Citation for published version:

M. Rodríguez-Pérez, M. Fernández-Veiga, S. Herrería-Alonso, M. Hmila and C. López-García, "Optimum Traffic Allocation in Bundled Energy-Efficient Ethernet Links," in *IEEE Systems Journal*, vol. 12, no. 1, pp. 593-603, March 2018, doi: 10.1109/JSYST.2015.2466086.

Link to published version: https://ieeexplore.ieee.org/document/7243300

Optimum Traffic Allocation in Bundled Energy Efficient Ethernet Links

Miguel Rodríguez-Pérez[a,*], Manuel Fernández-Veiga[a], Meriame Hmila[a], Sergio Herrería-Alonso[a], Cándido López-García[a]

[a] *Dept. Telematics Engineering, University of Vigo, Spain.*

Abstract

The energy demands of Ethernet links have been an active focus of research in the recent years. This work has enabled a new generation of Energy Efficient Ethernet (EEE) interfaces able to adapt their power consumption to the actual traffic demands, thus yielding significant energy savings. With the energy consumption of single network connections being a solved problem, in this paper we focus on the energy demands of link aggregates that are commonly used to increase the capacity of a network connection. We build on known energy models of single EEE links to derive the energy demands of the whole aggregate as a function on how the traffic load is spread among its powered links. We then provide a practical method to share the load that minimizes overall energy consumption with controlled packet delay, and prove that it is valid for a wide range of EEE links. Finally, we validate our method with both synthetic and real traffic traces captured in Internet backbones.

Keywords: Network interfaces, Link aggregation, Optimization methods, Energy efficiency

1. Introduction

Energy consumption is nowadays a global source of concern for both economic and environmental reasons. Networking equipment alone consumes 1.8% of the world's electricity, and that number is currently increasing at a 10% rate annually [1]. If we focus just on data centers, between 15 and 20% of electricity is used for networking [2]. These reasons are spurring the development of more power efficient networking equipment.

A direct result of these efforts is the IEEE 802.3az standard [3] which provides a new *idle mode* for Ethernet physical interfaces. This new mode only needs a small fraction of the power used in normal operation, but no traffic can be transmitted nor received while the interface stays in the idle mode. Since there is an implicit trade-off between energy consumption and frame delay, these new Energy Efficient Ethernet (EEE) interfaces need a governor that decides when to enter and exit this idle mode. In fact, several alternatives have already been proposed in the literature [4–7] and have been later validated by both empirical [8, 9] and analytic means [10–13]. These works have provided us with the tools needed to accurately estimate the power savings of EEE for any arrival traffic pattern with the more prevalent idle mode governors and to properly tune them to maximize energy savings.

With the energy consumption problem of single Ethernet links mostly solved we focus in this paper on the power demands of network connections formed by multiple EEE links, either by link aggregation [14] or some other proprietary means. Despite the fact that the existence of EEE for saving energy in the individual components of the bundle, the global consumption of an aggregate may be severely affected by how the incoming traffic is shared among its powered-on links. This is because although EEE methods do significantly reduce energy consumption, the energy profile of EEE interfaces does not grow linearly with the traffic load.

So, in this paper we tackle this problem and derive an optimum method for assigning traffic to each link forming a bundle in a way that minimizes overall power consumption for common traffic arrival patterns. We have found that sharing traffic among the links in a sequential *water-filling* mode, where traffic is transmitted on a given link only when all the previous ones are already being used at their maximum capacity is the most power-efficient approach. We build on known power models of individual EEE interfaces to prove that sequential water filling gets optimum results for various relevant traffic arrival patterns. We then use this result to propose a practical method that minimizes the energy usage of the bundle. The method is tested empirically with the help of both synthetic and real world traffic traces. Our results show that additional energy reductions of up to 50% are attainable when the traffic is properly spread among the links.

The rest of this paper is organized as follows. Section 2

*Corresponding author. Address Telematics Engineering Dept., Univ. of Vigo, 36310 Vigo, Spain. Tel.:+34 986813459; fax:+34 986812116.

Email addresses: `miguel@det.uvigo.es` (Miguel Rodríguez-Pérez), `mveiga@det.uvigo.es` (Manuel Fernández-Veiga), `meriame@det.uvigo.es` (Meriame Hmila), `sha@det.uvigo.es` (Sergio Herrería-Alonso), `candido@det.uvigo.es` (Cándido López-García)

provides a formal description of the problem at hand. Section 3 analyzes the concavity of the cost function of the main EEE algorithms. Section 4 details a practical algorithm to implement water-filling. The results are commented in Section 5. Finally, Section 6 ends the paper with our conclusions.

2. Problem Description

In transmission networks, it is customary to bundle several homogeneous links, i.e., links with similar transmission technology, as a cheap way for scaling up the aggregate transmission rate between two endpoints. The bundle can be seen and managed either as a set of independent links or as a unit by the traffic management algorithms and the upper layer protocols. In the latter case, the traffic is split among the individual links in the bundle considering the optimization of a given performance metric. We focus in this paper on the optimum allocation of traffic when the bundle components are Energy Efficient Ethernet (EEE) links (IEEE 802.3az [3]), from the point of view of total energy consumption minimization. The profile of energy consumption in EEE links has been analyzed in many works [8–13], and has been shown to be highly sensitive to the statistical variability of the incoming traffic. Thus, further gains in energy efficiency may be realized if the total traffic load offered to the bundle is properly allocated to individual links, especially if links are not identical in their transmission rates or hardware devices.

We consider a bundle comprising N identical transmission links. The traffic demand to the bundle is X, and $E(x_i)$ is the energy consumption of link $i = 1, \dots, N$, where x_i stands for the traffic rate in that link. Link capacities are denoted by C_i, for $i = 1, \dots, N$..

Our goal is to minimize the overall consumption of the bundle

$$E(x_1, \dots, x_N) = \sum_{i=1}^{N} E(x_i) \text{ s.t. } C \geq x_i \geq 0, \quad \sum_{i=1}^{N} x_i = X \tag{1}$$

where

$$E(x_i) = 1 - (1 - \sigma_{\text{off}})(1 - \rho_i) \frac{\mathrm{T}_{\text{off}}(\rho_i)}{\mathrm{T}_{\text{off}}(\rho_i) + \mathrm{T}_{\text{s}} + \mathrm{T}_{\text{w}}} \tag{2}$$

is the normalized energy consumption of link i, as shown in [13]. In (2), T_s and T_w are, respectively, the transition times needed to enter and exit the idle mode, σ_{off} is the fraction of energy consumed by the interface in the idle state compared to its energy consumption in the active state, $\rho_i = x_i/C$ is the normalized traffic load on the link and $\mathrm{T}_{\text{off}}(\rho_i)$ is the function that gives the average time spent by the interface in the idle state for a given input load. Note that $\mathrm{T}_{\text{off}}(\rho_i)$ depends on both the actual traffic arrival pattern and the idle state governor. So, (1) is a standard minimization problem amenable to analysis provided that $E(x_1, \dots, x_N)$ is a well-behaved function.

2.1. Optimum allocation

In this Subsection, we prove that for certain functions $\mathrm{T}_{\text{off}}(\cdot)$ the solution to the optimum allocation is a simple sequential water-filling algorithm: each link capacity is fully used before sending traffic through a new, idle link. Clearly, (1) is a concave separable optimization problem when the objective function is concave and we have the following simple result.

Proposition 1. *If $E(x_i)$, $i = 1, \dots, N$, is a strictly concave function, then there exists an ordering of links $\sigma(1), \dots, \sigma(N)$ such that $E(x_1, \dots, x_N)$ is minimum for $x_{\sigma(i)} = \min\{C_{\sigma(i)}, X - \sum_{j<i} x_{\sigma(j)}\}$.*

Proof. The proof is a direct consequence of the subadditivity of the $E(\cdot)$ function and is given in Appendix A. □

Now we derive sufficient conditions for the concavity of the cost function $E(\cdot)$. Recall from (2) that $E(\cdot)$ depends on some constants related to the interface hardware and the statistical variability of the incoming traffic. We will try to understand what conditions must satisfy T_{off}, which is the only traffic-dependent term. For clarity and simplicity, in the following we use the notations $f(\rho) = \mathrm{T}_{\text{off}}(\rho)$ and $t(\rho) = E(\rho)$. We will further assume that $f(\rho)$ is decreasing[1] and continuously differentiable in $\rho \in (0, 1)$.

Proposition 2. *Let $f(x)$ be a function $f : [0, 1] \to \mathbb{R}^+$, decreasing and with continuous derivatives. Let $a, b > 0$ and consider the function*

$$t(x) = 1 - a(1 - x) \frac{f(x)}{f(x) + b}. \tag{3}$$

Under these definitions, $t(x)$ is concave if

$$f''(x)(f(x) + b) \geq 2(f'(x))^2. \tag{4}$$

Proof. The proof is provided in Appendix B. □

Proposition 2 applies trivially to the functions $E(\cdot)$ setting $a = 1 - \sigma_{\text{off}}$ and $b = \mathrm{T}_\text{s} + \mathrm{T}_\text{w}$, and then we have derived a simple sufficient condition for the $\mathrm{T}_{\text{off}}(\cdot)$ term that makes $E(\cdot)$ concave and the optimization problem easily solvable.

3. Analysis of Frame and Burst Transmission

In this Section we check whether the known formulas for the average sleeping time in EEE satisfy the condition of Proposition 2. According to [13] the time $\mathrm{T}_{\text{off}}(\cdot)$ depends both on the incoming traffic characteristics and the threshold algorithm used to switch between the idle and the active states in the Ethernet interface. There are two main approaches, the *frame transmission* algorithm and the *burst transmission* one, that we consider next.

[1] $f(\rho)$ computes the average time spent by the interface in the idle state, so it is reasonable to assume it is decreasing when the traffic load is higher.

3.1. Frame Transmission

Frame transmission is a straightforward use of the idle mode. Under frame transmission, the physical interface is put in idle mode as soon as the last frame in the queue has been transmitted, and normal operation is restored as soon as new traffic arrives at the networking interface. For many common traffic patterns this operating mode does not produce great energy savings, as there is a transition period every time the interface changes its operating mode that draws some energy. From [13], for the frame transmission algorithm

$$\mathrm{T}_{\mathrm{off}}^{\mathrm{frame}}(\rho) = \int_{\mathrm{T_s}}^{\infty} (t - \mathrm{T_s}) f_{\rho,\mathrm{T_e}}(t)\mathrm{d}t, \tag{5}$$

where $f_{\rho,\mathrm{T_e}}(t)$ denotes the probability density function for traffic load ρ of the empty period, i.e., the time elapsed since the queue empties until the subsequent first arrival. When $f_{\rho,\mathrm{T_e}}(t)$ is unknown, equation (5) can be approximated by

$$\mathrm{T}_{\mathrm{off}}^{\mathrm{frame}}(\rho) \approx \left(\frac{1}{\mu\rho} - \mathrm{T_s}\right)^{+} \tag{6}$$

with μ^{-1} the average packet transmission duration. Closed formulas exist when the arrival process follows a Poisson or a deterministic distribution. In particular, for Poisson arrivals, we have

$$\mathrm{T}_{\mathrm{off}}^{\mathrm{frame}}(\rho) = \frac{\mathrm{e}^{-\mu\rho\mathrm{T_s}}}{\mu\rho}. \tag{7}$$

1) Poisson traffic: For proving the concavity under the assumption of Poisson arrivals, we start by noting that $f(\rho) = \mathrm{e}^{-\mu\rho\mathrm{T_s}}/(\mu\rho)$ and substitute this in (4) with $b = \mathrm{T_s} + \mathrm{T_w}$. The result is the condition

$$\left(\frac{\mu\mathrm{T_s^2}\mathrm{e}^{-\mu\rho-\mathrm{T_s}}}{\rho} + \frac{2\mathrm{e}^{-\mu\rho-\mathrm{T_s}}}{\mu\rho^3} + \frac{2\mathrm{T_s}\mathrm{e}^{-\mu\rho\mathrm{T_s}}}{\rho^2}\right)\cdot \left(\frac{\mathrm{e}^{-\mu\rho\mathrm{T_s}}}{\mu\rho} + \mathrm{T_s} + \mathrm{T_w}\right) > -2\left(\frac{\mathrm{e}^{-\mu\rho\mathrm{T_s}}}{\mu\rho^2} + \frac{\mathrm{T_s}\mathrm{e}^{-\mu\rho\mathrm{T_s}}}{\rho}\right)^2, \tag{8}$$

and after some routine simplifications this reduces to

$$(\mathrm{T_s}+\mathrm{T_w})\mathrm{e}^{\mu\rho\mathrm{T_s}}(2+\mu\rho\mathrm{T_s}(2+\mu\rho\mathrm{T_s})) > \mathrm{T_s}(2+\mu\rho\mathrm{T_s}). \tag{9}$$

But $\mu\rho\mathrm{T_s} > 0$ and $\mathrm{e}^{\mu\rho\mathrm{T_s}} > 1$, so

$$\begin{gathered}(\mathrm{T_s}+\mathrm{T_w})\mathrm{e}^{\mu\rho\mathrm{T_s}}(2+\mu\rho\mathrm{T_s}(2+\mu\rho\mathrm{T_s})) > \\ \mathrm{T_s}\mathrm{e}^{\mu\rho\mathrm{T_s}}(2+\mu\rho\mathrm{T_s}(2+\mu\rho\mathrm{T_s})) > \mathrm{T_s}(2+\mu\rho\mathrm{T_s}),\end{gathered} \tag{10}$$

and (4) is satisfied.

Note that it is important to ascertain that the link consumption function $E(\cdot)$ is concave for Poisson traffic since, notwithstanding that Poissonian models are not generally suitable, they are reasonably valid for real traffic in subsecond timescales [15] and also for aggregated traffic in the Internet core [16]. In any case, in Section 5 we test the validity of our assumptions with both synthetic and real traffic traces collected in Internet links.

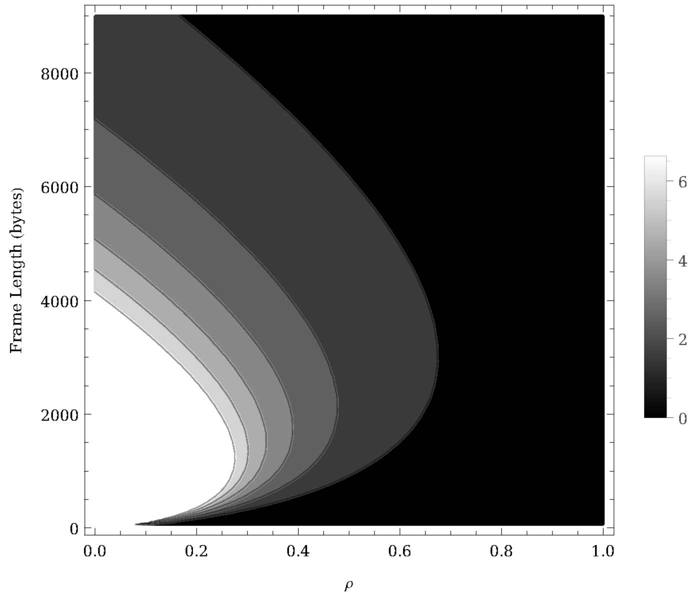


Figure 1: Contour plot of $h''(\rho)$ when $h(\rho) = \frac{\mathrm{T}_{\mathrm{off}}^{\mathrm{frame}}(\rho)}{\mathrm{T}_{\mathrm{off}}^{\mathrm{frame}}(\rho)+\mathrm{T_s}+\mathrm{T_w}}$ for a Poisson arrival process under the frame transmission energy-saving algorithm.

2) General traffic distributions: For unknown traffic distributions we must resort to the approximation given by (6), so we let $f(\rho) = 1/(\mu\rho) - \mathrm{T_s}$ and $b = \mathrm{T_s} + \mathrm{T_w}$. Now we can immediately substitute in $f''(\rho)(f(\rho) + b) > 2(f'(\rho))^2$ and get

$$\frac{2(\frac{1}{\mu\rho} + \mathrm{T_w})}{\mu\rho^3} > \frac{2}{\mu^2\rho^4}. \tag{11}$$

After some straightforward cancellations, this is

$$\frac{2\mathrm{T_w}}{\mu\rho^3} > 0 \tag{12}$$

which is obviously true.

Figure 1 shows for purposes of illustration a contour plot of $h''(\rho)$ for the function $h(\rho) = \mathrm{T}_{\mathrm{off}}^{\mathrm{frame}}(\rho)/(\mathrm{T}_{\mathrm{off}}^{\mathrm{frame}}(\rho) + \mathrm{T_s} + \mathrm{T_w})$. The traffic is Poissonian and the Ethernet link runs at 10 Gb/s ($b = \mathrm{T_s} + \mathrm{T_w} = 2.28\ \mu\mathrm{s} + 4.48\ \mu\mathrm{s}$, as mandated by the IEEE 802.3az standard [3]), with packet sizes between 64 and 9 000 bytes.

3.2. Burst Transmission

Burst transmission is a simple modification of frame transmission that waits until a given number of packets $\mathrm{Q_w}$ arrive at the network interface before exiting idle mode. To avoid excessive delays, there is a tunable parameter $\mathrm{T_{max}}$ that limits the wait for the $\mathrm{Q_w}$-th frame since the first frame arrives. The analysis of the burst transmission algorithm is more involved, for the reason that there is not one but two operating regimes depending on the traffic load. Fortunately, the two operating regimes (low and high traffic load, respectively) can be neatly separated by the approximate traffic threshold

$$\rho^* \approx \frac{\mathrm{Q_w} - 1}{\mu \mathrm{T_{max}}} \tag{13}$$

where $\mathrm{Q_w}$ and $\mathrm{T_{max}}$ are the tunable parameters in the burst transmission algorithm [11]. As in the previous Section, we will proceed and check whether, with burst transmission, the link energy consumption function is concave.

1) Low load regime, $\rho < \rho^*$: When the traffic load is low, the interface exits the low power mode before a backlog of $\mathrm{Q_w}$ packets accumulates at the queue due to the timer expiry after waiting for $\mathrm{T_{max}}$ seconds. The exact expression for the expected sojourn time in the low-power state is (see [13])

$$\mathrm{T_{off}^{burst,\ low}}(\rho) = \int_0^\infty (t + \mathrm{T_{max}} - \mathrm{T_s}) f_{\rho,\mathrm{T_e}}(t)\mathrm{d}t. \tag{14}$$

When $f_{\rho,T_e}(t)$ is unknown, (14) can be approximated by

$$\mathrm{T_{off}^{burst,\ low}}(\rho) \approx \frac{1}{\mu\rho} + \mathrm{T_{max}} - \mathrm{T_s}. \tag{15}$$

As in the frame transmission algorithm, there exist closed expressions for $\mathrm{T_{off}^{burst,\ low}}(\rho)$ for some distributions, and remarkably (15) is exact with Poissonian arrivals.

Proving the concavity of $E(\rho)$ in this case is direct. First, note that $f(\rho) = \mathrm{T_{off}^{burst,\ low}}(\rho) = \mathrm{T_{off}^{frame}}(\rho) + \mathrm{T_{max}}$, so that the derivatives f' and f'' are the same as in the frame transmission case, and hence plugging (15) into the condition $f''(x)(f(x)+b) > 2(f'(x))^2$ one can easily check that the inequality holds.

2) High load regime, $\rho > \rho^*$: When the traffic load is high, the packet burst is much more likely to reach its maximum size $\mathrm{Q_w}$ before the timer expires. Now, the expected sojourn time in the low-power state is given by

$$\mathrm{T_{off}^{burst,high}}(\rho) = \int_{\mathrm{T_s}}^\infty (t - \mathrm{T_s}) f_{\rho,\mathrm{Q_w}}(t)\mathrm{d}t, \tag{16}$$

where, as usual, $f_{\rho,\mathrm{Q_w}}(t)$ is the probability density function of the $\mathrm{Q_w}$-th frame arrival epoch after the interface has entered the idle mode. When the density is unknown, the expected time can be well approximated by

$$\mathrm{T_{off}^{burst,high}}(\rho) \approx \frac{\mathrm{Q_w}}{\mu\rho} - \mathrm{T_s} \tag{17}$$

whereas the exact formula for the case of Poissonian arrivals is

$$\mathrm{T_{off}^{burst,high}}(\rho) = \frac{\Gamma(\mathrm{Q_w}+1, \mu\rho\mathrm{T_s}) - \mu\rho\mathrm{T_s}\Gamma(\mathrm{Q_w}, \mu\rho T_s)}{\mu\rho\Gamma(\mathrm{Q_w})}. \tag{18}$$

Here, $\Gamma(\cdot)$ and $\Gamma(\cdot,\cdot)$ are the complete and incomplete Gamma functions, respectively [17].

In order to prove that Poissonian arrivals lead to concave energy consumption functions, simply substitute (18) into (4) to obtain after some tedious calculations the inequality

$$\begin{aligned}\mu\rho\Gamma(\mathrm{Q_w})^2 \mathrm{e}^{\mu\rho\mathrm{T_s}} \Big(&\mathrm{T_s}(\mu\rho\mathrm{T_s})^{\mathrm{Q_w}}((\mathrm{Q_w} - \mu\rho\mathrm{T_s})\Gamma(\mathrm{Q_w}, \mathrm{T_s}) + \\ &\mu\rho(\mathrm{T_s}+\mathrm{T_w})\Gamma(\mathrm{Q_w})) + \\ &2\mathrm{e}^{\mu\rho\mathrm{T_s}}\Gamma(\mathrm{Q_w}+1, \mu\rho\mathrm{T_s})((\mathrm{T_s}+\mathrm{T_w})\Gamma(\mathrm{Q_w}) \\ &-\mathrm{T_s}\Gamma(\mathrm{Q_w}, \mu\rho\mathrm{T_s}))\Big) > 0.\end{aligned} \tag{19}$$

All the constant terms appearing in the above inequality are positive, so this simplifies somewhat to

$$\begin{aligned}&\mathrm{T_s}(\mu\rho\mathrm{T_s})^{\mathrm{Q_w}} \Big(\mu\rho((\mathrm{T_s}+\mathrm{T_w})\Gamma(\mathrm{Q_w}) - \mathrm{T_s}\Gamma(\mathrm{Q_w}, \mu\rho\mathrm{T_s})) + \\ &\Gamma(\mathrm{Q_w}+1, \mu\rho\mathrm{T_s})\Big) + \Gamma(\mathrm{Q_w}+1, \mathrm{T_s}\mu\rho)((\mathrm{T_s}+\mathrm{T_w})\Gamma(\mathrm{Q_w}) - \\ &\mathrm{T_s}\Gamma(\mathrm{Q_w}, T_s\mu\rho)) > 0\end{aligned} \tag{20}$$

which holds true because

$$\mu\rho(\mathrm{T_s}+\mathrm{T_w})\Gamma(\mathrm{Q_w}) > \mu\rho\mathrm{T_s}\Gamma(\mathrm{Q_w}) > \mu\rho\mathrm{T_s}\Gamma(\mathrm{Q_w}, \mu\rho\mathrm{T_s}) \tag{21}$$

as a consequence of elementary properties of the Gamma functions. This implies that all the summands in the left side of (20) are positive, and (4) is satisfied.

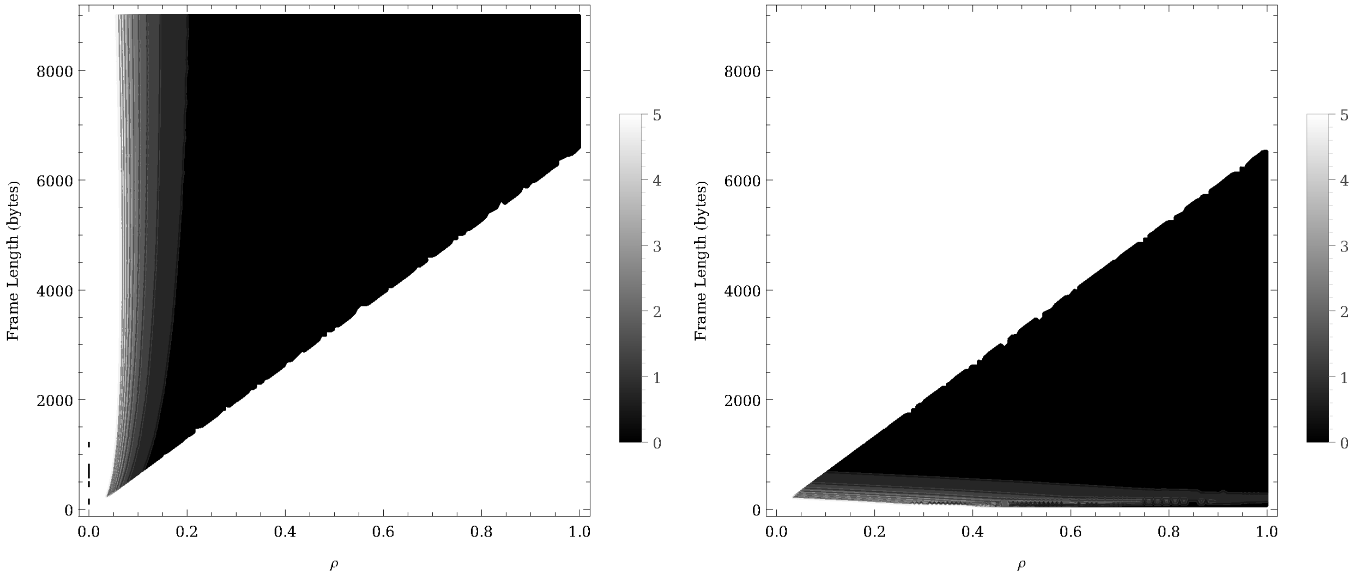
The last step is to prove concavity for the general approximation (17). A change of variable $m = \mathrm{Q_w}/\rho$ transforms (17) into (6) formally. Since $m > 0$, following the same steps as in frame transmission, one concludes that (17) also fulfills condition (4). Hence, the link energy consumption function $E(\cdot)$ is concave with burst transmission in the high-load regime, regardless the traffic arrival pattern.

A numerical illustration of the concavity is shown in Figure 2, which depicts the contour plots of $h''(\rho)$ for a 10 Gb/s Ethernet link as the traffic load and the packet size vary.

4. Delay Control

According to the previous sections, a straightforward application of a water-filling algorithm to share traffic among the bundle links provides maximum energy savings. However, if proper care is not taken, packet delay can grow uncontrolled as we explain next.

From a practical point of view there are many ways to implement a water filling algorithm. For instance, one could use separate queues for each link and only divert traffic to new links when the queue of the previous one overflows. Obviously, this approach exhibits the greatest delay. A second option is to limit the load factor in every link, and thus the delay, and divert traffic when this factor is reached. Its main drawback is that no link is used at its full capacity and so the energy savings are not maximum.



(a) Low load regime

(b) High-load regime

Figure 2: Contour plots of $h''(\rho)$ when $h(\rho) = \frac{\mathrm{T}_{\mathrm{off}}^{\mathrm{burst}}(\rho)}{\mathrm{T}_{\mathrm{off}}^{\mathrm{burst}}(\rho)+\mathrm{T}_{\mathrm{s}}+\mathrm{T}_{\mathrm{w}}}$ for a Poisson arrival process under the burst transmission energy-saving algorithm.

Another option, in the opposite extreme, is to have a common bundle queue and zero-length queues at the links. In this case, a new link is used if when a packet arrives, the previous link is busy transmitting a packet. The problem is that if the traffic load is not high enough, we will find that the first link is idle while the second one is transmitting, and that goes against the idea of the water-filling algorithm.

We propose a simple dynamic water-filling algorithm that can control average delay, while keeping the utilization factor of the links close to 1. The algorithm has one configuration parameter, the expected delay (d_e) and $N+1$ state variables, with N the number of links in the bundle, as it just keeps a record of the short term average delay (d_{av}), calculated with an exponentially weighted moving average, and the current queue length in each link measured in time units (q_i, $i = 1 \dots N$).

The algorithm works as follows. Each link in the bundle is assumed to have its own queue, so whenever a new packet arrives, the algorithm decides which queue should store it. For this the expected delay is compared with the current average delay. If $d_{av} < d_e$, the packet is stored in the queue of the first link. For every other case, a sequential search is started for a queue with a queue length smaller than the expected delay. If no queue is found, the packet is stored in the last queue. This is all summarized in Listing 1.

```
function packet_arrival
    if (d_av < d_e)
        return enqueue(link(1))

    foreach (l in Links)
        if (q_l < d_e)
            return enqueue(l)

    return enqueue(N)
```

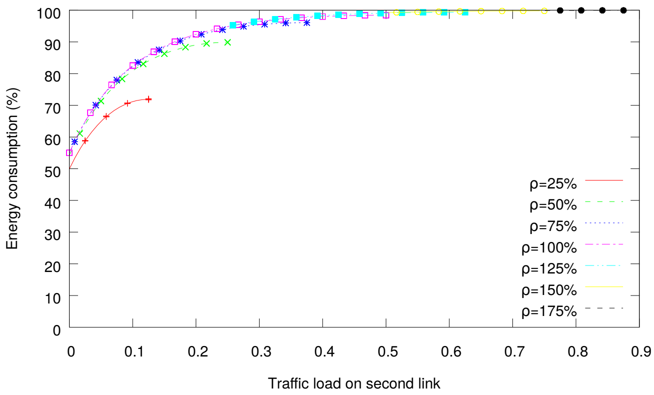
Listing 1: Dynamic water-filling algorithm.

5. Results

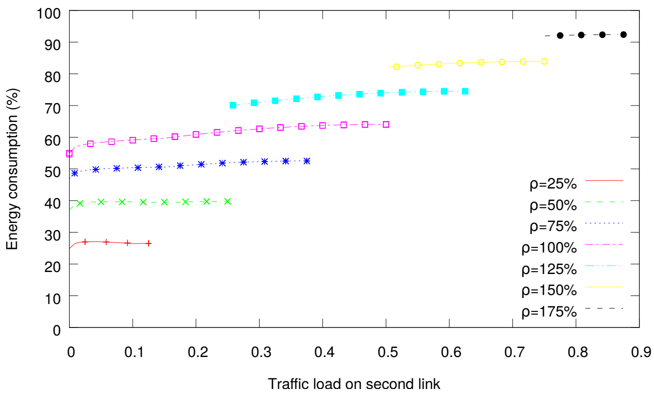
We have carried out several experiments to assess the effectiveness of our proposed sharing strategy. We have employed the ns-2 network simulator with an added module for simulating IEEE 802.3az links available for download at [18]. The simulated bundles have a varying number of 10 Gb/s links with 10GBASE-T interfaces, so $\mathrm{T_s} = 2.88\,\mu$s, $\mathrm{T_w} = 4.48\,\mu$s and $\sigma_{\mathrm{off}} = 0.1$, in accordance with several estimates provided by different manufactures during the standardization process of the IEEE 802.3az standard. For the *burst transmission* simulations we set up $\mathrm{T_{max}} = 100\,\mu$s and $\mathrm{Q_w} = 20$ frames, so that $\mu\mathrm{T_s} > 3.6$ frames, as recommended in [13].

5.1. Model Validation

The first set of experiments tests all possible traffic sharing alternatives in a simple 2-link bundle when it is fed with synthetic traffic. For the experiments we used a fixed frame size of 1 000 bytes and a varying arrival rate, so that the aggregated load ranged between 25 and 175%. Then, for each load we modified the share between the two links and, for each share, we run five simulations with different random seeds and a ten seconds duration.

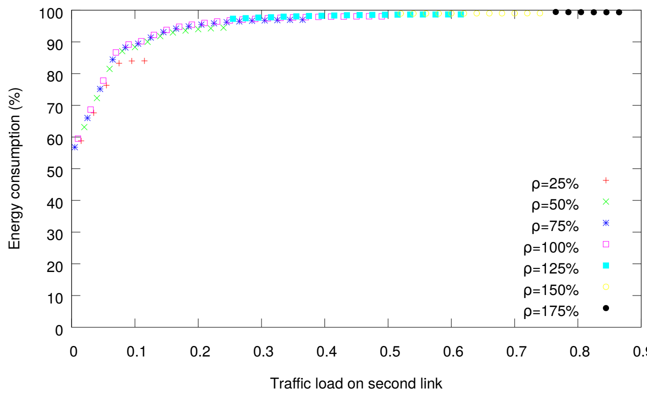


(a) Frame transmission

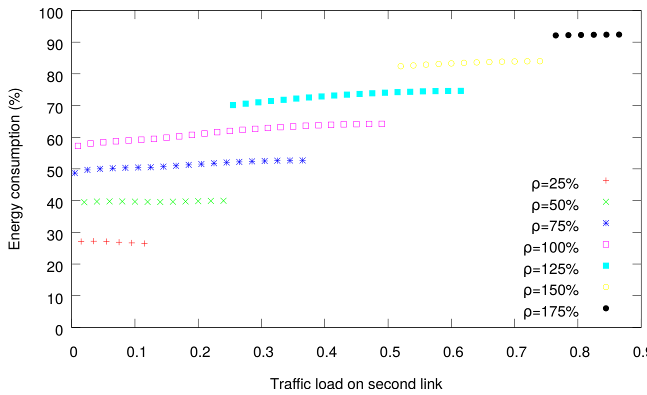


(b) Burst transmission

Figure 3: Results for a 2-link bundle with Poisson traffic as a function of excess traffic load on the second link.



(a) Frame transmission



(b) Burst transmission

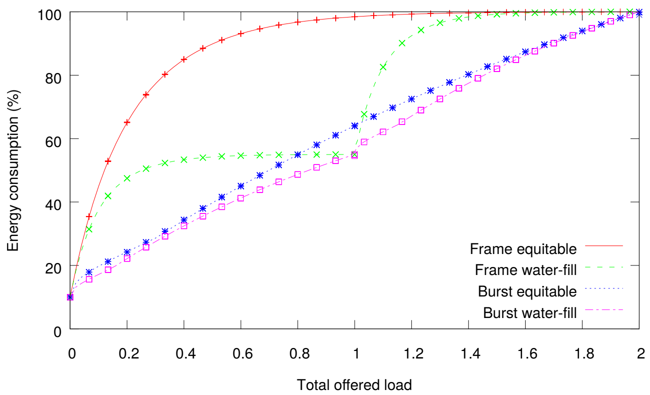
Figure 4: Results for a 2-link bundle with Pareto traffic as a function of excess traffic load on the second link.

Figure 3 shows the total energy consumption of the bundle versus the traffic load on the second link for Poisson traffic with both the frame and burst transmission algorithms. For clarity, we take advantage of the symmetry of the problem and only represent the results where load on the second link is smaller than that on the first. Figure 3 shows very clearly that there is very little variance among the different simulations for the same share and load and, at the same time, that the results match those provided by the model, plotted with continuous lines in the graph. It is also easy to see the increasing energy consumption with the traffic load on the second link. The closer the loads of both links are, the higher the energy needs. In fact, the minimum consumption is obtained when most load is concentrated on a single link, as predicted. Finally, we also observe that the benefit of aggregating load on a single link is much greater for frame than for burst transmission. This is a consequence of the fact that the energy profile of the burst transmission algorithm is more linear. Also, as expected, burst transmission needs less energy than frame transmission, although the difference is small when the water-fill share is used.

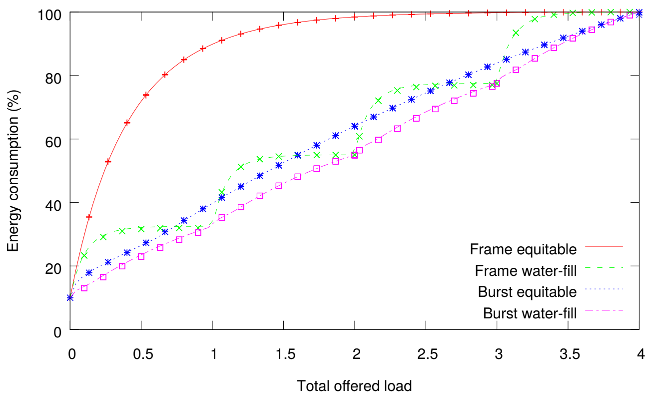
The results for Pareto traffic (with the shape factor α set to 2.5)[2] are plotted in Figure 4. Although the performance curves are not as smooth as for the Poisson traffic, the previous conclusions still hold. Again, the minimum consumption is obtained when most of the traffic is on a single link and then increases as the traffic on the second link increases. At the same time, the frame transmission algorithm benefits more than the burst transmission one.

Our second experiment compares the overall energy consumption of an Ethernet bundle for the full range of possible incoming traffic demands and two different sharing methods. The first, equitable share, spreads the traffic evenly across all the constituent links, while the second is the naïve water-filling method. Traffic follows a Poisson distribution and the frame size is 1000 bytes, as in the previous experiment. Figure 5 displays both the experimental and analytic results for two, four and eight-link aggregates. Again, frame transmission algorithm benefits more than burst transmission of the water-fill sharing algorithm. Further, as the number of links in the bundle in-

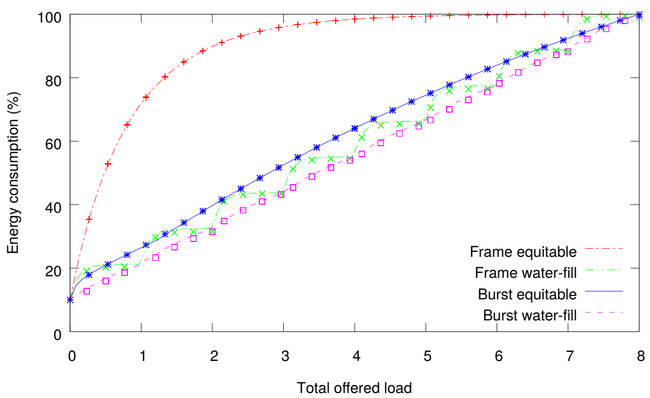
[2] Pareto distributions must be characterized with a shape parameter α greater than 2 to have a finite variance. However, the greater the α parameter is, the shorter the fluctuations, so a value of 2.5 is a good compromise to have finite variance along with significant fluctuations.



(a) Two links bundle



(b) Four links bundle



(c) Eight links bundle

Figure 5: Normalized global consumption of a bundle link for the different idle mode governors.

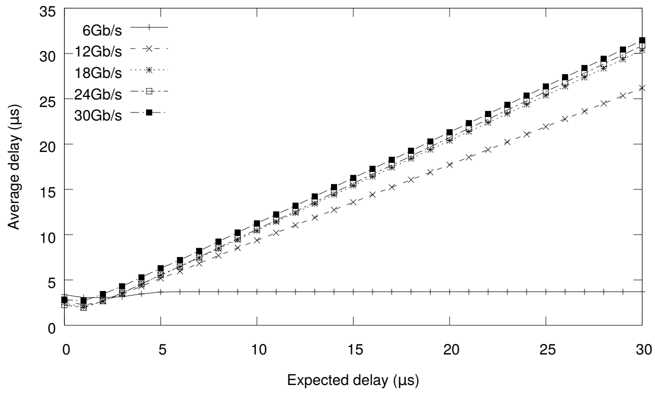


Figure 6: Obtained average delay versus configured delay for the dynamic water-fill algorithm.

creases, the energy demands of frame transmission, when using the water-fill procedure, approximate those of burst transmission.

5.2. Dynamic Water-filling Algorithm

The next set of experiments tests the behavior of the dynamic water-filling algorithm. We have employed real traffic traces captured on Internet backbones for the simulations. The traffic comes from the publicly available passive monitoring CAIDA dataset from 2013 [19] which provides anonymized traces from a 10 Gb/s backbone. We used one of these traces to feed traffic to a simulated 4-link bundle made of 10GBASE-T interfaces. Of all the available traces, we have chosen one with a relatively high demand of about 6 Gb/s on average. As that load is quite low for our simulated bundle of 40 Gb/s we made new traces of approximately 12, 18, 24 and 30 Gb/s combining traffic from independent adjacent traces.

The first experiment verifies that the algorithm is in fact able to control the average delay. For this we have fed all the traffic traces to a 4-link bundle, and configured the algorithm for different expected delays. The results are plotted in Figure 6.[3] In can be clearly seen an almost perfect relationship between the expected and the measured average delay for values greater than the transition times of the EEE links. The exception is the 6 Gb/s trace, that is bounded below 4 μs. This is expected, as the queue cannot grow larger when the capacity of a single link is greater than the offered traffic. The simulation with the 12 Gb/s trace shows a small drift of the average delay, but, in any case, the average delay is kept below the expected delay.

The second experiment shows the variation of power consumption versus expected delay. The results are shown in Figure 7. When the expected delay is too low, all links are used simultaneously, and the power savings are minimal. However, as the allowed delay increases, most of the

[3]Results for burst transmission have been omitted for the sake of brevity, but show a similar behavior.

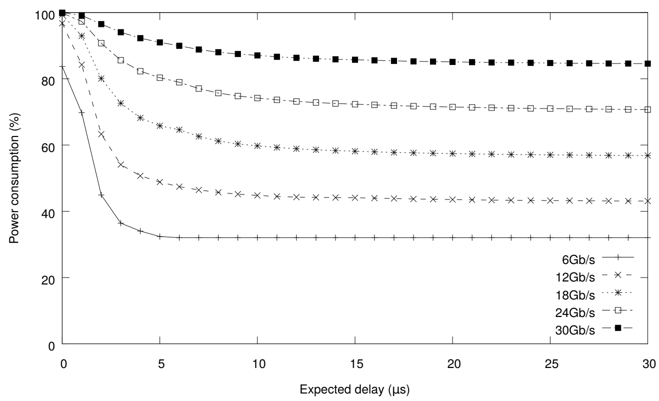


Figure 7: Power consumption vesus expected delay for the dynamic water-fill algorithm.

Bundle	Strategy	Link #1	Link #2	Link #3	Link #4
	Equit.	1.55	1.55	1.55	1.55
6.21	Naïve Water-fill	6.21	0	0	0
	Dyn. Frame	6.21	0	0	0
	Dyn. Burst	6.18	0.03	0	0
	Equit.	3.15	3.15	3.15	3.15
12.60	Naïve Water-fill	9	3.60	0	0
	Dyn. Frame	9.44	3.08	0.07	0
	Dyn. Burst	9.50	2.67	0.39	0.04
	Equit.	4.71	4.70	4.70	4.70
18.81	Naïve Water-fill	9	9	0.81	0
	Dyn. Frame	9.94	7.12	1.73	0.02
	Dyn. Burst	9.97	6.82	1.77	0.25
	Equit.	6.27	6.27	6.27	6.27
25.08	Naïve Water-fill	9	9	7.08	0
	Dyn. Frame	10	9.16	5.12	0.80
	Dyn. Burst	10	9.13	4.78	1.17
	Equit.	7.85	7.85	7.85	7.85
31.40	Naïve Water-fill	9	9	9	4.4
	Dyn. Frame	10	9.82	7.94	3.64
	Dyn. Burst	10	9.83	7.74	3.83

Table 1: Average traffic fed into each link for the real traffic simulations (in Gb/s).

traffic is transmitted by the first links and, despite the fact that all of them are powered on, we achieve large power savings thanks to the concavity of the cost function. It is important to notice that the maximum energy savings are already obtained starting from low delay target values. This allows to deploy the algorithm even in networks used by delay-sensitive applications.

In the last experiment we have compared the results obtained when sharing the traffic with three different strategies: spreading the traffic evenly across the four links, that we called *equitable*, a naïve implementation of the *water-fill* algorithm and, finally, the *dynamic* water-fill algorithm with a target delay of ten microseconds for the frame transmission algorithm and 20 μs for the burst one.[4] For the naïve implementation we have constrained the traffic load on any link to 90% to avoid excessive buffering.

The exact traffic rate of each trace and the different shares are detailed in Table 1. For the equitable and the naïve water-fill they have been determined beforehand, but for the dynamic algorithm the table lists the results obtained via simulation.

The results for both the frame transmission and the burst transmission algorithms are depicted in Figure 8. In every case the frame transmission algorithm needs more energy than the burst transmission one, but, at the same time, the savings resulting from applying the water-fill procedure are also greater. In fact, there is usually very little difference in the consumption of both EEE algorithms in that case. As expected, the equitable share draws more energy than the other two shares and the water-fill share is the one that produces the best results. Finally, the dynamic water-fill algorithm improves the results, but not substantially.

We have also measured the impact of the different algorithms on queuing delay. Figure 9 shows the average

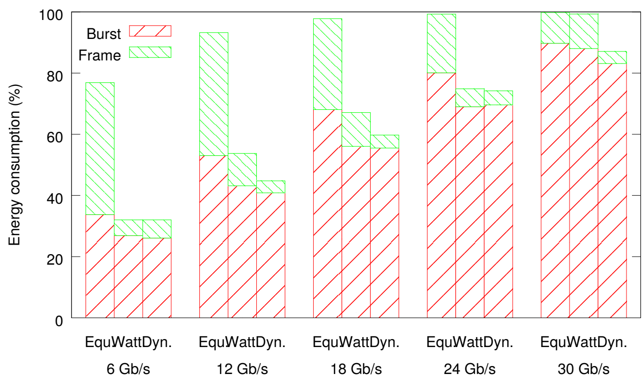


Figure 8: Energy consumption with real traffic traces when employing different strategies to share the traffic in a 4-link bundle.

queuing delay suffered by the traffic in the previous experiment. As it is the case for single EEE links [13], we observe that burst transmission always causes more delay than frame transmission. We also find that the different sharing methods impact on the queuing delay differently. In most cases, the equitable share produces the least delay while the naïve water-fill algorithm exhibits the largest one. This is unsurprising since we have driven the links near their capacity. However in all cases, the delay is still small and stays in the range of the tens of microseconds. The dynamic water-fill algorithm gets the best results, as its delay is in every case much lower than that of the naïve water-fill algorithm and never much larger than the equitable one.

6. Conclusions

This paper presents an optimum, yet simple, procedure for distributing traffic load among the links of a bundle that minimizes energy consumption when individual links

[4] In burst transmission power savings reach their maximum for a higher delay value than frame transmission. This is expected as burst transmission adds additional delay in the form of queuing before waking up a link.

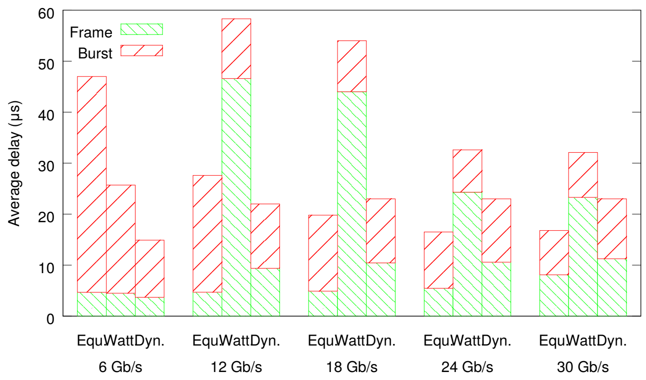


Figure 9: Average queuing delay with real traffic traces when employing different strategies to share the traffic in a 4-link bundle.

employ an EEE algorithm. As explained, the maximum energy savings are obtained when traffic is only transmitted on a link if all the previous ones in the aggregate are already being used at their maximum allowed load. The paper proves the optimality of the procedure for typical energy cost functions of individual Ethernet links.

The provided procedure is oblivious of the energy saving algorithm used in the links, whether it is the simple *frame transmission* algorithm or the more efficient *burst transmission* one. Moreover, we found that as the number of links forming the bundle increases, the difference in the total energy consumption between both algorithms vanishes when using our sharing procedure. Thus, for bundles made up of many links it is advisable to use the simpler *frame transmission* algorithm in the links, as it both reduces complexity and adds less latency to the transmitted frames.

We have also explored several alternatives to build a practical implementation of the water-filling idea to then present a simple practical implementation of the water-filling algorithm that is able to keep average delay controlled at a configurable target value while minimizing overall energy consumption.

Finally, we have tested our procedure with both synthetic and real traffic traces. In all cases, the obtained results match our expectations with the best results being obtained when the proposed sharing algorithm is employed.

Appendix A. Proof of proposition 1

In this Section, we prove that for the particular case of equal cost functions the solution to the optimum allocation is a simple sequential water-filling algorithm: each link capacity is fully used before sending traffic through a new, idle link.

We assume $X < \sum_i C_i$, otherwise the solution is trivial. It is easy to see that the constraints define a convex region $\mathcal{R}$. Since the objective function is concave, it follows that it attains its minimum at some of the extreme points of $\mathcal{R}$, namely $x_i = C_i$ for $i \in T \subseteq [1:N]$, $\quad 0 < x_j < C_j$ for one $j \in [1:N]$ and $x_k = 0$ for all $k \in T^C \setminus \{j\}$. In fact, when all the cost functions are equal, the optimal traffic allocation is to use the links in decreasing order of capacity. Assume, without loss of generality, that $C_1 > C_2 > \ldots > C_N$.[5] Fix two links i and j, $i > j$, and assume that a feasible solution is the vector $\mathbf{x} = (x_i^*, \ldots, x_N^*)$. Then, since E is a concave function it is also subadditive, and for $i > j$ and $\delta < \min\{x_i^*, C_j - x_j^*\}$ we have

$$E(x_i^*) + E(x_j^*) \geq E(x_i^* - \delta) + E(x_j^* + \delta). \tag{A.1}$$

Therefore, the vector $\tilde{\mathbf{x}} = (x_1^*, \ldots, x_j^* + \delta, \ldots, x_i^* - \delta, \ldots, x_N^*)$ is a better solution that $\mathbf{x}$. Iterating this argument as many times as necessary, it is immediate to conclude that

$$x_i^* = C_i \text{ for } i = 1, \ldots, s-1 \tag{A.2}$$

$$0 \leq x_s^* = X - \sum_{i=1}^{s-1} C_i < C_s \tag{A.3}$$

$$x_j^* = 0 \text{ for } j = s+1, \ldots, N \tag{A.4}$$

is the optimal solution, where $\sum_{i=1}^{s-1} C_i \leq X < \sum_{i=1}^{s} C_i$. □

To see (A.1), recall that for a concave function f and three ordered points $a < b < c$ it holds

$$\frac{f(b) - f(a)}{b - a} \geq \frac{f(c) - f(a)}{c - a}. \tag{A.5}$$

Just let $t = (b-a)/(c-a)$ so that $b = (1-t)a + tc$. By the definition of concavity

$$f(b) \geq (1-t)f(a) + tf(c) \tag{A.6}$$

which is (A.5). Similarly, for $a < b < c$

$$\frac{f(c) - f(a)}{c - a} \geq \frac{f(c) - f(b)}{c - b} \tag{A.7}$$

and combining (A.5) and (A.7) gives

$$\frac{f(b) - f(a)}{b - a} \geq \frac{f(c) - f(b)}{c - b}. \tag{A.8}$$

Now, use inequality (A.8) twice over the tuples $(x_1 - \delta, x_i, x_j)$ and $(x_i, x_j, x_j + \delta)$ to conclude (A.1). □

Appendix B. Proof of proposition 2

Consider the auxiliary function

$$u(x) = 1 - t(x) = a(1-x)\frac{f(x)}{f(x) + b} = g(x)h(x) \tag{B.1}$$

[5] If some links are of the same capacity, each permutation of the links lead to an equivalent solution of the problem.

where $g(x) \triangleq a(1-x)$ and $h(x) \triangleq f(x)/(f(x)+b)$. Strict concavity of $t(x)$ is equivalent to $u(x)$ being strictly convex or, alternatively, to $u''(x) > 0$. Taking the second derivative of $u(x)$ we get

$$u''(x) = g(x)h''(x) - 2ah'(x), \tag{B.2}$$

because $g'(x) = -a$. So, $u(x)$ is strictly convex if and only if $g(x)h''(x) > 2ah'(x)$. But

$$h'(x) = b\frac{f'(x)}{(f(x)+b)^2} < 0 \tag{B.3}$$

since we assumed $f(x)$ to be decreasing. With $g(x) > 0$ for $x \in [0,1]$, $a > 0$ and (B.3), (B.2) shows that $h(x)$ convex implies $u(x)$ convex. Finally,

$$h''(x) = b\frac{f''(x)(f(x)+b) - 2(f'(x))^2}{(f(x)+b)^3} \tag{B.4}$$

and $h(x)$ is a convex function —$t(x)$ is a concave function— if and only if $f''(x)(f(x)+b) > 2(f'(x))^2$, since $f(x)$ is nonnegative. □

Acknowledgments

Work supported by the European Regional Development Fund (ERDF) and the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (AtlantTIC).

References

[1] S. Lambert, W. V. Heddeghem, W. Vereecken, B. Lannoo, D. Colle, M. Pickavet, Worldwide electricity consumption of communication networks, Optics Express 20 (26) (2012) 513–524.

[2] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, N. McKeown, Elastictree: Saving energy in data center networks, in: USENIX Symposium on Networked Systems Design and Implementation, NSDI 2010, San Jose, CA, USA, 2010, pp. 249–264.

[3] IEEE 802.3az (Oct. 2010).
URL http://dx.doi.org/10.1109/IEEESTD.2010.5621025

[4] M. Gupta, S. Singh, Using low-power modes for energy conservation in Ethernet LANs, in: Proceedings of the IEEE INFOCOM, Anchorage, AK; USA, 2007, pp. 2451–2455. doi:10.1109/INFCOM.2007.299.

[5] M. Rodríguez Pérez, S. Herrería Alonso, M. Fernández Veiga, C. López García, Improved opportunistic sleeping algorithms for LAN switches, in: Proceedings of the IEEE Globecom, Honolulu, HI, USA, 2009.

[6] P. Reviriego Vasallo, J. A. Maestro, J. A. Hernández, D. Larrabeiti, Burst transmission for Energy-Efficient Ethernet, IEEE Internet Comput. 14 (4) (2010) 50–57.

[7] K. Christensen, P. B. N. Reviriego Vasallo, M. Bennett, M. Mostowfi, J. A. Maestro, IEEE 802.3az: the road to energy efficient Ethernet, IEEE Commun. Mag. 48 (11) (2010) 50–56.

[8] P. Reviriego Vasallo, J. A. Hernández, D. Larrabeiti, J. A. Maestro, Performance evaluation of Energy Efficient Ethernet, IEEE Commun. Lett. 13 (9) (2009) 697–699.

[9] P. Reviriego Vasallo, K. Christensen, J. Rabanillo, J. A. Maestro, An initial evaluation of energy efficient Ethernet, IEEE Commun. Lett. 15 (5) (2011) 578–580.

[10] S. Herrería Alonso, M. Rodríguez Pérez, M. Fernández Veiga, C. López García, Opportunistic power saving algorithms for Ethernet devices, Computer Networks 55 (9) (2011) 2051–2064. doi:10.1016/j.comnet.2011.02.012.

[11] S. Herrería Alonso, M. Rodríguez Pérez, M. Fernández Veiga, C. López García, A power saving model for burst transmission in energy-efficient Ethernet, IEEE Commun. Lett. 15 (5) (2011) 584–586. doi:10.1109/LCOMM.2011.040111.110547.

[12] M. A. Marsan, A. F. Anta, V. Mancuso, B. Rengarajan, P. Reviriego Vasallo, G. Rizzo, A simple analytical model for energy efficient Ethernet, IEEE Commun. Lett. 15 (7) (2011) 773–775.

[13] S. Herrería Alonso, M. Rodríguez Pérez, M. Fernández Veiga, C. López García, A GI/G/1 model for 10 Gb/s energy efficient Ethernet, IEEE Trans. Commun. 60 (2012) 3386–3395. doi:10.1109/TCOMM.2012.081512.120089.

[14] IEEE Std 802.1ax IEEE standard for local and metropolitan area networks — link aggregation (Nov. 2008). doi:10.1109/IEEESTD.2008.4668665.

[15] T. Karagiannis, M. Molle, M. Faloutsos, A. Broido, A nonstationary Poisson view of Internet traffic, in: Proceedings of the IEEE INFOCOM, Vol. 3, 2004, pp. 1558–1569.

[16] A. Vishwanath, V. Sivaraman, D. Ostry, How Poisson is TCP traffic at short time-scales in a small buffer core network, in: IEEE 3rd Intl. Symp on Advanced Networks and Telecommunication Systems (ANTS), 2009, 2009, pp. 1–3.

[17] M. Abramowitz, S. I.A., Handbook of mathematical functions, Dover, New York, USA, 1972.

[18] ns Network Simulator v2.35 with IEEE802.3az support, https://github.com/migrax/ns2/commits/ieee802.3az (Feb. 2014).

[19] The CAIDA UCSD anonymized 2013 Internet traces — 2013/08/15 13:14:00 UTC, https://www.caida.org/data/passive/passive_2013_dataset.xml.