# An Energy Aware Admission Control With Traffic Class Differentiation: From Theory to NS-2 Simulation

Hugo Zuger[1,2]
[1] CNRS, LAAS,
7 avenue du colonel Roche,
F-31400 Toulouse, France
hugo.zuger@laas.fr

Yann Labit[1,2]
[2] Univ de Toulouse,
UPS, LAAS,
F-31400 Toulouse, France
yann.labit@laas.fr

## ABSTRACT
Faced with increasing energy prices, financial crisis and environmental problems, the sector of information and communications technologies initiates the third industrial revolution. This revolution aims to intelligently allocate and distribute energy. The study, development and implementation of intelligent energy managers for computer networks is called green networking. The proposed paper aims to develop and test an energy-aware solution. EAAC-MQ, an Energy-Aware Admission Control for wired networks, has been created and tested with NS-2 (Network Simulator). This solution adapts its behavior to traffic classes (Multiple Queues, MQ) to meet the divers needs in network ressources. It reduces energy consumption, and therefore operating costs, about 20%, despite of reducing throughput and increasing delay.

## Keywords
Green Networking, Energy-Aware Networks, Admission Control, Energy consumption, Routers, Network Simulator NS-2, Traffic Class Differentiation.

## 1. INTRODUCTION
The Internet requires a large amount of energy to provide its services to its 2.4 billion users[1]. Servers, routers and switches consume almost 5.5% of the annual world energy production [11]. Moreover, this consumption increases of 25% each year. Consequently, the Internet is an important contributor to greenhouse gas emissions. According to [16], the Internet emits as much gas as civil aviation (nearly 2%). Without being alarmist, this quick carbon footprint shows the Internet as an energy-greedy system. Associated with the recent economic issues encounterred by companies, governments and private indiviudals, which lead to reduce costs, it is easy to understand that reducing ICT power consumption is became a central focus for the industry and even IEEE [13]. Economic and ecological contexts, not forgetting Moore's law, led to the creation of a new area of works and research named *green networking*. Green networking is the development of energy-efficient networking technologies and products, in order to minimize resource and energy use whenever possible, and indirectly cost.

[1]http://www.internetworldstats.com/stats.htm

Many works have already been done on this subject. Our state of the art focus only on wired network solutions. Some surveys such as [5] and [3] suggest to classify these green networking solutions into different categories. Five major types of solutions are underlined :

- Hardware and infrastructure engineering. These solutions aims to use energy-efficient materials to make network devices or to reduce network complexity [8], [7].
- Dynamic adaptation. Devices adapt themselves to the traffic load (idle logic, dynamic voltage scaling, adaptive link rate [9]).
- Smart sleeping (interface proxying). The goal is here to use a proxy to ensure network presence of a computer while it is in sleep mode [10].
- Router buffer management : optimize the number of buffers to use in order to reduce power consumption [4], [19].
- Energy-aware routing protocol. Gather the traffic in few links in order to reduce router consumption [2], [21].

However, we choose to focus on an original solution that has been introduced by Erol Gelenbe et al. in [18]. They created an admission control for wired networks which takes into account the estimated power consumption. This approach already existed in wireless network and is now adapted to wired networks. As far as we knew, this is the only work done about an energy-aware admission control for wired network. An admission control is a network mechanism verifying the availability of ressources such as bandwidth in order to decide wether or not a data can be safely transmitted. Here, the admission control estimates the power consumption of the network in order to determine if the consumption peak that a new flow would generate is acceptable or not. The decision is taken by comparing this peak and a threshold : if the peak is greater than the threshold, the flow will wait. Else, it is send. Waiting flows are re-processed later. This admission control can save 17% of power consumption. As some flows have to wait, the average delay increases (almost factor 8 in their tests on a test-bed [18]). Yet, none informations are given about throughput : is it the same ? Has it evolved ? If it does, how ? We can guess that with 90% of the Internet traffic being transmitted with TCP[2], delaying packets will decrease throughtput, as acknowledgements will also be delayed. Anyway, we do not know if the decreasing power consumption is (in percent) the same as a potential decreasing in throughput, or not.

In our opinion, this is not the only limit of this solution. It is centralised as there is only one machine that supervised the admission control for all the network. Thus, a high-frequency information exchange about link utilization is needed, leading to an increasing traffic, that has to been taken into account to evaluate the new performances. Moreover, this traffic has to have right of way in order to give to the supervisor the most accurate data about link utilizations. But nothing is said about the consequences of such a traffic on performances.

One of the main limits of Gelenbe paper is that there is no differentiation between types of traffic. All packets are keep in the same buffer, no matter if they belong to a VoIP communication, a P2P or FTP transfer or are part of a mail. In fact, such a choice can generate an even longer delay in some case. For example, in the case where many low-priority packets arrive in the buffer just before a single VoIP packet, this one would have to wait either the low-priority burst is taken care of or its timer ends. Moreover, as both types of packets have the same timer Tmax, the VoIP packet can not overtake the low-priority packets in some scenarii. As the delay increases a lot, something has to be done there.

In order to improve Gelenbe et al. model, we suggest a new model of energy-aware admission control for wired network that takes into account the diversity of Internet traffic. In this paper, we propose a model with different traffic classes, in which each one has a different Tmax and a different energy threshold.

First, we will detailled some notions about networking and energy consumption, which are essential to understand our solution. Secondly, each step of our solution will be described. The results of our simulations will be presented in the fourth section. We will then synthetise the results and conclude with several tracks.

# 2. NOTIONS

In this section we present some notions about networks that are necessary in order to understand our solution and its mechanisms. Thus, we focus on router power consumption and routing before explain our solution in details.

## 2.1 Router power consumption

Some works have been done about new models for hardware consumption, specially for routers. Different models exist and we are going to focus on two of them. First, *On/Off* model [17] is the power consumption model used in most of routers nowadays. The consumption is approximately the same regardless of traffic density as data rates are always the same. The second one is *Adaptive Link Rate* [9]. In this model, Ethernet links dynamically change their data rates in response to traffic levels. Thus, router power consumption is proportionnal to the traffic. The more important traffic is, the bigger the consumption is. This new model can reduce energy consumption (estimation : hundreds of millions of dollars in the USA alone each year) without degrading user performance. However, a negociation (via handshake mechanism) between the source and the destination is required : consequently, ALR can only be used if these two machines (and all possible intermediate machines) have this model implemented. The consumption is here the sum of a static part (chassis power consumption) and a dynamic part (proportionnal to traffic). Figure 1 shows both models on an example link (values are arbitrary). With ALR, maximum consumption is only reached when link utilization is maximum. As most of the time networks are overdesigned, this situation barely occurs.

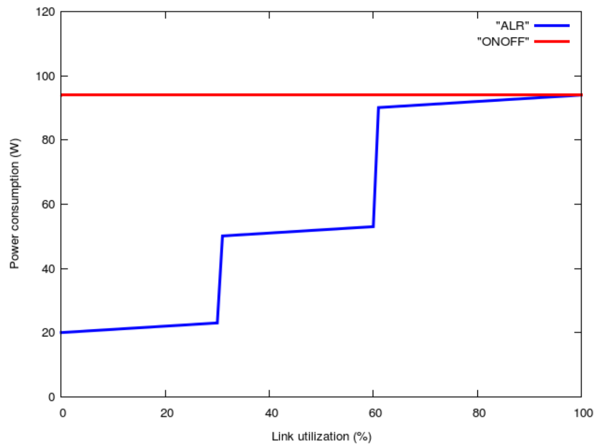


**Figure 1: A comparison of two power models**

## 2.2 Routing

Route computation aims to find the path between a machine A and a machine B. Among the different routing algorithms existing, we choose to use *Open Shortest Path First* (OSPF) [3]. It is a version of the shortest path problem applied to networks. A topology, which is a set of nodes connected together, can be seen as a graph. Thus, Dijkstra algorithm can be applied to determine the shortest path between two nodes. Dijkstra algorithm applied to networks can minimize either delay or number of hops.

Some green networking research [2] [21] focus on energy-aware routing protocols. These solutions aim to reduce consumption by turning “off” some links (a link is said to be turned off if there is no traffic on it). The traffic is gathered on few links in order to minimize the consumption of the network.

*Remark*: A possible future work would be using such a routing protocol in our solution, in order to improve its results.

[2]Transmission Control Protocol (RFC793), http://tools.ietf.org/html/rfc793

[3]OSPF Version 2, http://tools.ietf.org/html/rfc2328

At this step, we want to estimate the efficiency of our approach without energy-aware routing protocol.

### 2.3 Algorithm of Energy-Aware Admission Control - Multiple Queues

We named our solution Energy-Aware Admission Control - Multiple Queues, shorten EAAC-MQ. The goal of this admission control is to find the optimal period to send the packet. The optimal period is defined as follows : it is the period during which the consumption of the packet is the smallest and the delay added by the admission control acceptable according to the QoS required. The way the admission control proceeds is still by estimating the network energy consumption if the packets are send. According to the value of this estimation, the admission control decides whether or not the packets have to be send.

Here is its algorithm. Note that classifying and processing are parallel threads.

1. Thread 1 : Classify the incoming packet
2. Thread 2 : Packet processing
   (a) Choose the packet to process
   (b) Compute its route
   (c) Estimate the power consumption on this route
   (d) Take a decision : keep or send this packet ?
   (e) Process next packet

As things are, some settings have to be specified as our solution is not adapted to any topology. Here is a list of hypothesis and conditions that the topology which benefit from EAAC-MQ have to meet : EAAC-MQ is an energy-aware admission control for wired *Local Area Network* (LAN), with only one domain. Moreover, we suppose the topology known and well-sized (congestion is impossible, unless a link is down). The power consumption model of all network devices has to be *Adaptive Link Rate* (ALR) or equivalent.

## 3. EAAC-MQ MODEL

### 3.1 Algorithm

The principal of EAAC-MQ is to delay energy-greedy packets and thus favour packets which would consume less energy. This admission control applies a differentiated service, depending on traffic class the packet belongs to. The needs of this solution lead to this algorithm :

1. Classify incoming packets : the Classifier
2. Read one of the buffers : the Reader
3. Packet processing : the Admission Control

Each step is explained and detailled herebelow. Contrary to the centralised solution of Gelenbe, our is distributed : each client has its own admission control. It may be problematic with congestion, but, as we explained herebelow, traffic engineering is required to optimize EAAC-MQ results and we expect administrator to well size their networks.

### 3.2 Step 1 : The Classifier

#### *3.2.1 Trafic classes*

As previously explained, our admission control differentiates traffic classes, according to the QoS (Quality of Services) packets require. The different traffic classes can be seen as DiffServ classes. Each class is defined by a traffic type and two parameters, already used by Gelenbe [18] in his model. These two parameters are used in the admission control.

The first parameter is *Tmax*. This timer is used to limit the additionnal delay caused by the admission control. After *Tmax* seconds, the packet will be admitted on the network regardless of its estimated energy consumption. *Tmax* can be seen as the maximum amount of time the packet is willing to wait before being accepted on the network.

The second one is the *energy threshold*. It is the limit threshold beyond which the packet is not accepted on the network. In fact, if estimated consumption is greater than this threshold, the packet is either put in the WQ queue or stay in it. If estimated consumption is smaller than this threshold, the packet is admitted on the network. It is further explained in Step 3.

Each class has a different *Tmax* and *energy threshold* in order to differentiate the processing between classes :

- the biggest Tmax is, the lower the priority is. We consider that a low-priority packet is more disposed to wait than a high-priority one. Therefore priority packets should have the smallest Tmax if one wants to have the best results.
- the biggest the threshold is, the higher the priority is : the admission control is more tolerable with high-priority classes. Indeed, the probability to reject a packet decreases when the treshold increases. Thus, a high-priority packet is accepted more often than a low-priority one for a given utilization of the network. It is thus advised to respect this threshold hierarchy to get optimal results.

Consequently, it is important to choose wisely the *energy threshold* and the *Tmax* for each class. Having a too small *threshold* makes the admission control too selective as too many packets could be rejected and then be forced to wait the end of their *Tmax*. However, having a too big *threshold* increases energy consumption by accepting packets too easily. In this case, the admission control tends to be useless to reduce energy consumption.

These choices must be based on a previous analysis of the topology on which this admission control will be implemented. Number of routers and links, type of traffic, daily and weekly network utilization have to be taken into account to determine the optimal parameters according to the period. Traffic engineering is needed to have the best results on a given topology. A good choice of parameters is characteristic of the topology. We do not offer here a global method that gives the parameter values for a given topology. Such a method can be the subject of a future work. Anyway, for our simulations, we have analysed our test topologies and traffics to determine the best Tmax and threshold to use in these cases.

### 3.2.2 *Classify incoming packets*

As soon as a incoming packet is classified, it is put into its class buffer, called Request Queue. There are two types of queues in EAAC-MQ.

The first one is Request Queue. It is the queue that contains the incoming packets from a certain traffic class. Each traffic class has an associated RQ. RQs are FIFO queues.

The second one is Waiting Queue, the queue containing packets that had been rejected at least one time. Service policy is *Shortest Remaining Time* (SRT). Packets are here classified by decreasing remaining time before their Tmax expires. Thus packets having the smallest remaining time to wait are first processed. Such a policy has been chosen to decrease the average delay. An other possible policy would have been to process high-priority packets before other packets. However, we estimated that the energy economies would not have been as important as with SRT, because too many low-priority packets would have waited until Tmax. Moreover, as high-priority packets have a smaller Tmax than low-priority have, SRT policy puts most of the time high-priority packets at the head of the WQ. Note that the Waiting Queue is the same for any packet, regardless of its class. For example, with 2 traffic classes, there are 2 Request Queues and 1 Waiting Queue (see Figure 2 with n=2).

## 3.3 Step 2 : The Reader

The way buffers are read is different from Gelenbe solution. In its model, WQ is just read when RQ is empty. But this way increases the number of packets that have to wait till the end of their *Tmax*. We therefore chose an other method.

The scheduling discipline applied to RQs and WQ is *Weighted Round Robin*. WRR serves a given number of packets for each nonempty queue. This scheduling discipline is chosen in order to increase the differentiation between processings. Indeed, with such a discipline, high-priority classes can be traited more often than low-priority classes if one configures weights correctly, by giving the biggest one to high-priority classes.

Weights should be modify according to the period of day / week. These weights also have to be dynamically adapted, such as *Tmax* and *energy threshold*, reinforcing the need in traffic engineering paired with EAAC-MQ.

## 3.4 Step 3 : Packet processing by admission control

We can model the problem as follows :

$$\begin{cases} s_i = node\, with\, EAAC - MQ \\ d_i = destination\, node \\ R_i = route\, from\, s_i\, to\, d_i \\ bw_k = amount\, of\, data\, at\, the\, k^{th}\, router\, of\, R_i \\ \epsilon_i = size\, of\, incoming\, packet \\ C_i = traffic\, class\, of\, incoming\, packet \\ \alpha_i = power\, threshold\, associated\, to\, C_i \\ Tmax_i = Tmax\, associated\, to\, C_i \\ P_k(bw_k) = power\, consumption\, of\, the\, k^{th}\, router\, of\, R_i \end{cases}$$

The beginning of the admission control is classic : the packet header is read to determine the destination $d_i$ and this information is used to compute the route $R_i$ with the chosen algorithm. We choose OSPF but it is better to compute routes with an energy-aware routing protocol, as it decreases energy consumption. The implementation of such a protocol in EAAC-MQ is a possible future work.

Next, the used bandwidth on each link from this route is estimated. Here is the first difference : the estimate bandwidth is used to approximate the actual consumption $P_k(bw_k)$ of each router according to ALR model [9], the consumption model that we described earlier.

The packet size $\epsilon_i$ and $P_k(bw_k)$ are used to approximate the future consumption of $R_i$ :

$$\sum_{k \in R_i} [P_k(bw_k + \epsilon_i)]$$

The decision is taken by comparing the peak of consumption $\delta$ that this packet should create to the *threshold* $\alpha_i$. $\delta$ is defined as :

$$\delta = \sum_{k \in R_i} [P_k(bw_k + \epsilon_i) - P_k(bw_k)]$$

If $\delta \geq \alpha_i$, the packet joins the Waiting Queue, where it will wait either $Tmax_i$ or its next processing.

If $\delta < \alpha_i$, the packet is accepted on the network and is immediately send.

Figure 2 sums up EAAC-MQ algorithm in a general case with $n$ different traffic classes. For each $k \in [1, n]$, $C_k$ is a traffic class with its threshold $\alpha_k$, its timer $Tmax_k$ and a weight $N_k$ for the WRR buffer reading. The bigger $k$ is, the lower the priority is. We choose $N_k \geq N_{k+1}$ and $N_{WQ} < \min_{k \in [1,n]}(N_k)$ to favour the processing of high-priority packets. $Tmax_{WQ}$ is in fact a dynamic $Tmax$ : as it is a common buffer, each packet can have a different $Tmax$ depending on its traffic class.

As consequences of this model, we expected to reduce energy consumption but also to increase delays, as we keep some packets in buffers. Consequently, throughput should decrease if TCP is used. Indeed, to increase the delay may cause TCP to reduce the size of its window in order to avoid or reduce potential congestion.

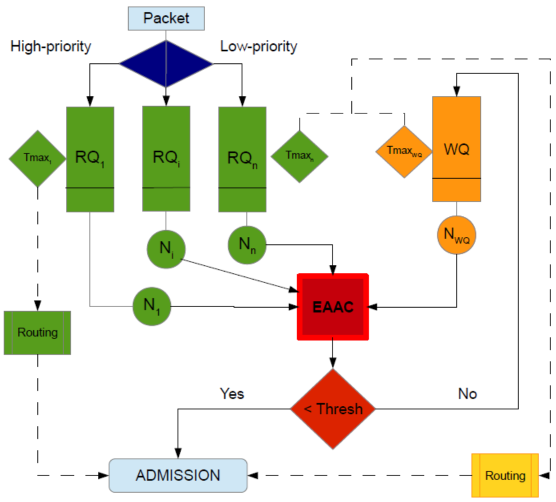


**Figure 2: EAAC-MQ algorithm in a nutshell**

# 4. SCENARII & SIMULATION RESULTS

## 4.1 Tools

Despite we knew that emulation gives more interesting results than simulation does [15], we were restrained by the router model used in the LAAS network emulator (LaasNetExp [15]). Indeed, they do not have an ALR consumption model but a "traditionnal" one. A intern study [1] shows that these routers have a constant consumption (model ON/OFF) regardless of the utilization of links connected to them. Thus we decided to rely only on simulation, that we did with the well-known event-simulator ns-2 [4] (version 2.35). This choice led us to find the appropriate ns tools to fulfil our objectives. An energy measurement tool and a realistic web traffic generator were necessary.

The official release of ns-2 has only a energy model for wireless nodes. As our solution aims to be apply to wired network, we were forced to implemente a different energy model. Our focus was on ECOFEN [14], a ns-2 tool provided a ALR energy model. This module periodically returns the energy consumption of each node. This module is used to estimate consumption for the admission control and get the global consumption. Consumption values for each entities is printed each second.

As a traffic generator, we chose PackMime-HTTP [6], a ns-2 http-1.1 traffic generator which had been developped by Bell Labs. We have chosen this generator among plenty of others because it is well documented in ns doc, it does not need an external trace file to generate traffic (contrary to Tmix[20]), it is compatible with ECOFEN (the energy model) and other ns traffic agents (CBR, FTP). Moreover, PackMime-HTTP generates output files from whom delays and throughput can be easily extracted. The most important parameter is *rate* : the bigger it is, the more important the number of new http connexions per second is. These modules are used with our own modul, EAAC-MQ, which had been implemented in the core of ns-2 in order to perform the simulations the results of which are presented in the following subsections.

[4]ns-2 official website : http://www.isi.edu/nsnam/ns/

## 4.2 Simulation topologies

Two different topologies are used in the simulations presented here below. The first one, on figure 3, is the simpliest topology : two nodes connected by a single full-duplex link (communications can be achieved in both directions). This topology will be referred as *Topology 1* thereafter.

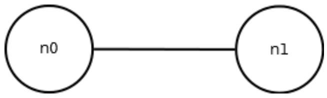


**Figure 3: Topology 1**

The second topology (referred as *Topology 2*) we used in our simulations is presented on figure 4.2 (routers are white, clients green and servers blue). This topology is the closest of the one Gelenbe used in his paper we could make with the informations found in the paper [18] (they specified nor link capacity neither link delay). There are 4 PCs (clients), 2 servers (one intern server and one extern) and 13 routers. All links have a 20 Mbps capacity and a 5 ms delay, except the link between *Server_2* and *Router_5* which has a 50 ms delay in order to characterize the distance between clients and *Server_2*, located somewhere on Earth. Link weights, used in OSPF algorithm, are chosen to make *Server_1* unreachable except for traffic heading to it. In fact, only traffic address to *Server_1* can reach it *(Server_1* is not used as a router). Moreover, we suppose clients, routers and servers have the same consumption model (dynamic part) but different static part, according to figures found in the state of the art (PC : 100 W, routers : 150 W, servers : 250 W). These choices have been made to make the model more realistic. PC consumption value is the average max consumption of 2-core computers, based on manufacturer datasheets from the website materiel.net[5], a french hardware distributor. Router value is taken from ECOFEN paper [14] and server value is based on an Intel Corporation report [12].

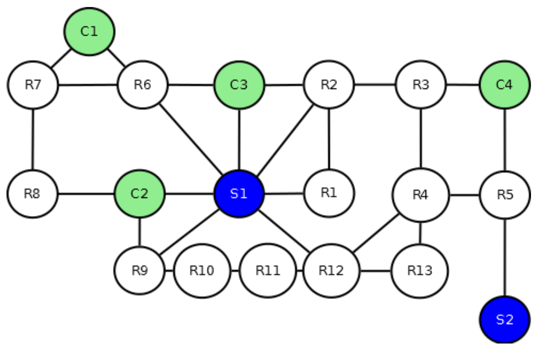


**Figure 4: Topology 2**

## 4.3 Traffic

Two different traffics are used in the simulations presented in this article : a TCP-based traffic (Pareto and Exponential distributions) in the first simulation and a HTTP traffic generated with Packmime-HTTP in the second one. The shape of the HTTP traffic is based on analysis made on the LAAS network with MRTG[6] and represented on figure 5. We can

[5]http://www.materiel.net/
[6]MRTG project website, http://oss.oetiker.ch/mrtg/

see traffic peaks during working hours (9.00 to 12.00 and 14.00 to 19.00) and very few traffic at night. In simulations using TCP-based traffics, the configurations are arbitrary.

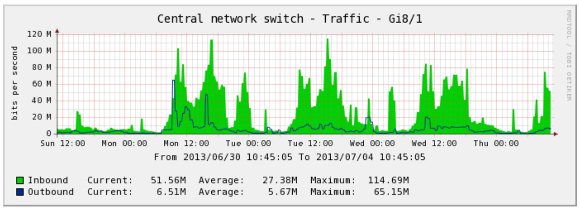


**Figure 5: Traffic on central switch at LAAS-CNRS**

## 4.4 Results and performances

The Simulation 1 takes place on Topology 1 on which two traffic classes (2C) are used : one with high priority (Pareto distribution) and one with low priority (Exponential distribution). These choices are arbitrary. The results are showed on figure 6 (energy consumption), figure 7 and figure 8 (throughputs).

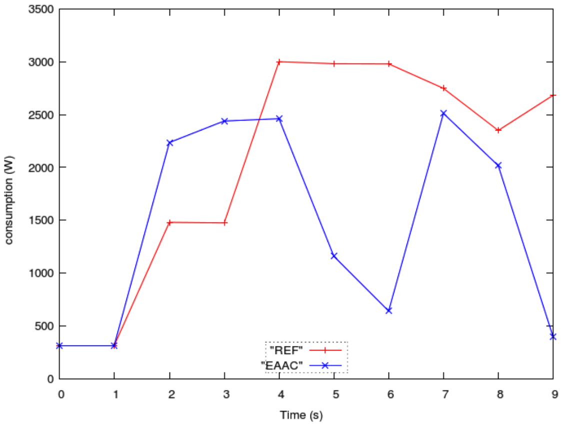


**Figure 6: Consumption in Simulation 1**

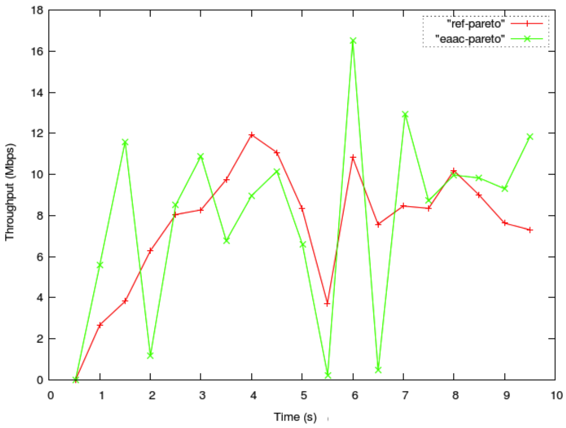


**Figure 7: High-priority traffic throughput in Simulation 1**

In this series, we reduce the energy consumption of 29% (consumption with our solution is in blue) and it results

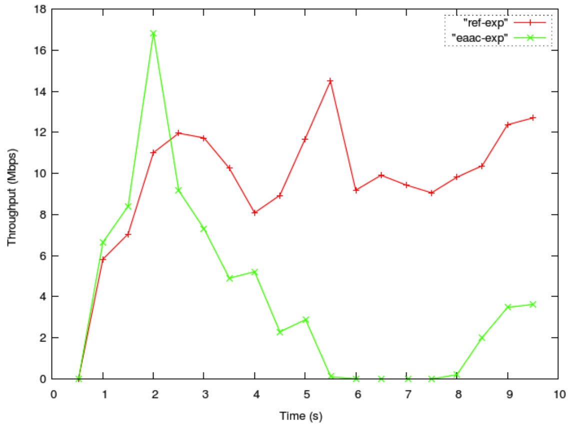


**Figure 8: Low-priority traffic throughput in Simulation 1**

in throughput losses of 32% (results with our solution in green). In this case, we can see that the low-priority traffic is more impact that the high-priority one. In fact, the admission control seems to sacrifice low-priority traffic to reduce consumption. Moreover, a focus on delays shows that the increase for low-priority (x4) is more important than for high-priority (x1.4). The differentiate processing is efficient, even if the impact on low-priority classes seems too important.

The Simulation 2 takes place on Topology 2 with only one traffic class this time, in order to analyze the performance in a simple case but on a more complex topology though. The HTTP traffic describe earlier is used in this simulation. Results are showed on figure 9 (energy consumption) and figure 10 (throughput). Green graphs represent performances with our solution implemented. Red ones represent the reference case (without the energy aware admission control).

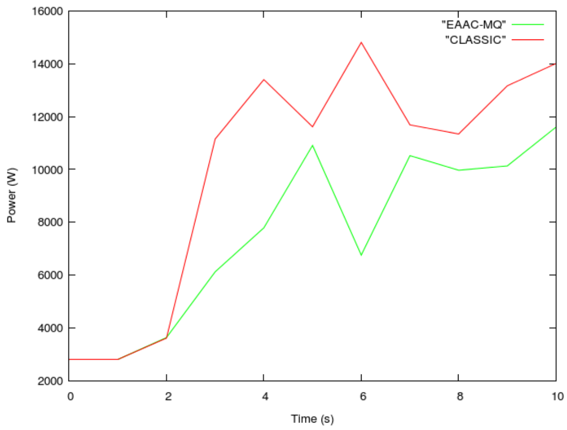


**Figure 9: Consumption in Simulation 2**

Again, our solution reduces energy consumption (by 27%). Throughput decreases by only 5% this time thanks to an optimal configuration of the admission control (threshold and $Tmax$). The delay is approximately doubled with our solution (x1.8).

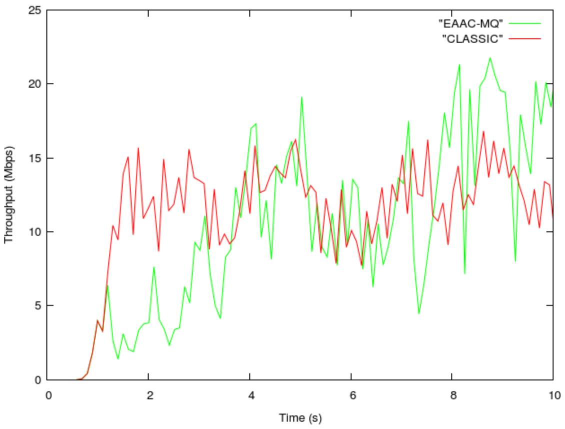


**Figure 10: Throughput in Simulation 2**

The table 1 sums up the results of the simulations we presented earlier. It gives the economy of energy obtained with our solution, the variation of throughput and delay compare to a reference simulation (without our solution).

| **Topology** | **Topology 1** | **Topology 2** |
|---|---|---|
| **Simulation** | **2C** | **1C** |
| **Consumption reduction** | -29% | -27% |
| **Throughput** | -32% | -5% |
| **Delay** | x 2 | x 1.8 |

**Table 1: Results of performance evaluation.**

It is important to note that informations related to ALR handshakes, routing matrices and energy consumption estimations do not spread on links in simulations, contrary to what really happens on networks. This difference has to be taken into account as these transmissions will modify the link utilizations and thus consumption estimations.

All in all, these simulations shows good results : the solution we developped reduces energy consumption at the cost of increasing delays and reducing throughput. But these consequences are the price to pay to reduce energy consumption (and cost). Future works aiming to improve our solution can be done : using an energy-aware routing protocol instead of OSPF will increase energy savings, an implementation on testbeds will expose our solution to real traffic. To base decisions on flows and not packets or to dynamically adapt threshold and Tmax configurations to moment of day/week should also improve results and performances. A preliminary traffic engineering approach should be included to help in the choice of EAAC-MQ model.
*Remark*: More experiences and improvment in ns-2 coding (ex: the minimum of throughput) are needed to be more efficient and complete.

## 5. CONCLUSION

We presented in this paper the model of an energy-aware admission control called EAAC-MQ that adapts its behavior to traffic classes to meet the divers needs in network ressources. It has been implemented with ns-2, the common network simulator. Thus, 25% of energy can be saved in average while reducing the impact on delays and throughput for high-priority traffics.

EAAC-MQ is one of the numerous pieces of the puzzle that energy-aware network is. Indeed, in respect with all the works done so far in this field, it is clear that future networks will need modifications at different levels in order to become green. Admission control is just a piece. We believe that it has to be paired with traffic engineering methods but also with energy-aware routing protocol to be optimal. Each approach are more or less efficient but we are expecting that best results will come from the combination of all of them. Obviously, this pooling needs compatibilty studies in order to analyze the influence that a solution would have on the others, to maximize both energy savings and user quality of experience.

## 6. ACKNOWLEDGEMENTS

We would like to thank the project FLOWER (CNRS PEPS - INSIS 2011-2013)[7] to which this work is linked.

## 7. REFERENCES

[1] A. Adelin, P. Owezarski, and T. Gayraud. On the impact of monitoring router energy consumption for greening the internet. In *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, pages 298–304, 2010.

[2] E. Amaldi, A. Capone, and L. Gianoli. Energy-aware {IP} traffic engineering with shortest path routing. *Computer Networks*, 57(6):1503 – 1517, 2013.

[3] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier. A survey of green networking research. *Communications Surveys Tutorials, IEEE*, 14(1):3–20, 2012.

[4] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli. Green networking with packet processing engines: Modeling and optimization, 2013.

[5] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti. Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *Communications Surveys Tutorials, IEEE*, 13(2):223–244, 2011.

[6] J. Cao, W. Cleveland, Y. Gao, K. Jeffay, F. Smith, and M. Weigle. Stochastic models for generating synthetic http source traffic. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1546–1557 vol.3, 2004.

[7] L. Ceuppens, A. Sardella, and D. Kharitonov. Power saving strategies and technologies in network equipment opportunities and challenges, risk and rewards. In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 381–384, 2008.

[8] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 457–465, 2008.

[9] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen. Reducing the energy consumption of ethernet

[7] Project FLOWER website, http://homepages.laas.fr/ylabit/page_flower.html

with adaptive link rate (alr). *Computers, IEEE Transactions on*, 57(4):448–461, 2008.

[10] M. Jimeno, K. Christensen, and B. Nordman. A network connection proxy to enable hosts to sleep and save energy. In *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, pages 101–110, 2008.

[11] J. Koomey. Estimating total power consumption by servers in the us and the world. 2007.

[12] L. Minas and B. Ellison. The problem of power consumption in servers. In *Energy Efficiency for Information Technology*, 2009.

[13] I. of Electrical and E. E. (IEEE). Ieee 802.3az energy efficient ethernet task force.

[14] A.-C. Orgerie, L. Lefevre, I. Guerin-Lassous, and D. Pacheco. Ecofen: An end-to-end energy cost model and simulator for evaluating power consumption in large-scale networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1–6, 2011.

[15] P. Owezarski, P. Berthou, Y. Labit, and D. Gauchard. Laasnetexp: a generic polymorphic platform for network emulation and experiments. volume 24, pages 24:1–24:9, 2008.

[16] G. A. Plan. An inefficient truth. December 2007.

[17] J. Restrepo, C. Gruber, and C. Machuca. Energy profile aware routing. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pages 1–5, 2009.

[18] G. Sakellari, C. Morfopoulou, T. Mahmoodi, and E. Gelenbe. Using energy criteria to admit flows in a wired network. In E. Gelenbe and R. Lent, editors, *Computer and Information Sciences III*, pages 63–72. Springer London, 2013.

[19] A. Vishwanath, V. Sivaraman, Z. Zhao, C. Russell, and M. Thottan. Adapting router buffers for energy efficiency. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 19:1–19:12, New York, NY, USA, 2011. ACM.

[20] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, and F. D. Smith. Tmix: a tool for generating realistic tcp application workloads in ns-2. *ACM SIGCOMM Computer Communication Review*, 36(3):65–76, 2006.

[21] M. Zhang, C. Yi, B. Liu, and B. Zhang. Greente: Power-aware traffic engineering. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pages 21–30, 2010.