

On the Energy Efficiency of Sleeping and Rate Adaptation for Network Devices

Timothée Haudebourg¹ and Anne-Cécile Orgerie²

¹ ENS Rennes - University of Rennes 1, Rennes, France

`timothee.haudebourg@ens-rennes.fr`

² CNRS - IRISA, Rennes, France

`anne-cecile.orgerie@irisa.fr`

Abstract. The ever-growing appetite of Internet applications for network resources has led to an unprecedented electricity bill for these telecommunication infrastructures. Several techniques have been developed to improve the energy consumption of network devices. As their utilization highly varies over time, the two main techniques for saving energy, namely sleeping and rate adaptation, exploits the lower workload periods to either put to sleep some hardware elements or adapt the network rate to the actual traffic level. In this paper, we compare two emblematic approaches of these energy-efficient techniques: Low Power Idle and Adaptive Link Rate. Our simulation-based study quantifies the reachable energy savings of these two approaches depending on the traffic characteristics. We show that, with little impact on the Quality of Service and consequent energy savings, Low Power Idle has a clear advantage. On the contrary, ALR is almost always consuming more than LPI and can reach unacceptable QoS levels. We also show that they can be combined to achieve better energy-efficiency, but at the cost of important QoS degradation.

Keywords: Energy efficiency; Wired networks; Sleeping; Adaptive Link Rate; Low Power Idle

1 Introduction

Information and Communications Technologies (ICT) currently consume around 5% of global electricity with one third of this consumption that is imputed to communication networks [19]. The multiplication of end-user devices leads to a rapid growth of the traffic. A recent study made by Cisco estimates that annual global IP traffic will pass the zettabyte threshold by the end of 2016 [13]. This ever-growing appetite of Internet applications for network resources has led to an unprecedented electricity bill for these telecommunication infrastructures.

Yet, it has been shown that network infrastructures are not used at their full capacity and present high redundancy for fault-tolerance and security purposes [11, 22, 25]. Moreover, network devices such as routers and switches are power-hungry even when they are little or not used [23, 16, 7]. The energy consumption of networks is not only incurred by powering networking equipment

(routers, switches, links, hubs, etc.), but also by end-hosts that demand high availability and full-time connectivity even if the network is not used [18].

The ideal power-proportionality has still not been reached by device manufacturers [4]. These observations have led to the proposition of various solutions to save energy in wired networks. Approaches found in literature can be categorized into two categories, both exploiting the lower charge periods to either put to sleep some hardware elements (sleeping) or adapt the network rate to the actual traffic level (rate adaptation).

The emblematic sleeping solution proposes a standardized Low Power Idle (LPI) mode [12] (norm IEEE 802.3az). The basic idea of this Energy-Efficient Ethernet (EEE) standard consists in sending packets as fast as possible and entering a low-power idle state when there is no data to transmit. The first network devices implementing this capability have appeared on mass market in 2013. Packet coalescing can be used to improve LPI performances at the cost of a slight latency increase [14, 9].

As for rate adaptation, the most famous implementation is Adaptive Link Rate (ALR) which has been proposed in 2005 [16, 18, 17]. It follows the idea of the Dynamic Voltage Frequency Scaling (DVFS) for CPUs adapted to the network device port rates. When full speed is not needed, a lower rate is negotiated between the network ports sharing a common link, thus incurring less power consumption [5]. Two buffer thresholds are employed to decide when to switch to a lower or a higher rate. Several policies have been proposed to adjust these buffer threshold with the aim of reducing oscillations between rates [5, 17].

While these two techniques pursue a common goal, they adopt radically different approaches. Moreover, while LPI is standardized and deployed for several years, ALR is still looking for a viable implementation resolving its oscillation issues. One can wonder which approach can provide the larger energy savings, at what cost for the Quality-of-Service, and whether they can be combined for a better result or not. The only study comparing both approaches that can be found in literature proposes a theoretical comparison based on models of sleeping and rate adaptation general techniques [20]. In particular, as this study was published in 2008, before the adoption of IEEE 802.3az, the employed sleeping model is using values differing by an order of magnitude from the one implemented in Low Power Idle (for the switching time for instance). This study does not try to combine both approaches.

In this paper, we propose a simulation-based comparison relying on an implementation of the two existing protocols (LPI and ALR) under various traffic conditions, and we provide a quantified study of both approaches separately and combined. This evaluation of the practical implementations of the two main energy-efficient techniques found in literature – namely sleeping and rate adaptation – quantifies the impacts of both techniques combined and separately, on energy consumption and quality of service (QoS).

Contrarily to previous work, we show that LPI has a clear advantage in terms of energy savings compared to ALR, and an even larger advantage on QoS for most of the traffic scenarios. Our results also indicate that combining

both protocols, LPI and ALR, reduces the energy saving dependence to packet coalescing. But, at the same time, it hugely impacts QoS, thus making LPI alone more suitable.

Section 2 introduces the context, shows the different solutions proposed in literature and presents a comparative study of these solutions. Section 3 describes our experimentation conditions. The validation simulations are conducted using our ECOFEN module implemented within the ns3 network simulator. Simulation results are provided in Section 4 along with considerations on how to improve these solutions. Section 5 concludes and presents future work.

2 State of the Art

Internet traffic presents a high dynamicity and variability [11, 22]. Typically, the Amsterdam Internet Exchange point [2] handles three times more traffic during its peak periods than during its low periods on a daily basis. This fluctuation leads to infrastructure over-provisioning and energy waste for QoS purpose. Indeed, networks usually stay fully operational at any time even during low periods.

2.1 Rate adaptation: Adaptive Link Rate (ALR)

ALR exploits low-traffic periods to downgrade the negotiated link rate and to save energy [16]. Practically, for backward compatibility reasons, 10 Gbps ports can operate at 1 Gbps, 100 Mbps or 10 Mbps, and consume less under lower rates. For instance, a port operating at 1 Gbps consumes 9 times more power than when operating at 100 Mbps [29]. However, switching between rates can be costly in terms of time and energy. For instance, switching from 100 Mbps to 10 Mbps requires 575.8 ms and consumes 0.8 Watts on average, while the port consumes 0.4 Watts at 100 Mbps and 0.1 Watt at 10 Mbps [29].

ALR [5] is relying on a MAC handshake protocol to negotiate the rate between the ports sharing the same link and requires the physical layers of both ports to be resynchronized [18]. This process explains the lengthy switching times. The policy for switching between rates is based on a dual threshold on the buffer occupancy: a lower threshold and an upper one [17]. When the upper threshold is reached, ALR initiates a switching to an upper rate, and symmetrically, when the buffer occupancy goes under the lower threshold, ALR switches to a lower rate. These two thresholds have to be carefully tuned in order to avoid oscillations [17].

2.2 Sleeping: Low Power Idle (LPI)

LPI also relies on low-traffic periods to save energy, but with a different approach: it puts the device in a low-power mode at the Ethernet physical layer level when there is no traffic for a short period of time. A refresh signal is then sent periodically to avoid renegotiating link parameters when coming back to normal

mode [12]. Typically, the energy consumption when the device is in low power mode can be as low as 10% that of the active mode [12].

As stated in the IEEE 802.3az standard, the length of the transitions to and from the LPI mode are not negligible and these transitions also consume energy [1]. For instance, according to the standard [1], from 100 Mbps it takes $200\mu s$ to go to LPI mode, and $30\mu s$ to wake up, while the transmission time for a 1500 byte frame is $120\mu s$. These values are three orders of magnitude lower than the rate switching times of ALR. In order to increase LPI energy savings, coalescing strategies have been developed to buffer incoming packets and to send them in burst, thus increasing sleeping duration [9]. A typical coalescing strategy consists of combining a timer and a buffer size threshold to trigger the sending burst [12].

Figure 1 summarizes the different options for using sleeping and rate adaptation techniques and the corresponding impact on power consumption.

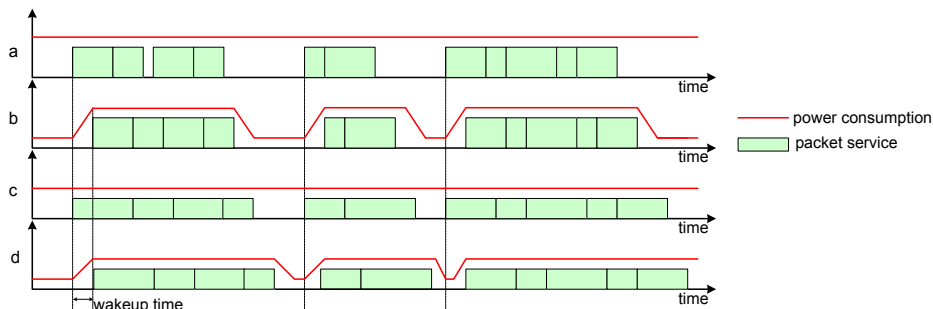


Fig. 1: Packet service times and power consumption in the following cases: (a) no power-aware optimizations, (b) sleeping, (c) rate adaptation, (d) sleeping and rate adaptation (from [8]).

3 Experimentation conditions

We evaluate the energy savings-QoS trade-off achieved by ALR, LPI, and ALR and LPI combined, and the impact of traffic characteristics on this trade-off. For comparing sleeping and rate adaptation techniques, we have implemented ALR and LPI protocols within the network simulator ns3 [21]. For the simulations, a single link between two nodes is considered with a 100 Mbps bandwidth and a latency of 5 ms. The injected traffic is described in the next section.

3.1 Energy consumption

The energy consumption for the network devices has been computed using the models provided in [24]. These models include per-byte and per-packet energy costs for Ethernet ports as measured on real hardware [28].

Concerning the energy consumption of ALR and LPI operations (switching), we have used real measurements from the literature: [26] for LPI (made on a RTL8111E Realtek NIC implementing draft 3.2 of the LPI standard) and [29] for ALR (using a Xilinx Virtex-II Pro NetFPGA). Concerning ALR, we have used the dual threshold policy described in [17]. For LPI, we have implemented the coalescing strategy with a fixed timer as described in [12]. Table 1 summarizes the main time and power costs per network port as utilized in our simulations for ALR and LPI.

Table 1: Time and power costs per network port utilized for the simulations

Category	State	Time (s)	Power (W)
Idle consumption	100 Mbps	-	0.4
ALR	10 Mbps \rightarrow 100 Mbps	72.4×10^{-3}	0.8
	100 Mbps \rightarrow 10 Mbps	574.8×10^{-3}	0.8
	at 10 Mbps	-	0.1
LPI	Time to awake T_w	30×10^{-6}	0.8
	Time to sleep T_s	200×10^{-6}	0.8
	in LPI mode	-	0.04

3.2 Traffic characteristics

Our experiments target a quantification of the energy consumption and the QoS parameters (latency and jitter) of ALR and LPI over a single link connecting two nodes. A single flow is injected from one node to the other one. We vary three main characteristics of this flow:

- α_d : the flow density, the ratio of sent bits per second over the link capacity. From the buffer threshold policy analysis, this utility ratio seems to be the main parameter influencing ALR.
- α_r : the flow regularity, it characterizes the delay variance between each sent packet. This parameter characterizes the time gap between consecutive packets, and thus may influence the ability of both approaches to save energy.
- α_c : the flow coalescence, the probability of packets to be sent by bursts. As explained in the description of LPI, coalescence can greatly help LPI to reach larger energy savings.

Each parameter is normalized (belongs to $[0, 1]$). Then, flows are generated according to these parameters as follows. The density parameter α_d allows to compute the mean delay between each packet: $T_{delay} = \frac{P_{size}}{C_{link} \times \alpha_d}$

With C_{link} the link capacity and P_{size} the packet size. The regularity parameter α_r determines the random variable Y used to define the delay between two consecutive packets. Following previous traffic studies [15, 6], we set: $Y = \alpha_r \times T_{delay} + (1 - \alpha_r) \times X$

With X a random variable following an exponential law of parameter $\lambda = \frac{1}{T_{delay}}$. When α_r goes to 1, this guarantees that the density parameter is respected, without having a perfectly (non realistic) regular flow. Finally, the coalescing parameter α_c determines the probability for a packet to be delayed and sent with the next one (if the next one is also delayed, they are both sent with the packet after the next one, and so on).

The graphs presented in this paper result from more than 30,000 simulations overall (with varying the three parameters). The link capacity is set to 100 Mbps for simulation duration constraints (a 1 Gbps link produces much longer simulations). ALR can reduce this rate to 10 Mbps if the utility ratio is under 0.1 following its buffer threshold policy. For each given triplet $(\alpha_d, \alpha_r, \alpha_c)$, the same flow is simulated with ALR, LPI, and ALR and LPI combined (thanks to the deterministic pseudo-random generator of ns3). The simulation is also run without any energy saving techniques to norm the energy consumption results of the different experiments (ie. the 100% in energy on the graphs represents this scenario). We observe that the energy consumption of the two ports without energy saving techniques is independent from the three parameters $(\alpha_d, \alpha_r, \alpha_c)$. Then the same experiments are repeated with a different seed for the random variables. Each point on the figures is thus the average value of 10 different runs.

4 Energy Savings and Performance Degradation of Sleeping and Rate Adaptation Techniques

4.1 Energy consumption

Figure 2 shows the energy consumption of the two connected ports with ALR as a function of the coalescence parameter α_c and the density parameter α_d . The energy is shown in percentage of the consumption without energy saving technique. As expected, energy savings occur only below 10% of link utilization (ie. $\alpha_d < 0.1$). The observed results for energy saved are similar to those observed in literature [29, 3].

In the best case, when the utilization is really low (α_d close to 0), the overall energy consumption reduces to 25%. Unfortunately, such a low bandwidth utilization is not frequent in all kinds of network. While it can often happen in access networks, it rarely occurs in core networks for instance. It can also be observed that, for beneficial traffic for ALR (traffic below 10% of link capacity), packet coalescence increases the energy savings by less than 10%.

Figure 3 presents the same experiment using LPI instead of ALR. As expected, the energy consumption in this case highly depends on the packet coalescence: it diminishes with the coalescence increase. These results are in line with previous estimations from the literature [14, 12]. This coalescence can be

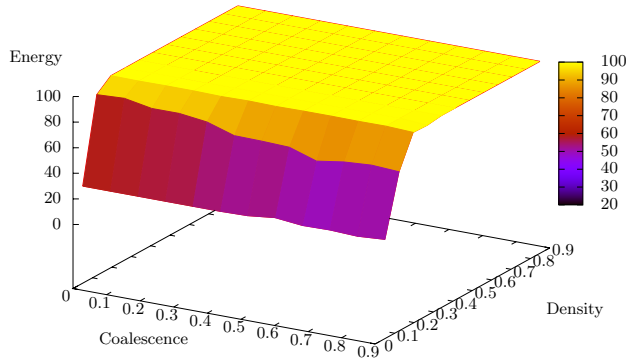


Fig. 2: Energy consumption when using ALR

reached either by traffic engineering at the application level or by utilizing coalescing buffers within the ports.

Similarly to ALR, LPI is sensitive to the link utilization. The lower α_d is, the higher energy savings are. In the worst scenario for LPI, it can still reach 40% of energy savings, and 80% in the best case.

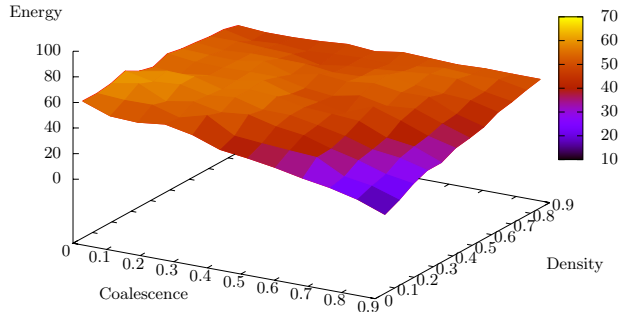


Fig. 3: Energy consumption when using LPI

In both cases ALR and LPI, we observe that the flow regularity has low impact on energy consumption, although it slightly improves the savings when the flow is less regular (α_r small). More generally, it appears that, from a pure energy saving point of view, LPI behaves better than ALR in most of the cases. Only under 10% of utilization and without packet coalescence, ALR allows for slightly greater energy savings than LPI. However, as shown in [29], with ALR, the link has to stay several seconds in the lower rate to amortize the switching energy cost.

Concerning ALR, it would probably perform better if more rates were available from a hardware point of view, as it is the case for the CPU frequencies and DVFS (Dynamic Voltage Frequency Scaling) [27]. However, the availability of

these rates in current network ports results from a side effect of the heterogeneity and backward compatibility principle that is essential in today’s networks. Indeed, the lower frequencies of a network port are used if the port of the corresponding device to which it is connected is not able to operate at this speed. Introducing intermediate rates without any compatibility requirement coming from previous Ethernet norms will need an important standardization effort from the manufacturers to offer identical rates for all new devices. Furthermore, it would need a thorough analysis on real hardware to measure the potential energy-related benefits of new rates.

Figure 4 shows the results with ALR and LPI combined. Although, higher energy savings are reached (more than 80%), the dependency to packet coalescence (parameter α_c) has disappeared. In this scenario, savings are bigger on average than the LPI-only case and packets coalescence is not useful any more. Indeed, when switching to lower rates, ALR increases per-packet transmission time and thus, naturally coalesces packets.

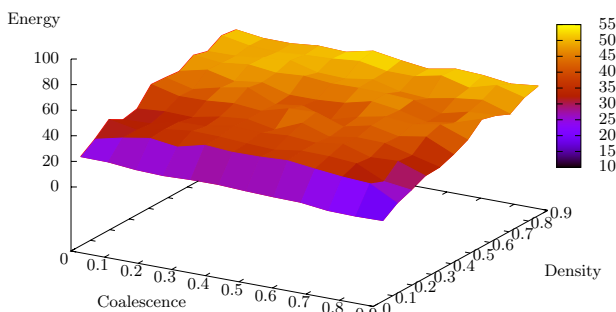


Fig. 4: Energy consumption while combining ALR and LPI

These experiments allow us to confirm that our implementations behave according to the literature for the ALR and LPI alone scenarios. Combining both approaches, we have found original results not previously studied in related work. Moreover, from a technical point of view, combining ALR and LPI required the design of a new algorithm to prioritize actions done by the two protocols. For instance, when the link utilization is low, if we use ALR first, idle periods are reduced because the rate is lower, and thus, LPI switches to low power mode for smaller periods although it could lead to better savings than ALR.

4.2 Quality of Service

In the three scenarios (ALR, LPI and ALR+LPI), the energy-efficient protocols are impacting the packet delivery, and thus the QoS experienced by the users. These impacts concern:

- the bandwidth, when ALR is reducing the link rate;

- the latency, when LPI is in low power mode (this adds the time to wake up the port), or when ALR is reducing the link rate;
- the jitter, when using packet coalescence to improve LPI performances for instance.

Here we will quantify these impacts on the QoS for the experiments performed in the previous section. For the simulations, the link latency was set to 5 milliseconds (typical LAN network). For LPI alone, the measured latency increase is negligible in most cases. Indeed, the time parameters of LPI (T_s , T_r and T_w) are in the order of microseconds, and thus have little impact on the end-to-end packet latency. In the worst case, which is for heavy and highly irregular traffic, the latency does not exceed 6.5 milliseconds. In this case, LPI has more chances to switch to low power mode while receiving new packets. The port needs then to completely switch to low power mode before waking up and sending the packets.

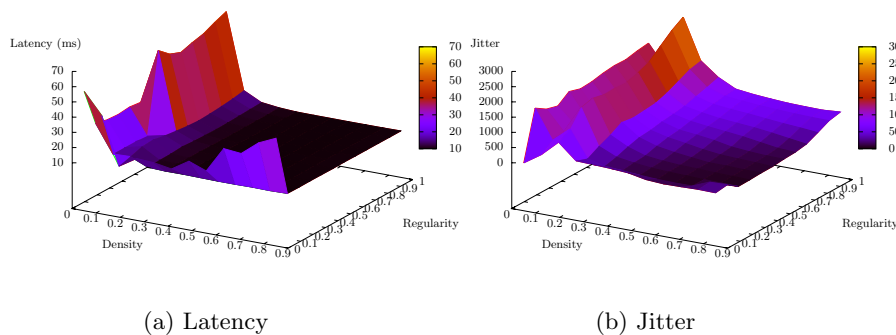


Fig. 5: Latency and jitter with ALR

Figure 5 shows the latency (in milliseconds) induced by ALR depending on flow density and flow regularity. In the worst case, when ALR is energy-efficient (link utilization under 10% = $\alpha_d \approx 0.1$), the latency can go up to almost 50 ms, against 5 ms usually. This result comes from ALR which reduces the link rate, and this automatically increases the reception delays. The jitter is particularly high when the link utilization is around 10% because this is when ALR tends to oscillate between high and low rate, thus inducing substantial oscillations on the latency. Under this threshold, latency is stabilized around 50 ms, and over this threshold, it is stabilized around 5 ms, thus keeping the jitter stable in these two regions.

In ALR and LPI cases, this QoS degradation comes from the switching times between states (rates for ALR and sleep/aware for LPI). ALR, with its switching time three orders of magnitude higher than LPI, performs poorly in comparison. Moreover, while LPI's switching times depends mostly on hardware, ALR's switching times result mainly from network protocol costs (physical layer resyn-

chronization) [29]. Switching times are thus more complex to reduce for ALR than for LPI.

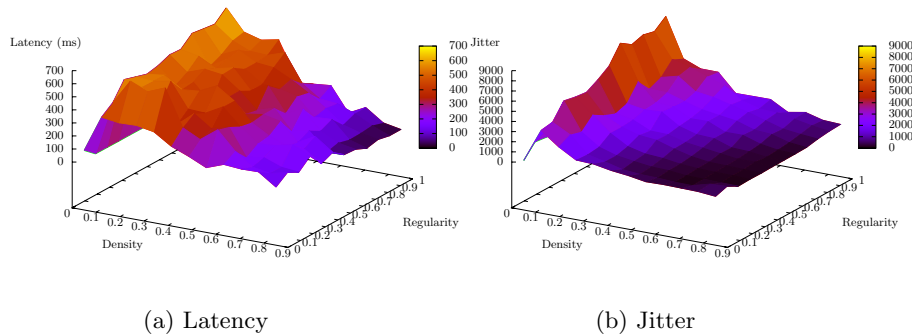


Fig. 6: Latency and jitter with ALR and LPI combined

If the combined use of ALR and LPI seems promising in terms of energy savings, its effects on QoS are less pleasing. Figure 6 shows the QoS provided when combining ALR and LPI. When flow density is small, latency and jitter are high. The worst case is still around $\alpha_d = 0.1$, but the latency is ten times worse than in the case of ALR alone, while the jitter is three times worse. ALR rate adaptation mechanisms are even less reactive when combined with sleeping periods due to LPI switching to low power mode. On a small network, such a poor QoS is not bearable for latency-critical applications like voice over IP. On a larger network, latency may accumulate at each device and results could even be worse from a QoS point of view.

4.3 Heterogeneous network

Our second scenario considers a realistic network based on the description of an Italian Internet Service Provider (ISP) given in [10]. This scenario aims at evaluating the performances of ALR and LPI on an heterogeneous network, where links do not present the same throughput. While the authors of [10] consider switching off entire nodes (i.e. routers or switches), here, we only act on the links and associated ports through ALR and LPI. Our approach is thus less disruptive and easier to apply in practice (no re-routing protocol needed). As we do not consider switching off entire network nodes, in the following experiments, we do not take into account the static energy consumption of these devices (it remains the same values for all the experiments).

The original network is hierarchical and presented in [10]; it consists of 8 core nodes, 52 backbone nodes, 52 metro nodes and 260 feeders. Due to long simulation duration and for symmetry reasons, we accurately simulate only a quarter of this network, the rest is aggregated into several nodes. The simulated network is presented in Figure 7.

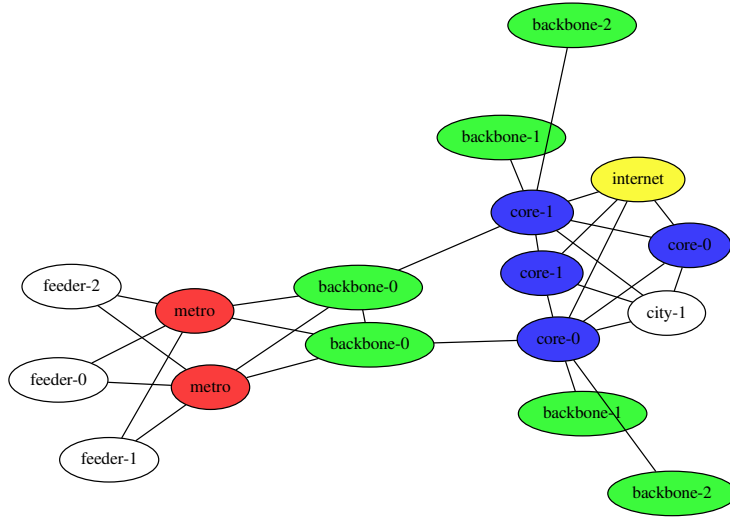


Fig. 7: Simulated representative section of the ISP network

The network has a hierarchical design, it is composed of:

- *core nodes*: they are connected among themselves through 50 Gbps links, each link being redundant for failure protection. They represent points of presence in a major city (4 core nodes per city). The second city is aggregated in one node called *city-1* in our scenario.
- *backbone nodes*: they are connected to the core nodes through 20 Gbps links, each backbone node being connected to two different core nodes. They are spread across the city.
- *metro nodes*: they are connected to two backbone nodes each through 20 Gbps links.
- *feeders*: they are connected to metro nodes through 10 Gbps links. They are responsible for bringing connectivity to the DSLAMs to which users are connected, so each feeder aggregates traffic from users in the same neighborhood or small town.
- *Internet node*: two central core nodes are connected to an Internet node by means of a 100 Gbps link, representing the gateway to access the Internet and the other networks (like an Internet exchange point).

In this scenario, only feeders and the Internet nodes are able to generate traffic, the other nodes only route the traffic. On Figure 7, the nodes aggregating the traffic from nodes of the original network design presented in [10] are *backbone-1*, *backbone-2*, *core-0*, *core-1* and *city-1*. For simulation reasons also, we have divided all the link throughputs by 1,000; so the smaller links have a 10 Mbps bandwidth instead of a 10 Gbps. Our experiment simulates one hour of real traffic, and in spite of our simplifications for reducing the simulation duration, it still lasts more than 40 hours on average (for 1 simulated hour).

Connections are distributed according to a normal law centered on 30 minutes, in order to show a peak of use in the middle of the simulation. Hence, the behaviors of ALR and LPI can be studied under different traffic loads. For a realistic traffic distribution, we consider that 70% of the connections are between the Internet node (outside the ISP) and one of the feeders, and 30% are between two feeders of the ISP [10].

In an heterogeneous network, ALR and LPI may present different behavior depending on the considered network part. Figure 8 compares the energy consumed by the different network parts during the simulation with ALR, LPI and ALR+LPI combined. The energy is expressed as the ratio between the considered consumption and the network consumption without any energy saving method (represented by 1).

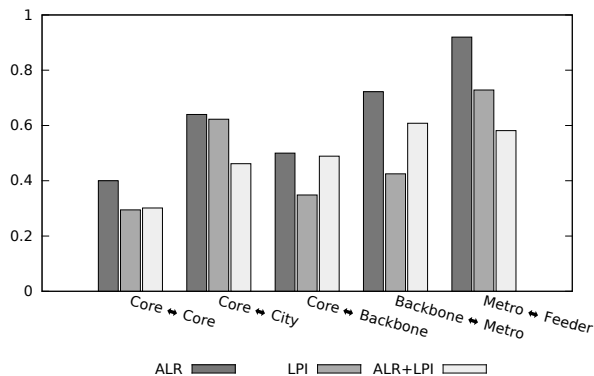


Fig. 8: Energy consumption ratio between the different energy saving options and the case where no energy saving technique is applied

One can see that the largest energy savings are realized on the most consuming links: at the core of the network between core nodes. For these devices, whatever the considered techniques, it leads to at least 60% of the energy saved compared to no energy reduction technique. However, unlike in the first scenario, LPI allows for the largest energy savings among the 3 considered options. Combining ALR and LPI allows for better energy savings at the edge of the networks for links between metro nodes and feeders. In this case, ALR+LPI can save up to 40% of the energy on average, against 25% with LPI only and 5% with ALR only.

Although overall energy savings are smaller on edge links, they are largely more numerous than core links. Indeed, considering the entire original network presented in [10], one can estimate that the energy consumed by the network ports at the edge (feeders) represent 36% of the overall energy consumed by all the ports of the network. One should notice that this value is different from the one presented in [10] as we only consider the dynamic energy consumption of

the network devices (consumption of the ports) which are the only elements on which one can act when not considering to switch off entire devices (contrary to what is done in [10]). Figure 9 presents the energy consumption proportion for the network ports by network category.

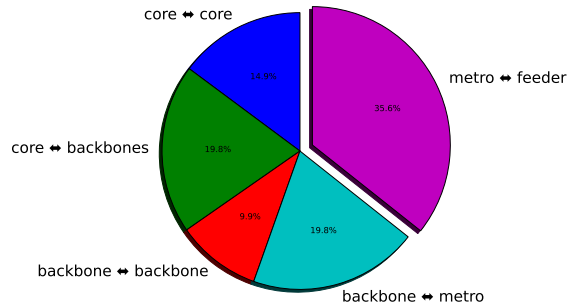


Fig. 9: Energy consumption of the network ports by category of network

The largest energy consumption part resides in the network between the metro network and the feeders. For this part, the ALR+LPI option reaches the lowest energy consumption. Yet, as outlined by the first scenario, ALR and ALR+LPI are leading to unacceptable QoS degradation that make them unsuitable for users. In this scenario, we consider only ISP point of view: residential and access networks (last mile) are not taken into account in the overall picture.

5 Conclusion

In this article, we provide a quantified comparison of ALR and LPI, two protocols implementing the two main energy-efficient approaches of the literature. This simulation-based study has been conducted in terms of energy savings and QoS. It appears that in the majority of the cases, LPI allows for more savings than ALR, independently from the network topology. Indeed, the limited number of available rates for current network ports (10Mbps, 100Mbps, 1Gbps, 10Gbps) constrains the energy efficiency of ALR to operate only under 10% of link usage. However, under this 10% limit, some rare cases show better savings for ALR than for LPI, in particular when the coalescence of the traffic is low.

The previous literature study comparing these two generic approaches was notably more to the benefit of rate adaptation than sleeping [20]. This study from 2008 was considering much higher switching costs for sleeping technique than the one achieved in practice with LPI, and switching costs between states (rates for rate adaptation and sleep/awake for sleeping) constitute the most influencing factor on energy savings and QoS.

This study also shows that combining LPI and ALR provides better energy savings than with LPI only. In particular, combining ALR with LPI would reduce the impact of coalescing on the reached energy savings. However, these energy savings come at the cost of significantly higher latency and jitter which are not acceptable for end-users. LPI alone would then be a more suitable option to save energy while keeping expected QoS levels. Consequently, with the current state-of-the-art hardware, ALR should stop being considered as a suitable solution by the community. Our future work include exploring future hardware architecture achieving a relative low static power dissipation compared to the dynamic part, and studying from which ratio ALR would start to be more beneficial.

Acknowledgments

Experiments presented in this paper were carried out using the Grid'5000 experimental test-bed, being developed under the Inria ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see <https://www.grid5000.fr>).

The authors would like to thank the reviewers for their valuable comments.

References

1. IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 3: CSMA/CD Access Method and Physical Layer Specifications Amendment 5: Media Access Control Parameters, Physical Layers, and Management Parameters for Energy-Efficient Ethernet. IEEE Std 802.3az-2010 (Amendment to IEEE Std 802.3-2008) pp. 1–302 (Oct 2010)
2. Amsterdam internet exchange (accessed in October 2016), <https://ams-ix.net/technical/statistics>
3. Anand, H., Reardon, C., Subramaniyan, R., George, A.: Ethernet adaptive link rate (alr): Analysis of a mac handshake protocol. In: 31st IEEE Conference on Local Computer Networks. pp. 533–534 (2006)
4. Barroso, L., Holzle, U.: The case for energy-proportional computing. *Computer* 40(12), 33–37 (2007)
5. Bennett, M., Christensen, K., Nordman, B.: Improving The Energy Efficiency Of Ethernet: Adaptive Link Rate Proposal. Ethernet Alliance White Paper (2006)
6. Benson, T., Akella, A., Maltz, D.: Network traffic characteristics of data centers in the wild. In: Conference on Internet measurement (IMC). pp. 267–280 (2010)
7. 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* (2011)
8. Bolla, R., Bruschi, R., Davoli, F., Cucchietti, F.: Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Comm. Surv. and Tutorials* 13(2), 223–244 (2011)
9. Chatzipapas, A., Mancuso, V.: Measurement-based coalescing control for 802.3az. In: IFIP Networking Conference (Networking) and Workshops. pp. 270–278 (2016)
10. Chiaraviglio, L., Mellia, M., Neri, F.: Energy-aware backbone networks: a case study. In: IEEE Int. Conf. on Communications (ICC) Workshops. pp. 1–5 (2009)

11. Christensen, K., Gunaratne, C., Nordman, B., George, A.: The next frontier for communications networks: power management. *Computer Communications* 27(18), 1758–1770 (2004)
12. Christensen, K., Reviriego, P., Nordman, B., Bennett, M., Mostowfi, M., Maestro, J.: IEEE 802.3az: the road to energy efficient ethernet. *IEEE Communications Magazine* 48(11), 50–56 (2010)
13. The Zettabyte Era: Trends and Analysis. Tech. rep., Cisco (2016)
14. De La Oliva, A., Hernández, T.R.V., Guerri, J.C., Hernández, J.A., Reviriego, P.: Performance analysis of energy efficient ethernet on video streaming servers. *Computer Networks* 57(3) (2013)
15. Ersoz, D., Yousif, M., Das, C.: Characterizing Network Traffic in a Cluster-based, Multi-tier Data Center. In: *International Conference on Distributed Computing Systems (ICDCS)* (2007)
16. Gunaratne, C., Christensen, K., Nordman, B.: Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed. *Int. Journal of Network Management* 15(5), 297–310 (2005)
17. Gunaratne, C., Christensen, K., Suen, S.: Ethernet adaptive link rate (alr): Analysis of a buffer threshold policy. In: *IEEE Global Telecommunications Conference (GLOBECOM'06)*. pp. 1–6 (2006)
18. Gunaratne, C., Christensen, K., Nordman, B., Suen, S.: Reducing the energy consumption of ethernet with adaptive link rate (alr). *Computers, IEEE Transactions on* 57(4), 448–461 (2008)
19. Impact of ICT on the Energy Consumption Around the World. Tech. rep., National Academy of Technologies of France (2014)
20. Nedeveschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., Wetherall, D.: Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In: *USENIX Symp. On Netw. Systems Design & Implementation (NSDI)*. pp. 323–336 (2008)
21. ns3 network simulator, <http://www.nsnam.org>
22. Odlyzko, A.: Data Networks are Lightly Utilized, and will Stay that Way. *Review of Network Economics* 2 (2003)
23. Orgerie, A.C., Dias de Assunção, M., Lefèvre, L.: A Survey on Techniques for Improving the Energy Efficiency of Large-scale Distributed Systems. *ACM Computing Survey* 46(4) (2014)
24. Orgerie, A.C., Lefèvre, L., Guérin-Lassous, I., Lopez Pacheco, D.: ECOFEN: an End-to-end energy Cost mOdel and simulator For Evaluating power consumption in large-scale Networks. In: *SustaInet: Workshop on Sustainable Internet and Internet for Sustainability* (2011)
25. Patel-Predd, P.: Energy-efficient ethernet: Ethernet connections waste lots of watts. it need not be so. *IEEE Spectrum Magazine* (2008)
26. Reviriego, P., Christensen, K., Rabanillo, J., Maestro, J.: An Initial Evaluation of Energy Efficient Ethernet. *IEEE Communications Letters* 15(5), 578–580 (2011)
27. Shang, L., Peh, L.S., Jha, N.: Dynamic voltage scaling with links for power optimization of interconnection networks. In: *International Symposium on High-Performance Computer Architecture (HPCA)* (2003)
28. Sivaraman, V., Vishwanath, A., Zhao, Z., Russell, C.: Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router. In: *IEEE INFOCOM Workshops*. pp. 331–336 (2011)
29. Zhang, B., Sabhanatarajan, K., Gordon-Ross, A., George, A.: Real-time performance analysis of adaptive link rate. In: *IEEE Conference on Local Computer Networks (LCN)*. pp. 282–288 (2008)