Leveraging Energy Saving Capabilities of Current EEE Interfaces via Pre-Coalescing[⋆,⋆⋆]

Miguel Rodríguez-Pérez[*], Sergio Herrería-Alonso, Raúl F. Rodríguez-Rubio, José Carlos López-Ardao

Telematics Engineering Department, EE Telecomunicación, Rúa Maxwell s/n, University of Vigo, 36310 Vigo, Spain.

Abstract

The low power idle mode implemented by Energy Efficient Ethernet (EEE) allows network interfaces to save up to 90 % of their nominal energy consumption when idling. There is an ample body of research that recommends the use of frame coalescing algorithms—that enter the low power mode as soon as there is no more traffic waiting to be sent, and delay the exit from this mode until there is an acceptable amount of traffic queued—to minimize energy usage while maintaining an acceptable performance. However, EEE capable hardware from several manufactures delays the entrance to the low power mode for a considerable amount of time (hysteresis). In this paper we augment existing EEE energy models to account for the hysteresis delay and show that, using the configuration ranges provided by manufacturers, most existing EEE networking devices are unable to obtain significant energy savings. To improve their energy efficiency, we propose to implement frame coalescing directly at traffic sources, before reaching the network interface. We also derive the optimum coalescing parameters to obtain a given target energy consumption at the EEE device when its configuration parameters are known in advance.

Keywords: Energy efficient Ethernet, Traffic coalescing, Local area networks, Modeling and simulation, Green communications

1. Introduction

Energy Efficient Ethernet (EEE) interfaces are already widespread, as they have been around us for the last ten years [1, 2]. These interfaces are able to

[⋆]This work was supported by the European Regional Development Fund (ERDF) and the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (atlanTTic).

[⋆⋆]This work was funded by the "Ministerio de Economía, Industria y Competitividad" through the project TEC2017-85587-R of the "Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad", (partly financed with FEDER funds).

[*]Corresponding author

avoid wasting precious energy when they are not transmitting data. For this, they implement one or more low power modes [1, 3] that only spend a fraction of the nominal power usage, but that cannot be employed during normal data transmission. However, going in and out from these low power modes is not free, as it takes some time and energy that are otherwise unavailable for doing useful work.

The compromise between energy efficiency and the time spent transitioning to and from low power idle (LPI) modes has been the subject of many in-depth analyses, as it has direct consequences in two very important metrics: energy consumption and traffic delay. A simple, yet efficient, way to improve energy efficiency consists in amortizing the transitions among several frames. Instead of exiting LPI as soon as a new frame arrives, it is more efficient to wait until a larger set of frames are waiting for transmission and then perform a single transition back to the active state for the whole set. This coalescing approach [4] clearly trades some delay for greater energy savings [5, 6]. Many coalescing proposals employ the number of queued frames as the condition to exit LPI, but time-based coalescers, that use the time since the first frame arrival while in LPI, can also be configured to obtain identical results [7, 8].

A complementary approach to avoid excessive transitions consists in delaying the entrance into the LPI mode when the transmission queue is drained. In this case, the EEE interface waits for new immediate arrivals during a small additional hysteresis time before entering the LPI mode. Thus, if the time until the next frame arrival is shorter than this hysteresis time, traffic does not get unnecessarily delayed and some energy is saved, since entering the LPI mode is not efficient enough to compensate for the energy consumed during the state transitions. However, if the frame arrives just after the hysteresis has ended, traffic gets delayed and the energy employed waiting during the hysteresis time becomes wasted. In fact, it has been proved that for low traffic loads, adding hysteresis results in greater energy usage [9].

Most hardware manufacturers have included EEE low power modes in their products. Many of them have chosen to implement both a hysteresis delay and a time-based frame coalescer to improve efficiency.[1] However, their tuning of these parameters, and the available configuration range, is too much conservative, as we will prove later. Minimum values higher than 20 µs are normal for the hysteresis length of 10 Gbit/s interfaces, even reaching hundreds of microseconds. Regarding the coalescing timer, when it is available, its maximum configurable value is sometimes too short to compensate for high hysteresis times. In Table 1 we provide a small sample of the available configuration ranges for both the hysteresis and coalescing timers of some popular EEE devices. Note that the last three devices feature 1 Gbit/s ports, but we added them for completeness, as they have been thoroughly analyzed in [15].

[1] Regretfully, not all manufacturers provide details about the actual implementation characteristics of their EEE mode. Even more, some simply let the administrator enable or disable the EEE mode, but not to tune its parameters.

Table 1: Available Configuration Options of Several Providers of EEE Capable Devices

Manufacturer	**Hysteresis (µs)**	**Coalescing (µs)**
Cisco Nexus 7000 [10]	20 and 600	6
Dell EMC N-Series [11]	600 to 4 294 967 295	0 to 65 535
QLogic bnx2 [12]	256 to 1 048 575	?
Intel X550 [13] & X710 [14]	1 to 63	?
D-Link DGS-1100–16 [15]	0	0
Level-One GEU-0820 [15]	0	0
SMC GS801 [15]	0	0

This paper analyzes the energy efficiency limits of actual networking devices. For this, we contribute a new model for EEE interfaces with hysteresis. The model is completed to include the joint effects of hysteresis and time-based coalescing. The model shows that for most practical loads, and with the available values for hysteresis and coalescing delays, the LPI mode in some EEE hardware is not sufficient to reduce energy consumption. Our second contribution is a proposal to deploy time-based coalescers before the Network Interface Card (NIC) so as to overcome the limitations of networking equipment. This technique helps to overcome some of the drawbacks of devices that employ high hysteresis values and can also augment the saving of devices that lack any coalescing capabilities. We provide an analytic model for this technique and also solve its proper tuning to obtain any desired approximation to the optimal efficiency.

The rest of this paper is organized as follows. Section 2 summarizes previous work dealing with EEE. In Section 3 we present a power model for the behavior of actual hardware and provide early estimations of the expected energy savings. The pre-coalescing solution is shown and modeled in Section 4. We test the behavior of our solution in Section 5 and finally, we present our conclusions in Section 6.

2. Related Work

The lack of any prescribed governing algorithm for the LPI mode of EEE in [1] led to the development of many competing proposals. The simplest ones arrived soon and consisted in simply entering LPI as soon as there was no traffic left to be transmitted to then resume normal operation as soon as there was newly available traffic. These proposals are collectively known as *frame transmission* algorithms and their energy and delay performance models are already well known [16, 17, 18, 19]. These energy consumption models show that frame transmission causes a high number of transitions to and from LPI that severely hinders its performance, thus rendering it a poor choice for most traffic patterns and loads—recall that during these transitions the network interface draws about the same power than an active one.

The solution to this problem is to amortize these transitions among several frames. For this, the interface is not immediately woken up when there is

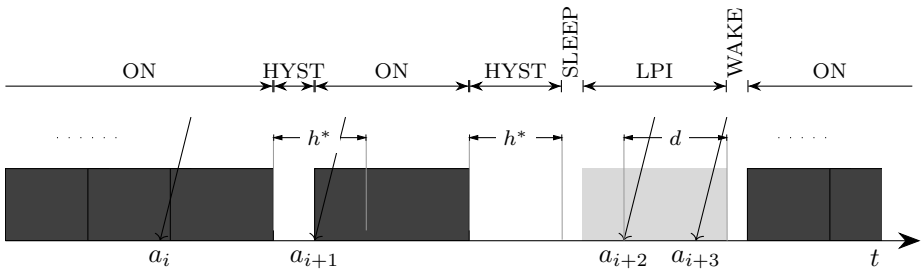


Figure 1: EEE operation with hysteresis and delay times.

newly available traffic, instead it is woken up when some more traffic has been accumulated. Some early works that propose this *frame coalescing* (or *burst transmission* as it is also known) are [2, 4]. There exist two main approaches for deciding when to wake up the interface. Size-based coalescers put an upper bound to the queue length while in LPI and resume normal operations once this threshold is reached.[2] On the other hand, time-based coalescers use the time spent in LPI, or since the first arrival while in LPI, to decide when to restart transmitting traffic. There is ample literature modeling the energy consumption [6] or even the energy-delay tradeoffs, either just for size-based [20] coalescers, for time-based [21] ones or even for the general case [22, 23, 7, 5, 24].

These models have been later employed to tune the frame coalescing algorithm, i.e., deriving the optimum upper bound, so as to obtain a given energy efficiency [7] or to meet a target average delay by limiting the maximum time in LPI [23, 25].

To the best of our knowledge, the introduction of some hysteresis time before entering LPI has not yet been considered in the research literature in the context of EEE, although it has been studied in analogous scenarios. Such is the case of cellular base stations, in order to reduce the number of transitions between different operating modes [26, 27]. However, as shown in [9], adding hysteresis can sometimes lead to increased energy consumption and it is largely unneeded when using a properly tuned frame coalescing algorithm.

3. Hardware Model

As it was already stated in the previous sections, currently deployed EEE interfaces use a time-based coalescing algorithm with hysteresis to drive the LPI mode, as represented in Fig. 1. That is, after the transmission queue becomes depleted, most EEE interfaces wait for the queue to remain empty for at least h^* seconds before entering the LPI mode, with h^* being the *hysteresis* time. Conversely, normal operation resumes d seconds after the first frame arrival in LPI. This coalescing time d is simply called *delay*. As aforementioned in

[2]To avoid excessively long waiting times, a safeguard in the form of a maximum sleeping time is usually implemented.

Section 2 the performance of EEE with hysteresis has not been studied yet in the literature, so we will provide an extended EEE energy usage model in this section taking it into account.

The purpose of the hysteresis time is to avoid entering LPI when the load is high, i.e., when frames arrive too close to each other. When this happens, entering and exiting LPI increases both delay and energy consumption, as the interface consumes about the same amount of energy during transitions as during regular transmissions. Hysteresis works on the idea that if the expected time in LPI is much smaller than the sum of the *sleep* and *wake up* transition times, the energy savings are probably not worth the added delay. Once the interface is in LPI, it is important to keep it in that state as long as possible to amortize the transition delays, but with a controlled latency. That is the objective of the *delay* value. In essence, it plays an analogous role to the queue threshold in EEE size-based coalescing algorithms [4, 5, 6].

In the following subsections, we will derive a power saving model for EEE interfaces with both a time-based coalescer, i.e., $d > 0$, and hysteresis, i.e., $h^* > 0$.

3.1. Energy Model Considering Hysteresis

As shown in [22], the average normalized energy consumption of an EEE interface can be calculated as

$$\sigma = 1 - (1 - \sigma_{\mathrm{LPI}})\rho_{\mathrm{LPI}}, \tag{1}$$

where σ_{LPI} is the normalized energy consumption of the LPI mode and

$$\rho_{\mathrm{LPI}} = \frac{\mathrm{E}\left[\mathrm{T_{LPI}}\right]}{\mathrm{E}\left[\mathrm{T_{cycle}}\right]} \tag{2}$$

is the normalized sleeping time, that is, the percentage of time the interface stays in LPI, as $\mathrm{T_{LPI}}$ is the duration of the LPI sub-period and $\mathrm{T_{cycle}}$ the time between two consecutive transitions from LPI to the normal operating mode.

Clearly, in a given cycle there can only be a single LPI period and, consequently, one transition to and from it. However, there can be several *ON* periods, as a single arrival during the hysteresis time prevents the interface from entering the low power mode. So

$$\mathrm{E}\left[\mathrm{T_{cycle}}\right] = \mathrm{T_S} + \mathrm{E}\left[\mathrm{T_{LPI}}\right] + \mathrm{T_W} + \mathrm{E}\left[n\right]\left(\mathrm{E}\left[\mathrm{T_{ON}}\right] + \mathrm{E}\left[h\right]\right), \tag{3}$$

with n the number of times the interface fails to enter LPI while in hysteresis, $h \leq h^*$ a random variable representing the actual length of a hysteresis interval, and $\mathrm{T_{ON}}$ the length of an ON interval. $\mathrm{T_S}$ and $\mathrm{T_W}$ are the transition times from the ON state to LPI and from LPI back to ON. They only depend on the interface characteristics, so we treat them as known constants.

If we assume a conservative service at the interface, it must hold that

$$\rho = \frac{\mathrm{E}\left[n\right]\mathrm{E}\left[\mathrm{T_{ON}}\right]}{\mathrm{E}\left[\mathrm{T_{cycle}}\right]}, \tag{4}$$

with ρ being the load factor, and, from (3) and (4), it immediately follows that

$$\rho_{\mathrm{LPI}} = (1-\rho)\frac{\mathrm{E}\left[\mathrm{T_{LPI}}\right]}{\mathrm{E}\left[\mathrm{T_{LPI}}\right] + \mathrm{E}\left[n\right]\mathrm{E}\left[h\right] + \mathrm{T_S} + \mathrm{T_W}}. \tag{5}$$

The average length of the hysteresis interval is simply

$$\mathrm{E}\left[h\right] = \lambda \int_0^\infty \min(t, h^*) f_e(t)\,\mathrm{d}t, \tag{6}$$

where $f_e(t)$ is the density function of the time between the end of the ON state and the next frame arrival. Additionally, if arrivals are independent, $\mathrm{E}\left[n\right]$ is just the average value of a geometric distribution parameterized by the probability that there are no arrivals during the hysteresis interval.

In the specific case of a Poisson process with interarrival times I and arrival rate λ it holds that

$$\mathrm{E}\left[n\right] = \frac{1}{\mathrm{P}(I > h^*)} = \mathrm{e}^{\lambda h^*}. \tag{7}$$

and, because of the PASTA property, $f_e(t)$ is the density function of an exponential distribution with parameter λ, so

$$\mathrm{E}\left[h\right] = \lambda \int_0^{h^*} t\mathrm{e}^{-\lambda t}\,\mathrm{d}t + \lambda h^* \int_{h^*}^\infty \mathrm{e}^{-\lambda t}\,\mathrm{d}t = \frac{(1-\mathrm{e}^{-\lambda h^*})}{\lambda}. \tag{8}$$

3.2. Model for Time-Based Coalescers with Hysteresis

We can extend the previous model to take into account the effect of the delay timer. We need to obtain the average time from the end of the transition to LPI until d seconds after the arrival of the first frame since the end of hysteresis. Particularizing directly for a Poisson arrival process, we get

$$\begin{aligned}\mathrm{E}\left[\mathrm{T_{LPI}}\right] &= \lambda \int_{(\mathrm{T_S}-d)^+}^\infty (t + d - \mathrm{T_S})\,\mathrm{e}^{-\lambda t}\,\mathrm{d}t \\ &= \begin{cases} \lambda^{-1} + d - \mathrm{T_S}, & \text{for } d > \mathrm{T_S}, \\ \frac{\mathrm{e}^{-\lambda(\mathrm{T_S}-d)}}{\lambda}, & \text{in every other case,} \end{cases}\end{aligned} \tag{9}$$

where $(x)^+ = \max\{x, 0\}$.

3.3. Model Results for Real Hardware

With the previous model, we can take a closer look at the expected energy performance of some representative hardware. According to configuration manuals from several manufactures, e.g., [10, 11], the delay parameter takes a default value in the order of 6 µs to 8 µs, while the minimum allowable hysteresis value can get as high as 600 µs for some of them (see Table 1). In Fig. 2 we compare the expected energy usage for an extremely ample range of traffic loads and different settings.

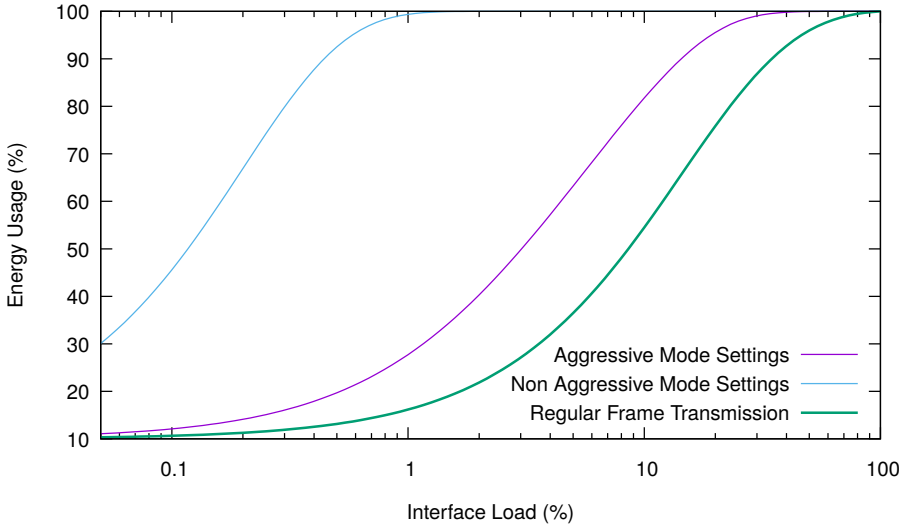


Figure 2: Expected energy usage for different values of h^* and d.

Table 2: Configuration Parameters for the EEE Interface

Interface Configuration	**Hysteresis (µs)**	**Delay (µs)**
Aggressive	20	6
Non Aggressive	600	6

We have employed two combinations of the hysteresis and delay parameters called *aggressive mode* and *non aggressive mode* as shown in Table 2. We use these values as representative of both extremes of available minimum hysteresis values, and use $d = 6\,\mu s$ as it is the default for [10]. We ignored interfaces such as [14] as, for practical purposes, their hysteresis can be deactivated.

Note that the non aggressive mode settings only make sense for loads well below 1 % and that even for the so-called aggressive settings we get much worse results than with the naive frame transmission algorithm with no coalescing whatsoever. Furthermore, the per frame maximum additional latency is smaller for frame transmission, as it adds a maximum delay of $T_S + T_W = 7.36\,\mu s$,[3] while typical interface settings are adding an additional delay of at least $\min\{d\} + T_W = 6\,\mu s + 4.48\,\mu s = 10.48\,\mu s$.

[3] Using $T_S = 2.88\,\mu s$ and $T_W = 4.48\,\mu s$ according to [1].

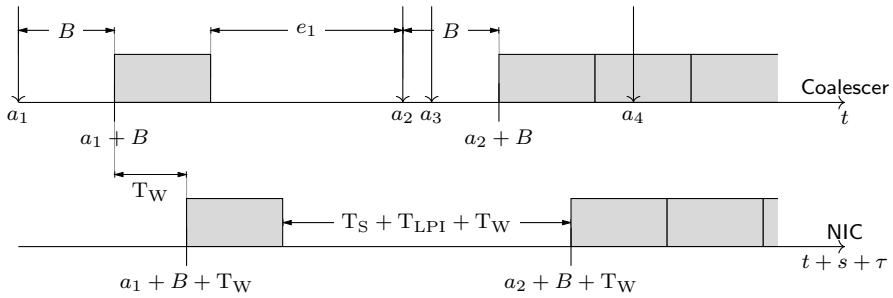


Figure 3: Time diagram of the pre-coalescing procedure. For simplicity, we assume $B + e_i > \mathrm{T_S}$ and $s_i = s$.

4. Pre-Coalescing

Such disappointing results may be overcome either by increasing coalescing delay if it is configurable or by performing frame coalescing *before* the traffic arrives at the EEE network interface (*pre-coalescing*). If the EEE NIC resides in the same hosts, this can be done by a traffic shaper placed just before the NIC. The technique can also be employed if the high hysteresis EEE NIC belongs to a switch. If its traffic is dominated by a single contributor, the latter can pre-coalesce the traffic before sending it to the switch. In both cases, the idea is to create artificial gaps between trains of coalesced frames (bunches) that let the interface become idle despite the hysteresis. This should result in an increase in the normalized average sleeping time. Pre-coalescing, when compared to traditional frame coalescing, has the extra advantage that it adds less latency for the same energy savings, as we will prove later.

The pre-coalescing algorithm works as follows. When an idling terminal has a new frame ready for transmission, it waits for B seconds before actually delivering the traffic to the network interface. Then, it dispatches frames at link rate until the queue depletes. Finally, when the coalescer delivers the last queued frame, it waits for the next frame arrival in order to form a new bunch.

4.1. Pre-Coalescer Energy Model

We now build an energy model of this frame pre-coalescer. We will assume for simplicity that the coalescer receives Poisson traffic with a general, albeit independent, frame size distribution. We are interested in the energy consumption of the network interface, but we cannot rely on the previous analysis as the traffic coming out of the coalescer shall not follow a Poisson process.

Fig. 3 shows a time diagram of the operation of both the coalescer and the network interface. In the diagram, a_i are the arrival times of different frames at the coalescer, $s_i = s$ is the corresponding frame size and e_i is the elapsed time since the transmission of the i-th bunch and the arrival of the next frame.[4] As

[4]In Fig. 3 we have made $s_i = s$ for simplicity, but the result holds for the general case.

the arrivals form a Poisson process of parameter λ, e_i is exponentially distributed because of the PASTA property.

We can notice in Fig. 3 how the operation states at the interface mimic those of the coalescer, with a small $\mathrm{T_W}+s+\tau$ delay, with τ being the propagation delay. When the coalescer and the interface reside in the same host, the propagation delay is $\tau = 0$. Furthermore, it is clear that the time between two consecutive active periods at the coalescer must comprise both EEE transitions ($\mathrm{T_S} + \mathrm{T_W}$) and the time spent in the low power mode ($\mathrm{T_{LPI}}$). As both the coalescer and the interface work at the same line rate, this must be equal to $e + B$: the time elapsed since the end of a bunch transmission until the next one starts being delivered by the coalescer. In other words,

$$\mathrm{T_S} + \mathrm{T_W} + \mathrm{E}\left[\mathrm{T_{LPI}}\right] = \mathrm{E}\left[e\right] + B = \lambda^{-1} + B, \tag{10}$$

so that

$$\mathrm{E}\left[\mathrm{T_{LPI}}\right] = \lambda^{-1} + B - \mathrm{T_S} - \mathrm{T_W}. \tag{11}$$

The cycle length is also identical both at the target interface and at the coalescer. We can take advantage of the fact that traffic arriving at the latter belongs to a Poisson process to obtain the average cycle length:

$$\mathrm{E}\left[\mathrm{T_{cycle}}\right] = \mathrm{E}\left[e\right] + B + \mathrm{E}\left[\mathrm{T_{busy}}\right], \tag{12}$$

where $\mathrm{T_{busy}}$ is the length of the coalescer busy cycle. For a M/G/1 system, $\mathrm{T_{busy}} = \frac{s'}{1-\rho}$, with s' the total work accumulated at the start of the cycle. In our case, the accumulated traffic at the start of transmission is just $\lambda B \cdot s$, so

$$\mathrm{E}\left[\mathrm{T_{cycle}}\right] = \mathrm{E}\left[e\right] + B + \lambda\frac{B \cdot s}{(1-\rho)}, \tag{13}$$

and therefore,

$$\rho_{\mathrm{LPI}} = (1-\rho)\frac{\lambda^{-1} + B - \mathrm{T_S} - \mathrm{T_W}}{(1-\rho)\lambda^{-1} + B}. \tag{14}$$

Please note that when $e + B < \mathrm{T_W}$, that is, the time elapsed since the end of the transmission until the arrival of the next frame at the coalescer plus the bunching period is less than the transition time, the interface stops being synchronized with the coalescer and (14) no longer holds. Fortunately, this case is not of practical interest. It will be later shown that such short B values produce negligible increments in energy savings.

4.1.1. Effects of Hysteresis and Delay

To model the effects of hysteresis, we proceed in a similar way. As long as $e + B > \mathrm{T_W} + h^*$, with h^* being the hysteresis configured in the interface, it stays in sync with the coalescer. The cycle length at the coalescer stays the same, as it is clearly unaffected by the behavior of the downstream interface. However, the duration between two consecutive transmissions at the interface has to accommodate the hysteresis time, so:

$$\mathrm{T_S} + \mathrm{T_W} + \mathrm{E}\left[\mathrm{T_{LPI}}\right] + h^* = \mathrm{E}\left[e\right] + B = \lambda^{-1} + B, \tag{15}$$

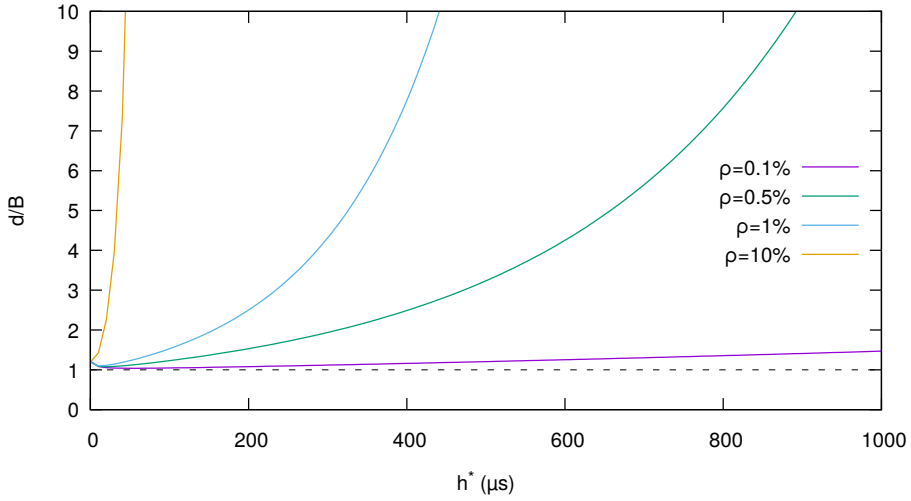


Figure 4: Relation between minimum delay (d) and bunch length (B) needed to reach the same consumption for different hysteresis values (h^*) and traffic loads (ρ).

and consequently,

$$\rho_{\mathrm{LPI}}^{\mathrm{hyst}} = (1-\rho)\frac{\lambda^{-1} + B - h^* - \mathrm{T_S} - \mathrm{T_W}}{\lambda^{-1}(1-\rho) + B}. \tag{16}$$

The case of added delay (time-based frame coalescing) at the interface in addition to bunching at the coalescer is more involved. The added wake up delay causes the interface to be delayed for d time units after the first cycle. If $d > B$, the states of the interface and the coalescer do no longer remain in sync and we cannot apply Poisson models at the latter to model the interface. However, usually $h^* \gg d$, so $B > d$. In this case, the cycle length at the coalescer obviously stays the same as in the previous case, and also at the interface, with the exception of the first interval that can be neglected for our analysis. Finally, the delay time is part of $\mathrm{T_{LPI}}$, so (14) holds for coalescing interfaces as long as $d < B$.

4.2. Pre-Coalescing vs. NIC Coalescing Frame Delay

The pre-coalescer energy model allows us to compare the benefits of pre-coalescing against time-based coalescing at the interface. In Fig. 4 we have represented the relation between the needed delay for a time-based coalescer (d) and the bunching length employed at the source (B) in order to guarantee the same energy consumption, assuming a Poisson traffic arrival process. For low traffic loads and small hysteresis values, both parameters remain close. However, when the load is moderate or high, pre-coalescing requires much smaller coalescing lengths. The reason is that pre-coalescing creates gaps between bursts of frames

before they get to the interface. These gaps are, by design, longer than the configured hysteresis, so the interface can always enter the LPI mode. However, traditional frame coalescing at the interface only takes place *after* the interface enters LPI. If the average interval between frame arrivals is smaller than the hysteresis time, as is the case with moderate loads, the interface seldom enters LPI. In this case, the few times the interface does enter LPI, it needs to stay a considerable amount of time on it to get the same energy savings as pre-coalescing.

4.2.1. Pre-Coalescing Algorithm Delay

The average frame delay can be modeled considering the coalescer+interface tandem as a GI/G/1 queue with added delay for the first customer in a busy period. In fact, the coalescer is akin to a queue that waits for B seconds before serving the first job in its cycle. The network interface should add no additional queuing delay, as both the coalescer and the interface work at the same rate. According to [28], and considering that waiting times and arrivals are actually uncorrelated—recall that the pre-coalescer uses a timeout B since the first frame arrival to start transmitting the bunch—the average waiting time of such a queue is simply

$$\mathrm{E}\left[W\right] = \frac{\lambda(\sigma_S^2 + \sigma_I^2)}{2(1-\rho)} + \frac{1-\rho}{2\lambda} + \frac{(B + \mathrm{T_W})^2 - \mathrm{E}\left[e^2\right]}{2(B + \mathrm{T_W} + \mathrm{E}\left[e\right])}, \tag{17}$$

where σ_S^2 is the transmission time variance, σ_I^2 is the variance of the arrival distribution and e is the random variable representing the length of the empty periods, that is, the interval length since the end of a cycle until the arrival of the next frame. In the case of a Poisson arrival process ($\sigma_I^2 = 1/\lambda^2$), recall that empty periods also follow an exponential distribution of parameter λ by the PASTA property, so that $\mathrm{E}\left[e\right] = 1/\lambda$ and $\mathrm{E}\left[e^2\right] = 2/\lambda^2$. Substituting all these values in (17), we finally arrive to

$$\mathrm{E}\left[W\right] = \frac{1 + \lambda^2\sigma_S^2}{2\lambda(1-\rho)} + \frac{1-\rho}{2\lambda} + \frac{\lambda^2(B + \mathrm{T_W})^2 - 2}{2\lambda\left(1 + \lambda(B + \mathrm{T_W})\right)}. \tag{18}$$

4.3. Tuning the Pre-Coalescing Algorithm

In this section we use (16) to derive the minimum B value needed for the interface to have a target energy profile. We consider two different example targets: matching the performance of the naive frame transmission algorithm and approximating the behavior of an ideal EEE interface. Throughout this section we will assume that $B > \{h^*, d, \mathrm{T_W}\}$.

4.3.1. Matching Frame Transmission Consumption

It is an established fact that the frame transmission algorithm achieves less than ideal energy savings for moderate and high traffic loads. However, interfaces using common hysteresis values get even worse results. On the other hand, for low traffic loads, frame transmission attains nearly-optimal results.

The energy usage of frame transmission under the hypothesis of Poisson traffic is well established in the literature [16, 18, 17, 22]. From (5) and (9), after making $h^* = 0$ and $d = 0$, we get that the normalized length of the frame transmission algorithm sleeping interval is

$$\rho_{\mathrm{LPI}}^{\mathrm{f}} = (1-\rho)\frac{\mathrm{e}^{-\lambda \mathrm{T_S}}}{\mathrm{e}^{-\lambda \mathrm{T_S}} + \lambda(\mathrm{T_S}+\mathrm{T_W})}. \tag{19}$$

Now, if we equal (19) to (16) and solve for B, after some straightforward simplifications we find that the optimum B value that matches frame transmission performance is

$$B_{\mathrm{f}}^* = \frac{\lambda\left[h^* + \mathrm{T_S} + \mathrm{T_W} + (\mathrm{T_S}+\mathrm{T_W})(\lambda(h^* + \mathrm{T_S} + \mathrm{T_W}) - 1)\mathrm{e}^{\lambda \mathrm{T_S}}\right] - \rho}{(\mathrm{T_S}+\mathrm{T_W})\,\lambda^2\,\mathrm{e}^{\lambda \mathrm{T_S}}}, \tag{20}$$

for all but the highest rates, as $\lambda^{-1} \gg \mathrm{T_S}$. Additionally, for the usual hysteresis values $h \gg \{\mathrm{T_S}, \mathrm{T_W}\}$, and applying the fact that, as long as the frame size stays below 1500 bytes, $s < \mathrm{T_S}$ for a 10 Gbit/s interface, we get that B_{f}^* can be approximated as

$$B_{\mathrm{f}}^* \approx h\left(1 + \frac{\lambda^{-1}}{\mathrm{T_S}+\mathrm{T_W}}\right). \tag{21}$$

4.3.2. Approximating Ideal EEE Interface Consumption

Pre-coalescing can also be used to approximate the results of an ideal EEE interface, that is, one that stays in LPI during $1-\rho$ of the total time. Consider δ the maximum deviation permitted from the optimum value. We firstly equate the target consumption to (16):

$$1 - \rho - \delta = \rho_{\mathrm{LPI}}^{\mathrm{hyst}} = (1-\rho)\frac{\lambda^{-1} + B_{\mathrm{i}}^* - h^* - \mathrm{T_S} - \mathrm{T_W}}{\lambda^{-1}(1-\rho) + B_{\mathrm{i}}^*}. \tag{22}$$

Then, we can solve again for B_{i}^*, the minimum B value that guarantees an acceptable approximation to an ideal EEE interface, and get

$$B_{\mathrm{i}}^* = (1-\rho)\frac{\lambda(h^* + \mathrm{T_S} + \mathrm{T_W}) - \rho - \delta}{\delta\lambda}. \tag{23}$$

As B_{i}^* depends on ρ, we obtain the load value that makes B_{i}^* maximum and obtain that

$$\rho_{\mathrm{i}}^* = \frac{\sqrt{\delta L}}{\sqrt{(h^* + \mathrm{T_S} + \mathrm{T_W})\mathrm{C} - L}}, \tag{24}$$

with C the link capacity and L the average frame size. Substituting (24) into (23) we obtain the B value that guaranties a given approximation to the ideal sleeping interval for any load:

$$\begin{aligned} B_{\mathrm{i}}^{\max} = {} & \frac{(h^* + \mathrm{T_S} + \mathrm{T_W})\mathrm{C} + (\delta - 1)L}{\delta \mathrm{C}} \\ & - \frac{2\sqrt{\delta L}\sqrt{(h^* + \mathrm{T_S} + \mathrm{T_W})\mathrm{C} - L}}{\delta \mathrm{C}}. \end{aligned} \tag{25}$$

If we restrict (25) to the usual operating conditions, then we can neglect the time needed for the transmission of a single frame ($s = L/\mathrm{C} \approx 0$). So, assuming that the hysteresis is at the very least an order of magnitude higher than the transitions times, we have

$$\begin{aligned} B_{\mathrm{i}}^{\max} &\approx \frac{h^*\mathrm{C} + (\delta - 1)L}{\delta \mathrm{C}} - 2\sqrt{\frac{Lh^*}{\delta \mathrm{C}} - \frac{L^2}{\delta \mathrm{C}^2}} \\ &\approx \frac{h^*}{\delta} + (\delta - 1)\frac{L}{\mathrm{C}} - \frac{2}{\sqrt{\delta}}\sqrt{\frac{L}{\mathrm{C}}\left(h^* - \frac{L}{\mathrm{C}}\right)} \\ &\approx \frac{h^*}{\delta}. \end{aligned} \tag{26}$$

5. Experimental Results

In this section we analyze the accuracy of the previous models via simulation. We focus on the normalized sleeping time instead of the energy consumption for several reasons. Firstly, it only depends on the LPI governing algorithm and not on the electrical characteristics of a given NIC. While most are similar, there can be variations among different manufactures. Secondly, NIC interfaces provide directly to operating system the time spent in each mode and the number of transitions, but not energy usage estimations. In any case, as it can be seen from (1) and (2), both are interchangeable. For the experiments we have developed a new EEE simulator with configurable hysteresis and delay to mimic real hardware. The simulator is available to download [29]. In all simulated scenarios, traffic arrivals follow either a Poisson or a Pareto process with shape parameter $\alpha = 1.8$ to validate our formulas with self-similar traffic of infinite variance. We employed constant frame sizes of 1500 bytes. Every simulation experiment has been repeated 20 times with different random seeds and 95 % confidence intervals have been calculated. However, considering that the intervals were very small, we have chosen not to represent them to avoid cluttering the figures in excess. Finally, the experiments can be divided into four groups: a first group devoted to assess the accuracy of the delay and hysteresis models, a second one about the bunching model, a third group that shows the optimum configuration of the bunching algorithm for a given target energy expenditure, and the final set of experiments that test the pre-coalescing algorithm in a hardware test-bed.

5.1. Model Validation

In the first set of experiments we validate the energy model for time-based coalescers with hysteresis developed in Section 3.2. Fig. 5 shows the normalized sleeping time defined in (2), as a hardware-independent proxy for the energy usage of an EEE interface, from an extremely low load value to an abnormally high one with a ranging hysteresis time from 0 to 1 ms. In the figure the filled points correspond to the Poisson traffic, the empty ones are the results for the Pareto process and the continuous lines are the model predictions. It comes

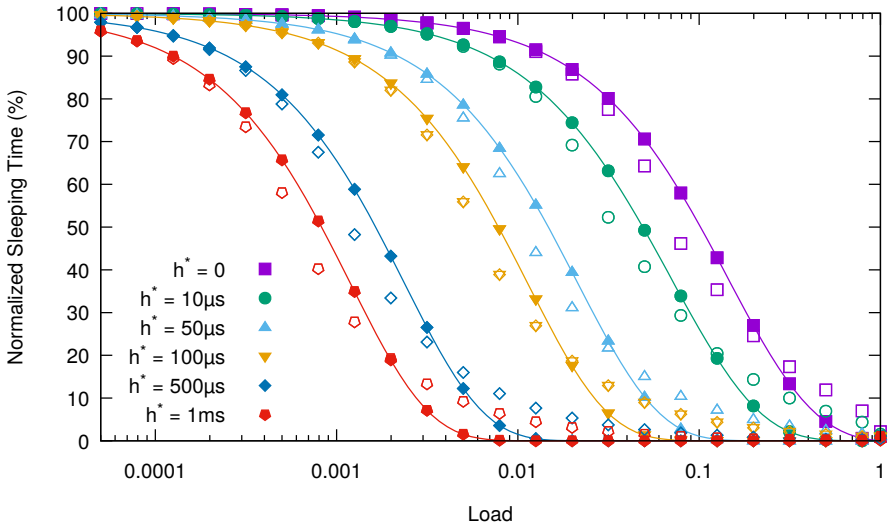
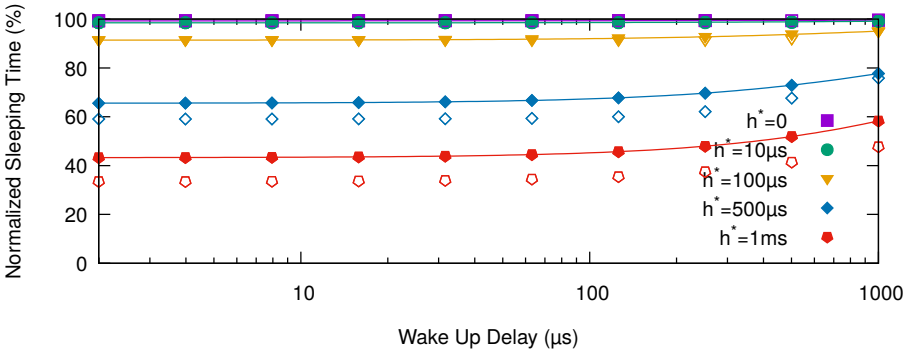


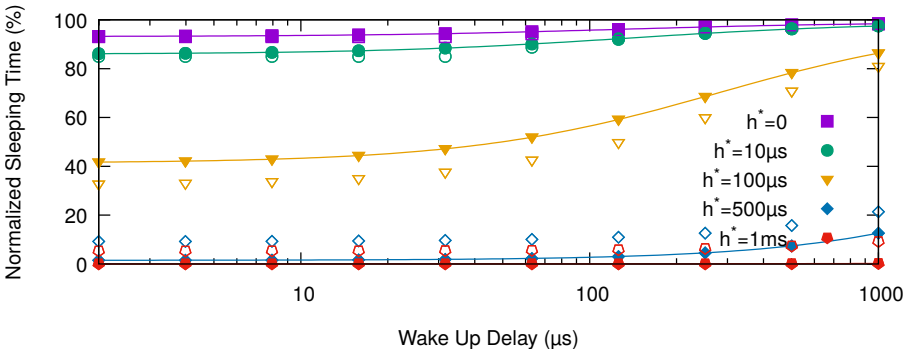
Figure 5: Normalized sleeping time versus load for different hysteresis values. Filled points correspond to Poisson traffic, empty points to Pareto traffic and continuous lines to model predictions.

immediately that the model has enough accuracy and is able to match the simulation values for every hysteresis and load combination. Only the results for the self-similar traffic show minor deviations from the model value. For this kind of traffic, the model overestimates the sleeping time at low loads, and underestimates it at the highest ones. As expected, we can also appreciate that there is an inverse relationship between hysteresis and energy efficiency. From an energy usage point of view it is better to configure the interface with no hysteresis, though not all commercial equipment allows to configure small enough hysteresis values.

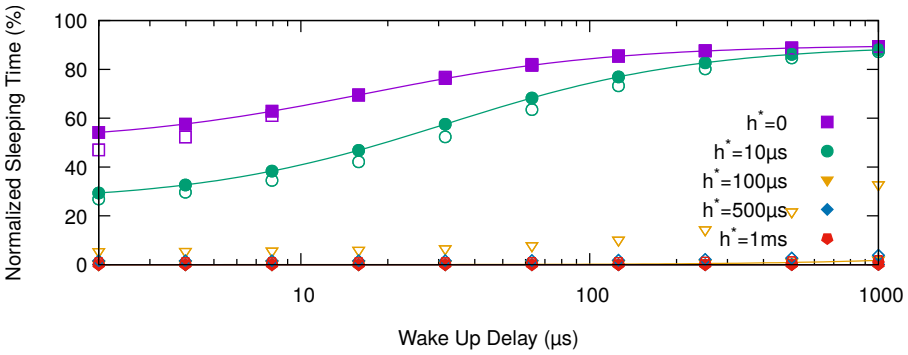
The effect of the wake up delay is represented in Fig. 6. As in the previous experiment, there is a very good match between the model predictions and the observed values. The results for the different traffic loads show a similar trend: sleeping time grows as the wake up delay increases, i.e., the time needed for the interface to return to active mode since the first frame arrival while idle, augments. However, for the wake up delay to take significant effect, it needs to be quite high when compared to the applied hysteresis. For instance, in Fig. 6a, for the $h^* = 500\,\mu s$ and $h^* = 1\,ms$ plots, we only see increasing the sleeping time when the delay surpasses the $200\,\mu s$ value. In Fig. 6b, for the $h^* = 100\,\mu s$ case, the sleeping time only starts to grow rapidly when the delay surpasses the $20\,\mu s$ level. Similar conclusions can be drawn from Fig. 6c for both the no hysteresis and the $h^* = 10\,\mu s$ cases. In every case we see that the hysteresis value has a much greater effect than the delay on energy consumption.



(a)



(b)



(c)

Figure 6: Normalized sleeping time versus wake up delay for different hysteresis values and both synthetic Poisson (filled points) and Pareto (empty points) traffic of loads 0.1 % (a), 1 % (b) and 10 % (c).

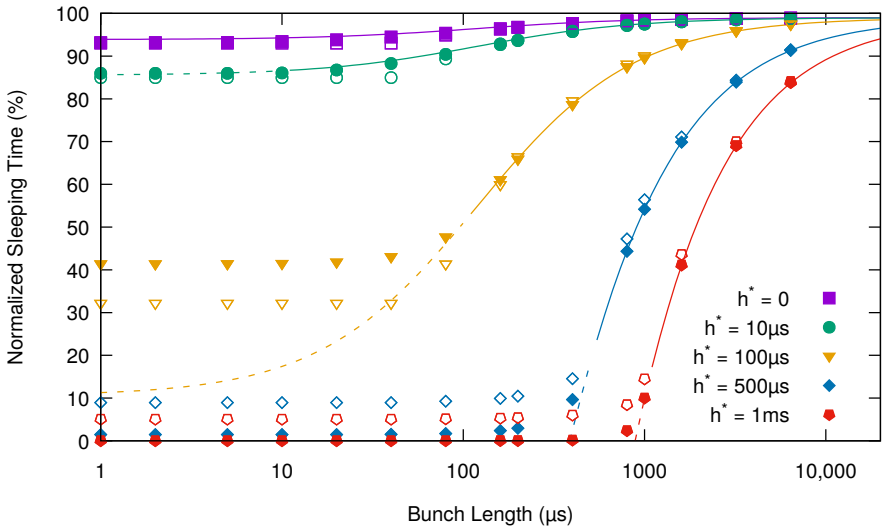


Figure 7: Normalized sleeping time versus bunch length for a 1% load and various hysteresis lengths with Poisson and self-similar traffic. Dashed lines represent values out of the model valid range.

5.2. Pre-coalescing Algorithm Evaluation

The next set of experiments focuses on validating the model for the pre-coalescing algorithm as in (16). To this end, we have fed both Poisson and self-similar Pareto traffic to a frame coalescer that implements our bunching algorithm. The software used has been released with an open-source license, and it is available for download [30].

Fig. 7 compares the model predictions to the results of applying an increasing bunching length to an EEE interface configured with different hysteresis values and 1 % load,[5] albeit without wake up delay. As in the previous experiments, the simulated results are shown with points and the model predictions with lines. To account for the fact that the model is only valid for bunching lengths greater than the interface hysteresis, we have employed dashed lines for those regions where the model is not expected to hold. All in all, the results confirm the accuracy of the model within its valid region. We can also appreciate how the bunching technique improves the energy efficiency of the interface, even if at the cost of increased delay. As a rule of thumb, we see that we need a bunch length, i.e., the maximum delay suffered by a frame, ten times the duration of the hysteresis for significant savings. On the other hand, for low bunching lengths, the model underestimates the obtained savings. This can be seen very clearly in the $h^* = 100\,\mu s$ results.

[5] We have experimented with different load values (from 0.1 % to 50 %), obtaining comparable

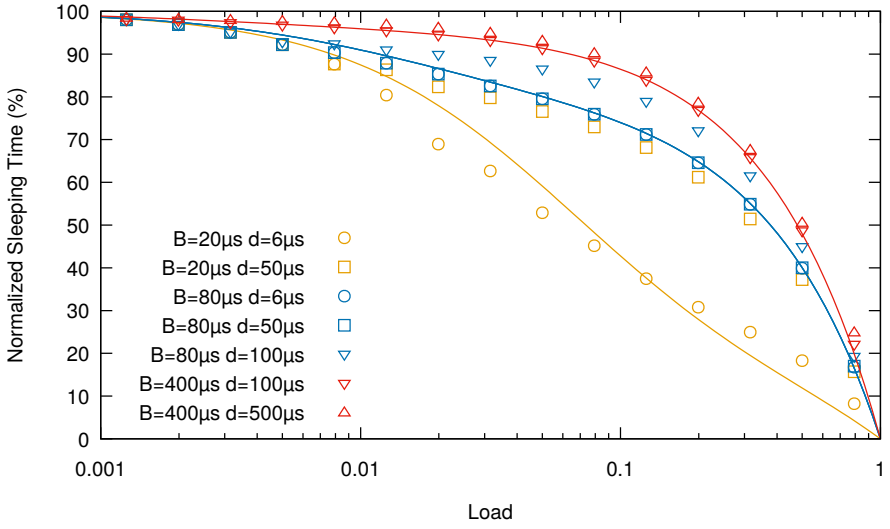


Figure 8: Normalized sleeping time versus load for different bunch length and delay combinations with self-similar traffic. Hysteresis is set to 10 µs. The continuous line represents the model (16).

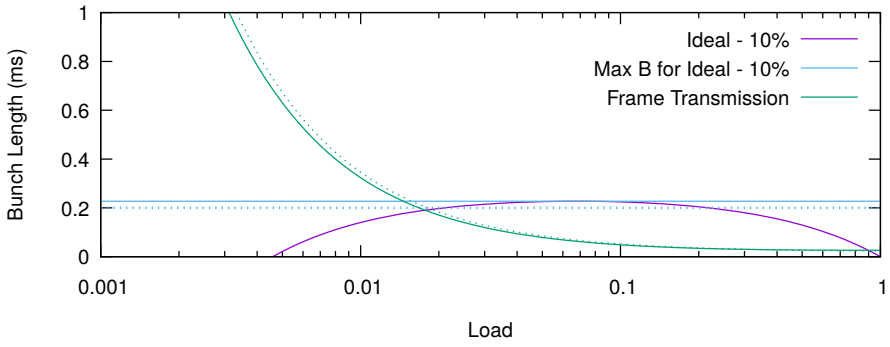
In Fig. 8 we represent the evolution of the sleeping time for different Pareto traffic loads and a set of different combinations of bunching length and wake up delay.[6] The hysteresis is otherwise fixed to a relatively low value of 10 µs so that delay effects are more evident. Recall that for the wake up delay to be noticeable, it has to be greater than the hysteresis, as discussed previously when presenting Fig. 6. It is clear again that the model (16) is a good predictor for the experimental results, at least as long as it stays in its valid region, that is, when the wake up delay is shorter than the bunching length ($d < B$). As expected, the higher the wake up time, the longer the interface stays in the LPI mode. We can also observe that it is not worth to use delay at the network interface if it is not greater than the bunching length.

5.3. Pre-coalescing Approximation to Frame Transmission and Ideal Energy Savings

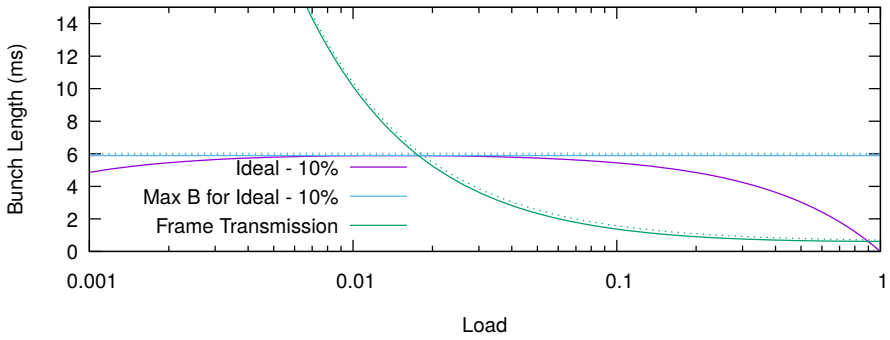
This set of experiments explores the tuning of the bunching length at the pre-coalescer in order for the NIC to obtain results comparable to those of: a) an ideal EEE interface, i.e., one that only stays awake while transmitting traffic and in LPI otherwise, with a 10 % error margin; b) and to those of an interface with no hysteresis nor delay that employs the frame transmission algorithm. We

results. We have omitted them in the paper for the sake of brevity.

[6] We have obtained similar results with Poisson traffic, but they have not been plotted to avoid cluttering the figure in excess.



(a)



(b)

Figure 9: Bunch length needed to obtain a given target consumption for varying Pareto traffic loads and (a) aggressive and (b) non aggressive settings.

consider two different configurations for the interface, with different delay and hysteresis parameters that we have called *aggressive* and *non aggressive* settings. The actual parameters have been selected in accordance with the characteristics of current actual hardware [10, 11] and are the ones shown in Table 2.

Fig. 9 shows the necessary bunching length to obtain energy savings in line with those of frame transmission and those of an ideal interface for varying traffic load conditions. At the same time, it shows the accuracy of the simplifications carried out in Section 4.3. The exact results are shown with continuous lines, while the approximated values with dotted lines. In any case, we can conclude that the approximations provide good enough results. It is also clear from both plots that (25), and consequently (26), permit to configure the bunching length without regards to the actual traffic load.

We can also deduce some additional conclusions. At the highest loads, the necessary bunching length diminishes both for the frame transmission target and for the approximation to the ideal interface. This is because at the highest loads the energy savings are always lowest. This is more pronounced in frame transmission, as it obtains modest savings from moderate loads onward. On the contrary, for the lowest loads it is almost impossible to get the same results as frame transmission does, as it is able to obtain almost ideal energy savings. However, if we use the ideal conditions with a 10 % error margin as a reference, we see that, for the aggressive settings (Fig. 9a), there is no need to employ pre-coalescing. On the contrary, for the conservative settings (Fig. 9b) and the represented loads, bunching is always needed. Obviously, if we further increased the load ranges to include even lower values, we would be able to find loads were bunching would not be needed. Please note, however, that the considered range of loads is already big enough to cover for all usual cases.

We have represented in Fig. 10 the normalized sleeping time obtained when using the approximated bunching lengths of Fig. 9. We can confirm that the calculated bunching lengths are enough to obtain the desired energy savings. In fact, for moderate to high loads, the employed bunching lengths slightly outperform their target value. The same happens for low loads and the ideal settings. In this case it is because no bunching is necessary to get or even improve the target of less than 10 % difference when compared to the ideal algorithm.

In Fig. 11 we show the effects on average frame delay of some of the recommended bunching lengths shown in Fig. 9. We have shown both the average frame delay and its 95 % percentile as empty and filled dots, respectively. The expected average values from the model (18) are represented with continuous lines. Results show that the model is a good predictor for the average value and that the delay is dominated by the pre-coalescing process. It is important to realize that, given that the bunching process dominates the delay, the worst case per-frame delay is very stable and can be approximated by the B parameter.

5.4. Actual Hardware Results

In our last experiment we measured the performance of the pre-coalescing technique with two 10 Gbit/s NICs. In particular we used two Intel X550 controllers [13] connected to an Intel Core i7-3770 computer running Ubuntu

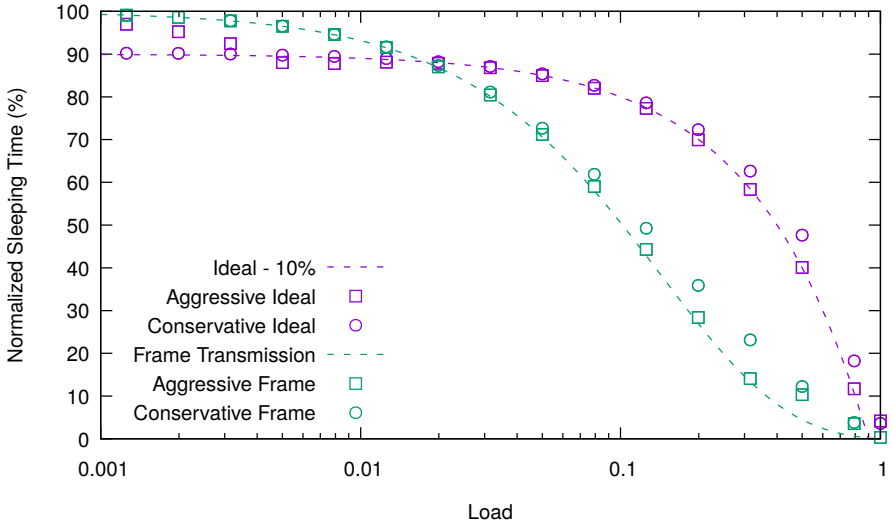


Figure 10: Normalized sleeping time when using the optimum bunch length to get a target consumption.

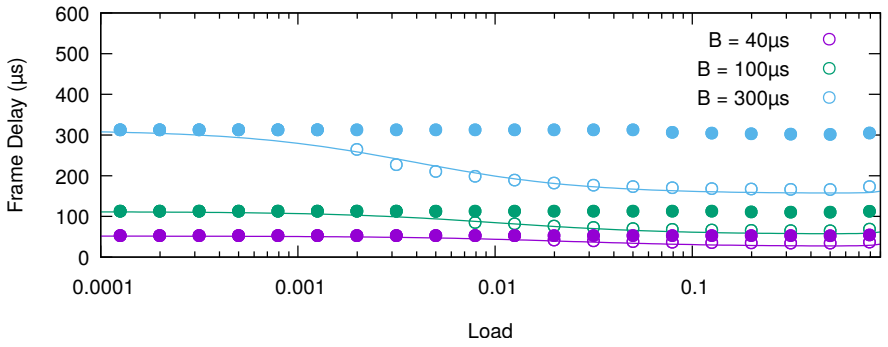
Table 3: Time Spent in LPI Mode on an Intel x550 Interface with $h^* = 20\,\mu s$

	No Pre-coalescing / 200 µs Pre-coalescing					
Rate	LPI events/s		Av. Duration		Time in LPI (%)	
10 Mbit/s	803	804	1.21 ms	1.21 ms	97.8	97.9
100 Mbit/s	7977	3004	98.6 µs	302 µs	78.6	90.6
850 Mbit/s	5334	2664	107 µs	260 µs	57.3	69.1

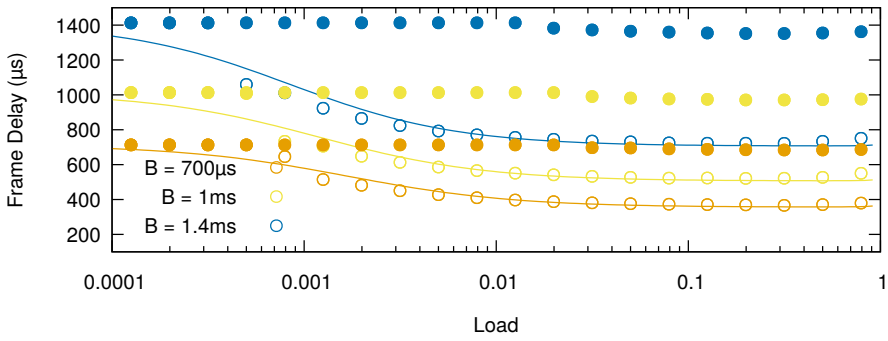
Linux 19.04. Sadly, the hardware does not report the accumulated time spent in LPI, only the number of times the interface enters LPI mode.[7]

We fed the NIC with self-similar traffic traces with packet sizes of 1500 bytes and different average rates. Then we captured the traffic received at the other end. With the number of LPI transitions and the gaps between the frames in the received trace, we have calculated the average time spent in LPI. Although this device allows to completely disable hysteresis, we have used $h^* = 20\,\mu s$ to account for the most advantageous value available from other vendors. A bunching length of 200 µs was chosen according to (26). The results are shown in Table 3. For very low rates, the LPI event rate only depends on the frame

[7] As a matter of fact, the official driver does not yet support EEE [31]. We have made available a simple tool to configure and query EEE mode in x550 NICS at `https://migrax.github.io/eee-X550/`.



(a)



(b)

Figure 11: Average frame delay for different traffic loads when using the pre-coalescing algorithm (empty circles). Filled circles show the 95 % percentile of frame delay: (a) $h^* = 20\,\mu s$ and (b) $h^* = 600\,\mu s$.

interarrival times (about 1.2 ms for 10 Mbit/s). As the rate increases the pre-coalescer starts grouping individual arrivals. For the 100 Mbit/s trace, frames arrive, on average, every 120 µs without pre-coalescing while, with pre-coalescing, a bunch of an average of two frames arrives every 120 µs + 200 µs = 320 µs. This results in almost 2.6 more transitions per second for the non pre-coalescing setup, yielding to lower energy savings due to the time spent in hysteresis and in transitions for each LPI event. Finally, for higher rate traces, several frames are served in the same busy period for both scenarios. This causes the event rate to lose direct dependence on individual frame interarrival times. Results show that, as expected, for a low hysteresis value, no pre-coalescing is necessary for very low average rates. However, as the rate increases, even for modest loads, pre-coalescing is able to significantly increase energy savings.

6. Conclusions

This paper presents a new model for EEE interfaces with hysteresis, and shows that many existing EEE networking devices are unable to obtain significant energy savings with their available configuration. To improve their energy efficiency, the paper proposes to implement frame coalescing before traffic reaches the interface, and provides an analytic model for this technique. This model allows us to derive the optimum coalescing parameters to obtain a desired target energy consumption at the EEE device, when the configuration parameters at the device are known in advance.

The study includes four groups of experiments. The first set demonstrates that the energy model considering hysteresis and delay has enough accuracy and is able to match the simulation values for every relevant hysteresis and load combination. From an energy usage point of view, it is always best to configure the interface with no hysteresis. Furthermore, the wake up delay needs to be quite high to compensate for the negative effects of hysteresis. The second group of experiments focus on validating the model for the bunching algorithm as in (16). The results confirm the accuracy of the model within its valid region (for bunching lengths longer than the interface hysteresis). Besides, we could appreciate how the bunching technique improves the energy efficiency of the interface, albeit at the cost of an increased delay. We can also see that it is not worth to use delay at the interface if it is not greater than the bunching length. The experiments also analyzed the ideal tuning of the bunching length to mimic frame transmission performance or to get close to ideal energy usage. The proper tuning results in a bunch length directly proportional to the hysteresis value, and inversely proportional to the maximum deviation from the ideal energy consumption. This bunch interval has direct consequences on frame delay. For the worst case and for almost any traffic load, the maximum frame delay is very close to the bunch interval. So, for delay-stringent applications, it is important to either employ interfaces with low hysteresis values, or to compromise energy savings allowing greater deviations from the ideal energy consumption. Finally, the fourth set of experiments validated the results on a real hardware test-bed equipped with EEE capable NICs.

The pre-coalescing technique is a valid approach to overcome the very high hysteresis values available at most EEE equipment. It has the additional advantage that it can take into account traffic needs, in contrast with using delay at the interface, which has to be the same for all the transit traffic. In any case, EEE equipment should provide the option to select lower hysteresis values or even no hysteresis at all to the network administrator. This is the best way to get significant energy savings at relevant traffic loads without excessive frame delay.

References

[1] IEEE Std 802.3az-2010 (Amendment to IEEE Std 802.3-2008). pages 1–302, October 2010. doi: 10.1109/IEEESTD.2010.5621025.

[2] Ken Christensen, Pedro Reviriego, Bruce Nordman, Michael Bennett, Mehrgan Mostowfi, and Juan Maestro. IEEE 802.3az: the road to energy efficient Ethernet. *IEEE Communications Magazine*, 48(11):50–56, November 2010. doi: 10.1109/MCOM.2010.5621967.

[3] IEEE Std 802.3bj-2014 (Amendment to IEEE Std 802.3-2012 as amended by IEEE Std 802.3bk-2013). 2014. doi: 10.1109/IEEESTD.2014.6891095.

[4] Pedro Reviriego, José-Alberto Hernández, David Larrabeiti, and Juan Antonio Maestro. Burst Transmission in Energy Efficient Ethernet. *IEEE Internet Computing*, 14(4):50–57, March 2010. doi: 10.1109/MIC.2010.52.

[5] Kyung Jae Kim, Shunfu Jin, Naishuo Tian, and Bong Dae Choi. Mathematical analysis of burst transmission scheme for IEEE 802.3az energy efficient Ethernet. *Performance Evaluation*, 70(5):350–363, May 2013. doi: 10.1016/j.peva.2012.12.001.

[6] Sergio Herrería Alonso, Miguel Rodríguez Pérez, Manuel Fernández Veiga, and Cándido López García. A Power Saving Model for Burst Transmission in Energy-Efficient Ethernet. *IEEE Communications Letters*, 15(5):584–586, 2011. doi: 10.1109/LCOMM.2011.040111.110547.

[7] Xiaodan Pan, Tong Ye, Tony T. Lee, and Weisheng Hu. Power Efficiency and Delay Tradeoff of 10GBase-T Energy Efficient Ethernet Protocol. *IEEE/ACM Transactions on Networking*, 25(5):2773–2787, October 2017. doi: 10.1109/TNET.2017.2703928.

[8] Sergio Herrería Alonso, Miguel Rodríguez Pérez, Manuel Fernández Veiga, and Cándido López García. Dynamic EEE coalescing: Techniques and bounds. *IEEE Transactions on Green Communications and Networking*, 3 (1):205–214, March 2019. doi: 10.1109/TGCN.2019.2893105.

[9] Sergio Herrería Alonso, Miguel Rodríguez Pérez, Manuel Fernández-Veiga, and Cándido López-García. An optimal dynamic sleeping control policy for single base stations in green cellular networks. *Journal of Network and Computer Applications*, 116:86–94, August 2018. doi: 10.1016/j.jnca.2018.05.014.

[10] *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*, 11 2017. URL `https://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5_x/nx-os/system_management/configuration/guide/sm_nx_os_cg.html`.

[11] *Dell EMC Networking N-Series N1100-ON, N1500, N2000, N2100-ON, N3000, N3100-ON, and N4000 Switches CLI Reference Guide*, November 2018. URL `https://downloads.dell.com/manuals/all-products/esuprt_ser_stor_net/esuprt_networking/esuprt_net_fxd_prt_swtchs/networking-n1500-series_administrator-guide_en-us.pdf`.

[12] Yuval Mintz. bnx2x: Added EEE support, June 2012. URL `https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=c8c60d88c59cbb48737732ba948663a3efe882aa`.

[13] *Intel Ethernet Controller X550 Datasheet*, November 2018. 333369-005, version 2.3.

[14] *Intel Ethernet Controller X710/XXV710/XL710 Datasheet*, November 2018. 332464-016, version 3.61.

[15] V Sivaraman, P Reviriego, Z Zhao, A Sánchez-Macián, A Vishwanath, J.A. Maestro, and C Russell. An experimental power profile of Energy Efficient Ethernet switches. *Computer Communications*, 50:110–118, 9 2014. ISSN 01403664. doi: 10.1016/j.comcom.2014.02.019.

[16] David Larrabeiti, Pedro Reviriego Vasallo, J.A. Hernández, J.A. Maestro, and M. Urueña. Towards an energy efficient 10 Gb/s optical Ethernet: Performance analysis and viability. *Optical Switching and Networking*, 8 (3):131 – 138, July 2011. doi: 10.1016/j.osn.2011.03.009. Special Issue on Green Communications and Networking.

[17] Marco Ajmone Marsan, Antonio Fernandez Anta, Vincenzo Mancuso, Balaji Rengarajan, Pedro Reviriego Vasallo, and Gianluca Rizzo. A simple analytical model for energy efficient Ethernet. *IEEE Communications Letters*, 15(7):773–775, July 2011. doi: 10.1109/LCOMM.2011.060111.110973.

[18] Raffaele Bolla, Roberto Bruschi, Alessandro Carrega, Franco Davoli, and Paolo Lago. A closed-form model for the IEEE 802.3az network and power performance. *IEEE Journal on Selected Areas in Communications*, 32(1):16–27, January 2014. doi: 10.1109/JSAC.2014.140103.

[19] Miguel Rodríguez Pérez, Sergio Herrería Alonso, Manuel Fernández Veiga, and Cándido López García. Delay Properties of Energy Efficient Ethernet Networks. *IEEE Communications Letters*, 21(10):2194–2197, October 2017. doi: 10.1109/LCOMM.2017.2726539.

[20] Mehrgan Mostowfi and Ken Christensen. An energy-delay model for a packet coalescer. In *2012 Proceedings of IEEE Southeastcon*, pages 1–6, Orlando, FL, March 2012. IEEE. ISBN 978-1-4673-1375-9. doi: 10.1109/SECon.2012.6196920.

[21] N. Akar. Delay Analysis of Timer-Based Frame Coalescing in Energy Efficient Ethernet. *IEEE Communications Letters*, 17(7):1459–1462, July 2013. doi: 10.1109/LCOMM.2013.051313.130245.

[22] Sergio Herrería Alonso, Miguel Rodríguez Pérez, Manuel Fernández Veiga, and Candido López García. A GI/G/1 Model for 10 Gb/s Energy Efficient Ethernet Links. *IEEE Transactions on Communications*, 60(11):3386–3395, November 2012. doi: 10.1109/TCOMM.2012.081512.120089.

[23] Angelos Chatzipapas and Vincenzo Mancuso. An M/G/1 Model for Gigabit Energy Efficient Ethernet Links With Coalescing and Real-Trace-Based Evaluation. *IEEE/ACM Transactions on Networking*, 24(5):2663–2675, October 2016. doi: 10.1109/TNET.2015.2477090.

[24] Jinli Meng, Fengyuan Ren, and Chuang Lin. Modeling and understanding burst transmission for energy efficient Ethernet. *Computer Networks*, 122: 217–230, July 2017. doi: 10.1016/j.comnet.2017.03.020.

[25] Angelos Chatzipapas and Vincenzo Mancuso. Measurement-based coalescing control for 802.3az. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 270–278, Vienna, Austria, May 2016. IEEE. ISBN 978-3-9018-8283-8. doi: 10.1109/IFIPNetworking.2016.7497210.

[26] Zhisheng Niu, Xueying Guo, Sheng Zhou, and P. R. Kumar. Characterizing Energy–Delay Tradeoff in Hyper-Cellular Networks With Base Station Sleeping Control. *IEEE Journal on Selected Areas in Communications*, 33 (4):641–650, 2015. doi: 10.1109/JSAC.2015.2393494.

[27] Xueying Guo, Zhisheng Niu, Sheng Zhou, and P. R. Kumar. Delay-Constrained Energy-Optimal Base Station Sleeping Control. *IEEE Journal on Selected Areas in Communications*, 34(5):1073–1085, May 2016. doi: 10.1109/JSAC.2016.2520221.

[28] KT Marshall. Bounds for some generalizations of the GI/G/1 queue. *Operations Research*, 16(4):841–848, August 1968. doi: 10.1287/opre.16.4.841.

[29] Miguel Rodríguez Pérez. HystEEE: A Rustified Simulator for 10Gb/s EEE with Configurable Hysteresis, October 2019. URL `https://migrax.github.io/HystEEE/`. version 0.6.0.

[30] Miguel Rodríguez Pérez. EEEBuncherSimul: A Simulator of the Buncher Algorithm for Actual EEE Interfaces, November 2018. URL `https://migrax.github.io/EEEBuncherSimul/`.

[31] *Intel Ethernet Controller X550 Feature Support Matrix*, March 2019. 335253-008, version 1.7.