



HAL
open science

Virtual resource description and clustering for virtual network discovery

Ines Houdi, Wajdi Louati, Djamel Zeghlache, Stephan Baucke

► **To cite this version:**

Ines Houdi, Wajdi Louati, Djamel Zeghlache, Stephan Baucke. Virtual resource description and clustering for virtual network discovery. ICC 2009 : IEEE International Conference on Communications, Jun 2009, Dresden, Germany. pp.1 - 6, 10.1109/ICCW.2009.5207979 . hal-01368460

HAL Id: hal-01368460

<https://hal.science/hal-01368460v1>

Submitted on 19 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Virtual Resource Description and Clustering for Virtual Network Discovery

Ines HOUIDI, Wajdi LOUATI, Djamel ZEGHLACHE

TELECOM & Management SudParis

9 rue Charles Fourier, 91011 Evry, France

E-mail: {ines.houidi, wajdi.louati, djamel.zeghlache}@it-sudparis.eu

Stephan BAUCKE

Ericsson Research

Ericsson Allee 1, 52134 Herzogenrath, Germany

E-mail: stephan.baucke@ericsson.com

Abstract— This paper explores description and clustering techniques for virtual networks provisioning from shared physical resources. The contribution consists of a virtual resource description schema to specify properties and relations between virtual resources and a conceptual clustering technique to facilitate matching and discovery. Conceptual clustering provides a hierarchical classification of virtual resources and concept descriptions for clusters. Performance results benefits of conceptual clustering for resource matching are reported.

Keywords: Network Virtualization; Virtual Resource Discovery; Virtual Resource Description; Conceptual Clustering

I. INTRODUCTION

Network virtualization has recently received considerable attention in both academia and industry as an important enabler for designing the Future Internet architecture [1-3]. The network virtualization concept allows the setup of separate *Virtual Networks* (VN) running simultaneously on a shared substrate network. Network virtualization is achieved via logical segmentation of the underlying physical network (or substrate) including the virtualization of network nodes and their interconnection. In essence, a Virtual Network, in its elementary form, consists of a group of Virtual Nodes interconnected via dedicated Virtual Links. The objective of virtualization is to make each VN appear to a customer/user as a dedicated physical network with dedicated resources, networking services and security policies. VNs can simultaneously support multiple network architectures, experiments and services (e.g. IPTV, VPNs, etc.).

Virtual networks can be supplied as a service by VN Provider(s) and delivered to VN users to support their infrastructures and services. As depicted in figure 1, there are essentially three main actors involved:

- Infrastructure Provider: the actor that owns and controls the substrate network. The infrastructure providers ensure virtualization of their physical resources (fixed and wireless nodes and links) to offer virtual resources (as a service) to VN providers.
- VN Provider: allocates and aggregates virtual resources made available by Infrastructure Providers to establish VNs for users/customers on demand.
- VN User/Customer: requests virtual networks from VN Providers. These actors correspond to end users, service providers and virtual network operators.

The infrastructure providers describe their virtual resources to make them known to potential actors, especially VN providers. Many of the network description specifications used in the current state of the art (such as cNIS, NDL, NM/PerfSONAR, vgDL) [4-7] can be used to describe physical network elements. However, these specifications do not define a schema for describing virtual resources and virtualization services. The first contribution of this paper consists of defining a preliminary virtual resource description schema to fill this gap, specify virtual resources properties and their relationships.

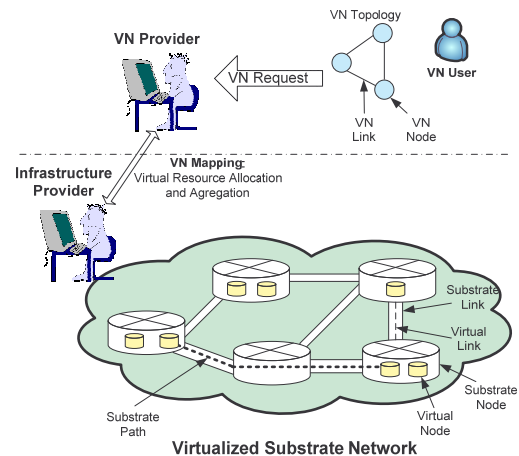


Figure 1. Virtual Networks Provisioning Framework

The infrastructure providers are expected to advertise and register the virtual resource descriptions in a resource discovery framework to make them known to VN providers. Upon receiving a VN request from users, the VN providers in cooperation with the infrastructure providers are responsible for mapping (or embedding) VNs to shared substrate networks. The VN embedding process consists in assigning the VN nodes and links to a specific set of virtual resources extracted from substrate resources (e.g. substrate nodes, substrate links/paths).

The VN embedding occurs in three steps known as discovery (includes matching), selection and binding:

1. Candidate discovery or matching: finds a set of VN candidates (i.e. the appropriate virtual nodes and links) that fulfil the VN request. VN providers will discover and

match user queries for VNs with the virtual resources offered by the infrastructure providers;

2. Candidate selection: chooses the best VN candidate using optimization algorithms such as those described in [8-11];
3. Candidate binding: allocates/reserves virtual resources from the substrate to set up the selected VN candidate.

In this paper we do not focus on how the resource discovery framework can be implemented and used for supporting virtual resource discovery. Since the state of the art is sufficiently rich in searching and finding techniques, we will rather describe how VN providers and infrastructure providers can benefit from description and clustering (i.e. classification) of network resources to facilitate the matching process.

Section II presents the proposed virtual resource description schema while section III is devoted to the resource discovery and matching framework. Section IV presents a similarity based matching algorithm for VN discovery. The implementation and evaluation of the matching algorithm based on conceptual clustering is reported in section V.

II. VIRTUAL RESOURCE DESCRIPTION SCHEMA

Infrastructure providers will have to describe the virtual resources and services they offer to inform VN Providers on the virtual resources basic properties but also on the degree of openness, flexibility, configurability and programmability of the offered resources. Infrastructure providers would also notify users of updates and evolution of the available virtual resources. Providing these descriptions and features is essential to build flexible and diversified future networks.

Virtual resources need to be described in terms of properties and functionalities much like services and devices/nodes are described in existing service architectures. In the most relevant state of the art, that of the description of computer networks, resources are described using a number of well known network description specifications such as cNIS [4], NDL [5], and NM/PerfSONAR [6]. The NML Working Group [7] proposed recently an extensible network description schema called the Network Mark-up Language (NML) for possible standardization within the OGF (Open Grid Forum). The objective of the NML schema is to provide an abstraction layer for inter-domain and hybrid networks, and to define an information model for service discovery and provisioning.

These existing specifications do not, however, provide schemas for describing virtual resources and virtualization services at the levels required for automated virtual network provisioning. The NML specification only includes a primitive definition of virtual links and nodes but does not provide finalised schemas and properties of virtual resources. The motivation of this work is to integrate in the description schema the properties and relations between virtual resources.

Figure 2 depicts a diagram representing the proposed virtual resource description schema. As shown in this diagram, a *Network Element*, seen as a basic building component, can be any of the following: *Node*, *Link*, *Interface*, or *Path*. Each Network Element has an identifier (or name), an availability parameter, and *functional* as well as *non-functional* attributes. Functional attributes define characteristics, properties and

functions of the network element (such as node/link type, execution environment, virtualization tool, OS, network stack, etc). Non-Functional attributes specify criteria and constraints related to the network element including performance, capacity, location, cost/price, QoS, etc.

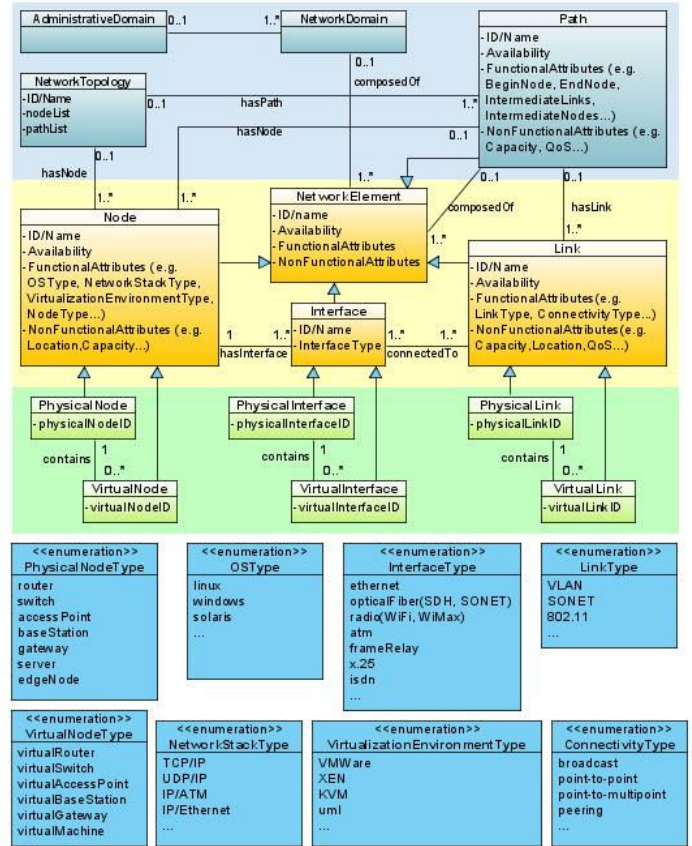


Figure 2. The proposed virtual resource description schema

Each Node in figure 2 has two sub-classes: virtual node and physical node. A physical node may contain one or multiple virtual nodes (e.g. virtual routers, virtual switches, virtual base stations). Each physical/virtual node has one or multiple physical/virtual interfaces. The interface types include, for example, Ethernet_Interface, ATM_Interface, Radio_Interface (e.g. WiFi, WiMax) and OpticalFiber_Interface (SDH, SONET). One or multiple physical/virtual interfaces may be connected to one or multiple physical/virtual links. A physical link may support one or multiple virtual links (by using layer 1 and/or layer 2 virtualization). Each link has two additional characteristics: connectivity type (broadcast, point-to-point, point-to-multipoint) and QoS related parameters (such as bandwidth, delay, jitter, packet loss).

The enumeration boxes in the lower part of figure 2 provide attributes and their associated values. For instance, nodes and links used to establish inter-provider peering will have attributes like *NodeType=edgeNode*, *LinkType=peering*, etc. As depicted at the top of figure 2, the proposed schema defines an abstract class called *Network Domain*. It represents a group of Network Elements forming a specific network domain (e.g. an Autonomous System). Each Network Domain is managed by one *Administrative Domain* (e.g. infrastructure provider,

VN provider). A *Path* is a group of *Network Elements* which can be composed of a single link or a sequence of links interconnected via intermediate nodes. A path can also be seen as a subclass of Network Elements. The functional attributes of each path include the links and nodes involved in the path establishment (e.g. begin/end nodes, intermediate nodes/links). The composition of a path (i.e. the set of intermediate nodes and links) is completely transparent to the VN provider which only sees the path endpoints. The *NetworkTopology* class represents the topology, the graph of connected network elements including all virtual and physical nodes and paths. Here we make no assumptions on the visibility of the physical resources given to the virtual network providers. Infrastructure providers actually decide how much information is disclosed to other parties on a case by case and SLA basis.

In this work, the extensible and standard XML schema has been used to express the proposed virtual resource description specification. This specification provides a common description framework for all infrastructure providers and ensures interoperability between their resource descriptions. Each node description includes the descriptions of all links directly connected to its network interfaces for completeness and simplicity reasons.

VN Query

A VN query schema/language is needed to specify the VN requests received by VN providers from VN users. The VN request includes typically the required nodes and links descriptions to be discovered and matched in the substrate. VN providers analyze and reformulate the VN queries into this format, syntax and structure to comply with the proposed description schema. The goal is to ensure reliable matching between the required resource descriptions (specified in the request) and the available resource descriptions (provided by the infrastructure provider).

III. HIERARCHICAL CONCEPTUAL CLUSTERING

This section addresses resource discovery and hierarchical conceptual clustering to facilitate VN provisioning.

A. Virtual Resource Discovery and Matching

Candidate discovery consists in finding a set of available virtual resources that comply with requirements specified by the VN request. Infrastructure providers advertise and register the virtual resource descriptions in the discovery framework using an advertisement process (e.g. publish/subscribe mechanisms). VN providers use this framework to discover and match available resources using VN requests. As shown in figure 3, resource discovery typically operates as a distributed set of repositories that will hold the physical/virtual resource descriptions. These distributed repositories, typically managed by the infrastructure providers, may act as a separate or shared secure overlay running on top of substrate networks.

As mentioned in section II, two types of attributes are defined: functional and non-functional attributes. The functional attributes include typically static parameters like node/link type, operating system, virtualization platform, geographical zone, some service guarantees (e.g. a minimum capacity for a link) or fixed link capacity (i.e. defined during

link virtualization), etc. In this work, only functional attributes are advertised in the resource discovery framework (see figure 3, step 0). The non-functional attributes, which include real-time parameters (e.g. processing power, memory, actual capacity, location, etc), will be registered into local repositories located in the substrate nodes (step 1). Indeed, advertising non-functional attributes in the discovery framework should be avoided since maintaining up-to-date real-time parameters requires a heavy monitoring process between the discovery framework and the substrate nodes. The discovery process will match the VN queries with the resource descriptions based only on the functional attributes advertised in the repositories. The non-functional attributes advertised in the local repositories will be considered only during the selection and binding processes.

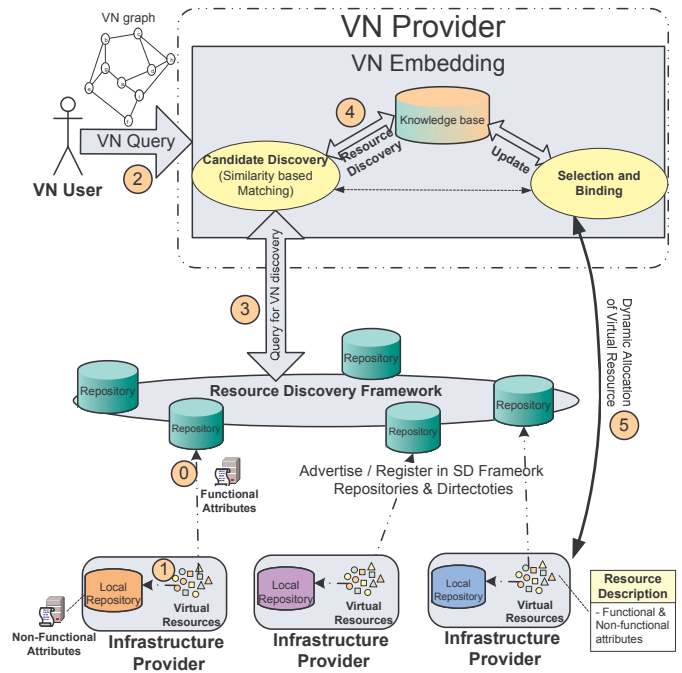


Figure 3. Allocation of virtual resources from substrate networks

B. Hierarchical Conceptual Clustering

Partitioning and organizing heterogeneous resource descriptions maintained by the repositories is crucial for reducing latency and overhead during the matching process. This paper proposes the use of similarity based clustering approaches to achieve this objective by grouping virtual resources with similar descriptions and properties into clusters. Resources grouped in clusters share common concepts, properties and capabilities. In particular, this work uses a conceptual clustering technique to classify virtual resources based on a concept description per cluster. Such clustering generates a hierarchy of clusters along with conceptual descriptions of each cluster. Providing meaningful descriptions of the clusters facilitates an understanding of the structure and organization of the virtual resources and simplifies the discovery process.

To produce the conceptual clustering of virtual resources, we propose the use of *Clustering Engines* located in the discovery framework. These engines are responsible for: (1)

collecting the functional attributes from the resource descriptions advertised in the repositories, (2) classifying these attributes into clusters based on their similarities, (3) generating a concept description for each cluster, and finally (4) producing a tree, called dendrogram, used to illustrate the organization of the clusters produced by a conceptual clustering algorithm. The dendrogram represents one possible structure to classify resources. In this work, we used the conceptual clustering algorithm *CLUSTER3* [13] to produce dendrograms for virtual resources.

The repositories maintain the virtual resource descriptions as well as the dendrograms representing the hierarchical classification of resources. An example of a dendrogram generation and a step by step scenario describing how a dendrogram can simplify the VN discovery and matching process is presented in the following section to illustrate the overall process.

Example of a dendrogram representation

Figure 4 provides a simple example of a dendrogram representation of ten virtual nodes and their associated links. To facilitate the dendrogram representation, we assumed that each virtual resource is described using only five attributes: VNt (VNtype), VE (Virtualization environment), OS (Operating System), NS (Network Stack), and VLt (LinkType). Each attribute has an assigned domain (DOM) that specifies all possible values the attribute can take:

- $DOM(VNt) = \{VSwitch, VRouter, VBasestation\};$
- $DOM(VE) = \{XEN, VMware, xVM\};$
- $DOM(OS) = \{Win, Linux, Solaris, Cisco, Juniper\};$
- $DOM(NS) = \{IP, ATM, FR, Eth\}.$
- $DOM(VLt) = \{VLAN, SONET, 802.11, WiMax\}$

As illustrated in figure 4, the conceptual clustering algorithm groups elements (i.e. virtual resources) into clusters that are concisely circumscribed by conjunctive statements. A conjunctive statement is an expression, typically defined in the variable valued logic system [12], which describes all elements in the cluster. For instance, the conjunctive statement of *cluster2* (figure 4) is as follows: $Cluster2 = [VN=Router]$.

To produce a clustering hierarchy (i.e. dendrogram), this algorithm is applied recursively to each cluster while generating at each step conceptual descriptions (in the form of conjunctions) of the clustering results. For example, the conjunctive statements of *cluster2.2* and *cluster2.2.1* are provided as follows:

- $Cluster2.2 = [VN=Router] \& [VE=VMware]$
- $Cluster2.2.1 = [VN=Router] \& [VE=VMware] \& [OS=Linux] \& [NS=IP] \& [VLt=VLAN \vee SONET]$

Cluster 2.2.1 includes two virtual nodes N2 and N8. In the conjunctive statement, the relational operators $\&$ and \vee denote respectively the AND and OR logical operators.

The conceptual clustering algorithm will provide accordingly recursive levels of generalizations of the virtual resources. At the bottom of figure 4, the conjunctive statements N0, N1...N9 represent the conceptual descriptions of the ten virtual nodes and their associated links.

Transparency and level of abstractions

Depending on the level of detail of the information that the infrastructure providers wish to disclose to virtual network providers, the dendrograms can provide extensive, rich or rather limited information about the available virtual resources. The infrastructure providers can mask as much detail about their infrastructure as they wish, and make the functional attributes either transparent or invisible to the VN providers. Consequently, the VN providers are limited to the information disclosed by the infrastructure providers and can therefore only achieve discovery and matching within the limits imposed by the infrastructure providers.



- N0: $[VNt=Switch] \& [VE=XEN] \& [OS=cisco] \& [NS=ATM] \& [VLt=SONET]$
- N1: $[VNt=Base\ station] \& [VE=VMWare] \& [NS=IP] \& [VLt=WiMax]$
- N2: $[VNt=Router] \& [VE=VMWare] \& [OS=Linux] \& [NS=IP] \& [VLt=SONET]$
- N3: $[VNt=Switch] \& [VE=XEN] \& [OS=cisco] \& [NS=FR] \& [VLt=VLAN]$
- N4: $[VNt=Router] \& [VE=VMWare] \& [OS=win] \& [NS=eth] \& [VLt=SONET]$
- N5: $[VNt=Router] \& [VE=xVM] \& [OS=Solaris] \& [NS=IP] \& [VLt=SONET]$
- N6: $[VNt=Switch] \& [VE=VMWare] \& [OS=juniper] \& [NS=ATM] \& [VLt=SONET]$
- N7: $[VNt=Base\ station] \& [VE=XEN] \& [NS=IP] \& [VLt=802.11]$
- N8: $[VNt=Router] \& [VE=VMWare] \& [OS=Linux] \& [NS=IP] \& [VLt=VLAN \vee SONET]$
- N9: $[VNt=Router] \& [VE=xVM] \& [OS=Solaris] \& [NS=IP] \& [VLt=VLAN]$

Figure 4. Example of a conceptual clustering hierarchy

To realize information hiding in order to control the amount of information that is disclosed by infrastructure providers, two possibilities are discussed in this paper. One possibility is to use the *NetworkDomain* class as an abstraction (i.e. the provider only discloses a set of *NetworkDomain* objects to the outside world without the *NetworkElement* objects it contains). This means, however, that *NetworkDomain* objects must provide sufficiently meaningful and suitable attributes for the VNet providers (e.g. specifying external interfaces that can be used for peering with other *NetworkDomains* and some high-level properties such as domain-wide QoS features). The alternative consists in specifying rules and policies for the advertisement process to control and manage the information (i.e. attributes, values, relationships) to be advertised in the discovery framework. Based on these rules, the advertisement process can tag some objects as transparent or invisible during the registration phase.

Knowledge base

Rather than querying the repositories each time a request is received, the VN providers can also maintain a history of these

dendrograms in their own knowledge base. This can reduce load and latency during the overall discovery process. The VN providers can also generate their own views or dendrograms as they discover, acquire and assess virtual resources. In this case, VN providers retrieve first the resource descriptions from the discovery framework and then use a clustering engine to generate the dendrogram to store the results in their own knowledge base.

C. Similarity based matching using dendrograms

This section provides a step-by-step scenario describing the VN discovery and mapping process (see figure 3). Upon receiving a VN request from users (step 2), the VN providers can either directly query the resource discovery framework (step 3), or use their own knowledge database (step 4) to discover virtual resources matching the VN requests. A Matching Algorithm proposed for the discovery and matching process is responsible for:

1. Extracting the required resource descriptions from VN requests (as input 1)
2. Retrieving from repositories the dendrogram describing the available resources (as input 2)
3. Matching the required resource descriptions with the concept descriptions of clusters, i.e. associating the required resources with suitable clusters in the dendrogram based on similarities. The matching algorithm operates as follows:

Similarity based matching algorithm

1. Start at root node of the dendrogram
2. If the root description does not satisfy the required description, the request is rejected
3. Else continue the matching process for children nodes
 - a. If no child description matches the requested description, stop the matching process and return to the last matched description
 - b. Else continue matching process for child sub tree
4. Go to step 3

The algorithm begins by examining the root of the tree (dendrogram). Next, the algorithm uses a descent procedure in the tree which consists in matching the requested resource description with the child descriptions: If the root description does not comply with the required description, the request will be rejected since there will be no descriptions in the dendrogram that would match the request. If the root description satisfies the required description, the matching is successful and will continue for children nodes. If there is a child that matches the required description, the algorithm pursues the search in the indicated child sub tree. These steps are repeated until the indicated child sub tree is null or there are no more child descriptions that match the requested description. Finally, the algorithm returns the most appropriate resource description from the cluster that matches best the required resource description.

Resource Matching Example

Assuming that a given VN request is composed of two virtual nodes *vnodel* and *vnodel2* interconnected via a virtual link *vlink1*:

- *vnodel*: $[VNt=Router] \& [VE=VMWare] \& [OS=Linux] \& [NS=IP] \& [VLT=SONET]$
- *vnodel2*: $[VNt=Switch] \& [VE=VMWare] \& [OS=Juniper] \& [NS=ATM] \& [VLT=SONET]$
- *vlink1*: $[VLT=SONET]$

The *vnodel* and *vnodel2* represent the required resource descriptions (as input 1). The matching algorithm will take for instance the dendrogram depicted in figure 4 (as input 2) and will try to find a match between the required resource descriptions and the clusters' concept descriptions. As described above, the goal of this algorithm is to search and find for each required resource description a suitable cluster in the dendrogram so that the concept description of this cluster includes the required resource description. As a result, the output of the matching algorithm will be as follows: *vnodel* belongs to *cluster2.2.1*. The nodes *N2* and *N8* are two candidates capable of supporting *vnodel*. The *vnodel2* will belong to cluster *N3* which includes only one virtual node.

D. Resource Selection and Binding

The VN provider is responsible for discovering and matching resources (in this paper using the dendrogram announced by the infrastructure provider). Using the matching algorithm, each required resource specified in the VN request will be associated to a cluster in the dendrogram. The set of available resources that constitute this cluster will represent the potential candidates capable of supporting the required resource. The Infrastructure Providers will select and identify candidate virtual resources to suggest to the VN provider. A selection process chooses the best resource (among the cluster elements) to map the required resources. Optimization algorithms as described in [8-11] can be used to find the optimal mapping according to multiple constraints and objectives (e.g. reducing cost). The non-functional attributes advertised in the local repositories are considered during this selection phase since the dynamic and real time aspect of such attributes can affect the overall optimization and selection process.

Once the best possible candidates have been determined by the selection process and accepted by the VN provider, the infrastructure provider allocates and reserves the virtual resources (also known as binding) to set up the selected VN candidates specified in the VN requests (step 5 in figure 3).

IV. PERFORMANCE EVALUATION

The objective of this section is to evaluate the time delay required by the matching algorithm to discover candidate resources (among *N* possible resources) that satisfy the required resource descriptions. The matching algorithm has been implemented using a tree data structure. The algorithm operates *N* XML schema descriptions representing the virtual nodes. The *N* virtual nodes are classified into a hierarchy of clusters represented by a dendrogram (a binary tree; i.e. each node has at most two children). An experiment has been set up to assess scalability of the developed matching algorithm by evaluating the effect of using different representations (linear or hierarchical with different tree heights) on the algorithm performance. A mono-processor PC (Pentium M, 1.6 GHz with 512 Mo of RAM) was used for this experiment.

Figure 5 depicts three time delay curves representing the time delay required by the matching algorithm to discover a required resource in linear and hierarchical structures. The lower curve represents the time delay needed to discover resources in a dendrogram with a balanced tree structure (i.e. height H of the dendrogram is equal to $O(\log(N))$). The middle curve represents the time delay required to discover resources in a dendrogram with an unbalanced tree structure (i.e. $H = O(N)$). The time delay needed to discover resources in the worst case ($H = O(N)$) is important compared to the average case ($H = O(\log(N))$). When the dendrogram is a binary tree, the time complexity of the matching algorithm is $O(H)$. Even in the worst case (i.e. the middle curve) where the height of the dendrogram is $O(N)$, the performance results show that using a hierarchical dendrogram can reduce time delay compared to non-hierarchical (linear) structures (i.e. the upper curve). The scalability improvement achieved by using a dendrogram is significant, compared to the linear case, when the number of virtual nodes increases.

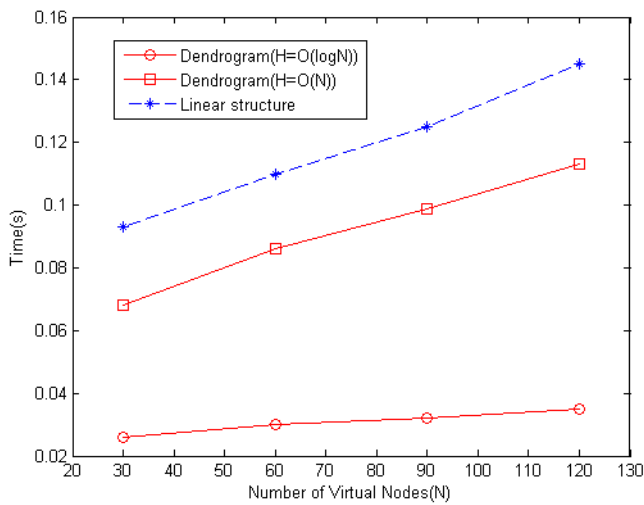


Figure 5. Resource discovery delay performance of matching algorithms

V. CONCLUSION

This paper explores how resource description, discovery and matching can ease Virtual Network embedding in virtualization frameworks. An initial resource description schema is presented to specify virtual resources offered by infrastructure providers. A conceptual clustering technique is also proposed to classify resources with similar characteristics into a hierarchy of clusters and to provide meaningful descriptions of these clusters. The hierarchical representation (in the form of a dendrogram) facilitates resource discovery. A similarity based matching algorithm to discover available resources maintained by the dendrogram classification is presented and evaluated. Dendrogram structures for representation of clustered resources and matching are shown to improve scalability.

Future work will consist of implementing and evaluating more thoroughly the overall mapping process especially for highly dynamic environments.

ACKNOWLEDGMENT

This work was supported in part by the European Union through the 4WARD project in the 7th Framework Programme. The views expressed in this paper are solely those of the authors and do not necessarily represent the views of their employers or the 4WARD project. The authors would like to thank Mr. Houssemed MEDHIOUB for his valuable contribution to the proposed resource description schema.

REFERENCES

- [1] The GENI project, <http://www.geni.net/>
- [2] The 4WARD project, <http://www.4ward-project.eu/>
- [3] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *IEEE Computer Magazine*, vol. 38, no. 4, pp. 34–41, 2005.
- [4] GÉANT2 common Network Information Service Schema Specification, <http://www.geant2.net>
- [5] NDL web page, <http://www.science.uva.nl/research/sne/ndl>
- [6] NM-WG, <http://nmwg.internet2.edu/>
- [7] NML-WG, http://www.ogf.org/gf/group_info/view.php?group=nml-wg
- [8] I. Houidi, W. Louati, and D. Zeglache, "A Distributed Virtual Network Mapping Algorithm," In *Proceedings of IEEE ICC 2008*, May 19-23, 2008, Beijing, China, pp 5634 - 5640.
- [9] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proceedings of IEEE INFOCOM*, 2006.
- [10] J. Lu and J. Turner, "Efficient mapping of virtual networks onto a shared substrate," Washington University, Tech. Rep. WUCSE-2006-35, 2006.
- [11] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM Com. Communication*, vol. 38, no. 2, pp. 17-29, April 2008.
- [12] R. S. Michalski, and R. Stepp, "Automated Construction of Classifications: Conceptual Clustering versus Numerical Taxonomy," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, No. 4, pp. 396 - 410, July, 1983.
- [13] W.D. Seeman, and R.S. Michalski, "The CLUSTER3 System for Goal-oriented Conceptual Clustering: Method and Preliminary Results," In *Proceedings of The Data Mining and Information Engineering 2006 Conference*, Prague, Czech Republic, July 11-13, 2006