Virtualizing Home Gateways for Large Scale Energy Reduction in Wireline Networks

Jean-Patrick Gelas[245], Laurent Lefèvre[145,] Teferi Assefa[3], Mulugeta Libsie[3]

[1] INRIA, France
[2] Université Claude Bernard, Lyon 1, France
[3] Addis Ababa University, Addis Ababa, Ethiopia
[4] LIP Laboratory, Ecole Normale Supérieure de Lyon, France
[5] University of Lyon, France

* Corresponding Author, laurent.lefevre@inria.fr, +33 4 72 72 82 28

Abstract

About 80-90% of the energy in today's wireline networks is consumed in the access network, with about 10W per user being dissipated mostly by the customer premises equipment (CPE). Home gateway is a popular equipment deployed at the end of networks and supporting a set of heterogeneous services (from network to multimedia services). These gateways are difficult to manage for network operators and consume a lot of energy. This paper explores the possibility to reduce the complexity of such equipment by moving services to some external dedicated and shared facilities. This paper will present first result towards virtualizing home gateway services towards some specific part of the network. When combined to quasi passive CPE, this approach can reduce the energy consumption of wired networks infrastructures. This research is done within the GreenTouch initiative which aims to increase network energy efficiency by a factor of 1000 from current levels by 2015 (http://www.greentouch.org).

1 Introduction

Recently energy saving and reduction of carbon emission are the concern of the whole world, and of all industries including Information & Communication Technology (ICT), due to energy cost rising and its impact to the environment [7, 8].

Hence, optimization of energy usage and minimizing its impact to the environment in the ICT sector becomes a crucial issue that requires immediate solutions. Above all, for network operators, working on energy efficiency is not only a matter of environmental protection, but also vital to minimize an ever increasing energy cost for their deployment of future network infrastructure.

To come up with a solution for such a problem, researchers tried to examine and identify the overall energy consumption share of the various parts of the telecom infrastructure. Based on their study, the majority share of the current energy consumption profile of a network infrastructure, more than 80% as in [1], is consumed by the equipment in customer premises.

Today end-users host Internet access box at home generally provided by their Internet provider. Those boxes are initially used to convert signal and protocols (Cable, xDSL, Optical) for the home network (wired and wireless). They also embed several services like a DHCP server, a NAT service and sometimes an administration interface. Recent box like the one from Free SAS embed a multitude of services from storage (VCR like) or console game. It means that these boxes must embed more and more processing power and storage capabilities for example. Thus, complexity becoming higher and higher, the probability for the Internet providers to deploy faulty boxes follow the same rule. We believe that complexity should be concentrated and easily accessible to the provider, while the network hardware, like boxes, spread all over the territory must stay simple and very reliable.

Hence, considering energy reduction on customer premises equipment, we can result in a significant contribution in minimizing the overall energy consumption of the industry and in lessening of carbon emission to the environment.

We reasonably suppose in this research work that most end users will be connected to their provider and have triple play services over a fiber link (FTTH : Fiber to the Home). This is something very common in large cities (in Europe) and even rural place close to those cities. It means that a fiber to Ethernet (or WiFi) converter will still be required. And then if we consider a replacement of the current Home Gateway (HG) by a quasi-passive device (assuming it is very simple (with no fancy features) and which consumes around 1Watt), and it is ideal to pull those resource intensive services towards the inner provider infrastructure

Hence, by relocating those network and application level services from HG into a virtual Home Gateway (vHGW) and if we used a node that can host around a 1000 vHGW's, we can achieve approximately 300% energy saving in the overall wire line telecom networks, and we can result in a minimum impact on the environment.

This paper is organized as follows. Section 2 and 3 respectively deal with experimental solution and results. Section 4 gives a state of the art summary of the area. Finally section 5 traditionally concludes this work and describes our future works.

2 Experimental evaluation of vHGW

2.1 Hardware architecture

Our experimental platform is composed by three Dell R610 computing nodes, respectively named AAA, BBB and CCC (see figure 2.1). All are equipped with quad-core dual processor (Intel Xeon E5506), run a recent GNU/Linux distribution (Debian 6.0 - Squeeze), and each consume power from 80W (idle) to 300W. Nodes are interconnected through 10 Gbps links (Myrinet). BBB use a two ports Myrinet card in order to be directly (back-to-back) connected to AAA and CCC. We apply a dedicated role to each physical node. AAA hosts data content servers (e.g video server). It is used to generate network traffic on demand. BBB hosts the virtual home gateway (vHGW) of the clients, thus we run as much LXC containers as there is clients. Finally, CCC is used to emulate clients who will consume the data generated by AAA. Obviously, the number of container in BBB equals number of containers in CCC. The data path goes through AAA to the containers in BBB to reach containers in CCC. In the rest of the paper we will focus on BBB.

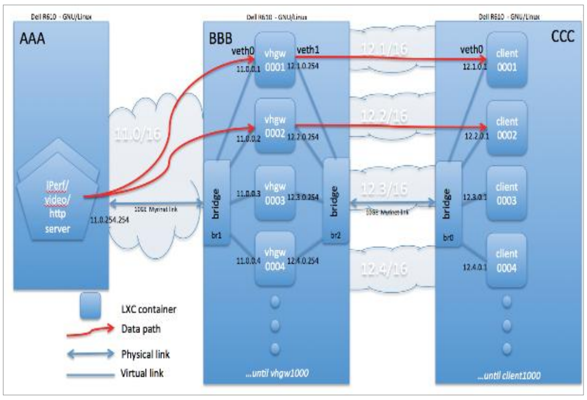

Figure 2.1: Experimental testbed for the experiments.

2.2 Software architecture

To achieve our goal to run a large set of virtualised boxes we had to choose an extremely lightweight virtualisation technology. When we want to run a very large number of virtual machines, the most common virtualisation solutions like XEN, KVM, VMware, ... based on full-virtualisation or even para-virtualisation show rapidly their limit or required extremely high end servers.

After a study of the common solutions available on the market and the context of our usage, we choose to keep only solutions offering simple isolation like vServer and LXC.

Due to the various advantages and simplicity of LXC, our experiment entirely uses this virtualisation solution. LXC [11] does not provide a virtual machine, but rather a virtual environment that has its own process and network space. It is similar to a *chroot* with more isolation. In brief, LXC uses *cgroups* [10] to create a restricted view of the host operating system. The benefits of LXC over KVM, VMWare or Xen, for example, is that it is light weight and provides the ability to host more virtual environments (without the need for CPU virtualisation support). Within the LXC guest environment, you can only see what the admin allows you to see of the host system; you can have a separate process space and also create a separate file system for the guest. The disadvantages include only support for Linux based guests. LXC is built into newer Linux kernels taking advantage of kernel support and does not required a heavily modified kernel (actually, *cgroups* is in the Linux kernel by default) and LXC tools are included in most distributions. It has virtually no overhead (we will see later that is not completely true) and provides a great flexibility because of its ability to share resources between different LXC guests.

3 Experimental results

In the context of this project, the more we host vHGW on one server the more we save energy. One of our first goals was to deploy as much as possible LXC containers. The first problem we have to take up was disk space. Because we chose to allocate a dedicated file system to each container, we had to replicate a file system as many times there are containers. Our file system was built thanks to the *debootstrap* tool (provided by Debian). While minimalistic, it is still too big (about 213 MB). After some raw cleanup we achieve 140 MB, which still provide a comfortable file system (i.e. with all the tools required in our experimental usage context). In an industrial context, with a well-defined list of services required for a final product, we predict that the file system can be reduced to less than 20MB. This can be done also by aggregating all the branches of the file system that can be mounted read-only and then shared by all containers (e.g. /lib directory). Finally, thanks to this file system size reduction we were able to replicate it as much as we wanted on the hard drive available on our server, namely 1000 times.

Our second goal was to launch all of those containers. It was more subtle to achieve this goal. Our very first experiment allows us to run "only" 124 containers. While it is already a good score, it was still far from our goal. LXC default configuration file open four *tty* (virtual terminal) that we do not use. By reducing this value to one, we noticed that by reducing the number of open file we were able to run more containers. Then, to go beyond we choose to tweak some default kernel configuration value, like the *inotify max_users_instances* (now set to 1024 to achieve our goal). Another problem we met was the extremely bad network performance. This was due to the limited size of the network neighbor table (caching MAC and IP address correspondence). Once we increased dramatically those table size (from 128 to 8192) the system did not spent it's time to flush the table anymore and worked again smoothly.

Finally, we were able to run, access and use 1000 LXC containers as vHGW hosted by only one physical server.

Immediately, we take advantage of this centralization to manage all those boxes. The deployment time is directly dependent of the file system size to replicate and hard drive technology used. On our SSD drive performances are really good but this disk does not provide enough space by now. This file system replication and customization (for each container) is made only once. Thus, this time duration is not critical.

Then, we wrote a set of tools to manage our deployed vHGW. Firstly, we had two tools to start and stop a set (or subset) of vHGW (lxc-starter.sh and lxc-stopper.sh) respectively. Then, we used another simple tool to check availability and good connectivity of all (or a subset) of the running vHGW (lxc-pinger.sh). Finally, we wrote another simple tool based on rsh (a kind of unsecured ssh) to run one (or more) command efficiently on each (or a subset) of vHGW. For example, an administrator can then apply very efficiently a routing service, NAPT service (Network Address & Port Translation), firewalling service, DHCP service, or software update on the fly without asking customer to reboot its box or, worst, to send it back by mail.

3.1 Network bandwidth sharing

Here, we propose to study the bandwidth network sharing in terms of fairness by the vHGW. We used Iperf to generate TCP streams as fast as possible, and UDP to generate stream at a given rate. The Iperf server runs on AAA, and Iperf clients run on each vHGW (one per container), i.e. on BBB, to study the behavior on one interface only (10GE Myrinet). The very first experiment shows that one TCP stream reach a maximal throughput of about 9Gbps, and two TCP stream share equally the bandwidth with 5Gbps for each.

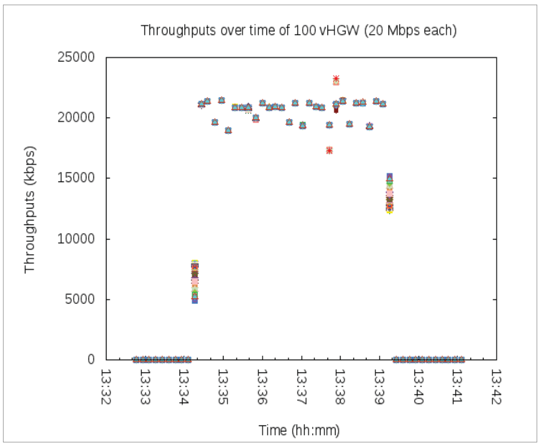

Figure 3.1: Bandwidth fairness among 100 vHGW transmitting data stream at 20Mbps.

The following experiment was used to study the bandwidth fairness between 100 vHGW. Each vHGW transmit one UDP stream at a given throughput to the Iperf server. In the 1st test we set the throughput at 20Mbps, while in the 2nd test we set the throughput to 100 Mbps. The total theoretical throughput is then 2Gbps and 10Gbps for the 1st and 2nd test respectively. We run those experiments for 5 minutes and obtain about 3000 points of measure (100 points each 10 seconds). As shown in figure 3.1, the fairness of the bandwidth utilization is good. Large majority of points superimposed. On the other hand, as shown in the figure 3.2, the "entropy" is high. No vHGW is able to maintain a constant throughput and loss rate is largely affected (not shown on the figures). In the 1st test, 2.7 % to 4.3 % of the packets is lost, whereas in the 2nd test, 79 % to 80% of the packets are lost.

Moreover, we noticed that in both experiment, all the cores of our CPUs were overloaded. The Htop tool shows that 100 processes are running and announce 100 % usage for the 8 cores..

Figure 3.3 shows the result of an experiment with 300 vHGW. Each vHGW is transmitting a 2 Mbps data stream. Total throughput of transmitted data is only 600 Mbps. There is nothing to say than it works finely.

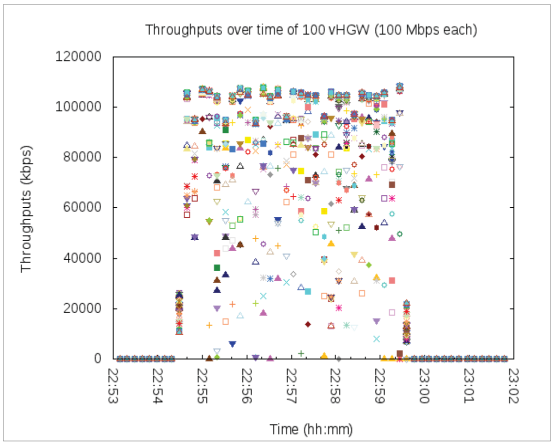

Figure 3.2 : Bandwidth fairness among 100 vHGW transmitting data stream at 100Mbps.

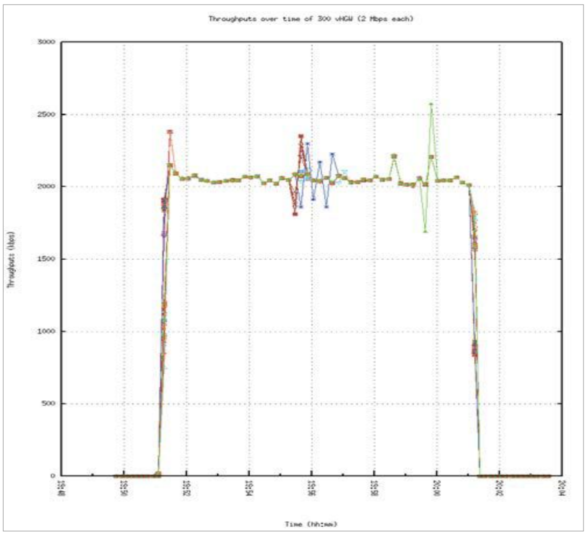

Figure 3.3: Bandwidth fairness among 300 vHGW, each transmitting a 2 Mbps data stream.

3.2 Energy cost of forwarding datastreams

In this experiment we measure power consumption of the node hosting one vHGW forwarding one datastream at different throughputs (respectively 1, 10,100 and 1000 Mbps) on a given period of time. As shown on figure 3.4 there is about 10 Watts only of difference between 1Mbps and 1Gbps. Consequently, for a large quantity of data to transmit we would recommend to send data as fast as possible to save energy.

3.3 Impact on context switch

Our host in a idle state shows about 80 context switch per second. In this experiment every single minute we start one more vHGW (It tooks about 16 hours to start 1000 vHGW). Figure 3.5 illustrates the results of this experiment. It shows that a large number of idle vHGW still impact the performance by increasing the number of context switch per second. When all the vHGW are launched, the number of context switch is about 500. The peak on the right is the consequence of the command launched by the administrator to halt all the vHGW. Then the system comes back to 80 context switch per second.

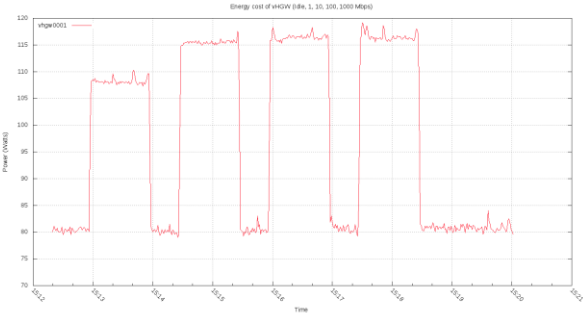

Figure 3.4 : Energy cost of one vHGW forwarding data streams at different speed.

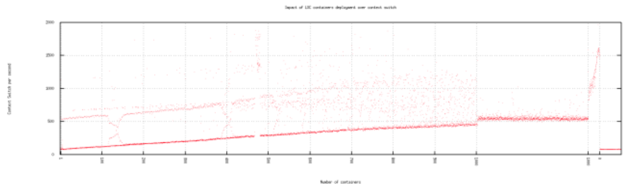

Figure 3.5 : Evolution of context switching while number of vHGW increase 1 to 1000, stay idle for a while, and finally are all halted at once.

3.4 Impact on Jitter and RTT

We also measure the impact over jitter and RTT (Round Time Trip) while number of vHGW hosted increase from 1 to 1000. To measure this we use respectively iPerf and ping. About jitter, measures show that this value remains almost constant (about 0.012 ms) whatever the number of vHGW. About latency, measures stay below 0.2 ms (with a std dev of about 0.022) under 100 running vHGW. At 500, value increase slightly (0.229 ms with a std dev of 0.42). And finally, at 1000 latency increase a lot with a high std dev value (2.33 ms with a std dev of 5.845).

3.5 Easy management

To illustrate the fact that it becomes really easy and efficient to manage all vHGW, we run the following experiment. About 500 vHGW are running. As shown on figure 3.6, at the beginning, 240 of them are forwarding a data stream of 10 Mbps each. Then, we (the operator) decide to filter 100 of them. Thus, we apply a iptable rule to drop the traffic from 100 of them (middle step on figure). Next, we apply the same filter rule on 100 more vHGW. Figure shows that the total throughput forwarded by the hosting node drops from 240 Mbps to 140 Mbps to 40 Mbps.

While we perform this scenario we also measure power consumption. Figure 3.7 shows that there is no gain to reduce number of streams. Conclusion, whatever is the number of stream (still not null) the power consumption turn around 162 Watts on this platform.

Figure 3.6: Applying filters on a subset of vHGW

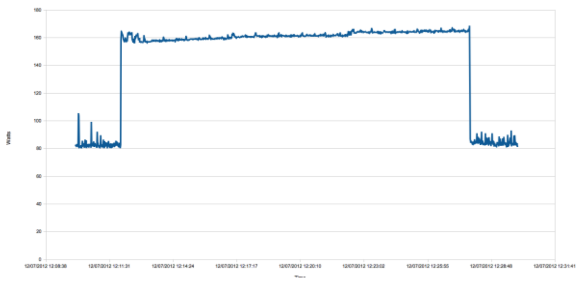

Figure 3.7: Reducing number of stream does not impact power.

3.6 Energy gains on idle box

We were wondering about the impact of the number of vHGW on power consumption when idle. For this experiment, we measured (one measure per second) the power (in Watts) consumed by the node (BBB). Our measuring device is directly connected behind the PSU (i.e. between the node and the wall socket). While measuring power, we deployed each 20 seconds one more vHGW. To reach the number of 1000 vHGW it spends about 6 hours. Figure 3.8 shows the result. Each point is one measure.

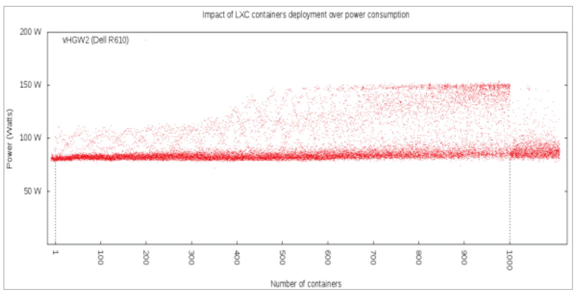

Figure 3.8: Impact on power consumption of 1 to 1000 deployed vHGW

We notice a high density line of points around 80 W. This is the value measured when our node is idle. The more vHGW are deployed, the more the density decrease. It means that the average value increase slightly, but less than 10 Watts. This additional consumption is probably due to the large number of context switching as shown by the Linux monitoring tools even when containers are unused. Thus, the number of vHGW deployed does not impact a lot on power, while processing performance will be clearly impacted.

If we consider idle box consuming 18W, replaced by simple box consuming only 6W and vHGW hosted on a server consuming 100W (considering all the vHGW idle), in this scenario we obtain a ratio gain of about 3 (*i.e.* we would consume three time less energy globally). % 1000 x 18W = 18.000 W compared to 1000 x 6W + 100W = 6.100 W

4 Related works

As the cost of energy is rising and protection of the environment from greenhouse effect is an important issue, energy reduction and its efficient utilization becomes a concern of every industry, and the whole world [2, 7, 8]. However, due to the advancement of Internet and broadband technologies, a huge and growing energy demand has been seen in the ICT and telecom industries.

According to current studies, the telecom infrastructure is the major contributor for the ever increasing energy demand in the ICT sector [2]. And surprisingly, more than 80% of this share is consumed by the Home Gateways (HGs) [9].

However, as access and aggregation networks become very powerful, and as customer interest rises for value added and content oriented services subscription, the home network configuration complexity and security becomes a concern [9]. Besides, as HG's service becoming complex in type and size of functionalities, obviously its energy consumption will rise dramatically, and its effect will be escalating energy demand in the telecom infrastructure.

Accordingly, one of the important contributions in reducing the energy consumption of HG's is a solution proposed in [3], an introduction of Network Protocol Agent (NPA). According to this study, the NPA will be always there to maintain network connectivity and monitoring any service interruption from interior and exterior environment, while power downing other functional blocks of the device during their idle time.

The Internet access box consumption: In 2007, the French magazine "60 millions de consommateurs"[1] did a comparative study about the ADSL box consumption which is largely deployed on the French territory. Besides the fact that those boxes are not always reliable, and this study shows their consumption is by far not negligible. According to the INC (French *Institut National de la Consommation*) a yearly cumulated average consumption in standby and working mode ranges from 143 and 263 kW/h depending on the model. Then, if we take into account the entire box in France, it required about 1.51 billion of kW/h power to supply all of them for a year (thus, the production of a nuclear reactor for 2.5 month). And surprisingly, another study also shows that putting in standby of HG elements doesn't have a significant impact on the overall energy consumption of HG devices as they consume more than 8W during their idle time even according to the European standard [4, 5].

Obviously, such devices cannot be completely switched off to insure a minimum set of services like telephony. We notice, without surprise, that the most consuming boxes are those embedding a hard drive. The hard drive is used mostly to record TV show. In the context of vHGW, this VCR functionality could be provided by the ISP also by storing users recording in a Cloud storage infrastructure for example. Sadly, a more recent study, from Alliance TICS, taking into consideration the apparition of the latest new box (e.g. Free revolution, 18W (ADSL box) + 13W (TV box)) shows that the French box consumption reach 5 TWh (including 3 TWh in standby mode).

However, the introduction of a vHGW [9] enables pulling of HG's services to NSPs premises for the purpose of reducing the network configuration complexity and security concern of HGs, and dictates a promising direction for the research community to consider vHGW's benefits from different perspectives.

Recently, another promising study also advocates the advantage of pulling some functionality of HGs to the access and/or backhaul network of the NSP, with a use of vHGWs [6]. According to this study, network service providers can be benefited in terms of reducing capital and operational expenditure and shorten time to market for new emerging services, while preserving subscriber's performance requirements.

Nevertheless, within the introduction of vHGW's, the potential of service relocation to address energy optimization (reduction) of HG's and also for the overall telecom infrastructure was not explored. Correspondingly, even though a number of researchers proposed various energy reductions and energy efficiency schemes for next generation wire-line networks, the potential of service relocation from HGs for energy saving remains untouched.

In this regard, the result of our preliminary work showed that the possibility of the relocation of some functionality of HGs, and the provision of those functionalities through a vHGW hosted in a node located in NSP premises. According to our study a significant energy saving has been achieved and the capability of vHGW's in executing the network and application level services, such as: routing, DHCP, firewalling and NAT, are confirmed.

Thus, by considering a replacement of the current HG by a quasi-passive device (which can consume around 1Watt) and if we suppose that end users have triple play services over a fiber link (FTTH), and pulling those network and application level services into a vHGW, and then having a server machine that can host around a 1000 vHGW's, we can obtain about 300% energy saving in the overall wire line telecom networks.

In conclusion, future box will still be always connected, however they will have to be cleverly designed to consume less (1-10W against 11-31W today), provide locally less services to ease a better reliability and improve maintenance efficiency.

5 Conclusion

In a near future, boxes (xDSL or FTTH) deployed in home customer will be reduced to the strict minimum to reduce energy consumption and failure probability, while services will remain and may become even more complex. In order to save energy while supporting users and internet service provider requirements, we propose to aggregate on dedicated servers a large number of virtualised box. Our approach still provides isolation between users while the provider takes advantage of this consolidation.

In this mostly experimental work, we prove that it was possible to host a large number of container to meet our requirements. The gains are of both terms: Energy and Management. To achieve this, a set of tools (framework) has been developed on purpose. While still experimental, it allows a rapid deployment and proved to be reliable.

We also show the borderlines of our approach when we reach the limits of the server hosting a large set of vHGW (~1000) or when the total throughputs forwarded by different vHGW reach maximum speed of our network infrastructure (~10Gbps). Additional experimental validations with various scenario of usage are currently under consideration.

[1] This magazine is edited by a French association for the defense of the consumers.

Acknowledgments

This research is partly supported by the GreenTouch initiative (http://greentouch.org).

6 References

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

[2] G. Koutitas and P. Demestichas, "A Review of Energy Efficiency in Telecommunication Networks", Telfor Journal, Vol. 2, No. 1, 2010.

[3] Wan-Ki Park, Chang-sic Choi, Il-woo Lee and Jonghyun Jang; "Energy efficient multi-function home gateway in always-on home environment", Consumer Electronics, IEEE Transactions, February 2010.

[4] Home Gateway Initiative, "Requirements for an energy efficient home gateway, HGI-RD009-R3", 2010.

[5] European Commission, Institute for the Energy Renewable Energy Unit, "Code of Conduct on Energy Consumption of Broadband Equipment", Version 3, 18 November 2008.

[6] Daniel Abgrall & Adam Kapovits, "How to build the next generation telecom infrastructure: Virtual CPE ", Eurescom study report, September 2011.

[7] Lange, C. Kosiankowski, D. Weidmann, R. Gladisch, A. "Energy Consumption of Telecommunication Networks and Related Improvement Options", Deutsche Telekom AG, Berlin, Germany, IEEE Journal, march-April 2011

[8] Michael C. Parker & Stuart D. Walker, "RoadMapping ICT: An Absolute Energy Efficiency Metric", Sch. of Comput. Sci. & Electron. Eng., Univ. of Essex, Colchester, UK , August 2011

[9] Koert Vlaeminck, "Service Enablers for Converged IP Access Networks" PhD Thesis, Ghent University, October 11, 2007

[10] cgroups,[Online] available: http://www.kernel.org/doc/Documentation/cgroups.

[11] "lxc Linux Containers: containers namespace cgroup virtualization", SourceForge, [Online] available: http://lxc.sourceforge.net/