

On the Optimal Allocation of Virtual Resources in Cloud Computing Networks

Chrysa Papagianni, Aris Leivadeas, Symeon Papavassiliou,
Vasilis Maglaris, Cristina Cervelló-Pastor, and Álvaro Monje

Abstract—Cloud computing builds upon advances on virtualization and distributed computing to support cost-efficient usage of computing resources, emphasizing on resource scalability and on demand services. Moving away from traditional data-center oriented models, distributed clouds extend over a loosely coupled federated substrate, offering enhanced communication and computational services to target end-users with quality of service (QoS) requirements, as dictated by the future Internet vision. Toward facilitating the efficient realization of such networked computing environments, computing and networking resources need to be jointly treated and optimized. This requires delivery of user-driven sets of virtual resources, dynamically allocated to actual substrate resources within networked clouds, creating the need to revisit resource mapping algorithms and tailor them to a composite virtual resource mapping problem. In this paper, toward providing a unified resource allocation framework for networked clouds, we first formulate the optimal networked cloud mapping problem as a mixed integer programming (MIP) problem, indicating objectives related to cost efficiency of the resource mapping procedure, while abiding by user requests for QoS-aware virtual resources. We subsequently propose a method for the efficient mapping of resource requests onto a shared substrate interconnecting various islands of computing resources, and adopt a heuristic methodology to address the problem. The efficiency of the proposed approach is illustrated in a simulation/emulation environment, that allows for a flexible, structured, and comparative performance evaluation. We conclude by outlining a proof-of-concept realization of our proposed schema, mounted over the European future Internet test-bed FEDERICA, a resource virtualization platform augmented with network and computing facilities.

Index Terms—Federated infrastructures, resource allocation, resource mapping, virtualization, cloud computing, quality of service

1 INTRODUCTION

CLOUD computing promises reliable services delivered through next generation data centers that are built on compute and storage virtualization technologies. According to Buyya et al., [1] “a cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and the consumers” and accessible as a composable service via web 2.0 technologies.

Therefore, with respect to cloud computing there exist the “as a service” definitions, which include software as a service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS). Each of these has a very different business value proposition. However, despite of the model adopted and followed, ultimately the goal of cloud computing is to create a fluid pool of virtual resources across computers, servers, and data centers that enable users to access stored data and applications on an as-needed basis.

In distributed computing environments, up to 85 percent of computing capacity remains idle [2]. Cloud IaaS emerged as a solution providing immediate and on demand access to computing resources with significant cost savings for the users. The same need for cost efficiency however applies also to the cloud IaaS providers. Cloud providers try to take advantage of the cloud’s elastic service provisioning model, by utilizing solely the needed capacity to satisfy targeted end-user quality of service (QoS) at any given time. In such an environment, the probability of all embedded requests in the cloud utilizing their maximal capacity requirements simultaneously is low [3]. Therefore, the capacity of physical resources can be multiplexed among requested resources allowing us to accommodate more requests [4]. The term thin provisioning is used to differentiate from the established overprovisioning methodology that plans capacity for peak workloads [3].

Moreover, in future Internet (FI) vision, where Internet connection of objects and federation of infrastructures become of high importance, the cloud involves two different key players: cloud computing and networking. For many cloud computing applications, network performance will be the key to cloud computing performance and its subsequent adoption. QoS delivery in the cloud is intrinsically integrated with the network, its infrastructure, and capacity. Therefore, the convergence between cloud computing and networking is becoming more a requirement than a desire, motivating and driving the creation of networked cloud paradigm.

Toward facilitating the efficient realization of this emerging paradigm, traditional cloud computing resources and networking related resources need to be jointly treated

• C. Papagianni, A. Leivadeas, S. Papavassiliou, and V. Maglaris are with the Division of Communication, Electronic and Information Engineering, School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechniou 9, Athens 15780, Greece. E-mail: chrisap@noc.ntua.gr, [arisleiv,papavass,maglaris]@netmode.ntua.gr

• C. Cervelló-Pastor and A. Monje are with the Department of Telematics Engineering, Universitat Politècnica de Catalunya, Castelldefels 08860, Barcelona, Spain. E-mail: [cristina, alvaro.monje]@entel.upc.edu.

Manuscript received 22 Apr. 2011; revised 6 Dec. 2012; accepted 20 Jan. 2013; published online 14 Feb. 2013.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number tcsl-2011-04-0260. Digital Object Identifier no. 10.1109/TC.2013.31.

and optimized. Therefore, one needs to consider the dynamic provisioning, configuration, reconfiguration, and optimization of both computing resources (e.g., servers) as well as networking resources (e.g., bandwidth pipes enabling interconnectivity through network infrastructure operating either seamlessly or providing fully reconfigurable network abstractions via appropriate virtualization models). Moreover, functional and nonfunctional characteristics of these resources need to be also taken into account [6]. Functional parameters define characteristics and properties of computing/networking resources, for example, operating system, supported virtualization environment, and so on, while nonfunctional parameters specify criteria and constraints of the various resources, e.g., maximum number of interfaces for each node, maximum disk space, and so on. Application QoS in this networked cloud environment is directly linked to the provisioning model adopted for computational resources and networking performance. Networking performance related metrics can be further viewed as objectives that need to be optimized and/or constraints that need to be satisfied. For instance, one feasible way to reduce delay along a communication path is by minimizing transit “hops”.

1.1 Paper Contributions and Structure

Promoting a unified management and control framework for delivering efficiently cloud IaaS, we propose a method for efficient mapping of user requests for virtual resources (denoted as virtual network requests) onto a shared substrate interconnecting previously isolated islands of computing resources. The problem essentially amounts to solving optimally the real-time problem of mapping virtual resources to substrate resources with limited assets (e.g., of virtual nodes and virtual links also known as virtual network embedding (VNE) problem [7]).

Specifically, in our work toward optimizing the cloud, we study and formulate the corresponding VNE problem in the envisioned networked cloud computing environment (in the following we will refer to this problem as networked cloud mapping (NCM)). Following the methodology introduced in [7] and the above considered cloud service paradigm, our work aims to:

1. extend the pool of shared resources to a layer 2/3 network topology including heterogeneous network infrastructure possibly across multiple domains;
2. provide a generic formulation for the resource mapping problem at hand capable of taking into consideration QoS requirements;
3. support QoS provisioning of cloud IaaS;
4. design and implement an experimentation simulation environment that allows a flexible and structured evaluation of the performance and efficiency of the proposed approach; and finally
5. provide a proof of concept of the operational efficiency of the proposed approach via a prototype implementation of the framework on an FI experimentation platform, namely the FEDERICA [8].

The remainder of this paper is organized as follows. In Section 2, a brief review of the related work regarding VNE problem is provided. In Section 3, the formulation of the optimal NCM problem is presented, supporting QoS provisioning of Cloud IaaS, indicating objectives related to cost efficiency of the resource mapping procedure and

network performance. In addition, the adopted methodology to treat and solve this problem is summarized. Section 4 provides an overview of the experimentation environment implemented to test the efficiency of the proposed solution. Within this environment, in Section 5, the experiment setup is described along with numerical results and relevant discussions. In addition, in Section 6, the application of the proposed resource mapping methodology in a prototype implementation utilizing the FEDERICA FI experimentation testbed is demonstrated. Finally, Section 7 concludes the paper.

2 RELATED WORK

The problem of assigning interconnected virtual nodes to the substrate network with constraints on virtual nodes and virtual links, can be reduced to the NP-hard multiway separator problem [9]. Most of the proposed approaches decompose the problem into the node mapping phase and the link mapping phase, to reduce the overall complexity of the problem. Researchers usually employ some greedy heuristic approach for node mapping, while link mapping is performed using (k) shortest path or multicommodity flow algorithms (e.g., [10], [11]). Recent approaches tend to solve the two problems either simultaneously (e.g., [6]) or providing some type of coordination among the two phases (e.g., [7]). In the proposed study, we follow the latter approach. The two phases are correlated in the sense that the node mapping phase facilitates the link mapping phase.

Additional ways to deal with the complexity of the problem is to restrict the search space in one or more dimensions. For example, one may omit admission control by assuming infinite capacity on the substrate network [12], [13], [14] or ignore either node or link requirements (e.g., [12], [13]). In several cases (e.g., [11], [13], [14], [15]), VNE requests are not handled upon arrival (offline variants), while recently several researchers (e.g., [7], [16]) investigated methodologies for handling virtual network requests as they arrive. Additional constraints have been imposed on the problem, such as location constraints for virtual resources (e.g., [7], [17]). The online version of the problem is studied hereafter supporting admission control dictated by the limits of substrate resources in terms of capacity. Upon the arrival of a request, its topology is assigned to the substrate network with the target to achieve balanced load on both substrate nodes and links.

Static formulations of the VNE problem have been studied, where during the life-time of the request, no change is allowed in resource assignment (e.g., [13], [14]). In the proposed methodology, the resource mapping solution is also static in the sense that resource allocation does not change for the duration of the lease from the cloud provider. However, complementary to our approach, dynamic reconfiguration of virtual network mapping (links or links/nodes) based on current traffic conditions, could be used to improve the efficient utilization of substrate resources (e.g., [10], [12], [14]). Although the latter is not treated in this paper, at the end of Section 5, some discussion about potential reconfiguration strategies and emerging tradeoffs is provided.

VNE algorithms suffer from scalability issues and hence request partitioning has been studied for mapping each part of a request to a different part of the substrate network (e.g., [14], [18]). Distributed algorithms have been proven to

enhance overall network resiliency as well as surmount scalability limitations and delays imposed by maintaining information centrally (e.g., [15], [19]). Apart from heuristic algorithms, metaheuristics have been also applied for solving the VNE problem (e.g., [20], [21]). An overview and extensive literature review is available in [22] and [23].

All of the aforementioned approaches study the intra-domain VNE problem. However, recently the interdomain VNE problem has been also tackled with (e.g., [6], [24], [25], [26]) by considering virtual network provisioning across multiple infrastructure providers. In all cases substrate resources are restricted to computing resources and bandwidth pipes. Houidi et al in [27] attempt to deal with heterogeneous substrate resources, where a method for resource discovery and a clustering technique of the attributes of the physical resources is proposed, in order to facilitate association of virtual and physical resources. Following, a matching algorithm is applied trying to find the best match between virtual and physical resources. Optimization algorithms however have not been applied that would optimize mapping of resources according to multiple constraints and objectives.

The objective of our work is to deal with the NCM problem, by efficiently embedding requests for interconnected virtual resources into a pool of heterogeneous substrate resources with finite capacities in realtime, while QoS related parameters are also taken into consideration. Moreover, multiple nonfunctional attributes of those resources are considered.

3 PROBLEM FORMULATION AND SOLUTION

A networked cloud request is modeled as a weighted undirected graph denoted by $G^V = (N^V, E^V)$ where N^V represents the set of virtual nodes and E^V the set of virtual links. Similarly, the substrate network is modeled as a weighted undirected graph $G^S = (N^S, E^S)$. Every node is associated with a resource of type $a \in A$ (e.g., server, router, etc.) so that $n^X \in V_a^X \subseteq N^X$, $a \in A$, $X \in \{V, S\}$ and $A \cup V_a^X = N^X$, $X \in \{V, S\}$. Based on its type, node $n^X \in V_a^X \subseteq N^X$ is attributed with an explicit set I of nonfunctional attributes, denoted as capacities, $c_i(n^X)$, $i \in I$, $n^X \in V_a^X$, $X \in \{V, S\}$ (e.g., CPU capacity, memory, storage capacity, number per type of available network interfaces, etc.). The vector of capacities for each node is denoted as $\mathbf{c}(n^X)$. Moreover, every edge $(n^X, m^X) \in E^X$, $\forall n^X, m^X \in N^X$, $X \in \{V, S\}$, is associated with a link of bandwidth capacity $bw(n^X, m^X)$.

3.1 Networked Cloud Mapping

Resource mapping determines the allocation of physical resources (substrate nodes, links, and paths) to the networked cloud request. Resource allocation does not change for the duration of the lease from the cloud provider while substrate resources are released upon request expiration. Request mapping is comprised of node assignment and link assignment. Specifically, node assignment is denoted as:

$$M^N : N^V \rightarrow N^S$$

where $M^N(n^V) \in V_a^S$, $n^V \in V_a^V \subseteq N^V$.

Expressing anticolllocation constraints for the computational resources, each virtual node from the same networked cloud request must be assigned to a different substrate node. Therefore,

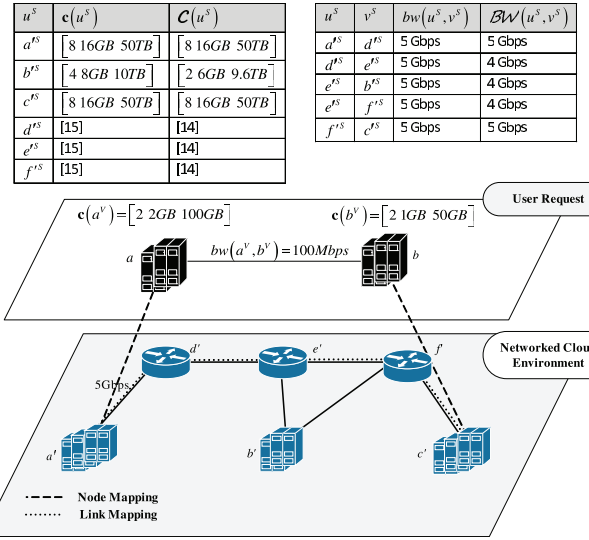


Fig. 1. Networked cloud environment and request mapping.

$$M^N(n^V) = M^N(m^V) \iff n^V \equiv m^V.$$

For a virtual node $n^V \in V_a^V$ to be mapped to substrate node $n^S \in V_a^S$ each requested capacity $i \in I$ must not exceed the remaining capacity \mathcal{C}_i of the substrate node n^S . That is,

$$c_i(n^V) \leq \mathcal{C}_i(n^S)$$

$$\mathcal{C}_i(n^S) = c_i(n^S) - \sum_{\substack{\forall m^V, \text{ where} \\ M^N(m^V) = n^S}} c_i(m^V).$$

On the other hand, every virtual link can be mapped to a single substrate path P^S for nonbifurcated routing or a set of substrate paths \mathbf{P}^S , when traffic is split in multiple routes among the mapping solutions of the virtual link end points. Under the assumption that flow bifurcation is enabled, that is traffic may be split among multiple paths, link mapping is denoted as

$$M^E : E^V \rightarrow \mathbf{P}^S$$

where $M^E(n^V, m^V) \in \mathbf{P}^S(M^N(n^V), M^N(m^V))$.

Similarly, the bandwidth capacity of the virtual link is subject to,

$$bw(n^V, m^V) \leq \sum_{\forall P^S \in M^E(n^V, m^V)} bw(P^S)$$

$$bw(P^S) = \min_{\forall (u^S, v^S) \in P^S} \mathcal{BW}(u^S, v^S)$$

$$\mathcal{BW}(u^S, v^S) = bw(u^S, v^S) - \sum_{\substack{\forall (j^V, k^V), (u^S, v^S) \\ \in M^E(j^V, k^V)}} bw(j^V, k^V).$$

The available bandwidth capacity of a substrate path is restricted to the available capacity \mathcal{BW} of the most loaded link in the path.

An example of a networked cloud request mapped to a substrate network is provided in Fig. 1. There are two types of nodes available (namely computational nodes and routers) so that $\{a^S, b^S, c^S\} \in V_1 \subseteq N^S$, $\{d^S, e^S, f^S\} \in V_2 \subseteq N^S$ and $\{a^V, b^V\} \in V_1 = N^V$. In the particular example, computational nodes are associated with {CPU cores, memory, disk space} while a router's capacity corresponds

to the number of logical router instances it can support. The values of nominal capacities and available resources for substrate nodes and links are provided in the preceding table within Fig. 1. Regarding the request, values over the virtual nodes and links represent computational and bandwidth requirements. Moreover, resource mapping is provided schematically in the figure where virtual nodes a^V and b^V have been mapped to nodes a'^S and b'^S , respectively, and virtual link (a^V, b^V) to path $\{(a'^S, d'^S), (d'^S, e'^S), (e'^S, f'^S), (f'^S, c'^S)\}$.

3.2 Hard/Soft QoS on Computational Resources

A simple scheme regarding provisioning of computing resources is employed, to cater for both elastic and hard cloud provisioning models. Specifically, hard and soft QoS provisioning is adopted to facilitate capacity requirements imposed by users. Soft QoS provisioning implies that no guarantees can be provided on meeting high-workload demands for elastic cloud services. Instead a best-effort approach is followed, exploiting the fact that virtual resources exhibit time-varying resource demand patterns with bursts of high-demand periods, intermixed with low-utilization regions [5]. Only a percentage of requested resources capacity is reserved, necessary to facilitate the operation of a virtual machine as, e.g., in the case of VMware's vStorage thin provisioning scheme [28], enabling the colocation of more virtual machines in the same physical node via oversubscription of physical resources. On the other hand, hard QoS provisioning guarantees performance at peak workloads via explicit reservation of maximum requested resources capacity.

Therefore, computational capacity requirements are modeled as follows: a virtual node $n^V \in V_a^V$ can be mapped to substrate node $n^S \in V_a^S$ in the case that a percentage $\mathcal{P}_{c_i(n^V)}$ of the requested capacity $i \in I$, as defined by the resources provisioning scheme (hard or soft QoS provisioning), will not exceed the available capacity \mathcal{C}_i of the substrate node n^S . That is,

$$\mathcal{P}_{c_i(n^V)} c_i(n^V) \leq \mathcal{C}_i(n^S)$$

$$\mathcal{C}_i(n^S) = c_i(n^S) - \sum_{\forall m^V, M^N(m^V)=n^S} \mathcal{P}_{c_i(m^V)} c_i(m^V).$$

3.3 Mixed Integer Programming Formulation

As already mentioned in Section 2, the node and link mapping phase are not independent of each other. To correlate the node and link mapping phase, the methodology proposed in [7] is adopted, without however posing any location constraints on virtual nodes. Specifically, the substrate network graph is augmented with the virtual nodes of the request. Every newly added (virtual) node in the augmented substrate graph is connected to every substrate node with infinite bandwidth. Hence, the augmented undirected substrate graph is denoted as $G^{S'} = (N^{S'}, E^{S'})$ where $N^{S'} = N^S \cup N^V$ and $E^{S'} = E^S \cup \{(n^V, n^S) | \forall n^V \in N^V, \forall n^S \in N^S\}$.

Every virtual link $(n^V, m^V) \in E^V$ with bandwidth requirements $bw(n^V, m^V)$ is considered a commodity in the augmented substrate graph originated at the virtual node $n^V \in N^{S'} \setminus N^S$ and ending at the virtual node $m^V \in N^{S'} \setminus N^S$. The resource allocation problem in the augmented substrate graph is formulated as a mixed integer programming (MIP)

$|E^V|$ -commodity flow problem, where the communication demands among the $N^{S'}$ nodes are specified as a $|N^{S'}| \times |N^{S'}|$ demand matrix. For the sake of simplicity superscripts V, S will be omitted in the following.

Variables

x_{uv}^{nm} : a binary variable set to 1 if there is traffic flow of the virtual link $(n, m) \in E^V$ routed via the augmented substrate link $(u, v) \in E^{S'}$.

f_{uv}^{nm} : the amount of traffic for the virtual link $(n, m) \in E^V$ routed over the link $(u, v) \in E^{S'}$ from u to v .

Objective

$$\begin{aligned} \min \quad & \sum_{uv \in E^S} \sum_{nm \in E^V} C_{uv} f_{uv}^{nm} \\ & + \sum_{a \in A} \sum_{nm \in E^V} \sum_{w \in V_a^S \subseteq N^S} \sum_{p \in V_a^V \subseteq N^{S'} \setminus N^S} D_w x_{pw}^{nm} \sum_{i \in I} c_i(p) \\ & + \sum_{w \in E^S} \sum_{nm \in E^V} C_{wv} x_{wv}^{nm}. \end{aligned} \quad (1)$$

Constraints

$$f_{uv}^{nm} \geq 0, \quad \forall u, v \in N^{S'}, \forall (n, m) \in E^V \quad (2)$$

$$x_{uv}^{nm} \in \{0, 1\}, \quad \forall u, v \in N^{S'}, \forall (nm) \in E^V \quad (3)$$

$$\begin{aligned} \sum_{w \in N^{S'}} f_{uw}^{nm} - \sum_{w \in N^{S'}} f_{wu}^{nm} &= 0, \quad \forall (n, m) \in E^V, \forall u \in N^{S'} \setminus \{n, m\} \\ \sum_{w \in N^{S'}} f_{nw}^{nm} - \sum_{w \in N^{S'}} f_{wn}^{nm} &= b(n, m), \quad \forall (n, m) \in E^V, n \in N^{S'} \\ \sum_{w \in N^{S'}} f_{mw}^{nm} - \sum_{w \in N^{S'}} f_{wm}^{nm} &= -b(n, m), \quad \forall (n, m) \in E^V, m \in N^{S'} \end{aligned} \quad (4)$$

$$\mathcal{P}_{c_i(p)} c_i(p) x_{pw}^{nm} \leq \mathcal{C}_i(w), \quad \forall p \in V_a^V \subseteq N^{S'} \setminus N^S, \quad \forall w \in V_a^S \subseteq N^S, \forall i \in I, \forall a \in A, \forall (n, m) \in E^V \quad (5)$$

$$(f_{uv}^{nm} + f_{vu}^{nm}) \leq \mathcal{BW}(u, v) x_{uv}^{nm} \quad \forall u, v \in N^{S'}, \forall (n, m) \in E^V \quad (6)$$

$$\sum_{nm \in E^V} (f_{uv}^{nm} + f_{vu}^{nm}) \leq \mathcal{BW}(u, v) \quad \forall u, v \in N^{S'} \quad (7)$$

$$\sum_{p \in V_a^V \subseteq N^{S'} \setminus N^S} x_{pw}^{nm} \leq 1, \quad \forall w \in V_a^S \subseteq N^S, \forall nm \in E^V, \forall a \quad (8)$$

$$\sum_{w \in V_a^S \subseteq N^S} x_{pw}^{nm} = 0, \quad \forall p \in V_a^V \subseteq N^{S'} \setminus N^S, \forall nm \in E^V, \forall a, a' \neq a \quad (9)$$

$$\sum_{w \in V_a^S \subseteq N^S} x_{pw}^{nm} = 1, \quad \forall p \in V_a^V \subseteq N^{S'} \setminus N^S, \forall nm \in E^V, \forall a. \quad (10)$$

$$x_{uv}^{nm} = x_{vu}^{nm}, \quad \forall u, v \in N^{S'}, \forall (n, m) \in E^V \quad (11)$$

$$x_{uv}^{nm} = x_{uv}^{nk} = x_{uv}^{lm}, \quad \forall u, v \in N^{S'} \setminus N^V, k, l \in N^{S'} \setminus N^S, (n, m) \in E^V \quad (12)$$

$$x_{uv}^{nm} \leq \lceil f_{uv}^{nm} + f_{vu}^{nm} \rceil, \quad \forall u, v \in N^{S'}, \forall (n, m) \in E^V. \quad (13)$$

1. The goal of the minimization problem (objective 1) is twofold;
 - a. *To minimize the cost of mapping the request into the substrate, as provided by the first two summation terms.* The cost of embedding a networked cloud request corresponds to the sum of substrate resources allocated to that request. Specifically, the first terms reflect the total amount of bandwidth allocated on substrate links that are parts of the substrate paths mapped to the requested virtual links. The second terms correspond to the total amount of computational resources that are allocated to the physical servers mapped to requested virtual nodes. Each of these terms multiplied by the corresponding monetary factor can provide the cost of embedding the particular request to the cloud provider resources. Weights C_{uv} and D_w can be adjusted to balance the load on the substrate links and nodes, respectively. As an example, in [7] the weights C_{uv} and D_w have been set equal to the inverse values of the available bandwidth of the link and the specific node-type available capacity, respectively. In this way, it is ensured that less loaded resources are preferred over more loaded ones. The same approach is followed in this study.
 - b. *To minimize the overall number of hops for a virtual link mapped on a substrate path, according to the third summation term.* The rationale behind it is that reducing the number of hops along a traffic route between two communication nodes is a common practice to provide QoS enhancements with regards to latency. Moreover, since the objective function expresses essentially a bi-objective problem, an appropriately defined weight has been also included in the third term of the summations to control the impact of the specific objective on the optimization process. In the particular case, it has been set equal to C_{uv} to associate the length of the substrate path mapped to the virtual link (m, n) and available capacity of links included in the path, since both are implicitly related to latency.
2. Constraints (2) and (3) provide the domain constraints of the two vector variables.
3. Flow conservation is guaranteed via constraints set (4).
4. Constraints set (5) ensures that the requested capacity $i \in I$ for a virtual node of type $a \in A$ that is mapped to substrate node $w \in V_a^S \subseteq N^S$ does not exceed the substrate node's available capacity.
5. Constraints set (6) and (7) ensure that the sum of all virtual flows that are routed through the substrate link (u, v) does not exceed its available bandwidth capacity.
6. Constraints sets (8) and (9) are adopted to ensure that at most one virtual node is linked to a substrate node of the same type $a \in A$.
7. Constraints set (10) assures that only one substrate node is selected for each virtual node.

8. Constraints set (11) and (13) ensure that the binary variable x_{uv}^{nm} is set whenever there is traffic routed of virtual link (n, m) over the substrate link (u, v) regardless of the direction. Constraints set (12) guarantees that the resource mapping solution is a connected graph.

3.4 Solution

In the previous sections, we have presented the NCM problem from a pool of heterogeneous physical resources with the goal of minimizing the mapping cost and the overall number of hops for every virtual link mapped to a substrate path. The problem is formulated as a MIP problem. MIPs provide a flexible and mathematically precise way of formulating many real-world problems. Such problems are known to be NP-Hard [29], and thus, large-scale instances and models exhibiting a high degree of symmetry are often computationally intractable. Integer programming is a commonly used technique for resource allocation and scheduling in wired and wireless networks. The main two problem types that MIP addresses in this field are: 1) network synthesis and 2) resource assignment problems [30]. However, due to the computational intractability of MIP models, efficient heuristics, problem preprocessing, and reformulations are proposed in the literature such as Lagrangian relaxation, problem decomposition, branch and bound and linear programming relaxation.

To solve this problem, the following methodology is applied. Specifically, the request is mapped to the networked cloud in two phases: 1) solving the flow allocation problem as was described in the previous section that results in substrate node mapping; and 2) allocating virtual links to the substrate.

Node Mapping Phase

The flow allocation problem as was described in the previous section is solved, taking into consideration virtual links as demands. Due to the nature of the MIP problem presented, the optimal fractional solution is computed for the problem's linear programming relaxation of the integer variable x_{uv}^{nm} , which can provide a solution at least as good as the integer one. The relaxed problem can be solved by any suitable linear programming method, in polynomial time (e.g., CPLEX dual simplex routine). A rounding technique is applied to obtain the integer solution of the aforementioned relaxed MIP problem. Randomized rounding for LP relaxations was introduced by Raghavan and Thompson [31] for multicommodity routing problems, where the fractional values contained in the optimal LP solution were treated as probabilities.

The randomized rounding technique proposed in [7] is adopted, where the correlation between the linear variable f_{uv}^{nm} and the binary variable x_{uv}^{nm} , during the LP relaxation process is maintained. Specifically, the substrate node that maximizes the x_{uv}^{nm}, f_{uv}^{nm} product per virtual link is selected.

Link Mapping Phase

Once the aforementioned node mapping procedure has been successfully completed, link mapping is achieved by solving the multicommodity flow allocation problem allowing traffic bifurcation [30]. Alternatively, a shortest path algorithm can be applied to restrict each flow to a single path.

4 EXPERIMENTATION ENVIRONMENT

A discrete event java-based simulator called simulator for controlling virtual infrastructures (CVI-Sim) was implemented to provide an extendable experimentation environment that will enable us to evaluate the performance of the proposed approach and the efficiency of the mapping solution. This generic purpose simulator can facilitate further research on the control and management plane of virtualized infrastructures. In this concept, CVI-Sim acts as an emulator of a resource mapping service, because it is designed to support importing actual resource specification files (e.g., PlanetLab RSpec [32]) to test substrate networks and networked cloud requests based on real-world deployments. Additionally, it allows for bulk random creation of requests for virtual resources, according to user selected probabilistic distributions on requests interarrival time and virtual resources lifetime as well as random generation of interconnected substrate resources. Finally, via an editing tool bench, graphic design of requests/substrates is also supported.

An extended resource set along with resource specific functional and nonfunctional parameters has been adopted in CVI-Sim, to allow easy evaluation of virtual resource mapping algorithms in heterogeneous virtual environments. Functional and nonfunctional parameters are grouped by type of resource; e.g., computing node parameters include operating system, supported virtualization environment, supported network stack, CPU computational capacity, available memory, total disk space, maximum number of interfaces, maximum number of supported VMs. Moreover a third set of simulation parameters has been defined to capture the need for request specific information; e.g., description of the request arrival process, distribution of the lifetime of a request, adoption of path splitting or unsplitable flows.

CVI-Sim is based on the JUNG software library [33] that provides a common and extendible language for the manipulation, analysis, and visualization of data that can be represented as a graph. The GUI of the simulator is implemented under JFC (Swing/AWT) framework [34]. The library CPLEX has been used to solve the relaxed MIP problem [35].

5 PERFORMANCE EVALUATION

In this section, the effectiveness of the proposed network cloud mapping problem formulation along with the performance of the proposed solution is evaluated via simulation. To better illustrate the efficiency and superior performance of the proposed NCM approach, we compare it against the corresponding performances of the following two well-known methods applied in the literature, in terms of a well-defined set of metrics.

- Greedy node mapping followed by a shortest path algorithm for the link mapping phase (G-SP) [14]
- Greedy node mapping followed by solving the multicommodity flow problem (G-MCF) [10].

5.1 Evaluation Setup and Metrics

Regarding substrate network creation, two types of nodes have been used (servers and routers) to provide a more

TABLE 1
Evaluation Metrics

Mapping Cost	$C(G^V) = \sum_{e^V \in E^V} \sum_{e^S \in E^S} f_{e^S}^{e^V} + \sum_{a=1}^A \sum_{n^V \in N_a^V} \sum_{i=1}^I c_i(n^V)$
Mapping Revenue	$R(G^V) = \sum_{e^V \in E^V} b(e^V) + \sum_{a=1}^A \sum_{n^V \in N_a^V} \sum_{i=1}^I c_i(n^V)$
Acceptance Ratio	The average percentage of accepted mapping requests
Hop Count	Average hop number per virtual link

realistic networked cloud environment. Each node is characterized from its type (server, router), its operating system (Windows, Linux, Android, Solaris, JUNOS, etc.), and its virtualization environment (Xen, VMware, KVM, JUNOS specific, etc.). Moreover, the nodes have different nonfunctional characteristics based on their type, e.g., CPU computational capacity, memory, storage for servers, and number of available logical router instances for routers. Matching the experimentation setup in [7], the available CPU capacity per server and available bandwidth capacity per substrate link are uniformly distributed in the interval [50-100]. These present unit values. Similarly, available storage and memory capacities are also defined as real numbers uniformly distributed between [50-100]. A maximum of 15 available logical routers can be instantiated on every physical router [36]. Substrate topologies are randomly generated as partial mesh topologies, while each substrate is comprised of 50 nodes. The probability of generating a specific type of node is 80 percent for servers and 20 percent for routers.

In the same fashion as in [7], the requested CPU capacity is uniformly distributed in the interval [0-20] for every requested virtual machine and requested bandwidth is uniformly distributed in the interval [0-50]. Requested storage and memory capacities are also defined as real numbers uniformly distributed between [0-20]. One logical router corresponds to each requested machine with routing capabilities. The number of virtual nodes per request is randomly selected by a uniform distribution between 2 and 10 with 50 percent connectivity, following similar setups in the literature [10], [14]. The probability of generating a specific type of node is 90 percent for VMs and 10 percent for virtual routers. The ratio between hard and soft QoS provisioning for networked cloud requests is set to 50 percent, while the percentage of resources capacity reserved $\mathcal{P}_{c_i(n^V)}$ is set to 50 percent.

NCM requests arrive according to a Poisson process with a varying rate (1 request per 100 time units to five requests per 100 time units; step 0.5). Each of them is assumed to have exponentially distributed lifetime with an average of 1,000 time units. Each simulation is executed for 1,000 requests and repeated for 10 iterations. The aforementioned experimentation setup is aligned with commonly used environments in the literature (e.g., [7], [10], [14]).

To quantify the performance of these approaches, we use the metrics presented in Table 1. The revenue metric is an indicator of the cloud provider's gain from accepting NCM requests, while the mapping cost reflects the corresponding cost for embedding a request and allocating substrate resources.

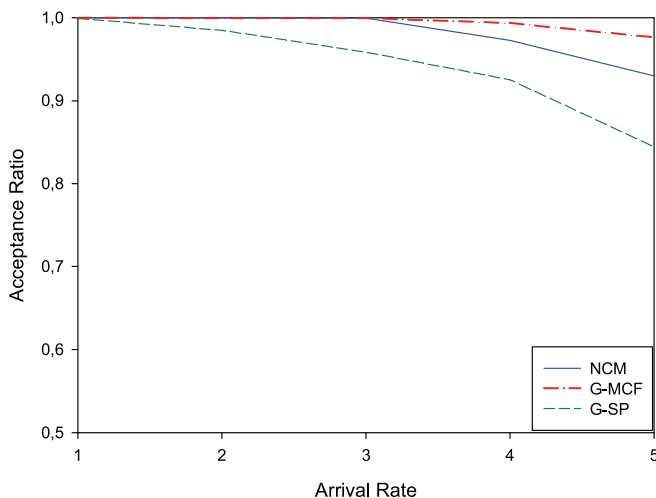


Fig. 2. Acceptance ratio.

5.2 Numerical Results and Comparison

Figs. 2 and 3 present the behavior of the three algorithms under consideration in this study, namely NCM, G-SP, and G-MCF, regarding the metrics of acceptance ratio and mapping revenue, respectively, as a function of increasing request arrival rate. As we can observe from those figures, NCM outperforms G-SP with regards to the cloud provider's revenue and number of networked cloud requests that are successfully embedded in the substrate. That effect is more pronounced as the request arrival rate increases leading to a more loaded substrate. On the other hand, NCM exhibits similar revenue and acceptance ratio to the G-MCF approach; both metrics are slightly increased for the G-MCF in higher arrival rates partially due to the fact that the algorithm tends to accept more requests with soft QoS resource provisioning (e.g., approximately 2 percent for arrival rate five requests per 100 time units).

Fig. 4 signifies that the NCM succeeds in reducing the number of hops along a traffic route between two communication nodes (i.e., according to the second objective in the problem formulation). Despite the fact that traffic bifurcation is allowed, NCM's behavior with regards to the particular metric is slightly enhanced in lower arrival rates

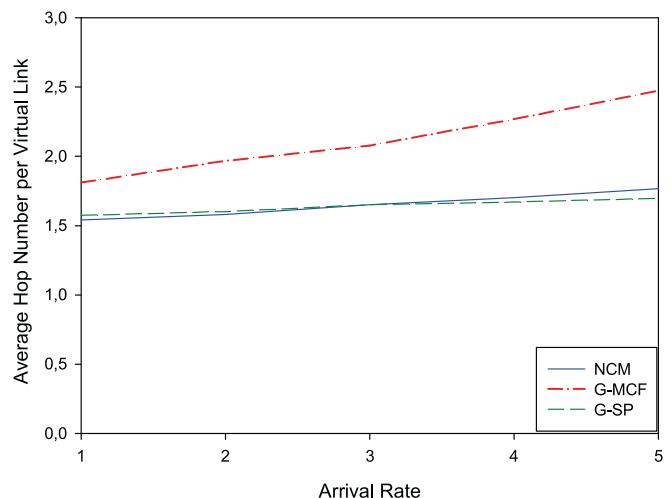


Fig. 4. Hop count.

compared to the G-SP, where the shortest substrate path between two nodes is always selected, due to more efficient node mapping in that terms. This decrease is also supported by the lower mapping cost of the NCM in comparison to the G-SP, as depicted in Fig. 5. On the other hand, as the request arrival rate increases, the average hop number per virtual link is slightly higher for the NCM because it maps successfully a larger number of requests than the G-SP. These two factors lead to a slight increase of the mapping cost for the NCM as depicted in Fig. 5.

Similarly, at low arrival rates the NCM and G-MCF exhibit similar behavior with regards to accepting incoming requests. Despite the fact that the G-MCF results in higher average hop number per virtual link, the mapping cost for the NCM is slightly higher (Fig. 5). The latter observation, along with the fact that revenues for the two algorithms are also similar, is an indicator that the proposed approach, due to the second objective in the problem formulation, tends to attain a smaller bifurcation ratio than the G-MCF. At higher arrival rates, the difference in cost increases slightly due to the difference in the number of accepted requests. The latter observation along with fact that the values of average hop number per virtual link are more diverse, is an indicator

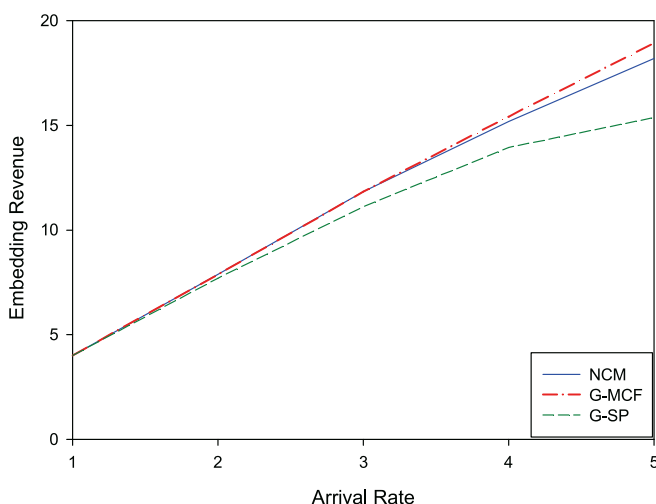


Fig. 3. Mapping revenue.

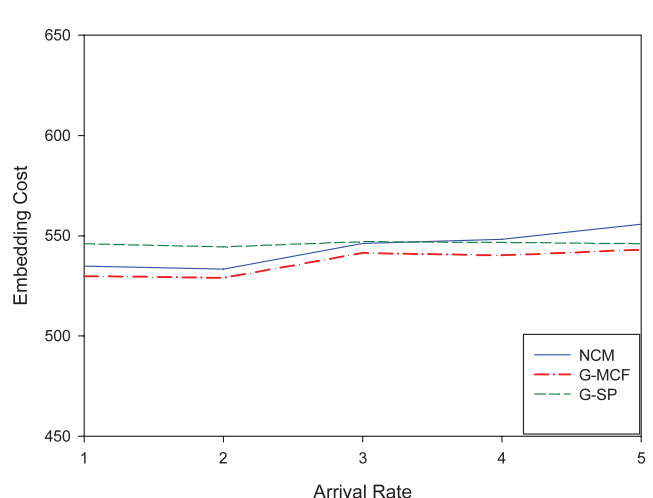


Fig. 5. Mapping cost.

TABLE 2
Resources Utilization

Algorithm	CPU(%)	Memory(%)	Link(%)
G-SP	54.3	55.9	31.7
NCM	57	56.2	32
G-MCF	58.8	61.7	35.5

that the proposed approach tends to embed requests that generate more revenue, instead of requests with a smaller node/link number that increase the acceptance ratio.

The key observation that the average hop number per virtual link (Fig. 4) is maintained at low values for the NCM approach, demonstrates the efficiency of the problem formulation and solution. This reduction of hops can be implicitly translated to improvement in the provided QoS parameters (e.g., delay). There is an obvious tradeoff between the number of hops on the adopted path(s) and the ability of accepting incoming requests, a fact witnessed by both G-SP and G-MCF (Fig. 2 and 4). The NCM on the other hand manages to overcome this tradeoff by exhibiting similar behavior to the G-MCF with regards to acceptance ratio and revenue while maintaining the number of hops on the adopted path(s) at similar levels with those achieved by the G-SP. Moreover, another key observation is that by controlling the bifurcation ratio for the NCM algorithm we can adjust the embedding cost, e.g., an increase in the bifurcation ratio would decrease the cost at the expense of a increase in the average hop number per virtual link.

It is noted that based on our evaluation all three algorithms present similar trend and comparable results regarding utilization of substrate resources, with NCM's performance positioned in between the corresponding performances of G-SP and G-MCF. Indicatively, server CPU, server memory, and link utilization, for an arrival rate of four requests per 100 time units is presented in Table 2. We should note that despite the fact that the number of accepted requests are very close for both NCM and G-MCF (2 percent difference), utilization is lower for NCM due to adopted load balancing feature.

The evaluation results demonstrate the benefits of the proposed approach for the cloud infrastructure provider and motivate its adoption in real systems. Specifically, it enables service differentiation (e.g., soft or hard QoS provisioning for capacity requirements, low delay substrate paths) in the sense that requests with different requirements can be served by providing the corresponding set of virtual resources at different cost. Furthermore, NCM tends to accept more requests for guaranteed resources (i.e., hard QoS), hence providing more revenue, assuming that an appropriate unit monetary cost per type of service is applied. In addition, it allows the provider to control the parameters of the algorithm and the corresponding substrate resources allocation process, to better meet its operational objectives. For example, adjusting the weight of the objective related to the hop number, the bifurcation ratio can be implicitly regulated. Fine tuning the tradeoff between the number of hops on the adopted path(s) and the ability of accepting incoming requests, enables each cloud provider to pursue their specific business goals.

5.3 Discussion on Virtual Topology Reconfiguration

As noted before, the proposed resource mapping solution is static in the sense that resource allocation does not change for the duration of the lease from the cloud provider. Although the load balancing feature of our approach can to some extent mitigate the potential effects of static request assignment since upon a request arrival, less loaded resources are preferred over more loaded ones, to further deal with the highly dynamic networked cloud environment, the proposed NCM algorithm could be complemented with the parallel application of an appropriate reconfiguration mechanism. Such a reconfiguration mechanism triggers node and path migration for the purpose of reallocating virtualized resources, according to a pre-specified reconfiguration policy.

Reconfiguration policies may vary depending on several operational parameters such as: 1) the degree of utilization of substrate resources which may result in existing service performance degradation or new service denial, 2) unbalanced allocation of substrate resources and/or resource consolidation, 3) implementation of service prioritization policies. The frequency of resources remapping along with the number of already assigned sets of virtual resources impacted by such a reconfiguration, determines to a large extent the involved migration overhead and corresponding cost. This cost includes among others, computational resources required for recomputing resource assignment, migration overhead and its impact on substrate networking resources, and possible service disruption cost.

In practice, a methodology similar to the one proposed in [14] could be adapted and executed periodically, where a threshold-based approach on the resource utilization could be employed for marking a resource for possible reconfiguration. Furthermore, instead of reconfiguring only the virtual topologies with marked resources, the residual lifetime of the requests could be also taken into account. The goal should be to avoid reconfiguring requests that are expected to expire soon. The cost of reconfiguration and migration of resources (i.e., [14]), as mentioned before, could reflect among others the expenses involved in recomputing the topology assignment (e.g., embedding cost), as well as the potential service disruption. Apart from periodical reconfiguration, in an alternative approach, the reconfiguration scheme could be activated upon request denial as means to decrease reconfiguration cost and complexity. A maximum value should be applied on the number of virtual topologies that are reconfigured on each occasion, because in a highly loaded networked cloud the number of reconfigurations could increase substantially.

6 APPLICATION OF THE PROPOSED FRAMEWORK ON A FI EXPERIMENTATION PLATFORM: A PROOF OF CONCEPT

A prototype of the proposed resource mapping algorithm has been implemented within networking innovations over virtualized infrastructures (NOVI) control framework [37], a real-time provisioning system of heterogeneous resources over federated shared infrastructures. NOVI is an experimentally driven research project under the FIRE initiative [38] with the goal to create a blueprint of FI federated

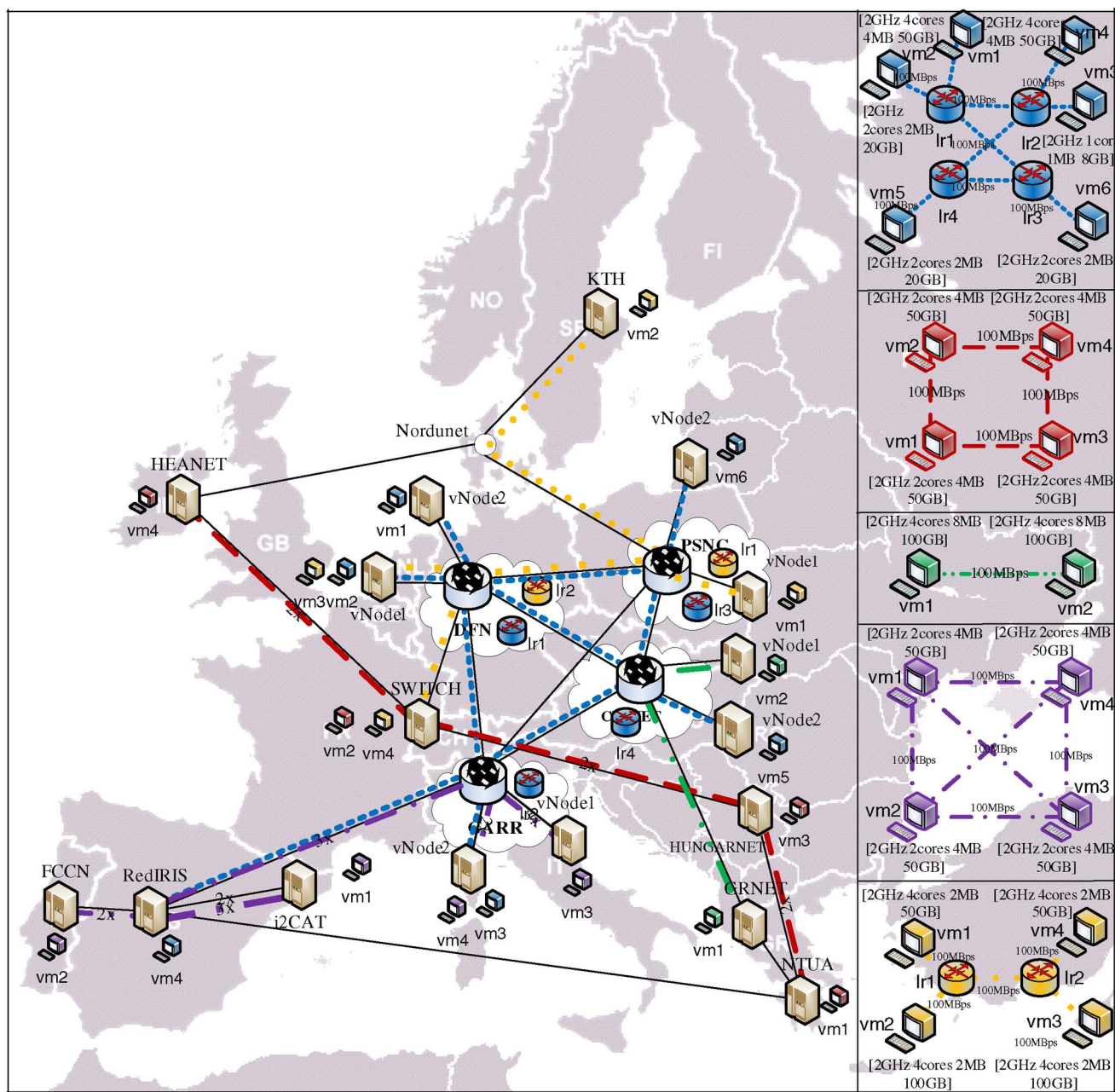


Fig. 6. NCM mapping on the FEDERICA substrate.

infrastructures. NOVI's aim is to address the issue of vertical federation by designing and prototyping a service portfolio based on combined virtualized facilities from shared infrastructures at different layers. NOVI control framework [39] is being deployed in a federated testbed including the FEDERICA [8] infrastructure. The latter is a resource virtualization platform, augmented with network and computing facilities hosted in European NREN Points of Presence.

In this section, the application and operation of the proposed resource mapping scheme over the FEDERICA experimental platform is demonstrated. The module responsible for resource mapping has been adapted to incorporate the embedding paradigm presented in the previous sections and enable allocation within the administrative domain of FEDERICA. The presented prototype

serves as a proof of concept regarding the efficiency of the embedding process.

To exhibit the functionality and operational differences of the two algorithms (NCM and G-SP) an illustrative example is presented in Fig. 6 and Fig. 7. Specifically, five networked cloud requests arrive in sequence for the FEDERICA substrate network, as depicted on each of the two figures. Only hard QoS provisioning is supported for these requests. The application of NCM results into mappings that reflect the objectives of the MIP problem presented in Section 3.3; that is, manages to balance the load across the entire substrate topology in comparison to the G-SP, reduces the average hop number per virtual link to 1.65 in comparison to the 2.28 of G-SP, while simultaneously the cost for embedding the requests is 1,072 for the NCM and 1,354 for the G-SP.

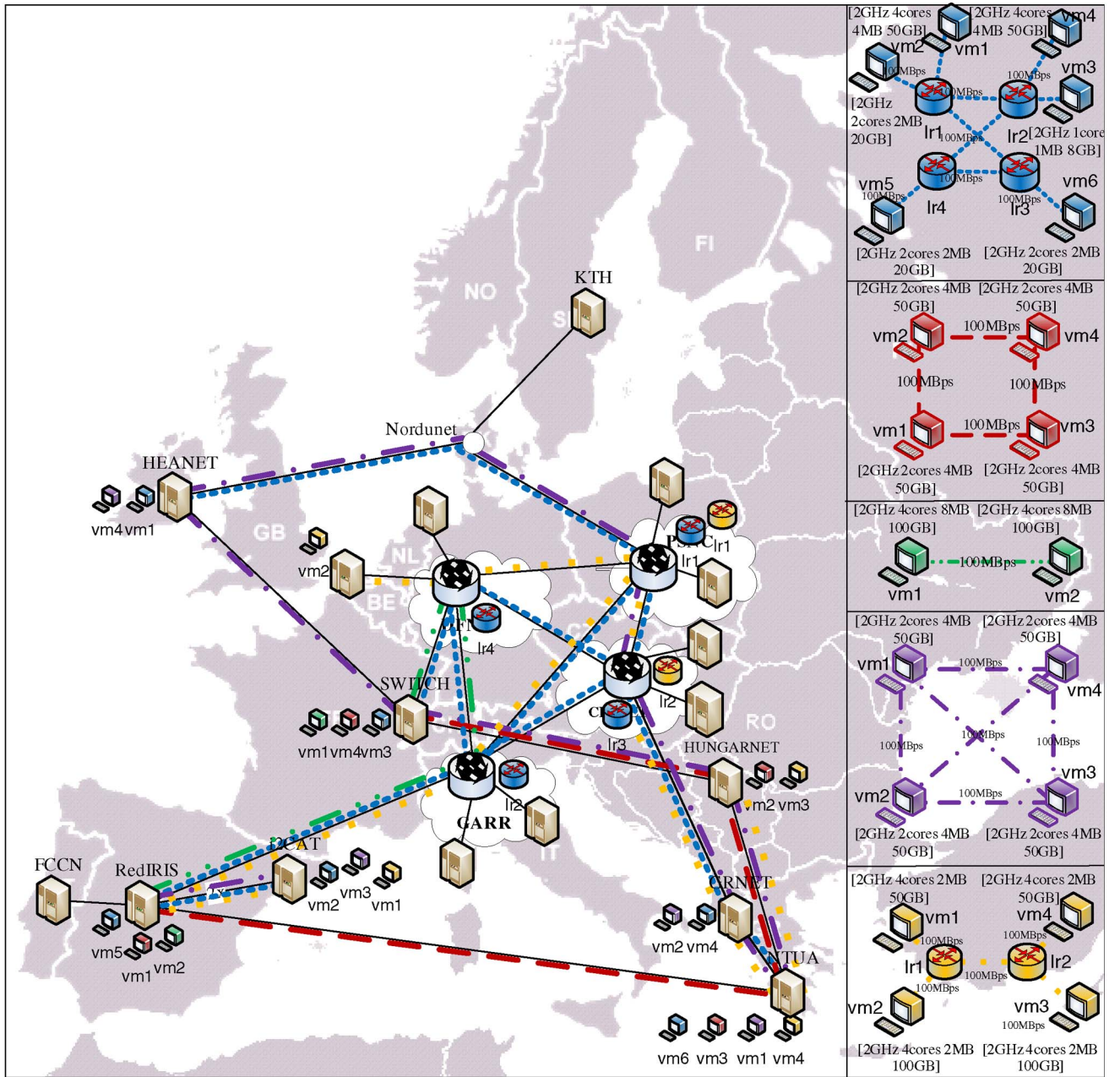


Fig. 7. G-SP mapping on the FEDERICA substrate.

7 CONCLUDING REMARKS

In this paper, we study the virtual resource allocation problem for networked cloud environments, incorporating heterogeneous substrate resources, and provide an appropriate approximation approach to address the problem. Specifically, for the node mapping phase, we provide a MIP problem formulation capable of taking into consideration QoS requirements. Appropriate relaxation and application of a randomized rounding technique leads to a polynomial-time solution. Following, link mapping is determined by solving the corresponding multicommodity flow problem. The proposed solution is compared against two well-known approaches on embedding virtual resource requests to a physical substrate. Based on extensive modeling and experimentation, utilizing *CVI-Sim*—a simulation/emulation environment that allows for a flexible and structured

evaluation of the performance and efficiency of the proposed approach, we conclude that the proposed NCM approach overall outperforms other commonly applied algorithms. Specifically, NCM provides a tradeoff between G-SP and G-MCF in terms of acceptance ratio of NCM requests and number of hops on the substrate, per virtual link. At the same time, NCM manages to embed requests that generate more revenue, at a cost similar to G-MCF. An appropriate reconfiguration strategy has been also adopted to deal with the highly dynamic networked cloud environment.

It should be noted that although in this paper the optimal NCM problem has been addressed considering heterogeneous resources, the main focus has been placed on wired networks and infrastructures. Extending however the proposed framework to take into account new and more dynamic heterogeneous environments and infrastructures, beyond the traditional Internet (e.g., wireless), presents

additional challenges, due to the nature of the wireless environment (e.g., coherence, isolation and uniqueness of nodes), the stochastic nature of the corresponding resources, as well issues associated with the existence of mobile nodes. The development of such a holistic framework is of high research and practical importance and is part of our current and future work.

ACKNOWLEDGMENTS

This work was supported in part by the European Commission, seventh Framework Program for Research and Technological Development, Capacities, Grant No. 257867 - NOVI.

REFERENCES

- [1] R. Buyya, C.S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities," *Proc. IEEE Int'l Conf. High Performance Computing and Communications (HPCC '08)*, pp. 5-13, Sept. 2008. doi: 10.1109/HPCC.2008.172.
- [2] "Creating Energy Efficient Data Centers," U.S. Dept. of Energy, May 2007.
- [3] D. Breitgand, A. Epstein, and B. Rochwerger, "Resource Management Mechanisms to Support SLAs in IaaS Clouds" *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*, pp. 288-307, IGI Global, 2012, doi:10.4018/978-1-4666-1631-8.
- [4] I.M. Lloriente, R.S. Montero, B. Sotomayor, B. Breitgand, and D. Maraschini, "On the Management of Virtual Machines for Cloud Infrastructures," *Cloud Computing: Principles and Paradigms*, pp. 157-191, John Wiley & Sons, Jan. 2011, doi:10.1145/1809049.1809052.
- [5] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, "Efficient Resource Provisioning in Compute Clouds via VM Multiplexing," *Proc. Seventh Int'l Conf. Autonomic Computing*, pp. 11-20, June 2010. doi:10.1145/1809049.1809052.
- [6] I. Houdi, W. Louati, W.B. Ameur, and D. Zeghlache, "Virtual Network Provisioning Across Multiple Substrate Network," *J. Computer Networks*, vol. 55, no. 2, pp. 1011-1023, 2011, dx.doi.org/10.1016/j.comnet.2010.12.011.
- [7] M. Chowdhury, M.R. Rahman, and R. Boutaba, "ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping," *IEEE/ACM Trans. Networking*, vol. 20, no. 1, pp. 206-219, Feb. 2012, doi: 10.1109/TNET.2011.2159308.
- [8] "The FEDERICA Project Website," <http://www.fp7-federica.eu>, 2013.
- [9] D. Andersen, "Theoretical Approaches To Node Assignment," *Unpublished Manuscript*, <http://www-2.cs.cmu.edu/~dga/papers/andersen-assign.ps>, 2002.
- [10] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration," *ACM SIGCOMM Computer Comm. Rev.*, vol.38, no. 2, pp. 17-29, Apr. 2008, doi:10.1145/1355734.1355737.
- [11] W. Szeto, Y. Iraqi, and R. Boutaba, "A Multi-Commodity Flow Based Approach to Virtual Network Resource Allocation," *Proc. IEEE GLOBECOM '03*, vol. 6, pp. 3004-3008, Dec. 2003, doi:10.1109/GLOCOM.2003.1258787.
- [12] J. Fan and M.H. Ammar, "Dynamic Topology Configuration in Service Overlay Networks: A Study of Reconfiguration Policies," *Proc. IEEE INFOCOM '06*, pp. 1-12, Apr. 2006, doi:10.1109/INFOCOM.2006.139.
- [13] J. Lu and J. Turner, "Efficient Mapping of Virtual Networks onto a Shared Substrate," Technical Report WUCSE-2006-35, Washington Univ. St. Louis, 2006.
- [14] Y. Zhu and M.H. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," *Proc. IEEE INFOCOM '06*, pp. 1-12, Apr. 2006, doi:10.1109/INFOCOM.2006.322.
- [15] I. Houdi, W. Louati, and D. Zeghlache, "A Distributed Virtual Network Mapping Algorithm," *Proc. IEEE Int'l Conf. Comm. (ICC '08)*, pp. 5634-5640, May 2008, doi:10.1109/ICC.2008.1056.
- [16] J. Lischka and K. Holger, "A Virtual Mapping Algorithm Based on Subgraph Isomorphism Detection," *Proc. ACM SIGCOMM '09*, pp. 81-88, Aug. 2009, doi:10.1145/1592648.1592662.
- [17] L. Lallemand and A. Reifert, "On Force-Based Placement of Distributed Services within a Substrate Network," *Proc. EUNICE/IFIP WG 6.6 Conf. Networked Services and Applications: Eng., Control and Management (EUNICE '10)*, pp. 65-75, June 2010, doi:10.1007/978-3-642-13971-0-7.
- [18] Y. Liu, Y. Li, K. Xiao, and H. Cui, "Mapping Resources for Network Emulation with Heuristic and Genetic Algorithms," *Proc. Int'l Conf. Parallel and Distributed Computing, Applications and Technologies (PDCAT '05)*, pp. 670-674, 2005, doi:10.1109/PDCAT.2005.166.
- [19] I. Houdi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, "Adaptive Virtual Network Provisioning," *Proc. ACM SIGCOMM '10*, Sept. 2010.
- [20] S. Zhang, Z. Qiant, S. Guo, and S. Lu, "FELL: A Flexible Virtual Network Embedding Algorithm with Guaranteed Load Balancing," *Proc. IEEE Int'l Conf. Comm. (ICC)*, pp. 1-5, June 2011, doi: 10.1109/icc.2011.5962960.
- [21] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNE-AC: Virtual Network Embedding Algorithm Based on Ant Colony Metaheuristic," *Proc. IEEE Int'l Conf. Comm. (ICC)*, pp. 1-6, June 2011, doi: 10.1109/icc.2011.5963442.
- [22] A. Haider, R. Potter, and A. Nakao, "Challenges in Resource Allocation in Network Virtualization," *Proc. ITC Specialist Seminar Network Virtualization*, May 2009.
- [23] N.M. Mosharaf, K. Chowdhury, and R. Boutaba, "A Survey of Network Virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862-876, Apr. 2010, doi:10.1016/j.comnet.2009.10.017.
- [24] M. Chowdhury, F. Samuel, and R. Boutaba, "PolyViNE: Policy-Based Virtual Network Embedding Across Multiple domains," *Proc. ACM SIGCOMM '10*, pp. 49-56, Sept. 2010, doi:10.1145/1851399.1851408.
- [25] F. Zaheer, J. Xiao, and R. Boutaba, "Multi-Provider Service Negotiation and Contracting in Network Virtualization," *Proc. IEEE Network Operations and Management Symp. (NOMS)*, pp. 471-478, June 2010, doi: 10.1109/NOMS.2010.5488487.
- [26] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and A. Yumerefendi, "Embedding Virtual Topologies in Networked Clouds," *Proc. Sixth Int'l Conf. Future Internet Technologies (CFI '11)*, pp. 26-29, June 2011, doi:10.1145/2002396.2002403.
- [27] I. Houdi, W. Louati, D. Zeghlache, and S. Baucke, "Virtual Resource Description and Clustering for Virtual Network Discovery," *Proc. IEEE Int'l Conf. Communications (ICC '09)*, pp. 1-6, June 2009, doi:10.1109/ICCW.2009.5207979.
- [28] "VMware vStorage Thin Provisioning," <http://www.vmware.com/files/pdf/VMware-vStorage-Thin-Provisioning-DS-EN.pdf>, 2013.
- [29] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.
- [30] M.G.C. Resende and P. Pardalos, *Handbook of Optimization in Telecommunication*. Springer, 2006.
- [31] P. Raghavan and C.D. Thompson, "Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs," *Combinatorica*, vol. 7, no. 4, pp. 365-374, 1987, doi:10.1007/BF02579324.
- [32] "PlanetLAB Website," <http://www.planet-lab.org>, 2013.
- [33] "JUNG 2.0.1 Website," <http://jung.sourceforge.net>, 2013.
- [34] "JFC(Swing/AWT) Website," <http://download.oracle.com/javase/6/docs/technotes/guides/swing>, 2013.
- [35] "IBM ILOG CPLEX Optimizer," <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>, 2013.
- [36] Juniper MX480, <http://www.juniper.net/>, 2013.
- [37] "NOVI FP7 STREP Project Website," <http://www.fp7-novi.eu>, 2013.
- [38] FIRE: Future Internet Research and Experimentation, <http://cordis.europa.eu/fp7/ict/fire>, 2013.
- [39] L. Lymberopoulos, M. Grammatikou, M. Potts, P. Grosso, A. Fekete, B. Belter, M. Campanella, and V. Maglaris, "NOVI Tools and Algorithms for Federating Virtualized Infrastructures," *Future Internet - From Technological Promises to Reality*, pp. 213-224, Springer-Verlag, 2012, doi: 10.1007/978-3-642-30241-1_19.



Chrysa Papagianni received the diploma degree and the PhD degree in electrical and computer engineering, both from NTUA in 2003 and 2009, respectively. She is a senior R&D associate at the Network Management and Optimal Design Laboratory (NETMODE), National Technical University of Athens (NTUA), Athens, Greece, since October 2010. Her research interests include the area of computer networks with emphasis on cloud computing,

virtualization, network optimization and management, and service provisioning.



Aris Leivadeas received the diploma degree in electrical and computer engineering from the University of Patras, Greece in 2008, and the MSc degree in mobile and personal communications from King's College London, United Kingdom, in 2009. He has been a member of the Network Management and Optimal Design Laboratory (NETMODE), National Technical University of Athens, Greece, since October 2010, where he is currently working toward the

PhD degree. His research interests include cloud computing, wired/wireless virtualization, and optimization.



Symeon Papavassiliou received the diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 1990, and the MSc and PhD degrees in electrical engineering from Polytechnic University, Brooklyn, NY, in 1992 and 1996, respectively. He is currently an associate professor with the Faculty of Electrical and Computer Engineering Department, National Technical University of Athens. From

1995 to 1999, he was a senior technical staff member at AT&T Laboratories in Middletown, New Jersey, and in August 1999 he joined the Electrical and Computer Engineering Department at the New Jersey Institute of Technology (NJIT), where he was an associate professor until 2004. He was awarded the Best Paper Award in IEEE INFOCOM '94, the AT&T Division Recognition and Achievement Award in 1997, the US National Science Foundation (NSF) Career Award in 2003, and the Best Paper Award in IEEE WCNC 2012. He has an established record of publications in his field of expertise, with more than 200 technical journal and conference published papers. His main research interests include the area of computer and communication networks with emphasis on wireless communications, resource allocation, and optimization, network virtualization and performance analysis, and evaluation of stochastic systems.



Vasilis Maglaris received the diploma degree in mechanical and electrical engineering from NTUA in 1974 and the PhD degree from Columbia University, New York in 1979. He is a professor of Electrical & Computer Engineering (ECE) at the National Technical University of Athens (NTUA) since 1989. He subsequently held various industrial and academic positions in the USA (1979-1989) and in Greece (1989-now). His research interests include Internet

technologies, network design and optimization and distributed computing systems. He is currently chairing the Policy Committee that governs GEANT the Pan-European next generation interconnection of more than 34 National Research & Education Networks in the extended European Research Area. He also served on the board of the Greek National Regulatory Authority on Telecommunications and Posts for two five-year terms (1995-2005). He is currently directing the Network Management & Optimal Design (NETMODE) Laboratory within the ECE School at NTUA. He has authored more than 100 research papers and regularly delivers lectures on Internet advances.



Cristina Cervelló-Pastor received the MSc degree in telecom engineering and the PhD degree in telecommunication engineering, both from the Barcelona School of Telecommunications Engineering (ETSETB), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. She is an associated professor in the Department of Telematics Engineering of UPC. Currently, she is the head of this Department. She is also the leader of the line of investigation on

Optical Networks within the BAMPLA group. The research trajectory has been centered on the field of routing in high speed networks and the development of new protocols and services in optical networks, taking part in diverse national and European projects (NOVI, FEDERICA, ATDMA, A@DAN, Euro-NGI, Euro-FGI, EURO-NF) and being responsible of various public and private funding R&D projects. In parallel, she has published diverse papers in national and international journals and conferences and she is supervising thesis in the field of routing and contention resolution in high-speed networks



Álvaro Monje received the degree in informatics engineering in 2009 from the Facultat d'Informàtica de Barcelona (FIB), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently with the NOVI project. He has worked in the FEDERICA project and previously for FIB as administrator and maintenance coordinator.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.