

Energy-Aware Weight Assignment Framework for Circuit Oriented GMPLS Networks

Original

Energy-Aware Weight Assignment Framework for Circuit Oriented GMPLS Networks / Ciarrocchi, M.; Bianzino, ARUNA PREM; Mellia, Marco; Donadio, P.; Parladori, G.. - STAMPA. - (2013). (Intervento presentato al convegno ITC 2013 tenutosi a Shangai, China nel 10 September 2013) [10.1109/ITC.2013.6662961].

Availability:

This version is available at: 11583/2519099 since:

Publisher:

Published

DOI:10.1109/ITC.2013.6662961

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Energy-Aware Weight Assignment Framework for Circuit Oriented GMPLS Networks

Marcos Ciarrocchi
Dipartimento di Elettronica
Politecnico di Torino
Torino, Italy
marcos.ciarrocchi
@studenti.polito.it

Aruna Prem Bianzino
Department of Computer
and Information Science
Linköping University
Linköping, Sweden
aruna.bianzino@liu.se

Marco Mellia
Consorzio Nazionale
Interuniversitario per le
Telecomunicazioni (CNIT)
Torino, Italy
mellia@polito.it

Pasquale Donadio
and Giorgio Parladori
Alcatel-Lucent Italia
Vimercate, Italy
(pasquale.donadio,giorgio.parladori)
@alcatel-lucent.com

Abstract—A branch of green networking research is consolidating. It aims at routing traffic with the goal of reducing the network energy consumption. It is usually referred to as Energy-Aware Routing. Previous works in this branch only focused on pure IP networks, e.g., assuming an Open Shortest Path First (OSPF) control plane, and best effort packet forwarding on the data plane. In this work, we consider instead Generalized Multi-Protocol Label Switching (GMPLS) backbone networks, where optical technologies allow to design “circuit switching” network management policies with strict bandwidth reservation policies.

We define a simple and generic framework which generates a family of routing algorithms, based on an energy-aware weight assignment. In particular, routing weights are functions of both the energy consumption and the actual load of network devices. Using such weights, a simple minimum-cost routing allows finding the current least expensive circuit, minimising the additional energy cost. Results obtained on realistic case studies show that our weight assignment policy favours a consistent reduction of the network power consumption, without significantly affecting the network performance. Furthermore, the framework allows to trade energy efficiently and network performance, a desirable property at which ISPs are looking for. Simple and robust parameter settings allow reaching a win-win situation, with excellent performance in terms of both energy efficiency and network resource utilization.

I. INTRODUCTION

The importance and penetration of Information and Communication Technologies (ICT) in today living is a matter of facts, with the Internet gluing together all the devices we own and paving the way for the communication society. The proliferation of connected devices impacts on the human footprint on the environment, with several reports indicating ICT as one of the major increasing causes of energy consumption [1], [2]. Even if not agreeing on the exact figures, all the estimations underline the importance of the problem, and on the urgency of reducing the environmental impact of the ICT sector. Continuously increasing energy costs are also pushing players to reduce the power consumption of ICT technologies. This surged a large effort to greening ICT technologies, and the Internet in particular [3], [4].

Considering backbone networks, power efficiency is a growing concern as core node capacity and density are constantly driven to new heights. Vendors are quickly offering new solutions in which each single chip, interface, line card and

node is designed to reduce its energy footprint. Overall, each element can operate at several “power-states”, corresponding to different switching and transmission capacities. That is, energy consumption of elements becomes a function of the offered load. As such, routing traffic demands such as to minimize the overall power consumption becomes an important part of network management.

Differently from previous works, in this paper we focus on “circuit switched networks”, e.g., modern networks in which the GMPLS-enabled backbone [5] offers dynamical setup of virtual circuits with Quality of Service (QoS) guarantees. In this scenario, customers generate *connection requests*, or “tunnels”, with a specific QoS requirement in terms of bandwidth and possibly delay. When processing a new request, the network control plane has to find a suitable route, and, if found, requested capacity is allocated on the data plane for the duration of the request. In case no path can be found, the connection is blocked. Traffic is thus specified in terms of circuits, that are born, stay active for some time, then are terminated. This kind of solutions are currently foreseen as best candidate for backbone networks to exploit the benefits of optical networks while reducing the complexity of the per-packet switching costs [6]. They enable cost-effective service delivery, and flexible traffic management.

The current GMPLS framework is considered to be the emerging control plane solution for future optical networks [6]. The main functionality that the GMPLS control plane offers in optical networks is the dynamic establishment and tear-down of end-to-end optical connectivity. GMPLS currently does not include any mechanism to take into account energy consumption parameters when identifying end-to-end paths or disseminating the status of network elements with respect to their power consumption. Actually, the standard operates in the direction to define an energy-aware control plane enabled by energy-aware routing algorithms and signalling as well as by specific energy optimisation mechanisms.

What is missing is the application of energy-aware routing algorithms to the GMPLS control plane suite. We propose a novel methodology which extends the Path Computation Element architecture [7] with the objective to minimise the overall energy consumption of inter-domain/intra-domain op-

tical networks, ensuring the agreed QoS.

To this extent, our work aims at investigating how to extend the previous finding to include energy efficiency in the circuit switching framework. We conduct extensive simulations by considering *real* scenarios, modelling an actual ISP topology and power consumption figures derived from actual network elements, therefore providing realistic experiments.

Results show that the proposed weight assignment enables to include power consumption efficiency at the control plane. Results show that up to 60% of energy consumption can be easily achieved without significantly affecting the network utilization. Furthermore, by setting weights in a proper and intuitive way it is possible to easily obtain a win-win situation in which both energy efficiency and network performance are maximised. This strengthens the proposed framework, making it a simple, intuitive and efficient proposal that can be easily and quickly implemented by extending the current GMPLS framework.

The paper is organised as follows: Section II describes the state of the art of Energy-Aware routing. Section III reports the description of the algorithm. Section IV details the realistic case studies considered for the algorithm evaluation. Section V presents the simulation results, and discusses the sensitivity of the results to the algorithm parameters. Section VI analyses hybrid routing policies. Finally, conclusions are drawn in Section VII.

II. RELATED WORKS

Routing based on minimum-cost path algorithms and leveraging on linear combinations of different routing weights has been proposed in the past. Yet, all proposals focused on the so called “QoS-routing”, i.e., looking for the route that maximises the QoS [8]. Energy consumption was not taken into account in any of the previous works to the best of our knowledge.

The only work that indirectly could consider energy-aware weight assignment is [9]. The authors propose a context-based path-selection algorithm for multi-radio systems, in which the routing weights of links are computed with the objective of minimising self interference, or maximising the network coding throughput. Network energy consumption is not explicitly taken into account, but the proposed framework is supposed to be extended also for other contexts. However, the resulting solutions are not intuitive.

In the context of pure packet switched networks, different green network management policies have been proposed. Solutions range from centralised omniscient algorithms [10], [11], to distributed on-line algorithms [12], [13]. We address the interested reader to [3], [4] for a complete overview. While centralised solutions suppose the perfect knowledge of all current traffic flows in the network, in distributed solutions, single network devices change their working state (e.g., by switching off some interfaces) and the traffic is routed on the network portion which remains active. In both cases, to the best of our knowledge, solutions to compute the set of link weights corresponding to the desired flow allocation are missing.

In general, centralised solutions considering *packet switched networks* analyse a “snapshot” of the network traffic and compute a proper flow allocation to reduce the network energy expenditure. A *circuit switched network* accounts, instead, for independent connection requests that asynchronously arrive at the network and need to be allocated on end-to-end circuits at their arrival. In this context it is hence possible to take into account only the current state of the network elements (e.g., their traffic load) in order to compute the best path to accommodate a new connection request, in order to minimise the network energy consumption and satisfy the QoS requirements. This difference reflects the different concept driving our solution with respect to the ones present in literature: proposed solutions for packet switched networks (e.g., [11]) try to periodically reroute the carried traffic to minimise energy consumption. This will result in the circuit switched network context in rerouting the allocated circuits at every new traffic request arrival, resulting in a non affordable complexity and lost of QoS. This is why we instead directly route new connection requests using the set of devices which express the best trade off among available capacity and energy consumption, naturally leaving devices idle when not needed. Note that previously allocated flows are not rerouted in any case.

III. ALGORITHM DESCRIPTION

A. Modelling Assumption

We model the communication network as a directed Graph $G(V, A)$. V is the set of vertexes, whose elements $i \in V$ represent network nodes, e.g., GMPLS routers. A is the set of arcs, whose elements $a = (i, j) \in A$ represent communication links, e.g., physical interfaces of given capacity.

We consider a circuit switching network, where a connection (or GMPLS tunnel) request r arrive at random time intervals, originated at a source node $s(r) \in V$, and destined to node $d(r) \in V$. Connections request a bandwidth amount $b(r)$, and hold for $\tau(r)$. The connection request may be rejected if there are not enough available resources in the network at the arrival time. If the connection request is accepted, it is routed over the *minimum cost* path.

Every link a is associated to a cost, i.e., a routing weight $w(a)$. Routing weights are defined on administrative bases, following the intuition that the higher the cost associated to a link, the less likely the link is chosen to route traffic. The cost $c(p_{s,d})$ of the path $p_{s,d}$ is defined as the sum of the routing weights assigned to the links being part of the path:

$$c(p_{s,d}) = \sum_{(i,j) \in p_{s,d}} w((i,j)) \quad (1)$$

A proper setting of the routing weights is the key instrument of network management to achieve better network performance, and to reduce the network power consumption. In the following we propose a dynamic routing weight assignment to choose the best path.

B. Link Weight Model

While considering a connection request, the routing weight $w(a)$ is associated to each link a on the basis of the following equation:

$$w(a) = A\rho(a) + B\Delta P(a) + C \quad (2)$$

where: $\rho(a)$ is the load on link a , $\Delta P(a)$ is the increase in the power consumption caused by the eventual routing of the request on link a . A, B, C are non-negative multipliers used to tune and define the desired kind of routing¹. In particular, the 3 following special cases are of relevant interest:

- $A = 1, B = C = 0$: **Least Loaded Path (LLP)** routing, i.e., a routing aiming at selecting the minimum congested path;
- $A = 0, B = 1, C = 0$: **Pure Green Path (PGP)** routing, i.e., a routing aiming at selecting the lowest power consuming path;
- $A = 0; B = 0, C = 1$: the classic **Shortest Path (ShP)** routing.

Finally, **Hybrid** schemes can be generated by tuning A, B, C to trade among energy saving and network utilization.

C. Modelling Nodes and Linecards

To take into account node energy consumption and load into the path computation, we perform a transformation of $G(V, A)$ as follows. We explode each node as in Fig. 1 by introducing two virtual nodes, i' and i'' , connected by a virtual link $v = (i', i'')$. The new link v has load $\rho(v)$ and power consumption $P(v)$ profiles equal to the one of the original node (i.e., i). This procedure allows node characteristics to be naturally accounted in the Dijkstra's algorithm, when link v is added in the path.

Formally, we consider hence for the path selection procedure an edited graph $G'(V', A')$. For each node $i \in V$, the set V' is including the two nodes i' and i'' . For each arc $a = (i, j) \in A$, the set A' is including the arc (i'', j') . Finally, for each node $i \in V$, the set A' is including the arc (i', i'') . As a consequence, a connection request from node s to node d in G , corresponds to a connection request from node s' to node d'' in G' . The cost of the path is then simply computed as in Eq. (1).

This process can be used to include the cost of other elements by transforming them into virtual links and nodes. For example, "virtual" links can be added between a node linecard and the corresponding node, to account for the linecard load level, and power consumption. This procedure may be needed if, e.g., a node is equipped with multiple linecards, that aggregates several physical ports.

¹ A and B are pure numbers, while C has the physical quantity of $[Power]^{-1}$. As $\rho(a)$ is a rate in percentage and $\Delta P(a)$ a Power, $w(a)$ results in a pure number. The multipliers are also responsible for the scaling of the equation elements, as $\rho(a) \in [0, 100]$, while $\Delta P(a)$ may have a much wider dynamics, and C is usually set to 1 for hop count.

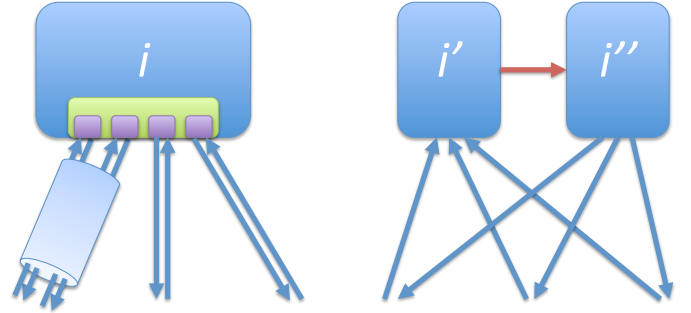


Fig. 1: Node duplication to account for network nodes in the path computation procedure: the real network devices mapped in the G graph (on the left) and their representation in the G' graph (on the right).

D. Routing New Connections

When a new connection request has to be routed, the network management procedures computes the weights for each element of the expanded graph G' . The traditional minimum cost path is computed, e.g., by running the Dijkstra's algorithm. Ties are broken by selecting one path at random among the minimum cost paths. A standard culling pre-processing is run so that those elements that do not have enough capacity to accommodate the request are given an infinite weight to exclude them from the calculation.

When considering the Pure Green Path routing, $w(a) = \Delta P(a)$. The path computation corresponds to an execution of the Dijkstra's algorithm in which the non-negative weights correspond to the increase in the network power consumption in the case in which the selected request is routed through the corresponding network device. Since the Dijkstra's algorithm returns the path corresponding to the lowest sum of routing weights, the resulting path will be the one corresponding to the overall lowest possible increase in the network power consumption². Therefore, the Pure Green Path routing results to be the optimal choice for the routing of the new request³.

IV. SCENARIO DESCRIPTION

Performance of the described algorithm has been evaluated considering a network scenario representing the real infrastructure of a national ISP in Italy. The considered network topology follows a typical hierarchical design, as reported in Fig. 2, in which five levels of nodes are present: Internet peering point, core, backbone, metro, and access nodes.

The core level is composed by few nodes densely interconnected by high-capacity links, and offering connectivity to the Internet by means of peering nodes (labeled as "Internet" in Fig. 2). Going down in the hierarchical levels, the number of nodes increases, while the link capacity decreases. Access nodes represent sources and destinations of traffic,

²We assume $P(a)$ to be an increasing function of a - see Sec. IV

³Note that the selected path is optimal under the current conditions and may no longer be optimal when other traffic requests are allocated, or previously allocated traffic requests are terminated.

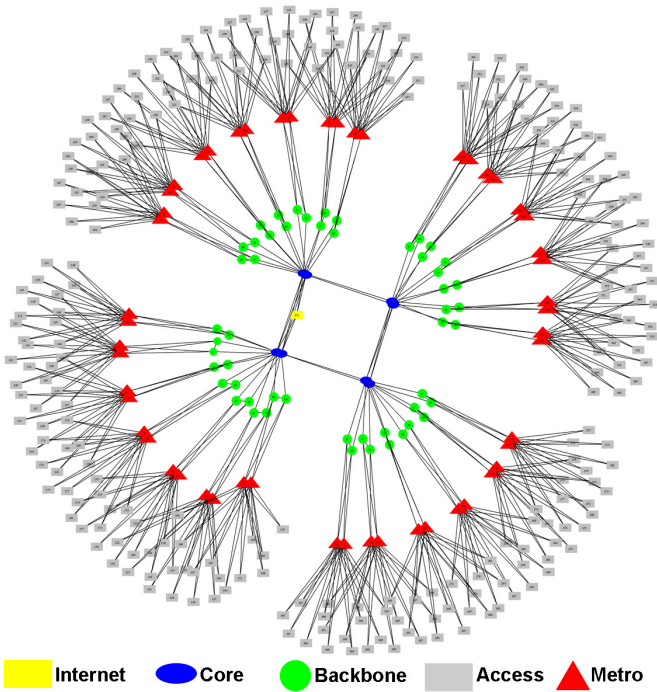


Fig. 2: Representation of the ISP network topology.

offering those nodes connectivity to end customers. Metro nodes aggregate the traffic coming from different areas in large cities or towns. They are assumed to be edge between the GMPLS enabled backbone, and the traditional packet switching technologies offered by access nodes. Metro nodes are interconnected to the core nodes by means of high capacity backbone nodes. Optical links are used, with different capacities depending on which nodes they interconnect. We assume all but access nodes support GMPLS switching, with the core and backbone nodes supporting direct optical switching as offered by optical cross connects technologies.

Links between access and metro nodes and between metro and backbone nodes are assumed to be up to 10 Gb/s. Links between backbone and core nodes are assumed to be up to 40 Gb/s. Finally, links between core nodes are assumed to be up to 100 Gb/s. Further details on the network scenario are described in [11] in which however, the routers are assumed to be packet switched nodes, i.e., IP routers.

A. Traffic Model

For simplicity, we considered in our simulations only the GMPLS-enabled four higher hierarchical levels of the considered network, not accounting for the access nodes. In the resulting network scenario, the metro nodes and the Internet peering point are the only sources and destinations of GMPLS tunnels.

Connection requests are generated considering a Poisson process, with mean arrival rate equal to λ and duration exponentially distributed with a mean of T s. Tunnels requests are associated to a source and a destination node, uniformly chosen among the *metro* node set and the *Internet peering point*.

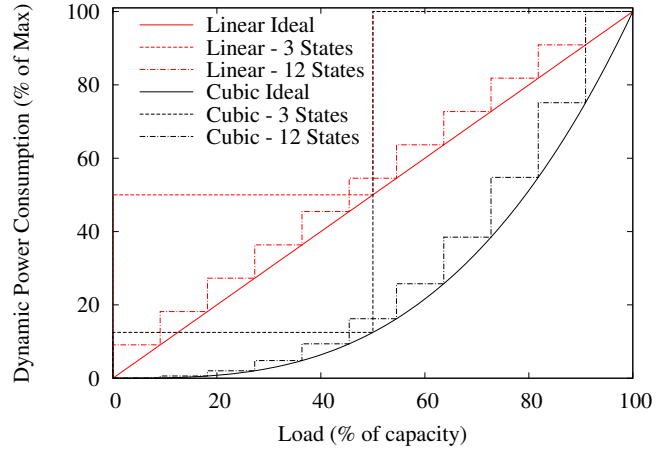


Fig. 3: Dynamic power consumption models considering different status distributions. Red curves refers to linear scaling, black curves to cubic scaling.

This simple traffic matrix reflects the symmetry properties of the topology, which has been designed to balance traffic among metro nodes. Each connection is characterised by a requested capacity that we assume to be uniformly distributed between b_{min} and b_{max} . In the following, we assume $b_{min} = 0.5Gb/s$, $b_{max} = 1.5Gb/s$, $T = 100$ hours. λ results then the only free parameter that defines the offered load to the the network. Tunnel requests for which it is not possible to find a path which guarantees the requested capacity result blocked.

B. Static Power Model

To evaluate the network power expenditure, we model the network devices considering real node and component power consumption figures, considering measurement results from Alcatel-Lucent. In particular, we consider the switching capacity of GMPLS routers as described in [11]. Each node is built by aggregating those network components listed in Tab. I so that the correct number of components are selected to match node desired peak capacity. Each node requires fan, power supply, Equipment Controller (EC)⁴, Shelf Controller (SC)⁵, switching fabric (matrix), and the number of linecards and interfaces to match the desired capacity. Static and dynamic power consumption of each node will thus change based on its configuration. Power consumption figures correspond to actual values measured experimentally on Alcatel 1850 TSS-160C Transport Service Switch.

The node power consumption accounts for all the GMPLS router common components, up to the linecard. The link power consumption accounts for the interfaces and includes optical transeiver costs. For instance, a network node equipped with

⁴The EC provides the hardware resources (physical interfaces) and the SW functionality (protocol stack) required for the communication between Network Element and Management system (OS, craft terminal, etc).

⁵The SC provide the resources to support the SW functions related to the control and management operation of the boards. As the SC is involved in critical activities (for instance EPS), it is 1+1 protected.

TABLE I: Power consumption figures for different network components.

Component	Description	Bandwidth (Gb/s)	Static Power (W)	Dynamic Power (W)
NODE	TSS160C			
Common Part	FAN	320.0	30.0	0.0
Common Part	Supply	320.0	50.0	0.0
Common Part	EC	320.0	20.0	0.0
Common Part	SC	320.0	18.0	0.0
Common Part	Matrix 1	320.0	100.0	50.0
Common Part	Matrix 2	320.0	100.0	50.0
Line Card 1	2x10 Gb/s	20.0	30.0	20.0
Interface	SFP 1x10 Gb/s	10.0	20.0	10.0
Line Card 2	10x1 Gb/s	10.0	30.0	20.0
Interface	SFP 1x1 Gb/s	1.0	5.0	5.0

a single linecard with two 10 Gb/s interfaces results in a static power consumption of 348 W, and a dynamic power consumption of 120 W. The two links, once active, have a static and dynamic power cost of 20 W and 10 W, respectively.

C. Dynamic Power Model

We consider network devices consuming a static power share - as computed above - as soon as they are switched on. The dynamic power consumption share is assumed proportional to the offered load level. As described in [14], the dynamic power consumption share is split into “steps”, accounting for different possible working states. Due to physical constraint, only a limited number of possible power-capacity states can be implemented in actual devices so that a step function relates device load to device power consumption, as sketched in Fig. 3. We assume that up to 12 working states can be defined, as described in [14]. An interesting question is the usefulness of 12 power states, e.g., what if only 6 or 2 power states are offered? In our power model, we thus consider a variable number of working states, ranging from 2 to 12 to answer the previous question.

Two different power vs. load curves are considered: a “linear” function, in which all steps are equal, representative of current devices, and a “cubic” function, in which state power consumption is following a cubic profile, as representative of network technologies adopting Dynamic Voltage Scaling (DVS)-like solutions. An example for the two different distributions, considering different number of states, is reported in Fig. 3. For instance, the “3 states” curves are characterised by the presence of 3 working states: an *idle state*, characterised by 0 dynamic power consumption - 0 capacity; a *low power state*, characterised by a dynamic power consumption lower than the maximum (i.e., 50% assuming linear scaling function, and 12.5% assuming cubic scaling function), but able to carry an amount of traffic smaller or equal to 50% of the device capacity; and a *full working state*, characterised by the maximum power consumption, and able to carry an amount of traffic smaller or equal to the device capacity.

In the following, we compare linear and cubic power-capacity functions to allow to investigate which are the benefits and impact of different technological constraints, e.g., the presence of DVS technology.

TABLE II: Default values of the simulation parameters.

Parameter	Symbol	Value
Number of Nodes	$ V $	113
Number of Links	$ A $	396
Connection request arrival rate	λ	1 hour
Average call duration	T	[10:600] hours
Connection bandwidth interval	$b_{min} - b_{max}$	0.5-1.5 Gbps

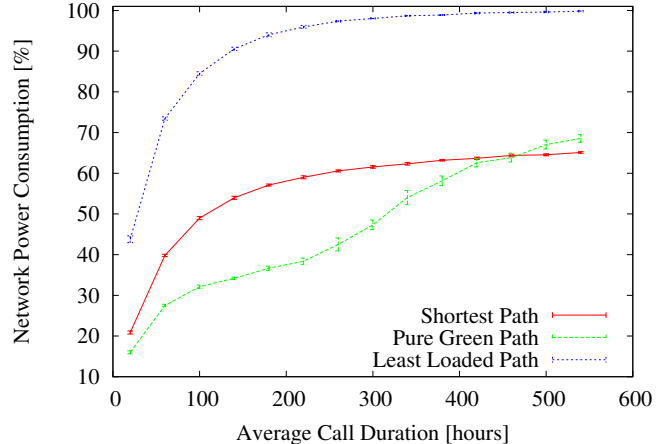


Fig. 4: Network power consumption, considering different weight assignment policies.

V. PERFORMANCE EVALUATION

The proposed algorithms have been implemented in a custom event-based simulator, written in C language. The simulator code and input description are available upon request. Events correspond to tunnel request arrivals and teardown.

Each simulation corresponds to a 100 days period of simulated time (corresponding to about 2400 connection requests per node on average), over which results are averaged, after excluding the warm-up period of 10 days. For each value, 10 simulations have been executed with different seeds. We report the average values among the 10 runs, together with the corresponding 95% confidence interval. Table II recaps the main scenario characteristics and the default values of the simulation parameters.

As a first step, we evaluate the total network power consumption when considering the three pure weight assignment policies, i.e., Least Loaded Path, Pure Green Path, and Shortest Path. Fig. 4 reports the corresponding network power consumption, normalised with respect to the maximum power consumption reached by any scenario, versus average call duration, i.e., for increasing offered load. The power consumption model uses 12 different states, and power consumption of devices grows linearly with their offered capacity. In order to evaluate the network performance, we report the percentage of blocked calls for each of the three weight assignment policies. Results are reported in Fig. 5.

As expected, both the total power consumption and the blocking probability are monotonically increasing with network offered load. The Least Loaded Path performs poorly in

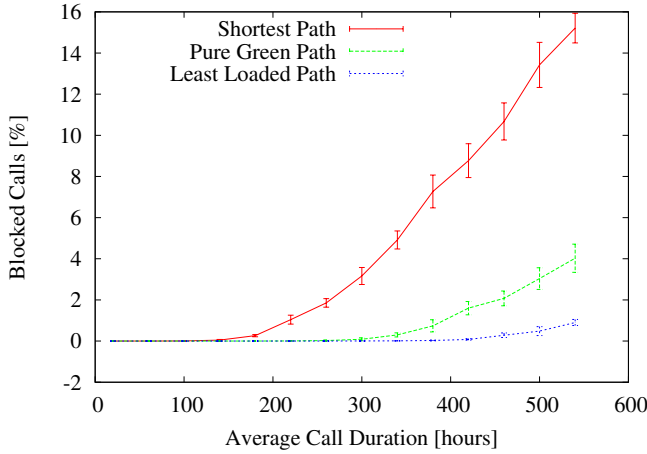


Fig. 5: Blocking probability considering different weight assignment policies.

terms of energy saving since it tends to spread tunnel requests on all possible paths thus quickly turning on all nodes and links. This results in a steep increase of the power consumption figure even at light load, caused mostly by the static power consumed by devices when turned on. The power consumption then flattens, with small increases due to load increase on devices which causes the dynamic power consumption figure to grow proportionally.

Notably, the Pure Green Path policy is able to achieve the lowest power consumption among the three policies, saving between the 30% and the 60% of power with respect to the most power-hungry Least Loaded Path policy. This is thanks to its preference to route new requests on possibly longer paths in which capacity is still available prior to turn on new elements. Only in case the capacity of already active elements does not allow to accommodate new requests, the Pure Green Path policy starts activating new paths and devices. This causes the energy consumption curve to face a second transient phase with steep growth (since the static power is higher than the dynamic power). This is reflected in the second part of the curve, for average call duration higher than 200 hours.

The Shortest Path policy power consumption curve follows a similar trend to the Least Loaded Path one, even if the power consumption is much smaller. In this case, the Shortest Path policy ignores both load and energy state of links, thus blindly taking always the same paths, ignoring possible longer but less congested or expensive paths. This is confirmed by the blocking probability curve in Fig. 5. It shows that the Shortest Path policy starts blocking new connection for values of the offered load much smaller than the one at which the Pure Green and Least Loaded routing starts blocking calls.

To support this intuition, we compare the network performance considering the load distribution on links. Results are reported in Fig. 6 considering an average call duration of $T = 100$ hours, for which call blocking probability is negligible for all algorithms. For each policy, Fig. 6 reports the average load of links, sorted by increasing load. Notice how

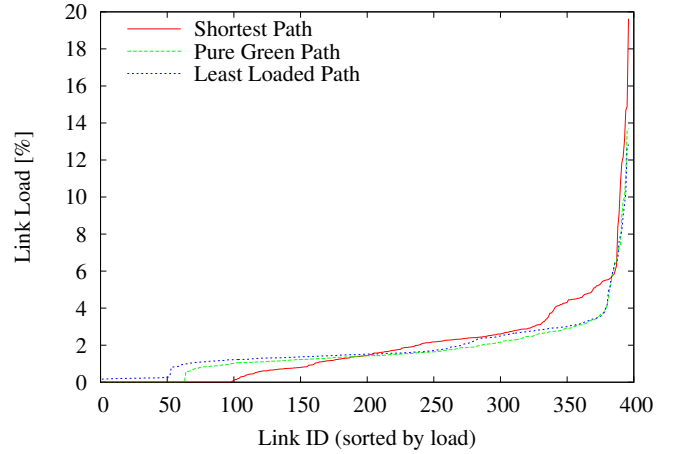


Fig. 6: Average link load distribution for $T = 100$ hours.

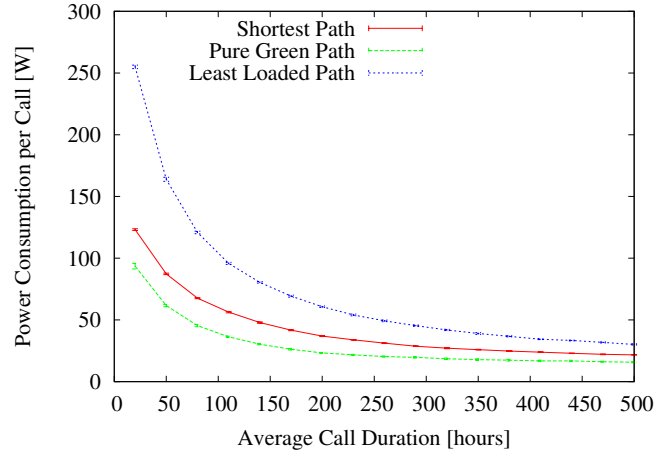


Fig. 7: Power consumption per connection, considering different weight assignment policies.

the Shortest Path concentrates the load on few bottleneck links which results thus congested and cause connection blocking. Some links are never used. On the contrary, the Least Loaded Path tends to spread traffic more evenly on more links. This at the cost of turning on all links. The Pure Green Path routing is carefully choosing those links that, being already powered on, allow to accommodate the requests without the need to power on other resources.

At last, to fairly compare the power consumption achievable by the three policies at the net of the blocked calls, we evaluate the power consumption per call, i.e., the total network power consumption over the total number of accepted calls. Results are reported in Fig. 7, and show that also in this case the Pure Green Path policy is able to guarantee the lowest possible power consumption among the three. Note how the saving per call is very significant for low values of offered load. Given future GMPLS optical networks are expected to work under light load, this result supports for the choice of Pure Green Path routing.

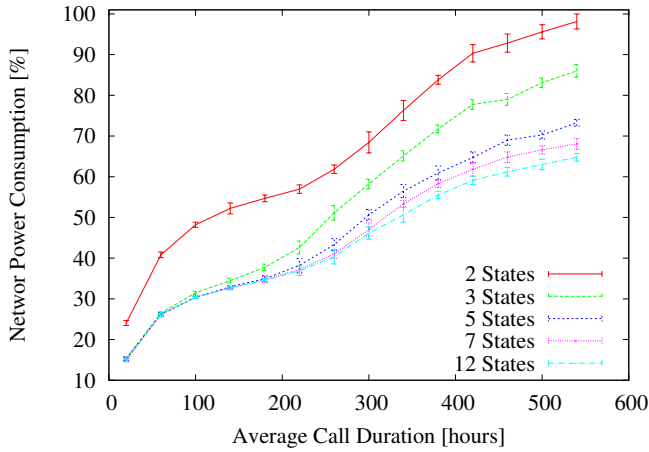


Fig. 8: Impact on the network power consumption of the variation of the number of power states, considering the cubic state distribution.

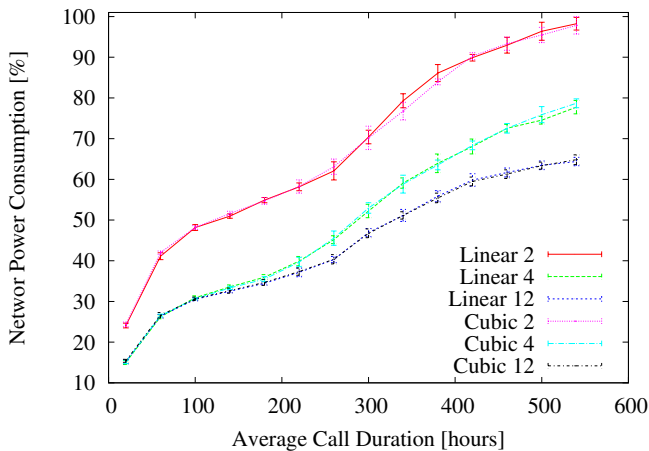


Fig. 9: Impact on the network power consumption of the variation of the number of power states, and their distribution.

A. Impact of Variation of the Number and Distribution of Power States

We evaluate the sensitivity of the results to the variation of the number of power states, and to their distribution (i.e., linear or cubic function of the offered capacity). This analysis aims at better understanding which direction should the constructor take in the design of network components, especially from the point of view of the number of power states to be included, and their distribution (i.e., the scaling technology to be adopted). We consider only the Pure Green Path routing here.

As a first step, we evaluated the power consumption resulting from the application of the Pure Green Path policy to devices with a variable number of power states. We assume the energy cost grows cubically. Results are reported in Fig. 8, for different average call duration. Power consumption is normalized as before with respect to the maximum power consumption among all scenarios. Fig. 9 compares the power

consumption resulting when considering linear or cubic distribution of power states with respect to the offered capacity. Also in this case, power consumption values are normalised with respect to the maximum computed power consumption for ease of comparison.

Results show that i) an increasing number of states is beneficial for the reduction of the network power consumption, but after a certain granularity, ii) the achievable reduction of the extra power saving strongly decreases. For instance, the gain reached with only 5 states is practically identical to the one guaranteed by adopting a finer grade adaptability. Similar findings have been presented in [15].

Considering the linear or cubic distribution of power states with respect to the offered capacity, a minor impact is observed, independently from the number of power states (Fig. 9). This is due to the limited impact that the dynamic power consumption has compared to the static power cost that has to be faced when turning on the device.

B. Effects on the Average Path length

We evaluate the weighted average path length resulting from each policy, as an indicator of the end-to-end delay. We weight the length of the paths by the amount of traffic they are carrying, as discussed in [16]. Being R the set of all the traffic requests in the considered period, and $L(p_{s,d})$ the length of the path $p_{s,d}$, in number of hops, the weighted average path length results as:

$$\bar{L} = \frac{\sum_{r \in R} b(r) L(p_{s(r),d(r)})}{\sum_{r \in R} b(r)} \quad (3)$$

Results for increasing average call duration are reported in Fig. 10. 12 power states following a cubic distribution are considered here. Consider first the Pure Green Path policy. Results show that, after network load of about $T = 200$ hours, the residual capacity becomes a scarce resource and the routing reacts activating longer paths. A similar behaviour is shown also by the ShP routing policy, as it starts filling the shortest paths, and only after they are saturated it starts looking for longer alternatives. The effect is mitigated in this case by the increasing number of blocked calls, and by the fact that minimising the path length is the primary goal of this routing policy. On the other hand, the LLP policy starts initially by spreading the traffic among all the available paths, including very long ones, while, with increasing load, the amount of tunnels routed on long paths becomes negligible, as they are less likely to be available. The average path length for the LLP policy becomes even shorter than for the other policies after a given network load. This is because - for high network loads - the calls suffer blocking with the other policies. Fig. 11 reports results for varying number of power states, confirming that this metric is independent from the number of considered power states.

VI. HYBRID ROUTING

Eq. (2) allows to easily build possible hybrid policies. In particular, it is interesting to observe which are the trade-

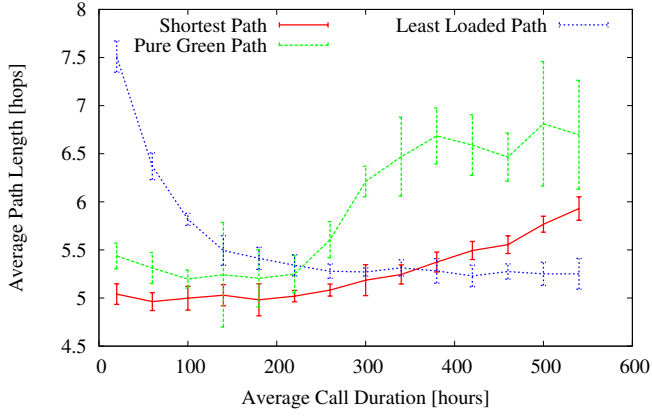


Fig. 10: Average path length considering different weight assignment policies, for increasing average call duration.

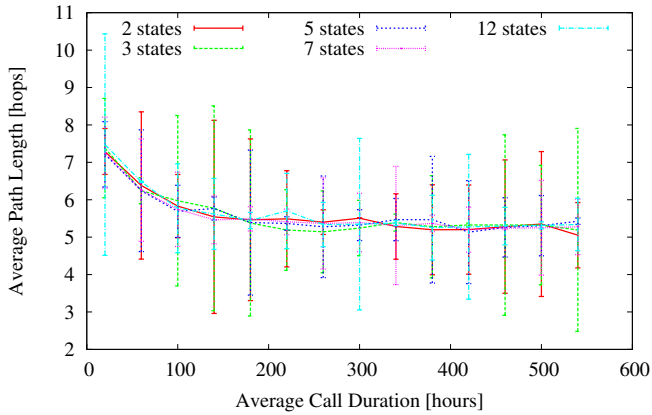


Fig. 11: Impact of the number of power states on the Average Path Length, for increasing average call duration.

offs among energy efficiency, and blocking probability, i.e., between Pure Green and Least Loaded routing. We considered $C = 0$, as the Shortest Path policy has shown worst performance. This allows us to simplify the evaluation by fixing the parameter $B = 1$, and by evaluating the performance when varying the value of the parameter A .

Intuitively, when the value of A is low, the Pure Green Path behaviour is predominant, as $A\rho(a) \ll \Delta P(a)$; for big values of A , the Least Loaded Path behaviour becomes predominant, as $A\rho(a) \gg \Delta P(a)$. For intermediate values, we expect hybrid schemas to be generated. Fig. 12 reports the normalised power consumption for growing values of the parameter A . The power consumption model uses 12 different states, and power consumption of devices grows linearly with their offered capacity. An average Call Duration of 500 hours is considered here. Results reflect the expected behaviour, with network power consumption growing for increasing values of the A parameter. Notice that large values of A are needed to allow the load component to become the predominant part of the weight in Eq. (2).

For the same scenarios, Fig. 13 reports the variation of the

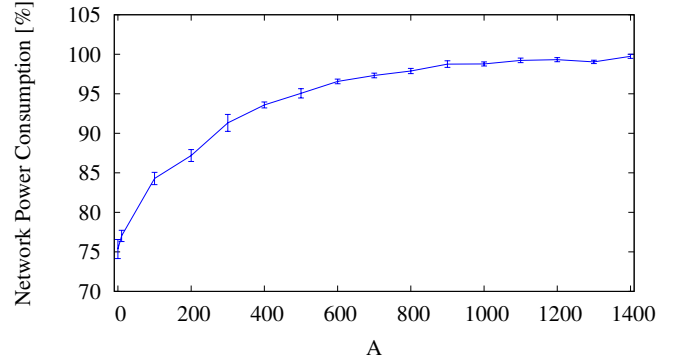


Fig. 12: Impact on the network power consumption of the variation of the parameter A .

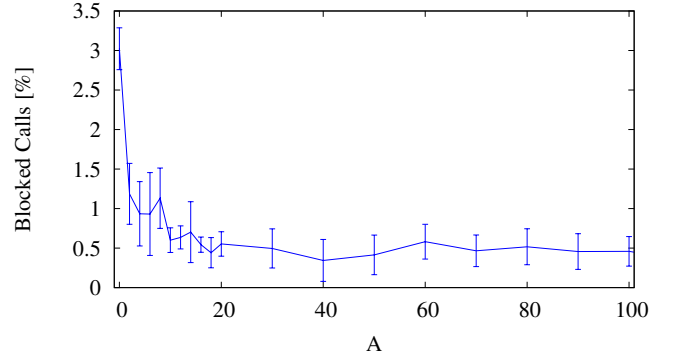


Fig. 13: Impact on the call blocking probability of the variation of the parameter A .

blocking probability. It shows a rapid dropping of the rate of blocked calls for increasing values of the A parameter. Blocking probability stabilises to the value corresponding to the Least Load Path routing, for $A > 10$ (see Fig. 5). This results in the routing algorithm to prefer avoiding increasing power consumption and then choosing the least congested path among the minimum extra energy cost paths.

To verify this statement, Fig. 14 reports the normalised power consumption for different network loads and for different Hybrid algorithms. As we can see, curves corresponding to increasing values of the parameter A gradually approach the curve corresponding to the Least Loaded Path policy (reported for comparison purpose in Fig. 14, by a solid blue line), starting from the Pure Green Path policy (see also Fig. 4). Notice the marginal extra energy consumed when selecting $A = 10$, a case in which the blocking probability is already minimised.

We can hence gauge the trade off corresponding to a proper tuning of the parameters of Eq. (2), i.e., $A = 10, B = 1, C = 0$. For the considered network scenario, the Hybrid algorithm is able to provide the same saving guaranteed by the Pure Green Path policy, on the one hand, and the same minimum blocking probability guaranteed by the Least Loaded Path routing, on the other hand. This allows to obtain a win-

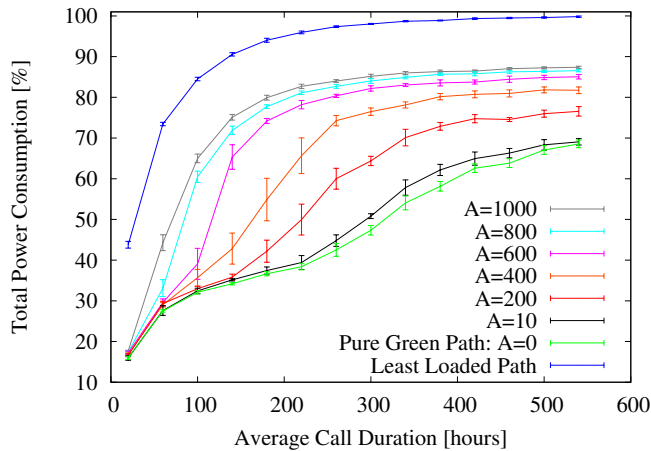


Fig. 14: Impact on the network power consumption of the variation of the parameter A , for different average call duration.

win situation in which both energy efficiency and network utilization are maximised. Note that the selected parameter tuning, i.e., $A = 10, B = 1, C = 0$, behaves exactly as the Pure Green Path policy for low loads (i.e., average call duration $T \leq 100$ hours), while gradually trades small amounts of energy to achieve the low blocking probability of the Least Loaded Path routing, for increasing network loads.

VII. CONCLUSIONS

In this paper, for the first time to the best of our knowledge, we considered the problem of the design of energy efficient network management algorithms considering the circuit oriented paradigm as offered by GMPLS backbone topology, in which tunnel requests can be routed on the minimum cost path. We designed a simple and novel framework which allows the definition of a family of routing algorithms, based on the definition of a cost function that weights traffic load, energy cost and path length. Hybrid routing policies can be easily defined; they allow trading between power consumption, on the one hand, and network performance, on the other hand. Some ingenuity is required to include the node cost so that their energy consumption can be taken into account during the minim cost path computation.

The defined routing policies have been evaluated over a realistic network scenario, considering different power models reflecting different technological scenarios. Thorough performance evaluation shows that the proposed framework allows to considerably reduce the power consumption of GMPLS backbone networks, without practically affecting the network performance, and to easily tune the achievable energy saving as a function of the required network performance.

We are currently working in the implementation of the proposed framework in the GMPLS testbed available at Alcatel-Lucent laboratories.

ACKNOWLEDGMENTS

This work has been supported by the FP7 Integrated Project “Econet - low Energy Consumption NETWORKs” funded by the

European Commission, and by the Swedish national Graduate school in computer science (CUGS).

REFERENCES

- [1] Global Action Plan. An Inefficient Truth. Global Action Plan Report, <http://globalactionplan.org.uk>, December 2007.
- [2] Molly Webb. SMART 2020: Enabling the Low Carbon Economy in the Information Age. The Climate Group. London, June 2008.
- [3] A.P. Bianzino, C. Chaudet, D. Rossi, and J.L. Rougier. A Survey of Green Networking Research. *IEEE Communication Surveys and Tutorials*, 14(1):3–20, 2012.
- [4] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Communication Surveys and Tutorials*, 13(2):223–244, 2011.
- [5] E. Mannie. Generalized Multi-Protocol Label Switching (GMPLS) Architecture. IETF RFC 3945, <http://www.ietf.org/rfc/rfc3945.txt>, October 2004.
- [6] S. Azodolmolky, R. Nejabati, E. Escalona, R. Jayakumar, N. Efstathiou, and D. Simeonidou. Integrated OpenFlow-GMPLS control plane: an overlay model for software defined packet over optical networks. *Optics Express*, 19(26):B421–B428, 2011.
- [7] A. Farrel, J.-P. Vasseur, and J. Ash. A Path Computation Element (PCE)-Based Architecture. IETF RFC 4655, <http://www.ietf.org/rfc/rfc4655.txt>, August 2006.
- [8] J.M. Jaffe. Algorithms for finding paths with multiple constraints. *Networks*, 14(1):95–116, 1984.
- [9] S. Das, Y. Wu, R. Chandra, and Y.C. Hu. Context-based Routing: Technique, Applications and Experience. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 379–392, San Francisco, USA, April 2008.
- [10] N. Vasić and D. Kostić. Energy-aware traffic engineering. In *1st International Conference on Energy-Efficient Computing and Networking (e-Energy 2010)*, Passau, Germany, April 2010.
- [11] L. Chiaraviglio, M. Mellia, and F. Neri. Minimizing ISP Network Energy Cost: Formulation and Solutions. *IEEE/ACM Transactions on Networking*, ISSN: 1063-6692, 2011.
- [12] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini. An Energy Saving Routing Algorithm for a Green OSPF Protocol. In *IEEE INFOCOM Workshop, 2010*, San Diego, USA, March 2010.
- [13] A.P. Bianzino, L. Chiaraviglio, M. Mellia, and J.L. Rougier. GRiDA: Green Distributed Algorithm for Energy-Efficient IP Backbone Networks. *Computer Networks*, 53(14):3219–3232, September 2012.
- [14] Low Energy Consumption NETWORKs (ECONET) project. Deliverable 4.1 - Definition of Energy-Aware States. <https://www.econet-project.eu/Deliverables>, September 2011.
- [15] Edoardo Bonetto, Marco Mellia, and Michela Meo. Energy Profiling of ISP Points of Presence. In *IEEE ICC’12 Workshop on Green Communications and Networking*, Ottawa, CA, June 2012.
- [16] A.P. Bianzino, C. Chaudet, S. Moretti, D. Rossi, and J.L. Rougier. The Green-Game: Striking a Balance between QoS and Energy Saving. In *23rd International Teletraffic Congress (ITC 2011)*, San Francisco, USA, September 2011.