ClubADSL: When Your Neighbors are Your Friends

Domenico Giustiniano*, Eduard Gomà Llairo*, Julian Morillo Pozo$^+$, Alberto Lopez Toledo*, Pablo Rodriguez*

Telefonica Research, Barcelona, Spain; $^+$Universitat Politècnica de Catalunya, Barcelona, Spain.

***Abstract*—ADSL is becoming the standard form of residential and small-business broadband Internet access due to, primarily, its low deployment cost. These ADSL residential lines are often deployed with 802.11 Access Points (AP) that provide wireless connectivity. Given the density of ADSL deployment, it is often possible for a residential wireless client to be in range of several other APs, belonging to neighbors, with ADSL connectivity. While the ADSL technology has showed evident limits in terms of capacity (with speeds ranging 1-10 Mbps), the short-range wireless communication can guarantee a much higher capacity (up to 20 Mbps). Furthermore, the ADSL links in the neighborhood are generally under-utilized, since ADSL subscribers do not connect 100% of the time. Therefore, it is possible for a wireless client to simultaneously connect to several APs in range and effectively aggregate their available ADSL bandwidth.**

In this paper, we introduce ClubADSL, a wireless client that can simultaneously connect to several APs in range on different frequencies and aggregate both their downlink and uplink capacity. ClubADSL is a software that runs locally on the client-side, and it requires neither modification to the existing Internet infrastructure, nor any hardware/protocol upgrades to the 802.11 local area network. We show the feasibility of ClubADSL in seamlessly transmitting TCP traffic, and validate its implementation both in controlled scenarios and with current applications over real ADSL lines. In particular we show that a ClubADSL client can greatly benefit from the aggregated download bandwidth in the case of server-client applications such as video streaming, but can also take advantage of the increased upload bandwidth greatly reducing download times with incentive-based P2P applications such as BitTorrent.

I. Introduction

Wireless local area networks (WLANs) were traditionally envisioned with the goal to increase the coverage range for connecting to the Internet. As a typical example, home wireless connection is nowadays a standard “de-facto” for residential ADSL subscriptions, extending the Internet access provided by the ADSL line. These lines, however, are generally low speed connections and hence represent the bottleneck in the end-to-end communication [1].

On the other hand, it is well known that wireless connectivity to the Access Point (AP) is generally at higher speed with respect to the ADSL wired connection. These APs, commonly based on the IEEE 802.11b/g standard, provide high speed wireless connectivity that can achieve up to 20 times the speed of the ADSL lines they are connected to. Furthermore, the density of ADSL deployments with wireless connectivity tends to be high, especially in urban areas, as can be confirmed by a simple scanning with a WLAN-enabled station [2]. These ADSL links in the neighborhood are generally under-utilized, since users are not connected 100% of the time.

As a consequence, it is theoretically possible to simultaneously connect to several APs and aggregate their available bandwidth, leveraging in the fact that the high speed wireless card needs to connect to each ADSL line for a short period of time in order to collect all the pending data.

In this work, we present ClubADSL, a wireless client that can connect to the APs that are in the client coverage range, and aggregate their available ADSL bandwidth. With ClubADSL, the client is able to associate to multiple APs, independently of their operative frequency. ClubADSL works without any modification to the existing infrastructure, it is a software-only add-on running on the client-side and does not require any hardware/protocol upgrades.

We show the feasibility of ClubADSL in seamlessly transmitting TCP traffic, and validate its implementation both in controlled scenarios and with current applications over real ADSL lines. In particular we show that a ClubADSL client can greatly benefit both from i) the aggregated download bandwidth, in the case of server-client applications such as video streaming, and ii) the increased upload bandwidth, greatly reducing download times with incentive-based P2P applications such as BitTorrent.

The rest of this manuscript is organized as follows. Section II gives the related work and Section III presents the ClubADSL architecture and implementation. Section IV explores the limits of our ClubADSL implementation in a controlled environment. Section V shows the performance of ClubADSL over actual ADSL lines. Finally Section VI presents the conclusions.

II. Related work

The idea of connecting to multiple APs through one single radio was first presented in [3]. The authors rely on the Power Save mode feature of the WLAN standard to switch among different Wi-Fi nodes (APs or Ad-hoc) in a Time Division Multiple Access (TDMA) fashion. A client can inform the node that it is going into PS mode — so that it can buffer packets directed to the client — and switch radio-frequency to other nodes, only to come back to the original node before the PS period expires. Switching between networks was transparent to the applications, but at a high cost (30-600 msec). Similarly, [4] dedicates a constant 10% of time for scanning procedures or supporting a seamless hand-off between WLANs.

On top of the Power Save feature, [5] focuses on the connection to multiple APs and selects the percentage of

Alberto Lopez Toledo is supported by the Institució Catalana de Recerca i Estudis Avançats (ICREA). This work was completed while Julian Morillo Pozo was an intern at Telefonica Research.

978-1-4244-4671-1/09/$25.00 ©2009 IEEE

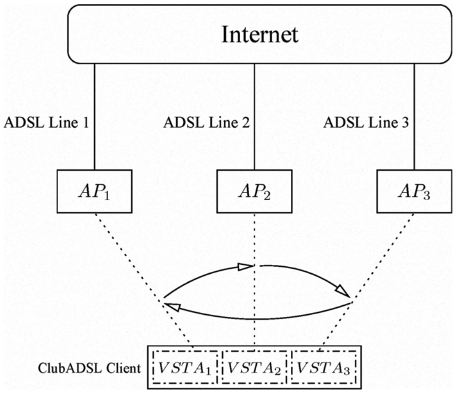

Fig. 1. Example of ClubADSL topology

connection on each AP in order to maximize the aggregate throughput based on the Knapsack algorithm. The system has an average switching cost of 2.8 msec plus another 2.8 msec of standard deviation.

III. Architecture

Fig. 1 shows an example of a ClubADSL client connected to 3 APs. In the scenario, AP_1 may represent the home AP while AP_2 and AP_3 the neighborhood APs. In ClubADSL, the wireless driver on top of the single-radio card is *virtualized*, i.e., appears as independent Virtual STAtions (*$VSTA_i$*) associated to their respective Access Point AP_i. Each of these $VSTA_i$ connects to Internet via its backhaul AP_i, and independently of the AP frequency.

ClubADSL selects in a TDMA fashion the APs to connect to and assigns the control of the card to a $VSTA_i$ for a given time, called *duty cycle*. During the *duty cycle*, $VSTA_i$ transmits/receives frames over the backhaul AP while the other *VSTAs* (and the corresponding *APs*) can only buffer packets. The sum of the *duty cycles* gives the amount of time to transmit/receive on each $VSTA_i$ and re-connect to the first virtual station, and is called *wireless period* (see Fig 2).

In the rest of this Section we describe the architectural design of ClubADSL at both network and MAC layers.

Network Layer The ClubADSL routing functionalities are implemented using the Click modular router [6]. In Click, elements can be flexibly configured to perform tasks for packet processing such as packet classification and scheduling, and then connected together in a flow graph to compose a protocol. Click features a simple tunneling mechanism that allows protocols to interface with the standard Linux TCP/IP stack. Thus, any application built on TCP/IP can be used with the ClubADSL client. Using the Click framework, we built the ClubADSL network layer with the following elements:

- A **load balancer** that splits traffic across the different $VSTA_i$ on a per-flow base. A flow is identified by its IP address and its port. When a new flow arrives, ClubADSL decides which AP to assign it in order to satisfy the ratios given by the percentage of connection on each AP_i[1]. Then it records the assignment in a hash table so that subsequent packets will be mapped to the AP based on the information stored in the table [5].
- A **reverse-NAT module** that assures that the packets leaving the host do that with the correct source IP address (i.e. the one corresponding to the outgoing $VSTA_i$) and that the incoming packets are presented to the OS with the expected IP address.

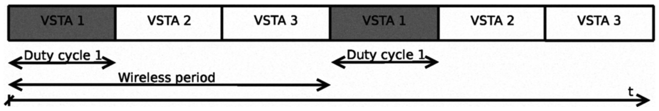

Fig. 2. Relation between *duty cycle* and *wireless period.*

MAC Layer The ClubADSL MAC layer driver is based on the latest stable MadWiFi version 0.9.4 [7] and allows the creation of N virtual interfaces in station mode. Each interface is a virtual station $VSTA_i$ associated to an Access Point AP_i (Fig. 1) on a given radio-frequency.

The ClubADSL MAC layer is composed by the following elements:

- A **multi-queue manager**, responsible for managing the N virtual client queues. The basic mechanism to keep alive a number of connections while switching from one AP to another without loosing the association to each AP_i is based on the 802.11 power saving (PS) mechanism. The selected $VSTA_i$ keeps the control of the card during a given *duty cycle* f_i. During this time it transmits/receives frames based on the 802.11 DCF protocol while the other virtual stations (and the corresponding APs) can only buffer packets. Particularly, referring to Fig 2:
  - During the reserved *duty cycle*, $VSTA_1$ transmits and receives data according to the 802.11 DCF protocol. The other *VSTAs* are in PS mode, and hence they (and the corresponding APs) can only buffer packets.
  - when the *duty cycle* expires, $VSTA_1$ sends a frame to inform AP_1 that is going to PS mode and waits for its MAC ACK. According to the 802.11 protocol, AP_i starts to buffer the packets directed to it.
  - ClubADSL assigns the control of the card to $VSTA_2$ and switches to the AP_2 radio-frequency.
  - $VSTA_2$ sends a frame to announce that it can send/receive traffic and waits for its MAC ACK.
- A **virtual client switching scheme**, that minimizes the throughput degradation problem caused by the switching cost delay. This is achieved using i) a MAC queue per *VSTA* ii) regular data traffic buffered in the MAC *VSTA* layer queue to switch the PS state. Note that we still send probe messages in the case of empty MAC layer queues (i.e. no traffic sent by $VSTA_i$).

IV. Validation in a controlled environment

The experimental results in a controlled environment show that:

- ClubADSL can efficiently aggregate the throughput of the APs it is connected to.

[1] Currently we are implementing a scheduler for the management of this set of duty cycles. In this version, the duty cycles are fixed via user-space commands.

978-1-4244-4671-1/09/$25.00 ©2009 IEEE

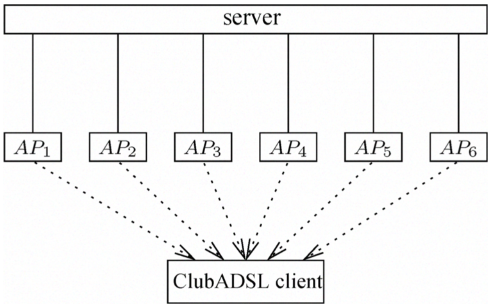

Fig. 3. Topology for the client throughput assessment

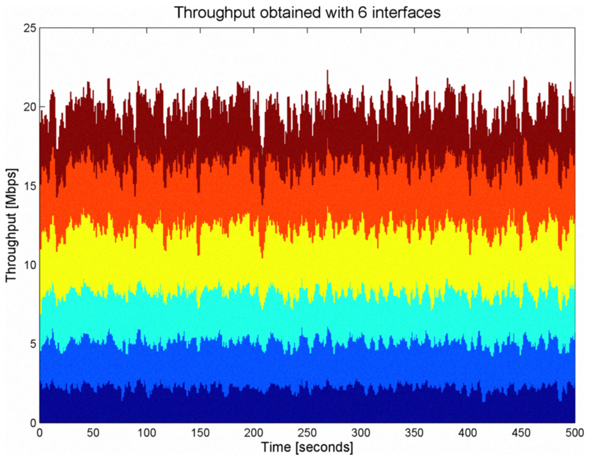

Fig. 4. TCP aggregate throughput. Each color represents the throughput component given by each virtual station at each second

- ClubADSL prototype implementation increases the consumption of the laptop of at most 1.6% respect to standard stations connected to only one AP.

In the tests, for the transport layer, we used a Linux standard TCP Reno with delayed ACK option enabled. At MAC level, automatic rate selection, wireless multimedia extensions, and RTS/CTS mechanism were disabled, together with any non-standard compliant features at MAC level. The ClubADSL client and the APs used a H/W queue with best effort parameters and a physical rate set to 54 Mbps. Throughput measures were based on the iperf tool.

A. Throughput

In this Section, we evaluate the throughput that the ClubADSL client can achieve using six APs on six different radio-frequencies connected to a server via Ethernet LAN connections at 100 Mbps, as shown in Fig. 3. In the test, each *VSTA* connected to the corresponding AP for a *duty cycle* of 15msec, resulting in a total wireless period of 90msec.

Fig. 4 shows the experimental aggregate downlink TCP throughput collected at the ClubADSL client. The figure depicts with a different color band the throughput achieved on each AP versus the time. The sum of the contributions is very close to the theoretical maximum of 21.7 Mbps that can be achieved connecting 100% of the time to only 1 AP.

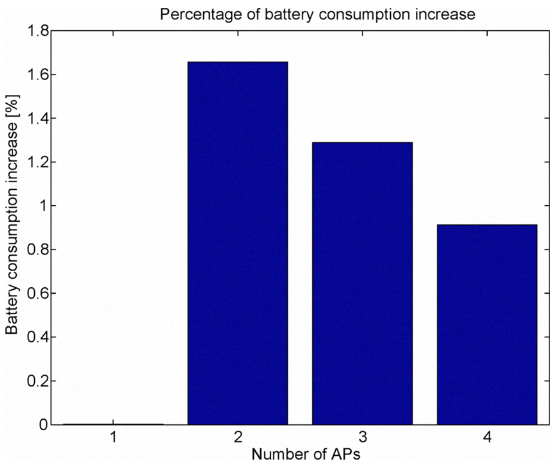

Fig. 5. Battery consumption increase in the ClubADSL prototype implementation

This throughput is achieved despite the switching cost between virtual interfaces. Particularly, in the implementation, we incurred in a channel-switching cost — i.e. the time where it cannot transmit/receive any traffic — of 1.2 msecs for uplink traffic and 1.5 msec for downlink traffic. This cost is less than half of the ones obtained with the time-division implementations given in [4] and [5]. The bulk of this cost is due to the hardware operation delay, which is in the order of 800 μsec in our Atheros chipset-based cards. This cost is hardware dependent and in other (more efficient) chipset implementations is reduced to 200-500 μsec [4], [8].

B. Power Consumption

In order to evaluate the power consumption of the ClubADSL prototype implementation we have deployed a test where the ClubADSL station connects to a certain number of APs, and simultaneously transmits TCP flows in uplink to the APs for one hour time. For each test, the ClubADSL station was switching among the APs with a *wireless period* of 100 msecs. We have then gathered statistics on the power consumption as reported by the Linux Kernel and summarized the results in Fig 5.

Particularly, in the figure, we show the increase of battery consumption versus the number of APs, compared to the reference scenario of a standard station connected to 1 AP. As the ClubADSL station connects to two APs, the extra-processing and high temporal accuracy required by the implementation introduce a cost in terms of power consumption of around 1.6%.

This cost decreases with the number of APs because of the higher impact of the switching time in the tests. In order to explain this finding, we note that higher is the number of APs higher is the number of switching procedures per second[2]. During this switching time, the 802.11 card cannot send/receive any traffic and the Radio-Frequency Transceiver

[2] Based on the selected *wireless period*, we had 20 switching/sec with 2 APs, 30 switching/sec with 3 APs and 40 switching/sec with 4 APs.

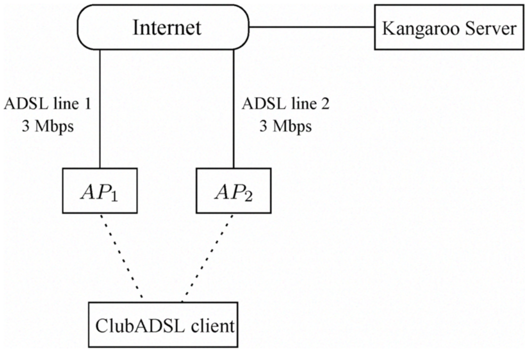

Fig. 6. Testbed deployed for testing the HD video quality with the ClubADSL client

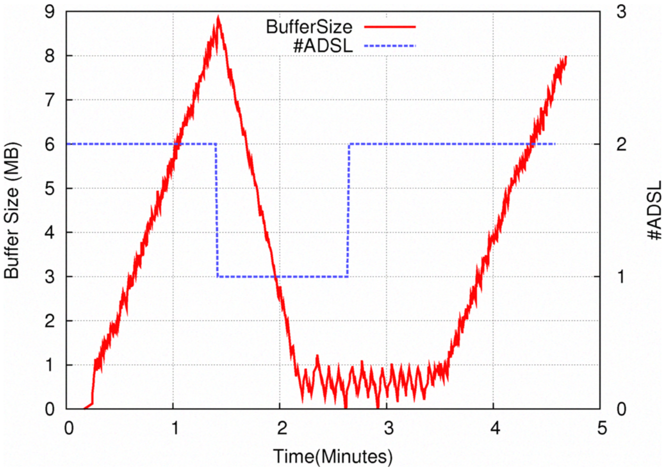

Fig. 7. HD video streaming with 1/2 ADSL lines

circuitry is mostly in idle state. Hence, a trade-off exists between the power consumption and the switching costs.

V. ClubADSL Quality Assessment using Real Applications

In this Section we present the experimental evaluation of the ClubADSL implementation in a real deployment. The tests have been performed in our research laboratory using two dedicated ADSL lines at 3 Mbps downlink speed, hence providing to the ClubADSL client a total theoretical bandwidth of 6 Mbps. The experimental results show that:

- ClubADSL increases the observed download speed of applications such as video-streaming, web browsing, and FTP downloads.
- ClubADSL aggregates the upload available capacity in incentive-based peer-to-peer applications.

A. Video streaming over the ClubADSL

High-definition (HD) video requires a high capacity connection. To test the video quality that can be afforded by the ClubADSL client, we setup a testbed composed by a ClubADSL client connected to the Video-on-Demand Kangaroo server [9], via two dedicated ADSL lines at 3 Mbps. The scenario is depicted in Fig. 6, with the Kangaroo server — deployed in the Internet, externally to the ADSL lines — sending a 4 Mbps sport video (plus control protocol overhead).

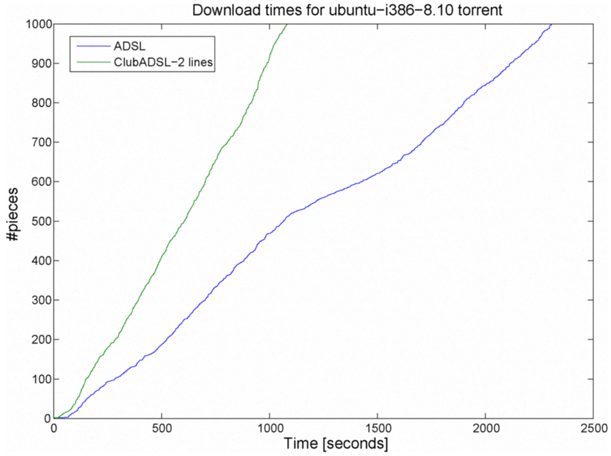

Fig. 8. Single download speed comparison

Fig. 7 shows, with a dashed line, the number of APs being used by ClubADSL, and with a solid line the buffer size stored in the video player[3]. The buffer is a representation of the audio/video quality perceived by the user during the video streaming.

We can see that while ClubADSL is connected to two APs, the download speed is higher than the video rate, with a buffering rate of around 6 MB/min (0.8 Mbps), corresponding to the difference between the download speed and the video quality rate. At around 1 minute and 25 seconds, we disassociate from 1 AP. At this point, the buffer size of Kangaroo decreases since one ADSL connection is not enough for the high definition TV. The 8.7 MB buffered video information is depleted in 44 seconds, at which moment the Kangaroo client on the ClubADSL station is not able to properly show a constant HD video-streaming. At 2:37, we re-associate the ClubADSL client to the second AP. Within a minute, the Kangaroo client (which is by itself a prototype implementation) refreshes the TCP connections and opens a new TCP flow on this new AP. After this time, the client starts to buffer packets and show the video again without interruptions.

Concluding, the high granularity in terms of TCP flows offered by P2P video-streaming applications is exploited by the ClubADSL client to increase the quality observed by end-users. Generally, also other server-based applications such as web browsing, and FTP downloads, observe increased download speed, because of the availability of multiple TCP flows for each application.

B. Peer-to-Peer download over the ClubADSL client

In Section V-A we showed how applications can benefit from the increased available download capacity. However,

[3] In Kangaroo, each video is divided in segments and the Kangaroo application is divided in 2 parts: i) the P2P core that downloads different segments from the server and other users. ii) the player that plays segments in a sequential way. The video buffer that is shown in the plot represents the amount of segments that are downloaded by the P2P core.

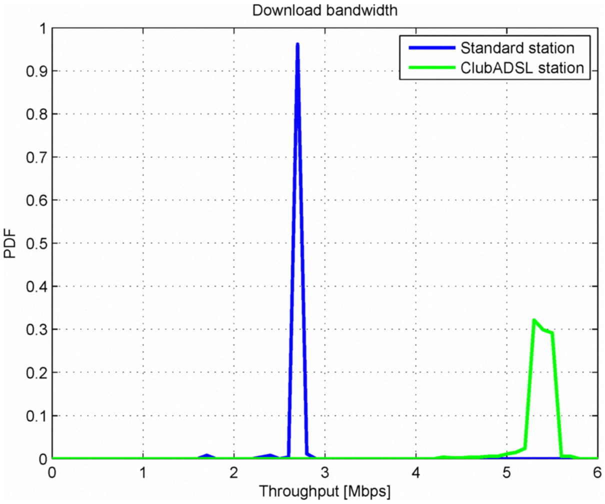

Fig. 9. P2P download speed comparison

ClubADSL does not only aggregate the download bandwidth, but also the upload available capacity. This is crucial in incentive-based peer-to-peer applications such as BitTorrent [10], in which peers contributing to a content (torrent) will preferably send data to those peers from which they receive more data. This “tit-for-tat” strategy is considered to be one of the cornerstones of the high performance achieved by the BitTorrent protocol [11]. In that sense, the upload capacity of a peer, and not its download capacity, will mostly determine the achieved download speed, or in other words, will determine how many peers will be willing to send data to them because they are good “contributors”. ClubADSL produces such a desired effect.

Fig. 8 shows the download times of the P2P file pieces for one single download of a popular (and legal) content via BitTorrent. Being a well-seeded popular content, download times did not vary greatly in subsequent tests, and we show one particular run for illustrative purposes. As a result of the doubling of the upload capacity, the file pieces were downloaded at roughly the double speed by the ClubADSL client respect to the standard 802.11 client connected to one 3Mbps ADSL line.

In order to evaluate how this result improves the throughput download, Fig. 9 also shows the download speed of ClubADSL compared to the standard 802.11 client. We can see that, in general, while not completely saturating the download capacity (as BitTorrent is not a server-based protocol) the obtained download speed of ClubADSL is the double of the one achieved by the standard 802.11 station.

VI. Conclusion

The never-ending increase of demand for high speed services requires a complete re-thinking about not only the possible advancements in infrastructure deployments, but also to what extent the current technology can be used. This work has shown how the current bottleneck for home connections, represented by ADSL, can be overcome by taking advantage both of the wireless nature of the connection to the ADSL line and the generally low ADSL utilization in the neighborhood.

We have presented ClubADSL, a wireless client that can connect to different APs in range and aggregate their unused ADSL bandwidth. ClubADSL works without any modification to existing infrastructure, it is a software-only add-on running on the client-side and does not require any hardware/protocol upgrades. We have given a performance evaluation, both in controlled environments and in real deployments, over applications as video-streaming and P2P.

Future work aims at studying the fairness among multiple ClubADSL stations and the performance benefits in a deployment with real users.

Acknowledgments

We gratefully acknowledge the help of Xiaoyuan Yang for the video-streaming tests.

References

[1] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. W. Biersack “Performance Limitations of ADSL Users: A Case Study”, PAM 2007.
[2] D. Han, A. Agarwala, D. G. Andersen, M. Kaminsky, K. Papagiannaki, S. Seshan. “Mark-and-Sweep: Getting the ”Inside” Scoop on Neighborhood Networks”, IMC 2008.
[3] R. Chandra, P. Bahl, and P. Bahl. “MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card”. In INFOCOM, 2004.
[4] A. J. Nicholson, S. Wolchok, and B. D. Noble “Juggler: Virtual Networks for Fun and Profit”, To appear in IEEE Transactions on Mobile Computing.
[5] S. Kandula, K. Lin, T. Badirkhanli and D. Katabi, “FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput”, NSDI 2008, San Francisco, CA.
[6] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M.F Kaashoek, “The Click modular router”, in ACM Transactions on Computer Systems, volume 18, pages 263-297, August 2000.
[7] The MadWiFi driver, http://madwifi-project.org/
[8] datasheets.maxim-ic.com/en/ds/MAX2820-MAX2821A.pdf
[9] Xiaoyuan Yang, Pablo Rodriguez, “Canguro: un sistema de disco virtual para VOD interactivo”, NAI award, 2007.
[10] B. Cohen, “Incentives build robustness in BitTorrent,” in Proc. of First Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, Jun 2003.
[11] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, “Measurements, analysis, and modeling of BitTorrent-like systems.” in Proc. of ACM IMC’05, Berkeley, CA, 2005.