Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2014, Article ID 910658, 14 pages
http://dx.doi.org/10.1155/2014/910658

Research Article

Evaluation of Power Saving and Feasibility Study of Migrations Solutions in a Virtual Router Network

V. Eramo, S. Testa, and E. Miucci

Department of Information Engineering, Electronics and Telecommunications (DIET), Sapienza University of Rome, 00184 Rome, Italy

Correspondence should be addressed to V. Eramo; vincenzo.eramo@uniroma1.it

Received 9 November 2013; Accepted 8 January 2014; Published 30 March 2014

Academic Editor: Yan Luo

The power consumption of the network equipment has increased significantly and some strategies to contain the power used in the IP network are needed. Among the green networking strategies, the virtualization class and in particular the deployment of migrating virtual routers can lead to a high energy saving. It consists in migrating virtual routers in fewer physical nodes when the traffic decreases allowing for a power consumption saving. In this paper we formulate the problem of minimizing the power consumption as a Mixed Integer Linear Programming (MILP) problem. Due to the hard complexity of the introduced MILP problem, we propose a heuristic for the migration of virtual routers among physical devices in order to turn off as many nodes as possible and save power according to the compliance with network node and link capacity constraints. We show that 50% of nodes may be turned off in the case of a real provider network when traffic percentage reduction of 80% occurs. Finally we also perform a feasibility study by means of an experimental test-bed to evaluate migration time of a routing plane based on QUAGGA routing software.

1. Introduction

The number of Internet users today is over 2 billion and it will increase in the next few years. This will lead to a significant growth of energy consumption; just consider that in 2012 the Internet energy consumption has been the 9,8% of the overall energy consumption in the USA [1]. All the strategies proposed in the literature to reduce the power consumption of Internet or to improve the utilization of the network devices already deployed (nodes, links) are indicated as Green Networking. Within this big area, four classes of strategies can be identified according to the type of network equipment considered for reducing the power consumption and the mechanism used to obtain this gain: resource consolidation, selective connectedness, proportional computing, and virtualization [1]. The resource consolidation class is related to the dimensioning strategies to reduce the global consumption due to devices underutilized at a given time. The selective connectedness class regroups distributed mechanism allowing single pieces of equipment to go idle for some time, as transparently as possible to the rest of the network devices [2–5]. The proportional computing category, introduced in [6], starts from the assumption that a device can exhibit different energy consumption profiles as a function of its utilization level and this energy-aware profiles offer different optimization opportunities. The Virtualization class takes into account the mechanisms allowing more than one service to operate on the same hardware improving its utilization [7, 8].

We have to notice that a huge part of the energy in a network is used to supply switching nodes and data center [1]. Then it is also important to underline the fluctuating pattern of the traffic in the network in a day: we observe traffic peak during the day time and large reductions during the night time. Starting from these two assumptions and focusing on the virtualization class features we are interested in proposing and evaluating solutions based on router virtualization and migration that allow for a reduction of the power consumption in Internet networks. The proposed solutions employ the traffic daily variations: as the network traffic volume decreases in the night, virtual routers can be migrated to a smaller set of physical routers and the unneeded

physical routers can be shut down or put into hibernation mode to save power. Obviously this physical node sharing mechanism is largely known in literature as consolidation [1, 9, 10] and it is inspired by virtual machine consolidation allowing for energy saving in cloud datacenter environment [11–14]. Recently energy-aware virtual network embedding through consolidation has been studied [15–17]. These few studies propose solutions to reduce energy consumption in environments in which virtual networks are embedded in a shared substrate run by an infrastructure provider. In this paper we study the case in which one only virtual router layer is hosted on an MPLS/IP network supporting the virtual router migration. When the traffic decreases, some virtual routers can migrate towards other MPLS/IP nodes by employing the reconfiguration capacity of the MPLS layer. The problem of choosing the migrating virtual routers and the MPLS/IP nodes to host them can be modeled as an optimization problem, but due to its high complexity, we introduce a heuristic to evaluate the power saving and the effectiveness of the virtual router migration technique. We also study the impact that the virtual router migration has on the MPLS layer in terms of number of label switched paths (LSP) to be reconfigured.

The rest of the paper is organized as follows. In Section 2 we present the main technical features to implement the virtual router migration (VRM). A Mixed Integer Linear Programming (MILP) formulation of the power consumption minimization problem in a virtual router network is given in Section 3. In Section 4 we describe the proposed heuristic. Some simulation results are shown in Section 5. A feasibility study is carried out in Section 6 where we describe an experimental test-bed to evaluate the migration time of QUAGGA-based routing plane. The conclusions and future research items are finally reported in Section 7.

2. Virtual Router Migration Technique

We start considering the Virtual ROuters On the Move (VROOM) paradigm proposed in [7], in which the assumption, confirmed in practice, is that a logical router instance can migrate among physical nodes. Clearly the migration cannot take place without verifying some constraints; in particular before and after the migration all the logical configurations or states must remain the same. Therefore the IP addresses must remain the same as well as the routing protocol configurations and the overall logical topology; furthermore we want to avoid EGP/IGP reconvergence and routing protocol adjacencies loss. The possible applications of this paradigm are different; for example, it can be used to perform planned maintenance without service disruption. In our contest we want to use it for power saving strategies: when the traffic decreases significantly we want to move the virtual routers from a physical device to another in order to turn off the first machine and save power. An example of migration is reported in Figure 1. Seven Physical Elements (PE) and seven virtual routers (VR) are shown. Initially a VR is located in each PE as indicated in Figure 1(a). The migration of two VRs allows us to switch off two PEs as indicated in Figure 1(b) and to save in network power consumption.

There are three main aspects that make possible the VROOM paradigm: first of all the possibility of creating virtual router instances, then a clear separation between control and data plane, and finally the dynamic binding of virtual interfaces on the physical ones. The migration process starts creating a copy of all the information about the control plane of the virtual router (link state database, network interfaces configuration, etc.), then they are sent to the new physical device and here the data plane is cloned, finally the virtual interfaces are mapped on the new physical ones. Another technological aspect that makes possible the migration of virtual router instances is the deployment of reconfigurable Transport Network, that is to say, a protocol layer between IP and physical layer able to easily reconfigure the paths of IP flows. For example, MPLS or an optical network with its lightpaths can be seen as reconfigurable transport network. An IP/MPLS network is shown in Figure 2 before (a) and after (b) the migration of a virtual router. On the top of IP physical routers (PHYs) the virtual routers (VRs) connected by virtual links (L) are depicted. Each virtual link is mapped at the MPLS level on a label switched path (LSP), that is, a list of connected MPLS routers. For example, when the VR-1 hosted by the physical node PHY-A migrates to PHY-B the logical links L2 and L3 need to be remapped using the MPLS substrate: in particular the corresponding LSP-2 and LSP-3 change their path. After migration the physical node PHY-A which remained idle is turned off.

3. A Mixed Integer Linear Programming Formulation of the Migration Problem to Minimize the Power Consumption

For our research we have considered an IP/MPLS network on which a virtual network is mapped. Each IP physical router hosts a virtual router and each virtual link is mapped on a label switched path (LSP) of the MPLS substrate network. In this scenario, when a migration node has to be moved, the reconfigurability of the MPLS network is exploited to displace all of the LSPs on which the virtual links are mapped so that the overall virtual network topology remains the same.

The virtual nodes have to be moved so that the power consumption is minimized and both link bandwidth and node processing capacity are available to move virtual links and virtual router, respectively. Next we adapt the Mixed Integer Linear Programming (MILP) formulation proposed in [17] for our migration problem.

(1) Notation and Parameters

(i) N: number of virtual routers/physical nodes;

(ii) PHY_n $(n = 1, \ldots, N)$: nth physical nodes;

(iii) VR_i $(i = 1, \ldots, N)$: ith virtual router;

(iv) $b_{h,k}$ $(h, k = 1, \ldots, N)$: offered traffic between VR_h and VR_k;

(v) $\Theta_{\mathrm{PHY},i}$ $(i = 1, \ldots, N)$: set of physical nodes in which VR_i may migrate;

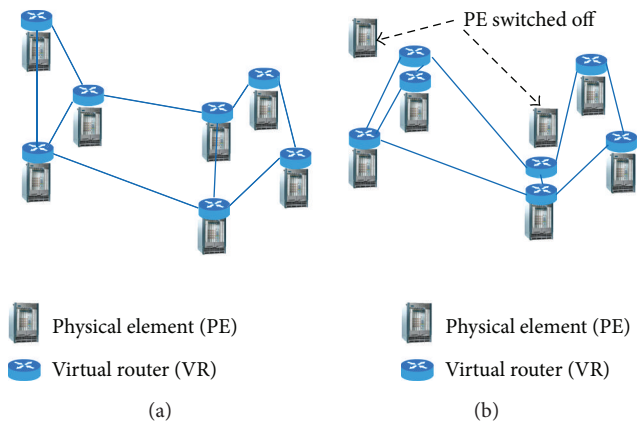

Figure 1: An example of virtual router migration for reducing the power consumption in Internet. The network is composed of seven physical elements (PE) and seven virtual routers (VR) (a). Two VRs migrated and two PEs can be switched off (b).

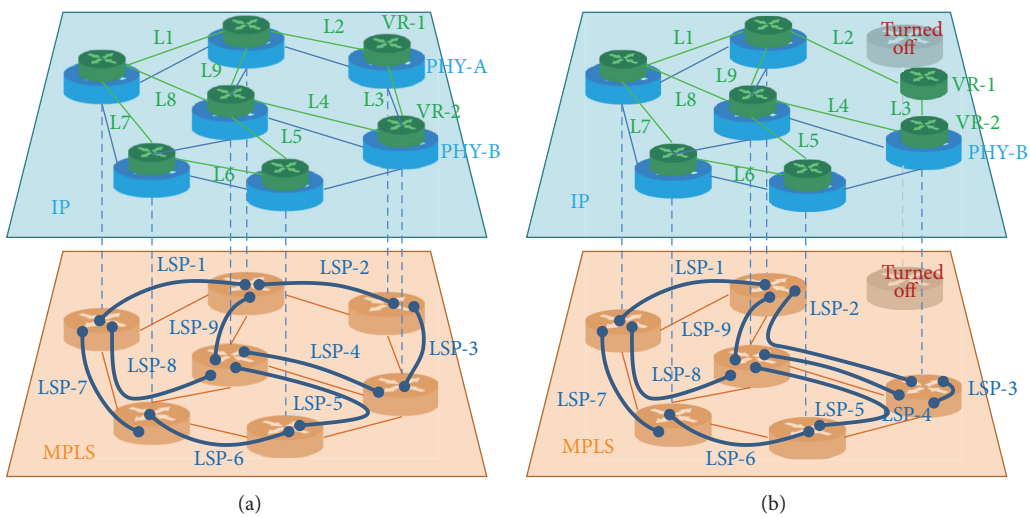

Figure 2: IP/MPLS network configuration before (a) and after (b) the migration of a virtual router.

(vi) $\lambda^{\max}_{\mathrm{PHY}_n}$ $(n = 1, \ldots, N)$: maximum traffic that PHY_n can handle;

(vii) P_{PHY_n} $(n = 1, \ldots, N)$: consumption power of the physical node PHY_n; it takes into account mainly the power consumed by the chassis and the route processor;

(viii) $P^{\mathrm{link}}_{i,j}$ $(i, j = 1, \ldots, N)$: power consumed by the Line Card in PHY_i connecting PHY_i to PHY_j;

(ix) $d_{\max}$: maximum degree of the nodes in the physical network;

(x) $c_{i,j}$ $(i, j = 1, \ldots, N)$: link bandwidth if P_{PHY_i} and P_{PHY_j} are connected;

(xi) β: link capacity overprovisioning factor.

(2) Variables

(i) α^j_i $(i, j = 1, \ldots, N)$: binary variable assuming the value 1 if VR_i is allocated to PHY_j; otherwise its value is zero;

(ii) γ_i $(i = 1, \ldots, N)$: binary variable assuming the value 1 if PHY_i is switched on after the migration; otherwise its value is zero;

(iii) $\delta_{i,j}$ $(i, j = 1, \ldots, N)$: binary variable assuming the value 1 if the link connecting PHY_i and PHY_j is switched on after the migration;

(iv) $z^{i,j}_{h,k} = \alpha^i_h \alpha^j_k$ $(i, h, j, k = 1, \ldots, N)$: binary variable introduced to make linear the optimization problem;

(v) t^j_i $(i, j = 1, \ldots, N)$: amount of bandwidth allocated between PHY_i and PHY_j to support the bandwidth demand of the virtual routers mapped on them

$$t_{i,j} = \sum_{h=1}^{N} \sum_{k=1}^{N} b_{h,k} \alpha^i_h \alpha^j_k; \tag{1}$$

(vi) $f_{h,k}^{i,j}$ $(i, h, j, k = 1, \ldots, N)$: amount of bandwidth allocated between PHY_h and PHY_k and passing through the link connecting PHY_i and PHY_j.

(3) Constraints

(i) A constraint introduces the variables $z_{i,h}^{j,k}$ to avoid the nonlinearity on the formulation. Then expression (1) becomes

$$t_{i,j} = \sum_{h=1}^{N} \sum_{k=1}^{N} b_{h,k} z_{h,k}^{i,j}. \tag{2}$$

(ii) The following source, destination, and input/output flow conservation constraints hold:

$$\begin{gathered}
\sum_{h=1\ (h\neq j)}^{N} f_{i,j}^{h,i} - \sum_{h=1\ (h\neq j)}^{N} f_{i,j}^{i,h} = -t_{i,j}, \quad i, j = 1, \ldots, N \\
\sum_{h=1\ (h\neq j)}^{N} f_{i,j}^{h,j} - \sum_{h=1\ (h\neq j)}^{N} f_{i,j}^{j,h} = t_{i,j}, \quad i, j = 1, \ldots, N \\
\sum_{i=1}^{N} f_{h,k}^{i,s} - \sum_{j=1}^{N} f_{h,k}^{s,j} = 0, \quad h, k = 1, \ldots, N;\ s \neq h;\ s \neq k.
\end{gathered} \tag{3}$$

(iii) The correlation between the variables $z_{h,k}^{i,j}, \alpha_h^i, \alpha_k^j$ has to be introduced to guarantee $z_{h,k}^{i,j}$ will be 1 only if α_h^i and α_k^j are 1:

$$\begin{gathered}
\sum_{i=1}^{N} z_{h,k}^{i,j} = \alpha_k^j, \quad j, h, k = 1, \ldots, N \\
\sum_{j=1}^{N} z_{h,k}^{i,j} = \alpha_h^i, \quad i, h, k = 1, \ldots, N \\
\alpha_h^i + \alpha_k^j - z_{h,k}^{i,j} \leq 1, \quad i, j, h, k = 1, \ldots, N.
\end{gathered} \tag{4}$$

(iv) The following link and node processing capacity constraints are introduced by the following expressions:

$$\begin{gathered}
\sum_{h=1}^{N} \sum_{k=1}^{N} f_{h,k}^{i,j} \leq (1-\beta) c_{i,j} \delta_{i,j}, \quad i, j = 1, \ldots, N \\
\sum_{i=1}^{N} \sum_{h=1(h\neq i)}^{N} \alpha_i^j b_{i,h} + \sum_{i=1}^{N} \sum_{h=1(h\neq i)}^{N} \alpha_i^j b_{h,i} \leq \lambda_{\mathrm{PHY}_j}^{\max}, \\
j = 1, \ldots, N.
\end{gathered} \tag{5}$$

(v) We assume that a virtual router is constrained to be mapped on one only physical node belonging to the physical node set in which it can migrate:

$$\begin{gathered}
\sum_{j=1}^{N} \alpha_i^j = 1, \quad i = 1, \ldots, N \\
\alpha_i^j = 0, \quad j \notin \Theta_{\mathrm{PHY},i} \ \ (i = 1, \ldots, N).
\end{gathered} \tag{6}$$

(vi) Constraints have to be introduced to guarantee that a physical node is active when at least one of its incoming or outgoing link is active:

$$\begin{gathered}
\sum_{j=1}^{N} \delta_{i,j} + \sum_{j=1}^{N} \delta_{j,i} \geq \gamma_i, \quad i = 1, \ldots, N \\
\sum_{j=1}^{N} \delta_{i,j} + \sum_{j=1}^{N} \delta_{j,i} \leq 2 d_{\max} \gamma_i, \quad i = 1, \ldots, N.
\end{gathered} \tag{7}$$

(vii) Finally we introduce a constraint that guarantees the switching off of both ingoing and outgoing link of a physical node:

$$\delta_{i,j} = \delta_{j,i}, \quad i, j = 1, \ldots, N. \tag{8}$$

(4) Objective

(i) We have to minimize the total power consumption, that is,

$$\min \left(\sum_{i=1}^{N} \gamma_i P_{\mathrm{PHY}_i} + \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_{i,j} P_{i,j}^{\mathrm{link}} \right). \tag{9}$$

4. A Migration Heuristic for Power Saving

The optimization problem we have defined in Section 3 is complex and can be solved in the case of few physical nodes/virtual nodes. For this reason we propose a heuristic, referred to as Maximum Energy Efficiency (MEE) and whose the main steps are reported in Algorithm 1. Let us introduce the following notations:

(i) $\Lambda_{\mathrm{PHY}} = \{\mathrm{PHY}_n, n = 1, \ldots, N\}$: set of all physical nodes;

(ii) $\Lambda_{\mathrm{VR}} = \{\mathrm{VR}_n, n = 1, \ldots, N\}$: set of virtual nodes;

(iii) λ_{VR_n} $(n = 1, \ldots, N)$: total traffic ingoing/outgoing in/from the VR_n;

(iv) $s_{\max} = \max_{i=1,\ldots,N} \mathrm{Card}(\Theta_{\mathrm{PHY},i})$: the maximum of the cardinalities of the sets $\Theta_{\mathrm{PHY},i}$ $(i = 1, \ldots, N)$.

The following variables are also introduced:

(i) λ_{PHY_n}: total traffic incoming/outgoing in/from all the virtual routers hosted in PHY_n;

(ii) Γ_{PHY_n}: set of PHY_n's adjacent nodes in the physical network.

The proposed heuristic is based on turning off less energy efficient nodes and the migration of VRs towards more energy efficient nodes. For this reason the energy efficiency η_{PHY_n} of the physical node PHY_n $(n = 1, \ldots, N)$ is introduced and defined as

$$\eta_{\mathrm{PHY}_n} = \frac{\lambda_{\mathrm{PHY}_n}}{P_{\mathrm{PHY}_n}^{\mathrm{tot}}}, \tag{10}$$

```
(1) /*LSP set up phase*/
(2) map each virtual router VR_n on the corresponding physical node PHY_n
(3) set up LSPs in the MPLS network carrying traffic incoming/outgoing
    to/from VR_n according to capacity constraints and a mapping policy
(4) /*Physical node turning off phase*/
(5) while Λ_PHY ≠ ∅ do
(6)     find PHY_n | η_PHY_n = min_{PHY_s∈Λ_PHY} η_PHY_s
(7)     ζ = Θ_PHY,n
(8)     while ζ ≠ ∅ do
(9)         find PHY_m | η_PHY_m = max_{PHY_s∈ζ} η_PHY_s
(10)        if (λ_PHY_m + λ_PHY_n ≤ λ^max_PHY_m) ∧ (LSPs to/from
            PHY_m carrying the traffic to/from VR_n can be set
            up according to capacity constraints and a mapping
            policy) then
(11)            map VR_n into PHY_m
(12)            turn off PHY_n
(13)            for each PHY_s ∈ Γ_PHY_n do
(14)                update η_PHY_s
(15)            end for
(16)            update λ_PHY_m and η_PHY_m
(17)            if PHY_m ∈ Λ_PHY then
(18)                Λ_PHY = Λ_PHY \ {PHY_m}
(19)            end if
(20)            for each VR_s ∈ Λ_VR do
(21)                if PHY_n ∈ Θ_PHY,s then
(22)                    Θ_PHY,s = Θ_PHY,s \ {PHY_n}
(23)                end if
(24)            end for
(25)            go to line (30)
(26)        else
(27)            ζ = ζ \ {PHY_m}
(28)        end if
(29)    end while
(30)    Λ_PHY = Λ_PHY \ {PHY_n}
(31) end while
```

Algorithm 1: Maximum energy efficiency.

where $P^{\text{tot}}_{\text{PHY}_n}$ is the sum of the node power consumption and the active links power consumption of PHY_n; in particular if the physical node has L_n active ingoing/outgoing links, $P^{\text{tot}}_{\text{PHY}_n}$ can be expressed as

$$P^{\text{tot}}_{\text{PHY}_n} = P^{\text{C,RP}}_{\text{PHY}_n} + \sum_{l=1}^{L_n} P^{\text{LC}}_{n,l}, \tag{11}$$

where $P^{\text{C,RP}}_{\text{PHY}_n}$ is the chassis and route processor power consumption of PHY_n and $P^{\text{LC}}_{n,l}$ ($l = 1, \ldots, L_n$) is the power consumption of the lth Line Card. We assume that the node power consumption and the active link power consumption are independent of the offered traffic. In such a way we model the power consumption of today's devices that consume a large amount of static power and a very limited amount of power depending on the current load [18, 19]. However the proposed heuristic can be easily extended to the case in which the node power consumption is dependent on the offered traffic [20].

Next we illustrate the main steps of the proposed heuristic. A preliminary step is needed to map the VRs on the corresponding physical nodes (PHY_s) (line 2) as well as the virtual links on the LSPs of the MPLS network according to a mapping strategy (line 3).

When the IP/MPLS network is correctly configured, MEE chooses the physical device with the least energy efficiency (line 6). Then it choses another physical node in which to migrate the virtual router hosted by the selected node. This physical node is chosen among the nodes of the set in which the virtual node can migrate ($\Theta_{\text{PHY},n}$). In particular the set ζ is introduced and containing the physical nodes belonging to set $\Theta_{\text{PHY},n}$ (line 7). These nodes are selected in order decreasing of energy efficiency (line 9) and the algorithm verifies two constraints (line 10). The first one is that the new node has the necessary capacity to manage the incoming/outgoing traffic of the VR migrating on it. The second constraint refers to the possibility of rerouting the LSPs corresponding to the virtual links afferent to the migrating VR: in particular this problem takes into account

the constraint on the residual physical links capacity on which the LSPs have to be mapped. In Section 5 we will show some results about two LSPs rerouting approaches. The first one is based in solving of a Multicommodity Flow (MCF) problem [21] enabling the splitting of LSPs on more than one path; the commodities and the capacities of the MCF problem are the flows ingoing/outgoing the VR to be migrated and the residual capacities of the MPLS network, respectively. The second approach is based on rerouting each LSP one one shortest path.

If these constraints are verified, the migration of the VR in the new PHY node takes place (line 11), the old physical node is turned off (line 12), and the energy efficiency of the adjacent nodes to the turned off node are properly updated (line 14). The amount of traffic λ_{PHY_m} managed by the new PHY node and its energy efficiency η_{PHY_m} are updated considering the contribution of the VR migrated (line 16). To make simple the heuristic, we remove the physical node PHY_m from Λ_{PHY_m} (line 18), guaranteeing that the VRs on it will not migrate towards other physical nodes. By updating the sets $\Theta_{\mathrm{PHY},s}$ $(s = 1, \ldots N)$, future migrations towards the turned off node are avoided (line 22). Finally regardless of whether the VR can be moved or not, the relative physical node is removed from Λ_{PHY} (line 30) in order to avoid a too long simulation time. The algorithm stops when all the nodes that can be turned off have been explored.

Next we report a complexity analysis of the proposed algorithm. When the logical links are remapped on the shortest paths by using the Dijkstra algorithm, the complexity can be evaluated according to the following remarks: (i) the proposed heuristic performs N steps in which at each step the least energy efficient physical node, among the switched on ones, is selected; the complexity of these operations is $O(N^2)$; (ii) the physical nodes in which to try migrating a VR are selected in decreasing order of energy efficiency; the complexity of these operations is $O(s_{\max}^2)$; (iii) the LSP rerouting is accomplished by using the Dijkstra algorithm whose complexity, when heap binary data structure is used, is $O(M \log N)$, M being the number of links of the MPLS network. According to these remarks the heuristic complexity is $O(N^2 M s_{\max}^2 \log N)$ when LSP rerouting based on shortest path is accomplished.

When the proposed heuristic is based on a MCF rerouting, the complexity is polynomial because a linear programming formulation can be given for an MCF problem.

5. Numerical Results

The power consumption saving that the proposed heuristic allows us to obtain has been evaluated when both two-level hierarchical networks and more general provider networks are taken into account. In the first case we consider a network scenario inspired by the real network of an Internet Service Provider whose structure is hierarchical [22]. The network is composed by few core nodes that are highly interconnected by means of high-capacity links. It is also composed of edge nodes that are used to interconnect aggregation nodes to core nodes. The aggregation nodes are the ones to which users are directly connected. A Digital Subscriber Line Access Multiplexer (DSLAM) and an Optical Line Termination (OLT) in PONs are typical examples of aggregation node. Each node is dual-homed; that is, it is connected to the closest pair of edge nodes to guarantee alternate paths in case of failure. We assume that the network is composed of $N = x + x \times y + x \times y \times z$ nodes where x indicates the number of core nodes, y the number of the edge nodes for each core node, and z the number of aggregation nodes for each edge node. The core network has always a ring topology and furthermore we consider a connectivity factor p which gives the probability that two core nodes not adjacent in the ring are connected with a link. In Figure 3 an example of this topology is shown when $x = 6$, $y = 3$, and $z = 4$.

Next we describe the considered traffic model, how the traffic is routed, and how the link capacities are dimensioned. Only the aggregation nodes can be source and destination of traffic which is assumed initially uniformly distributed in the range $[v_{\min}, v_{\max}]$. The offered traffic is scaled by a factor α chosen according to the dimensioning procedure next illustrated. It is routed in the IP network exploiting the shortest paths between each couple source-destination exchanging data. When all the traffic is routed in the network, the aggregated traffic on an IP virtual link is carried by one LSP in the underlying MPLS network connecting directly the end nodes of the IP virtual links. At the beginning we have as many LSPs as the number of virtual links in the IP network. The capacity of the physical link is dimensioned considering the amount of traffic that has been routed through it. In particular we assume that the network links are dimensioned with capacity values belonging to the set $\vartheta \equiv \{C_i \; i = 1, \ldots N\}$ with $C_i < C_{i+1}$ $(i = 1, \ldots, N-1)$. The link capacities and the scale factor α are chosen as follows: (i) the highest load link is equipped with the highest capacity C_N and the scale factor α is chosen so that this link carries a traffic equal to $(1-\beta)C_N$, where the factor β is the capacity overprovisioning factor; (ii) each remaining link of the network is dimensioned with the highest capacity value $C \in \vartheta$ such that the scaled traffic carried on the link is smaller than or equal to $(1-\beta)C$.

Next we evaluate the power consumption saving when a traffic reduction occurs and the virtual router migration is performed according to the heuristic introduced in Section 2. We assume that only core or edge nodes can be turned off and, due to the particular topology, when an edge node is turned off, its two adjacent edge nodes cannot be switched off to avoid isolation of the aggregation nodes connected to it. We also assume that migration of a VR can occur only in physical nodes adjacent to the physical nodes in which the VRs is hosted before the migration.

A first set of results is illustrated in Figures 4 and 5 in which we report the average number and the percentage of switched off nodes, respectively, as a function of the traffic percentage reduction γ in a hierarchical network with $x = 6$, $y = 3$, and $z = 4$. Each curve in Figures 4 and 5 is related to a specific value of the connectivity degree p. Each reported value is the mean of values obtained for five traffic realizations. When $\gamma = 0$ the link capacities are dimensioned according to the previous illustrated procedure. The offered traffic between each couple of aggregation nodes

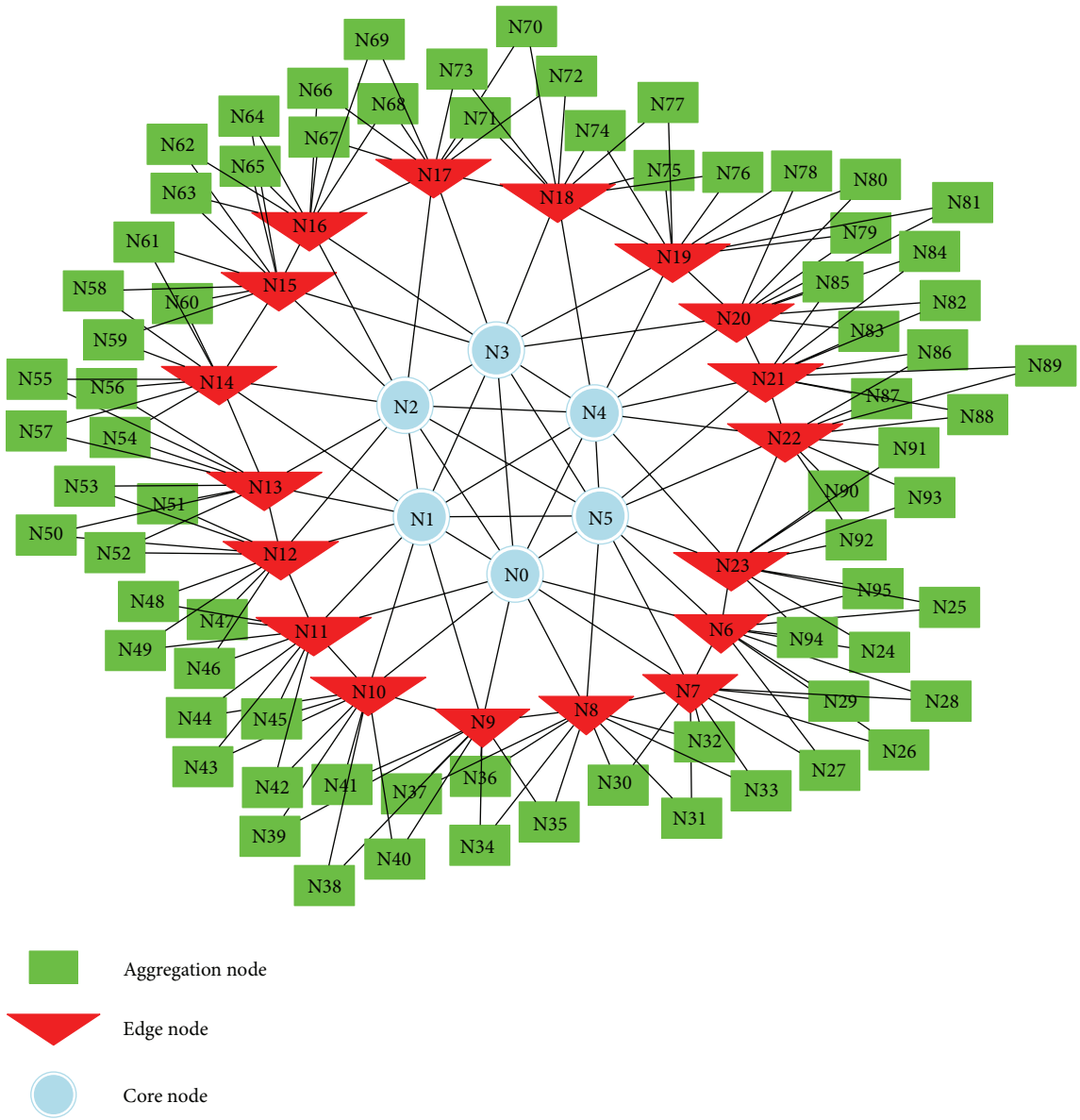

Figure 3: Hierarchical Network topology with $x = 6$, $y = 3$, and $z = 4$.

is characterized by parameter values $v_{\min} = 0{,}5$ and $v_{\max} = 1{,}5$. The following $N = 7$ capacity values are available: $C_1 = 100$ Mb/s, $C_2 = 622$ Mb/s, $C_3 = 1$ Gb/s, $C_4 = 2{,}488$ Gb/s, $C_5 = 2 \times 2{,}488$ Gb/s, $C_6 = 3 \times 2{,}488$ Gb/s, and $C_7 = 4 \times 2{,}488$ Gb/s. The spare capacity is determined by a parameter value $\beta = 0{,}2$. We also assume that when a traffic reduction happens and the virtual router migration procedure is activated, a migration is performed when it is guaranteed that the processing load of each core (edge) node is not higher than the one of the highest load core (edge) node in the traffic condition $\gamma = 0$.

The power consumption values used for the chassis, the route processor, and the Line Cards in edge and core nodes are the ones measured by the authors in [23] and reported in Table 1. We show in Figures 4 and 5 some results that compare two strategies for the remapping of virtual links on new LSPs after the migration of a virtual router. The first one is based on the solution of a Multicommodity Flow (MCF)

Table 1: Values of power consumed by the physical router components.

	Power consumption
Chassis and route processor	
Edge router	220 W
Core router	400 W
Line Cards	
Fast Ethernet	26 W
OC-12 622.08 Mbps	18 W
Gigabit Ethernet	30 W
OC-48 2.488 Gbps	70 W

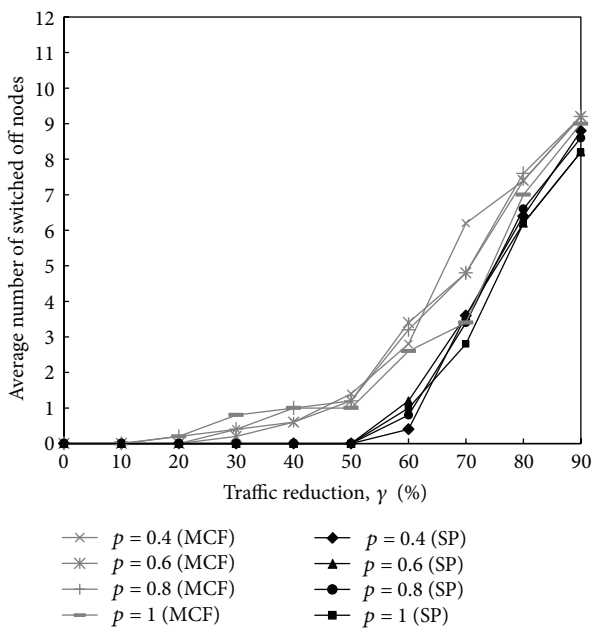

Figure 4: Average number of switched off nodes obtained using MEE Heuristic with MCF and Dijkstra rerouting approach and in the case of the Network 6-3-4.

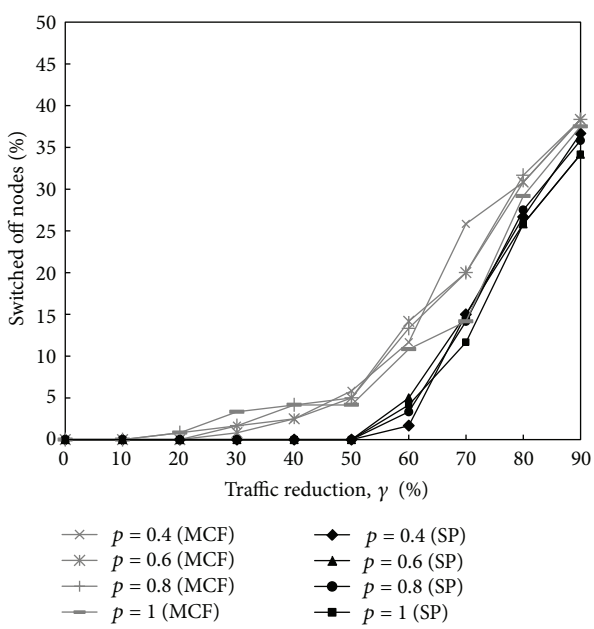

Figure 5: Percentage of switched off nodes obtained using MEE Heuristic with MCF and Dijkstra rerouting approach and in the case of the Network 6-3-4.

problem in order to remap the virtual links; in this case a traffic demand between two virtual routers can be rerouted on more than one path in order to better use the resources on the physical link. In the second strategy it is remapped on the Shortest Path (SP) chosen applying the Dijkstra algorithm. It is easy to see from Figures 4 and 5 that more nodes can be switched off when the traffic decreases. That is obviously due

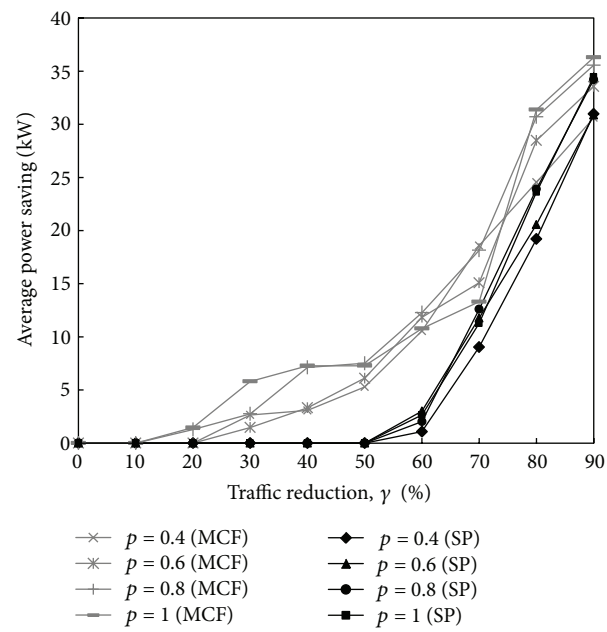

Figure 6: Power consumption saving obtained using MEE Heuristic with MCF and Dijkstra rerouting approach and in the case of the Network 6-3-4.

to the higher availability of resource in the links and nodes. The performance is little dependent on factor p and this is due to the way in which we scale the traffic on the links: the greater the value of p is, the greater the factor α is and this leads to a more connected network but with a higher level of traffic load. The curves related to the shortest path rerouting approach show lower performance and this is due to the fact that in this case a single path is chosen for each LSP to reroute with respect to the case of MCF in which a traffic demand between two virtual routers can be split on more than one path. As a matter of example we notice from Figure 4 that when $\gamma = 80\%$, seven and six nodes can be switched off in the case of rerouting strategies based on MCF and SP, respectively. That leads to switch off the 33% and 27% of nodes as shown in Figure 5. Notice that we evaluate the percentage of switched off nodes only taking into account the nodes possible to switch off.

The power saved turning off the nodes is shown in Figure 6 and the same values in percentage compared with the total power consumption of the overall network is reported in Figure 7.

Finally we report in Figure 8 the average number of LSPs that is necessary to reroute as a result of the shutdown of the nodes. As expected the curves shown in this figure follow the trend of those in Figures 4 and 5 since they are quantities directly dependent. In particular the shortest path rerouting approach, using Dijkstra algorithm, needs to reconfigure a lower number of LSPs but, at the same time, it turns off fewer physical nodes than the MCF rerouting approach. This second strategy allows us to turn off nodes already when the traffic load is around 30% and this is paid with an increasing

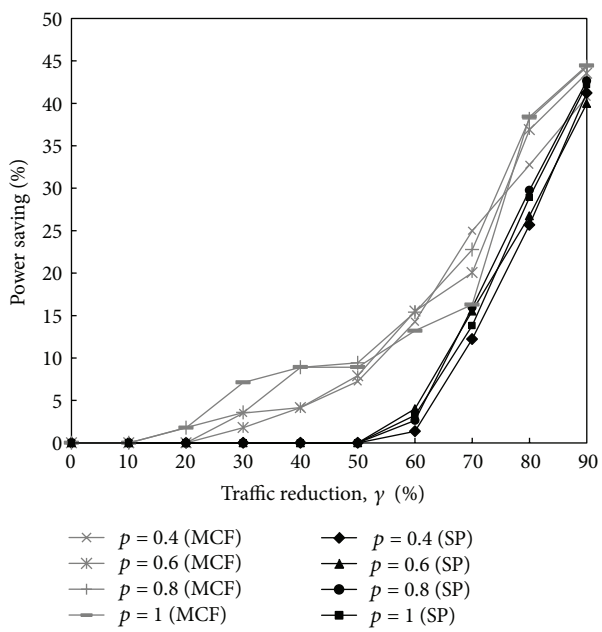

Figure 7: Percentage of power consumption saving obtained using MEE Heuristic with MCF and Dijkstra rerouting approach and in the case of the Network 6-3-4.

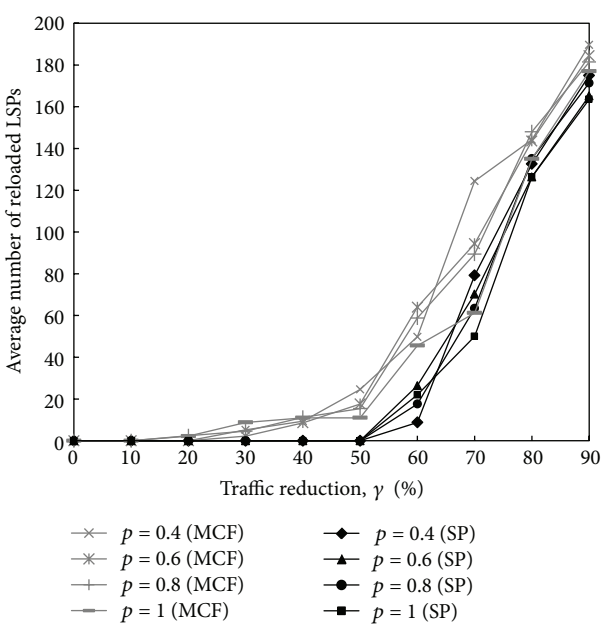

Figure 8: Number of Reloaded LSPs using MEE Heuristic with MCF and Dijkstra rerouting approach for the Network 6-3-4.

number of paths that need reconfiguration in the MPLS domain.

Next we evaluate the power consumption saving in the case of a real provider network. We consider the EBONE network [5] composed of 159 nodes and 614 links and reported in Figure 9(a). We evaluate the power consumption saving when the proposed heuristic with LSP rerouting based on shortest path is applied. We assume that (i) all of the nodes are core nodes; (ii) the constraint that a VR hosted in leaf node cannot migrate; (iii) the constraint that a VR can migrate only in physical nodes adjacent to the physical node in which it is initially hosted; (iv) the traffic is generated among any node couple according to the procedure before described with $\nu_{\min} = 0{,}5$ and $\nu_{\max} = 1{,}5$; (v) the availability of Gigabit Ethernet Line Cards and in particular the links can be dimensioned with capacity values equal to nC_{GE} ($n = 1, \ldots, 10$) with $C_{\mathrm{GE}} = 1$ Gb/s.

We report in Figures 9(b), 9(c), and 9(d) and in white color, the nodes that the proposed migration heuristic allows us to obtain in the case of $\gamma = 0\%$, $\gamma = 60\%$, and $\gamma = 80\%$, respectively. Average number and percentage of switched off nodes, power consumption saving and relative percentage, and number of LSPs rerouted are reported in Table 2 for the EBONE network in the case of traffic reduction γ from 0 to 80%. We can notice that a big power saving can be obtained. For instance when $\gamma = 80\%$, as much as 60% of nodes can be switched off.

6. Experimental Test-Bed for the Migration Time Evaluation of a Quagga-Based Routing Plane

We illustrate an experimental test-bed to evaluate the migration time of a routing plane based on QUAGGA routing software. The operation mode correctness of the OSPF routing protocol is also verified. The realized test-bed allows for an evaluation of the migration time as a function of the number of nodes of an emulated network. The section is organized as follows. The software router architecture and the used software are described in Section 6.1. The test-bed realized to evaluate the routing plane migration time is illustrated in Section 6.2 where the main numerical results are also shown.

6.1. Software Router Architecture. There are three main aspects that make possible the virtual router migration paradigm [7]: first of all the possibility to create virtual router instances, then a clear separation between routing and data plane, and finally the dynamic binding of virtual interfaces on the physical ones. A Software Router (SR) equipped with software modules for the implementation of the virtual router migration paradigm is illustrated in Figure 10. It is based on the following main software modules: *Linux* Operating System, *Linux Containers (LXC)* [24] virtualization software, *QUAGGA* [25] routing software, and the *Linux bridge* [26] software. Thanks to the Linux Containers we are able to divide the resources of a Personal Computer (PC) among different virtual routers instances. Each of them has an independent control plane to execute applications, configurations, routing protocols instances, Routing Information Base (RIB), and also an own data plane managing interfaces and Forwarding Information Base (FIB). The isolation of different virtual routers makes possible the migration of one of them transparently with respect to the others. The virtual routers are activated and deactivated by the *VR-Manager*. The virtual interfaces of the virtual routers are mapped on the host physical ones thanks to the *Linux bridge* software that makes

Table 2: Number and percentage of switched off nodes, power consumption saving and relative percentage, and number of LSPs rerouted are reported for the EBONE network in the case of traffic reduction γ from 0 to 80%.

γ	Number of turned off nodes	Percentage of turned off nodes	Power consumption saving (W)	Percentage of power consumption saving	Number of rerouted LSPs
0%	30.8	23%	21332	18%	92.8
20%	50.6	38%	35336	30%	164.8
40%	68.8	52%	49444	42%	280
60%	77.6	59%	58376	50%	364
80%	79.8	60%	60624	52%	384

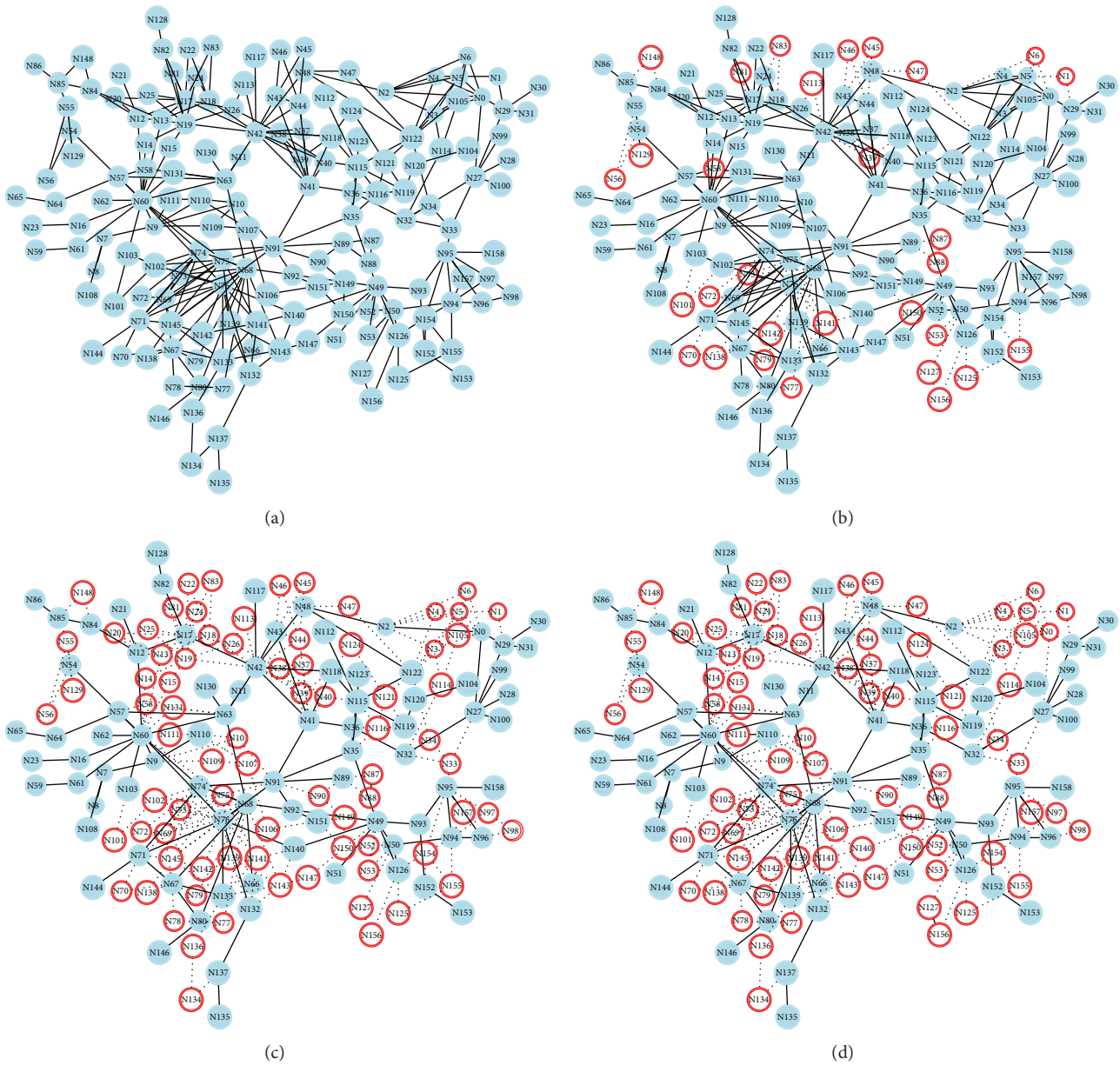

Figure 9: EBONE Network Topology (a); virtual router migration and switching off of physical nodes when $\gamma = 0\%$ (b), $\gamma = 60\%$ (c) and $\gamma = 80\%$ (d), respectively.

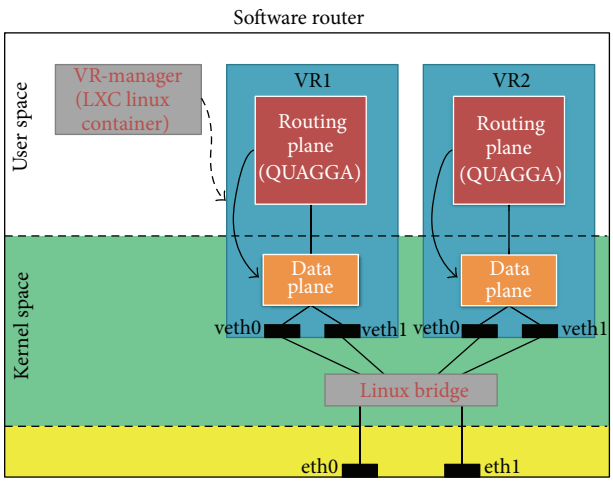

Figure 10: Software Router Architecture supporting virtualization.

possible setting dynamically the association between virtual routers interfaces and physical ones. Each VR is equipped with QUAGGA routing software [25]. To allow the routing plane to migrate, we have realized a Quagga's patch for the OSPF routing plane migration. It is mainly composed of three software modules: the first one copies all of the routing information (LSA database, configuration files, etc.) to be migrated in a text file appropriately encoded; (ii) the second one manages the transferring of the text file containing the routing information between the two SRs in which the migration occurs; the third one decodes the text file and recreates the OSPF data structure in the VR in which the migration occurred.

The key instants of virtual router migration process from a SR-A to a SR-B are shown in Figure 11. When the migration event starts a virtual container is opened in the destination SR. In the interval $[t_0, t_1]$ all of the routing information is copied in the text file. With the aim to transfer the minimum amount of information, we establish that all the virtual routers in the network have the same Quagga's executable files. For this reason only the information contained in both the Quagga's configuration file and the Link State Database (LSDB) data structure containing the Link State Advertisements (LSA) is appropriately encoded and stored in the text file to be transferred from SR-A to SR-B. This phase takes a time depending on the network dimension: as the number of nodes increases, the number of LSAs and thus the dimension of the LSDB increases, so the transferring of this data will take a longer time when the network dimension increases. During the copy of the LSDB it is necessary that no other subprocess accesses the memory of LSDB to modify it, so in this step it is frozen and the update LSAs that the VR may receive are discarded. In the interval $[t_1, t_3]$ ($[t_2, t_4]$ in SR-B) the text file is transferred from SR-A to SR-B. In this phase the update LSAs received are stored in the LSDB in SR-A but are not acknowledged; in such a way the neighbour router which has generated the update LSA is induced, according to the OSPF protocol operation mode, to retransmit it until the migration process is completed. Then

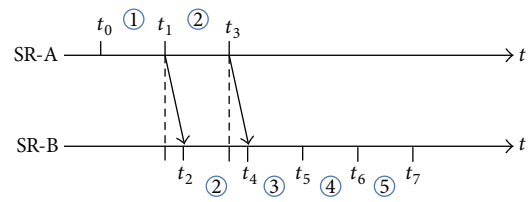

Figure 11: Main steps of virtual router migration between SR-A and SR-B.

SR-B receives the update LSA when retransmissions occur. In the interval $[t_4, t_5]$, the text file received by SR-B is decoded and from this information both the Quagga's configuration file and the LSDB data structure are rebuilt. In this interval the shortest path tree is calculated starting from the network information collected in the LSDB. At this point the Quagga's zebra demon clones the VR's data plane in SR-B and updates the routing table. Also in these two last intervals VR in SR-A goes on to update the LSDB, without acknowledging any LSA received. Finally in the interval $[t_6, t_7]$, IP addresses are assigned to the virtual interfaces of the VR and these last ones are mapped on the SR-B ones thanks to the Linux Bridge software.

6.2. Test-Bed for the Evaluation of the Routing Plane Migration Time. The realized test-bed is shown in Figure 12 and is composed of a DELL PowerConnect 5524 switch with 24 ports GbE and three PC DELL Optiplex 990 with the following hardware features:

(i) CPU Intel Core i7 2600 @ 3.40 GHz,

(ii) SDRAM DDR3 8 GB,

(iii) Hard disk SATA II 1TB, 7200 rpm,

(iv) 2 Network interfaces (only one used for measures).

Each PC is equipped with Linux Ubuntu 10.10 32-bit OS (kernel version 2.6.38). The SR-A and SR-B are the ones involved in the migration process. VR-A and VR-B are executing in SR-A and SR-B, respectively. The testing PC has the function to generate routing and data traffic. The first one allows for the emulation of any network topology in the SR-A and SR-B. The second one allows the migration process to be disturbed by injecting traffic in the network links.

The following software tools have been used.

(i) *LSA Generator* [27] is an open source software installed in the testing PC. It can generate and send all types of LSA defined within the OSPF protocol. A text file, describing the network topology, is given as input to LSA generator that by injecting appropriate LSAs towards SR-A and SR-B allows any network topology to be emulated in them.

(ii) *RUDE* (*Real-time UDP Data Emitter* [28]) is an open source UDP traffic generator in which the packet rate and dimension can be set. The generated traffic is analyzed in another node of the network using the **CRUDE** (*Collector for RUDE* [28]) software.

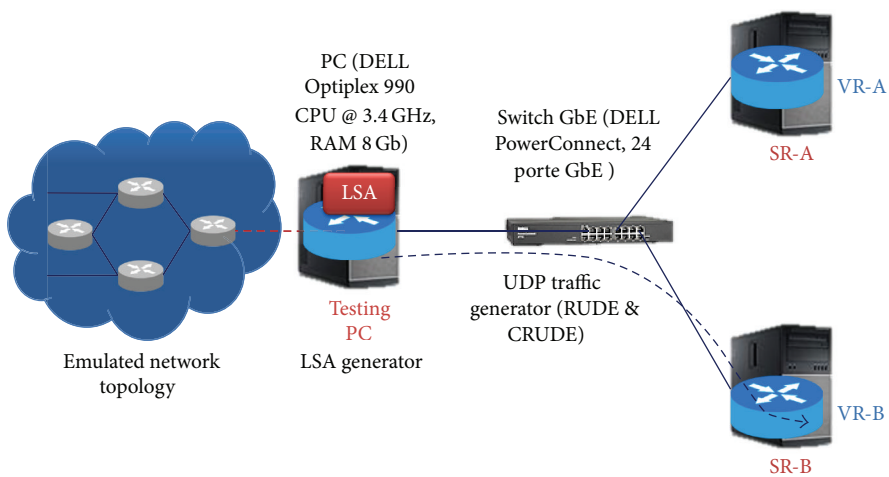

Figure 12: Test-bed for the evaluation of the migration time.

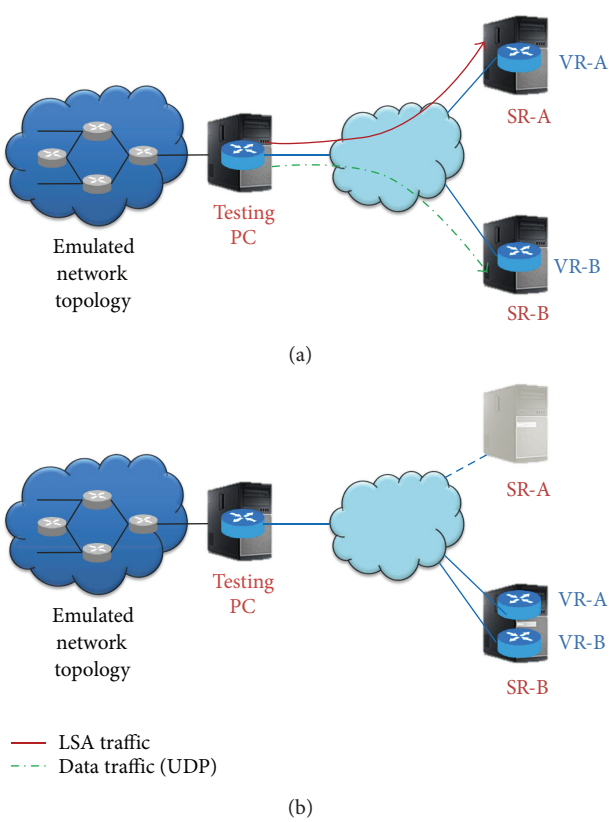

Figure 13: Test-bed scenario before (a) and after (b) the migration of VR-A from SR-A to SR-B.

(iii) *Tshark* [29] monitors and analyses the OSPF routing traffic incoming in the different virtual routers we consider in the simulation.

By means of the introduced test-bed, we have evaluated the Quagga's routing plane migration time and we have verified the operation mode correctness of the OSPF routing protocol when a migration occurs. The test is performed in two phases.

(i) Phase-1. As indicated in Figure 13(a), the testing PC generates the emulated network topology sending LSA towards the SR-A so that the SR-A's LSDB is updated. All of the performed tests are based on fully meshed router networks with each router connected to each other through a different transit network [30]. Hence the testing PC sends a Router LSA and a Network LSA for each router and transit network of the emulated network topology, respectively. It also sends UDP traffic towards the SR-B so that the migration process is disturbed.

(ii) Phase-2. When the emulated network topology is acquired by SR-A, the virtual router VR-A is moved from SR-A to SR-B as shown in Figure 13(b). We have evaluated the various components of the migration time by inserting timers in the Quagga's source codes. The test has been repeated several times varying the number N of nodes (routers and transit networks) from 50 to 500 and then varying the bit rate f_{UDP} of UDP traffic from 1 Mbps up to 700 Mbps, carried in links with 1 Gbps capacity. Finally in this phase we have also verified the operation mode correctness of the OSPF routing protocol.

The migration time is shown in Figure 14 as a function of the number N of nodes of the emulated OSPF network. Moreover the figure provides the information related to the time required by the different migration steps: copy of the routing plane in SR-A ($t_1 - t_0$), transfer of encoded data from SR-A to SR-B ($t_4 - t_2$), and installation on the new physical device SR-B ($t_6 - t_4$) of the routing plane. In this first case study the UDP data traffic rate f_{UDP} is equal to zero. As we can observe the overall migration time grows linearly with the number N of nodes of the emulated network topology and thus with the amount of data of the routing plane that is necessary to transfer. From the results we can notice low

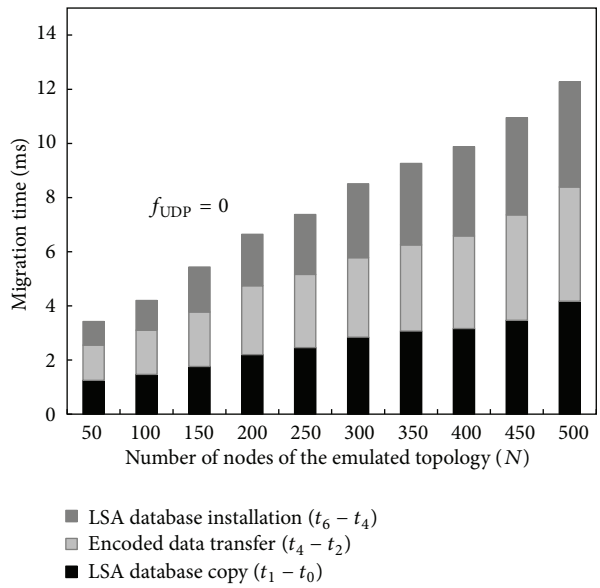

Figure 14: Migration time components as a function of the number N of nodes in the emulated network. The UDP traffic rate f_{UDP} is equal to zero.

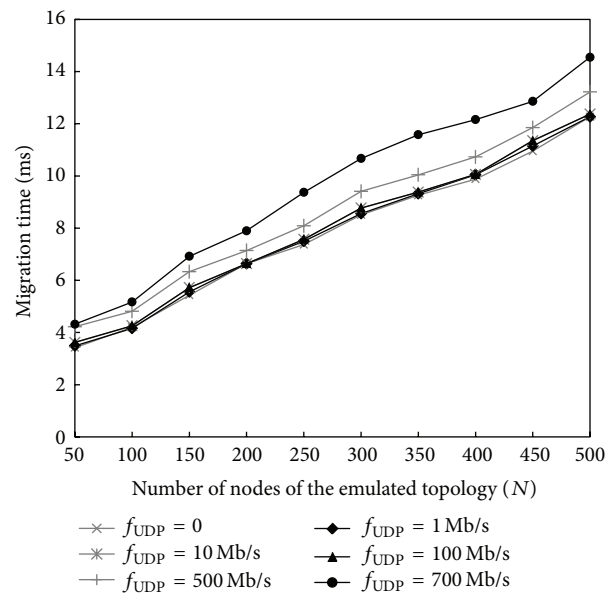

Figure 15: Migration time as a function of the number N of nodes in the emulated network. The UDP traffic rate f_{UDP} is varied from 0 to 700 Mb/s.

migration time that, even in the case of network with $N = 500$ nodes, is smaller than 12 ms.

The migration time is shown in Figure 15 varying the number N of nodes in the emulated network topology, when we consider the UDP data flow sent from the testing PC to the SR-B. Increasing the bit rate, the UDP packets flow tends to saturate the link capacity ingoing the SR-B and this leads to a growing delay in the transferring component $(t_4 - t_3)$ of the routing information from SR-A to SR-B. Even the installation delay component $(t_6 - t_4)$ of the routing plane is increased due to the fact that the SR-B's CPU has to process the UDP packets received. As we can expect the whole migration time grows when the link is stressed by the UDP traffic. For example when the link ingoing the SR-B is 70% loaded, the migration time is increased by 20%.

We have also verified if the migrant virtual router maintains, during the migration, the adjacencies with the testing PC. In fact according to the OSPF protocol, an IP router sends HELLO packets to inform its neighbours that it is active. Two timeouts are used. The first one is the HELLO timeout that establishes the time distance between HELLO packets. The second one is the DEAD timeout that represents the time after which a neighbour is declared not reachable if HELLO packets are not received. In our tests we have considered a HELLO timeout equal to 10, 5, 2, and 1 second and a DEAD timeout set to 4 times the HELLO timeout. We have verified that the VR-A never loses the adjacencies with the testing PC and the VR-B. That is due to the fact that the overall migration time is much lower than the DEAD timeout. Due to the lack of a retransmission mechanism of update LSAs in *LSA Generator*, we are not able to verify the operation mode correctness of the OSPF routing protocol when a topological change occurs and update LSAs are lost. The only consideration we can do is that the very low migration times leads to a low probability that an update LSA is lost.

7. Conclusion

In this paper a heuristic, called Maximum Energy Efficiency (MEE), for virtual router migration in an IP/MPLS network is proposed. The effectiveness of the MEE heuristic has been evaluated in a hierarchical network with core, edge, and aggregation nodes. The simulation results show that the 33% and 27% of nodes can be switched off when the traffic is reduced of 80% in the case of rerouting strategies based on MCF and Dijkstra, respectively. This higher power saving is paid with a higher number of LSPs to be reroute on the MPLS network. We have also evaluated the power consumption saving in real provider networks. In particular the proposed heuristic has been applied to evaluate the power saving in the EBONE network composed of 159 nodes and 614 links. We have shown that a 60% power saving can be reached for a 80% traffic reduction.

Finally the routing plane migration time of a Software Router equipped with QUAGGA routing software has been evaluated in an experimental test-bed made of three PCs: a testing PC and two other PCs between which a virtual router migration is trigged off. The testing PC is able to emulate a network topology and allows for an evaluation of the migration time as a function of the number of nodes in the emulated network. We have evaluated a migration time less than 12 ms in the case of an emulated network of 500 nodes and when the network links are 70% loaded.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The research leading to these results has received funding by MIUR with PRIN 2009 SFINGI (Software router to Improve Next-Generation Internet).

References

[1] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A survey of green networking research," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 3–20, 2012, First Quarter.

[2] K. J. Christensen, C. Gunaratne, B. Nordman, and A. D. George, "The next frontier for communications networks: power management," *Computer Communications*, vol. 27, no. 18, pp. 1758–1770, 2004.

[3] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, "An energy saving routing algorithm for a green OSPF protocol," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM '10)*, San Diego, Calif, USA, March 2010.

[4] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, "An OSPF enhancement for energy saving in IP networks," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '11)*, pp. 325–330, Shanghai, China, April 2011.

[5] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, and A. V. Vasilakos, "An OSPF-integrated routing strategy for QoS-aware energy saving in IP backbone networks," *IEEE Transactions on Network and Service Management*, vol. 9, no. 3, pp. 254–267, 2012.

[6] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, 2007.

[7] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual routers on the move: live router migration as a network-management primitive," in *Proceedings of the ACM Conference on Data Communication (SIGCOMM '08)*, pp. 231–242, Seattle, Wash, USA, August 2008.

[8] X. Chen and C. Philips, "Virtual router migration and infrastructure sleeping for energy management of IP over WDM networks," in *Proceedings of the International Conference on Telecommunications and Multimedia*, Chania, Greece, July 2012.

[9] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, Barcelona, Spain, April 2006.

[10] N. Mosharaf, K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 783–791, Rio de Janeiro, Brazil, April 2009.

[11] S. Srikantaiah, A. Kansal, and F. Zao, "Energy aware consolidation for cloud computing," in *Proceeding of the 2008 Conference on Power Aware Computing and System*, USENIX Association, 2008.

[12] B. Guenter, N. Jain, and C. Williams, "Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning," in *Proceedings of the 26th IEEE Conference on Computer Communications (INFOCOM '11)*, pp. 1332–1340, Shanghai, China, April 2011.

[13] Q. Huang, F. Gao, R. Wang, and Z. Qi, "Power consumption of virtual machine live migration in clouds," in *Proceedings of the 3rd International Conference on Communications and Mobile Computing (CMC '11)*, pp. 122–125, Qingdao, China, April 2011.

[14] H. Liu, H. Jin, C. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster Computing*, vol. 16, no. 2, pp. 249–264, 2013.

[15] S. Su, Z. Zhang, X. Cheng, Y. Wang, Y. Luo, and J. Wang, "Energy-aware virtual network embedding through consolidation," in *Proceedings of the IEEE INFOCOM Workshop on Communications and Control for Sustainable Energy Systems: Green Networking and Smart Grids (INFOCOM '12)*, Orlando, Fla, USA, April 2012.

[16] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. D. Meer, "Energy efficient virtual network embedding," *IEEE Communications Letters*, vol. 16, pp. 756–759, 2012.

[17] J. F. Botero and X. Hesselbach, "Greener networking in a network virtualization environment," *Computer Networks*, vol. 57, pp. 2021–2039, 2013.

[18] L. Chiaraviglio, D. Ciullo, M. Mellia, and M. Meo, "Modeling sleep mode gains in energy-aware networks," *Computer Networks*, vol. 57, pp. 3051–3066, 2013.

[19] A. Adelin, P. Owezarski, and T. Gayraud, "On the impact of monitoring router energy consumption for greening the Internet," in *Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (Grid '10)*, pp. 298–304, Bruxelles, Belgium, October 2010.

[20] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti, and S. Singh, "The potential impact of green technologies in next-generation wireline networks: is there room for energy saving optimization?" *IEEE Communications Magazine*, vol. 49, no. 8, pp. 80–86, 2011.

[21] C. Barnhart, N. Krishnan, and P. H. Vance, "Multicommodity flow problems," in *Encyclopedia of Optimization*, pp. 2354–2362, Springer, 2009.

[22] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, June 2009.

[23] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 457–465, April 2008.

[24] LXC Linux Containers, http://lxc.sourceforge.net.

[25] Quagga Routing Software, http://www.quagga.org.

[26] Linux Bridge, http://sourceforge.net/projects/bridgelinux/.

[27] LSA Generator, LSA Generator Software SPOOF, http://www.cs.ucsb.edu/rsg/Routing/download.html.

[28] RUDE and CRUDE, http://rude.sourceforge.net.

[29] Wireshark, http://www.wireshark.org.

[30] V. Eramo, M. Listanti, N. Caione, I. Russo, and G. Gasparro, "Optimization in the shortest path first computation for the routing software GNU Zebra," *IEICE Transactions on Communications B*, vol. E88, no. 6, pp. 2644–2649, 2005.