UNIVERSITY CARLOS III OF MADRID

Department of Telematics Engineering

Master of Science Thesis

Using Energy Efficient Ethernet (802.3az) in Web Hosting Centers

Author: **Angelos Chatzipapas**
Dipl. Eng. in Computer Engineering and Informatics

Supervisors: **Antonio de la Oliva, Ph.D.**
Vincenzo Mancuso, Ph.D.

Leganés, September 2012

Abstract

The contribution of this master thesis is twofold. First we present an analytical model of IEEE 802.3az and second we summarize the results of our measurement campaign conducted in collaboration between Institute IMDEA Networks and InterHost. The analytical model uses simple traffic parameters to estimate the power consumption of the newly released standard for Energy Efficient Ethernet (EEE), namely IEEE 802.3az. With our measurements, we have characterized the behavior of the aggregate traffic flowing through one of the InterHost company's firewalls at one of their web hosting centers located in Madrid, Spain. We used the collected data in order to evaluate the potential for power saving that could be achieved by replacing the company's gigabit wired links with EEE connections. In the thesis, we plot the daily load and the potential power saving computed by simulating the measured traffic with an EEE link simulator as well. Using the presented model for predicting the EEE power saving from few statistical traffic parameters, we show that the EEE power saving for 1000Base-T links can be estimated with good accuracy. Finally, to show the importance of precise measurements, we first collect high precision timestamps for packet arrivals by means of specialized high resolution hardware, then we post-process the collected traces and introduce uniformly distributed random noise to the original timestamps. Results show that (i) substantial power saving could be achieved, higher that 40% in the peak hour and as high as 90% overnight, and that (ii) EEE power saving predictions can be biased by timestamp errors and so, high precision hardware is needed. Last but not least, packet coalescing for EEE is discussed to further reduce the power consumption of links under medium to high load operation.

Table of Contents

List of Figures

Chapter 1

Introduction

During the past years a lot of effort has been made to increase processing, communication, switching speed and data storage without any effort to optimize the power consumption. According to [1], about $14\ TWh$ were consumed in 2005 by the telecom core network in EU-25[1] and the consumption is expected to increase to about $30\ TWh$ by 2020. We can understand that, although this power consumption is useful for the human beings, it is also potentially harmful for our environment since it produces an augmented amount of CO_2 emissions and highly contributes to the greenhouse effect. The current threat to the environment could turn into a much more serious threat in the near future, since there is a growing demand of new generation devices that require connection to the Internet (such as televisions, white goods, etc.). In addition existing network connected devices are now increasing their bandwidth demands (e.g., Web servers, databases, etc.). In fact, the Internet traffic might grow with the number of data centers in the network and the number of users that demand higher amounts of traffic such as bigger files, videos, TV over IP etc. Hence, as the demand for much traffic rises, especially in developing countries, more and more energy consumption is expected for networking.

In order to protect the environment and obtain lower service cost, Internet Service Providers (ISPs) and network operators are currently trying to deploy new strategies to achieve lower energy bills and reduced energy consumption. There are three main roadmaps to achieve these goals:

1. The hardware optimization using more efficient electronics in order to reduce the power consumption of the components (the hardware producers are responsible for it).

2. The rising usage of renewable power sources (solar panels, wind generator etc.).

3. The use of power saving models and energy efficient algorithms (ISPs and network operators are responsible for it).

[1]The first 25 European countries that were forming European Union at that period: Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, The Netherlands, United Kingdom

This thesis investigates to the third roadmap, using the recently approved Energy Efficient Ethernet (EEE) standard for power saving issues in Local Area Networks. Indeed, according to [2], based on estimations for the year 2005, EEE can potentially obtain significant reductions of about $3\ TWh$ per year.

Legacy Ethernet is a power-unaware standard which consumes a constant amount of power independently on the actual traffic, flowing through the wires. However, low speed Ethernet cards consume about $200\ mW$ which is not significant considering the power consumption of a server or a home PC, so that Ethernet power saving strategies were not relevant for low speed connections. In contrast, new high speed Gigabit interface cards may consume up to $20\ W$ which makes reasonable the introduction of a power saving mechanism. In fact, taking into account that the amount of Web Hosting Centers and server farms has been extremely increased due to the new trends and services (YouTube, Facebook, Twitter etc.), there are now billions of running interfaces that consume a constant amount of power. In addition, Ethernet links are basically inactive most of the time. Therefore, a new power aware Ethernet standard (standardized late 2010) was introduced to minimize the power consumption of the links, namely IEEE 802.3az, or EEE (Energy Efficient Ethernet).

Later in this thesis, we present an analytical model, that uses simple statistical parameters, (such as mean interarrival time, variance etc.) and models the power consumption of an EEE link over time. We use real traces to evaluate the potential EEE power saving, and we focus exclusively on $1\ Gbps$ links, which are the most widely adopted ones, and on the impact of uplink/downlink traffic correlation on such saving. With our results we show that EEE might save more than 40% of the link power most of the time, with peaks of 90% or more during night hours. We also unveil that high precision packet timestamps are key to achieve high accuracy estimations via simulation, and to enable the use of simplified analytical computations. In particular, noisy measurements severely impact the quality of EEE power saving estimates as soon as the maximum timestamp deviation due to noise reaches a few milliseconds, which is below the typical timestamp accuracy of non-dedicated network hardware, i.e., of inexpensive but imprecise driver timestamping. This justifies using specialized high accuracy timestamping hardware.

Finally we investigate on packet coalescing for EEE. The idea is to coalesce packets in a buffer of limited size until either the buffer is full and the interface transmits the coalesced packets, or a timeout expires and the interface starts the transmission. This method allows to improve power performance when traffic is not scarce and packet arrivals have short spacing.

The rest of the thesis is organized as follows. Chapter 2 presents EEE standard and how it may lead to potential energy savings. Chapter 3 describes an analytical model about the power consumption of the links. Chapter 4 shows evaluation results about the potential power savings that can achieved in web hosting centers and the importance of precise packet timestamping. Chapter 5 describes the related work and possible enhancements, i.e. coalescing, that improve EEE performance and power save. Chapter 6 summarizes and concludes the thesis.

Chapter 2

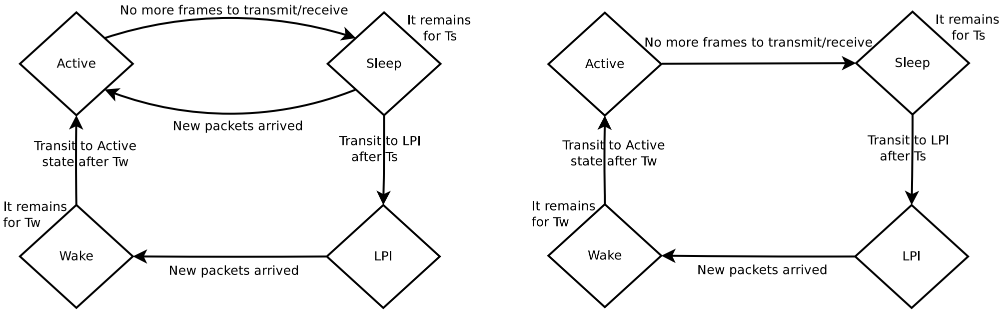
Energy Efficient Ethernet - (EEE)

Energy Efficient Ethernet 802.03az [3] was standardized in September 2010 and aims to provide significant power saving in LANs. Formerly, the evolution of LANs led towards higher link speeds for faster communication and higher bandwidth, in order to satisfy the increased demand for data (link speeds from $10\ Mbps$ to $10\ Gbps$). The electricity consumption of relatively "old" network interfaces remained in very low levels so the main concern of Ethernet component producers was not to save power. For example, in $100\ Mbps$ Ethernet links, the Ethernet devices consume about $200\ mW$ of power [4]. However, the new high speed Ethernet ($1\ Gbps$ or faster) requires several Watts of power consumption [5] for each interface. Considering a usual server that consumes around $200\ W$, a simple Ethernet device contributes to 10% of this amount. Indeed, data and Web Hosting Centers have a huge number of interface cards which finally generate a high cost for the center (electricity bills). Thus, the idea of reducing the power consumption of Ethernet links appears in the foreground.

Legacy Ethernet consumes a constant amount of power with or without traffic which makes it totally inefficient with typical Ethernet traffic profiles. This behavior results in a huge waste of power since it is well known that Ethernet links are inactive most of the time with utilization factors from 5% for a home PC to 30% for heavy loaded data servers [6–8]. EEE aims to reduce this waste of power and approach power proportionality, i.e., a power consumption proportional to the served traffic. The EEE standard introduces four new states for the links, state "Active" (A) which represents the busy period, state "Low Power Idle" (LPI) in which there is no traffic and the link consumes 90% less power that in state A according to measurements [9], and states "Sleep" (S) and "Wake Up" (W) which correspond to the time spent during switching from state A to LPI and vice versa, respectively [10]. In the following subsection, we describe how EEE links behave.

2.1 Behavior of EEE Links

The behavior of EEE links is illustrated in Fig. 2.1. Packet transmissions may occur only in state A. When there is no traffic to serve, the link switches to LPI state, where it consumes about 10% of the power consumption in state A [9]. The transition interval T_S between states A and LPI is denoted as the state S of the link. When a new frame arrives in either side of the link for transmission, the link switches back to state A after a transition



(a) State transition diagram for 100Base-T and 1000Base-T EEE links.

(b) State transition diagram for 10GBase-T EEE links.

Figure 2.1: State transition diagrams for various link speeds.

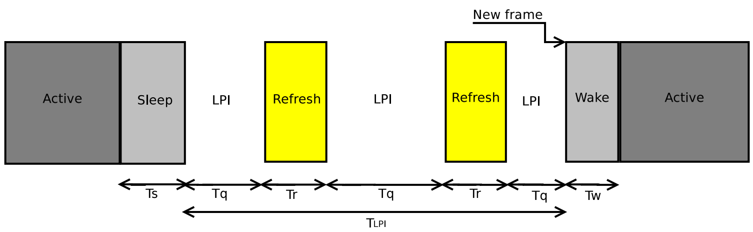


Figure 2.2: Channel transitions during a whole cycle.

time T_W which defines the state W of the link. According to EEE, the duration of T_S and T_W varies among the different link speeds. Table 2.1, indicates the duration of the two states for 100Base-Tx, 1000Base-T and 10GBase-T links. The table also shows the transmissions time and the efficiency (the time to transmit a frame over the time to transmit a frame plus $T_W + T_S$) for a whole cycle of a big and a small frame. Furthermore, the standard defines a “Refresh” message, that checks the channel state in LPI, ensures that the receiver elements still follow the channel conditions and maintains the duration of T_W in low levels. Fig. 2.2 shows the channel state transitions over time for a whole cycle (note that an LPI interval is composed by several idle intervals T_q and refresh intervals T_r).

Table 2.1: Time required for W, S and Frame Transmission [μs]

Speed	Minimum T_S	Minimum T_W	Transm. time for 1500 $bytes$ frame (efficiency)	Transm. time for 150 $bytes$ frame (efficiency)
100Base-T	200	30	120 (48%)	12 (4.8%)
1000Base-T	182	16	12 (5.7%)	1.2 (0.57%)
10GBase-T	2.88	4.48	1.2 (14.6%)	0.12 (1.46%)

EEE defines slightly different behaviors for different link speeds. First, in 100Base-Tx and 10GBase-T links, EEE works independently in the two traffic directions, i.e., each link direction can go to LPI independently of the traffic activity in the other direction. In contrast, in 1000Base-T links, EEE can enter in state LPI only when both link directions have no data to send. Second, 100Base-Tx and 1000Base-T links treat differently a new frame that arrives during state S than 10GBase-T links. As illustrated in the transition diagrams in Fig. 2.1b, $10\ Gbps$ EEE links have to execute the whole sequence of states S-LPI-W before switching back to the state A, while in Fig. 2.1a, $100\ Mbps$ and $1000\ Mbps$ EEE links in state S can switch directly to state A if there is a new packet arrival.

At this point, it is important to mention that, in states S and W, EEE links practically consume as much power as in state A (see to Reviriego et al. [9]). So, the actual EEE saving is obtained when the link remains in state LPI. As a consequence, the goal of most research studies is to find how the links can remain in state LPI for the longest possible duration.

Since the scope of using EEE is to reduce the power consumption of the Ethernet links, it is of great interest to evaluate the performance of EEE under different working conditions. Therefore, models and simulators are needed to produce such a performance evaluation. In particular, we have developed a simple analytical model for 1000Base-T EEE links and a C++ simulator. In the next Chapter we show the analytical model that estimates the power consumption of 1000Base-T EEE links. In Chapter 4 we will introduce our simulations to evaluate the performance of EEE over real traffic traces.

Chapter 3

Analytical Model

We model the behavior of the EEE link as an $M/G/1$ queue in which the packet service rate is non-zero only in state A, where it equals a constant R, corresponding to the link speed. We denote by S_p the size of a single packet and by $E[S_p]$ the average packet size. Frames arrive in batches of random size $N_b \geq 1$ packets, according to a Poisson process with rate λ [11]. Arrivals in batches simulate as much as possible the behavior of real networks, since usually packets arrive at the network interfaces back-to-back. In addition, $N_b = 1$ denotes the Poisson arrival of single packets. The following model is based on the one described in [10].

Cycle analysis. The behavior of 1000Base-T EEE links over time can be seen in Fig. 3.1a. The figure shows a whole cycle starting with the arrival of a batch that causes the transition from state LPI to state A, continuing with an exchange between busy B_i and incomplete periods S and ending after a complete period S of duration T_S followed by an period LPI.

For 100Base-Tx and 10GBase-T EEE links the behavior is different since after the link enters in state S, it cannot immediately exit but after completing both states S and W, as can be seen in 3.1b. The total length of the cycle is defined by T_C and its average value is given by $E[T_C]$.

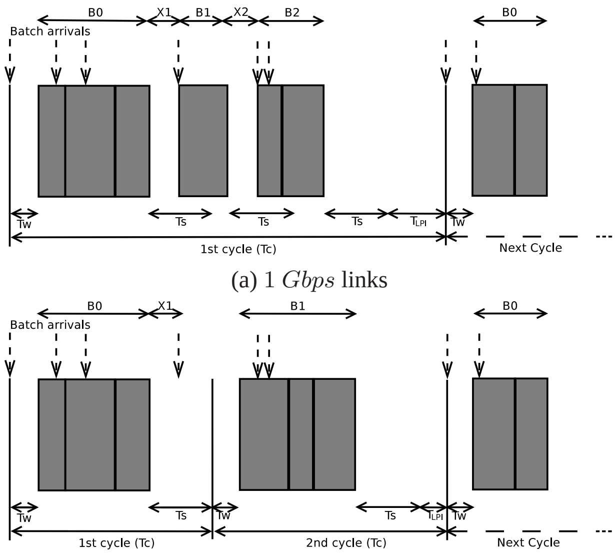
To compute the total power consumption of the system model all we need to do is to find the utilization factor of each state, and multiply it by the power consumption of the corresponding state. To do that we will use results from the renewal theory and we will estimate the average time spent in each state during a cycle over the average cycle duration. The utilization factor is denoted as ρ_α where α defines the four possible states of the link (A, S , W and LPI). Thus the total power consumption of the proposed model is given by:

$$P_{total} = \rho_A P_A + \rho_S P_S + \rho_W P_W + \rho_{LPI} P_{LPI}, \text{with} \sum_\alpha \rho_\alpha = 1 \tag{3.1}$$

Reviriego et al [9] prove that $P_A \simeq P_S \simeq P_W$ and therefore Eq. 3.1 is transformed to:

$$P_{total} = (1 - \rho_{LPI}) P_A + \rho_{LPI} P_{LPI} \tag{3.2}$$

Next, we want to derive the cycle parameters for the two cases of (1) 100 $Mbps$ and 10 $Gbps$ EEE link and (2) 1 $Gbps$ EEE links. Exceptionally, even though the standard specifies that 1 $Gbps$ EEE links can go to state LPI only when there is no traffic in both link



(a) 1 $Gbps$ links

(b) 100 $Mbps$ and 10 $Gbps$ links

Figure 3.1: System cycle for EEE links.

directions, we assume unidirectional link behavior for the following reasons. First, earlier measurements [10] have shown that it is a good approximation to use unidirectional traffic and that the superposition of the traffic in the two link directions can model with acceptable deviation bidirectional links when the load in one direction is very low or the loads in the two directions are strongly asymmetric. Second, the correlation of traffic in the two link directions is difficult to model. Indeed, there are no models available in the literature that consider bidirectional traffic on 1 $Gbps$ EEE links.

3.1 1 $Gbps$ Unidirectional EEE Links

To find the utilization factors we first need to compute the time that the link stays in each state. Fig. 3.1a shows the different state transitions of the link during a whole cycle over time. The analysis in based on [10].

Interval T_W. The wake-up interval has a fixed duration of T_W.

Interval T_S. The sleep interval has a fixed duration of T_S.

Interval T_{LPI}. State LPI state lasts until the first batch arrival in the interface. Since arrival processes are Poisson with rate λ its average is:

$$E[T_{LPI}] = \frac{1}{\lambda}. \tag{3.3}$$

Interval B_0. B_0 denotes the state A after the link has exited from the state LPI. In an M/G/1 queue with batch arrivals and arrival rate λ the average duration of a busy period is given by the number of batches received (one batch that caused the transition to state A and the batches received during T_W), the mean service rate of a batch $E[S_p]E[N_b]/R$ (where R corresponds to the service rate of the channel) and the overall utilization factor

$\rho = \frac{\lambda}{R}E[S_p]E[N_b]$. Thus [12],

$$E[B_0] = \frac{(1+\lambda T_W)E[S_p]E[N_b]}{R(1-\rho)} \tag{3.4}$$

Interval B_i, for $i > 0$. B_i denotes state A after the link has exited without completing state S (intervals B_1, B_2 etc. in Figs. 3.1a, 3.1b). This expression is similar to Eq. 3.4 with the difference that there is not the interval T_S.

$$E[B_i] = \frac{E[S_p]E[N_b]}{R(1-\rho)}, \forall i > 0. \tag{3.5}$$

Interval X. When the sleeping time T_S is not completed due to an arrival, there is an interval X between the end of a busy period and the beginning of the next busy period. Otherwise, there is no X interval at all. X is then a random variable with truncated exponential distribution and rate λ. Therefore, its distribution is as follows:

$$f_X(t) = \frac{\lambda_1 e^{-\lambda t}}{1-e^{-\lambda T_S}}, \quad t \in [0, T_S]. \tag{3.6}$$

Accordingly, the average of X is as follows:

$$E[X] = \frac{1}{\lambda} - \frac{T_s}{e^{\lambda T_S} - 1}. \tag{3.7}$$

Number of repetitions N. Busy intervals $B_i, i \geq 1$, occur if the residual interarrival time at the end of the previous busy interval is shorter than T_S. We call this residual interarrival interval X. Since arrivals are Poisson, the probability of having no arrivals in T_S is $P_0 = e^{-\lambda T_S}$. Thereby, the number $N \geq 0$ of busy periods of type B_i in a cycle, i.e., not counting B_0, can be seen as the number of consecutive successes of a geometric random variable N with success probability $1 - P_0$. Hence, its average value is:

$$E[N] = \frac{1-P_0}{P_0} = e^{\lambda T_S} - 1. \tag{3.8}$$

Theorem 1. *In unidirectional 1000Base-T EEE links the average cycle duration is given by:*

$$E[T_C^{'}] = T_W + T_S + E[N] \cdot E[X] + E[B_0] + E[B_1] + \frac{1}{\lambda}; \tag{3.9}$$

$$\textbf{\textit{or}}\ E[T_C^{'}] = \frac{\lambda T_W + e^{\lambda T_S}}{\lambda(1-\rho)}. \tag{3.10}$$

Proof. By summing up all the previous intervals the proof follows. □

Therefore,

Corollary 1. *For* 1 *Gbps* *unidirectional EEE links, the utilization factors* $\rho_A^{'}, \rho_L^{'}, \rho_S^{'}, \rho_W^{'}$ *are given by:*

$$\rho_A^{'} = \frac{E[B_0] + E[N]E[B_i]}{E[T_C^{'}]}; \tag{3.11}$$

$$\rho_L^{'} = \frac{1}{\lambda E[T_C^{'}]}; \tag{3.12}$$

$$\rho_S^{'} = \frac{e^{\lambda T_S} - 1}{E[T_C^{'}]}; \tag{3.13}$$

$$\rho_W^{'} = \frac{T_W}{E[T_C^{'}]}. \tag{3.14}$$

3.2 $100\ Mbps$ and $10\ Gbps$ Unidirectional EEE Links

The utilization factors for these two cases can be found by following the same procedure as previously. The difference is that in case that a packet arrives in state S, it has to wait for state S to complete and for the link to wake up (T_W). In Fig. 3.1b we can see the state transitions of $100\ Mbps$ and $10\ Gbps$ links during a whole cycle over time. We observe that for similar traffic the behavior of the links is different compared with $1\ Gbps$ links. Thus, the interval B_i (for $i > 0$) lasts for:

$$E[B_i] = \frac{(1 + \lambda(T_S + T_W - E[X]))E[S_p]E[N_b]}{R(1-\rho)}, \forall i > 0; \tag{3.15}$$

where interval X denotes the elapsed time before the arrival of the first packet in state S.

Some analytical results from Sec. 3.1 can be reused here. I.e., $E[B_0]$ is the same as in Eq. 3.4, $E[N]$ is like in Eq. 3.8 and $E[X]$ is the same as in Eq. 3.7, the result for the average cycle time is given by the next Theorem:

Theorem 2. *In unidirectional 100Base-T and 10GBase-T EEE links the average cycle duration is given by:*

$$E[T_C^{''}] = T_W + T_S + E[N] \cdot (T_W + T_S + E[B_i]) + E[B_0] + \frac{1}{\lambda}; \tag{3.16}$$

$$\textbf{\textit{or}}\ E[T_C^{''}] = \frac{1 + \lambda(T_S + T_W)e^{\lambda T_S}}{\lambda(1-\rho)}. \tag{3.17}$$

Therefore,

Corollary 2. *For $100\ Mbps$ and $10\ Gbps$ unidirectional EEE links, the utilization factors $\rho_A^{''}, \rho_L^{''}, \rho_S^{''}, \rho_W^{''}$ are given by:*

$$\rho''_A = \frac{E[B_0] + E[N]E[B_i]}{E[T''_C]}; \tag{3.18}$$

$$\rho''_L = \frac{1}{\lambda E[T''_C]}; \tag{3.19}$$

$$\rho''_S = \frac{T_S + E[N]T_S}{E[T''_C]} = \frac{e^{\lambda T_S} T_S}{E[T''_C]}; \tag{3.20}$$

$$\rho''_W = \frac{T_W + E[N]T_W}{E[T''_C]} = \frac{e^{\lambda T_S} T_W}{E[T''_C]}. \tag{3.21}$$

3.3 Deriving Model Parameters from Measurements

Results reported in Sections 3.1 and 3.2 show that the model requires the knowledge of some parameters. However, parameters T_S, T_W and R are determined by the Ethernet link speed, while ρ, λ, N_b and S_p depend on the traffic. These parameters can be given as arbitrary input to the model, as well as estimated from real traces.

$E[S_p]$ and ρ can be directly estimated from a trace. For the rest, all we need to do is to compute the first two moments of the packet interarrival time, namely the average $E[Y]$ and the variance $V[Y]$. In fact, assuming that batches are exponentially distributed with success parameter p_b we have that:

$$E[N_b] = \frac{1}{1 - p_b}. \tag{3.22}$$

Then let Y be the interarrival time between packets. Since we consider arrival in batches, the average interarrival time between packets $E[Y]$ is given by:

$$E[Y] = 0 \cdot p_b + \frac{1}{\lambda} \cdot (1 - p_b) = \frac{1 - p_b}{\lambda}; \tag{3.23}$$

where in Eq. 3.23 the first term is the delay that packets suffer within the same batch (virtually zero delay) times the success probability p_b and the second term is the delay that packets suffer due to the next batch, times the probability of success in this case (1-p_b).

Similarly, the variance is given by the next formula:

$$V[Y] = \frac{\sqrt{1 - p_b^2}}{\lambda}. \tag{3.24}$$

Solving Eq. 3.23 and Eq. 3.24 we can find λ and p_b and then we can also compute $E[N_b]$. With the above, all parameters are computed. Therefore, we remark that using the model for real traffic traces only requires to estimate from the trace the following basic parameters: $\rho, E[S_p], E[Y], and V[Y]$.

Chapter 4

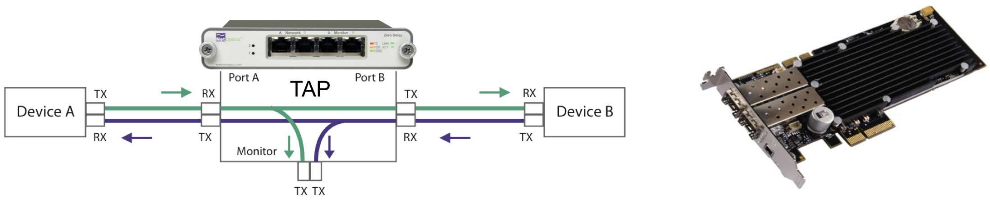
Energy Efficiency in Web Hosting Centers

In this section, we summarize the results of our measurement campaign conducted in collaboration between Institute IMDEA Networks and InterHost. With our measurements, we have characterized the behavior of the aggregate traffic flowing through one of the InterHost's firewalls at one of their web hosting centers located in Madrid, Spain. We have used the collected data in order to evaluate the potential for power saving that could be achieved by replacing the company's gigabit wired links with EEE connections. Furthermore, we plot the daily load and the potential power savings computed by simulating the measured traffic with a C++ EEE link simulator, and using the model that we previously explained in Chapter 3 for predicting the EEE and it is further analyzed in [10].

4.1 Motivation

The goal of our measurement is to collect enough data to characterize the traffic behavior of a real commercial installation, and to provide an estimate of the power saving that might be achieved at the data center by replacing existing Ethernet links with IEEE 802.3az energy saving Ethernet links. We need these traffic measurements to compute the first two moments of the packet interarrival rate, the average packet size and the offered link load, and use these parameters to feed the EEE model that we previously analyzed in Chapter 3. As explained in Chapter 3, the model in Chapter 3 is able to predict the average time spent by an EEE link in state LPI, thus allowing to estimate the amount of power saving that can be achieved with a given input traffic pattern, when EEE links are adopted.

The model does not consider that, for 1000Base-Tx links, the IEEE802.3az standard specifies that the link can go to low power state only when no traffic is present in both link directions. So the model ignores the correlation of traffic flows in the two link transmission directions. However, results shown in [10] demonstrate that that model can be suitably used to predict the power saving of EEE 1000Base-Tx links when the link is strongly asymmetrically loaded in the two transmission directions, and when the less loaded direction works at few percents of its capacity. To achieve such prediction, it is enough to consider the traffic in the most loaded link direction only (cf. Table II in [10]).



(a) Measurement architecture and TAP device (image from NetOptics website).

(b) The Endace DAG card (image from Endace website).

Figure 4.1: Measurement tools.

Here we want to test our model's performances under a broader range of realistic traffic matrices, and we want to take into account the correlation of traffic flows in the two link directions. Obtaining real data is usually very difficult, so that synthetic traffic generator is the obvious, though less accurate, alternative. In our case, InterHost allowed us to monitor the traffic at the interface between Internet and one of InterHost's firewalls. This gives us the advantage of using real data and real traffic that yields more realistic results. However, in respect of user's and company's privacy and security, we only capture few bytes of each packet, and we do not inspect the payload. Actually, our goal is to capture the arrival time of the packet and its size, which is enough for our purposes.

To achieve our goal, we need to collect precise and clock-synchronized timestamps for packet arrivals over the two link directions. Therefore we need precise measurement tools and a measurable source of real traffic.

In this rest of this Section we show that a huge amount of power might be saved at InterHost by replacing the legacy Ethernet links with EEE links, the power saving being a periodic function of the time of the day and the day of the week. More than 40% of the power might be saved most of the time, with peaks of 90% or more during the night hours. We also show that noisy measurements might severely impact the quality of our EEE power saving estimate as soon as the maximum timestamp deviation due to noise reaches a few milliseconds, which is below the typical timestamp accuracy of non-dedicated network hardware, i.e., of inexpensive but imprecise driver time-stamping. This justifies using specialized high accuracy time-stamping hardware.

4.2 Unintrusive Measurements at InterHost

All we need for our EEE power saving estimate is to compute the first two moments of the packet interarrival time, the average packet size, and the offered load. However, we can also simulate the EEE operation by using real traffic traces as an input arrival process [10]. With the simulator we can compute the exact EEE power saving, and compare this value with the estimate yielded by the model. The simulator needs an input trace file containing, for each packet, the timestamp of the packet's arrival time, and the size of the packet. Therefore, we do not need to parse the content of any packet, and there is no need to save the content of each packet for post-processing. All we need is an accurate sniffer, whose architecture is described in what follows.

To take the measurements we need a server to store the data and a *tap* to sniff and duplicate real packets without affecting the traffic. The tap is shown in Figure 4.1a. We use a NetOptics passive device which is inserted in a 1000Base-TX Ethernet link and duplicates each and every signal over the link.[1] The NetOptics device, as shown in the figure, is also able to separate the traffic in the two directions, i.e., it provides the traffic in the two directions over two separate cables to be connected to a monitor device.

We use a Linux Dell Xeon server with $1\ Tbyte$ of memory to store the monitored data. The two monitoring ports of the tap are connected to a digital capture card, specifically a high accuracy two-port Endace DAG card as the one shown in Fig. 4.1b.[2] The DAG card is a dedicated capture device with dedicated CPU and memory, able to capture 100% of the traffic at up to $10\ Gbps$ over each direction. Furthermore, the DAG card has a unique time-stamping engine that guarantees clock synchronization to the nanosecond over the two monitoring ports.

The tap is placed within the link that we want to measure, i.e., in front of one of InterHost's firewalls, and the DAG is activated once per hour to collect at most 100 bytes per packet for 200 seconds. The link speed is $1\ Gbps$.

We keep as well a remote *ssh* connection with the server, so that we can periodically transfer the collected traces to a Linux server located in our lab at Institute IMDEA Networks. Once the traces are in our lab, we post-process them with the *tshark* packet analyzer and create simplified and anonymous trace files containing only the timestamp of each packet arrival and its size in bytes.

4.3 Using the Collected Traces (EEE Simulation and Model)

We have been capturing traffic traces from November 26, 2011 to the present days (end of August 2012). Therefore, we now have enough data to characterize the traffic behavior over the measured InterHost's link. In particular, we observe a daily typical maximum traffic of about 4-5% in the most loaded link direction, with the exceptional value of 11% on one traffic direction on February 8, 2012.

After collecting a trace, we post-process the captured packet fragments to generate a simple two-column file containing a timestamp and a packet size for each link transmission direction. Once the simplified traces are ready, we first simulate the EEE operation with an ad-hoc developed C++ simulator [10], then we extract the statistical parameters needed to run the EEE model and estimate the EEE power saving though the model discussed in Chapter 3 as well.

In the following subsections we show our measurements together with the EEE power saving as computed via the simulator in each of the sampled traffic intervals. Model results will be compared to the simulation in Section 4.6.

4.4 Traffic Behavior and EEE Power Saving

In Fig. 4.2 we plot the monthly load in each direction for February 2012, which is representative of our measurements and the power saving that might be achieved by means of

[1] http://www.netoptics.com/products/network-taps
[2] http://www.endace.com/endace-dag-high-speed-packet-capture-cards.html

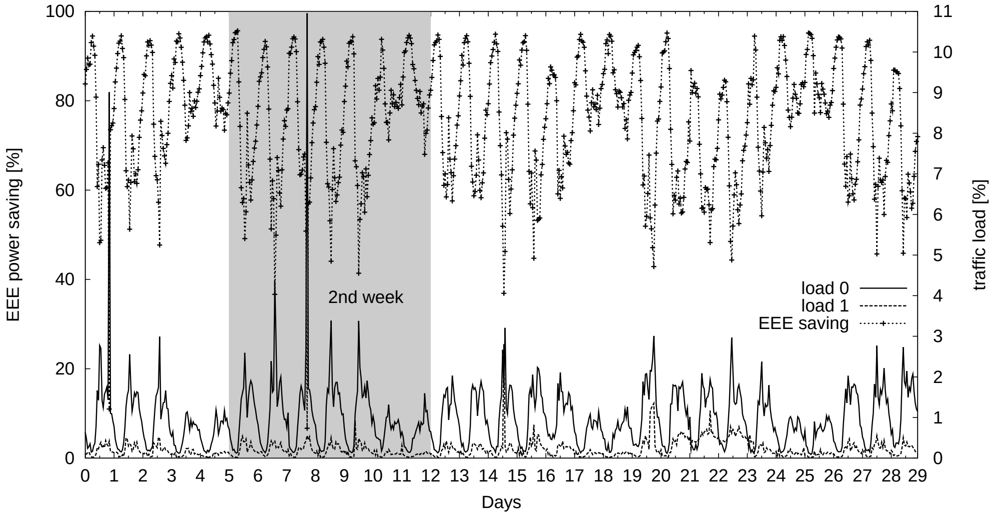


Figure 4.2: The power savings during a whole month.

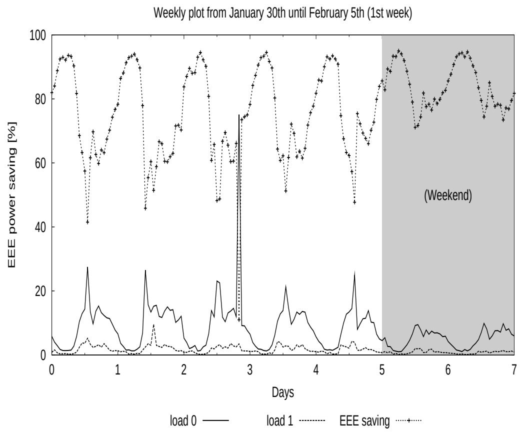
EEE links. Our observations are that we have a maximum traffic load of about 5% in the most loaded direction, but for holidays and weekends, when the peak load is below 2%. In the figure, traffic patterns are quite regular, showing higher traffic activity over 5 consecutive days (corresponding to the Monday-Friday period), followed by low traffic intensity over the weekend It is also evident that overnight the traffic is very low.

Processing the collected data with the EEE simulator reveals that overnight and during the weekend, EEE might save 70-90% of the power with respect to legacy Ethernet. Even during weekdays the power saving might be at least about 40% during the peak hours.

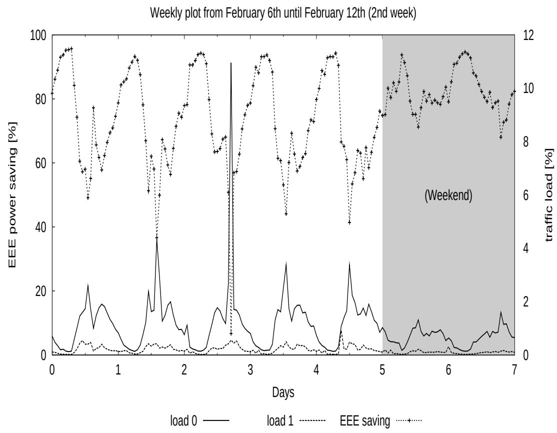
Considering that a gigabit Network Interface Card (NIC) consumes about 2 W (e.g., see Intel 1G datasheets [13]), the estimated power saving for a single link with two NICs connected sums up to 1.1 to 2.6 $KWh/month$, these numbers being relevant for large data centers or web hosting companies running tens of thousands of links. For example, consider a typical large data center (more than 4650 m^2 nowadays [14]) with $\sim 40,000$ servers (like Google's data centers [15]). According to [16] each server has three network ports on average. Therefore, in total, the data center runs about 120,000 network ports, each of which will typically be very low loaded. Thereby, the power saving could be close to the peak most of the time. Considering the average price of electricity in Europe, i.e., ~ 0.15 $Euros/KWh$, using EEE links might generate a potential economy of more than $280,000$ $Euros/year$. The case of 10 $Gbps$ (and 100 $Gbps$) links is even more promising, since they consume even more power, i.e., $4.5 - 20$ W according to [13].

4.5 Zooming over the Weekly Saving

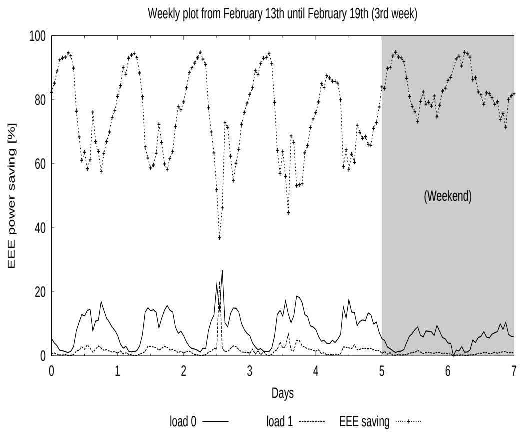
In Fig. 4.3 we plot the weekly results of February divided into four weeks. We can better observe the slightly increased traffic demands during Mondays and the low traffic demands during weekends and overnight. The daily spikes occur at about 1 pm and 6 pm, e.g., just before leaving the office (either before lunch time or before going home). This



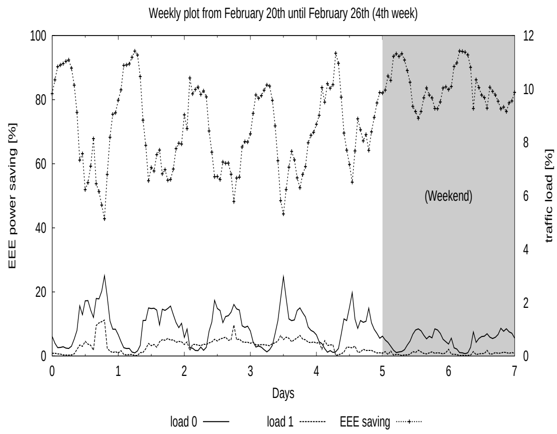
(a) Power saving during the first week of February.



(b) Power saving during the second week of February.



(c) Power saving during the third week of February.



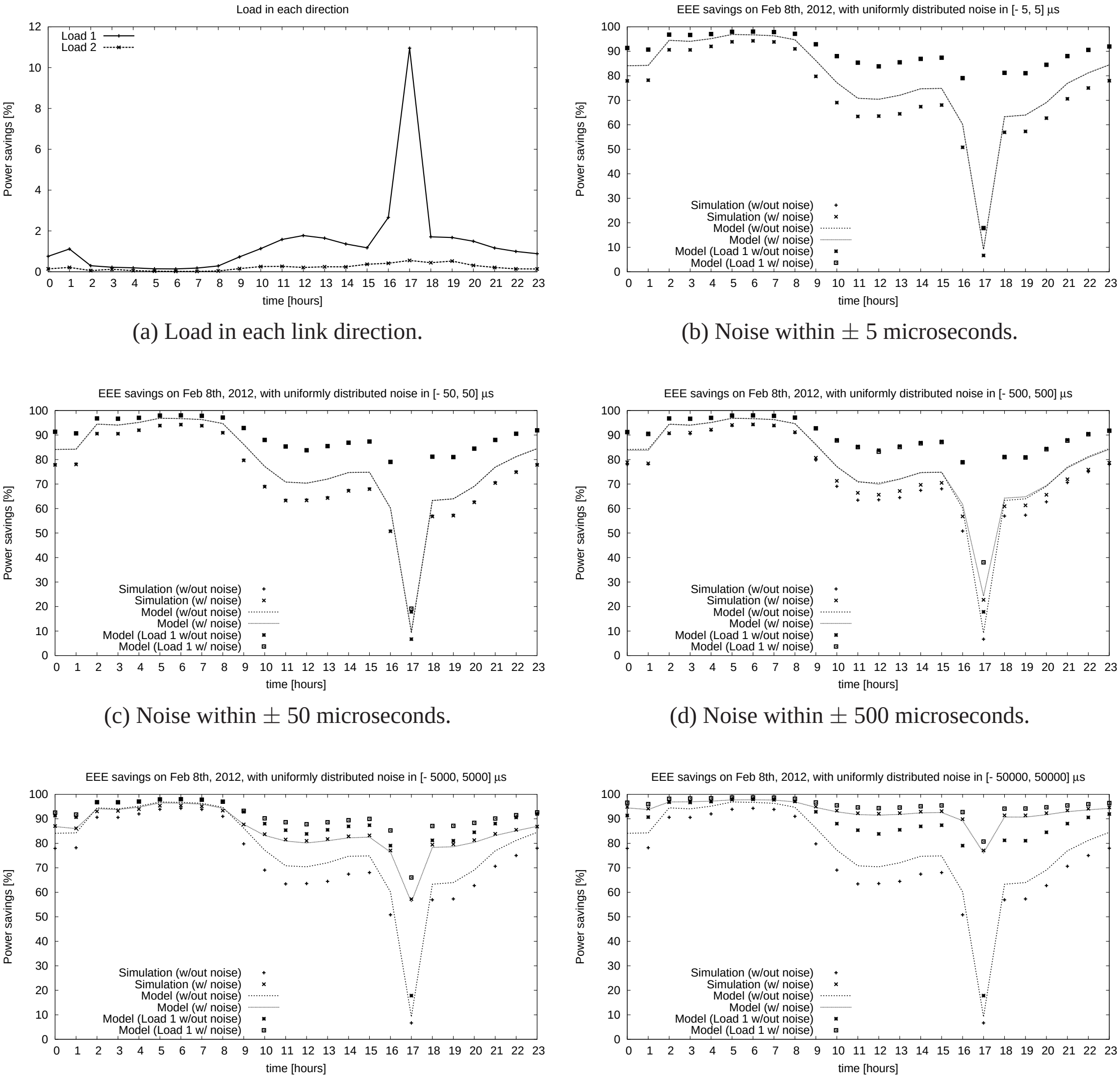
(d) Power saving during the forth week of February.

Figure 4.3: Weekly plots.

traffic distribution over time clearly depends also on the nature of the websites hosted at InterHost premises, about which we have no information. However, the measured traffic patterns are qualitatively in line with other traffic patterns reported in literature.

4.6 Impact of Noisy Measurements

In the previous subsection, we have plotted the power saving that might be achieved with EEE, based on the load and the traffic characteristics. We have used a simulator to compute such a power saving gain. In contrast, here we plot the EEE power saving gain based on simulation and model (of Chapter 3), and show the impact of noisy measurements on the quality of EEE power saving estimates. Since the model runs for unidirectional traffic only, we feed it with either the traffic of each link direction separately, or with a single trace representing both traffic directions. Specifically, when the model is computed



(a) Load in each link direction.

(b) Noise within $\pm$ 5 microseconds.

(c) Noise within $\pm$ 50 microseconds.

(d) Noise within $\pm$ 500 microseconds.

(e) Noise within $\pm$ 5 milliseconds.

(f) Noise within $\pm$ 50 milliseconds.

Figure 4.4: Noise introduction.

over the aggregate traffic measured over the two link directions, we merge the traffic traces obtained for the two directions. As a consequence, the model cannot completely capture the correlation of traffic in the two directions. In fact, the IEEE 802.3az standard says that link interfaces can enter the low power state only when *both link directions* are idle. Instead, we are simplifying the model to cope with a unique link direction, possibly resulting from the merging of the two real link directions.

In particular, we picked the day with the maximum traffic recorded over the entire measurement campaign, which corresponds to February 8th, 2012, and we perturbed the originally collected timestamps by adding zero-mean uniformly distributed noise. The load registered over the daily samples is depicted in Fig. 4.4a, and it achieves a maximum value of 11% over the most loaded link direction. That load is not affected by our timestamp alterations.

The extremes of the added noise span from $\pm$ 5 microseconds in Fig. 4.4b and go up to $\pm$ 50 milliseconds in Fig. 4.4f. Each plot contains two lines that indicate the power saving estimates obtained with the model computed on the aggregate link traffic (merging of the two traffic directions), with and without the artificially added noise. The figures also report the EEE power saving as estimated by the model when considering only the traffic in the most loaded link direction (which is Load1 in the figures), with and without timestamping noise. Finally, the figures include the EEE computed through the simulations, with and without noise. Therefore, in each plot, the values obtained without noise are always the same, thus representing the benchmark for these experiments.

From the figures, we note that model and simulator report similar trends, with close but distinct power saving estimates. Differences are well explained considering that the model does not capture the bidirectional nature of the EEE power saving mechanism. Limited differences are due to the low load measured on both link directions. The model which only consider the most loaded link is obviously the one reporting the highest power saving. Similarly, the model considering the aggregate traffic yields the lower power saving, since it does not consider that packets belonging to opposite traffic directions might be served in parallel.

Let's now consider the impact of noise. As we can see in Figs. 4.4b to 4.4f, with the introduction of noise in the order of up to one millisecond, the curves obtained with and without noise almost overlap. In contrast, with the introduction of more noise, we can see that the graphs obtained with noisy measurements tend to achieve higher power saving. In fact, adding noise to timestamps contributes to break the traffic correlation between the two directions, so that (i) simulator and model yield very similar results, and (ii) power saving occurs with roughly the product of probabilities of having each link direction idle. As a result, about 80% power saving can be erroneously estimated even for the peak hour.

We remark that noisy measurements brings to erroneous power saving estimates, and conclude that timestamp errors larger that a millisecond are not tolerable to achieve accurate estimates. Considering that millisecond accuracy in time-stamping is barely achievable with ordinary operating systems and driver-operated time-stamping, which depends on system interrupts, we also conclude that dedicated traffic measurement tools are needed, as the ones that we have used for our measurement campaign.

Chapter 5

Related Proposals

A few number of similar works on EEE appeared recently in the literature. In the following subsections, we first present the related analytical model in [10], second we show the results of existing evaluations on EEE links based on [4, 9] and last we introduce enhancements for future EEE systems that might boost the performance of EEE in terms of power saving [18, 19].

5.1 Evaluation of EEE

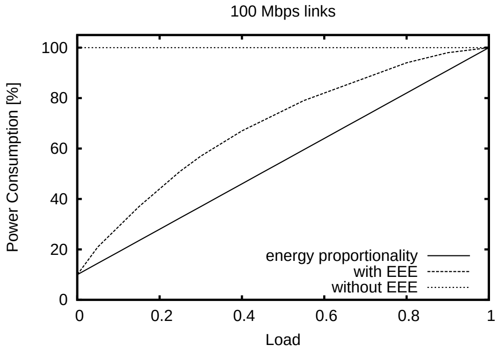
5.1.1 Model

In [10], Ajmone Marsan et al. model the behavior of EEE links with independent traffic in the two link directions, and validate their model by means of real traces for $1\ Gbps$ and $10\ Gbps$ links. The authors describe the behavior of EEE links in a similar manner as we also did in this thesis. Additionally, using a simulator that was developed by the authors, they predict the power saving of real traces with EEE links instead of the legacy Ethernet. They make use of traces obtained by Google's data centers and CAIDA [20, 21]. The final simulation results are very close to the ones from the model. However, they evaluate $10\ Gbps$ EEE links using CAIDA traces for backbone links, so that their results cannot be used for data centers using $1\ Gbps$ links. Similarly, the number of $1\ Gbps$ traces considered in that paper is limited to a few units so that their results cannot be easily generalized to broader scenarios. Moreover bidirectional Google datacenter traces are not synchronized with high precision since the authors could not access the data center and use high precision tools.

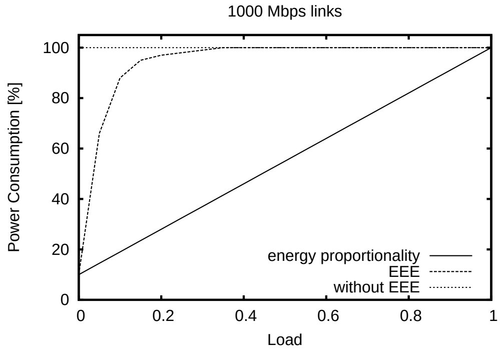
5.1.2 Simulation Results

In [4] the authors first simulate the behavior of EEE links and then estimate the power consumption of different scenarios by capturing real traces.

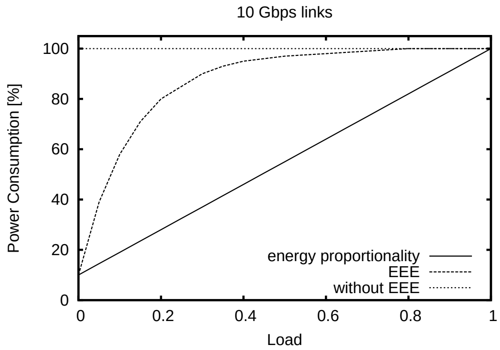
The simulations were realized for three different link speeds: $(1)\ 100\ Base-Tx$, $(2)\ 1000\ Base-T$ and $(3)\ 10\ GBase-T$. For every example case it is assumed that each link direction is independent of the other and so the link is allowed to be, e.g., in state LPI in one direction and in state A in the other direction. Like we previously said this is not the case for 1000Base-T links, but for simplicity and similarly to what we did in Chapter 3,



(a) 100Base-T



(b) 1000Base-T



(c) 10GBase-T

Figure 5.1: Power consumption vs. Load (data from [4]).

the authors of [4] use the unidirectional behavior assumption also for $1\ Gbps$ EEE links. In their paper is further assumed that large frames of $10000\ bits$ are exchanged and they arrive according to Poisson distribution. This identifies the best case for EEE links since, comparing the transmission time of such highly aggregated traffic with the interval $T_S + T_W$, we observe that the overhead produced is less significant. In fast networks, if we only send small packets (e.g., acknowledgments) the power overhead could be huge since T_{frame} (the time needed to transmit a frame) is relatively small compared with $T_S + T_W$. Last assumption in [4] is that state LPI consumes about 10% the power of state A. Simulations proposed in [4] are compared on one side with the ideal case of energy proportionality [17], where power consumption is given by the formula:

$$P_{ideal} = P_{Active} \cdot Load + P_{LPI} \cdot (1 - Load); \tag{5.1}$$

and on the other side with the case of legacy power-unaware Ethernet links. The results of the simulations are illustrated in Fig. 5.1. We can observe that, for $100\ Mbps$ links, energy consumption is more energy proportional than it is for $1000\ Mbps$ and $10\ Gbps$ links. This is because the overhead in time due to state S and W is very large compared with the data transmission in $Gbps$ links.

Next, the authors of [4] measure the traffic and the energy consumption in real traffic patterns. They study four different scenarios in order to have a more wide view of EEE behavior. These are:

- A user who is connected to its ADSL router using 100 $Mbps$ connection and downloads some video files.
- Two users are connected between each other using 100 $Mbps$ LAN connection to exchange a file.
- A university access link of 1000 $Mbps$ with highly multiplexed Internet traffic.
- Some traces from Google's data centers.

The first scenario is typical for residential users with ADSL connection from 1 $Mbps$ to 10 $Mbps$ where the load between the router and the user is very low. Nowadays, newer Ethernet cards with link speeds of 1 $Gbps$ or more between the router and the user result in lesser load. In the second scenario the authors generate heavy traffic for the link in one direction (continuous traffic), while in the other direction only "ACKs" are transmitted and the link is very low loaded. The third scenario has a medium traffic of about 10-17% in each direction. In the last scenario various traces from Google' s data centers have agnostically been used.

Except from the first scenario where the links save about 90% of energy, the rest of the scenarios perform very poorly in terms of energy consumption. We can see in [4] that when the frame size reaches its maximum value (1500 $Bytes$) the power consumed is proportional to the traffic, while for medium and small packet sizes the power consumption of EEE approaches 100%, i.e., same as legacy Ethernet links.

5.1.3 Performance Measurements of Real Cards

In [9], the authors realize measurements in NIC cards and they observe their power consumption behavior.

The experiments are made using two NIC cards with integrated EEE functionality. Since the measurements are based on the power consumed by the NIC cards, this power represents both PHY and MAC layers plus the PCIe interface and the integrated voltage regulator of the card. Accordingly, the expected results show less power saving (in percentage) than expected, since these elements (MAC and PCIe interfaces) do not benefit from EEE. Initially, different length of cables were used, to test if this factor influences the power consumption of the card but since there was only a small variation of about 2%, a fixed cable length was used for other experiments. Last but not least, the highest supported speed by the cards was mostly used for the experiments, that is 1000 $Mbps$, while in some of them 100 $Mbps$ speed was used for reference reasons.

Three scenarios were tested.

- The power consumption of the link in the case of no traffic versus a heavy loaded link.
- The power consumption of the link during state transitions.
- The power consumption of the cards for various traffic loads.

In the first scenario, the two NICs exchange a video file (continuous traffic of big packets) and the authors measure the power consumption with and without EEE. A reduction of about 70% is observed for $1000\ Mbps$ links and about 30% for $100\ Mbps$ links. The power saving is not as high as expected because the additional components of the NIC card do not benefit from the use of EEE as we previously said. Additionally, since we use the same card for both link speeds the energy overhead produced by these components is more dominant for the low speed than for the high speed.

In the second scenario, scarce traffic (which produces frequent state transitions) is generated to challenge the link at both speeds. More specifically, small packets of $250\ bytes$ are used with a data rate of $10\ Mbps$ so the packets arrive almost equally spaced every $T_S + T_W$ and the link is either in state A or in one of the transition states. The power consumption for this case is similar to the power unaware schemes. So, we understand that the power consumption of transition states is similar to the power consumption of the state A.

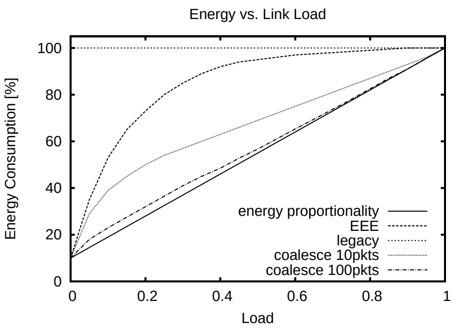
In the last scenario the video file is again exchanged for various data rates over $1000\ Mbps$ links. Two observations were made. First, when the link load is over 6% there is no power saving, because we have a huge energy overhead due to continuous EEE state transitions. Second, we observe that the link is active in both directions because we have high data traffic in one direction and we receive scarce ACKs traffic in the other direction.

5.2 Boosting Energy Efficiency with Packet Coalescing

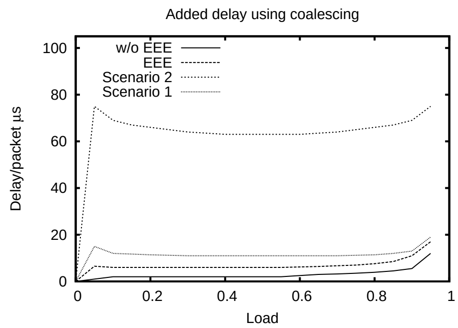
The previous evaluations shows that EEE behaves quite well when there is scarce traffic on the links and when large packets of $1500\ bytes$ are transmitted over the network, like for video files. However, when there is heavier traffic (e.g., $load > 15\%$) then EEE behaves similar to power-unaware schemes. This is because of the overhead produced by the state transitions of the links. For instance, with a traffic pattern of small packets ($100 - 200\ bytes$) equally spaced between them (every 150-200 μs), a $1000\ Mbps$ link will spend 95% of the time in EEE state transitions. So, since the idea of a “low power” state is brilliant and this is the method that is usually suggested in order to reduce the power consumption of a device (e.g., base stations, mobile devices, monitors, laptops, smart phones, televisions), further enhancements towards this direction are needed to achieve the initial goal for power reduction even when traffic is not very low and/or packet sizes are small. As we previously said, one possible policy going in this direction is the hardware optimization using more efficient electronics to transit faster (open research area for hardware producers) and a second possible policy is the use of efficient algorithms to handle both the packets and the link and remain for greater time in state LPI.

In this section we shortly investigate on the second policy, namely packet coalescing, and we show an efficient algorithm to reshape traffic and handle packet more efficiently in terms of power. Initial studies have been realized in [18, 19]. The basic idea is as follows: when there are no packets to transmit switch to state LPI, then store the new arriving packets in a buffer and wait until the buffer fulls or a timer expires. Then switch to state A and transmit. Next, we get into more details about how this policy works.

Packet coalescing wants to further improve the power efficiency of EEE. By collecting bursts (set of packets) of traffic, the link remains in the state LPI for more time and, depending on the maximum timeout that we allow the link to remain in state LPI, we save more or



(a) Power Consumption with coalescing.



(b) Delay of packets using various coalescing sizes.

Figure 5.2: Impact of coalescing on power save and on delay (data from [18]).

less power. For implementing coalescing we need two things.

- A coalescing buffer to collect and store the packets. Simulation results in [18, 19] show that depending on the size of packet coalescing, EEE can obtain results very close to the energy proportionality. The bigger the buffer size the more the power saving as we see in Fig. 5.2a. Indeed, Fig. 5.2a illustrates that a burst of ten packets reduces the power consumption by half than EEE without a coalescer. With a burst of hundred packets we see that the power consumption approaches the ideal case (data from [18]).

- A timer $T_{coalesce}$ can trigger the buffer interface to start sending packets and empty the coalescing buffer in cases of low traffic, to avoid waiting for hours in case of scarce traffic, e.g., a few "ACKs". So, either the buffer fulls and we start sending the packets together with the ones that arrive, or we wait until the timer expires.

However, there are two main drawbacks due to coalescing. First, the added delay to the packets can reach at least $T_{coalesce} + T_W$ for the worst case and second, sending bursts of packets to the network where in the worst case they are as big as the buffer size is not a desirable property for IP networks. In fact, bursts will flood the network and may cause buffer overflows to intermediate routers. Both of these drawbacks degrade the network performance.

In Fig. 5.2b (data taken from [18]) we illustrate the impact of large buffers and long timers on packet delay. Compared with power-unaware schemes, the first scenario (ten packets and 12 μs timer) adds an average packet delay of 10 μs while the second scenario (hundred packets and 120 μs timer) adds an average packet delay of about 60 μs. These results show that coalescing can have, potentially, a very reduced impact on delay, while allowing EEE to save energy as in low traffic/big packet scenarios. The authors went one step further and they simulate some Internet-like topology measurements with ns-2 simulator, to evaluate the impact of packet bursts on power saving and packet delay. They show that packet bursts save significant power with negligible latency (tens of ms) due to burstification.

Chapter 6

Conclusions

Until recently, there were not many enhancements in the network core in terms of energy efficiency. Higher network speeds, bigger capacity, fast routing are the results of an increased demand of network services such as video files, movies, online TV, etc. Network companies deployed a lot of kilometers of copper and fiber together with routers and servers and a lot of new Ethernet ports especially in developing countries to satisfy the customers and adapt to their needs. Furthermore, higher Ethernet speeds require more power to operate properly since they have more complex electronics in order to meet the standards. This led to an increased power consumption. Thus, a new Ethernet standard was proposed by IEEE, to deal with energy efficiency, namely Energy Efficient Ethernet-EEE (IEEE 802.3az).

The idea behind EEE is simple. Four new states are introduced, “Active”, “Low Power Idle” (LPI), “Sleep” and “Wake Up”. When there is data to send over the Ethernet link the link operates as usual in “Active” state but when there is no data to send, it transits to LPI state, consuming only 10% of the initial power. The transition time between “Active” and LPI state defined as the “Sleep” state and the transition time between LPI and “Active” state defined as the “Wake Up” state.

A priori, this method looks fine when the traffic is scarce (<15%) and long intervals exist between the packets. In contrast, in cases where traffic is heavy and short intervals separate the packets, EEE may not provide significant improvement since it has a lot of power overhead due to state transitions. Therefore, in this thesis, we studied the power save in real scenarios and reported on novel proposals aiming at enhancing EEE performance when traffic is not scarce and spacing among packets is short. The main goal of our work was to collect and analyze unique data on bidirectional traffic in a real network, and estimate the potential savings that can be achieved by adopting the recently released IEEE Standard 802.3az instead of the legacy Ethernet. Overall, we observed traffic patterns yielding the possibility to save at least 40%, and up to more than 90%, in each gigabit link. Such a power saving would represent a non-negligible operational cost reduction for a data center, in the order of several hundreds of thousands of Euros per year. Moreover, we analyzed the importance of high precision traffic measurements on the power saving estimation. Our analysis unveils that precise timestamping is needed to use analytical models, which allows EEE power saving estimation with no need for time consuming simulations.

Another goal of our work was to describe various techniques that have been proposed to reduce the power consumption of EEE even in cases of high traffic or short spacing among the packets. These techniques are based on packet coalescing which can improve power consumption to energy proportionality, i.e., the power consumption can be made proportional to the traffic load at the price of additional packet delay.

References

[1] European Commission (DG INFSO), *Study of the impacts of ICT on energy efficiency*, 2007

[2] C. Gunaratne, K. Christensen, B. Nordman, S. Suen, *Reducing the energy consumption in Ethernet with Adaptive Link Rate(ALR)*, IEEE Trans. Computers, vol. 57, no. 4, pp. 448-461, 2008.

[3] IEEE Std 802.3az: *Energy Efficient Ethernet-2010*.

[4] P. Reviriego, J.A. Hernandez, D. Larrabeiti, J.A. Maestro, *Performance Evaluation of Energy Efficient Ethernet*, IEEE Communication letters, vol. 13, no.9, pp. 697-699, Sep. 2009.

[5] R. Sohan, A. Rice, A.W. Moore, *Characterizing 10 Gbps Network Interface Energy Consumption*, 35th IEEE Conference on Local Computer Networks, Oct. 2010.

[6] M. Gupta, S. Grover, S. Singh, *A feasibility study for power management in LAN Switches*, IEEE International Conference on Network Protocols, pp.1092-1648, 2004.

[7] B. Nordman, *EEE savings estimate*, IEEE 802.03az Task Force Information, Mar. 2008.

[8] V. Mancuso, A. Chatzipapas, *On IEEE 802.3az Energy Efficiency in Web Hosting Centers*, to appear in IEEE Communication Letters, 2012.

[9] P. Reviriego, K. Christensen, J. Rabanillo, J.A. Maestro, *An Initial Evaluation of Energy Efficient Ethernet*, IEEE Communication letters, No. 5, Vol. 15, May 2011.

[10] M. Ajmone Marsan, A. Fernandez Anta, V. Mancuso, B. Rengarajan, P. Reviriego Vasallo, G. Rizzo, *A Simple Analytical Model for Energy Efficient Ethernet*, IEEE Communication letters, No. 99, pp. 1-3, June 2011.

[11] R. Jain, S. Routhier, *Packet trains-Measurement and a New Model for Computer Network Traffic*, IEEE Journal of Selected Areas in Communications, vol.4, no.6, pp.986-995, Sep. 1986.

[12] L. Kleinrock,*Queueing Systems, volume 1: Theory*, Wiley 1975.

[13] R. Sohan, A. Rice, A.W. Moore, K. Mansley, *Characterizing 10 Gbps Network Interface Energy Consumption*, in 35th IEEE Conference on Local Computer Networks, Oct. 2010.

[14] S. Bapat, *The Future of Data Centers (... and the Stuff That Goes In Them)*, http://www.e3s-center.org/events/09/e3s-symposium.htm, June 2009.

[15] Google, *Google Data Center Video Tour*, http://www.google.com/about/datacenters/events/2009-summit.html#tab0=4, Apr. 2009.

[16] U.S. Environmental Protection Agency, Energy Star Program, *Report to Congress on Server and Data Center Energy Efficiency*, Public Law 109-431, Aug. 2007.

[17] L.A. Barroso, U. Holze, *The case for energy-proportional computing*, IEEE Computer, vol. 40, no. 12, pp. 33-37, Dec.2007.

[18] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, J.A. Maestro, *IEEE 802.3az: The Road to Energy Efficient Ethernet*, IEEE Communications Magazine. Vol.48, no.11, pp.50-56, Nov. 2010.

[19] P. Reviriego, J.A. Maestro, J.A. Hernández, D. Larrabeiti, *Burst Transmission for Energy Efficient Ethernet*, IEEE Internet Computing, Jul. 2010.

[20] C. Walsworth, E. Aben, K.C. Claffy, D. Andersen, *The CAIDA anonymized 2009 Internet Traces*, http://www.caida.org/data/passive/passive_2009_dataset.xml.

[21] K.C. Claffy, D. Andersen, and P. Hick, *The CAIDA anonymized 2010 Internet traces*, http://www.caida.org/data/passive/passive_2010_dataset.xml.