Off-Peak Energy Optimization for Links in Virtualized Network Environment

Ebrahim Ghazisaeedi, *Student Member, IEEE,* Changcheng Huang, *Senior Member, IEEE*

***Abstract*—Energy consumption in Information and Communication Technology (ICT) is estimated to be 10% of the total energy consumed in industrial countries. Besides, the population of ICT customers is growing. In order to handle the increasing traffic demands, service providers need to expand their network infrastructure. The recent proposed network virtualization technology helps slow down the infrastructure expansion by allowing the coexistence of multiple virtual networks over a single physical network. Although Virtualized Network Environment (VNE) slows down the infrastructure expansion and therefore controls power consumption, it is essential to develop new techniques to decrease VNE's energy consumption. In this paper, we discuss multiple novel energy saving reconfiguration methods that globally/locally optimize VNE's link power consumption, during off-peak time. The proposed fine-grained local reconfiguration enables the providers to adjust level of the reconfiguration, and accordingly control possible traffic disruptions. An Integer Linear Program (ILP) is formulated for each solution according to two power models, and considering the impact of traffic splitability. Because the formulated ILPs are not scalable to large network sizes, a novel heuristic algorithm is also suggested. The simulation results prove the proposed solutions are able to save notable amount of energy in physical links during off-peak time.**

***Index Terms*—Virtualized Network Environment; Energy Efficiency; Energy-Efficient Virtualized Network Environment; Link Reconfiguration**

I. Introduction

Information and Communication Technology (ICT) plays a fundamental role in our everyday life. It is difficult to imagine a world without the infrastructure that connects people and transfers their information across the globe. Significant advantages of communication networks have stimulated the demand for this technology. It is predicted that the size of the Internet doubles every 5.32 years [1]. The increase in users' demand, availability of broadband access and the new services offered by ICT, have triggered the warnings about energy consumption of communication technology [2].

In order to handle the growing demands, Internet Service Providers (ISPs) need to expand their physical infrastructure, such as adding extra servers, routers, switches and links. This correspondingly increases power consumption which necessitates controlling and decreasing network's energy usage.

Recently, virtualization has been proposed to share resources in network environment [3]. A VNE supports the coexistence of multiple virtual networks (VNs) over a single substrate network [4]. VNE embedding process maps requested virtual nodes and links onto substrate nodes and paths, respectively. It allocates traffic capacities to virtual links in substrate paths. A VNE uses the actual resources more efficiently by sharing the substrate network's capacity among multiple virtual networks. Sharing the physical resources and allowing coexistence of multiple virtual networks on a single substrate help slow down the infrastructure expansion, and consequently slow down the growing ICT's energy consumption. Nonetheless, it is also essential to furthermore decrease a VNE's energy consumption with additional energy saving techniques, even though VNE already decreases power consumption by concept. An energy-aware VNE slows down the infrastructure expansion as well as network's energy consumption.

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 underutilized 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 peak and minimum off-peak traffic rate over their substrate network [5]. However, today's substrate networks are provisioned to support VNs' peak time traffic demands, with some additional over-provisioning accommodating unexpected traffic rates [5]. 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, e.g. [6], [7], that estimate the future traffic by looking at the current traffic state. 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 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 [5]. Besides, reconfiguring mapping of embedded VNs may require additional signalling traffic that is necessary for notifying all the involved routers [8]. This may introduce significant work load for the signalling controller especially when the reconfiguration tries to make changes to

Ebrahim Ghazisaeedi, and Changcheng Huang are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada. E-Mail: {eghazisaeedi; huang}@sce.carleton.ca

Manuscript Submitted: November 2014, Revised: April 2015, Accepted: May 2015

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

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.

Due to simpler technical implementation and high potential of energy saving in network links [9], this paper is restricted to power saving solutions for links in VNE. In this paper, we discuss multiple novel energy saving solutions that optimize VNE's link power consumption during the off-peak time, according to two defined power models. We approach the problem by reconfiguring mapping of the already embedded virtual links.

First, we study a reconfiguration problem that re-maps every virtual link according to its off-peak traffic demand, in order to to minimize VNE's link power consumption during the off-peak period. This is a coarse-grained (the granularity is a virtual link) and global (it re-maps every virtual link) reconfiguration method. We study this problem as our benchmark, because it delivers the most optimum level of energy saving for this problem. However, as it is discussed, this is not a safe approach. We formulate this method as a Mixed Integer Linear Program (MILP).

Second, we propose a novel reconfiguration methodology. We define a stress rate for a substrate link. Accordingly, a solution is proposed to minimize VNE's link power consumption during the off-peak time. This method *may* set a less stressed substrate link into sleep mode for the off-peak time. We re-map an allocated traffic capacity to a virtual link in a less stressed substrate link *if* we set the substrate link into sleep mode. This is a fine-grained (the granularity is an allocated traffic capacity to a virtual link in a substrate link) and local (it does not re-map every allocated traffic capacity to virtual links) reconfiguration strategy. This method enables the providers to change level of the reconfiguration by adjusting the stress rate's threshold, and therefore control the possible traffic interruptions. Clearly, there is a trade-off between energy saving level and the possible traffic interruptions. In order to study the impact of traffic splitability on energy saving capability of our solutions, we formulate the latter method as a MILP for splittable traffic, and as a Binary Integer Linear Program (BILP) for non-splittable traffic.

The formulated optimization solutions are $\mathcal{NP}$-hard and therefore they are not scalable to large network sizes. Hence, a novel heuristic algorithm is also proposed. The simulation results confirm the heuristic algorithm can achieve closely to the optimum points set by the optimization program. While, it is scalable to large network sizes.

This paper is organized as follows: The related works and our contributions in this paper, are discussed in Section II. Two physical link power models are defined in Section III. The optimization programs are formulated in Section IV and the suggested heuristic is discussed in Section V. The performance of the ILPs 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 terms of network virtualization and energy saving solutions 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 [10]–[13] 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 [10], and consolidating VNs to the smallest number of substrate network elements in [11]–[13]. Modifying the VNE embedding algorithms in order to achieve a power-efficient VNE suffers from a major difficulty. 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.

Some other papers proposed heuristics that modify the already mapped VNs. Juan Fleipe Botero in [14] 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. However, the virtual links might be mapped onto physical paths. Finally, their reconfiguration problem is not formulated mathematically. An off-line heuristic reconfiguration algorithm is proposed in our previous work [15]. The algorithm tries 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 [16] 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) [17]. Besides, their solution does not enable the providers to adjust level of the reconfiguration, and control the possible traffic disruptions.

Since we do not reconfigure mapping of allocated virtual nodes in this paper, the problem might seem similar to the classic routing problems for multi-layer network design. For example, Chuansheng Xin and his co-authors in [18] proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

a mapping method to design a static virtual topology for WDM optical networks. They pre-compute possible routes between source and destination of every requested traffic demand, and then their ILP chooses the best routes according to their objective. [9], [18], and the other existing approaches for energy-efficient routing problem, e.g. [19], [20], are mapping methods. They route every requested traffic demand, between fixed nodes. This is similar to our coarse-grained global reconfiguration method that re-maps every virtual link during the off-peak time. However, as it is discussed in the previous section, it is not safe to re-allocate every traffic demand during the off-peak period. Thus, energy-aware routing of every traffic demand is quite different from locally reconfiguring mapping of VNs and simultaneously considering 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 [21] proposed a MILP that reroutes off-peak traffic in order to minimize energy consumption during the off-peak period. However, their approach, similar to [15] and [8], assumes fixed VN mapping to decrease the possible traffic disruptions. But, as it is discussed, assuming fixed VN mapping reduces the level of energy saving. Furthermore, they do not provide a tool, so the providers could adjust level of the reconfiguration. Similarly, authors in [5] suggested a MILP to reroute off-peak traffic to save energy. Different from previous papers [8], [15], [21], the method in [5] first pre-computes static mappings for VNs, according to their off-peak load. They do no consider energy consumption at this step. Then, the MILP reroutes the off-peak traffic of every virtual link to the pre-computed paths, to save energy. They do not let the MILP to modify mapping of virtual links, in order to reduce the possible traffic interruptions and decrease program's complexity. However, pre-computing off-peak mapping and then searching in them to save energy do not provide the most optimum result, because it is possible to reroute traffic only to the substrate links in the pre-computed mappings. Besides, rerouting off-peak traffic of every virtual link is not a good practice, while the providers are not able to control the possible traffic interruptions.

Our Contributions: *a)* Different from previous research studies [10]–[14] 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 [16]. *c)* We define a stress rate for a substrate link. So, we propose a fine-grained local reconfiguration approach that *may* set a less stressed substrate link into sleep mode for the off-peak time. Accordingly, we re-map an allocated traffic capacity to a virtual link in a less stressed substrate link *if* we set the substrate link into sleep mode. Our solution makes a decision about which allocated traffic capacities of which virtual links, are necessary to be re-mapped. This increases the complexity of the problem in comparison to the classic routing problems in [5], [8], [9], [18]–[21]. However, it enables the providers to change level of the reconfiguration by adjusting the stress rate's threshold, and therefore control the possible interruptions. We show the proposed fine-grained local reconfiguration is able to reach the same level of energy saving as the coarse-grained global reconfiguration (the benchmark), by relaxing stress rate' threshold. This is a novel approach different to 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 [5], [8], [15], [21], and so it has larger degree of freedom to save energy. The simulation results prove the significant improvement in saving power by our method, in comparison to the state-of-the-art. *e)* We discuss how differently we should approach the problem in the case of non-splittable traffic in comparison to splittable traffic, to have wide enough search zone for re-mapping. *f)* We also present a heuristic reconfiguration algorithm that could achieve closely to the optimum results, but much faster than the BILP. *g)* We evaluate the proposed solutions by extensive simulations. To the best of our knowledge, there is not such a comprehensive study in the literature that consider simultaneously global/local, coarse-grained/fine-grained VNE reconfiguration for splittable/non-splittable traffic, according to two power models.

III. Physical Link Power Model

We study two power models to define the physical link's power consumption. The first power model considers constant amount of power consumption ($P^m(l_s^{i,j})$) for an active physical link. $P^m(l_s^{i,j})$ is the maximum link power consumption of substrate link $l_s^{i,j}$. $l_s^{i,j}$ denotes a substrate link that connects ith substrate node to jth substrate node . According to this model, the actual traffic load on the link does not affect the physical link's power consumption. We call this model *Fixed* link power model. Consequently, the actual power consumption $\tilde{p}(l_s^{i,j})$ of a physical link $l_s^{i,j}$ could be found by Equation 1, where $\alpha(l_s^{i,j})$ refers to $l_s^{i,j}$ state. $\alpha(l_s^{i,j})$ is 1 when the link is active, otherwise it is 0.

$$\tilde{p}(l_s^{i,j}) = \alpha(l_s^{i,j}) P^m(l_s^{i,j}) \quad (1)$$

The second power model assumes a base amount of power $P^b(l_s^{i,j})$ that keeps the physical link $l_s^{i,j}$ operational. However, different from the previous model, the power consumption varies linearly, between the base power $P^b(l_s^{i,j})$ (when there is no traffic on the link) and the maximum power $P^m(l_s^{i,j})$ (when the link is fully utilized). We call this model *Semi Proportional* link power model. Equation 2 defines actual power consumption of a physical link $l_s^{i,j}$ according to this power model.

$$\tilde{p}(l_s^{i,j}) = \alpha(l_s^{i,j}) P^b(l_s^{i,j}) + \frac{r(l_s^{i,j})}{C_b(l_s^{i,j})} (P^m(l_s^{i,j}) - P^b(l_s^{i,j})) \quad (2)$$

Where, $C_b(l_s^{i,j})$ stands for the bandwidth capacity of substrate link $l_s^{i,j}$, and $r(l_s^{i,j})$ is the traffic load on $l_s^{i,j}$. Note that $P^b(l_s^{i,j})$, and $P^m(l_s^{i,j})$ are normally defined for different ranges of link bandwidth capacity, based on the link's length, and type of the cable. Some numerical amounts for $P^b(l_s^{i,j})$ and $P^m(l_s^{i,j})$ are given in [22].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

Today's networks are designed based on *Fixed* link power model, so it is a common model that is widely used [22]. Nevertheless, it is not efficient that an active physical link consumes a constant amount of power, regardless of its traffic load. Therefore, physical links are expected to get modified, so their power consumption will be more adaptive to their traffic load. The *Semi Proportional* link power model brings a traffic adaptive power model that physical link's power consumption is changing according to its traffic load. These power models are validated by measurements against actual physical links in the previous studies, e.g. [23], [24].

IV. Integer Linear Programs

Towards decreasing total link energy consumption in VNE during the off-peak hours, the general problem description is the following:

Given:

- Physical substrate network topology
- Allocated virtual networks' topologies
- For each substrate link: Its bandwidth capacity, and every allocated traffic capacity to a virtual link, in the substrate link
- Off-peak traffic demands of every virtual network (determined by VNs' customers or network traffic prediction techniques)

Find:

- Modified off-peak link mapping of VNE that leads to minimum substrate network's link power consumption during the off-peak time

Constraints:

- Supporting off-peak traffic demands

The most optimum result for this problem could be achieved by the coarse-grained global link reconfiguration program. We call this approach, off-peak link energy optimization by global link reconfiguration. We consider the result of this method as the benchmark. Nevertheless, as it is discussed, this method might cause uncontrolled traffic disputations. Therefore, we propose the fine-grained local link reconfiguration program for the defined problem. The latter approach is called off-peak link energy optimization by local link reconfiguration. In this regard, first, we model VNE mathematically. Then, according to both *Fixed* and *Semi Proportional* link power models, we define ILPs for both of the approaches. Since the traffic type (splittable/non-splittable) has a major impact on solution methodology, we formulate off-peak link energy optimization by local link reconfiguration problem, for both splittable and non-splittable traffic.

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 network environment. Since the graph is directed, we have higher level of flexibility in terms of rerouting traffic flows.

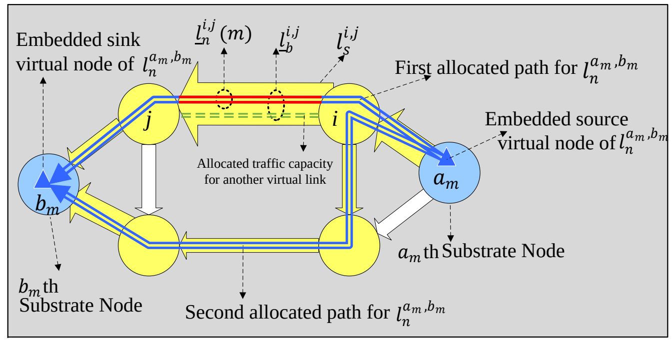

Fig. 1. Example: An embedded virtual link onto a substrate network

Similar to the substrate network model, nth virtual network, from 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 nth virtual network vertices and edges, respectively. $L_n = |E_n|$ denotes total number of virtual links in nth virtual network.

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 qualified substrate nodes. A virtual link could be mapped onto a single substrate link, or multiple substrate links which makes a substrate path. If traffic is splittable, 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. The allocated virtual links of nth VN are given as a set of ordered allocated virtual node pairs (a_m, b_m), $m = 1, 2, \ldots, L_n$. $l_n^{a_m,b_m}$ represents m_{th} virtual link, belonging to nth VN, that connects the virtual node mapped onto a_mth substrate node to the virtual node mapped onto b_mth substrate node. Off-peak traffic demand $\acute{r}_n^m$ of each virtual link $l_n^{a_m,b_m}$ is also given. In addition, $\underline{l}_n^{i,j}(m)$ represents the allocated traffic capacity to $l_n^{a_m,b_m}$ in $l_s^{i,j}$. $\acute{r}_n^{i,j}(m)$ denotes the off-peak traffic demand of $\underline{l}_n^{i,j}(m)$, and it is known for every allocated traffic capacity in any substrate link. During the the off-peak period, the reserved traffic capacity for $\underline{l}_n^{i,j}(m)$ 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. We might aggregate all the allocated traffic capacities in a substrate link. $\underline{l}_b^{i,j}$ denotes the bundled allocated traffic capacities in a substrate link $l_s^{i,j}$, and $\acute{r}_b^{i,j}$ is its associated off-peak traffic demand. Besides, $C_b(l_s^{i,j})$ of each substrate link $l_s^{i,j}$ is specified.

For example, Figure 1 demonstrates a substrate network and a mapped virtual link $l_n^{a_m,b_m}$ onto the network. Since traffic is splittable in this example, two substrate paths are allocated to the virtual link. Figure 1 also shows $\underline{l}_n^{i,j}(m)$ and $\underline{l}_b^{i,j}$ in $l_s^{i,j}$.

B. Programs based on Fixed Link Power Model

According to *Fixed* link power model, an active physical link consume a constant amount of energy, regardless of its traffic load. In this paper, we assume all the physical links in the substrate network are in the same range of bandwidth capacity. Therefore, active substrate links consume the same amount of power. Accordingly, the power saving solution has to put maximum number of physical links into sleep mode, in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

order to minimize the network's link power consumption. We assume the ideal case, in which a physical link in sleep mode consumes no power. The following optimization programs in this section are designed according to this methodology.

1) Off-peak Link Energy Optimization by Global Link Reconfiguration (OL-GLs-F): In this section, we intend to re-map every virtual link for the off-peak time, to minimize number of active substrate links during that period. We re-map every virtual link according to its known off-peak traffic demand. This is a coarse-grained solution, as the granularity is a virtual link. Besides, it is a global optimization, because we re-map every virtual link. The traffic is assumed to be splittable in this problem, to give larger degree of freedom for re-mapping. This program is expected to deliver the most optimum energy saving level for our problem. But, it might cause uncontrollable traffic interruptions. We study this program as the benchmark. When this problem is formulated according to the *Fixed* link power model, it is called OL-GLs-F.

This problem could be formulated as a multi-commodity flow problem. In the context of this problem, a virtual link is a commodity. We have L_n commodities for nth virtual network. OL-GLs-F is written as a MILP as follows:
Optimization Variables:

- $\alpha(l_s^{i,j})$ is an auxiliary binary variable. $\alpha(l_s^{i,j})$ is 1 when $l_s^{i,j}$ is active. Otherwise, $\alpha(l_s^{i,j})$ is 0.
- $f_n^{i,j}(m)$ is a real-valued variable. $f_n^{i,j}(m)$ is the re-allocated traffic capacity to $l_n^{a_m,b_m}$ in $l_s^{i,j}$.

Objective Function: The objective function in Equation 3 minimizes number of active substrate links for the off-peak time.

$$Minimize \sum_{(i,j)\in E_s} \alpha(l_s^{i,j}) \tag{3}$$

Constraints: The first constraint in Equation 4 is flow conservation constraint that maintains the flow balance on the nodes and re-allocates off-peak traffic demand of every virtual link.

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

The second constraint in Equation 5 ensures the total re-allocated traffic capacities $r(l_s^{i,j})$ in each substrate link $l_s^{i,j}$ is less than its physical bandwidth capacity.

$$r(l_s^{i,j}) \le C_b(l_s^{i,j}), \quad \forall (i,j) \in E_s \tag{5}$$

where:

$$r(l_s^{i,j}) = \sum_{\{n|G_n\in\Phi\}} \sum_{m=1}^{L_n} f_n^{i,j}(m) \tag{6}$$

A constraint in Equation 7 is added to make the program linear. Note that B is a large integer number. It must be large enough to be greater than the largest amount of $r(l_s^{i,j})$. $r(l_s^{i,j})$ could be 0 or greater than 0. First, assume $r(l_s^{i,j}) = 0$, so according to the objective function and constraint in Equation 7, $\alpha(l_s^{i,j})$ will be 0. Second, assume $r(l_s^{i,j}) > 0$. In this case, $\alpha(l_s^{i,j})$ must be 1 to convince the constraint.

$$r(l_s^{i,j}) \le B\alpha(l_s^{i,j}), \quad \forall (i,j) \in E_s \tag{7}$$

In addition, the variables must hold the following bounds:

$$f_n^{i,j}(m) \ge 0, \quad \forall (i,j) \in E_s, \forall n \in \{n|G_n \in \Phi\}, m = 1,2,\ldots,L_n \tag{8}$$

$$\alpha(l_s^{i,j}) \in \{0,1\}, \quad \forall (i,j) \in E_s \tag{9}$$

The formulated MILP for OL-GLs-F is a type of VNE embedding problem. MILP is $\mathcal{NP}$-hard in general, because ILP is $\mathcal{NP}$-hard. Besides, VNE embedding process is $\mathcal{NP}$-hard [4]. In consequence, the defined MILP for off-peak link energy optimization by global link reconfiguration problem is $\mathcal{NP}$-hard, and therefore the optimization solution is not scalable to the large network sizes.

2) Off-peak Link Energy Optimization by Local Link Reconfiguration: The previous reconfiguration approach might cause uncontrolled traffic interruptions, as discussed. Therefore, we propose a second methodology. First, we define a stress rate for a substrate link. Then, we develop solutions that *may* set a less stressed substrate link into sleep mode for the off-peak time. We re-map an allocated traffic capacity to a virtual link in a less stressed substrate link *if* we set the substrate link into sleep mode. Different from the previous approach, this is a fine-grained solution as the granularity is an allocated traffic capacity to a virtual link, in a substrate link. Besides, this is a local optimization, because it does not re-map every allocated traffic capacity in any physical link. This method enables the providers to change level of the reconfiguration by adjusting the stress rate's threshold, and therefore control the possible traffic interruptions.

The stress rate $\tilde{s}(l_s^{i,j})$ of a substrate link $l_s^{i,j}$ denotes the intensity of involved VNs and the total off-peak traffic demand in the link. A VN is involved in a substrate link $l_s^{i,j}$, if at least one of its embedded virtual links passes through $l_s^{i,j}$. Assume $\eta(l_s^{i,j})$ as the number of VNs involved in substrate link $l_s^{i,j}$, Equation 10 defines $\tilde{s}(l_s^{i,j})$.

$$\tilde{s}(l_s^{i,j}) = \frac{\eta(l_s^{i,j})}{|\Phi|} \times \frac{\sum_{\{n|G_n\in\Phi\}} \sum_{m=1}^{L_n} \acute{r}_n^{i,j}(m)}{C_b(l_s^{i,j})} \tag{10}$$

$\tilde{s}(l_s^{i,j})$ considers two parameters. The first parameter ($\frac{\eta(l_s^{i,j})}{|\Phi|}$) is the fraction of the number of involved VNs in the substrate link, over total number of active VNs. This parameter denotes intensity of the involved VNs in $l_s^{i,j}$. This is an important factor. If a substrate link is highly intense in regard to the involved VNs, traffic of large number of involved VNs passes through the link. Therefore, sleeping such a substrate link might affect normal operations in 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 link, over bandwidth capacity of the link. This is essential, since sleeping a substrate link with high traffic utilization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

might cause large traffic interruptions. Higher link stress rate means larger number of VNs is involved, or the link is more utilized for the off-peak time. In this regard, we do not re-map the allocated traffic capacities in substrate links with $\tilde{s}(l_s^{i,j}) \geq \mathcal{T}$, in order to control traffic disruptions due to the reconfiguration. $\mathcal{T}$ is stress rate's threshold, and it is a real number between 0 and 1. Providers could adjust $\mathcal{T}$. Decreasing $\mathcal{T}$ degrades amount of power the programs could save, but also reduces the traffic interruptions due to the reconfiguration. This is because smaller number of physical links are considered for power saving. The impact of setting different values of $\mathcal{T}$ on energy saving ability of the solution is discussed in Section VI.

The traffic might be splittable or non-splittable. In splittable case, the traffic demand of each virtual link could be carried by one or multiple paths in the substrate network. However, if the traffic is non-splittable each virtual link's traffic demand may be required to follow the same path through the network, rather than be divided among multiple paths. It is expected to save higher amounts of energy when traffic is splittable, because we are more flexible in terms of finding alternative paths during the reconfiguration. This is a major restriction that has an important impact in the solution methodology. In this regard, we formulate the off-peak link energy optimization by local link reconfiguration program, for both splittable and non-splittable traffic.

a) Splittable Traffic (OL-LLs-F): Because the traffic is assumed to be splittable, it is the same to aggregate all the allocated traffic capacities in a physical link and then re-allocate the bundled traffic capacity onto multiple paths, or re-allocate every single allocated traffic capacity in the link onto multiple paths. In order to simplify the program we re-allocate the bundled traffic capacities according to their off-peak traffic demand. In this problem, we re-map the bundled allocated traffic capacity in a less stressed substrate link *if* we set the substrate link into sleep mode. The off-peak link energy optimization by local link reconfiguration problem for splittable traffic, that is defined according to *Fixed* link power model, is called OL-LLs-F.

The defined reconfiguration problem could be formulated as a multi-commodity flow problem. In the context of this problem, the bundled allocated traffic capacity $\underline{l}_b^{i,j}$ in a substate link $l_s^{i,j}$ with $\tilde{s}(l_s^{i,j}) < \mathcal{T}$, is a commodity. Each commodity $\underline{l}_b^{i,j}$ is associated with a known off-peak traffic demand $\acute{r}_b^{i,j}$. OL-LLs-F is formulated as a MILP as follows:

Optimization Variables:

- $\alpha(l_s^{i,j})$ is an auxiliary binary variable. $\alpha(l_s^{i,j})$ is 1 when the substrate link $l_s^{i,j}$ is active, otherwise $\alpha(l_s^{i,j})$ is 0.
- $f^{x,y}(\underline{l}_b^{i,j})$ is a real-valued variable. It is the re-allocated traffic capacity to $\underline{l}_b^{i,j}$ in $l_s^{x,y}$. $l_s^{x,y}$ is a substrate link connects xth substrate node to yth substrate node. Similarly, $f^{i,j}(\underline{l}_b^{x,y})$ is a real-valued variable.
- $\beta(\underline{l}_b^{i,j})$ is a binary variable. It shows $\underline{l}_b^{i,j}$ status, after reconfiguration. It is 0 in the case $\underline{l}_b^{i,j}$ is removed, after reconfiguration. Otherwise, $\beta(\underline{l}_b^{i,j})$ is 1.

Objective Function: The same objective as Equation 3.

Constraints: The constraints in Equations 5, 7, and the followings: The constraint in Equation 11 is flow conservation constraint that re-allocates off-peak traffic demand of a removed commodity. *The program does not re-allocate off-peak traffic demand of every bundled allocated traffic capacity. It re-allocates the off-peak traffic demand of a commodity if the commodity is removed.* If the program decides to remove a bundled allocated traffic capacity $\underline{l}_b^{i,j}$, $\underline{l}_b^{i,j}$ status changes, and so $1-\beta(\underline{l}_b^{i,j})$ is equal to 1. Therefore, the constraint in Equation 11 requires re-allocating one or multiple alternative paths from ith substrate node to jth substate node, which support its off-peak traffic demand $\acute{r}_b^{i,j}$. Nevertheless, if $\underline{l}_b^{i,j}$ is not removed during the reconfiguration process, $1-\beta(\underline{l}_b^{i,j})$ is equal to 0, and therefore $\underline{l}_b^{i,j}$ is not re-mapped.

$$\sum_{\{y|(x,y)\in E_s\}} f^{x,y}(\underline{l}_b^{i,j}) - \sum_{\{y|(y,x)\in E_s\}} f^{y,x}(\underline{l}_b^{i,j}) = \begin{cases} (1-\beta(\underline{l}_b^{i,j})\acute{r}_b^{i,j} & \text{if} \quad x=i \\ (\beta(\underline{l}_b^{i,j})-1)\acute{r}_b^{i,j} & \text{if} \quad x=j \\ 0 & \text{otherwise} \end{cases}, \quad \forall x\in V_s, \forall (i,j)\in E_s \tag{11}$$

In this problem, different from OL-GLs-F, there might be two types of allocated traffic capacity in a substrate link, during the off-peak time. The first type is an un-reconfigured bundled allocated traffic capacity. This means $\underline{l}_b^{i,j}$ was allocated and it is not removed in the reconfiguration process, therefore $\beta(\underline{l}_b^{i,j}) = 1$. In this case, $\beta(\underline{l}_b^{i,j})\acute{r}_b^{i,j}$ is equal to $\acute{r}_b^{i,j}$ that is its reserved amount of traffic capacity for the off-peak period. If the program re-allocates $\underline{l}_b^{i,j}$, $\beta(\underline{l}_b^{i,j}) = 0$. So, its original allocated traffic capacity is no longer reserved. The second type is the re-allocated bundled traffic capacity of other links (like $l_s^{x,y}$), in substrate link $l_s^{i,j}$. $f^{i,j}(\underline{l}_b^{x,y})$ is the re-allocated traffic capacity to $\underline{l}_b^{x,y}$ in $l_s^{i,j}$, for the off-peak period. Equation 12 calculates the total allocated traffic capacities $r(l_s^{i,j})$ in $l_s^{i,j}$, for this problem.

$$r(l_s^{i,j}) = \beta(\underline{l}_b^{i,j})\acute{r}_b^{i,j} + \sum_{(x,y)\in E_s} f^{i,j}(\underline{l}_b^{x,y}) \tag{12}$$

Moreover, the constraint in Equation 13 avoids re-mapping a bundled allocated traffic capacity in substrate links with $\tilde{s}(l_s^{i,j}) \geq \mathcal{T}$, in order to decrease the traffic interruptions. This constraint enables providers to control level of the reconfiguration, and therefore control the possible interruptions.

$$\beta(\underline{l}_b^{i,j}) = 1\,, \quad \forall (i,j)\in\{(i,j)|(i,j)\in E_s, \tilde{s}(l_s^{i,j})\geq\mathcal{T}\} \tag{13}$$

Furthermore, the variables must hold the bound in Equation 9, and the followings:

$$f^{x,y}(\underline{l}_b^{i,j}) \geq 0\,, \quad \forall (x,y)\in E_s, \forall (i,j)\in E_s \tag{14}$$

$$\beta(\underline{l}_b^{i,j}) \in \{0,1\}\,, \quad \forall (i,j)\in E_s \tag{15}$$

b) Non-Splittable Traffic (OL-LLns-F): Since traffic is non-splittable, it is not possible to re-allocate an allocated traffic capacity to a virtual link in a substrate link, onto multiple substrate paths. Consequently, aggregating all the allocated traffic capacities in a substrate link and then re-allocating the bundled traffic capacity, is not an efficient

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

approach. This is because there would be a smaller number of alternative paths that could support the bundled off-peak traffic demand. In this problem, we re-map an allocated traffic capacity to a virtual link in a less stressed substrate link *if* we set the substrate link into sleep mode. We also might re-map an allocated traffic capacity to a virtual link in a less stressed substrate link *if* re-mapping the traffic capacity provides enough bandwidth capacity in the substrate link for re-mapping of another traffic capacity, which leads to minimum total link power consumption. The off-peak link energy optimization by local link reconfiguration problem for non-splittable traffic, that is formulated according to the *Fixed* link power model, is called OL-LLns-F.

This problem could be formulated as a BILP in category of multi-commodity flow problems. Different from splittable form, an allocated traffic capacity $\underline{l}_n^{i,j}(m)$ to a virtual link $l_n^{a_m,b_m}$ in a substrate link $l_s^{i,j}$ with $\tilde{s}(l_s^{i,j}) < \mathcal{T}$, is a commodity. OL-LLns-F is formulated as a BILP as follows:

Optimization Variables:

- $\alpha(l_s^{i,j})$ is an auxiliary binary variable. $\alpha(l_s^{i,j})$ is 1 when the physical link $l_s^{i,j}$ is active, otherwise $\alpha(l_s^{i,j})$ is 0.
- $z^{x,y}\left(\underline{l}_n^{i,j}(m)\right)$ is a binary variable. If the re-allocated path for commodity $\underline{l}_n^{i,j}(m)$ passes through $l_s^{x,y}$, $z^{x,y}\left(\underline{l}_n^{i,j}(m)\right) = 1$. Otherwise, $z^{x,y}\left(\underline{l}_n^{i,j}(m)\right) = 0$. Similarly, $z^{i,j}\left(\underline{l}_n^{x,y}(m)\right)$ is a binary variable.
- $\beta(\underline{l}_n^{i,j}(m))$ is a binary variable. It shows $\underline{l}_n^{i,j}(m)$ status, after reconfiguration. It is 0 in the case $\underline{l}_n^{i,j}(m)$ is removed, after reconfiguration. Otherwise, $\underline{l}_n^{i,j}(m)$ is 1.

Objective Function: The same objective as Equation 3.

Constraints: The constraints in Equations 5, 7, and the followings: If the program decides to remove an allocated traffic capacity $\underline{l}_n^{i,j}(m)$, $\beta(\underline{l}_n^{i,j}(m))$ will be equal to 0. Therefore, Equation 16 needs to route a single unit of data from ith substrate node to jth substrate node. Because variable $z^{x,y}\left(\underline{l}_n^{i,j}(m)\right)$ is binary, the unit of data could not be splitted. Besides, the constraint in Equation 17 limits the program routing, so maximum number of incoming and outgoing flows of every commodity, in any node, is two flows. This maintains a single loopless path. Thus, the driven route will be used as a replaced path for $\underline{l}_n^{i,j}(m)$. If an allocated traffic capacity $\underline{l}_n^{i,j}(m)$ is not removed ($\beta(\underline{l}_n^{i,j}(m)) = 1$), it will not be re-allocated.

$$\sum_{\{y|(x,y)\in E_s\}} z^{x,y}\left(\underline{l}_n^{i,j}(m)\right) - \sum_{\{y|(y,x)\in E_s\}} z^{y,x}\left(\underline{l}_n^{i,j}(m)\right) = \begin{cases} 1-\beta(\underline{l}_n^{i,j}(m)) & \text{if} \quad x=i \\ \beta(\underline{l}_n^{i,j}(m))-1 & \text{if} \quad x=j \\ 0 & \text{otherwise} \end{cases}, \quad \forall x \in V_s, \forall n \in \{n|G_n \in \Phi\}, m=1,2,\ldots,L_n, \forall(i,j)\in E_s \tag{16}$$

$$\sum_{\{y|(x,y)\in E_s\}} z^{x,y}\left(\underline{l}_n^{i,j}(m)\right) + \sum_{\{y|(y,x)\in E_s\}} z^{y,x}\left(\underline{l}_n^{i,j}(m)\right) \le 2, \quad \forall x \in V_s, \forall n \in \{n|G_n \in \Phi\}, m=1,2,\ldots,L_n, \forall(i,j)\in E_s \tag{17}$$

The total allocated traffic capacities $r(l_s^{i,j})$ in a physical link $l_s^{i,j}$ during off-peak period, is the summation of total unreconfigured allocated traffic capacities ($\beta(\underline{l}_n^{i,j}(m) = 1)$) in $l_s^{i,j}$ as well as the re-allocated traffic capacities of other links (like $l_s^{x,y}$) in $l_s^{i,j}$. $r(l_s^{i,j})$ is calculated in Equation 18. $z^{i,j}\left(\underline{l}_n^{x,y}(m)\right)\acute{r}_n^{x,y}(m)$ is the re-allocated traffic capacity to $\underline{l}_n^{x,y}(m)$ in $l_s^{i,j}$.

$$r(l_s^{i,j}) = \sum_{\{n|G_n\in\Phi\}} \sum_{m=1}^{L_n} \left\{\beta(\underline{l}_n^{i,j}(m))\acute{r}_n^{i,j}(m)\right\} + \sum_{(x,y)\in E_s} \sum_{\{n|G_n\in\Phi\}} \sum_{m=1}^{L_n} \left\{z^{i,j}\left(\underline{l}_n^{x,y}(m)\right)\acute{r}_n^{x,y}(m)\right\} \tag{18}$$

The constraint in Equation 19 prevents the program to re-map an allocated traffic capacity in a physical link $l_s^{i,j}$ with $\tilde{s}(l_s^{i,j}) \ge \mathcal{T}$.

$$\beta(\underline{l}_n^{i,j}(m)) = 1, \quad \forall(i,j) \in \{(i,j)|(i,j)\in E_s, \tilde{s}(l_s^{i,j}) \ge \mathcal{T}\}, \forall n \in \{n|G_n \in \Phi\}, m=1,2,\ldots,L_n \tag{19}$$

In addition, the variables must hold the bound in Equation 9, and the followings:

$$z^{x,y}\left(\underline{l}_n^{i,j}(m)\right) \in \{0,1\}, \quad \forall(x,y)\in E_s, \forall n \in \{n|G_n \in \Phi\}, m=1,2,\ldots,L_n, \forall(i,j)\in E_s \tag{20}$$

$$\beta(\underline{l}_n^{i,j}(m)) \in \{0,1\}, \quad \forall(i,j)\in E_s, \forall n \in \{n|G_n \in \Phi\}, m=1,2,\ldots,L_n \tag{21}$$

The formulated integer linear programs for off-peak link energy optimization by local link reconfiguration problem, either for splittable traffic (OL-LLs-F), or non-splittable traffic (OL-LLns-F), could be reduced to the problem discussed in [25] that is a simple two-commodity integer flow problem. It is proven at [25] that this simple two-commodity integer flow problem is $\mathcal{NP}$-hard. Hence, the formulated programs are $\mathcal{NP}$-hard.

C. Programs based on Semi Proportional Link Power Model

The previous section developed power saving programs for VNE's links, conforming to *Fixed* link power model. As it is discussed in Section III, *Semi Proportional* link power model defines a traffic adaptive power model for a physical link. Based on this link power model, large portion of consumed energy by any physical link is for keeping the link operational. Nonetheless, different from *Fixed* link power model, traffic load on the link also changes its power consumption. Hence, every physical link in the substrate network does not consume the same amount of energy. In this regard, it is possible reduce link power consumption by either setting the link into sleep mode, or by rerouting its traffic load to other physical links with higher bandwidth capacity. However, we could save larger amount of energy by sleeping the link, in comparison to rerouting its traffic load.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

The objective functions in previous programs are required to be modified, so they optimize the energy based on *Semi Proportional* link power model. In this regard, the objective function in Equation 3 needs to be replaced by Equation 22.

$$Minimize \sum_{(i,j)\in E_s} \left\{ \alpha(l_s^{i,j}) P^b(l_s^{i,j}) + \frac{r(l_s^{i,j})}{C_b(l_s^{i,j})} \left(P^m(l_s^{i,j}) - P^b(l_s^{i,j}) \right) \right\} \tag{22}$$

By modifying the objective function and keeping the same constraints and bounds, the programs reconfigure mapping of allocated VNs, according to *Semi Proportional* link power model defined in Equation 2.

Note that the off-peak link energy optimization by global link reconfiguration problem, which is formulated according to *Semi Proportional* link power model, is called OL-GLs-SP. Besides, the off-peak link energy optimization by local link reconfiguration problem that is defined according to *Semi Proportional* link power model is called OL-LLs-SP in the case of splittable traffic, and OL-LLns-SP in the case of non-splittable traffic.

V. Heuristic Algorithm

The discussed BILP for OL-LLns-F in Section IV-B2b is $\mathcal{NP}$-hard, and therefore the optimization solution is not scalable to large network sizes, due to its long executing time. In this section, we propose a scalable heuristic algorithm for OL-LLns-F.

Pseudo code of the proposed heuristic algorithm is shown in Algorithm 1. The algorithm checks the possibility of setting VNE's physical links into sleep mode during the off-peak time. This process must be done precisely in order to guarantee that network supports off-peak traffic demands of all the involved VNs. In this regard, the algorithm first calculates some metrics and then sorts the physical links in order to check the link removal possibility. Afterwards, it tries to find an alternative path for off-peak traffic of every allocated traffic capacity to a virtual link in the removed physical link. This process is done in multiple phases.

During the off-peak period, the available bandwidth capacity in $l_s^{i,j}$ is represented by $\breve{C}_b(l_s^{i,j})$. $\breve{C}_b(l_s^{i,j})$ is equal to its physical capacity subtracted by total reserved off-peak traffic capacities for virtual links in $l_s^{i,j}$. Equation 23 defines $\breve{C}_b(l_s^{i,j})$. Besides, G_s^T is off-peak substrate topology. At first, G_s^T is the same as substrate network topology.

$$\breve{C}_b(l_s^{i,j}) = C_b(l_s^{i,j}) - \sum_{\{n|G_n\in\Phi\}} \sum_{m=1}^{L_n} \acute{r}_n^{i,j}(m) \tag{23}$$

Because the substrate links with higher stress rate are more essential in regard to traffic demands and the possible interruptions, the algorithm starts setting substrate links into sleep mode from the link that has the lowest stress rate. It sorts the substrate links with $\tilde{s}(l_s^{i,j}) < \mathcal{T}$ in ascending order based on $\tilde{s}$. The list is represented by S_L.

In the next phase, the algorithm removes the physical links that are capable to be set into sleep mode, from G_s^T, and at the end it returns G_s^T as the energy-efficient off-peak substrate topology. This phase also ensures the rearranged network accommodates the off-peak traffic demands. In this regards, there must be a single replaced path for each removed traffic capacity that supports its off-peak traffic demand. The algorithm tries to find such an alternative path for every allocated traffic capacity to a virtual link in every physical link with $\tilde{s}(l_s^{i,j}) < \mathcal{T}$. The algorithm uses Dijkstra algorithm as the preferred routing algorithm to find the shortest alternative path, while every active physical link cost is assumed as 1. It is needed to check eligibility of every substrate link on the path in regard to the available off-peak bandwidth capacity. Three conditions might happen while the algorithm searches for such a replaced path:

- There is such an alternative path in G_s^T. So, the algorithm updates the respective $\breve{C}_b$, and allocates the respective traffic capacity in all the substrate links over the path.
- There is an alternative path in G_s^T, but one or some of the substrate links over the path do not support the off-peak traffic demand. Therefore, the algorithm places the respective physical link back to G_s^T, cancels all the previous capacity and traffic modifications, and aborts checking process for rest of the allocated traffic capacities in this physical link.
- There is no alternative path in G_s^T. Hence, the algorithm places back the respective physical link to G_s^T, cancels all the previous capacity and traffic modifications, and aborts checking process for rest of the allocated traffic capacities in this physical link.

After checking process for all of the removed physical

Algorithm 1 Heuristic Algorithm for OL-LLns-F

for all (i,j) such that $(i,j) \in E_s$ **do**
    if $\tilde{s}(l_s^{i,j}) < \mathcal{T}$ **then**
        place the link in S_L in ascending order based on $\tilde{s}$
    end if
end for
for all (i,j) such that $l_s^{i,j}$ is the top unchecked link in S_L **do**
    remove (i,j) from G_s^T
    for all n such that $G_n \in \Phi$ **do**
        for all m such that $m = 1, 2, \ldots, L_n$ **do**
            if there is an alternative path from node i to node j (by Dijkstra) in G_s^T **then**
                for all (x,y) such that $l_s^{x,y}$ is on the alternative path **do**
                    $\breve{C}_b(l_s^{x,y}) = \breve{C}_b(l_s^{x,y}) - \acute{r}_n^{i,j}(m)$
                    if $\breve{C}_b(l_s^{x,y}) < 0$ **then**
                        $\breve{C}_b(l_s^{x,y}) = \breve{C}_b(l_s^{x,y}) + \acute{r}_n^{i,j}(m)$
                        place (i,j) back to G_s^T
                        undo all the previous capacity and traffic modifications respective to $l_s^{i,j}$
                        break and go for next substrate link in S_L
                    else
                        $\acute{r}_n^{x,y}(m) = \acute{r}_n^{x,y}(m) + \acute{r}_n^{i,j}(m)$
                    end if
                end for
            else
                place (i,j) back to G_s^T
                undo all the previous capacity and traffic modifications respective to $l_s^{i,j}$
                break and go for next substrate link in S_L
            end if
        end for
    end for
end for
return G_s^T

links, G_s^T is returned as the energy-efficient off-peak substrate topology.

It is expected the suggested heuristic for OL-LLns-F is much simpler and faster than the BILP. The largest loop, starts in line 6 and ends in line 29, determines the complexity of the proposed heuristic. This loop runs for every physical link, so its complexity is $\mathcal{O}(|E_s|)$. The first sub-loop starts in line 8 runs for every VN, and therefore its complexity is $\mathcal{O}(|\Phi|)$. The second sub-loop starts in line 9 runs for every virtual link in the respective virtual network. Considering the worst case, the complexity of this sub-loop is $\mathcal{O}(|E_v^m|)$, where E_v^m is the set of edges of the involved virtual network with the largest number of virtual links. The heuristic calls Dijkstra algorithm in line 10. The complexity of Dijkstra algorithm in the worst case is $\mathcal{O}\left(|E_s| + |V_s|logV_s|\right)$. The third sub-loop starting in line 11 checks the capability of every substrate link on the found path. So, its complexity is $\mathcal{O}(|E_s|)$. In the worst case scenario, the heuristic might need to check all the physical links again, in order to undo the capacity and traffic modifications for each re-allocated traffic capacity. So, complexity of the undoing function is $\mathcal{O}(|E_s||\Phi||E_v^m|)$. Hence, the complexity of the proposed heuristic is $\mathcal{O}\left(|E_s|^3|\Phi|^2|E_v^m|^2(|E_s| + |V_s|log|V_s|)\right)$. Consequently, the proposed heuristic algorithm is much simpler and it could be solved in a polynomial time.

VI. Evaluation

The proposed energy saving solutions are supposed to reduce total link power consumption in VNE during off-peak hours. However, they need to guarantee the off-peak traffic requirements. In order to evaluate their effectiveness, several random VNE setups have been evaluated.

Recently, Waxman algorithm [26] is widely used by the researchers to generate random virtual/substrate topologies for VNE [13], [14], [27]–[29]. Therefore, in this paper, substrate and virtual networks' topologies are generated by Waxman algorithm. Waxman generates random network topologies based on two parameters, λ and μ. As λ grows the probability of having an edge between any nodes in the topology is increased. As μ grows there is a larger ratio of long edges to short edges. In this paper, we choose the Waxman parameters, for both substrate and virtual networks' topologies, as $\lambda = \mu = 0.5$, in the area size of 100×100. After creating random substrate and virtual networks' topologies, the substrate links' capacity and virtual links' peak demand are generated randomly with a uniform distribution. The bandwidth capacity of each physical link is a random amount between 100Mbps and 200Mbps, but each virtual link's bandwidth demand is generated randomly between 40Mbps and 80Mbps. Both randomly generated substrate and virtual networks are symmetric, so if there is a link from ith node to jth node with a specific amount of bandwidth capacity, there is also a link from jth node to ith node with the same amount of bandwidth capacity. In the next step, the created virtual nodes are mapped to the substrate nodes randomly with the uniform distribution. Afterwards, every generated virtual link's peak bandwidth demand is allocated on a substrate path through a state-of-art heuristic algorithm.

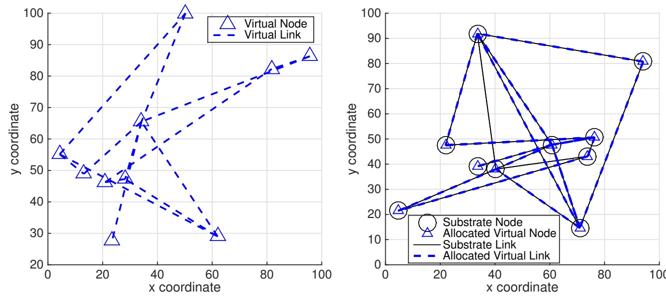

Fig. 2 (a) A Random VN (b) A Random Substrate Network

For example, Figure 2a shows a randomly generated virtual network. Besides, Figure 2b demonstrates a randomly generated substrate network that includes allocated nodes and links of the generated virtual network in Figure 2a.

As it is discussed, the formulated ILPs are $\mathcal{NP}$-hard, so they are not scalable to large network sizes. Therefore, we assess capability of the defined ILPs on small random simulation setups, similar to the other related works in [13]–[15], [22]. The ILPs are solved by MOSEK solver [30]. Nonetheless, theoretical complexity analysis reveals the proposed heuristic algorithm is much simpler, and therefore it is scalable to large network sizes. Hence, performance of the suggested heuristic is examined on large random simulation setups.

Every small random simulation setup contains 10 randomly generated VNEs. Each VNE in a small random simulation setup has 2 random virtual networks that are allocated on a single random substrate network, while every substrate and virtual network has 10 nodes. The average number of physical links in the small random simulation setups is 30. Furthermore, every large random simulation setup includes 10 randomly generated VNEs. All the VNEs in a large random simulation setup have at least 2 random virtual networks that are mapped on 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 590. 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 solved the formulated MILPs for OL-GLs-F, and OL-LLs-F on a small random simulation setup, while traffic is assumed to be splittable. Both have been solved for different amounts of off-peak traffic ratio. Off-peak traffic ratio is the fraction of network's off-peak traffic rate by its peak traffic rate. The average number of physical links in sleep mode during the off-peak period has been probed and shown in Figure 3a. The results illustrate both of OL-GLs-F and OL-LLs-F are able to set notable number of physical links into sleep mode during this time. Besides, the number of physical links in sleep mode is decreasing by increasing the off-peak traffic ratio. This is because increasing off-peak traffic ratio increases the amount of traffic programs need to re-allocate, so they are more limited in terms of finding alternative paths.

OL-GLs-F is expected to deliver the most optimum level of energy saving. However, different from OL-GLs-F, OL-LLs-F enables the providers to adjust level of the reconfiguration and control the possible traffic disruptions. This is possible

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

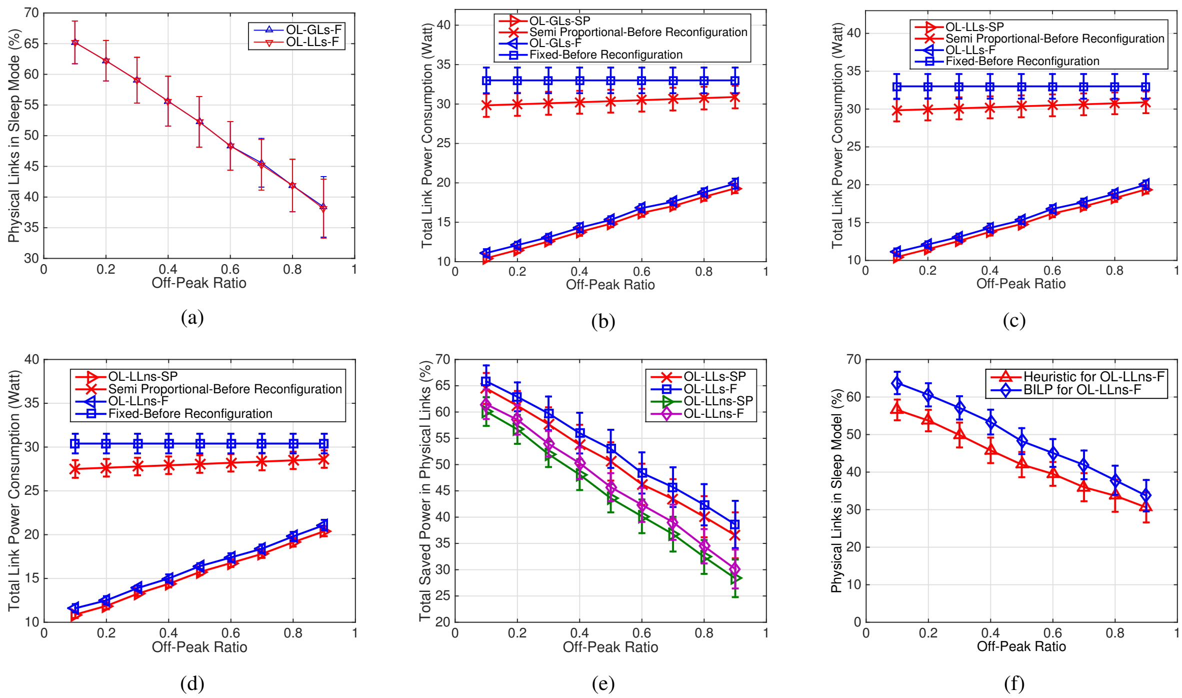

Fig. 3. (a) Off-peak link energy optimization by global link reconfiguration vs. local link reconfiguration (splittable traffic). (b) Total link power consumption of off-peak link energy optimization by global link reconfiguration. (c) Total link power consumption of off-peak link energy optimization by local link reconfiguration for splittable traffic. (d) Total link power consumption of off-peak link energy optimization by local link reconfiguration for non-splittable traffic. (e) Total saved power with off-peak link energy optimization by local link reconfiguration for splittable and non-splittable traffic. (f) BILP vs. heuristic for off-peak link energy optimization by local link reconfiguration (non-splittable traffic).

through the constraint in Equation 13 that prevents the local link reconfiguration solution to modify the allocated traffic capacities in physical links with stress rate larger than a threshold, set by the providers. Although this approach could help decreasing the possible traffic disruptions, it needs to provide the possibility of achieving the maximum energy saving level for the providers. Since the size of generated VNEs are small in the small simulation setup, the stress rate for most of the physical links is less than chosen stress rate threshold of 0.6. Therefore, Figure 3a confirms when the constraint in Equation 13 is relaxed, OL-LLs-F could achieve the same energy saving level as OL-GLs-F.

In addition, we measured total link power consumption for both off-peak link energy optimization by global link reconfiguration and local link reconfiguration solutions on a small random simulation setup. This measurement has been done according to both *Fixed* and *Semi Proportional* link power models. The total link power consumption has been measured for off-peak ratio range of 0.1 to 0.9, before and after applying the proposed energy saving solutions. According to [22] and because the random physical links' capacities are generated uniformly in the range of 100-200Mbps, P^b is 0.9Watt and P^m is 1.0Watt for any physical link. The measurement results for off-peak link energy optimization by global link reconfiguration, and off-peak link energy optimization by local link reconfiguration for splittable and non-splittable traffic, are shown in Figures 3b, 3c, and 3d, respectively.

The results in Figures 3b, 3c, and 3d, demonstrate all of the formulated programs are able to reduce VNE's link power consumption, effectively. Considering the results of when *Fixed* link power model is used, the total link power consumption for any off-peak traffic rate is constant, while no energy saving solution is employed. Nevertheless, the power consumption is changing with the traffic rate even before applying any energy saving solution, when *Semi Proportional* link power model is used. By applying any of the proposed energy saving techniques (global/local link reconfiguration), the total link power consumption based on both link power models will be decreased. Note that increasing the off-peak ratio raises the total link power consumption, as the larger number of links will be left activated.

In the same simulation setup, we calculated the percentage of power saved in physical links, by the formulated programs for off-peak link energy optimization by local link reconfiguration problem. This is tested in the case of splittable and non-splittable traffic, for both *Fixed* and *Semi Proportional* link power models. The results are shown in Figure 3e. The outcome of programs shows the rate of power we could save in physical links is decreasing when the off-peak ratio is increasing, because the programs are more limited in terms of finding alternative paths. Besides, Figure 3e illustrates it is probable to save higher amounts of energy when traffic is splittable, in comparison to when traffic is non-splittable. This is because the programs are more flexible in terms of finding alternative paths when they could split the traffic to multiple paths. Moreover, Figure 3e confirms we could reach higher rates of energy saving when the objective function is formulated according to the *Fixed* link power model, in comparison to when the objective function is formulated based on *Semi Proportional* link power model. This is mainly due to two reasons. First, since the power consumption is varying based on traffic load in *Semi Proportional* link power model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

rather than being a constant amount based on *Fixed* link power model, a physical link's power consumption in the case of *Semi Proportional* link power model is less than when *Fixed* link power model is employed. Consequently, the amount of saved energy with *Semi Proportional* link power model might be less than the amount of saved energy with *Fixed* link power model, if the program sets the same physical links into sleep mode. In addition, if the energy saving program sets a physical link into sleep mode, it needs to find an alternative path to support the off-peak traffic of the removed link. In the case of *Semi Proportional* link power model, the rerouted traffic increases the power consumption over the alternative path. However, in the case of *Fixed* link power model, since traffic load does not affect link power consumption, the rerouted traffic does not increase the power consumption over the alternative path.

Furthermore, we solved the formulated BILP for OL-LLns-F and compared its ability in terms of sleeping physical links to the proposed heuristic for the same problem. The average results are measured for different off-peak traffic ratios on a small random simulation setup, and shown in Figure 3f. The BILP results set the optimum points, while the heuristic algorithm still reveals reasonable results. However, the heuristic algorithm is much simpler and faster in terms of required run time. Note that in Figure 3f, the difference between BILP and heuristic results is smaller when off-peak traffic ratio is high, because there are less energy saving opportunities when off-peak traffic rates are high. When off-peak ratio is low, it is more probable to find alternative paths for off-peak traffic demands, and as the BILP is more effective than the heuristic, the BILP saves higher levels of energy in comparison to the heuristic. Nevertheless, when off-peak ratio is high, there will be less alternative options to reallocate the traffic, so the BILP and the heuristic work more closely.

It is also important to evaluate effectiveness of the proposed heuristic algorithm for OL-LLns-F. We assess the ability of the heuristic based on different factors on the defined large random simulation setups.

The heuristic is formulated based on *Fixed* link power model. In this regard, the total link power consumption is measured for different off-peak ratios, before and after applying the proposed heuristic on a large random simulation setup. The average results are shown in Figure 4a. The power consumption is constant before applying the heuristic. By applying the heuristic, the total VNE's link power consumption will be reduced. Note that increasing the off-peak ratio raises the total link power consumption, as the larger number of links are left active.

Besides, it is essential to check ability of the heuristic for different numbers of involved virtual networks, as increasing the number of virtual networks adds several constraints in terms of sleeping a single physical link. The ability of the proposed heuristic on setting physical links into sleep mode during the off-peak hours, is tested over two large random simulation setups. In the first large random setup, each VNE contains two virtual networks, but in the second large random setup, every VNE includes three virtual networks. The effectiveness of the algorithm is evaluated for the range of off-peak traffic ratios. The average results for both setups are shown in Figure 4b. Figure 4b shows it is probable to set notable number of physical links during the off-peak period, by implementing the suggested heuristic algorithm. For the first setup, when the off-peak traffic ratio is 0.1, the proposed heuristic sets 89.1230% of the physical links into sleep mode. This happens while the reconfiguration heuristic still accommodates off-peak traffic demands of involved VNs. In addition, Figure 4b confirms mapping an extra virtual network onto the substrate network degrades the ability of heuristic in terms of saving power. This is because the algorithm assesses the allocated traffic capacities to every virtual link in each substrate link in order to find a replaced path. By adding new virtual networks, new virtual links are mapped onto physical links, and therefore there are more constraints for the algorithm. Consequently, smaller number of physical links is capable to be set into sleep mode over off-peak hours. Moreover, decreasing off-peak ratio decreases the difference between outcome of the first and second simulation setups. When off-peak traffic rate is low, the programs are more flexible in terms of finding alternative paths. Hence, the energy saving ability of the programs is less affected by adding extra virtual networks when off-peak ratio is low, in comparison to when we have high off-peak traffic rates.

Moreover, it is explained in Section V that the proposed heuristic, similar to the formulated BILP for OL-LLns-F, does not re-allocate the allocated traffic capacities in physical links with $\tilde{s}(l_s^{i,j}) \geq \mathcal{T}$, in order to decrease service disruption due to reconfiguration. Figure 4c studies the effect of changing stress rate threshold $\mathcal{T}$ on capability of the heuristic for setting physical links into sleep mode, over a large random simulation setup. Figure 4c shows decreasing $\mathcal{T}$, decreases number of physical links the heuristic sets into sleep mode, because smaller number of substrate links are considered for power saving. Although setting smaller $\mathcal{T}$ decreases amount of power the solutions could save, it reduces the traffic interruptions, due to reconfiguration. Consequently, the providers could control possible traffic interruptions by adjusting $\mathcal{T}$. Note that because of the specific chosen amounts of physical and virtual links' bandwidth capacity in our defined large random simulation setups, the stress rate of most of the physical links in the considered simulation setup is less than 0.3. Consequently, the heuristic outcome is almost constant for $\mathcal{T}$ of greater than 0.3.

Additionally, as mentioned in Section II, the suggested heuristic for OL-LLns-SP is expected to be more effective compared to our previous work [15] and other similar studies in [8], [21]. We compared the outcome of the proposed heuristic in this paper to our previous algorithm in [15], over a large random simulation setup. As the result is clear in Figure 4d, this link reconfiguration algorithm is able to set higher number of physical links into sleep mode over off-peak hours, with the same constraints. This is because the methods in [8], [15], [21] does not modify the allocation of mapped virtual networks, while they only reroute the traffic to the already allocated traffic capacities to virtual links. Nonetheless, in this paper, we reconfigure mapping of virtual links to reach higher energy saving rates.

Furthermore, re-allocating the traffic capacities in the other

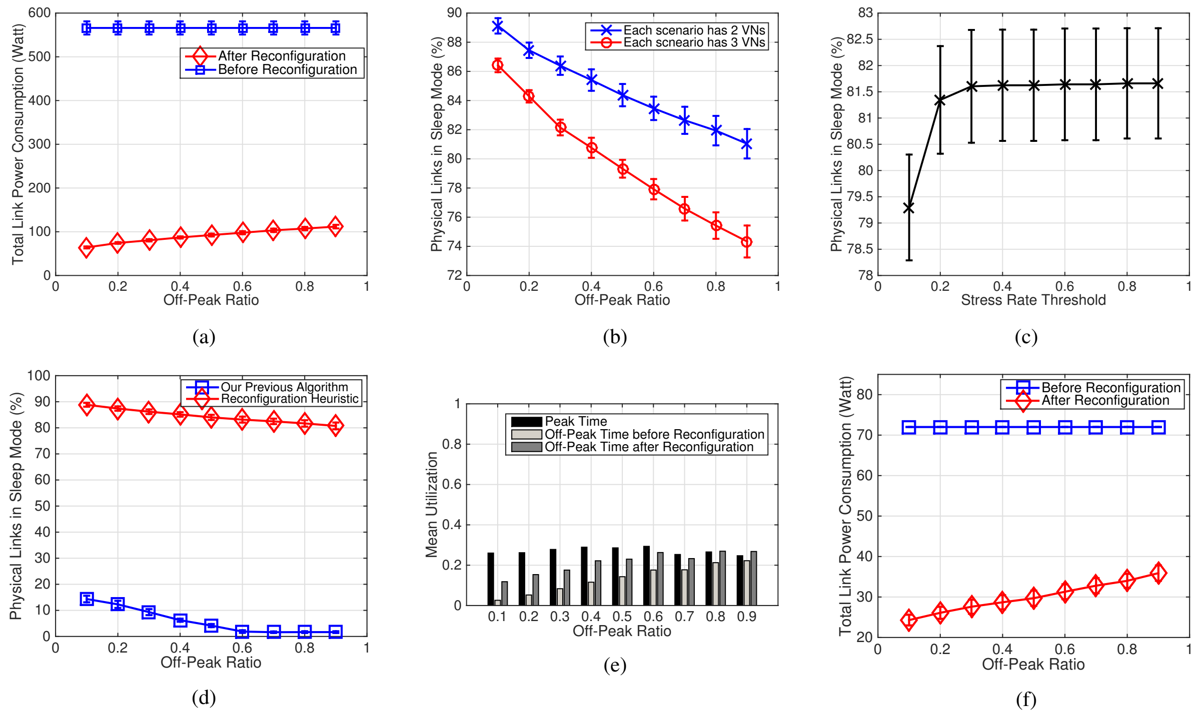

Fig. 4. (a) Total link power consumption of off-peak link energy optimization by local link reconfiguration heuristic. (b) Link reconfiguration heuristic for different numbers of involved VNs. (c) Effect of changing stress rate threshold on heuristic's outcome (d) Link reconfiguration heuristic vs. our previous algorithm. (e) Mean link utilization over different configurations. (f) Total link power consumption before and after applying the heuristic on the GÉANT simulation setup.

substate links causes changes to the link utilization. Accordingly, it is needed to make sure the increased utilization is controlled and does not cause congestion. Link utilization for three different configurations is tested on a large random simulation setup, and the average results are shown in Figure 4e. The first configuration is for peak time when allocated bandwidth is able to handle "Worst-Case" scenarios. The second configuration is for the off-peak period while no energy saving algorithm is implemented. Over this period the links are less utilized while the same bandwidth capacity is allocated and consume the same power as the peak time. After applying our suggested heuristic, the average link utilization is increased, but it is still less than the maximum utilization.

It is also necessary to validate the effectiveness of our proposed approach against a real topology. In this regard, we tested the proposed heuristic on a new random simulation setup that contains 10 randomly generated VNEs. Each VNE in this new random simulation setup, has 2 random virtual networks that are mapped onto the GÉANT network topology [31], while every virtual network has 10 nodes. GÉANT has 22 nodes and 36 bidirectional links. We considered GÉANT network as the substrate network, because it is a real universal topology. Figure 4f shows the results for this setup. The results confirm the heuristic is able to effectively reduce the network's link power consumption.

All the formulated ILPs in this paper are $\mathcal{NP}$-hard, while complexity of the proposed heuristic algorithm is $\mathcal{O}\left(|E_s|^3|\Phi|^2|E_v^m|^2(|E_s|+|V_s|log|V_s|)\right)$. So, it is expected the proposed heuristic needs less run time in comparison to the formulated integer linear programs. We verified run time for each method (when the objective is defined based on *Fixed* link power model) on a single small random VNE that has a substrate network and 2 virtual networks, while each has 10 nodes. The run time is measured for each formulated optimization program as well as the heuristic, when off-peak ratio is 0.5. The run time of the MILPs for OL-GLs-F and OL-LLs-F are 215.7 seconds and 3,302.4 seconds, respectively. Besides, the run time of the BILP for OL-LLns-F is 20,322.0 seconds. However, the run time of the heuristic for OL-LLns-F is only 0.0409 seconds, which is so small in comparison to the required run time for the formulated BILP of the same problem. The run time for local link reconfiguration BILP for non-splittable traffic is higher than the MILP of the same problem for splittable traffic. This is because the BILP needs to find an alternative path for every allocated traffic capacity in substrate links, while the MILP has to find an alternative path for every bundled traffic capacity. Therefore, the BILP for non-splittable traffic has more constraints than the MILP for splittable traffic. In addition, the run time of local link reconfiguration MILP for splittable traffic is larger than the run time for global link reconfiguration MILP, since the local link reconfiguration is more complex and has larger number of constraints than the global link reconfiguration.

The simulation results prove the suggested energy saving solutions are able to reduce VNE's link power consumption, during the off-peak period, effectively. Besides, the proposed heuristic is a simple and fast algorithm that works closely to the optimum points. Note that every simulation setup is quite large to cover a substantial number of random topologies in order to verify the effectiveness of the proposed solutions. Besides, the calculated confidence intervals confirm the results are precise enough to reveal significances of suggested energy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2440246, IEEE Transactions on Cloud Computing

saving methods.

VII. Conclusion

ICT's energy consumption is growing fast, according to the latest published reports. ISPs are needed to expand their infrastructure in order to handle the higher traffic load. VNE technology helps to slow down the infrastructure expansion. Nonetheless, it is also essential to have energy saving techniques that decrease VNE's energy consumption. In this paper, we discussed multiple novel energy saving solutions that globally/locally optimize VNE's link power consumption, during off-peak time. The proposed fine-grained local reconfiguration enables the providers to adjust level of the reconfiguration, and accordingly control the possible traffic disruptions. An Integer Linear Program (ILP) is formulated for each problem. Since the ILPs are $\mathcal{NP}$-hard, a novel heuristic algorithm is also proposed. Simulation results show the energy saving solutions are noticeably effective and the heuristic achieves closely to the optimum points. Because physical nodes are also essential power consumers in VNE, it is necessary to develop energy saving techniques that minimize both node and link power consumption in VNE, in the future works,.

References

[1] G.-Q. Zhang, G.-Q. Zhang, Q.-F. Yang, S.-Q. Cheng, and T. Zhou, "Evolution of the internet and its cores," *New Journal of Physics, IOP Publishing*, vol. 10, no. 12, p. 123027, 2008.
[2] 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?" *Communications Magazine, IEEE*, vol. 49, no. 8, pp. 80–86, 2011.
[3] J. S. Turner and D. E. Taylor, "Diversifying the internet," in *Global Telecommunications Conference (GLOBECOM), IEEE*, vol. 2, 2005, pp. 6 pp.–760.
[4] N. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks, Elsevier*, vol. 54, no. 5, pp. 862–876, 2010.
[5] 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.
[6] H. Feng and Y. Shu, "Study on network traffic prediction techniques," in *International Conference on Wireless Communications, Networking and Mobile Computing, IEEE*, 2005, pp. 1041–1044.
[7] A. S. San-Qi, "A predictability analysis of network traffic," *Computer Networks, Elsevier*, vol. 39, no. 4, pp. 329–345, 2002.
[8] 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.
[9] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *International Conference on Communications (ICC), IEEE*, 2009, pp. 1–6.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] J. F. Botero and X. Hesselbach, "Greener networking in a network virtualization environment," *Computer Networks, Elsevier*, vol. 57, no. 9, pp. 2021–2039, 2013.
[15] 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.
[16] 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.
[17] 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 architecuture, IEEE Press*, 2014, pp. 301–312.
[18] C. Xin, B. Wang, X. Cao, and J. Li, "Logical topology design for dynamic traffic grooming in wdm optical networks," *Journal of Lightwave Technology, IEEE*, vol. 24, no. 6, p. 2267, 2006.
[19] 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.
[20] 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.
[21] 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.
[22] 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.
[23] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, *A power benchmarking framework for network devices*, ser. NETWORKING, Springer, 2009, pp. 795–808.
[24] C. Gunaratne, K. Christensen, and B. Nordman, "Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed," *International Journal of Network Management, Wiley Online Library*, vol. 15, no. 5, pp. 297–310, 2005.
[25] 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.
[26] B. M. Waxman, "Routing of multipoint connections," *Selected Areas in Communications, IEEE*, vol. 6, no. 9, pp. 1617–1622, 1988.
[27] A. Fischer, J. F. B. Vega, M. Duelli, D. Schlosser, X. H. Serra, and H. D. Meer, "Alevin-a framework to develop, compare, and analyze virtual network embedding algorithms," *Open-Access-Journal Electronic Communications of the EASST*, 2011.
[28] J. Zhu and T. Wolf, "Vnmbench: a benchmark for virtual network mapping algorithms," in *21st International Conference on Computer Communications and Networks (ICCCN), IEEE*, 2012, pp. 1–8.
[29] M. T. Beck, A. Fischer, H. de Meer, J. F. Botero, and X. Hesselbach, "A distributed, parallel, and generic virtual network embedding framework," in *International Conference on Communications (ICC), IEEE*, 2013, pp. 3471–3475.
[30] 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.
[31] "Geant; the pan european data network project."

Ebrahim Ghazisaeedi received his M.Sc. degree in Mobile and Satellite Communications from the University of Surrey, England, in 2011. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at the Department of Systems and Computer Engineering, Carleton University, Canada. His main research interests are in communication networks, network virtualization, and network optimization.

Changcheng Huang received his Ph.D. degree in Electrical Engineering from Carleton University, Canada, in 1997. Since July 2000, he has been with the Department of Systems and Computer Engineering at Carleton University, where he is currently a professor. His research interests are stochastic control in computer networks, resource optimization in wireless networks, reliability mechanisms for optical networks, network protocol design and implementation issues.