# Optimizing Link Sleeping Reconfigurations in ISP Networks with Off-Peak Time Failure Protection

Frederic Francois, Ning Wang, Klaus Moessner, and Stylianos Georgoulas

***Abstract*—Energy consumption in ISP backbone networks has been rapidly increasing with the advent of increasingly bandwidth-hungry applications. Network resource optimization through sleeping reconfiguration and rate adaptation has been proposed for reducing energy consumption when the traffic demands are at their low levels. It has been observed that many operational backbone networks exhibit regular diurnal traffic patterns, which offers the opportunity to apply simple time-driven link sleeping reconfigurations for energy-saving purposes. In this work, an efficient optimization scheme called Time-driven Link Sleeping (TLS) is proposed for practical energy management which produces an optimized combination of the reduced network topology and its unified off-peak configuration duration in daily operations. Such a scheme significantly eases the operational complexity at the ISP side for energy saving, but without resorting to complicated online network adaptations. The GÉANT network and its real traffic matrices were used to evaluate the proposed TLS scheme. Simulation results show that up to 28.3% energy savings can be achieved during off-peak operation without network performance deterioration. In addition, considering the potential risk of traffic congestion caused by unexpected network failures based on the reduced topology during off-peak time, we further propose a robust TLS scheme with Single Link Failure Protection (TLS-SLFP) which aims to achieve an optimized trade-off between network robustness and energy efficiency performance.**

***Index Terms*—Energy-aware network management, green networks, traffic engineering, link sleeping, network robustness.**

## I. Introduction

Computer networks are becoming more and more pervasive in our modern society where an increasing number of network devices are being connected to the public Internet in order to support the delivery of exponentially increasing volume of digital content objects. On the other hand, new technologies for energy efficiency in network devices have not been able to sufficiently offset the rapid increase in the overall energy consumption of todays computer networks. High energy consumption creates new practical issues for network operators in their daily operations, such as increased network operating costs and also high greenhouse gas emissions that threat our living environment as a long-term effect. These problems are likely to become even worse in the future if no green networking technologies become available, due to the increasing popularity of bandwidth-hungry applications such as Internet television (IPTV) services deployed across the global Internet.

During the last decade, various techniques have been proposed for making wired network infrastructures such as ISP backbone networks more energy efficient. These techniques can be classified as either *disruptive*, where completely new features need to be supported by device hardware such as dynamic voltage scaling (DVS) [1]–[4], or alternatively *evolutionary*, where incremental approaches such as link sleeping reconfigurations can be leveraged based on *legacy* network systems [5]–[7]. While disruptive approaches aim at long-term solutions for greening the future Internet, short or mid-term evolutionary schemes are of equal significance, as the environmental issue and the scarcity of energy resources are already regarded as imminent threats.

A distinct example of evolutionary approaches is *energy-aware network management* based on legacy network infrastructures. As a more specific paradigm, the concept of *energy-aware traffic engineering* (ETE) has been proposed in the literature, which relies on traffic optimization techniques to achieve energy efficiency in operational networks. In general, ETE optimizes the distribution of traffic among different paths within the network so that the overall energy consumption can be reduced while still maintaining the targeted service guarantee to end users. This is typically achieved through careful reconfiguration of network devices, for instance by putting a subset of nodes/links to sleep mode, or by adaptively reducing the traffic transmission rate when the network is relatively idle with low traffic load. In this work, we propose a practical and efficient approach with time-driven network topology reconfigurations called *Time-driven Link Sleeping* (TLS). The rationale behind is that, despite the existence of traffic dynamics in normal network operations, most ISP networks have regular and predictable diurnal traffic behavioral patterns [7]–[12]. Such a feature offers the opportunity for simple network topology control rather than relying on complex online ETE paradigms based on continuous network monitoring. The proposed TLS scheme aims to compute a reduced network topology which will produce the *optimized combination of the number of sleeping links and the unified duration of their sleeping period*. Such period is effectively off-peak operation time as compared to peak time period when the full network topology is applied. By unified duration we mean that the start and the end of the sleeping period (i.e. the off-peak

Manuscript received May 14, 2012; revised August 8, 2012. The associate editor coordinating the review of this paper and approving it for publication was F. De Turck.

The research leading to these results has been performed within the UniverSelf project (www.UniverSelf-project.eu) and received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257513.

time window) on a daily basis are always kept the same. This energy-saving approach is not only feasible but also attractive for actual deployment due to its simplicity in network reconfigurations. First of all, ISP network resources are normally over-provisioned in order to safely accommodate peak-time traffic demands as the worst case scenario. Such a strategy provides opportunities for reducing the working network topology during the off-peak time and saving energy while still maintaining the similar degree of over-provisioning that occurs during peak-time operation. A distinct feature of our proposed TLS paradigm is that the (pre-determined) time window between the activation of the reduced topology reconfiguration and its termination is unified on a daily basis. Such a time-driven reconfiguration approach significantly eases the corresponding operations for energy saving on the network operator side. The enforcement of the normal peak-time topology and the reduced topology can directly be achieved based on existing multi-topology routing protocols such as MT-OSPF [13] and MT-ISIS [14] for avoiding routing disruptions during the transient topology switching period.

Based on such multi-topology aware routing protocols, the transition between the two topologies at the beginning/end of the off-peak time window is performed through the scheduled remarking of multi-topology identifiers (MT-IDs) carried by flow packets. From signaling point of view, such scheduled transition can be fulfilled by any manager-agent based network management protocol such as SNMP, where the central manager can signal to all ingress routers to simultaneously remark packets for topology switching.

Intuitively, removing sleeping network devices from the working topology reduces the network capability in carrying customer traffic. In case of further unexpected network failures occurring in the residual topology during off-peak time, the operational network may become vulnerable to post-failure traffic congestion. Effectively, single link failures are known to occur frequently in operational networks [15], and indeed it is important to eliminate, or at least minimize such risk when computing the reduced topology for energy saving purpose. Towards this end, we further extend the basic TLS scheme to take into account single link failure protections, and propose an advanced scheme called *Time-driven Link Sleeping with Single Link Failure Protection* (TLS-SLFP). TLS-SLFP ensures that in the event of any single link failure scenario during the off-peak time, the network still remains always fully connected and also that sufficient network resources are available for carrying ongoing traffic demands without service disruptions even with the reduced network topology.

To realistically evaluate the proposed TLS and TLS-SLFP schemes, we use in our simulation the GÉANT network topology which consists of 23 Points-of-Presence (PoP) nodes and 74 links, together with its historical traffic matrices over five consecutive days for performance evaluation. The results show that TLS is able to achieve significant energy savings for a variety of constraint policies on the maximum allowable link utilization. For instance, 18.6% energy savings over 5 days can be achieved when the allowable Maximum Link Utilization (MLU) constraint in the off-peak window is set to 80%. In this case, around 45% of the links can be reconfigured to sleep mode, and the daily off-peak time duration using this reduced network topology configuration is 10 out of 24 hours. By taking into account traffic uncertainty, we also introduce traffic scaling factor which is used to scale up traffic demands based on the real traffic matrix by multiplying every traffic demand between each Source-Destination pair by a given value larger than 1.0. According to our experiments, there are still sufficient opportunities for energy savings even if the off-peak traffic demands are scaled up to 30%. Such energy savings are achieved through a simple time-driven network reconfiguration operation, but without imposing the need for any additional intelligence or complexity to the network itself. It is also our observation (and expectation) that the TLS-SLFP scheme achieves less energy-saving gains compared to plain TLS, due to the more conservative configuration strategy against potential network failures. On the other hand, there still exist opportunities for energy savings in this case, and the trade-off between network robustness and energy efficiency will be quantified in the performance evaluation section. To the best of knowledge, the proposed TLS and TLS-SLFP schemes are the first to jointly optimize link sleeping and sleeping duration for energy saving in ISP networks, with additional awareness of network robustness against network failures in the latter scheme.

The rest of the paper is organized as follows. In Section II, an overview of existing ETE methods is presented along with the analysis of the characteristics of offline and online ETE approaches. Section III introduces the problem formulation that TLS aims to solve. The specification of TLS is described in detail along with the evaluation of the algorithm based on the operational network GÉANT. Section IV describes TLS-SLFP which is an extension of TLS which takes into account single link failure protection during the off-peak time. Finally, Section V concludes the paper.

## II. Related Work

Reduction of power consumption in computer networks can be achieved through various strategies. The first strategy is the use of *green hardware* for networking devices. At the physical level, energy-efficient circuit design can be used [2], [16], [17], while at the functional level, dynamic frequency and voltage scaling can be applied [18]–[20]. Secondly, it is also possible to achieve energy efficiency through next-generation *green networking protocols*. Green protocols can allow, for example, proxies to take over the functionality of network devices that will be put into sleep mode in an autonomic manner [12], [21], [22]. Such an approach has been illustrated in [12] where a router can transparently transfer all the virtual links from one line card to another so that the former can be put to sleep. Another example is to extend the OSPF protocol by making a subset of routers share the routing table of other routers rather than calculating their own, so that fewer active links are used [21]. The third strategy is the use of *green network management paradigms*, such as ETE, where traffic demands are routed appropriately in order to promote either link rate adaptation [23]–[25] and/or link sleeping [7], [21], [26], [27] by taking into account the energy consumption characteristics of links and line cards. Future energy savings based on the exploitation of different possible energy consumption profiles of links have also been investigated in [20] and [28].

The principle of ETE was first introduced in [29], and there has been a comprehensive introduction and analysis of existing ETE solutions in [30] and [31]. ETE schemes can be classified into *offline* and *online* solutions. Offline ETE mainly refers to the computation of static network configurations (e.g. planned node/link sleeping) based on forecasted traffic behavioral patterns. On the other hand, online ETE can be used as a complementary approach in order to deal with short timescale traffic dynamics with on-the-fly network reconfigurations. Taking link/node sleeping reconfiguration as an example, most offline ETE paradigms aim to compute reduced network topologies based on one single traffic matrix (TM) for energy-saving purposes [20], [21]. However, due to the dynamics of traffic patterns in operational networks, existing offline ETE paradigms may exhibit inefficiency due to inaccurate traffic forecasting, as has always been the case in conventional offline TE approaches. In order to circumvent this limitation, complementary online ETE solutions have been designed to allow fast local network reconfigurations to handle rapid changes in traffic demands [23], [24], [29] but the high complexity in realizing these approaches has made them less attractive in practical deployments.

The majority of ETE solutions in the literature uses an offline optimization strategy [7], [21], [26], [27], [32] because of its simpler implementation compared to online energy saving methods [23], [24], [29]. Furthermore, offline ETE solutions can make use of global optimization methods because of their long timescale and this will result in more predictable network behavior compared to the case where local optimization methods are used by online ETE solutions [33].

## III. Time-driven Link Sleeping (TLS)

### A. Scheme Overview

Same as many other ETE schemes involving rerouting of customer traffic upon topology changes, in TLS two distinct network topologies are configured for daily operations. During the normal peak-time operation hours, the original *full topology* is applied for handling customer traffic, as it happens in common practice. During off-peak time, the *reduced network topology* that specifically excludes the scheduled sleeping links is used for energy saving purposes. In order to seamlessly perform topology switching without incurring routing disruption, we propose to use multi-topology routing protocols [13], [14] as the underlying routing platform. Specifically, two routing topologies are configured, one with the full physical topology while the other excluding the links scheduled to sleep. Once the operation has entered the scheduled off-peak time, individual routers simultaneously activate the reduced routing topology by remarking the multi-topology identifier (MT-ID) of packets from the default full topology MT-ID to the reduced topology MT-ID. As a result, if a router has detected from its forwarding table that the original valid next-hop towards the destination in the default full topology is not available in the reduced topology, then the alternative next-hop in the reduced topology towards that destination is used for actual packet forwarding. Of course, if the same next-hop is available in both topologies, it will be used in any case upon MT-ID remarking. When the scheduled off-peak time expires, all source routers remark customer packets back to the full topology MT-ID. The mapping of MT-ID to the two topologies can be flexibly determined by the network operator, for instance default MT-ID = 0 represents use of the fully peak-time topology, while an unreserved MT-ID code point can represent the reduced off-peak topology with the exclusion of sleeping links.

For the sake of overall performance optimality, a technical challenge for such a time-driven approach is that the reduced network topology should not be exclusively optimized with respect to the *number of sleeping links*, but it should also support *as long an operation period as possible* with this reduced network capacity, but without causing network congestion incurred due to customer traffic dynamics. It is not difficult to infer that there is a trade-off which needs to be optimized, between the *number* of sleeping links in the reduced topology and the *duration* of the off-peak time when this topology is actually enforced in operation. This is because, if an excessive number of links is excluded from the network based on a completely greedy approach focusing on link removal, then the resulting reduced topology is not able to handle even minor traffic increases, and in this case the full topology has to be restored, which leads to *very short duration* for the use of the reduced topology.

In order to determine the optimized combination of the reduced topology and its applied operation duration, it is important to consider how traffic behavioral patterns can play a role. Towards this end, traffic forecasting (based on monitoring) is needed which produces distinct traffic matrices at regular time intervals (e.g. every tens of minutes), as is the case in operational networks (GÉANT [34] and Abilene networks [35]). It should be noted that network configurations based on one single traffic matrix are not sufficiently robust in dealing with traffic dynamics, as this has been indicated in conventional traffic engineering paradigms [30]. Therefore, we consider multiple traffic matrices as input in order to more accurately produce the optimized combination of the reduced topology and its off-peak operation duration. To ease the *time-driven* reconfiguration operations at the ISP side, it is also desired that the off-peak configuration starts at exactly the same time and has the same duration on a daily basis (e.g. 7:00PM-7:00AM). According to our proposed solution, a single synthetic traffic matrix is computed based on multiple TMs located at the expansion points, which represent the lowest MLU within each day. This synthetic traffic matrix, along with the other traffic matrices at the expansion points, is used to determine the optimized set of sleeping links. Similarly, determining the off-peak duration on a daily basis also takes into account traffic matrix patterns during each day.

It is worth mentioning that, although traffic dynamics exhibit some similar patterns on a daily basis, it is also important to make sure the same off-peak topology can be applied in a unified time-driven manner even though there can be traffic pattern variations at the same (off-peak) time in each day. Input traffic matrices to TLS can also be scaled up, through the use of a traffic scaling factor, so as to enhance the robustness of the calculated reduced network topology to potential changes in the traffic patterns such as spikes.

In the next section, we will first formulate the overall

TABLE I
DEFINITION OF SYMBOLS

| Variable | Description |
|---|---|
| $A_z$ | Set of sleeping links in the reduced topology during off-peak time |
| $I_z$ | Set of consecutive traffic matrix intervals (time) supported with $A_z$ |
| $G(V,E)$ | Directed graph with $V$ being set of nodes and $E$ being set of links |
| $c_{ij}$ | Bandwidth capacity of link from node $i$ to $j$ |
| $t^{sd}$ | Traffic demand from node $s$ to $d$ |
| $f_{ij}^{sd}$ | Traffic demand from $s$ to $d$ that traverses link from $i$ to $j$ |
| $f_{ij}$ | Total traffic demand on link from $i$ to $j$ |
| $\alpha$ | Maximum allowable fractional utilization of link capacity |
| $P(t^{sd})$ | Set of links taken by $t^{sd}$ from node $s$ to $d$ |

optimization problem, followed by joint schemes of computing the reduced topology and determining its enforced duration. The outcome is an optimized network topology coupled with a unified time window for its configuration on a daily basis.

### *B. Problem Formulation and Proposed Algorithm*

*1) Problem Formulation:* The overall problem formulation on optimizing the reduced topology and its duration can be expressed as:

$$\text{maximize } max\{|I_z| \times |A_z|\} \tag{1}$$

subject to:

$$\sum_{j=1}^{|V|} f_{ij}^{sd} - \sum_{j=1}^{|V|} f_{ji}^{sd} = \begin{cases} t^{sd}, & \forall\ s,d,i=s \\ -t^{sd}, & \forall\ s,d,i=d \\ 0, & \forall\ s,d,i\neq s \end{cases} \tag{2}$$

$$f_{ij} < \alpha c_{ij}, \quad \forall\ i,j \tag{3}$$

$$P(t^{sd}) \cap A_z = \emptyset, \quad \forall\ s,d,z \tag{4}$$

Equation (1) is the objective function which jointly optimizes the reduced topology (in terms of the number of sleeping links) and the duration of its configuration (in terms of the number of *consecutive* traffic matrix intervals which can be supported by that reduced topology). The overall off-peak time duration can be calculated from the number of traffic matrices, which are captured at *regular* intervals covered by the period. Constraint (2) is the standard flow conservation constraint in IP routing. Constraint (3) ensures that all active links in the reduced topology have utilization below a given threshold determined by the ISP. That is, with the reduced topology, the maximum link load should not exceed the given threshold $\alpha$ (in terms of the fraction of the link capacity) at any time during the off-peak time. The value of $\alpha$ is normally determined by individual ISPs according to their own traffic management policies. Such a value can be set to be consistent with the one associated with the peak-time operation with full topology. Constraint (4) makes sure that only the active links can carry the traffic during the off-peak time period. In addition, it is also required that the reduced network topology should not be broken due to the link removals, and also on a daily basis the starting/ending time for the off-peak topology configuration should be the same.

It should be noted that the simplified problem of identifying the optimal number of links for sleeping based on a single traffic matrix has been proven to be NP-complete [23], [29]. Therefore, there is no known computationally-efficient approach to optimally solve the problem and heuristics need to be used. Now we introduce our proposed TLS scheme below based on heuristics.

*2) Time-driven Link Sleeping Algorithm:* Fig. 1 illustrates the overall TLS approach. The curve in the figure indicates the actual maximum link utilization (MLU) dynamics pattern across a given period (5 days[1]). Such an MLU curve is effectively plotted according to the monitored traffic matrices at a certain time interval. In this example, we illustrate that with the TM measurements taking place every 2 hours, 12 TMs are produced within each 24 hour period. In Fig. 1, $M_{w,n}$ indicates the $n^{th}$ TM in day $w$. Such historical TM information is used for computing future off-peak network configurations and this is analogous to the forecasted traffic matrix used in general traffic engineering. First of all, multiple sampled TMs from different days in the considered period are used for computing a synthetic off-peak TM, based on which sleeping links are initially identified in the reduced network topology. The purpose of using multiple TMs instead of a single one is for robustness reasons, as traffic conditions even at the same time point in different days may vary. In addition, in order to compute the actual off-peak configuration period (i.e. the time window) on a daily basis, the same starting point for computation on each day needs to be identified, i.e. this point is located at the same $n$ for every day $w$. Starting from this point, an algorithm is designed for expanding (in both directions) the off-peak configuration window size incrementally in order to maximize the overall energy savings based on the reduced off-peak topology. As the final result, the reduced topology can be configured in a unified way on a daily basis, for instance between traffic matrix $M_{w,n1}$ and $M_{w,n2}$ within each $w^{th}$ day. For instance, in Fig. 1 the values of n1 and n2 for identifying the unified daily off-peak time are 4 and 7 respectively.

The overall heuristic-based approach consists of the following three sequential stages: (1) computing the synthetic traffic matrix and the starting point for off-peak window size expansion, (2) greedy link removal for the provisional reduced topology, and (3) the joint expansion of the off-peak window size and the finalization of the reduced topology to be applied during the off-peak period.

*a) Stage 1, Computing the synthetic TM and the starting point for off-peak window size expansion:* As

[1] To clearly illustrate the daily off-peak configuration windows, we purposely start from midday on the first day with peak traffic volume in the network. Also it is not necessary for the peak/valley MLU to occur exactly at the same time point on each day. This figure is only for illustration purposes.

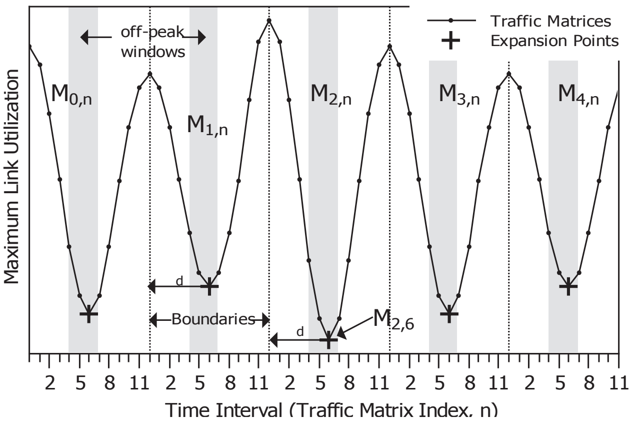

Fig. 1. Illustration of the overall approach.

previously indicated, the off-peak operation window size can be represented with a sequence of consecutive TMs at regular intervals, for instance every 2 hours as shown in Fig. 1. The MLU of each traffic matrix is calculated to identify the traffic matrix with the *lowest* MLU across the whole period (denoted as TM*) together with its location in the sequence of the TMs (e.g. TM $M_{2,6}$ in Fig. 1). The $2^{nd}$ index of this traffic matrix with the lowest MLU (i.e. 6 in this specific example) is set to be the *expansion point* of the daily off-peak window. To enable the same daily off-peak window, the location of the expansion point in every other day is at the same position as TM*. This is represented in Fig. 1 through the calculation of distance $d$ from the beginning of each day. Put in other words, since TM* in the figure is $M_{2,6}$, then the expansion point in other days should be $M_{w,6}$ (i.e. $d = 6$), where $w$ represents the index of the days under consideration. It is worth mentioning that we do not assume the actual lowest MLU always takes place exactly at the expansion point in the other days. For simplicity reasons this is not exhibited in Fig. 1. These expansions points are the starting points that will be used in the $3^{rd}$ stage to finally determine the actual location and the size of the off-peak windows.

As for the traffic matrix information to be used for computing the reduced topology with link removal, it is not efficient if only one single sampled TM is used as input, say TM* with the lowest MLU across the whole period. A more robust strategy is to take into account multiple TMs in order to obtain an optimized network configuration across all scenarios during the period. As such, we first compute a synthetic traffic matrix based on individual TMs located on the daily expansion points. In this example, the overall synthetic TM is computed according to the 5 TMs shown in Fig. 1 (with the + sign). For each traffic volume entry in the synthetic TM, the value is effectively the average of the corresponding entries from the 5 input TMs at the daily expansion points.

The computational complexity of calculating all the shortest paths of all the Source-Destination pairs in a graph and assigning all the traffic demands of one traffic matrix to the edges of the graph is given by $O(|V|^2.(|E| + |V|.log|V|))$ based on Dijkstra's shortest path algorithm. Since this procedure has to be done for each traffic matrix and there are $T$ traffic matrices in one day and in total $W$ days are considered as input, the overall complexity to determine the whole set of MLUs is $O(T.W.|V|^2.(|E| + |V|.log|V|))$ . This is actually the overall complexity for this stage since the complexity of finding the expansion points and calculating the synthetic traffic matrices does not dominate the overall complexity.

*b) Stage 2, Greedy link removal:* In this stage, a reduced network topology is provisionally computed based on the synthetic TM and the traffic matrices at the expansion points obtained from the previous stage. Further fine-tuning of this topology will be performed jointly with the determination of the actual off-peak operational window size in the next stage. The computation of this provisional reduced topology is achieved by the iterative greedy removal of the links from the full network topology. The detailed operation of the link removal in this stage is shown in Fig. 2.

Specifically, when considering the actual removal of any link $l^*$, the corresponding utilization of all the links is updated in the whole residual topology. If link $l^*$ under consideration cannot be removed, it is put into the *Link Removal Failure List* so that it will not be further considered. The condition for a successful link removal is that the residual network topology still remains connected and also the resulting MLU does not exceed the predefined constraint $\alpha$ based on the input synthetic TM. In addition, considering the requirement that the daily off-peak windows need to have the same starting time and duration, for robustness concerns the traffic matrices at all expansion points are also tested for compliance with the constraints (3) and (4) (shown in subsection III.B.1) each time one link is considered for removal from the network topology. At the end of this operation, the network topology *with the fewest t remaining active links*, which still satisfies constraints (3) and (4), is determined as the reduced topology. This topology will be further refined jointly with the optimization of the off-peak time window size in the next stage.

For each candidate link for removal, *Stage 2* has to find the alternative shortest paths of all the Source-Destination pairs of the graph and route all the traffic demands of the synthetic traffic matrix in order to check for violation of the MLU constraint. This step of *Stage 2* has complexity of $O(|E|.|V|^2.(|E| + |V|.log|V|))$ given that maximum $|E|$ links need to be considered for removal. The complexity of checking for full connectivity of the graph after each link removal is $O(|V|.(|E| + |V|.log|V|))$ based on node visiting using shortest path algorithm. Since the complexity for connectivity check is lower than the previous complexity, the overall complexity of *Stage 2* is $O(|E|.|V|^2.(|E| + |V|.log|V|))$.

*c) Stage 3, Determination of combined off-peak time window size and the final reduced topology:* The final stage of TLS is to compute the actual off-peak window size jointly with the fine-tuning of the number of sleeping links. The flowchart in Fig. 3 shows the detailed operation in this stage. To start with, the maximum number of *consecutive* TMs that can be supported by the reduced network topology computed in the previous stage is identified. The expansion of the daily off-peak windows starts from the expansion points obtained in *Stage 1*, which is identified by $M_{w,b}$. From this point the expansion is performed in both directions *sequentially* and *independently*. This is because the traffic behavior pattern

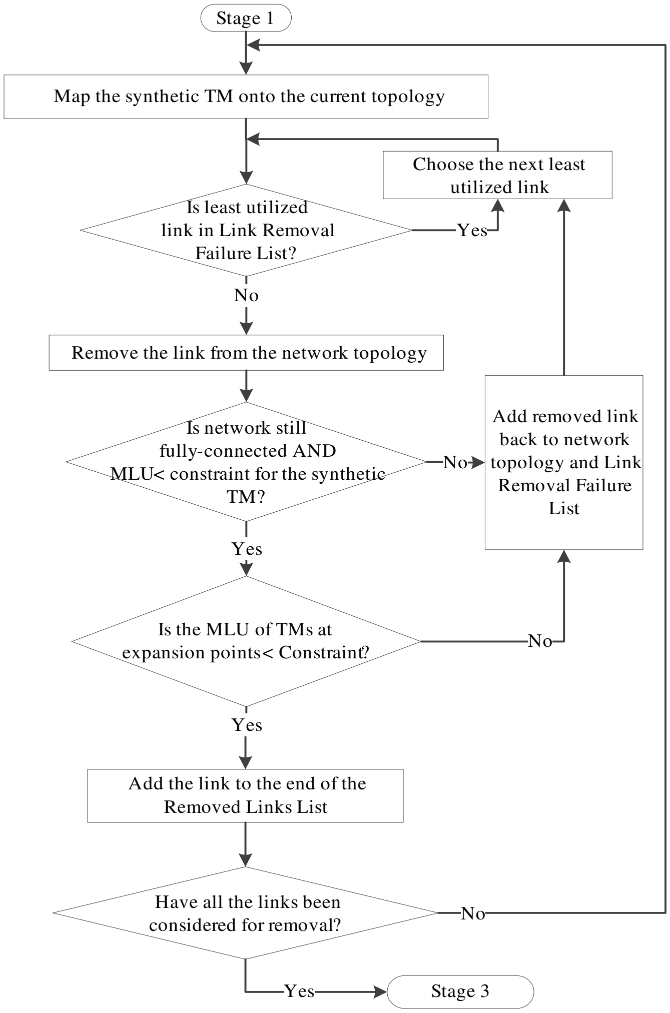
Stage 1
Map the synthetic TM onto the current topology
Choose the next least utilized link
Is least utilized link in Link Removal Failure List?
Yes
No
Remove the link from the network topology
Is network still fully-connected AND MLU< constraint for the synthetic TM?
No
Add removed link back to network topology and Link Removal Failure List
Yes
Is the MLU of TMs at expansion points< Constraint?
No
Yes
Add the link to the end of the Removed Links List
Have all the links been considered for removal?
No
Yes
Stage 3

Fig. 2. Flowchart for greedy link removal algorithm.

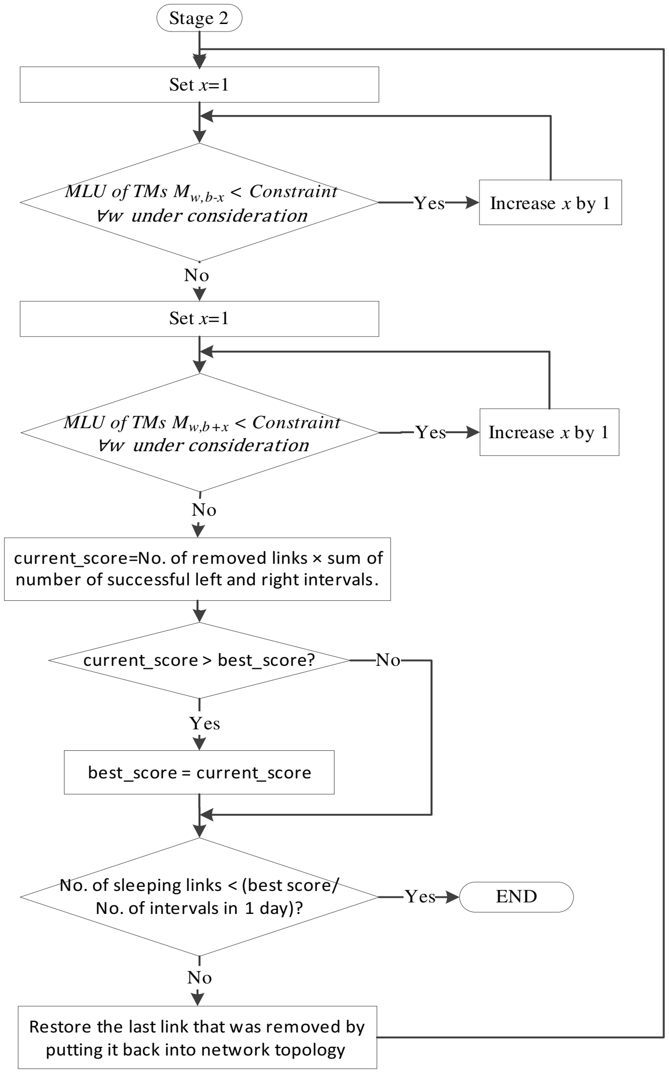
Stage 2
Set x=1
MLU of TMs M_{w,b-x} < Constraint ∀w under consideration
Yes
Increase x by 1
No
Set x=1
MLU of TMs M_{w,b+x} < Constraint ∀w under consideration
Yes
Increase x by 1
No
current_score=No. of removed links × sum of number of successful left and right intervals.
current_score > best_score?
No
Yes
best_score = current_score
No. of sleeping links < (best score/ No. of intervals in 1 day)?
Yes
END
No
Restore the last link that was removed by putting it back into network topology

Fig. 3. Off-peak window determination stage flowchart.

is generally not symmetric in the two directions from the expansion points. In each direction, if the projected MLU of the next TM does not exceed the predefined constraint for *all* considered days, then the time duration between that next TM and the current one can be included in the off-peak time window.

After the maximum number of consecutive TM intervals is obtained, TLS reconsiders the possibility of restoring some potentially sleeping links from the reduced topology determined in *Stage 2* and checks if this leads to an improved value of the overall objective function. The consideration of adding sleeping links is in the reverse order by which the links were removed from the network topology in the previous stage. This step is necessary because it is possible to have a sufficient increase in the number of consecutive TM intervals supported (i.e. enlarged off-peak time window size) to compensate for a reduced number of sleeping links and therefore, obtain better performance according to the overall objective function.

The termination condition for the iterative consideration of restoring sleeping links depends on the best value of the objective function so far. The objective function is formed by multiplying the number of sleeping links with the number of consecutive off-peak traffic matrix intervals. Using the best objective value obtained so far, the minimum number of sleeping links needed to obtain an improved objective value can be calculated. In *Stage 3*, TLS iteratively reduces the number of sleeping links to find out whether a longer off-peak period can be obtained so that the overall objective value can be improved. The restoration of sleeping links is stopped when the number of sleeping links is not sufficient to improve the best value of the objective cost function even if the entire time period is considered as off-peak.

For each restoration of the sleeping link, *Stage 3* has to find the new shortest path of all the Source-Destination pairs of the graph and route all the traffic demands of all the traffic matrices until violation of the MLU constraint. Therefore, the overall complexity of this stage is $O(E.T.W.|V|^2.(|E|+|V|.log|V|))$.

### *C. Performance Evaluation*

*1) Experiment Setup:* We evaluate TLS based on the operational GÉANT network topology and its published traffic traces. The topology consists of 23 Points-of-Presence (PoPs) and 74 unidirectional links of varying bandwidth

TABLE II
CHARACTERISTICS OF SET OF TRAFFIC MATRICES

| MLU | Value(%) |
|---|---|
| Max. | 90.9 |
| Min. | 30.9 |
| Mean | 58.6 |
| $1^{st}$ Quartile | 44.5 |
| $2^{nd}$ Quartile | 55.9 |
| $3^{rd}$ Quartile | 74.2 |

TABLE III
ENERGY SAVINGS OF TLS

| $\alpha$ | $X$ | $\lvert A_z \rvert$ | $O$ | $Y(\%)$ |
|---|---|---|---|---|
| 0.9 | 1 | 33 | 915 | 28.3 |
| | 1.1 | 33 | 600 | 18.6 |
| | 1.2 | 33 | 510 | 15.8 |
| | 1.3 | 33 | 420 | 13.0 |
| 0.8 | 1 | 33 | 600 | 18.6 |
| | 1.1 | 33 | 450 | 13.9 |
| | 1.2 | 33 | 375 | 11.6 |
| | 1.3 | 33 | 360 | 11.1 |
| 0.7 | 1 | 33 | 435 | 13.5 |
| | 1.1 | 33 | 360 | 11.1 |
| | 1.2 | 33 | 315 | 9.76 |
| | 1.3 | 32 | 405 | 12.2 |
| 0.6 | 1 | 33 | 330 | 10.2 |
| | 1.1 | 32 | 420 | 12.6 |
| | 1.2 | 32 | 405 | 12.2 |
| | 1.3 | 32 | 375 | 11.3 |

capacities. As far as the operational duration is concerned, we consider 480 consecutive traffic matrices at 15-minute intervals from Monday midday to Saturday midday according to the historical GÉANT traffic matrix data set [34]. The statistical characteristics of the traffic matrices during this period are given in Table II.

As previously mentioned, the network operator may have different policies in setting the constraint threshold for the maximum allowable link utilization based on the reduced topology during the off-peak time. This is effectively reflected by the value of $\alpha$ in Inequality (3) in the problem formulation. In our simulation experiment, we consider different values for $\alpha$ which represent the conservativeness in the link removal operation. A higher value of $\alpha$ indicates that a higher MLU is tolerable and in this case, more energy can be conserved due to either increased number of links to be removed and/or expanded off-peak time duration allowed. In our experiments we consider the following values for $\alpha$: 0.6, 0.7, 0.8 and 0.9, meaning that the maximum allowable MLU is 60%, 70%, 80% and 90%, respectively. It should be noted that, the most risky scenario ($\alpha = 0.9$) is at a similar level as the maximum allowable MLU during peak time (90.9%, see Table II). In addition, we also scale up the actual traffic demand in the TMs by 10% to 30% in order to evaluate the energy efficiency with higher traffic volume scenarios. Through this way, it is possible to test the performance of TLS under a variety of network performance policies (expressed through the MLU constraint) and different levels of operator conservativeness/trust with respect to the validity of historical traffic matrices. The scaling factor is represented by $X$.

*2) Simulation Results:* The proposed scheme was evaluated with different constraint thresholds ($\alpha$) and traffic volume scaling factors ($X$) as shown in Table III. In the table, $O$ is the total off-peak time duration (in terms of minutes) per 24-hour operation and $Y$ represents the energy savings (in terms of percentage) over the whole operation period under consideration.

It can be observed from Table III that, as the MLU constraint becomes more stringent, the off-peak window size and number of sleeping links decrease. The window size decreases more significantly compared to the decrease in the number of sleeping links with this specific traffic pattern. This phenomenon is more noticeable when the traffic matrices are scaled up and the MLU constraint is kept constant. The most important result from Table III is that TLS is able to achieve 28.3% energy savings over 5 consecutive working days while having a maximum MLU constraint of 90% (which is still not higher than the worst-case MLU at peak time that has been observed). In this case, the degree of bandwidth over-provisioning between peak time and off-peak time is effectively synchronized during daily network operations. In most cases, more conservative MLU threshold settings result in less energy savings being achieved due to the shrunk off-peak window size and/or reduced number of sleeping links. On the other hand, there are observed cases which do not follow this trend. This is because TLS does not always remove the links in the same order in *Stage 2*, as the MLU constraint is reached at different points in the link removal process because of either a change in the magnitude of the traffic demands due to a change in the traffic scaling factor or a different MLU constraint value. Since every link removal may either reduce or increase the opportunity for further link removals, the energy savings obtained are not always reduced when the scaling factor is increased due to this unpredictability in the future link removal order.

Figs. 4, 5, 6 and 7 show the *actual* MLU performance with time across the 5 considered days for different $\alpha$ constraint values. During the peak-time operation period (outside the dark area), the MLU performance is based on the full network topology. There are two MLU curve segments in the off-peak duration: the solid curves represent the MLU performance based on the actual reduced topology with the exclusion of sleeping links, while the dashed curves indicate the expected MLU performance based on the original full topology. Our observation on the MLU comparison is as follows: intuitively, the MLUs in the off-peak windows are expected to be higher when the *reduced network topology* is used rather than the *full network topology*. This is because the number of links available to carry the traffic demands is lower in the former scenario which uses the reduced topology. An interesting observation, however, is that there are also cases where the *full network topology* gives a higher MLU compared to the *reduced network topology*. According to our analysis, this is

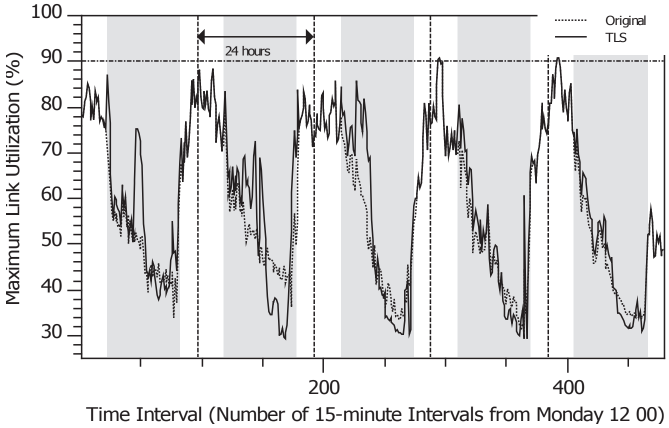

Fig. 4. Variation of MLU with reduced topology of $\alpha = 0.9$.

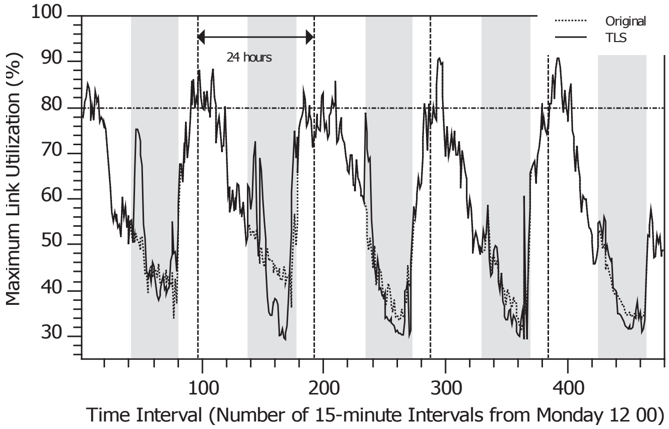

Fig. 5. Variation of MLU with reduced topology of $\alpha = 0.8$.

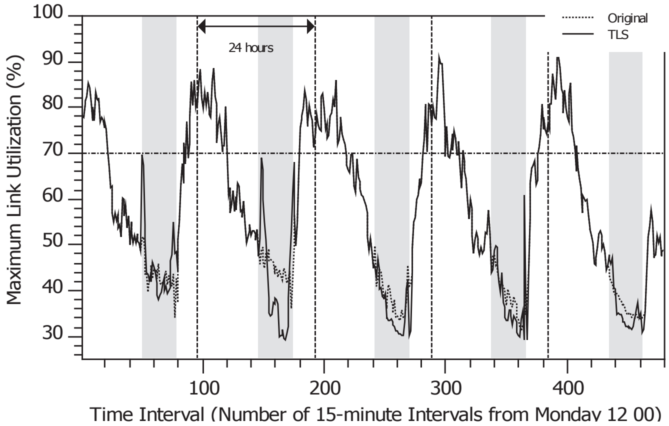

Fig. 6. Variation of MLU with reduced topology of $\alpha = 0.7$.

because the greedy link removal operation of TLS may remove a link which originally injects traffic into the most loaded link of the network based on the full topology. The removal of such a link can divert some customer traffic away from that most loaded link to other alternate paths, in which case the overall MLU is decreased.

This scenario is illustrated with a simple example in Fig. 8 and Table IV. The figure shows a small topology in which the links have different capacities, and the IGP link weights are in *inverse proportion* to the link capacities. Table IV shows

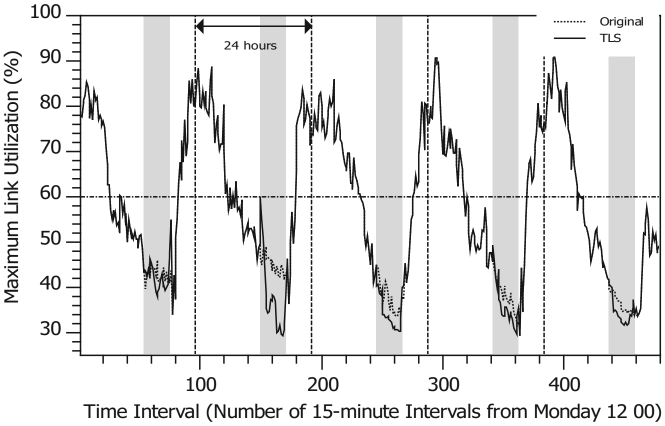

Fig. 7. Variation of MLU with reduced topology of $\alpha = 0.6$.

the traffic matrix information to be mapped onto the topology. Initially, there is no link removal and the traffic demand from node A to C is routed through node B and it consumes 10 units of bandwidth on the link B-C, which is the most loaded link in the network (note that the link is also carrying 60 units of traffic from B to C) with utilization being 70%. When the least-loaded link A-B is removed from the topology, then the traffic demand from A to C will be diverted on to the path A-D-C. The link utilization increase on links A-D and D-C due to the diverted traffic (to 30%) is not enough to make any of them become the most loaded link in the network. Although link B-C remains the most loaded link, its utilization is reduced to 60% and hence the maximum MLU in the network is reduced after the link removal.

It should be noted that the original IGP link weight settings in the GÉANT network are not optimized for traffic engineering but for delay [11], and this explains why TLS is able to further reduce the MLUs in the off-peak windows even with a reduced topology. Overall, TLS is very efficient as it is able to reconfigure 33 out of 74 links to sleep mode during the off-peak time in most experiments shown in Table III, which represents more than 40% of the links.

Interestingly, it can be also seen in Fig. 4, 5, 6 and 7 that most of the off-peak MLU values obtained using the original full network topology are similar to the ones obtained using the TLS reduced topology, meaning that the worst-case link load is not increased substantially due to the reduced topology. This is due to the fact that the small amount of load which was previously carried by the sleeping links has been rerouted mostly to the remaining active links with relatively low or medium utilization. This is shown in Fig. 9 where the utilization percentage of the sorted links is plotted for the original and TLS topology using an off-peak traffic matrix. It can be observed that on the left part of the graph, TLS has lower utilization values due to the sleeping links which have zero utilization, while the original topology has a set of links with low utilization. On the right part of the graph though, it can be seen that TLS has larger values than the original due to the loads of the sleeping links being transferred to the other active links.

In addition to the above performance evaluation based on the same 5-day traffic matrices as original input, we also

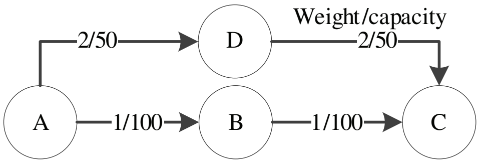

Fig. 8. Illustrative network topology.

TABLE IV
TRAFFIC DEMANDS AND PATHS FOR THE ILLUSTRATIVE NETWORK TOPOLOGY IN FIG. 8

| Source→<br>Destination | Demand | Initial Path | Path After removal<br>of link A-B |
|---|---|---|---|
| A→C | 10 | A-B-C | A-D-C |
| B→C | 60 | B-C | B-C |
| A→D | 20 | A-D | A-D |
| D→C | 20 | D-C | D-C |
|  | MLU | 70% (B-C) | 60% (B-C) |

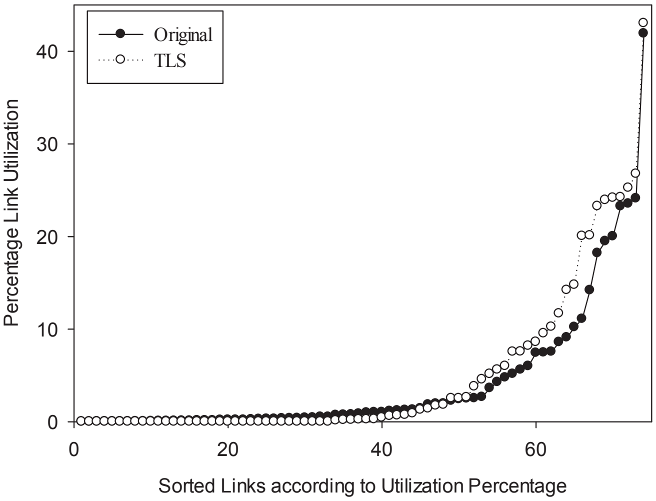

Fig. 9. Percentage utilization of links in the network.

TABLE V
CONFIGURATION PERFORMANCE ON OTHER WEEKS

| $\alpha$ | Week 1 |  | Week 2 |  | Week 3 |  | Week 4 |  | Week 5 |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  | $H$ | $U(\%)$ | $H$ | $U(\%)$ | $H$ | $U(\%)$ | $H$ | $U(\%)$ | $H$ | $U(\%)$ |
| 0.9 | 0 | 82.6 | 0 | 74.1 | 0 | 73.1 | 0 | 65.8 | 0 | 69.4 |
| 0.8 | 0 | 68.5 | 0 | 65.5 | 0 | 57.5 | 0 | 44.5 | 0 | 57.5 |
| 0.7 | 0 | 59.7 | 0 | 60.9 | 0 | 50.0 | 0 | 42.4 | 0 | 48.6 |
| 0.6 | 0 | 48.0 | 0 | 57.5 | 0 | 50.1 | 0 | 40.8 | 0 | 43.2 |

applied the same resulting configuration on 5 other different weeks in order to have a more comprehensive evaluation of the proposed TLS scheme. Table V shows the performance of the configuration when applied to these 5 other weeks where $H$ is the number of off-peak traffic matrices which produce an MLU above the set constraint $\alpha$ and $U$ is the actual MLU observed during the off-peak period. It can be observed that the value of $H$ is always 0 and that $U$ is always below the constraint. These results indicate promising robustness capability of the TLS scheme against varied traffic patterns in different testing periods.

### D. Summary

In this section, we presented the TLS algorithm and evaluated its performance. TLS determines a *reduced network topology* with scheduled sleeping link reconfiguration and the actual off-peak time period during which this network topology is used. The scheme aims to identify an optimized trade-off between the number of sleeping links and the configuration duration for the reduced topology in order to achieve maximum energy savings while maintaining overall network performance. We demonstrated through simulations based on the GÉANT network and its actual traffic traces, that the proposed scheme is able to achieve significant energy savings in the daily operations. Some interesting observations have also been obtained and analyzed, such as the ability of TLS to reduce the MLU with fewer active links, which provides insights for the design of future traffic-engineering aware network optimization schemes for energy efficiency.

## IV. TLS WITH SINGLE LINK FAILURE PROTECTION

### A. Scheme Overview

During the off-peak time with the exclusion of sleeping links, the reduced network topology is more vulnerable to traffic congestion in event of any further unexpected link failure because of a decrease in the connectivity options and the capacity of the network. The proposed TLS algorithm does not consider the single link failure scenario which is known to occur frequently in operational networks. In order to take into account traffic protection requirements against unexpected failures during off-peak time, we extend the plain TLS scheme to be robustness aware, and we call the new scheme *Time-driven Link Sleeping with Single Link Failure Protection* (TLS-SLFP). TLS-SLFP aims to optimize the reduced network topology so that it always remains fully connected, and is able to support the worst-case off-peak traffic demands after the failure of any single link during off-peak time. First of all, from the basic connectivity point of view, the further reduced (post-failure) network topology should still remain fully connected. Therefore, an additional constraint must be added to cater for this new requirement. In addition, a still fully-connected post-failure topology does not necessarily ensure that all the traffic demands will be sufficiently accommodated by the further reduced topology. Hence we consider the second requirement from the viewpoint of bandwidth support, in which case the original link removal operation needs to make sure that sufficient bandwidth resources are provided by the reduced topology in the presence of any single link failure.

### B. Algorithm Extension from TLS

*1) Proposed Algorithm:* Since the objective of TLS-SLFP is an extended problem formulation of the plain TLS, the corresponding algorithm is largely built on top of TLS. TLS-SLFP has four stages which are described below.

*a) Stage 1, Computing the synthetic TM and the starting point for off-peak window size expansion:* This stage is the same as *Stage 1* for TLS.

*b) Stage 2, Greedy link removal with full network connectivity check for single link failure scenarios: Stage 2* of TLS has been extended for taking into account the new constraints on post-failure connectivity of the reduced topology when determining sleeping links. Specifically, when a link $l^*$ is considered for scheduled sleeping, the following condition needs to be satisfied: The removal of $l^*$ and of any other single link (considered as unexpected single link failure scenarios) should still ensure the residual network topology remains fully connected. As such, the connectivity check is based on $L - 2$ single link failure scenarios jointly with the removal of $l^*$, where $L$ is the total number of links within the physical network topology. For each link removal in the modified *Stage 2*, an additional connectivity check needs to be done for each of the remaining active links when they are projected to fail. The complexity of the actual connectivity check for any single link failure is $O(|E|.|V|.(|V|.log|V| + |E|))$. This additional step in the modified Stage 2 gives a new complexity of $O(|E|.(|V|^2.(|E|+|V|.log|V|)+|E|.|V|.(|V|.log|V|+|E|)))$.

*c) Stage 3, Determination of combined off-peak time window size and the final reduced topology:* This stage is the same as *Stage 3* for TLS.

*d) Newly added Stage 4, Optimizing sleeping link selection to avoid traffic congestion during single link failures:* A new *Stage 4* has been included in TLS-SLFP which makes sure that no traffic congestion (projected utilization of any active link exceeding 100%) occurs given any single link failure scenario in the robust reduced topology for the off-peak traffic demands. The first step of this stage calculates the number of single link failure scenarios that *cannot* be supported by the reduced network topology produced at the end of *Stage 3*. In order to test for single link failures in this reduced network topology, each active link is removed in turn from the topology and we check if each of the residual topologies is able to accommodate the traffic demand.

After identifying the total number of unsupported single link failure scenarios, TLS-SLFP identifies what impact each sleeping link in the reduced topology will have on this number if the sleeping link were to be restored back to the topology again. This step produces a list of ranked sleeping links for potential restoration according to their ability to reduce the number of unsupported single link failure scenarios. The next step of this stage involves the restoration of sleeping link candidates back to the working topology according to the list produced in the previous step. After each sleeping link candidate is considered for restoration, the number of unsupported single link failure scenarios is re-calculated, and if this number has not decreased, the link can remain sleeping and the next link candidate in the list is tested. This procedure is applied to ensure that the minimum number of sleeping link candidates needs to be restored back to the working topology, and therefore, conserve the energy-savings achieved in *Stage 3* as much as possible.

In this stage, the first step involves the calculation of the number of single link failures which cause the MLU to go above 100% in order to see if optimization of the number of sleeping links is needed. This step has complexity of $O(|E|.T.W.|V|^2.(|E| + |V|.log|V|))$. The next step involves the calculation of the number of unsupported single link failures for each sleeping link restoration. The complexity of this step is $O(|E|^2.T.W.|V|^2.(|E| + |V|.log|V|))$ which dominates the overall complexity of this stage. The next major step of *Stage 4* is to calculate the remaining unsupported single link failures after each sleeping link restoration. The complexity of this step is the same as the previous step. Therefore, the overall complexity of this stage can be simplified to $O(|E|^2.T.W.|V|^2.(|E| + |V|.log|V|))$.

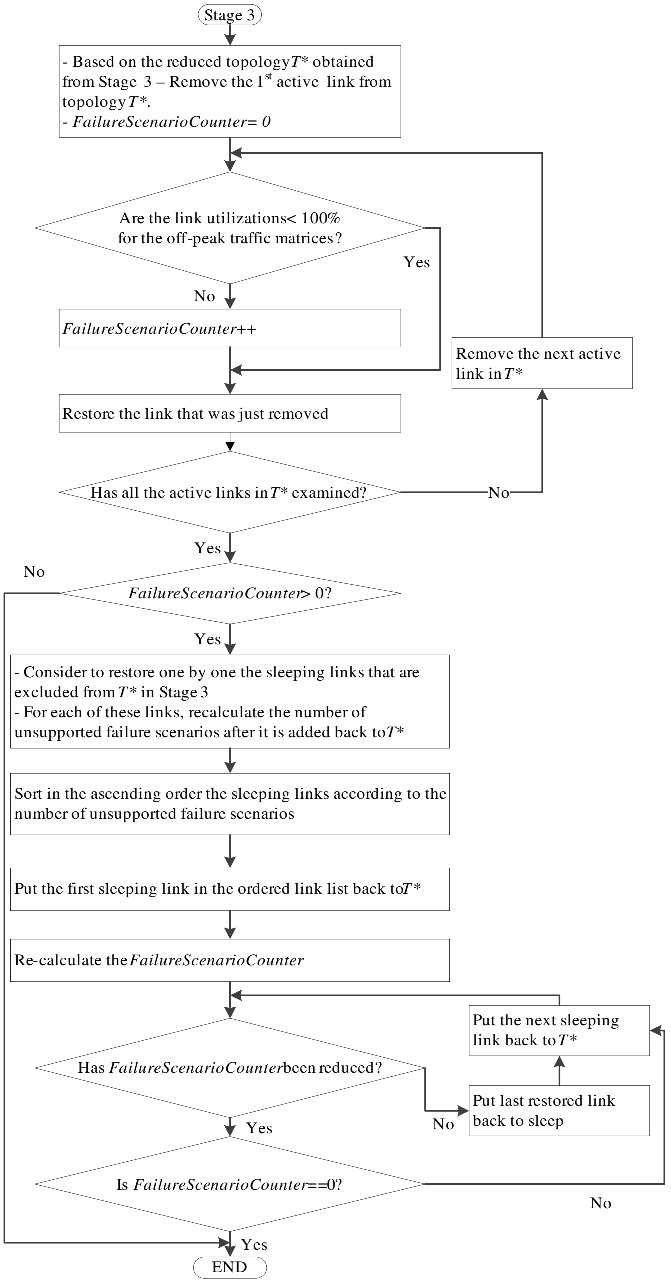
Stage 3
- Based on the reduced topology T* obtained from Stage 3 – Remove the 1st active link from topology T*.
- FailureScenarioCounter= 0
Are the link utilizations< 100% for the off-peak traffic matrices?
Yes
No
FailureScenarioCounter++
Remove the next active link in T*
Restore the link that was just removed
Has all the active links in T* examined?
No
Yes
No
FailureScenarioCounter> 0?
Yes
- Consider to restore one by one the sleeping links that are excluded from T* in Stage 3
- For each of these links, recalculate the number of unsupported failure scenarios after it is added back to T*
Sort in the ascending order the sleeping links according to the number of unsupported failure scenarios
Put the first sleeping link in the ordered link list back to T*
Re-calculate the FailureScenarioCounter
Put the next sleeping link back to T*
Has FailureScenarioCounter been reduced?
Put last restored link back to sleep
No
Yes
Is FailureScenarioCounter==0?
No
Yes
END

Fig. 10. Flowchart showing optimization of the set of sleeping links in TLS-SLFP.

*2) Performance Evaluation:* The TLS-SLFP scheme was evaluated based on the same GÉANT network topology and traffic matrices that were used for TLS, since this allows for the change in performance to be directly compared. Figs. 11, 12, 13 and 14 indicate the energy saving comparison

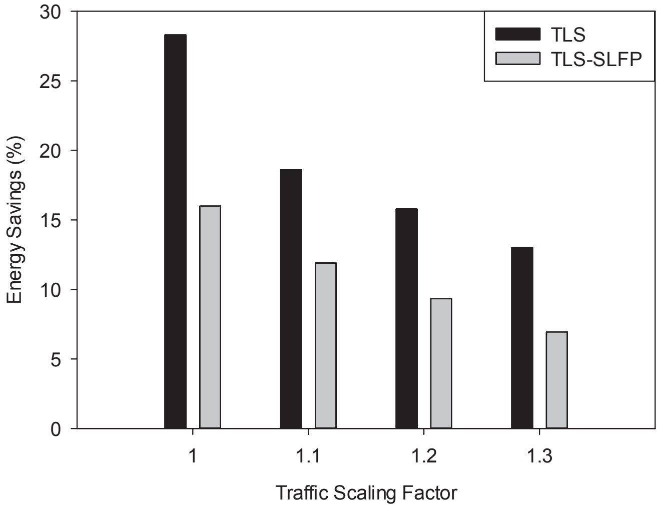

Fig. 11. Energy-saving gains with reduced topology of $\alpha = 0.9$ under different traffic scaling factors.

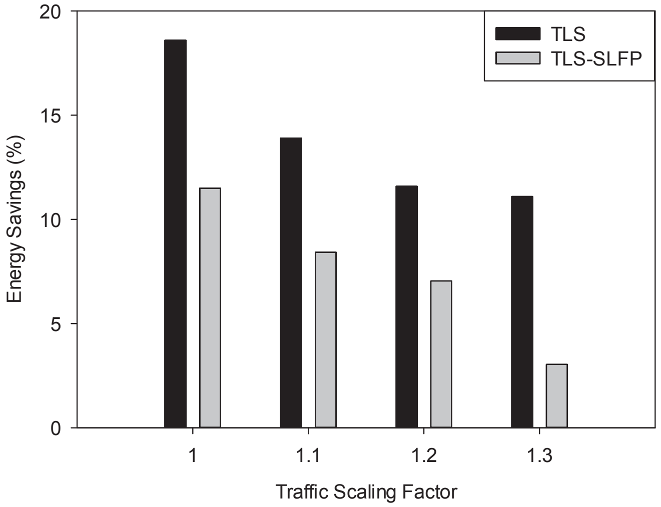

Fig. 12. Energy-saving gains with reduced topology of $\alpha = 0.8$ under different traffic scaling factors.

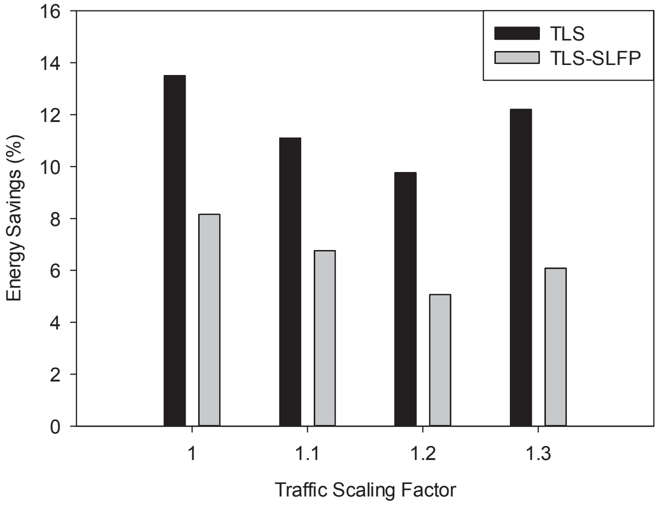

Fig. 13. Energy-saving gains with reduced topology of $\alpha = 0.7$ under different traffic scaling factors.

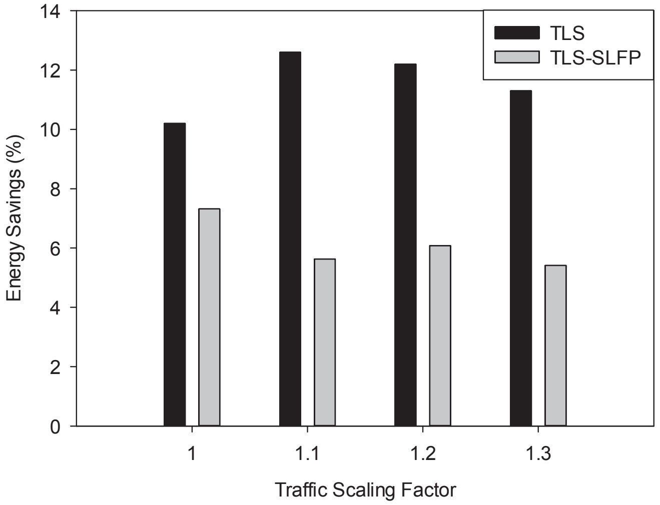

Fig. 14. Energy-saving gains with reduced topology of $\alpha = 0.6$ under different traffic scaling factors.

between TLS and TLS-SLFP based on different values of MLU constraint $\alpha$. All the figures indicate that the overall energy saving gains are reduced as new constraints are added to TLS in order to take into account single link failure protection in TLS-SLFP.

Fig. 11 shows the performance comparison between TLS and TLS-SLFP when the MLU constraint is set to 90%. It can be seen that in all cases involving different traffic scaling factors, there is a decrease in energy savings when the additional constraint is applied for single link failure protection. For instance, the energy saving gain is reduced from 28.3% in TLS to 16% in TLS-SLFP in the scenario without traffic volume scaling. Figs. 12, 13 and 14, where the MLU constraint is set to 80%, 70% and 60% respectively, show a similar pattern as observed in Fig. 11.

Table VI shows a detailed breakdown of the performance of TLS-SLFP. One interesting observation from this table is that TLS-SLFP tends to push more links to the sleep mode when the MLU constraint is lowered while decreasing the off-peak window duration. Overall, the energy savings decrease as the MLU constraint is lowered because the MLU threshold is reached more quickly during the off-peak window expansion process. It is also worth mentioning that, the energy saving gains do not always decrease when the MLU constraint is kept constant and the scaling factor is increased. This is due to the same reason as in the TLS scenario where the link removal order changes as MLU constraint and scaling factor change.

### C. Summary

In this section, a new algorithm called TLS-SLFP was introduced as an extension to TLS to take into account single link failure protection which is known to occur frequently in operational networks. TLS-SLFP has been shown to be able to achieve reasonable energy savings even though there have been additional constraints on the algorithm such as more stringent network connectivity and capacity constraints in order to support single link failure events.

TABLE VI
ENERGY SAVINGS OF TLS-SLFP

| $\alpha$ | $X$ | $\lvert A_z \rvert$ | $O$ | $Y(\%)$ |
|---|---|---|---|---|
| 0.9 | 1 | 12 | 1425 | 16.0 |
| | 1.1 | 18 | 705 | 11.9 |
| | 1.2 | 17 | 585 | 9.33 |
| | 1.3 | 17 | 435 | 6.94 |
| 0.8 | 1 | 13 | 945 | 11.5 |
| | 1.1 | 13 | 690 | 8.42 |
| | 1.2 | 20 | 375 | 7.04 |
| | 1.3 | 9 | 360 | 3.04 |
| 0.7 | 1 | 20 | 435 | 8.16 |
| | 1.1 | 20 | 360 | 6.76 |
| | 1.2 | 20 | 270 | 5.07 |
| | 1.3 | 16 | 405 | 6.08 |
| 0.6 | 1 | 20 | 390 | 7.32 |
| | 1.1 | 20 | 300 | 5.63 |
| | 1.2 | 16 | 405 | 6.08 |
| | 1.3 | 16 | 360 | 5.41 |

## V. CONCLUSIONS

Since it has been observed that the traffic patterns of many operational ISP networks are generally regular and predictable on a daily basis, such a feature can be exploited by applying simple time-driven energy-aware network reconfigurations with low operational complexity. In this work, an algorithm called Time-driven Link Sleeping (TLS) was proposed which jointly optimizes the reduced network topology and the period of off-peak operation during which this reduced topology can be used. The novelty of TLS is that it performs optimization with respect to a period of traffic behavioral dynamics (represented by multiple traffic matrices) compared to only one instance of traffic matrix, as is the case in the vast majority of existing works. The proposed scheme enables a simple and practical energy management paradigm that eases the complexity of ISP operations for energy conservation. TLS has been evaluated based on the GÉANT network and its real historical traffic traces, and the results show that the proposed scheme is able to achieve up to 28.3% energy saving effects during daily operations.

We have also taken into account the network robustness requirement against single link failures when determining sleeping links for off-peak time operation. The rationale behind is to avoid potential broken topology or traffic congestion due to unexpected single link failures that might occur to the reduced topology during the off-peak time. Towards this end, TLS has been further extended to cater for any single link failure scenario, which is known as Time-driven Link Sleeping with Single Link Failure Protection (TLS-SLFP). Performance evaluation on TLS-SLPF has shown that the proposed scheme is able to obtain good trade-off between energy efficiency performances and network robustness against common single link failures.

## REFERENCES

[1] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, “A clean slate 4D approach to network control and management,” *Computer Commun. Rev.*, vol. 35, pp. 41–54, 2005.

[2] M. Baldi and Y. Ofek, “Time for a ‘Greener’ Internet,” in *Proc. 2009 IEEE International Conference on Communications Workshops*, pp. 1–6.

[3] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, “Power awareness in network design and routing,” in *Proc. 2008 IEEE INFOCOM*, pp. 457–465.

[4] B. Sanso and H. Mellah, “On reliability, performance and Internet power consumption,” in *Proc. 2009 International Workshop on Design of Reliable Communication Networks*, pp. 259–264.

[5] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, “Reducing network energy consumption via sleeping and rate-adaptation,” in *Proc. 2008 USENIX Symposium on Networked Systems Design and Implementation*, pp. 323–336.

[6] G. Da Costa, J. P. Gelas, Y. Georgiou, L. Lefevre, A. C. Orgerie, J. M. Pierson, O. Richard, and K. Sharma, “The GREEN-NET framework: energy efficiency in large scale distributed systems,” in *Proc. 2009 IEEE International Symposium on Parallel Distributed Processing*, pp. 1–8.

[7] F. Francois, N. Wang, K. Moessner, and S. Georgoulas, “Optimization for time-driven link sleeping reconfigurations in ISP backbone networks,” in *Proc. IEEE Network Operations and Management Symposium*, pp. 221–228.

[8] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, “Structural analysis of network traffic flows,” in *Proc. 2004 Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 61–72.

[9] A. S. Antonio Nucci and N. Taft, “The problem of synthetically generating IP traffic matrices: initial recommendations,” Narus, Sprint and Intel, Tech. Rep., 2005.

[10] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, “Experience in measuring backbone traffic variability: models, metrics, measurements and meaning,” in *Proc. 2002 ACM SIGCOMM Workshop on Internet measurment*, pp. 91–92.

[11] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, “Providing public intradomain traffic matrices to the research community,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 83–86, Jan. 2006.

[12] R. Bolla, R. Bruschi, A. Cianfrani, and M. Listanti, “Enabling backbone networks to sleep,” *IEEE Network*, vol. 25, no. 2, pp. 26–31, Mar.-Apr. 2011.

[13] P. Psenak, S. Mirtorabi, A. Roy, L. Nguyen, and P. Pillay-Esnault, “Multi-Topology (MT) Routing in OSPF,” RFC 4915 (Proposed Standard), Internet Engineering Task Force, June 2007.

[14] T. Przygienda, N. Shen, and N. Sheth, “M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs),” RFC 5120 (Proposed Standard), Internet Engineering Task Force, Feb. 2008.

[15] K. W. Kwong, R. Gue andrin, A. Shaikh, and S. Tao, “Balancing performance, robustness and flexibility in routing systems,” *IEEE Trans. Network and Service Management*, vol. 7, no. 3, pp. 186–199, Sep. 2010.

[16] L. G. Roberts, “A radical new router,” *IEEE Spectrum*, vol. 46, no. 7, pp. 34–39, July 2009.

[17] M. Yamada, T. Yazaki, N. Matsuyama, and T. Hayashi, “Power efficient approach and performance control for routers,” in *Proc. 2009 IEEE International Conference on Communications Workshops*, pp. 1–5.

[18] R. Bolla, R. Bruschi, and A. Ranieri, “Energy-aware equipment for next-generation networks,” in *Proc. 2009 International Conference on Future Internet Technologies*, pp. 8–11.

[19] M. Mandviwalla and N.-F. Tzeng, “Energy-efficient scheme for multiprocessor-based router linecards,” in *Proc. 2006 International Symposium on Applications on Internet*, pp. 156–163.

[20] J. C. C. Restrepo, C. G. Gruber, and C. M. Machuca, “Energy profile aware routing,” in *Proc. 2009 IEEE International Conference on Communications Workshops*, pp. 1–5.

[21] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, and A. Vasilakos, “An OSPF-integrated routing strategy for QoS-aware energy saving in IP backbone networks,” *IEEE Trans. Network and Service Management*, no. 99, pp. 1–14, 2012.

[22] L. Irish and K. J. Christensen, “A ‘Green TCP/IP’ to reduce electricity consumed by computers,” in *Proc. 1998 IEEE Southeastcon*, pp. 302–305.

[23] N. Vasić and D. Kostić, “Energy-aware traffic engineering,” in *Proc. 2010 International Conference on Energy-Efficient Computing and Networking*, pp. 169–178.

[24] C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, and M. Mellia, “Energy saving and network performance: a trade-off approach,” in *Proc. 2010 International Conference on Energy-Efficient Computing and Networking*, pp. 41–50.

[25] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, “Reducing the energy consumption of Ethernet with adaptive link rate (ALR),” *IEEE Trans. Computers*, vol. 57, pp. 448–461, 2008.

[26] W. Fisher, M. Suchara, and J. Rexford, “Greening backbone networks: reducing energy consumption by shutting off cables in bundled links,” in *Proc. 2010 ACM SIGCOMM Workshop on Green Networking*, pp. 29–34.

[27] M. Zhang, C. Yi, B. Liu, and B. Zhang, “GreenTE: power-aware traffic engineering,” in *2010 IEEE International Conference on Network Protocols*.

[28] G. N. Rosario, G. Garroppo, S. Giordano, and M. Pagano, “Energy aware routing based on energy characterization of devices: solutions and analysis,” in *2011 International Workshop on Green Communications*.

[29] M. Gupta and S. Singh, “Greening of the Internet,” in *Proc. 2003 SIGCOMM*, pp. 19–26.

[30] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, “A survey of green networking research,” *IEEE Commun. Surveys Tutorials*, no. 99, pp. 1–18, 2010.

[31] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, “Energy efficiency in the future Internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures,” *IEEE Commun. Surveys Tutorials*, no. 99, pp. 1–22, 2010.

[32] L. Chiaraviglio, M. Mellia, and F. Neri, “Reducing power consumption in backbone networks,” in *Proc. 2009 IEEE International Conference on Communications*, pp. 2298–2303.

[33] N. Wang, K. Ho, G. Pavlou, and M. Howarth, “An overview of routing optimization for Internet traffic engineering,” *IEEE Commun. Surveys Tutorials*, vol. 10, no. 1, pp. 36–56, 2008.

[34] “The TOTEM Traffic Engineering Toolbox.” Available: http://totem.run.montefiore.ulg.ac.be

[35] “Yin Zhang’s Abilene Traffic Matrices.” Available: {http://www.cs.utexas.edu/users/yzhang/research/AbileneTM/}