An OSPF Enhancement for energy saving in IP Networks

Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Polverini
DIET Department - University of Roma "Sapienza", Via Eudossiana 18, 00184 Roma, Italy
e-mail: {cianfrani,eramo,listanti}@infocom.uniroma1.it

***Abstract*—This paper deals with a strategy to save energy in an IP network during low traffic hours allowing a subset of IP router interfaces to be put in sleep mode by means of an Energy Aware Routing (EAR) strategy. The EAR is fully compatible with OSPF and is based on the "Shortest Path Tree (SPT) exportation" mechanism, consisting in sharing the SPTs among couple of routers. The EAR strategy is able to control the set of links to be put in sleep mode through the concept of "move". This approach gives the network operator the possibility to control the network performance and allows a smoothed QoS degradation strategy to be implemented. A formulation of the EAR problem is presented and will be demonstrated that this problem can be traced back to the well-known problem of the maximum clique search in an undirected weighted graph. A heuristics, called Max_Compatibility, is presented and, as shown in the performance evaluation study, it allows to save about 30% of network links with a negligible increase of network path lengths and link loads.**

***Index Terms*—Energy efficient networks, IP Routing, Performance Evaluation.**

I. Introduction

The reduction of energy consumptions is currently one of the key challenges also in the Internet world [1], [2]. In this field Internet has great opportunities since network resources are dimensioned on the basis of peak-hour traffic, neglecting traffic daily fluctuations, though traffic during night hours is less than 80% of peak hours. Energy saving strategies should be able to adapt the overall network infrastructure to the actual traffic conditions.

Energy-saving strategies at network-level [3] are here considered. They consist of coordinated mechanisms aiming at modifying network routing so that a subset of network devices, i.e. routers and links, can be put in a low power state (sleep state). Unfortunately, these strategies have impact on network control and routing protocols. Recent studies [4]–[6] propose power-aware routing algorithms based on the knowledge of the traffic demand and QoS constraints; these conditions cannot be assured by the current routing protocols (e.g. OSPF [7]); consequently, these algorithms can be exclusively viewed as off-line strategies. Moreover their implementation would need a strong modification of the IP routing architecture. Our proposal aims at overcomes these limitations by defining an on-line strategy fully compatible with routing protocol operation.

A two layers IP-over-WDM network is considered. The IP logical topology is mapped on an optical WDM network, composed of Optical Cross-Connects (OXCs) and fiber links. Each IP link, considered as unidirectional, corresponds to a dedicated WDM lightpath. A GMPLS control plane [8] is considered; in case of failures, is able to reconfigure the optical layer, by establishing/deleting lightpaths. For these reasons, the IP logical topology is assumed to be pre-determined and time invariant.

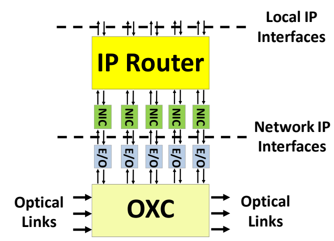


Fig. 1. Architecture of an IP-over-WDM node.

The figure 1 shows a simplified architecture of a network node. The OXC in the optical layer originates/terminates lightpaths, or, if the node is a transit node for a lightpath, switches it between the relevant incoming and outgoing optical interfaces. A specific IP interface (NIC: Network Interface Card) is associated to each logical IP link originated or terminated by the node; a NIC interacts with Optical Layer by means of an E/O converter.

The energy saving strategy here proposed is based on the ability to put to sleep a subset of IP interfaces during low traffic periods. This is obtained by modifying the network paths used to route IP traffic flows. If an IP interface is put to sleep, the associated IP link is switched in a stand-by state and the packet flows crossing it are routed through other active IP links. This approach does not lead to changes of the logical IP topology and is completely independent of the optical layer operation.

In [9] we proposed an IP level energy saving routing algorithm, called distributed Energy-Aware Routing (d-EAR) algorithm. It is based on the "Shortest Path Tree (SPT) exportation" mechanism, that consists in using only a subset of routers SPTs to select the routing paths. In this paper we propose an advanced version of EAR strategy that overcomes the limitations of the old d-EAR solution. The "SPT exportation" has been enhanced by defining the concepts of "move" that provides the means to early evaluate all the positive and negative effects of an exportation. So, EAR is able to keep under control the set of links to be put in the sleep mode and, by applying the concept of "compatibility" among moves, can modulate the network performance and allows a QoS strategy

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

to be implemented. Finally, EAR intrinsically guarantees the absence of cycles, so it avoids the exchange of OSPF control messages during a reconfiguration phase needed by d-EAR solution.

The problem of finding a set of compatible moves, able to minimize the network consumption, is discussed and it is shown that it can be formulated as the maximum clique problem in a graph. As this problem is known to be NP-hard, a heuristics, called Max_Compatibility heuristic, is proposed aiming at finding the set of moves characterized by the maximum number of links to be switched off. The energy saving obtained through the EAR algorithm, in terms of percentage of links put in low power mode, can reach about the 30% with a negligible increase of the network path lengths and link loads.

The paper is organized as follows: in Sec. II the general principles of the EAR strategy are described; in Sec. III the ”move” concept and its properties are introduced. The general formulation of the problem of the network consumption minimization and the heuristics used for its solution are discussed in Sec. IV. Finally, in Sec. V the performance of EAR strategy are evaluated in case of a reference network.

II. The Energy Aware Routing (EAR) strategy

The most popular intra-AS routing protocol is the Open Shortest Path First (OSPF) [7]. OSPF-LSA messages allow each IP node to compute its own Shortest Path Tree (SPT), i.e. the set of shortest paths from itself to all the network destinations. Our scope is to define an IP level energy saving mechanism compatible with OSPF operation; in particular, it is to be avoided that link state changes (link switch off or on) be seen by OSPF as link failures (or restorations) determining a topology change. This for two main reasons: i) OSPF would generate a LSA flooding round to update the router topology databases with a useless signaling traffic generation; ii) a path re-computation procedure following to a link state change is too slow, as a matter of example, it could take several seconds or tens of seconds. Analogously, a strategy based on OSPF link weights modification, so as to deny traffic flow in a certain links, would be complex to handle and would fall in the previous two problems. EAR strategu , thanks to the SPT exportation mechanism, is able to reduce the number of IP router interfaces used to forward packets with no IP topology changes: it can be implemented by using the same OSPF data structures and does not interfere with OSPF normal operation.

A. The EAR idea and its implementation

The basic idea of the path computation strategy adopted in EAR is that the IP links used to route traffic can be reduced if a subset of SPTs is used instead of the full set (one for each router) as in case of classical OSPF. In this way, the IP interfaces not more used can be switched in a sleep mode. Although hardware implementation aspects are out of the scope of this paper, we assume that NICs support low power mode of operation, analogously to the solution adopted in IEEE 802.3az Energy Efficient Ethernet [10]. Possibly, a periodic wake up of the whole interface allowing the exchange of OSPF Hello Packets could be needed to maintain the adjacency of IP routing protocols; in this way, the IP routing protocol still maintains the same logical topology, so as to avoid a flooding phase of LSA packets.

It is to be observed that, by assuming that all the IP routers have a consistent view of the IP network topology in their LSA databases, each router is able to compute the SPT associated to each other. Normally, a router only computes its own SPT via the Djikstra’s algorithm; diversely, the EAR operation is based on the concept of ”SPT exportation”: a set of routers, called ”exporters”, forces the use of their SPTs to another set of routers, called ”importers”. Practically, a router belonging to the importer set identifies its own exporter and computes the SPT associated to it by assuming the exporter itself as the root node; successively, having the exporter’s SPT as a reference, the importer router will determine its own path tree, named Modified Path Tree (MPT), using the same IP links of the exporter’s SPT.

The EAR strategy can be implemented in a centralized way with a limited impact on OSPF routing protocol implementation. One of the network routers can be elected as the EAR coordinator and its function is to compute the routers roles by means of the LSA database and of the heuristic explained in the following sections. The coordinator can start the energy saving phase sending a specific OSPF message to each importer router specifying the associated exporter router. Of course this new OSPF message needs to be implemented into the OSPF code. The EAR implementation we propose allows the EAR coordinator to activate the energy saving strategy when traffic decreases under fixed thresholds. In fact, routers roles depend only on network topology and so they need to be updated only when a topological modification, notified by a new LSA reception, occurs. Moreover, if a topological modification takes place during the energy saving phase, we suggest to recompute the network paths not considering the exportation mechanism, as happens in today OSPF network: in this way the coordinator can recompute router roles and afterwards re-activate the energy saving phase.

B. EAR mode of operation

The role of a router, i.e. importer of exporter, is assigned according to the set of links, called *target links*, that are candidate to be switched off. This approach gives the operator the possibility to control the network performance and allows a smoothed QoS degradation strategy to be implemented.

Let T be the set of target links, i.e. the set of links candidate to be switched off, for a target link $l \in T$ a specific exportation has to be identified, i.e. a couple of importer and exporter nodes. The importer unambiguously corresponds to the router R_i being the source of the link l; this is quite obvious since, if the target link l has to be put in a sleep mode, the network paths of R_i have to be recomputed so that l will not be used by R_i any more. The identification of the exporter router is more complex. A candidate to be an exporter router for the link l must meet the following two conditions: i) the link l does not have to belong to its SPT; ii) it has to belong to the set of neighbors of R_i. The first condition is quite obvious, whereas the second one is introduced to minimize the paths

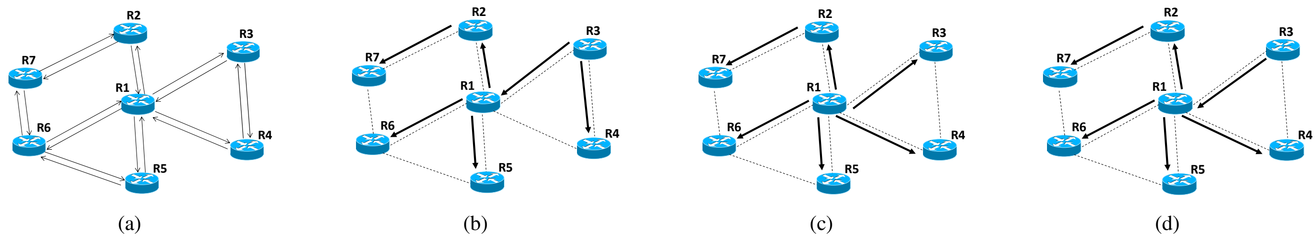


Fig. 2. a) Example of a network graph with OSPF weights. b) SPT computed by $R3$, $SPT(R3)$. c) SPT computed by $R1$, $SPT(R1)$. d) EAR algorithm performed by $R3$ if $R1$ is the exporter router, $MPT(R3, R1)$.

length increase. If more than one router satisfies the previous conditions the choice will depend on the predictable effects of the exportation. In the following, we indicate with $M(l)$ the set of exportations that can be carried out in order to switch off the link l.

As an example, in Fig.2a a simple network topology composed by 7 IP routers and 18 direct links is shown; for sake of simplicity all the IP links have unitary OSPF weights. Let's assume that the target link is the link $(R3, R4)$. The router $R3$ will be the importer router, whereas, $R1$ could be a feasible exporter router because it is a neighbor of $R3$ and the link $(R3, R4)$ does not belong to its SPT (see Fig.2c). The MPT of $R3$ resulting from the exportation is shown in Fig.2d, the link $(R3, R4)$ is not more used to route traffic and can be put in a low power state.

It is to be noted that, the set $M(l)$ could be empty, i.e. it is not always possible to find an exporter that allows to put to sleep a target link. This could be due to two reasons: i) an importer neighbor not having the target link in its SPT does not exist or ii) a previous exportation has limited the possible exportations that can be executed. These aspects led to the definition of the concept of "move" that is discussed in the next Section.

III. The concept of move and its properties

An IP network can be represented by means of a weighted directed graph $G(V, E)$ where V is the set of nodes representing the IP routers and E is the set of directed edges being the IP links. For each directed edge $e \in E$, $S(e)$, $E(e)$ and $w(e)$ denote the source node, the end node and the weight associated to the link, respectively. Let N and L be the cardinalities of V ($N = ||V||$) and E ($L = ||E||$), respectively. The set of all the shortest paths from the node v to all the other nodes, i.e. the SPT associated to the node v, is indicated as $SPT(v)$.

A node x is adjacent to a node i if there exists a directed link e from i to x, i.e. $S(e) = i$ and $E(e) = x$. An exportation, being the effect of sleeping a target link l outgoing from i, could be carried out having i as importer and x as exporter, so the node i could be able to compute a Modified Path Tree (MPT) having $SPT(x)$ as reference. It follows that the link l identifies the target link and the link e represents the exportation able to switch off the link l.

It is to be noted that there could be different ways to put in sleep mode the link l; let $M(l)$ be the set of exportations associated to a target link l. The choice of one of these alternatives leads to a specific network state and determines different consequences on energy saving and network performance. For this reason, from now on we indicate a generic exportation $m \in M(l)$ with the term *move*.

As each direct link identifies a move having the source node as importer and the destination node as exporter, therefore the maximum number of moves that can be performed in a network described by a graph $G(V, E)$ is equal to L. We are interested to investigate the relationships between moves; in particular, our goal is to define simple rules to establish if two moves can be performed one after another or if they are mutually exclusive, i.e. the execution of one excludes the application of the other. This concept can be summarized introducing the following definition of *compatibility* between a couple of moves:

- Compatibility condition: given a couple of moves m_1 and m_2, the move m_2 is said to be *compatible* with the move m_1, and we use the notation $m_2 \propto m_1$, if m_2 can still be performed after the execution of m_1; otherwise the move m_2 is said to be *incompatible* with the move m_1.

It is to be observed that the compatibility is symmetric, i.e. if $m_2 \propto m_1$ so $m_1 \propto m_2$, that means that the two moves m_1 and m_2 can be executed in any order.

Given a move m, we are interested to define the set $M_{NC}(m)$ of all the moves *incompatible* with m. As a first step, we fix the set of *conditions* that have to be verified so that two moves are compatible; they can be classified in *procedural* conditions and *performance* ones.

The *procedural* conditions can be derived from the "exportation" definition itself; in particular, if an exportation, having i as importer node and x as exporter node, is performed, the following conditions (C1-C3) hold:

C1) i cannot be the importer of a next exportation;
C2) i cannot be the exporter of a next exportation;
C3) x cannot be the importer of a next exportation.

The *performance* conditions represent the constraints that we establish to control the effects of the routing path changes on network performance. In particular, we introduce the following Rule 1:

- Rule 1: an SPT associated to an exporter node cannot be modified by any exportation, i.e. the routing path trees of exporter nodes must remain the shortest ones.

This rule determines a twofold effect: i) a great part of new the network paths is identical to the shortest ones and ii) the rerouted paths are close enough, in terms of number of hops, to the shortest ones. If the Rule 1 holds, it is possible to prove that in a network with equal edges weights, the maximum increase of network paths in terms of number of hops is equal to two. For space reason the proof is omitted. The application of the Rule 1 implies two further extra-conditions:

C4) a link used by the SPT of an exporter cannot be put in sleep mode;

C5) a node having a path modified by an exportation cannot be an exporter.

If the conditions C1-C5 hold, it is possible to demonstrate that the new network path routing is loop free.

Once defined the concept of move and defined the compatibility conditions between two moves, the problem of energy saving in an IP network can be equivalently formulated as the problem of finding a set of compatible moves able to minimize the network consumption by assuring a given level of QoS.

In the next section we focus on the specific problem of finding the set of compatible moves assuring the highest energy saving; in other words we try to give a response to the following question: *given a network topology, which is the set of compatible moves that allows to put in low power mode the maximum number of links?*. In the following, this problem will be called briefly *EAR problem*.

IV. Solution of the EAR problem

Let us consider an IP network modeled by a directed graph $G(V, E)$. We define an undirected graph $H = (M, C)$, where each node $m \in M$ represents a move m and the edges $c \in C$ indicate the compatibility relationships between the moves: if there exists an edge c between the nodes m_1 and m_2, then the moves m_1 and m_2 are compatible. Moreover, each node m is characterized by a weight $w(m)$ which represents the number of IP links that the move m allows to put to sleep. In $H = (M, C)$, a set of compatible moves is represented by a clique $K_H(n)$ where n is the number of nodes of the clique. A clique is a complete subgraph of H in which every pair of nodes is adjacent.

Each clique $K_H(n)$ of H is characterized by its weight $W(K_H)$, given by the sum of the weights of all the clique nodes. In our case, since two compatible moves cannot switch off the same IP links, $W(K_H)$ is the number of IP links that the set of compatible moves represented by $K_H(n)$ allows to put to sleep. So the EAR problem is equivalent to find the complete subgraph of H characterized by the maximum weight. A simplified version of this problem, obtained if every node has the same weight, is known in literature as the maximum clique problem of a graph, and it is *NP-hard*; as a consequence our problem is *NP-hard*, too. In the next subsection, we propose a heuristic, called Max_Compatibility heuristic, that aims at finding the largest set of compatible moves. In addition we propose a further heuristic, the Min_Used_Links heuristic, that tries to put to sleep a particular subset of network links considering a particular link property and not the moves compatibility.

A. The Max_Compatibility Heuristic

To explain the Max_Compatibility heuristic, let us introduce some further notations. Each move m_i is characterized by a compatibility vector $\mathbf{c}_i$, representing the compatibility relationships among m_i and all other moves:

$$\mathbf{c_i} = \{c_{ij}, 1 \leq j \leq L\},\ with\ \ c_{ij} = \begin{bmatrix} 0 & m_i \not\propto m_j \\ 1 & m_i \propto m_j \\ 0 & i = j \end{bmatrix} \quad (1)$$

The compatibility degree g_i of the move m_i represents the number of moves compatible with m_i:

$$g_i = \sum_{j=1}^{L} c_{ij}. \quad (2)$$

A possible solution S_k, that allows to switch off a set of IP interfaces, is represented by a set of compatible moves:

$$S_k = \{m_k,\ \ k \in K\},\ \ c_{ij} = 1\ \forall i, j \in K\ with\ i \neq j. \quad (3)$$

We define an utility function $U(S_k)$ to compare different solutions:

$$U(S_k) = \sum_{k \in K} w(m_k), \quad (4)$$

where $w(m_k)$ is the number of IP interfaces that the move m_k puts in sleep mode. In this way each solution is characterized by its ability to save energy. Of course, we are interested in the detection of the best solution S_k^*, that maximizes $U(S_k)$:

$$U(S_k^*) = max\ U(S_k)\ \ \forall k. \quad (5)$$

It is to noted that, if we consider the the maximum clique problem, i.e. a simplified version of the EAR problem characterized by constant weights, the solution S_k^+ corresponds to the set of moves with maximum cardinality. So

$$||S_k^+|| = max\ ||S_k||\ \ \forall k. \quad (6)$$

The Max_Compatibility Heuristic, described in Algorithm 1 pseudocode, has been defined considering both the previous problems. In the first part the heuristic tries to solve the maximum clique problem by detecting a set of possible solutions S_k characterized by the highest number of moves. In the second part, the best solution, in terms of energy saving, is chosen, by evaluating the function $U(S_k)$.

In particular, the heuristic immediately selects the max compatibility move m_M, the one having the highest compatibility degree g_M (Line 1). In this way we force the solution to contain the move m_M. The next step (Lines 3-8) defines g_M candidate solutions S_j that will be evaluated during algorithm execution; each set S_j is initially composed by the move m_M and a compatible move m_j. Moreover, two extra data structures are introduced: the set M_j containing the moves that could be inserted into S_j in next steps, that is composed by all moves compatible with both m_M and m_j; the vector $\mathbf{c_{Sj}}$ that represents the compatibility vector of the moves m_M and m_j considered at the same time. In Lines 9-16 the sets S_j are fulfilled; to each solution S_j the move in M_j having the highest compatibility with S_j is added, until no more residual compatible moves are remained (i.e. $M_j = \{\emptyset\}$); the set M_j and the compatibility vector $\mathbf{c_{Sj}}$ are re-computed each time a new move is inserted into S_j. Finally (Line 17) the best solution S_j^* in terms of energy saving, is chosen.

Algorithm 1 Max_Compatibility Heuristic

Find m_M s.t. $g_M = \max_{j \in L} g_j$
$j = 1$
for all m_k t.c. $c_{Mk} = 1$ **do**
  $S_j = \{m_M, m_k\}$
  $j = j + 1$
  $M_j = \{m_i$ t.c $c_{iM} = 1$ AND $c_{ik} = 1\}$
  $\mathbf{c_{Sj}} = \mathbf{c_M}$ AND $\mathbf{c_k}$
end for
for $j = 1$ to g_M **do**
  while $M_j \neq \{\emptyset\}$ **do**
    $m_l = \max_{m_k \in M_j}(\mathbf{c_{Sj}}$ AND $\mathbf{c_k})$
    $S_j = S_j \cup \{m_l\}$
    $M_j = M_j - \{m_l\} - \{m_i \in M_j$ s.t. $c_{il} = 0\}$
    $\mathbf{c_{Sj}} = \mathbf{c_{Sj}}$ AND $\mathbf{c_l}$
  end while
end for
$S_j^* = \max_{1 \le j \le g_i} U(S_j)$

B. The Min_Used_Links Heuristic

The Min_Used_Links heuristic orders the links on the basis of the number of paths crossing them and less used links are put in the sleep mode.

The Min_Used_Links heuristic is based on the knowledge of the SPTs associated to every router; in fact, to determine link utilization, the heuristic associates to each link l_i a value n_i, representing the number of routers having the link l_i in their SPT. After SPTs computation, the links l_i are ranked according the n_i value. At each step the link having the lower n_i value is extracted from the list and a feasible move is searched and the relevant link switched off. Afterwards, the new SPTs are computed and the links list is updated. The procedures terminates when, considering the compatibility conditions defined in Section IV, there are no more feasible exportations to be executed.

V. Performance evaluation

The performance of EAR strategy will be evaluated by comparing results arising from application of Max_Compatibility and Min_Used_Links heuristics to real network topologies available thanks to the Rocketfuel project [11]. These heuristics will be also compared with the old version d-EAR [9]. The networks considered for the comparison are: i) Ebone composed of 159 nodes and 614 links; ii) Exodus (244 nodes and 1080 links); iii) Abovenet (366 nodes, 1932 links) and iv) Sprint (516 nodes, 3186 links). In all the network topologies we have always assumed equal OSPF links cost; ; we have obtained similar results in the case of variable OSPF weights, but for space reasons these results are not shown here. Considering that the IP links are directed, in the following we indifferently refer to IP links and IP interfaces. In particular, the expression "the IP link is in the sleep mode" synthetically indicates that the source IP interface has been switched off.

The figure of merit used for the comparison is the maximum percentage of links η_{eMAX} that is possible to put in low power mode if no QoS constraints are considered, i.e.

$$\eta_{eMAX} = L_e/(L_D - L_{min}). \tag{7}$$

herein: L is the number of IP links; L_e is the number of links that the specific strategy allows to put to sleep; L_D is the number of links belonging to at least one SPT when classical Djikstra routing is performed; L_{min} is the minimum number of links guaranteeing the network connectivity.

The range of variation of η_{eMAX} is [0,1]. In particular, $\eta_{eMAX} = 0$ is obtained if the classical Dijkstra's algorithm is performed implying $L_e = 0$; $\eta_{eMAX} = 1$ is reached is the residual active topology is a tree (i.e. $L_{min} = 2(R-1)$ where R is the number of network routers).

In Figure 3 the results obtained through the application of the three energy aware strategies are reported.

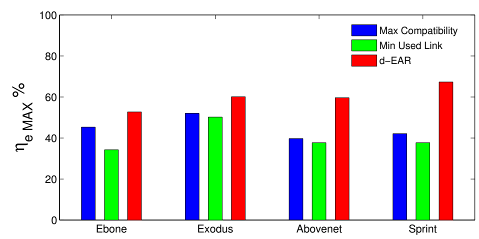
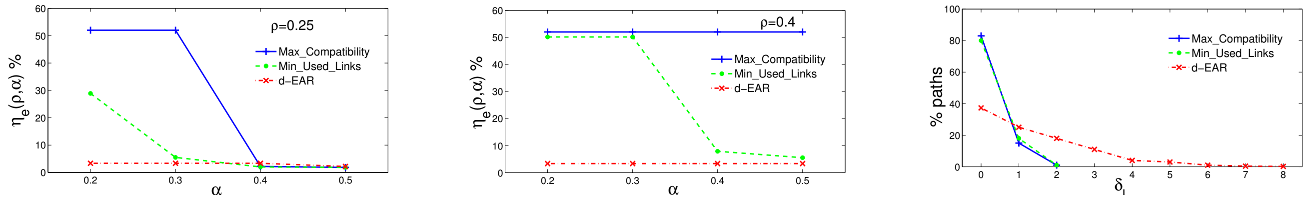


Fig. 3. Percentage of IP interfaces in sleep mode.

Two preliminary observations arise: 1) for all the network topologies the highest percentage of links put in low power mode is achieved by the d-EAR strategy ; ii) the Max_Compatibility heuristic always provides better performance with respect to the Min_Used_Links one. The first remark can be explained by considering that the new heuristics performance are limited by the performance conditions (C4 and C5). As for the second observation, the performance differences have to be imputed to the specific heuristics strategy: the Max_Compatibility tries to detect the larger set of compatible moves, in order to have the higher number of links to power down, while the Min_Used_Links tries to minimize the effects of paths re-computation, powering down the less used links. In order to highlight the behavior of the three policies regarding the QoS aspects, the previous results have to be examined more in detail, so we focus our attention on the Exodus topology. Anyway, it is to be noted that the general conclusions that will be derived are independent of the specific topology and hold whichever network is examined.

The real energy saving opportunities are computed considering that a network needs to assure satisfying level of QoS; therefore, we evaluate η_{eMAX} as a function of the maximum link utilization ρ obtained after the path rerouting due to the switch off of a set of links. A reference traffic matrix has been generated according to guidelines reported in [12]; in particular, we supposed that each router generates traffic towards any other: the 40% is high bit-rate traffic, between 1 Mbit/s and 80 Mbit/s, and the remaining 60% is low bit-rate traffic, up to 1 Mbit/s. Both high and low bit rate traffics have a lognormal distribution. The link capacities are dimensioned on the basis of peak-hours traffic, fixing the target maximum link utilization equal to 0.25. The daily traffic fluctuations are taken into account introducing α parameter, which represents the scaling factor of the traffic matrix and of course $\alpha \le 1$.



(a) (b) (c)

Fig. 4. The real energy saving obtained for different traffic conditions when a) $\rho = 0.25$ and b) $\rho = 0.4$; c) distribution of paths length increase δ_l

By indicating with $\eta_e(\rho, \alpha)$ the percentage of links that is possible to switch off as a function of the maximum link utilization ρ and of the traffic scaling factor α, Figures 4a and 4b show $\eta_e(\rho, \alpha)$ vs. α, considering two different values of maximum allowable link utilization ρ. The "hard" policy does not allow any increase of the ρ with respect to the target design value, i.e. $\rho = 0.25$, whereas the "soft" policy allows a maximum link load equal to 0,4 to be reached. By observing the curves shown in the previous Figures, two main conclusions can be derived. First of all, the d-EAR solution is not able to exploit its theoretical power saving capabilities: in all network scenarios d-EAR reaches maximum value of $\eta_e(\rho, \alpha)$ equal to about 4%. This result highlights the importance of the performance conditions (C4 and C5) that were introduced in the new heuristics. The second remark is even more interesting: the Max_Compatibility heuristic has always better performance with respect to the Min_Used_Links one. This is an unexpected result since, in principle, the Min_Used_Links, trying to put to sleep the less used links, would have to minimize the path re-routing effects. The better performance reached by Max_Compatibility heuristic can be explained focusing on the compatibility relationship: a move with a high compatibility degree is able to "coexist" with a high number of different moves, so allowing a good solution in terms of maximum number of links to power down, but it is also able to change the network paths so that the impact on link utilization is minimized. So we can affirm that the Max_Compatibility heuristic provides the best performance in terms of real power saving.

The last parameter we have considered to compare the performance of the proposed energy saving strategies is the distribution of path lengths in terms of number of hops δ_l shown in Figure 4c. This parameter could give a preliminary indication of the expected delay increase due to the path re-routing. The Figure 4c, on one hand, indicates that the old d-EAR policies could entail extra path lengths up to 7-8 hops, whereas, on the other hand, confirms that both Max_Compatibility and Min_Used_Links heuristics assures a maximum increase of path length equal to 2 as discussed in Section III. Moreover, it is to be observed that about the 80% of network paths do not change their lengths. This last result further highlights the importance of Rule 1 introduced to make the exportation mechanism a feasible power saving strategy.

VI. Conclusions

In this paper a novel strategy, called EAR, to save energy in an IP-over-WDM network during low traffic hours has been proposed. EAR strategy is fully compatible with OSPF and it is based on the the concept of sequence of "moves" aiming at determining the best mode to switch off a set of links and to reroute the paths crossing them. EAR is able to modulate the network performance and allows a QoS strategy to be implemented. The problem of finding the best set of moves has been formulated as the classical problem to determine the maximum clique in a graph. Two heuristics have been proposed and their performance have been evaluated. The obtained results show that it is possible to reduce the number of active links, putting in sleep mode about the 30% of network links, maintaining the link load less than 30-40%. Further studies are now being carried out on the definition of a new version of the OSPF protocol able to support EAR and on its implementation in an experimental test-bed.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreements n 247674 (STRONGEST-Scalable, Tunable and Resilient Optical Networks Guaranteeing Extremely-high Speed Transport) and from the Italian Research Ministry under the PRIN EFFICIENT grant.

References

[1] J. G. Koomey, "Estimating total power consumption by servers in the U.S. and the world", Staff Scientist, Lawrence Berkeley National Laboratory and Consulting Professor, Stanford University. Feb. 2007
[2] EC DG-INFSO, Impacts of Information and Communication Technologies on Energy Efficiency, Sept. 2008.
[3] M. Gupta, S. Singh, "Greening of the Internet", Proceedings of ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.
[4] G. Shen, R. S. Tucker, "Energy-Minimized Design for IP Over WDM Networks", Journal of Opt. Comm. Net., Vol. 1, No. 1, June 2009.
[5] L. Chiaraviglio, M. Mellia, F. Neri, "Energy-aware Backbone Networks: a Case Study", GreenComm 2009, Dresden, Germany.
[6] M. Zhang, C. Yi, B. Liu, B. Zhang, "GreenTE: Power-Aware Traffic Engineering", IEEE ICNP 2010, Kyoto, Japan.
[7] J. Moy. "OSPF Version 2" , Request for Comments 2328, April 1998.
[8] "Generalized multi-protocol label switching (GMPLS) architecture", RFC 3945, October 2004
[9] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, E. Vittorini, "An Energy Saving Routing Algorithm for a Green OSPF Protocol", IEEE INFOCOM 2010, San Diego (USA), March 2010.
[10] IEEE P802.3az Energy Efficient Ethernet Task Force Available at: http://grouper.ieee.org/groups/802/3/az/public/index.html.
[11] Rocketfuel Project [Online]. Available *http://www.cs.washington.edu/research/networking/rocketfuel/*.
[12] A. Nucci, A. Sridharan, N. Taft "The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations", ACM SIGCOMM Computer Communication Review, Vol. 35, No. 2, pp. 19-32, July 2005.