Keeping the Connectivity and Saving the Energy in the Internet

Francesca Cuomo, Anna Abbagnale, Antonio Cianfrani, Marco Polverini

DIET, University of Rome "Sapienza", Via Eudossiana 18, 00184, Rome, Italy
francesca.cuomo@uniroma1.it, abbagnale@infocom.uniroma1.it, {cianfrani,polverini}@net.infocom.uniroma1.it.

***Abstract*—Nowadays a big effort is spent to reduce the Internet energy consumption. Actual Internet topologies have space to power off some links and devices to reduce the energy consumed in off-peak periods still guaranteeing connectivity among terminals. In this work we leverage the algebraic connectivity of the graph modeling an ISP network in order to define the ESACON (*Energy Saving based on Algebraic CONnectivity*) algorithm. We then consider the network connectivity as a first target performance to be assured. To this aim we identify a metric based on the algebraic connectivity that, on one side, allows to switch off several links with the consequent significant energy saving and, on the other side, still preserves network connectivity and network performance for efficiently supporting the Internet traffic. We find that ESACON achieves better performance with respect to similar topology-aware approaches; moreover ESACON performance are comparable with ones of a complex traffic-aware solution.**

***Index Terms*—Energy consumption, Green Internet, Graph theory**

I. Introduction

Recent literature estimates the power consumption of the Internet to be about 0.4% of electricity consumption in broadband-enabled countries and forecasts that this could approach 1% as access rates increase [1]. This consumption is due to both the customer premises equipments and to the switching and routing equipments. As already discussed in the literature there is space in the current Internet topologies to power off some links and devices to reduce the energy consumed in off-peak periods still guaranteeing connectivity among terminals [2], [3]. To this aim we propose to leverage the algebraic connectivity of the graph, modeling an ISP network in order to define an algorithm to power off some network links. By following guidelines given by the results of the paper in [4], we consider the network connectivity as a first target performance to be assured. Specifically, we start by measuring the current connectivity of an ISP and we provide an effective method for reducing this connectivity of a little percentage (around $1\% - 10\%$) by contemporary cutting a big slice (in the order of $20 - 130kW$) of energy consumed to power on line-cards of the network routers. Since the proposed metric to power off the links embraces several interesting topological characteristics, we are also able to counterbalance the resulting increment of links load that can arise in the off-peak periods when traffic is injected in the network, as well to limit the length of the resulting network paths. This is an interesting side effect that makes our proposal innovative in the field of the works dealing with procedures for pruning the network topology.

Our main contributions can be outlined as follows:

- We derive an algorithm for identifying network links that can be powered off by keeping the network connectivity above a suitable threshold. To the best of our knowledge, this is the first algorithm that jointly reduce the number of links in the network and keeps the network connectivity and other topological properties. The proposed algorithm considers and leverages the unique characteristics of the *algebraic connectivity* parameter.
- We discuss that our proposal can be implemented in a router with a quite reduced complexity and consequent reduced time execution. So the algorithm can be implemented in a centralized or in a distributed scenario.
- We compare the performance results of our algorithm with other similar approaches proposed in the literature. On one side we perform a comparison with a *topology-aware* solution and we show that our proposal achieves better performance. On the other side we also consider a *traffic-aware* solution and we show that our proposal presents comparable performance even if it is "blind" with respect to traffic to be supported in the network.

The paper is structured as follows. Section II introduces the adopted model and metric to measure the network connectivity, while Section III describes the algorithm and the implementation aspects. The analysis of the proposed solution is presented in Section IV. Finally Section V concludes the paper and sketches the future work.

II. Properties of Algebraic Connectivity

Graph theory is a powerful tool for modeling the structure of an IP network. A graph is represented as $G = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ is the set of vertices and $\mathcal{E}$ is the set of edges. The *adjacency matrix* of a graph G, denoted with $A(G)$, is a $N \times N$ binary matrix where $N = |\mathcal{N}|$ and the generic element a_{ij} is equal to 1 if $(i, j) \in \mathcal{E}$ and equal to 0 otherwise. The adjacency matrix of simple bidirectional graph is symmetric and has all diagonal elements equal to 0. The degree matrix $D(G)$ of G is a diagonal matrix with the generic element d_{ii} equal to the the degree of node i i.e., to the number of edges incident on i; i.e., $d_{ii} = \sum_j a_{ij} = \sum_j a_{ji}$.

In graph theory the connectivity of a graph $G(\mathcal{N}, \mathcal{E})$, is evaluated by using the *Laplacian matrix* ($L(G)$) [5]. This matrix is equal to the difference between $D(G)$ and $A(G)$. The Laplacian matrix of a bidirectional graph is symmetric and all its row and column sums are equal to 0. The eigenspectrum of $L(G)$ is defined as the set of its N eigenvalues, denoted as $\lambda(G)$, that can be ordered from the smallest to the greatest ($\lambda_1(G) \leqslant \lambda_2(G) \leqslant ... \leqslant \lambda_{\mathcal{N}}(G)$). The eigenvalues of the Laplacian matrix measure the connectivity of the graph.

Theorem 1 [6]: the smallest eigenvalue of the Laplacian of a bidirectional graph G is equal to 0 (i.e., $\lambda_1(G) = 0$) and the number of eigenvalues equal to 0 is the number of connected components of G.

Consequently, $\lambda_2(G) = 0$ iff G is disconnected; $\lambda_2(G)$ is generally called *algebraic connectivity*.

In this paper we propose to use the *algebraic connectivity* as a parameter to provide a methodology to switch off some links of Internet still keeping the resulting topology robust, to overcome some node and edge failures. In fact it is wide documented in the literature that $\lambda_2(G)$ encompasses several interesting characteristics of a network topology:

- it measures the stability and the robustness of complex network models [4]; a small perturbation in the network configuration will be attenuated back to the equilibrium with a rate proportional to $\lambda_2(G)$;
- the average distance between any two vertices of G is inversely proportional to $\lambda_2(G)$ [7];
- for any bidirectional graph G, the second eigenvalue of its Laplacian is upper bounded by its node connectivity which is equal to the minimum number of nodes whose deletion from G causes the graph to be disconnected or reduces it to a 1-node graph; it is then a lower bound on the degree of robustness of the graph to node and edge failures;
- if a graph G^l is obtained from G by delating the edge l, it has been proved that $\lambda_2(G^l) < \lambda_2(G)$; this means that the variation $\Delta^l = \lambda_2(G) - \lambda_2(G^l)$ measures the loss of connectivity produced by the removal of edge l.

By using the last property it is possible to evaluate the impact of an edge on the algebraic connectivity. Let l_1 and l_2 be two edges of graph G and let G^{l_1} and G^{l_2} be the graphs obtained from G by removing l_1 and l_2, respectively. In order to evaluate which of these two edges has a higher impact on the algebraic connectivity it is sufficient to compare Δ^{l_1} and Δ^{l_2}: we can state that l_2 has a higher impact on the algebraic connectivity compared with l_1 if $\Delta^{l_1} < \Delta^{l_2}$.

III. ESACON ALGORITHM

In this framework we model the Internet topology (i.e., the topology of an autonomous system in IP) with a bidirectional graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of routers and $\mathcal{E}$ is the set of bidirectional links connecting these routers. Let $N = |\mathcal{N}|$ and $E = |\mathcal{E}|$ be the cardinalities of sets $\mathcal{N}$ and $\mathcal{E}$, respectively.

A. Algorithm description

The proposed algorithm is named ESACON (*Energy Saving based on Algebraic CONnectivity*) and it is composed of two main steps:

1) the creation of an ordered list of links, denoted as $\mathcal{L}$;
2) the identification of a set of links to be switched off.

The pseudo-code in Algorithm 1 describes the two steps performed to run ESACON. As for the step 1, the input is the network topology G and the output is an ordered list of the links composing the network, $\mathcal{L}$. The criterion used to order the network's links is the algebraic connectivity: network's links are ordered based on their impact on the algebraic connectivity, since the aim is to switch off those links that have a low impact on the network connectivity. The algorithm associates to each bidirectional link $l \in \mathcal{E}$ the variation $\Delta^l = \lambda_2^{in}(G) - \lambda_2(G^l)$, where $\lambda_2^{in}(G)$ is the initial value of λ_2 for the graph G and $\lambda_2(G^l)$ is the algebraic connectivity that the network would have if the bidirectional link l was switched off (lines 6-10 of the Algorithm 1). We name $\vec{L}$ a vector containing the E bidirectional links and $\mathcal{L}$ is the ordered list obtained from $\vec{L}$ sorting in increasing order the values Δ^l. Links at the top of the list entail a lower reduction of the algebraic connectivity compared with those links at the end of the list.

As for the step 2, the input is the ordered list $\mathcal{L}$ and the output is the set of those links that can be switched off still keeping the connectivity over a given threshold. The set of these links is denoted with $\mathcal{S}$. In order to evaluate the loss of connectivity produced by powering off some links, we introduce the *residual normalized algebraic connectivity* defined as:

$$\gamma = (\lambda_2^{fin} / \lambda_2^{in}) * 100 \quad (1)$$

where λ_2^{fin} is the algebraic connectivity of a reduced graph G', obtained delating some links from the graph G. The step 2 has as many iterations as the number of bidirectional links of the network are. At each iteration the algorithm tries to include a new link in the set $\mathcal{S}$. At the generic iteration i, the algorithm checks if the link in position $i-th$ of the list, indicated with $\mathcal{L}[i]$, can be switched off. To this aim it is necessary to calculate the algebraic connectivity (λ_2^{fin}) of the graph $G' = (\mathcal{N}, \mathcal{E} - \mathcal{S} - \mathcal{L}[i])$. G' is obtained from the graph G deleting the link $\mathcal{L}[i]$ of the current iteration and the links already in $\mathcal{S}$ (line 15). ESACON verifies if γ is greater than a given threshold, denoted as γ_{th} (line 17). If this is the case the link $\mathcal{L}[i]$ can be switched off and it is added to the set $\mathcal{S}$ (lines 18-19), otherwise it remains active and the algorithm goes to the $(i+1)-th$ iteration.

The main goal of ESACON is to identify the largest number of links that can be switched off by keeping γ greater than a given threshold (denoted as γ_{th}). In the following Section we show that even if γ_{th} is kept quite high (around 90%-99%), ESACON is able to switch off a great number of links without compromising some key network performance.

B. ESACON implementation

ESACON algorithm proposes an innovative approach to save energy in a wired network, detecting a subset of links to put to sleep; the algorithm complexity is equal to $\mathcal{O}(E \cdot N^2)$, because for each bidirectional link the algebraic connectivity

Algorithm 1 $ESACON$

Input: a network graph $G(\mathcal{N}, \mathcal{E})$;
/* *STEP 1* */
$E = |\mathcal{E}|$;
λ_2^{in} is the algebraic connectivity of graph G;
$\vec{L} = zeros$;
for $each\ l \in \mathcal{E}$ **do**
  $\vec{L}[l] = l$:
  $\lambda_2^l = \lambda_2(G^l)$ where $G^l = (\mathcal{N}, \mathcal{E} - l)$;
  $\Delta^l = \lambda_2^{in} - \lambda_2^l$;
end for
$\mathcal{L}$ = sort $\vec{L}$ in increasing order based on the values Δ^l; /**Output of STEP 1**/
/* *STEP 2* */
$\mathcal{S} = \emptyset$;
for $i = 1, 2, ..., E$ **do**
  $\lambda_2^{fin} = \lambda_2(G')$ where $G' = (\mathcal{N}, \mathcal{E} - \mathcal{S} - \mathcal{L}[i])$;
  $\gamma = (\lambda_2^{fin}/\lambda_2^{in}) * 100$;
  if $\gamma \geq \gamma_{th}$ **then**
    switch off $\mathcal{L}[i]$;
    $\mathcal{S} = \mathcal{S} \cup \mathcal{L}[i]$;
  end if
end for
Outputs: $\mathcal{S}$ /*list of network's links that are switched off*/ and $G^{fin} = (\mathcal{N}, \mathcal{E} - \mathcal{S})$ /*final network topology*/.

(having a computation complexity equal to $\mathcal{O}(N^2)$) of the network obtained removing that link has to be computed. ESACON implementation is strictly related to the network scenario we consider: in particular we suppose to operate in a Autonomous System where a link-state intra-AS routing protocol, such as OSPF, is working. The availability of the network topology in every network node (thanks to the OSPF database) makes possible two different solutions: a centralized implementation and a distributed one.

In the first case a single network router executes ESACON algorithm and detects all the links to put to sleep. After that, a control message needs to be notified to each involved router, i.e. a router having at least a link to power down; this operation can be performed by means of a specific "Energy-saving" OSPF control message, that should be introduced into OSPF protocol. Finally, a control messages exchange phase is needed to notify links not more used to route traffic and so to change network paths.

In case of a distributed implementation every network node executes ESACON to evaluate if some of its links have to be power down, i.e. if they are in $\mathcal{S}$. The nodes then switch off their links in $\mathcal{S}$ and notify their new topological situation by mean of classical OSPF control messages, as for the last phase of centralized implementation. In this distributed solution the only main requirement is that all nodes have the same topological view, i.e. the same G. This can be achieved by running a classical link-state routing protocol. In both centralized and distributed cases we can classify ESACON as *topology-aware* approach.

Comparing the two approaches we can conclude that the centralized implementation has the advantage of performing the ESACON algorithm in just one node, while the distributed one does not have to perform an extra control message exchange phase that need the introduction of new OSPF messages. In our opinion, considering that: i) the algorithm complexity is limited to $\mathcal{O}(E \cdot N^2)$ and ii) the algorithm can also be executed off-line (the energy saving phase can be a planned operation), the distributed implementation is the most effective one.

IV. Performance analysis

In this analysis we use as main performance indexes:

1) the percentage of links that ESACON allows to switch off, denoted with η;
2) the mean utilization of the residual active links (i.e., on the links of network modeled by $G^{fin} = (\mathcal{N}, \mathcal{E} - \mathcal{S})$); it is indicated with U and expressed in percentage;
3) the energy that can be saved by applying ESACON, denoted with ξ (W);
4) the increase of number of hops in the network paths.

As for item 1), $\eta = S/(E - E_{min})$, where $S = |\mathcal{S}|$ is the number of links that can be switched off and E_{min} the minimum number of links that guarantees the network connectivity, that is shown to be equal to $E_{min} = (N - 1)$. As for the item 2) we generate a reference traffic matrix according to guidelines reported in [8]. In particular, we suppose that each router generates traffic towards any other router and the composition is: 40% of traffic is high bit-rate traffic, between $1Mbit/s$ and $80Mbit/s$, and the remaining 60% is a low bit-rate traffic, until $1Mbit/s$. The capacity of each link, indicated with C_l, is selected considering the availability of $2.5Gbit/s$ modules and imposing that the maximum link utilization, in the initial network, is lower than 25%. The mean utilization of the residual active links is computed as:

$$U(\%) = \frac{\sum_{l=1}^{E-S} U_l^{fin}}{E - S} \cdot 100 \qquad (2)$$

where U_l^{fin} is the utilization of the link l in the final topology.

As for item 3), we consider that each 2.5 $Gbit/s$ module consumes $\Xi = 140\ W$ [9], therefore a link of capacity $C_l\ Gbit/sec$ consumes an amount of energy equal to $(C_l/2.5) \cdot \Xi\ W$. In addition we consider that a link cannot be completely powered down, because of redundancy and fast recovery reasons. Therefore, a switched off link is a link in standby mode and it consumes a certain amount of energy; in this case we suppose that a standby link consumes about the 20% of an active link. As for the item 4) we evaluated, in the initial Internet topology G, the number of hops of a path, obtained by running Dijkstra as done in OSPF, connecting each possible source router with each possible destination router. Then we repeated the same analysis for the final topology G^{fin}, measuring the percentage of paths whose length increases of a certain number of hops due to the

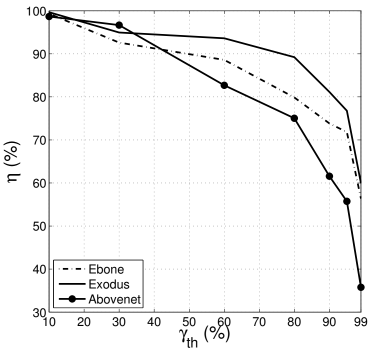


Fig. 1. η parameter as function of threshold γ_{th}

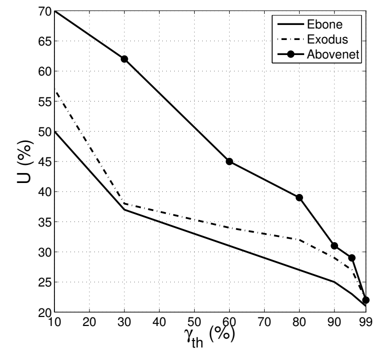


Fig. 2. U parameter as function of threshold γ_{th}

removal of links included in $\mathcal{S}$.

In this performance analysis we use real ISP topologies captured thanks to the Rocketfuel engine, namely "EXODUS ($N = 244$ and $E = 540$)", "EBONE ($N = 159$ and $E = 307$)" and "ABOVENET ($N = 366$ and $E = 968$)" [10]. The performance analysis includes:

- a discussion on the effect of γ_{th} on η and U;
- a comparison with a random switch off mechanism;
- a comparison with a topology-aware solution [3];
- a comparison with a traffic-aware approach [2].

A. Tuning of γ_{th}

Figure 1 presents the behavior of η as a function of γ_{th}. In this figure we do not consider $\gamma_{th} = 0$, since in this case the final network would be disconnected, therefore we use as minimum value of γ_{th} 10% . It can be noticed that as the threshold increases the number of links that can be switched off decreases due to the connectivity constraints. On the contrary, if we use low values of γ_{th} this has an effect on the mean utilization of the residual active links, that is the mean utilization of the residual active links increases when γ_{th} decreases, since the same amount of traffic has to be redistributed on a lower number of links compared with the initial topology G. This can be observed in Figure 2 where we plot the behavior of U as a function of γ_{th}.

Figure 3 shows the distribution of the path length increase for two different value of γ_{th} (95% and 99%). It is possible to notice that when the threshold increases the percentage of paths that do not increase the number of hops increases too. This may result in a lower increment of data delivery delay with respect to the case of low value of γ_{th}. This consideration is confirmed by the fact that the average distance between two vertexes of a graph is inversely proportional to the algebraic connectivity.

B. Comparison with a random switch-off

To have the first term of comparison for ESACON we defined one algorithm which switches off links by randomly selecting them in the original topology with the only constraint of keeping the network connected, that is assuring

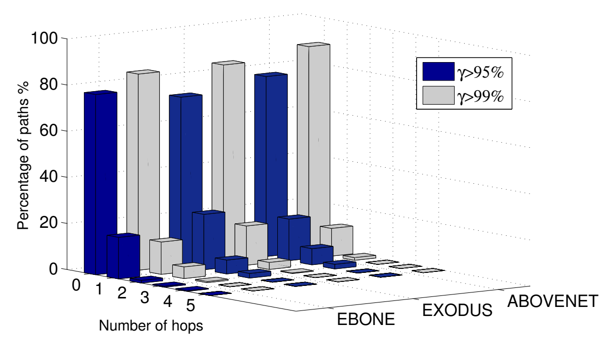


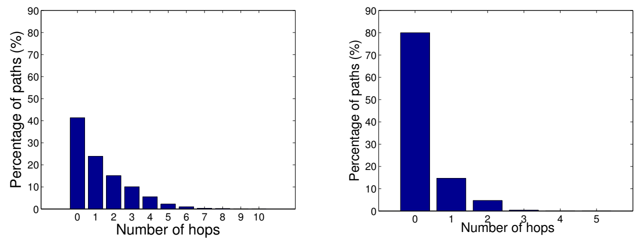
Fig. 3. Distribution of the path length increase

that $\lambda_2(G^{fin}) > 0$. We named this method RANDOM.

We analyze the performance of RANDOM and ESACON in terms of residual normalized connectivity and mean utilization, trying to keep constant the number of switched off links. To this aim, we set, for each of the three topologies, the number of links that ESACON allows to switch off keeping the residual normalized connectivity above $\gamma_{th} = 99\%$, obtaining the relevant value of U. Then we switch off the same number of links by applying RANDOM and we calculate γ and U achieved with this method. Table I reports the obtained results. It is possible to notice that even though RANDOM achieves the same η parameter, it produces a very high reduction of algebraic connectivity compared with ESACON, making worse the mean utilization on each active links. Besides we evaluate also the distribution of the path length increase. It can be noticed that by using RANDOM paths are longer on average with respect to ESACON. Figure 4 shows, for the EBONE topology, the distribution of the path length increase, both in case of RANDOM and in case of ESACON with $\gamma_{th} = 99\%$. In case of RANDOM the percentage of paths that maintain the same number of hops,

TABLE I
COMPARISON OF RANDOM AND ESACON IN TERMS OF γ AND U ON EQUAL TERMS OF η.

		RANDOM	ESACON
EBONE	γ (%)	35.1	99
	U (%)	28.5	21
EXODUS	γ (%)	40.9	99
	U (%)	35.9	22
ABOVENET	γ (%)	71.2	99
	U (%)	25	22

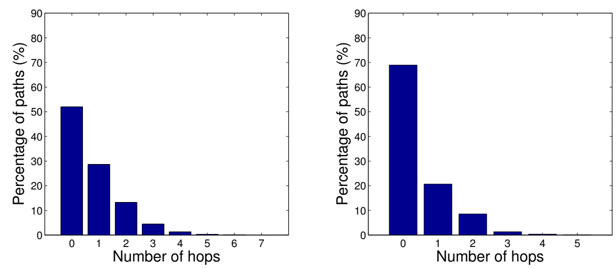


(a) RANDOM (b) ESACON with $\gamma_{th} = 99\%$

Fig. 4. Distribution of the path length increase, in case of EBONE topology.

TABLE II
COMPARISON OF d-EAR AND ESACON IN TERMS OF η, ξ AND U

		d-EAR	ESACON
EBONE	η (%)	52.7	73.8
	ξ (Wh)	21952	30352
	U (%)	21.8	24
EXODUS	η (%)	60.1	81.2
	ξ (Wh)	65296	74144
	U (%)	29	29
ABOVENET	η (%)	59.6	61.6
	ξ (Wh)	91280	136080
	U (%)	31.5	31



(a) d-EAR (b) ESACON with $\gamma_{th} = 90\%$

Fig. 5. Distribution of the path length increase, in case of EBONE topology.

after the deletion of the number of links deductible in Table I, is lower with respect to ESACON. Moreover, in case of RANDOM the average length of the network path increases, since there are also some paths that vary the length of 7 hops.

C. Comparison with topology-aware approaches

In this Subsection we compare ESACON with another topology-aware method for energy saving proposed in literature: distributed Energy-Aware Routing (d-EAR) [3].
d-EAR is a routing algorithm that is able to save energy performing the "exportation" mechanism. An oriented link is put to sleep forcing its source router, referred as the *importer*, to use the Shortest Path Tree (SPT) of a specific neighbor, referred as the *exporter*. The routers role, and so the links in energy save mode, is defined on the basis of the node-degree strategy: the routers having the highest degree "export" their SPTs to all their neighbors. The d-EAR main feature is its OSPF compliance. In this analysis we set for ESACON γ_{th} equal to 90%, for all the three topologies, and liken its performance to the ones of d-EAR. Table II shows the results of this evaluation. For all the topologies, ESACON switches off a greater number of links compared with d-EAR, entailing a greater amount of saved energy. In particular, in case of ABOVENET topology, even if the η parameter obtained with ESACON is slightly greater with respect to d-EAR, the ξ parameter of ESACON is significantly grater with respect to d-EAR, since ESACON switches off links characterized by a greater capacity compared with those powered down by d-EAR. Moreover the obtained mean utilizations are comparable with the two methods, expect in case of EBONE topology, where the mean utilization achieved with ESACON is slightly greater than the one in case of d-EAR.
During this comparison, we also evaluated the distribution of the path length increase, both in case of d-EAR and in case of ESACON with $\gamma_{th} = 90\%$. Figure 5 shows this analysis for the EBONE topology: it can be noticed that by using d-EAR paths are longer, on average, with respect to ESACON.
We can conclude that ESACON has better performance with respect to a different topology-aware approach.

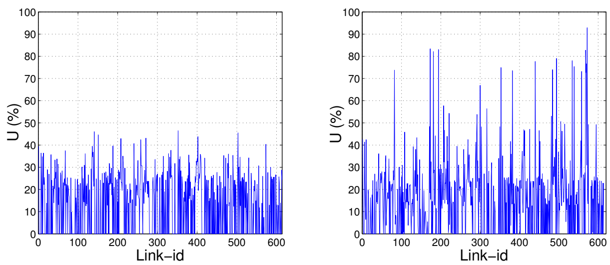
D. Comparison with traffic-aware approaches

As explained in previous Sections ESACON is a topology-aware algorithm, so it does not take into account traffic conditions. To better characterize ESACON performance we compare it with a traffic-aware algorithm [2], referred as ESTA (Energy Saving Traffic-Aware) in the following. In particular we consider the Minimum Flow (MF) version of ESTA algorithm, that detects the set of links to put to sleep on the basis of link load. In more detail, at each step ESTA evaluates a single network link, starting from the less used one, and it tries to reconfigure the network: all network paths are computed not using the link under evaluation and maintaining the maximum link utilization under a fixed value u_{th}; if the previous conditions are respected ESTA put to sleep the considered link. The previous operations are performed for each network link. In this way ESTA is able to optimally reconfigure the network distributing traffic among active links.

It is clear that ESTA algorithm will have better performance with respect to ESACON one because of traffic knowledge; our scope is to use ESTA as a reference point to evaluate ESACON impact on traffic performance. ESACON and ESTA have different configuration parameters, the connectivity threshold γ_{th} in the first case and the maximum utilization threshold u_{th} in the second one; to compare the algorithms we fix the parameters so that to obtain the same performance in terms of

TABLE III
COMPARISON OF ESACON AND ESTA IN TERMS OF η, ξ AND U

	η (%)	ξ (Wh)	U (%)
ESACON ($\gamma_{th} = 90\%$)	73.8	30352	24
ESTA ($u_{th} = 47\%$)	93	34496	24



(a) ESTA with $u_{th} = 47\%$ (b) ESACON with $\gamma_{th} = 90\%$

Fig. 6. Distribution of the link utilization when the mean utilization is equal to 24%, in the case of EBONE topology.

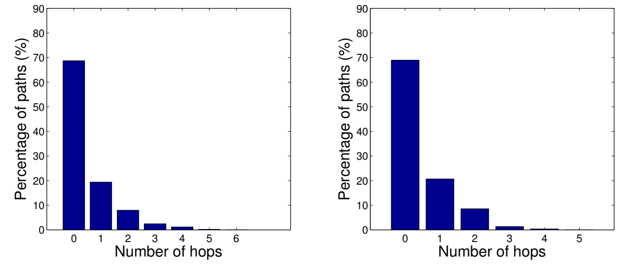
mean utilization on active links in both cases. We focus our attention on EBONE topology and we fix γ_{th} equal to 90% and u_{th} equal to 47% so that to have in both cases a mean utilization $U(\%)$ on active links equal to 24%.

In Table III the results of ESTA and ESACON algorithms are reported. The main result is that ESTA is able to switch off a greater percentage of network links: the reason is that the network is highly under-dimensioned, with a maximum link utilization lower than 25% with modules of $2.5Gbit/s$, and so ESTA can exploit available bandwidth to reconfigure network paths. In Figure 6 the distribution of links utilization is reported and it is shown that ESACON have different peaks while ESTA is able to equally split network traffic among active links; even if ESTA has better performance, as expected, ESACON results highlight its ability to maintain links utilization limited under a reasonable value ($U \leq 50\%$ for the 95% of links). Finally in Figure 7 the distribution of paths increase, in terms of number of hops, is reported. A detailed analysis shows that ESACON performance are slightly better than ESTA one: in the case of ESTA the percentage of paths increasing their length is greater and there are also some paths having a length increase equal to 6 hops, that never happens in ESACON case.

ESACON performance highlights its ability to reconfigure a network to obtain power saving with no knowledge of traffic matrix. ESACON is a "low-impact" energy saving solution, because it does not need to modify traffic engineering protocols; moreover, considering the comparison with a traffic-aware solution, it could represent a good starting point to implement a future algorithm where traffic QoS aspects are taken into account.

V. CONCLUSIONS

In this paper we proposed an energy saving solution for IP networks: the innovative aspect of our solution is that it exploits the algebraic connectivity. We defined a routing



(a) ESTA with $u_{th} = 47\%$ (b) ESACON with $\gamma_{th} = 90\%$

Fig. 7. Distribution of the paths length increase when the mean utilization is equal to 24%, in the case of EBONE topology.

algorithm, named ESACON, that is able to detect a set of links to power off maintaining the network connectivity above a fixed threshold. The links to power down are the ones that minimize the algebraic connectivity decrease of the network. We compared the ESACON performance, in terms of energy saving and mean links utilization, with respect to different solutions: we proved that ESACON has better performance with respect to similar topology-aware solution and that it has comparable performance with respect to a complex traffic-aware solution. These results highlight the interesting features of our solution, that represent a good trade-off between topology-aware solutions, with worse performance, and traffic-aware solutions, requiring a full knowledge of traffic conditions.

ACKNOWLEDGMENT

Authors would like to thank Katia Tedesco for the analysis of the ESACON performance.

REFERENCES

[1] J. Baliga, R. Ayre, K. Hinton, W. Sorin, and R. Tucker, "Energy Consumption in Optical IP Networks," *Lightwave Technology, Journal of*, vol. 27, no. 13, pp. 2391 –2403, July 2009.
[2] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Proceedings of the 2009 IEEE international conference on Communications*, ser. ICC'09, 2009.
[3] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, "An Energy Saving Routing Algorithm for a Green OSPF Protocol," in *IEEE INFOCOM Conference on Computer Communications Workshops*, 2010.
[4] A. Jamakovic and S. Uhlig, "On the relationship between the algebraic connectivity and graph's robustness to node and link failures," in *NGI 2007*, May 2007, pp. 96–102.
[5] C. Godsil and G. Royle, *Algebraic Graph Theory*, April 2001.
[6] B. Mohar, "The Laplacian spectrum of graphs," in *Graph Theory, Combinatorics, and Applications*. Wiley, 1991, pp. 871–898.
[7] ——, "Eigenvalues, diameter, and mean distance in graphs," *Graphs and Combinatorics*, vol. 7, no. 1, pp. 53–64, 1991.
[8] A. Nucci, A. Sridharan, and N. Taft, "The problem of synthetically generating IP traffic matrices: initial recommendations," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 19–32, July 2005.
[9] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-Aware Traffic Engineering," in *EEE ICNP*, 2010.
[10] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," *SIGCOMM Comput. Commun. Rev.*, 2002.