

Energy-Efficient Sleeping Modes for Next-Generation Core Networks

Raffaele BOLLA¹, Roberto BRUSCHI², Antonio CIANFRANI³, Marco LISTANTI³

¹*DIST – University of Genoa, Via all’Opera Pia 13, Genoa, 16145, Italy*

Tel: +39 010 353 2075, Fax: +39 010 353 2154, Email: raffaele.bolla@unige.it

²*National Inter-University Consortium for Telecommunications (CNIT),
Via all’Opera Pia 13, Genoa, 16145, Italy Email: roberto.bruschi@cnit.it*

³*INFOCOM – University of Rome “La Sapienza”, via Eudossiana 18, 00184 Rome, Italy
Emails: antonio.cianfrani@uniroma1.it, listanti@infocom.uniroma1.it*

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.

Keywords: Green Networking, Standby Network Capabilities.

1. Introduction

In the last few years, Telecom Operators (Telcos), Internet Service Providers (ISPs) and public organizations around the world reported statistics of network energy requirements and the related carbon footprint, showing an alarming and growing trend. The Global e-Sustainability Initiative (GeSI) [1] estimates an overall network energy requirement of about 21.4 TWh in 2010 for European Telcos, and foresees a figure of 35.8 TWh in 2020 if no green network technologies will be adopted. The sole introduction of novel low-consumption silicon technologies cannot clearly cope with such trends, and be sufficient for drawing ahead current network equipment towards a greener future Internet [2].

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 [3].

Unfortunately, energy consumption of current network devices is substantially flat [4], 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. [5] 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.

In this respect, 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. As shown in [3], 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.

As far as network end-hosts are concerned, this problem was already analyzed by Christensen et al., which proposed to cover sleeping hosts with green connectivity proxies [8]. Unfortunately, given the intrinsic nature and the architecture of devices and protocols, backbone networks require a slightly different approach to support sleeping primitives.

2. Objectives

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 [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 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.

3. 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.

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.

4. 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:

- shorter times for waking up and recovering normal operations;
- 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 signalling 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

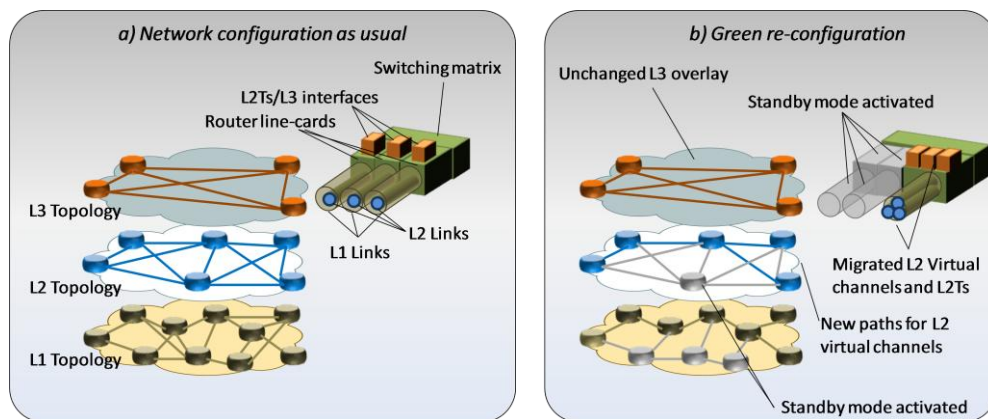


Figure 1. State-of-the-art backbone network and device scenario in subfigure a), and, in subfigure b), the proposed approach to enable network devices to selectively sleep their components. The approach is fully based on network re-configuration at L2, and aims at managing standby primitives in a transparent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

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.).

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 5).

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.

5. Methodology

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. Subsection C discusses the role of the TE in managing standby states.

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 (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.

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.

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 6.B, 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.

C. *The Role of Energy-aware TE*

The use of standby primitives in a single line-card clearly impacts not only on the local node, but potentially on the whole network, since the re-allocation of L2VLs across different nodes is needed. So, network nodes cannot decide in an independent way when and which line-cards put to sleep, but they need to cooperate for agreeing upon a new network-wide configuration.

With this aim, we propose a traffic engineering criterion able to guide periodical L2 reconfiguration in order to reduce the overall energy consumption of network nodes, while meeting QoS and operational constraints. The criterion exploits daily fluctuations in traffic volumes and network resource over-provisioning.

Referring to regions, delimited by fixed thresholds, on incoming traffic volumes, the criterion decides to put to sleep or to wake-up modules of the network routers, and consequently remap L2VLs. The L2 remapping process aims at statistically maintaining the same QoS features of previous L2Ts, in terms of maximum link load and back-up path availability.

When a threshold is exceeded, and incoming traffic volumes enter a new region, the criterion ranks physical links on the basis of their traffic load. Starting from the one with the minimum load, each physical link is “tested” as follows. For each L2VL crossing the physical link, the criterion searches alternative network paths with enough available bandwidth, and not crossing the physical link under test. If at least one such path exists, the physical link is a candidate for entering standby mode. The check is then performed on the other links on both line-cards at the candidate physical link ends. If all the PHYs are candidates, the line-card can enter standby mode. Since remapped L2VLs have to terminate at the same pair of logical nodes with respect to the original L2VLs, at least one line-card per router will remain always active.

The optimization procedure ends when all the network physical links are checked.

6. Results

This section is organized as follows. Subsection A analyzes the potential impact of the optimization criterion for the energy-aware traffic engineering methodology introduced in section 5.C.

Subsection B shows some performance evaluation results obtained with a modular Software Router prototype, which includes both the selective standby capabilities of section 4, as well as the smart support introduced in subsection 5.B.

A. *The potential impact of energy-aware TE*

In order to analyze the potential impact of the simple traffic engineering criterion introduced in section V.C, we considered two different physical networks that have the same features of the one studied in [3], 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 maximum traffic load on physical links equal to 50%;
- 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 [3].

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 [4], 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.

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%

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 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 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 4 and 5.

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 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-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

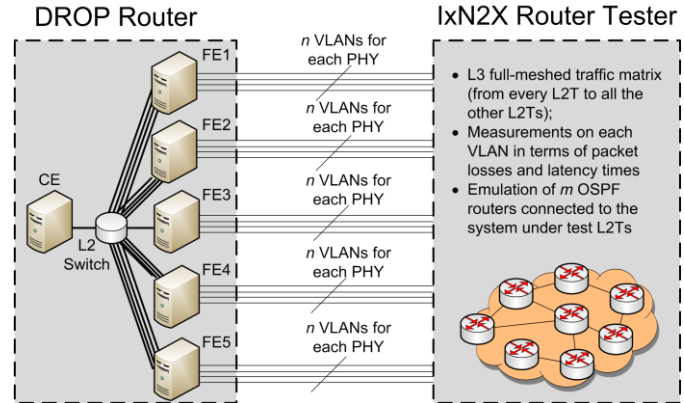


Figure 2. Experimental testbed utilized to evaluate the performance of the proposed approach. The DROP router is composed by 5 FEs, 1 CE and a L2 switch used for internal traffic switching. The IxN2X router tester is used as testing means at both data- and control-planes, since (i) it generates IP over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

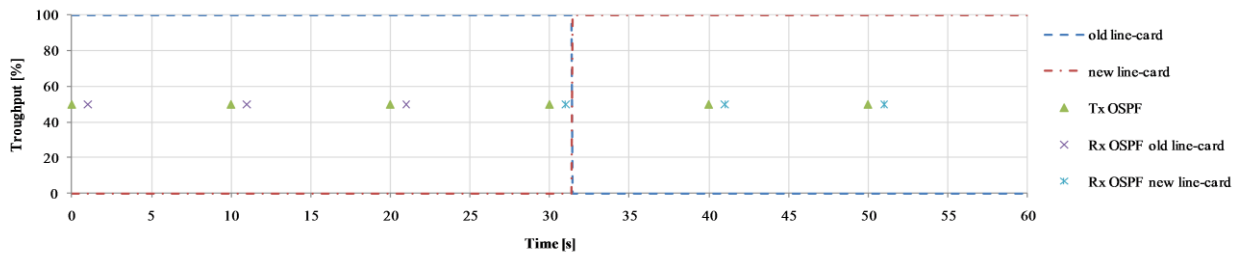


Figure 3. Throughput of traffic carried by a L2VL (VLAN) during a re-mapping process on both the line-card going to sleep, and the new one. The figure also reports the time instants (measured by the IxN2X) of OSPF signaling packets generated and received by the DROP router.

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, the development of specific standby techniques for network devices can sensibly reduce these times.

7. Conclusions

In this contribution, 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 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] 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.
- [3] 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.
- [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] 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.
- [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.

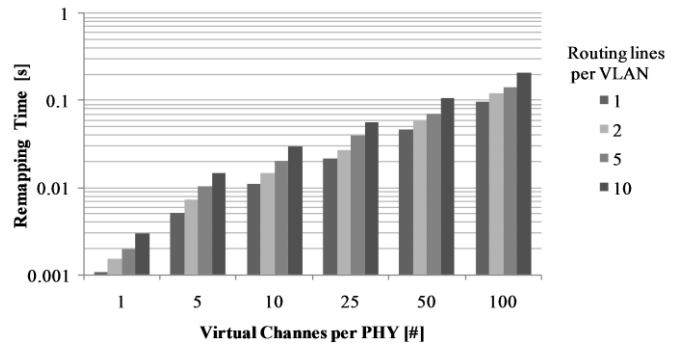


Figure 4. Remapping times according to the number of L2VLs per PHY, and to the number of routing table entries to be updated during the migration process.