

Enabling Backbone Networks to Sleep

Raffaele Bolla, Roberto Bruschi, Antonio Cianfrani, and Marco Listanti

Abstract—Today, backbone networks of Telecom operators deploy a large number of devices and links. This is mainly due to both redundancy purposes for network service reliability, and resource over-dimensioning for maintaining quality of service during rush hours. Unfortunately, current network devices do not have power management primitives, and have constant energy consumption independently of their actual workloads. Starting from these considerations, we propose a viable approach to introduce and to support standby modes in backbone network devices. This approach can be effectively used to almost halve the energy requirements of the whole Telecom core network. Our main idea consists of periodically reconfiguring nodes and links to meet incoming traffic volumes and operational constraints of real-world networks, such as reliability, stability, quality of service, and re-convergence times. To this purpose, the approach we propose directly exploits the main features of both backbone device architectures, as well as the network protocol stack.

I. INTRODUCTION

RECENTLY, Telecom Operators (Telcos), Internet providers have raised their interest in energy efficiency for wired networks and service infrastructures, making it a high priority objective [1]. This interest is motivated by the increase in energy prices, the continuing growth of customer population, the spreading of broadband access, and the expanding offer of services. Indeed, the increase in the volume of data traffic follows Moore’s law, by doubling every 18 months, and, by operating in a business-as-usual scenario in the next future, network device scalability might be constrained by energy consumption aspects. In this respect, Baliga *et al.* [2] stated that today’s networks rely very strongly on electronics, despite the great progresses of optics in transmission and switching, and outlined how energy consumption of the network equipment is a key factor of growing importance. In this sense, they suggested that the ultimate capacity of the Internet might eventually be constrained by energy density limitations and associated heat dissipation considerations, rather than by the bandwidth of the physical components.

To support new generation network infrastructures and related services for a rapidly growing customer population,

R. Bolla is with the Department of Communication, Computer and Systems Science (DIST), University of Genoa, Italy (e-mail: raffaele.bolla@unige.it).

R. Bruschi is with the National Inter-University Consortium for Telecommunications (CNIT), Italy. (phone: +39-010-3532057; fax: +39-010-3532154; e-mail: roberto.bruschi@cnit.it)

A. Cianfrani and M. Listanti are with the DIET Department, University of Rome “Sapienza”, via Eudossiana 18, 00184 Roma, Italy (e-mail: antonio.cianfrani@uniroma1.it, listanti@infocom.uniroma1.it).

providers need an ever-larger number of devices, with sophisticated architectures able to perform increasingly complex operations in a scalable way. For instance, high-end routers are increasingly based on complex multi-rack architectures; historic data from manufacturers’ datasheets show continuously raising capacities, by a factor of 2.5 every 18 months. However, silicon technologies improve their energy efficiency at a slower pace following Dennard’s law (namely, by a factor 1.65 every 18 months) with respect to routers’ capacities and traffic volumes [3].

Moreover, it is well known that network links and devices are provisioned for busy or rush hour load, which typically exceeds their average utilization by a wide margin. While this margin is seldom reached, nevertheless the overall power consumption in today’s networks is determined by it and remains more or less constant even in the presence of fluctuating traffic loads [4]. Today’s backbone networks are specifically designed to be extremely over-dimensioned in terms of switching capacity and of number of deployed links and nodes. Their switching capacity is usually larger than twice rush-hour traffic volumes in order to guarantee zero-loss and minimum latency packet forwarding. Moreover, links and nodes are often deployed in fully redundant way to meet network reliability constraints [5].

It is a common opinion that the sole introduction of novel low-consumption silicon technologies cannot effectively cope with such trends, and be sufficient for drawing ahead current network equipment towards a greener Future Internet. But the above-depicted situation suggests the possibility of adapting network energy requirements to the actual traffic profiles .

Advanced features for power management are already available in the largest part of hardware technologies, today used for building network devices. Silicon of network interfaces and of device internal chips already has the possibility of entering standby modes, or of scaling its working speed, and consequently of lowering their energy requirements. However, their activation is generally hindered by network protocols and architectures themselves, since they are specifically designed to be always available at the maximum speeds.

Emerging research approaches to network control, routing and traffic engineering [6] [7] aim at dynamically turning network portions off during light utilization periods, in order to minimize the energy requirements, while meeting the operational constraints and current switching workloads.

For instance, elements in standby (e.g., links or nodes) do literally “fall off” the network, since they are not able to exchange protocol signaling messages to maintain their “network presence” [8]. In core network scenarios, given the

features of routing and traffic engineering protocols, the falling off of any elements generally triggers all network nodes to exchange signaling traffic, and to re-converge towards new network logical topologies and/or configurations, causing transitory network instabilities and signaling traffic storms.

Starting from these considerations, we propose a viable approach to introduce standby primitives into next-generation devices, and to smartly support them in order to meet network operational and performance constraints. We exploit two features already and largely present in today's networks and devices: the network resource virtualization and the modular architecture of nodes. These features give us the opportunity of using the same base concepts already applied in other fields (e.g., data-centers): decoupling physical elements (e.g., a line-card), which may be put in standby, from their (virtual) functionalities and resources, so that the latter can be migrated towards other active physical elements of the same device.

In a different scenario and with other aims, the idea of virtual router migration was already investigated in [9]. However, in such work the authors suggested the migration of the entire router entity (i.e., its control and data planes among remote physical platforms. On the contrary, our approach aims at maintaining router entities bound to physical platforms: in this way we can directly control physical nodes, and avoid them to fall off the network.

The paper is organized as follows. Section II introduces the reference backbone network scenario. Section III discusses how hardware standby primitives can be introduced into network devices. The approach to support them while meeting network operational constraints is described in section IV. Section V argues how traffic engineering can be applied to coordinate standby capable nodes in order to reduce the overall network energy consumption. Performance evaluation results on the viability and the potential impact of the proposed approach are in section VI. Finally, the conclusions are drawn in section VII.

II. THE NETWORK SCENARIO

We considered a network scenario similar to the state-of-the-art backbone networks deployed by Telcos, where IP nodes have highly modular architectures, and work with a three-layer protocol stack.

In more detail, we consider an IP network (L3) overlaid over a Wavelength Division Multiplexing (WDM) optical network (L1). A Layer 2 (L2) protocol (e.g., the Multi-Protocol Label Switch (MPLS) or the Ethernet protocols) is used to optimally map IP traffic on the physical infrastructure, and to implement value-added network features and services (e.g., Quality of Service (QoS), virtual private networks, mechanisms for fast fault recovery, etc.).

In such environment, physical channels carry multiple "virtual" L2 links (e.g., Label Switching Path (LSP) for MPLS or a Virtual LAN (VLAN) for Ethernet), which directly connect two or more nodes working at L3. Then, each LSP and/or VLAN constitutes a different link at the IP layer.

The path of L2 links on the physical topology is usually

determined by using a constrained-based routing algorithm taking into account physical capacity and QoS features. To this purpose, classical IP routing protocols, such as Open Shortest Path First (OSPF), are used within Traffic Engineering (TE) extensions. Moreover a control protocol, such as Generalized-MPLS (GMPLS), is required to dynamically manage L2 virtual topology.

Regarding devices, we focus on high-end network routers with modular architectures, composed by a switching matrix and multiple line-cards. Every line-card has one or more Physical interfaces (PHY), and is assumed to include full packet processing capabilities at L2 and L3. As shown in Figure 1.a, each line-card includes multiple PHYs, each one carrying a number of L2 Virtual Links (L2VL). L2VLs are terminated on the line-card itself through virtual network interfaces, called L2 Terminations (L2Ts), which, by definition, are also the network interfaces at layer 3. Thus, IP links are realized by means of L2Ts on two or more nodes.

III. STANDBY PRIMITIVES FOR NETWORK DEVICES

Current network devices do not include sleeping/standby capabilities. However, these capabilities are key features of general purpose hardware across all market segments.

Sleeping/standby primitives are founded on power management mechanisms that allow hardware modules in a device freezing their operations, while maintaining their "context" information (e.g., configurations, running tasks, etc.). When sleeping, hardware elements have very low energy requirements: energy is substantially needed only to refresh memory for maintaining context data, and to optionally leave some sub-modules (e.g., a network interface) powered on, awaiting for external wake up messages.

Current sleeping technologies in general purpose systems allow entire PCs and laptops entering and waking up from standby states in time periods shorter than 2 s. These intervals are substantially due to the time required to save (or to load) a large amount of context information for operating systems and running applications. Considering the high customization of network device, which generally include specialized hardware requiring less "context" data than general purpose PCs, we can reasonably suppose that future specific developments of such primitives for network devices will achieve much shorter wake-up and sleeping times.

Given the nature of networks protocols, putting entire backbone devices in standby status would not be a practical approach. This is because devices have to maintain their network presence by replying to signalling messages, otherwise they fall out the network and cause a new re-convergence of routing and traffic engineering protocols (e.g., OSPF, GMPLS, etc.). Thus, in order to transparently manage standby primitives and avoid the network falling off, devices must always maintain active control-plane processes, and some connectivity towards other nodes to exchange signaling messages.

Starting from these considerations, we assume next-generation network devices to have the capability of selectively putting in standby status some of their physical

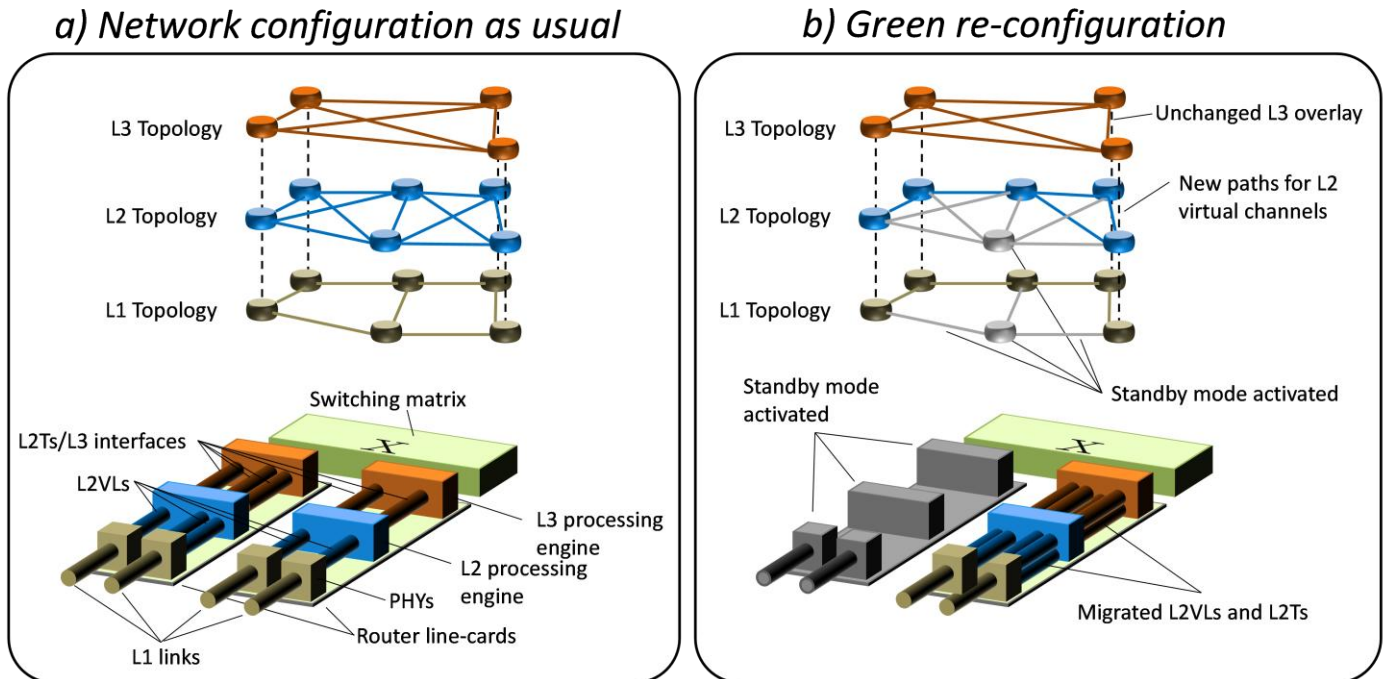


Figure 1. State-of-the-art backbone network and device scenario in subfigure a), and, in subfigure b), the proposed approach to enable network devices to selectively sleep their components. The approach is fully based on network re-configuration at L2, and aims at managing standby primitives in a transparent way with respect to the L3 overlay.

components. Throughout the paper we refer to line-cards as the “minimum granularity” building block that can be put in sleeping state, but the approach we propose would be even more beneficial if applied at each line-card sub-component (e.g., PHYs, packet processing engines, etc.).

Our idea simply consists of putting to sleep those portions of the device data-plane that are not currently used, like redundant link interfaces, or that are so lightly utilized that their jobs may be temporarily transferred to other active line-cards (see Section IV).

Finally, for the sake of completeness, we have also to underline that the introduction of standby primitives must go with the development of specific watchdog tasks, which periodically wake up sleeping hardware and check for possible faults or anomalies.

IV. SMARTLY SUPPORTING STANDBY PRIMITIVES

As already introduced, we exploit two features already present in today’s networks and devices: the network resource virtualization and the modular architecture of network nodes. These features give us the opportunity of decoupling physical elements, such as line-cards that may be put in standby, from their (virtual) functionalities and resources, so that the latter can be migrated towards other active physical elements of the same device.

In more detail, our idea is mainly based on the exploitation of today’s L2 protocols for backbone networks (e.g., mainly MPLS and Ethernet), since:

- they are specifically used to manage the virtualization of the physical network infrastructure;

- they already include efficient mechanisms for rapidly moving/migrating L2VLs across the network (e.g., the fault recovery procedures).

In order to avoid unwanted drawbacks in network behavior, our solution is completely transparent to the L3: IP routing protocols are unaware of network changes and so control message exchange and L3 reconfiguration are avoided.

The rest of this section is organized as follows. Sub-section A describes the main drawbacks in using standby primitives without any explicit network support. Subsection B introduces the approach we propose for smartly supporting standby.

A. Standby primitives without smart support

The sole adoption of standby primitives may cause significant drawbacks in network operational behavior.

For instance, if a line-card entered standby status, all packet forwarding operations would stop, and no further signaling messages could be received and/or transmitted by that line-card. Consequently, its PHYs, L2VLs, and L2Ts would fall out the network, as the entire line-card would fault. This triggers fault protection mechanisms for L2VLs to re-converge towards a new L2 topology. Since the terminations of such L2 channel are involved in the topology change, also modifications to the IP logical overlay are highly probable. If the IP logical topology changes, L3 routing protocols must re-converge in their turn, and find new optimal paths.

All this can be summarized in:

- no negligible amount of signaling traffic across the whole network;
- slow network re-convergence, since both L2 and L3 routing/traffic engineering protocols are involved, and

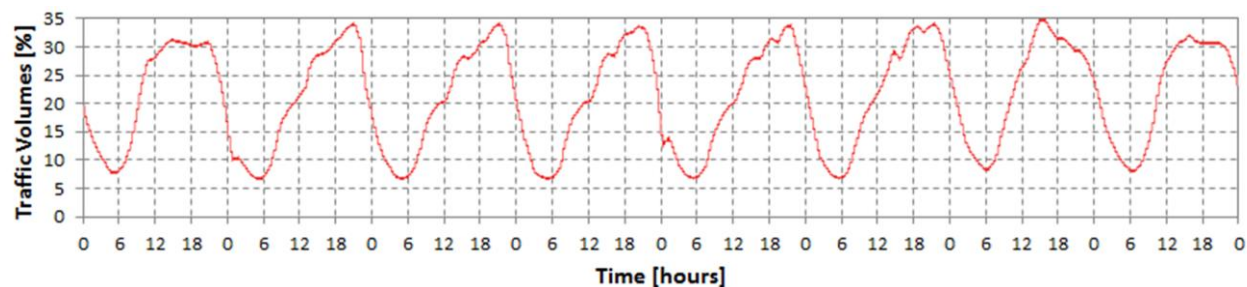


Figure 2. Typical daily trend of traffic volumes in a physical interface of a router deployed in a Telco network. These values are reported as percentage with respect to the physical link capacity.

IP protocols generally require long re-convergence times;

- double re-convergence at L2 and L3, which may lead to unwanted traffic paths across the network.

B. Introducing the smart support

In order to avoid the above mentioned drawbacks, standby modes have to be explicitly supported with special techniques to maintain the “network presence” of sleeping components. Our idea consists of making line-cards left active to “cover” sleeping parts, without the device losing any networking resource/functionality. So, before a line-card enters standby status, it has to transfer its resources and activated functionalities to other cards that will remain active.

It is worth noting that such resources and functionalities to be moved are substantially the ones related to all L2VLs and L2Ts carried by the line-card PHYs.

As shown in Figure 1.b, we fully exploit the L2 protocols to migrate L2VLs from the line-card entering standby to other line-cards. This obviously requires a new L2VL re-mapping on the physical network topology, since each L2VL has to enter the device from the PHYs of other line-cards.

Up to this point, the proposed procedure looks very similar to ones involved in fault recovery events, except from the fact the L2 resource re-mapping is made before the line-card become unavailable, and then, by using suitable re-allocation mechanisms, L2VL migrations can be performed without traffic losses and/or service interruptions [9].

The step beyond, and the most innovative part of our approach consists of making this L2 re-mapping, and then also standby hardware transitions, totally transparent to the IP layer.

If each L2VL of the sleeping line-card is re-mapped on another active line-card, than the network node sees the same number of L3 interfaces (i.e., the L2Ts), which connect the local router to the same set of IP nodes, as before the L2VL migration. In other words, the full re-mapping at L2 results in a L3 overlay topology substantially identical to the starting one.

Even if no re-convergence of IP routing would be required, standard routers usually considers the L2Ts of re-mapped L2VLs as new network interfaces, since they are allocated on different PHYs and line-cards. A L2T before the migration generally differs from the new one in the interface name/identifier.

Capitalizing on such considerations, our approach simply

consists of maintaining the same identification parameters of its old copy in the new L2T. In this way, and as demonstrated by the prototype introduced in section VI.B, IP routing protocols are unaware of both the L2 re-mapping and line-card sleeping/wake-up events.

V. THE ROLE OF ENERGY-AWARE TE

Standby state of line cards cannot be locally managed by routers, since it requires the re-allocation of a number L2VLs across the network. For this reason, we propose to use a Network Control Unit (NCU), i.e., a network node devoted collect traffic load information from the routers, and to consequently apply a traffic engineering criterion to perform the L2VLs reconfiguration, and to meet the QoS constraints.

We suppose the NCU performing its main tasks, thanks to the knowledge of the topology and the network traffic conditions, by means of routing protocols with traffic extensions (e.g., OSPF-TE), and to the ability of managing L2VLs, by means of the GMPLS protocol.

The NCU continuously monitors if incoming traffic volumes exceed fixed thresholds. When traffic volumes exceed such thresholds, the NCU executes the TE algorithm and detects which line-cards of routers have to sleep or to wake-up.

In order to avoid too frequent and fluctuating network re-configurations, we think the NCU to use only few traffic thresholds, able to confine typical “night and day” profiles of traffic volumes. In more detail and as shown in Fig. 2, traffic volumes usually exhibit a well-known sinusoidal-like behavior during the 24 hours, with rush hours in the afternoon and low-volume nightly hours. The minimum levels of traffic generally corresponds to the 20-30% of the peak volumes.

Our idea is that fixing few thresholds (e.g., one at the half of the peak load, or three at the 25%, 50% and 75% of the peak) will be a reasonable compromise between energy-saving and network stability.

The TE algorithm is devoted find the optimal network configuration that can be achieved re-mapping L2VLs on the physical topology. The optimal configuration is the one that maintains the QoS constraints in terms of maximum links utilization and back-up availability, and that allows sleeping the larger number of line-cards.

The L2VL re-mapping problem can be defined as an integer linear programming (ILP) problem. However, given the intrinsic complexity of such kind of problems, we propose a

simple TE heuristic, that can be easily adopted to save a considerable amount of energy.

The TE algorithm we propose is based on the links traffic load, and can be summarized as follows.

- 1) The set of all routers line-cards $\{LC_{ij}\}$ (where i and j represent the identifiers of the router and of the line-card, respectively) is ranked in increasing order of traffic load.
- 2) The minimum load line-card is selected, LC_{min} , and it is removed from $\{LC_{ij}\}$.
- 3) LC_{min} “standby” is evaluated: all the L2VLs using the physical interfaces of LC_{min} are re-mapped: a new path not using LC_{min} is searched. If a new path for all the involved L2VLs is found, LC_{min} can enter standby and $\{LC_{ij}\}$ is updated, otherwise it has to remain active.
- 4) if all the network line-cards are not tested, return to the step 2.

The algorithm idea is really simple: it tries to pass in standby mode the less used line-cards, so that to minimize the impact of L2VLs re-mapping on the whole network and at the same time to maximize the number of sleeping line-cards.

VI. PERFORMANCE EVALUATION

This section is organized as follows. Subsection A analyzes the potential impact of the TE methodology. Subsection B evaluates the performance of an energy-aware prototype, which includes both the selective standby capability and the smart support of subsection IV.B.

A. The potential impact of energy-aware TE

In this section, we evaluate the impact on energy consumptions of a typical Telcos network, of the simple TE approach proposed in section V. We considered an IP overlay with a fully meshed topology, where each IP link was realized by means of a single pairs of L2T: so even the starting L2 topology is a full mesh one. Regarding the physical network, we considered the two different topologies analyzed in [5], that are composed by 159 nodes and 614 links, and 244 nodes and 1080 links, respectively. We also suppose the presence of a redundant copy for each link. For the sake of simplicity, every line-card is thought to host a single PHY. The physical network has been dimensioned by fixing the maximum load:

- a reference IP traffic matrix, representing the peak-hours traffic, has been generated;
- L2VLs have been allocated on the physical topology by using a simple shortest path routing strategy.
- the capacity of the physical links has been defined

TABLE I

MAXIMUM NUMBER OF LINE-CARDS THAN CAN BE PUT TO SLEEP, ENERGY CONSUMPTIONS AND SAVINGS ACCORDING TO DIFFERENT TRAFFIC VOLUMES AND THE TWO PHYSICAL TOPOLOGIES.

η	Network topology 1 (159 nodes, 614 links)			Network topology 2 (244 nodes, 1080 links)		
	Sleeping links [#]	Consumption [kW]	Savings [%]	Sleeping links [#]	Consumption [kW]	Savings [%]
100%	0%	235.5	0%	0%	400.0	0%
75%	42%	130.9	44.4%	45.7%	216.0	46.0%
50%	48%	126.5	46.3%	51%	208.2	48.0%
25%	51.6%	111.5	52.6%	55.4%	181.9	51.6%

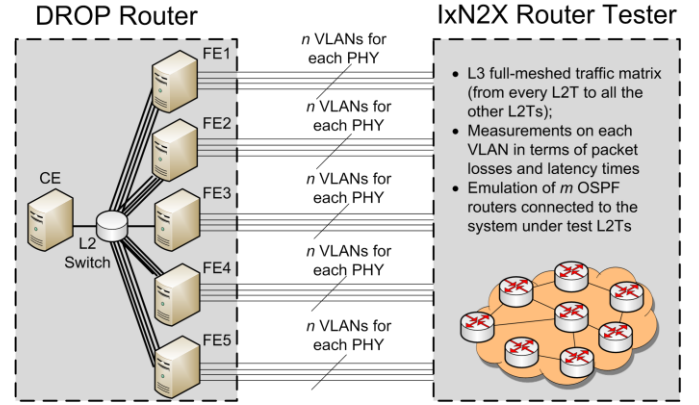


Figure 3. Experimental testbed utilized to evaluate the performance of the proposed approach. The DROP router is composed by 5 FEs, 1 CE and a L2 switch used for internal traffic switching. The IxN2X router tester is used as testing means at both data- and control-planes, since (i) it generates IP over VLAN traffic flows; (ii) it measures the DROP router data-plane performance in terms of throughput, packet losses and latencies; (iii) it emulates OSPF routers connected to the system under test.

fixing a maximum traffic load equal to 50% and the availability of physical interfaces with capacity multiple of 2.5 Gbit/s.

In this way, we obtained two physical networks able to satisfy a specific IP traffic matrix with a high over-provisioning degree, similar to the ones of real-world Telcos' backbones [5]. Then, we applied, by means of simple numerical calculations, the traffic engineering optimization criterion according to different levels of traffic loads η . In detail, we used three different traffic levels, where η is equal to 75%, 50% and 25% of the maximum-load traffic matrix, respectively. The results are reported in Table I, and are expressed as percentage of physical line-cards that can be put to sleep. In more detail, the results in Table I demonstrate that also in the presence of high traffic volumes ($\eta=75\%$), more than 40% of line-cards can enter standby modes. When traffic levels decrease, standby primitives can be enabled on more than 50% of line-cards.

Then, we supposed our network to be composed by Cisco GSR 12008 routers. Exploiting the measurements in [4], we suppose nodes to consume 400 W without line-cards. Each line-card consumes 70 W when active, and 10 W in standby status. Table I outlines that network energy absorption can be reduced by a figure of more than 40%.

B. The energy-aware router prototype

In order to develop a modular router prototype with standby capabilities, we used an existing open-source SW framework, called DROP [10]. In detail, DROP allows aggregating multiple SW routers, based on the Linux operating system and components-off-the-shelves (COTS) hardware, to work as a single modular IP router.

As shown in Figure 3, reporting the testbed we used, a number of SW routers, namely forwarding elements (FEs), are devoted to perform data-plane operations,

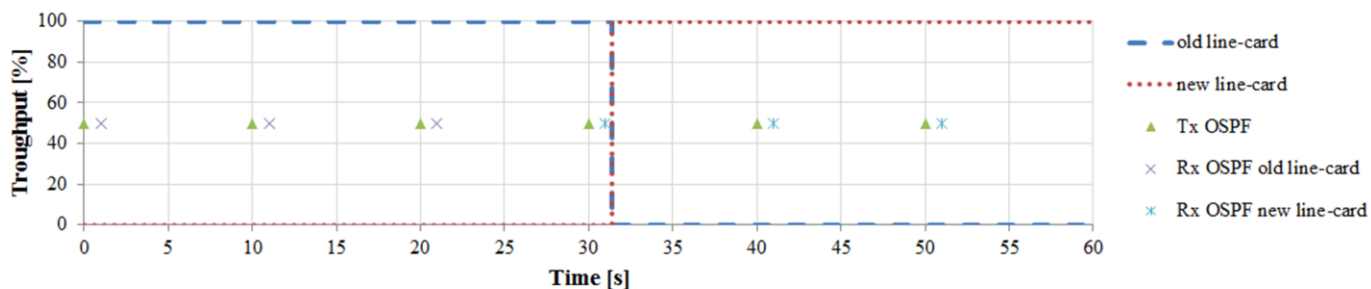


Figure 3. Throughput of traffic carried by a L2VL (VLAN) during a re-mapping process on both the line-card going to sleep, and the new one. The figure also reports the time instants (measured by the IxN2X) of OSPF signaling packets generated and received by the DROP router.

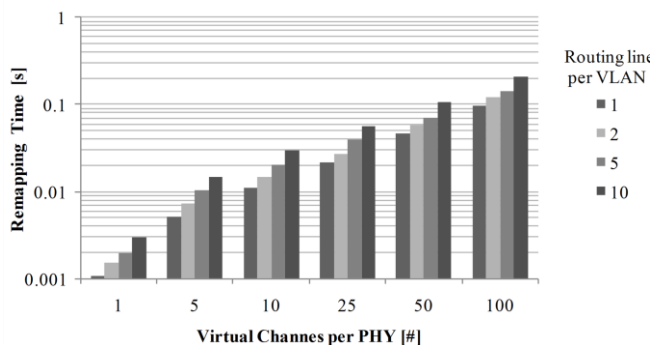


Figure 4. Remapping times according to the number of L2VLs per PHY, and to the number of routing table entries to be updated during the migration process.

while a single SW router works as central control element (CE), and runs signaling protocols' applications for the whole aggregated router. A L2 switch is internally used as switching matrix.

Each FE is realized with a dual Xeon 5550 based server, capable of entering the ACPI sleeping state. Every FE can be thought of as a single line-card of the node, and hosts 4 physical links. The L2VLs are realized by means of IEEE 802.1q VLANs.

Regarding benchmarking tools, we used the Ixia N2X router tester, which allows generating and measuring traffic flows with high accuracy, and also emulating the presence of others OSPF routers (Figure 3).

The DROP architecture was extended in order to support standby primitives and their smart operation, as per sections III and IV. When the DROP control element receives a signaling message asking for the sleeping of a line-card and the relative re-mapping of its L2VLs to other forwarding elements, it starts allocating identical copies of the VLAN interfaces (including their IP configurations) to be remapped on the other elements. During such process, DROP maintains the same identifiers between the old copies and new ones. So, for a short time period, the router has two identical copies of the L2T placed on different line-cards.

When the allocation process is fulfilled, and the new VLANs are ready to be used, the DROP control element sends an acknowledgment message, and waits for a further reply. As soon as this reply is received, DROP updates its routing tables, in order to use the re-mapped interfaces, and starts de-allocating the old L2Ts, and the old line-card can finally enter the standby mode. Similar operations are performed in case of

line-card wakeup.

Figures 4 and 5 show some results obtained to evaluate the performance of the above-introduced implementation. In more detail, the results in Figure 4, which reports the throughput measured by the IxN2X during a VLAN remapping, show that the traffic crossing the DROP router switches the output line-card without any forwarding service interruption (in all tests no packets were lost). Moreover, Figure 4 also shows the reception instants of OSPF Hello packets crossing the VLAN: we can see how the OSPF adjacency is maintained also after the VLAN re-mapping. Thus, the proposed solution is transparent to the L3.

Figure 5 reports experimental measurements of the time periods that elapse from the reception of a sleeping request message, to the completion of traffic switching among line-cards. The measures were repeated for a different number of VLANs per PHY, as well as for a different number of routing table lines that have the re-mapped VLAN as output interface and, consequently, have to be updated during the migration process.

The obtained results show remapping times scaling in an almost linear way with respect to the number of involved VLANs and routing table entries. The maximum measured time is equal to 200 ms, and corresponds to the case with 100 VLANs per PHY, and 10 routing table entries per VLAN to be updated. Then, the time to sleep or to wake up a FE, measured with our testbed, is about 2-3 s. As previously sketched, specific standby techniques for network devices can sensibly reduce these times.

VII. CONCLUSION

We dealt with the use of standby primitives in backbone network devices. We considered state-of-the-art device architectures and protocol stack, which are usually deployed in current Telcos' core networks.

We discussed potential drawbacks on network performance and operational behavior, and we proposed a comprehensive approach to smartly support such primitives avoiding network instabilities and traffic signaling storms. The proposed solution allows dynamically managing standby primitives according to the network traffic volumes, and some QoS and resilience performance constraints. To this purpose, the proposed solution exploits in depth the virtualization capabilities of L2 protocols and the modularity level of today's network devices, and it allows managing hardware

standby and wakeup events in a fully transparent way with respect to the IP layer.

The proposed approach has been experimentally validated by means of an energy-aware modular router prototype.

REFERENCES

- [1] Global e-Sustainability Initiative (GeSI), "SMART 2020: Enabling the low carbon economy in the information age", <http://www.theclimategroup.org/assets/resources/publications/Smart2020Report.pdf>.
- [2] J.Baliga, R.Ayre, K.Hinton, R.S.Tucker, "Photonic switching and the energy bottleneck," Proc. Internat. Conf. Photonics in Switching, San Francisco, CA, USA, 2007.
- [3] R.Bolla, R.Bruschi, F.Davoli, F.Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures" IEEE Communications Surveys and Tutorials, to appear in vol.13, no.4, Dec. 2011.
- [4] J.Chabarek, J.Sommers, P.Barford, C.Estan, D.Tsiang, S.Wright, "Power Awareness in Network Design and Routing," Proc. IEEE Infocom'09, Phoenix, AZ, Apr. 2008.
- [5] R.Bolla, R.Bruschi, K.Christensen, F.Cucchietti, F.Davoli, S.Singh, "The Potential Impact of Green Technologies in Next Generation Wireline Networks - Is There Room for Energy Savings Optimization?," IEEE Communications, to appear in Jul. 2011 issue.
- [6] A.Cianfrani, M.Listanti, V.Eramo, M.Marazza, E.Vittorini, "An energy saving routing algorithm for a green OSPF protocol," Proc. IEEE Infocom'10, San Diego, CA, Mar. 2010.
- [7] J.Restrepo, C.Gruber, C.Machoca, "Energy Profile Aware Routing," Proc. IEEE GreenComm'09, Dresden, Germany, June 2009.
- [8] B.Nordman, K.Christensen, "Proxying: The Next Step in Reducing IT Energy Use," IEEE Computer, Vol. 43, No. 1, pp. 91-93, January 2010.
- [9] Y.Wang, E.Keller, B.Biskeborn, J.van der Merwe, J.Rexford, "Virtual Routers on the Move: Live Router Migration as a Network-Management Primitive," Proc. ACM SIGCOMM, Seattle, WA, Aug. 2008.
- [10] R.Bolla, R.Bruschi, G.Lamanna, A.Ranieri, "DROP: An Open-Source Project towards Distributed SW Router Architectures," Proc. IEEE GlobeCom'09, Honolulu, Hawaii, USA, Dec. 2009.