Energy-Aware Mapping and Live Migration of Virtual Networks

Esteban Rodriguez, Gustavo P. Alkmim, Nelson L. S. da Fonseca, *Senior Member, IEEE*, and Daniel Macêdo Batista, *Member, IEEE*

***Abstract*—Network virtualization facilitates the deployment of new protocols and applications without the need to change the core of the network. One key step in instantiating virtual networks (VNs) is the allocation of physical resources to virtual elements (routers and links), which can be then targeted for the minimization of energy consumption. However, such mappings need to support the quality-of-service requirements of applications. Indeed, the search for an optimal solution for the VN mapping problem is NP-hard, and approximated algorithms must be developed for its solution. The dynamic allocation and deallocation of VNs on a network substrate can compromise the optimality of a mapping designed to minimize energy consumption, since such allocation and deallocation can lead to the underutilization of the network substrate. To mitigate such negative effects, techniques such as live migration can be employed to rearrange already mapped VNs in order to improve network utilization, thus minimizing energy consumption. This paper introduces a set of new algorithms for the mapping of VNs on network substrates designed to reduce network energy consumption. Moreover, two new algorithms for the migration of virtual routers and links are proposed with simulation showing the efficacy of the algorithms.**

***Index Terms*—Cloud, green networks, next-generation networking, virtualization, virtual network mapping.**

I. Introduction

THE minimalist approach and the independence of specific network technologies at the link layer have enabled the global spread of the Internet, with the core of the Internet designed essentially to the forwarding of packets. However, as a consequence of such minimalism, various attempts have been made to provide missing features in its original design.

Network virtualization has been proposed to overcome these limitations. It allows the definition of virtual networks (VNs) composed of virtual routers and links; these are then hosted by routers and links in real networks, which are called "network substrates." Network virtualization allows the coexistence of different protocol stacks and architectures on the same substrate, without the need for modifying the physical network. Moreover, it imposes no restrictions on these protocols and architectures.

One of the main issues in network virtualization is the efficient mapping of VNs onto the substrate network [1], [2]. This mapping determines the allocation of routers and links of a VN onto the routers and links of the substrate network. However, the search for the optimal mapping of VNs is an NP-hard problem [3].

One of the problems faced by telecommunication companies and Internet Service Providers (ISPs) in recent years is an increase in energy consumption due to the growing spread of broadband access and the expansion of services offered [4]. According to Bolla *et al.* [5], [6], the increase in the volume of network traffic follows Moore's law, doubling every 18 months, whereas the improvement of energy efficiency in silicon technologies follows Dennard's law, increasing only by a factor of 1.65 every 18 months. There is, thus, a constant increase in power consumption in communication networks, a consumption corresponding to 2%–10% of the power consumed in the world today, and this is expected to increase in the coming years.

Advances in hardware have allowed the design of energy-efficient network devices by the adoption of "power on demand" operation [7]–[9]. Techniques employed on the physical layer have made transmission more energy efficient, but future advances in energy-aware networking are expected to take place at the architectural level. In this context, network virtualization plays a key role, since it can use physical devices in a more efficient way, thus reducing the need for the acquisition of new devices.

This paper addresses the issue of energy savings in VN mapping. It introduces an algorithm based on integer linear programming and an algorithm based on the decomposition of the original model to reduce the computational demand required to find a solution to the problem, a decomposition made possible for networks with over 400 routers. Moreover, this paper introduces two algorithms to guide live migration of VNs for the reestablishment of minimum energy consumption after the termination of a VN and the consequent change in the state of network. One unique characteristic of the algorithm proposed is the consideration of various assumptions not considered in previous papers to make the modeling of the problem more realistic. Numerical examples evaluate the tradeoff between energy consumption and bandwidth consumption. Results show that the proposed algorithm promotes substantial energy saving, yet supports the quality-of-service (QoS) requirements of VNs. This paper revises the mathematical formulation and the associated numerical results presented in [10] and [11] and

Manuscript received November 10, 2014; revised June 2, 2015; accepted July 30, 2015. This work was supported in part by Ministerio Ciencia y Tecnología y Telecomunicaciones de Costa Rica (MICITT/CONICIT), by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAESP).

E. Rodriguez, G. P. Alkmim, and N. L. S. da Fonseca are with the Institute of Computing, State University of Campinas, 13083-852 Campinas-SP, Brazil (e-mail: nfonseca@ic.unicamp.br).

D. M. Batista is with the Institute of Mathematics and Statistics, University of São Paulo, 05508-090 São Paulo-SP, Brazil.

Digital Object Identifier 10.1109/JSYST.2015.2467159

expands the performance evaluation of the proposed algorithms by validating them in larger network substrates.

This paper is organized as follows. Section II describes related work and situates the contribution of this paper in relation to a previously published paper. Section III introduces the energy consumption model used in this paper. Section IV presents the mathematical formulation of the mapping problem, whereas Section V presents the proposed algorithms. Section VI shows the performance evaluation of the proposed algorithm, and Section VII concludes this paper.

II. Related Work

The seminal work in [12] discussed the energy consumption in networks, and it is based on various investigations reported in the literature. This work stated that the energy consumption on the Internet must indeed be taken into consideration, and it explored the possibility of putting devices to sleep to promote energy savings.

The idea of using virtualization as a technique for the reduction of energy consumption was highlighted in [13] as one of the strategies of green networking. This paper also provided a categorization of green networking strategies.

In [14], there was an extensive data collection of power consumption elements in IP over wavelength division multiplexing (WDM) networks, such as the power consumption of the chassis, line cards, and the processor of the routers, which were used as data for this paper.

A mixed-integer linear programming (MILP) model for the design of energy-efficient WDM networks was introduced in [15], and two heuristics were developed to attempt to minimize the number of physical components used in the network, as well as their power consumption. It considered an array of physical elements, as well as traffic aggregation. The power consumption value of line cards was computed, as was the consumption of a chassis distributed over the number of line cards, and this has been used to calculate the power consumption of the physical links in this paper. The work in [15] considered the energy consumed by the ports of an IP router as a measure of the total energy consumption of a router. This differs from the approach used in this paper, since the power consumption of the physical router in this paper has been broken down into several components.

The research in [16] considered a wide area network and evaluated the possibility of turning off elements without violating QoS requirement constraints. It considered that the energy consumption of the physical links is much lower than that of the nodes. VNs, however, are not considered.

The work in [2], by the same authors of this paper, approaches the VN mapping problem in detail, surveying an extensive collection of virtual networking technologies and algorithms, and proposing several algorithms. The work reported considered a variety of parameters and has been used as the basis for this paper.

The work in [17] also introduced a solution for the VN mapping problem. The objective was to allocate a set of VN requests to a reduced group of physical network equipment, and a mixed-integer program was proposed as a solution. Results showed that it is possible to reduce the energy consumption (up to 35% of the node consumption and 25% of the bandwidth consumption), but the parameters considered in this paper, such as locality constraint, memory, and operating system images, were ignored. The scenarios in [17] involved much smaller networks, with 15 physical nodes and VN requests of only five virtual nodes, than those considered here.

The work in [18] extended the VN mapping algorithm for increasing energy efficiency that was presented in [17]. Variation of the problem includes selection criteria, which allows several virtual nodes to be mapped onto the same substrate node. The VN scenarios explored used the Waxman topology algorithm [19], whereas in this paper, actual power consumption values of actual devices are used [14], [20], [21]. Moreover, this paper utilizes the generally preferred Barabasi–Albert 2 topology generator [22].

Moreover, no constraints on the alignment of virtual topologies with physical topologies are assumed here, and it is possible that the topology of physical routers and links allocated to a VN be the same as that of the requested VN, but this happens only if that topology is the one that minimizes bandwidth allocation. Moreover, such an alignment is not necessary to guarantee the QoS requirements of applications, which are indeed assured by the constraints of the mapping problem.

The work presented in [23] was one of the first to introduce the concept of migration of operating systems under "live constraints," known as live migration. This made a clear distinction between hardware and software and facilitates fault management, load balancing, and low-level system maintenance. The work in [24] used Xen for virtual machines migration when machines are far apart. Such migration can be usually achieved within a downtime of 1–2 s. The paper showed that some of the transfer time involved in live migration across large geographical distances is often negligible. The work in [25] also modeled the transfer of virtual machines and proposed an algorithm to reduce the downtime. It made clear that the present proposal for live migration is realistic.

The authors in [26] used a specific application to quantify the effect of live migration on the Internet. Their work also indicated advantages of live migration, such as improved manageability, performance, and fault tolerance. The authors suggested the adoption of load balancing criterion to migrate the workload of a server.

The work in [27] compared live migration approaches based on Xen and OpenFlow. It showed that live migration can be employed as a green networking strategy. However, no concrete proposal to reduce the energy consumption was presented. This paper introduced algorithms that can be employed for live migration as a green networking strategy.

The work in [28] explored the impact of live migration in data centers. Assumptions were made to reduce the energy consumption during migration; however, the work did not introduce any technique for such reduction.

The work in [29] and [30] presented algorithms to reduce the power consumption in clouds using migration. The solution in [29], named SERCON, tried to minimize the number of migrations, whereas that in [30] presented a dynamic threshold-based scheme, which rearranges virtual machines

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

depending on their utilization. Both showed that migrations can reduce energy consumption in networks, although the work was focused on data centers.

The work in [31] proposed an optimization model for reducing the energy consumption of VN provisioning, which considers both communication and processing power consumption in cloud-based data centers. The authors considered both workload-independent and workload-dependent power consumption and introduced a heuristic for VN provisioning to reduce energy consumption while complying with service level agreements. They showed that packing VNs into a minimum set of physical links and nodes can lead to hotspots in the cloud infrastructure, which, in turn, results in large blocking of VN request. They suggested that a tradeoff between energy consumption and blocking should be considered.

The work in [32] proposed a trade-off between maximizing the number of VNs that can be accommodated by an ISP and minimizing the energy costs of the whole system. The authors pointed out that one of the challenges is the quantification of the cost of the energy for complex physical network infrastructure of an ISP, specifically to the joint consideration of the price of electricity and the consumption of energy. They used a discrete-time model to characterize the dynamics of price changes and distinguished nodes in the infrastructure as host nodes and router nodes. Heuristics based on particle swarm were proposed and were shown to produce solutions with greater reduction in energy consumption than do other proposals in the literature.

The work in [33] showed that power savings are generally attained at the expense of increased propagation delays. The authors proposed delay- and power-minimized provisioning to minimize propagation delay, maximize the power savings in IP over WDM networks, and minimize the power consumption overhead in data centers. Simulation results showed that the proposal achieved low delays and low power provisioning in an environment dominated by cloud services.

The work in [34] proposed a holistic approach for a large-scale cloud system in which the cloud services are provisioned by several data centers interconnected over a backbone network. The authors proposed an MILP formulation for virtual machine placement that aims to minimize both power consumption of the virtualized backbone network and resource usage in data centers. Since the MILP formulation required long-running computations, the problem was partitioned into two subproblems, namely, intra and inter data centers virtual machine placement. Moreover, a heuristic was proposed to reduce the computational time during reconfiguration of the virtualized backbone topology.

The work in [35] proposed an energy-efficient VN embedding approach for cloud computing networks, in which power savings are introduced by consolidating network resources and data center resources. Energy consumption of an IP over WDM network was modeled using an MILP to reduce the power consumption by reducing the number of activated nodes and links. Results showed that the model achieved a maximum power saving of 60% of that of other approaches in the literature. The authors also proposed a heuristic for real-time energy optimization. Furthermore, they studied the impact of delay and node location constraints on the energy efficiency of VN embedding and showed how such embedding can impact the design of optimally located data centers for minimal power consumption in cloud networks. An additional contribution is the emphasis that power savings and spectral efficiency can be obtained by VN embedding in optical orthogonal-division multiplexing networks.

TABLE I
COMPARISON BETWEEN MAPPING ALGORITHMS

Reference	**Virtual Network Mappings**	**Green Networking**	**Live Migration**	**Joint bandwidth/power reduction**
[36]	no	yes	no	no
[37]	yes	yes	no	no
[1]	yes	yes	yes	no
[3]	yes	yes	no	no
[15]	yes	yes	no	no
[16]	no	yes	no	no
[17]	yes	yes	no	yes
[2]	yes	no	no	no
[18]	yes	yes	no	no
[29]	yes	yes	yes	no
[30]	no	yes	yes	no
[31]	yes	yes	no	no
[32]	yes	yes	no	yes
[33]	yes	yes	no	no
[34]	yes	yes	yes	no
[35]	ye	yes	no	no
This Proposal	yes	yes	yes	yes

Reference	**Processor Cores**	**Bandwidth**	**Locality Constrains**	**Images for Instantiating**
[36]	no	no	no	no
[37]	no	no	no	no
[1]	yes	no	no	no
[3]	no	no	no	no
[15]	yes	yes	no	no
[16]	no	yes	no	no
[17]	no	yes	no	no
[2]	yes	yes	yes	yes
[18]	no	yes	no	no
[29]	no	yes	no	no
[30]	no	no	no	no
[31]	yes	yes	no	no
[32]	yes	yes	yes	no
[33]	no	yes	no	no
[34]	yes	yes	no	no
[35]	yes	yes	yes	no
This Proposal	yes	yes	yes	yes

Reference	**Link Delay**	**Available memory and image size**	**Repository location of images**	**Mapping Finalization Time**
[36]	no	no	no	no
[37]	no	no	no	no
[1]	yes	no	no	no
[3]	no	no	no	no
[15]	no	no	no	no
[16]	no	no	no	no
[17]	no	no	no	no
[2]	yes	yes	yes	yes
[18]	no	no	no	yes
[29]	no	no	no	no
[30]	no	no	no	no
[31]	no	no	no	no
[32]	no	no	no	no
[33]	yes	no	no	no
[34]	no	no	no	no
[35]	yes	no	no	no
This Proposal	yes	yes	yes	yes

Table I compares the solutions of the aforementioned papers with the one in this paper, which is unique, since it considers sets of images with different sizes, different times required to instantiate virtual routers, different locations of repositories in which the operating system images are stored, and different sizes of the available memory of the physical routers, as well as a general reduction in energy consumption and live migration strategies to further minimize the energy consumption. This work thus provides a significant advance in the mapping of green VN onto substrate networks, since it provides a more realistic assessment of operational networks.

III. Energy Consumption Model

The evaluation of the approaches for mapping VNs onto physical substrates requires a realistic power consumption model, which breaks the power consumption of the physical elements down into the components related to the physical router and those of the physical links.

A. Physical Router

The power consumption of a router consists of both traffic-dependent and traffic-independent components. The traffic-independent component typically represents around 90% of power consumption of the router. Three major components contribute to this power consumption: the router chassis, the router processor, and the line cards [14], [20], [21].

Most of this power consumption is accounted for by the router chassis, since all the components of the router must be cooled and protected by the chassis. This consumption is constant. Thus, it is crucial to minimize the number of chassis to be turned "on" in the network. A chassis should be turned off only when none of its processors is active. The cost of the electronic components of the router, such as the motherboard and its physical memory, are encompassed in the cost of the chassis as a whole.

Line cards, which are required to connect a physical interface to the router, consume considerably less power, but this power is added to that of the other elements. Moreover, since the number of line cards is determined by the number of physical links in the network, their power consumption is traffic dependent, with intensive traffic resulting in a larger number of slots required to satisfy the bandwidth demands of the network.

Router processors are assumed to be limited to a single physical processor at a time per slot. Each slot can have cores, which can potentially be used for the allocation of different virtual routers in order to satisfy their processing demands. Regardless of the number of processors that are active at a time, the chassis remains active.

The incorporation of all three of these components distinguishes the present model from other models in the literature, since a constant value for the energy consumption of the router is generally attributed.

B. Physical Links

The model proposed in [15] and [38] is employed here for the determination of the power consumption of the physical link. In this model, the energy cost of a physical link depends on the length of that link, given by the number of amplifiers required along it. A span distance $\mathcal{P}^s$ determines that every $\mathcal{P}^s$ kilometers (km), a new amplifier is required to properly propagate the signal. The following equations give the energy cost of a physical link:

$$\mathcal{P}^{L}_{u,v} = \mathcal{P}^{\#A}_{u,v} \times \mathcal{P}^{CA} \quad (1)$$

$$\mathcal{P}^{\#A}_{u,v} = \left\lceil \frac{J_{u,v}}{\mathcal{P}^s} - 1 \right\rceil + 2 \quad (2)$$

where

1) $\mathcal{P}^{L}_{u,v}$ gives the power consumed by the physical link (u, v);
2) $\mathcal{P}^{\#A}_{u,v}$ gives the number along amplifiers of the link (u, v);
3) $\mathcal{P}^{CA}$ gives the energy consumption of each amplifier;
4) $J_{u,v}$ gives the length of the link (u, v).

In this paper, it is assumed that all links have a single fiber, with two amplifiers added at the endpoints of the link, where signal amplification is needed.

IV. Proposed Model

This paper models requests for VN establishment on network substrates that arrive dynamically. Each request specifies the topology of the VN, the resources demanded by the VN elements, and the QoS requirements, which include a bound on the time permitted for the instantiation of the VNs and location constraint for that of the nodes of the VN.

The model proposed takes the request for the establishment of the VN and tries to select which elements of a network substrate should be allocated for the instantiation of a VN being requested. The selection criteria aim at minimizing the energy consumption, while satisfying the requirements of the request.

The model is based on two 0–1 integer linear programming (ILP) submodels. It differs from that in [2] by the introduction of the two submodels for the reduction of the computational memory demand. Table II presents the notation used in the model.

The values of $\mathcal{P}^{\text{chassis}}$, $\mathcal{P}^{L}$, $\mathcal{P}^{\text{card}}$, and $\mathcal{P}^{\text{core}}$ denote the power consumption of the chassis, physical link, line cards, and cores, respectively.

The inclusion of D, K, $L_{n,m}$, $E_{m,i}$, B, G, S, $T_{n,i}$, I, and $R_{n,i}$ in the model makes it unique and more realistic than the previous one due to the consideration of such diverse aspects of real networks. The maximum delay allowed along the network links (D, K) affects the QoS furnished for applications sensitive to delays, particularly those involving video and audio. The specific image required by a virtual router should be defined, and the content of each repository must be known $(R_{n,i})$ to determine from which repository the image should be downloaded $(I, E_{m,i})$. Locality restrictions and the size of the images must be known since routers have a limited storage capacity (B, G) and the size of the image impacts on the download time. Moreover, users may have policy issues that prevent the utilization of certain physical routers $(L_{n,m})$ or can restrict the solution to employ certain energy-efficient sites. Furthermore, the maximum time acceptable for the instantiation of the VN is related to the urgency of the VNs and service priority $(S, D, K, T_{n,i})$.

In the formulations, the following variables define the state of occupancy of the network substrate.

1) K_n denotes the number of cores allocated in the physical router n.
2) $O_{u,v}$ denotes the number of virtual links that use the physical link u, v.

TABLE II
NOTATION USED IN THIS PAPER

Notation	Description
N	set of physical routers
F	set of physical links, with the physical link (n_1, n_2) connecting the physical routers n_1 and $n_2 \in N$
M	set of virtual routers
V	set of virtual links, with the virtual link (m_1, m_2) connecting the virtual routers m_1 and $m_2 \in M$
I	set of images stored in the repository, each corresponding to a file with an operating system and a specific software ready to be instantiated on a physical router
A	number of available cores in the physical routers, with $A(n)$ giving the number of cores of router n
P	set of the number of cores requested by virtual routers, with $P(m)$ giving the number of cores required by the virtual router m
C	set of values of available bandwidth for the physical links, with $C(f)$, $f \in F$, giving the available bandwidth of the link f
Q	set of bandwidth values requested by virtual links, with $Q(v)$, $v \in V$, giving the bandwidth required by the virtual link v
D	set of values of delays in the physical links, with $D(f)$, $f \in F$, giving the delay along the link f
K	set of values of maximum delay allowed on a virtual link, with $K(v)$, $v \in V$, representing the maximum delay allowed along the virtual link v
$L_{n,m}$	defines restrictions on the location of physical routers, with the value of the variable being 1 if the virtual router m can be mapped onto the physical router n; otherwise, it is 0. If a user does not want a virtual router m to be mapped onto a given physical router due to any policy reason the variables of $L_{n,m}$ must be 0
$R_{n,i}$	provides details about the location where images are stored; if the image i is located in a repository with a direct link to the physical router n, the value of the variable is 1; otherwise, it is 0
$E_{m,i}$	related to software restrictions. If the image i contains all the software requirements (operating system, protocol stacks and kernel modules) required by the virtual router m, the value of the variable is 1; otherwise, it is 0
B	set of values of the memory available in the physical routers, with $B(n)$ representing the memory available in the router n
G	set of image sizes, with $G(i)$ representing the size of the image i
S	time threshold for instantiating the virtual network
$T_{n,i}$	represents the time the physical router n takes to boot the image i
$\mathcal{P}^{chassis}$	denotes the power consumption of the chassis
$\mathcal{P}^{L}$	denotes the power consumption of the physical link
$\mathcal{P}^{card}$	denotes the power consumption of the linecards
$\mathcal{P}^{core}$	denotes the power consumption of the cores
K_n	denotes the number of cores allocated in the physical router n
$O_{u,v}$	denotes the number of virtual links that use the physical link u, v
$X_{n,m,i}$	has a value of 1 if the virtual router m is mapped onto the physical router n using the image i; otherwise, it is 0
$Y_{u,v,w}$	has a value of 1 if the physical path used by the virtual link w includes the physical link (u, v); otherwise, it is 0
$Z_{u,v,m}$	has a value of 1 if the physical path (u, v) is used to transfer the image requested by the virtual router m; otherwise, it is 0
U_n	has a value is 1 of the physical router (n) is to be powered on; otherwise, it is 0
$W_{u,v}$	has a value is 1 of the physical path (u, v) is to be powered on; otherwise, it is 0

The values of K_n and $O_{u,v}$ are used for the computation of α_n and $\beta_{u,v}$, which simplifies the objective function, i.e.,

$$\alpha_n = \left\lceil \frac{K_n}{K_n + 1} \right\rceil \tag{3}$$

$$\beta_{u,v} = \left\lceil \frac{O_{u,v}}{O_{u,v} + 1} \right\rceil. \tag{4}$$

The values of α_n and $\beta_{u,v}$ determine whether a router and a physical link, respectively, are already in use.

The solution of the problem is given by the following binary variables: $X_{n,m,i}$, $Y_{u,v,w}$, $Z_{u,v,m}$, U_n, and $W_{u,v}$.

The mapping of VNs is based on the sequential execution of two ILPs. The first, i.e., ILP-Green-Mapping, maps the VNs onto the substrate. The second, i.e., ILP-Green-Image, determines the path in the substrate used to transfer the images.

ILP-Green-Mapping is formulated as follows:

Minimize

$$\mathcal{P}^{\text{chassis}} \sum_{n \in N} (\alpha_n + (1 - \alpha_n) U_n) + \mathcal{P}^{\text{core}} \sum_{n \in N} \sum_{m \in M} \sum_{i \in I} (X_{n,m,i} \times P(m)) + \left(2\mathcal{P}^{\text{card}}_{u,v} + \mathcal{P}^{L}_{u,v}\right) \sum_{(u,v) \in F} (\beta_{u,v} + (1 - \beta_{u,v}) W_{u,v})$$

subject to

$$\sum_{n \in N} \sum_{i \in I} X_{n,m,i} = 1 \quad \forall\, m \in M \tag{C1}$$

$$\sum_{m \in M} \sum_{i \in I} X_{n,m,i} \le 1 \quad \forall\, n \in N \tag{C2}$$

$$\sum_{m \in M} \sum_{i \in I} P(m) \times X_{n,m,i} \le A(n) \quad \forall\, n \in N \tag{C3}$$

$$X_{n,m,i} = 0 \quad \forall\, n \in N,\ \forall\, m \in M,\ \ \forall\, i \in I | L_{n,m} = 0 \text{ or } E_{m,i} = 0 \tag{C4}$$

$$\sum_{w' \in V} Y_{u,v,w'} \times Q(w') \le C(w) \quad \forall\, w = (u, v) \in F \tag{C5}$$

$$\sum_{u \in N} \sum_{v \in N} Y_{u,v,w} \times D(u, v) \le K(w) \quad \forall\, w \in V, (u, v) \in F \tag{C6}$$

$$\sum_{m \in M} \sum_{i \in I} X_{n,m,i} \times G(i) \le B(n) \quad \forall\, n \in N \tag{C7}$$

$$Y_{u,v,w} = 0 \quad \forall\, u,\ \forall\, v,\ \forall\, w \in V | (u, v) \notin F \tag{C8}$$

$$\sum_{f \in N} Y_{n,f,w} - \sum_{f \in N} Y_{f,n,w} = \sum_{i \in I} X_{n,a,i} - \sum_{i \in I} X_{n,b,i} \quad \forall\, w = (a, b) \in V,\ \forall\, n \in N \tag{C9}$$

$$X_{n,m,i} \le U_n \quad \forall\, n \in N,\ \forall\, m \in M,\ \forall\, i \in I \tag{C10}$$

$$U_n \le \sum_{m \in M} \sum_{i \in I} X_{n,m,i} \quad \forall n \in N \tag{C11}$$

$$Y_{u,v,w} \le W_{u,v} \quad \forall v \in V, \forall (u, v) \in F \tag{C12}$$

$$W_{u,v} \le \sum_{v \in V} Y_{u,v,w} \quad \forall\, (u, v) \in F \tag{C13}$$

$$X_{n,m,i} \in \{0, 1\} \quad \forall\, n \in N,\ \forall\, m \in M,\ \forall\, i \in I \tag{C14}$$

$$Y_{u,v,w} \in \{0, 1\} \quad \forall\, u,\ \forall\, v,\ \forall\, w \in V \tag{C15}$$

$$U_n \in \{0, 1\} \quad \forall\, n \in N \tag{C16}$$

$$W_{u,v} \in \{0, 1\} \quad \forall\, u,\ \forall\, v \in V. \tag{C17}$$

The objective function minimizes the power consumed by the arriving request. Constraint (C1) establishes that a virtual router be assigned to a single physical router and that a single image is used to instantiate it. Constraint (C2) limits the number of virtual routers that can be allocated per request to a single physical router. Only a single virtual router is allowed to be allocated in a physical router per request. Constraint (C9) ensures that the set of physical links that comprise a virtual link be a valid path. It compares the in-degree and the out-degree of each physical router n.

Constraints (C3) and (C7) express the limitations of the physical routers. They ensure that each physical router will allocate no more than the maximum capacity of cores and memory, respectively.

Constraint (C4) guarantees that the virtual routers will be instantiated only using images that meet its software requirements and geographic location.

Constraints (C5) and (C6) express the limitations of the physical links. Constraint (C5) ensures that the bandwidth available for each physical link is greater than that required for all of the virtual links using it. Constraint (C6) establishes that the total delay in the physical path allocated to a virtual link should not exceed the delay threshold allowed for that virtual link.

Constraint (C8) guarantees that if (u, v) does not correspond to a physical link, it will not be used in the mapping.

Constraints (C10) and (C11) express the energy constraints of the physical routers. Constraint (C10) ensures that no core can be assigned to a given router without turning on the device first. Constraint (C11) ensures that if the router is powered on, then at least one core will be assigned to that router.

Constraints (C12) and (C13) express the energy constraints of the physical links. Constraint (C12) ensures that a virtual link can be used on the physical link (u, v) only if the physical link is powered on. Constraint (C13) ensures that if the link is powered on, then at least one virtual link will be assigned to that physical link.

Constraints (C14), (C15), (C16), and (C17) define the domains of the binary variables.

After solving ILP-Green-Mapping, the values of $X_{n,m,i}$ and $Y_{u,v,w}$ are used as input for the second model, the ILP-Green-Image problem.

The ILP-Green-Image problem is formulated as follows:

$$\text{Minimize} \sum_{m \in M} \sum_{u \in N} \sum_{v \in N | (u,v) \in F} Z_{u,v,m} \times D(u,v) + \frac{Z_{u,v,m} \times G(i | X_{n,m,i} = 1)}{C(u,v)} \text{ subject to}$$

$$\sum_{m \in M} Z_{u,v,m} = 0 \quad \text{(C18)}$$
$$\forall u,\ \forall v | (u,v) \notin F$$

$$\sum_{j \in N} Z_{u,j,m} - \sum_{j \in N} Z_{j,u,m} = X_{n,m,i} \times R_{u,i} - X_{n,m,i} \times \left(1 - \left\lceil \frac{|u-n|}{\alpha} \right\rceil \right) \quad \text{(C19)}$$
$$\forall m \in M,\ \forall i \in I,\ \forall n, u \in N,\ \alpha = |N|$$

$$Z_{u,v,m} \in \{0,1\} \quad \text{(C20)}$$
$$\forall u,\ \forall v,\ \forall m \in M.$$

This objective function minimizes the time required to instantiate the VN. The time needed for the instantiation of each virtual router is the sum of the times required to transfer the image and to boot the operating system, assuming that two or more images can be transferred simultaneously through the same physical link.

Constraint (C18) guarantees that (u, v) will not be used if it does not belong to the considered substrate. Constraint (C19) establishes that the set of physical links allocated for the transfer of an image consists of a valid path in the substrate network. Constraint (C20) defines the domains of the variables.

V. Algorithms Proposed

This section presents the virtual mapping algorithm and the live migration algorithms based on the formulation presented in the previous section.

A. VN Mapping Algorithms

The models proposed were implemented using the *IBM ILOG CPLEX Optimizer* [39].

The ILP is solved using *CPLEX*'s dynamic search algorithm. It is based on the *branch and cut* [40] algorithm. In the formulation, both *ILP-GREEN-Mapping* and *ILP-GREEN-Image* search for a solution using all the nodes of the search tree. The Opt algorithm used to solve the ILP formulations is presented in Algorithm 1.

Algorithm 1: Opt Algorithm

Input: Substrate γ, virtual network request δ.
Result: Mapping of δ in γ using paths θ to transfer the images.
1 Define γ and δ as input parameters for *ILP-GREEN-Mapping*;
2 Search the entire tree using *ILP-GREEN-Mapping* and obtain the values of variables $X_{n,m,i}$ and $Y_{n,u,w}$ related to best feasible solution found;
3 **if** *ILP-Mapping cannot find a solution* **then**
4 | Block the request;
5 **else**
6 | Use γ, δ and variable $X_{n,m,i}$ as input parameters for *ILP-Image*;
7 | Search the entire tree using *ILP-GREEN-Image* and obtain the values of variables $Z_{n,u,m}$ that gives the best solution;
8 | **if** *ILP-GREEN-Image cannot find a solution* **then**
9 | | Block the request;
10 | **else**
11 | | Return the mapping of δ in γ defined values of variables $X_{n,m,i}$ e $Y_{n,u,w}$;
12 | | Return the paths θ defined values of variable $Z_{n,u,m}$.

This algorithm takes a long time to find solutions for substrates with more than 20 nodes. Simulations with substrate of more than 25 nodes can take more than two days. The need for such extended time period motivated the implementation of a heuristic, denominated *Root*, which limits the search on the root of the search tree, considerably reducing the execution time in relation to the optimal implementation.

The *Root* algorithm differs from the implementation of the optimal implementation in the composition of lines 2 and 7, which are replaced respectively by the following.

1) *Line 2*: Stop the search for solutions for *ILP-GREEN-Mapping* at the root of the search tree and use the values of the variables $X_{n,m,i}$ and $Y_{n,u,w}$.
2) *Line 7*: Stop the search for the solutions for *ILP-GREEN-Image* at the root of the search tree and use the values of variable $Z_{n,u,m}$.

B. Migration Algorithms

The central idea of remapping VNs is to restore a network configuration that consumes minimal energy. One of the options offering most promising opportunities for remapping is the termination of a VN, since virtual routers can be "packed" together on a reduced number of physical routers, thus allowing the deallocation of all virtual routers hosted on a given physical router so that it can be turned off.

The algorithms presented in this paper are to be executed when existing VNs have terminated, and those VNs can be rearranged to save energy.

These algorithms assume that the transfer of information about the state of running processes is negligible given the high capacity of the physical links. Moreover, it is assumed that the times required 1) to boot the VNs, 2) to redirect the traffic to the new VN, 3) to finalize the transferring of the old VN, and 4) to interrupt the services of the old VN are also negligible [24].

Algorithm 2, which is called Live Migration Nodes Recently Used (LM_NRU), aims at reducing the energy consumption by reallocating the virtual routers on those physical routers that hosted the routers of the terminated VNs. Before explaining the LM_NRU algorithm, certain mathematical notations are introduced in Table III.

TABLE III
NOTATION USED IN THIS PAPER

Notation	Description
$P = \{p_i\}$	set of physical routers;
$C = \{c_{ij}\}$	set of physical links (channels), with C_{ij} connecting physical routers (P_i, P_j);
$V^k = \{V_i^k\}$	set of virtual routers of virtual network k;
$L^k = \{l_{ij}^k\}$	set of virtual links of the virtual the network k, with l_{ij}^k connecting virtual routers (V_i^k, V_j^k);
$VN_k = (V^k, L^k)$	virtual network k;
$VN = \{VN_k\}$	set of active virtual networks;
$A(t) = [a_{ijk}(t)]$	matrix with elements identified as $a_{ijk} = 1$ if the virtual router $V_j^k \in VN_K$ is allocated on the physical router $P_i \in P$ at time t;
$F(t) = [f_{ij}(t)]$	matrix with elements $f_{ij}(t)$ defining the allocated bandwidth on the physical link $c_{ij} \in C$;
$S(VN, t) = (A(t), F(t))$	state of the substrate network hosting VM at time t;
$E(S(VN, t))$	energy consumption of $S(VN, t)$
t_k	time at which virtual network k is terminated, $t_k \in R^+$
$D_k = \{P_k \in P \| a_{ijk}(t_k^-) = 1$ and $a_{ijk}(t_k^+) = 0\}$	set of physical routers on which a virtual router of VN_k was allocated
$G^k = \{VN_m \| a_{ijk}(t_k^-) = 1$ and $a_{ijk}(t_k^+) = 0_i^d \in D_k\}$	set of virtual networks which have virtual routers allocated to the same physical router that hosted a virtual router of the terminated virtual network k

Algorithm 2: LM_NRU Algorithm

Input: Z - terminating virtual network
Input: $S(VN, t_z$ - state of the substrate when VN_z terminates
Result: S' - state of the substrate
1 $VN' = VN \cap G^z$;
2 for all $VN_i \in G^z$;
3 **if** *RE-MAP* (VN_i) **then**
4 | $VN' = VN' \cup \{VN_i\}$;
5 **else**
6 | return $S(VN, t_z^+)$;
7 **if** $E(S(VN', t_z^+)) \leq E(S(VN, t_z^t))$ **then**
8 | return $S(VN', t_z^+)$;
9 **else**
10 | return $S(VN, t_z^+)$;

The LM_NRU algorithm receives as input the state of the network substrate at the time of VN termination, and it returns to a state of that network, which will minimize energy consumption.

In line 1, all VNs, which have virtual routers allocated to the same physical router that has hosted a virtual router of the terminating VN, are "unmapped."

In lines 2–4, attempts are made to remap all unmapped VNs. The aim is to produce a new mapping, which will consume less energy than was being consumed when the VN z terminated. RE-MAP is a binary function that returns true when the remapping of VN_i is feasible. In this case, a new mapping of VNs, including VN_i, is considered (line 7).

Otherwise, the process of remapping of all unmapped VNs is interrupted, and the substrate remains in the same state as it was when the VN terminated.

If all unmapped VNs are remapped and the new substrate state consumes less energy, this new state is given as the solution by the algorithm, and the VNs will migrate to the physical routers suggested by this solution. If the new mapping does not lead to a reduction in energy consumption, however, no migration will be undertaken.

Algorithm 3 [the Live Migration-ALL (LM_ALL) algorithm] was designed to reduce the energy consumption by reallocating all virtual routers of the network. It differs from Algorithm 2 in that it deallocates all virtual routers, whereas the LM_NRU algorithm deallocates only a part of the network. The LM_ALL algorithm "unmaps" all VNs (line 1) and then tries to remap all of them (lines 2–6). The rest of Algorithm 2 is similar to that of Algorithm 3.

Algorithm 3: LM_ALL Algorithm

Data: Z - terminating virtual network
Data: $S(VN, t_z$ - state of the substrate when VN_z terminates
Result: S' - state of the substrate
1 $VN' = \{\}$;
2 for all $VN_{i \neq z}$;
3 **if** *RE-MAP* (VN_i) **then**
4 | $VN' = VN' \cup \{VN_i\}$;
5 **else**
6 | return $S(VN, t_z^+)$;
7 **if** $E(S(VN', t_z^+)) \leq E(S(VN, t_z^t))$ **then**
8 | return $S(VN', t_z^+)$;
9 **else**
10 | return $S(VN, t_z^+)$;

TABLE IV
VALUES OF THE PARAMETERS USED IN THE SIMULATION

Parameter	Value
Bandwidth of physical link	[1, 10] Gbps
Number of images in the network	3
Simulation time	5000s
Average duration per request	360s
Number of virtual nodes per request	[3, 6]
Bandwidth of each virtual link	[100, 300] Mbps
Upperbound to time required to instantiate a virtual network	100s
RAM memory	768MB
Image size	128MB
Cores per physical router	6
Cores per virtual router	[2, 5]
Physical link delay	Defined by BRITE
Virtual link delay	15 × the value defined by BRITE
Upperbound to the time required to instantiate an image in a router	10s

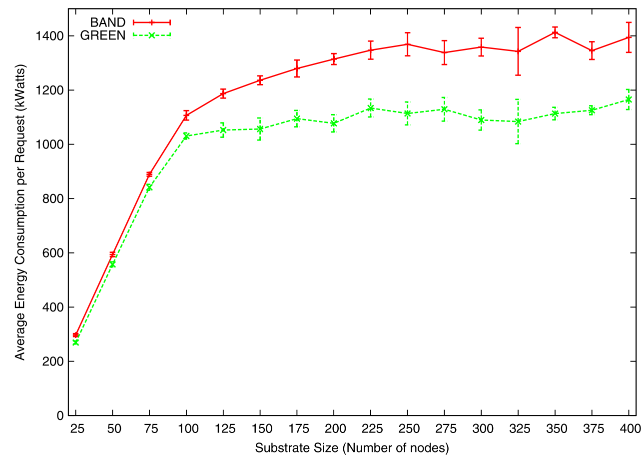


Fig. 1. Average power consumption per request as a function of the number of nodes of the substrate.

VI. PERFORMANCE EVALUATION

A. *Configuration of Experiments*

To evaluate the performance of the proposed algorithms, another optimization model was developed containing similar constraints, but with the minimization of the allocated bandwidth as the objective function. For the remainder of this paper, the model proposed is referred to as GREEN, whereas that which minimizes the allocated bandwidth is referred to as BAND [2]. The evaluation was conducted in dynamic scenarios, in which the availability of resources on the substrate network varies as a function of time.

The average energy consumption per request, the amount of bandwidth allocated per request, and the substrate utilization of physical nodes and physical links were evaluated as a function of the substrate size (number of physical routers), as well as the interarrival time of requests.

The effectiveness of the proposed models was assessed using a simulator implemented in C++. The simulator received as input a description of the substrate network. The simulator processed requests for VN establishment one at a time. VNs were mapped onto the network substrate using the VN mapping algorithm presented in Section V-A. The algorithms presented in Section V-B were executed when a VN request was deallocated; resources were then released for future allocation by incoming VNs or remapped ones.

Requests for virtual establishment arrive according to a Poisson process, and the lifetimes of these VNs were exponentially distributed. The topologies of both physical and VNs were generated using the BRITE network topology generator and the Barabasi–Albert 2 algorithm [41]. Both the capacity of the substrate network links and the requested bandwidth were drawn from a uniform distribution. Delay requirements for data transfer should be greater than the propagation delay of the physical links. To assure this, the values of link weights provided by BRITE were multiplied by a random value from a uniform distribution. The number of cores demanded by these VNs is also taken from a uniform distribution. The parameter values used in the simulations are given in Table IV.

Confidence intervals with a 95% confidence level were derived using the independent replication method. The ILP formulations were implemented using the CPLEX optimization library version 12.0. Simulations were executed on a computer equipped with two Intel Xeon 2.27-GHz processors, with six cores each, and 6 GB of random access memory, running the operating system Debian GNU/Linux Squeeze.

Although simulations were derived for interarrival times of requests varying from 25 to 300 s, the results in this paper are shown only for an interarrival time of 25 s, since these higher loads result in less differences between the results derived by the GREEN and BAND algorithms, due to rapid achievement of saturation of the network substrate. Results are shown as a function of the number of nodes in the network substrate. Each point on the curve shows the results for ten different topologies. For each topology, ten simulations with different seeds for the random number generator were executed.

B. *Evaluation of the Mapping Algorithm*

Fig. 1 shows the average energy consumption per request as a function of the number of nodes in the substrate. The average energy consumption increases for both models since a higher number of nodes in the substrate implies that the mapping tends to use a large number of nodes. The energy consumption of the chassis is large compared with the other elements; thus, the impact on the consumption per request increases, since there will be a larger number of active nodes. For substrates with more than 100 nodes, the GREEN model tends to allocate cores in routers already active (powered on), and the growth of energy consumption per request stabilizes. However, for the BAND model (GREEN does not share this tendency to utilize active routers), the consumption per request tends to increase a larger number of active routers. The consumption of BAND stabilizes when there are 250 nodes. The energy reduction produced by the GREEN model was on the order of 20% for substrate sizes larger than 100 nodes.

Fig. 2 shows the average bandwidth allocated per request as a function of the number of nodes of the substrate. The bandwidth consumption per request is greater for the GREEN model since it tends to use routers already powered regardless of the location of these routers, which can imply on long

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

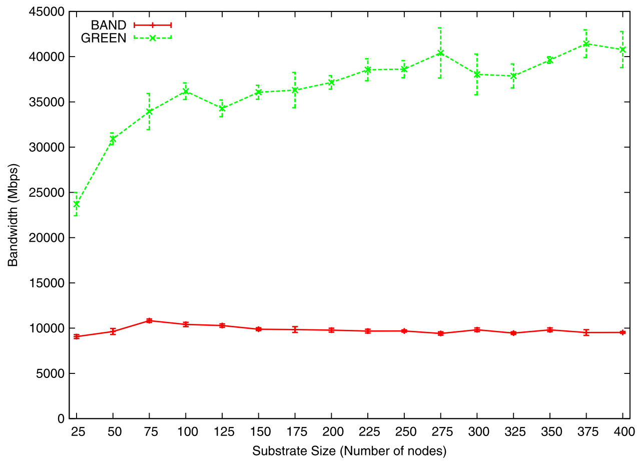


Fig. 2. Average allocated bandwidth per request as a function of the number of nodes of the substrate.

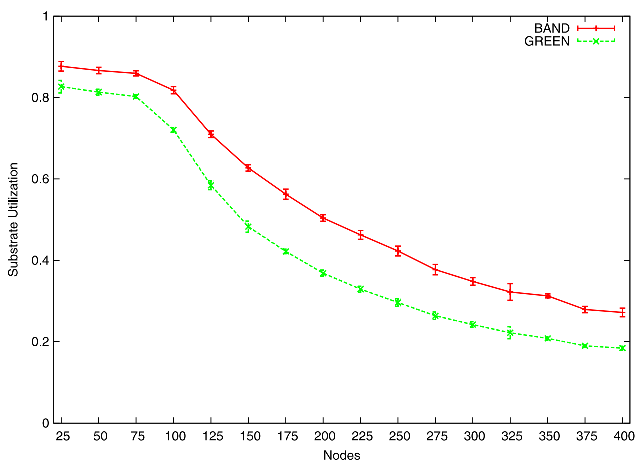


Fig. 3. Substrate nodes utilization as a function of the number of physical nodes.

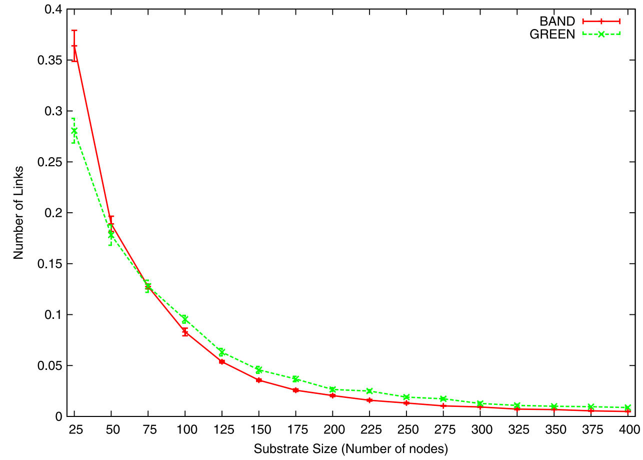


Fig. 4. Substrate links utilization as a function of the number of physical nodes.

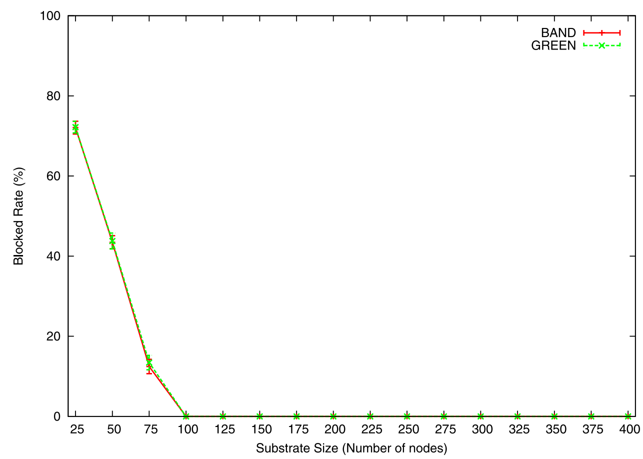


Fig. 5. Blocking ratio as a function of the number of physical nodes.

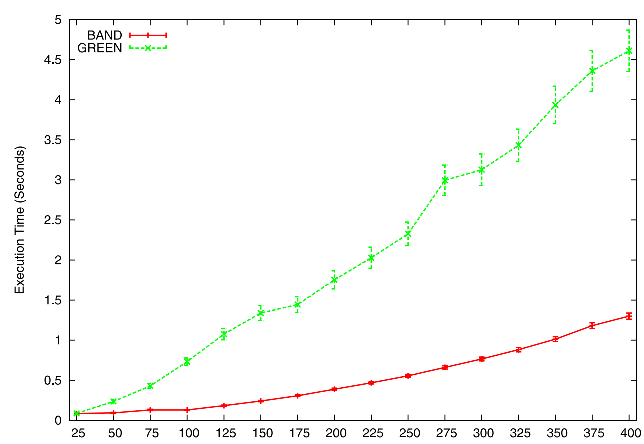


Fig. 6. Execution time of the BAND and GREEN algorithms as a function of the number of physical nodes.

paths between nodes. Although the bandwidth consumption demanded by the GREEN model is greater than that consumed by the BAND model, it does not violate the QoS requirements of the requests, since these are guaranteed by the constraints defined in the formulation.

Fig. 3 shows the utilization of physical nodes, which is the ratio between the physical routers occupied and the total number of physical routers, as a function of the substrate size. The utilization in both scenarios decreases as the physical substrate sizes increase, because there are more nodes available to allocate the incoming requests. The GREEN model reduces the utilization of these nodes, which is at most 14% of that produced by BAND. This difference is almost constant regardless of the substrate size, averaging 10%. This is due to that the GREEN model tends to use powered routers whenever possible, thus concentrating the VN requests on a smaller physical substrate.

Fig. 4 shows the utilization of physical links, which is the ratio between the physical links occupied and the total number of physical links available, as a function of the substrate size. For substrates larger than 100 nodes, the GREEN model uses more physical links than BAND. Such trend does not happen for substrate with 25, 50, and 75 because requests saturate, the physical nodes are saturated with requests, and, consequently, the network is unable to handle enough requests for this trend to be revealed. The blocking ratio and the execution time of the BAND and GREEN algorithms as a function of the number of physical nodes are shown in Figs. 5 and 6, respectively.

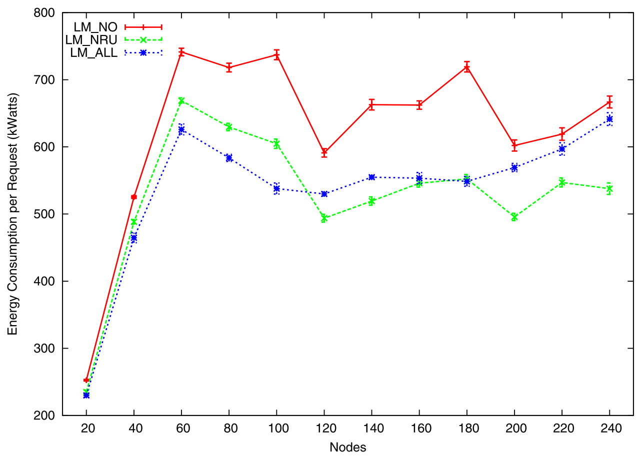


Fig. 7. Average power consumption per request.

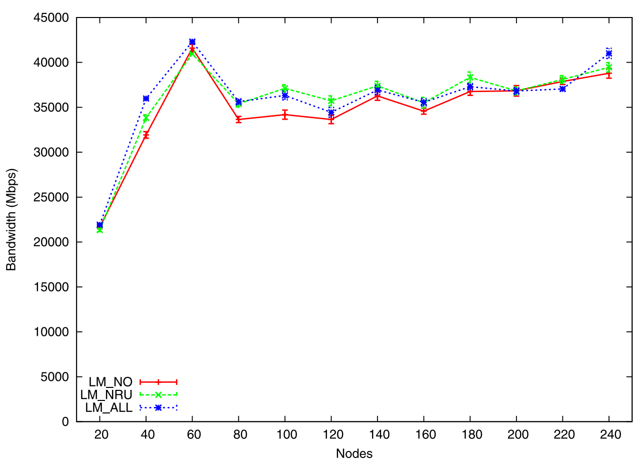


Fig. 8. Average allocated bandwidth per request as a function of the number of physical nodes.

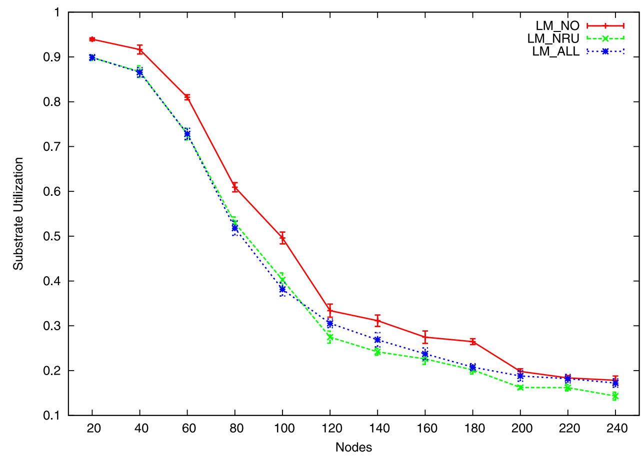


Fig. 9. Substrate utilization as a function of the substrate size.

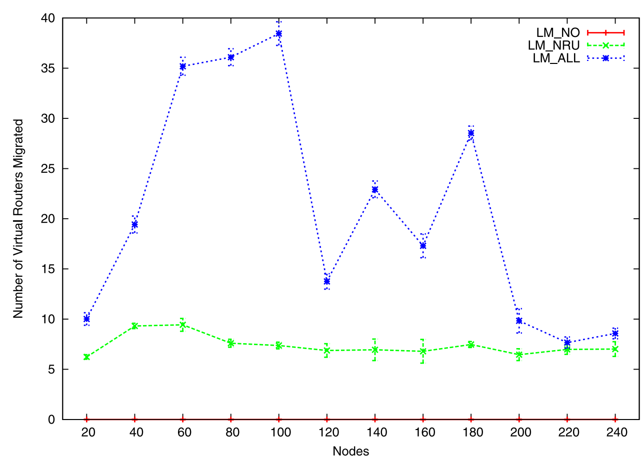


Fig. 10. Average virtual routers migrated per event of VN termination as a function of the number of physical nodes.

C. Results of Live Migration Algorithms

Fig. 7 shows the average energy consumption per request using the scheme with no live migration (LM_NO) and with the two proposed algorithms, for network substrates of at most 240 nodes. Both LM_ALL and LM_NRU show significant energy savings per request when compared with LM_NO, which was, on average, 20% and up to 25% for substrates with more than 60 routers. However, for small substrate (less than 40 routers), no significant difference was observed since there is little room for optimized mapping of VNs with such a reduced number of physical routers. Energy savings furnished by the LM_NRU and LM_ALL models show no significant differences, which favors the adoption of the LM_NRU algorithm, since it has less overhead.

The proposed algorithms also consumed a similar amount of bandwidth even when no migration occurred (see Fig. 8). Although the no-migration scheme does tend to consume the lowest amount of bandwidth, this reduction can reach 8%. When the substrate size increases, this difference becomes negligible.

Fig. 9 shows the substrate utilization as a function of the substrate size. The proposed algorithms show a significant reduction in utilization, although the difference between them is negligible. The reduction is around 13% in relation to a no-migration scheme, and this difference remains almost constant regardless of the substrate size. This difference is due to the concentration of a larger number of VNs since the reallocation of VN resources and the smaller number of physical routers are used in a no-migration scheme, and it is the result of savings obtained from the reduction of the number of physical router chassis powered.

Figs. 10 and 11 display the average number of virtual router and virtual link migrations per VN termination event, respectively. These figures clearly show that LM_ALL takes advantage of all available choices in the remapping of the VNs. As the number of physical nodes increases, the number of virtual routers and virtual links migrated decreases since various VNs are mapped onto the same physical nodes and links onto which they were originally mapped. The number of migrations of both virtual routers and links per event produced by LM_NRU is almost constant. LM_NRU migrates 41% fewer virtual routers than LM_ALL and 38% fewer virtual links than LM_ALL. In relation to the energy savings resulting from the use of these algorithms, LM_NRU is more attractive than LM_ALL since it produces a lower number of migrations.

The LM_ALL algorithm remaps all active networks at each termination of a VN and rearranges the VNs to contribute to a reduction in energy consumption. Therefore, the number of

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

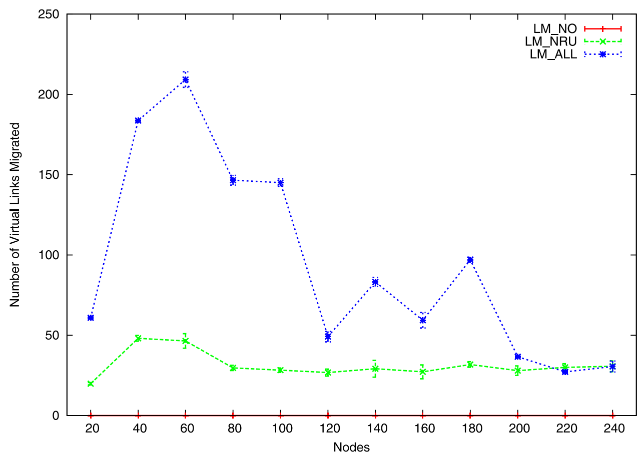


Fig. 11. Average virtual links migrated per event of VN termination as a function of the number of physical nodes.

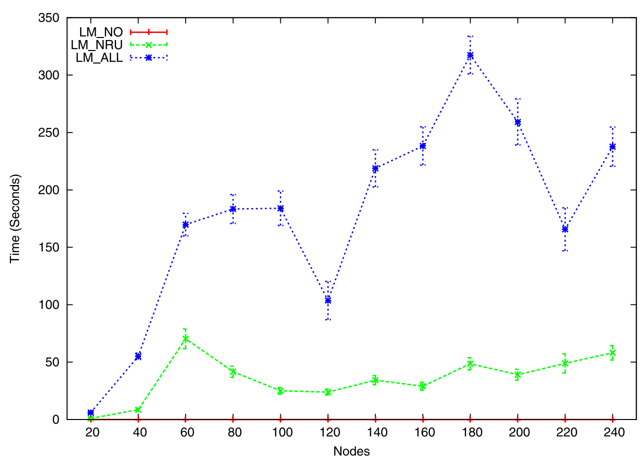


Fig. 12. Execution time per event of VN termination as a function of the number of physical nodes.

nodes and virtual links migrated will be greater than that for the LM_NRU algorithm. The large variation in LM_ALL curves in Figs. 10 and 11 can be explained by the fact that each point on the curve corresponds to the results for ten different topologies of the network substrate, which led to a large number of mapping possibilities.

Fig. 12 shows the average execution time per event of VN termination. LM_ALL leads to more sharply increased demand than does the LM_NRU algorithm. LM_ALL is at most 750% slower than LM_NRU as a result of fewer deallocation and reallocation that needs to be done. Considering the similarity in energy savings and the time complexity and overhead costs of the LM_ALL algorithm, the adoption of the LM_NRU algorithm for the live migration of VNs is recommended.

VII. Conclusion

Network virtualization is a promising technique used nowadays to resolve various issues related to the lack of flexibility of the network core. In addition to the benefits highlighted in the previous sections, virtualization is crucial for the saving of energy in the development of a greener Internet.

This paper has presented novel algorithms for reducing the network energy consumption. A new model minimizing energy consumption in the mapping of VNs onto network substrates has been presented. This model is based on an ILP model and considers various realistic parameters generally neglected by previous papers in the literature. Moreover, this paper proposes the use of live migration to further reduce energy consumption and presents two algorithms for this purpose.

One of the main contributions of this paper is the consideration of real-life parameters omitted in previous studies for the green VN mapping problem and the use of live migration for the reduction of energy consumption in core networks, such as the availability of preconfigured images for each virtual router limitations of flash memory in physical routers. Moreover, each VN request specifies a bound for the time to be used in mapping it onto the substrate.

In order to minimize the energy consumption per request, a two-step ILP was developed. Due to the complexity of the problem (NP-hard), a heuristic was developed to reduce the execution time of the problem. This proposal was then compared with another one that minimize only the bandwidth consumption. The results show that the energy savings per request can be on the order of 50% for some scenarios. Moreover, the QoS requirements of the VNs demands were always supported.

The remapping of VNs is a technique that can reduce network energy consumption up to 18%, when compared with a no-migration scheme, without compromising bandwidth demands. The LM_NRU algorithm leads to less overhead than that produced by the LM_ALL algorithm, yet energy savings are similar. Consequently, the LM_NRU algorithm is recommended for VN remapping with live migration.

The proposals in this paper try to answer the call for the ever-growing demands for a decrease in energy consumption in communication networks [42], [43]. They can be used for the virtualized networks of the future, which will definitely have to refrain from consumption of energy in the face of growing network demands.

References

[1] N. Chowdhury, M. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 783–791.
[2] G. P. Alkmim, D. M. Batista, and N. L. S. Fonseca, "Optimal mapping of virtual networks," in *Proc. IEEE GLOBECOM*, 2011, pp. 1–5.
[3] I. Houidi, W. Louati, and D. Zeghlache, "A distributed and autonomic virtual network mapping framework," in *Proc. ICAS*, 2008, pp. 241–247.
[4] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient cloud content delivery in core networks," in *Proc. IEEE GLOBECOM Workshops*, Atlanta, GA, USA, Dec. 9–13, 2013, pp. 420–426.
[5] R. Bolla *et al.*, "The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization?" *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 80–86, Aug. 2011.
[6] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Commun. Surveys Tuts.*, vol. 33, no. 2, pp. 223–244, 2nd Quart. 2011.
[7] J. Wu, S. Rangan, and H. Zhang, *Green Communications: Theoretical Fundamentals, Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2013.
[8] X. Dong, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficiency of optical OFDM-based networks," in *Proc. IEEE ICC*, Budapest, Hungary, Jun. 9–13, 2013, pp. 4131–4136.
[9] M. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Switch-off transients in cellular access networks with sleep modes," in *Proc. IEEE ICC Workshops*, Jun. 2011, pp. 1–6.

[10] E. Rodríguez, G. P. Alkmim, D. M. Batista, and N. L. S. da Fonseca, "Live migration in green virtualized networks," in *Proc. IEEE ICC Workshops*, Budapest, Hungary, Jun. 9–13, 2013, pp. 2262–2266.

[11] E. Rodríguez, G. P. Alkmim, D. M. Batista, and N. L. S. da Fonseca, "Green virtualized networks," in *Proc. IEEE ICC*, Ottawa, ON, Canada, Jun. 10–15, 2012, pp. 1970–1975.

[12] M. Gupta and S. Singh, "Greening of the Internet," in *Proc. SIGCOMM Conf. Appl., Technol., Architect., Protocols Comput. Commun.*, 2003, pp. 19–26.

[13] R. Dario, B. A. Prem, C. Claude, and R. Jean-Louis, "A survey of green networking research," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 3–20, 1st Quart. 2009.

[14] F. Idzikowski, "Power consumption of network elements in IP over WDM networks," Telecommun. Netw. Group, Tech. Univ. Berlin, Berlin, Germany, TKN Tech. Rep. Ser. TKN-09-006, Jul. 2009.

[15] G. Shen and R. S. Tucker, "Energy-minimized design for IP over WDM networks," *J. Opt. Commun. Netw.*, vol. 1, no. 1, pp. 176–186, Jun. 2009.

[16] L. Chiaraviglio, M. Mellia, and F. Neri, " Energy-aware networks: Reducing power consumption by switching off network elements, " in *Proc. FEDERICA-Phosphorus Tutorial Workshop*, 2008, pp. 1–6.

[17] J. Botero *et al.*, "Energy efficient virtual network embedding," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 756–759, May 2012.

[18] A. Fischer, M. Beck, and H. de Meer, "An approach to energy-efficient virtual network embeddings," in *Proc. 5th Int. Workshop ManFI*, 2013, pp. 1–6.

[19] B. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[20] Cisco, Brochures for Devices CRS-1 and Components, CISCO ONS 15501 and Others, accessed 06/2011, 2011. [Online]. Available: http://www.cisco.com/

[21] Product data sheet, T Series Core Routers, Sep. 2015. [Online]. Available: http://www.juniper.net/us/en/products-services/routing/t-tx-series/

[22] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE," Accessed at 12/20/2010, 2010. [Online]. Available: http://www.cs.bu.edu/brite/

[23] C. Clark *et al.*, "Live migration of virtual machines," in *Proc. 2nd Conf. Symposium NSDI*, 2005, vol. 2, pp. 273–286.

[24] F. Travostino *et al.*, "Seamless live migration of virtual machines over the MAN/WAN," *Future Gener. Comput. Syst.*, vol. 22, no. 8, pp. 901–907, Oct. 2006.

[25] F. Moghaddam and M. Cheriet, "Decreasing live virtual machine migration down-time using a memory page selection based on memory change PDF," in *Proc. ICNSC*, 2010, pp. 355–359.

[26] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," *CoRR*, vol. abs/1109.4974, 2011.

[27] P. S. Pisa *et al.*, "OpenFlow and Xen-based virtual network migration," in *Proc. World Comput. Congr., Network Future Conf.*, Brisbane, Qld., Australia, Sep. 2010, pp. 170–181.

[28] S. Kikuchi and Y. Matsumoto, "Impact of live migration on multi-tier application performance in clouds," in *Proc. IEEE 5th Int. Conf. CLOUD Comput.*, Jun. 2012, pp. 261–268.

[29] A. Murtazaev and S. Oh, "Sercon: Server consolidation algorithm using live migration of virtual machines for green computing," *IETE Tech. Rev.*, vol. 28, no. 3, pp. 212–231, May/Jun. 2011.

[30] S. Richa, P. Nidhi, and D. Hiteshi, "Power aware live migration for data centers in cloud using dynamic threshold," *Int. J. Comput. Technol. Appl.*, vol. 2, no. 6, pp. 2041–2046, Nov./Dec. 2011.

[31] G. Sun *et al.*, "Power-efficient provisioning for online virtual network requests in cloud-based data centers," *IEEE Syst. J.*, vol. 9, no. 2, pp. 427–441, Jun. 2015.

[32] S. Su *et al.*, "Energy-aware virtual network embedding," *IEEE/ACM Trasn. Netw.*, vol. 22, no. 5, pp. 1607–1620, Oct. 2014.

[33] B. Kantarci and H. Mouftah, "Designing an energy-efficient cloud network [invited]," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 4, no. 11, pp. B101–B113, Nov. 2012.

[34] B. Kantarci, L. Foschini, A. Corradi, and H. T. Mouftah, "Design of energy-efficient cloud systems via network and resource virtualization," *Int. J. Netw. Manage.*, vol. 25, no. 2, pp. 75–94, Mar./Apr. 2015. [Online]. Available: http://dx.doi.org/10.1002/nem.1838

[35] L. Nonde, T. El-Gorashi, and J. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *J. Lightw. Technol.*, vol. 33, no. 9, pp. 1828–1849, May 2015.

[36] W. Szeto, Y. Iraqi, and R. Boutaba, "A multi-commodity flow based approach to virtual network resource allocation," in *Proc. IEEE GLOBECOM*, vol. 6, Dec. 2003, pp. 3004–3008.

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

[38] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz, "Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios," in *Proc. 14th Conf. ONDM*, Feb. 2010, pp. 1–6.

[39] IBM ILOG CPLEX Optimization Studio V12.2, IBM, Armonk, NY, USA, 2010. Accessed at 12/20/2010. [Online]. Available: http://publib.boulder.ibm.com/infocenter/cosinfoc/v12r2/index.jsp?topic=/ilog.odms.cplex.help/Content/Optimization/Documentation/CPLEX/

[40] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bull. Amer. Soc.*, vol. 64, no. 5, pp. 275–278, Sep. 1958.

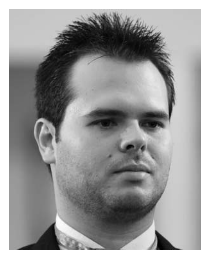
[41] R. Albert and A. L. Barabási, "Topology of evolving networks: Local events and universality," *Phys. Rev. Lett.*, vol. 85, no. 24, pp. 5234–5237, Dec. 2000.

[42] J. vom Brocke, R. T. Watson, C. Dwyer, S. Elliot, and N. Melville, "Green information systems: Directives for the is discipline," *Commun. Assoc. Inf. Syst.*, vol. 33, no. 30, pp. 509–520, 2012.

[43] X. Dong, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Low carbon emission IP over WDM network," in *Proc. IEEE ICC*, Kyoto, Japan, Jun. 5–9, 2011, pp. 1–6.

Esteban Rodriguez received the Bachelor's degree in computer science from the University of Costa Rica, San José, Costa Rica, and the M.Sc. degree in computer science from the State University of Campinas, Campinas, Brazil, in 2013.

He is currently with the Institute of Computing, State University of Campinas.



Gustavo P. Alkmim received the B.Sc. degree in computer science from Federal University of Lavras, Lavras, Brazil, in 2009 and the M.Sc. degree in computer science from the State University of Campinas, Campinas, Brazil, in 2012. He is currently working toward the Ph.D. degree in computer science at the State University of Campinas.

His research interests include cloud networks, virtualization, and Internet of things.

Nelson L. S. da Fonseca (S'92–M'95–SM'00) received the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, CA, USA, in 1994.

He is currently a Full Professor with the Institute of Computing, State University of Campinas, Campinas, Brazil. He has supervised over 50 graduate students. He has authored or coauthored over 350 papers.

Prof. Batista served as the Vice President for Member Relations of the IEEE Communications Society (ComSoc), the Director of the Latin America Region, and the Director of Online Services. He founded the IEEE Latin-American Conference on Communications and was technical chair of over ten IEEE conferences. He was the Editor-in-Chief of the IEEE Communications Surveys and Tutorials. He is a Senior Editor of the IEEE Communications Surveys and Tutorials and the IEEE Communications Magazine and a Member of the Editorial Board of Computer Networks, Peer-to-Peer Networking and Applications, Journal of Internet Services and Applications, and the International Journal of Communication Systems. He was a recipient of the Elsevier Computer Network Journal Editor of Year 2001 award, the Medal of the Chancelor of the University of Pisa (2007), and the 2012 IEEE ComSoc Joseph LoCicero Award for Exemplary Service to Publications.

Daniel Macêdo Batista (M'11) received the B.Sc. degree from Federal University of Bahia, Salvador, Brazil, in 2004 and the M.Sc. and Ph.D. degrees from the State University of Campinas, Campinas, Brazil, in June 2006 and January 2010, respectively, all in computer science.

Since 2011, he has been a Professor with the Department of Computer Science, University of São Paulo, São Paulo, Brazil. His research interests include traffic engineering, grid and cloud networks, Internet of things, and network security.