

Introducing Standby Capabilities into Next-generation Network Devices

Raffaele Bolla
DIST, University of Genoa
Italy
raffaele.bolla@unige.it

Roberto Bruschi
CNIT
Italy
roberto.bruschi@cnit.it

Antonio Cianfrani, Marco Listanti
University of Rome "Sapienza"
Italy
antonio.cianfrani@uniroma1.it
listanti@infocom.uniroma1.it

ABSTRACT

In this contribution, our main objective is to introduce and to support standby modes in next-generation devices for backbone networks. 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 is mainly founded on two features already and largely present in today's networks and devices: the network resource virtualization and the modular architecture of nodes. By means of a Linux SW router prototype, we demonstrated that the proposed approach allows dynamically sleeping pieces of hardware in several network nodes, while maintaining good network performance levels, and without causing undesired network instabilities.

Categories and Subject Descriptors

C.2.6 [Routers]: Energy-aware Routers

General Terms

Design, Experimentation, Measurement, Performance.

Keywords

Green networks, Router Power Management, SW Router.

1. INTRODUCTION

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. In more detail, 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 [1].

Unfortunately, energy consumption of current network devices is substantially flat [2], and mainly depends on their maximum switching capacities, rather than their actual workloads. So, redundant or lightly utilized links and

devices consume the same amount of energy than devices switching large volumes of traffic. Baliga et al. [3] suggested that the ultimate capacity of the Internet of tomorrow might eventually be constrained by energy density limitations in Telcos' points of presence, where backbone devices are deployed, rather than by the bandwidth of next-generation devices.

Emerging research approaches to network control, routing and traffic engineering [4] [5] 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. As shown in [1], the massive adoption of such selective network turn-offs can enable next-generation backbone networks to almost halve their energy requirements, especially if applied to redundant hardware, too. Widespread concerns and criticisms on this kind of approaches are mainly founded on the fact that turned-off elements (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". Moreover, 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.

The approach is mainly founded on 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 [6]. However, in such work the authors suggested the migration

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM PRESTO 2010, November 30, 2010, Philadelphia, USA.

Copyright 2010 ACM 978-1-4503-0467-2/10/11 ...\$10.00.

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 key advantage of our solution with respect to a simple switching-off consists of reduced recovery times, as well as in the possibility of managing device standby and wake-up events in a transparent way with respect to the IP layer, avoiding useless signaling storms and slow network re-convergences.

The paper is organized as follows. Section 2 introduces the reference backbone network scenario. Section 3 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 4. Section 5 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 6. Finally, the conclusions are drawn in section 7.

2. THE NETWORK SCENARIO

As far as the network protocol stack is concerned, 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) in the case of MPLS protocol or a Virtual LAN (VLAN) in the case of Ethernet protocol), which directly connect two or more nodes working at L3. Then, each LSP and/or VLAN constitutes a different logical sub-network 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) or Intermediate system to intermediate system (IS-IS), 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 network 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.

3. 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 hardware sub-modules (e.g., a network interface) powered on, awaiting for external wake up messages. The main advantages of using standby modes with respect to simply switching off hardware components consist in: (i) shorter times for waking up and recovering normal operations; (ii) the ability of performing minor operations also when components are sleeping.

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.

However, given 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 signaling messages, otherwise they fall out the network and cause a new re-convergence of routing and traffic engineering protocols (e.g., OSPF, IS-IS, GMPLS, etc.). Thus, in order to transparently manage standby primitives and avoid the network falling off, devices must always maintain active control-plane processes, as well as 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 their physical components and building blocks at the data-plane level. 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.).

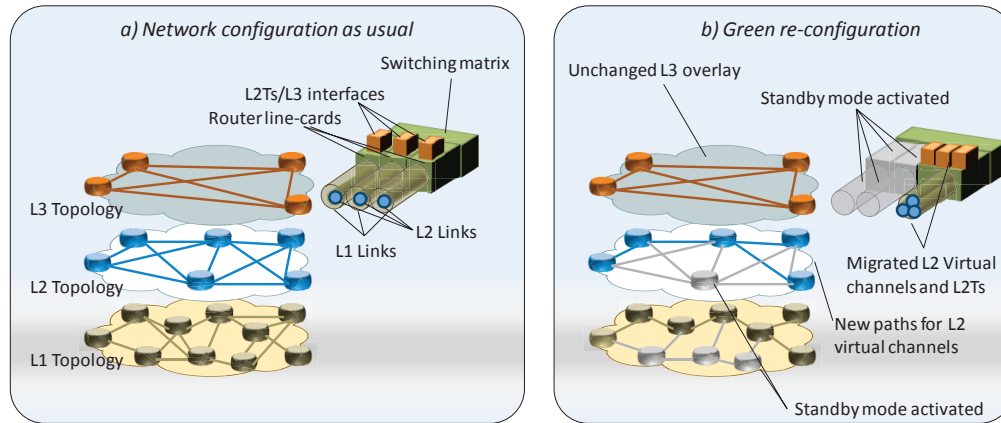


Figure 1. State-of-the-art backbone network and device scenario in subfigure a), and, in subfigure b), the proposed approach to enable network devices to selectively sleep their components. The approach is fully based on network re-configuration at L2, and aims at managing standby primitives in a transparent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

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 4).

4. SMARTLY SUPPORTING STANDBY PRIMITIVES

The network resource virtualization and the modular architecture of network nodes 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., 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.

4.1 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 (i.e., the IP interfaces) 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: moreover, IP protocols are well-known to require long re-convergence times;
- double re-convergence at L2 and L3, which may lead to unwanted traffic paths across the network.

4.2 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 [6].

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. In detail, 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 4.2, IP routing protocols are unaware of both the L2 re-mapping and line-card sleeping/wake up events. It is worth noting that the operations similar to the ones for supporting the line-card sleeping events, can be applied also for managing wake-up events.

5. THE ROLE OF ENERGY-AWARE TRAFFIC ENGINEERING

The use of standby primitives in a single line-card can be really exploited only if an energy-aware traffic engineering strategy is introduced. In particular this strategy should be able to change the L2 topology so that a set of router line-card can pass in standby mode. The criterion exploits daily fluctuations in traffic volumes and network resource over-provisioning, maintaining the QoS constraints. Thus, our scope is to provide a traffic engineering strategy that supports energy saving capabilities of real network devices. In particular, we propose a centralized algorithm that, on the basis of the different layers network topologies and of the traffic demand, is able to re-compute new paths for a subset of L2VLs. The new paths are computed in order to empty the traffic load incoming to certain router line-cards, and so can enter standby modes, and save energy. This strategy can be implemented as an off-line solution that needs to “monitor” the incoming traffic: when traffic go under a fixed threshold, the algorithm allows for line-card sleeping, while when this threshold is exceeded, the sleeping line-cards are re-activated. The new L2 topology maintains the same QoS constraints in terms of maximum link load and back-up availability; the only performance degradation is latency increase due to the L2VL paths lengths increase. The algorithm for re-allocating L2VLs can be summarized as follows:

- 1) routers line-cards are ranked on the basis of their traffic load;
- 2) the line-card having the minimum load is selected and its standby state is “tested”;

- 3) for each physical link of the line-card, all the L2VLs using that link are detected. For each L2VL a new path not using the router line-card is searched, meeting the maximum link load constraint. If a new path for each involved L2VL is found, the new paths are activated and the line-card passes in standby state; otherwise this line-card remain active;

the last two steps are executed recursively until all the line-card are checked.

The idea of the algorithm is quite simple: it tries to put in standby mode the less used physical links in order to maximize energy saving. Different strategies to optimize the solution of this problem can be defined, but this is out of the scope of this paper. our solution has to show that even a really simple algorithm allows for a considerable energy saving during low traffic hours.

6. PERFORMANCE EVALUATION

This section is organized as follows. Subsection 1 analyzes the potential impact of the optimization criterion for the energy-aware traffic engineering methodology introduced in section 5. Subsection 2 shows some performance evaluation results obtained with a modular Software Router prototype, which includes both the selective standby capabilities of section 3, as well as the smart support introduced in subsection 4.2.

6.1 The potential impact of energy-aware Traffic Engineering

In order to analyze the potential impact of the simple traffic engineering criterion introduced in section 5, we considered two different physical networks that have the same features of the one studied in [1], and that are composed by 159 nodes and 614 links, and 244 nodes and 1080 links, respectively. We suppose the presence of a redundant copy for each link. For the sake of simplicity, every line-card is thought to host a single physical link.

The IP overlay has a fully meshed topology, and each IP link has been realized by means of single pairs of L2T: so even the L2 topology is a full mesh one. L2VLs have been allocated on the physical topology by using a simple shortest path routing strategy.

A reference maximum-load traffic matrix has been considered, and, after L2VLs’ allocation, the physical network has been dimensioned by considering two conditions: (a) a maximum traffic load on physical links equal to 50%; (b) 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 [1].

Starting from these physical topologies, 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 obtained are reported in Table I, and are expressed in terms of average percentage of physical line-cards per router 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 [2], we suppose nodes to have an energy consumption of 400 W without line-cards. Each line-card consumes 70 W when active, and 10 W in standby status. Starting from these data, Table II outlines that, by using the proposed approach, network energy absorption can be reduced by a figure of more than 40%. Moreover, given that maximum network utilizations are (and will be also in the future) usually less than 40%, we can state that savings larger than 45-50% would be easily achieved.

6.2 The energy-aware router prototype

We used an existing open-source SW framework, called DROP [7]. 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 2, which reports the testbed we used, a number of SW routers, namely forwarding elements (FEs), are devoted to perform data-plane operations, 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 forwarding element is realized with a dual Xeon 5550 based server, capable of entering the ACPI S3 sleeping state and equipped with 8 Gigabit Ethernet interfaces. Four such interfaces are used for internal router connectivity, and the other ones for external connections.

Thus, each forwarding element can be thought of as a single line-card of the modular platform, and hosts 4 physical links. The L2VLs are realized by means of IEEE 802.1q VLANs. As far as the validation and benchmarking

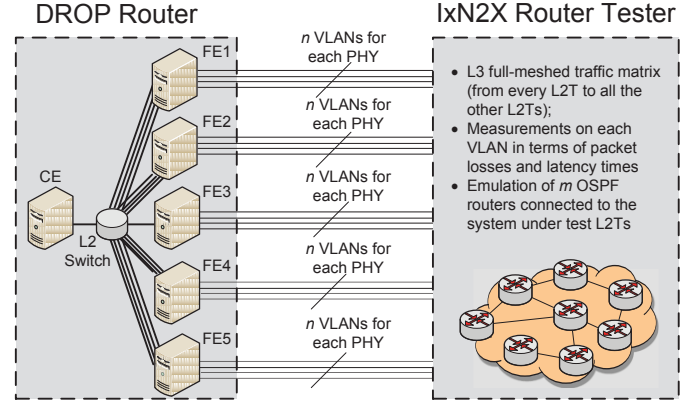


Figure 2. Experimental testbed utilized to evaluate the performance of the proposed approach. The DROP router is composed by 5 FEs, 1 CE and a L2 switch used for internal traffic switching. The IxN2X router tester is used as testing means at both data- and control-planes, since (i) it generates IP over 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.

tools are concerned, we used a professional Ixia router tester, called IxN2X, which allows to generate and to measure traffic flows with high accuracy. The IxN2X is used also to emulate the presence of OSPF routers connected to DROP (Figure 2). The DROP architecture was extended in order to support standby primitives and their smart operation, as per sections 3 and 4. In more detail, 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 (the entire L2Ts including their IP configurations) to be remapped on the other elements. During such process, DROP maintains the same names and 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

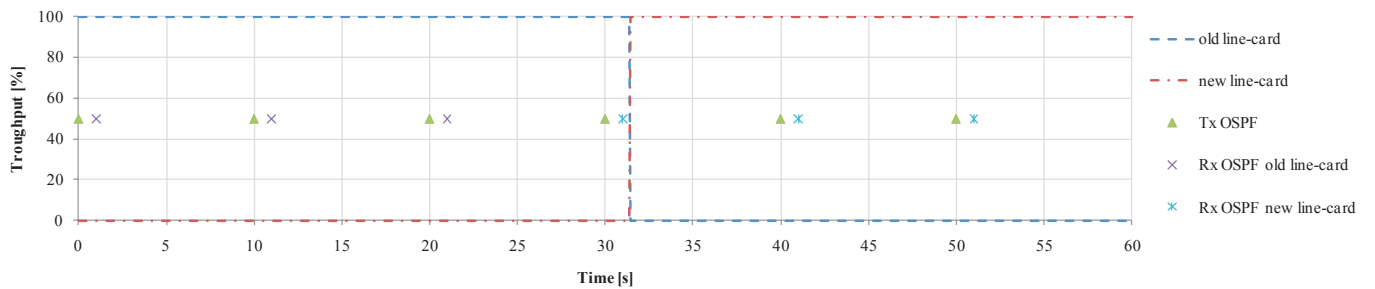


Figure 3. Throughput of traffic carried by a L2VL (VLAN) during a re-mapping process on both the line-card going to sleep, and the new one. The figure also reports the time instants (measured by the IxN2X) of OSPF signaling packets generated and received by the DROP router.

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 3 and 4 show some results obtained to evaluate the performance of the above-introduced implementation. In more detail, the results in Figure 3, 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 3 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-

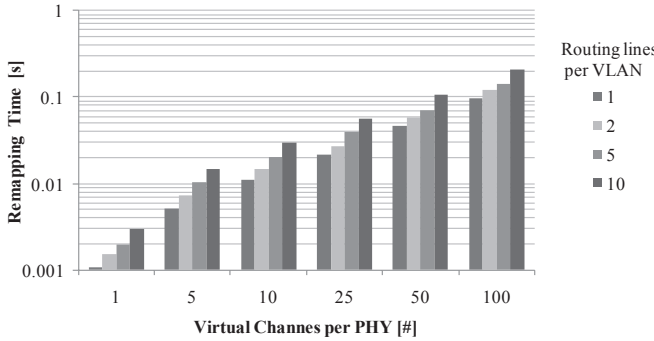


Figure 4. Remapping times according to the number of L2VLs per PHY, and to the number of routing table entries to be updated during the migration process.

mapping. Thus, the proposed solution is transparent to the L3. Figure 4 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 remapped 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.

7. CONCLUSIONS

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 traffic volumes incoming to the network, and some QoS and resilience performance

Table I. Maximum number of line-cards than can be put to sleep according to different traffic volumes.

η	Network topology 1 (159 nodes, 614 links)	Network topology 2 (244 nodes, 1080 links)
75%	42%	45.7%
50%	48%	51%
25%	51.6%	55.4%

Table II. 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)	
	Consumption [kWh]	Savings [%]	Consumption [kWh]	Savings [%]
100%	235.5	0%	400.0	0%
75%	130.9	44.4%	216.0	46.0%
50%	126.5	46.3%	208.2	48.0%
25%	111.5	52.6%	181.9	51.6%

constraints. To this end, the proposed solution exploits in depth the virtualization capabilities of L2 protocols and the modularity level of today's network devices, managing hardware standby and wakeup events in a fully transparent way with respect to the IP layer.

ACKNOWLEDGMENTS

This work was supported by the ECONET and the TREND European projects, funded under the Framework Programme 7, theme ICT-2009.1.1.

REFERENCES

- [1] 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 Nov. 2010 issue.
- [2] 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.
- [3] 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.
- [4] 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, 2010.
- [5] J.Restrepo, C.Gruber, C.Machoca, "Energy Profile Aware Routing," Proc. IEEE GreenComm'09, Dresden, Germany, June 2009.
- [6] 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.
- [7] 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