Investigo
Universidade de Vigo

Biblioteca Universitaria
Área de Repositorio Institucional

Accepted manuscript.

This article has been accepted for publication in *IEEE Transactions on Communications*. The final version of record is available at DOI 10.1109/TCOMM.2017.2700310

Citation for published version:
S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga and C. López-García, "Optimizing Dual-Mode EEE Interfaces: Deep-Sleep is Healthy," in *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3374-3385, Aug. 2017,
doi: 10.1109/TCOMM.2017.2700310

General rights:
© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

Optimizing Dual-Mode EEE Interfaces: Deep-Sleep Is Healthy

Sergio Herrería-Alonso, Miguel Rodríguez-Pérez, *Member, IEEE,*
Manuel Fernández-Veiga, *Senior Member, IEEE,* and Cándido López-García

***Abstract*—The IEEE 802.3bj standard defines two potential low power operating modes for high speed Energy Efficient Ethernet (EEE) physical interfaces working at 40 and 100 Gb/s: a not-so-efficient low power mode that requires very short transition times to restore normal operation (Fast-Wake) and a highly efficient low power mode with longer transition times (Deep-Sleep). In this paper, we present a new frame coalescing mechanism that dynamically adjusts the coalescing queue threshold in order to minimize the energy consumption of dual-mode EEE interfaces and maintains, at the same time, the average frame delay close to a target value. The proposed mechanism has been validated through simulation under different types of traffic (Poisson, self-similar and real Internet traffic). Additionally, we show that, with the current transition times and efficiency profiles of the standardized low power modes, our proposal renders the Fast-Wake mode unnecessary in most practical scenarios.**

***Index Terms*—Energy efficiency, IEEE 802.3bj, Energy Efficient Ethernet, coalescing**

I. Introduction

To reduce energy consumption of Ethernet links, the IEEE published in 2010 the IEEE 802.3az standard [1], known as *Energy Efficient Ethernet* (EEE). This norm provides a new operating mode to be used in Ethernet physical interfaces (PHYs) when there is no data to transmit. When PHYs are in this low power idle (LPI) mode, they only draw a small fraction of the energy needed for normal operations, although they are unable to send traffic through their attached links. Probably, the most straightforward way to manage EEE interfaces consists of entering LPI whenever the transmission buffer becomes empty and restoring normal functioning when there is new traffic to transmit. However, this approach is not very efficient since PHYs consume about the same power during state transitions (to/from the LPI mode) as in the active state and transition times are of the same order as a single frame transmission time. In fact, energy savings can be greatly improved if the number of state transitions is reduced, for example, by making the PHYs wait to accommodate a few frames in the transmission buffer before exiting LPI (*frame coalescing*). EEE has shown to be very effective reducing energy consumption in 100 Mb/s, 1 Gb/s and 10 Gb/s Ethernet links, specially when some coalescing control policy is applied [2], [3], [4].

The problem of relatively long transition times is even more severe in 40 Gb/s and 100 Gb/s Ethernet PHYs since, under these higher rates, transmission times are significantly lower while transition times remain similar. The IEEE 802.3bj amendment deals with these high speed interfaces [5] and defines for them an optional low power mode, known as *Deep-Sleep*, identical to the LPI mode defined for the Ethernet interfaces with lower rates. Thus, PHYs in the Deep-Sleep mode just consume a small portion of the power consumed when active (around 10%) but, since all signaling between the sender and the receiver is stopped, resuming normal operation requires excessively long transition times.

Since high speed Ethernet PHYs are expected to be used in data centers and backbone links operating at moderate loads, IEEE 802.3bj also defines a new mode called *Fast-Wake* which maintains attached links aligned. Although clock synchronization must be maintained in this mode, some of the PHY components at the higher layers can be turned off thus obtaining some limited power savings, but with very short transition times. For example, [6] suggests that consumptions around 70–80% of the peak could be obtained with Fast-Wake. Therefore, it is expected that IEEE 802.3bj PHYs support both Fast-Wake and Deep-Sleep modes to get significant reductions on their energy consumption without causing noticeable effects in frame delay nor additional frame losses.

As with IEEE 802.3az interfaces with just a single low power mode, the standard does not suggest any algorithm to manage the two possible low power modes of dual-mode EEE interfaces. Any mechanism proposed to govern dual-mode EEE interfaces must answer the three following questions:

1) When to enter a low power mode.
2) Which of the two low power modes should be entered.
3) When to exit the low power mode.

Clearly, to maximize power savings, dual-mode EEE interfaces should enter a low power mode every time their transmission buffer gets empty since energy-aware interfaces should not stay active when they have no data to transmit. However, the answer to the last two questions is not obvious and depends, in fact, on the traffic load.

In this paper, we present a new coalescing mechanism to manage the low power modes of dual-mode EEE interfaces able to minimize their energy consumption under the constraint of a target average frame delay. We firstly show how to select the most convenient low power mode according to the configured target delay. Then, since the proposal applies a frame coalescing technique, the interface will remain in the selected low power mode until the transmission buffer reaches a certain threshold. Unfortunately, a single queue threshold value does not suit well for all the possible traffic loads [7], [8], [9], so we also derive an adaptive algorithm to adjust the

The authors are with the Department of Telematics Engineering, University of Vigo, 36310 Vigo, Spain (e-mail: sha@det.uvigo.es).

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

coalescing queue threshold according to the target delay and the existent traffic conditions.

Finally, an unexpected but important finding of this study is that, with the current transition times and efficiency profiles of the standardized low power modes, our proposal renders the Fast-Wake mode unnecessary in most practical scenarios since the best energy savings are often achievable with the Deep-Sleep mode.

The rest of this paper is organized as follows. Section II presents some works in the field that preceded our paper. In Section III we derive some analytic results to find out which is the most convenient low power mode for dual-mode EEE interfaces in a given scenario. Section IV presents an adaptive algorithm able to adjust the coalescing queue threshold according to the configured target delay and the actual traffic conditions. Then, our proposal to govern dual-mode EEE interfaces is presented in Section V. Section VI shows some results obtained through simulation. Finally, the main conclusions of this work are summarized in Section VII.

II. Related Work

A. Coalescing

Nowadays, most communication systems and computing equipment can enter a low power state during idle periods to save energy while some (or all) of their functions remain frozen. Unfortunately, the duration of the transitions between the low power and the active states are not negligible and a significant amount of energy is wasted while executing them. To reduce the overhead of state transitions, multiple individual jobs can be collected, or coalesced, into a single burst of jobs. Clearly, coalescing jobs extends idle periods thus reducing the number of state transitions and, therefore, improving the energy efficiency of the system. However, coalescing also increases the delay of job processing, so there is a trade-off between the energy consumption and the system performance [10], [11].

Three types of coalescing can be defined based on when to exit the low power mode and return to the active state. Time-based coalescers end the coalescing period after the expiration of a predefined timer started upon the arrival of the first job. Size-based coalescers wake up when the number of collected jobs (or the amount of work to do) reaches a predefined threshold. Finally, hybrid coalescers use a combination of both. These are the most interesting coalescers since they allow maintaining both the coalescing queue length and the maximum queueing delay controlled at the same time.

Diverse coalescing schemes have been successfully applied to reduce the energy consumption in many different computing equipment [12], [13], mobile devices [11], [14], [15] and networking technologies including Wi-Fi [16], [17], optical [18], [19] and cellular [20], [21] networks. Next we review the application of coalescing techniques to EEE interfaces.

B. Frame Coalescing in EEE

Energy efficiency of EEE interfaces can be greatly improved using the *frame coalescing* (or *burst transmission*) wake-up strategy [2], [3], [4]. Instead of exiting the LPI mode as soon as there is new traffic to transmit, PHYs using this mechanism remain asleep until the amount of queued data in the transmission buffer reaches a certain threshold (Q_{w}). Certainly, coalescing frames into bursts increases their delay. To avoid excessive delays, the maximum coalescing time, that is, the maximum time an interface can be in the LPI mode since the first frame is buffered for transmission, must be limited ($W_{\max}$).

Several mathematical analyses modeling EEE behavior can be found in the literature. Some models do not consider the effect of coalescing and just assume that PHYs awake as soon as a new frame is ready for transmission [22], [23], [24]. There are also some works that just provide simple models for the energy consumption of EEE interfaces using frame coalescing with Poisson traffic [25], [26]. The most interesting models are those addressing the trade-off between energy consumption and frame delay when using this algorithm in 1000BASE-T [8], [9] and 10GBASE-T links [27], [28], [29], [30], [31].[1] We will build our proposal on the M/G/1 model presented in [28] since high speed EEE interfaces operate similarly to 10GBASE-T interfaces and this model provides very accurate and easy-to-use approximations to the energy savings and the delay that frames suffer due to coalescing.

There also exist some previous works proposing the dynamic adaptation of coalescing parameters to actual traffic conditions. In [7], the authors present a dynamic coalescing algorithm that adjusts the queue threshold Q_{w} to achieve a given predefined energy efficiency ratio without taking any delay constraint into consideration. In contrast, other works propose to adapt the maximum coalescing time $W_{\max}$ to meet a target average delay [9], [32], but, unfortunately, they can only be applied to 1000BASE-T links. In addition, all these dynamic schemes rely on an additional gain parameter controlling the speed of adjustment of the coalescing parameter to the target value. This gain parameter would have important effects on the system performance, so it must be carefully configured to guarantee a good compromise between the system stability and a fast response to changing traffic conditions.

C. Dual-Mode EEE

To the best of our knowledge, only two algorithms had been proposed to make use of the two new low power modes of 802.3bj-2014 interfaces [33], [34]. Both algorithms put dual-mode EEE interfaces to sleep as soon as their transmission buffers get empty. The first proposal [33] assumes that, when there are no more frames to transmit, a dual-mode EEE interface always enters the Fast-Wake mode, after a short transition of length $T_{\mathrm{s}}^{\mathrm{f}}$. Then, the interface remains in Fast-Wake either until a frame arrives or for a maximum period of length T_{idle}. In the former case, the interface would directly return to the active state after a transition of length $T_{\mathrm{w}}^{\mathrm{f}}$ while,

[1] Different Ethernet speeds have different EEE specifications and transition mechanisms. For example, state transitions in 10GBASE-T links can occur in both directions of the link independently while 1000BASE-T links can enter the LPI mode only when there is no traffic in both directions. Additionally, in 1000BASE-T links, transitions from active to the LPI mode are immediately interrupted when a new frame arrives to the interface. In contrast, the interruption of the sleep transition is not supported by 10GBASE-T links.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

in the latter one, it would transition to Deep-Sleep after an additional $T_{\rm s}^{\rm d}$ period and remain in this mode until a new frame arrives. Eventually, a long transition of length $T_{\rm w}^{\rm d}$ will be required to return to the active state.[2]

This simple technique to manage dual-mode interfaces can be easily improved just making the PHYs wait to accommodate a few frames in the transmission buffer before exiting the low power mode.[3] As with EEE interfaces with just a single low power mode, frame coalescing can also be applied in dual-mode interfaces to decrease the number of state transitions and thus reduce their energy consumption. With frame coalescing, interfaces in the Fast-Wake mode would switch to Deep-Sleep as long as less than $Q_{\rm w}^{\rm f}$ frames arrive during the $T_{\rm idle}$ period. Otherwise, they would resume normal operation when the $Q_{\rm w}^{\rm f}$-th frame arrives. Clearly, frame coalescing could also be applied to Deep-Sleep so that interfaces remain in this mode until a total of $Q_{\rm w}^{\rm d}$ frames are buffered for transmission. Obviously, $Q_{\rm w}^{\rm f} \leq Q_{\rm w}^{\rm d}$ to avoid useless transitions to Deep-Sleep.

The two algorithms presented so far use Fast-Wake as an inevitable prior step before reaching the more efficient Deep-Sleep mode. However, as suggested in [34], it would be more effective to enter Deep-Sleep directly if the traffic load is low enough. Note that, under low traffic conditions, the rate of state transitions would be significantly small and, therefore, only a minor amount of energy would be wasted in them. Consequently, dual-mode interfaces could immediately enter Deep-Sleep and take advantage of its tiny consumption without spending too much energy in the transition periods.

Taking this into account, [34] uses frame coalescing in the following manner. Depending on the traffic load, active dual-mode interfaces can transition to Fast-Wake or directly to Deep-Sleep. If Fast-Wake is selected, the interface stays in this mode for a fixed period of length $T_{\rm idle}$. Then, it returns to the active mode to transmit the coalesced frames or transitions to Deep-Sleep if the transmission buffer is still empty. On the other hand, if Deep-Sleep is used, the interface will remain in this mode until $Q_{\rm w}^{\rm d}$ frames are buffered for transmission or the timer $W_{\max}$ expires, whatever happens first. In [34], the $Q_{\rm w}^{\rm d}$ threshold is configured to match the amount of frames that the interface is able to send in a $W_{\max}$ period.

To determine if the traffic load is low enough to go straight to Deep-Sleep, [34] computes the number of coalesced frames in the previous coalescing period and compares it to a given threshold. If the number of coalesced frames is below this threshold, it is considered that the traffic load is low enough and the interface will directly transition to Deep-Sleep. Otherwise, it will enter Fast-Wake. [34] arbitrarily sets this threshold to half the $Q_{\rm w}^{\rm d}$ threshold. Analytical models to evaluate the power consumption and the average frame delay of dual-mode EEE interfaces using this scheme are presented in [35] and [36], respectively.

Unfortunately, all these schemes cannot be configured to maintain the average frame delay around a target value under any possible traffic load. Moreover, they present some implementation issues. As stated in the standard [5], to switch between Fast-Wake and Deep-Sleep modes, it is necessary to exchange LLDP signaling between the attached links before entering the preferred low power mode [37]. For this reason, switching between low power modes cannot be done as fast as required by these schemes. The standard presumes that dual-mode EEE interfaces are intended to use only one of the two low power modes for a significant period of time.

III. Low Power Mode Selection

As shown in [34], the most convenient low power mode for a dual-mode interface depends on the traffic load it receives. Indeed, the slow but efficient Deep-Sleep mode should be used with low loads since just a small amount of transitions will be induced under these traffic conditions. Conversely, if the traffic load is high, the Fast-Wake mode should be preferred to reduce the length of the transition periods and, hence, the energy wasted in them. There must exist, therefore, an arrival rate threshold $\tilde{\lambda}$ that delimits whether these interfaces operate under low or high load conditions. This threshold could be used to decide the best low power mode for the interface according to actual traffic conditions.

To determine the operating conditions of dual-mode EEE interfaces, we will elaborate on the energy model developed in [28] for EEE interfaces with just a single low power mode. Throughout the paper, we will assume that frame arrivals follow a Poisson process with average arrival rate λ. Poisson traffic provides a valid approximation to the traffic load of this kind of high speed Ethernet interfaces since it is expected that they would be mostly used in data centers or backbone links at the Internet core [38]. We also assume that service times follow an arbitrary distribution function with mean service rate μ. Obviously, the utilization factor $\rho = \lambda/\mu$ must be less than 1 to assure system stability. Finally, we assume that the interface has a transmission buffer of infinite capacity.[4]

According to the model presented in [28], the energy consumed on an EEE interface compared with that consumed on a power-unaware interface that is always active is given by

$$\varphi = 1 - (1 - \varphi_{\rm off})(1 - \rho)\frac{\overline{T}_{\rm off}}{\overline{T}_{\rm off} + T_{\rm s} + T_{\rm w}}, \qquad (1)$$

where $\varphi_{\rm off}$ is the portion of the active mode energy consumption demanded when the interface is in the low power mode, $T_{\rm s}$ is the time required to enter the low power mode, $T_{\rm w}$ is the time required to exit the low power mode and restore normal operation and $\overline{T}_{\rm off}$ is the mean time the interface stays in the low power mode in each busy cycle. Since with Poisson traffic all inter-arrival times are identically and exponentially distributed, the arrival time of the $Q_{\rm w}-$th frame is Erlang$-Q_{\rm w}$

[2]Throughout the paper, the superscript f (d) will be used to refer those parameters related to the Fast-Wake (Deep-Sleep) mode.

[3]Throughout the paper, the queue thresholds will be specified in frames for simplicity. However, in a real setting, they should be specified in bytes to handle frames of different sizes.

[4]Note that, if the queue threshold is chosen carefully, frame losses will be negligible. For example, an interface transitioning to awake from Deep-Sleep will only receive, in average, $\lambda T_{\rm w}^{\rm d}$ more frames before it is active again. Therefore, the buffer should be correctly dimensioned to accommodate, at least, $Q_{\rm w}^{\rm d} + \mu T_{\rm w}^{\rm d}$ frames plus a safety margin.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

TABLE I
EEE PHYSICAL CHARACTERISTICS OF 40 GB/S DUAL-MODE INTERFACES.

Parameter	Fast-Wake	Deep-Sleep
T_s (μs)	0.18	0.72 (from FW), 0.9 (from active)
T_w (μs)	0.34	5.5
φ_off	0.7	0.1

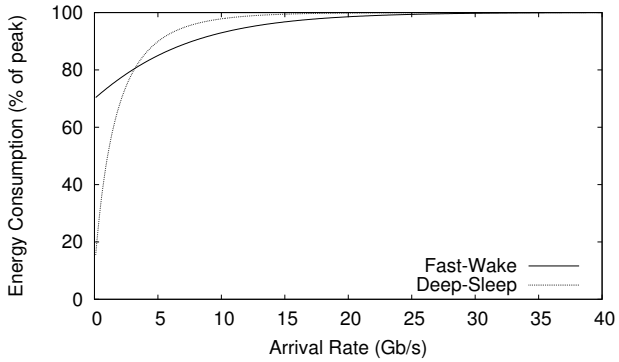
distributed and, therefore, according to [28], $\overline{T}_\mathrm{off}$ when using frame coalescing can be calculated as

$$\overline{T}_\mathrm{off} = \frac{\Gamma(Q_\mathrm{w}+1, \lambda T_\mathrm{s}) - \lambda T_\mathrm{s}\Gamma(Q_\mathrm{w}, \lambda T_\mathrm{s})}{\lambda \Gamma(Q_\mathrm{w})}, \tag{2}$$

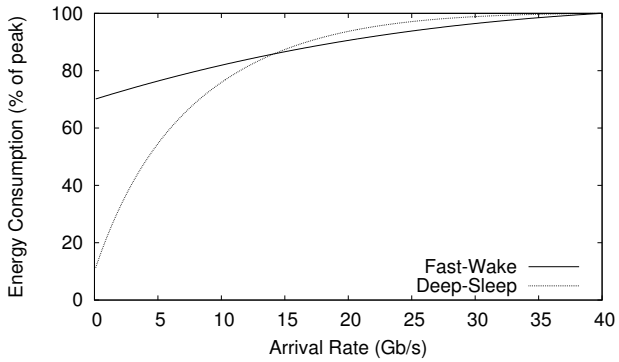
where $\Gamma(q,x) = \int_x^\infty t^{q-1}\mathrm{e}^{-t}\,\mathrm{d}t$ is the upper incomplete gamma function and $\Gamma(q) = \Gamma(q,0)$.

We now use (1) and (2) to compute the expected energy consumption on a 40 Gb/s dual-mode EEE interface that only enters the Fast-Wake mode or the Deep-Sleep mode. We considered the EEE parameters shown in Table I as revealed in the IEEE 802.3bj-2014 standard and also used in previous works [33], [34]. The standard does not provide exact values for φ_off. Although it will depend on the concrete PHY, it is expected that the Deep-Sleep mode just needs around one tenth of the power under normal operation (the same as the low power mode in conventional EEE [39]), while Fast-Wake would draw an amount of energy close to 70% of the peak consumption using a best-case estimation as suggested in [6].

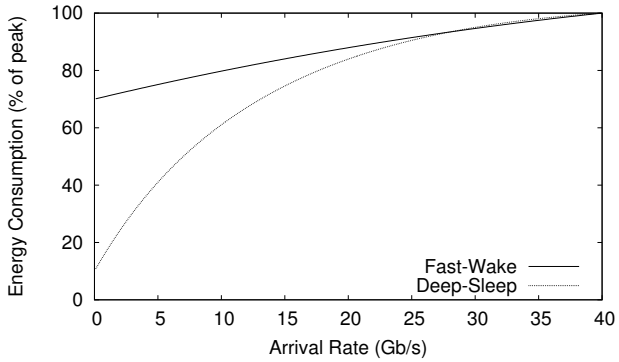
Figure 1 shows the energy consumed with three different coalescing configurations. As shown in Fig. 1(a), if no coalescing is applied ($Q_\mathrm{w} = 1$ frame) and the arrival rate does not exceed 3 Gb/s approximately, the interface consumes less energy using Deep-Sleep than using Fast-Wake since, under these conditions, the number of transitions is very low and the interface stays in the low power mode most of the time. In contrast, for higher arrival rates, the Fast-Wake mode obtains greater energy savings since the amount of time in the low power mode is significantly reduced and no longer compensates for the long Deep-Sleep transition periods. Figures 1(b) and 1(c) show the energy consumed when some coalescing is applied. Note that, the higher the queue threshold is, the wider the range of arrival rates in which the Deep-Sleep mode consumes less energy. There must exist, therefore, a queue threshold value from which the Deep-Sleep mode is always more efficient than Fast-Wake for all the possible arrival rates. We will derive this value analytically in the following subsection.



(a) No coalescing ($Q_\mathrm{w} = 1$ frame).



(b) Frame coalescing with $Q_\mathrm{w} = 5$ frames.



(c) Frame coalescing with $Q_\mathrm{w} = 10$ frames.

Fig. 1. Energy consumption in a 40 Gb/s dual-mode EEE interface that only enters the Fast-Wake mode or the Deep-Sleep mode.

A. Arrival Rate Threshold

In this subsection, we compute the arrival rate threshold $\tilde{\lambda}$ that determines the minimum rate from which the interface should use Fast-Wake. Note that, for such an arrival rate, the interface would consume the same amount of energy with both low power modes, that is, $\varphi^\mathrm{d} = \varphi^\mathrm{f}$, and, therefore, from (1)

$$(1-\varphi^\mathrm{d}_\mathrm{off})\frac{\overline{T}^\mathrm{d}_\mathrm{off}}{\overline{T}^\mathrm{d}_\mathrm{off} + T^\mathrm{d}_\mathrm{s} + T^\mathrm{d}_\mathrm{w}} = (1-\varphi^\mathrm{f}_\mathrm{off})\frac{\overline{T}^\mathrm{f}_\mathrm{off}}{\overline{T}^\mathrm{f}_\mathrm{off} + T^\mathrm{f}_\mathrm{s} + T^\mathrm{f}_\mathrm{w}}. \tag{3}$$

In addition, assuming that $\lambda T_\mathrm{s} \ll Q_\mathrm{w}$,[5] then $\Gamma(Q_\mathrm{w}, \lambda T_\mathrm{s}) \approx \Gamma(Q_\mathrm{w})$ and $\overline{T}_\mathrm{off}$, the mean time in the low power mode computed in (2), can be well approximated as $Q_\mathrm{w}/\lambda - T_\mathrm{s}$. Substituting this value into (3), we have

$$(1-\varphi^\mathrm{d}_\mathrm{off})\frac{Q_\mathrm{w} - \tilde{\lambda}T^\mathrm{d}_\mathrm{s}}{Q_\mathrm{w} + \tilde{\lambda}T^\mathrm{d}_\mathrm{w}} = (1-\varphi^\mathrm{f}_\mathrm{off})\frac{Q_\mathrm{w} - \tilde{\lambda}T^\mathrm{f}_\mathrm{s}}{Q_\mathrm{w} + \tilde{\lambda}T^\mathrm{f}_\mathrm{w}}, \tag{4}$$

and solving for $\tilde{\lambda}$, we get

$$\tilde{\lambda} = \frac{\sqrt{b^2 - 4a(1-c)} - b}{2a} Q_\mathrm{w}, \tag{5}$$

[5]This assumption is quite realistic considering the expected EEE parameters of dual-mode interfaces. Note that, for the worst-case scenario, that is, $\lambda = 40$ Gb/s and $T_\mathrm{s} = 0.9\,\mu$s, Q_w should be much greater than 4500 bytes (that is, just three 1500-byte frames) what is usual in reasonable coalescing configurations as we will show in the following sections.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

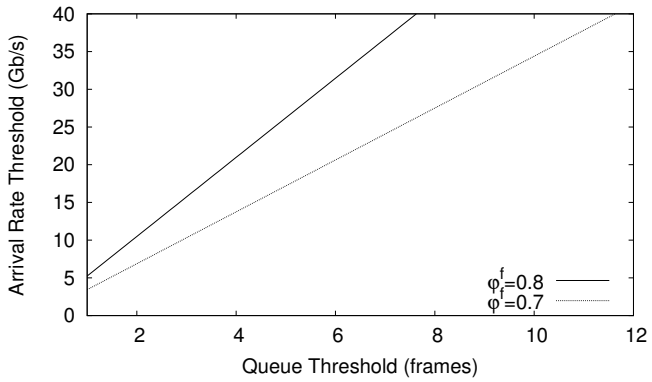


Fig. 2. Arrival rate thresholds for a 40 Gb/s dual-mode EEE interface.

where $a = cT_{\rm s}^{\rm d}T_{\rm w}^{\rm f} - T_{\rm s}^{\rm f}T_{\rm w}^{\rm d}$, $b = T_{\rm w}^{\rm d} - T_{\rm s}^{\rm f} + c(T_{\rm s}^{\rm d} - T_{\rm w}^{\rm f})$ and $c = (1-\varphi_{\rm off}^{\rm d})/(1-\varphi_{\rm off}^{\rm f})$.

Figure 2 shows the arrival rate thresholds obtained using (5) for a 40 Gb/s dual-mode interface with the EEE parameters shown in Table I and different coalescing queue thresholds. Clearly, the relationship between the arrival rate threshold and the queue threshold is proportional linear and the greater the queue threshold is, the wider the range of arrival rates in which the Deep-Sleep mode is preferred. Additionally, as suggested in the previous section, it can be seen that a small queue threshold of just twelve 1500-byte frames is enough to always make Deep-Sleep the preferred mode under any traffic load. The queue threshold $\tilde{Q}_{\rm w}$ that always makes Deep-Sleep the preferred mode regardless of traffic conditions can be easily derived just solving (5) for $Q_{\rm w}$ and fixing $\tilde{\lambda} = \mu$, that is,

$$\tilde{Q}_{\rm w} = \frac{2a\mu}{\sqrt{b^2 - 4a(1-c)} - b}. \tag{6}$$

Particularizing (6) for a 40 Gb/s dual-mode interface with the usual EEE parameters and considering 1500-byte frames, we get $\tilde{Q}_{\rm w} = 11.63$ frames if $\varphi_{\rm off}^{\rm f} = 0.7$ (or $\tilde{Q}_{\rm w} = 7.63$ frames if $\varphi_{\rm off}^{\rm f} = 0.8$) as shown in Fig. 2.

IV. Adaptive Frame Coalescing

As shown in previous works [7], [8], a good tuning of the queue threshold is key for the performance of the coalescing algorithm. If the queue threshold is too high, frames can get excessively delayed. On the contrary, setting a too low threshold reduces the power savings. An additional problem is that a single threshold value cannot perform well for any possible incoming traffic. As shown later, when the traffic load is low, increasing the threshold, even from modest values, produces unacceptable large increments on frame delay with only marginal increments on power savings. Under these circumstances, a low threshold is desirable, as it provides small latencies with satisfying energy savings. For high traffic loads, the situation is just reversed. If the threshold were not increased, power savings would be greatly diminished.

In this section we present a method to dynamically accommodate the $Q_{\rm w}$ parameter to the incoming traffic. Our aim is to minimize power consumption while trying to maintain the average frame delay $\overline{W}$ close to a given target value. As shown in [28], with Poisson traffic, the average frame delay when using frame coalescing in an EEE interface is given by the formula

$$\overline{W} = \frac{1+\lambda^2\sigma_S^2 + (1-\rho)^2}{2\lambda(1-\rho)} - \frac{Q_{\rm w}-1}{\lambda Q_{\rm w}} + \frac{Q_{\rm w}-3+(Q_{\rm w}+\lambda T_{\rm w}-1)^2}{2\lambda(Q_{\rm w}+\lambda T_{\rm w})}, \tag{7}$$

where σ_S^2 is the variance of the service time. As expected, the average delay depends on the selected queue threshold, so we could adjust the frame delay just adapting the queue threshold to the existent traffic conditions. Thus, from (7), the queue threshold $Q_{\rm w}^*$ required to reach a given target delay W^* must hold the following condition

$$Q_{\rm w}^{*3} + (2\lambda T_{\rm w} - 2\lambda k - 3)Q_{\rm w}^{*2} + (\lambda^2 T_{\rm w}^2 - 2\lambda^2 T_{\rm w} k - 4\lambda T_{\rm w})Q_{\rm w}^* + 2\lambda T_{\rm w} = 0, \tag{8}$$

where $k = W^* - \dfrac{1+\lambda^2\sigma_S^2 + (1-\rho)^2}{2\lambda(1-\rho)}$.

Certainly, this cubic equation can be solved using one of the several algebraic methods known to compute its roots. However, for low traffic loads, that is, $\rho \ll 1$ and $\lambda T_{\rm w} \ll Q_{\rm w}$, the average frame delay in (7) can be approximated as

$$\overline{W} \approx \frac{T_{\rm w}}{2} + \frac{Q_{\rm w}-1}{2\lambda}, \tag{9}$$

and, therefore, the queue threshold $Q_{\rm w}^*$ required to achieve the target delay W^* can be simply computed as

$$Q_{\rm w}^* = (2W^* - T_{\rm w})\lambda + 1. \tag{10}$$

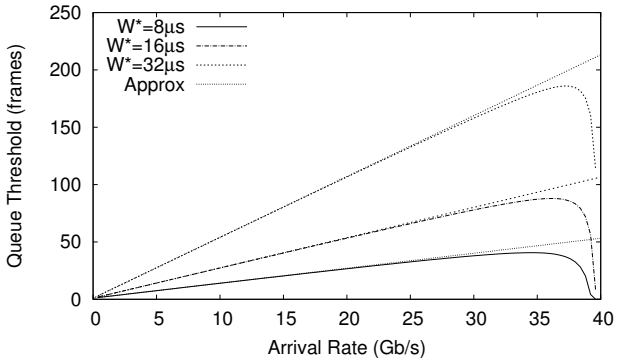
Figure 3 shows the queue thresholds required to reach different target delays for all the range of possible arrival rates considering equally sized 1500-byte frames. Using the awakening times shown in Table I, (8) has only one valid real root, that is, a root greater than one, for all the traffic loads. Additionally, the queue thresholds obtained with the approximation (10) are also depicted in the graphs. Note that we get very accurate values with this approximation except at the highest (and extremely unlikely) rates.[6] As expected, for a given target delay, higher traffic loads require greater queue thresholds. In addition, greater queue thresholds are demanded with increasing target delays. Also note that the Deep-Sleep mode involves slightly smaller queue thresholds than the Fast-Wake mode since Deep-Sleep entails a longer awakening time.

Finally, it must be taken into account that, due to the inevitable awakening times required to exit the low power modes, not all the target delays can be reached. In fact, those requiring queue thresholds smaller than one frame are not actually achievable. So, from (10), it can be easily seen that a target delay cannot be achieved if $2W^* - T_{\rm w} < 0$, that is, $W^* < T_{\rm w}/2$. On the other hand, if the queue threshold is close to one frame, then (9) cannot be used for very low arrival rates because its second term becomes indeterminate. Note that this is the case when (10) is used to configure the queue threshold

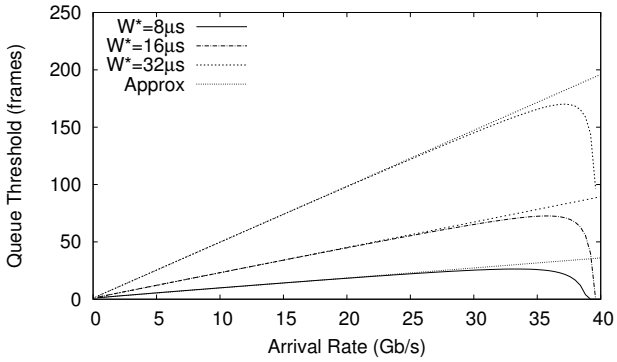
[6]The approximation is really accurate except for those load factors greater than 90%. Fortunately, such high traffic loads are extremely unlikely in an operative Ethernet interface. In any case, the energy-saving algorithm could be temporarily suspended if the traffic load reaches such a large value.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310



(a) Fast-Wake.



(b) Deep-Sleep.

Fig. 3. Queue thresholds required to achieve different target delays.

and the arrival rate is low enough for the selected target delay ($Q_{\mathrm{w}}^{*} \approx 1$). Nevertheless, it can be proved that, in this scenario, the average frame delay tends to T_{w} just computing the limit of the expression (7) for $Q_{\mathrm{w}} = 1$ when $\lambda \to 0$. Note that, if $Q_{\mathrm{w}} = 1$ and the inter-arrival time between consecutive frames is sufficiently long, a frame arrival at the sleeping interface will awake it and, T_{w} seconds later, the interface will transmit the arriving frame and enter the low power mode again. If $W^{*} \geq T_{\mathrm{w}}$, this issue is not really a problem since the delay constraint will be fulfilled anyway but, otherwise, it must be taken into account if the EEE interface is expected to work at such low loads.

V. Adaptive Coalescing for Dual-Mode EEE

In the previous section, we showed that the average frame delay can be kept around a target value if the queue threshold is properly adapted to actual traffic conditions using (10). On the other hand, in Section III-A, we computed the arrival rate threshold that determines the less consuming low power mode for a dual-mode interface as a function of its EEE parameters and the queue threshold. As shown in Fig. 2, even with small queue thresholds, Deep-Sleep is the preferred low power mode in most scenarios, so it is very likely that this keeps being the case when using our Q_{w} adjustment method under normal conditions. In particular, from (5) and (10), it follows that Deep-Sleep will be the most efficient low power mode if

$$\lambda < \frac{\sqrt{b^2 - 4a(1-c)} - b}{2a} \left((2W^{*} - T_{\mathrm{w}}^{\mathrm{d}})\lambda + 1 \right), \qquad (11)$$

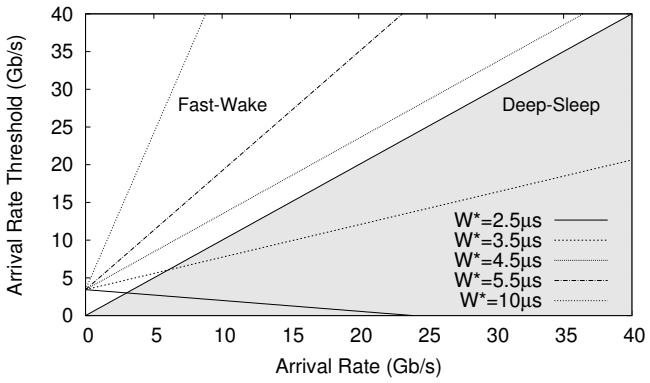


Fig. 4. Arrival rate thresholds when using the proposed Q_{w} adjustment method for different target delays.

what is true for all $\lambda < \mu$ when

$$W^{*} > \frac{T_{\mathrm{w}}^{\mathrm{d}}}{2} + \frac{a}{\sqrt{b^2 - 4a(1-c)} - b} - \frac{1}{2\mu} = \widetilde{W}. \qquad (12)$$

Therefore, dual-mode interfaces configured with a target delay greater than $\widetilde{W}$ should always opt for the Deep-Sleep mode under any traffic load. Figure 4 shows the arrival rate thresholds obtained using the EEE parameters shown in Table I for different target delays. With these parameters, $\widetilde{W}$ is $4.35\,\mu$s. As expected, for those target delays greater than $4.35\,\mu$s, the arrival rate is always lower than the arrival rate threshold, so Deep-Sleep will be the preferred low power mode in all cases. On the contrary, for those target delays lower than $\widetilde{W}$, there exists an arrival rate value (the point of intersection with the diagonal) from which the interface should use the Fast-Wake mode. This arrival rate threshold can be easily calculated just solving (11) for λ:

$$\widetilde{\lambda} = \left(T_{\mathrm{w}}^{\mathrm{d}} - 2W^{*} + \frac{2a}{\sqrt{b^2 - 4a(1-c)} - b} \right)^{-1}. \qquad (13)$$

A. Ideal Algorithm to Manage Dual-Mode EEE

As a consequence of our analysis, we propose the following ideal Algorithm 1 to manage dual-mode EEE interfaces. This algorithm uses the well-known frame coalescing technique to reduce energy consumption but now the coalescing queue threshold is dynamically adjusted according to existent traffic conditions when the transmission buffer gets empty, just before entering the low power mode. Our proposal only requires configuring one straightforward parameter: the desired average frame delay, W^{*}. Recall from Section IV that only W^{*} values greater than $T_{\mathrm{w}}^{\mathrm{f}}/2$ are achievable. In addition, W^{*} values below $T_{\mathrm{w}}^{\mathrm{d}}/2$ can only be obtained using the Fast-Wake mode. In contrast, if the selected W^{*} is greater than $\widetilde{W}$, the interface should always enter Deep-Sleep and adjust the queue threshold as suggested in (10) to achieve the desired target delay. Finally, only for those W^{*} in between, the interface should compute the arrival rate threshold using (13) to select the most efficient low power mode under the actual traffic conditions.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

Algorithm 1 Ideal algorithm to manage dual-mode EEE (executed when the transmission buffer gets empty).

if $W^* < T_{\mathrm{w}}^{\mathrm{f}}/2$ **then**
  Remain active
else if $W^* < T_{\mathrm{w}}^{\mathrm{d}}/2$ **OR** $(W^* < \widetilde{W}$ **AND** $\lambda > \tilde{\lambda})$ **then**
  Update $Q_{\mathrm{w}} \leftarrow (2W^* - T_{\mathrm{w}}^{\mathrm{f}})\lambda + 1$
  Enter Fast-Wake
else
  Update $Q_{\mathrm{w}} \leftarrow (2W^* - T_{\mathrm{w}}^{\mathrm{d}})\lambda + 1$
  Enter Deep-Sleep
end if

B. Avoiding Switching Between the Low Power Modes

Using the EEE parameters shown in Table I, the former algorithm turns into the following simple rules to choose the most convenient low power mode according to the configured target delay:

- Target delays lower than $T_{\mathrm{w}}^{\mathrm{f}}/2 = 0.17\,\mu$s cannot be achieved.
- Fast-Wake must always be used when the target delay is in the range $[T_{\mathrm{w}}^{\mathrm{f}}/2, T_{\mathrm{w}}^{\mathrm{d}}/2] = [0.17, 2.75]\,\mu$s.
- Deep-Sleep must always be used for target delays greater than $\widetilde{W} = 4.35\,\mu$s.
- Finally, if the target delay is in the range $[T_{\mathrm{w}}^{\mathrm{d}}/2, \widetilde{W}] = [2.75, 4.35]\,\mu$s, the preferred low power mode depends on the actual traffic load.

As previously stated in Section II, dual-mode EEE interfaces should use just one of the two possible low power modes for a significant period of time due to the requirement of LLDP signaling to switch between them. Note that, with our proposal, the selected low power mode only depends, except for those target delays in the small range $[2.75, 4.35]\,\mu$s, on the configured target delay. Since this parameter would be configured by the manufacturer or by the administrator of the interface sporadically, the selected low power mode should remain the same for long in most scenarios.

To completely avoid unpleasant low power mode changes, we must make a decision about the preferred mode within the range $[T_{\mathrm{w}}^{\mathrm{d}}/2, \widetilde{W}]$. Figure 5 shows the arrival rate thresholds obtained using (13) for target delays in the range $[2.75, 4.35]\,\mu$s. It can be seen that, for those loads greater than 10 Gb/s, the arrival rate threshold is lower than the traffic load in almost the entire range of target delays. Therefore, we propose to employ the Fast-Wake mode regardless of the actual arrival rate for all the target delays in this range thus avoiding undesired changes in the low power mode in case of bursty traffic. Note that this is the right selection except for low load scenarios with target delays slightly greater than $T_{\mathrm{w}}^{\mathrm{d}}/2$.

C. Practical Algorithm to Manage Dual-Mode EEE

Since the energy-saving algorithm that manages the low power modes is out of the scope of the EEE standard, its logical implementation pertains to the MAC control layer, usually defined in software as part of the firmware of the interface. The conditions upon which the low power modes are activated and suspended are thus controlled by software/firmware within the hardware-specific network driver.

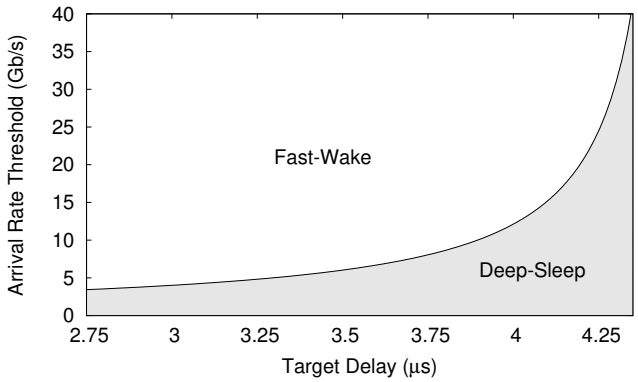


Fig. 5. Arrival rate thresholds when using adaptive coalescing with target delays in the range $[2.75, 4.35]\,\mu$s.

The MAC layer and the PHY unit of Ethernet interfaces communicate through a standardized interface named Medium Independent Interface (MII). The MII interface includes a special signal, known as the LPI signal, which enables the MAC layer to instruct the PHY unit for entering into a sleep mode [1]. Additionally, the standard also defines the state variable LPI_FW to control the low power mode to be used [5]. This Boolean variable is set to true when the interface is to use the Fast-Wake mode, and false when the interface is to use the Deep-Sleep mode. Therefore, the PHY will operate in the low power mode indicated by the LPI_FW variable while the MAC layer maintains the LPI signal active on the MII interface.

Taking into account all these practical considerations, we now propose Algorithm 2 to manage dual-mode EEE interfaces. The LPI_FW variable defaults true and should only be set to false if the optional Deep-Sleep mode is supported and the configured target delay W^* exceeds $\widetilde{W}$. Then, each time the transmission buffer gets empty, the MAC layer just has to update the queue threshold according to current traffic conditions using (10) and raise the LPI signal to instruct the PHY to enter the corresponding low power mode as indicated by the LPI_FW variable. In the opposite direction, when the $W_{\max}$ timer expires or the queue threshold is reached, whatever happens first, the MAC layer deactivates the LPI signal so that the PHY exits the low power state and resumes normal operation. Recall that the $W_{\max}$ timer must be started when the first frame arrives at the sleeping PHY.

To apply the proposed algorithm the MAC layer only needs to measure the arrival rate. The average arrival rate $\widehat{\lambda}$ can be directly estimated at the end of each coalescing cycle, that is, when the transmission buffer gets empty, simply dividing the number of frames received during the current coalescing cycle by its duration. The estimation of the arrival rate and the computation of the next queue threshold only require a few lines of code in the interface firmware, and hardly increase the computational complexity.

Although we have particularized our analysis for 40 Gb/s interfaces, this algorithm can also be applied without changes

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

Algorithm 2 Practical algorithm to manage dual-mode EEE.

A) *At initialization (or after a target delay update)*:
LPI_FW $\leftarrow$ TRUE (Fast-Wake)
if $W^* \geq \widetilde{W}$ **then**
  LPI_FW $\leftarrow$ FALSE (Deep-Sleep)
end if
B) *Each time the transmission buffer gets empty*:
if LPI_FW $==$ TRUE **then**
  Update $Q_{\mathrm{w}} \leftarrow (2W^* - T_{\mathrm{w}}^{\mathrm{f}})\widehat{\lambda} + 1$
else
  Update $Q_{\mathrm{w}} \leftarrow (2W^* - T_{\mathrm{w}}^{\mathrm{d}})\widehat{\lambda} + 1$
end if
Activate LPI signal
C) *When the $W_{\max}$ timer expires or the Q_{w} threshold is reached*:
Deactivate LPI signal

to 100 Gb/s PHYs since they operate in the same way and their EEE parameters are identical to those shown in Table I as stated in the standard [5]. The only difference is that, for 100 Gb/s interfaces, the delay threshold is slightly higher ($\widetilde{W} = 4.44\,\mu$s).

VI. Evaluation

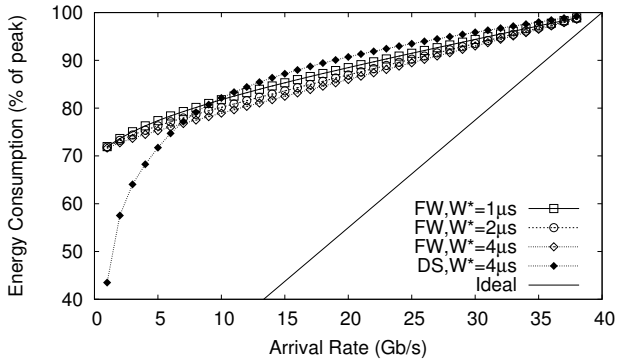
To evaluate the power savings that can be obtained in dual-mode EEE interfaces, we have conducted several experiments on an open-source in-house simulator, available for download at [40]. We simulated a 40 Gb/s interface receiving Poisson traffic with an average arrival rate varying from 1 to 38 Gb/s. The frame size was set to 1500 bytes. Regarding the PHY features, we set the transition times and the efficiency profiles of the low power modes to those shown in Table I.

Each simulation experiment was run for ten seconds and repeated ten times using different random seeds. The average and the 95 % confidence intervals (CIs) of every performance measure have been calculated but CIs are not shown in the graphs since all of them are small and just clutter the figures.

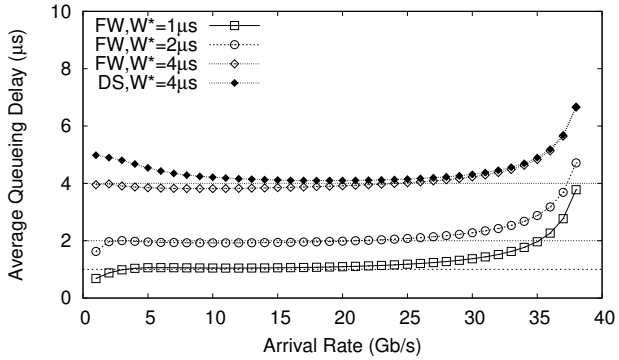
A. Adaptive Coalescing with Poisson Traffic

Firstly, we conducted several simulation experiments to validate Algorithm 2 with different target delays from 1 to 32 μs. We set $W_{\max} = 2W^*$ in all the simulated scenarios. Figures 6(a) and 6(b) show, respectively, the energy consumption and the average queueing delay obtained with those target delays below $\widetilde{W} = 4.35\,\mu$s, that is, those that, according to Algorithm 2, lend to use the Fast-Wake mode ($W^* = 1$, 2 and 4 μs). Only the results obtained using Deep-Sleep with a target delay of 4 μs are also shown for comparison. Recall that target delays below $T_{\mathrm{w}}^{\mathrm{d}}/2 = 2.75\,\mu$s cannot be achieved using this sleep mode.[7] As expected, our proposal gets significant energy savings while maintaining the average queueing delay around the target value at the same time. Obviously, energy savings are slightly higher with increasing target delay. The

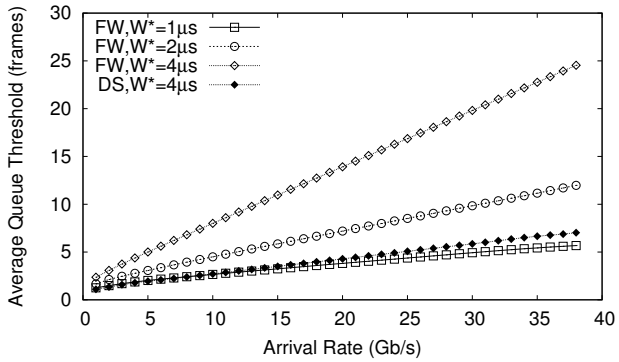
[7]The results using Fast-Wake are always shown with unfilled points while filled points are used to show the results with Deep-Sleep.



(a) Energy savings.



(b) Average queueing delay.



(c) Average queue threshold.

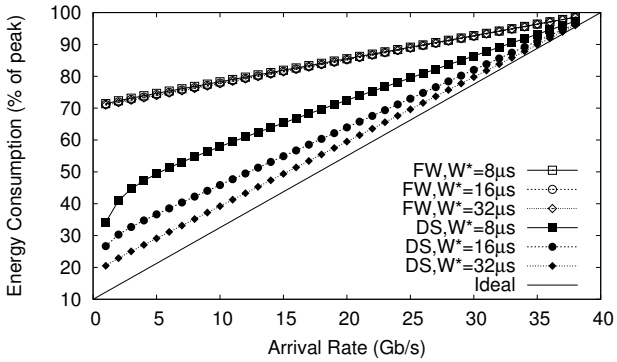
Fig. 6. Dynamic coalescing with Poisson traffic and low target delays.

results also confirm that, with a target delay of 4 μs, greater energy savings could be obtained at the lowest rates using Deep-Sleep while, at medium and high rates, Fast-Wake is the most efficient mode.

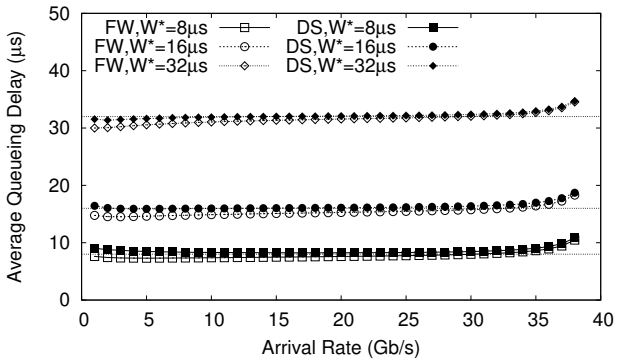
It can also be observed a slight increment in the average queueing delay at the highest (and most unlikely) rates. This is because our algorithm uses approximation (10) to compute the queue threshold required to achieve the target delay and this approximation, as seen in Fig. 3, overestimates the queue threshold at these rates thus causing greater queueing delays. On the other hand, the slight bias observed at the lowest rates is a direct consequence of configuring the queue threshold with values very close to one frame. As previously explained in Section IV, when $Q_{\mathrm{w}} \approx 1$ and the traffic load is low, the average frame delay tends to T_{w}. This is not really a problem

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

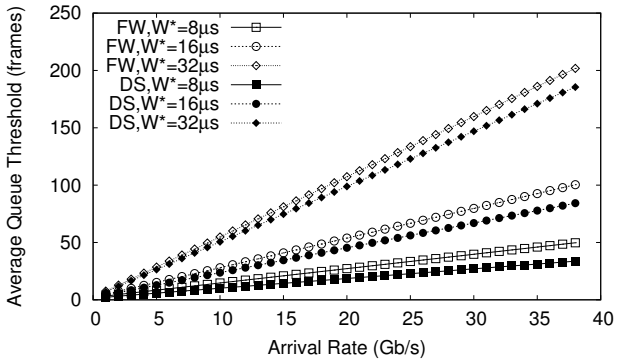
This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310



(a) Energy savings.



(b) Average queueing delay.



(c) Average queue threshold.

Fig. 7. Dynamic coalescing with Poisson traffic and high target delays.

for the Fast-Wake scenarios since all the target delays are greater than $T_{\mathrm{w}}^{\mathrm{f}} = 0.34\,\mu$s and, therefore, the delay constraint is still fulfilled but it actually increases the frame delay above the target delay in the Deep-Sleep scenario at the lowest loads.

Additionally, the average queue thresholds are shown in Fig. 6(c). Our proposal adjusts the queue threshold to actual traffic conditions just choosing higher queue thresholds as traffic load increases. These higher thresholds enable greater power savings without sacrificing frame delay, since the time required to reach them is lower as frame inter-arrival times decrease. Also note that Deep-Sleep computes smaller queue thresholds than Fast-Wake to reach the target delay since it requires a longer awakening time.

Results obtained with the target delays that induce to use the Deep-Sleep mode ($W^* = 8$, 16 and 32 μs) are shown in Fig. 7. The results obtained using Fast-Wake are also shown for comparison. Again, our dynamic algorithm is able to minimize energy consumption and bound the average frame delay at the same time but note that, with Deep-Sleep, the impact of the target delay on energy efficiency is stronger. Indeed, the energy consumed with a target delay of just 32 μs is quite close to that consumed by an idealized interface that could stay in the Deep-Sleep mode all the time there is no traffic to transmit. In contrast, increasing the target delay with Fast-Wake barely has an effect on energy consumption and, due to its intrinsic limitations, the energy consumption is above 70% under all traffic conditions. Finally, note that the delay bias observed in the previous experiments at the lowest loads is not seen with these higher target delays since the queue threshold is now configured with values sufficiently greater than 1.

B. Delay Distribution

As shown in the previous simulation experiments, our proposal is able to maintain the average delay around the target value. Now, we examine the queueing delay distribution to gain a deeper insight into the effects of the mechanism on data traffic. Figure 8 shows the empirical cumulative distribution function (CDF) of the queueing delay for different arrival rates (5, 20 and 35 Gb/s) and target delays. As it can be seen, the queueing delay approximately follows an uniform distribution on the interval $[0, W_{\max}]$. Also note that the queueing delay can grow slightly larger than $W_{\max}$. Recall that this parameter just limits the maximum time the interface can stay in a low power mode, so, if the interface exits the low power mode when the $W_{\max}$ timer expires, some frames will suffer an additional delay due to the transition required to return to the active state.

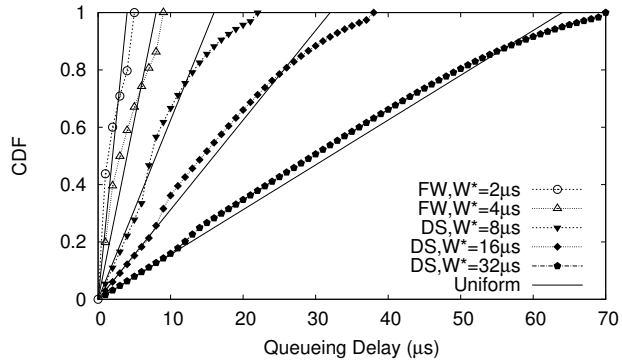
C. Adaptive Coalescing with Self-Similar Traffic

We have conducted some extra simulations to test our proposal under more realistic conditions. It is well-known that packetized traffic like Internet traffic exhibits self-similar characteristics and that self-similar processes can be described using heavy-tailed distributions. So, to validate our proposal with self-similar traffic, in the following experiments frame inter-arrival times will follow a Pareto distribution with shape parameter $\alpha = 1.5$ (that is, with Hurst parameter $H = (3 - \alpha)/2 = 0.75$) [41]. When inter-arrival times follow a heavy-tailed distribution like this, the process counting the number of arrivals in a given time interval is asymptotically self-similar [42].

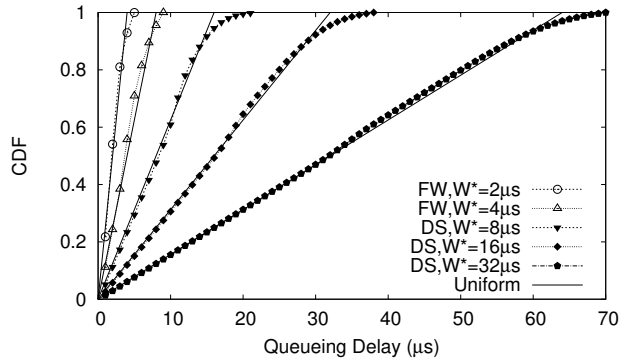
Figure 9 shows that the results obtained with Pareto traffic are very similar to those obtained with Poisson traffic. Despite using a Poissonian model to derive our proposal, it still works well with self-similar traffic, except for the highest (and most unlikely) rates. Note that, at low and moderate rates, the resetting effects of frame coalescing with buffers of small size weakens the effects of long-range dependence traffic. However, at the highest rates, the frequency of coalescing periods is greatly reduced, so the relevance of long-range dependence becomes more significant thus obtaining longer queueing delays than in the previous experiments since queue lengths are expected to be larger with Pareto traffic [43].

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

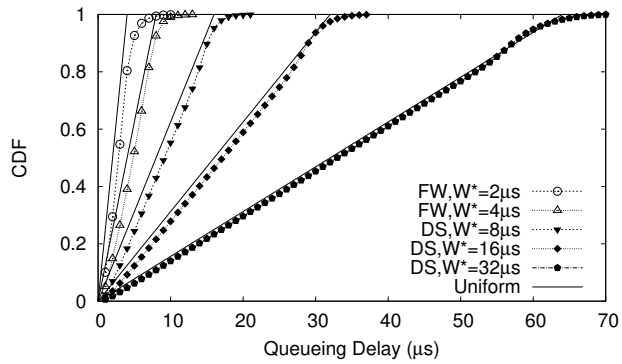
This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310



(a) $\lambda = 5$ Gb/s.

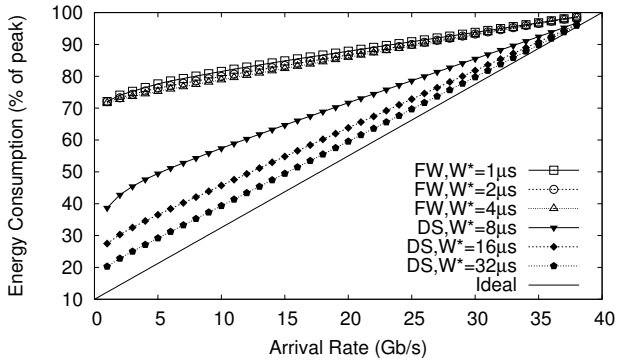


(b) $\lambda = 20$ Gb/s.

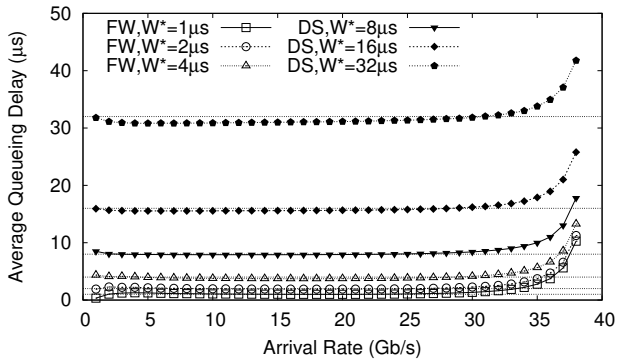


(c) $\lambda = 35$ Gb/s.

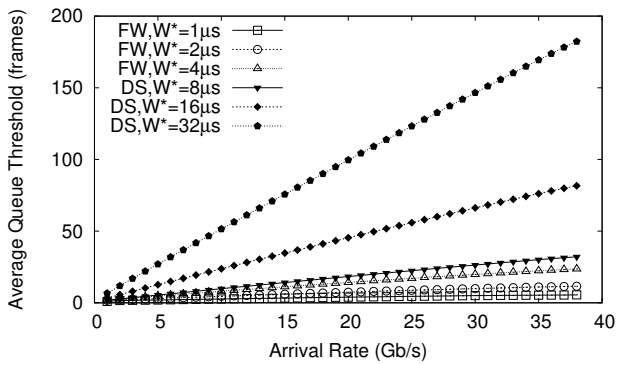
Fig. 8. Cumulative distribution function of the queueing delay.



(a) Energy savings.



(b) Average queueing delay.



(c) Average queue threshold.

Fig. 9. Dynamic coalescing with Pareto traffic.

D. Adaptive Coalescing with Real Traffic

We have also evaluated our proposal using real world traffic traces publicly available from the CAIDA archive [44]. The analyzed CAIDA traces were collected during 2015 on a 10 Gb/s backbone Ethernet link. Though 10 Gb/s EEE links only have a single LPI mode, we assumed that the traced PHY behaves as the previously simulated dual-mode interface and uses the same configuration settings. Figure 10 shows the obtained results. As in the previous experiments with simulated traffic, our proposal is able to achieve notable energy savings while keeping the average queueing delay close to the configured target value. The Poissonian-like nature of aggregated traffic explains the validity of our proposal in this kind of scenarios.

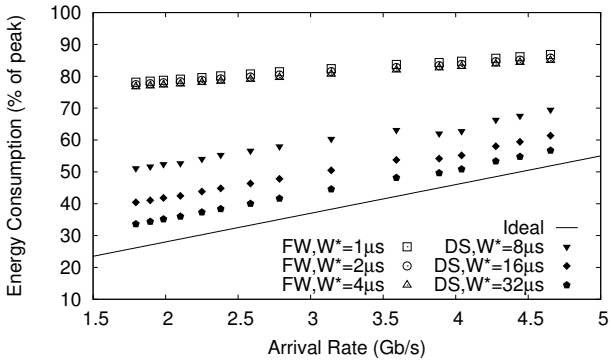
VII. Conclusions

This paper presents the sole mechanism to manage high speed EEE interfaces with two low power modes (Fast-Wake and Deep-Sleep) that is able to minimize their energy consumption and, simultaneously, to bound the average frame delay. With this mechanism, the interface selects the most convenient low power mode according to the configured target delay and remains in the selected mode until the transmission buffer reaches a certain threshold. The coalescing queue threshold is dynamically adjusted using an adaptive algorithm in accordance with the existent traffic conditions and the target delay.

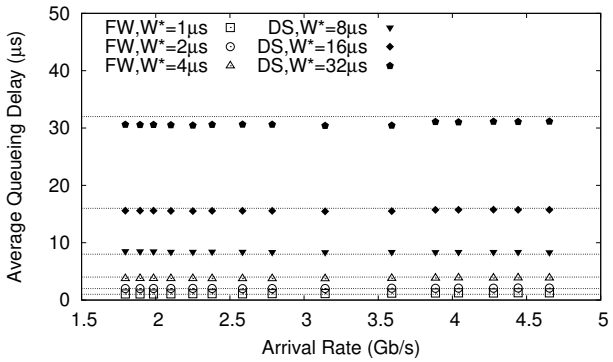
An additional valuable feature of our proposal is that it is very simple to configure. Only the desired average frame delay (and the maximum frame delay) must be configured.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

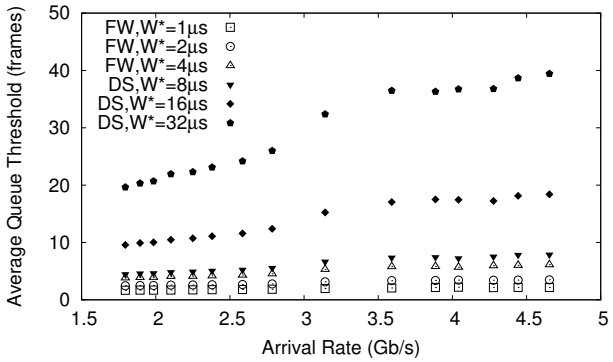
This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310



(a) Energy savings.



(b) Average queueing delay.



(c) Average queue threshold.

Fig. 10. Results for CAIDA traces.

Finally, an unexpected but important conclusion of our study is that the Fast-Wake mode becomes unnecessary in most practical scenarios. We have demonstrated that, when operating the new mechanism with target delays above $4.35\,\mu$s in $40\,$Gb/s interfaces (or $4.44\,\mu$s in $100\,$Gb/s interfaces), Deep-Sleep is always the preferred mode since it obtains the greatest energy savings. Only stringent scenarios requiring target delays below $4.35\,\mu$s (or $4.44\,\mu$s) could benefit from the Fast-Wake mode. We suggest, therefore, that manufacturers should either reduce the power consumption of the Fast-Wake mode if possible, or concentrate their efforts on the Deep-Sleep mode since this would be the most employed mode in practice and can be more easily optimized.

As future work, we plan to modify the firmware of a legacy dual-mode EEE interface to implement the proposed mechanism and then apply a similar measurement setup to that used in [39] to measure the energy savings and the frame delays obtained with our proposal in a real network.

ACKNOWLEDGMENTS

Support for CAIDA's Internet Traces is provided by the National Science Foundation, the US Department of Homeland Security, and CAIDA Members.

Work 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).

REFERENCES

[1] "IEEE Std 802.3az-2010," Oct. 2010. [Online]. Available: http://dx.doi.org/10.1109/IEEESTD.2010.5621025
[2] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, "IEEE 802.3az: the road to energy efficient Ethernet," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 50–56, 2010.
[3] P. Reviriego, J. A. Maestro, J. A. Hernández, and D. Larrabeiti, "Burst transmission for Energy-Efficient Ethernet," *IEEE Internet Comput.*, vol. 14, no. 4, pp. 50–57, Jul. 2010.
[4] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Opportunistic power saving algorithms for Ethernet devices," *Computer Networks*, vol. 55, no. 9, pp. 2051–2064, Jun. 2011.
[5] "IEEE Std 802.3bj-2014 amendment 2: Physical layer specifications and management parameters for 100 Gb/s operation over backplanes and copper cables," Sep. 2014. [Online]. Available: http://dx.doi.org/10.1109/IEEESTD.2014.6891095
[6] H. Barrass, "Options for EEE in 100G," Presentation at IEEE P802.3bj meeting, Jan. 2012. [Online]. Available: http://www.ieee802.org/3/bj/public/jan12/barrass_01a_0112.pdf
[7] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Bounded energy consumption with dynamic packet coalescing," in *Networks and Optical Communications (NOC), 17th European Conference on*, Jun. 2012, pp. 1–5.
[8] A. Chatzipapas and V. Mancuso, "Modelling and real-trace-based evaluation of static and dynamic coalescing for energy efficient Ethernet," in *ACM e-Energy'13*, Berkeley, CA, May 2013, pp. 161–172.
[9] ——, "An M/G/1 model for gigabit energy efficient Ethernet links with coalescing and real-trace-based evaluation," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2663–2675, Oct. 2016.
[10] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli, "Green network technologies and the art of trading-off," in *IEEE INFOCOM Workshops*, Shanghai, China, Apr. 2011, pp. 301–306.
[11] H.-H. Choi, K.-H. Lee, and J.-R. Lee, "Analysis of tradeoff between energy consumption and activation delay in power management mechanisms," *IEEE Commun. Lett.*, vol. 16, no. 3, pp. 414–416, Mar. 2012.
[12] W. Wu, P. Demar, and M. Crawford, "Sorting reordered packets with interrupt coalescing," *Computer Networks*, vol. 53, no. 15, pp. 2646–2662, Oct. 2009.
[13] R. Bruschi, P. Lago, A. Lombardo, and G. Schembra, "Modeling power management in networked devices," *Computer Communications*, vol. 50, p. 95–109, Sep. 2014.
[14] R. Wang, J. Tsai, C. Maciocco, T.-Y. C. Tai, and J. Wu, "Reducing power consumption for mobile platforms via adaptive traffic coalescing," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1618–1629, Sep. 2011.
[15] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Adaptive DRX scheme to improve energy efficiency in LTE networks with bounded delay," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2963–2973, Dec. 2015.
[16] D. Camps-Mur, M. D. Gomony, X. Pérez-Costa, and S. Sallent-Ribes, "Leveraging 802.11n frame aggregation to enhance QoS and power consumption in Wi-Fi networks," *Computer Networks*, vol. 56, no. 12, pp. 2896–2911, Aug. 2012.
[17] X. Zhou and A. Boukerche, "AFLAS: An adaptive frame length aggregation scheme for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 855–867, Jan. 2017.
[18] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "On the use of the doze mode to reduce power consumption in EPON systems," *J. Lightw. Technol.*, vol. 32, no. 2, pp. 285–292, Jan. 2014.

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TCOMM.2017.2700310

[19] G. R. de los Santos, P. Reviriego, and J. A. Hernández, "Packet coalescing strategies for energy efficient high-speed communications over plastic optical fibers," *IEEE J. Opt. Commun. Netw.*, vol. 7, no. 4, pp. 253–263, Apr. 2015.

[20] J. Wu, S. Zhou, and Z. Niu, "Traffic-aware base station sleeping control and power matching for energy-delay tradeoffs in green cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 4196–4209, Aug. 2013.

[21] X. Guo, Z. Niu, S. Zhou, and P. R. Kumar, "Delay-constrained energy-optimal base station sleeping control," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1073–1085, May 2016.

[22] M. A. Marsan, A. F. Anta, V. Mancuso, B. Rengarajan, P. R. Vasallo, and G. Rizzo, "A simple analytical model for energy efficient Ethernet," *IEEE Commun. Lett.*, vol. 15, no. 7, pp. 773–775, Jul. 2011.

[23] D. Larrabeiti, P. Reviriego, 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*, vol. 8, no. 3, pp. 131–138, Mar. 2011.

[24] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, and P. Lago, "A closed-form model for the IEEE 802.3az network and power performance," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 1, pp. 16–27, Jan. 2014.

[25] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "A power saving model for burst transmission in energy-efficient Ethernet," *IEEE Commun. Lett.*, vol. 15, no. 5, pp. 584–586, May 2011.

[26] ——, "How efficient is energy efficient Ethernet?" in *3rd Intl. Congress on Ultramodern Telecommunications and Control Systems (ICUMT 2011)*, Budapest (Hungary), Oct. 2011.

[27] ——, "Optimal configuration of energy-efficient Ethernet," *Computer Networks*, vol. 56, no. 10, pp. 2456–2467, Jul. 2012.

[28] ——, "A GI/G/1 model for 10 Gb/s energy efficient Ethernet links," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3386–3395, Nov. 2012.

[29] M. Mostowfi and K. Christensen, "An energy-delay model for a packet coalescer," in *IEEE Southeastcon*, Orlando, FL, Mar. 2012.

[30] K. J. Kim, S. Jin, N. Tian, and B. D. Choi, "Mathematical analysis of burst transmission scheme for IEEE 802.3az energy efficient Ethernet," *Performance Evaluation*, vol. 70, no. 5, pp. 350–353, May 2013.

[31] J. Meng, F. Ren, W. Jiang, and C. Lin, "Modeling and understanding burst transmission algorithms for energy efficient Ethernet," in *IEEE/ACM 21st International Symposium on Quality of Service (IWQoS)*, Montreal, Canada, Jun. 2013.

[32] A. Chatzipapas and V. Mancuso, "Measurement-based coalescing control for 802.3az," in *IFIP Networking Conference*, Vienna, Austria, May 2016.

[33] M. Mostowfi, "A simulation study of energy efficient Ethernet with two modes of low-power operation," *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1702–1705, Oct. 2015.

[34] ——, "Packet coalescing for dual-mode energy efficient Ethernet: A simulation study," in *Simulation Tools and Techniques (SIMUTools '15), Eighth EAI International Conference on*, Aug. 2015.

[35] M. Mostowfi and K. Shafie, "An analytical model for the power consumption of dual-mode EEE," *IET Electronics Letters*, vol. 52, no. 15, pp. 1308–1310, Jul. 2016.

[36] ——, "Average packet delay in dual-mode EEE: An analytical model," *IET Electronics Letters*, vol. 52, no. 21, pp. 1759–1761, Oct. 2016.

[37] A. Marris, "Comments on EEE operation," Presentation at IEEE P802.3bj meeting, Sep. 2013. [Online]. Available: http://www.ieee802.org/3/bj/public/sep13/marris_3bj_01_0913.pdf

[38] A. Vishwanath, V. Sivaraman, and D. Ostry, "How Poisson is TCP traffic at short time-scales in a small buffer core network?" in *Advanced Networks and Telecommunication Systems (ANTS), IEEE 3rd International Symposium on*, Dec. 2009.

[39] P. Reviriego, K. Christensen, J. Rabanillo, and J. A. Maestro, "An initial evaluation of energy efficient Ethernet," *IEEE Commun. Lett.*, vol. 15, no. 5, pp. 578–580, May 2011.

[40] S. Herrería-Alonso, "DualModeEeeSimulator: A Java program that simulates a dual-mode EEE link," Oct. 2016. [Online]. Available: https://github.com/sherreria/DualModeEeeSimulator

[41] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[42] J. Gordon, "Pareto process as a model of self-similar packet traffic," in *IEEE Global Telecommunications Conference (GLOBECOM'95)*, Nov. 1995.

[43] C. M. Harris, P. H. Brill, and M. J. Fischer, "Internet-type queues with power-tailed interarrival times and computational methods for their analysis," *INFORMS Journal on Computing*, vol. 12, no. 4, pp. 261–271, Nov. 2000.

[44] "The CAIDA UCSD anonymized Internet traces 2015 - Dates used: 20150219, 20150521." [Online]. Available: http://www.caida.org/data/passive/passive_2015_dataset.xml

Copyright (c) 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.