

Enabling Backbone Networks to Sleep

Raffaele Bolla, University of Genoa

Roberto Bruschi, National Inter-University Consortium for Telecommunications (CNIT)

Antonio Cianfrani, and Marco Listanti, University of Rome "Sapienza"

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 overdimensioning for maintaining quality of service during rush hours. Unfortunately, current network devices do not have power management primitives, and have constant energy consumption independent of their actual workloads. Starting from these considerations, we propose a viable approach to introduce and 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 reconvergence times. To this purpose, the approach we propose directly exploits the main features of both backbone device architectures and the network protocol stack.

Recently, telecom companies (telcos) and 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, 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 multirack 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 (i.e., by a factor of 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 overdimensioned 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 a 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 current network equipment toward 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 reconverge toward new net-

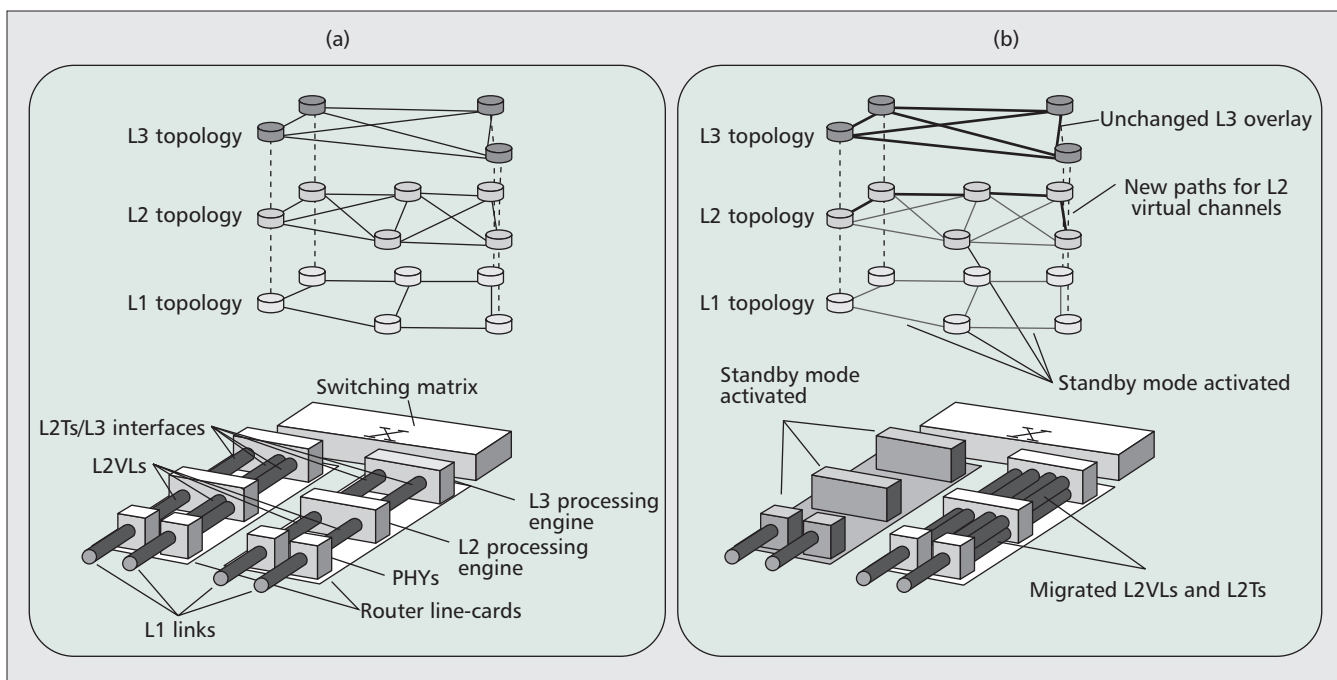


Figure 1. State-of-the-art backbone network and device scenario in subfigure a), and, in subfigure b), the proposed approach to enable network devices to selectively sleep their components. The approach is fully based on network re-configuration at L2, and aims at managing standby primitives in a transparent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

work 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: 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 article is organized as follows. We introduce the reference backbone network scenario. We discuss how hardware standby primitives can be introduced into network devices. The approach to support them while meeting network operational constraints is described. We then argue 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 potential impact of the proposed approach are presented. Finally, the conclusions are drawn in.

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 (layer 3, L3) overlaid over a wavelength-division multiplexing (WDM) optical network (layer 1, L1). A layer 2 (L2) protocol (e.g., multi-protocol label switching, MPLS, or Ethernet) is used to optimally map IP traffic on the physical infrastructure, and to implement value-added network features and services (quality of service [QoS], virtual private networks, mechanisms for fast fault recovery, etc.).

In such an environment, physical channels carry multiple "virtual" L2 links (e.g., label switching paths [LSPs] 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 (PHY) interfaces, and is assumed to include full packet processing capabilities at L2 and L3. As shown in Fig. 1a, 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 L3. Thus, IP links are realized by means of L2Ts on two or more nodes.

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 manage-

ment mechanisms that allow hardware modules in a device to freeze their operations while maintaining their “context” information (e.g., configurations, running tasks). 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, waiting 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 signaling 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 components. Throughout the article 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 subcomponent (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 so lightly utilized that their jobs may be temporarily transferred to other active line cards.

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.

Smartly Supporting Standby Primitives

As already introduced, we exploit two features already present in today’s networks and devices: 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 toward 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., fault recovery procedures).

In order to avoid unwanted drawbacks in network behavior, our solution is completely transparent to L3: IP routing proto-

cols are unaware of network change, so control message exchange and L3 reconfiguration are avoided.

The rest of this section is organized as follows. The next subsection describes the main drawbacks in using standby primitives without any explicit network support. The subsection after that introduces the approach we propose for smartly supporting standby.

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 cause network fallout, as the entire line card would fault. This triggers fault protection mechanisms for L2VLs to reconverge toward a new L2 topology. Since the terminations of such an L2 channel are involved in the topology change, modifications to the IP logical overlay are also highly probable. If the IP logical topology changes, L3 routing protocols must reconverge in turn, and find new optimal paths.

All this can be summarized in:

- No negligible amount of signaling traffic across the whole network
- Slow network reconvergence, since both L2 and L3 routing/traffic engineering protocols are involved, and IP protocols generally require long re-convergence times
- Double reconvergence at L2 and L3, which may lead to unwanted traffic paths across the network

Introducing 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 “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 Fig. 1b, 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 remapping 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 remapping, and then also standby hardware transitions, totally transparent to the IP layer.

If each L2VL of the sleeping line card is remapped on another active line card, 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 remapping at L2 results in an L3 overlay topology substantially identical to the starting one.

Even if no reconvergence of IP routing would be required, standard routers usually considers the L2Ts of remapped L2VLs as new network interfaces, since they are allocated on

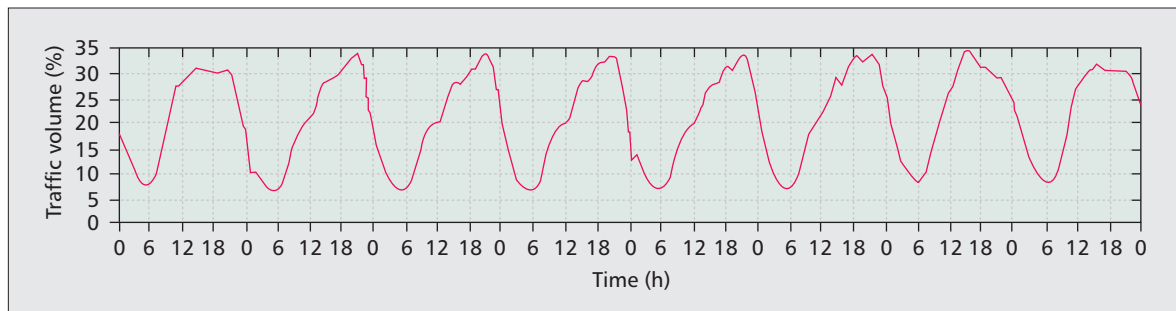


Figure 2. Typical daily trend of traffic volumes in a physical interface of a router deployed in a Telco network. These values are reported as percentage with respect to the physical link capacity.

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 later, IP routing protocols are unaware of both the L2 remapping and line card sleeping/wake-up events.

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), a network node devoted to collecting 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 reconfigurations, we think the NCU should use only a 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 well-known sinusoidal-like behavior over 24 hours, with rush hours in the afternoon and low-volume nightly hours. The minimum levels of traffic generally correspond to 20–30 percent of peak volumes.

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

The TE algorithm is devoted to finding the optimal network configuration that can be achieved by remapping L2VLs on the physical topology. The optimal configuration is the one that maintains the QoS constraints in terms of maximum link utilization and backup availability, and allows the largest number of line cards to sleep.

The L2VL remapping problem can be defined as an integer linear programming (ILP) problem. However, given the intrinsic complexity of such problems, we propose a simple TE heuristic that can easily be adopted to save a considerable amount of energy.

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

- The set of all router line cards $\{LC_{ij}\}$ (where i and j represent the identifiers of the router and line card, respectively) is ranked in increasing order of traffic load.
- The minimum load line card is selected, LC_{min} , and removed from $\{LC_{ij}\}$.
- LC_{min} “standby” is evaluated: all the L2VLs using the PHY interfaces of LC_{min} are remapped: 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.
- If all the network line cards are not tested, return to step 2.

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

Performance Evaluation

This section is organized as follows. The next section 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 an earlier section.

The Potential Impact of Energy-Aware TE

In this section, we evaluate the impact on energy consumptions of a typical telco network of the simple TE approach proposed earlier. 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], which 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 hour 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 fixing a maximum traffic load equal to 50 percent and the availability of physical interfaces with capacity a multiple of 2.5 Gb/s.

In this way, we obtained two physical networks able to satisfy a specific IP traffic matrix with a high degree of overprovisioning, similar to the those of real-world telcos’ backbones [5]. The we applied, by means of simple numerical calculations, the TE optimization criterion according to different levels of traffic loads η . In detail, we used three different traffic levels, where $\eta = 75, 50$, and 25 percent of the maximum load traffic matrix, respectively. The results are reported in Table 1, and are expressed as the percentage of physical line cards that can be put to sleep. In more detail, the results

η	Network topology 1 (159 nodes, 614 links)			Network topology 2 (244 nodes, 1080 links)		
	No. sleeping links	Consumption (kW)	Savings (%)	No. 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%
29%	51.6%	123.8	47.4%	55.4%	203.5	49.1%

Table 1. Maximum number of line cards that can be put to sleep, energy consumption, and savings according to different traffic volumes in two physical topologies.

in Table 1 demonstrate that also in the presence of high traffic volumes ($\eta = 75$ percent), more than 40 percent of line cards can enter standby modes. When traffic levels decrease, standby primitives can be enabled on more than 50 percent of line cards.

We then supposed our network to be composed by Cisco GSR 12008 routers. Exploiting the measurements in [4], we supposed nodes consuming 400 W without line cards. Each line card consumes 70 W when active and 10 W in standby mode. Table 1 shows that network energy absorption can be reduced by a figure of more than 40 percent.

The Energy-Aware Router Prototype

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

As shown in Fig. 3, reporting the testbed we used, a number of switching routers, or forwarding elements (FEs), are

devoted to perform data plane operations, while a single switching router works as a central control element (CE), running signaling protocols' applications for the whole aggregated router. An L2 switch is internally used as a 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 four 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 other OSPF routers (Fig. 3).

The DROP architecture was extended in order to support standby primitives and their smart operation, as per earlier sections. When the DROP control element receives a signaling message asking for the sleeping of a line card and the relative remapping of its L2VLs to other FEs, it starts allocating identical copies of the VLAN interfaces (including their IP configurations) to be remapped on the other elements. During such a process, DROP maintains the same identifiers between the old and new copies. 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 remapped interfaces and starts de-allocating the old L2Ts, and the old line card can finally enter standby mode. Similar operations are performed for line card wakeup.

Figures 4 and 5 show some results obtained to evaluate the performance of the implementation introduced above. In more detail, the results in Fig. 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, Fig. 4 also shows the reception instants of OSPF Hello packets crossing the VLAN: we can see how the OSPF adjacency is also maintained after the VLAN remapping. Thus, the proposed solution is transparent to L3.

Figure 5 reports experimental measurements of the time periods that elapse from the reception of a sleeping request message to the com-

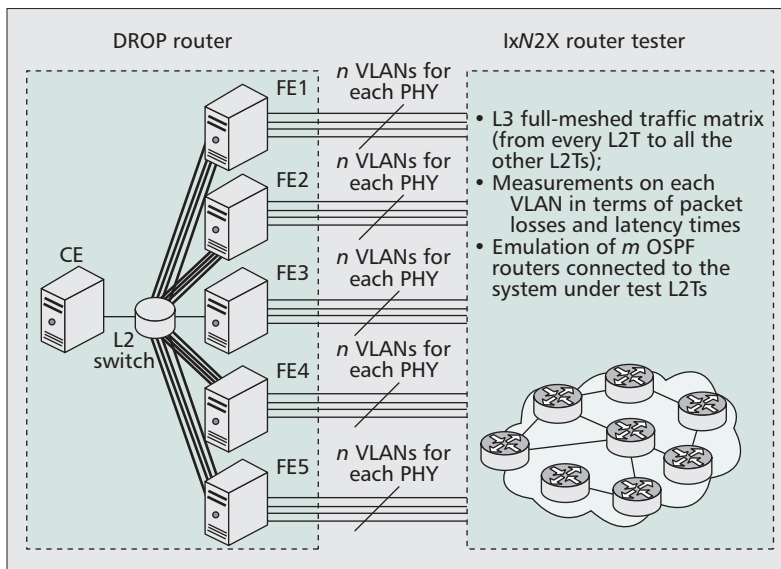


Figure 3. Experimental testbed utilized to evaluate the performance of the proposed approach. The DROP router is composed by 5 FEs, 1 CE and a L2 switch used for internal traffic switching. The IxN2X router tester is used as testing means at both data- and control-planes, since (i) it generates IP over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

pletion 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 almost linearly 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 wake up an FE, measured with our testbed, is about 2–3 s. As previously sketched, specific standby techniques for network devices can sensibly reduce these times.

Conclusion

We deal with the use of standby primitives in backbone network devices. We consider the state-of-the-art device architectures and protocol stack usually deployed in current telcos' core networks.

We discuss potential drawbacks on network performance and operational behavior, and propose a comprehensive approach to smartly support such primitives, avoiding network instabilities and traffic signaling storms. The proposed solution allows dynamic management of standby primitives according to network traffic volume, 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 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 *et al.*, "Photonic Switching and the Energy Bottleneck," *Proc. Int'l. Conf. Photonics in Switching*, San Francisco, CA, 2007.
- [3] R. Bolla *et al.*, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," *IEEE Commun. Surveys and Tutorials*, to appear, vol. 13, no. 4, Dec. 2011.
- [4] J. Chabarek *et al.*, "Power Awareness in Network Design and Routing," *Proc. IEEE INFOCOM '09*, Phoenix, AZ, Apr. 2008.
- [5] R. Bolla *et al.*, "The Potential Impact of Green Technologies in Next Generation Wireline Networks — is There Room for Energy Savings Optimization?," *IEEE Commun. Mag.*, to appear, July 2011 issue.
- [6] A. Cianfrani *et al.*, "An Energy Saving Routing Algorithm for A Green OSPF protocol," *Proc. IEEE INFOCOM '10*, San Diego, CA, Mar. 2010.
- [7] J. Restrepo, C. Gruber, and C. Machoca, "Energy Profile Aware Routing," *Proc. IEEE GreenComm '09*, Dresden, Germany, June 2009.
- [8] B. Nordman and K. Christensen, "Proxying: The Next Step in Reducing IT Energy Use," *IEEE Computer*, vol. 43, no. 1, Jan. 2010, pp. 91–93.
- [9] Y. Wang *et al.*, "Virtual Routers on the Move: Live Router Migration as a Network-Management Primitive," *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.
- [10] R. Bolla *et al.*, "DROP: An Open-Source Project towards Distributed SW Router Architectures," *Proc. IEEE GLOBECOM '09*, Honolulu, HI, Dec. 2009.

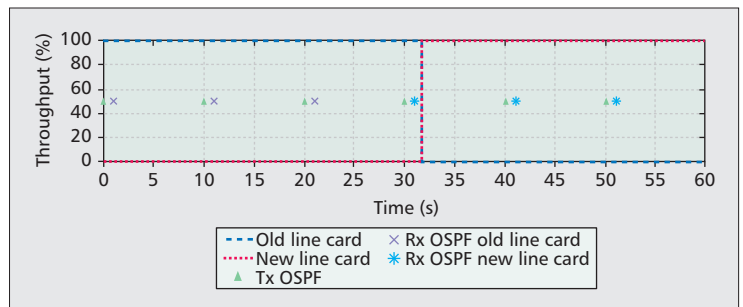


Figure 4. Throughput of traffic carried by an L2VL (VLAN) during a remapping process on both the line card going to sleep and the new one. The figure also reports the time instants (measured by the IxN2X) of OSPF signaling packets generated and received by the DROP router.

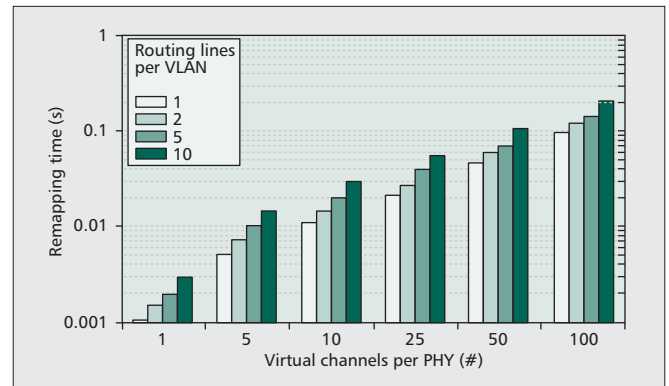


Figure 5. Remapping times according to the number of L2VLs per PHY and the number of routing table entries to be updated during the migration process.

Biographies

RAFFAELE BOLLA [M] (raffaele.bolla@unige.it) received his "Laurea" degree in electronic engineering in 1989 and his Ph.D. degree in telecommunications in 1994, both from the University of Genoa, Italy. He is currently an associate professor at the Department of Communications, Computer and Systems Science (DIST) of the University of Genoa. His main current research interests are in the future Internet and green networking. He is principal investigator of the ECONET project.

ROBERTO BRUSCHI [M] (roberto.bruschi@cni.it) received his M.Sc. degree in telecommunication engineering in 2002 and his Ph.D. degree in electronic engineering in 2006 from the University of Genoa. He is currently a researcher with the National Inter-University Consortium for Telecommunications (CNIT) in the University of Genoa Research Unit. His main research interests include future Internet, green networking, and software routers.

ANTONIO CIANFRANI [M] (antonio.cianfrani@uniroma1.it) received his M.S. degree in telecommunications engineering and Ph.D. degree in Information and Communication Engineering from the University of Rome "Sapienza" in 2004 and 2008, respectively. He is currently a PostDoc at the DIET Department of the University of Rome "La Sapienza." His main scientific contribution are on green networking, routing algorithms in IP networks and switching architectures in optical networks.

MARCO LISTANTI [M] (listanti@infocom.uniroma1.it) received his Dr. Eng. degree in electronics engineering from the University Sapienza of Roma in 1980. In 1981, he joined the Fondazione Ugo Bordoni, where has been leader of the group "TLC network architecture" until 1991. In November 1991 joined the INFOCOM Dept. of the University of Roma "La Sapienza," where he is Professor of Switching Systems. His current research interests focus on traffic control in IP networks and on optical networking.