Insomnia in the Access

or How to Curb Access Network Related Energy Consumption

Eduard Goma[†], Marco Canini[⋆], Alberto Lopez Toledo[†], Nikolaos Laoutaris[†],
Dejan Kostić[⋆], Pablo Rodriguez[†], Rade Stanojević[‡⋆], and Pablo Yagüe Valentín[†]

[†]Telefonica Research, [⋆]EPFL, [‡]Institute IMDEA Networks

[†]{goma,alopezt,nikos,pablorr,payv}@tid.es, [⋆]{marco.canini,dejan.kostic}@epfl.ch,
[‡]rade.stanojevic@imdea.org

ABSTRACT

Access networks include modems, home gateways, and DSL Access Multiplexers (DSLAMs), and are responsible for 70-80% of total network-based energy consumption. In this paper, we take an in-depth look at the problem of greening access networks, identify root problems, and propose practical solutions for their user- and ISP-parts. On the user side, the combination of *continuous light traffic* and *lack of alternative paths* condemns gateways to being powered most of the time despite having Sleep-on-Idle (SoI) capabilities. To address this, we introduce Broadband Hitch-Hiking (BH^2), that takes advantage of the overlap of wireless networks to aggregate user traffic in as few gateways as possible. In current urban settings BH^2 can power off 65-90% of gateways. Powering off gateways permits the remaining ones to synchronize at higher speeds due to reduced crosstalk from having fewer active lines. Our tests reveal speedup up to 25%. On the ISP side, we propose introducing simple inexpensive switches at the distribution frame for batching active lines to a subset of cards letting the remaining ones sleep. Overall, our results show an 80% energy savings margin in access networks. The combination of BH^2 and switching gets close to this margin, saving 66% on average.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Access schemes*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design, Experimentation, Measurement

Keywords

Energy, Broadband access networks

[⋆]Work partly done when the author was with Telefonica Research.

1. INTRODUCTION

Recognizing the importance of improving the energy efficiency of the Information and Communication Technologies $(ICT)^1$, recent research efforts focused on reducing the energy consumption of datacenters [2, 3, 4], networks [5, 6, 7], and networked computers [8, 9, 10]. Out of the overall ICT energy expenditure, around 37% goes to powering telecommunication infrastructures [11]. The annual energy consumption figures reported by telecommunication companies are indeed staggering — Telecom Italia 2.1 TWh [12], France Telecom - Orange 3.7 TWh [13], Telefonica 4.5 TWh [14], Verizon 9.9 TWh [15]. In this paper, we seek energy saving opportunities in broadband access networks including: (i) on the user side, modem, wireless Access Point (AP), and router (hereafter collectively referred to as *gateway*), and (ii) on the ISP side DSLAM modems and line cards. Access networks consume 70-80% of the overall energy going into powering wired networks [16]. The above devices, despite being smaller than backbone/metro devices, are responsible for a major share of this fraction, due to their sheer number and high per bit consumption [17].

Insomnia at the user part. Like most ICT devices, access network devices are not energy proportional [18], *i.e.*, they consume close to maximum power even if only lightly loaded. Until the long term vision of energy proportional computing becomes reality, the most practical approach for cutting down on energy consumption is to implement Sleep-on-Idle (SoI) mechanisms which by now have become obligatory [19]. Herein, however, access networks have a problem that backbones and metro networks don't have – *they lack alternative paths*. This means that a household, or a small office connecting through a single Digital Subscriber Line (DSL) or cable connection can only power off its gateway when there is absolutely no traffic to be sent or received. This is possible when all terminal devices are powered off, but highly unlikely if some of them are online, and especially if the user is actively engaged.

Home gateways might take up to a minute to boot and synchronize their modem with the DSLAM, so traffic inactivity periods need to be sufficiently long. However, it is known [8, 10, 9], and we also demonstrate in Sec. 2, that most usages of the Internet (including leaving a machine idle

---

[1]While the energy consumption attributed to ICT might be smaller than in other areas like transportation, manufacturing, *etc.*, the ongoing exponential increase of Internet traffic on one side [1], and the successful efforts for reducing energy consumption in other societal systems on the other, will only increase the importance of greening ICT.
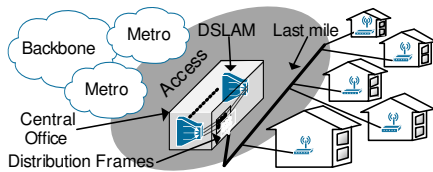
Figure 1: Illustration of a residential DSL broadband network. User gateways are connected to DSLAM ports via telephone lines in shared bundles up to few km long.

to maintain network presence) include at least some *continuous light traffic* that puts packets on the wire at time intervals that are much shorter than this. The problem is exacerbated by the fact that broadband connections typically serve multiple terminal devices like PCs, home entertainment systems and smart-phones, any of which can be injecting light, or not so light traffic.

Inflexibility at the ISP part. As shown in Fig. 1, the individual copper twisted pairs of nearby DSL subscribers are bundled in a common cable that travels all the way to the ISP central or regional office where individual pairs split again and terminate on the Main Distribution Frame (MDF) from where short local connections bring them to the Handover Distribution Frame (HDF) and ultimately to the ISP terminating modems. Each modem occupies a port in a line card and multiple line cards are controlled by a DSLAM. The insomniac state of the user part implies that ISP modems and line cards cannot sleep either. Later in the paper we will show that it is possible to put a large percentage of gateways to sleep yielding substantial energy savings for users. This implies that an equal percentage of ISP modems can also be powered off through SoI. The total energy gains however for an ISP are rather small. The reason is that a single ISP modem consumes around 1 W whereas the shared circuitry of the line card that hosts it consumes at least about 100 W. Line cards, however, are unlikely to sleep, even if they employ SoI. This is due to the inflexibility in terms of *lack of switching capability* at the HDF that permits a single active port to keep a card with 12-72 ports awake.

Our contributions and results. Our main contribution is a rigorous quantification of energy saving margins in access networks, and a breakdown of the gains between user and ISP parts. Overall, using a combination of trace-driven simulation and system prototype deployment (Sec. 5), we demonstrate that there exists an 80% energy saving margin at access networks. Straightforward techniques that do not require any expensive substitution of equipment can get close to this margin and save 66% of the current consumption. This will hopefully serve as a call to arms for ISPs and users to offer and use techniques in the spirit of those that we developed for demonstrating the claimed margins. Our specific technical contributions include the following:

• *Traffic aggregation:* We show that by aggregating user traffic to a minimum number of gateways it is possible to overcome the lack of alternative paths and the continuous light traffic problems and save up to 72% of the energy consumed by individual users without negatively impacting on their QoS. To demonstrate this, we introduce Broadband Hitch-Hiking (BH[2]), a distributed algorithm implemented in the driver of wireless home devices (Sec. 3). In dense urban areas where most users are, BH[2] permits devices to direct their light traffic to neighboring gateways within range, thus letting the local gateway to power off through SoI.

• *Impact on crosstalk[2]:* Apart from energy savings, BH[2] yields an additional – surprising – "bonus" for QoS. Owing to the broadband lines being powered off by BH[2], the remaining copper twisted pairs on the common cable connecting users to an ISP office are able to achieve higher data rates due to reduced crosstalk [20] (Sec. 6). Our detailed experiments using an Alcatel DSLAM with 24 VDSL2 modems and cable lengths from 50 up to 600 m show substantial gains – having half of the lines powered off gives the remaining ones a speedup of around 15% whereas powering off 75% of the lines increases the speedup to 25%.

• *Line switching:* Going over to the ISP side, we show that the probability of a line card powering off using SoI decreases exponentially with the number of modems it carries (Sec. 4). We propose solving this problem using switches at the HDF for terminating lines at different DSLAM ports depending on their state (active/inactive). We develop a model for deciding how large a switch needs to be and obtain the very positive result that even tiny 8×8 switches (8 DSL lines being able to rearrange between a fixed set of 8 ports on different line cards) permit batching together the active lines to a minimum number of line cards, letting the remaining ones sleep. Our experiments show that these simple switches allow us to power off a percentage of line cards that tracks well the percentage of gateways that BH[2] can power off.

Putting it all together, simple aggregation at the user part and switching at the ISP part can save 66% of total energy consumed in access networks. Extrapolating to all DSL users world-wide, assuming comparable link utilizations and wireless gateway density that we observe, the savings collectively amount to about 33 TWh per year, comparable to the output of 3 nuclear power plants in the US or equivalent to half of the energy used by US datacenters in 2006[3].

2. CHALLENGES IN GREENING ACCESS NETWORKS

DSL is the most widely deployed broadband access network technology (58% of broadband subscriptions as of June 2010 in OECD countries [22]). DSL works on top of ordinary twisted copper pairs used originally for telephony. DSL enables digital transmission between a modem at the user's premises and a second modem at the ISP (Fig. 1). ISP modems reside inside a DSL Access Multiplexer (DSLAM) which contains one such device for each serviced phone line. Similar to IP routers, a DSLAM is often designed as a shelf supporting a number of slots for line cards. A DSL line card typically services 12 to 72 lines. Since DSL is by far the most widespread access technology connecting over 320 million subscribers world-wide [21], and the one for which we have extensive datasets, we will focus our study around it. Notice, however, that our proposed techniques operate one level above the component level so they are applicable to other technologies as well. Next, we look at the reasons behind the high energy consumption of access networks.

2.1 Huge number of devices

The access is the only part of the network where there exists a direct proportionality between the number of net-

---

[2]The electromagnetic interference produced by other DSL lines packed closely in the same cable bundle.

[3]Analysis based on public data available at http://www.eia.gov, http://www.energystar.gov and in [21].

work devices and the number of customers. Indeed each subscriber has its own gateway and modem whereas a second modem is terminating every customer line at the DSLAM. Individual DSLAMs, despite servicing up to few thousands customers, are still by far the most numerous shared device of a network. In [16], it is shown that the number of DSLAMs is at least one order of magnitude more than metro devices and two orders of magnitude more than core devices.

2.2 Very high energy consumption per bit

Due to their sheer number, access devices need to be as cheap as possible. This requirement has an impact on several aspects including their efficiency in terms of energy consumption. Due to lack of energy proportionality [18], the lowest energy consumption per bit transmitted is achieved when a device is fully utilized. Using data sheets for different devices, researchers have computed the ratio of maximum transmission capacity to maximum energy consumption and reported the per bit consumption of devices at the different levels of a network [6, 17, 23]. From these reports it is clear that access devices consume two to three orders of magnitude more energy per bit transmitted than core devices.

2.3 Poor statistical multiplexing

Access devices not only have higher per bit consumption than other network devices under full load but have even worse per bit consumption under typical load. This is due to the fact that dedicated access devices are typically much less utilized than shared core devices (while always consuming close to maximum power [18]). Take for example an ADSL line. Its utilization depends almost exclusively on the behavior of a single customer (whether an individual or a family). Therefore, the amount of statistical multiplexing achieved is much lower than in higher levels of the network, *e.g.*, in an access or a core router. To get an idea of the level of utilization of typical ADSL lines, we plot in Fig. 2 the median and the average utilization of a set of 10K ADSL subscribers of a large commercial ISP providing subscriptions with 1-20 Mbps downlink, and 256 Kbps to 1 Mbps uplink. We observe a very low average utilization throughout the day that does not exceed 9% even during the peak hour as also noted by others [24, 25, 26]. Such levels of utilization are smaller by a multiplicative factor than the level of utilization of backbone links (typically 30-50%). This makes the average per bit consumption of access devices yet another multiplicative factor worse than their corresponding per bit consumption under maximum load.

2.4 Failure of Sleep-on-Idle

The above mentioned low utilization should make access devices a prime target for applying simple "Sleep-on-Idle" (SoI) techniques that power off devices when there is no traffic. Unfortunately, SoI is inhibited on access devices by the following two problems. First, as noted earlier *in the access there exist no alternative wired paths*. If a DSL line is put to sleep, the customer is effectively disconnected from the network. Disconnection is ok as long as the customer has no traffic to send or receive. This, however, brings us to the second problem – "*continuous light traffic*". We are referring to low average rate traffic (see Fig. 2) that however is constantly present as long as one or more terminal devices are on. Gateways and modems might take up to a minute to boot and synchronize and thus cannot sleep using SoI
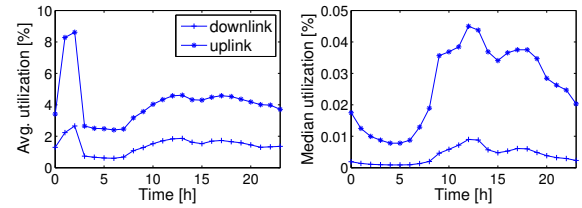

Figure 2: Daily average and median utilization of access links in a commercial ADSL provider (July 2009).
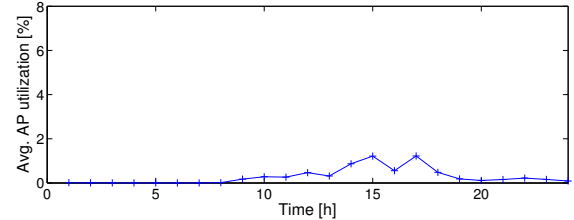

Figure 3: Average downlink utilization of access 6 Mbps links when using the UCSD CRAWDAD traces [27].

under such traffic. Web browsing, email, chat, or leaving a machine on to maintain online presence generate such light continuous traffic [8, 9, 10].

We want to illustrate this effect with data collected from real users. Given that packet-level traces of residential users are not publicly available, we use traces of wireless activity in the UCSD Computer Science building from [27] (see Sec. 5.1 for a detailed description). We assume for each AP in the building a backhaul bandwidth of 6 Mbps, which is the average downlink speed of the 10K residential ADSL subscribers presented in Fig. 2. We then compute the sum of inter-packet gaps in second-long bins (*i.e.*, 0-1 s, 1-2 s, *etc.*) for the peak hour (16-17 h), and show them in Fig. 4. We can see that for more than 80% of the time the inter-packet gaps are lower than 60 s, despite the utilization being as low as 1%. This continuous light traffic effectively condemns the SoI technique to a maximum saving of only 20%. Our results in Fig. 4 are strikingly consistent with the distribution of the inter-packet gaps of the non-publicly available dataset of residential traffic analyzed in Fig. 3 of [8]. In addition, the UCSD dataset matches very well with the aggregate utilization levels from 10K ADSL customers presented in Fig. 2. Therefore, we will use it later for our evaluation.

2.5 Summary

Access networks involve a huge number of devices that are particularly inefficient in terms of per bit energy consumption. Therefore it is not surprising that they end up consuming a very high percentage (70-80% [16]) of the total energy spent in networks costing ISPs millions per year. Powering off access networks is inhibited by the lack of alternative paths which implies that the single connecting path can only be put to sleep when there is no user traffic. This, however, is seldom possible due to continuous light traffic that does not permit SoI functionalities to be effective.

3. GREENING THE USER PART

In this section, we develop a simple technique for aggregating the traffic of multiple users to a sufficient subset of gateways letting the rest sleep. *Aggregation* is the key to solving the lack of alternative paths and continuous traffic
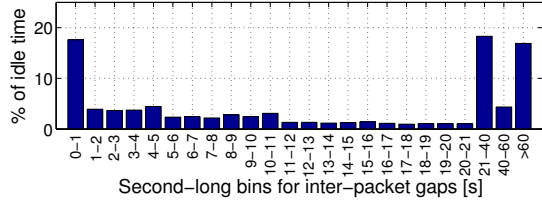
Figure 4: Histogram of the fraction of the idle time made up of inter-packet gaps of different size for the peak hour in Fig. 3.

problems discussed before. Currently, the simplest form of aggregation is via the wireless channel, taking advantage of the overlap of coverage area from multiple WiFi gateways in typical urban areas. In the future, femtocells or other wireless technologies can be used for the same purpose. Next, we formulate the problem and then present the design and implementation of a practical and efficient algorithm.

3.1 Formulation

Let U denote a set of users and let $d_i(t)$ denote the traffic demand of user $i \in U$ at time slot t. Let also G be a set of gateways and let c_j denote the capacity of gateway $j \in G$, *i.e.*, the speed of its broadband connection to the Internet. Additionally, with $w_{i,j}$ we denote the maximum available bandwidth between user i and gateway j due to the physical characteristics of the wireless channel. The parameters *backup* (a nonnegative integer) and $q \in (0,1]$ denote the minimal number of backup gateways each user must connect to in order to do smooth hand-offs (more on this shortly) and the maximum allowed utilization of a gateway for protecting the QoS of the local user, respectively.

To formulate the problem of minimizing the number of online gateways we define o_j, a binary decision variable that becomes 1 iff gateway j is online. We also define the variables a_{ij} that becomes 1 iff i's traffic is "assigned" (routed) to gateway j. Our optimization problem at time slot t can now be stated as the following binary integer program:

$$\underset{o_j, a_{i,j}}{\text{minimize}} \quad \sum_{j \in G} o_j \tag{1}$$

$$\text{subject to} \quad \sum_{j \in G} a_{ij} \geq 1 + backup, \quad \forall i \in U$$

$$d_i \cdot a_{ij} \leq w_{ij}, \quad \forall i \in U, j \in G$$

$$\sum_{i \in U} d_i \cdot a_{ij} \leq q \cdot c_j \cdot o_j, \quad \forall j \in G$$

It is easy to see that the decision version of the above problem is NP-complete (reduction from SET-COVER to a version of our problem with no backups and infinite capacity at gateways). Next, we describe BH^2, a simple distributed heuristic algorithm that we developed for the above problem. BH^2 runs on user terminals permitting them to direct traffic to remote gateways in range under certain conditions. The operation of the algorithm can be summarized easily by looking at the following two cases:

User connected to its home gateway: If the load of the home gateway falls below a *low threshold*, the algorithm looks for available remote gateways in range that are both not heavily loaded (*i.e.*, their load is below a *high threshold*), and not candidates for going to sleep (their load is above the *low threshold*). If the number of gateways that meet

the above conditions is greater than the minimum number of *backup*, the algorithm selects one randomly among them with a probability proportional to their load, and redirects its traffic there. The randomness in the selection is introduced to prevent synchronization. The use of *backup* gateways is introduced to allow users to perform smooth hand-offs if they need to leave the remote gateway.

User connected to a remote gateway: Similar to when it is at its home, if the load of the remote gateway falls below the *low threshold*, the algorithm looks for available gateways in range whose load is below the *high threshold* and above the *low threshold*. If the number of candidate gateways is enough to meet the backup requirement, it selects among them randomly with probability proportional to their load. If the minimum number of *backup* gateways cannot be met, or if the load of the assigned remote gateway increases above the *high threshold*, the algorithm returns the user to its home gateway, waking it up if necessary.

3.2 BH^2: Implementation

BH^2 needs the following to be in place: (i) gateways need to implement some form of SoI that puts them to sleep after a period of traffic absence; (ii) gateways need to wake up when traffic reappears; and (iii) terminal devices running BH^2 need to be able to estimate the load of all gateways in range in order to compare with the above mentioned thresholds and also be able to route traffic through the assigned gateway — all irrespective of the channel in which the gateways operate.

For (i), there already exist several implementations in the literature (see *e.g.*, [8]). Regarding (ii) technologies such as Wake-on-WirelessLAN (WoWLAN) [28] or Remote Wake Technology (RWT) [29] are readily available in popular hardware and operating systems. Note that to wake up a system using WoWLAN or RWT, it is necessary to know the MAC address of the target device. In BH^2 users can only wake their own home gateway, so knowing the MAC address is not a problem.

Finally, for (iii) we use the methods described in [30] and [31], that allow a user to be simultaneously connected to all the gateways in range. Briefly, the user's wireless card is virtualized, *i.e.*, it appears as independent virtual cards associated to each available gateway. The solution then is to rely on the standard 802.11 Power Saving (PS) mode to implement a Time-Division Multiple Access (TDMA) by sequentially cycling through the gateways in a round-robin fashion devoting enough time to the selected gateway to collect the bandwidth from its backhaul, and a small amount of time to the other gateways in order to be able to estimate their load. The load estimation employs a trick that relies on the fact that every 802.11 frame sent by a gateway carries a MAC Sequence Number (SN) in the header [30]. The user can listen periodically to the traffic sent by the monitored gateways and store its SNs. By counting the SNs, the amount of packets traversing the gateway backhaul, and hence its load, can be estimated.

3.3 Discussion

We implemented BH^2 on Linux laptops by only modifying the open source wireless card drivers, without requiring any explicit communication with gateways, or other terminals (see Sec. 5.3). To realize, however, the benefits of BH^2 a gateway needs to allow remote terminals to connect to it.
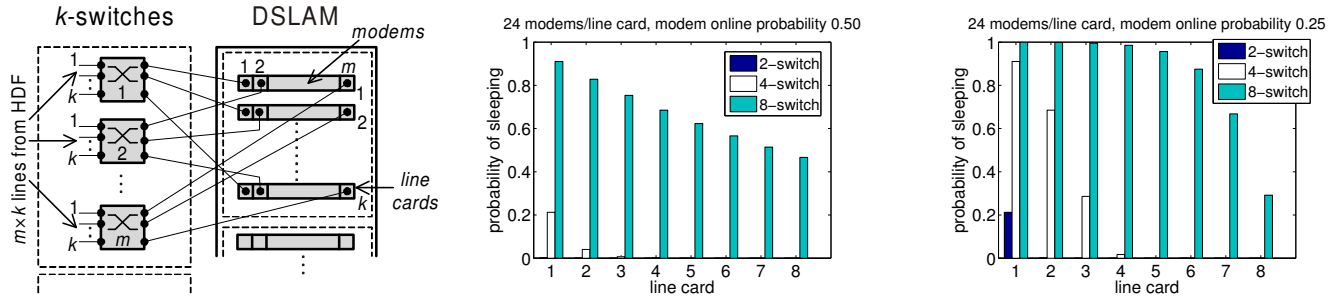
Figure 5: [left] k-switches and connection to DSLAM. [middle and right] Probability of line card 1,2,...,k sleeping on a batch of k line cards connected through m k-switches. Each line card has $m = 24$ modems and each modem is active with probability $p = 0.5$ (middle) or $p = 0.25$ (right).

In our testbed this is not an issue, but on a real deployment it implies the cooperation of the owner of the gateway.

The wide adoption of such technologies is clearly a non trivial matter since it involves incentive, security, and privacy issues. Our implementation surely does not solve such issues since it was developed merely for demonstrating the technical feasibility of BH[2] and to permit computing energy gains in a real setting. However our hope is that due to the important energy gains that we report, users and ISPs will find ways to realize them in practice. We are confident that this is possible. Security and privacy concerns can be addressed with off-the-shelf solutions that already exist and have been deployed commercially[4]. Similar to the case of collaborative downloading [32], we believe that the right incentives can be found since (i) there is no penalty for users from participating — they all save energy, the gains are almost balanced, and QoS is preserved as we show later, and (ii) ISPs or regulators can provide additional incentives if there is need (*e.g.*, ISPs could temporarily increase a user's backhaul capacity to allow large uploads or downloads). Finally, notice that although we have solved the aggregation problem in a distributed way, more centralized/coordinated techniques, potentially involving changes to the gateway, can be developed for offering strict accountability and strengthened security (see *e.g.*, [33, 32]).

4. GREENING THE ISP PART

Aggregation is the key primitive in reducing the consumption at the user part of access networks. In this section, we will argue that *switching* is the key primitive for saving energy on the ISP part by putting line cards to sleep. By switching we mean the ability to terminate a customer's twisted pair in a modem/line card of choice at the DSLAM instead of having it always fixed to the same port.

4.1 Problem description

Consider a line card carrying m modems and assume that in a certain time slot individual gateways terminating in these modems have traffic with probability p (independently of each other) and with probability $1-p$ they don't, and so can be powered off using SoI. Then, the probability that the entire line card can be put to sleep is $(1-p)^m$, *i.e.*, it *drops exponentially with the line card size* m. This means that even if instead of continuous traffic we had well behaved burst traffic that utilizes a link fully and then leaves it idle

[4]For example, FON uses double ESSID (one private, one shared), combined with RADIUS-based WPA authentication (http://www.fon.com).

for a long period, the probability that a 48 port card can sleep under just 5% utilization would still be only 8% – *i.e.*, it is highly unlikely that line cards will sleep using just SoI.

What can be done to improve this? If we knew in advance the traffic profile of users, *i.e.*, had $p_i(t)$ for each user $i \in U$ and each time slot t of the day, we would assign users i and j to the same line card if their profiles were similar ($p_i \approx p_j$) and to different ones otherwise. The rationale is that if a user is active we would like also the remaining users to be active (to fully utilize the line card) and the reverse (to be able to put it to sleep).

The above strategy has several practical problems including that p_i's are not known in advance, are changing, and that traffic is not well behaved but rather continuous as shown in Sec. 2. For these reasons, our proposal is to introduce switching at the HDF for being able to select dynamically based on state (on/off) the line card/port to which a line terminates. A full switching capability that allows any line to terminate at any port can power off $\lfloor n \cdot (1-p)/m \rfloor$ line cards, where n is the total number of ports of the DSLAM. Percentage wise this is asymptotically equal to the percentage of powered off gateways.

4.2 Our proposal: small k-switches are enough

A full switching capability maximizes the number of line cards that can be put to sleep but incurs a cost that depends on n, the number of ports of a DSLAM (can be 1000 or more). In this section, we argue that much smaller constant size switches (as low as 8×8) suffice for getting close to optimal performance.

We propose to use a series of k-switches each of which gets k lines from the HDF and terminates them at k modems at the DSLAM, allowing any mapping between lines and modems. We connect arbitrary lines to the switch at the HDF but take care to connect modems that belong to k different line cards at the DSLAM. As a simple convention (but this is not necessary or constraining), we assume that line cards are batched in groups of k (as shown in Fig. 5) and that a given k-switch connects to one modem at each line card at the same position (*e.g.*, 1th, 2nd, ..., m^{th}). The operation of the switch is simple — it checks the state of each line and when a line is inactive, it maps that line to the next free modem starting from the first line card and going down until it finds an unallocated port. Effectively the k-switch packs the inactive lines on the top part of its k positions and the active lines on its bottom part. This way, m k-switches try to batch the active lines on a minimum number of line cards out of the k that they cover. Of course, unlike with

a full switching capability, they might fail to do so, *e.g.*, if there is a switch whose k lines happen to be all active.

Next, we compute the impact of having these switches on the ability of line cards to power off through SoI. Assuming that individual lines are active with probability p independent of each other, it is easy to verify that the probability that the lth line card out of a set of k line cards can be put to sleep is:

$$P\{l^{th} \text{ line card sleeps}\}$$
$$= P\{\text{at least } l \text{ out of } k \text{ lines at every switch are inactive}\}$$
$$= (P\{\text{at least } l \text{ out of } k \text{ lines of a switch are inactive}\})^m$$
$$= \left(1 - \sum_{i=0}^{l-1}(1-p)^i p^{k-i}\right)^m$$
(2)

We can use Eq. (2) to select how big k needs to be in order to be able to put a good number of line cards to sleep given m (property of the line card) and p (performance of BH^2 under a given traffic load, topology, *etc.*). Fig. 5 shows that even very small switches of size $k = 4$ or 8 are in position to put to sleep a good number of line cards even in the case that BH^2 is able to turn off only half of the gateways ($p = 0.5$, something that is not at all uncommon according to the results shown later in Sect. 5.2.2). The above results are derived assuming independent traffic between users but as we will show experimentally later, such small switch sizes are indeed sufficient for reaping most of the benefits.

4.3 Discussion

Regarding the feasibility and the cost of installing 4- or 8-switches at the HDF, we note that switches of much greater size have already been constructed by large vendors, and used by several ISPs, albeit, for different applications than the one we envision here. For example, switches for Automated MDF (ADF) are available from companies like Network Automation[5] (ranging from $k = 20$ up to $k = 160000$) and Telepath Networks[6] ($k = 100$ up to $k = 100000$ with ms switching times). Given that the cost of switching depends on k, we believe that their cost will be more than covered by their contribution in achieving great reductions in energy consumption as we show next. Finally, we note that the power consumption of the switches themselves is negligible as these provide simple line switching, not datagram switching, and therefore do not need a packet processor or other complicated circuitry. They are simple micro-electromechanical relays that operate with near-zero power consumption [34] and have very low manufacturing costs [35].

5. EVALUATION

5.1 Evaluation methodology

Metrics: Our main quantitative evaluation metric is the total *energy savings* of the different schemes with respect to a *no-sleep* operation. Two indicators of performance are the number of gateways and the number of DSLAM line cards that each scheme can put to sleep. We also analyze how the schemes impact on the *completion time* of the network flows compared to the *no-sleep* scheme. As a measure of *fairness*,

we also look at the distribution of the sleeping time of the gateways compared to the basic *Sleep-on-Idle* scheme.

Traffic traces: We use the packet-level wireless traces from the CRAWDAD repository in [27]. The traces were obtained during the 24 hours of Thursday, January 11, 2007, by monitoring the activity of the wireless clients in the four-story UCSD Computer Science building. The traces contain packets of 272 clients accessing 40 APs. For simplicity we only consider downlink traffic. The dataset is detailed in [36].

Scenario: Since the traces do not contain topology information, we use the algorithm proposed in [37] to generate a wireless overlap topology. The resulting topology has node degrees that follow the distribution of per-household wireless networks in a residential area, as measured in [38]. The resulting average number of networks in range of a client is 5.6, consistent with previous studies (*e.g.*, [39]), and also with our own independent measurements. We uniformly distribute the 272 clients over the 40 gateways, and assign a wireless channel capacity of 12 Mbps between a client and its home gateway, and, based on the findings in [40], we assign half of that capacity (*i.e.*, 6 Mbps) between a client and gateways adjacent to the home gateway. As discussed in Sec. 2.4, we use ADSL speeds of 6 Mbps for consistency with our residential traces depicted in Fig. 2. On the ISP side, we consider a single DSLAM with 48 ports distributed in 4 line cards of 12 ports each. The gateways are connected to the ports randomly, as shown by our analysis of the distribution of the ADSL attenuations measured at two production DSLAMs covering more than 2K users in two major European cities (see Appendix). Since we only have 4 line cards, the k-switch schemes use 12 4-switches, where the ith 4-switch is connected to the ith port of each line card, following a configuration similar to the one in Fig. 5.

Power consumption: We performed detailed measurements of the power consumption of a Netgear WNR 3500L wireless router for different loads and distances from the clients. We obtained an average consumption of 5 W, with less than 10% of variation across the load range. We also measured the power consumption of a Telsey CPVA642WA ADSL gateway, and obtained a consumption of about 9 W mostly constant across all utilization levels. This implies that in terms of bits/Joule it is more efficient to operate the devices at higher utilization levels. We further measured the power consumption of the DSLAM we used in our experiments (reported later in Sec. 6). The ISAM 7302 datasheet reports for the shelf a typical consumption of 21 W and 53 W max. Each DSL line card is reported to consume typically 98 W, and a maximum of 112 W. We use the above figures as inputs for our evaluation.

Algorithms for comparison:

No-sleep: Users only connect to their home gateways. Gateways and line cards never go to sleep. This scheme represents today's regular residential network operation, and it is our baseline for comparison.

Sleep on Idle (SoI): Users only connect to their home gateways. Gateways go to sleep after an *idle timeout* (see below). When a gateway goes to sleep, the corresponding modem on the DSLAM also goes to sleep, and if all the modems in a line card are sleeping, the entire line card is put to sleep. When new traffic arrives there is a *wake-up time* (also see below), that includes the time that the gateway, the line
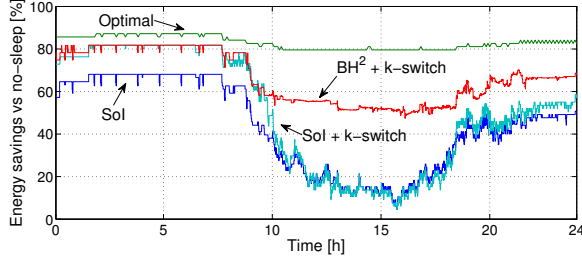
Figure 6: Energy saving vs. not sleeping.

Figure 7: Number of online gateways for the aggregation schemes.

card (if necessary) and the DSLAM modem take to wake up, and the time needed for the modem synchronization.

SoI + k-switch: Same as *SoI*, but DSL lines are connected to 12 4-switches as described before in the scenario. To prevent the disruption of active flows, the switching operations happen only when the gateway is being woken-up.

BH² + k-switch: Users employ the BH² algorithm described in Sec. 3.1. We use a *low threshold* and *high threshold* of 10% and 50% respectively. BH² decides which gateways to use every 150 s, with a random offset to prevent synchronizations. When a user is assigned to a new gateway, it routes all its new traffic to the new gateway. However, its existing flows are not dropped, but remain at the current gateway until they finish. If BH² has to wake up the user's home gateway in order to return to it, the user's traffic remains routed over the current remote gateway until the home gateway becomes operative. Similar to the *SoI + k-switch scheme*, the DSL lines are connected to 12 4-switches. Unless explicitly stated, BH² refers to the scheme with one backup.

Optimal: Every minute, the optimal assignment of users to gateways is computed by solving the centralized ILP of Eq. (1) and assuming that the users' flows can then instantaneously "migrate" to the optimal assignment. Also, the DSL lines are connected to a full-switch in the DSLAM, covering all the available ports. Every minute, all the active ports are switched optimally to minimize the number of active line cards. The switching "migrates" all active flows with zero downtime and no disruption. Note that *optimal* is certainly infeasible in practice with current technology, but represents a useful upper bound of the potential savings from aggregating traffic at the user side and packing active ports on the DSLAM line cards.

Wake-up time and idle timeout: We measured the *wake-up time* using several gateways connected to both commercial ADSL lines and our DSLAMs in the testbed, and we obtained an average time of 60 s. Note that the ADSL resynchronization can be as high as 3 minutes in some cases. Following a similar analysis with [9], we computed the idle timeout that has a low probability of putting the device to sleep right before a new packet arrives (and hence paying the *wake-up time* penalty). This is justified by the results of Fig. 4 showing that even in peak hours, roughly 82% of the inter-packet gaps are lower than 60 s.

Sensitivity analysis: We performed extensive sensitivity analysis and selected the parameters that provide the best performance in a wider range of situations — we tested both the convergence and stability of the BH² algorithm under different loads, by scaling up to 3 times up and down the DSL capacities. We tested a large range of the *low threshold* and the *high threshold*, and selected the ones that provided a good trade-off between convergence and stability. In par-
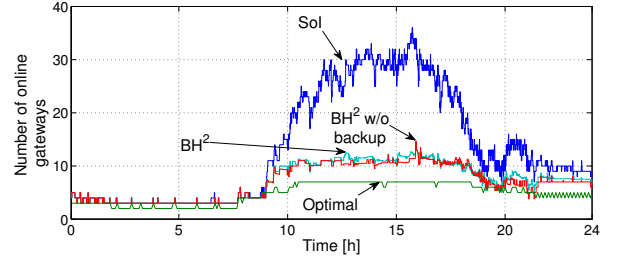
ticular, we saw that 50% utilization in the gateways was an accurate estimator of a future saturation (given the low load this does not happen often). Also, a 10% of *low threshold* absorbed most of the high frequency changes in load. We paid special attention to oscillations, and selected the values that minimized the number of gateway changes, especially those that required powering on the sleeping gateways. For the employed traces, executing BH² every 2.5 minutes and estimating load over 1-minute intervals achieved this goal.

5.2 Results

We evaluate the performance of the different algorithms over the scenario discussed above using simulation. For each scheme, we run the experiments 10 times and average the results for every second of the day. The simulation starts with all the gateways sleeping.

5.2.1 Energy savings

Fig. 6 shows the energy savings of the different schemes compared to *no-sleep* for the duration of the day. Some important observations:
• During off-peak hours, most schemes can achieve energy savings greater than 60%. However, *optimal* can consistently achieve 80% savings compared to *no-sleep*.
• During peak hours, both *SoI* and *SoI + k-switch* schemes suffer considerably, dropping to less than 20%.
• *BH² + k-switch* tracks the *optimal* much better, and achieves consistently at least 50% savings, even during peak hours.
• Focusing on *SoI* and *SoI + k-switch*, we see that unlike *SoI*, the k-switches allow *SoI + k-switch* to match the performance of *BH² + k-switch* during off-peak hours, but become ineffective during peak hours.

5.2.2 The effect of aggregation on the user side

To understand and interpret the above results we look deeper at gateway aggregation. Fig. 7 shows the number of active gateways during the course of the day for *BH²*, *SoI* and the *optimal* aggregation. Also, for the sake of comparison, we show *BH²* when one backup is required, and also when no backup is enabled. The main observations are:
• During off-peak hours, there is almost no traffic in our dataset so all schemes can carry the traffic with 3-4 gateways online out of the total 40.
• *SoI* powers on many gateways (up to 95% of them at 15h) during the peak hours.
• On the contrary, *BH²* manages to track closely the *optimal* in terms of active gateways, even during peak.
• Using a backup does not penalize performance in terms of number of active gateways.

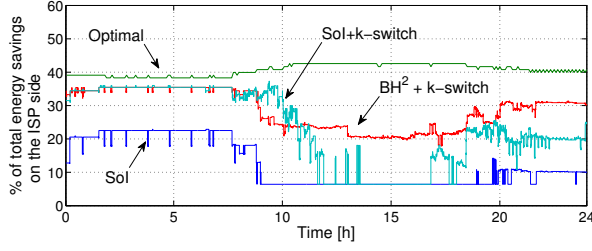The above results verify our main proposition that the

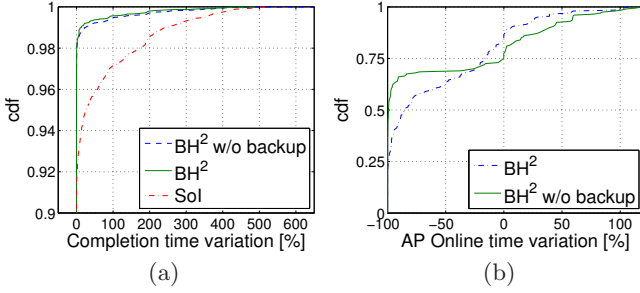**Figure 8: Contribution of the ISP part to the total energy savings.**

Figure 9: [left] **Increase in flow completion time vs.** *no-sleep*. [right] **Increase in gateway online time vs.** *SoI*.

lack of alternative paths combined with continuous traffic discussed in Fig. 4 do not permit *SoI* to be effective, despite operating under an average load of less than 2%. They also explain why *SoI + k-switch* can match the energy savings of $BH^2 + k\text{-}switch$ during off-peak hours – most user terminals are switched off so both of them use the same number of active gateways. Last, these results reveal why the *k*-switches do not make much difference when used with SoI (*SoI + k-switch*) during peak hours – *SoI* just fails to power off gateways therefore there is not much that the *k*-switches can do to power off line cards (the *p* of Sec. 4 is close to 1). Next, we show that the picture changes completely when *k*-switches are combined with BH^2 .

5.2.3 The effect of switching on the ISP side

Fig. 8 shows what percentages of the total savings of the various schemes correspond to the ISP side (DSLAM). We observe the following:
• Under *optimal* and $BH^2 + k\text{-}switch$, ISP-side savings due to switching are a substantial part of the overall savings, 40% and 30% (day average), respectively. This highlights that reaping the full energy savings requires actions at both the user and ISP sides.
• *SoI* saves very little for the ISP during peak hours as it only powers off terminating modems but no line cards. *SoI+k-switch* does only marginally better. The reason is that the *k*-switch does not have enough inactive lines at hand to be able to put entire cards to sleep.

Looking at the number of online cards, during off-peak hours we can see that all schemes can cope with just a single line card. During peak hours though, the average number of online cards varies significantly – *optimal*: 1, BH^2 +full-switch: 2, BH^2 +*k-switch*: 2.88, *SoI*+full-switch: 3, *SoI+k-switch*: 3.74, *SoI*: 3.99. Comparing BH^2 and *SoI* with full-switch and with 4-switch, we can verify also experimentally that even very small switches track closely the performance of full switching.
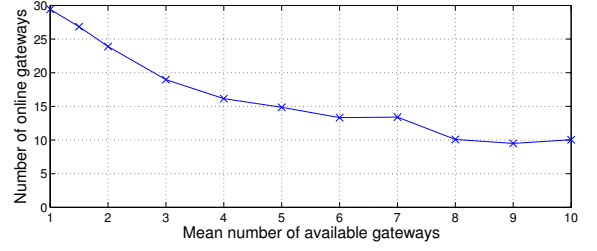

Figure 10: The impact of gateway density on the number of gateways that can sleep.

5.2.4 A look into QoS

An obvious question is whether powering off gateways and migrating to neighbors affects the QoS of the users. While the definition of QoS is ample, we focus as in [33] on whether the schemes increase the *completion time* of flows compared to *no-sleep*. Fig. 9a plots the CDF of the percentage of variation of flow completion time with respect to *no-sleep*. We can see the following:
• Even for *SoI*, only 8% of the flows see their completion time increased. Moreover, the increase can be as high as 7 times the original.
• BH^2 schemes perform much better, with as few as 2% of the flows being affected, and less heavily.
• Having a backup gateway slightly reduces the impact on completion time for BH^2 .

The few flows that see a large percentage-wise increase in their duration are short lived-flows (few seconds) that happen to coincide with waking up of a sleeping gateway, and thus get stretched by an additional 60 s. Finally, with low utilizations such as the one observed in our traces, having a backup gateway does not significantly change the behavior. We have observed, however, that as utilization increases, the positive effect of the backup is more noticeable.

5.2.5 The effect of gateway density

The results shown up to this point are all obtained using the same wireless overlapping topology and might raise the question of whether they are only valid for this gateway density. We assess the effect of the network density over the user aggregation capabilities of BH^2 running simulations where the mean number of networks that a user can connect to varies from 1 (*i.e.*, the user can only connect to the home gateway) to 10. We use a binomial distribution to generate the connectivity matrices with different mean number of gateways per user.

Fig. 10 shows the mean number of online gateways during the peak hours (from 11am to 7pm) versus the mean number of gateways users can connect to. As one might expect, the results show that the mean number of online gateways decreases with the increasing available gateway density. However, even in a low density deployment the number of online gateways is substantially reduced. For example, when users have just two neighboring gateways available on average, the number of online gateways is reduced to 19 (35% fewer powered-on gateways than when users can only connect to their home network).

5.2.6 Fairness

In this section, we examine whether the energy savings are shared in a fair manner among the different gateways. Fig. 9b shows the CDF of the variation in online time for
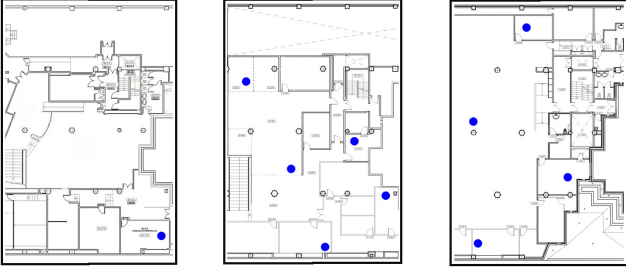
Figure 11: Testbed deployment. Gateways and terminals have been deployed over 3 floors, first floor [left], mezzanine [middle], and second floor [right]. Each circle represents a gateway, while terminals are placed nearby the gateways, one terminal per gateway. Obstacles, like walls and desks are present between all gateway links.

the gateways when using BH^2 compared to *SoI*. We want to see whether the traffic aggregation performed by BH^2 creates inequalities on the amount of online time a gateway experiences compared to running the simple *SoI* scheme (no change would be considered *fair*). We observe the following:
• As expected, BH^2 maintains a larger number of gateways always sleeping, hence the 25% of gateways with 100% decrease in online time.
• BH^2 increases the online time of 14% of the gateways compared to *SoI*.
• BH^2 without backup shows a more unfair situation, with several gateways completely eliminating their online time, and a larger number of them increasing it.

We see that using one backup gateway results in a more fair distribution of the sleeping times, while not harming performance (Fig. 7), therefore we opt for keeping it.

5.3 Realistic deployment

Testbed description: We deployed a testbed spanning three floors of a multi-story building. The testbed consists of 10 commercial 3 Mbps ADSL subscriptions with their corresponding gateways and 10 BH^2 terminals, *i.e.*, the "owners" of each line. The gateways are distributed approximately every 850 sq. ft. to emulate an average residential apartment size (see Fig. 11) and are randomly set to independent radio-frequencies in the 2.4 GHz ISM band. Similar to our evaluation scenario, each BH^2 terminal is in range of approximately 5.5 gateways and can communicate over the wireless channel at an average speed higher than 6 Mbps.

BH^2 implementation details: We implemented the BH^2 algorithm described in Sec. 3.1 on Linux laptops equipped with a single-radio Atheros-based wireless card. The BH^2 algorithm is implemented in the MadWiFi 0.9.4 driver [41] and the Click modular router 1.6.0 [42]. BH^2 terminals communicate with gateways at different radio-frequencies using the TDMA techniques described in Sec. 3.2. During the time BH^2 is connected to a gateway, it transmits and receives traffic according to the standard 802.11 DCF protocol. BH^2 uses a TDMA period of 100 ms, of which 60% is devoted to the gateway currently selected by BH^2, and the rest is distributed evenly among the rest of the gateways in range to collect their utilization statistics[7]. To be transparent to the applications, BH^2 implements a Reverse-NAT module that

---

[7]We verified that 60% of the wireless card's time is enough to collect all the bandwidth available at any gateway, since wireless capacities are higher than ADSL backhaul speeds.
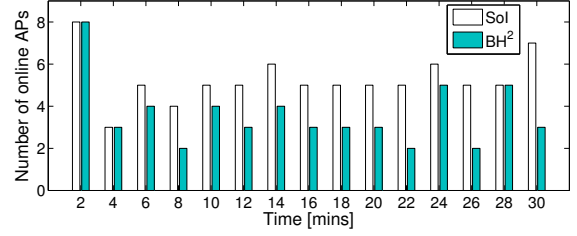

Figure 12: Number of online APs for a live experiment in the testbed in Fig. 11.

ensures packets leave the terminals with the correct source IP address, while exposing a single *dummy* IP address to the upper layers [30].

Gateways: We use off-the-shelf Linksys DD-WRTv24, running unmodified firmware. All wireless routers and terminals have the wireless multimedia extensions and the RTS/CTS handshake disabled. Any non-standard 802.11 feature is also disabled, and H/W queues are set up with 802.11 best effort parameters. Each wireless router is connected to a Zyxel P-600 modem that provides the ADSL connectivity.

Methodology: We use the traces described in Sec. 5.1 as our source of data. For each flow, we record the timestamp t and the amount of bytes b reported in the traces and we *replay* it: at the specified time t the terminal makes a *HTTP request* to download b bytes of the DVD image of a very popular Linux distribution from the local national repository.

Since the testbed has just a few clients, each BH^2 terminal replays the flows of all clients that are *originally associated* with one of traced APs selected at random. When the BH^2 algorithms selects a new gateway, the terminal starts routing the new flows through it. However it does not modify the existing downloads, *i.e.*, they will continue through the same gateway until finished.

The BH^2 algorithm runs independently and in a totally distributed manner. Each BH^2 terminal wirelessly monitors the load in the gateways as described in Sec. 3.2 and takes independent decisions based on the *low* and *high thresholds*. However, since our gateways do not have any SoI capabilities, they do not actually go to sleep. Instead, we emulate the state of the gateway as follows: a script running in a central server monitors the load of the gateways and flags them as "sleeping" when the idle timeout expires. Each BH^2 terminal checks the status of the gateway in the server via an independent local area network. If a terminal decides to "wake-up" a gateway, it changes the status on the server as "waking-up". The server automatically updates the status to "active" after the appropriate *wake-up time*.

Results: We conducted numerous experiments to verify the correct operation of BH^2. Specifically, we used the BH^2 laptops in browsing, YouTube video streaming, BitTorrent and even P2P live video streaming sessions. We did not experience performance problems (*i.e.*, glitches, video rebuffering or choppy audio) even after several gateway changes.

To validate BH^2 performance, we made 10 independent experiments that *replay* the traces using 9 laptops, each of them having one of the 9 gateways of Fig. 11 as their "home" gateway. In each run we randomly assign one of the APs of the CRAWDAD traces to a gateway in our testbed. The corresponding laptop replays all the clients in the traces that were originally associated with the AP represented by that gateway. Our testbed allows a client to connect to a max-

imum of 3 gateways. Fig. 12 shows the number of active gateways from 15:00 to 15:30 h for BH^2 without a backup and *SoI*. We observe the following:

• Of the 9 gateways, on average BH^2 puts 5.46 to sleep (60%), while *SoI* only puts to sleep 3.72 (41%).

• BH^2 consistently outperforms *SoI* at all times, even for the small load of our traces and 3-gateway limitation we imposed in our implementation.

These experiments show that in our realistic experiments, BH^2 yield energy savings that *doubles* those of *SoI*, consistent with the results we reported earlier through simulation.

5.4 Summary

The results of this section have demonstrated that there is an 80% margin for energy savings in access networks. Simple aggregation and switching techniques like BH^2 + *k-switch* can save 66% on average, of which 2/3 go to users and 1/3 to the ISP. Extrapolating to all DSL users world-wide, the savings collectively amount to about 33 TWh per year.

6. A CROSSTALK BONUS

Apart from the energy gains, the aggregation effect of BH^2 permits modems to lock at higher speeds due to lower crosstalk. In this section, we present a number of experiments with a real DSLAM and copper lines to demonstrate that the speedup can be as high as 25%.

6.1 Crosstalk

Crosstalk [20] refers to the electromagnetic coupling between lines in the same cable bundle. Crosstalk increases with attenuation ($\sim$cable length) and signal frequency. It also depends on the distance between lines inside the bundle and it is worst for adjacent lines (*e.g.*, 1 and 2 in Fig. 13a).

To deal with varying conditions of crosstalk, ADSL and VDSL adapt the frequency plan to the line length and crosstalk noise. To do so, there are two options while initializing the connection: (i) maximize the bit rate subject to the currently sensed line conditions and crosstalk while leaving a safe margin of at least 6 dB, or (ii) maximize the noise guard margin while having a bit rate fixed (usually set according to the subscribed plan). Once the connection is established it is being monitored and adjusted whereas re-synchronization occurs if the noise margin falls to 0 dB.

6.2 Experimental setup and methodology

Our testbed consists of an Alcatel 7302 ISAM DSLAM equipped with a 48-port, NVLT-C line card and 24 VDSL2 modems[8]. Each modem is connected through a cable bundle of 25 twisted pairs (Fig. 13a) to a switchboard that allows us to vary the length of the twisted pair connecting the modem to the DSLAM, as illustrated in Fig. 13b.

We measure the actual bit rate as we vary the number of active lines using the following methodology. First, we define 5 random orders in which to activate the 24 lines. The sequences activate 4 lines at a time up to 12 lines, and then 2 at a time up to 24 lines. At each step in a sequence, we activate certain lines and force each one to resynchronize, one at a time in random order.

We use two different line length setups and two different service profiles for a total of four set of experiments. Specifically, we experiment with a fixed line length of 600 m for

---

[8]21 modems are Huawei HG520v, other 3 are Zyxel P-870HW.

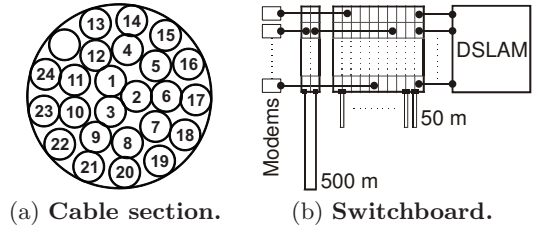(a) **Cable section.**   (b) **Switchboard.**
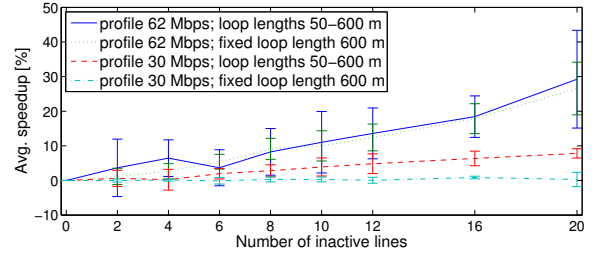
Figure 13: Diagrams of the measurement testbed.

Figure 14: **Average speedup as more lines become inactive. Standard deviations are plotted as error bars. Following the profile order of the legend, the baselines are at 41.3, 43.7, 27.8, and 29.7 Mbps.**

all lines and with line lengths chosen to match a real distribution of lengths between 50 and 600 m as given to us by a large telco. We use two different service profiles: (i) the first with plan downstream bit rate of 30 Mbps, (ii) the second with plan downstream bit rate of 62 Mbps.

6.3 Results

Based on the average bit rate measured as we vary the number of active lines, we compute the average per-line speedup as the relative bit rate gain w.r.t. the baseline rate of having all lines active. Fig. 14 presents the average and standard deviation of the per-line performance increase, computed over all random sequences each of which is measured twice to account for the non-deterministic nature of the measured medium.

Considering the profile at 62 Mbps, it can be seen that there is an increase in bit rate of 1.1-1.2% for each modem that becomes inactive. When half of the modems are off, the remaining modems can obtain 13.6% more bandwidth, whereas when increasing the powered off modems to around 75% the speedup climbs to 25%.

7. RELATED WORK

Greening networks. Improving the energy efficiency of the Internet is currently a very active area of research (see [43] for a detailed survey). Most of the efforts have focused on backbone networks. For instance, Chabarek *et al.* [6] presented an energy-aware network design methodology. Vasić *et al.* [7] proposed a traffic engineering scheme that matches active network devices to the current traffic load. Heller *et al.* [44] discussed a similar approach for datacenter networks.

Arguing for hardware support of energy-aware capabilities, Nedevschi *et al.* [5] quantified the savings that can be achieved by shaping traffic into bursts to enable putting network components to sleep during idle times and by adapting the rate of network operation to the offered load. Applying such techniques to DSL is not straightforward due to two main issues: (i) putting a DSL modem to sleep would tear

down the connection and restoring it with the current DSL specifications would require a new synchronization phase that can last tens of seconds, and (*ii*) rate adaption may lower the transmission power to the point where crosstalk from other active lines becomes a severe impairment.

Greening access networks. Approaches for saving energy in access networks include those that advocate network re-engineering and works that improve energy efficiency of broadband technologies. Bianco *et al.* [45] provided an analysis of the energy consumption trends in Telecom Italia and discussed the deployment of VDSL2 in FTTx architectures. Tsiaflakis *et al.* [46] and Guenach *et al.* [47] suggested a reformulation of dynamic spectrum management for minimizing power consumption of DSL. These works showed that adding objectives or constraints for limiting the transmit power in DSL modems ultimately reduces date rate performance and may render active lines less stable.

Looking at the enterprise environment, Jardosh *et al.* [33] proposed an approach for powering off APs in enterprise wireless networks. Although the underlying ideas and objectives are similar to BH^2, the two settings are substantially different. Jardosh *et al.* assumed a centralized controller that can map terminals to APs in a setting where APs communicate through a shared backbone. In our case, assuming a centralized controller would be challenging in terms of both scalability and amount of change to the existing architecture. However, we consider this work complementary and some of its techniques may be utilized to provide a centralized solution to the problem BH^2 solves.

Exploiting neighbor's wireless. Building on the observation of increasingly frequent high-density wireless deployments, several schemes have been proposed to aggregate bandwidth from neighboring wireless networks. Systems like FatVAP [31] and THEMIS [30] demonstrated the feasibility of aggregating the backhaul capacity of multiple APs using a single virtualized wireless card. PERM [38] discussed how to schedule flows across accessible APs based on predicting of flow round-trip time or traffic volume. CUBS [26] and Link-alike [39] focused on sharing the idle upload bandwidth of neighbors to improve P2P applications and file uploads. COMBINE [32] presented a system for collaborative downloading and specified a framework of incentives that could also help address several practical issues in our proposal. While different in their specifics all of them are in essence trying to maximize the transmission capacity. Our work is dual to that objective as it attempts to minimize the committed energy for carrying a given amount of traffic. At the same time, BH^2 takes advantage of some of these techniques to carry traffic on different gateways.

Network connectivity proxies. Tackling the problem of reducing power consumption from a different angle, several works (*e.g.*, Nedevschi *et al.* [8], Reich *et al.* [9], Agarwal *et al.* [10]) argued that putting end hosts to sleep yields substantial energy savings that, however, are difficult to obtain because users keep unused end hosts fully powered-on only to maintain their network presence. They suggested using network connectivity proxies that allow idle hosts to sleep and still maintain full network presence. Unlike our proposal, such proxies cannot be used when the user is actively using its terminal. By maintaining the connections, they also eliminate energy saving opportunities at the ISP.

8. CONCLUSION

The biggest fraction of wired network energy is consumed by the access network, even though the individual access devices themselves have small power requirements. Unfortunately, straightforward techniques for sleeping during idle periods are inefficient in this environment because of the continuous lightweight traffic, and the absence of alternative paths that could carry the traffic.

In this paper, we formalize the problem of saving energy in the predominantly DSL-based access networks, and describe two simple techniques: (*i*) wireless user traffic aggregation that enables access devices to firmly sleep, and (*ii*) switching at the ISP side that significantly increases the number of power-hungry line cards that can sleep. Our thorough evaluation using trace-based simulation and a live prototype shows that for typical urban settings it is possible to save 66% of access network energy on average, while being within 7-35% of the computationally-hard optimal aggregation and impractical optimal switching. Moreover, our results show that both of our proposed techniques significantly contribute to the overall savings, which argue for their individual importance. Finally, we demonstrate using live experiments that energy-saving in the access network brings a surprising speedup in achievable access link bit rates due to reduced crosstalk.

9. REFERENCES
 [1] Cisco Visual Networking Index: Forecast and Methodology, 2008-2013., 2009.
 [2] K. Church, A. Greenberg, and J. Hamilton. On Delivering Embarrassingly Distributed Cloud Services. In *Proc. of HotNets-VII*, 2008.
 [3] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez. Greening the Internet with Nano Data Centers. In *Proc. of CoNEXT'09*, 2009.
 [4] S. Nedevschi, S. Ratnasamy, and J. Padhye. Hot Data Centers vs. Cool Peers. In *Proc. of HotPower'08*, 2008.
 [5] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In *Proc. of NSDI'08*, 2008.
 [6] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *Proc. of INFOCOM'08*, 2008.
 [7] N. Vasić and D. Kostić. Energy-Aware Traffic Engineering. In *Proc. of e-Energy'10*, 2010.
 [8] S. Nedevschi, J. Chandrashekar, J. Liu, B. Nordman, S. Ratnasamy, and N. Taft. Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems. In *Proc. of NSDI'09*, 2009.
 [9] J. Reich, A. Kansal, M. Gorackzo, and J. Padhye. Sleepless in Seattle No Longer. In *Proc. of USENIX ATC'10*, 2010.
[10] Y. Agarwal, S. Savage, and R. Gupta. SleepServer: A Software-Only Approach for Reducing the Energy Consumption of PCs within Enterprise Environments. In *Proc. of USENIX ATC'10*, 2010.
[11] The Climate Group. SMART 2020: Enabling the low carbon economy in the information age, 2008.

[12] Telecom Italia 2009 Sustainability Report.

[13] France Telecom 2009 Corporate Responsibility Report.

[14] Telefonica 2009 Corporate Responsibility Report.

[15] Verizon 2009-2010 Corporate Responsibility Report.

[16] R. Bolla, R. Bruschi, K. Christensen, F. Cucchietti, F. Davoli, and S. Singh. The Potential Impact of Green Technologies in Next-Generation Wireline Networks – Is There Room for Energy Saving Optimization? *IEEE Communications Magazine*, 2011.

[17] R. S. Tucker. Modelling Energy Consumption in IP Networks. In *Cisco Green Research Symposium*, 2008.

[18] L. A. Barroso and U. Hölzle. The Case for Energy-Proportional Computing. *Computer*, 40(12):33–37, 2007.

[19] European Commission. Code of Conduct on Energy Consumption of Broadband Equipment ver. 3, Nov 2008.

[20] P. Golden, H. Dedieu, and K. Jacobsen. *Fundamentals of DSL Technology*. Auerbach Publications, 2005.

[21] Point topic. World Broadband Statistics: Short report Q3 2010. http://point-topic.com/content/dslanalysis/.

[22] OECD Broadband Portal. http://www.oecd.org/sti/ict/broadband.

[23] U. Lee, I. Rimac, and V. Hilt. Greening the Internet with Content-Centric Networking. In *Proc. of e-Energy'10*, 2010.

[24] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing Residential Broadband Networks. In *Proc. of IMC'07*, 2007.

[25] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *Proc. of IMC'09*, 2009.

[26] E. Tan, L. Guo, S. Chen, and X. Zhang. CUBS: Coordinated Upload Bandwidth Sharing in Residential Networks. In *Proc. of ICNP'09*, 2009.

[27] Y.-C. Cheng. CRAWDAD trace ucsd/cse/tcpdump/wired (v. 2008-08-25), 2008.

[28] Microsoft Technet. Configure Wake on Wireless LAN. http://technet.microsoft.com/en-us/library/ee851581(WS.10).aspx.

[29] Intel. Remote Wake Technology. http://www.intel.com/support/chipsets/rwt/.

[30] D. Giustiniano, E. Goma, A. Lopez Toledo, I. Dangerfield, J. Morillo, and P. Rodriguez. Fair WLAN Backhaul Aggregation. In *Proc. of MobiCom'10*, 2010.

[31] S. Kandula, K. C.-J. Lin, T. Badirkhanli, and D. Katabi. FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput. In *Proc. of NSDI'08*, 2008.

[32] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath. COMBINE: Leveraging the Power of Wireless Peers through Collaborative Downloading. In *Proc. of MobiSys'07*, 2007.

[33] A. P. Jardosh, K. Papagiannaki, E. M. Belding, K. C. Almeroth, G. Iannaccone, and B. Vinnakota. Green WLANs: On-demand WLAN Infrastructures. *Mobile Networks and Applications (MONET)*, pages 1–26, 2009.

[34] G. M. Rebeiz. RF MEMS switches: status of the technology. In *Proc. of TRANSDUCERS'03*, 2003.

[35] J. Bouchaud, B. Knoblich, H. Tilmans, F. Coccetti, and A. EI Fatatry. RF MEMS roadmap. In *Proc. of EuMIC*, 2007.

[36] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benkö, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker. Automating Cross-Layer Diagnosis of Enterprise Wireless Networks. In *Proc. of SIGCOMM'07*, 2007.

[37] F. Viger and M. Latapy. Efficient and Simple Generation of Random Simple Connected Graphs with Prescribed Degree Sequence. In *Proc. of COCOON'05*, 2005.

[38] N. Thompson, G. He, and H. Luo. Flow Scheduling for End-host Multihoming. In *Proc. of INFOCOM'06*, 2006.

[39] S. Jakubczak, D. G. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan. Link-alike: Using Wireless to Share Network Resources in a Neighborhood. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12(4), 2008.

[40] D. Han, A. Agarwala, D. G. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan. Mark-and-Sweep: Getting the "Inside" Scoop on Neighborhood Networks. In *Proc. of IMC'08*, 2008.

[41] Madwifi project. http://madwifi-project.org.

[42] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. M. Kaashoek. The Click Modular Router. *ACM Trans. Comput. Syst.*, 18(3):263–297, August 2000.

[43] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Communications Surveys and Tutorials*, 13(4), 2011.

[44] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. ElasticTree: Saving Energy in Data Center Networks. In *Proc. of NSDI'10*, 2010.

[45] C. Bianco, F. Cucchietti, and G. Griffa. Energy consumption trends in the Next Generation Access Network – a Telco perspective. In *Proc. of INTELEC'07*, 2007.

[46] P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen. Green DSL: Energy-Efficient DSM. In *Proc. of ICC'09*, 2009.

[47] M. Guenach, C. Nuzman, J. Maes, and M. Peeters. On Power Optimization in DSL Systems. In *GreenComm'09*, 2009.

APPENDIX

We investigate if there is any correlation between the geographical distance of two customers and the relative distances of their ports in the DSLAM. We collect information from two production ADSL2+ DSLAMs of a large ISP in two major cities. We discuss one of them, since both yield similar results. Fig. 15 shows the distribution of the port attenuations measured in each of the 14 active line cards of 72 ports each. We plot the attenuations from n to $n + 100$, where n is not disclosed to protect sensitive information.

It is important to know that the attenuation measured for a line indicates the reduction in signal strength on that line, and represents the natural deterioration of the signal over distance from the MDF. We assume that there are no other external sources of attenuation, such as poor/unequal quality of the copper wires or the MDF connections. The production DSLAMs we measure are being actively maintained, and the ISP routinely discards lines that show subpar quality. In ADSL2+, a difference of 1dB in attenuation corresponds to a cable length of roughly 230 feet (70 m). Fig. 15 shows that across all line cards the attenuation has a similar Gaussian distribution with a standard deviation of one mile with minimal variations in mean. Given this randomness in the attenuation distribution, we assume that the assignment of gateways to DSLAM ports is random, and does not depend on their geographical proximity.
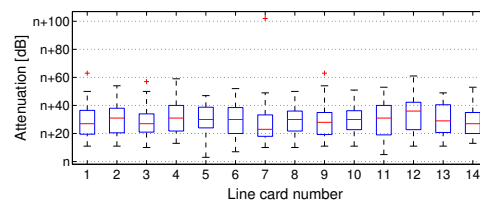

Figure 15: Distribution of the attenuations observed in a production DSLAM of a large ISP.