D4.5 Final WP4 report on testbeds and field-trials

Grant Agreement Number:	**257740**
Project Acronym:	**TREND**
Project Title:	**Towards Real Energy-efficient Network Design**
Funding Scheme:	Network of Excellence
Project Coordinator	Name: Marco Ajmone Marsan Phone: + 39 011 5644032 Fax: +39 011 5644099 e-mail: ajmone@polito.it

Due Date of Delivery:	M36+3 (30/11/2013)
Actual Date of Delivery:	22/12/2013
Workpackage:	WP4 : Instrumenting the network for power saving
Nature of the Deliverable	R
Dissemination level	PU
Editors:	Ivaylo Haratcherev

Abstract:

WP4 coordinates experimental activities, with implementation and testing of the technologies and procedures studied in the project. This report covers the progress made during the last (third) year of the project.

Keyword list: yearly, final, activities, experimental, testbeds, field-trials

Disclaimer

The information, documentation and figures available in this deliverable are written by the TREND Consortium partners under EC co-financing (project FP7-ICT-257740) and does not necessarily reflect the view of the European Commission.

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user uses the information at its sole risk and liability.

Table of Contents

1. Executive Summary

This document presents the final yearly report for WP4 of TREND. TREND is a Network of Excellence (NoE) funded by the Seventh Framework Program of the European Union (FP7/2007-2013) under grant agreement n. 257740. The aim of the NoE is to assess and reduce the energy consumption of telecommunication networks.

The document contains the objectives, scope, roles and progress report for all tasks and activities in WP4 during the third year of TREND. It also contains a record of produced and planned papers, as well of mobility actions.

In the beginning of the last year of TREND a new experimental plan was devised to address the issues brought up in the second project review report. As a result, five new lines of experimentation have been introduced:

- Experiments on wireless access devices
- Implementation and measurement of FUFL on the FUB testbed
- Measurement and management of the energy cost of networking intensive applications: P2P and OSN
- Development of the TREND Energy Saving eMulator (TESM)

The first research activity focuses on energy efficiency of the devices that provide wireless access and it is related to the corresponding research activities in WP2. The experimental activity is focused both on validation on a real WLAN of the algorithms investigated in the project for the adaptation of the capacity of a wireless access network to the number of active users, and on the validation on a testbed of the effectiveness of the green 3G femto that was prototyped in WP2.

The second experimental activity: "Implementation and measurement of FUFL on the FUB testbed" is a result of the collaboration with FUB, which allowed TREND to use its very large testbed to experiment one of the proposed algorithms for energy efficiency in core networks - FUFL (Fixed Upper Fixed Lower).

The third experimental activity looks at the application level, which is more and more driving all the internal network characteristics, including energy consumption.

Finally, the experimental activity: "Development of the TREND Energy Saving eMulator (TESM)" is by itself a significant step in the direction of the "big picture" in green networking. Indeed, it allows a network operator to describe its network, and to explore the power saving options and the corresponding improvements. TESM leverages many of the previous TREND activities, from the collection of power consumption data (WP1), through the definition of traffic and network mid-term forecasts (WP3), to the identification of power saving algorithms (WP3).

2. Experiments on wireless access devices

2.1 Experiments of WLAN AP switch on/off in dense WLAN

Partners: PoliTo

Summary:

Dense Wireless LAN (WLAN) scenarios are typical of corporate or campus networks [1]. While the high density of APs is needed to provide enough capacity to guarantee good service to a potentially large population of users, it also introduces some redundancy and overlapping areas. A high number of access points (APs) fits the peak of user demand, but is superfluous during periods of low activity, e.g., at night, leading to considerable power wastage. The high capacity combined with redundant coverage opens the path for the design of algorithms that reduce energy waste by carefully and dynamically activating only the amount of capacity that is strictly required based on the number of users and the amount of traffic they generate. Some works [2-4] in TREND WP2 propose strategies to adapt the number of operating APs to the amount of needed capacity, by switching off the redundant APs when traffic is low, while still providing full coverage. These analytical studies show that the proposed strategies allow to greatly reduce the density of active APs, and, thus, save significant amount of energy.

The objective of this experimental activity is to implement some of the algorithms proposed in WP2 for dense WLANs, with the twofold objective of validating the feasibility of the proposed approaches in real networks and to investigate the actual performance under a real traffic scenario. The experimental activities are organized in two phases, devoted, respectively, to: i) testbed, and ii) field experiments.

Results:

A. Cisco Router testbed

The testbed setup is represented in Figure 1. The testbed comprises several WLAN APs providing overlapping coverage areas. WLAN APs are connected to a Cisco router equipped with the EnergyWise Technology [5]. The APs are powered through the IEEE 802.3af/at Power-over-Ethernet (PoE) standard interfaces of the router. A management PC connected to the router collects measurements of a number of parameters for each AP (the amount of carried traffic, the number of associated users, and the power consumption) by sending SNMP requests to the AP. Based on the collected data, the management PC implements some of the energy saving algorithms proposed in the context of WP2 and operates by switching on/off the PoE ports of the router to switch on/off the APs. We investigate the APs switching transient phases, including transient duration, and possible QoS degradation.

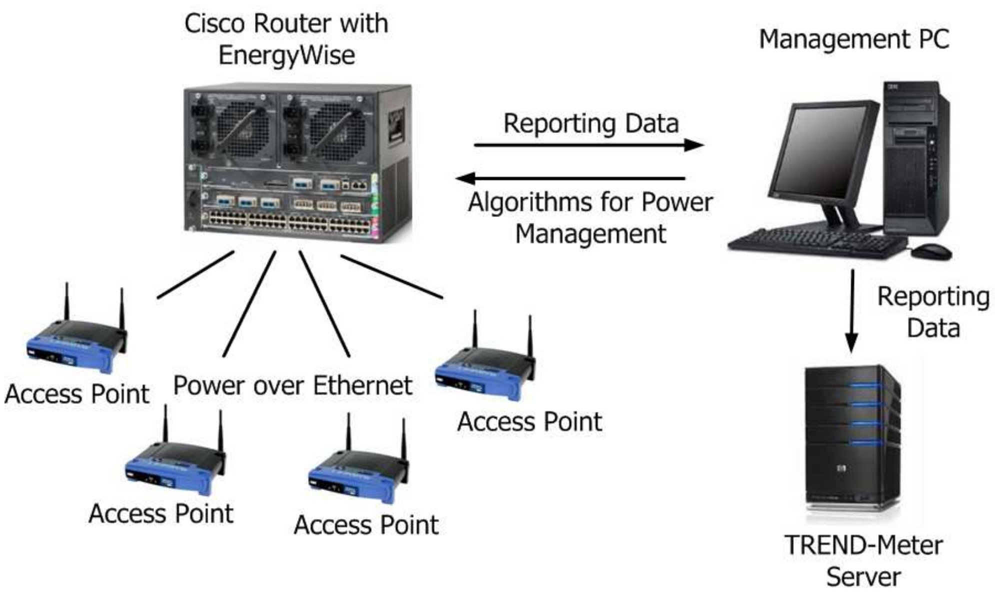


Figure 1: Testbed setup

Two energy saving strategies are implemented in the testbed:

1) Switching-on/off strategy based on number of associated users.

In this experiment, two APs are deployed in the same area, and our algorithm is based on the number of associated users, as in [3]. The (k+1)-st AP is switched on when the number of users in the network reaches kTh, but the number of active APs decreases from k+1 to k only when the number of users in the network becomes less than or equal to kTh-Tl, where the hysteresis Tl is used for preventing APs from turning on and off too frequently. The thresholds Th and Tl can be set to any value depending on different environment and QoS requirements. In our experiment, we set the threshold on the number of users Th of an AP to be 5, and the hysteresis Tl in the algorithm to be 1. Figure 2 shows the total power consumption of the two APs when the number of associated users varies.

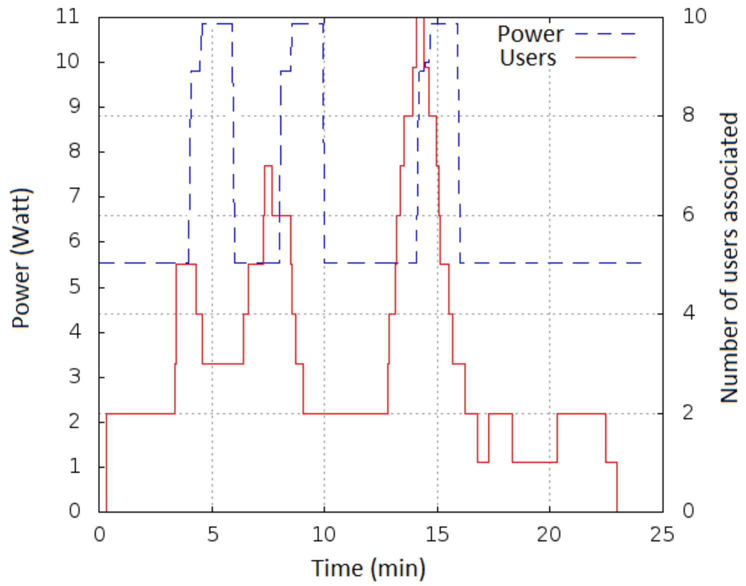


Figure 2: Power consumption of two APs according to the on/off algorithm based on number of associated users

We found out that around 60-secend boot time is needed for an AP after it gets power from PoE. Therefore, in the power consumption curve of Figure 2 there is a gradual increase before the power consumption of the second AP becomes constant. It is also shown that there is a short latency (around 10-15 seconds) between the user behavior variation and the response of the AP. The latency comes from the control process, because during this period the management PC is doing the procedure of: 1) detecting the users' association/de-association by SNMP protocol; 2) running the control algorithm; 3) sending signals to turn on/off the second AP; 4) getting the new power consumption value. Also, the power consumption of an AP is almost constant no matter how many users are associated to it.

2) Switching-on/off strategy based on aggregate traffic

In this experiment, we implement an algorithm to turn on/off the APs based on the aggregate traffic transmitted by the cluster of APs covering a common area. In the algorithm, we turn on/off a specific number of APs depending on the total traffic of all APs and a threshold of throughput for each AP. In the experiment, we generate traffic according to a trace based on real history data in PoliTo WiFi networks, in which the peak value (8 Mbps) is reached around 12 noon (as shown in Figure 3). We have 10 APs covering this testing area. The threshold of the throughput of each AP is set to 800 kbps. The results in Figure 3 show that the number of working APs is adaptive to the aggregated traffic. Note that the threshold of the throughput can be set to any value depending on different environment and QoS requirement. We generate UDP traffic with fixed packet size and data rate.

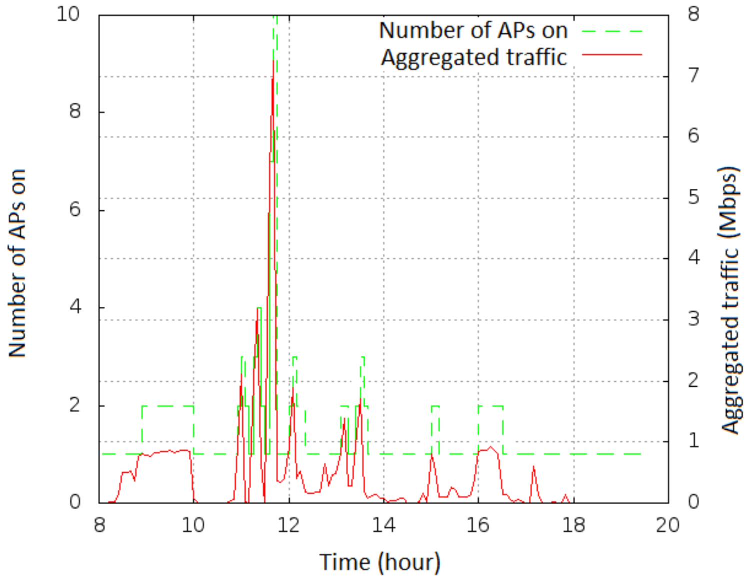


Figure 3: Number of APs powered on based on the aggregate throughout

B. The field experiment in operative networks:

During the second phase of the experimental activity, the algorithms investigated during the first phase are being implemented on a portion of the operative WLAN of the Politecnico di Torino Campus. In particular, 3 APs that provide capacity in the area of a student study room of the Department of Electronics and Telecommunications have been connected to the Cisco router, in a configuration similar to the one of the first phase. The experiment is still ongoing, so far we have got some preliminary results for the traffic traces of the 3 APs and the energy saving has been calculated through simulation for a typical day. Figure 4 shows the number of associated users for the 3 APs in a typical day (11/11/2013).

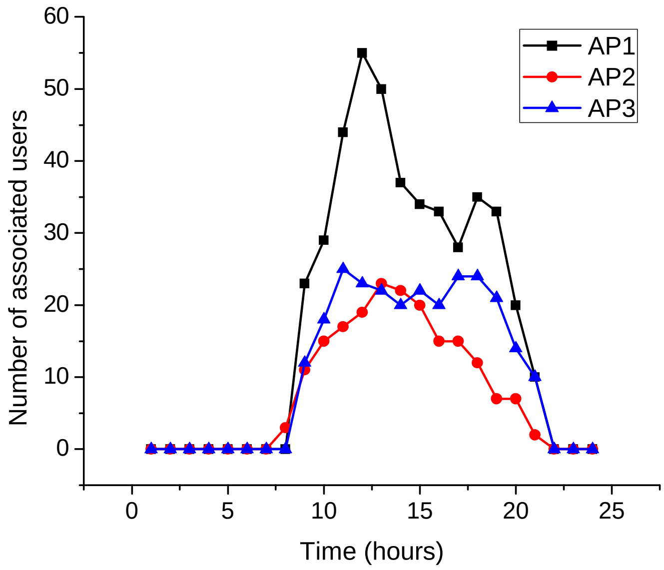


Figure 4: Number of associated users on the 3 APs during one working day

We set that AP2 and AP3 can be switched off to save energy, while AP1 is always on to ensure connectivity. We also set that the maximum number of users associated to an AP to 40, according to the traces we collect for the week from 11/11/2013 to 17/11/2013, because the maximum total number of users associated in one hour for 3 APs is 102 in this week. We set that AP2 is going to be turned on when the total number of associated users is over 40, AP3 is going to be turned on when the total number of associated users is over 80. The minimum time period for which an AP can be turn on/off is 1 hour. Table 1 shows the simulation results for the off-time of the 3 APs and the energy that can be saved for the typical working day 11/11/2013.

Table 1: The off-time of each AP and the energy savings in a day

AP ID	Off-time for one day (hours)	Energy saving (%)
AP1	0	0.0
AP2	12	50
AP3	21	87.5
Average saving		45.8

On Saturdays and Sundays, AP2 and AP3 could be shut down for most of the time due to the very few users and low utilization of the WiFi network, because there is almost no people in the study room where the 3 APs are located. We found that the average energy saving for the total week from 11/11/2013 to 17/11/2013 is 54.9%.

However, the results in Table 1 are based on the simulation with a fixed threshold of number of associated users and a fixed period of on/off time of AP (1 hour). In different network scenarios, there are some dynamic issues that may need to be considered in our future work, e.g. the probability of the connectivity loss when handing over a user from one AP to another at the moment of shutting down the former AP, more flexible definition for the threshold of the number of associated users or traffic load of an AP, the frequency of turning on/off APs in a day, etc. Therefore, we are going to design and apply some dynamic algorithms for different network areas, in order to adjust the threshold of the number of associated users and traffic load for each AP in real time, in order to make the energy saving scheme more flexible and scalable for different network scenarios.

2.2 Remote monitoring and control for energy-efficient femto

Partners: ALBL-F

Summary:

The first part of this activity consisted in the validation and improvement of a power-consumption model, in order to estimate in real-time the power consumption of a 3G femto. This estimation model was created in a previous study, however it had not been verified and had to be checked against the actual femto power-consumption. If the accuracy of the model turned out to be insufficient, a subsequent step of improvement would be carried out.

The second part was to develop “on-board” power consumption visualization and power-control web interface for the “green” femto. This functionality allows for direct monitoring of the femto power consumption from outside the femto via a web interface and a HTTP server implemented on the device. The same interface enables simple power-control of the femto, for example manual or scheduled transition to and back from a standby mode.

Finally, the third part comprised the integration of the femtocell’s power consumption and throughput data real-time reporting into the TREND-meter tool.

Results:

The Power estimation model evaluation

The power estimation model aims at calculating the power consumption based on information about the state of the different parts or components of the femtocell, namely the RF part, the TCXO heater, the LEDs and the different ICs like the baseband processor.

The power consumption of the Femtocell’s different parts was measured in real time with a test bench that includes a National Instrument Data Acquisition Device (DAQ) NI USB-6009 and the Labview software. The power consumption was then stored in a file to be compared with the power estimated with the model for the same time period in which the real measurements were taken. To evaluate the model’s performance in all conditions and situations, the femto was cycled multiple times via various modes while the comparison was carried out.

After several iterations, the power estimation model was refined, and an average accuracy of 98% has been achieved. Figure 5 shows typical model performance in terms of relative error for different situations.

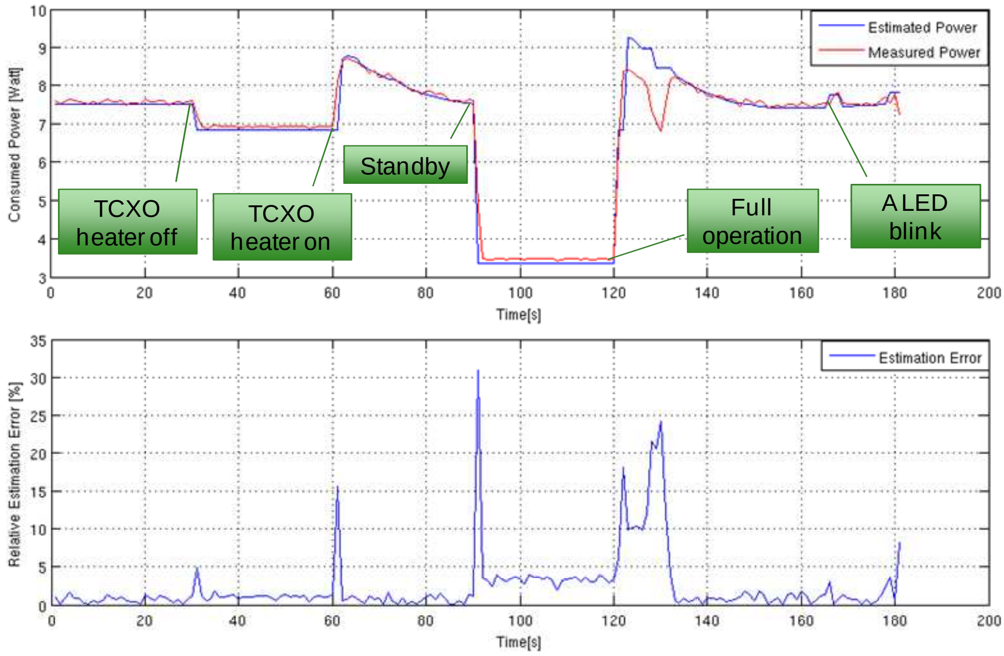


Figure 5: Relative error of the power estimation model

Power consumption visualization and control web interface

An HTTP server was implemented on the femto to provide a user-friendly WEB interface that allows remote control of the femtocell's standby mode - manually or by setting up an automatic schedule.

The WEB interface also provides a chart to track femto power consumption in real-time. The chart is updated each second. An example screenshot of the interface is shown on Figure 6.

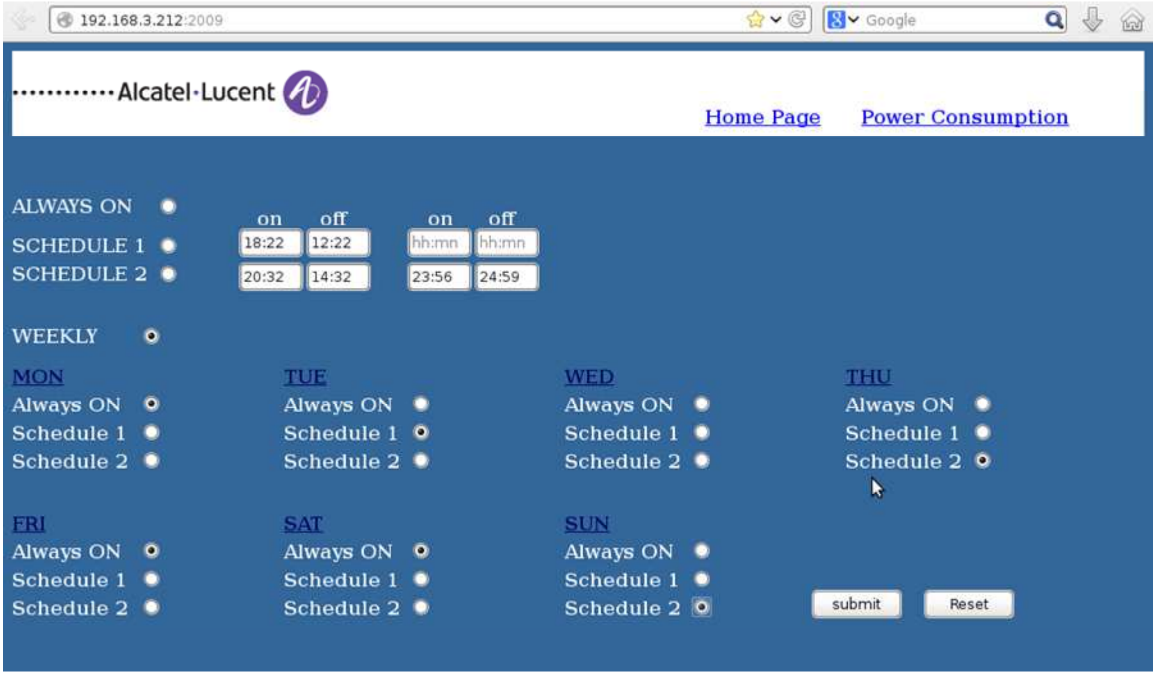


Figure 6: Femtocell's user WEB interface

Integration into TREND-Meter

In the final part of this activity, the energy-estimation module in the femto was coupled with the TREND-meter tool, providing regular real-time measurements for the TREND-meter database. Figure 7 shows the interaction of the femto with the tool.

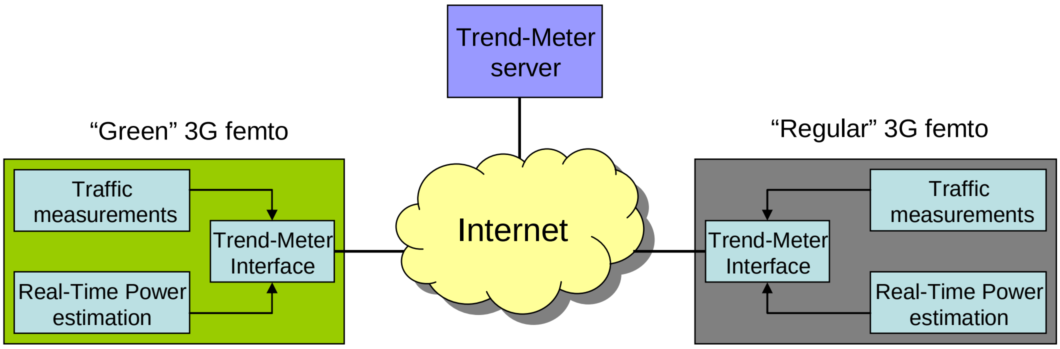


Figure 7: Femtocell integration in TREND-Meter

On Figure 8, typical measurement graphs are shown.

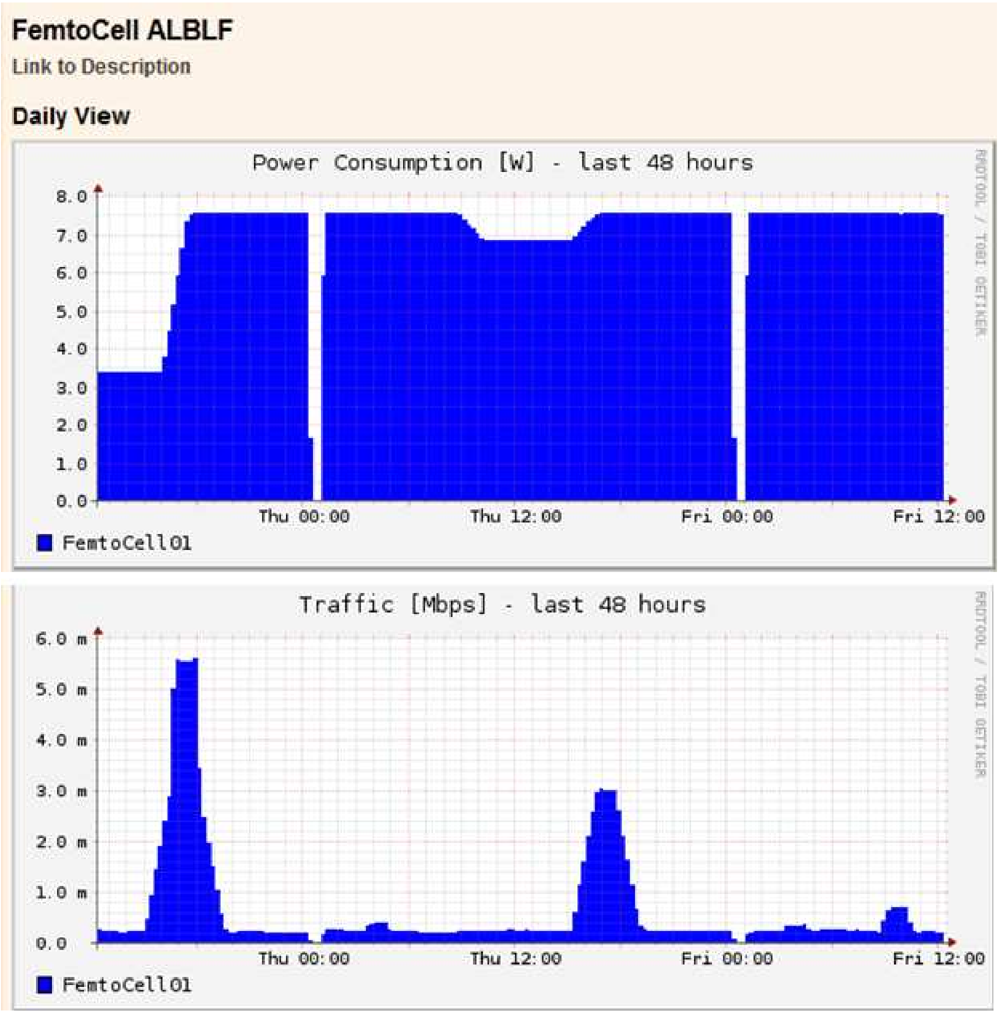


Figure 8: Femtocell graphs in TREND-Meter

2.3 *Extraction of usage and mobility patterns of campus WLAN users*

Partners: TUB and PoliTo

Summary:

In this study we focused on the extraction of the usage and mobility patterns of WLAN users, which can be beneficial to develop switching on/off strategies. These patterns represent how the user demand differs from one time period to another as well as between various locations in a campus. Understanding these variations can be helpful to design more energy-efficient off-line algorithms [6], proposed to save power in dense WLANs.

Although publicly accessible WLAN traces collected from different environments (e.g., campuses and enterprises) have been used to not only design the off-line algorithms but also evaluate the performance of them, most of the existing traces are not up-to-date, leading to less accurate results. In fact, the existing traces may completely reflect neither the growth of the user demand for capacity nor the changing needs for mobility, which stem from the growing deployment of portable devices (e.g., smartphones, tablet etc.). Therefore, it seemed vital to collect new traces and analyse them in order to assess the spatial and temporal dependencies of the user demand.

Our target environment was the main campus of PoliTo (size: ~ 0.225 km^2), where 83 APs (IEEE 802.11 devices [7], working on 2.4 GHz) are deployed to provide the coverage and the capacity to about 17,000 users. The central controller of the WLAN facilitates the monitoring and the control of the network, and provides reports of the system status [8]. The traces were collected from the database of the central controller, in which administrator-defined data has been collected periodically. The traces used for this study were collected from 5th June to 5th July, 2013, and included the following information:

- IP and MAC addresses of the users,
- association and de-association time of user sessions,
- name, location and the SSID of each AP,
- average session throughput, traffic (per user session) handled by the AP, number of sent and received frames,
- average SNR for client session, average RSSI for client session.

Figure 9 illustrates our proposed framework for the analysis of the collected traces.

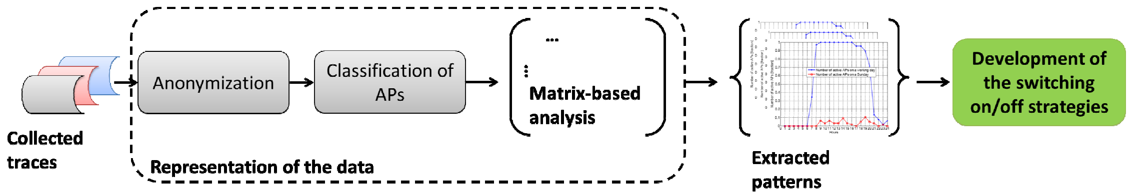


Figure 9: Proposed framework for the analysis of the collected traces

In order to assure the anonymity of the users, we removed the user ID from the traces, and the hashed values were also substituted for MAC addressed of the user devices.

After classification of the APs depending on the traffic handled by them, we analysed the following characteristics to extract the usage and mobility patterns of the WLAN users:

- Client behavior (e.g., connection session length)
- Characteristics of traffic generated by the clients
- Characteristics of the WLAN (e.g., active time of the APs)
- Client mobility (e.g., the number of roaming events)

It is worth noting here that the development of new switching on/off strategies is beyond the scope of this study, but the results can be beneficial to the development of these strategies nevertheless.

The following are some of the most important results of the analysis:

- During a working day, about one third of the events are roaming events. Hence, locations where the remaining events take place, can be virtually estimated. This is one of the key messages of this study. Furthermore, in comparison to previous studies (e.g., [9]), an increase in the number of roaming events has been identified.
- The number of roaming events to the immediate neighbouring APs is far too low, which means there is a large overlapping area between most of the neighbouring APs. Hence, the switching on/off strategies are applicable in this scenario.
- The applicability of an aggressive switching-off strategy, e.g., [10] in the nights (from 9 p.m. to 7 a.m.) and particularly on Sundays is recognized.
- We state that to save power at the campus, more than one off-line algorithm is required regarding the spatial dependency of user demand. In other words, contrary to assumptions made in most of previous studies, the APs placed in different locations may handle different amount of traffic during a given time period (e.g., from 10 a.m. to 11 a.m.).
- It has been shown that on Sundays it is only required to keep alive the APs placed in the vicinity of the entrances of the campus (a case for the partial coverage provision).
- It is reported that virtually all the APs are active less than 12 hours per day.
- A dramatic decrease in number of active APs (i.e., at least one user associates with the AP) on Sundays in comparison to working days has been identified. However, it can be seen that the percentage of the active APs out of the total number of APs is increased in comparison to what has been reported in [9]. This can reflect the recent growth of the user demand for capacity.

To summarize, in order to develop more energy-efficient on/off switching strategies an up-to-date collection of traces reflecting changes in the usage and mobility patterns of WLAN users is required, and was so far missing in the research literature. To this end, such an up-to-date trace has been collected and analysed as result of this experimental activity.

Outcome:

The results of this study will be published in a conference paper and a technical report.

During this study period, Fatemeh Ganji (TUB) visited PoliTo (30/06/2013 – 12/07/2013 and 01/07/2013 – 04/07/2013).

2.4 An experimental study on the detection of WLAN-user connectivity initiation in the low SNR regime

Partners: TUB

Summary:

The main goal of this experimental study was to verify the practical applicability of the very aggressive WLAN on-off switching strategy proposed by TUB in WP2 [10], namely, a purely WLAN-based approach to keep alive only a skeleton consisting of a small number of WLAN

APs that provides the coverage required to detect the user connectivity initiation. In [10], it has been analytically demonstrated that the number of APs remaining in operation is upper bounded by the minimum SNR of the successfully decoded Probe-Request frames sent by the user attempting to connect to the WLAN.

To complement these results we performed two following experiments:

- Using an experimental setup, consisting of a commercially available WLAN AP [11] and a user device (laptop), we verified that the user connectivity attempt can be detected with a given probability of detection, when the SNR of the successfully decoded Probe-Request frames is above a given threshold. For this experiment we applied the model recommended and verified in [13], which shows the relation between the probability of a successful decoding of a frame and its SNR. Using this model, we measured the threshold of the SNR, where the probability of decoding the Probe Request frame is virtually 100%.
- It is clear that the SNR of the frames that can be successfully decoded at the receiver is higher than the SNR of the frames that cannot be decoded. Hence, even a more energy-efficient WLANs can be introduced, if the connectivity attempt of an incoming user can be identified from patterns of changes in energy level of the channel that are distinguishable from the noise level in the lower SNR regime (below the threshold specified in the first experiment). Thus, second and more importantly, by adding an additional measurement equipment (spectrum analyser [12]) to our setup, we verified whether the user connectivity attempt can be still detected, even when the Probe Request frames are not decoded (correct reception of a full frame is not possible), by observing if the changes in energy level of the channel can indicate transmission attempts.

The results of the first experiment showed that the threshold for successful decoding of the entire Probe-Request frame is equal to 0 dB for the device under tests [11] (received signal strength, RSS, as well as noise floor of the AP, is equal to -96 dBm).

In the second experiment, we aimed at going below the threshold obtained from the first experiment, when the SNR of the received Probe Request frame is not sufficiently high to decode the full frame correctly. Under such unfavourable channel conditions, we claim that patterns of changes in the energy level of the channel can represent the transmission of the Probe Request frames, which are not successfully decoded.

To this end, we proposed a detector of Probe-Request transmission, whose task was to identify the transmission of a train of Probe Request frames, sent with a given pattern (the known number of re-transmission and time interval between two consecutive frames), with a pre-defined probability of unsuccessful detection (false negative) within a given detection delay. In our experiments, we examined quantitatively the performance of the proposed detector in terms of successful detection of the transmission of trains of Probe Request frames in the low SNR regime. To this end, we collected the PSD (power spectral density) values, measured by the spectrum analyser [12]), when the Probe-Request frames were not decoded by the AP. The collected data was further fed to the PRB-REQ transmission detector (a Matlab code) and the processing was done offline in Matlab.

Our results reveal that the transmission of the Probe-Request frames, with a given pattern, can be detected, when the average SNR varied between -1.75 dB and 2.25 dB, corresponding to ~-102.5 dBm ≤ avg. RSS ≤ ~ -98.5 dBm. It is worth noting here that the noise floor is not constant during our experiment and it is measured for each train of PRB-REQ frames (average noise floor of the spectrum analyser is about -102 dBm). It has been shown that the probability of false negatives (i.e., the transmission of the Probe-Request frames is not detected) varies between 1% (RSS= -98.5 dBm) and 25% (RSS= -101.5 dBm).

Moreover, we performed numerical analysis to address the potential power saving, which can be achieved with this improved version of our aggressive switching-off strategy, based on the proposed Probe Request-transmission detector. In order to calculate the power saving, similar to [10], we compared the power consumption of a reference WLAN with density of 2960 APs/km^2 [14] with the power consumption of the APs (IEEE 802.11b devices) providing only the detection coverage over the given target area (1 km^2). We considered the indoor scenario defined in [10], namely, avg. RSS of -98.5 dBm, noise floor equal to -100.75 dBm, transmission power of the user device set to 20 dBm, power consumption of each AP approximated as 10 W, path-loss exponent and standard deviation of the slow fading were equal to 2.5 and 6.8 dB, respectively. We have shown that, when the target area is covered by the APs providing only the detection coverage instead of providing the full capacity (i.e., 11 Mbps with the IEEE 802.11b standard), the power saving potential is in the range of 98.6% of the total consumed power. Although the power saving potential obtained in this experiment is significantly high, it can be achieved at the price of an increase in the number of false negatives, representing an increase in delay of user detection.

Outcome:

The most important results obtained in this series of experiments will be published in a conference paper.

3. Implementation and Measurement of FUFL on the FUB Testbed

Partners/Collaborating Institutions: CNIT-UniRoma1, FUB, TUB

Summary: The aim of this activity was to validate whether it is feasible to implement in an operational network the Energy-Aware Adaptive Routing Solutions (EA-ARSs) proposed within WP3. We used the testbed that FUB has access to [6], and targeted the EA-ARS called FUFL (Fixed Upper Fixed Lower) [16][17]for implementation. FUFL requires no traffic rerouting, but just simple traffic monitoring of load on network interfaces. The testbed is based on Gigabit Ethernet (GbE), so we targeted switching off GbE links with their interfaces instead of lightpaths, as originally proposed in [16].

After configuring the testbed according to the demonstration needs (three Juniper M10/M10i routers interconnected into a logical ring with logical links consisting of two Gigabit Ethernet links each, as shown in Figure 10), we successfully implemented not only FUFL, but also DUFL (Dynamic Upper Fixed Lower) [16], which uses rerouting in order to increase the number of idle network interfaces that can be switched off in low demand hours.

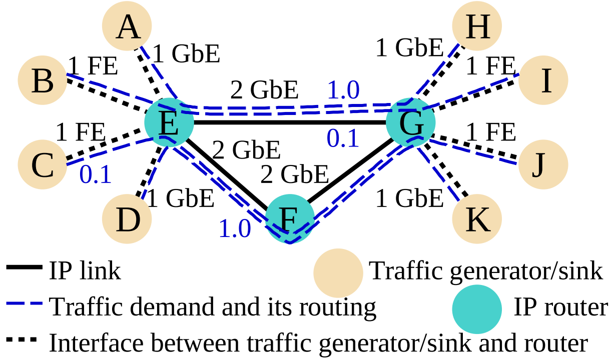


Figure 10: Base logical topology configured on the testbed

We loaded the testbed with traffic generated by Testing Platform Spirent SPT-3U, Anritsu MD1230B and a linux Personal Computer. Due to limitations of these devices and limited efforts available for this activity, we used random traffic (between nodes A and H, and nodes D and K in Figure 10) and sine-like traffic (between nodes B and I, and nodes C and J in Figure 10), and not the traffic measured by the TREND-Meter. The artificial traffic used in our study does not affect the main aim of this activity though, i.e., validation and demonstration of EA-ARSs proposed within WP3. Given the maximum value of the random traffic 1 Gbps, and of the sine-like traffic 0.1 Gbps, we chose the IP routing so that activation and deactivation of network interfaces is triggered (see Figure 10).

Our results show that it is possible to automatically and remotely adapt the network configuration, switch off interfaces, and save energy using off-the-shelf equipment. The power savings reached up to 16% for DUFL and up to 14% for FUFL (see Figure 11), however these values were not the primary objective of this activity (absolute values of power savings did not exceed 83 W). More importantly, no traffic losses were experienced in our experiments. The packets suffered from minor increase of delay (up to 30 ms). We experienced relatively long time required to reconfigure the network, where accessing the routers was the most time consuming action (11.54 s). We believe that this drawback can be easily overcome in the operational network of the future – network devices are expected to have dedicated (quick) access methods in order to reconfigure them according to the changing traffic demands.

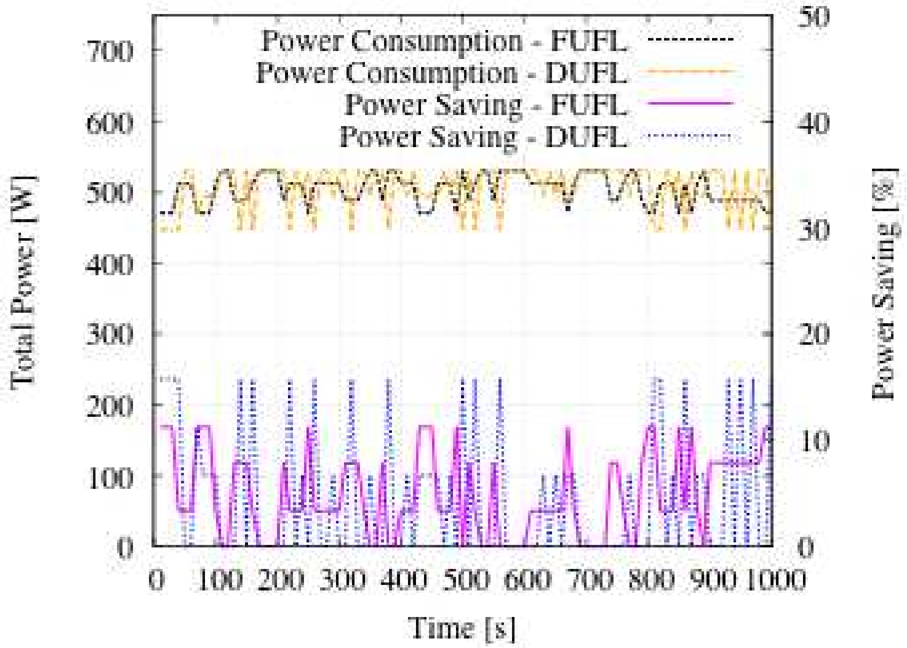


Figure 11: Power consumption and power saving for FUFL and DUFL experiments on the testbed

Outcome: Main results were published in the paper presented at the Tyrrhenian International Workshop on Digital Communications in Genoa, Italy in September 2013 [18].

A paper titled "Facing the reality: validation of energy saving mechanisms on a testbed" was submitted to Journal of Electrical and Computer Engineering [19], where we extensively analyze related work (publications regarding experiments on the MiDORi [20] and CARISMA testbeds [21]).

A mobility action of Filip Idzikowski (TUB) to Rome (both CNIT-UniRoma1 and FUB) was performed between 16/03/2013 and 20/03/2013. Filip Idzikowski together with Luca Chiaraviglio, Antonio Cianfrani and Angelo Coiro (all CNIT-UniRoma1) visited FUB on 19/03/2013.

4. Measurement and management of the energy cost of networking intensive applications: P2P and OSN

Partners: CNIT and UC3M

Description

The rapid growth of the Internet brings new challenges for the management and distribution of content to users. While the Internet Protocol was initially designed to be used in a Peer to Peer (P2P) fashion [23], the appearance of content providers change the system to a client-server model, in which the content was mainly stored in servers and it was consumed by final users. Due to economic and performance issues, these servers have been typically grouped forming data centers, in which a lot of them share resources. In the past years this trend in the content distribution has been reversed with the introduction of the P2P system for sharing the content [24] and distributing it among different data centers, forming Content Delivery Networks (CDNs) [25].

These content delivery mechanisms have the same role of distributing content to the end-users, even if they have different structures. These structural differences generally deal with the performance in terms of latency and bandwidth. To meet the demands of faster and more reliable Internet applications and services, the number of network devices had to increase enormously, triggering the concern about the huge amount of energy consumed. From current estimations [26] is noted that the Information and Communication Technology sector (ICT) consumes between 2% and 10% of the worldwide energy consumption, and these values are expected to increase in the future years, if no radical changes in Internet technology design will be undertaken. Moreover, this figure becomes even more impressive if we consider not only networking devices (e.g., routers, switches, etc) inside telecom and home networks, but also the "networked" ones [27][28].

From the point of view of end-hosts, previous works showed that the CPU is one of key components with largest power consumption [29]. The CPU energy behavior is dynamic in the sense that it depends on the workload. The workload of a CPU arises from the applications that are running on it, and on the activities performed at the kernel level to support them. Moreover, CPU's already implement advanced power management features, which can be set through the Advanced Configuration and Power Interface (ACPI) [30]. This standard interface completely hides the CPU internal techniques to reduce power consumption, which may differ depending on the processor, to the OSs and SW applications. The ACPI standard introduces two main different power savings mechanisms, namely performance (P-) and power (C-) states, which can be independently employed and tuned for the largest part of today's processors. However, due to the effectiveness and the simplicity of usage, the power management of modern OSs tends mainly to rely on C-states.

Starting from these considerations, the objective of this joint experimental activity is to provide an in-depth experimental analysis of the impact of file distribution applications on the energy behaviour of a final user networked device. In this work we decided to evaluate two of the most popular protocols in the Internet: HTTP and BitTorrent. In a HTTP environment, we used the Epiphany Browser, and in the case of the BitTorrent protocol, we exploited two different programs: Vuze and Transmission.

With this analysis we aim to answer some important concerns: *What protocol is more energy efficient for content distribution for final users? Is always the same protocol the best under different scenarios? Is the performance affected by network conditions?* To this purpose, we set up a complete testbed that allowed us to perform a number of measurements for analyzing the energy consumption of the CPU of a standard PC under different scenarios. The testbed will be described below.

Experimental Testbed

In order to collect the largest possible set of data, we set up a complete and simple testbed (Figure 12) which involves three main elements: the System Under Test (SUT) that is a commercial off-the-shelf PC running the Debian Linux operating system, a Linux based PC acting as Gateway, and a multi-channel Data AcQuisition (DAQ) used to collect a high number of DC power consumption probes.

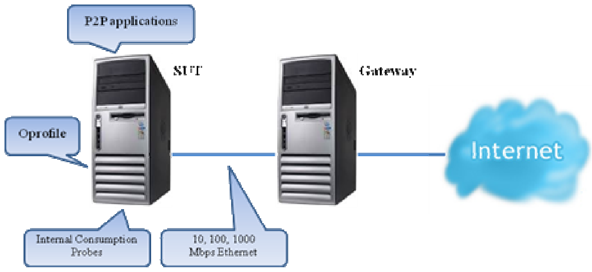


Figure 12. Experimental Testbed

The System Under Test

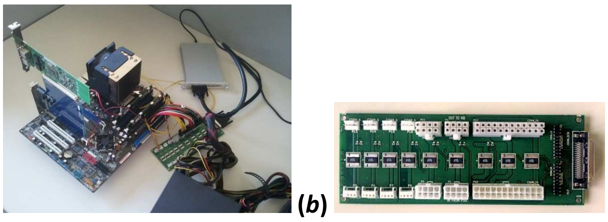
The SUT is a Linux workstation equipped with an Intel i5 processor running at 2.68GHz, with 4 physical cores and a maximum power consumption of 95W [33]. The workstation is equipped with 2 GB of DDR3 RAM, and an Intel PRO Gigabit Ethernet adapter. The Operating System (OS) is the Linux Debian 5.0.6, and the kernel version is the “vanilla” 3.4, which supports symmetric multi-processor (SMP).

The power management of the Intel i5 processor is configurable through the ACPI standard [30], which provides today a standardized interface between the hardware, where power management capabilities are realized, and the software. This standard interface completely hides the CPU internal techniques to reduce power consumption, which may differ depending on the processor, to the OSs and SW applications. The ACPI standard introduces two main different power savings mechanisms, namely performance (P-) and power (C-) states, which can be independently employed and tuned for the largest part of today-s processors. However, due to the effectiveness and the simplicity of usage, the power management of modern OSs tends mainly to rely on C-states.

Regarding the C-states, the C0 is the active power state where the CPU executes instructions, while C1 through Cn states corresponds to sleeping or idle modes, where the processor consumes less power and dissipates less heat. As the sleeping state (C1, …, Cn) becomes deeper, the transition between the active and the sleeping state (and vice versa) requires longer time and more energy [34]. C-states are usually very effective primitives (more than P-states), since they literally allow to quickly shut off a number of internal CPU components, avoiding intrinsic energy wasting due to leakage current. However, when components are re-turned on, they generally need a no negligible additional start-up. [35] shows the behavior and the power consumption of an Intel Core i5 processor when receiving IP packets.

C-state transitions are mainly driven by the OS and device I/O operations. When the CPU finishes serving its job backlog, the scheduler of the OS may decide to enter in a low power idle mode (C1, …, Cn). In such states, the CPU can wake only by some scheduled activities in the OS, or by an interrupt coming from an I/O component (like a NIC, a keyboard, etc.). In this respect, it is worth nothing that network traffic dynamics can heavily influence interrupt generation from I/O hardware, and then the effectiveness of C-state transitions.

Power measurements



(*a*) (*b*)

Figure 13. The end-host equipped with PCI-e, DDR3 and ATX risers connected to the DAQ system (a) and the ATX riser board prototype (b).

Regarding the power measurement tools, since PC hardware does not usually include power probes to independently measure the power absorption of CPU, we used a riser board for ATX power connectors described in [35], which allows putting some current and voltage probes on the CPU supply rails. The probe outputs are collected by an external DAQ device. Figure 13(a) shows the end-host equipped with the ATX riser board connected to the DAQ system. Figure 13(a) also shows other riser boards for PCI and RAM busses, but we neglect it since the consumption of these components is substantially constant.

Internal SW measurements

The internal SW measurements have been carried out by using a profiler, called Oprofile [36]. It is an open source tool that realizes a continuous monitoring of system dynamics with a frequent and quite regular sampling of CPU HW registers. Oprofile allows the effective evaluation of the CPU utilization of both Linux user- and kernel-space with a very low computational overhead.

Network Performance Measurements

In order to evaluate the network performance indexes, we enabled Tcpdump, to collect traffic traces for analysis. Finally, the different access link speeds were emulated by setting the bandwidth of the Ethernet link between the “gateway” and the SUT to 10, 100, and 1000 Mbps.

Methodology

By exploiting the testbed described above, we performed several tests in order to understand the advantage and drawbacks, regarding the energy consumption of different types of content distribution under different network conditions. It is important to note that the tests for each scenario have been repeated at least 10 times for each experiment and we present here the average results.

The wide range of file sharing applications over the Internet gives the final users the possibility to choose, in a very easy and usually free way, the application that best meets their needs. In this work we decided to evaluate two of the most popular protocols in the Internet: HTTP and BitTorrent.

In a HTTP environment, we used the Epiphany Browser. This application is based on the typical client-server model. In the case of the BitTorrent protocol, we exploited two different programs. The first one is Transmission, a very simple command line client. The second one is Vuze, one of the most used BT clients nowadays. Vuze is a complete environment that allows users to download several pieces of content in parallel, and it has integrated some tools

Table 1. TEST CASES

BOTTLENECK BANDWIDTH [MBPS]	*DOWNLOAD*	*SIZE*
1000 - 100 - 10	Debian-6.0.7-amd64-CD-1.iso	676 MB
	Debian-6.0.5-amd64-DVD-1.iso	4.64 GB
	Parker.avi (Film)	2.5 GB

not related at all with the download of the content as a video player.

For our tests, we considered some network conditions that may represent some common cases in the today's Internet. As shown in Table 1, the selected scenarios provide different end-to-end bandwidth. The 10Mbps link can be used to understand the energy consumption in final user devices connected to a commercial DSL or cable network. The 100Mbps bottleneck could represent a residential FTTH network or the connection of a small company. Finally, the 1Gbps link (the speed of our network) can represent the speed used in bigger companies or even in a Datacenter.

A characteristic of P2P environments is the competition or cooperation of resources. In this respect, we are going to evaluate the case in with the user have to compete for the resources in order to complete the download, we refer this case as competitive download and the case where all peers collaborate (collaborate download). In order to evaluate the first case we downloaded the most popular movie in the major BitTorrent tracker during our experiments. It is important to point out that due to some limitations it was impossible to simulate a competitive download using HTTP. We explored the option to download the same movie from a commercial service as Mega.com or bitshare.com, but these systems usually limit the download rate in an artificial way, making impossible an automatic download. For that reason we decided to download the movie from a server placed in a university in a different country.

In the case of collaborative download, we decided to download a Debian ISO. For the HTTP protocol, we chose download an ISO from the closer server to our lab in order to minimize the delays. Moreover, in the case of BitTorrent, we could see how the absence of free riders and malicious users in the swarm make very easy to download the content. Finally, in order to evaluate if the download size influences the energy consumed by the end-host, we repeated the experiment with two different Debian ISOs: a CD and a DVD. Table 1 shows a summary of these different experiment setups.

Results:

Network Performance

The time needed to finish a download, plays an important role in the device performance, because this is the minimum time that our system should be on for completing the distribution of the file. Figure 14 reports the average download time for every download content with each different file distribution application: a) HTTP, b) Vuze and c) Transmission. From this figure, we can see that the amount of data downloaded and the speed link impact the download time. In addition, we can observe that the download time is not reduced in the same proportion with the increase of the speed link. Moreover, it is worth to note that HTTP needs more time than Vuze and Transmission to download the same amount of data, due to the fact that HTTP uses a single source instead of the multiple sources exploited by BitTorrent applications. In particular, in the case of the collaborative download, HTTP needs between 60% and a 486% more time to finish a download than BT.

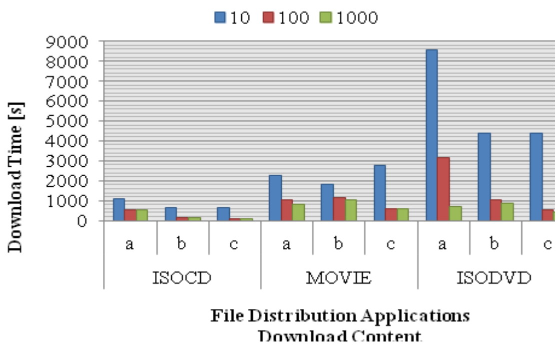


Figure 14. Download time of file distribution applications

Energy Consumption

Regarding the energy consumption, we calculated the average energy needed to download a megabyte of information for each one of our tests. The obtained results are reported in Figure 15. Analyzing this figure, we can outline how in presence of lower bottleneck speeds the energy consumption increases in a significant way. Furthermore, it is possible to note that the energy consumption is influenced by the downloading time, resulting in higher energy requirements by the HTTP protocol in the case of collaborative downloads, and especially when the download speed is low. In particular, a collaborative download needs between 8% and 50% less energy per megabit using BT than using HTTP, depending on the link conditions. In the competitive case, we can observe that the energy consumption with BitTorrent applications is higher than HTTP, in average HTTP needs 23% less energy to complete the download.

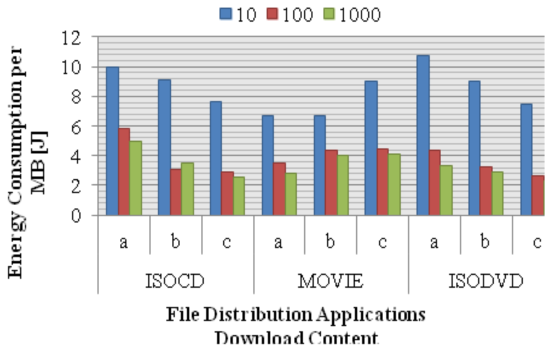


Figure 15. Energy Consumption per MB of file distribution applications

These increasing values of energy consumption are due to the C-state transition overheads, as discussed before. Every time a new group of packets arrives, the CPU SUT has to move from the C_3 state to the C_0 one for the elaboration of the incoming traffic, increasing in this way the CPU uptime. In this respect, Figure 16 shows the average CPU uptime values, that is, the time the CPU has spent in the C_0 state. From this figure we can note that in the competitive case the CPU uptime using BT applications is higher than HTTP. This behaviour could depend on the available bandwidth shared by the peers and on the number and type of such peer connections; due to the fact that the selected movie was a popular download, the churn were high (peers arrive and leave the swarm over time), while the available bandwidth per peer was quite low (i.e., a fraction of the upload link in a typical ADSL connection). These dynamics and limitations lead to a greater use of CPU.

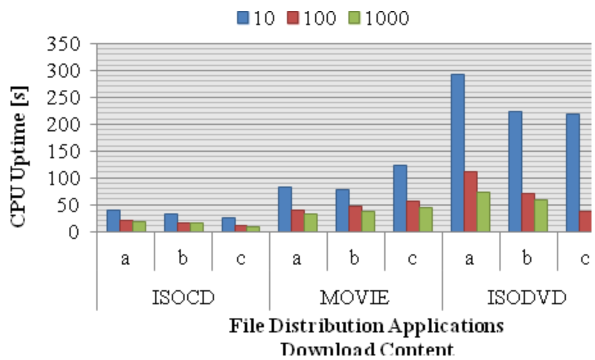


Figure 16. Average values of CPU uptime of file distribution applications

Profiling CPU SW activities

As demonstrated by the previous results, the CPU activity time is an important factor that affects the energy consumption of the SUT. In order to understand insights of the computational complexity of SUT SW components, we carried out some test using Oprofile. This tool, as introduced in section II, allows estimating which source-code function of the Linux kernel or of an application is utilizing the CPU, and which is its time share of the overall uptime. For a better understanding, we grouped together the source code functions into 6 categories, each one representing a different functional block of the SUT SW architecture. Table 2 shows the selected categories.

Table 2. Oprofile Function Categories for the Entire SUT SW

Name	Description
App	Vuze BT and Transmission BT applications
TCP/UDP	traffic processing at the TCP and UDP layer
ACPI	the ACPI and especially the functions related to idle-active transitions and vice versa
Sched	the operating system scheduler
IRQ	management of HW interrupts, mainly due to the NIC.
Other	other spurious sources of CPU activity

Figure 19, Figure 18 and Figure 17 report the CPU usage of the different categories in HTTP, Transmission and Vuze respectively. From these figures we can see the differences between this 3 file distribution applications. In Figure 19, we can notice that the App category is almost imperceptible. The weight of the Epiphany browser, used in our tests, does not affect the CPU activity time (up to 0.28%). This is because the browser uses the HTTP protocol and work with a single client-server connection for delivering content. Instead, in Figure 18 and Figure 17 we can see that the App category account for 20% - 35% of the CPU activity time with Vuze, and between 2% and 15% with Transmission BT client. This significant difference is due to the properties of these applications. Vuze is written in Java and needs a Java Virtual Machine (JVM) for a proper operation, while Transmission is a light-weight BT client written in C and can be run without external applications.

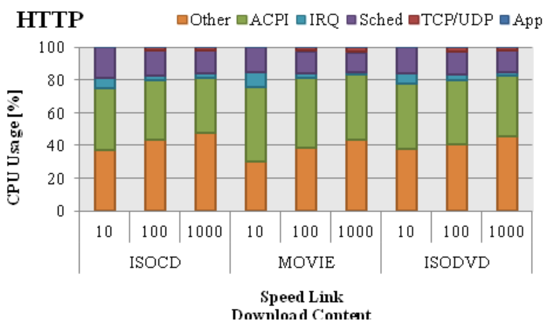


Figure 19. Breakdown of the CPU Usage of different download contents with HTTP protocol

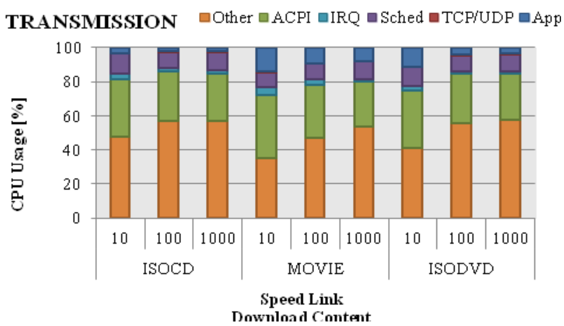


Figure 18. Breakdown of the CPU Usage of different download contents with a Transmission BT client.

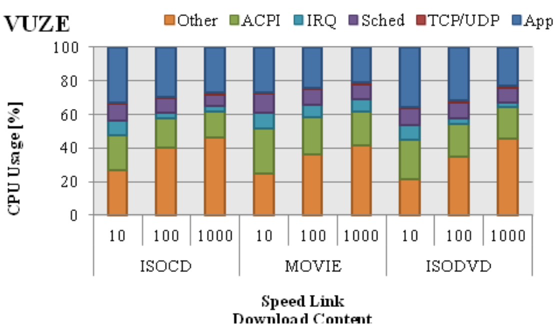


Figure 17. Breakdown of the CPU Usage of different download contents with a Vuze BT client

The TCP/UDP category appears to not significantly affect the CPU Uptime, up to 3% with HTTP (a single client-server connection), and less than 1% in Vuze and Transmission cases because the network management functionalities are considered as methods implemented in the program code of the application. The OS scheduler, who decides when to run which process, plays an important role in the CPU utilization, it accounts for 12% - 18% in HTTP, and between 8% and 12% in Vuze and Transmission cases. These values seem to be dependent on the HW interrupts, (2% - 6% in HTTP, 3% - 9% in Vuze, and 2% - 5% in Transmission). In fact, the OS scheduler is woken up by HW interrupts generated by the NIC for serving the incoming traffic. The OS scheduler is also recalled when the CPU completes its job backlog, and then is ready to enter into low power idle state, thereby executing ACPI functions, which in HTTP case range between 34% and 45% of the CPU uptime, 15% - 23% in Vuze cases, and 26% - 37% in Transmission cases. As we can notice, the ACPI category has a significant impact on the overall CPU uptime as a result of limitations on the speed link. From these results, we can see that the OS scheduler, IRQ and ACPI functions increase significantly in presence of lower bottleneck speeds (scenarios with a high number of active-idle transitions), thereby affecting the energy consumption of the SUT.

Conclusions

In this joint experimental activity, we evaluate the impact of the HTTP and BitTorrent file sharing systems on the energy behaviour of a final user networked device. For this purpose, we analyzed the case of collaborative and competitive downloads with the selected file distribution applications. From the obtained results, it is possible to note how in the collaborative cases BitTorrent exploits the ability to use multiple sources at the same time reducing in a significant way the downloading time, allowing lower energy consumptions, between 8% and 50%. In the competitive cases, the dynamics of the sharing peers influence the CPU behavior, increasing the CPU activity time due to the large number of active-idle state transitions, thereby increasing the energy consumption of the BitTorrent applications by 23% more than HTTP.

5. Development of the Trend Energy Saving eMulator (TESM)

Partners: CNIT, INRIA, TUB

Summary: During this activity, we have developed a tool for emulating virtual ISP networks, taking as input one of the network scenario provided by FT and one of the energy-aware algorithm developed in the literature.

Results: We have first built the network emulator tool. The network emulator is based on virtual machines emulating routers. Among the possible software available for virtualization, we have chosen Netkit (http://wiki.netkit.org/index.php/Main_Page). Netkit provides the possibility to create "light" virtual machines, each of them equipped with the basic command needed in a Linux distribution. In particular, the Quagga software is already available and it is used to emulate virtual routers.

In the first task of this work, we have first created an automated procedure to build a network of virtual machines already ready to work properly. In particular, we have generated a set of programs that, given as input the description of the topology in terms of links and OSPF weights, automatically performs the following task:

- creation of the virtual machines
- setting up the IP address plan
- setting up the interfaces for each node
- assigning the IP address to each interface
- building the configuration files for Zebra and Ospf
- setting up a server on each virtual machine to accept the command sent by the central controller to apply an energy-saving algorithm.

An example of input file containing the description of the topology is reported in the following:

```
1    1    33
2    1    35
```

The first number is the link ID, while the second and the third number are the link end-points (defined with IDs).

An example of file containing OSPF weights is the following:

```
8 23 20.000000
8 24 60.000000
```

where the first two numbers represents the end-points of the link while the third number is the link cost.

Based on this information the emulator generates the configuration file for each router. In particular, it first creates a start-up script like the following one:

```
1   /sbin/ifconfig eth0 151.100.1.1 netmask 255.255.255.0 broadcast 151.100.1.255 up
2   /sbin/ifconfig eth1 151.100.2.1 netmask 255.255.255.0 broadcast 151.100.2.255 up
3   echo "1" > /proc/sys/net/ipv4/ip_forward
4   chmod 775 /var/run
5   /usr/local/sbin/zebra -f /etc/zebra/zebra.conf & > /dev/null 2> /dev/null
6   /usr/local/sbin/ospfd -f /etc/zebra/ospf.conf & > /dev/null 2> /dev/null
7   perl /etc/serverclient/server.pl &
```

The script is run every time a router is booted. The main task of the script are the following ones:

- setting up of the network interfaces (IP address, netmask, and broadcast access) (line 1 and 2 in this case)
- abilitation of the forwarding capability needed to run the node as a router (line 3)
- starting the demons Zebra and OSPF, based on the configuration file automatically generated by the emulator (line 5 and 6)
- starting the program to accept the communication from the central controller (line 7).

Moreover, a configuration file for Quagga is created. An example is reported in the following;

```
hostname r1
password zebra
interface eth0
ip ospf hello-interval 10
ip ospf dead-interval 40
interface eth1
ip ospf hello-interval 10
ip ospf dead-interval 40
router ospf
router-id 151.100.2.1
network 151.100.1.0/24 area 0.0.0.5
network 151.100.2.0/24 area 0.0.0.5
line vty
no exec-timeout
```

```
15   log file /usr/local/etc/ospfd_r2.log
16   interface eth0
17   ip ospf cost 39
18   interface eth1
19   ip ospf cost 40
```

In particular, in line 1 and 2 the hostname and password are inizialized. Then, the OSPF settings, including the hello intervals, are setup (line 3-8). Moreover, the specific OSPF configuration settings are applied, including the router id and the network for each link (line 9-12). Finally, OSPF specific costs are assigned (line 16-19).

In order to run a centralized algorithm, we have developed a server program which is running on each machine. Here we report an example of the server:

```
use IO::Socket::INET;
open(STDOUT, '>',"/etc/serverclient/log.out") or die $!;
#the time function gives you the timestamp
open(STDERR, '>',"/etc/serverclient/log.err") or die $!;
# auto-flush on socket
$| = 1;
# creating a listening socket
my $socket = new IO::Socket::INET (
    LocalHost => '0.0.0.0',
    LocalPort => '7777',
    Proto => 'tcp',
    Listen => 5,
    Reuse => 1
);
die "cannot create socket $!\n" unless $socket;
print "server waiting for client connection on port 7777\n";
while(1)
{
    # waiting for a new client connection
    my $client_socket = $socket->accept();
    # get information about a newly connected client
    my $client_address = $client_socket->peerhost();

    my $client_port = $client_socket->peerport();
```

```
        print "connection from $client_address:$client_port\n";
        # read up to 1024 characters from the connected client
        my $data = "";
        $client_socket->recv($data, 1024);
        print "received data: $data\n";
        print "executing command\n";
        $result = `$data`;
        print "command executed\n";
        # write response data to the connected client
        #$data = "ok";
        $client_socket->send($result);
        # notify client that response has been sent
        shutdown($client_socket, 1);
    }
$socket->close();
```

The server accepts the connection on a TCP port, and executes the command received by the client. In this way, remote commands can be sent from the centralized unit to each router.

An example of client is reported in the following:

```
use IO::Socket::INET;
# auto-flush on socket
$| = 1;
# create a connecting socket
my $socket = new IO::Socket::INET (
    PeerHost => '151.100.32.1',
    PeerPort => '7777',
    Proto => 'tcp',
);
die "cannot connect to the server $!\n" unless $socket;
print "connected to the server\n";

# data to send to a server
my $req = "/usr/local/bin/vtysh -c \'en\' -c \'configure terminal\' -c \'router ospf\' -c \'energy_saving router-id_ex 151.100.55.1\' 2>&1";
```

```
14    my $size = $socket->send($req);
15    print "sent data of length $size\n";
16    # notify server that request has been sent
17    shutdown($socket, 1);
18    # receive a response of up to 1024 characters from server
19    my $response = "";
20    $socket->recv($response, 1024);
21    print "received response: $response\n";
22    $socket->close();
```

Specific commands to set up an energy saving algorithm are sent in lines 13-14. Moreover, the client checks the response of the server (line 19-21). The central controller implements an energy-aware algorithm to switch off the links of the network. In particular, we have implemented the ESIR algorithm proposed in [37]. The Energy Saving IP Routing Strategy (ESIR) is designed to be integrated into the OSPF routing protocol. The path computation strategy, realized by means of a modified Dijkstra algorithm, is distributed and fully compatible with OSPF mechanisms. At the same time, ESIR is able to satisfy QoS requirements by maintaining traffic load on all the network links under fixed configurable values. The modified version of the Dijkstra algorithm is able to select a subset of paths to route the traffic, leaving the unused interfaces to enter low power mode. The main advantage of this approach is that the IP topology does not change, and consequently no exchange of Link State Advertisements packets is needed. To achieve this goal, the set of routers is divided in importers and exporters. The main idea is the exportation mechanism, which allows to share a Shortest Path Tree between neighbors routers, so that the overall set of active links is minimized. Only the importer routers modify their shortest path tree, starting from the tree of the exporter routers. This allows to avoid to trigger an entire path re-computation inside the network. In order to work properly, the exporter has to be the neighbor of the importer. Moreover, an importer can be object of a single exportation, i.e. it can receive the shortest path tree from a single exporter. Finally, once a node has become an exporter, its shortest path tree cannot be modified any more.

We have implemented the modified version of the Dijkstra algorithm used by ESIR in the Quagga software, and then we have modified the Netkit software in order to automatically create the virtual machines with the modified Quagga software already installed. In particular, we have performed the following steps:

- launching a single virtual machine with writing permission on the original file system of Nettkit, with the following command

  ```
  vstart vm --eth0=tap,10.0.0.1,10.0.0.2 -M 256 -W
  ```
  In this way, a virtual machine called vm is created, with 256 MB of RAM, writing permission on the Netkit filesystem (option -W) and a tap interface to connect the virtual machine with the host machine
- we configure a DNS server inside the virtual machine by modiyfing the file /etc/resolv.cong

- we remove the original version of Quagga with the command apt-get --purge remove quagga
- we remove unstable from /etc/apt/sources.list, since for default Netkit uses also the unstable release of the packages
- we update the repository by running the apt-get update command
- we install the gcc compiler, and the following packages that are needed for the Quagga installation: gawk, make, libreadline5, libreadline-gplv2-dev
- we copy the modified source code of Quagga inside the virtual machine
- we add the Quagga user with adduser quagga
- inside the folder contaning the modified version of Quagga, we install the package by running the following commands:
  ./configure --enable-vtysh
  make
  make install
  chown root.quagga /var/run
- Finally, we set up the permission of the folder /var/run and we run the command ldconfig to update the caches.

At the end of the process, each new virtual machine created includes the modified version of Quagga.

In our experiments, we have chosen the reference network provided by Orange in WP3. In particular, the network is composed of 38 nodes and 72 links. A screenshot of the 38 virtual machines emulating the network is reported in Fig. 20

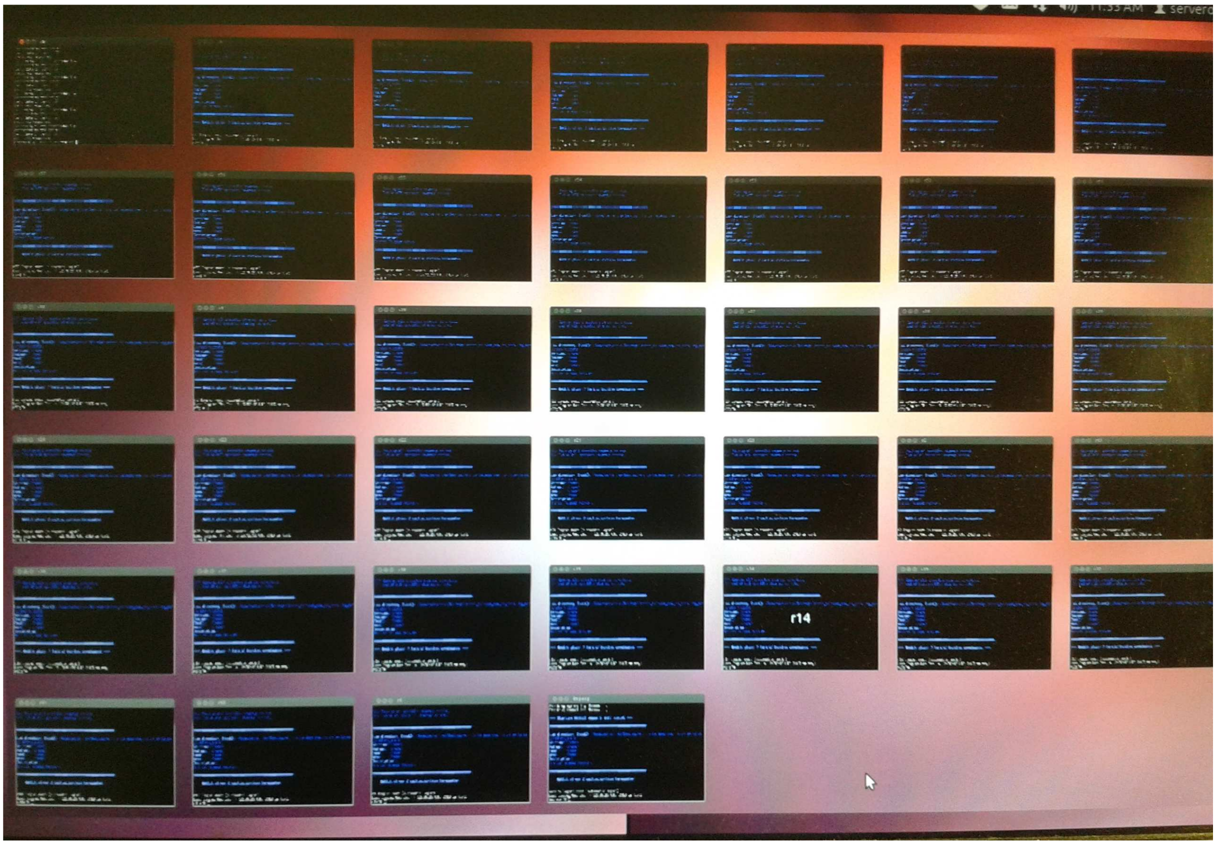


Fig. 20: The 38 virtual machines emulating the Orange scenario

Given the network provided by Orange, we have tested the effectiveness of the proposed algorithm. In particular, we have used the set of traffic matrices provided by Orange, and we have chosen the matrix corresponding to the off-peak traffic. Fig. 21 reports the network configuration obtained. The links that are powered off are reported in green lines. As can be seen, ESIR is able to power off some links in the network, still guaranteeing connectivity and QoS constraints. Fig. 22 and Fig. 23 reports instead the network configuration considering the mid-peak and the peak traffic matrices. Clearly, since traffic is higher compared to the off-peak, the number of links powered off is lower compared to the off peak case.

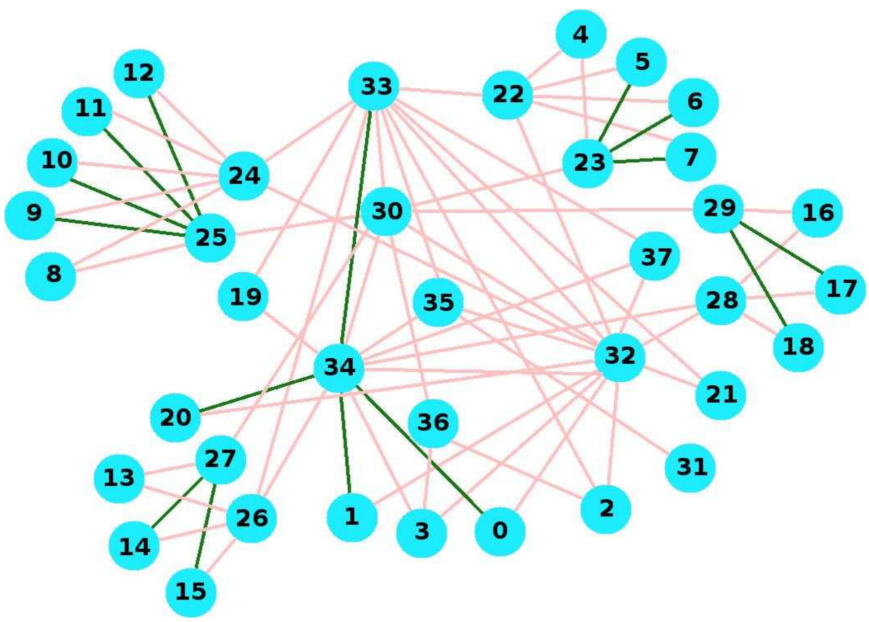


Fig.21: The off-peak network configuration. The green links are the ones powered off.

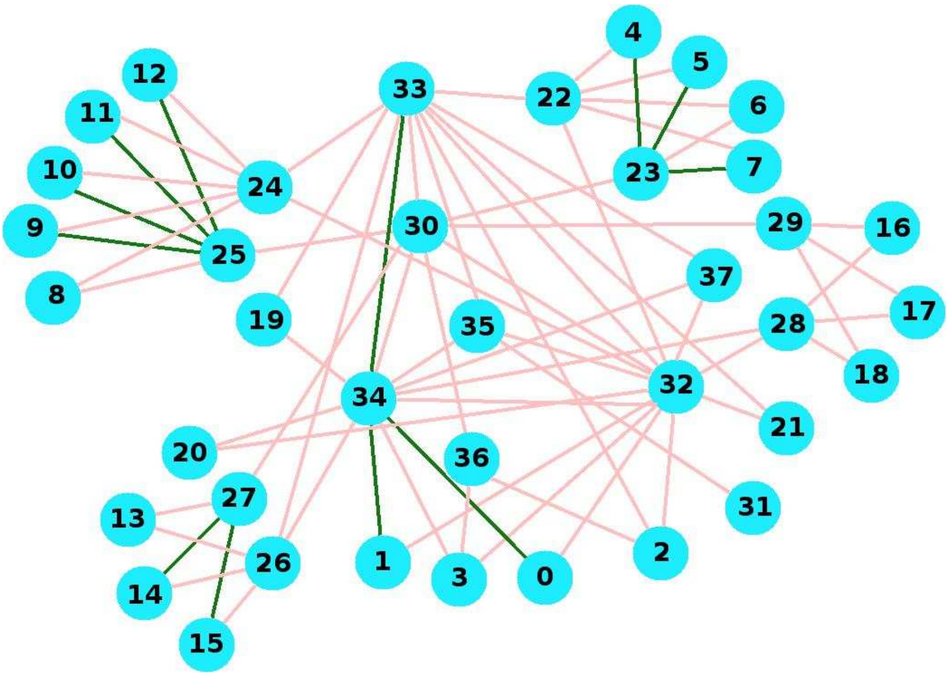


Fig. 22: The mid-peak network configuration.

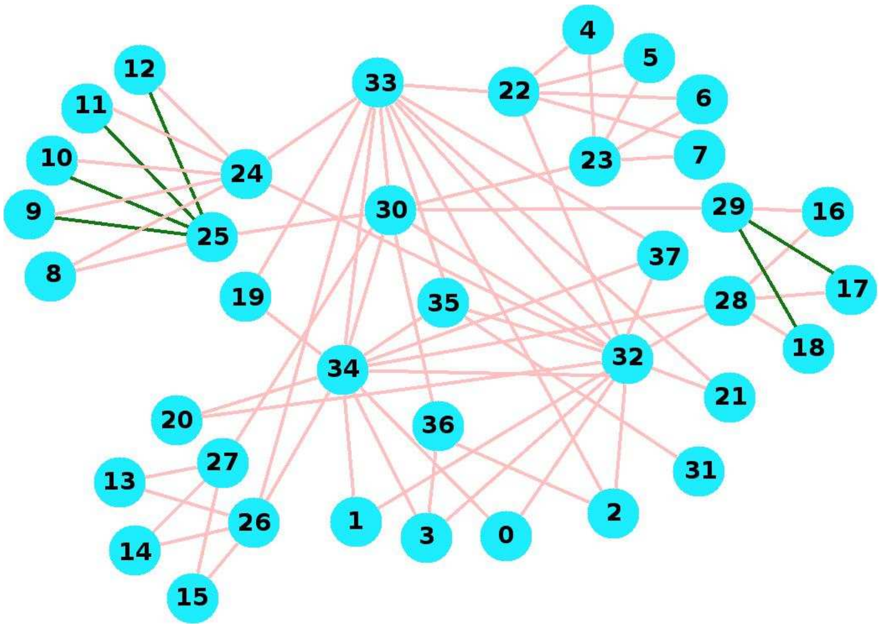


Fig. 23: The peak network configuration.

Papers

Published papers

Involved partners	Authors	Title	Conference/ Journal	Presentation/ publication date
A-LBLF, PoliTo, TUB, CNIT-Genoa, CNIT-UniRoma1, UC3M, FUB, UTH	I. Haratcherev, M. Meo, Y. Zhang, Y. Hu, A. Conte, F. Idzikowski, Ł. Budzisz, F. Ganji, R. Bolla, O. Jarmilo Ortiz, R. Bruschi, A. Cianfrani, L. Chiaraviglio, A. Coiro, R. Gonzalez, C. Guerrero, E. Tego, F. Matera, S. Keranidis, G. Kazdaridis, and T. Korakis	The TREND Experimental Activities on "green" Communication Networks	Tyrrhenian International Workshop on Digital Communications, Genoa, Italy	24/09/2013
CNIT	R. Bolla, R. Bruschi, O. M. Jaramillo Ortiz, P. Lago	The Energy Consumption of TCP	Proc. of 3rd ACM/IEEE International Conf. on Future Energy Systems (e-Energy 2013)	May 2013
CNIT	R. Bolla, R. Bruschi, O. M. Jaramillo Ortiz, R. Rapuzzi	Enabling the TCP Segmentation Offload to Save Energy	Proc. of the 24th Tyrrhenian International Workshop on Digital Communications	Sept. 2013

Planned (or submitted) papers

Involved partners	Authors (if known)	Title / Topic	Conference/ Journal	Planned date
FUB, TUB, CNIT-	E. Tego, F. Idzikowski, L. Chiaraviglio,	Facing the reality: validation of energy saving mechanisms on	Journal of Electrical and Computer Engineering	submitted

UniRoma 1	A. Coiro, and F. Matera	a testbed		
TUB	F. Ganji, A. Zubow, Ł. Budzisz, A. Wolisz	An experimental study on the detection of WLAN-user connectivity initiation in the low SNR regime	Under discussion	planned
TUB, PoliTo	F. Ganji, , Ł. Budzisz, F. Getachew, N. Li, M. Ricca, Y. Zhang, M. Meo, A. Wolisz	Extraction of usage and mobility patterns of campus- WLAN users	Under discussion	planned
TUB, PoliTo	F. Ganji, , Ł. Budzisz, F. Getachew, N. Li, M. Ricca, Y. Zhang, M. Meo, A. Wolisz	Extraction of usage and mobility patterns of campus- WLAN users (extended version)	Under discussion	planned

Mobility

Past mobility

Involved partners	Person	Topic	Period
TUB, FUB, CNIT-UniRoma1	F. Idzikowski	TESM and FUFL on the FUB testbed (WP3 and WP4)	16/03/2013 – 20/03/2013
PoliTo, TUB	F. Ganji	Collection of WLAN traces in dense WLAN to verify on/off strategies proposed in WP2 (WP2 and WP4)	21/05/2013 – 31/05/2013
PoliTo, TUB	F. Ganji	Extraction of usage and mobility patterns of campus WLAN users (WP2 and WP4)_	30/06/2013 – 12/07/2013
PoliTo, TUB	F. Idzikowski	Energy-aware IP-over-WDM networks, experimental activities, big picture, and teaching on metro/core networks (WP3 and WP4)	01/07/2013 – 04/07/2013
UC3M - CNIT	Roberto Gonzalez	Experimental activity for measurement and management the energy cost of networking intensive applications: P2P and OSN	20/01/2013 - 25/01/2013
CNIT – UC3M	Olga Jaramillo	Experimental activity for measurement and management the energy cost of networking intensive applications: P2P and OSN	03/06/2013 - 07/06/2013
CNIT – PoliTO	Olga Jaramillo	Participation to the TREND PhD School	01/07/2013 - 05/07/2013

References

[1] W. Lemstra and V. Hayes, J. Groenewegen,“ The Innovation Journey of Wi-Fi: The Road To Global Success,” Cambridge Univ. press, 2011.

[2] F. Ganji, Ł. Budzisz, and A. Wolisz, “Assessment of the power saving potential in dense enterprise WLANs” TKN Group, TU Berlin, Tech. Rep. TKN-13-003, Apr. 2013.

[3] M. Ajmone Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, “A Simple Analytical Model for the Energy-Efficient Activation of Access Points in Dense WLANs,” Proc. e-Energy, Passau, Germany, Apr. 2010.

[4] A. P. Couto da Silva, M. Meo, and M. Ajmone Marsan, “Energy-Performance Trade-off in Dense WLANs: a Queuing Study,” Computer Networks, vol. 56, no. 10, pp. 2522–2537, July 2012.

[5] Cisco EnergyWise Technology, http://www.cisco.com/en/US/products/ps10195/index.html, Last access: June, 2013

[6] Ł. Budzisz , F. Ganji , G. Rizzo, M.A. Marsan, M. Meo, Y. Zhang, G. Koutitas, L. Tassiulas, S. Lambert, B. Lannoo , M. Pickavet , A. Conte , I. Haratcherev , and A. Wolisz, “Dynamic Resource Provisioning for Energy Efficiency in Wireless Access Networks: a Survey and an Outlook,” submitted to IEEE Communications Surveys and Tutorials, September 2013.

[7] “Cisco Aironet 1260 Series Access Point Data Sheet,” [Online]. Available: http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps10980/data_sheet_c78-593663.html

[8] “Cisco Wireless Control System Configuration Guide, Release 7.0”, [Online]. Available: http://www.cisco.com/en/US/docs/wireless/wcs/7.0/configuration/guide/WCS70cg.html

[9] T. Henderson, D. Kotz, and I. Abyzov, “The changing usage of a mature campus-wide wireless network,” In Proc. of the 10th annual international conference on Mobile computing and networking. ACM, 2004.

[10] F. Ganji , Ł. Budzisz , A. Wolisz , ”Assessment of the Power Saving Potential in Dense Enterprise WLANs,” 24th annual IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'13), London, Great Britain, September 2013.

[11] “CRC-CORAL USER’S GUIDE (CRC-COgnitive Radio Aware Learning Platform User’s Guide) Version 2.0,” Communications Research Centre Canada, 2010.

[12] R&S- FSV product brochure, “R&S®FSV Signal and Spectrum Analyzer-Signal analysis at its best,” Rohde and Schwarz, 2009- 2011.

[13] A. Vlavianos, L.K. Law, I. Broustis, S.V. Krishnamurthy, and M. Faloutsos, “Assessing link quality in IEEE 802.11 Wireless Networks: Which is the right metric?,” In Proc. of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008.

[14] W.Lemstra, V.Hayes, J. Groenewegen,“ The Innovation Journey of Wi-Fi: The Road To Global Success,” Cambridge University press, 2011.

[15] A. Valenti, A. Rufini, S. Pompei, F. Matera, S. Di Bartolo, C. Da Ponte, D. Del Buono, and G. Beleffi, “QoE and QoS comparison in an anycast digital television platform operating on passive optical network,” in Proc. of the Networks, Rome, Italy, October 2012.

[16] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz, “Dynamic routing at different layers in IP-over-WDM networks – maximizing energy savings,” Optical Switching and Networking, Special Issue on Green Communications, vol. 8, no. 3, pp. 181–200, July 2011.

[17] F. Idzikowski, L. Chiaraviglio, R. Duque, F. Jiménez, and E. Le Rouzic, “Green horizon: Looking at backbone networks in 2020 from the perspective of network operators,” in Proc of the ICC, Budapest, Hungary, June 2013.

[18] I. Haratcherev, M. Meo, Y. Zhang, Y. Hu, A. Conte, F. Idzikowski, Ł. Budzisz, F. Ganji, R. Bolla, O. Jarmilo Ortiz, R. Bruschi, A. Cianfrani, L. Chiaraviglio, A. Coiro, R. Gonzalez, C. Guerrero, E. Tego, F. Matera, S. Keranidis, G. Kazdaridis, and T. Korakis, "The TREND Experimental Activities on “green” Communication Networks," in Proc. of Tyrrehenian International Workshop on Digital Communications, Genoa, Italy, September 2013.

[19] E. Tego, F. Idzikowski, L. Chiaraviglio, A. Coiro, and F. Matera, "Facing the reality: validation of energy saving mechanisms on a testbed," submitted to Journal of Electrical and Computer Engineering.

[20] “MiDORi Network Technologies Project (MiDORi),” http://midori.yamanaka.ics.keio.ac.jp/, July 2009.

[21] A. Morea, S. Spadaro, O. Rival, J. Perelló, F. Agraz, and D. Verchere, “Power Management of Optoelectronic Interfaces for Dynamic Optical Networks,” in Proc. of the ECOC, Geneva, Switzerland, September 2011.

[22] A. Morea, J. Perelló, F. Agraz, and S. Spadaro, “Demonstration of GMPLS-controlled device power management for next generation green optical networks,” in Proc. of the OFC, Los Angeles, USA, March 2012.

[23] D. Clark, “The Design Philosophy of the DARPA Internet Protocols”, Proc. SIGCOMM 1988.

[24] Ares Blanco, Jairo, and Eduard Aibar Puentes. "The evolution of P2P networks for file exchange: The interaction between social controversy and technical change." Actes d'Història de la Ciència i de la Tècnica 4 (2012): 55-74.

[25] SAROIU, S., GUMMADI, K. P., DUNN, R. J., GRIBBLE, S. D., AND LEVY, H. M. 2002. An Analysis of internet content delivery systems. In Proceedings of the 2002 USENIX/ACM Symposium on Operating System Design and Implementation.

[26] GESI, “Smart 2020,” report, DOI=http://www.smart2020.org/_assets/files/02_Smart2020Report.pdf

[27] Bolla R., Bruschi R., Carrega A., Davoli F., Suino D., Vassilakis C., Zafeiropoulos A., “Cuting the energy bills of Internet Service Providers and telecoms through power management: An impact analysis,” Computer Networks, Elseiver, vol. 56, no. 10, pp. 2320-2342, July 2012.

[28] Bolla R.,Bruschi R., Christensen K., Cucchietti F., Davoli F., Singh S., “The Potential Impact of Green Technologies in Next Generation Wireline Networks – Is There

Room for Energy Savings Optimization?," IEEE Communications Magazine, vol. 49, no. 8, pp. 80-86, Aug. 2011.

[29] K. Cameron, R. Ge, and X. Feng. Designing Computational Clusters for Performance and Power. Advances in Computers, 69(2), May 2007.

[30] The Advanced Configuration and Power Interface (ACPI), DOI=http://www.acpi.info/

[31] http://www.vuze.com/

[32] http://www.transmissionbt.com/

[33] Intel® Core™ i5 Processor Datasheet, DOI=http://download.intel.com/design/processor/datashts/322164.pdf

[34] "Energy-Efficient Platforms – Considerations for Application Software and Services," Intel Whitepaper March 2011, http://download.intel.com/technology/pdf/Green_Hill_Software.pdf

[35] R. Bolla, R. Bruschi, O. M. Jaramillo Ortiz, P. Lago, "The Energy Consumption of TCP" Proc. of the 4th International Conference on Future Energy Systems, e-Energy'13, Berkeley, USA, May 2013.

[36] Oprofile, DOI= http://oprofile.sourceforge.net

[37] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, "An OSPF-Integrated Routing Strategy for QoS-aware Energy Saving in IP backbone Networks", IEEE Transactions on Network and Service Management, vol. 9, no. 3, September 2012.