Energy-Efficient Virtual Link Reconfiguration for Off-Peak Time

Ebrahim Ghazisaeedi, Changcheng Huang
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
{eghazisaeedi; hunag}@sce.carleton.ca

***Abstract*—Energy consumption in Information and Communication Technology (ICT) is a large portion of total energy consumed in industrial countries. Virtualized Network Environment (VNE) has recently emerged as a solution to address the challenges of future Internet. It is essential to develop novel techniques to reduce VNE's energy consumption. In this paper, we propose an energy saving method that optimizes VNE's energy consumption during the off-peak time. This method reconfigures mapping for some of the embedded virtual links in the off-peak period. The proposed strategy enables providers to adjust the level of the reconfiguration, and accordingly control probable traffic disruptions due to the reconfiguration. This problem is formulated as a Binary Integer Linear Program (BILP). Since the defined BILP is $\mathcal{NP}$-hard, a novel heuristic algorithm is also suggested. The proposed energy saving methods are evaluated over random VNE scenarios. The results confirm the defined solutions are able to save notable amounts of energy during off-peak period, while still accommodating off-peak traffic demands of involved virtual networks.**

I. INTRODUCTION

Several reports from different Information and Communication Technology (ICT) organizations over the world confirm the increasing demand of energy in this technology, which is a major concern for future Internet. In the case that no green technology will be deployed in communication networks, Global e-Sustainability Initiative (GeSI) predicts 35.8TWh energy consumption for European telecom operators in 2020 [1].

Recently, virtualization has emerged in communication networks. Virtualized Network Environment (VNE) supports the coexistence of multiple Virtual Networks (VNs) over a single substrate network. VNE embedding process maps requested virtual nodes and links onto substrate nodes and paths, respectively. Network virtualization has been regarded as a promising technology for flexibly utilizing shared network resources. Consequently, the corresponding solutions to energy saving in this type of networks become essential.

In fact, virtual networks' traffic loads change over time. Virtual networks might be highly utilized during a period of time (peak time, e.g. day hours), while they are under-utilized during another notable period of time (off-peak time, e.g. night hours). Traffic variations in virtual networks correspondingly change substrate network's utilization. The reports for 40 North American and 25 European network providers reveal 60% difference between the peak and the minimum off-peak traffic rate over their substrate network [2]. However, today's substrate networks are provisioned to support VNs' peak time traffic demands, with some additional over-provisioning accommodating unexpected traffic rates [2]. The substrate network's elements are always switched on, neglecting the traffic behaviour.

Network providers could determine the off-peak time period of the substrate network and traffic demands of each VN in that period, through given information by VNs' customers, or network traffic prediction techniques. During the off-peak period, it is possible to reduce VNE's energy consumption by reconfiguring mapping of the already embedded VNs according to their decreased traffic demands. In this context, virtual networks are accepted and embedded onto the substrate network by a normal (not energy-efficient) VNE embedding process to accommodate the peak traffic behaviour. The reconfiguration technique is run during normal network operations, upon networks go from the peak period to the off-peak period, to save energy in the off-peak period. However, when the traffic load changes from the peak level to an off-peak level, some traffic flows that last in the both time periods might suffer from traffic disruptions imposed by applying the reconfiguration [2]. Besides, reconfiguring mapping of embedded VNs may require additional signalling traffic that is necessary for notifying all the involved routers [3]. This may introduce significant work load for the signalling controller especially when the reconfiguration tries to make changes to a large number of nodes at the same time. Consequently, it may not be a good practice to reconfigure mapping of every embedded virtual node/link.

In this paper, we assume substrate and virtual networks are homogeneous. All the substrate and virtual nodes are assumed to be switches/routers to reflect network environment. This is the case in the most of existing related research studies. We intend to reduce VNE's energy consumption during the off-peak period, by reconfiguring mapping for some of the embedded virtual links for that period. We do not reconfigure mapping of the embedded virtual nodes to prevent possible traffic disruptions due to moving embedded access/edge virtual nodes. An access/edge virtual node connects the VN to other networks (on different substrate networks). So, moving such a virtual node to other part of the substrate network might be infeasible. However, it is still probable to save energy in capable intermediate substrate nodes and substrate links. An intermediate substrate node is a physical node that no virtual node is embedded onto it, while it only forwards the traffic of other virtual nodes. In this regard, we define a stress rate for an intermediate substrate node. Accordingly, a solution is proposed to minimize total energy consumption of intermediate substrate nodes and substrate links during the off-peak time. This method *might* set less stressed intermediate substrate nodes and their respective substrate links into sleep mode for the off-peak time. We re-map a virtual link *if and only if* we sleep at least one intermediate substrate node over its embedded path. This strategy enables the providers to change the level of the reconfiguration by adjusting the stress rate's threshold. Therefore, they can control the possible traffic interruptions of the reconfiguration. Clearly, there is a trade-off between energy saving level and the possible traffic interruptions.

We formulate a Binary Integer Linear Program (BILP)

978-1-4799-5952-5/15/$31.00

for this problem. Since the formulated BILP is $\mathcal{NP}$-hard, a heuristic algorithm is also suggested. The simulation results confirm the heuristic algorithm can achieve closely to the optimum points set by the formulated BILP, and it is scalable to large network sizes. This heuristic could be implemented in Software-defined Networking (SDN) controllers to optimize network energy consumption during the off-peak period.

This paper is organized as follows: The related works and our contributions in this paper are discussed in Section II. A power model is studied in Section III. The BILP is defined and formulated in Section IV and the suggested heuristic is discussed in Section V. The performance of the BILP as well as the heuristic algorithm are evaluated in Section VI. The paper will conclude in Section VII.

II. RELATED WORKS

The literature is rich in regard to VNE and energy saving methods for communication networks. But, they have been studied separately. There are few very recent works that concerned about energy consumption in VNE. We review them in this section.

Four papers [4]–[7] tried to save energy in VNE by making its embedding procedure energy-aware. This has been done by modifying the link weights based on physical links' power consumption in [4], and consolidating VNs to the smallest number of substrate network elements in [5]–[7]. Nonetheless, when VNE embedding algorithms are modified to map the resources energy-wise, several extra constraints will be added to the embedding procedure. Accordingly, the embedding algorithm has a smaller set of physical node and link candidates to choose from. This decreases the network's admittance ratio for new virtual network requests, which is not cost efficient for the providers. The main economic objective of providers is to reject the minimum number of virtual network requests. Thus, these solutions are not profitable for them in long term.

Authors in [8] offered a heuristic algorithm that reconfigures mapping of accepted VNs at each embedding phase to save energy. This approach has the same problem of energy-efficient embedding methods. Because reconfiguring mapping of accepted VNs at each embedding phase for their life time, still might make capacity bottlenecks that decrease network admittance rate for new VNs. Moreover, their heuristic assumes each virtual link is only mapped onto a single physical link, which is not a real assumption. Our previous work [9] proposed an algorithm to maximize the number of sleep mode physical links during the off-peak period of VNE. It reroutes the off-peak traffic of already embedded virtual links to other already allocated traffic capacities. It does not change mapping of VNs. Assuming fixed VN mapping prevents us to reroute a VN's off-peak traffic to substrate links that no traffic capacity is allocated in them to that particular VN. This decreases the level of energy we could save. Authors in [10] suggested a method to move embedded virtual machines (VMs) onto servers, to other servers. Their solution is run over time periodically to consolidate the VMs. Nevertheless, moving allocated VMs and setting the servers into sleep mode is expensive, if it is not impossible, due to two reasons. First, normally large amount of data is distributed over large number of servers, and it is not profitable/possible for the providers to move data of a server to another one. Second, waking up servers from sleep mode (in the case of unexpected demand, or going back to peak time), imposes hundreds of milliseconds delay to the tasks that might violate Service Level Objectives (SLOs) [11]. Besides, their solution does not enable the providers to adjust the level of the reconfiguration, and control the possible traffic disruptions.

Since we do not reconfigure mapping of virtual nodes in this paper, the problem might seem similar to the classic energy-efficient routing problems for multi-layer network design. Most of the existing approaches for energy-efficient multi-layer routing problem, e.g. [12], [13], route every requested traffic demand between fixed nodes, considering energy consumption. This is quite different from reconfiguring mapping of VNs, in terms of the possible traffic disruptions that might happen during the reconfiguration. To the best of our knowledge there are few papers that studied classic energy-efficient reconfiguration problem and considered the possible traffic interruptions. Authors in [14] proposed a Mixed Integer Linear Program (MILP) that reroutes off-peak traffic in order to minimize energy consumption during the off-peak period. However, their approach, similar to [2], [3], [9], assumes fixed VN mapping to decrease the possible traffic disruptions. Their method reroutes the off-peak traffic of every virtual link to the pre-computed paths, to save energy. But, as it is discussed, assuming fixed VN mapping reduces the level of energy saving. Besides, rerouting off-peak traffic of every virtual link is not a good practice, as discussed.

Our Contributions: *a)* Different from previous research studies [4]–[8] our method does not decrease the network admittance ratio for new virtual networks. This is because we reconfigure mapping of the already accepted VNs only for the off-peak period, and they could be reconfigured back to their peak mapping in the case of unexpected new demand. *b)* We do not move VMs, so our method does not have the difficulties of [10]. *c)* We define the stress rate for an intermediate physical node. So, we approach the reconfiguration problem by formulating a single optimization program that *may* set less stressed intermediate physical nodes and their respective physical links into sleep mode. Accordingly, a virtual link is re-mapped *if and only if* the program sets at least one intermediate physical node across its embedded physical path into sleep mode. Therefore, our solution makes a decision about which intermediate physical nodes are required to be set into sleep mode, and which virtual links are necessary to be re-mapped. This increases the complexity of the program in comparison to the classic energy-efficient routing programs [2], [3], [12]–[14]. However, it enables the providers to control the possible interruptions by adjusting the stress rate's threshold. This is a novel approach different from any existing research studies. *d)* As a consequence of this novel approach, our solution is not limited to a sub-topology as the case in [2], [3], [9], [14], and so it has larger degree of freedom to save energy. *e)* We present a heuristic algorithm that could achieve closely to the optimum results, but much faster than the BILP. *f)* We evaluate the proposed solutions by extensive simulations.

III. POWER MODEL

We assume a constant amount of power consumption for an active physical node or link, regardless of its traffic load. This is a common model which today's networks are designed based on it, and it is also widely used [15].

In this regard, the ith substrate node v_s^i drains $\tilde{p}^m(v_s^i)$ in the case it is active, regardless of its traffic load. $\tilde{p}^m(v_s^i)$ is maximum power consumption of v_s^i. $\alpha(v_s^i)\tilde{p}^m(v_s^i)$ is actual power consumption of v_s^i, where $\alpha(v_s^i)$ indicates the status of v_s^i. If v_s^i is active, $\alpha(v_s^i)$ is 1. Otherwise, $\alpha(v_s^i)$ is 0. According to [16], maximum power consumption $\tilde{p}^m(v_s^i)$ for v_s^i is equal to $C_b(v_s^i)^{\frac{2}{3}}$, where $C_b(v_s^i)$ is the bandwidth capacity of v_s^i.

Similarly, $\alpha(l_s^{i,j})\tilde{p}^m(l_s^{i,j})$ is actual power consumption of substrate link $l_s^{i,j}$. Where, $l_s^{i,j}$ stands for the substrate link that connects the ith substrate node v_s^i to the jth substrate node v_s^j. $\tilde{p}^m(l_s^{i,j})$ is maximum power consumption of $l_s^{i,j}$. Normally, $\tilde{p}^m(l_s^{i,j})$ is defined for different ranges of bandwidth capacity, and according to the link's length and the type of the cable. Some numerical values for $\tilde{p}^m(l_s^{i,j})$ are given in [15].

The power model determines the energy saving methodology. In this paper, we assume all the substrate links are in the same range of bandwidth capacity, so active substrate links consume the same amount of power. This is the common case in today's real networks. Based on the defined power model, we intend to set a device (intermediate substrate node, or substrate link) into sleep mode, to save energy effectively. In the case of unexpected traffic behaviour or higher traffic demands, the network elements could be woken up. We assume the ideal case in which a device consumes no power in sleep mode. Since physical nodes consume much larger amounts of power in comparison to physical links, it is more essential to set an intermediate substrate node into sleep mode rather than a substrate link.

IV. BINARY INTEGER LINEAR PROGRAM

In this problem, the substrate network topology, embedded virtual networks' topologies, the bandwidth capacity of every substrate node/link, as well as the off-peak traffic demands of each VN, are known. It is required to find modified mapping of virtual links for the off-peak period. This mapping leads to minimum energy consumption of intermediate substrate nodes and substrate links during the off-peak time. This is subject to supporting determined off-peak traffic demands. This reconfiguration strategy is a local optimization problem, because it does not reconfigure mapping of every virtual link, in order to reduce the traffic interruptions. In this regard, first, we model VNE mathematically. Afterwards, we define a binary integer linear program for the problem.

A. Network Model

The substrate network is modelled as a directed graph $G_s = (V_s, E_s)$ where V_s is the set of substrate vertices, and E_s is the set of substrate edges. Vertices represent nodes and edges denote links in the network environment. Since the graph is directed, we have the higher level of flexibility in terms of rerouting traffic flows.

Similar to the substrate network model, the nth virtual network, from the set of all the involved virtual networks Φ, is also modelled as a directed graph $G_n = (V_n, E_n)$. V_n and E_n stand for the nth virtual network's vertices and edges, respectively. $N_n = |V_n|$ and $L_n = |E_n|$ denote the total number of virtual nodes and the total number of virtual links of the nth virtual network, respectively.

In VNE embedding procedure, a requested virtual network G_n is mapped onto the substrate network G_s: $G_n \rightarrow G_s$. Virtual nodes are embedded onto the chosen physical nodes. A virtual link could be mapped onto a single physical link, or multiple physical links which makes a physical path. If traffic is splittable, the requested traffic capacity for a virtual link could be allocated in multiple substrate paths. However, if traffic is non-splittable each demanded traffic capacity is allocated only in one path. In this paper, we study the problem for non-splittable traffic, in order to avoid out-of-order packet delivery. The embedded virtual links of the nth VN are given as a set of ordered embedded virtual node pairs (a_m, b_m), $m = 1, 2, \ldots, L_n$. $l_n^{a_m,b_m}$ represents the m_{th} virtual link, belonging to the nth VN, that connects the virtual node mapped onto the a_mth substrate node $v_s^{a_m}$ to the virtual node mapped onto the b_mth substrate node $v_s^{b_m}$. In addition, $\varrho_n^m(v_s^i)$ is 1, if the mapped path for $l_n^{a_m,b_m}$ includes v_s^i. Otherwise, $\varrho_n^m(v_s^i)$ is 0. Moreover, the off-peak traffic demand $\acute{r}_n^m$ for each virtual link $l_n^{a_m,b_m}$ is also given. Besides, $\acute{r}_n^{i,j}(m)$ denotes the off-peak traffic demand flows through allocated traffic capacity to $l_n^{a_m,b_m}$ in $l_s^{i,j}$. $\acute{r}_n^{i,j}(m)$ in every physical link over the mapped paths is known. During the the off-peak period, the reserved traffic capacity for a virtual link $l_n^{a_m,b_m}$ in a physical link $l_s^{i,j}$ is equal to its off-peak traffic demand $\acute{r}_n^{i,j}(m)$, and rest of the physical link's bandwidth capacity could be shared. Besides, the bandwidth capacity $C_b(l_s^{i,j})$ of each physical link $l_s^{i,j}$, and the bandwidth capacity $C_b(v_s^i)$ of each physical node v_s^i are known.

B. Program

In this problem, a virtual link is re-mapped *if and only if* the program sets at least one intermediate substrate node across its embedded path into sleep mode. However, we do not intend to sleep highly stressed intermediate substrate nodes. So, we do not reconfigure mapping of virtual links that all the substrate nodes along their embedded paths are highly stressed. The stress rate $\tilde{s}(v_s^i)$ of a substrate node v_s^i indicates intensity of involved VNs and total off-peak traffic demand in the substrate node. A VN is involved in a substrate node v_s^i, if at least one of its virtual nodes is mapped onto v_s^i, or at least one of its mapped virtual links passes through v_s^i. Assume $\eta(v_s^i)$ as the number of virtual networks involved in physical node v_s^i, the following equation defines $\tilde{s}(v_s^i)$.

$$\tilde{s}(v_s^i) = \frac{\eta(v_s^i)}{|\Phi|} \left(\frac{\sum_{\{j|(i,j)\in E_s\}} \sum_{\{n|G_n\in\Phi\}} \sum_{m=1}^{L_n} \acute{r}_n^{i,j}(m)}{C_b(v_s^i)} + \frac{\sum_{\{j|(j,i)\in E_s\}} \sum_{\{n|G_n\in\Phi\}} \sum_{m=1}^{L_n} \acute{r}_n^{j,i}(m)}{C_b(v_s^i)} \right) \quad (1)$$

$\tilde{s}(v_s^i)$ considers two parameters. The first parameter ($\frac{\eta(v_s^i)}{|\Phi|}$) is the fraction of the number of involved VNs in the substrate node, over the total number of active VNs. This parameter denotes intensity of the involved VNs in v_s^i. This is an important factor, because a highly intense substrate node in regard to the involved VNs, means traffic of the large number of involved VNs passes through the substrate node. Therefore, sleeping such a substrate node might affect normal operations in the large number of VNs. The second parameter (all the terms except the first parameter) concerns about the off-peak traffic demand by finding the fraction of total off-peak traffic passes the substrate node, over bandwidth capacity of the node. This is essential, since sleeping a substrate node with high traffic utilization might cause large traffic interruptions. In this regard, we do not set intermediate substrate nodes with $\tilde{s}(v_s^i) \geq \mathcal{T}$ into sleep mode, in order to decrease traffic disruption. $\mathcal{T}$ is the stress rate's threshold, and it is a real number between 0 and 1. Providers could adjust $\mathcal{T}$. Decreasing $\mathcal{T}$ degrades the amount of energy the program could save, but also reduces the traffic interruptions due to the reconfiguration. This is because the smaller number of physical node and links are considered for energy saving. The impact of setting different values of $\mathcal{T}$ on energy saving ability of the solution is discussed in Section VI.

The described problem could be formulated as a BILP in the category of multi-commodity flow problems, as follows:

Optimization Variables: $\alpha(v_s^i)$ is a binary variable that denotes the status of v_s^i. $\alpha(v_s^i)$ is 1 in the case v_s^i is active, otherwise it is 0. Similarly, $\alpha(l_s^{i,j})$ is a binary variable that refers to the status of $l_s^{i,j}$. Besides, ω_n^m is a binary variable that is 1 in the case at least one intermediate substrate node over the embedded path for $l_n^{a_m,b_m}$ is set into sleep mode. Otherwise, ω_n^m is 0. Furthermore, $z_n^{i,j}(m)$ is a binary variable. $z_n^{i,j}(m)$ is 1 in the case the re-mapped path for $l_n^{a_m,b_m}$ includes $l_s^{i,j}$. Otherwise, $z_n^{i,j}(m)$ is 0.

Objective Function: This program aims to minimize energy consumption of intermediate substrate nodes and substrate links. Equation 2, and the constraint in Equation 4 that allows only intermediate substrate nodes to be set into sleep mode, preserve this objective.

$$Minimize \left\{ \sum_{i \in V_s} \alpha(v_s^i) \tilde{p}^m(v_s^i) + \sum_{(i,j) \in E_s} \alpha(l_s^{i,j}) \tilde{p}^m(l_s^{i,j}) \right\} \tag{2}$$

Constraints: If the program puts an intermediate physical node into sleep mode, it needs to suggest a single alternative path for every virtual link passing the substrate node. In this regard, the constraint in Equation 3 sets the binary variable ω_n^m to 1, if at least one substrate node v_s^i, that is on the allocated path for $l_n^{a_m,b_m}$ ($\varrho_n^m(v_s^i) = 1$), is set into sleep mode ($\alpha(v_s^i) = 0$) by the program. Otherwise, ω_n^m is 0. The program derives ω_n^m for every virtual link. ω_n^m determines whether $l_n^{a_m,b_m}$ is needed to be re-mapped during the off-peak period, or not. Note that B_1, B_2, and B_3 are integer numbers, while they must be large enough to be greater than the left hand side of their respective equations.

$$\omega_n^m \leq \sum_{i \in V_s} \varrho_n^m(v_s^i) \left(1 - \alpha(v_s^i)\right) \leq B_1 \omega_n^m \,, \qquad \forall n \in \{n | G_n \in \Phi\}, m = 1, 2, \ldots, L_n \tag{3}$$

The constraint in Equation 4 requires routing a single unit of data from the virtual node mapped onto $v_s^{a_m}$ to the virtual node mapped onto $v_s^{b_m}$, if ω_n^m is 1. As variable $z_n^{i,j}(m)$ is binary, the unit of data could not be splitted. Besides, the constraint in Equation 5 limits the program routing. So, the maximum number of incoming and outgoing flows of each virtual link in any node is two, to maintain a single loopless path. Thus, the driven route will be a single path from $v_s^{a_m}$ to $v_s^{b_m}$, which will be used as a replaced path for $l_n^{a_m,b_m}$. In the case ω_n^m is 0, $l_n^{a_m,b_m}$ is not re-mapped.

$$\sum_{\{j | (i,j) \in E_s\}} z_n^{i,j}(m) - \sum_{\{j | (j,i) \in E_s\}} z_n^{j,i}(m) = \begin{cases} \omega_n^m & if \quad i = a_m \\ -\omega_n^m & if \quad i = b_m \\ 0 & otherwise \end{cases} \,, \qquad \forall i \in V_s, \forall n \in \{n | G_n \in \Phi\}, m = 1, 2, \ldots, L_n \tag{4}$$

$$\sum_{\{j | (i,j) \in E_s\}} z_n^{i,j}(m) + \sum_{\{j | (j,i) \in E_s\}} z_n^{j,i}(m) \leq 2 \,, \qquad \forall i \in V_s, \forall n \in \{n | G_n \in \Phi\}, m = 1, 2, \ldots, L_n \tag{5}$$

It is not feasible the alternative path for a re-mapped virtual link passes through the sleeping physical nodes. The constraint in Equation 6 confirms the total incoming and outgoing diverted flows passing through a sleeping physical node ($\alpha(v_s^i) = 0$) is 0.

$$\sum_{\{j | (i,j) \in E_s\}} \sum_{\{n | G_n \in \Phi\}} \sum_{m=1}^{L_n} z_n^{i,j}(m) + \sum_{\{j | (j,i) \in E_s\}} \sum_{\{n | G_n \in \Phi\}} \sum_{m=1}^{L_n} z_n^{j,i}(m) \leq B_2 \alpha(v_s^i) \,, \qquad \forall i \in V_s \tag{6}$$

Additionally, the total allocated traffic capacity to virtual links in every substrate link must be equal or less than the physical link's bandwidth capacity. Equation 7 preserves this constraint.

$$r(l_s^{i,j}) \leq C_b(l_s^{i,j}) \,, \quad \forall (i,j) \in E_s \tag{7}$$

In this problem, there could be two types of allocated traffic capacity to virtual links, in a substrate link. The first type is the reserved off-peak traffic capacity for virtual links that are not re-mapped by the program. $1 - \omega_n^m$ determines whether $l_n^{a_m,b_m}$ is re-mapped or not. If the program re-maps $l_n^{a_m,b_m}$, ω_n^m is 1. Therefore, $1 - \omega_n^m$ is 0, which means $l_n^{a_m,b_m}$ is re-mapped and so its off-peak traffic capacity in its original path is not reserved, anymore. On the other hand, if $l_n^{a_m,b_m}$ is not re-allocated, $1 - \omega_n^m$ is 1, and its off-peak traffic capacity in its original path is reserved. The second type is the off-peak traffic capacity allocated to the reconfigured virtual links in the substrate link. $z_n^{i,j}(m)\acute{r}_n^m$ determines the amount of allocated off-peak traffic capacity to a reconfigured virtual link $l_n^{a_m,b_m}$ in $l_s^{i,j}$. If the off-peak traffic demand $\acute{r}_n^m$ of $l_n^{a_m,b_m}$ is re-mapped through $l_s^{i,j}$, $z_n^{i,j}(m)\acute{r}_n^m$ is equal to $\acute{r}_n^m$. Otherwise, this amount is 0. Equation 8 calculates the total allocated traffic caapcity in a physical link $l_s^{i,j}$.

$$r(l_s^{i,j}) = \sum_{\{n | G_n \in \Phi\}} \sum_{m=1}^{L_n} \left((1 - \omega_n^m)\acute{r}_n^{i,j}(m) + z_n^{i,j}(m)\acute{r}_n^m\right) \tag{8}$$

The total allocated traffic capacity $r(v_s^i)$ in a substrate node v_s^i also must be equal or less than its physical bandwidth capacity $C_b(v_s^i)$, as expressed in Equation 9.

$$r(v_s^i) \leq C_b(v_s^i) \,, \quad \forall i \in V_s \tag{9}$$

Equation 10 defines the total allocated off-peak traffic capacity in a substrate node.

$$r(v_s^i) = \sum_{\{j|(i,j)\in E_s\}} r(l_s^{i,j}) + \sum_{\{j|(j,i)\in E_s\}} r(l_s^{j,i}) \qquad (10)$$

The constraint in Equation 11 ensures a substrate node with $\tilde{s}(v_s^i) \geq \mathcal{T}$ remains active. Besides, if all the intermediate physical nodes along the embedded path for a virtual link are highly stressed, the constraint in Equation 3 confirms the virtual link is not re-mapped.

$$\alpha(v_s^i) = 1\,, \quad \forall i \in \{i|i \in V_s, \tilde{s}(v_s^i) \geq \mathcal{T}\} \qquad (11)$$

Moreover, the constraint in Equation 12 makes the program linear by setting binary variable $\alpha(l_s^{i,j})$ to 1, if there is any allocated traffic capacity to a virtual link in $l_s^{i,j}$. Otherwise, $\alpha(l_s^{i,j})$ is 0.

$$r(l_s^{i,j}) \leq B_3\alpha(l_s^{i,j})\,, \quad \forall(i,j) \in E_s \qquad (12)$$

In addition, the variables must hold the following bounds:

$$z_n^{i,j}(m) \in \{0,1\}\,, \\ \forall(i,j) \in E_s, \forall n \in \{n|G_n \in \Phi\}, m = 1,2,\ldots,L_n \qquad (13)$$

$$\alpha(l_s^{i,j}) \in \{0,1\}\,, \quad \forall(i,j) \in E_s \qquad (14)$$

$$\alpha(v_s^i) \in \{0,1\}\,, \quad \forall i \in V_s \qquad (15)$$

$$\omega_n^m \in \{0,1\}\,, \quad \forall n \in \{n|G_n \in \Phi\}, m = 1,2,\ldots,L_n \qquad (16)$$

The formulated binary integer linear program could be reduced to the problem discussed in [17] that is a simple two-commodity integer flow problem. It is proven in [17] that this simple two-commodity integer flow problem is $\mathcal{NP}$-hard. Hence, our formulated BILP is $\mathcal{NP}$-hard.

V. HEURISTIC ALGORITHM

The discussed BILP is $\mathcal{NP}$-hard, and therefore the optimization solution is not scalable in the case of large network sizes, due to the long executing time. In this section, we propose a heuristic algorithm for the same problem, which is expected to be scalable to large network sizes. Pseudo code of the proposed heuristic algorithm is shown in Algorithm 1.

The available bandwidth capacity $\breve{C}_b(l_s^{i,j})$ during the off-peak time in $l_s^{i,j}$ is equal to $C_b(l_s^{i,j}) - \sum_{\{n|G_n\in\Phi\}}\sum_{m=1}^{L_n}\acute{r}_n^{i,j}(m)$. Besides, the available off-peak bandwidth capacity $\breve{C}_b(v_s^i)$ in v_s^i is $C_b(v_s^i) - \left(\sum_{\{j|(i,j)\in E_s\}}\sum_{\{n|G_n\in\Phi\}}\sum_{m=1}^{L_n}\acute{r}_n^{i,j}(m) + \sum_{\{j|(j,i)\in E_s\}}\sum_{\{n|G_n\in\Phi\}}\sum_{m=1}^{L_n}\acute{r}_n^{j,i}(m)\right)$. Moreover, the algorithm makes an auxiliary off-peak substrate topology G_s^T. At the first, G_s^T is the same as substrate network topology.

Since we do not reconfigure mapping of virtual nodes, the heuristic only sets capable intermediate physical nodes into sleep mode. Because the substrate nodes with the higher stress rate $\tilde{s}(v_s^i)$ are more essential in regard to bandwidth demands and traffic disruptions, the algorithm starts setting physical nodes into sleep mode from the node that has the smaller number of involved VNs, and lower off-peak utilization. This happens when the algorithm sorts the intermediate substrate nodes with $\tilde{s}(v_s^i) < \mathcal{T}$, in ascending order based on $\tilde{s}$. The list is represented by S_L.

Algorithm 1 Heuristic Algorithm

```
for all $i \in V_s$ do
    if $\tilde{s}(v_s^i) < \mathcal{T}$, and $v_s^i$ is an intermediate substrate node then
        place $v_s^i$ in $S\_L$, in ascending order based on $\tilde{s}(v_s^i)$
    end if
    if $\tilde{s}(v_s^i) = 0$ then
        remove $v_s^i$ and all its respective physical links from $G_s^T$
    end if
end for
for all $i$ such that $v_s^i$ is the top unchecked physical node in $S\_L$ do
    remove $v_s^i$ and all its respective physical links from $G_s^T$
    for all virtual links $l_n^{a_m,b_m}$ pass through $v_s^i$ do
        find $K$-shortest path from $v_s^{a_m}$ to $v_s^{b_m}$ in $G_s^T$
        for all $K$ found paths do
            for all $(x,y)$ such that $l_s^{x,y}$ is on the alternative path do
                $\breve{C}_b(l_s^{x,y}) = \breve{C}_b(l_s^{x,y}) - \acute{r}_n^m$
                if $\breve{C}_b(l_s^{x,y}) < 0$ then
                    $\breve{C}_b(l_s^{x,y}) = \breve{C}_b(l_s^{x,y}) + \acute{r}_n^m$
                    undo all the modifications respective to $v_s^i$
                    break, and go for next found path
                else
                    $\acute{r}_n^{x,y}(m) = \acute{r}_n^{x,y}(m) + \acute{r}_n^m$
                end if
            end for
            for all $j$ such that $v_s^j$ is an intermediate physical node on the alternative path do
                $\breve{C}_b(v_s^j) = \breve{C}_b(v_s^j) - 2\acute{r}_n^m$
                if $\breve{C}_b(v_s^j) < 0$ then
                    $\breve{C}_b(v_s^j) = \breve{C}_b(v_s^j) + 2\acute{r}_n^m$
                    undo all the modifications respective to $v_s^i$
                    break, and go for next found path
                end if
            end for
            if the path is capable then
                break from checking the rest of the found paths
            end if
        end for
        if reallocation was successful for $l_n^{a_m,b_m}$ then
            remove the previously allocated traffic capacities to $l_n^{a_m,b_m}$ in any physical link
        else
            place $v_s^i$ and all its respective physical links back to $G_s^T$
            undo all the modifications respective to $v_s^i$
            break, and go for next substrate node in $S\_L$
        end if
    end for
end for
return $G_s^T$
```

In the next step, the algorithm removes the top unchecked physical node v_s^i in S_L, and its respective physical links, from G_s^T. Nevertheless, there must be a single alternative path for every virtual link $l_n^{a_m,b_m}$ that was passing through the removed substrate node v_s^i. The algorithm finds K loopless shortest paths from the source node of the virtual node to its sink node. Our preferred routing algorithm to find K loopless shortest paths is Yen's algorithm [18], while the cost of every physical link is assumed to be 1, according to the objective. Note that the value of K is adjustable, and its effect on the heuristic's outcome is discussed in Section VI. The alternative path must support the off-peak traffic demand $\acute{r}_n^m$ of the respective virtual link $l_n^{a_m,b_m}$. So, a capable alternative path needs to satisfy some constraints.

First, the available off-peak bandwidth capacity $\breve{C}_b(l_s^{x,y})$ of every physical link $l_s^{x,y}$ on the alternative path must be equal or greater than the off-peak traffic demand $\acute{r}_n^m$ of the virtual link. Second, the available off-peak bandwidth capacity $\breve{C}_b(v_s^j)$ of every physical node v_s^j on the alternative path also must support off-peak traffic demand $\acute{r}_n^m$ of the virtual link. If the found path satisfies these constraints, the algorithm updates $\breve{C}_b(l_s^{x,y})$ and $\breve{C}_b(v_s^j)$ of all the physical link and nodes on the path, respectively, and stops checking the rests of the shortest paths. Besides, it removes the previously allocated traffic capacities to the virtual link. However, if one or multiple physical nodes or links on the found path do not support the demanded traffic, the heuristic checks the next shortest path. In the case there is no capable alternative path, the algorithm puts the physical node and its respective physical links back to G_s^T, discards the modifications, and checks the removal possibility of the next substrate node in S_L. After the checking process for all of the physical nodes in S_L, G_s^T is returned as the energy-efficient off-peak substrate topology.

The complexity of this heuristic is $\mathcal{O}(K|V_s||\Phi||E_v^m|(|V_s|^2 log|V_s| + |E_s|^2|\Phi||E_v^m| + |V_s||E_s||\Phi||E_v^m|))$. Where, E_v^m is set of the edges of the involved virtual network with the largest number of virtual links. Therefore, the heuristic is much faster than the BILP.

VI. EVALUATION

We assess several random VNE setups to study the impact of different parameters on the energy saving capability of the discussed BILP and the heuristic.

Substrate and virtual networks' topologies are generated by Waxman algorithm [19], similar to several other studies, e.g. [7], [8]. In this paper, we choose the Waxman parameters, for both substrate and virtual networks' topologies, as $\lambda = 0.5$ and $\mu = 0.5$, in the area size of 100×100, as the other studies. The substrate links' bandwidth capacity and the virtual links' peak demand are generated randomly with the uniform distribution. The bandwidth capacity of each physical link is a random amount between 100Mbps and 200Mbps, but each virtual link's peak bandwidth demand is generated randomly between 50Mbps and 100Mbps. The bandwidth capacity of every physical node is a constant amount of 1Gbps. The randomly generated substrate networks are symmetric. Each created virtual node is mapped onto a substrate node, with the uniform distribution. Afterwards, every generated virtual link's peak bandwidth demand is allocated in a substrate path through a state-of-the-art algorithm that does not concern about energy efficiency.

We assess the capability of the defined BILP on small random simulation setups, similar to the other related works in [7]–[9], [15]. The BILP is solved by MOSEK solver [20]. But, the performance of the suggested heuristic is examined on large random simulation setups, as it is scalable to large scenarios. A small random simulation setup contains 10 randomly generated VNEs. Each VNE has 2 random virtual networks that are mapped onto a single random substrate network, while the substrate network has 15 nodes and each virtual network has 5 nodes. The average number of physical links in the small random simulation setups is 55. Furthermore, every large random simulation setup includes 10 randomly generated VNEs. All the VNEs in a large random simulation setup have 2 random virtual networks that are mapped onto a single random substrate network, while the substrate network has 50 physical nodes and each virtual network has 20 virtual nodes. The average number of physical links in the large random simulation setups is 570. We assume $\mathcal{T} = 0.6$, unless otherwise stated. The average results including confidence intervals with the confidence level of 90% are calculated for each setup.

First, we study the power saving capability of the formulated BILP. We measured network's total node and link power consumption, on different off-peak ratios, before and after applying the defined solution. The off-peak ratio is the fraction of network's off-peak traffic rate by its peak traffic rate. The measurement results for the off-peak ratio range of 0.1 to 0.9, are shown in Figure 1a. The results reveal the formulated BILP reduces total node and link power consumption of the simulated virtualized networks, noticeably. Besides increasing the off-peak traffic ratio increases the amount of traffic demands needs to be re-allocated, so the program is more limited in terms of finding alternative paths for each re-mapped virtual link. In result, the number of intermediate physical nodes and physical links in sleep mode is decreasing by increasing the off-peak traffic demand. Therefore, the program could save lower amounts of power.

Second, we measured network's total node and link power consumption before and after applying the heuristic algorithm, on the defined range of off-peak traffic ratio. The heuristic is simulated when K is 1, 2, 3, 4 or 5. The results are presented in Figure 1b. The results confirm the proposed heuristic algorithm is able to reduce VNE's total node and link power consumption, effectively. In addition, increasing K raises the amount of power heuristic could save, as it is more probable to find alternative paths.

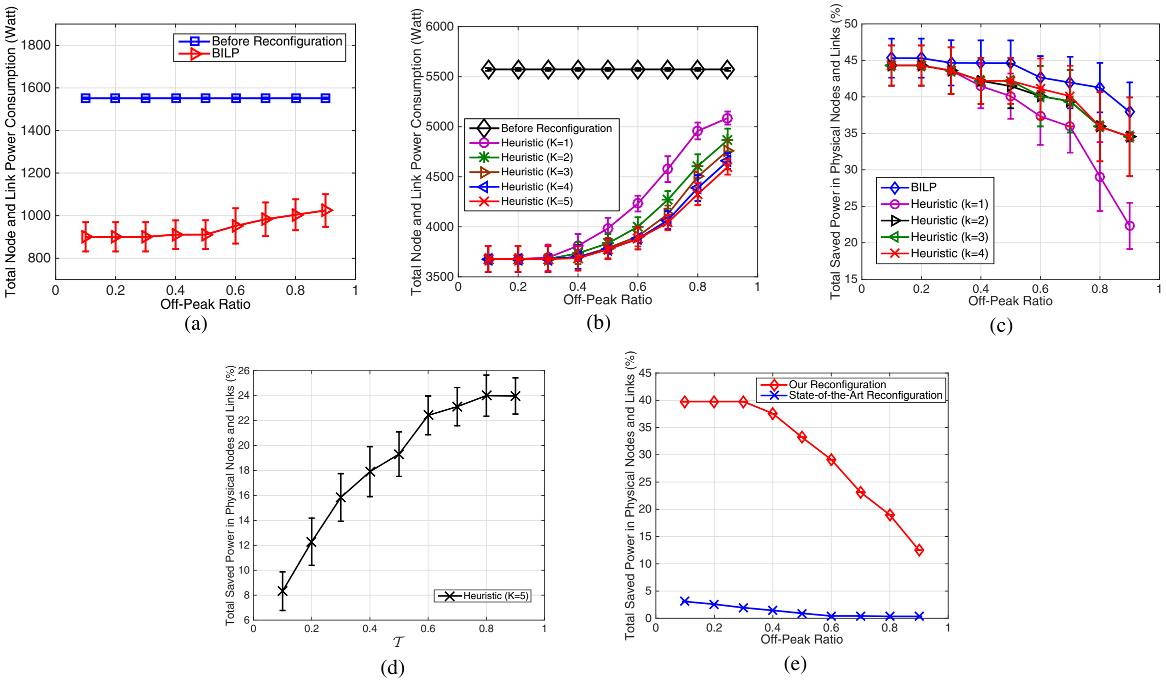
Figure 1c shows the percentage of power the BILP saves, as well as the percentage of power the proposed heuristic for the same problem could save. This is tested for different values of K, on the defined range of off-peak ratio. The heuristic works closely to the optimum points set by BILP, while the heuristic is significantly faster than the BILP. The results reveals it takes about 40,000 seconds to run the BILP on a small simulation setup. But, it takes about only 3 seconds to run the heuristic on the same setup.

Figure 1d studies the effect of changing $\mathcal{T}$ on the capability of the heuristic on energy saving, when off-peak ratio is 0.5 and $K = 5$. Figure 1d shows decreasing $\mathcal{T}$, decreases the amount of power heuristic saves. This is because smaller number of substrate nodes and links are considered for energy saving. Although decreasing $\mathcal{T}$ decreases the amount of power formulated solutions could save, it reduces the traffic interruptions due to the reconfiguration. Providers could adjust $\mathcal{T}$ to control the possible traffic interruptions.

Moreover, Figure 1e compares the results of our suggested heuristic to the discussed state-of-the-art reconfigurations in [9], [14]. The results confirms our proposed approach could save much larger rates of power in comparison to the existing approaches, while the provider could control the possible interruptions. This is because the the state-of-the-art methods assume fixed VN mapping.

VII. CONCLUSION

ICT's power consumption is growing fast. Virtualized network environment has been recently emerged in this technology. Therefore, it is vital to develop solutions that decrease



(a) (b) (c)

(d) (e)

Fig. 1. (a) Total node and link power consumption based on the off-peak ratio before and after applying the BILP. (b) Total node and link power consumption based on the off-peak ratio before and after applying the heuristic with different values of K. (c) Total saved power in physical nodes and links by the BILP and the heuristic with different values of K. (d) The impact of different values of $\mathcal{T}$ on the total saved power by the heuristic. (e) Comparison between our reconfiguration heuristic and the state-of-the-art reconfiguration.

energy consumption in VNE. In this paper, we have formulated a BILP that reduces energy consumption of VNE during the off-peak period. The proposed method reconfigures mapping for some of the mapped virtual links for the off-peak period. This solution enables providers to adjust the level of the reconfiguration, and accordingly control possible traffic disruptions due to the reconfiguration. We also suggested a novel energy saving heuristic algorithm for the same problem. The defined BILP and the heuristic are tested over randomly generated VNEs. Simulation results show the proposed solutions are noticeably effective and the heuristic achieves closely to the optimum points.

REFERENCES

[1] S. Ricciardi, D. Careglio, G. Santos-Boada, J. Sol-Pareta, U. Fiore, and F. Palmieri, "Saving energy in data center infrastructures," in *First International Conference on Data Compression, Communications and Processing (CCP), IEEE*, 2011, pp. 265–270.

[2] G. Rizzelli, A. Morea, M. Tornatore, and O. Rival, "Energy efficient traffic-aware design of on-off multi-layer translucent optical networks," *Computer Networks, Elsevier*, vol. 56, no. 10, pp. 2443–2455, 2012.

[3] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz, "Dynamic routing at different layers in ip-over-wdm networks-maximizing energy savings," *Optical Switching and Networking, Elsevier*, vol. 8, no. 3, pp. 181–200, 2011.

[4] A. Fischer, M. T. Beck, and H. D. Meer, "An approach to energy-efficient virtual network embeddings," in *International Symposium on Integrated Network Management (IM), IFIP/IEEE*, 2013, pp. 1142–1147.

[5] S. Su, Z. Zhang, X. Cheng, Y. Wang, Y. Luo, and J. Wang, "Energy-aware virtual network embedding through consolidation," in *Computer Communications Workshops (INFOCOM WKSHPS), IEEE*, 2012, pp. 127–132.

[6] B. Wang, X. Chang, J. Liu, and J. K. Muppala, "Reducing power consumption in embedding virtual infrastructures," in *Globecom Workshops (GC Wkshps), IEEE*, 2012, pp. 714–718.

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

[8] J. F. Botero and X. Hesselbach, "Greener networking in a network virtualization environment," *Computer Networks, Elsevier*, vol. 57, no. 9, pp. 2021–2039, 2013.

[9] E. Ghazisaeedi, N. Wang, and R. Tafazolli, "Link sleeping optimization for green virtual network infrastructures," in *Globecom Workshops (GC Wkshps), IEEE*, 2012, pp. 842–846.

[10] M. F. Zhani, Q. Zhang, G. Simon, and R. Boutaba, "Vdc planner: Dynamic migration-aware virtual data center embedding for clouds," in *International Symposium on Integrated Network Management (IM), IFIP/IEEE*, 2013, pp. 18–25.

[11] D. Lo, L. Cheng, R. Govindaraju, L. A. Barroso, and C. Kozyrakis, "Towards energy proportionality for large-scale latency-critical workloads," in *Proceeding of the 41st annual international symposium on Computer architecture, IEEE Press*, 2014, pp. 301–312.

[12] G. Shen and R. Tucker, "Energy-minimized design for ip over wdm networks," *Journal of Optical Communications and Networking, IEEE/OSA*, vol. 1, no. 1, pp. 176–186, 2009.

[13] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing isp network energy cost: formulation and solutions," *Transactions on Networking (TON), IEEE/ACM*, vol. 20, no. 2, pp. 463–476, 2012.

[14] Y. Zhang, M. Tornatore, P. Chowdhury, and B. Mukherjee, "Energy optimization in ip-over-wdm networks," *Optical Switching and Networking, Elsevier*, vol. 8, no. 3, pp. 171–180, 2011.

[15] A. P. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. Rougier, "Energy-aware routing: a reality check," in *GLOBECOM Workshops (GC Wkshps), IEEE*, 2010, pp. 1422–1427.

[16] R. Tucker, J. Baliga, R. Ayre, K. Hinton, and W. Sorin, "Energy consumption in ip networks," in *ECOC Symposium on Green ICT, ITU*, 2008, p. 1.

[17] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *16th Annual Symposium on Foundations of Computer Science, IEEE*, 1975, pp. 184–193.

[18] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[19] B. M. Waxman, "Routing of multipoint connections," *Selected Areas in Communications, IEEE*, vol. 6, no. 9, pp. 1617–1622, 1988.

[20] E. D. Andersen and K. D. Andersen, *The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm*, ser. High performance optimization. Springer, 2000, pp. 197–232.