Implementation and Evaluation of a Sleep-Proxy for Energy Savings in Networked Computers

Enida Sheme
Department of Computer Science
Polytechnic University of Tirana
enida.sheme@gmail.com

Neki Frasheri
Department of Computer Science
Polytechnic University of Tirana
neki.frasheri@gmail.com

Marin Aranitasi
Department of Computer Science
Polytechnic University of Tirana
m_aranitasi@hotmail.com

ABSTRACT
In enterprise networks, idle desktop machines rarely sleep, because users and IT departments want them to be always accessible. While some solutions have been proposed, few of them have been implemented even more evaluated in real network environments. In this paper, we implement and evaluate a sleep proxy system, based on existing proposed architecture for this Proxy.

This system is tested in 6 different PC machines of a real network. The results of the experiments show that machines can sleep almost 55% of the experimenting time (which is translated into energy savings) while maintaining their network accessibility to user satisfaction. However, there is a need of "cooperation" between IT procedures and sleep proxy system in order to gain better performance and thus less dissipated energy.

Keywords
Power consumption, energy savings, networked computers, network traffic, proxying Ethernet adapter

1. INTRODUCTION
There has been a lot of researches lately regarding the wasted energy consumed especially by the IT environment, [11], [3], [6], [1]. One of the most frequent cases is that of office machines left on despite of user activity on them.

The focus of this paper is exactly enterprise's PCs, the power management of which would save up significant energy.

The main reason PC users and IT administrators want to leave PCs on all the time is they want to access them whenever they want to, or whenever it is planned a broadcasting update or process. Among many ideas, lately the focus is back to Sleep Proxy concept, which was first introduced in 1998 [9] and later researched. A Sleep Proxy allows a PC to pass into a Sleep state, meanwhile it preserves PC's network presence and awakens it if necessary. The architecture of this Sleep Proxy system is composed by a client which periodically updates the server with information it needs to represent this client while it is asleep, and the server (Sleep Proxy) that based on a designed policy on how to manage network traffic protocols, will eventually ignore some packets, answer to some requests, and awake the sleeping computer for other packets.

The aim of developing a Sleep Proxy system is to encourage users to enable power management features, assuring them that they can have access to their computers even when they are asleep. So, in our work we emphasize this by testing a real system in an office environment, showing and highlighting the differences between non-proxy system and proxy system network regarding their energy consumption.

In Section 2, we mention related articles, papers and publications regarding this area. In Section 3 we will give a preview of the design and implementation of Sleep Proxy server, how are all the components related to each other in Hardware and Software level. Section 4 describes the experimental environment, also the tests regarding the power consumption without and with Proxy server implemented in the experiment, in order to get an idea of energy savings this system can provide. The results of the experiments and their interpretation are given in section 5, where we emphasize the need of synchronization between IT staff and the Proxy system in order to get the best results. At last we come with our conclusions.

2. RELATED WORK
The Sleep Proxy idea was first introduced in 1998 by Kenneth Christensen in an International Journal [9]. Although there were some technologies such as "Wake on LAN" and "Advanced Configuration and Power Interface Specification" in 90's regarding saving energy issue, due to some of their disadvantages, there are no more researches in the area. Instead, attention was back to Sleep Proxy in 2004 by Bruce Nordman [8]. An interesting paper comes in 2007 for an implemented Proxy specifically designed for Universal Plug and Play Protocol (UPnp) [12]. Later in 2007, Bruce Nordman and Keneth Christensen came with another work where they propose solution [10] for a Sleep Proxy which can manage protocols such as ARP, DHCP, ICMP. Another scheme similiar to Sleep Proxy comes in 2008 called Somniloquy [13]. It offers a hardware implementation of Sleep Proxy in a so-called "gumstix", thought as a predecessor of future NIC. We can also mention [4]: Sleep Proxy as a standard of European Computer Manufacturers Association in November 2009. Researches go on and another paper of 2010' [7], illustrates designing, implementing and testing of a Sleep Proxy in Microsoft Offices.

3. DESIGN AND IMPLEMENTATION ISSUES
In this section we show some design and implementation issues based on previous known architecture of Sleep Proxy Server, expressed in papers such as [2], [8] and [9]. The Sleep Proxy system is based on a client/server architecture where every client can fall asleep, being represented on the network by the Sleep Proxy server.

BCI'12, September 16–20, 2012, Novi Sad, Serbia.

Local Proceedings also appeared in ISBN 978-86-7031-200-5, Faculty of Sciences, University of Novi Sad.

3.1 Client Server Communication Protocol

Based on the characteristics of what we want our Sleep Project to do, we specify below the client server communication. That means what will the server do and what will the client do in our system.

The client will eventually:

- provide to the proxy server necessary information that server will need in order to represent the sleeping computer on the network. This will be done by periodically (every 10 seconds) sending an INFO message that contains these data: PC_name, IP address, MAC address, Operating System, state, local time
- "request" to the proxy server to change his status to sleep when it is going to sleep. This will happen automatically by server when it doesn't get an INFO message after 20 seconds (twice the time of default time set for periodic sending of INFO)
- notify the server whenever the client is awaked by any event. This will happen by automatically sending INFO
- know and process any waking packet that comes from the server (which means to support some kind of WoL technology)
- optionally can configure the default time for periodic sending of INFO, IP address of server, communication port with the server. The default values are set in the programming code.

The server will eventually:

- be listening to INFO messages from the clients and store these data in a table
- keep a list of tables to keep track and information about computers on the network
- change status of a computer from Sleep to Awake in its list whenever that clients notifies it is awaked (it gets INFO messages)
- prevent any network traffic to wake the sleeping computer in vain by ignoring or answering to these kind of packets
- wake up the sleeping computer by sending a WoL packet whenever there is a network traffic too important not to be answered by the client itself

3.2 Implementation Issues

Based on the survey of [11] regarding network chattering, we decide to implement this reaction policy for the Proxy server:

- it will answer back to ARP requests, will send ARP probes broadcasts on the network when necessary
- wake up sleeping computer by sending WoL packet in case of TCP requests toward that computer
- ignore all the others

Programming this system will need:

- Java RMI for building the client / server applications (as a programming environment is chosen "Eclipse IDE for Java Developers", Version 3.5.2 "Helios Service Release"),
- Jpcap library (Java packet capturing) which serves for capturing, analyzing and processing network packets.

4. EXPERIMENTS

4.1 Experiment Preview

The experiment environment is composed by 6 computers clients and one server. 6 computers have these parameters:

- 4 of them: Processor Intel(R) Core Duo CPU 2.0 GHz, RAM 1 GB, HDD 160 GB, Operating System 32-bit Windows XP, TFT screen
- 2 of them: Processor Intel(R) Core Duo CPU 2.0 GHz, RAM 1 GB, HDD 160 GB, Operating System 32-bit Windows 7.

The server has these parameters: Processor Intel(R) Core Duo CPU 2.4 GHz, RAM 2 GB, HDD 160 GB, Operating System 32-bit Windows 7.

In every client is installed Java JDK and executes the client.jar application, which is the client Proxy application. Meanwhile the server executes the Server Proxy application. The network topology is Bus.

There will be held these experiments to test the Sleep Proxy system. *The first testing*: we will measure the consumed energy and the network traffic (number of packets) passing to them during the time 16:00 to 8:00 of the next day, for three continuous days, providing so 3 samples for each computer, without "sleep proxy" implemented. *The second testing*, will include the 7-th computer, the server Sleep Proxy, and we will measure the same parameters for 16 hours, in three days. So we will have 3 other samples for each of 6 client computers. Based on the results of the experiments we will compare the energy consumed and network traffic, also the time of each computer passed on sleeping state with Sleep Proxy system implemented, which is automatically translated into energy savings.

4.2 Measuring Tools

4.2.1 Joulemeter

The power consumption is measured by a specific tool produced by Microsoft called Joulemeter. We will measure the power consumption for 16 hours, for 3 days.

4.2.2 Wireshark

Wireshark is the program from which we will get information about the network traffic in the experimental client machine, in terms of number of packets, saved in generated .pcap files.

5. RESULTS AND INTERPRETATION

The first testing results regarding the power consumption and total number of packets are shown in Table 1 below.

Table 1: 3-day Average Power Consumption and total nr. of packets without implemented proxy on the network

	3-day Average	
	Power Consum. (W)	*Total number of packets*
C1	*132.9*	*208734*
C2	*117.7*	*56450*
C3	*115.15*	*139321*
C4	*115.46*	*182600*
C5	*118.9*	*196852*
C6	*134.1*	*289457*

As we can see from the table, Computer C1 and C6 have higher power consumption and this happens because of the CRT screen that consumes more power. The same data of Table 1 are shown in a graphical form in Figure 1.

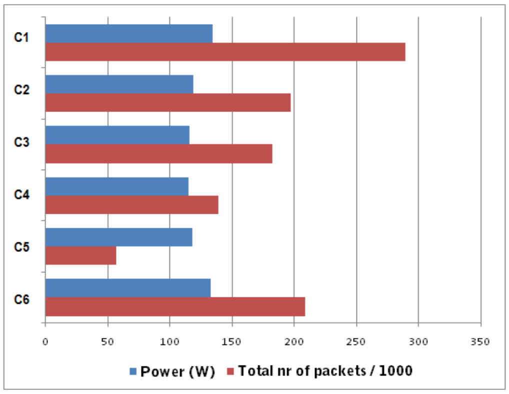

Figure 1: Graphical representation of 3-day Average Power Consumption and total number of packets without implemented proxy on the network.

The results about the second testing, when Sleep Proxy is implemented on the network, regarding power consumption and total number of packets are shown in Table 2, also in a graphical form in Figure 2:

Table 2: 3-day Average Power Consumption and total number of packets with implemented proxy on the network

	3-day Average	
	Power Consum. (W)	*Total number of packets*
C1	*132.8*	*107337*
C2	*117.4*	*49413*
C3	*115.17*	*71492*
C4	*115.5*	*72880*
C5	*118.16*	*97320*
C6	*134.16*	*122369*

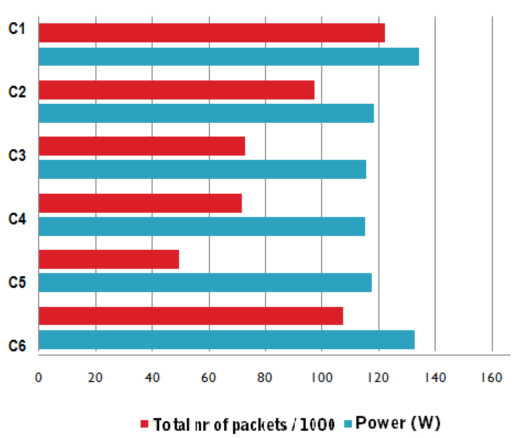

Figure 2: Graphical representation of 3-day Average Power Consumption and total number of packets with implemented proxy on the network.

During the 16 experimenting hours, using a Sleep Proxy server, it was provided a percentage of sleeping time in a range of 44% to 65%, with an average sleeping time of 8.71 hours, illustrated in Table 3 and Figure 3.

Table 3: Average and percentage of sleeping hours during the 16 experimental hours with Sleep Proxy

Computer	Average	% sleep hours
C1	8.1	50%
C2	7.2	44%
C3	9.6	60%
C4	10.4	65%
C5	8.8	55%
C6	8.16	51%

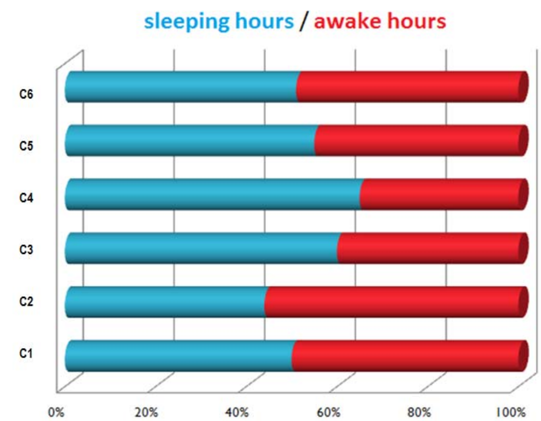

Figure 3: Percentage of experiment hours in sleeping or awake state with implemented proxy.

Now using the formula Energy = Power * Time, we calculate each of 6 computers * 16 experimental hours = energy consumed (kWh) per each computer. It is computed that the daily average energy consumption per computer is 1.97 kWh. Translated in annual energy consumption, if 6 computers would be left on from Monday to Thursday afternoons, it is computed an average energy consumption of 376kWh per each computer, or about 376 MWh for 1000 computers of a big company in a year.

Meanwhile using the Sleep Proxy system on the experimental network shows these calculated results:

*Energy consumed = 3W * sleeping hours + Power consum. * Awake hours*

It is computed that, using Sleep Proxy, the daily average energy consumption per computer is 0.92 kWh. In comparison to the value just given above, without Sleep Proxy of 1.97 kWh, we save energy up to 53%. So, computers with proxy consume 53% less power than without proxy. The annual calculation would be 176 MWh for 1000 computers in a year.

The difference with and without proxy is approximately 208MWh, translated into 54.2% energy savings.

The figures 4 and 5 show respectively the difference in a graphical form of Daily Average energy consumed and Daily Average number of packets in the NIC by the client with and without proxy.

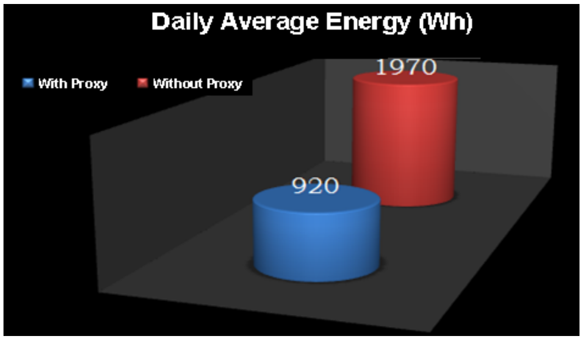

Figure 4: Difference in daily average energy consumed by the client with and without proxy.

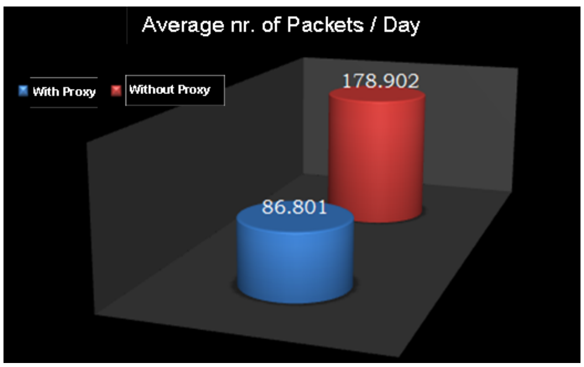

Figure 5: Difference in daily average number of packets in the client NIC with and without proxy.

6. CONCLUSIONS

The results of this paper highlights the importance of power management features application and encourage this by evaluating a Sleep Proxy system, making a comparison of energy consumed between two situations: without Sleep Proxy implemented, and with Sleep Proxy implemented on the network.

The system we implemented can provide significant energy savings, assuring computers's availability on the network and a high level of transparency for other computers of the network. Due to our experiments, we calculate that in our system of 6 computers, daily average energy consumption in a Sleep Proxy system is 0.92 kWh per each computer. Compared to the daily average energy consumption value without implemented Sleep Proxy, of 1.97 kW, *the energy savings results up to* 53%, so computers in a Sleep Proxy environment consume 53% less energy than the same computers in a Non-Sleep Proxy environment. These values belong to only one computer on the network, but as the size of the company grows the energy saving becomes more and more significant in an annual period, up to hundreds of MWh. The energy savings is achieved by longer sleeping time in the presence of a Sleep Proxy Server. In a non-Sleep Proxy environment, sleeping time is almost 0 hours. Meanwhile the results of the experiments with Sleep Proxy implemented, show that during 16 hours of experimental time, the 6 computers stayed in sleeping time from 44% to 65% of the time, translated in 7.2 to 10.4 sleeping hours, from a total of 16 hours. Knowing that Sleep proxy is a system which is closely related to other Software components, brings to the need for some Software policies that would help in a better performance of the Sleep Proxy system. We can mention here procedures and routines such as backup, patches, automatic updates, scheduled by the IT department. Of course, IT department policies and schedules regarding these fields play a non-trivial role in Sleep Proxy performance. Sleep Proxy is active during non-official time (16-8 of the next day) so it would be much helpful from the IT department to be synchronized with this system in order to get the best results from Sleep Proxy.

7. REFERENCES

[1] Allan Chen "Berkeley Lab Researcher Leads Development of a Standard to Reduce Energy Use of Computers Connected to the Internet".

[2] Bruce Nordman, Ken Christensen "Improving the Energy Efficiency of Ethernet-Connected: A Proposal for Proxying", Version 1.0, September 2007.

[3] Chamara Gunaratne, Ken Christensen and Bruce Nordman "Managing energy Consumption Costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed", *INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT, October 2005.*

[4] ECMA Standards, "Final Draft, Proxzzzy for sleeping hosts"; November2009,www.ecma-international.org /publications/files/ECMA-ST/ECMA-393.pdf

[5] "EPA Announces New Computer Efficiency Requirements" – October 2006, http://yosemite.epa.gov/opa/admpress.nsf/4a3d7e51caf96c7a85257359003f533e/113b0c0647fee41585257210006474f1!OpenDocument

[6] Greg Shultz "Use Windows XP's power management features to lower energy bills", November 2002, http://articles.techrepublic.com/510010878111054827.html

[7] Joshua Reich, Michel Goraczko, Aman Kansal, Jitendra Padhye – "Sleepless in Seattle No Longer", Microsoft Research, *Proceedings of the 2010 USENIX Conference, June 2010.*

[8] K. Christensen, C. Gunaratne, B. Nordman, and A. George, "The Next Frontier for Communications Networks: Power Management," *Computer Communications, Vol. 27, Nr. 18, f. 1758-1770*, 2004.

[9] Kenneth J. Christensen, Franklin 'Bo' Gulledge – "Enabling Power Management for Network-attached Computers", *International Journal of Network Management*, Vol. 8, Nr.2, pp. 120-130, 1998.

[10] M. Allman, K. Christensen, B. Nordman, dhe V. Paxson, "Enabling an Energy-Efficient Future Internet Through Selectively Connected End Systems," Sixth Workshop on Hot Topics in Networks (HotNets-VI), November 2007.

[11] Sergiu Nedevschi, Jaideep Chandrashekar, Junda Liu, Bruce Nordman, Sylvia Ratnasamy, Nina Taft – "Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems", *NSDI Conference, April 2009.*

[12] UPnP Low Power Architecture, Version 1.0, UPnP Forum, August 27, 2007.

[13] Yuvraj Agarwal, Steve Hodges, James Scott, Ranveer Chandra, Paramvir Bahl, Rajesh Gupta "Somniloquy: Maintaining Network Connectivity While Your Computer Sleeps", 2008.