Measurement-Based Coalescing Control for 802.3az

Angelos Chatzipapas* and Vincenzo Mancuso+
*Universidad Carlos III de Madrid, Madrid, Spain, +IMDEA Networks Institute, Madrid, Spain
Email: {angelos.chatzipapas, vincenzo.mancuso}@imdea.org

***Abstract*—IEEE 802.3az standard (EEE), is the energy-aware alternative to legacy Ethernet. To save energy by extending the sojourn in the *Low Power Idle* state of EEE, packet coalescing has been proposed. While coalescing improves by far the energy efficiency of EEE, it is still far from achieving energy consumption proportional to traffic. Moreover, coalescing can introduce high delays. In this work, we use sensitivity analysis to evaluate the impact of coalescing timers and buffer sizes, and to shed light on the delay incurred by adopting coalescing schemes. Accordingly, we design and study measurement-based coalescing control solutions that tune the coalescing parameters on-the-fly, thus adapting the link to the instantaneous load and controlling the coalescing delay experienced by the packets. Our results show that, by relying on run-time delay measurements, simple and practical adaptive coalescing schemes outperform traditional static and dynamic coalescing. Notably, our schemes double the energy saving benefit of legacy EEE coalescing and allow to control the coalescing delay.**

***Index Terms*—IEEE 802.3az; Coalescing; Data Centers; Efficiency; Sensitivity; Simulation.**

I. Introduction

More than 20% of the energy consumption in data centers is due to the network operation, which establishes network as the second biggest energy consumer in data centers [1]. While high-speed Ethernet cards constantly absorb a considerable part of a server's consumption—e.g., 10 *Gbps* cards consume ∼15 W [2]—recent studies have shown that network links are underutilized: ∼40% are "comatose" and another ∼40% of the links are loaded no more than 10% [3]. Hence, there is a clear need for introducing a network-wide energy saving mechanism.

To this goal, IEEE 802.3az [4], known as Energy Efficient Ethernet (EEE), introduces a Low Power Idle state (LPI) for unutilized links. However, in terms of energy saving, EEE underperforms even under low traffic conditions due to LPI transitioning delays [5], [6] and thereby more advanced solutions are needed. Packet coalescing [7], [8] has been proposed to enforce longer sojourns in LPI state, thus improving the energy proportionality of EEE. However, coalescing has a cost, i.e., additional queueing delay for packets.

Using sensitivity analysis, this paper discusses the properties of coalescing techniques for EEE gigabit links, and proposes the design of delay-controlled adaptive coalescing schemes that effectively trade off energy saving and delay guarantees. Specifically, the work (i) analytically studies the performance of gigabit EEE links with coalescing using real data traces that have been captured in an operational web hosting center, (ii) proposes *measurement-based coalescing control* algorithms (MBCC) that almost halve the energy consumption of EEE links with respect to legacy coalescing, while maintaining the coalescing delay bounded and (iii) shows that significant economy can be achieved in a typical data center (∼$1.7M/year).

ISBN 978-3-901882-83-8 © 2016 IFIP

Our goal is to design a new class of adaptive coalescing algorithms for EEE links, namely MBCC. To achieve this goal, we analytically build on top of the analysis we presented in [6], which accurately models the behavior of coalescing buffers in gigabit EEE links with static coalescing parameters.[1] Specifically, our prior work [6] accounts for the fact that energy saving features of gigabit EEE links are triggered by the traffic activity in both link directions simultaneously. Namely, gigabit EEE links exhibit a *bidirectional behavior*. On the one hand, the model of [6] allows to estimate both the potential energy saving and the coalescing delay, but, on the other hand, it does not show how to configure the coalescing parameters optimally. Here, we derive a sensitivity analysis of the coalescing delay and energy saving with respect to the coalescing timer duration and the coalescing buffer size, and use it to design measurement-based control schemes that outperform legacy coalescing schemes.

Our new analytical study reveals the importance of coalescing parameters in different scenarios, and unveils that by adjusting the sole coalescing timer duration, it is possible to tune the link performance to achieve near-optimal energy saving, while incurring controlled coalescing delay.

Exploiting our analytical findings, *we design a simple measurement-based delay-controlled distributed adaptive coalescing scheme* in which network cards at the edge of the link coordinate by running a simple distributed algorithm to sense the delay incurred by packets. Our proposal uses the sensed delay as control signal to trigger the dynamic adaptation of the coalescing timer in the direction identified through the analysis.

Notably, our study goes beyond existing results on dynamic/adaptive coalescing [6], [9], [10]. Indeed, the key and novel feature of MBCC proposal, which makes it different from the class of dynamic algorithms studied in [6], is that we explicitly account, through measurements, for the delay experienced by packets in the EEE link.

With our approach, adaptive coalescing can outperform static coalescing by a large factor. We validate the superiority of our MBCC schemes with respect to other existing solutions by

[1] We focus on gigabit links because they present the most challenging behavior for both modeling and implementation of coalescing strategies, as explained in Section II, and they are the most commonly deployed links in data centers. However, the algorithms presented in this paper can be used for the whole EEE link speed range.

using real traffic traces that we have captured in an operational web hosting center.

The rest of the paper is organized as follows. Section II describes the basic functionality of EEE, with and without coalescing, and explains the basic results available for the modeling of gigabit EEE links. Section III presents a sensitivity analysis of the parameters of EEE with coalescing. In Section IV we design MBCC. In Section V we benchmark our schemes and legacy ones. In Section VI we discuss related work. Finally Section VII concludes the paper.

II. Background

A. EEE Gigabit links

The goal of EEE is to achieve *energy proportionality*, i.e., that energy consumption be proportional to link load. EEE introduces (i) a low power state (namely Low Power Idle - LPI), in which the link does not serve traffic and consumes about 10% of the energy consumed by legacy Ethernet, (ii) an Active state (state A) which performs like legacy Ethernet serving the traffic, (iii) a Sleep state (state S), which is the transition of the link from state A to state LPI, and (iv) a Wake Up state (state W) which is the transition from state LPI to state A. In LPI, a "Refresh" message is sent every T_q time units, in order to check the condition of the link (e.g., connection, interference level, synchronization, etc.) and therefore to save time and resources when the link resumes its activity. For different Ethernet speed, e.g., 100 $Mbps$, 1 $Gbps$, and 10 $Gbps$, EEE has different specifications and transition mechanisms among states. Next, we describe the interesting and most deployed case of 1 $Gbps$ links where, unlike in the other cases, the traffic in both link directions has to be taken into consideration in order to switch between states.

In Fig. 1 we can see the specific EEE state transition graph for gigabit links, in which states L and C are introduced to differentiate pure idle and idle-with-coalesced-packets during LPI, as described later in Section II-B for the case of coalescing operation. The gigabit EEE link can start the transition to state LPI (state S) only when both link directions are inactive. If there is no arrival during an interval T_s (time to switch-off part of the electronics and go to sleep) the link successfully enters state LPI. In contrast, if there is an arrival during the transition in either of the two directions, the link switches back to state A in order to serve the packet. The link remains in state LPI as long as there is no packet arrival. As soon as a packet arrives, the link transits back to state A, which takes T_w time units, i.e., the time required to switch on all electronic parts (state W). For gigabit EEE links, the minimum values (which are typically implemented) for T_s, T_w and T_q are 182 μs, 16 μs and 20 ms, respectively. Note that energy-saving operations of gigabit EEE links are equally affected by arrivals in any link direction, so we refer to such a behavior as *bidirectional*.

B. EEE links with coalescing

Studies of EEE [5], [6] have shown that it is very inefficient and it does not provide any significant energy saving benefit for network loads that exceed a few percents (>5%). The main reasons for this behavior are that (i) the interarrival time between packets may prevent the link to enter state LPI (inter-packet spacing less than T_s) and (ii) packet arrivals do not allow long sojourns in state LPI, and thus most of the time is spent in transitioning. For instance, gigabit links spend 12 μs to serve a 1500-byte packet, against the 182 μs plus 16 μs required for the transition from state A to state LPI and back to state A passing through states S and W.

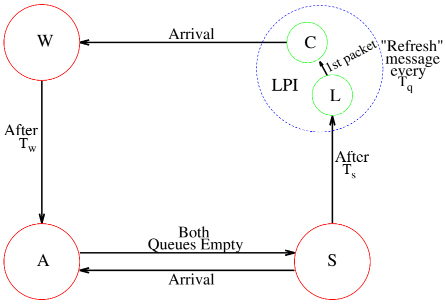


Fig. 1: State transitions for 1 $Gbps$ links with coalescing.

To face the above described issues, packet coalescing has been proposed. Coalescing prolongs the duration of state LPI since it introduces (i) two buffers of N_c packets, one for each link direction, where packets can be stored while the link is in state LPI and (ii) a timer of duration T_c which counts down from the arrival of the first packet in state LPI.

As depicted in Fig. 1, coalescing introduces two new states, which detail coalescing operations within state LPI: (i) state L, where the link enters after state S and in which it remains until it receives a packet in either of the two link directions, and (ii) state C, during which multiple packets are coalesced. Both in state L and state C, the link behaves (and absorbs low power) like in state LPI of a legacy EEE link.

The packet that triggered the transition from state L to state C starts the timer T_c, and the transition from state C to state W occurs after the timer T_c expires or when one of the two coalescing buffers fills up. We denote with τ_c the variable-size interval during which the link remains in state C.

C. Performance of Gigabit EEE links with coalescing

For the analysis presented in this paper, we build on the model presented in our prior work [6], which is the only accurate model that considers the bidirectional behavior of gigabit EEE links. For ease of presentation, here we report from [6] the expressions for the energy saving factor η_{LPI} and the average delay D_i for packets transmitted in direction i, where $i \in \{1, 2\}$ indicates the two possible link directions:

$$\eta_{LPI} = \frac{\frac{1}{\lambda_1+\lambda_2} + E[\tau_c]}{E[T_{cycle}]}; \qquad (1)$$

$$D_i = \frac{\sum_{\alpha \in \{A,S,L,C,W\}} n_\alpha^{(i)} D_\alpha^{(i)}}{\lambda_i E[T_{cycle}]}, \quad i \in \{1, 2\}. \qquad (2)$$

In the above expressions, the parameters λ_i represent the packet arrival rates in the two link directions, $E[\tau_c]$ is the average time that the link spends in state C, $E[T_{cycle}]$ is the average time spent between two consecutive transitions to state L (i.e., a system cycle), $n_{\alpha}^{(i)}$ corresponds to the number of packets received in link direction i in state α (denoting one of the states A, S, L, C, W), $D_{\alpha}^{(i)}$ is the average delay that the packets suffer in state α and direction i. From the results in [6], it is also possible to see that $\lambda_1 > \lambda_2 \Rightarrow D_2 > D_1$, so that the least loaded link suffers the highest delay. As concerns the duration of state C, we elaborate on the results of [6] and obtain the following expression for $E[\tau_c]$:

$$E[\tau_c] = \sum_{k=0}^{N_c-2} \sum_{j=0}^{N_c-2} \frac{\lambda_1^k \lambda_2^j}{k!j!} \int_{t=0}^{T_c} t^{k+j} e^{-(\lambda_1+\lambda_2)t} dt. \qquad (3)$$

As it is clear from the above expression, $E[\tau_c]$ increases with both T_c and N_c, and so does $E[T_{cycle}]$, which strongly depends on $E[\tau_c]$. Below we report an approximation for $E[T_{cycle}]$ from [6], expressed as a function of loads and arrival rates in the two link directions, i.e., ρ_i and λ_i, respectively:

$$\begin{aligned} E[T_{cycle}] =& (T_w + E[\tau_c]) \left[1 + \frac{1}{\lambda_1+\lambda_2} \left(\frac{\lambda_1 \rho_1}{1-\rho_1} + \frac{\lambda_2 \rho_2}{1-\rho_2} \right) \right] \\ &+ \frac{e^{(\lambda_1+\lambda_2)T_s}}{\lambda_1+\lambda_2} \left[\frac{\rho_1}{1-\rho_1} + \frac{\rho_1^2 (2-\rho_1)(\lambda_1 \rho_2 + \lambda_2)}{2\lambda_1 (1-\rho_1\rho_2)(1-\rho_1)^2} \right. \\ &+ \left. \frac{\rho_2}{1-\rho_2} + \frac{\rho_2^2(2-\rho_2)(\lambda_2\rho_1 + \lambda_1)}{2\lambda_2(1-\rho_1\rho_2)(1-\rho_2)^2} + 1 \right]. \qquad (4) \end{aligned}$$

By defining two positive coefficients a and b, that only depend on arrival rates, loads, and EEE parameters T_w and T_s, the previous result can be expressed as a linear function:

$$E[T_{cycle}] = a + b\, E[\tau_c], \qquad (5)$$

where a and b are constants that can be computed by comparing (4) and (5). In the above formulas, the dependency on T_c and N_c is concentrated in the term $E[\tau_c]$, therefore $E[T_{cycle}]$ grows with both T_c and N_c.

The analysis of $D_{\alpha}^{(i)}$ and $n_{\alpha}^{(i)}$ can be found in [6]. Here it is sufficient to recall that $D_L^{(i)}$, $D_C^{(i)}$, $D_W^{(i)}$, $n_C^{(i)}$ and $n_A^{(i)}$ can be expressed as a constant plus a term proportional to $E[\tau_c]$, and thus they also grow with T_c and N_c.

III. Sensitivity Analysis of EEE with Coalescing

We now proceed with a novel study on the sensitivity analysis of EEE performance with respect to the coalescing parameters. Specifically, we want to study the change of both energy saving and average packet delay when we modify either T_c or N_c. Thus, we apply the method of partial derivatives with respect to T_c and N_c.

The partial derivatives with respect to either T_c or N_c for both η_{LPI} and D_i show a dependence on the partial derivative of $E[\tau_c]$ as can be seen from the analysis presented in Section II-C. Thus, next we report the partial derivative of $E[\tau_c]$ (and $E[T_{cycle}]$), the rest is mere calculation.

A. Partial derivatives with respect to T_c

The partial derivative of $E[\tau_c]$ with respect to T_c is

$$\frac{\partial E[\tau_c]}{\partial T_c} = \sum_{k=0}^{N_c-2} \sum_{j=0}^{N_c-2} \frac{\lambda_1^k \lambda_2^j}{k!j!} T_c^{k+j} e^{-(\lambda_1+\lambda_2)T_c} > 0, \quad \forall T_c > 0; \quad (6)$$

and the partial derivative of $E[T_{cycle}]$ with respect to T_c is given by the following expression:

$$\begin{aligned} \frac{\partial E[T_{cycle}]}{\partial T_c} =& \frac{\partial}{\partial T_c} \left\{ E[\tau_c] \left[1 + \frac{1}{\lambda_1+\lambda_2} \left(\frac{\lambda_1\rho_1}{1-\rho_1} + \frac{\lambda_2\rho_2}{1-\rho_2} \right) \right] \right\} \\ =& \left[1 + \frac{1}{\lambda_1+\lambda_2} \left(\frac{\lambda_1\rho_1}{1-\rho_1} + \frac{\lambda_2\rho_2}{1-\rho_2} \right) \right] \frac{\partial E[\tau_c]}{\partial T_c} \qquad (7) \\ =& \, b \, \frac{\partial E[\tau_c]}{\partial T_c} > 0, \quad \forall T_c > 0. \qquad (8) \end{aligned}$$

Finally, we get the partial derivative of the energy saving η_{LPI} with respect to T_c as follows:

$$\begin{aligned} \frac{\partial \eta_{LPI}}{\partial T_c} =& \frac{\frac{\partial E[\tau_c]}{\partial T_c} E[T_{cycle}] - \frac{\partial E[T_{cycle}]}{\partial T_c} \left(\frac{1}{\lambda_1+\lambda_2} + E[\tau_c] \right)}{E^2[T_{cycle}]} \\ =& \frac{a - \frac{b}{\lambda_1+\lambda_2}}{(a + b\, E[\tau_c])^2} \frac{\partial E[\tau_c]}{\partial T_c}. \qquad (9) \end{aligned}$$

From the above expressions, it is clear that the energy saving is a monotonic function of T_c, and moreover $\frac{\partial \eta_{LPI}}{\partial T_c} > 0, \forall T_c > 0$. Therefore the delay monotonically increases with T_c.

Similarly, the partial derivative of the delay D_i with respect to T_c is:

$$\begin{aligned} \frac{\partial D_i}{\partial T_c} =& \frac{\left(\frac{\rho_i}{2\mu_i(1-\rho_i)} - D_i \right) \frac{\partial E[T_{cycle}]}{\partial T_c} - \frac{\rho_i}{2\mu_i(1-\rho_i)} \frac{\partial E[\tau_c]}{\partial T_c}}{E[T_{cycle}]} \\ &+ \frac{\left(\frac{1}{\lambda_1+\lambda_2} + T_w + E[\tau_c] \right) (1+\rho_i) \frac{\partial E[\tau_c]}{\partial T_c}}{E[T_{cycle}]}. \qquad (10) \end{aligned}$$

Also in this case it is possible to show that $\frac{\partial D_i}{\partial T_c} > 0, \forall T_c > 0$ as far as loads are not extremely high.In practice, high loads prevent any EEE benefit [6], [7], [8], and therefore we can safely assume that the delay monotonically increases with T_c under the circumstances in which energy saving can be achieved.

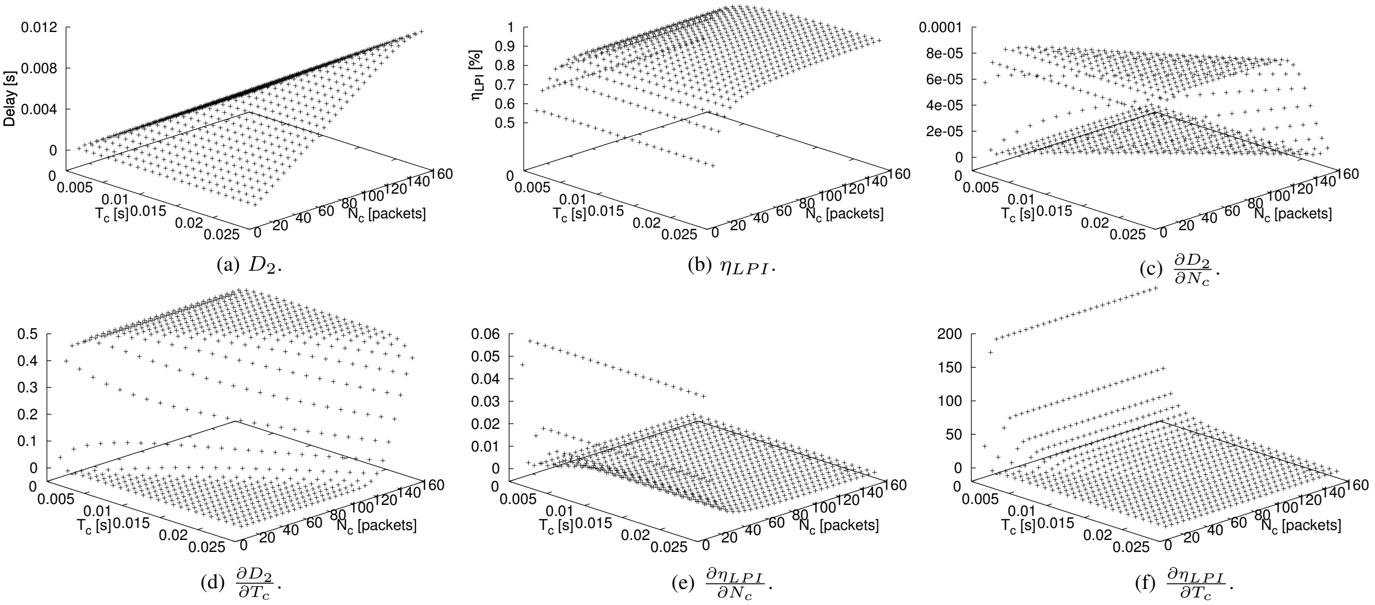
B. Partial derivative with respect to N_c

Regarding the partial derivative of $E[\tau_c]$ with respect to N_c, since N_c takes only integer values (it refers to packets) we consider the forward difference between $E[\tau_c]$ computed at $N_c + 1$ and at N_c:

$$\begin{aligned} \frac{\partial E[\tau_c]}{\partial N_c} \approx& \Delta_{N_c}[E[\tau_c]](N_c) = \frac{E[\tau_c](N_c+1) - E[\tau_c](N_c)}{1} \\ =& \sum_{j=0}^{N_c-2} g_{\lambda_1\lambda_2}(N_c-1, j) + \sum_{k=0}^{N_c-2} g_{\lambda_1\lambda_2}(k, N_c-1) \\ &+ g_{\lambda_1\lambda_2}(N_c-1, N_c-1) > 0, \quad \forall N_c \geq 2; \quad (11) \end{aligned}$$

where $g_{\lambda_1\lambda_2}(k,j) = \frac{\lambda_1^k \lambda_2^j}{k!j!} \int_{t=0}^{T_c} t^{k+j} e^{-(\lambda_1+\lambda_2)t} > 0, \ \forall T_c > 0$.

With the above, the partial derivatives of $E[T_{cycle}]$, η_{LPI}, and D_i with respect to N_c have the same form as their partial derivatives with respect to T_c. Therefore, we can conclude that energy saving and delay grow monotonically with N_c as well.



(a) D_2.

(b) η_{LPI}.

(c) $\frac{\partial D_2}{\partial N_c}$.

(d) $\frac{\partial D_2}{\partial T_c}$.

(e) $\frac{\partial \eta_{LPI}}{\partial N_c}$.

(f) $\frac{\partial \eta_{LPI}}{\partial T_c}$.

Fig. 2: Coalescing delay, energy saving, and their partial derivatives with respect to T_c and N_c. Since the delay due to coalescing is higher in the least loaded link direction, we show only the delay for packets transmitted in that direction (D_2).

TABLE I: Maximum η_{LPI} for $D_{target} \leq 1\ ms$ (ρ_1, ρ_2, λ_1, λ_2 are taken from real traffic traces)

ρ_1 [%]	ρ_2 [%]	λ_1 [pkts/s]	λ_2 [pkts/s]	$Max\{\eta'_{LPI}\}$ [%]	T'_c [ms]	N'_c [packets]	$Max\{\eta''_{LPI}\}$ [%]	T''_c [ms]	N''_c [packets]
0.11	5.25	2186	4343	82.09	≥ 3	≥ 32	82.09	=2	100
10.54	0.66	10410	5324	62.84	≥ 7	≥ 22	60.02	=2	100
0.57	32.68	10051	27459	15.34	≥ 9	$= 205$	8.74	≥ 5	100
1.01	40.52	17091	34042	1.80	≥ 10	$= 255$	0.75	≥ 4	100
5.06	0.5	5409	3809	77.50	≥ 5	≥ 15	66.55	$= 1$	100
1.14	17.93	9639	17320	37.59	≥ 7	≥ 75	31.17	$= 3$	100
0.20	0.06	310	268	92.72	$= 1$	≥ 15	92.72	$= 1$	100

C. Discussion

The partial derivatives with respect to either T_c or N_c show the strong dependency of D_i and η_{LPI} on $E[\tau_c]$ (and on $E[T_{cycle}]$ but this also depends on $E[\tau_c]$). Furthermore, the value of $E[\tau_c]$ grows with T_c and N_c, and we have shown that both η_{LPI} and D_i monotonically grow with T_c and N_c.

To graphically see the impact of T_c and N_c on the delay, D_i, and the energy saving, η_{LPI}, we plot in Fig. 2 an example of partial derivatives, representing the behavior of η_{LPI} and D_i for different T_c and N_c values when the offered load is $\rho_1 = 5.06\%$ and $\rho_2 = 0.5\%$. These loads correspond to a load profile of a traffic trace we collected on a gigabit link in a large web hosting center. Moreover, this is a representative link load since, according to [3], about 80% of the links operate with less than 10% of load, so that the selected case represents a medium load case.

Specifically, Fig. 2a illustrates the behavior of the delay experienced in the most loaded link direction (which is the highest of the two average delays). The figure shows that the delay quickly grows to unacceptable values with both T_c and N_c. The energy saving η_{LPI} also grows, but it does it faster with small values of T_c and N_c, and afterwards it saturates. Overall, the impact of T_c and N_c seems similar. However, the study of the partial derivatives presented in Fig. 2 unveils that both delay and energy saving are more sensitive to changes in T_c rather than in N_c. Indeed, Figs. 2c, 2d, 2e, and 2f point out that the partial derivatives with respect to T_c are up to three orders of magnitude higher than the ones with respect to N_c. We have observed the same behavior for a large range of load combinations, although the results are not shown here due to space limitations. Therefore, we can say that T_c is more important than N_c in the control of delay and energy saving in EEE. Another important observation is that the impact of N_c saturates for relatively small values of the coalescing buffer size, i.e., implementing buffer sizes of 100 packets allows to achieve the highest possible energy saving.

To validate the above observations, we report in Table I a few representative case studies corresponding to different combinations of average loads ρ_1 and ρ_2 as observed in real traffic traces for the two link directions. In the table, for each case, we report the maximum energy saving that can be achieved by manually varying T_c and N_c subject to an average delay below 1 ms (we denote with η'_{LPI} the energy saving factor that can be achieved subject to a given delay contraint). Additionally, we report the values T'_c and N'_c at which η'_{LPI} is maximized. Moreover, for the case without delay constraints but still the delay is below 1 ms, we fix the value of N_c to N''_c =100 packets (larger values do not improve the energy saving gain) and we check again the maximum value of the energy saving factor, which we denote as η''_{LPI}, achievable by varying T_c only. In the table, we report the value T''_c of the coalescing timer which maximizes the energy saving.

From Table I, we can observe that energy saving in the two cases is very close, so that we can think of fixing the size of the coalescing buffer and using an adaptive coalescing algorithm that, by adjusting the sole coalescing timer T_c, is able to achieve near optimal energy savings while keeping bounded the average delay of the packets due to coalescing. Noticeably, Table I also shows that small values of T_c are needed to achieve optimal (or near-optimal) performance figures, so that the optimal value of T_c can be searched in a small range. We next use the results of this section to design a novel adaptive coalescing algorithm based on run-time delay measurements.

IV. Measurement-Based Coalescing Control

Differently from existing approaches, we use analytical results on the sensitivity of D_i and η_{LPI} to make *run-time* educated decisions on how to adapt the coalescing parameters to meet a maximum target delay D_{target}.

The analysis tells that T_c and N_c behave qualitatively in a similar way. Specifically, fixing one of the two parameters limits the maximum achievable energy saving, although, by tuning the other parameter, it is possible to adjust the energy saving from zero to the maximum while increasing the delay monotonically. Therefore, to implement an adaptive coalescing algorithm, it is enough to fix one parameter between T_c and N_c to a sufficiently high value (which guarantees that near-maximal energy saving can be achieved), and adapt the remaining parameter.

The analysis also unveils that η_{LPI} and D_i values are more sensitive to T_c rather than toN_c. With the above consideration, jointly to the fact that N_c is limited to integer values, T_c results to be a better candidate for the fine tuning of energy and delay tradeoff when coalescing is adopted.

Therefore, we design an adaptive coalescing algorithm in which only T_c is adjusted. Moreover, in our algorithm, we implement a simple yet effective mechanism to detect when the coalescing is causing excessive delay and timely react. What we include in the algorithm is a low-pass filter to estimate the average coalescing delay D_i. When the link switches to state W, the dynamic timer algorithm tunes $T_c \in [T_c^{\min}, T_c^{\max}]$ based on the experienced (measured) average delay and D_{target}. The pseudocode of our heuristic is reported in Algorithm 1.

The analysis says that increasing T_c increases both η_{LPI} and delay at any load, so when the average delay is below or above the target, the algorithm increments or decrements the T_c value, respectively. The advantages of our approach are twofold: (i) given that T_c is tuned after exiting state C, the delay adaptation procedure is almost immediate (a few milliseconds), which allows to instantly react to changes in packet delay; (ii) our adaptive algorithm adapts quickly to any changes in traffic load simply by estimating the packet delay. Load variations occur very often in the daily patterns and so our simple T_c adaptation mechanism can produce great benefit for EEE.

With the above, we have defined not one but an entire class of delay-controlled MBCC algorithms, which differ in the way the value of T_c is tuned. For example, additive or multiplicative increases and decreases can be used. In the following, we simply use either (i) an additive increase/decrease approach with fixed step δ or (ii) an additive increase/multiplicative decrease approach with fixed additive step δ and multiplicative decrease percentage γ. We only focus on those two schemes because, first, multiple increase schemes provide less fairness than additive increase schemes and second, multiple increase schemes wildly oscillate and are a source of instability, thus leading to poor performance [11], [12]. Note that, due to the high sensitivity of delay and energy saving with respect to variations of T_c, the possible values of δ and γ have to be small enough to cause small changes in the adaptive timer.

Algorithm 1: MBCC: Adaptive Coalescing Timer

```
Input: run-time average estimate of delays $D_1$ and $D_2$
while $C \rightarrow W$ do
    if $(D_1\&\&D_2) \leq D_{target}$ then
        if $T_c < T_c^{\max}$ then
            $T_c = \max\{T_c + \delta, T_c^{\max}\}$
    else
        if $T_c > T_c^{\min}$ then
            $T_c = \max\{T_c - \delta, T_c^{\min}\}$ or
            $T_c = \max\{(1-\gamma)T_c, T_c^{\min}\}$
```

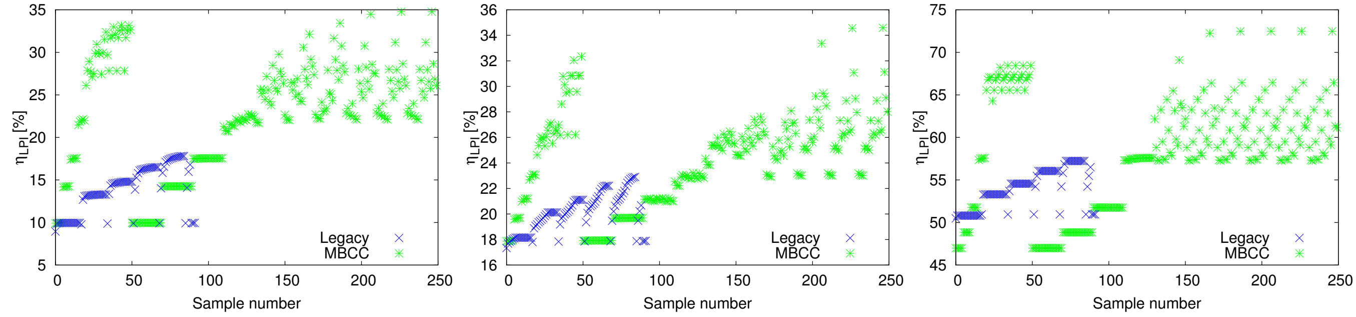
Note that, in gigabit EEE links, the two directions are correlated and therefore the algorithm has to run distributed over the two network cards at the edge of the link, although this requires only a few overhead messages to be transmitted from one card to the other to signal state transition events. However, such messages can be piggybacked by regular EEE state control messages, since each link edge just needs to send one bit to tell the other edge whether the measured delay is exceeding D_{target} or not.

V. Performance Evaluation

In this section we evaluate MBCC by implementing a delay-controlled adaptive coalescing timer algorithm. We benchmark MBCC against legacy static coalescing algorithms, for which it is known that dynamic adaptation based on coalescing events (such as buffer overflows or timer timeouts) does not improve performance if a target delay has to be guaranteed [6]. In the following, we first evaluate the achievable energy savings obtained by different configurations of legacy coalescing and MBCC for a set of representative traffic traces. Afterwards, we illustrate the behavior of energy savings and delays over time, when the load keeps changing. For our experiments, we use the real traffic traces we have been allowed to collect in Satec, a large web hosting center in Madrid, Spain.

A. Experimental setup

For our performance evaluation we monitored a typically low loaded link ($\rho_i \leq$10%) and a backbone link ($\rho_i \geq$10%) for a few minutes every one hour during a period of one year. Moreover, we modified the NS-3 simulator in order to (i) import the collected traces and (ii) simulate EEE with those traces as input, with both legacy coalescing and MBCC. Furthermore, we picked a few traffic traces, with loads



(a) $\rho_1 = 1.14\%$, $\rho_2 = 18\%$.

(b) $\rho_1 = 0.57\%$, $\rho_2 = 32.7\%$.

(c) $\rho_1 = 0.11\%$, $\rho_2 = 5.3\%$.

Fig. 3: Achievable energy saving for MBCC vs. Legacy coalescing with $D_{target} \leq 1\ ms$.

TABLE II: Legacy Coalescing: list of T_c and N_c combinations

Parameter	Value
T_c [μs]	200/500/700/1000/1200/**1300**/1400/1500/1700/2000
N_c [packets]	2/5/**10**/11/13/15/17/20/25/30/40/50/60/70/80/90/100

spanning from low to high, to further compare the achieved η_{LPI} using MBCC or legacy coalescing with bounded delay.

In our study, legacy coalescing schemes require the calibration of T_c and N_c based on the expected traffic characteristics or based on, e.g., the peak traffic. However, to guarantee low delay under low load conditions, both T_c and N_c have to be tuned to values well below the ones that achieve the best energy performance under medium or high traffic. In particular, since our criterion is to regulate the average coalescing delay of the packets crossing the EEE link under all traffic conditions, a (T_c, N_c) combination with small values has to be universally adopted to cope with the delay under scarce traffic conditions ($\sim$0.1% in the less loaded direction). Therefore, legacy coalescing has the disadvantage that it needs to be tuned on the *off-peak* traffic conditions. Apparently, so far, this has not been considered a great disadvantage for EEE links. In fact, energy savings are expected to be harvested only under low to medium traffic conditions. However, we argue that even though low loaded links represent about 40% of a data center links, there is still another 60% of the links from which additional energy savings could be potentially obtained.

To evaluate legacy coalescing, we test a range of values for T_c and N_c, as reported in Table II, under different traffic conditions. In contrast, for MBCC with our delay-controlled adaptive timer heuristic (Algorithm 1), we consider a fixed N_c value, such as the one selected based on the results reported in Table I (i.e., we could select the value $N_c = $100 packets), whereas the T_c value is automatically adapted according to the traffic. We assign D_{target} as initial value for the timer T_c. Other configuration parameters for MBCC are the adaptation coefficients δ and γ. The range of values for N_c, δ and γ can be read in Table III.

All tested parameters span over large intervals, to thoroughly explore their impact by means of our simulations.

B. Achievable energy saving

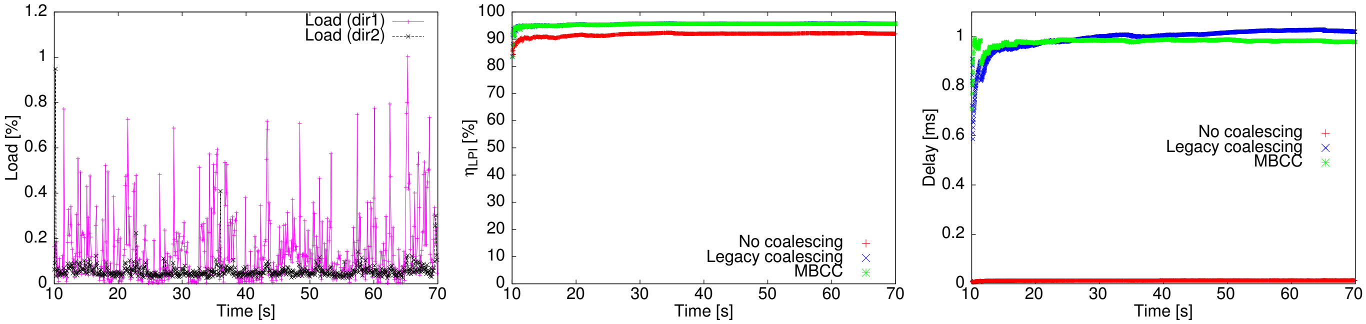
Here we compare legacy coalescing and MBCC under a variety of configuration choices, as reported in Tables II and III. In particular, we report our results for energy saving subject to average coalescing delay, D_{target}, not exceeding 1 ms. We think that this is a reasonable upper bound for the average delay in a point-to-point link. Indeed, according to [13] a connection between East and West coast in the US has at least four hops that create 28.4 ms of average delay. Thus, we consider that adding 1 ms due to the use of EEE in a data center connected to such a network is acceptable.

TABLE III: MBCC with Adaptive Coalescing Timer (Algorithm 1): list of parameters

Parameter	Value
δ [μs]	10/30/100/300/**1000**
γ [%]	**10**/25/50/75
N_c [packets]	2/5/10/20/50/75/100/200/500/**1000**

In Fig. 3 we plot η_{LPI} for three different load combinations (ρ_1, ρ_2). For legacy coalescing we run the simulation of a trace with a given combination (T_c, N_c) and we get the average delay of the packets in both directions and η_{LPI}. The delay can be higher or lower than D_{target} but we report the energy saving only for those combinations that give average coalescing delay below D_{target} (Table II). For MBCC, since it guarantees that the delay is below D_{target}, we report all points corresponding to all the tested combinations of parameters (Table III). Notably, the best results achieved with legacy coalescing in any of the depicted scenarios are very far from the best results of MBCC. Indeed, MBCC practically doubles the gain achieved by legacy coalescing.

Moreover, in our experiments we have observed that a particular combination performs best for legacy coalescing under any of the tested load combinations, i.e., (T_c=1300 μs, N_c =10 packets), reported in boldface in Table II. In contrast, for the case of MBCC, we have observed high variability in the configuration that achieves the best results in the various cases. In particular, considering that in each subfigure of Fig. 3 the first 50 samples for MBCC use only the parameter δ to adapt T_c (additive increase/decrease), while the remaining 200 samples use both δ and γ (additive increase, multiplicative decrease) for the adaptation of T_c, we can conclude that using both δ and γ is slightly more convenient. However, we have also observed that many configurations are equivalent, in particular when N_c is small (below 20), the performance is determined by N_c only, and changing δ and γ does not affect the results. In contrast, with higher N_c values δ and γ can be responsible for a fluctuation of 10-15% of energy saving. More in general,

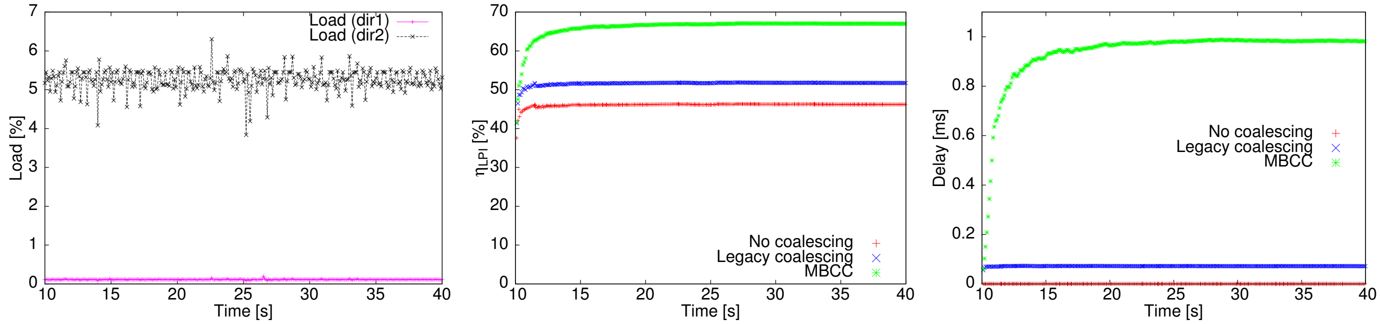


(a) Load.

(b) Energy saving (Legacy and MBCC overlap).

(c) Higher delay of the two directions.

Fig. 4: Low load ($\rho_1 = 0.2\%$, $\rho_2 = 0.06\%$). MBCC and legacy coalescing practically save the same amount of energy.

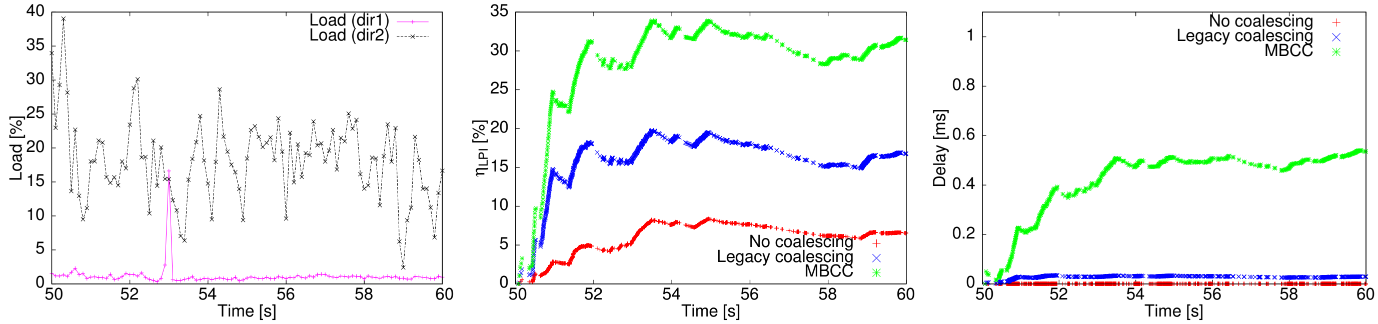


(a) Load.

(b) Energy saving.

(c) Higher delay of the two directions.

Fig. 5: Medium load ($\rho_1 = 0.11\%$, $\rho_2 = 5.3\%$). MBCC largely outperforms legacy coalescing at the expenses of delay (without exceeding the available delay budget).



(a) Load.

(b) Energy saving.

(c) Higher delay of the two directions.

Fig. 6: Highly variable load ($\rho_1 = 1.44\%$, $\rho_2 = 18.0\%$). MBCC doubles energy savings with respect to legacy coalescing.

our results indicate that bigger values of δ and N_c allow bigger energy saving. Instead, a bigger value of γ reduces the energy benefit. The topmost points in all the cases correspond to the combination (N_c = 1000 packets, δ = 1000 μs, γ = 10%), which is reported in boldface in Table III.

Now we select a near-optimal configuration for MBCC, and we compare its performance with the best configuration of the legacy coalescing scheme. We use an additive increase, additive decrease scheme with δ =100 μs, and N_c =100 packets for MBCC, and T_c = 1300 μs, and N_c =10 packets for legacy coalescing. With those configurations, in Figs. 4, 5, and 6 we plot the behavior over time of η_{LPI} and the higher of the two delays D_i for three different load combinations. In these figures, in addition to the performance MBCC and legacy coalescing, we also report the performance of EEE links without coalescing. Fig. 4 illustrates the case of low load. Specifically, as shown in Fig. 4a, the load in either link direction does not exceed 1%, and energy saving of 90-95% can be achieved with or without coalescing (see Fig. 4b). As concerns delay, Fig. 4c shows that coalescing introduces considerable delay with respect to the case of plain EEE without coalescing. However, the delay, D_{target}, is below 1 ms. The medium load case of Fig. 5 shows how MBCC manages to tradeoff delay for energy saving, while keeping the delay below 1 ms. Indeed, Fig. 5b shows the huge energy saving gain due to the delay-controlled coalescing operation of our proposal. In Fig. 6 we show a very dynamic case which combines high load with frequent and rapid load changes. We can still observe that our MBCC approach achieves a sevenfold gain with respect to plain EEE and a twofold gain with respect to legacy coalescing, while retaining the caused delay well below 1 ms. The impact of traffic variability is clear in the behavior of η_{LPI} and in the experienced delay.

Interestingly, the performance comparison shows that MBCC is able to maintain a constant gain over time with respect to the other schemes.

In conclusion, the energy benefit due to delay-aware MBCC is remarkable under any traffic condition, including under quickly variable traffic conditions. Configuring MBCC schemes is easy, since it only requires to make reasonably simple decisions on the maximum size of the coalescing buffer (in the order of 100 packets) and on the δ parameter (in the order of milliseconds). The γ parameter is optional and, if used, has to be chosen as a small factor (in the order of 10%).

C. Economical impact

The importance of EEE with MBCC can be seen in the following simple economical analysis. Let us consider a large data center, e.g., the one of OVH[2]. This data center contains 360,000 physical servers, and each server has on average 3 connected network ports [14]. Assuming that all network ports have gigabit links, each port may consume between 2 W and 13 W using legacy Ethernet [2]. Typical load distributions are $\sim$ 40% of the links at almost zero load ($\leq$ 0.1%), $\sim$ 40% between 0.1% and 10% of load, and the rest of the links operate at higher loads [3]. Therefore we can use the results of Figs. 4, 5 and 6 for an approximated economical analysis. Moreover, considering that the average cost of electricity in USA is about \$0.1/KWh, we can roughly estimate the cost of electricity for the network equipment of the servers of the aforementioned data center, using legacy Ethernet, plain EEE, EEE with legacy coalescing, or EEE with MBCC.

Thus, we will consider that on average an Ethernet card consumes 5 W and we further consider as averaged load values the ones we have in Figs. 4, 5 and 6. With our calculations, the annual electricity bill of data center servers just due to the network would be ~\$4.73M using legacy Ethernet. This amount could be reduced almost by half accounting to ~\$2.23M by adopting EEE. EEE with legacy coalescing could further deduct another ~\$133K from the bill and, finally, MBCC could allow to save another ~\$400K resulting in a final bill of ~\$1.7M per year. Therefore, the adoption of MBCC could potentially reduce the electricity cost of a data center by $\sim$ 65% if compared with legacy Ethernet and by $\sim$ 25% if compared with plain EEE. Practically, MBCC would quadruplicate the cost saving attainable with legacy coalescing.

In this simple estimate we exclude switches and other equipment such as air conditioning, CPU processing or server fans which could further contribute to the electricity cost reduction. Moreover faster Ethernet cards, i.e., 10, 40 and 100 $Gbps$, consume even more energy (at least two, three and five times more, respectively), so that the potential for energy saving is greater for higher data rates.

The cost of implementing coalescing is just adding a buffer to the NIC to support the packet aggregation but this might not be a problem since NICs have already integrated memory buffers and thus all we need is to reserve some space for coalescing. The cost of measurement-based coalescing control is negligible since it only requires software modifications on the driver side in order to apply the timer adaptation. Therefore, we believe that EEE with MBCC adjusting the coalescing timer is worth further research interest.

VI. Related Work

Since the standardization of 802.3az in 2010 a few works appeared in the literature that try to model its behavior and predict accurately the amount of energy saving and the experienced delay both for EEE and EEE with coalescing.

1) ***EEE modeling:*** There are works which model EEE with a good accuracy, although they do not consider the effect of coalescing. Among them, [15], [16], [17] are the most representative. In [15] the authors present an $M/G/1$ model for 1 $Gbps$ links with unidirectional traffic. In [16] a two state model is proposed for unidirectional traffic and transition times are assumed to be multiples of the frame transmission time. Bolla et *al.* [17] present a complete framework for EEE links, from 100 $Mbps$ to 10 $Gbps$, that takes into account the bidirectional nature of 1 $Gbps$ links.

As already discussed, packet coalescing techniques promise the largest energy saving gain, and thus analytical models exist to predict their behavior. The first work that showed the outperformance of EEE with coalescing over the legacy EEE is [7]. The authors analyze the energy consumption improvement of the link using a buffer of 10 or 100 packets at the cost of limited additional delay. In [18] the authors develop a $GI/G/1$ model which approximates the energy saving and the delay that the packets suffer due to coalescing. The authors of [19] develop a $D/D/1$ model to estimate the energy consumed and the corresponding average delay of the packets, although they evaluate their model only with synthetic Poisson traffic. Kim et *al.* [20] present a similar mathematical analysis and evaluation based on synthetic traffic but using an $M/G/1$ queueing system. In [21] Meng et *al.* show a markovian model for 10 gigabit links and only big 1500-byte packets which estimates the energy saving and the average maximum delay of the first packet. In all the above mentioned models, the dependency of EEE operations on the traffic in both link directions is neglected, so that they cannot be realistically used for gigabit links with coalescing.

Differently from other proposals mentioned above, in [6] the authors propose a model specifically designed for gigabit links with coalescing. The model is based on the correlated behavior of two $M/G/1$ queues. Using simple parameters such as average packet size and average load, the model is able to estimate the energy consumption and the coalescing delay when coalescing parameters are static. The results of [6] also show that static coalescing offers performance levels as high as dynamic coalescing algorithms. However, that paper does not consider the class of measurement-based algorithms for the control of coalescing parameters that we propose in this paper. Indeed, here we have extended the analytical results of [6] to show the advantages of MBCC schemes for dynamic coalescing.

[2] OVH.com presentation: http://www.youtube.com/watch?v=4e97g7_qSxA

*2) **Dynamic Coalescing:*** This research area, i.e., EEE with dynamic coalescing is very new and quite active. Google has recently patented a series of adaptive algorithms in [9] for 10 gigabit links. In particular they suggest a modified version of EEE in which states A and LPI have fixed intervals and those intervals can only be adapted by a term Δ based on the type of data traffic to be transmitted. In [10] the authors propose a dynamic coalescing queue algorithm that adapts the buffer size N_c according to the difference between an ideal energy proportional saving model and the one proposed in their paper, but it lacks of complete performance evaluation using different parameters for the dynamic queue part. Moreover the results they provide show that static coalescing outperforms or at least achieves results similar to the dynamic scheme. The authors in [6] proposed two dynamic coalescing algorithms that adapt the coalescing timer T_c and the coalescing buffer size N_c, respectively. The event that triggers the adaptation of the corresponding parameter is either the timer expiration, or the fill-up of the buffer, with no further considerations on the network performance. For instance, T_c is always increased after a timer expiration, while N_c is always incremented if the coalescing buffer N_c fills up. That paper studies various parameters for timer and buffer size increase and decrease and, differently from our new work, [6] concludes that dynamic schemes do not outperform legacy coalescing. In [22] the authors propose an adaptive scheme to adapt the duration of the coalescing timer for passive optical networks which adopt EEE, based on a neural network-based algorithm which optimizes the duration of the state LPI versus the Wake-Up time. However, this scheme does not consider delay, which is the key factor for the applicability of EEE.

VII. Conclusion

In this paper we have used sensitivity analysis to understand the impact of coalescing parameters, such as timer T_c and buffer size N_c, on the energy saving and the delay experienced over Energy Efficient Ethernet (EEE) links with coalescing. The analysis reveals that optimizing energy saving subject to delay constraints is possible by simply adapting T_c. Therefore, based on the coalescing properties analytically studied, we have designed MBCC, a class of adaptive coalescing algorithms which adapts T_c according to the delay sensed by the link. MBCC achieves dramatic gain with respect to legacy coalescing algorithms, for which dynamic adaptation has been proven unnecessary and unfruitful. Specifically, we validated the superiority of MBCC with real traffic traces collected in a large web hosting center, and we showed that our proposal can even double the energy saving benefit with respect to legacy coalescing schemes. Moreover, from a purely economical point of view, MBCC can reduce the electricity cost of a data center by 65%. Notably, if compared to EEE with legacy coalescing, MBCC would quadruplicate cost savings on large data centers' electricity bill.

Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness under the Ramon y Cajal Grant (ref: RYC-2014-01335), and under Grant TEC201455713-R (HyperAdapt).

References

[1] J. Arjona Aroca, A. Chatzipapas, A. Fernández Anta, and V. Mancuso, "A measurement-based analysis of the energy consumption of data center servers," in *ACM e-Energy '14*, Jun. 2014, pp. 63–74.

[2] R. Sohan, A. Rice, A. Moore, and K. Mansley, "Characterizing 10 Gbps network interface energy consumption," in *IEEE LCN 2010*, Oct. 2010.

[3] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 92–99, Jan. 2010.

[4] IEEE Std. 802.3az, "Energy Efficient Ethernet," 2010.

[5] P. Reviriego, K. Christensen, J. Rabanillo, and J. A. Maestro, "Initial evaluation of Energy Efficient Ethernet," *IEEE Communications Letters*, vol. 15, no. 5, pp. 578–580, May 2011.

[6] 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*, May 2013, pp. 161–172.

[7] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, "IEEE 802.3az: The road to Energy Efficient Ethernet," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 50–56, Nov. 2010.

[8] P. Reviriego, J. A. Maestro, J. A. Hernandez, and D. Larrabeiti, "Burst transmission for Energy Efficient Ethernet," *IEEE Computer Society*, vol. 14, no. 4, pp. 50–57, Jul. 2010.

[9] W.-C. Chang, W.-C. Lo, C.-S. Li, and M. Chang, "Adaptive pause time Energy Efficient Ethernet PHY," Jan. 2015, uS Patent 8,942,144.

[10] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernandez-Veiga, and C. López-García, "Bounded energy consumption with dynamic packet coalescing," in *IEEE NOC 2012*, Jun. 2012, pp. 1–5.

[11] V. Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM computer communication review*, vol. 18, no. 4, 1988, pp. 314–329.

[12] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN systems*, vol. 17, no. 1, pp. 1–14, 1989.

[13] B.-Y. Choi, S. Moon, Z.-L. Zhang, K. Papagiannaki, and C. Diot, "Analysis of point-to-point packet delay in an operational network," *Computer networks*, vol. 51, no. 13, pp. 3812–3827, Sep. 2007.

[14] S. Bapat, "The future of data centers (... and the stuff that goes in them)," in *1st Berkeley E3S Symposium*, Jun. 2009.

[15] M. Ajmone Marsan, A. Fernandez Anta, V. Mancuso, B. Rengarajan, P. Reviriego Vasallo, and G. Rizzo, "A simple analytical model for Energy Efficient Ethernet," *IEEE Communications Letters*, vol. 15, no. 7, pp. 773–775, Jun. 2011.

[16] D. Larrabeiti, P. Reviriego, J. A. Hernandez, J. A. Maestro, and M. Uruena, "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 JSAC*, vol. 32, no. 1, pp. 16–27, Jan. 2014.

[18] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "A GI/G/1 model for 10Gb/s Energy Efficient Ethernet links," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3386–3395, Nov. 2012.

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

[20] K. J. Kim, S. Jin, N. Tian, and B. D. Choi, "Mathematical analysis of burst transmission scheme for IEEE 802.3az Energy Efficient Ethernet," *Elsevier Performance Evaluation*, vol. 70, no. 5, pp. 350–363, May 2013.

[21] J. Meng, F. Ren, W. Jiang, and C. Lin, "Modeling and understanding burst transmission algorithms for Energy Efficient Ethernet," in *IEEE/ACM IWQoS 2013*. IEEE, 2013, pp. 1–10.

[22] S. Lee and K.-Y. Li, "Adaptive state transition control for energy-efficient gigabit-capable passive optical networks," *Photonic Network Communications*, Apr. 2015.