(G_V)$ *of a VNet* $G_V = (V_V, E_V)$ *is defined as the average specificity of its nodes* $v_V \in V_V$*:* $\sigma(G_V) = \sum_{v_V \in V_V} \sigma(v_V)/|V_V|$.

Note that $\sigma(G_V) \in [0, 1]$, where $\sigma(G_V) = 1$ and $\sigma(G_V) = 0$ denote the maximal and the minimal specificity, respectively. (We will focus on scenarios where $|T(p_1) \times \ldots \times T(p_k)| > 1$.)

III. EVALUATION

This section studies the Price of Specificity (PoS) in different scenarios. In order to avoid artifacts resulting from approximate or heuristic embeddings, we consider an optimal embedding algorithm ALG. ALG is based on *Mixed Integer Programming* [7] and is also used in our network virtualization prototype architecture [8]. It also allows us to study the effect of VNet migrations.

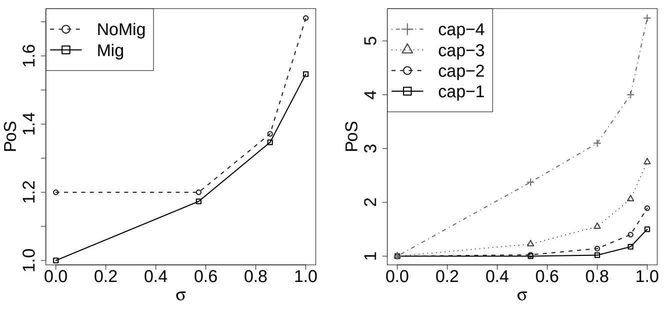

Fig. 2. *Left:* Impact of migration on the PoS. There are five 4-star VNets arriving over time on a 40-nodes Igen substrate with eight different substrate node types. *Right:* Impact of substrate node capacities on the PoS. The figure shows the result from embedding a 5-star VNet while changing the node capacity of the underlying substrate from one VNet node per substrate node up to four nodes per substrate node.

Setup. We consider two different properties $P = \{p_1, p_2\}$ with four different types each ($T(p_1) = \{t_1^1, t_2^1, t_3^1, t_4^1\}$, $T(p_2) = \{t_1^2, t_2^2, t_3^2, t_4^2\}$). Our substrate network is generated using the *Igen topology generator* [6]. The (standardly 100) substrate node types are chosen independently at random from the (standardly 16) base types, such that each type occurs equally often (up to rounding), and the virtual node types are chosen independently uniformly at random according to the specificity.

Furthermore, we assume that the nodes in the substrate all have a capacity of one unit, and that the links have an infinite capacity. The virtual nodes and links of the VNet have a demand of one unit. Finally, we allow the embedding algorithm to split a virtual link into multiple paths.

As for the VNets we will focus on star topologies. In the following we denote a star with one center node and $x - 1$ leaves an *x-star*. In most cases we study 4- or 5-stars. This is motivated by the tasks we observe at the *Google* compute cell with 11k machines (see `http://code.google.com/p/googleclusterdata/`, file: `TraceVersion2`). The frequency of VNets with exactly three nodes is about 73%.

General observations. In our experiments we find most of our expectations confirmed. If we increase the substrate size the PoS is decreasing. The same applies to a decreased load in the substrate. Surprisingly the PoS, while it is still higher, is not increasing significantly when changing the VNet from a 4-star to a 6-star. Our results also confirm the intuition that the relationship between the distribution of the requested and the supplied resource types is important: While skewed distributions can yield better allocations, they entail the risks of a high PoS if the demand does not perfectly match.

Impact of Migration. We study a scenario where five 4-star VNets arrive over time on a 40 node Igen substrate with eight different substrate node types. We only study runs where all five VNets have been embedded, resulting in a load of 50% on the substrate. This avoids heavily loaded substrate scenarios as well as scenarios of abundant capacity. Figure 2 (*left*) shows the aggregated PoS over all five VNets. We show averaged values as the embedding costs for an already embedded VNet can change over time in the migration scenario. Interestingly, migration is already effective even without specificity on the VNets (compare PoS Mig:1 - NoMig:1.2). This is due to embeddings which are initially optimal regarding resource costs but use resources that might be more effective in later embeddings, i.e., nodes with a higher degree. While the impact of migration is rather low for the following specificities, it is again recognizable for fully specified VNets.

Generally migration lowers the resource costs in our scenarios, but there is no specific relationship between its effects and the specificity of the VNets.

Impact of Capacity. A substrate provider can optimize VNet embeddings according to its needs, and e.g., minimize link resource costs (see ALG). The capacities in the infrastructure will not fit exactly the specifications of the VNets in general. Therefore the embedding of several VNet nodes on one substrate node promises lower link resource costs while the restriction on link capacities may yield higher costs.

Figure 2 (*right*) shows a scenario where the node capacities are increased up to four units. The curves show similar trends and there is a higher PoS noticeable with increasing node capacity. This is a consequence of using the substrate nodes at full capacity without VNet node specificities whereas a higher specificity prevents this due to the different VNet node types. A PoS is also observed on specificities where there was no additional costs before. At a specificity of $\sigma \approx 0.533$, the PoS for the scenario with a node capacity of four is already higher than the PoS of a fully specified VNet in a scenario with node capacity one.

An Out-Sourcing Scenario. While our previous simulations may also describe geographical types (as a property), we now change the experiments in a way that some VNet nodes assume a fixed position in the substrate. This use-case can for example represent a company which owns certain servers and runs its own network, but out-sources computation or storage to the cloud. While the locations of the infrastructure in the corporate network is given, the out-sourced resources may be specified at different granularity.

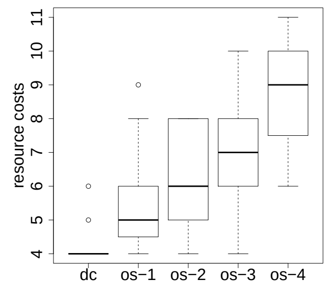

Fig. 3. Boxplot for the data center scenario (dc - zero nodes fixed) and four different out-sourcing scenarios (os1 - one fixed node, etc.), for $\sigma \approx 0.53$.

The impact of the fixed node locations can be understood as a function of the link resource costs of the out-sourcing scenarios. Figure 3 shows a boxplot comparing these costs with each other for a specificity of $\sigma \approx 0.53$. In the data center scenario the VNet is almost always embedded with the smallest possible amount of link resources while an increasing number of fixed nodes also increases the variance as well as the used link resources. Nonetheless, these resource costs remain below those of the out-sourcing scenarios.

IV. ON THE USE OF MIGRATION

Before concluding our contribution, let us briefly discuss an interesting result from our simulations in more detail: How does migration influence the Price of Specificity?

In order to study the impact of migration on the PoS we extend the Definition II.2 towards several embedded VNets.

Definition IV.1 (Price of Specificity ρ for multiple VNets)**.** *Given a sequence of VNets* $\mathcal{G}_k = (G_V^1, ..., G_V^k)$*, the PoS* $\rho(\mathcal{G}_k)$ *is defined as the average PoS over all VNets* $\rho(\mathcal{G}_k) = \sum_{i=1}^{k} \rho(G_V^i)/k$*.*

Given the PoS definition for a sequence of VNets, we will make the following assumptions: (1) We focus on embedding algorithms ALG which greedily accept all incoming VNets if possible, while trying to minimize the corresponding embedding costs. (2) Migration itself is only causing no/negligible costs regarding the PoS. (3) The substrate links have unbounded link capacities.

Theorem IV.2. *In scenarios satisfying Assumptions (1)-(3), migration can only decrease the overall PoS of the embedded VNets, meaning* $\rho_1(\mathcal{G}_k) \leq \rho_0(\mathcal{G}_k)$ *for any sequence of VNets* $\mathcal{G}_k$*, with* ρ_1 *representing the PoS for the migration scenario and* ρ_0 *representing the one without.*

However, note that if link capacities are limited, Theorem IV.2 no longer holds. We will refer to this phenomenon as the *Migration Paradox*.

Theorem IV.3. *Generally, there are scenarios where migration can* increase *the PoS.*

Consider a scenario where a certain resource is scarce and due to link capacity constraints only a migration could enable an embedding. Then, the embedding of the VNet, as well as the migration are probably resource costly and additionally might block the embedding of further VNets with lower resource costs. This results in a higher PoS for the sequence of VNets.

V. RELATED WORK

For a good survey of the network virtualization paradigm and existing embedding algorithms, the reader is referred to [5] and [3]. Our contribution is orthogonal to the embedding and resource allocation / optimization line of research. In fact, the Price of Specificity could be studied for each embedding algorithm reviewed above. In order to focus on the main properties of the Price of Specificity, we use a an optimal embedding approach for our evaluation, and ask the question how the VNet specification effects the cost.

There are many networking domains where economical aspects play a central role, for instance in the Internet backbone, in the cloud, in wireless spectrum allocation, or in the grid where storage and computational resources come at a certain price. Especially federation (e.g., collaboration of different ISPs or cloud providers) and fairness issues (over multiple resource types) have gained much attention recently. A review of these research fields is beyond the scope of this paper, and for a good introduction to some classic tradeoffs, we refer the reader to the V-Mart [9] paper (and the references therein) which attends to the inter-domain embedding problem and uses an auction-based model, and in the context of computing resources, to the GridEcon a market place proposed in [1].

In the context of the relatively new concept of network virtualization itself, not much work on economical aspects exists yet. PolyViNE [4] is a decentralized, policy-based inter-domain embedding protocol ensuring competitive prices for service providers. In more general context, Antoniadis et al. [2] employ coalitional game theory to study how participants should share the value of federation in virtualized infrastructures (in the context of ISP interconnections, peer-to-peer systems, the Grid, or cloud computing).

However, we are not aware of any studies on the effect of specificity on the VNet embedding cost.

VI. CONCLUSION

We believe that today we have a very good understanding of the technological solutions that are required to implement virtual networks (see e.g., the recent advances in OpenFlow and SDN technology), and also much progress has been made over the last years on the algorithmic and optimization side. However, little is known on the economical consequences.

We believe that the concept of Price of Specificity can help providers to reason about VNet pricing and embedding aspects, especially in competitive markets where infrastructure providers operate within small budget margins.

REFERENCES

[1] J. Altmann, C. Courcoubetis, G. D. Stamoulis, M. Dramitinos, T. Rayna, M. Risch, and C. Bannink. GridEcon: A market place for computing resources. In *Proc. GECON*, 2008.

[2] P. Antoniadis, S. Fdida, T. Friedman, and V. Misra. Federation of virtualized infrastructures: sharing the value of diversity. In *Proc. CoNEXT*, 2010.

[3] A. Belbekkouche, M. Hasan, and A. Karmouch. Resource discovery and allocation in network virtualization. *IEEE Communications Surveys Tutorials*, 2012.

[4] M. Chowdhury, F. Samuel, and R. Boutaba. PolyViNE: Policy-based virtual network embedding across multiple domains. In *Proc. SIGCOMM VISA*, 2010.

[5] M. K. Chowdhury and R. Boutaba. A survey of network virtualization. *Elsevier Computer Networks*, 2010.

[6] B. Quoitin, V. V. den Schrieck, P. Franois, and O. Bonaventure. Igen: Generation of router-level internet topologies through network design heuristics. In *Proc. ITC*, 2009.

[7] G. Schaffrath, S. Schmid, and A. Feldmann. Optimizing long-lived cloudnets with migrations. In *Proc. UCC*, 2012.

[8] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A. Wundsam, M. Kind, O. Maennel, and L. Mathy. Network virtualization architecture: Proposal and initial prototype. In *Proc. SIGCOMM VISA*, 2009.

[9] F. Zaheer, J. Xiao, and R. Boutaba. Multi-provider service negotiation and contracting in network virtualization. In *Proc. NOMS*, 2010.