Dynamic EEE Coalescing: Techniques and Bounds

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

***Abstract*—Frame coalescing is one of the most efficient techniques to manage the low power idle (LPI) mode supported by Energy Efficient Ethernet (EEE) interfaces. This technique enables EEE interfaces to remain in the LPI mode for a certain amount of time upon the arrival of the first frame (time-based coalescing) or until a predefined amount of traffic accumulates in the transmission buffer (size-based coalescing). This paper provides new insights on the practical efficiency limits of both coalescing techniques. In particular, we derive the fundamental limits on the maximum energy savings considering a target average frame delay. Additionally, we present new open-loop adaptive variants of both time-based and size-based coalescing techniques. These proposals dynamically adjust the length of the sleeping periods in accordance with actual traffic conditions to reduce energy consumption while keeping the average delay near a predefined value simultaneously. Analytical and simulation results show that the energy consumption of both proposals is comparable to the fundamental limits. Consequently, we recommend the usage of the time-based algorithm in most scenarios because of its simplicity as well as its ability to bound the maximum frame delay at the same time.**

***Index Terms*—Energy efficiency, IEEE 802.3az, energy efficient Ethernet, coalescing, energy-delay trade-off**

I. Introduction

CURRENT Ethernet interfaces have the ability to save power by entering a low power idle (LPI) mode whenever there is no traffic to transmit. This LPI mode, defined in the Energy Efficient Ethernet (EEE) amendment to the Ethernet standard [1], only needs around $10\,\%$ of the energy used during normal operation. However, transitions to and from the LPI mode consume energy and take some time to complete, so proper care has to be placed to decide when and for how long to use this LPI mode. Possibly to spur innovation, the standard does not offer guidelines as for how to employ this mode, leaving the task to devise efficient LPI controllers to hardware designers. Probably, the most popular family of governing algorithms is the one based on *frame coalescing* (also known as *burst transmission*) [2], [3], [4]. These algorithms strive to minimize energy usage by staying in the LPI mode until a significant amount of traffic is ready for transmission. Unfortunately, this has the undesired side-effect of increasing frame delay, so a careful balance between traffic delay and energy consumption is required.

Coalescing algorithms can be fundamentally subdivided into two complementary categories based on the signal used to abandon the LPI mode: time-based coalescers and size-based ones. The first kind determines the amount of coalesced traffic in the LPI mode indirectly, firing a timer when the first frame is queued for transmission. When the timer expires, the interface returns to the normal operating mode. Size-based coalescers, on the contrary, exit the LPI mode when a predefined amount of traffic accumulates. In both cases, the proper tuning of the timer value, or the queue threshold, is critical to get good performance without suffering excessive QoS degradation [5], [6]. A further complication stems from the fact that actual traffic characteristics influence the coalescing parameter tuning, so there is no single value that performs well enough for any traffic load [7].

This paper provides new insights on the practical efficiency limits of coalescing techniques. In particular, we derive the fundamental limits on the maximum energy savings considering a target average frame delay. Our second contribution is a couple of new open-loop dynamic coalescing algorithms: a time-based one and another from the size-based family. Both algorithms dynamically adjust their corresponding coalescing parameters in accordance with actual traffic conditions to reduce energy consumption while keeping the average delay near a predefined value. To asses their relative goodness, we also compare both algorithms against the practical bounds under different traffic conditions. The obtained results show that if the target delay is higher than a few microseconds, the energy consumption of both proposals closely approximates the fundamental limits. Finally, as a result of this comparison, we provide guidelines for the selection of the most convenient algorithm in accordance with the allowable delay characteristics. In any case, we can anticipate that in most scenarios, the time-based algorithm is to be preferred because of its simplicity as well as its ability to bound the maximum frame delay at the same time.

The rest of the paper is organized as follows. Related work is reviewed in Sect. II. Section III presents the energy consumption and delay models on which we will build our proposals. To facilitate the understanding of the dynamic techniques and the computation of the practical bounds, we summarize in Sect. IV and V the Poisson models developed in [8] for both time-based and size-based coalescing algorithms. In Sect. VI the new adaptive versions of both coalescing algorithms are presented. Then, in Sect. VII, we find a lower bound for the energy consumed under the constraint of a target average delay. The proposed dynamic techniques are mathematically analyzed and evaluated through simulation experiments in Sect. VIII and, based on the obtained results, we provide some guidelines for their application in Sect. IX. Finally, in Sect. X we summarize the main conclusions of this work.

II. Related Work

A. EEE Coalescing

Ordinarily, EEE interfaces enter the LPI mode every time the transmission buffer gets empty and, if no coalescing

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

is applied, they resume normal operation as soon as new traffic arrives. Unfortunately, this simple algorithm does not usually provide satisfying results since it triggers an excessive number of mode transitions and a great amount of energy is wasted on them [9]. To reduce the frequency of transitions, coalescing algorithms enable EEE interfaces to remain in the LPI mode until a significant amount of traffic is ready for transmission [2], [3], [4]. Certainly, coalescing frames into bursts extends idle periods but, sadly, also increases traffic delay. If the coalescing algorithm is configured with a long timer duration (or a high queue threshold), frames may suffer excessively large delays. On the contrary, if the coalescing parameter is configured with a too low value, only modest energy savings will be achieved. There is, therefore, a trade-off between energy consumption and frame delay [5], [6]. Moreover, traffic characteristics affect the coalescing parameter tuning, so there is no single value that performs well enough for any traffic load. Consequently, coalescers should dynamically tune the coalescing parameter according to actual traffic conditions to achieve the desired performance [7].

The research community has already provided some dynamic tuning algorithms over the last years. In [7] the authors tune a size-based coalescer to obtain a predefined energy efficiency target. The tuning of a time-based coalescer (and a size-based coalescer) to meet a target average delay is discussed in [10] (in [11], [12], resp.). All the aforementioned approaches rely on a feedback loop for tuning the coalescing parameter so, when traffic conditions are themselves dynamic, there is a convergence period. As is usual for closed-loop systems, the speed of convergence is controlled with a feedback gain parameter that must be carefully configured to guarantee system stability as well as a rapid response to varying traffic conditions. In contrast, we propose new open-loop dynamic algorithms that do not rely on any feedback signal to adapt to actual traffic conditions and maintain system stability.

Finally, [13] provides rules to select the appropriate queue threshold and timer duration values to comply with the average delay requirement when using both size-based and time-based coalescing jointly. [14] also obtains similar results for the case of sleeping base stations using size-based coalescing. Unfortunately, these papers do not provide a dynamic method for adjusting the coalescing parameters, they give no indication of how and when to estimate the traffic variables required to compute the optimal coalescing values and the computation of the optimal queue threshold is excessively complicated to be carried out in real-time.

B. EEE Analytical Models

Many analytical models evaluating EEE behavior have been developed in recent years. Some of them do not consider coalescing and just assume that EEE interfaces awake by the first frame arrival [15], [16], [17]. Other works just provide models for the energy consumed using coalescing with Poisson traffic [18], [19]. More interesting are those models addressing the energy-delay trade-off. For instance, [20] presents a deterministic model for size-based coalescers while [21] models time-based coalescers for Poisson arrivals. Most models analyze the general case that considers the joint use of time-based and size-based coalescing algorithms, both for 1000BASE-T interfaces [11], [12] and for 10GBASE-T ones [6], [8], [13], [22], [23].

In this paper, we focus on $10\,\mathrm{Gb/s}$ (and faster) EEE interfaces,[1] so we build on the GI/G/1 model proposed in [8] for 10GBASE-T interfaces since it provides precise but easy-to-use expressions for the computation of the energy consumption and the average frame delay when using coalescing.

III. Energy Consumption and Delay Models

This paper builds on the analytical model developed in [8], so we assume that frame arrivals follow a general distribution with independent interarrival times I_n, $n = 1, 2, \ldots$, and average arrival rate λ while the service times S_n, $n = 1, 2, \ldots$, are a set of random variables with equal distribution function and mean service rate μ. Furthermore, we assume an utilization $\rho = \lambda/\mu < 1$, thus assuring system stability.

Figure 1 shows an example of EEE operations when using coalescing. In particular, the example shows a complete coalescing cycle during which k frames are received and sent (from the $(n+1)$-th to the $(n+k)$-th frame). To maximize energy savings, the EEE interface is put to sleep as soon as its transmission buffer gets empty and, after a short transition of length T_s, it enters the LPI mode. The interface remains in the LPI mode for a period of random length T_off. Once the wake-up condition dictated by the coalescing technique is met, the interface abandons the LPI mode and, after a transition of length T_w, it starts transmitting the frames received while sleeping. Note that, after the interface is put to sleep, there exists a period of random length T_e with no frames in the transmission buffer (with no job to do). Also note that all the received frames in the coalescing cycle have to wait some amount of time in the transmission queue before they can be transmitted. We denote by W_n the queuing delay experienced by the n-th frame.

A. Energy Consumption

According to [8], the energy consumed by an EEE interface compared with that consumed by a power-unaware interface that does not support the LPI mode is given by

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

where φ_off is the ratio of the energy consumed by the EEE interface in the LPI mode to that consumed in the active state and $\overline{T}_\mathrm{off}$ is the average time the EEE interface remains in the sleeping mode in each coalescing cycle. This expression was obtained making the usual assumption that EEE interfaces approximately consume the same amount of energy during transitions as in the active state. Note that, since φ_off, T_s and T_w are intrinsic characteristics of EEE interfaces, we only need to compute the average length of sleeping periods, $\overline{T}_\mathrm{off}$, to obtain their energy consumption. We assume that $\varphi_\mathrm{off} = 0.1$,

[1] In $10\,\mathrm{Gb/s}$ (and faster) EEE interfaces, frame arrivals cannot interrupt transitions from active to the LPI mode. In addition, mode transitions can take place in each link direction in an independent way.

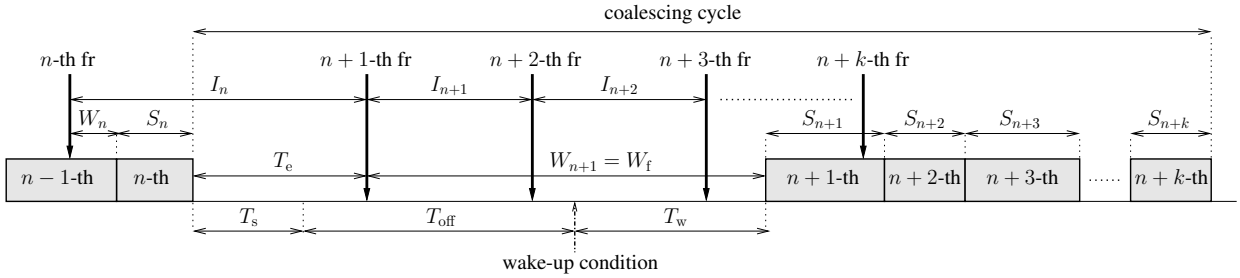


Fig. 1. Example of EEE coalescing operations.

$T_\mathrm{s} = 2.88\,\mu\mathrm{s}$ and $T_\mathrm{w} = 4.48\,\mu\mathrm{s}$ as the standard states for typical 10GBASE-T interfaces.

B. Average Queuing Delay

It is shown in [8] that the average queuing delay when using coalescing is given by the following expression:

$$\overline{W} = \overline{W}_0 + \frac{\overline{W_\mathrm{f}^2} - \overline{T_\mathrm{e}^2}}{2(\overline{W}_\mathrm{f} + \overline{T}_\mathrm{e})} - \frac{\lambda \mathrm{cov}(W, I)}{1-\rho}\,, \tag{2}$$

where W_f is the random queuing delay experienced by the first frame in the coalescing cycle, T_e is the empty period, that is, the random time elapsed since the interface is put to sleep until the first frame arrives, $\mathrm{cov}(W, I)$ is the covariance between the queuing time experienced by a frame and the time until the next frame arrives and

$$\overline{W}_0 = \frac{\lambda^2(\sigma_I^2 + \sigma_S^2) + (1-\rho)^2}{2\lambda(1-\rho)} \tag{3}$$

is a fixed term independent of the coalescing algorithm, with σ_I^2 and σ_S^2 being, respectively, the variances of interarrival and service times.

In the following sections this general model will be particularized for both time-based and size-based coalescing techniques with Poisson traffic. Although it is well-known that frame arrivals are not exponentially distributed [24], they are often modeled as Poisson processes for analytic simplicity. In addition, Poisson traffic can serve as a reasonable approximation to real traffic in sub-second time scales [25] and to aggregated traffic in Internet core links [26].

IV. Time-based Coalescing

We summarize in this section the model developed in [8] for the time-based coalescing technique with Poisson traffic.

A. Average Length of Sleeping Periods

With time-based coalescing, the sleeping period ends when a predefined timer of duration V started upon the arrival of the first frame expires. Therefore, the interface clearly remains $T_\mathrm{e} + V - T_\mathrm{s}$ seconds in the LPI mode. Assuming $V > T_\mathrm{s}$ to guarantee that every transition to sleep makes the interface spend some time in the LPI mode, the average length of sleeping periods is given by

$$\overline{T}_\mathrm{off}^\mathrm{tb} = \int_0^\infty (t + V - T_\mathrm{s}) f_{T_\mathrm{e}}(t)\,\mathrm{d}t\,, \tag{4}$$

where $f_{T_\mathrm{e}}(t)$ is the probability density function of the duration of empty periods. Due to the memoryless property of Poisson traffic, empty periods and interarrival times are identically distributed, so $f_{T_\mathrm{e}}(t) = f_{I_n}(t) = \lambda \mathrm{e}^{-\lambda t}, t \geq 0$. Then, solving integral (4), we get

$$\overline{T}_\mathrm{off}^\mathrm{tb} = 1/\lambda + V - T_\mathrm{s}\,. \tag{5}$$

B. Average Queuing Delay

In this case, the first frame in the coalescing cycle will always wait $V + T_\mathrm{w}$ since the sleeping timer is just fired when it arrives, so $\overline{W}_\mathrm{f} = V + T_\mathrm{w}$ and $\overline{W_\mathrm{f}^2} = (V + T_\mathrm{w})^2$. In addition, $\mathrm{cov}(W, I) = 0$ since the waiting time of a frame is independent of the arrival time of the next frame and $\overline{T}_\mathrm{e} = 1/\lambda$ (and $\overline{T_\mathrm{e}^2} = \sigma_I^2 + \overline{T}_\mathrm{e}^2 = 1/\lambda^2 + 1/\lambda^2 = 2/\lambda^2$) due to the Poisson memoryless property. Substituting all these values in (2), we finally get

$$\overline{W}^\mathrm{tb} = \overline{W}_0 + \frac{\lambda^2(V + T_\mathrm{w})^2 - 2}{2\lambda(1 + \lambda(V + T_\mathrm{w}))}\,. \tag{6}$$

V. Size-based Coalescing

In this section we summarize the model developed in [8] for size-based coalescers with Poisson traffic.

A. Average Length of Sleeping Periods

With size-based coalescing, a sleeping interface wakes up when the number of frames queued in the transmission buffer reaches a predefined threshold Q_w, so

$$\overline{T}_\mathrm{off}^\mathrm{sb} = \int_{T_\mathrm{s}}^\infty (t - T_\mathrm{s}) f_{Q_\mathrm{w}}(t)\,\mathrm{d}t\,, \tag{7}$$

where $f_{Q_\mathrm{w}}(t)$ is the probability density function of the time elapsed since the interface is put to sleep until the arrival of the $Q_\mathrm{w}-$th frame. With Poisson traffic, all interarrival times are identically and exponentially distributed, so the arrival time of the $Q_\mathrm{w}-$th frame is Erlang-Q_w distributed and $f_{Q_\mathrm{w}}(t) = \lambda^{Q_\mathrm{w}} t^{Q_\mathrm{w}-1} \mathrm{e}^{-\lambda t}/(Q_\mathrm{w} - 1)!$. Then, solving integral (7), we get

$$\overline{T}_\mathrm{off}^\mathrm{sb} = \frac{\Gamma(Q_\mathrm{w} + 1, \lambda T_\mathrm{s}) - \lambda T_\mathrm{s} \Gamma(Q_\mathrm{w}, \lambda T_\mathrm{s})}{\lambda \Gamma(Q_\mathrm{w})}\,, \tag{8}$$

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

B. Average Queuing Delay

With this technique, the first frame arriving in the coalescing cycle will wait for the arrival of the next $Q_{\mathrm{w}} - 1$ frames, so

$$\overline{W}_{\mathrm{f}}^{\mathrm{sb}} = \int_0^\infty (t + T_{\mathrm{w}}) f_{Q_{\mathrm{w}}-1}(t)\,\mathrm{d}t\,. \tag{9}$$

As just explained, when considering Poisson traffic, $f_{Q_{\mathrm{w}}-1}(t)$ is the Erlang-$Q_{\mathrm{w}} - 1$ probability density function, so solving this integral, we get $\overline{W}_{\mathrm{f}} = (Q_{\mathrm{w}} - 1)/\lambda + T_{\mathrm{w}}$, and then $\overline{W_{\mathrm{f}}^2} = \sigma_{W_{\mathrm{f}}}^2 + (\overline{W}_{\mathrm{f}})^2 = (Q_{\mathrm{w}} - 1)\sigma_I^2 + ((Q_{\mathrm{w}} - 1)/\lambda + T_{\mathrm{w}})^2$. In addition, $\mathrm{cov}(W, I)$ has a positive value since the queuing delay of the first $Q_{\mathrm{w}} - 1$ frames depends on the next interarrival times. [27] proves that, in this scenario, $\mathrm{cov}(W, I)$ is given by

$$\mathrm{cov}(W, I) = \frac{(1 - \rho)(Q_{\mathrm{w}} - 1)\sigma_I^2}{Q_{\mathrm{w}} - 1 + \lambda \overline{T}_{\mathrm{e}}}\,. \tag{10}$$

Finally, substituting these values in (2), we have

$$\overline{W}^{\mathrm{sb}} = \overline{W}_0 - \frac{Q_{\mathrm{w}} - 1}{\lambda Q_{\mathrm{w}}} + \frac{(Q_{\mathrm{w}} + \lambda T_{\mathrm{w}} - 1)^2 + Q_{\mathrm{w}} - 3}{2\lambda(Q_{\mathrm{w}} + \lambda T_{\mathrm{w}})}\,. \tag{11}$$

VI. Dynamic Coalescing

As previously shown, both energy consumption and average queuing delay depend on the value configured for the coalescing parameter (V or Q_{w}). Consequently, the average queuing delay can be kept around a desired value τ if the coalescing parameter is dynamically and suitably configured according to actual traffic conditions. For example, with time-based coalescing, equating (6) to τ and solving for V, we get that

$$V^* = \tau - \overline{W}_0 - T_{\mathrm{w}} + \frac{1}{\lambda}\sqrt{1 + \left(1 + \lambda(\tau - \overline{W}_0)\right)^2} \tag{12}$$

is the timer duration required to reach the average delay τ.

Similarly, equating (11) to τ, we find that, with size-based coalescing, the coalescing threshold Q_{w}^* needed to keep the average delay τ must meet the condition

$$\begin{aligned} &Q_{\mathrm{w}}^{*3} + \left(2\lambda T_{\mathrm{w}} - 2\lambda(\tau - \overline{W}_0) - 3\right) Q_{\mathrm{w}}^{*2} \\ &\quad + \left(\lambda^2 T_{\mathrm{w}}^2 - 2\lambda^2 T_{\mathrm{w}}(\tau - \overline{W}_0) - 4\lambda T_{\mathrm{w}}\right) Q_{\mathrm{w}}^* \\ &\quad + 2\lambda T_{\mathrm{w}} = 0, \end{aligned} \tag{13}$$

that can be resolved using any of the multiple algebraic methods known to find the roots of cubic equations.

Therefore, to guarantee a given average delay, adaptive coalescing techniques should dynamically adjust the coalescing parameter following the guidelines just provided in this section. We recommend to make this adjustment each time the transmission buffer gets empty, just before putting the interface to sleep at the beginning of a new coalescing cycle. This assures a quick reaction to variable traffic conditions.

However, note that the computation of the optimal coalescing parameters requires accurate estimations of the average arrival rate λ and the $\overline{W}_0$ delay, as shown in (12) and (13). But the computation of $\overline{W}_0$ requires, in turn, to estimate the average service rate (required to compute the utilization ρ) and the variances of both interarrival and service times, as shown in (3). This could certainly hinder the implementation of the dynamic techniques. Hence, to simplify $\overline{W}_0$ computation, we propose to assume Poisson arrivals ($\sigma_I^2 = 1/\lambda^2$) with deterministic frame sizes ($\sigma_S^2 = 0$), so that $\overline{W}_0$ can be approximated as

$$\overline{W}_0 \approx \frac{1 + (1 - \rho)^2}{2\lambda(1 - \rho)}, \tag{14}$$

and it is only necessary to measure the average arrival rate and the average service rate. These variables can be estimated at the beginning of each new coalescing cycle, just before adjusting the corresponding coalescing parameter. For example, the average arrival rate could be computed just dividing the amount of frames received in the last coalescing cycle by its duration. Similarly, the average service rate could be estimated just dividing the nominal rate of the interface by the average size of the frames received in the last coalescing cycle.

On the other hand, recall that cubic equation (13) must be resolved to obtain the optimal queue threshold required to reach the target delay. This computation can be greatly simplified if we assume that Q_{w} is configured with a large value. Under this condition, the average queuing delay in (11) can be approximated as

$$\overline{W}^{\mathrm{sb}} \approx \overline{W}_0 + \frac{Q_{\mathrm{w}} + \lambda T_{\mathrm{w}} - 3}{2\lambda}, \tag{15}$$

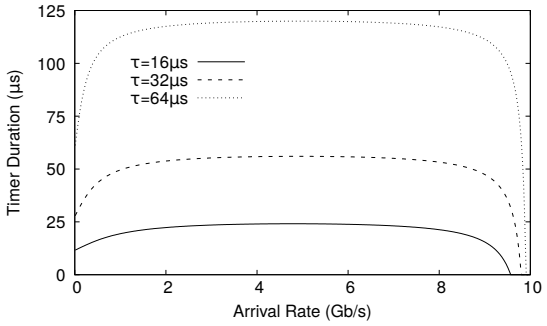
and then the optimal queue threshold can be estimated as

$$Q_{\mathrm{w}}^* \approx 2\lambda\left(\tau - \overline{W}_0 - \frac{T_{\mathrm{w}}}{2}\right) + 3. \tag{16}$$

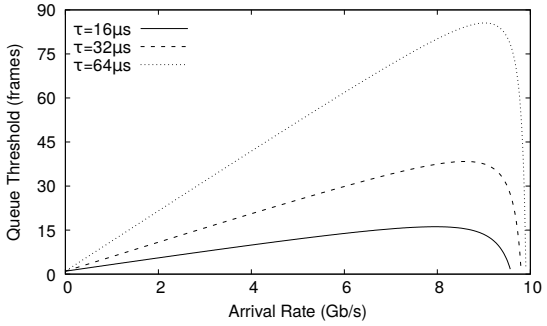
Figure 2 shows the timer durations and the queue thresholds computed using (12) and (16) to maintain different average queuing delays ($\tau = 16$, 32 and 64 µs) with 1500-byte frames. The queue thresholds obtained solving cubic equation (13) are not shown in Fig. 2(b) since they are indistinguishable from those obtained with approximation (16). Note that the optimal coalescing parameters take invalid values ($V^* < 0$, $Q_{\mathrm{w}}^* < 1$) for the highest (and most unlikely) rates. At very high rates (those higher than 9.5 Gb/s), the transmission buffer fills up due to traffic arriving faster than it is processed and the average queuing delay, even in the absence of any sleeping algorithm, will exceed the target delay. So, under these extreme traffic conditions, the delay constraint cannot be fulfilled and the interface should remain active without ever entering the LPI mode thus preventing the queuing delay from increasing even more. Note that this is not a shortcoming of the proposed techniques, but simply an undesired effect of the unavoidable increase on frame delay due to an excessive load.

VII. Lower Bound for Energy Consumption Given a Target Average Delay

Clearly, the more time the EEE interfaces remain in the sleeping mode, the less amount of energy they will consume. Therefore, to compute a lower bound for the energy consumed by EEE interfaces under the constraint of a given average delay, we must previously obtain an upper bound for the average length of sleeping periods ($\overline{T}_{\mathrm{off}}$) under this condition. We will build on the general model presented in Sect. III. Assuming that the average queuing delay equals the target value τ, substituting $\overline{W_{\mathrm{f}}^2} = \sigma_{W_{\mathrm{f}}}^2 + (\overline{W}_{\mathrm{f}})^2$ and $\overline{T_{\mathrm{e}}^2} = \sigma_{T_{\mathrm{e}}}^2 + (\overline{T}_{\mathrm{e}})^2$ in (2),



(a) Timer durations.



(b) Queue thresholds.

Fig. 2. Coalescing parameters required to reach different target delays.

and solving for $\overline{W}_{\rm f}$, we get that the average queuing delay of the first frame in each coalescing cycle must hold

$$\overline{W}_{\rm f} = \tau - \overline{W}_0 + \frac{\lambda {\rm cov}(W,I)}{1-\rho} + \sqrt{\left(\tau - \overline{W}_0 + \frac{\lambda {\rm cov}(W,I)}{1-\rho} + \overline{T}_{\rm e}\right)^2 + \sigma^2_{T_{\rm e}} - \sigma^2_{W_{\rm f}}}. \quad (17)$$

On the other hand, it can be easily seen from Fig. 1 that $\overline{W}_{\rm f} = T_{\rm s} + T_{\rm w} + \overline{T}_{\rm off} - \overline{T}_{\rm e}$, so equating this and (17), and solving for $\overline{T}_{\rm off}$, we have

$$\overline{T}_{\rm off} = \tau - T_{\rm s} - T_{\rm w} - \overline{W}_0 + \frac{\lambda {\rm cov}(W,I)}{1-\rho} + \overline{T}_{\rm e} + \sqrt{\left(\tau - \overline{W}_0 + \frac{\lambda {\rm cov}(W,I)}{1-\rho} + \overline{T}_{\rm e}\right)^2 + \sigma^2_{T_{\rm e}} - \sigma^2_{W_{\rm f}}}. \quad (18)$$

Next we will compute an upper bound for $\overline{T}_{\rm off}$. Firstly, note from (18) that greater sleeping periods are obtained with longer (and more varying) empty periods. Let the $n-$th frame be the last frame in a busy cycle. As shown in Fig. 1, $T_{\rm e} = I_n - W_n - S_n$, so, $\overline{T}_{\rm e} < 1/\lambda - 1/\mu$ because the last frame in a busy cycle always has to wait some amount of time in the transmission queue ($W_n > 0$). Moreover, $\sigma^2_{T_{\rm e}} = \sigma^2_I + \sigma^2_S + \sigma^2_{W_n} + 2{\rm cov}(W_n, S_n) - 2{\rm cov}(W_n, I_n) - 2{\rm cov}(I_n, S_n)$, but ${\rm cov}(W_n, S_n) = {\rm cov}(I_n, S_n) = 0$ since the waiting time experienced by a frame is independent of its length and it is assumed that frame lengths do not depend on the arrival process. Thus, $\sigma^2_{T_{\rm e}} < \sigma^2_I + \sigma^2_S + \sigma^2_{W_n}$, and, since $\sigma^2_{W_n} < \overline{W^2_n} < \overline{(I_n - S_n)^2} = \sigma^2_{I_n - S_n} + \overline{(I_n - S_n)}^2 = \sigma^2_I + \sigma^2_S + (1/\lambda - 1/\mu)^2$, we get that $\sigma^2_{T_{\rm e}} < 2(\sigma^2_I + \sigma^2_S) + (1/\lambda - 1/\mu)^2$.

Additionally, note that longer sleeping periods can be obtained with greater ${\rm cov}(W,I)$ terms, so, to avoid that this covariance is zero, we assume that the waiting time of a frame depends on the arrival time of the next frame as is the case with size-based coalescing. The ${\rm cov}(W,I)$ term can be computed by conditioning on the arrival times of the frames within the coalescing cycle. Note that, for those frames arriving while the interface is sleeping, this covariance is σ^2_I while for those ones arriving once the interface has been reactivated, this term is zero. Averaging over the coalescing cycle, we get that ${\rm cov}(W,I) = \sigma^2_I \overline{N}_{\rm off}/\overline{N}$ where $\overline{N}_{\rm off}$ is the average number of frames received while the interface is sleeping and $\overline{N}$ is the average number of frames served in the whole coalescing cycle. Clearly, $\overline{N}_{\rm off} = \lambda(\overline{W}_{\rm f} - T_{\rm w})$. In addition, it is well-known that, in a GI/G/1 system with vacations, $\overline{N} = \lambda(\overline{W}_{\rm f} + \overline{T}_{\rm e})/(1-\rho)$ where $\overline{W}_{\rm f} + \overline{T}_{\rm e}$ is the average vacation time. Therefore, we get that

$${\rm cov}(W,I) = (1-\rho)\frac{\overline{W}_{\rm f} - T_{\rm w}}{\overline{W}_{\rm f} + \overline{T}_{\rm e}}\sigma^2_I < (1-\rho)\sigma^2_I. \quad (19)$$

Then, assuming that $\sigma^2_{W_{\rm f}} = 0$ (as in the best case with time-based coalescing) and substituting the upper limits of $\overline{T}_{\rm e}$, $\sigma^2_{T_{\rm e}}$ and ${\rm cov}(W,I)$ in (18), we obtain the following upper bound for $\overline{T}_{\rm off}$:

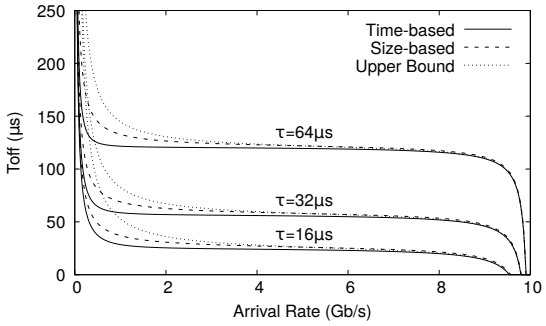
$$\overline{T}_{\rm off} < \tau - T_{\rm s} - T_{\rm w} - \overline{W}_0 + \lambda\sigma^2_I + \frac{1-\rho}{\lambda} + \sqrt{\left(\tau - \overline{W}_0 + \lambda\sigma^2_I + \frac{1-\rho}{\lambda}\right)^2 + 2(\sigma^2_I + \sigma^2_S) + \left(\frac{1-\rho}{\lambda}\right)^2}. \quad (20)$$

Finally, this value can be substituted in (1) to obtain a lower bound for the energy consumed by EEE interfaces under the constraint of a given average queuing delay.

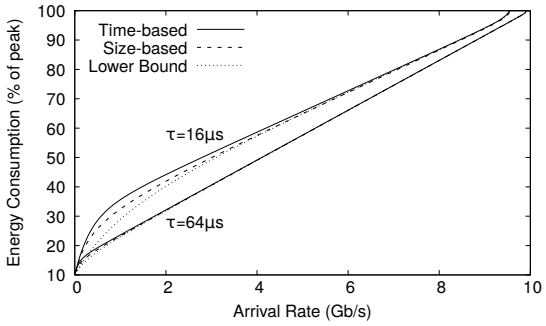
VIII. Evaluation

We firstly used the main results obtained in the analytical model to compute the energy consumed by EEE interfaces when using dynamic coalescing with Poisson traffic and 1500-byte frames. Figure 3(a) shows the average lengths of sleeping periods for both time-based and size-based coalescing techniques when the coalescing parameters are configured following the guidelines provided in Sect. VI. The upper bound given by (20) is also shown in the graph. The results evidence that, at low rates, size-based coalescing achieves longer sleeping periods, although there is still some room for improvement. However, from moderate to high rates, the duration of the sleeping periods obtained with both techniques are practically identical and match the upper bound. Also note that the average length of the sleeping periods for the highest rates tends to zero since, under these extreme conditions, the sleeping algorithm should be temporarily suspended to prevent the queuing delay from increasing even more.

Figure 3(b) shows the energy consumed with the proposed techniques and the lower bound obtained when substituting (20) in (1). As expected, dynamic size-based coalescing achieves greater energy savings since it provides longer sleeping periods, especially at low rates. However, note that, for a moderate target delay of just 64 µs, the energy consumed with

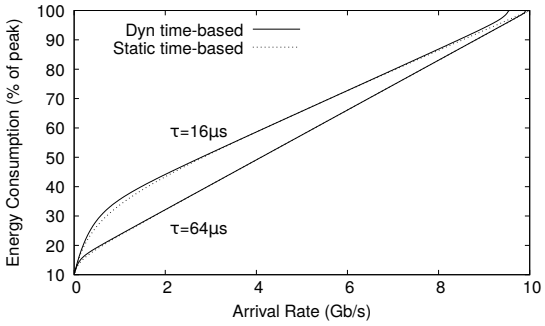


(a) Average length of sleeping periods.

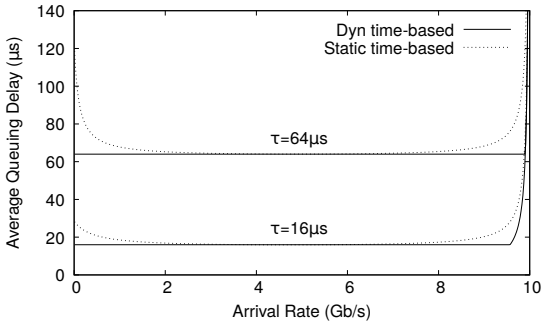


(b) Energy consumption.

Fig. 3. Analytical results with Poisson traffic.



(a) Energy consumption.



(b) Average queuing delay.

Fig. 4. Comparison between dynamic and static time-based coalescing.

both techniques is very similar and, remarkably, very close to the lower bound, even at the lowest rates. And, although not shown in the graph for clarity, the same occurs for greater target delays. Therefore, our proposals are able to minimize energy consumption in practice except when the target delay is configured with an excessively small value.

A. Comparison with Static Coalescing

We then compared the proposed schemes with static coalescing techniques in which the coalescing parameter is configured with a fixed value regardless of actual traffic conditions as those analyzed in [8], [22], [23]. To ensure a fair comparison, the coalescing parameters of the static coalescers were configured with the values required to get an average delay equal to the corresponding target delay when $\lambda = 5$ Gb/s. Thus, after using (12) and (16) to compute the coalescing parameters required for a target delay of $16\,\mu$s ($64\,\mu$s), we set the timer duration for the static time-based coalescer to $24\,\mu$s ($120\,\mu$s) and the queue threshold for the static size-based one to 12 frames (52 frames).

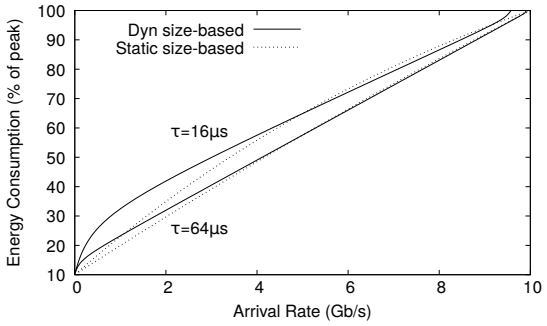
Figure 4 shows the energy consumption and the average queuing delay when using both dynamic and static time-based coalescers. With a $16\,\mu$s target delay, static time-based coalescing consumes a little less energy than the dynamic one at the lowest and the highest loads, but the average delay is moderately increased as well. If the target delay is increased to $64\,\mu$s, then frames still suffer longer delays with static coalescing at extreme loads but without obtaining significant energy savings this time.

Figure 5 shows the results obtained with both dynamic and static size-based coalescers. When using static size-based coalescing, deviations from the target delay are huge, especially at the lowest rates where the obtained delays are just unacceptable since reaching the queue threshold takes an excessively long time. At high rates, the queue threshold is reached too soon, so the average delay is below the target value but at the expense of a small increment in energy consumption.

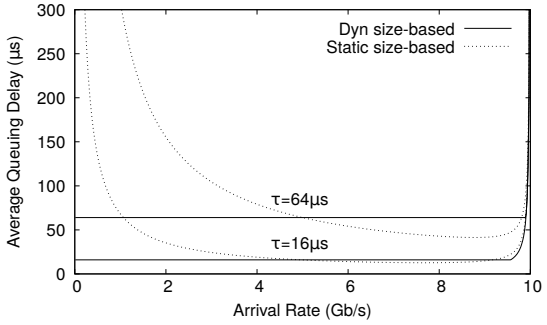
For acceptable performance, static coalescers need to jointly use both time-based and size-based techniques. Thus, at rates lower than the configured rate threshold (5 Gb/s in these experiments), the sleeping timer will expire before reaching the queue threshold in most occasions. Therefore, the coalescer will behave as a time-based one and the queuing delay will be bounded. Conversely, at rates higher than 5 Gb/s, the queue threshold will be likely reached before the sleeping timer expires, so the coalescer will behave as a size-based one thus reducing the frame delay at the cost of a slight increase in energy consumption. In any case, with static coalescing, it is impossible to maintain the average delay near the target value for all the possible traffic loads. At low rates, the average delay will exceed the desired value while, at high rates, the delay will get excessively reduced, thus wasting some energy needlessly.

B. Dynamic Coalescing with Pareto Traffic and Bimodal Frame Sizes

We also evaluated the proposed techniques with an open-source in-house simulator [28] under more stringent conditions. In the following simulation experiments, we considered

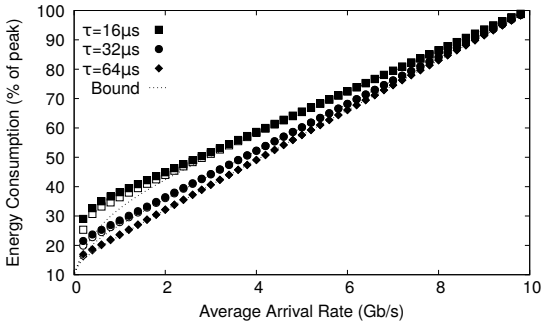


(a) Energy consumption.

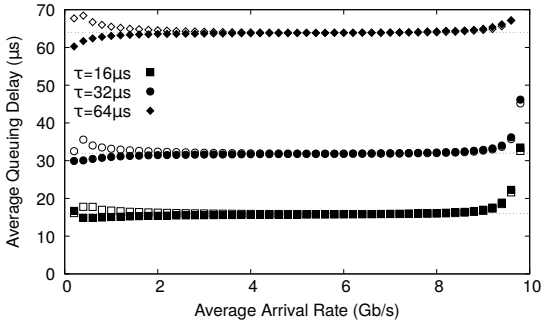


(b) Average queuing delay.

Fig. 5. Comparison between dynamic and static size-based coalescing.



(a) Energy consumption.



(b) Average queuing delay.

Fig. 6. Simulation results with Pareto traffic. Results with time-based (size-based) coalescing are shown with filled (unfilled) points.

Pareto interarrival times with shape parameter $\alpha = 2.5$ to validate our formulas with self-similar traffic.[2] Additionally, to approximate real Internet traffic, frame sizes were set to follow a bimodal distribution with 54% of frames having a size of 100 bytes and the rest with a size of 1500 bytes [29].

Figure 6 shows that the results obtained under these conditions are similar to those achieved with Poisson traffic. Despite the fact that our proposals were derived from a Poisson model, they are still able to keep the average delay near the target value and get energy consumption very close to the lower bound for most loads and target delays. Only at the lowest rates, the measured delay slightly deviates from the target value due to the effects of Pareto long-range dependence.

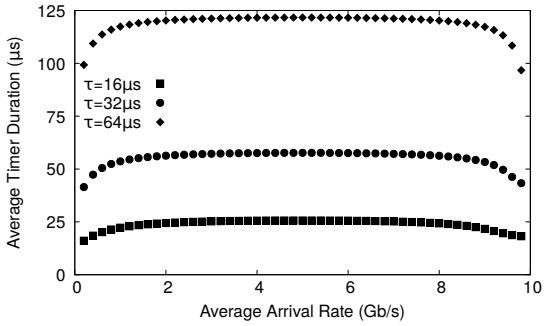
The average values taken by the coalescing parameters are also shown in Fig. 7. As expected, the proposed techniques appropriately tune their corresponding coalescing parameters according to the target delay and actual traffic conditions. Note that, except for the highest and the lowest arrival rates, the duration of the sleeping timer with time-based coalescing is roughly constant for a given target delay. On the other hand, size-based coalescing selects greater queue thresholds as traffic load increases except at the highest rates, in which the queue threshold must be decreased to cope with the unavoidable increase on frame delay due to excessive load. These results are in line with those obtained analytically in Sect. VI (and shown in Fig. 2).

[2]Note that Pareto distributions must be characterized with a shape parameter α greater than 2 to have finite variance. On the other hand, the greater the α parameter, the shorter the fluctuations, so a value of 2.5 is a good trade off to have finite variance along with significant fluctuations.

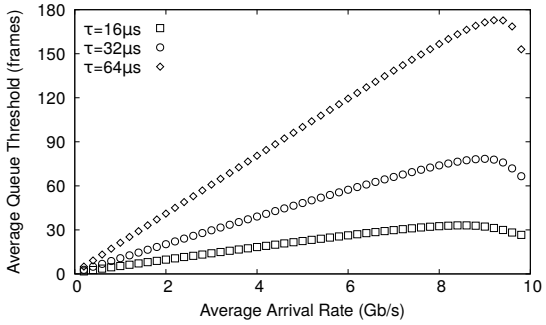
C. Dynamic Coalescing with Real Traffic

Additionally, we evaluated the proposed techniques with real world traffic traces. In particular, we analyzed several CAIDA traces captured on a 10 Gb/s backbone Ethernet link [30]. Figure 8 shows the results obtained with these traces. As in the previous experiments with synthetic traffic, our proposals, when configured with moderate target delays, are able to almost minimize energy consumption while keeping the average queuing delay close to the target value. Remarkably, the proposed techniques still work well even under these more realistic conditions with interarrival times and frame sizes measured in a real link since, as shown in Fig. 9, they are able to properly adjust the coalescing parameters according to the target delay and actual traffic conditions.

Finally, to go deeper into the effects of coalescing on data traffic, we computed the cumulative distribution function (CDF) of the queuing delay in the simulated scenarios. CDFs for different arrival rates and target delays are shown in Fig. 10. Seemingly, the queuing delay with time-based coalescing practically follows an uniform distribution on the interval $[0, 2\tau]$ while, with size-based coalescing, the queuing delay appears exponentially distributed and a significant amount of frames suffer delays greater than 2τ, especially at lower rates. For instance, if a size-based coalescer configured with a 64 µs target delay is applied to the trace with $\lambda = 1.8$ Gb/s, almost 14 % of the frames experience a delay higher than 128 µs, and even 3 % of them suffer delays greater than 192 µs. Therefore, to avoid that some frames suffer excessive delays, size-based coalescers should necessarily incorporate an additional

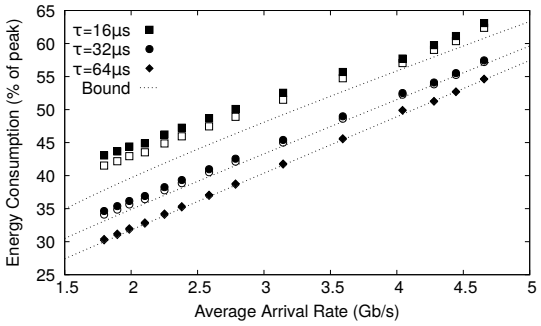


(a) Average timer duration (time-based coalescing).

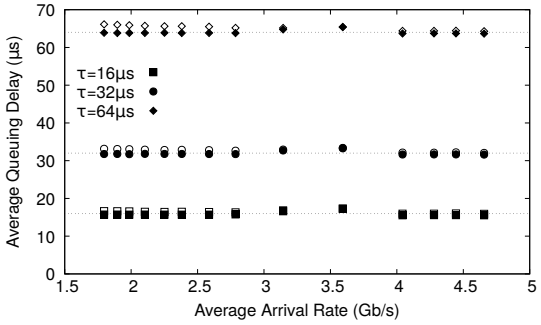


(b) Average queue threshold (size-based coalescing).

Fig. 7. Coalescing parameters with Pareto traffic.



(a) Energy consumption.



(b) Average queuing delay.

Fig. 8. Simulation results for CAIDA traces. Results with time-based (size-based) coalescing are shown with filled (unfilled) points.

timer that limits the maximum time the interface spends in the low power mode.

IX. Recommendations

In this section, we provide some guidelines for the selection of the most convenient dynamic algorithm in scenarios with different delay requirements. Following the results obtained in the previous experiments, we recommend that, in those scenarios with flexible delay requirements, EEE interfaces implement dynamic time-based coalescers with target delays in the range 32–128 µs so that significant energy savings can be obtained without too much QoS degradation.

Recall that, with target delays in this range, although the size-based coalescing algorithm induces slightly longer sleeping periods than the time-based one, this barely has noticeable effects on energy consumption and both algorithms approximately consume the same amount of energy. Consequently, we recommend using time-based coalescing since it is easier to implement as it just requires firing a timer while size-based coalescing requires a frame counting module to trigger the wake-up. Moreover, it is not worth exploring new and more advanced (and surely more complex) coalescing techniques since the energy consumed when using time-based coalescing with target delays from a few tens of microseconds is already close enough to the lower bound.

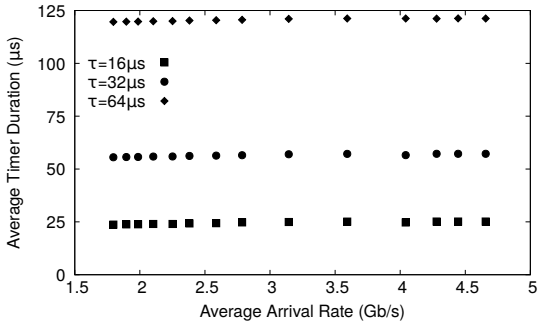
Also, recall that time-based coalescing implicitly limits the maximum frame delay to twice the target delay. In contrast, with size-base coalescing, a significant amount of frames experience excessively long delays. This undesirable side-effect of size-base coalescing can only be avoided if an additional timer that limits the maximum coalescing time is also included, that is, applying time-based coalescing at the same time.

Finally, for those scenarios with stringent delay requirements, we recommend using size-based coalescing configured with a target delay lower than 32 µs to achieve greater energy savings. Probably, in such scenarios, a static timer that bounds the maximum queuing delay should be also fired to avoid annoying delays.

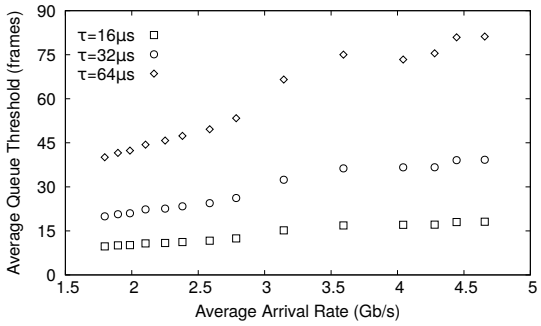
X. Conclusions

This paper provides new and helpful insights on the practical efficiency limits of dynamic coalescing techniques. We presented new open-loop adaptive versions of both time-based and size-based coalescing techniques that dynamically adjust the coalescing parameter according to actual traffic conditions under the constraint of a given average frame delay. We have also derived the fundamental limits on the maximum energy savings when considering a target average frame delay and compared them with the energy savings obtained when using our proposals.

Analytical and simulation results show that the energy consumption of both proposals greatly approximates to its fundamental limits when the target delay is configured with values larger than just a few tens of microseconds. Based on our experiments, we have also provided guidelines for the selection of the most appropriate coalescing technique in accordance with the allowable delay characteristics. In particular, we recommend the application of the dynamic time-



(a) Average timer duration (time-based coalescing).



(b) Average queue threshold (size-based coalescing).

Fig. 9. Coalescing parameters for CAIDA traces.

based algorithm in most scenarios because of its simplicity and its ability to bound the maximum frame delay.

As future work, we plan to research new and more energy efficient dynamic coalescing techniques for those scenarios with very stringent delay requirements.

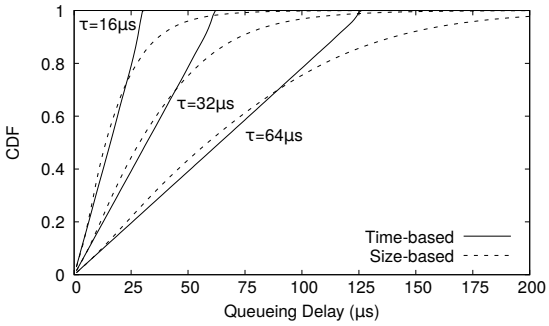
ACKNOWLEDGMENTS

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

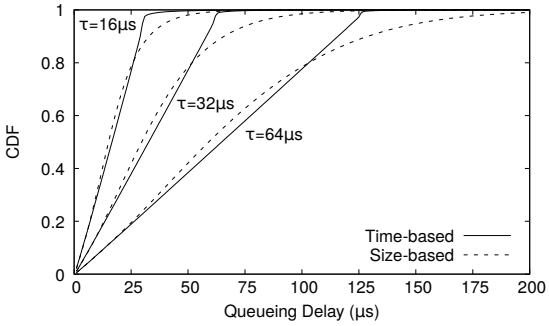
This work was supported by the European Regional Development Fund (ERDF) and the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (atlanTTic), and by the "Ministerio de Economía, Industria y Competitividad" through the project TEC2017-85587-R of the "Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad," (partially financed with FEDER funds)

REFERENCES

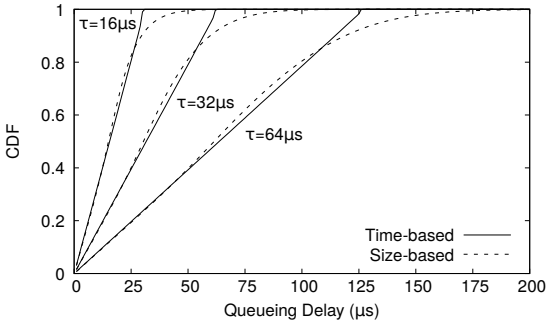
[1] "IEEE Std 802.3az-2010," Oct. 2010. [Online]. Available: http://dx.doi.org/10.1109/IEEESTD.2010.5621025
[2] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, "IEEE 802.3az: the road to energy efficient Ethernet," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 50–56, 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.



(a) $\lambda = 1.8$ Gb/s.



(b) $\lambda = 3.1$ Gb/s.



(c) $\lambda = 4.7$ Gb/s.

Fig. 10. CDF of the queuing delay measured for CAIDA traces.

[5] 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.
[6] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Optimal configuration of energy-efficient Ethernet," *Computer Networks*, vol. 56, no. 10, pp. 2456–2467, Jul. 2012.
[7] ——, "Bounded energy consumption with dynamic packet coalescing," in *Networks and Optical Communications (NOC), 17th European Conference on*, Jun. 2012, pp. 1–5.
[8] ——, "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.
[9] 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.
[10] A. Chatzipapas and V. Mancuso, "Measurement-based coalescing control for 802.3az," in *IFIP Networking Conference*, Vienna, Austria, May 2016.
[11] ——, "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.
[12] ——, "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.
[13] X. Pan, T. Ye, T. T. Lee, and W. Hu, "Power efficiency and delay tradeoff

of 10GBase-T energy efficient Ethernet protocol," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2773–2787, Oct. 2017.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] N. Akar, "Delay analysis of timer-based frame coalescing in energy efficient Ethernet," *IEEE Commun. Lett.*, vol. 17, no. 7, pp. 1459–1462, Jul. 2013.

[22] 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.

[23] J. Meng, F. Ren, W. Jiang, and C. Lin, "Modeling and understanding burst transmission for energy efficient ethernet," *Computer Networks*, vol. 122, pp. 217–230, Jul. 2017.

[24] V. Paxson and S. Floyd, "Wide area traffic: the failure of poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, Jun. 1995.

[25] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary Poisson view of Internet traffic," in *IEEE INFOCOM*, Mar. 2004, pp. 1558–1569.

[26] 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), 2009 IEEE 3rd International Symposium on*, Dec. 2009.

[27] D. P. Heyman and K. T. Marshall, "Bounds on the optimal operating policy for a class of single-server queues," *Operations Research*, vol. 16, no. 6, pp. 1138–1146, Nov. 1968.

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

[29] D. Murray and T. Koziniec, "The state of enterprise network traffic in 2012," in *18th Asia–Pacific Conference on Communications*, Oct. 2012, pp. 179–184.

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