Virtual Network Embedding: A Survey

Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann de Meer, and Xavier Hesselbach

***Abstract*—Network virtualization is recognized as an enabling technology for the future Internet. It aims to overcome the resistance of the current Internet to architectural change. Application of this technology relies on algorithms that can instantiate virtualized networks on a substrate infrastructure, optimizing the layout for service-relevant metrics. This class of algorithms is commonly known as "Virtual Network Embedding (VNE)" algorithms. This paper presents a survey of current research in the VNE area. Based upon a novel classification scheme for VNE algorithms a taxonomy of current approaches to the VNE problem is provided and opportunities for further research are discussed.**

***Index Terms*—Virtual networks, network virtualization, virtual network embedding, embedding algorithms, network mapping**

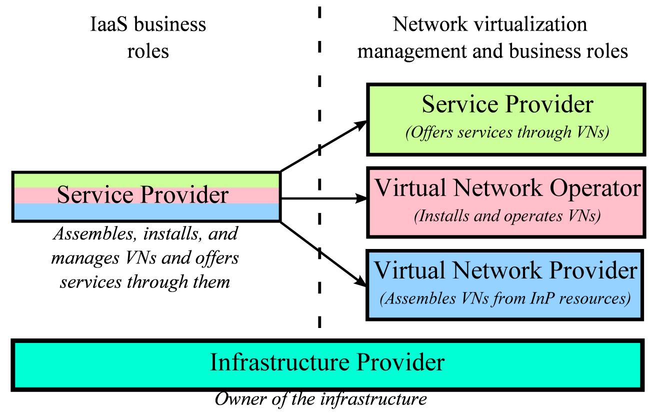

Fig. 1. Future Internet business model

I. Introduction

NETWORK virtualization [1], [2], [3], [4] is one of the most promising technologies for the future Internet. Introduced as a means to evaluate new protocols and services [5], it has already been actively used in research testbeds like G-Lab [6] or 4WARD [7], applied in distributed cloud computing environments [8] and is, by now, seen as a tool to overcome the resistance of the current Internet to fundamental changes. As such, network virtualization can be thought of as an inherent component of the future Internet architecture [9]. Indeed, even today network virtualization approaches are applied in the telecommunication market. An example for this is OpenFlow [10], which experiences strong support by the industry within the Open Networking Foundation [11].

In network virtualization, the primary entity is the *Virtual Network* (VN). A VN is a combination of active and passive network elements (network nodes and network links) on top of a *Substrate Network* (SN). Virtual nodes are interconnected through virtual links, forming a virtual topology. By virtualizing both node and link resources of a SN, multiple virtual network topologies with widely varying characteristics can be created and co-hosted on the same physical hardware. Moreover, the abstraction introduced by the resource virtualization mechanisms allows network operators to manage and modify networks in a highly flexible and dynamic way.

Future Internet architectures will be based on the *Infrastructure as a Service* (IaaS) [12] business model that decouples the role of current *Internet Service Providers* (ISPs) into two new roles: The *Infrastructure Provider* (InP) who deploys and maintains the network equipment and the *Service Provider* (SP), in charge of deploying network protocols and offer end-to-end services. The introduction of network virtualization separates the management and business roles of the SP [13] by identifying three main players (see Fig. 1): The *Virtual Network Provider (VNP)* which assembles virtual resources from one or more InPs, the *Virtual Network Operator (VNO)* which installs, manages and operates the VN according to the needs of the SP, and the SP which is free of management and concentrates on business by using the VNs to offer customized services.

The problem of embedding virtual networks in a substrate network is the main resource allocation challenge in network virtualization [14] and is usually referred to as the *Virtual Network Embedding* (VNE) problem. Through dynamic mapping of virtual resources onto physical hardware, the benefit gained from existing hardware can be maximized. Optimal dynamic resource allocation, leading to the self-configuration and organization of future networks, will be necessary to provide customized end-to-end guaranteed services to end users. This optimality can be computed with regard to different objectives, ranging from QoS, economical profit, or survivability over energy-efficiency to security of the networks. Fig. 2 illustrates how network virtualization makes use of embedding algorithms in order to allocate virtual resources on a physical infrastructure in an optimal way. The VNO uses embedding algorithms to decide which virtual resources to request from the VNP, who, in turn, instantiates them by using the InPs substrate resources.

Virtual Network Embedding deals with the allocation of virtual resources both in nodes and links. Therefore, it can be divided in two sub-problems: *Virtual Node Mapping* (VNoM) where virtual nodes have to be allocated in physical nodes and *Virtual Link Mapping* (VLiM) where virtual links connecting these virtual nodes have to be mapped to paths connecting the corresponding nodes in the substrate network.

Manuscript received July 27, 2012; revised November 23, 2012.

A. Fischer, M.T. Beck and H. De Meer are with the University of Passau, Innstr. 43, 94032 Passau, Germany (e-mail: andreas.fischer,demeer}@uni-passau.de). M.T. Beck is also with the Stadtwerke Passau, Regensburger Str. 29, 94036 Passau, Germany (e-mail: michael.beck@stadtwerke-passau.de).

J.F. Botero and X. Hesselbach are with the Universitat Politècnica de Catalunya, Jordi Girona Street 1-3, 08034 Barcelona, Spain (e-mail: {jf-botero,xavierh}@entel.upc.edu).

Digital Object Identifier 10.1109/SURV.2013.013013.00155

1553-877X/13/$31.00 © 2013 IEEE

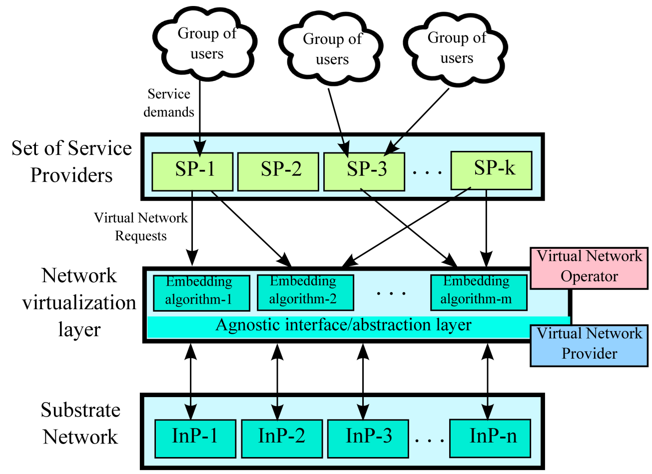

Fig. 2. Resource allocation in future Internet

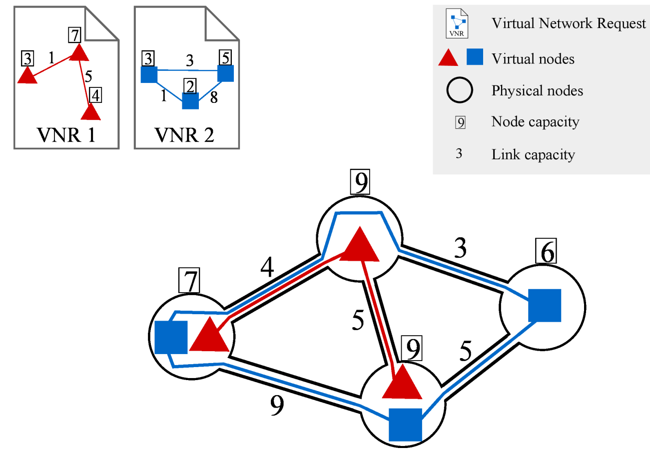

Fig. 3. Two virtual networks mapped onto one substrate network

A recent survey by Belbekkouche et al. [15] provides a description of the main approaches proposed for resource discovery and allocation (including VNE) in network virtualization. The survey presented here goes beyond just describing the main VNE approaches: A novel VNE classification scheme is presented. The VNE problem is considered in all its variants and current proposals coming from the research community are classified. In particular: a formal and generic mathematical *formulation* of the VNE problem, the different *parameters* that can be considered in the embedding for substrate as well as for virtual networks, the main *embedding objectives* which relate to VNE, the possible alternatives to *decompose* the VNE problem and to solve the *coordination* between VNoM and VLiM, the set of possible *optimization strategies* to solve VNE, the different *metrics* used to evaluate the performance of the solutions, a number of VNE *simulators*, a detailed *classification* of the existing approaches based on the employed strategy, and the emerging *research directions* currently considered by the research community are presented.

The remainder of this paper is organized as follows: Section II formulates the VNE problem and presents its different categories. Different variants, strategies, parameters and metrics used by VNE algorithms as well as the main software tools are presented in Section III. A classification of the existing VNE approaches is presented in Section IV. Section V highlights emerging research directions in the VNE field. Finally, Section VI concludes this survey.

II. Virtual Network Embedding

Several papers have provided specific formulations for VNE. In this survey, a general VNE formulation and a novel categorization is introduced, that allows to easily identify each variant of the problem.

A. Problem formulation

The application of virtualization mechanisms to network resources leads to the question how the virtualized resources should be realized by the substrate resources. It is important to note that substrate resources can be virtual themselves. This is commonly referred to as nested virtualization. In that case, only the lowest layer has to consist of physical resources.

The hardware abstraction provided by the virtualization solution provides a common denominator, allowing any substrate resource to host virtual resources of the same type. Typically, a substrate resource is partitioned to host several virtual resources. For example, a virtual node can, in principle, be hosted by any available substrate node. Moreover, a single substrate node can host several virtual nodes. Thus, the mapping of virtual nodes to substrate nodes describes a $n:1$ relationship (a strict partition of substrate resources).

In some cases, substrate resources can also be combined to create new virtual resources. This is the case for a virtual link which spans several links (i.e. a path) in the substrate network. In this case, a virtual link between two virtual nodes v and w is mapped to a path in the substrate network that connects the substrate hosts of v and w. Each substrate link may then be part of several virtual links. As such, the mapping of virtual links to substrate paths describes a $n:m$ relationship (both, a partition and a combination of substrate resources).

Fig. 3 depicts a scenario, where two virtual networks with three nodes each are hosted on one substrate network with four nodes. It can be seen that substrate nodes can host several virtual nodes (up to two in this example). Likewise, substrate links can host more than one virtual link. Moreover, one of the virtual links spans two substrate links, thus representing a virtual resource combined from several substrate resources.

Typically, there are some restrictions to be considered during the mapping. Most obviously, the candidate substrate resources for a mapping have to be able to support the performance requirements of the virtual resources. For example, a 1000 MBit/s virtual link can not be mapped to a path containing a 100 MBit/s substrate link. Likewise, the CPU power requested by a virtual node has to be less than (or equal to) the CPU power actually provided by a substrate node. If redundancy is required, even more substrate resources may have to be reserved. Nevertheless, substrate resources should be spent economically. Therefore, the mapping has to be optimized. This problem of mapping virtual resources to substrate resources in an optimal way is commonly known as the Virtual Network Embedding problem. This is typically

TABLE I
TERMINOLOGY USED THROUGHOUT THIS PAPER

Term	Description
$SN = (N, L)$	SN is a substrate network, consisting of nodes N and links L
$VNR^i = (N^i, L^i)$	VNR^i denotes the i^{th} Virtual Network Request, consisting of nodes N^i and links L^i
$\dot{R} = \prod_{j=1}^{m} R_j$	$\dot{R}$ contains resource vectors for all resources $R_1, ..., R_m$
$cap : N \cup L \rightarrow \dot{R}$	The function cap assigns a capacity to an element of the substrate network (either node or link)
$dem_i : N^i \cup L^i \rightarrow \dot{R}$	The function dem_i assigns a demand to an element of VNR^i (either a node or a link)
$f_i : N^i \rightarrow N$	f_i is the function that maps a virtual node of VNR^i to a substrate node (VNoM)
$g_i : L^i \rightarrow SN' \subseteq SN$	g_i is the function that maps a virtual link of VNR^i to a path in the substrate network (VLiM)

modeled by annotating a Virtual Network Request (VNR) with node and link demands. Likewise, the substrate network (SN) is annotated with node and link resources (also depicted in Fig. 3). Demands and resources then have to be matched in order to complete the embedding. This means that virtual resources are first *mapped* to candidate substrate resources. Only if all virtual resources can be mapped, the entire network is then *embedded* and substrate resources are actually spent. If VNRs arrive one at a time, reconfiguration might be necessary, reverting the previous embedding and calculating a new mapping.

Formally, the VNE problem can be described as follows (see Table I): Let $SN = (N, L)$ be a substrate network where N represents the set of substrate nodes and L the set of substrate links and let $VNR^i = (N^i, L^i)$ be a set of $i = 1, ..., n$ Virtual Network Requests where N^i and L^i represent the set of virtual nodes and virtual links of the VNR i respectively. Furthermore, let $\dot{R} = \prod_{j=1}^{m} R_j$ be a vector space of resource vectors over resource sets $R_1, ..., R_m$ and let $cap : N \cup L \rightarrow \dot{R}$ be a function that assigns available resources to elements of the substrate network. Finally, for each VNR^i, let $dem_i : N^i \cup L^i \rightarrow \dot{R}$ be a function that assigns demands to elements of all Virtual Network Requests. Then, a Virtual Network Embedding consists of two functions $f_i : N^i \rightarrow N$ and $g_i : L^i \rightarrow SN' \subseteq SN$ for each VNR^i such that $\forall n^i \in N^i : dem_i(n^i) \leq cap(f_i(n^i))$ and $\forall l^i \in L^i : \forall l \in g_i(l^i) : dem_i(l^i) \leq cap(l)$. f_i is then called a node mapping function (VNoM) and g_i is called a link mapping function (VLiM). Together, they form an embedding for VNR^i. It is not required that these functions are calculated by a single entity – calculation can be split among multiple entities.

Solving the VNE problem is $\mathcal{NP}$-hard, as it is related to the multi-way separator problem [16]. Even with a given virtual node mapping, the problem of optimally allocating a set of virtual links to single substrate paths reduces to the unsplittable flow problem [17], [18], and thus also is $\mathcal{NP}$-hard. Therefore, truly optimal solutions can only be gained for small problem instances. Thus, currently the main focus of work within the research community is on heuristic or meta-heuristic approaches.

B. A Virtual Network Embedding taxonomy

The generic problem described in the previous paragraph is characterized by three constraints. Depending on the scenario, modification and relocation of virtual resources may be necessary. As such, VNE approaches either have to be static (i.e. with unchanging infrastructures) or dynamic (taking changes in virtual and substrate infrastructure into account). Moreover, virtual networks might be spread over the substrate infrastructure of multiple InPs. In this case, VNE has to be performed in a distributed way with multiple entities contributing to the mapping. Finally, depending on the scenario, virtual resources may be realized either concise, i.e. minimizing substrate resource usage, or redundant, combining multiple substrate resources to realize one virtual resource.

Instead of a particular property of a VNE algorithm, these constraints are rather different variants of the underlying VNE problem. Thus, all VNE approaches proposed in the literature can be categorized according to whether they are *Static* or *Dynamic*, *Centralized* or *Distributed*, and *Concise* or *Redundant*. These six concepts will be described here and used later on in Section IV as a taxonomy to classify current VNE approaches.

1) Static vs. Dynamic: In most real-world situations, VNE has to be tackled as online problem. That is, VNRs will not be known in advance. Instead, they arrive to the system dynamically and can stay in the network for an arbitrary amount of time. To be realistic, the VNE algorithm has to handle the VNRs as they arrive, rather than attending a set of VNRs at once (offline VNE). While in principle, all approaches can be operated in an online manner, static VNE approaches do not contemplate the possibility of remapping one of more VNRs to improve the performance of the embedding in the SN. Several effects lead to a need for relocation of parts of (or even complete) virtual networks:

- *Fragmentation of SN's resources*: Over time, as new VNRs arrive and are embedded and others expire and release their resources from the SN, the embedding becomes fragmented and the ratio of accepted VNRs diminishes, resulting in a long-term revenue abatement.
- *Changes in the VN*: Before its lifetime expires, a VN may change in terms of topology, size and resources due to new requirements demanded by its users.
- *Changes in the SN*: The SN can also suffer from long-term changes. InPs must be updating its networking infrastructure from time to time to cope with scalability issues and, hence, the SN extends its size and current VNEs can find different and more optimal allocations.

Dynamic VNE approaches try to reconfigure the mapped VNRs in order to reorganize the resource allocation and

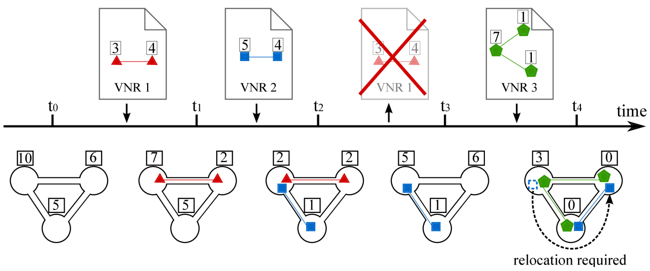

Fig. 4. Relocation of mapped VNRs in online VNE

optimize the utilization of SN resources. For example, Fig. 4 shows the online embedding of three VNRs that arrive and leave the SN at different times. The last one of them (VNR 3) cannot be embedded if the resources of the SN are not first defragmented by reconfiguring the already mapped VN 2.

A good example of dynamic VNE can be found in [19]. Here, the authors realize that most of the VNR rejections are caused by the bottlenecked substrate links. To improve the rejection rate and the load balance in the SN, they propose a reactive and iterative algorithm called Virtual Network Reconfiguration (VNRe). The algorithm is reactive because it just acts when a VNR is rejected and works as follows: In first place it sorts the mapped virtual nodes by their suitability for migration, then it migrates the most suitable virtual node and its attached virtual links to another substrate node and tries to map again the VNR. If the network can not be mapped, the next iteration of the algorithm migrates the following virtual node and the process is repeated until the VNR is mapped or until a predefined number of iterations is reached. Performance results show a significant increase of mapped VNRs after VNRe is applied.

2) Centralized vs. Distributed: The VNE problem can be solved in either a centralized or in a distributed way. Both approaches are fundamentally different, each having its own advantages and disadvantages.

a) Centralized: In a centralized approach there will be one entity which is responsible for performing the embedding. This can be a dedicated machine computing optimal solutions to the problem. The advantage of this approach lies in the fact that the mapping entity is at every step of the mapping aware of the overall situation of the network (i.e. it has global knowledge). This facilitates more optimal embeddings. On the other hand, a centralized entity presents a single point of failure – if it fails, the entire mapping process fails. Moreover, there may be scalability problems in large networks, where a single mapping entity may be overwhelmed by the number of VNRs to handle.

b) Distributed: Contrary to centralized solutions, a distributed approach utilizes multiple entities for computing the embeddings. There may be some internal organization in how the mapping is distributed among the participating entities, or it may be organized completely ad-hoc. The advantage of such an approach lies in its better scalability. Since the load is distributed among several nodes, each individual node will be better able to cope with the embeddings. However, one has to pay for this with synchronization overhead. In particular, each node needs sufficient information about the global state of the network. The more information is available to a node, the better the results will be. However, there is also increased overhead – as such, the situation becomes a trade-off between communication cost and quality of the embeddings.

One specialization of this approach is the situation, where multiple Infrastructure Providers each only map part of a Virtual Network (InterInP VNE). In this case, even if individual InPs use a centralized approach within their own network, the overall procedure is to be considered distributed.

3) Concise vs. Redundant: A failure of a single substrate entity will affect all virtual entities that are mapped upon it. Therefore, in environments where fault-sensitive applications are deployed inside the virtual networks, it can be advisable to set-up backup resources that can be used as fall-back resources in case the corresponding primary resources fail. To do that, the embedding result itself can be redundant to be resilient regarding node and/or link failures. Otherwise, the embedding result is referred to be "concise" if there is no redundancy.

a) Concise: The embedding results of *concise* approaches only use as many substrate resources as necessary to meet the demands of the virtual networks. There is no reservation of additional, redundant resources. This means that in case some of the substrate devices fail, there is no guarantee that the virtual network can recover from failure. However, since the approaches aim to be concise and to only use as much resources as necessary, the saved resources can be used to embed further virtual resources.

b) Redundant: A redundant approach, however, reserves additional resources for the virtual entities that can be used in case some substrate resources fail at run-time. In general, there is a trade-off between the reliability of an embedding and its embedding costs: The higher the degree of reliability, the higher the embedding costs, the more resources are used, and the less virtual entities can be embedded.

After the embedding has been done, the substrate entities have to be monitored. In case an instance fails, there needs to be a fall-back mechanism that is able to switch from the primary instance that has failed to one of its backup instances (e.g., update routing tables) and activate it.

One can also include embedding algorithms that map virtual link demands in multiple paths of the SN inside the redundant category. Single-path approaches map virtual links to exactly one communication path within the substrate network. In contrast, multi-path approaches might split demanded bandwidth of virtual links to multiple substrate paths. In case a virtual link is mapped to multiple substrate paths and some of these paths fail due to a failure within the network, packages can still be routed through the remaining communication links that have been set up between the substrate nodes by changing the splitting ratio. No reconfiguration has to be done, so it is fully transparent. However, bandwidth constraints have to be considered to avoid overloading of substrate link capacity.

C. A notation scheme for VNE algorithms

Each of the categories in the previous section are mutually independent. An algorithm can be, for example, centralized, dynamic, and redundant at the same time. Based upon this, a generic notation can be derived to determine the class a specific VNE algorithm belongs to. Thus, VNE algorithms are described with the following syntax:

[C|D]/[S|D]/[C|R]

The first character denotes, whether the algorithm is **C**entralized or **D**istributed. Likewise, the second character denotes whether the algorithm is **S**tatic or **D**ynamic. Finally, the third character denotes whether the algorithm is **C**oncise or **R**edundant. So, an algorithm denoted as **C/D/R** will be a centralized, dynamic, redundant algorithm. This allows for quick categorization of any given algorithm and proper comparison with similar approaches.

III. Computing optimized embeddings

This section elaborates on: the different *parameters* that can be taken into account for VNE, the possible *objectives* that the embedding may pursue, the *problem decomposition and coordination* applied by existing VNE approaches, the *optimization strategies* used by those approaches to solve VNE, the various *metrics* that can be used to judge the quality of VNE, and a set of publicly available *tools* to evaluate the performance of VNE algorithms.

A. VNE parameters

The resources investigated in the VNE problem are attributed with parameters. On the one hand, substrate resources have individual capacities and qualities. On the other hand, virtual resources each have their respective requirements. For example, a substrate node can provide a certain computing capacity relating to the CPU available to it. Per contra, a virtual node will require a certain computing capacity in order to properly compute routing information. These parameters are of paramount importance in order to achieve a valid embedding. Here, only the different kinds of parameters are discussed. A list of possible parameters is given both in [20] and [21].

One important distinction is that between *linear* and *non-linear* parameters. The linear programming (LP) techniques described in section III-D require linear parameters to find optimal solutions. If non-linear parameters (e.g. path loss probability) are considered, LP can not be used.

Apart from classifying linear and non-linear parameters, parameters can also be categorized according to several other dimensions. As a first step, one can distinguish between *node* and *link* parameters. Node parameters are attributes that refer to nodes, like computing power. Link parameters are attributes that refer to links, like bandwidth. However, there is a problem arising with such a strict model. When a virtual link is mapped to a path in the substrate network, the computing power of substrate nodes on the path may have an impact on the bandwidth of the virtual link [22]. As such, virtual nodes are somewhat easier to map than virtual links, since the latter consist of a combination of substrate node and links, whereas the former takes only substrate nodes into account.

Realizing this, one can further distinguish between *primary* and *secondary* parameters [20]. Primary parameters are parameters that can be directly assigned to a substrate resource and, likewise, can be explicitly required by a virtual network request. Secondary parameters, on the other hand, depend on other (primary) parameters. For example, the probability of packet loss at a node depends both on its computing power and the size of its memory. Primary parameters will be easier to regard during the embedding, since they can be directly matched, whereas secondary parameters have to be calculated first.

Packet loss can also be a property of links (e.g. wireless links). In that case, one can realize that there is also a difference between *consumable* resources and *static* resources. Consumable resources are resources that are consumed when a virtual entity is mapped. The standard examples for this are computing power and bandwidth. In some VNE approaches though, the variability of the demanded bandwidth in a mapped virtual link is taken into account either by considering it a stochastic variable [23] or by opportunistically sharing it among different flows [24], [25]. Static resources, on the other hand, do not depend on the number of virtual resources mapped to a substrate resource. The loss probability of a wireless link in the substrate network will stay the same, no matter how many virtual links are mapped to it (provided the consumable resource "bandwidth" is not depleted). Static resources are easy to match during the embedding. Consumable resources, on the other hand, are what makes the embedding typically $\mathcal{NP}$-complete.

Finally, one can further distinguish between *functional* and *non-functional* parameters. Functional parameters are parameters that specify low-level functionality. Computing power and bandwidth are again good examples. Non-functional parameters, on the other hand, are high-level properties of the respective entities. Examples are security or resilience of an entity. The matching of functional parameters is typically straightforward (although possibly hard to compute), whereas non-functional parameters will require more elaborate approaches in an embedding algorithm.

B. Main embedding objectives

VNE consists of finding the optimal f_i and g_i functions in order to solve VNoM and VLiM with respect to a particular objective. This section describes the objectives that have been pursued by existing approaches to solve VNE.

1) Provide QoS-compliant embeddings: Virtual Network Requests are installed and operated by the VNO according to a set of quality of service constraints defined by the service provider. These QoS requirements must be fulfilled by the virtual network embedding performed by the VNO.

There are several situations where these requirements are explicit in the request. For instance, a virtual network that provides VoIP services needs to count on medium bandwidth, low delays and high CPU requirements. Another example could be a virtual network offering P2P services that must provide medium bandwidth requirements, no relevant delay bounds and medium CPU requirements [26]. This can be achieved, e.g., by minimizing substrate resource stress (i.e. by distributing load equally across the substrate network).

2) Maximize the economical profit of the InP: From the InP point of view, a natural objective of an online embedding algorithm would be to maximize the economic benefit of

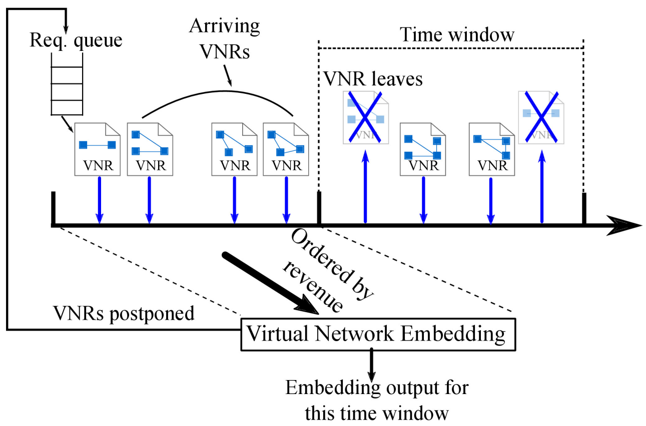

Fig. 5. Ordering by revenue in online VNE

accepting VNRs (long-term average revenue). This objective is directly proportional to maximize the number of embedded VNRs (acceptance ratio). In order to reach this goal, VNE approaches should try to minimize the resources spent by the SN to map a VNR, also known as embedding cost. In this way, it is easier to embed the next VNRs, resulting in an increment of the VN acceptance ratio. However, as the revenue depends on the VNRs, a pre-ordering of the set of VNRs to be embedded may benefit the long-term average revenue. This pre-ordering must be carried out taking into account the online nature of the virtual network embedding problem.

The behavior of a typical online VNE algorithm is shown in Fig. 5. Arriving VNRs are processed within time windows as well as in a request queue. If a request within a time window can not be embedded, it is deferred to the request queue and, in the next time window, tried again. This procedure is repeated until the time the VNR is willing to wait expires. In this case, the request is dropped.

To maximize the revenue, inside the time window, the VNRs can be decreasingly ordered by revenue. In [27], Chowdhury et al. present an approach (WiNE algorithm) that implements this procedure. Obtained results show that WiNE improves the mapped revenue when compared against online algorithms performing the embedding at the arrival of a VNR.

3) Provide survivable VNEs: Resilience in terms of VNE can be brought into play by integrating fallback resources within the substrate network. Backup nodes/links can be setup either for all or just for some specific primary nodes/links that may fail. At any time, consistency of the network topology has to be guaranteed, especially regarding the resources that were defined to be resilient towards failure. Recovery from failures should be transparent for the user. That is, he should not notice that the network switched to the backup resources. Even when using time-sensitive applications, the user should not notice that something went wrong. This especially requires that, for selecting backup resources, all QoS requirements of the primary entities have to be considered.

The backup resources themselves can be either dedicated or shared [28]. Dedicated means that for each virtual network a complete backup network can be set up and backup resources are fully dedicated to the virtual networks and independent from each other. However, this is resource inefficient, since for each virtual resource that gets embedded a dedicated substrate entity is needed. In some cases it might also be acceptable to share and reuse the backup resources in order to reduce the footprint on the substrate network caused by the additional backup resources. Usually, a higher degree of reused backup resources results in lower reliability, and vice versa.

Furthermore, (shared) backup resources can be allocated either in advance (i.e., before the *first* virtual network embedding request arrived) or "on-demand" (i.e., allocated for each embedding request). Shared on-demand backup resources can be assigned at embedding time, that is, each time a virtual network request arrives [29], [28], [30]. Shared pre-allocation algorithms, however, define some specific backup resources in the configuration phase, i.e., before any virtual network request arrives [31].

C. Problem decomposition and coordination

As it has been already stated in the formulation section (see Section II-A), the VNE problem is solved when its two sub-problems VNoM, represented by the f_i function, and (VLiM), represented by the g_i function, are solved. Looking at different InPs, one can decompose the InterInP problem (i.e. the embedding across several InPs) into a set of IntraInP problems (i.e. a set of embeddings within each InP). If these problem decompositions are not coordinated, optimization in one part can jeopardize optimization in another part. This subsection will discuss the options to handle this kind of coordination.

One alternative for VNoM/VLiM coordination is to solve each sub-problem in an isolated and independent way. In this case, VNoM must be solved in first place because it provides the input to solve VLiM. This variant is called *uncoordinated VNE*. On the contrary, some VNE approaches have improved the performance of the solution by providing coordination between the two phases. This variant, called *coordinated VNE*, can be solved either in two different and coordinated stages, or in just one stage. Finally, some VNE approaches look for the mapping of a VNR across heterogeneous InPs. This variant, called *InterInP coordination* aims to split the VNRs in different sub-requests and find the most adequate InP to map each of them.

1) Uncoordinated VNE: A lack of coordination between VNoM and VLiM implies that the solution is generated in two different stages. The first stage solves VNoM and provides the function f_i to the VLiM, which is then solved in a second stage. An example that demonstrates this lack of coordination between the VNE sub-problems was proposed in [32]. The main goal of this approach is to maximize the long-term average revenue. VNE is solved in two independent phases: VNoM follows a greedy algorithm, that chooses, for each virtual node, a set of eligible substrate nodes and then assigns one of them based on its amount of available resources. The aim is to assign the virtual nodes with bigger demands to the substrate nodes with bigger resources. Depending on the assumption taken for the SN, VLiM is solved in two different ways: Single path mapping using one k-shortest path [33] solution for increasing k, when each virtual link must be mapped just to a single path in the SN and Multiple path

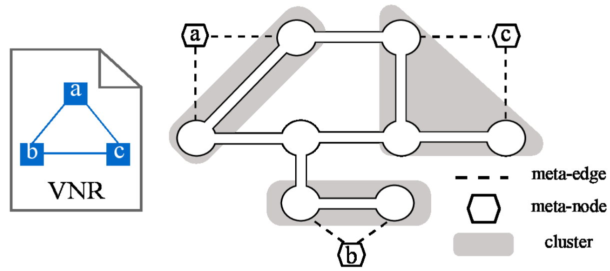

Fig. 6. Augmented substrate graph with meta-nodes and meta-edges for a VNR (cf. [36])

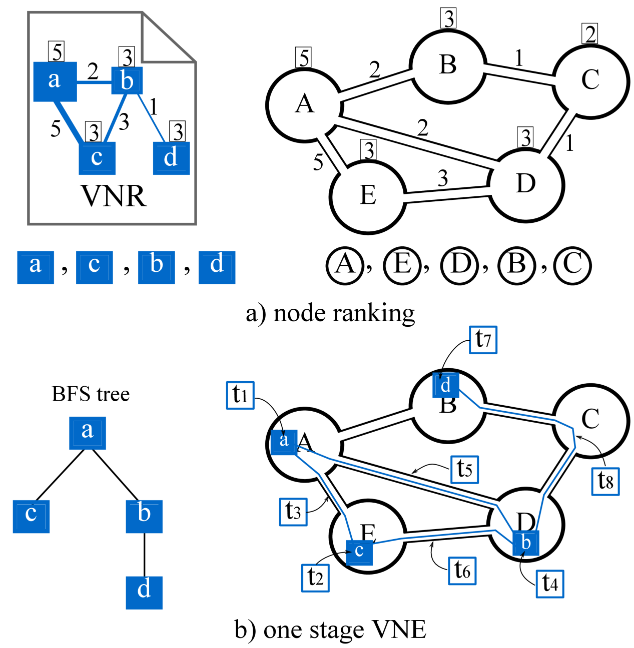

Fig. 7. VNE in one stage, as proposed in [37]

mapping when each virtual link demand can be carried by several paths in the SN. In the latter case, VLiM is reduced to the Multicommodity Flow Problem (MCF) problem [34] that provides a multi-path routing solution for each virtual link using optimal linear programming algorithms [35].

However, the lack of coordination between node and link mapping might result in neighboring virtual nodes being actually widely separated in the substrate topology. This fact increases the cost of single/multi paths used to solve the virtual link mapping phase resulting in low acceptance ratio and, therefore, low long-term revenue.

2) Coordinated VNE: A coordination between node and link mapping is desirable. If VNoM is performed without considering its relation with link mapping, the solution space is restricted and the overall performance of the embedding decreases. Coordination of VNE can be achieved in *two stages* when f_i is provided trying to obtain a result that optimizes the result of g_i. Alternatively, coordination can be also performed in *one stage* by solving the VNoM and VLiM at the same time.

a) Two stages coordinated VNE: An approach that illustrates the coordination in two stages was first proposed by Chowdhury et al. in [36]. Its objective is to minimize the embedding cost. A new set of node constraints is added: geographical location for substrate and virtual nodes and a non-negative distance per VNR indicating how far a virtual node of the VNR can be of its demanded location.

The node mapping stage starts by creating an augmented graph over the SN, introducing a set of meta-nodes, one per virtual node, each connected to a cluster of candidate SN nodes obeying location and capacity constraints. Over this augmented graph, the algorithm solves VNoM by proposing a Mixed Integer Programming (MIP) formulation. The MIP main goal is to solve the VNE trying to minimize the embedding cost, considering that the added meta-nodes map the requested virtual nodes (see Fig. 6). To avoid the $\mathcal{NP}$-completeness of the MIP, its linear programming relaxation is solved and virtual nodes are mapped to real substrate nodes (not meta-nodes) by rounding the obtained solution in two different ways: deterministically or randomly (each resulting in a different algorithm: DViNE and RViNE). After that, VLiM is performed following the same two solutions proposed in [32].

Coordination between both stages is strong, since the MIP formulation used for the virtual node mapping also considers the mapping of virtual links in the augmented SN between source and destination meta-nodes. Therefore, the substrate nodes chosen to map the nodes of the VNR are likely suited to provide virtual link mappings with low embedding cost.

b) One stage coordinated VNE: Solving the VNE in one single stage implies that virtual links are mapped at the same time as virtual nodes. When the first virtual node pair is mapped, the virtual link between them is also mapped and, as each virtual node is mapped, the virtual links connecting it with already mapped virtual nodes are also mapped.

One good example for this variant is the approach proposed in [37] where one of the most important parameters of a network node (substrate or virtual) is its position inside the topology. Topological attributes of a node have a direct impact on the efficiency of the embedding. Taking that into account, a node ranking approach, inspired by the PageRank algorithm used by Google's search engine, is proposed to measure the topology incidence of a node. When topology attributes are incorporated in node mapping, the acceptance ratio and the link mapping efficiency are improved. This is due to the fact that, given two nodes that are equal in resources, the node with the more capable neighborhood will be chosen, leading to a higher success probability for the embedding.

Fig. 7 shows how the approach works. Based on breadth-first search (BFS), the proposed algorithm (*RW-BFS*) solves VNE in one stage considering the impact of node mapping on the link mapping stage. The first step of *RW-BFS* is to calculate node ranks for substrate and virtual nodes as shown in Fig. 7a). It then builds a BFS tree of the VNR, where the root node is the virtual node with greatest rank and the children are placed from left to right based on their rank. VNE is performed by going through the BFS tree and mapping each virtual node in the first feasible substrate node of its list and, at the same time, mapping the virtual links incident to that virtual node onto the substrate shortest paths that satisfy the BW demands. Fig. 7b) shows how the one stage VNE is performed. At t_1 and t_2

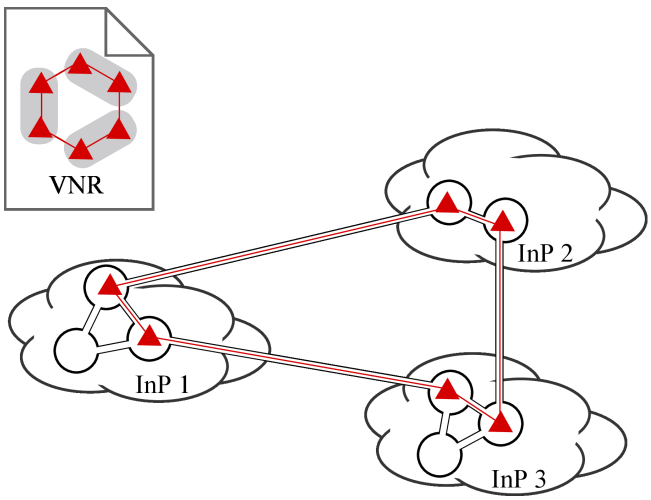

Fig. 8. Inter-InP VNE process

the virtual nodes a and c are mapped on top of the substrate node A and E respectively. At t_3 the virtual link between already mapped nodes (a and c) is mapped. Subsequent virtual nodes and links are mapped following the same procedure, combining, in this way, the virtual node and link mapping in one single stage.

3) InterInP Coordination: It is worth noting that previous algorithm variants consider the VNE in the single-InP scenario (IntraInP VNE). However, in realistic settings, VNRs must be mapped on top of a set of SNs managed by different InPs. Each InP provider should be able to embed parts of the virtual network and connect them using the external links among InPs. Therefore, to minimize the embedding cost, the SP splits the VNRs in several sub-requests and map each of them in the most convenient SN. Inside each InP, the sub-requests are mapped using ordinary VNE algorithms. Fig. 8 shows a split VNR and its mapping to a set of SNs belonging to different InPs.

A good example of InterInP VNE is presented in [38]. Here, the authors address the conflict of interest between SPs and InPs. On the one hand, each InP strives to optimize the allocation in its equipment by getting requests for their high-margin equipment while offloading unprofitable work onto their competitors. On the other hand, the SPs are interested in satisfying their demands while minimizing their expenditure. The approach proposes a distributed protocol that coordinates the InPs and ensures competitive embedding pricing for the SPs.

D. Optimization strategies

The VNE problem is $\mathcal{NP}$-hard (see section II-A). Therefore, for large problem sizes (i.e. large SN and VNRs size) the time to find the optimal solution becomes unaffordable. Taking this into account, three different types of approaches have been used to solve VNE. *Exact solutions* propose optimal techniques to solve small instances of the problem and to create baseline solutions that represent an optimal bound for heuristic-based VNE solutions. *Heuristic-based solutions* are not fixed on obtaining the global optimum. Instead, they try to find a good solution while keeping execution time low. Usually, heuristic solutions suffer from the problem that they can get stuck in a local optimum that can be far away from the real optimum. *Metaheuristic solutions* improve the quality of the result by escaping from local optima in reasonable time.

1) Exact solutions: Optimal VNE solutions can be achieved by means of Linear Programming (LP). More exactly, Integer Linear Programming (ILP) can be used to optimally formulate the VNE including the virtual node and link mapping subproblems in the same formulation. Although integer linear programs are in many practical situations $\mathcal{NP}$-complete, there are exact algorithms (e.g. branch and bound, branch and cut and branch and price [39]) that solve small instances of the problem in reasonable time. Software tools implementing these algorithms, commonly called *solvers*, are available either as open-source (e.g. GLPK [40]) or proprietary (e.g. CPLEX [41]).

Some approaches have used ILP to solve VNE. For example, in [42], the VNE ILP formulation seeks for the minimization of the embedding cost and the maximization of the acceptance ratio. The novel energy aware VNE introduced in [43] uses an ILP exact formulation where the goal is to embed the VNs' demand in a reduced set of equipment in the SN to save energy by switching off the remaining SN resources.

2) Heuristic solutions: Execution time is crucial in VNE. Network virtualization deals with dynamic online environments where VNRs arrival time is not known in advance. Therefore, to avoid delay in the embedding of a new VNR, the execution time of VNE algorithms should be minimized. Accordingly, heuristic-based VNE solutions are proposed. They attempt to find an acceptable solution, compromising optimality for short execution time.

One good example of heuristic-based VNE approaches is presented in [44]. It proposes virtual node and link mapping in a single stage. To do so, VNE is reduced to the well known $\mathcal{NP}$-hard Subgraph Isomorphism Detection (SID) problem. In graph theory, an isomorphism of graphs G and H ($G \simeq H$) is a bijection between the vertex sets of G and H, $m : V(G) \to V(H)$, such that any two vertices i and j of G are adjacent in G if and only if $m(i)$ and $m(j)$ are adjacent in H. The $\mathcal{NP}$-complete SID problem tries to find a subgraph G_{sg} of G ($G_{sg} \subset G$) such that $G_{sg} \simeq H$. A modification of an existing heuristic is proposed to solve the VNE. It consists of finding an isomorphic subgraph (representing the VNR), accomplishing the VNR demands inside the substrate network, and using a hop limit constraint for the substrate paths used for the mapping of virtual links.

3) Metaheuristic solutions: VNE can be seen as a combinatorial optimization problem where an optimal solution is sought over a discrete search-space. As the optimal solution for large instances of these problems is hard to find, metaheuristics like simulated annealing [45], genetic algorithms [46], ant colony optimization [47], particle swarm optimization [48] or tabu search [49] can be used to find near-optimal solutions by trying to improve a candidate solution with regard to a given measure of quality. The following two approaches provide good examples of metaheuristic-based solutions for VNE:

An approach based on the Max-Min Ant Colony metaheuristic [50] has been recently proposed to solve the VNE in [51]. The problem is divided into a set of solution components (equivalent to small parts of the overall solution) and then, a set of parallel artificial ants are launched to iteratively explore the search space until a predefined number of iterations is reached. During each iteration, each ant builds incrementally the solution by transiting from one solution component to another. After all ants finish their full solution, the best one is selected as the solution of that iteration. Finally, the best solution among all iterations is chosen as the final embedding solution.

A particle swarm optimization (PSO) based approach is proposed in [52]. PSO is a population-based stochastic global optimizer that generates near-optimal solutions in low computing time with stable convergence. In this paper, a PSO based VNE (*VNEUEPSO*) is proposed where each particle is a possible VNE solution that iteratively improves its position according to a fitness function (embedding cost). Finally the approximate optimal VNE solution is obtained through the evolution process of the particles. Each time, particles change their positions by modifying VNoM, VLiM is used then to guarantee the feasibility of the solution.

E. Embedding metrics

Metrics are necessary to evaluate the quality of a successful embedding. They are used to compare different VNE approaches and to quantify advances in optimization. Within this survey paper, different metrics are structured according to the embedding objectives indicated in Section III-B with "Other metrics" denoting metrics that don't exactly match one of the three objectives:

- Quality of Service metrics
- Cost-related metrics
- Resilience metrics
- Other metrics

Table II gives an overview over the various metrics discussed in this paper. Depending on the application scenario, one can compute for most metrics the average, the maximum, and/or the minimum value.

1) Quality of Service metrics: Quality of Service metrics aim to measure the impact of an embedding with respect to the service quality when using the virtual network. For example, when a user wants to run realtime applications like video telephony on a virtual network, the impact of the actual embedding that was chosen should not be perceptible. In addition to typical QoS metrics, metrics like the average length of paths between nodes should be taken into account because of the likely effect on forwarding delay. Furthermore, utilization of network resources should also be considered: When a lot of different virtual entities are mapped upon a physical resource, the physical resource has to manage the load somehow. This could also include some additional scheduling overhead for switching between different virtual allocations.

a) Path length: The path length metric measures the number of links between two substrate nodes that are mapped to two interconnected virtual nodes. The longer a corresponding path, the more resources had to be reserved for the embedding of the virtual link. Since every substrate node (except the receiving node) that is part of a path will take some time to forward packages sent via this path, the quality of service is influenced by the path length. In general, the package delay increases in connection with the length of a path.

b) Stress level: The stress level of a substrate entity reflects the number of virtual entities that are mapped onto it. The more virtual entities use the same substrate resource, the higher the impact regarding possible side effects. For example, mapping many virtual nodes onto a single substrate node keeps the CPU of the host operating system busy. A high substrate link stress might result in some additional packet delay because the resources of the substrate link and the host interfaces communicating through this link have to be shared between virtual entities.

c) Utilization: Utilization measures, in each SN entity (node or link), the sum of the spent substrate resources due to the mapping of virtual entities divided by the total amount of resources. This metric is a more precise measurement tool than the stress level metric, because it also takes into account the magnitude of the resource usage instead of simply counting the number of virtual entities that use a resource. For example, to measure the utilization of a substrate node, the sum of mapped CPU resources divided by the CPU capacity of the node could be used. The usage of a substrate link could be based on the sum of mapped bandwidth resources divided by the total link bandwidth capacity.

d) Throughput: Throughput is measured after the embedding of a virtual network has been done. For each virtual node pair, packages between the corresponding substrate node pair are generated and sent through the path connecting these nodes. Then, the maximal data rate that can be transmitted through all connections between source-destination pairs is determined.

e) Delay: Delay describes the amount of time needed for a packet to go from one node in the network to another node. With regard to VNE, algorithms can optimize the mapping of virtual resources so as to minimize delay between virtual nodes.

f) Jitter: Jitter measures the variance in packet interarrival times. In a network virtualization scenario, jitter can be inherent to the substrate network (e.g. due to unreliable links) or introduced by virtualization itself (e.g. due to concurrent resource usage by two different virtual entities). The latter is strongly dependent on the employed virtualization solution and cannot be solved by VNE. Jitter that is inherent to the substrate network, on the other hand, can be taken into account by VNE algorithms. However, so far (to the best of the authors knowledge) there is no VNE approach that focuses on minimizing jitter.

2) Resource spending metrics:

a) Cost: Cost in the VNE context refers to the amount of substrate resources that were used for embedding virtual networks. Cost is usually determined by summing up all CPU and bandwidth resources of the substrate network that have been reserved for VNRs. However, besides CPU and bandwidth, additional types of resources can also be taken into account, and different types of resources can optionally be

TABLE II
METRICS FOR VIRTUAL NETWORK EMBEDDING

Optimization goal	Metric	Description
Quality of Service	Path length	Describes the number of substrate links that are spanned by a virtual link on average
	Stress level	Describes the number of virtual entities realized by a substrate entity
	Utilization	Describes the sum of all spent substrate resources due to VNE divided by the sum of all provided substrate resources
	Throughput	Describes the data rate achievable between virtual nodes
	Delay	Describes the time a packet needs to travel across a virtual link
	Jitter	Describes the variance in inter-arrival times of packets on a virtual link
Resource spending	Cost	Describes the sum of all spent substrate resources for embedding VNRs
	Revenue	Describes the sum of all demanded resources of VNRs
	Cost/Revenue	Describes the ratio between spent substrate resources and provided virtual resources
	Acceptance ratio	Describes the number of VNRs that could be embedded
Resilience	Number of backups	Describes the number of available backup resources
	Path redundancy	Describes the diversity of paths in multi-path embeddings
	Cost of resilience	Describes the number of additional nodes required to maintain resiliency
	Recovery blocking probability	Describes the ratio of unrecoverable failure scenarios vs. all failure scenarios
	Number of migrations	Describes the number of virtual nodes that have to be moved in case of failure
Other	Runtime of the algorithm	Describes the time a VNE algorithm will take for an embedding of a certain size
	Number of coordination messages	Describes the number of messages that have to be exchanged in a distributed environment in order to complete the embedding
	Active substrate nodes	Describes the number of substrate nodes that have to be powered on in order to realize the hosted virtual infrastructures

weighted in dependence on their value range. Cost is directly related to the length of substrate paths: The longer the length of a path, the more substrate resources are used and the higher the costs for an embedding.

b) Revenue: Revenue refers to the sum of virtual resources that were actually requested by the virtual entities. This value is usually computed by applying the same scheme that was used to determine Cost.

c) Cost/Revenue: To compare algorithms with respect to their embedding results, typically many different network topologies with varying size and properties are generated. Depending on these random topologies, Costs vary and therefore impede a fair comparison.

So in addition to Costs, Revenue is typically also taken into account. By dividing Cost by Revenue, varying Cost values are balanced. The higher the value, the more resources were needed to embed the VNs.

d) Acceptance ratio: The acceptance ratio metric measures the number of virtual network requests that could be completely embedded by the embedding algorithm, divided by the total number of virtual network requests.

3) Resilience-related metrics:

a) Number of backups: The number of backups metric counts the number of backup resources that are set up for a virtual entity. Several additional substrate entities can be reserved to serve as a replacement in case the entity hosting the virtual entity fails.

b) Path redundancy: Path redundancy measures the ratio between the number of backup paths to the number of direct paths. Some redundancy algorithms set up backup paths that can be used in case some parts of the network break down. Therefore, the metric refers to the amount of additional resources that are used to backup the embedded network.

c) Cost of resilience: Related to the number of additional nodes required to maintain resilience, this metric measures the ratio of total number of running nodes and number of backup nodes. In contrast to path redundancy, this metric does not focus on connectivity resources but includes demanded node resources.

d) Recovery blocking probability: When a substrate entity fails, the substrate network has to perform re-organizing actions to recover from failure. Especially, compensatory resources have to be allocated to catch the outage. Some approaches do not reserve these extra capacities in advance, so the system has to identify suitable backup resources at run time. Due to limited capacities of the entities inside the substrate network, this might fail which results in a failure of recovery. The recovery blocking probability metric measures the ratio of the number of unrecoverable failure scenarios to the total number of failure scenarios.

e) Number of migrations: The number of migrations refers to the number of virtual nodes that need to be migrated to new facility nodes in case corresponding substrate nodes fail. Typically, at least the virtual nodes hosted on the failing substrate nodes have to be moved. Other constraints, like maximum path length, might however trigger even more migrations. Since migrations are resource intensive, they should be kept to a minimum.

4) Other metrics:

a) Runtime of the algorithm: Runtime of algorithms compares algorithms with respect to the time that they need to compute an actual embedding result. For most real-life systems, runtime is a crucial factor, with a direct tradeoff between timely completion and quality of embedding results (e.g., cost/revenue).

b) Number of coordination messages: For a distributed embedding approach, various messages between substrate nodes need to be exchanged for coordination purposes. The number of coordination messages is one related metric that can be used to determine and compare the communication overhead between different distributed approaches.

c) Active substrate nodes: The number of active substrate nodes is related to the average length of substrate paths, because additional nodes are used to forward communication data between end nodes. Therefore, the probability that previously switched off nodes are selected to forward data rises. Regarding energy efficiency, the number of nodes that need to be turned for an embedding can be a rough estimation of energy consumption. In general, the ratio between running nodes and the total number of substrate nodes should be taken into account.

F. Simulation tools

In order to evaluate VNE algorithms, simulation tools have been developed by a number of authors. Typically, algorithms are run with a randomly generated set of scenarios. Each of these scenarios consists of a SN and a number of VNRs to be embedded. Appropriate parameters are assigned to both substrate and virtual resources. After the algorithms have tried to embed the VNRs, the results are evaluated using one or more of the metrics described in Section III-E. VNE practitioners can use these simulation tools to develop and test their own VNE algorithms or to compare the performance of existing algorithms with regard to new metrics. This subsection presents three exemplary implementations of VNE simulation tools. Although this list is far from being comprehensive, it still indicates where one can start with practical VNE experiments.

The first example is the Vineyard VNE simulation tool by Chowdhury[1]. It has been used to validate the D-ViNE and R-ViNE VNE algorithms [36], [27]. Vineyard uses the GT-ITM network topology generator for random scenario generation. It is written in C++ with some accompanying shell scripts. It can evaluate revenue, cost, stress, and acceptance ratios. It has been extended in several other papers [31], [53], [54] to cover new VNE aspects.

Another example for a VNE simulation tool is the VNE Simulator developed by Yu[2]. Similar to Vineyard, the tool is written in the C programming language with some supporting shell scripts. It also uses the GT-ITM network topology generator to generate network scenarios. It has been introduced in [32] to demonstrate the advantages of path splitting in virtual networks. Since then it has also been used and extended in a number of other publications [37], [55], [56].

[1]http://www.mosharaf.com/ViNE-Yard.tar.gz

[2]https://github.com/minlanyu/embed

The authors of this paper have collaborated to create an easy to use VNE simulation environment called "ALEVIN" [20]. The results have been made available online[3]. ALEVIN uses a Waxman generator to create random network topologies. ALEVIN is a Java project with well-defined interfaces to implement new VNE algorithms and metrics. It comes with a number of pre-implemented algorithms drawn from several VNE papers. Moreover, it comes with extensive users and developers documentation, making it easy for others to jumpstart their evaluation. Several of the metrics mentioned in section III-E have already been implemented in ALEVIN. By now, ALEVIN has been used in a number of publications [57], [58], [43] to compare VNE algorithms and evaluate new approaches to the VNE problem.

IV. A CLASSIFICATION OF VNE APPROACHES

In this section, the taxonomy developed in Section II-C is used to classify the different approaches proposed to solve VNE by the research community. The categories discussed in Section II-C are used to group similar approaches. The two Tables III and IV list approaches in their respective categories. Each approach is further characterized, considering the coordination of VNE subproblems, as introduced in Section III-C. Moreover, for each approach the applied optimization strategy is shown (see Section III-D). Further comments describe individual contributions by each approach. Approaches that appear in more than one category are marked with an asterisk (*). This happens, when an approach can be implemented in two ways (e.g. either static or dynamic).

The **C/S/C** category groups the set of approaches that take the straightforward solution, where aspects such as distributed behavior, dynamicity and redundancy are not considered. Algorithms in this category will operate in a centralized manner, expecting full knowledge of VNRs in advance, and providing concise solutions to the problem. The first VNE algorithms that have been proposed took this approach, with many other algorithms following. These approaches are listed in the upper part of Table III.

The VNE approaches belonging to the **C/D/C** category are grouped in the second part of Table III. This set of proposals is able to perform a dynamic reconfiguration of currently mapped VNRs once a new VNR arrives. Mappings are concise and the algorithms run on a central instance. Dynamic approaches typically are more challenging – this leads to fewer publications in this category, compared to centralized, static approaches.

The VNE approaches belonging to the **C/S/R** category are again centralized and static. However, they consider redundancy. An example for this are embeddings that use multi-path virtual link mapping solutions. The VLiM stage is $\mathcal{NP}$-hard if each virtual link must be allocated to a single path in the SN (cf. Sec. II-A). However, if the virtual link can be mapped to multiple paths, VLiM may be reduced to the well-known multicommodity flow problem that can be solved in affordable execution time. As a consequence, a significant number of the existing approaches solve the VNE using multi-path for

[3]http://alevin.sf.net/

TABLE III
TAXONOMY OF CONCISE VNE APPROACHES

Category	Reference	Optimization	Coordination	Contribution
C/S/C	[26] Inführ and Raidl (2011)	Exact	One Stage	Provides delay, location and routing constraints
	[59] Liu et al. (2011)	Exact	One Stage	Exact VNE based on correspondence matrices
	[60] Trinh et al. (2011)	Exact	One Stage	Exact VNE problem with SLA QoS guarantees
	[61] Pages et al. (2012)	Exact/Metaheuristic	One Stage	Introduces the VNE for optical networks
	[44] Lischka and Karl (2009)	Heuristic	One Stage	Provides one stage VNE. Based on SID
	[62] Di et al. (2010)	Heuristic	One Stage	Improvement of the approach in [44]
	[63] Ghazar and Saaman (2011)	Heuristic	One Stage	Introduces hierarchical management of the SN
	[64], [65] Yun et al. (2011-2012)	Heuristic	One Stage	First VNE approach in wireless multihop networks. Introduces metrics and feasibility measures for wireless VNE
	[56] Chen et al. (2012)	Heuristic	One Stage	Reduces resource fragmentation
	[66] Yu et al. (2012)	Heuristic	One Stage	One step VNE that increases coordination
	[67] Liu et al. (2011)	Heuristic	Two Stages	Improves coordination based on nodes proximity
	[24], [25] Sheng et al. (2011-2012)	Heuristic	Two Stages	Opportunistic resource sharing to deal with load fluctuation
	[68] Li et al. (2012)	Heuristic	Two Stages	Topology awareness to enforce VNE coordination
	[69] Lu and Turner (2006)	Heuristic	Uncoordinated	Embedding in specific backbone-star VN topologies
	[32] Yu et al.* (2008)	Heuristic	Uncoordinated	Utilizes the KSP algorithm [33] for VLiM
	[70] Razzaq and Siraj (2010)	Heuristic	Uncoordinated	Different K values in KSP based VLiM
	[71] Razzaq et al. (2011)	Heuristic	Uncoordinated	Investigates the VNE impact of bottlenecked nodes
	[72] Nogueira et al. (2011)	Heuristic	Uncoordinated	VNE considering SN resources heterogeneity
	[73] Leivadeas et al.* (2011)	Heuristic	Uncoordinated	Introduces VNE for wireless network testbeds
	[22], [57] Botero et al. (2011-2013)	Heuristic	Uncoordinated	Introduces hidden hop constraints
	[74] Zhu and Ammar* (2006)	Heuristic	Uncoordinated	Provides a balanced link and node stress in the SN
	[51] Fajjari et al. (2011)	Metaheuristic	One Stage	Max-Min Ant Colony metaheuristic VNE approach
	[75] Cheng et al. (2012)	Metaheuristic	One Stage	Accelerates convergence of PSO VNE metaheuristic with topology aware node ranking [37]
	[76] Zhang et al. (2012)	Heuristic	Uncoordinated	Maps one virtual node in several substrate nodes
	[77] Di et al. (2012)	Heuristic	One Stage	Coordinated VNE reducing the number of backtracks by carefully chosing the first virtual node to map
	[78] Abedifar and Eshghi (2012)	Heuristic	Uncoordinated	Introduces VNE in the optical domain trying to minimize the number of λs per link
	[79] Aris Leivadeas et al. (2012)	Heuristic	Coordinated	Considers importance of virtual nodes for embedding
	[80] Tae-Ho Lee et al. (2012)	Heuristic	InterInP	clustering of virtual networks in multi-provider environment
C/D/C	[19] Fajjari et al. (2011)	Heuristic	One Stage	Migration of nodes with bottlenecked adjacent links
	[81], [82] Bienkowski et al. (2010)	Heuristic	Two Stages	Migration when service access position changes
	[74] Zhu and Ammar* (2006)	Heuristic	Uncoordinated	Reduce the cost of periodic reconfigurations
	[83] Fan and Ammar (2006)	Heuristic	Uncoordinated	Reduces the cost of VNRs reconfiguration
	[84] Cai et al. (2010)	Heuristic	Uncoordinated	Reconfiguration based on SN evolution
	[85] Shun-li and Xue-song (2011)	Heuristic	Uncoordinated	Identifies mapped virtual nodes and links with not optimal mapping and migrate them to save SN resources
	[86] Sun et al. (2012)	Heuristic	Uncoordinated	Introduces the VNE problem for evolving VNRs
D/S/C	[87], [88] Houidi et al. (2010)	Heuristic	Uncoordinated	First distributed approach to solve VNE. Proposes a VNE protocol to manage the communication among substrate nodes
	[89] Xin et al. (2011)	Heuristic	InterInP	Introduces the InterInP VNE for networked clouds
	[90] Lv et al. (2011)	Heuristic	InterInP	InterInP VNE using hierarchical virtual resource organization
	[42] Houidi et al.* (2011)	Exact/Metaheuristic	InterInP	VNR is split assigning each subVN in different InPs. Provides exact and heuristic splitting approaches
	[91] Leivadeas et al.* (2012)	Heuristic	InterInP	Graph partitioning InterInP VNE using a heuristic integrating a min k-cut algorithm followed by subgraph isomorphism
D/D/C	[92] Marquezan et al. (2010)	Heuristic	Uncoordinated	First distributed dynamic approach. Reorganizes the SN when VNs demands change

TABLE IV
TAXONOMY OF REDUNDANT VNE APPROACHES

Category	Reference	Optimization	Coordination	Contribution
C/S/R	[42] Houidi et al.* (2011)	Exact	One Stage	First approach providing an ILP exact solution
	[93] Zhang et al. (2011)	Exact	One Stage	Optimal resilient solution attaining an enhanced QoS mapping. Provides diversified substrate back-up paths
	[43] Botero et al. (2012)	Exact	One Stage	Introduces the energy aware VNE
	[94] Wang and Wolf (2011)	Exact	One Stage	Redefines the VNR as a traffic matrix
	[95], [96], [97] Shamsi and Brockmeyer (2007-2009)	Heuristic	One Stage	Recover link failures by providing backup paths with intermediate nodes
	[98] Koslovski et al. (2010)	Heuristic	One Stage	Introduces reliabilty as a service offered by the InP. Reliable VNEs based on subgraph isomorphism detection
	[99] Yu et al. (2010)	Heuristic	One Stage	Introduces failure-dependent protection with a back-up solution for each regional failure
	[100] Lv et al. (2012)	Heuristic	One Stage	Introduces looses to multicast VNE in wireless mesh networks
	[36], [27] Chowdhury et al. (2009-2011)	Heuristic	Two Stages	Coordination in VNE using multi-path for VLiM
	[31] Rahman et al. (2010)	Heuristic	Two Stages	Upon a failure, the economic penalty is minimized by the pre-reservation of a bandwidth quota for back-up in SN links
	[53] Butt et al.* (2010)	Heuristic	Two Stages	VNE awareness of the SN bottlenecked resources
	[29] Yeow et al. (2010)	Heuristic	Two Stages	Introduces sharing among back up resources. Reduces resources allocated for redundancy
	[101] Sun et al. (2011)	Heuristic	Two Stages	Resilient VNE optimizing the embedding cost and reducing computational complexity
	[30] Yu et al. (2011)	Heuristic	Two Stages	Resilient VNE analyzing failures in substrate nodes
	[32] Yu et al.* (2008)	Heuristic	Uncoordinated	Introduces the multi-path approach for VLiM
	[102] Gao et al. (2010)	Heuristic	Uncoordinated	Improvement of the approach in [36]
	[103] Yang et al. (2010)	Heuristic	Uncoordinated	Divides the SN in regions to reduce VNE complexity
	[104] Zho et al. (2010)	Heuristic	Uncoordinated	Maps one virtual node to multiple substrate nodes
	[105] Chen et al. (2010)	Heuristic	Uncoordinated	Reactive resiliency protection approach against failures during the online VNE process. Considers just substrate link failures
	[106] Yu et al. (2011)	Heuristic	Uncoordinated	Proactive VNE approach offering protection against SN link failures for links with high stress
	[23] Sun et al. (2011)	Heuristic	Uncoordinated	Introduces stochastic BW demand to the VNE
	[55] Lu et al. (2011)	Heuristic	Uncoordinated	Introduces load balancing in links
	[28] Guo et al. (2011)	Heuristic	Uncoordinated	Proactive resilient VLiM approach sharing back-up paths
	[37] Cheng et al. (2011)	Metaheuristic	Two Stages	Introduces topology-awareness in VNE
	[107] Sheng et al. (2011)	Metaheuristic	Two Stages	Embedding time depends on VNR lifetime. Uses simulated annealing metaheuristic
	[52] Zhang et al. (2012)	Metaheuristic	Two Stages	Introduces particle swarm optimizaton (PSO) metaheuristic
	[108] Sun et al. (2012)	Metaheuristic	Two Stages	Introduces VNE in multi-datacenter environments
	[109] Lv et al. (2012)	Metaheuristic	Uncoordinated	Introduces VNE in wireless mesh networks
	[91] Leivadeas et al.* (2012)	Heuristic	Two Stages	Uses the approach in [27] to solve the VNE for an arbitrary pool of heterogeneous resources
	[54] Masti and Raghavan (2012)	Heuristic	Two Stages	VNE considering the residual capacity of the substrate links
	[110] Zhang et al. (2012)	Exact/Heuristic	One Stage	Recover link failures providing disjoint SN backup paths
C/D/R	[53] Butt et al.* (2010)	Heuristic	Two Stages	Reactive reconfiguration of virtual links and nodes causing rejection to less critical SN regions
	[32] Yu et al.* (2010)	Heuristic	Uncoordinated	Reconfigure the embedding by changing the splitting ratio in the multipath VLiM solution
	[111] Schaffrath et al. (2010)	Exact	One Stage	ILP-based VNE. Dynamically reconfigures existing mappings
	[112] Chen et al. (2011)	Heuristic	Two Stages	Periodic reconfiguration of SN nodes with high utilization
D/S/R	[38] Chowdhury et al. (2010)	Heuristic	InterInP	First InterInP VNE proposal. Mediates between InP and SP interests. VNR is split across InPs and embedded locally
D/D/R	[113] Houidi et al. (2010)	Heuristic	Two Stages	Fault-tolerant VNE that acts upon node and link failures

VLiM. The first part of Table IV shows the taxonomy of the approaches in this category.

The second part of Table IV groups the set of approaches that use redundancy to perform the reconfiguration of already mapped VNRs. The algorithms still run on a central instance. It can be seen that only few approaches try to combine redundancy and dynamicity.

Until now, there has been little interest of the research community in finding distributed solutions for the VNE. This is likely due to the fact that distributed algorithms are significantly harder to implement. Moreover, it will be more difficult to reach near-optimal solutions with a distributed algorithm. As such, there are only a few approaches to list here. The lower parts of Tables III and IV show the existing approaches in the **D/S/C**, **D/S/R**, **D/D/C** and **D/D/R** categories, respectively.

V. Emerging Research Directions

This section highlights the future research directions of VNE. Three main fields that may propel the VNE research in the near future are identified: approaches inside the categories of *distributed* proposals, VNE proposals looking for the optimization of *new objectives* and the application of VNE in specific network *environments*.

A. Distributed VNE

Since the VNE problem is $\mathcal{NP}$-hard, finding an optimal embedding is computationally complex. Therefore, most approaches tend to relax the original problem, focusing on near-optimal solutions. However, even these heuristic approaches do not scale well for large networks [87]. Instead of relying on a single, central node that computes all the embeddings, VNRs can be distributed to multiple substrate nodes. In this way, load will be spread, possibly increasing scalability. Tables III and IV show that, currently, there is a lack of distributed solutions for the VNE.

The only static and concise approach that performs Intra-InP VNE in a distributed way was proposed by Houidi et al. [87]. Although it solves the weaknesses of centralized approaches, it suffers from two main problems: excessive number of coordination messages and suboptimal embedding cost. Therefore, a good line of future research is the definition of distributed Intra-InP approaches trying to reduce the message overhead by, for instance, using clustering techniques that, in turn, confine the embedding tasks to a limited subset of the SN, thereby reducing embedding cost.

Until now, there is also a lack of VNE distributed solutions providing resiliency. The main interest of the research community has focused on offering resiliency by setting up back-up resources at provisioning time in a centralized way. However, the field of reactive and distributed embedding reconfiguration that acts in real time upon a failure has been only treated in one paper so far [113]. Unfortunately, it does not evaluate the optimality of the solution after reconfiguration and results in a considerable overhead of interchanged messages for big SNs.

B. Emerging VNE objectives

Recently, VNE has acquired importance in two fields: Green Networking and Security.

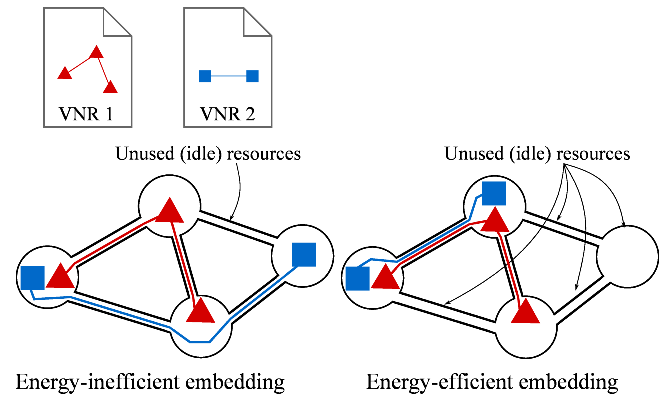

Fig. 9. Energy-efficiency of different embeddings

1) Green Networking: Resource consolidation (several virtual instances in one physical resource), will be an enabler for energy savings in future infrastructure networks. Currently, some green strategies to allocate resources in cloud computing environments have been proposed. In [114], Chang and Wu propose an heuristic approach that seeks for the minimization of the computing and communications power of applications instantiated in a cloud substrate. In [115], Yang et al. propose a Green Power Management (GPM) to perform the virtual machine migration in cloud environments thanks to a dynamic resource allocation that seeks the ideal load balancing amongst virtual machines.

In network virtualization environments, if the VNE main goal changes to the minimization of the energy consumption, the SN can be dynamically dimensioned for current traffic demand rather than for peak demand. Due to power consumption insensitiveness of current network equipment to traffic load [116], the best approach to minimize the energy consumption is to switch off or hibernate as many networking resources (nodes and links) as possible without compromising the network performance. The energy-aware dynamic VNE should be performed during low traffic demand periods, when some routers and interfaces can be switched off by rerouting the traffic to a smaller set of consolidated network equipment, as shown in Fig. 9. Due to limited resource capacities, this sometimes leads to longer substrate paths between entities, which not only has side-effects to communication delays but also increases embedding costs. So in general, there is a trade-off between these objectives which should be taken into account.

The first approach proposed in this area is called Energy Aware VNE (VNE-EA) [43] and its main goal is to minimize the switched on nodes and links after a VNR is mapped. However, VNE-EA uses exact ILP and just provides an optimal bound to evaluate future heuristic based algorithms. Furthermore, networking resources are assumed to be homogeneous with regard to their energy consumption which is not a realistic assumption. The study also concludes that, as VN traffic load increases, the solution saves energy but at high costs in terms of acceptance ratio.

Future research could be focused on the provision of heuristic energy-aware strategies that find near-optimal solutions within reasonable time. It is important also to avoid rejecting

VNRs when shifting to more energy efficient VNE solutions. Besides, topology and load dependence are also non-negligible aspects in energy consumption. Current ISPs topologies are hierarchical with many network elements consuming low energy amounts at the bottom and few network elements consuming higher amounts in the top. The recent development of green ICT equipment [117] - where the energy consumption depends on the load - can also be subject of future research in energy-aware VNE.

2) Security: In a virtualized future Internet, VNOs are envisioned to rent infrastructure via VNPs from InPs, as indicated in Section I. Virtual networks can be hosted on the hardware of multiple different InPs. At the same time, one InP can host networks from multiple VNPs/VNOs. As such, there will be security requirements on the one hand by the VNOs, and on the other hand by the InPs. Both parties have an interest to protect their respective assets (i.e. nodes and links – either physical or virtual).

In such an environment, the different stakeholders each have their own security requirements. A VNO is interested in having his network hosted on hardware that can offer a sufficient level of security. On the other hand, an InP will want to ensure that virtual networks are properly secured and do not run havoc on his equipment. Finally, different VNOs may distrust one another and require that their virtual infrastructure is not cohosted on the same physical equipment in order to minimize the risk of cross-virtualization attacks.

This issue can not simply be solved by installing additional software in either the virtual or the physical nodes. Instead, it is necessary to avoid mappings that will increase risk for one of the stakeholders. Thus, there is a need for security-aware VNE algorithms. A security-aware VNE algorithm will have to take these requirements into account, trying to minimize risk exposure for all involved stakeholders. As such, it is necessary to allow VNOs and InPs to express their security needs. A VNE algorithm should then try to match those requirements as closely as possible [118].

C. VNE environments

The maturity of network virtualization is a motivating factor for its application in specific network environments. For this reason, VNE strategies are migrating from theoretical to real scenarios trying to deal with their specific constraints.

1) Wireless networks: Wireless networks have become one of the main type of access technologies nowadays, and virtualization is expected to be applied in wireless scenarios as well. Wireless links have broadcast nature and, consequently, the main distinctive feature of wireless network virtualization is how to virtualize links. Up to know, there has been some work on link virtualization based on time-division multiplexing (TDM) [119], [120], frequency-division multiplexing [121] and space division multiplexing [122], [123].

The main challenge to overcome in VNE for wireless network comes from the broadcast nature of wireless links that may cause interference with other wireless links [124].

Recently, some approaches dealing with VNE in wireless environments have been proposed. An approach to solve VNE in TDM-based wireless virtualization environments was proposed in [64], [65], here the authors introduce the *feasibility checking* to examine whether an embedding solution is feasible (not easy in wireless environments due to interference) and *embedding performance* to provide a measure on how good an embedding solution is. The approach, however, does not consider the time-varying conditions that could strongly affect the embedding performance.

VNE solutions for FDM-based virtualization has been recently provided in [73], [109], [100]. The approach in [73] introduces VNE for wireless network testbeds based on FDM link virtualization, while in [109], [100], the authors introduce VNE for wireless mesh networks (WMNs).

As can be noted, until now, VNE in space division multiplexing-based wireless virtualization environments remains unexplored. Besides, existing approaches miss some paramount characteristics of wireless environments that should be subject of future work, namely:

- Mobility: One of the main features of wireless network environments, is the mobility of their nodes. As a consequence, VNE dynamic approaches that consider nodes' mobility as a trigger for mapping reconfiguration could be incorporated to current proposals.
- Distribution: Some wireless network environments are characterized by the lack of a centralized management entity, e.g. ad-hoc networks. Therefore, distributed VNE approaches in wireless networks deserve more interest in the near future.

2) Optical networks: The concept of virtual optical networks (VONs) is introduced in [61], in turn, virtual optical network mapping is proposed. The VNE problem in optical environments is defined as the maximization of the number of mapped VONs from the demand set given the limited capacity of the optical SN. Two different variants of the problem are proposed:

- *Transparent optical mapping*: In this variant of the problem, optically transparent end-to-end services are provisioned over the VON. Transparent services mean that the optical VN is not assumed to have electronic termination capabilities in nodes. Therefore, the VON must allocate the same set of wavelengths for every virtual link.
- *Opaque optical mapping*: This variant assumes electronic capabilities in the optical VN, so that it is not necessary to allocate the same set of wavelengths for each virtual link thanks to the Optical-Electrical-Optical (OEO) conversion.

The problem is formulated and solved with ILP techniques. To reach feasible running times for larger scenarios, a metaheuristic based on GRASP [125] is also proposed.

Another VNE approach in the optical domain is presented in [78]. Here, Abedifar and Eshghi propose a VLiM approach that routes the virtual links on the physical infrastructure and assigns the available wavelengths to the resulting light paths trying to minimize the used wavelenghts per physical link.

These are the first approaches in the optical networks field. However, they open some questions, e.g. how to include the effect of Physical Layer Impairment (PLI) degradations in the optical VNE feasibility, especially in larger networks scenarios.

VI. CONCLUSION

Virtual Network Embedding is a central problem to be solved when networks are virtualized. Optimizing the embedding of multiple virtual networks on one substrate network is computationally difficult for a number of important metrics. Multiple algorithms approaching this problem have been discussed in the literature, so far.

This paper presented a survey of current work in this area. A formal description of the VNE problem was provided. A categorization of VNE algorithms along three distinct dimensions (static vs. dynamic, centralized vs. distributed, concise vs. redundant) was developed. A list of optimization metrics was presented. A number of algorithmical approaches to the VNE problem was discussed. Finally, this information was used to create a taxonomy of VNE algorithms proposed in the literature.

There are a number of opportunities for future work in this area. It is to be noted, that the category of distributed VNE algorithms has received only sparse attention, so far. This provides an excellent point for future work, in particular since centralized algorithms will always be prone to the criticism of having a single-point of failure. Moreover, there are also novel directions of VNE research like energy-efficiency or security which have also been largely neglected by the scientific community up to now. New work in this area will have to define appropriate new metrics for the VNE problem and develop algorithms that will optimize according to energy-saving or security-enhancing goals. The application of virtualization to real networking environments, e.g. wireless networks, is currently being studied by the scientific community. The definition of algorithms that study the VNE problem with a focus on environment-based constraints is also an exciting branch for future research.

ACKNOWLEDGMENTS

This work has received support by the EU projects All4Green (FP7 STREP, grant no. 288674), EuroNF (FP7 NoE, grant no. 216366), EINS (FP7 NoE, grant no. 288021), and the Spanish Government project MICINN (grant no. TIN2010-20136-C03-01).

REFERENCES

[1] N. M. M. K. Chowdhury, "Network virtualization: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20–26, Jul. 2009.

[2] A. Berl, A. Fischer, and H. de Meer, "Virtualisierung im future internet - virtualisierungsmethoden und anwendungen," *Informatik-Spektrum*, vol. 33, no. 2, pp. 186–194, Apr. 2010.

[3] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Network virtualization: Implementation steps towards the future internet," in *Proc. Workshop on Overlay and Network Virtualization at KiVS*, Kassel, Germany, mar 2009.

[4] A. Berl, A. Fischer, and H. de Meer, "Using system virtualization to create virtualized networks," in *Workshops der Wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen (WowKiVS2009)*, Kassel, Germany, Mar. 2009, vol. 17, EASST.

[5] L. Peterson, S. Shenker, J. Turner *et al.*, "Overcoming the internet impasse through virtualization," *IEEE Computer*, vol. 38, no. 4, pp. 34–41, Apr. 2005.

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

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

[8] P. Endo, A. de Almeida Palhares, N. Pereira, G. Goncalves, D. Sadok, J. Kelner, B. Melander, and J. Mangs, "Resource allocation for distributed cloud: concepts and research challenges," *IEEE Network*, vol. 25, no. 4, pp. 42 –46, july-august 2011.

[9] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," *Computer Communication Review*, vol. 37, no. 1, 2007.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[11] ONF, "Open networking foundation," `https://www.opennetworking.org/`, 2011.

[12] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: A study of infrastructure as a service (iaas)," *Int J. Eng. and Information Technology*, vol. 2, pp. 60–63, 2010.

[13] 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. 1st ACM workshop on Virtualized infrastructure systems and architectures*, ser. VISA '09. New York, NY, USA: ACM, 2009, pp. 63–72.

[14] A. Haider, R. Potter, and A. Nakao, "Challenges in resource allocation in network virtualization," in *20th ITC Specialist Seminar*, vol. 18, 2009, p. 20.

[15] A. Belbekkouche, M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Commun. Surveys Tutorials*, vol. PP, no. 99, pp. 1–15, 2012.

[16] D. G. Andersen, "Theoretical approaches to node assignment," Dec. 2002, unpublished Manuscript.

[17] J. Kleinberg, "Aproximation algorithms for disjoint paths problems," Ph.D. dissertation, Massachusetts Institute of Technology, 1996.

[18] S. Kolliopoulos and C. Stein, "Improved approximation algorithms for unsplittable flow problems," in *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, oct 1997, pp. 426 –436.

[19] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "Vnr algorithm: A greedy approach for virtual networks reconfigurations," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, dec. 2011, pp. 1 –5.

[20] A. Fischer, J. F. Botero, M. Duelli, D. Schlosser, X. Hesselbach, and H. De Meer, "ALEVIN - a framework to develop, compare, and analyze virtual network embedding algorithms," *Electronic Communications of the EASST*, vol. 37, pp. 1–12, 2011.

[21] D. Stezenbach, M. Hartmann, and K. Tutschku, "Parameters and challenges for virtual network embedding in the future internet," in *First IEEE Workshop on Algorithms and Operating Procedures for Federated Virtualized Networks (FEDNET2012)*, 2012.

[22] J. F. Botero, X. Hesselbach, A. Fischer, and H. De Meer, "Optimal mapping of virtual networks with hidden hops," *Telecommunication Systems*, vol. 52, no. 3, pp. 1–10, 2013, the original publication is available at www.springerlink.com (March 2011).

[23] G. Sun, H. Yu, L. Li, V. Anand, Y. Cai, and H. Di, "Exploring online virtual networks mapping with stochastic bandwidth demand in multi-datacenter," *Photonic Network Communications*, pp. 1–14, 2011.

[24] S. Zhang, Z. Qian, B. Tang, J. Wu, and S. Lu, "Opportunistic bandwidth sharing for virtual network mapping," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, dec. 2011, pp. 1 –5.

[25] S. Zhang, Z. Qian, B. Tang, J. Wu, and S. Lu, "An opportunistic resource sharing and topology-aware mapping framework for virtual networks," in *INFOCOM 2012. 31th IEEE International Conference on Computer Communications*, 2012, to appear.

[26] J. Inführ and G. R. Raidl, "Introducing the virtual network mapping problem with delay, routing and location constraints," in *Proceedings of the 5th international conference on Network optimization*, ser. INOC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 105–117.

[27] M. Chowdhury, M. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206 –219, feb. 2012.

[28] T. Guo, N. Wang, K. Moessner, and R. Tafazolli, "Shared backup network provision for virtual network embedding," in *Communications (ICC), 2011 IEEE International Conference on*, june 2011, pp. 1 –5.

[29] W.-L. Yeow, C. Westphal, and U. Kozat, "Designing and embedding reliable virtual infrastructures," in *Proc. second ACM SIGCOMM*

workshop on Virtualized infrastructure systems and architectures, ser. VISA '10. New York, NY, USA: ACM, 2010, pp. 33–40.

[30] H. Yu, V. Anand, C. Qiao, and G. Sun, "Cost efficient design of survivable virtual infrastructure to recover from facility node failures," in *Communications (ICC), 2011 IEEE International Conference on*, june 2011, pp. 1 –6.

[31] M. R. Rahman, I. Aib, and R. Boutaba, "Survivable virtual network embedding," in *Networking*, 2010, pp. 40–52.

[32] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM CCR*, vol. 38, no. 2, pp. 17–29, Apr. 2008.

[33] D. Eppstein, "Finding the k shortest paths," *SIAM J. Comput.*, vol. 28, pp. 652–673, February 1999.

[34] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[35] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, 1997.

[36] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM*. IEEE Infocom, Apr. 2009.

[37] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 38–47, April 2011.

[38] M. Chowdhury, F. Samuel, and R. Boutaba, "Polyvine: policy-based virtual network embedding across multiple domains," in *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, ser. VISA '10. New York, NY, USA: ACM, 2010, pp. 49–56.

[39] L. A. Wolsey, *Integer Programming*. New York: John Wiley & Sons, 1998.

[40] GLPK, "GNU Linear Programming Kit," `http://www.gnu.org/software/glpk`, 2008.

[41] ILOG, Inc, "ILOG CPLEX: High-performance software for mathematical programming and optimization," `http://www.ilog.com/products/cplex/`, 2012.

[42] I. Houidi, W. Louati, W. B. Ameur, and D. Zeghlache, "Virtual network provisioning across multiple substrate networks," *Computer Networks*, vol. 55, no. 4, pp. 1011 – 1023, 2011, special Issue on Architectures and Protocols for the Future Internet.

[43] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. de Meer, "Energy efficient virtual network embedding," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 756 –759, may 2012.

[44] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proc. 1st ACM workshop on Virtualized infrastructure systems and architectures*. New York, USA, Aug. 2009, pp. 81–88.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[46] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992.

[47] D. M., "Optimization, learning and natural algorithms," *Ph.D. Thesis, Politecnico di Milano, Italy*, 1992.

[48] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, nov/dec 1995, pp. 1942 –1948 vol.4.

[49] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, pp. 533–549, May 1986.

[50] T. Stützle and H. H. Hoos, "Max-min ant system," *Future Gener. Comput. Syst.*, vol. 16, pp. 889–914, June 2000.

[51] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "Vne-ac: Virtual network embedding algorithm based on ant colony metaheuristic," in *Communications (ICC), 2011 IEEE International Conference on*, june 2011, pp. 1 –6.

[52] Z. Zhang, X. Cheng, S. Su, Y. Wang, K. Shuang, and Y. Luo, "A unified enhanced particle swarm optimization-based virtual network embedding algorithm," *International Journal of Communication Systems*, 2012.

[53] N. F. Butt, N. M. M. K. Chowdhury, and R. Boutaba, "Topology-awareness and reoptimization mechanism for virtual network embedding," in *Networking*, 2010, pp. 27–39.

[54] S. Masti and S. Raghavan, "Vna: An enhanced algorithm for virtual network embedding," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, August 2012, pp. 1 –9.

[55] B. LU, T. HUANG, Z. kai WANG, J. ya CHEN, Y. jie LIU, and J. LIU, "Adaptive scheme based on status feedback for virtual network mapping," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 5, pp. 87 – 94, 2011.

[56] X. Chen, Y. Luo, and J. Wang, "Virtual network embedding with border matching," in *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, jan. 2012, pp. 1 –8.

[57] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. de Meer, "Flexible VNE Algorithms Analysis using ALEVIN," in *Euroview 2011*, Aug. 2011.

[58] M. Duelli, D. Schlosser, J. F. Botero, X. Hesselbach, A. Fischer, and H. De Meer, "Vnreal: Virtual network resource embedding algorithms in the framework alevin," in *Proc. of the 7th Euro-NF Conf. on Next-Generation Internet (NGI 2011)*. IEEE, 2011, pp. 1–2.

[59] W. Liu, Y. Xiang, S. Ma, and X. Tang, "Completing virtual network embedding all in one mathematical programming," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, sept. 2011, pp. 183 –185.

[60] T. Trinh, H. Esaki, and C. Aswakul, "Quality of service using careful overbooking for optimal virtual network resource allocation," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*, may 2011, pp. 296 –299.

[61] A. Pages, J. Perello, S. Spadaro, and G. Junyent, "Strategies for virtual optical network allocation," *IEEE Commun. Lett.*, vol. 16, no. 2, pp. 268 –271, february 2012.

[62] H. Di, L. Li, V. Anand, H. Yu, and G. Sun, "Cost efficient virtual infrastructure mapping using subgraph isomorphism," in *Communications and Photonics Conference and Exhibition (ACP), 2010 Asia*, dec. 2010, pp. 533 –534.

[63] T. Ghazar and N. Samaan, "A hierarchical approach for efficient virtual network embedding based on exact subgraph matching," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, dec. 2011, pp. 1 –6.

[64] D. Yun and Y. Yi, "Virtual network embedding in wireless multihop networks," in *Proc. 6th International Conference on Future Internet Technologies*, ser. CFI '11. New York, NY, USA: ACM, 2011, pp. 30–33.

[65] D. Yun, J. Ok, B. Shin, S. Park, and Y. Yi, "Embedding of virtual network requests over static wireless multihop networks," *CoRR*, vol. abs/1207.1878, 2012.

[66] H. Yu, V. Anand, C. Qiao, H. Di, and X. Wei, "A cost efficient design of virtual infrastructures with joint node and link mapping," *J. Network and Systems Management*, vol. 20, pp. 97–115, 2012.

[67] J. Liu, T. Huang, J. ya Chen, and Y. Liu, "A new algorithm based on the proximity principle for the virtual network embedding problem," *Journal of Zhejiang University - Science C*, vol. 12, no. 11, pp. 910–918, 2011.

[68] X.-l. Li, H.-m. Wang, C.-g. Guo, B. Ding, X.-y. Li, W.-q. Bi, and S. Tan, "Topology awareness algorithm for virtual network mapping," *Journal of Zhejiang University - Science C*, vol. 13, pp. 178–186, 2012.

[69] J. Lu and J. Turner, "Efficient mapping of virtual networks onto a shared substrate," Washington University in St. Louis, Tech. Rep. WUCSE-2006-35, 2006.

[70] A. Razzaq and M. Rathore, "An approach towards resource efficient virtual network embedding," in *Evolving Internet (INTERNET), 2010 Second International Conference on*, sept. 2010, pp. 68 –73.

[71] A. Razzaq, P. Sjodin, and M. Hidell, "Minimizing bottleneck nodes of a substrate in virtual network embedding," in *Network of the Future (NOF), 2011 International Conference on the*, nov. 2011, pp. 35 –40.

[72] J. Nogueira, M. Melo, J. Carapinha, and S. Sargento, "Virtual network mapping into heterogeneous substrate networks," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, july 2011, pp. 438 –444.

[73] A. Leivadeas, C. Papagianni, E. Paraskevas, G. Androulidakis, and S. Papavassiliou, "An architecture for virtual network embedding in wireless systems," in *Proc. 2011 First International Symposium on Network Cloud Computing and Applications*. IEEE Computer Society, 2011, pp. 62–68.

[74] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proc.*, 2006, pp. 1–12.

[75] X. Cheng, S. Su, Z. Zhang, K. Shuang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology awareness and optimization," *Computer Networks*, vol. 56, no. 6, pp. 1797 – 1813, 2012.

[76] S. Zhang, J. Wu, and S. Lu, "Virtual network embedding with substrate support for parallelization," in *To appear in Proc. IEEE Global Telecommunications Conference*, 2012.

[77] H. Di, H. Yu, V. Anand, L. Li, G. Sun, and B. Dong, “Efficient online virtual network mapping using resource evaluation,” *J. Network and Systems Management*, vol. 20, pp. 468–488, 2012.

[78] V. Abedifar and M. Eshghi, “A novel routing and wavelength assignment in virtual network mapping based on the minimum path algorithm,” in *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*, july 2012, pp. 204 –208.

[79] A. Leivadeas, C. A. Papagianni, and S. Papavassiliou, “Socio-aware virtual network embedding,” *IEEE Network*, vol. 26, no. 5, pp. 35–43, 2012.

[80] T.-H. Lee, S. Tursunova, and T. Choi, “Graph clustering based provisioning algorithm for virtual network embedding,” in *NOMS*. IEEE, 2012, pp. 1175–1178.

[81] M. Bienkowski, A. Feldmann, D. Jurca, W. Kellerer, G. Schaffrath, S. Schmid, and J. Widmer, “Competitive analysis for service migration in vnets,” in *Proc. second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, ser. VISA ’10. New York, NY, USA: ACM, 2010, pp. 17–24.

[82] M. Bienkowski, A. Feldmann, J. Grassler, G. Schaffrath, and S. Schmid, “The wide-area virtual service migration problem: A competitive analysis approach,” *IEEE/ACM Trans. Netw. (ToN)*, 2013.

[83] J. Fan and M. H. Ammar, “Dynamic topology configuration in service overlay networks: A study of reconfiguration policies,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1 –12.

[84] Z. Cai, F. Liu, N. Xiao, Q. Liu, and Z. Wang, “Virtual network embedding for evolving networks,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, dec. 2010, pp. 1 –5.

[85] Z. Shun-li and Q. Xue-song, “A novel virtual network mapping algorithm for cost minimizing,” *Cyber Journals: J. Sel. Areas Telecommunications (JSAT)*, vol. 02, no. 01, pp. 1–9, January 2011.

[86] G. Sun, H. Yu, V. Anand, and L. Li, “A cost efficient framework and algorithm for embedding dynamic virtual network requests,” *Future Generation Computer Systems*, no. 0, pp. –, 2012, available online 25 August 2012.

[87] I. Houidi, W. Louati, and D. Zeghlache, “A distributed virtual network mapping algorithm,” in *Communications, 2008. ICC ’08. IEEE International Conference on*, may 2008, pp. 5634 –5640.

[88] I. Houidi, W. Louati, and D. Zeghlache, “A distributed and autonomic virtual network mapping framework,” in *Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on*, march 2008, pp. 241 –247.

[89] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and A. Yumerefendi, “Embedding virtual topologies in networked clouds,” in *Proc. 6th International Conference on Future Internet Technologies*, ser. CFI ’11. New York, NY, USA: ACM, 2011, pp. 26–29.

[90] B. Lv, Z. Wang, T. Huang, J. Chen, and Y. Liu, “Virtual resource organization and virtual network embedding across multiple domains,” in *Multimedia Information Networking and Security (MINES), 2010 International Conference on*, nov. 2010, pp. 725 –728.

[91] A. Leivadeas, C. Papagianni, and S. Papavassiliou, “Efficient resource mapping framework over networked clouds via iterated local search based request partitioning,” *IEEE Trans. Parallel Distrib. Syst.*, vol. PP, no. 99, p. 1, 2012.

[92] C. Marquezan, L. Granville, G. Nunzi, and M. Brunner, “Distributed autonomic resource management for network virtualization,” in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, april 2010, pp. 463 –470.

[93] X. Zhang, C. Phillips, and X. Chen, “An overlay mapping model for achieving enhanced qos and resilience performance,” in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, oct. 2011, pp. 1 –7.

[94] C. Wang and T. Wolf, “Virtual network mapping with traffic matrices,” in *Architectures for Networking and Communications Systems (ANCS), 2011 Seventh ACM/IEEE Symposium on*, oct. 2011, pp. 225 –226.

[95] J. Shamsi and M. Brockmeyer, “Qosmap: Qos aware mapping of virtual networks for resiliency and efficiency,” in *Globecom Workshops, 2007 IEEE*, nov. 2007, pp. 1 –6.

[96] J. Shamsi and M. Brockmeyer, “Efficient and dependable overlay networks,” in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, april 2008, pp. 1 –8.

[97] J. Shamsi and M. Brockmeyer, “Qosmap: Achieving quality and resilience through overlay construction,” in *Internet and Web Applications and Services, 2009. ICIW ’09. Fourth International Conference on*, may 2009, pp. 58 –67.

[98] G. Koslovski, W.-L. Yeow, C. Westphal, T. T. Huu, J. Montagnat, and P. Vicat-Blanc, “Reliability support in virtual infrastructures,” in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, dec. 2010, pp. 49 –58.

[99] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, “Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, dec. 2010, pp. 1 –6.

[100] P. Lv, Z. Cai, J. Xu, and M. Xu, “Multicast service-oriented virtual network embedding in wireless mesh networks,” *IEEE Commun. Lett.*, vol. 16, no. 3, pp. 375 –377, march 2012.

[101] G. Sun, H. Di, H. Yu, L. Li, and V. Anand, “The Framework and Algorithms for the Survivable Mapping of Virtual Network onto a Substrate Network,” *IETE Technical Review*, vol. 28, no. 5, pp. 381–391, 2011.

[102] X. Gao, H. Yu, V. Anand, G. Sun, and H. Di, “A new algorithm with coordinated node and link mapping for virtual network embedding based on lp relaxation,” in *Communications and Photonics Conference and Exhibition (ACP), 2010 Asia*, dec. 2010, pp. 152 –153.

[103] F. Yang, Z. kai Wang, J. ya Chen, and Y. jie Liu, “Vlb-vne: A regionalized valiant load-balancing algorithm in virtual network mapping,” in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, june 2010, pp. 432 –436.

[104] Y. Zhou, Y. Li, D. Jin, L. Su, and L. Zeng, “A virtual network embedding scheme with two-stage node mapping based on physical resource migration,” in *Communication Systems (ICCS), 2010 IEEE International Conference on*, nov. 2010, pp. 761 –766.

[105] Y. Chen, J. Li, T. Wo, C. Hu, and W. Liu, “Resilient virtual network service provision in network virtualization environments,” in *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, dec. 2010, pp. 51 –58.

[106] Y. Yu, C. Shan-zhi, L. Xin, and W. Yan, “Rmap: An algorithm of virtual network resilience mapping,” in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, sept. 2011, pp. 1 –4.

[107] S. Zhang, Z. Qian, S. Guo, and S. Lu, “Fell: A flexible virtual network embedding algorithm with guaranteed load balancing,” in *2011 IEEE International Conference on Communications (ICC),*, june 2011, pp. 1 –5.

[108] G. Sun, H. Yu, V. Anand, L. Li, and H. Di, “Optimal provisioning for virtual network request in cloud-based data centers,” *Photonic Network Communications*, pp. 1–14, 2012.

[109] P. Lv, X. Wang, and M. Xu, “Virtual access network embedding in wireless mesh networks,” *Ad Hoc Networks*, no. 0, pp. –, 2012.

[110] X. Zhang, X. Chen, and C. Phillips, “Achieving effective resilience for qos-aware application mapping,” *Computer Networks*, vol. 56, no. 14, pp. 3179 – 3191, 2012.

[111] G. Schaffrath, S. Schmid, and A. Feldmann, “Optimizing long-lived cloudnets with migrations,” in *Proc. 5th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, 2012.

[112] D. Chen, X. Qiu, Z. Qu, S. Zhang, and W. Li, “Algorithm for virtual nodes reconfiguration on network virtualization,” in *Advanced Intelligence and Awareness Internet (AIAI 2011), 2011 International Conference on*, oct. 2011, pp. 333 –337.

[113] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, “Adaptive virtual network provisioning,” in *Proc. second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, ser. VISA ’10. New York, NY, USA: ACM, 2010, pp. 41–48.

[114] R.-S. Chang and C.-M. Wu, “Green virtual networks for cloud computing,” in *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on*, aug. 2010, pp. 1 –7.

[115] C.-T. Yang, K.-C. Wang, H.-Y. Cheng, C.-T. Kuo, and W. C. C. Chu, “Green power management with dynamic resource allocation for cloud virtual machines,” in *Proc. 2011 IEEE International Conference on High Performance Computing and Communications*, ser. HPCC ’11, sept. 2011, pp. 726 –733.

[116] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, “Power awareness in network design and routing,” in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, april 2008, pp. 457 –465.

[117] A. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, “A survey of green networking research,” *IEEE Commun. Surveys Tutorials*, vol. 14, no. 1, pp. 3 –20, quarter 2012.

[118] A. Fischer and H. De Meer, “Position paper: Secure virtual network embedding,” *Praxis der Informationsverarbeitung und Kommunikation*, vol. 34, no. 4, pp. 190–193, 2011.

[119] G. Smith, A. Chaturvedi, A. Mishra, and S. Banerjee, “Wireless virtualization on commodity 802.11 hardware,” in *Proc. second ACM*

international workshop on Wireless network testbeds, experimental evaluation and characterization, ser. WinTECH '07, 2007, pp. 75–82.

[120] S. Perez, J. Cabero, and E. Miguel, "Virtualization of the wireless medium: A simulation-based study," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, april 2009, pp. 1 –5.

[121] S. Singhal, G. Hadjichristofi, I. Seskar, and D. Raychaudhri, "Evaluation of uml based wireless network virtualization," in *Next Generation Internet Networks, 2008. NGI 2008*, april 2008, pp. 223 –230.

[122] A. P. Jayasumana, Q. Han, and T. H. Illangasekare, "Virtual sensor networks - a resource efficient approach for concurrent applications," in *Proc. International Conference on Information Technology*, ser. ITNG '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 111–115.

[123] R. Mahindra, G. Bhanage, G. Hadjichristofi, I. Seskar, D. Raychaudhuri, and Y. Zhang, "Space versus time separation for wireless virtualization on an indoor grid," in *Next Generation Internet Networks, 2008. NGI 2008*, april 2008, pp. 215 –222.

[124] K.-m. Park and C.-k. Kim, "A framework for virtual network embedding in wireless networks," in *Proc. 4th International Conference on Future Internet Technologies*, ser. CFI '09. New York, NY, USA: ACM, 2009, pp. 5–7.

[125] M. G. Resende and C. C. Ribeiro, "Greedy randomized adaptive search procedures: Advances, hybridizations, and applications," in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds. Springer US, 2010, vol. 146, pp. 283–319.

Andreas Fischer (http://www.net.fim.uni-passau.de/fischer) received his Diploma in Computer Science from University of Passau in 2008. Since then he has been a scientific assistant at the Computer Networks and Computer Communications group of Prof. De Meer. His research interests are network virtualization and network resilience in virtualized environments. He is currently involved in the EU FP7 NoE EINS: "The Network of Excellence in Internet Science". Previously, he was involved in the EU FP7 projects AutoI, ResumeNet, and EuroNF.

Juan Felipe Botero (https://sites.google.com/site/juanfebotero/) is a Ph.D student at the at the Technical University of Catalonia, UPC, in Barcelona, Spain. In 2006 he received his Computer Science Degree from the University of Antioquia, Colombia, and the M.Sc. degree in Telematics Engenieering in 2008 from the Technical University of Catalonia, UPC, in Barcelona, Spain. In 2008, he joined BAMPLA (Design, and evaluation of Broadband Networks and Services) group in the Telematics Engineering Department and since then he is pursuing his Ph.D. His main research interests include Quality of Service and resource allocation in virtual networks.

Michael Till Beck (http://www.net.fim.uni-passau.de/beck) wrote his master thesis on energy-efficient virtual network embeddings. Since May 2011 he has been working as a research associate at the chair for Computer Networks and Computer Communications under the supervision of Prof. Hermann de Meer pursuing his PhD. He was involved in the ResumeNet project and is currently active in the All4Green project. Since January 2012, he is also working as a research assistant for Stadtwerke Passau GmbH, an energy provider. He is interested in energy efficiency and sustainability related topics. Furthermore, he is also interested in virtual networks and network resilience.

Hermann de Meer (http://www.fim.uni-passau.de/demeer) is currently appointed as Full Professor of computer science (Chair of Computer Networks and Communications) and is director of the Institute of IT Security and Security Law (ISL) at the University of Passau. He had been an Assistant Professor at Hamburg University, Germany, a Visiting Professor at Columbia University in New York City, USA, Visiting Professor at Karlstad University, Sweden, a Reader at University College London, UK, and a research fellow of Deutsche Forschungsgemeinschaft (DFG). He chaired one of the prime events in the area of Quality of Service in the Internet, the 13th international workshop on quality of service (IWQoS 2005, Passau). He has also chaired the first international workshop on self-organizing systems (IWSOS 2006, Passau) and the first international conference on energy-efficient computing and networking (e-Energy 2010, Passau). He currently holds several research grants funded by the Deutsche Forschungsgemeinschaft (DFG) and by the EU (FP7).

Xavier Hesselbach (http://www-entel.upc.edu/xavierh/) is an Associate Professor at ETSETB of the UPC and IEEE member since 1992. He received the M.S. degree in Telecommunications Engineering in 1994 and the PhD. degree in 1999, from the Universitat Politècnica de Catalunya (UPC). In 1993, he joined the Design, modelling and evaluation of broadband networks group in the Telematics Engineering Department of the UPC. His research interests include networks virtualization, resources managenent, broadband networks, quality of service and green networking. He has been involved in several national and international projects, and is author of 4 books and more than 50 national and international publications in conferences and journals. In 1994 he received the award from the COIT/AEIT of Spain for the best Master Thesis on Networks and Telecommunication Services.