Green Link Weights for Disruption Free Energy Aware Traffic Engineering

Obinna Okonor, Ning Wang, Stylianos Georgoulas and Zhili Sun
Institute for Communication Systems
University of Surrey, Guildford, Surrey
GU2 7XH, United Kingdom

Abstract - Energy Aware Traffic Engineering has been gaining steadily increasing interest due to the cost reduction benefits it can offer to network operators and for environmental reasons. While numerous approaches exist that attempt to provide energy reduction benefits by intelligently manipulating the network devices and their configuration, most of them suffer from one fundamental shortcoming; however small adaptations to a given network configuration they suggest, they all lead to temporal service disruptions that make these schemes less appealing to network operators. The more frequent the reconfigurations, to closely optimize the network attributes in response to changing traffic demands, the more frequent the service disruptions. Driven by this need for seamless service provisioning we put forward a framework for disruption free energy aware traffic engineering, which leverages on selective link sleeping and wake-up in a disruption free manner. The framework allows for maximizing the opportunities for disruption free reconfigurations based on intelligent link weight settings, assisted by a dynamic scheme that optimizes the reconfigurations in response to changing traffic conditions. As our simulation based evaluation shows the framework is capable of achieving significant energy saving gains, while at the same time providing robustness in terms of disruptions avoidance and resilience to congestion.

***Keywords* – link weight setting, routing convergernce, energy efficiency, traffic engineering, disruption-free operations, congestion avoidance.**

I. INTRODUCTION

The ever increasing power consumption of Internet-based networks has become a major concern for network operators due to the also ever increasing electricity cost as well as for environmental reasons and more stringent regulations associated with carbon emissions that are being put forward by local and international authorities. Research has shown that the Internet alone contributed up to 5.5% of the world's power consumption in 2007 with a prediction of 20 - 25% annual growth [1] while ICT alone accounts for 2% of the greenhouse effect annually [2], [3]. With the introduction of bandwidth hungry and Quality of Service (QoS) demanding applications which call for more efficient manipulation of the available network resources and even require network capacity upgrades, the operators are faced with the difficult task of having to accommodate these increasing traffic demands, without increasing the associated electricity cost or -if possible- even reducing it. Furthermore, since the energy consumption of networking devices does not scale with the device load (i.e. energy consumption is in many cases almost steady once a device is switched on irrespective of its actual load [4], [5]), while over provisioning -even if considered economically viable as a capital investment- may help in circumventing the QoS degradation issue due to lack of resources, this approach leads to high and unnecessary energy consumption.

The energy aware traffic engineering (ETE) techniques that have been introduced during the past few years in the direction of providing energy reduction benefits in internet-based networks rely (with certain variations) on the key concept that since networks are usually over provisioned to satisfy "peak-time" demands, resources such as network links or even entire network nodes can be switched off when the traffic volume allows for it. As such, the various ETE techniques try to make the network topology, configuration and capacity follow the trends that the traffic exhibits. The changes that are suggested by the various ETE schemes can be based on offline calculated estimations, on a periodic basis or triggered based on events (e.g. violation of allowable utilization thresholds), etc. For core networks though that use Open Shortest Path First (OSPF) as the underlying routing protocol, any change in the topology, however big or small, means that routing tables need to be updated (re-convergence process). This can lead to oscillatory behaviors and transient service disruptions (packet loss, packet re-ordering, excessive delay and jitter) due to inconsistency in the routing tables across the network during that period. This may not be tolerable for applications like real-time streaming but also for TCP applications due to the out-of-order packet delivery which can trigger re-transmissions and throttle down the speed of TCP connections; all these can make the adoption of such solutions less appealing in realistic scenarios. And the more dynamic a scheme, the more frequent the reconfigurations it puts forward, the more frequent the disruptions.

Driven by the need to overcome this major shortcoming of service disruptions in OSPF-based IP networks, linked with OSPF re-convergence in response to changes in the network topology dictated by energy saving operations, in this paper we push forward a framework for disruption free energy aware traffic engineering. The framework relies on selective link sleeping and wake-up when needed and performs ***two*** key operations; ***first*** it tries offline to maximize the opportunities for disruption free operations by intelligently setting the OSPF link weights in a way that the number of links that can be manipulated for traffic engineering operations in a disruption free way can be maximized (details on this concept are provided in forthcoming sections). While intelligent link weight setting for optimizing certain network objectives has been proposed in the past (and has been proven to be an NP-hard problem [6]-[9] even when considering static network conditions), to the best of our knowledge this is the first time that service disruption is being considered together with other network objectives during the link weight setting process. In addition, in real networks that exhibit dynamic traffic conditions, a single link weight setting aiming to account for all the traffic conditions that may be encountered may lead to sub-optimal performance, while frequent link weight setting (tuned to cope for shorter-term traffic conditions) can again lead to the re-convergence issues mentioned above and the associated service disruption. This is not an issue with our scheme; since the link weight setting operation is part of a broader framework, it is possible to perform it only **once** in an offline fashion and still circumvent the performance deficiencies that such a static link weight setting might have otherwise induced.

Then given a candidate set of available disruption free operations, an online operation is performed which selects the optimal, in terms of energy efficiency and network performance objectives, link wake-up operations needed each time in response to changing traffic conditions. This online operation plays a key role in allowing for one single link weight setting to be kept throughout the runtime operation of a network, without compromising neither energy efficiency, nor the network performance. Also, contrary to other ETE schemes that "blindly" revert to using the full topology when network congestion is detected, the online operation allows making small incremental disruption free updates to the topology when these are needed, sacrificing energy savings each time only to the lowest degree possible.

It is worth noting that while network links themselves do not consume very high energy (fiber links along with their amplifiers account only for 7% of the power budget of a network with the remaining 93% being consumed by routers themselves [10]), the line cards inside an IP router typically consume around 43% of the total power of the router [11]. As such, putting a link to sleep allows putting its associated line card to sleep, which when considered together represent a very high percentage of the overall energy budget consumed by a network. This work builds on our previous work in [12]. Contrary to the approach presented there though, here the target is not to reduce service disruptions due to OSPF re-convergence but to completely nullify the OSPF re-convergence issue and its consequences. As our simulation based evaluation, using the Abilene and GEANT network topologies and their publicly accessible traffic traces [13]-[15], will show, following the methodology in this work, robustness to service disruptions and network congestion can be achieved still, achieving significant energy saving gains in this process.

The rest of the paper is organized as follows. In Section II we present the key idea based upon which our work builds to allow for disruption free reconfigurations and the problem formulation. Then in Section III we present in details our framework and its two complementary (offline and online) operations. The performance of our framework using extensive simulations with real traffic traces in real network topologies is presented in Section IV. Section V presents related work in the area of energy aware traffic engineering and finally, Section VI, concludes the paper summarizing the main findings.

II. FRAMEWORK BUILDING NOTIONS

The key building notion for our work is that in an OSPF-based network, all links are not the same when it comes to the effect they can have on OSPF re-convergence. Some links unavoidably can lead to the need for routing table updates network-wide, which can lead to transient service disruptions and deterioration, when removed or restored to the topology. Whereas, some other links do not lead to the need for routing tables updates network-wide when they are removed/restored. But instead they lead to local routing table update only of the directly adjacent upstream node (head node). This operation may cause some very minimal service disruption (e.g. in case the new path to be followed by the affected packets has a significantly shorter delay than the old path, which may lead to some packet reordering) but this disruption is very much contained spatially but also temporally.

In the rest of this Section we present in more details why this is the case in OSPF-based networks and how this can be capitalized upon from our framework to support disruption free energy aware traffic engineering decisions.

A. Link Role in OSPF-based routing

To visualize how in an OSPF-based network, different links can affect in different ways the OSPF re-convergence process one can consider the example topology of Fig. 1 where R_1 to R_6 represents different network nodes, x in x[y] represents the link weight while y is the link bandwidth in Mbps. Link weights can range between and 65535 [16] and based on them (using the Dijkstra algorithm), each router/node calculates it's the Routing Information Base (RIB) and the corresponding Forwarding Information Base (FIB) by computing the Shortest Path Tree (SPT) towards all destinations from itself; in principle each entry in the FIB shows the next hop where a packet should be forwarded in order to reach its designated destination. With reference to Table I, for example, and based on the links weights of the topology in Fig. 1, the FIB for node R_1 shows that a packet destined for node R_6 should be forwarded to node R_3. As it will be explained, we can classify the links into two categories of interest; with the link weight determining to which category each link belongs to.

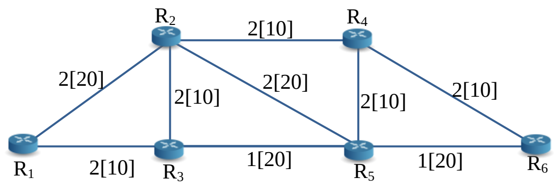

Fig. 1. Example topology for illustrating the ways links can be different in terms of OSPF re-convergence.

TABLE I
NEXT-HOP INFORMATION FOR NODES IN TOPOLOGY OF FIGURE 1

S\D	R_1	R_2	R_3	R_4	R_5	R_6
R_1	-	R_2	R_3	R_2	R_3	R_3
R_2	R_1	-	R_3	R_4	R_5	R_5
R_3	R_1	R_2	-	R_5	R_5	R_5
R_4	R_2	R_2	R_5	-	R_5	R_6
R_5	R_3	R_2	R_3	R_4	-	R_6
R_6	R_5	R_5	R_5	R_4	R_5	-

Stub link: a link is considered as a stub link if, given the link weight settings, it carries traffic generated only from the immediately adjacent upstream node (head node of the link) and not from any other nodes in the network. For example, the link between routers R_2 and R_5 is such a link because based on the current links weight settings in the network, no other router, further upstream from R_2 uses it. R_2, which is the head node of that link is the only router that uses it for traffic forwarding towards any destination in the network. In a similar manner, both links between R_2 and R_3 are stub links. What this implies is that if any of these links were to be removed the only traffic that would be eventually diverted would be the traffic originating from the head nodes of these links. This can be verified by viewing the new FIB table, assuming for example that the link between routers R_2 and R_5 has been removed.

TABLE II.
NEXT-HOP INFORMATION ILLUSTRATING THE EFFECT OF A STUB LINK REMOVAL.

S\D	R_1	R_2	R_3	R_4	R_5	R_6
R_1	-	R_2	R_3	R_2	R_3	R_3
R_2	R_1	-	R_3	R_4	***R_3***	***R_3***
R_3	R_1	R_2	-	R_5	R_5	R_5
R_4	R_2	R_2	R_5	-	R_5	R_6
R_5	R_3	R_2	R_3	R_4	-	R_6
R_6	R_5	R_5	R_5	R_4	R_5	-

When that link is removed, the FIB will remain the same except for the attributes in ***bold italics*** of the source node R_2 in row 3 columns 7 and 8 (see also the respective columns in Table I). That is, when that link is removed, its traffic will be diverted through the new shortest path following the principle of OSPF. Both traffics from R_2 to R_5 and R_6 respectively will be diverted through $R_2 \rightarrow R_3 \rightarrow R_5$ and $R_2 \rightarrow R_3 \rightarrow R_5 \rightarrow R_6$ respectively. As such, the FIB entry will change from R_5 to R_3 in both cases. This can also be confirmed for all other stub links in the network.

In a similar manner, if any of these links were to be restored back to the topology with the same link weights, then again the only traffic that would be diverted to flow through them would be the traffic originating from their respective head nodes. In all cases these changes in topology would be propagated to all routers in the network, but for all routers - apart from the head routers- they would not lead to any updates in their respective routing tables. As such, the service disruption linked with the OSPF re-convergence process is nullified if only these links are to be removed/restored.

Transit link: a link is considered as a transit link if, given the link weight settings, it carries traffic generated not only from its head node but also from other upstream routers. For example, the link between R_5 and R_6 is such a link, because routers R_1 and R_2 for example also use it to route their traffic towards R_6. The removal of this link from the topology would mean that the routing tables of both remote routers R_1 and R_2 would need to be updated. This can be illustrated by viewing the FIB entries in this case in Table III.

TABLE III.
NEXT-HOP INFORMATION ILLUSTRATING THE EFFECT OF A TRANSIT LINK REMOVAL.

S\D	R_1	R_2	R_3	R_4	R_5	R_6
R_1	-	R_2	R_3	R_2	***R_2***	***R_2***
R_2	R_1	-	R_3	R_4	R_5	***R_4***
R_3	R_1	R_2	-	R_5	R_5	R_5
R_4	R_2	R_2	R_5	-	R_5	R_6
R_5	R_3	R_2	R_3	R_4	-	***R_4***
R_6	R_5	R_5	R_5	R_4	R_5	-

As it can be seen (in ***bold italics***) in this case of a transit link removal, the FIB entries of not only the immediately adjacent upstream node, but of remote nodes further upstream would need to be updated. Since the update process of the routing tables is not synchronized, this means that while the routing tables are recalculated, an inconsistent "routing view" among the affected routers can lead to loops, oscillations, congestion and result to service disruption. In a similar manner, restoring such links back to the topology will have similar effects.

B. Framework Overview

Our framework capitalizes on this key difference with respect to link roles in the OSPF re-convergence process in a twofold way. As already mentioned, whether a link behaves as a stub or a transit link depends solely on the link weight settings. This means that it is possible by intelligently manipulating the link weights to increase the number of stub links. Our framework as such first tries in an offline manner to maximize the number of ***sleeping stub links***. Maximizing the number of sleeping stub links means that the opportunities of disruption free link restoration, when the traffic conditions call for this, are also maximized. To do so the offline part of our framework tries to maximize the number of sleeping stub links **when** the traffic conditions are at their lowest, making use of the well-established and acknowledged Genetic Algorithms in this maximization process. Optimizing the link weights in such a way, when the traffic conditions are at their lowest, increases the number of stub links that can be put initially to sleep, increasing therefore the disruption free link restoration operations available during runtime in response to the changing traffic conditions. The lowest traffic conditions are identified based on the use of historical traffic matrices, identifying the matrix with the lowest Maximum Link Utilization (MLU), and we will be referring to this topology as the pruned (or reduced) topology. We will also be referring to the offline part of our scheme as **G**enetic **A**lgorithm – assisted **s**tub **Li**nk **S**leeping **O**ptimization **A**lgorithm (GA-s-LiSOA).

The online part of our framework complements the offline part and then periodically monitors the actual network conditions and attempts to restore back to the topology the minimum number of previously sleeping stub links in case this is needed to resolve network congestion. This means that contrary to other schemes, the online part of our framework makes only small incremental changes to the topology every time, rather than waking up all links when network congestion is detected. As such energy efficiency gains are sacrificed only to the extent that it is needed to do so to ensure adequate network performance. And in all cases, these incremental changes are performed in a disruption free manner. It is worth noting that the existence of this proactive online part allows the use of the lowest MLU traffic matrix during the offline part in order to maximize the energy savings, since any traffic conditions that would otherwise lead to congestion are handled during runtime by the online part. This is in contrast to using the highest MLU TM that would not maximize energy savings due to the reduced number of removed links as a result of the very high level of traffic rate.

The runtime operation of our framework, which we will be referring to as **s**tub **Li**nk **W**ake-up **O**ptimization **T**echnique (s-LiWOT), can be implemented in a NetFlow enabled operational network [17], together with the use of Traffic Engineering Link State Advertisements (TE-LSAs) [18] [19], based on which a centralized controller obtains a complete view of the traffic conditions network-wide.

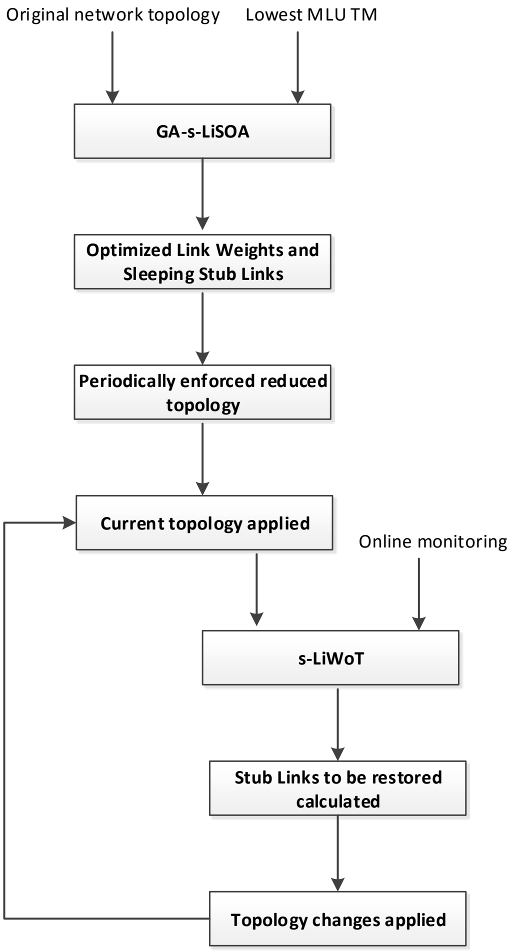

Fig. 2. Overview of LWO-DFTE.

At relatively long intervals that can be chosen by the network operator, (i.e. every 24 hours or every week) the operator can revert back the pruned topology as a whole and the online part of our framework can again start performing the operations needed in response to the actual traffic conditions. It is worth noting that this operation of reverting to the pruned topology is also again disruption free, despite the fact that multiple links may need to change state -from active to switched off- since all the links involved in this state transition are again only stub links. Fig. 2 shows an overall schematic of our framework, to which we will be referring to as **L**ink **W**eight **O**ptimized **D**isruption **F**ree energy-aware **T**raffic **E**ngineering (LWO-DFTE).

C. Problem Formulation

In this section, we provide an Integer Liner Programming (ILP) formulation of the problem we are trying to solve. Let us represent our network as a directed and connected network graph $G(V, E)$, where V represents the network nodes/routers and E represents the set of network links. $N = |V|$ is the total number of nodes and $L = |E|$ is the total number of links in the network. Table IV below summarizes the list of symbols associated with the problem formulation.

TABLE IV: DESCRIPTION OF VARIABLES

Variable	*Description*
P_{ef}	Energy efficiency
q_{ij}	Binary variable depicting on and off
ℓ_{ij}	Link connecting nodes i & j in that direction
t^{sd}	Total traffic demand from source (s) to destination (d)
ℓ_{ij}^{sd}	Traffic flow from s - d that is routed through ℓ_{ij}
L_{ij}	Total traffic load on link ℓ_{ij}
T	Threshold value for network link utilisation conditions
c_{ij}	Link Bandwidth
S_c^{sd}	Set of links in congested path routing traffic from s to d
S_n^{sd}	Set of links in new path routing traffic from s to d
ℓ_c	Congested link
E_s	Set of sleeping links
w_{ij}	Link weight connecting nodes i & j

We therefore formulate an ILP formulation of our problem where the objective function is to maximise P_{ef} subject to the constraints as explained below:

$$max\ P_{ef} \tag{1}$$

Subject to:

$$\sum_{j=1}^{N} \ell_{ij}^{sd} - \sum_{j=1}^{N} \ell_{ji}^{sd} = \begin{cases} t^{sd}, & \forall s,d,i=s \\ -t^{sd}, & \forall s,d,i=d \\ 0, & \forall s,d,i \neq s,d \end{cases} \quad (2)$$

$$\mathrm{w_t} = \sum_{\substack{i=j=1 \\ i\neq j}}^{N} q_{ij} \ell_{ij}^{sd} w_{ij} \quad (3)$$

$$L_{ij} \leq c_{ij} T \qquad \forall \{i,j\} \in E \quad (4)$$

$$S_n^{sd} \cap \ell_c = \emptyset \qquad \forall s,d \,|\, \ell_c \notin E_s \quad (5)$$

The objective function of our research is stated in equation (1) which is to maximize the total energy efficiency where the total energy ($\mathcal{P}_\mathrm{t}$) consumed in the network can be stated as -

$$\mathcal{P}_\mathrm{t} = \sum_{\substack{i=j=1 \\ i\neq j}}^{N} q_{ij} \, \mathcal{P}_{ij} \ell_{ij} \quad (6)$$

Equation (2) is the flow conservation constraint such that $\ell_{ij}^{sd} \geq 0$ where $\{s,d\} \in V.$ This is also supported by equation (3) for shortest path routing such that for any source destination path $\mathrm{sd_p}$, $\mathrm{w_t} \leq \mathrm{sd_p}$ and $q_{ij} = 1$. Equation (4) preserves the individual link utilization and by extension, the maximum link utilisation (MLU) in such a manner that there is no congestion. The determining factor here is T (which is dependent on the network administrator since L_{ij} and c_{ij} are network dependent). If the MLU is more than T, the network is considered to be congested. Therefore, the threshold value should be less than 100% so that congestion can be controlled in a proactive manner. More so, the higher the threshold value, the more the chances of links to enter sleeping mode and fewer links to need to be waken up to handle traffic upsurge. Equation (5) ensures that the selection of the minimum number of links to wake up does not involve the congested link and should be such that if –

$$S_c^{sd} = \{s, R_1, R_2, R_3, \ldots, d\} \quad (7)$$

where R_1, R_2, R_3 are the set of nodes of the congested path and the congested link is, for example, $\ell_c = \ell(R_1 \rightarrow R_2)$, then, S_n^{sd} must not contain $\ell(R_1 \rightarrow R_2)$. The objective of our framework is to maximize the energy efficiency which we define as:

$$P_{ef} = \frac{\sum \mathrm{P_f} - \sum \mathrm{P_p} + \sum \mathrm{P_w}}{\sum \mathrm{P_f}} \quad (8)$$

where $\mathrm{P_f}$ and $\mathrm{P_p}$ represent the energy consumption of the full and pruned topologies respectively while $\mathrm{P_w}$ is the energy consumption of the waken up links. As it has been shown in [6]-[9], determining energy efficient routing is NP-hard, even without considering the service disruption together with the other network and energy efficiency objectives. Taking this into account, we propose two well thought and reasoned heuristics for realizing the offline and the online part of our framework.

III. Framework Description

A. *Green Link Weight Setting for Maximizing Disruption Free Operations (Offline Part)*

In order to maximize the number of sleeping stub links when the network conditions are at their lowest, so as to maximize the subsequent opportunities for link restoration in response to changing traffic conditions, we employ a Genetic Algorithm on top of a variation of the **L**ink **S**leeping **O**ptimization **A**lgorithm (LiSOA) that was originally presented in [12]. The key differences with the work in [12] though are that:

- contrary to that version of LiSOA, we do not consider transit links for removal from the topology so as to nullify all disruption inducing operations; this is why we are referring to the version of LiSOA considering only stub link removal as s-LiSOA.
- with the introduction here of the GA, we do not take for granted the link weights as they are, but instead we attempt to optimize them so as to maximize the number of sleeping stub links under the lowest MLU traffic matrix; this is one of the key novelties of the work proposed in this work.

1) *s-LiSOA*

s-LiSOA, for a given set of link weights and the other existing properties of a topology (e.g. link capacities), attempts to put as many stub links as possible in the topology to sleep, assuming that the lowest MLU traffic matrix is mapped on the topology. To do so, s-LiSOA starts by mapping the lowest MLU TM on the full topology and ranks the stub links in ascending order based on their link utilization. Then it removes the least utilized stub link from the topology and evaluates the effects of this removal operation in terms of MLU in the remaining topology and disconnections in connectivity. If the remaining topology remains fully connected and the MLU remains below a threshold that is deemed as acceptable by the operator, then the stub link can be removed, otherwise it is restored back to the topology and the second least utilized stub link in the list is considered for removal. This process continues until all stub links have been considered for removal. Ranking the stub links in this order increases the chances of stub links being successfully removed from the topology, since the lower the utilization (traffic load) on a link the higher the chances that this traffic load can be assigned to alternative links without causing congestion to the corresponding routes. Considering the underlying IP network routing paradigm that assumes the existence of bidirectional connectivity between any pair of neighboring nodes, we assume a link layer tunneling mechanism is in place in order to emulate full bidirectional connectivity in the case of single (unidirectional) links [20] [21]. This allows our framework to address the removal and (as it will be shown afterwards) restoration of unidirectional links.

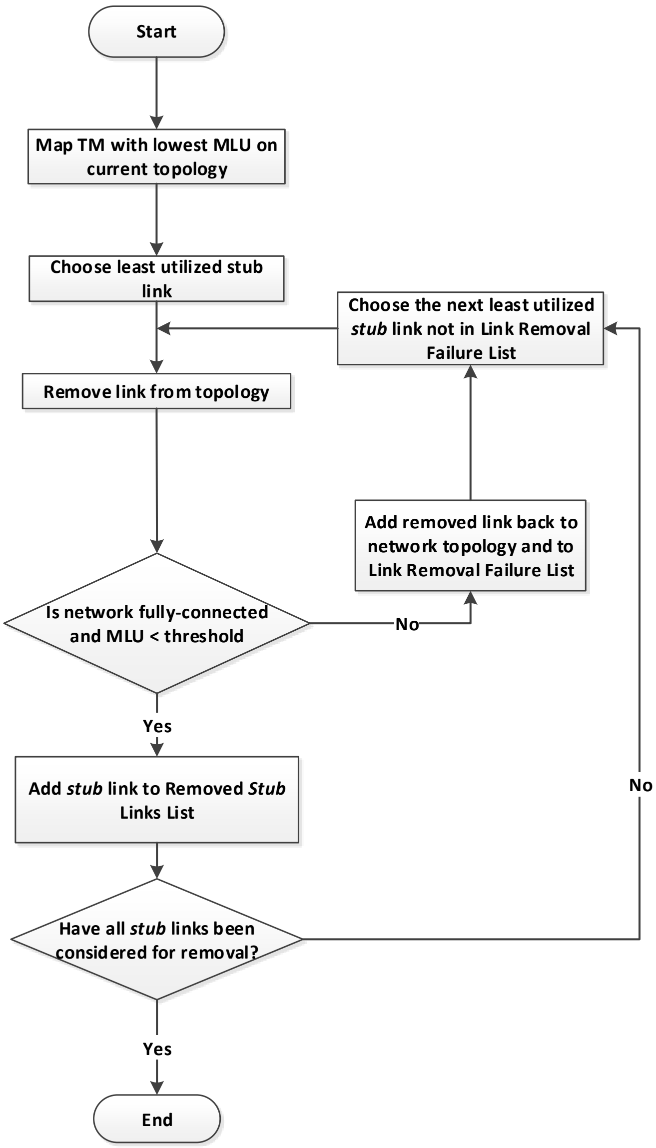

Fig. 3. s-LiSOA operations for stub link removal.

2) *GA-assisted s-LiSOA*

GA attempts to identify the set of link weights that maximize the number of sleeping stub links under the lowest (least MLU TM) traffic conditions. To do so the GA starts by generating an initial population of chromosomes where each chromosome is represented by an ordered sequence of genes, with each gene representing a link and its associated weight. Link weights generated can be an integer value in the range [1, 65535].

Since the objective is to maximize the number of sleeping stub links, s-LiSOA for the least MLU TM is run for the set of link weights in each chromosome and the number of sleeping stub links as an outcome of the s-LiSOA operation is used as the fitness value. The parent chromosomes are then sorted based on this fitness value and then a children generation is created using crossover and mutation operations and their fitness value is also calculated in a similar manner. The parent and children generation are then merged and sorted based on the fitness value and they are used to generate a new children population. This process continues for a pre-determined number of generations or until the improvement between successive generations becomes negligible.

The outcomes of this process are a) the calculation of the optimized link weights to be applied and remain the same throughout the operational runtime, and b) the pruned/reduced topology that is to be applied periodically and be incrementally modified by the online part of our framework.

B. Disruption Free Link Restoration (Online Part)

The disruption free link restoration during runtime is based on the scheme originally introduced in [12] as LiWOT. Contrary to the original scheme, the s-LiWOT version of it guarantees completely disruption free, in terms of re-convergence, link restoration rather than disruption minimizing link restoration, as was the case for LiWOT. To achieve this, s-LiWOT considers only stub links to be restored to address congestion issues, which as shown in previous sections ensures that no service disruption linked to routing tables recalculation takes place. s-LiWOT operates in a periodic fashion by intercepting the TE-LSAs that have been set to be broadcasted periodically and, since it operates in a centralized controller, it can reconstruct the full current network status in terms of congestion and maintain a logical view of the network. If there is no congestion detected (the congestion can be defined by the operator as the existence of an MLU value exceeding a certain threshold; this threshold can be set depending on the level of QoS the operator wants to ensure for the traffic) then no action is taken. If, however, there exist any links with MLU exceeding the allowable threshold value, then s-LiWOT tries to relieve the congestion in a way that not only does not create any service disruption but also sacrifices the energy savings as little as possible.

To do so, as soon as one or more links are detected as being congested, the flows on them (at the granularity of ingress-egress aggregates) are ranked in decreasing order with respect to their bandwidth demands and their source nodes are identified. Then for the flow with the higher bandwidth demands, the scheme searches whether there exists a currently sleeping stub link, the restoration of which can lead to the routing of that flow through an alternative path that does not traverse the congested links and, as such, can assist in reducing the MLU of the congested links. This process is followed for flows in the congested links as long as the congestion persists. Considering the flows in this process in decreasing order of their bandwidth demands offers the benefit that the minimum number of flows will need to be rerouted to resolve congestion, meaning that the minimum number of sleeping stub links will need to be restored and, as such, that the energy savings reduction is kept at the minimum possible. It is worth noting that only the very final outcome of s-LiWOT is applied and enforced to the actual topology. All intermediate steps taken to identify the final set of sleeping stub links that need to be restored are performed using the logical network view that the centralized controller can construct based on the intercepted TE-LSAs.

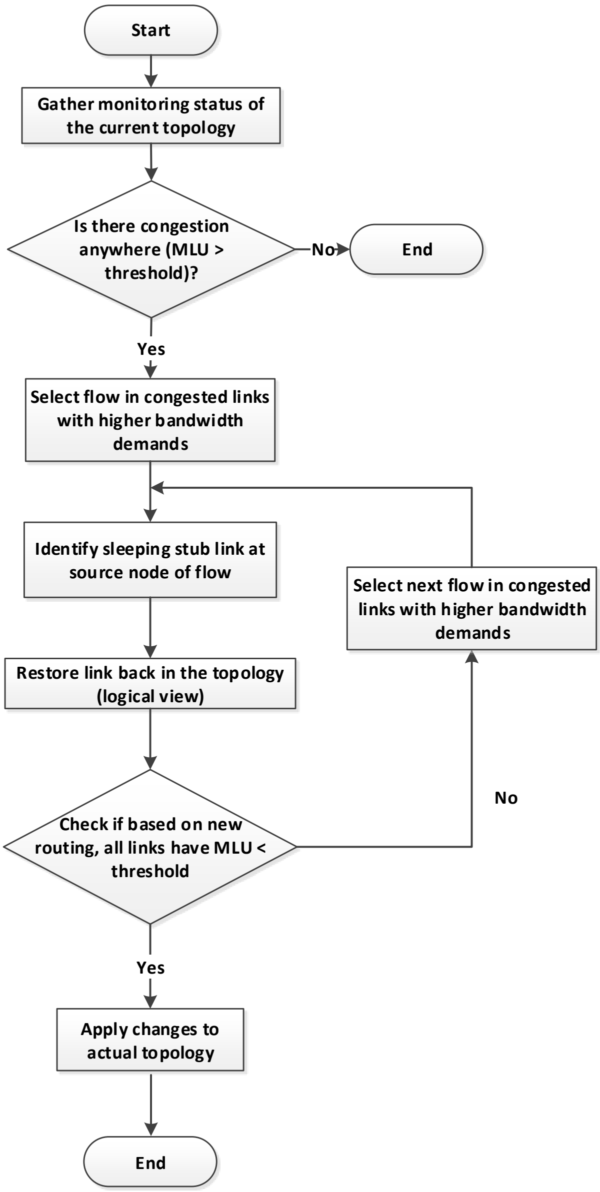

Fig. 4. s-LiWOT operations for stub link restoration.

IV. Performance Evaluation

A. Evaluation Setup

In order to evaluate the performance of our overall framework (LWO-DFTE) we performed a simulation based evaluation using the operational topologies of the Abilene and GEANT network topologies together with their actual traffic traces. The Abilene topology (USA) consists of 12 Point-of-Presence (PoP) nodes and 30 unidirectional links of varying delay and bandwidth capacities whereas the GEANT topology (Europe) is bigger consisting of 23 PoP nodes and 74 links. All links between any pair of nodes are logically bidirectional and symmetrical (formed by two separate unidirectional links of the same properties). For both topologies the following power model applies to the respective links [22].

TABLE V.
POWER CONSUMPTION RATE OF ROUTER LINE CARDS

Line card	Speed (kbps)	Power (Watts)
1-Port OC-192	9953280	174
2-Port OC-48	4976640	160
1-Port OC-48	2488320	140
1-Port OC-3	155520	60

The traffic traces used correspond to 1 week of operational runtime, corresponding to 672 TMs at 15 mins interval. With respect to our LWO-DFTE framework, this in principle means that:

- GA-s-LiSOA ran only once using as input the TM with the lowest MLU among all the 672 TMs available for the considered time period. The MLU threshold used during the link removal process was set to 100% to maximize the number of sleeping stub links. The link weights calculated by GA-s-LiSOA were applied and remained intact throughout the evaluated period.
- The reduced topology calculated also by GA-s-LiSOA was periodically enforced every 24 hours, using the time point corresponding to beginning of the off-peak period as the reference point for reverting to the reduced topology.
- s-LiWOT was run periodically at 15 mins intervals between two consecutive reduced topology reversion operations every 24 hours. The MLU threshold used during the link addition process was set to 90%; i.e. a link was considered as being congested if its utilization exceeded this value.

To also examine the robustness of the framework with respect to traffic unpredictability, for the evaluation of our framework we “inflated” the used traffic matrices by 10%. That is we assumed that the historical traffic matrices used by our framework for deriving the reduced topology were underestimating the actual future traffic demands by 10%. To evaluate the merit of the link weight optimization process itself in terms of the energy efficiency gains it brings, we also compare LWO-DFTE against two other approaches

- DFTE; in this approach there is no link weight optimization involved. Instead the scheme takes the existing original link weights as they are, applies s-LiSOA to calculate the reduced topology to be enforced every 24 hours, and applies s-LiWOT every 15 mins to make incremental updates to the topology.
- InvCaP-DFTE; in this approach again there is no link weight optimization involved. However, instead of relying on the existing link weights as they are, the scheme applies links weights according to the inverse capacity of the links. This approach can be beneficial for congestion avoidance since the bandwidth capacity of links is taken into account when deriving the routing of traffic through them. s-LiSOA is applied to calculate the reduced topology to be enforced every 24 hours, and s-LiWOT is applied every 15 mins to make incremental updates to the topology.

B. Link Weight Optimization energy Efficiency Benefits

In order to investigate the benefits with respect to energy efficiency that link weight optimization can bring, we compare LWO-DFTE against DFTE and InvCap-DFTE for both Abilene and GEANT topologies so we have a performance evaluation comparison in two different in terms of size but also in terms of traffic patterns topologies.

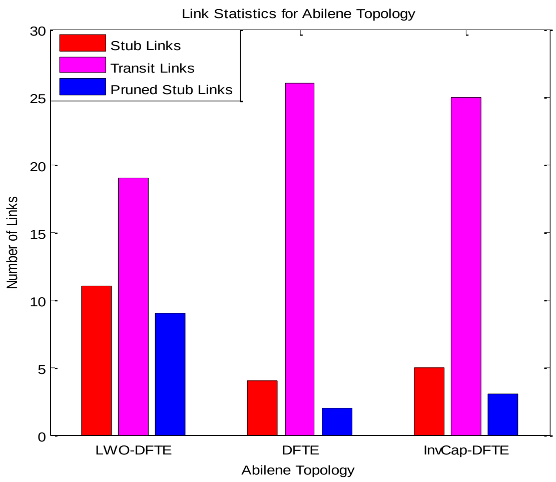

Fig. 5. Link statistics for LWO-DFTE, DFTE and InvCap-DFTE for the Abilene topology.

As it can be seen from Fig. 5, for the case of the Abilene network, all three schemes (i.e. all three link weight settings – existing original, optimized and inverse capacity) lead to a situation where the outcome of the offline part of the schemes is enough on its own to ensure the requested network performance in terms of MLU. As such, no stub link needs to be restored back to the topology at any point.

However, as Fig. 5 shows, by optimizing the link weights one can increase considerably the number of sleeping stub links, which in this case translates to an increase in energy savings since all these sleeping stub links remain in sleeping mode throughout the considered period. As Fig. 6 and Fig. 7 show, this increase in energy savings introduced by the link weight optimization (LWO-DFTE) does not come with the penalty of causing congestion to the network; on the contrary LWO-DFTE manages to still keep the MLU well below the set threshold and actually increases considerably the average link utilization (ALU) of links, which means that traffic is more evenly distributed among the links leading to a more balanced use of the network resources and reduction in link underutilization. It is worth noting that the LWO-DFTE scheme having higher MLU than the other schemes is not an issue, as long as it does not exceed the MLU threshold.

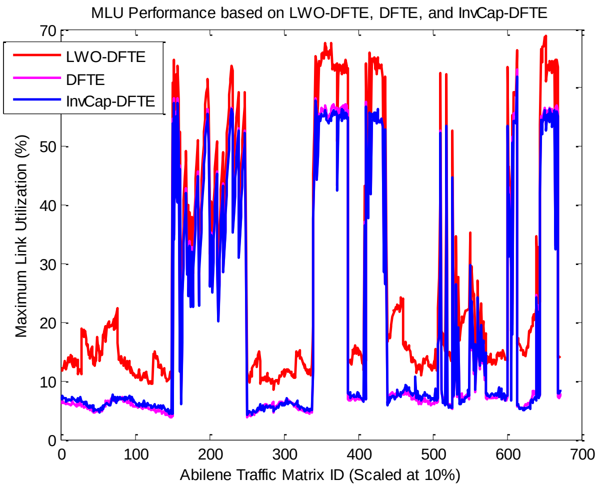

Fig. 6. MLU performance for LWO-DFTE, DFTE and InvCap-DFTE for the Abilene topology.

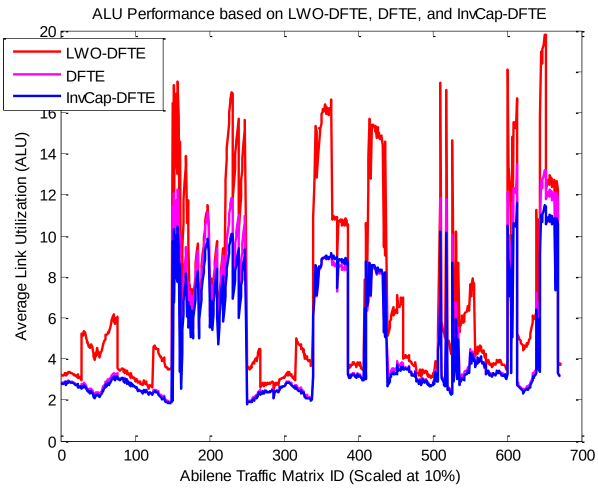

Fig. 7. Average link utilization performance for LWO-DFTE, DFTE and InvCap-DFTE for the Abilene topology.

For the case of the GEANT topology, as it can be seen from Fig. 8 and Fig. 9, the use of optimized link weights and also inverse capacity link weights leads to situations where, contrary to the case of the Abilene topology, the s-LiWOT part of the schemes needs to restore sleeping stub links during the operational runtime. While LWO-DFTE needs on average to restore slightly more sleeping stub links than InvCap-DFTE, given that the reduced topology that the offline part of LWO-DFTE uses as baseline, has almost twice as much sleeping stub links than the reduced topology of InvCap-DFTE, LWO-DFTE achieves still significantly higher energy savings despite the slightly increased number of restored sleeping stub links during the operational runtime.

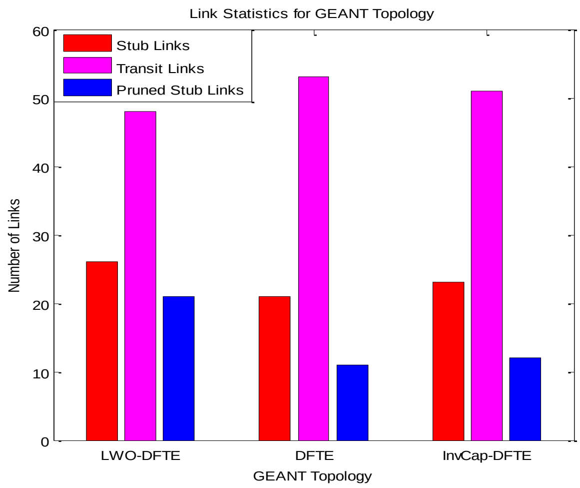

Fig. 8. Link statistics for LWO-DFTE, DFTE and InvCap-DFTE for the GEANT topology.

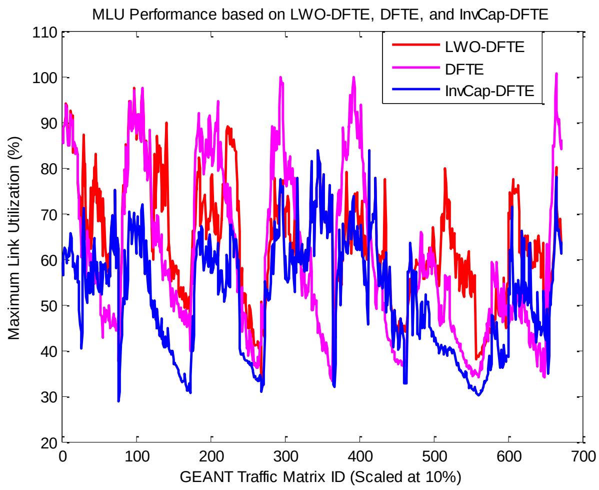

Fig. 10. MLU performance for LWO-DFTE, DFTE and InvCap-DFTE for the GEANT topology.

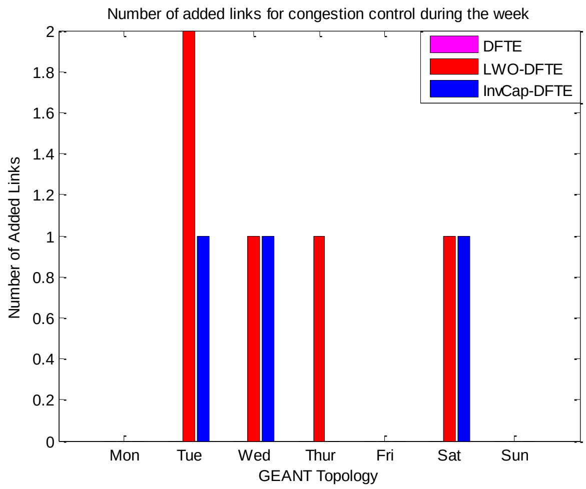

Fig. 9. Restored sleeping stub links for LWO-DFTE, DFTE and InvCap-DFTE for the GEANT topology.

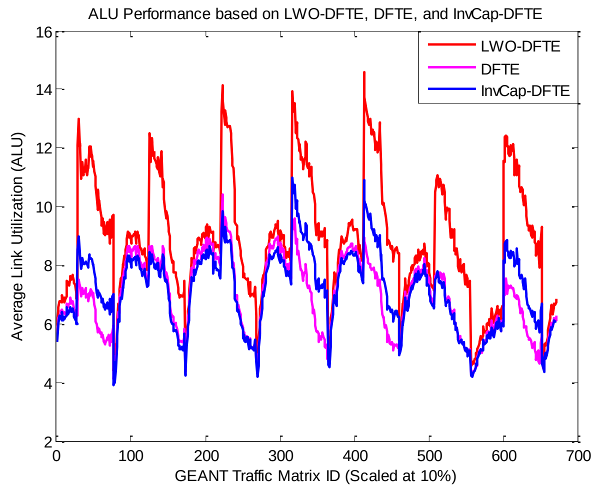

Fig. 11. Average link utilization performance for LWO-DFTE, DFTE and InvCap-DFTE for the GEANT topology.

As Fig. 10 and Fig. 11 below show, despite the much higher energy saving gains achieved by LWO-DFTE compared to the other two schemes, this again does not come with a reduction in the ability to relieve network congestion. On the contrary, similar to the Abilene topology case, LWO-DFTE is capable to always maintain the MLU below the set threshold and increase the average link utilization, balancing the use of resources and reducing link underutilization.

The following Table VI summarizes our main findings with respect to energy efficiency and link utilization performance, which confirm the ability of LWO-DFTE to provide increased energy efficiency gains together with a more balanced use of the resources, while meeting the network congestion performance objectives (expressed through the MLU in this case) compared to the other schemes that do not include the link weight optimization step.

TABLE VI.
ENERGY EFFICIENCY AND LINK UTILIZATION STATISTICS

	GEANT			**Abilene**		
	DFTE	LWO-DFTE	InvCap-DFTE	DFTE	LWO-DFTE	InvCap-DFTE
P_{ef} (%)	13.03	27.82	14.33	6.75	29.74	8.81
ALU	6.90	8.87	7.13	4.85	6.85	4.57
Sleeping stub links	11	21	12	2	9	3

It is also worth noting that our original scheme, as presented in [12], which does not attempt to completely nullify service disruption but rather just minimize it, needs to wake up on average 3 transit links on a daily basis to resolve congestion, when the same traffic conditions as the ones used for the evaluation of LWO-DFTE are presented to it. This would translate to significant service disruption on average 4 times per day due to routing re-convergence (resetting to the pruned topology plus the 3 transit links restoration every day) and which is something that the framework presented in this work (LWO-DFTE), does not suffer from.

C. Comparison with other schemes

In addition to the comparison above, we also compare the performance of our framework (LWO-DFTE) against other existing schemes so that we have an objective assessment of the performance of our framework; we implemented the following two schemes from the literature -

- The first scheme is the Mixed Integer Linear Programming based algorithm for Energy-aware Weights Optimization (MILP-EWO) as proposed in [1]. This scheme performs offline link weight optimization based on the historical traffic matrices and to derive three different topologies to be applied afterwards in a time-driven manner during the morning, afternoon and evening correspondingly, leading to service disruption 3 times per day.
- The second scheme is the Disruption-Free Green Traffic Engineering with NotVia Fast Reroute (DGTE-NotVia) [23]. This scheme also uses historical traffic matrices but does not perform any link weight optimization. It derives a reduced topology that can be applied during the off-peak period (evening) using NotVia Fast Reroute paths for diverting traffic away from the sleeping links.

The main strength of our scheme is that it provides a proactive mechanism such that when congestion occurs, it can incrementally select the minimum number of stub links in order to control congestion (also see Fig. 9 as an example of the low number of added links). Considering that in practice, diurnal traffics are dynamic in nature and therefore can increase randomly, we subjected these schemes to the same traffic conditions using 10% "inflated" traffic demands. As shown in Figures 12 and 13 respectively, while our framework increases the number of active network links in order to handle such traffic rise and prevent congestion, in both MILP-EWO and DGTE-NotVia schemes traffic continues to rise, above the 90% MLU threshold, especially in the case of DGTE-NotVia. This is due to the fact that since they operate in a time-driven way, but are based on offline calculations only, they cannot cope with cases where the actual traffic demands differ even mildly from the ones their offline calculations were based on.

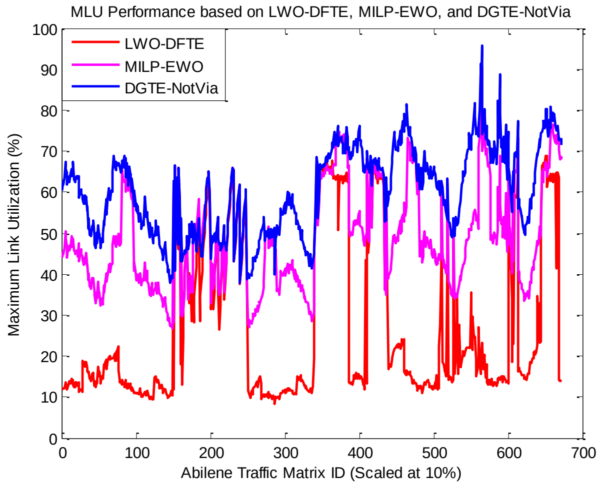

Fig. 12. MLU performance for LWO-DFTE, MILP-EWO and DGTE-NotVia for the Abilene topology.

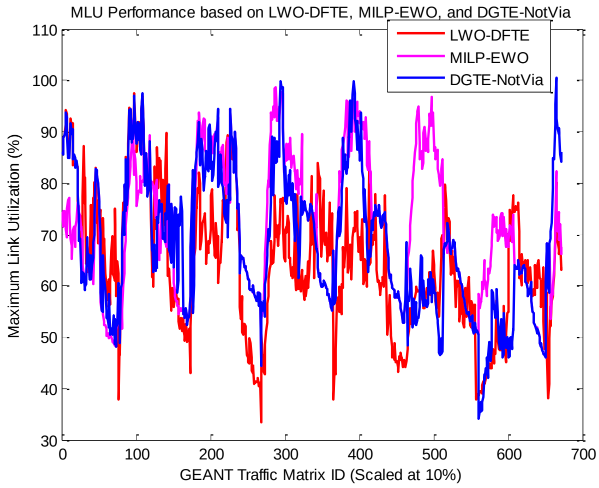

Fig. 13. MLU performance for LWO-DFTE, MILP-EWO and DGTE-NotVia for the GEANT topology.

In terms of energy efficiency, which comes in some cases at the expense of effective traffic control, as Table VII shows, MILP-EWO achieves higher gains especially in the evening when traffics are at their lowest rate (higher number of removed links). This, however, comes at the cost of induced

TABLE VII.
ENERGY EFFICIENCY AND LINK UTILIZATION STATISTICS

	GEANT			**Abilene**		
	LWO-DFTE	MILP-EWO	DGTE-NotVia	LWO-DFTE	MILP-EWO	DGTE-NotVia
P_{ef} (%)	27.82	46.35	26.81	29.74	37.75	18.25
Pruned links (morning/after noon/evening)	21	33/30/36	0/0/38	9	13/11/11	0/0/11

service disruption on a daily basis (3 times) and this could be detrimental especially for real time applications. The DGTE-NotVia scheme achieves on average lower energy efficiency gains due to the fact that it is applicable only in the evenings on a daily basis, reverting to the full topology during the peak time.

In order to show the effect that the link weight optimization process of our framework may have, when not applied with energy efficiency in mind, we applied the link weights calculated by our framework (LWO) and the link weights calculated by the MILP-EWO scheme (MILP) without however proceeding to the switching off of any links; that is the full topologies with these links weights were applied throughout. As Table VIII shows, the link weights calculated by the offline part of our framework, can still keep the MLU below the target value in all cases, even reducing it further compared to the original link weights (GEANT case).

TABLE VIII. MLU STATISTICS.

	GEANT			**Abilene**		
	LWO	MILP	Original	LWO	MILP	Original
MLU	88.68	94.29	91.43	64.88	61.91	63.24

V. RELATED WORK

Current practices of offline ETE schemes rely on static routing configurations that optimize network performance over a period of time based on historical traffic matrices. The authors of [24]-[27], [5] aim to put some links to sleep during the off peak period by mapping one of the TMs to the full topology. In [28], a time driven link sleeping algorithm that deploys a multi topology routing protocol was proposed. They aim to remove the least loaded links and then apply different sets of TMs to such reduced topology so as to optimize it for a particular period of time. This is different from the previous approaches as they claim that by considering different TMs, the reduced topology will be robust to diurnal traffic patterns. However, this comes with the challenge of providing backup links and thereby saving less energy in the process. The authors of [23] relied on the NotVia Fast Reroute scheme for the diversion of traffics from the sleeping links only during the off-peak time in a disruption free manner. In their scheme, based on the historical, off peak TMs, the algorithm determines a set of network links that will be configured to sleeping mode in a time driven manner. This implies that in the event of any traffic surge, the only alternative would be to revert to the full topology, thereby sacrificing energy savings. More so, it is obvious that the scheme introduces more overhead to the FIBs since each router will have to compute FIB entries for each sleeping link [29]. In general, most offline schemes are very simple to implement and robust to network scenarios where diurnal traffic patterns are relatively regular and predictable [30]. However they become suboptimal or even problematic in case of unpredictable traffic. In other to cope with this limitation, new online ETE schemes were introduced.

Online ETE are schemes that do not depend on any historical TM for their operation. Instead, they respond to the current state of the network at any particular point in time in order to deal with short timescale traffic dynamics. This makes it possible for a network to be adaptive in the way it uses its resources. In a nutshell, while offline approaches aim to provide a pruned topology (based on the available historical TMs) that will be applied during the off peak period, the online approaches make on-the-fly network reconfiguration decisions based on the current network status. Examples of such online schemes were proposed in [31]-[33]. The authors of [31] employed the knowledge of the current link load to switch off links, in combination with the history of past decisions. A resource management approach was employed in [32] in other to address energy efficiency. This is achieved by intelligent traffic distribution at the network edges. The authors of [33] proposed an energy critical paths approach to energy savings. In their work, offline energy paths are identified, then are deployed online while other sets of paths are set to low power mode. It is worth noting that none of these works tried to explicitly address the issue of service disruption due to routing re-convergence.

With respect to link weight optimization, while many approaches exist for link weight optimization towards network performance objectives, very few works have attempted to consider energy efficiency as an additional objective. The authors of [34] aim to optimize link weights so that there is limited number of active shortest paths for routing while still maintaining a congestion free network. In [1], link weight manipulation was exploited. That is to say that for each set of TM, there should be a corresponding link weight setting to switch to, thereby demanding as many reconfigurations as possible. The authors of [35] worked on optimizing the number of network configuration. They showed that their algorithm can select a minimum number of network configurations and their time duration in a day. In [36], pre-defined sets of link weights are set for a routing on demand architecture. They showed that such a scheme can achieve a trade-off between energy savings and maximum link utilization while still maintaining network packet delay within acceptable bounds. The only link weight optimization work that took traffic disruption into account is in [37]. The authors of [37] aim to choose only one set of link weights based on the different sets of TMs. However, this approach does not support traffic dynamics or guarantee robustness when subjected to real -possibly unpredictable-network conditions since is an offline approach that is based on historical TMs.

To the best of our knowledge, the framework proposed in this work is the first one that can provide energy efficiency gains through completely disruption free operations without making any rigid assumptions about the existence of well-defined and predictable traffic patterns. This is achieved by having two complementary parts (offline and online) inter-working in harmony; the offline part optimizing the link weight setting for energy saving purposes and the online part dynamically adjusting the topology in response to the current actual network conditions.

VI. CONCLUSIONS

In this paper we presented a novel framework towards energy efficiency in OSPF based network. The key novelty of this framework is its ability to provide significant energy saving gains without incurring any service disruption due to routing tables re-convergence in this process. In addition, the framework does not make any rigid assumptions about traffic dynamics which makes it suitable for networks that do not exhibit well-defined and predictable traffic patterns. This is achieved by the proper inter-working of two complementary schemes which operate at different time scales (offline/online).

As we showed through an extensive simulation-based evaluation using two real networks, different in sizes and traffic patterns, and through a comparison with other schemes, our framework is capable of providing significant energy savings and lead to a more balanced use of resources while always meeting the network congestion performance objectives. As we also showed our framework is also robust to discrepancies between historical traffic matrices it uses for its offline part and the actual network conditions it is subjected to during operational runtime.

Our framework is also easy to implement since it does not require any additional monitoring information on top of what can be provided through the combined use of NetFlow and TE-LSAs or any other modifications to the underlying (OSPF) routing protocol, which makes it also appealing from a practical implementation point of view.

References

[1] E. Amaldi, A. Capone, and L.G. Gianoli, "Energy-Aware IP Traffic Engineering with Shortest Path Routing" Comput. Network, 2013.

[2] C. Forster, I. Dickie, G. Maile, H. Smith, and M. Crisp, "Understanding the environmental impact of communication systems," OFCOM, Tech.Rep., 2009.

[3] H. Toure, "ICTs and climate change — The ITU perspective, "Climate Action, 2008 [Online]. Available: http://www.climateactionprogramme.org

[4] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study," in Proc. 2009 of the IEEE International Conference on Communications, June 2009, pp. 1–5.

[5] A. Adelin, P . Owezarski and T . Gayraud, "On the Impact of Monitoring Router Energy Consumption for Greening the Internet" in IEEE/ACM International Conference on Grid Computing, 2010.

[6] B. Fortz and M. Thorup "Increasing Internet Capacity Using Local Search" Technical Report IS-MG 2000.

[7] M. Pioro, A. Szentesi, J. Harmatos, A. Juttner P. Gajowniczek, and S. Kozdrowski, "OSPF related network optimization problems" Proc. IFIP ATM IP, 2000, UK.

[8] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-Aware Traffic Engineering," in *Proc. 2010 of IEEE International Conference on Network Protocols*, October 2010.

[9] M. A. Rodrigues and K. G. Ramakrishnan, "Optimal Routing in Data Networks" International Telecommunication Symposium (ITS) 1994.

[10] G. Shen and R. Tucker, "Energy-Minimized Design for IP Over WDM Networks," *IEEE J. Opt. Commun. Netw.*, vol. 1, no. 1, pp. 176–186, June 2009.

[11] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-Aware Traffic Engineering," in *Proc. 2010 of IEEE International Conference on Network Protocols*, October 2010.

[12] O. Okonor, N. Wang, Z. Sun, and S. Georgoulas, "Link Sleeping and Wake-up Optimization for Energy Aware ISP Networks, *in Proc. 2014 of IEEE International Symposium on Computers and Communications*, June 2014.

[13] The TOTEM Traffic Engineering Toolbox, http://totem.run.montefiore.ulg.ac.be

[14] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," Comput. Commun. Rev., vol. 36, pp. 83–86, 2006.

[15] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in Proc. SIGMETRICS, 2003, pp. 206–217.

[16] G. Retvari, and T. Cinkler "Practical OSPF Traffic Engineering" IEEE Communications Letters, VOL. 8, No. 11, November 2004.

[17] Netflow Analyzer, https://www.manageengine.com/products/netflow/

[18] D. Katz, K. Kompella, and D. Yeung, "Traffic Engineering (TE) Extension of OSPF Version 2" IETF RFC 3630, September 2003.

[19] L. Berger, I.Bryskin, A. Zinin, and R. Coltun "The OSPF Opaque LSA Option" RFC 5250, July 2008

[20] Cisco IOS IP Configuration Guide, "Configuring Unidirectional Link Routing",http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfudlr.html

[21] E. Duros, W.Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang "A Link-Layer Tunneling Mechanism for Unidirectional Links" RFC 3077, 2001

[22] Cisco 12000 series Packet Over SONET/SDH (POS) line cards, http://www.cisco.com/en/US/prod/collateral/routers/ps6342/product_data_sheet0900aecd803fd7b9.pdf

[23] N. Wang, C. Michael, and K. H. Ho, "Disruption-Free Green Traffic Engineering with NotVia Fast Reroute" in IEEE Communications Letters, October 2011.

[24] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in Proc. 2009 of the IEEE International Conference on Communications, June 2009, pp. 2298–2303.

[25] 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. Netw. Service Manag., no. 99, pp. 1–14, 2012.

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

[27] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: PowerAware Traffic Engineering," in Proc. 2010 of IEEE International Conference on Network Protocols, October 2010.

[28] F. Francois, N. Wang, K. Moessner, and S. Georgoulas, "Optimization for Time-driven Link Sleeping Reconfigurations in ISP Backbone Networks" in Network Operations and Management Symposium (NOMS), IEEE, April 2012.

[29] Q. Li, M. Xu, J. Wu, P. Lee, x. Shi, D. chiu and Y. Yang,"A unified approach to routing protection in IP networks", in IEEE Transactions on Network and Service Management 2012.

[30] B. Quoitin, S. Uhlig "Providing Public Intra-domain Traffic Matrices to the Research Community" ACM SIGCOMM Computer Communication Review, NY USA 2006.

[31] A. Bianzino, L. Chiaraviglio, and M. Mellia, "Grida: A green distributed algorithm for backbone networks," in Proc. 2011 of IEEE GreenCom, September 2011, pp. 113–119.

[32] M. Charalambides, D. Tuncer, L. Mamatas, and G. Pavlou, "Energy-Aware Adaptive Network Resource Management," in Proc. 2013 of the IFIP/IEEE Integrated Management Symposium, May 2013.

[33] N. Vasic, D. Novakovic, S. Shehar, P. Bhurat, M. Canini, and D. Kostic, "Identifying and Using Energy-Critical Paths," in Proc. 2011 of ACM CoNEXT, December 2011.

[34] F. Francois, N. Wang, K. Moessner, S. Georgoulas, "Green IGP Link Weights for Energyefficiency and Load-balancing in IP Backbone Networks", in: International Conferences on Networking (IFIP NETWORKING), 2013, pp. 1–9.

[35] L. Chiaraviglio and A. Cianfrani, "On the Effectiveness of Sleep Modes in Backbone Networks with Limited Configurations" in Software, Telecommunications and Computer Networks (SoftCOM), Italy, 2012.

[36] M. Shen, H. Liu, K. Xu, N. Wang, and Y. Zhong, "Routing on demand: toward the energy-aware traffic engineering with OSPF," in Proc. 2012 of the 11th International IFIP TC6 Conference on Networking, 2012, pp. 232–246.

[37] J. Moulierac, and T Phan "Optimizing IGP Link Weights for Energy-efficiency in a Changing World" Project-Teams COATI Inria France, May 2014.