POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Ingegneria Elettronica e delle Comunicazioni

–

XXIV ciclo

Tesi di Dottorato

Energy aware control algorithms for computer networks

Marco Ricca

Tutore
prof. Paolo Giaccone

Coordinatore del corso di dottorato
prof. Ivo Montrosset

Febbraio 2012

Summary

This thesis describes the research activity that was carried out in the *Telecommunication Network Group* at the *Electronics and Telecommunication Department* (former *Electronics Department*) of the *Politecnico di Torino*.

The main motivation of this work is to investigate techniques to reduce the power consumption inside a network element. It is enough to consider the high energy demand associated to the telecommunication networks field. As practical consequence the power consumption has become a relevant parameter and it represents a critical constraint for the network designers looking both the whole network infrastructure and the network elements like switches, routers and servers.

The PhD has been focused mainly on two research areas of interest, the first one was the power consumption inside the switching fabric of an high speed router. The target was to analyze the effect of the dynamic power inside a switching fabric, to evaluate a set of optimization strategies in order to minimize the power consumption and to achieve the best trade-off between power, high performances and packet delays; the crossbar was used as reference switching architecture for this study. Looking at the consumption side, generally speaking, it is possible to define two families of switching fabrics:

- **Bit-rate independent switching fabric**, in which the consumption does not depend on the number of transported bits; this family is typical of optical switching fabrics
- **Bit-rate dependent switching fabric**, where the total consumption is proportional to the data transmission bit-rate, this family is typical of electronic switching fabrics

The second research activity was carried at the *Alcatel-Lucent Bell Laboratories*, based in New Jersey (USA) and over a period of 9 months. The study of the power consumption across several network elements that are commercially available for the "corporate" market.

We started from a set of collected larger number of power measurements over these network elements and thanks to them we were able to develop a linear mathematical model to describe the power consumption of a generic network element.

Acknowledgements

So, it was a long but funny story, a nice journey...

First of all I wish to thank my supervisor Paolo, or in "a more official way" Prof. Paolo Giaccone, for his support during my research activities and for his patience. During this three-year period I made agaist with him at least about a thousand of bets on my results (and I had lost all of them.. try to figure out how many "ice creams" I had to pay).

I would like to thank all the people in the Telecommunication Network Group in Torino, in particular Prof. Andrea Bianco who provides an important contribution and feedback on my work.

A special thanks is for Andrea Francini who gave the opportunity to visit USA and the Bell Laboratories.

The last, but probably more important, "thank you" is for the patience and support of my parents, my brother, my family who experienced all ups and down of my research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Network elements and power costs

What is, in a whole picture, the total energy and total power consumption of Internet? It is not easy to get a unique answer, considering the continuous emergence of new technologies and their utilization over Internet, but it is well known that Internet power consumption is still growing due to the increasing number of network elements connected together [1]. Only in the USA region it represents more of the 2% of overall power consumption and it has been estimated that it is going to grow up to 8% [2].

The foresight for the whole telecommunication network sector reports that the overall consumption will increase threefold until 2020: from 150 GW to 450 GW [3]. Making a comparison to 2000, the overall power consumption has increased twice as much and the consumption associated only to the network elements represents the 40% of all.

In a packet telecommunication network, roughly speaking, data among users are transfered across network elements (like routers, switches and servers) through communication links.

Focusing on network elements that are commercially available, they are always operative but often most of them are underutilized: this represents a large waste of power consumption. As a consequence, this single contribution causes a twofold outcome: from one side the power demand requested for the telecommunication networks sector is still growing, on the other side it is possible to take account this constraint for the design of future generations network elements.

1.2 How to reduce power?

The roadmap drew to reduce power consumption costs associated to telecommunication networks sector, suggests a twofold optimization over the whole network resources and over network nodes.

The following example can be used as toy scenario. Assume that a source wants to send data to a destination and assume that there are multiple paths to reach the destination. Now, let suppose that one path is underutilized, i.e. the transmission bandwidth is lower than the maximum bandwidth available over that link. In one case, that link can be turned off and the traffic data flow to another one. In the latter, transmitter and receiver can be slowed down the speed of that link. In both cases, through these actions, it is possible to save power.

The discussion over this topic has been appeared insistently since 2000 and it is strongly connected to the claim to achieve high performances and scalability. It is enough to remember the transition from the *Mbit/s* to the *Gbit/s* domain.To achieve these results the price to pay is a significant growth in term of power consumption.

The aim of this work is to investigate techniques to reduce the power consumption inside a network element. Next sections describe a brief overview of this work. On one side it was considered the analysis of the power consumption for switching fabrics described in Section 1.3. The switching fabric is only a component of a network element, like an high speed router. As a consequence, described in Section 1.4, there is the energy profiling characterization of a commercial available network element in order to exploit "rate adaptation" techniques to reduce the power consumption.

1.3 Power aware switching fabric

Focusing on current generation of network devices, the aggregate bandwidth of high speed routers is growing fast, due to the increased traffic demand in the Internet. To support traffic growth, in core routers a switching fabric that must switch data at increasing speed is often implemented on a single integrated circuit. As a results, power consumption in high speed switching fabrics has become one of the most critical design issues, mainly due to high integration level on a single chip, that implies very high power spatial density [18].

The power consumption of a packet switching fabric is a sum of many contributions: supply power, data transfer and control power. Depending on the specific employed technology, the relative importance of each contribution is different.

Traditional electronic switching fabrics are based on CMOS technology. Roughly speaking, in an electronic crossbar, which is one of the simplest and most widely deployed switching fabric architectures, the output line is connected to the input line through a logic gate. The activation of the logic gate corresponds to selecting the

proper crosspoint in the crossbar fabric. The power consumption depends strongly on the electric charges that are moved to charge/discharge the input/output lines, for each bit transmission. Thus, the total power depends mainly on the amount of data transferred, and it is increasing with the bit rate.

The hardware design of such fabric is becoming more and more critical, because of the large pin count and the high bit rate. Indeed, if f is the maximum digital signal frequency, the power consumption of a CMOS device is proportional to f^3 [17]. In a $N \times N$ single-chip crossbar with N^2 crosspoints, each implemented through proper logic blocks, there are[1] $\Theta(N^2)$ CMOS components (i.e., a fixed number for each crosspoint), and the total power consumption becomes proportional to R^3N, where R is the data-transmission bit rate and N is the maximum number of data simultaneously flowing across the switching fabric.

In integrated circuits, Dynamic Voltage and Frequency Scaling (DVFS) [17], a classical technique used to control the power consumption, is based on the idea of jointly varying the power supply voltage and the peak signal frequency.

The main idea is to reduce the power when the traffic load is low, extending the packet transmission duration through bit voltage and frequency reduction. Indeed, networks are typically provisioned for worst-case or peak-hour traffic. However, several measurements (see for example [19]) show that backbone utilization rarely exceeds 30%, thus suggesting that exploiting low traffic conditions can be a significant asset to reduce power.

We propose a set of algorithms for power control that operate on an estimated traffic matrix to assess the potential power gain that can be obtained exploiting DVFS. We take an idealized approach based on fluid model, i.e., we disregard the interaction with packet scheduling algorithms that select the packets to be transferred across the switching fabric.

In electronic switches, it is critical the high density of power to dissipate on a single chip. Considering the power for the data transfer, another way to address this problem can be to consider as alternative the optical switching architecture. Optical switching fabrics usually offer a good scalability with the line rate. This is mainly due to the fact that the optical device dynamics are decoupled from the bit rate: the energy (and power) consumption is largely independent of the number of transported bits and depends mainly on the power supply and on the switch control. This fact holds for some optical linear analog switches [4] and for switching technologies based on latching electromechanical systems (like MEMS [5] or NEMS [6]).

Motivated from such optical technologies, we focus just on the power spent *to change the configuration of the switching fabric*, assuming that the amount of energy

[1] In Landau notation, function $g(n)$ is $\Theta(h(n))$ if, for $n \to \infty$, $k_1 h(n) < g(n) < k_2 h(n)$ for some positive constants k_1 and k_2.

depends only on the number of input-output connections added and/or removed inside the switching fabric.

We study how to achieve high throughput while minimizing the number of connections that change inside the switching fabric. Intuitively, our approach is based on changing the switching configuration in a "lazy" way, i.e., trying to keep the switching configuration as similar as possible in consecutive timeslots. Since the power is the energy averaged on a time scale much larger than the packet duration, we propose a frame-based approach, in which the packet scheduler pre-computes the configuration of the switching fabric once during the frame duration.

Under both scenarios, electronic and optical switching fabrics, we consider specifically IQ (Input-Queued) switches, since they are the reference architectures for high-performance packet switches thanks to their good scalability. Indeed, memory access speed does not increase linearly with the number of switch ports, as in OQ (Output-Queued) architectures. We only concentrate on the power of the crossbar chip, not considering the power contribution of other components of the switching architecture.

1.4 Power characterization of network elements

In packet networks, the term "rate adaptation" designates a broad set of methods aimed at establishing a direct relationship between sustained workload and energy consumption. In an ideal framework for energy efficiency, the network design is optimized to minimize energy consumption under full-load traffic conditions [26].

Rate adaptation additionally ensures that the energy-workload function is linear and that the network consumes no energy when there are no packets to transport [27]. To support such behavior, rate adaptation schemes provide the network systems with a discrete set of operating states, where each state maps a fixed traffic processing rate onto a respective power consumption level. The scope of the control exercised by a rate adaptation scheme can range from large subsets of network links and nodes [28], [29], [30] to individual sections of a single traffic processing chip [31]. Hence, for the sake of clarity we partition "rate adaptation" techniques based on their timescale of operation, which is defined by the switching time needed to transition between states and ultimately depends on the size of the targeted system.

Demand-timescale rate adaptation (DTRA) techniques control the state of network links and nodes based on expected or measured trends in traffic demands between network endpoints [28], [29], [30]. DTRA state transitions involve network signaling and system-level power cycles, so their timescale ranges from seconds to minutes. Packet-timescale rate adaptation (PTRA) techniques adjust the clock frequency and supply voltage of data-path hardware components to locally maintained

workload indicators such as queue lengths and traffic arrival rates [32], [33]. The timescale of PTRA state transitions ranges from microseconds to milliseconds depending on the underlying integrated circuit technology. Bit-timescale rate adaptation (BTRA) also applies to data-path hardware components. Compared to PTRA, BTRA transitions are much faster to execute (down to nanoseconds) because they only involve control of the system clock (e.g., by gating of the clock signal), at the expense of reduced power savings. To assess the energy-saving benefits that may derive from the application of the different types of rate-adaptation techniques we conduct power measurement experiments on a set of network systems that are commercially available. In the case of network-wide DTRA techniques, the energy profiles that result from the measurements quantify the benefits of enabling and disabling network ports and possibly also entire line cards and systems based on expected traffic demands; in the case of PTRA and BTRA techniques, instead, the profiles identify the energy saving margins that are available for the introduction of rate adaptive hardware components.

The energy profile of a network element maps system and traffic configurations onto power consumption levels, typically by means of a simplified linear model. Examples of system configuration variables can include the number of cards plugged into the chassis (in slotted systems), the number of ports that exchange traffic over network links and the transmission capacity provisioned for those ports. Traffic configuration variables include the traffic arrival rate at each network port and the statistical distribution of packet sizes and packet inter-arrival times at ports where traffic is present. While energy profiles are commonly available for computing systems and processors, studies that focus on networking systems and components have started appearing in the literature only recently [34], [35], [36] and suffer from important limitations. In fact, as we discuss below, the energy profiles presented in those studies are not always complete in the identification of system configuration variables [34] or in the modeling of critical system components [35], or rely mostly on manufacturer power-rating data rather than experimental measurements [36]. Profiling approaches that condense the energy-efficiency properties of a system into a small number of scalar indices [26], [37], [38] are unfit to support the fine-tuned state-setting decisions that are at the core of all rate adaptation methods.

1.5 Organization of the thesis

We focus in Chapters 2 and 3 on the problem of minimizing the power spent to *control* the configuration of the switching fabric, by neglecting the contribution due to the data transfer and to the supply power. After the characterization for crossbar-based IQ switches, Chapter 4 is focused on the energy profiling made over a set of different network equipments available for commerce in order to highlight

the achievable benefits introduced with "rate adaptation" technologies. Exploiting these technologies we assess how compatible they are with existing network elements and identify the design upgrades that can maximize their energy savings in new generations of network systems. Finally, Chapter 5 draws the conclusions of this thesis.

1.5.1 Power control for crossbar-based input-queued switches

Bit-rate independent crossbar

Chapter 2 is organized as follows. Section 2.1 defines the scheduling problem and Section 2.2 describes the algorithms we propose to solve it.

The performances of such algorithms are investigated in terms of energy and throughput both analytically and by simulation under different traffic scenarios in Section 2.3, whereas Section 2.4 discusses the performances in terms of delays. Finally, Section 2.5 discusses the related work.

Bit-rate dependent crossbar

In Chapter 3 we propose to exploit DVFS for the power control of a single-chip crossbar, to reduce the power consumption at the cost of increasing packet delays at low-medium loads without sacrificing switch throughput.

The Chapter 3 is organized as follows. The system model is defined in Section 3.1, while Section 3.2 formalizes the optimal crossbar chip power control problem, describes its properties, and proposes a set of algorithms to solve it. Performance results in Section 3.3 show the possible power gain of our approach.

Details on the hardware architecture for a 410 Gbps crossbar are provided in Section 3.4, where we show that the synthesis results well fit those of the theoretical model.

1.5.2 Energy profiling of network element

In Chapter 4, an overview of existing models for energy profiling and instances of "rate adaptation" techniques from the literature is described in Section 4.1. Section 4.2 lists technical specifications for the network systems that we target for energy profiling. Section 4.3 describes the auxiliary equipment of our experimental testbed.

In Section 4.4 we introduce our new model for energy profiling. In Section 4.6 we illustrate the measurement methodology that we follow for estimation of the parameters of the linear model.

In Section 4.7 we present and discuss the results of our power measurement experiments.

Chapter 2

Frame-Scheduling with Energy Reconfiguration Costs

2.1 Problem Definition

We consider an $N \times N$ synchronous (slotted) IQ switch as shown in Fig. 2.1. Time is slotted and input and output ports are assumed to be slot synchronized. Fixed-size packets are received and stored at inputs. Input queues are organized according to the classical *Virtual Output Queueing* (VOQ) architecture. Under this architecture there is one separate *First In First Out* queue (FIFO) at each input port for each output port, for a total of N^2 queues in the switch. The queue VOQ_{ij} stores, at input port i, packets that must be routed to output port j. The IQ architecture ensures high scalability in line rate and number of ports, and the VOQ scheme is theoretically optimal from the performance point of view.

At each timeslot, a packet scheduler [22] chooses a switching configuration i.e. an input/output port interconnection pattern to select the set of packets transferred simultaneously through the crossbar. This configuration satisfying the constraints that at most one packet is sent from each input and to each output, to avoid output conflicts in a timeslot.

This problem can be modeled as a matching problem in a bipartite graph. Each left hand side vertex corresponds to an input and each right hand side vertex corresponds to an output. An edge connects input i to output j if the corresponding VOQ_{ij} is not empty. In each timeslot the scheduler computes a *matching*, i.e. a subset of edges with no vertex in common, corresponding to a feasible switching configuration. A graphical example of this problem is shown in Fig. 2.2 where $N = 5$, the bipartite graph is shown on the left side of Fig. 2.2 while the scheduler choice, i.e. a matching, is represented on the right side Fig. 2.2.

A matching can be represented by an $N \times N$ binary matrix $M = [m_{ij}]$, denoted

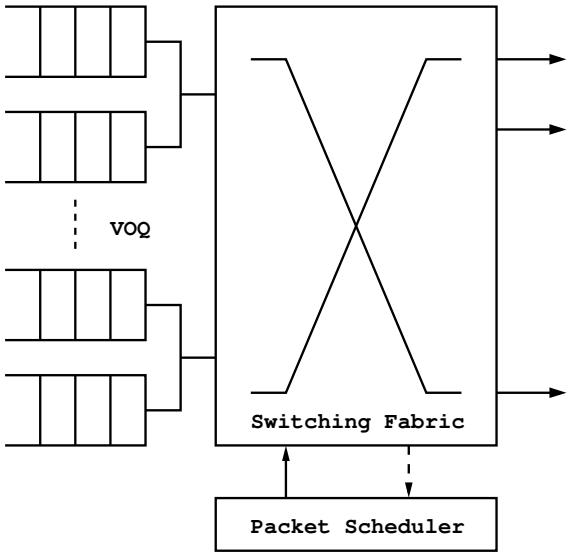

Figure 2.1. Logical structure of an IQ switch with VOQ architecture

as *matching matrix*, in which $m_{ij} = 1$ if and only if input i is connected to output j, and at most one element is set to 1 in each row and in each column:

$$\sum_{k=1}^{N} m_{ik} \leq 1 \qquad \sum_{k=1}^{N} m_{kj} \leq 1 \quad \forall i, j.$$

The set of all matching matrices is denoted by $\mathcal{M}$. A matching M is:

- *complete* if exactly one element is set to 1 in each row and in each column, and it is represented by a permutation matrix

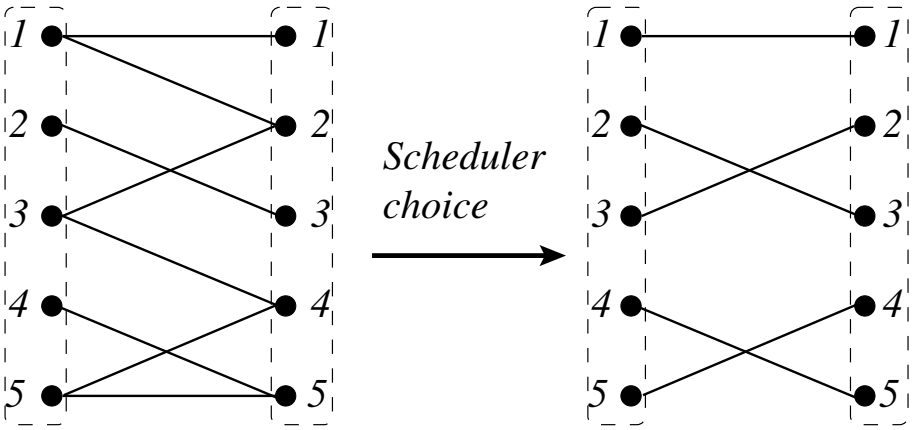

Figure 2.2. Bipartite graph (left) and a proposed matching (right)

- *non-null* if at least one element is set to 1:

$$\sum_{i,j} m_{ij} \geq 1$$

2.1.1 Energy Model

The goal is to minimize the energy consumption required to modify the switching configuration in consecutive timeslots. More precisely, if input i was connected to output j at timeslot t, and input i becomes connected to output $k \neq j$ at timeslot $t+1$, two energy costs arise: e_d is the energy required to delete the connection from input i to output j and e_a is the energy required to add the new connection from input i to output k. The total energy cost of a new switching configuration is obtained as the sum of the energy costs required to delete all connections selected in the previous timeslot and not selected in the current timeslot, plus the energy cost required to add all connections selected in the current timeslot and not selected in the previous one.

The actual values of e_d and e_a depend on the considered switching technology; e.g. in the case of bi-stable latching MEMS, with forces acting on the micro-mirrors only when mirrors change position, we can assume $\hat{e} = e_d = e_a$, i.e. the energy cost to remove or to add a connection is the same. In this case, the minimum value of energy required to modify the output at which an input is currently connected is $2\hat{e}$. Note that the approach here presented can be easily extended to the general case of $e_d \neq e_a$ or to other values of energy consumptions. In the following, we normalize the energy costs to $\hat{e}$ or, equivalently, set $\hat{e} = 1$.

Define $E\left(M^h, M^k\right)$ as the total amount of energy spent to modify matching $M^h = [m^h_{ij}]$ in matching $M^k = [m^k_{ij}]$. The total amount of energy can be computed by counting the number of edges that are either removed from M^h or added to M^h to obtain M^k:

$$E\left(M^h, M^k\right) = \hat{e} \sum_{i=1}^{N} \sum_{j=1}^{N} |m^h_{ij} - m^k_{ij}|.$$

By construction, $E\left(M^h, M^k\right) = E\left(M^k, M^h\right)$.

2.1.2 Frame Scheduling

We assume that the scheduler operates on a frame basis [7]. The scheduler samples the state of input queues at the beginning of a *sampling period*, that lasts T timeslots; i.e. the queues are sampled at timeslot $t = nT$, for any $n \in \mathbb{N}$. Then it computes a *frame*, i.e. a sequence of matchings, to empty the input queues before the next sampling period. Finally, the switching fabric is configured according to the frame to serve the packets during the current sampling period, i.e. for $nT \leq t < (n+1)T$.

If the queues are not empty at $t = (n+1)T$, the residual packets are kept and will be served in one of the subsequent sampling periods. Note that the two phases of computing the frame and serving the packets can be pipelined in subsequent scheduling periods; this allows to amortize the time to compute a new frame on the whole sampling period, at the acceptable cost of increasing the delays of T timeslots.

Let $R = [r_{ij}]$ be an $N \times N$ request matrix, where r_{ij} is the number of packets enqueued at VOQ_{ij}, sampled at timeslot $t = nT$, for $n \in \mathbb{N}$. The maximum row and column sum of R is denoted by T_R:

$$T_R = \max \left\{ \max_{j=1...N} \sum_{i=1}^{N} r_{ij}, \quad \max_{i=1...N} \sum_{j=1}^{N} r_{ij} \right\}$$

The *frame maximum load* is defined as $\rho = T_R/T$ and R is said to be *admissible* when $\rho \leq 1$.

The frame $\mathcal{F}_R^{\mathcal{A}}$ computed by a specific scheduler $\mathcal{A}$ on the request matrix R is defined as an ordered sequence of K distinct and non-null matchings:

$$\mathcal{F}_R^{\mathcal{A}} = \{(M^k, \phi_k)\}_{k=1}^{K}$$

with $M^k \in \mathcal{M}$ and $\phi_k \in \mathbb{N}$ is the number of consecutive timeslots in which matching M^k is used to configure the switching fabric. Each distinct matching appears always in consecutive timeslots to minimize the considered energy cost. To serve all packets in R, it must hold:

$$R = \sum_{k=1}^{K} \phi_k M^k \tag{2.1}$$

Let $F_R^{\mathcal{A}} = \sum_{k=1}^{K} \phi_k$ be the frame duration, i.e. the total number of slots used to transfer packets and empty the queues. Note that T is fixed, whereas $F_R^{\mathcal{A}}$ varies with R. An admissible request matrix R is said to be *sustainable* by scheduling algorithm $\mathcal{A}$ if during a sampling period *all* the packets in the request matrix are transferred, i.e. if

$$F_R^{\mathcal{A}} \leq T \tag{2.2}$$

Due to the Birkhoff-von Neumann theorem [23], the minimum frame duration to serve R is T_R slots. In general, $F_R^{\mathcal{A}} \geq T_R$ and we define the *frame-expansion factor* S as[1]:

$$S = \frac{F_R^{\mathcal{A}}}{T_R} \tag{2.3}$$

Combining (2.2) and (2.3), R is sustainable if $\rho S \leq 1$. We assume that R is always sustainable, thus $1/S$ can be seen as the normalized *maximum sustainable load*

[1]Even if S depends on R and $\mathcal{A}$, we omitted them from the notation for the sake of conciseness

according to an algorithm $\mathcal{A}$. The total *energy* cost to configure the switching fabric according to frame $\mathcal{F}_R^{\mathcal{A}}$ is:

$$E(\mathcal{F}_R^{\mathcal{A}}) = \sum_{k=1}^{K-1} E(M^k, M^{k+1})$$

Note that $E(\mathcal{F}_R^{\mathcal{A}})$ is independent of the values of ϕ_k. The corresponding *power*, evaluated over the sampling period, is:

$$P(\mathcal{F}_R^{\mathcal{A}}) = \frac{E(\mathcal{F}_R^{\mathcal{A}})}{T}$$

In this work we aim at finding scheduling policies that maximize throughput (i.e., the maximum sustainable load) and minimize the power, under a generic request matrix.

2.1.3 A Toy Example

To understand the possible tradeoffs between throughput and power, we consider the case of a constant uniform request matrix R, where $r_{ij} = u$, $\forall i, j$, and u is a fixed positive integer. In this case, $T_R = Nu$.

Let $D^k = [d_{ij}^k] \in \mathcal{M}$ be the permutation matrix corresponding to the i-th diagonal, i.e. $d_{ij}^k = 1$ if and only if $|(i-1) + (k-1)|_N = (j-1)$ for some positive integer k, where $|x|_N$ is the module-N operator. Let us consider three possible frames, all of them satisfying (2.1):

- $\mathcal{F}_1 = \{(D^1,1), \ldots, (D^N,1), \ldots, (D^1,1), \ldots, (D^N,1)\}$: the matchings are cyclically selected among all the N diagonals in a round robin fashion, keeping each matching for one timeslot.
- $\mathcal{F}_2 = \{(D^1, u), \ldots, (D^N, u)\}$: the matchings are cyclically selected among all the N diagonals in a round robin fashion, keeping each matching for u timeslot. The same matching is used only in consecutive timeslots within the frame.
- $\mathcal{F}_3 = \{(U^{11}, u), \ldots, (U^{1N}, u), \ldots, (U^{N1}, u), \ldots, (U^{NN}, u)\}$, where U^{ij} is a matching with only one edge, from input i to output j.

Table 2.1 reports the corresponding sustainable load (when $T = T_R$) and power.

From the throughput point of view, $\mathcal{F}_1$ and $\mathcal{F}_2$ provide an optimal scheduling, whereas $\mathcal{F}_3$ is inefficient. From the power point of view, $\mathcal{F}_2$ and $\mathcal{F}_3$ provide an optimal scheduling, because they use the same matching in consecutive timeslots. As a conclusion, under such constant traffic matrix, $\mathcal{F}_2$ is the best frame, achieving optimality in terms of both power and throughput.

Table 2.1. Power consumption and performance for constant uniform request matrix

Frame	$F_R^{\mathcal{A}}$	S	Max sustainable load	Energy per frame
$\mathcal{F}_1$	T_R	1	1	$2uN^2$
$\mathcal{F}_2$	T_R	1	1	$2N^2$
$\mathcal{F}_3$	NT_R	N	$1/N$	$2N^2$

2.2 Energy-Aware Frame Scheduling

Our energy-aware frame-scheduling problem can be modeled as a two-objective optimization problem: define a frame that minimizes the energy consumption, due to switching fabric reconfigurations whilst maximizing the throughput (or, equivalently, minimizing the frame duration $F_R^{\mathcal{A}}$).

We solve this problem in two steps the first one to maximize the throughput and the second one to minimize energy:

- **matching selection**

  given the request matrix R, define an algorithm $\mathcal{A}$ that computes an *unordered* frame $\mathcal{U}_R^{\mathcal{A}} = \{M^k, \phi_k\}_{k=1}^K$ such that

  1. condition (2.1) is satisfied
  2. condition $|\mathcal{U}_R^{\mathcal{A}}|$ the corresponding frame duration is minimized

  The objective is to serve all the packets in R to maximize throughput.

- **frame sorting**

  compute the final frame $\mathcal{F}_R^{\mathcal{A}}$ by ordering $\mathcal{U}_R^{\mathcal{A}}$ to minimize the energy consumption due to switching reconfigurations.

2.2.1 Matching Selection

We consider five different algorithms for the matching selection. The first four are iterative algorithms, exploiting the same generic decomposition algorithm *Gen-DEC*, whose pseudo-code is reported below.

At each iteration of *Gen-DEC*, a specific algorithm $\Omega(R)$ computes a matching matrix M on R. Then, the value of the minimum element in R among those selected by the matching matrix M is subtracted from all selected elements in R, and a residual request matrix is obtained. The process iterates until R becomes empty.

Since, at each iteration, at least one element (at most N elements) of R becomes zero, N^2 iterations are needed in the worst case to fully schedule R.

Gen-DEC (*Input*: R; *Output*: $\mathcal{U}_R$)

$\mathcal{U}_R = \emptyset, k = 1, R(k) = R$ // initialize

while $R(k) \neq 0$ // while $R(k)$ is not completely zero

{

$M^k = \Omega(R(k))$ // find a matching

$\phi_k = \min_{1 \leq i,j \leq N} \{m_{ij}^k r_{ij}(k) | r_{ij}(k) > 0\}$ // find minimum

$R(k+1) = R(k) - \phi_k M^k$ // subtract

$\mathcal{U}_R = \mathcal{U}_R \cup \{(M^k, \phi_k)\}$ // frame update

$k = k + 1$ // start a new iteration

}

BvN

A *Gen-DEC* based algorithm, exploiting the Birkhoff-von Neumann decomposition [23] on R, satisfying condition (2.1). $\Omega(R)$ is a MSM (Maximum Size Matching) on R, i.e. the matching with the largest number of edges corresponding to non-null elements of R. The MSM algorithm complexity is $O(N^{2.5})$.

The BvN decomposition is "optimal", because it achieves the minimum frame duration (equal to T_R) and the minimum frame-expansion ratio $S = 1$. The overall computational complexity is $O(N^{4.5})$.

GMax

A *Gen-DEC* based algorithm, where $\Omega(R)$ is a greedy maximum weight matching on R. The algorithm selects the element in R with the maximum value, then it removes the corresponding row and column from R, and repeats the process until all the rows and columns in R are considered.

The complexity of each iteration is $O(N^2 \log N)$ (needed to sort the N^2 values in R in the initial step); hence, the overall computational complexity is $O(N^4 \log N)$.

GExa

A *Gen-DEC* based algorithm. $\Omega(R)$ is a maximal size matching with the constraint that a queue is always served in consecutive timeslots until it becomes empty. More formally, if $M_{ij}^{k-1} = 1$ and $M_{ij}^k = 0$, then $r_{ij}^k = 0$. Otherwise, on the remaining

input-output pairs, $\Omega(R)$ computes a maximal size matching. This is equivalent to the exhaustive service decomposition discussed in [8].

Since the complexity of a greedy maximal size matching is $O(N^2)$, then the overall computational complexity is $O(N^4)$.

GMin

A *Gen-DEC* based algorithm. $\Omega(R)$ is a greedy minimum weight matching on R. Thus, the algorithm chooses the smallest elements in R, then it removes the corresponding row and column from R, and repeats the process until all the rows and columns in R are considered. Thus, the complexity is $O(N^4 \log N)$.

Diag

The matching selection is based on a precomputed set of N *covering diagonals* $D^k = [d^k_{ij}]$ on R, i.e. matchings with no elements in common and able to cover all the elements in R. Formally, $d^k_{ij} d^h_{ij} = 0$ for any $h \neq k$, and $\sum_{k=1}^{N} d^k_{ij} = 1$ for any i, j. The matching duration ϕ_k is chosen equal to the maximum value of the elements in the request matrix selected by D^k, i.e. $\phi_k = \max_{i,j}\{d^k_{ij} r_{ij} | r_{ij} > 0\}$ and the frame duration is $\sum_{k=1}^{N} \phi_k$.

The total number of iterations is N, each of the iterations having a complexity $O(N)$ (the maximum value among N elements of R must be found). Hence, the overall computational complexity is $O(N^2)$.

2.2.2 Frame Sorting

In this second step of the frame definition algorithm, the matchings found in the frame $\mathcal{U}_R$ are ordered to minimize the energy consumption due to reconfigurations in consecutive timeslots. One simple way to model this problem is to consider an auxiliary graph. Each matching in $\mathcal{U}_R$ is associated with a vertex, and any pair of vertexes is connected by an edge, thus creating a complete graph by construction.

The cost of the edge connecting the vertex of M^k with the one of M^h is defined as the energy needed to move the switching fabric configuration from one matching to the other, i.e. $E(M^k, M^h)$. The cost of any path in the auxiliary graph corresponds to the energy needed to follow the particular sequence of matchings defined by the path.

The frame sequence $\mathcal{F}^{\mathcal{A}}_R$ minimizing the energy consumption can be computed from $\mathcal{U}_R$ by finding the minimum-cost Hamiltonian cycle, also known as the TSP problem, which is NP-complete. However, in our scenario, the edge costs satisfy the triangle inequality, and the problem reduces to a metric TSP [9], which is still NP-complete, but it can be simply approximated.

We consider the following algorithms to sort $\mathcal{U}_R$:

- **No-Sort** (NS) leaves the sequence of matching unmodified.
- **Best-Sort** (BS) is a greedy algorithm that finds an approximated minimum cost cycle by visiting all the vertexes: it chooses, at each step, the minimum cost edge towards an unvisited vertex. The initial vertex is chosen at random.
- **Worst-Sort** (WS) is a greedy algorithm that heuristically finds the maximum cost Hamiltonian cycle: starting from a random vertex, at each step, the maximum cost edge towards an unvisited vertex is chosen. This algorithm permits to define a worst-case sequence from the energy consumption point of view, and it is useful to highlight the impact of the frame-sorting phase.

The above frame-sorting algorithms can be freely combined with the matching-selection algorithms defined in the previous section.

As remainder we use the notation:

(matching-selection)-(frame-sorting)

to denote the particular pair of algorithms considered in our investigations: e.g. GMax-BS, GExa-NS, etc.

2.3 Performance Results

2.3.1 Traffic Scenarios

We compare the performance of the previously presented algorithms for several families of randomly generated request matrices. The choice of such families is arbitrary, but each of them is aimed to test the performance of the algorithms under some specific scenario.

The first family is denoted as *Average Sum* (AS): the matrix elements r_{ij} are i.i.d. random variables, and satisfy the constraints:

$$E\left[\sum_{i=1}^{N} r_{ij}\right] = E\left[\sum_{j=1}^{N} r_{ij}\right] = \mu N$$

i.e. the sum of each row and column is, on average, equal to a constant μN. Hence, μ represents the average number of packets arrived to each input during T timeslots. Let $\textsc{Geom}(x)$ be a geometric distribution with average x. Among the family of AS request matrices, we consider:

- *Uniform* (Uni-AS): $r_{ij} = \textsc{Geom}(\mu)$. The coefficient of variation[2] of the elements in R is $C_v = 1$. This is a common testbed used in assessing the performance of switches. Furthermore, the variance of the elements is small.

- *Bidiagonal* (Bid-AS): let $M^1, M^2 \in \mathcal{M}$ be two randomly chosen permutation matrices. Set

$$r_{ij} = \textsc{Geom}(\alpha\mu N)d^1_{ij} + \textsc{Geom}((1-\alpha)\mu N)d^2_{ij}$$

  (with $0 < \alpha < 1$), i.e. R is obtained by summing two permutation matrices with random weights for each non-null element. Computing a maximum size matching on such family of matrices is difficult using any greedy approach, and for this reason this family is considered critical during the matching selection phase.

- *Bimodal* (Bim-AS):

$$r_{ij} = \begin{cases} 0 & \text{with probability } p \\ \textsc{Geom}(\mu) & \text{otherwise} \end{cases} \tag{2.4}$$

  Since the coefficient of variation is

$$C_v = \sqrt{\frac{(1+p)\mu - 1 + p}{\mu(1-p)}} \approx \sqrt{\frac{1+p}{1-p}}$$

  we can set the values of p and μ to obtain a given C_v. For example, setting $p = 0.601$ and $\mu = 100$ gives $C_v \approx 2$. Note that, just for this scenario, the average sum of the rows and columns is $(1-p)\mu N$. This family is similar to the uniform one, but with a larger variance.

We also consider the family of Perfect Sum (PS) matrices, whose rows and columns sum exactly to a constant μN:

$$\sum_{i=1}^{N} r_{ij} = \sum_{j=1}^{N} r_{ij} = \mu N$$

Obviously, the elements $\{r_{ij}\}$ are not i.i.d.. PS matrices are an extension to the integer domain of double stochastic matrices, for which the BvN [23] decomposition was originally defined. Similarly to AS matrices, we consider the following PS families:

[2]Given a probability distribution, the coefficient of variation C_v is defined as the ratio of its standard deviation σ and its mean μ:

$$C_v = \frac{\sigma}{\mu}$$

Table 2.2. Energy per packet for Diag (and GExa) algorithm

Request matrix	Total energy	Total packets	Energy per packet
Uni-AS, Uni-PS	$2\hat{e}N^2$	μN^2	$\frac{2\hat{e}}{\mu}$
Bid-AS, Bid-PS	$4\hat{e}N$	μN^2	$\frac{4\hat{e}}{\mu N}$
Bim-AS	$2\hat{e}(1-p)N^2$	$\mu(1-p)N^2$	$\frac{2\hat{e}}{\mu}$

- *Uniform* (Uni-PS): choose a set of μN random permutation matrices $M^k \in \mathcal{M}$ and compute
$$R = \sum_{k=1}^{\mu N} M^k$$
Uni-PS matrices are characterized by elements with low variance, because, for the Central Limit Theorem, $C_v \to 0$, as $N \to \infty$.

- *Bidiagonal* (Bid-PS): let $M^1, M^2 \in \mathcal{M}$ be two random permutation matrices
$$R = \alpha\mu N M^1 + (1-\alpha)\mu N M^2$$
with $0 < \alpha < 1$.

2.3.2 Performance of Diag algorithm

The energy and throughput performance of Diag algorithm can be evaluated analytically for all the considered scenarios. The energy consumption of Diag can be evaluated easily because the energy cost is always equal to $2\hat{e}$ for each non-null r_{ij}. Hence, for a frame of k distinct matchings, with $k \leq N$, the total energy cost is always upper bounded by $2\hat{e}kN$. Note that this holds independently from the matching sorting, since the final frame $\mathcal{F}_R^A$ is independent from it. Table 2.2 provides the energy per packet for all the considered traffic scenarios.

Note that for Uni-AS and Bid-AS the energy per packet is an upper bound on the actual energy costs (due to possibly zero values in R), for Bim-AS it is an average, whereas for Uni-PS and Bid-PS this energy value is exact. In Sec. 2.3.3 we will show that the results in Table 2.2 hold also for GExa.

To evaluate the throughput, we leverage some results from classical i.i.d. extreme value theory [10], exploiting the properties of the maximum among N i.i.d. random variables. Let γ be the Euler constant ($\gamma \approx 0.58$).

Lemma 1 (Bimodal case)**.** *Consider a set of N i.i.d. random variables $\{X_i\}_{i=1}^N$, in which*

$$X = \begin{cases} 0 & \text{with probability } p \\ U & \text{with probability } 1-p \end{cases}$$

where U is a random variable exponentially distributed with average $1/\lambda$

$$U \sim \lambda e^{-\lambda x} \qquad \text{for } x \geq 0$$

Then, for $N \to \infty$ the average value of the maximum:

$$E\left[\max_{i=1,\ldots,N} X_i\right] \to \frac{1}{\lambda}(\gamma + \log N + \log(1-p))$$

Proof. Let Y be the random variable corresponding to the maximum among N samples: $Y = \max_{i=1,\ldots,N}\{X_i\}$. Following standard methodology, the corresponding cumulative distribution function (CDF) of Y can be obtained as follows:

$$\begin{aligned} F_Y(y) = P(Y \leq y) &= P\left(\max_{i=1,\ldots,N}\{X_i\} \leq y\right) \\ &= \prod_{i=1}^{N} P(X_i \leq y) = \prod_{i=1}^{N} P(X \leq y) \\ &= \prod_{i=1}^{N} F_X(y) = F_X(y)^N \end{aligned}$$

To compute the CDF of X, we recall that for U

$$F_U(x) = 1 - e^{-\lambda x} \qquad \text{for } x \geq 0$$

After some simple computation $F_X(x) = 1 - (1-p)e^{-\lambda x}$, for $x \geq 0$ and

$$F_Y(x) = \left(1 - (1-p)e^{-\lambda x}\right)^N \tag{2.5}$$

Apply the following change of variable:

$$(1-p)e^{-\lambda x} = \frac{e^{-y}}{N}$$

then

$$x = \frac{1}{\lambda}(y + \log N + \log(1-p))$$

Exploiting (2.5):

$$F_Y\left(\frac{y+\log N+\log(1-p)}{\lambda}\right) = P\left(Y \leq \frac{y+\log N+\log(1-p)}{\lambda}\right) = \left(1-\frac{e^{-y}}{N}\right)^N$$

For $N \to \infty$:

$$\left(1-\frac{e^{-y}}{N}\right)^N \to e^{-e^{-y}}$$

and

$$P(\lambda Y - \log N - \log(1-p) \leq y) = e^{-e^{-y}}$$

for $-\infty < y < +\infty$. After defining

$$Z = \lambda Y - \log N - \log(1-p) \tag{2.6}$$

we obtain $P(Z \leq y) = e^{-e^{-y}}$, which corresponds to the Gumbel-type distribution [11] whose average is the Euler constant γ. Hence, by combining (2.6) with $E[Z] = \gamma$, for $N \to \infty$

$$E[Y] \to \frac{\gamma + \log N + \log(1-p)}{\lambda}$$

□

Lemma 2 (Exponential case)**.** *Consider a set of N i.i.d. random variables $\{X_i\}_{i=1}^N$. If all X_i are exponentially distributed with average $1/\lambda$, then*

$$E\left[\max_{i=1,\ldots,N} X_i\right] \to \frac{1}{\lambda}(\gamma + \log N) \qquad \textit{for } N \to \infty \tag{2.7}$$

Proof. The bimodal case for $p = 0$ corresponds to the exponential case. Just apply Lemma 1 to get the assert. □

Lemma 3 (Gaussian case)**.** *Consider a set of N i.i.d. random variables $\{X_i\}_{i=1}^N$. If all X_i have normal distribution with average a and variance b^2: $X \sim \mathcal{N}(a, b^2)$, then, for $N \to \infty$: $E\left[\max_{i=1,\ldots,N} X_i\right] \to a + b\Gamma(N)$, with function $\Gamma(N)$ defined as*

$$\Gamma(N) = (2\log N)^{\frac{1}{2}} - \frac{1}{2}(2\log N)^{-\frac{1}{2}}(\log(4\pi) + \log\log N)$$

We are now ready to present the theorem regarding the performance of Diag algorithm under different request matrices.

Theorem 1. *Let $R = [r_{ij}]$ be a Uni-AS request matrix, with $E[r_{ij}] = \mu \gg 0$. R is sustainable under Diag algorithm with a frame-expansion factor S that can be upper bounded by*

$$E[S] \leq \frac{\log N + \gamma}{\left(1 + \frac{\Gamma(N)}{\sqrt{N}}\right)}$$

Proof. To evaluate $E[S]$, we start to compute the average value of $F_R^{\mathcal{A}}$ and then we evaluate the average value of T_R. Let us focus on $F_R^{\mathcal{A}}$. Let C_d be the maximum element along the d-th diagonal of R. By construction, under Diag policy, $F_R^{\mathcal{A}} = \sum_{d=1}^{N} C_d$. We now wish to evaluate the average frame size $E[F_R^{\mathcal{A}}]$. Note that $C_1, C_2, \ldots, C_N$ are i.i.d. random variables. Then

$$E[F_R^{\mathcal{A}}] = NE[C_d] \tag{2.8}$$

$C_d = \max_{i=1\ldots N}\{A_i\}$, i.e. the maximum of N i.i.d. random variables A_i, distributed as each element of R. Since r_{ij} is geometrically distributed with average $\mu \gg 0$, we can approximate A_i with an exponential distribution with average μ. By Lemma 2 and (2.8),

$$E[F_R^{\mathcal{A}}] = \mu N(\log N + \gamma) \tag{2.9}$$

Let us now focus on T_R. Define T'_R and T''_R as the maximum row and column sums of R, i.e.

$$T'_R = \max_{j=1,\ldots,N} \sum_{i=1}^{N} r_{ij} \qquad T''_R = \max_{i=1,\ldots,N} \sum_{j=1}^{N} r_{ij}$$

From the Birkhoff-von Neumann theorem [23], $T_R = \max\{T'_R, T''_R\}$. Since all r_{ij} are i.i.d., we can focus on a generic row i of R and evaluate the sum B_i of the corresponding values: $B_i = \sum_{j=1}^{N} r_{ij}$. Thanks to the Central Limit Theorem, the distribution of B_i tends to the Normal distribution

$$B_i \sim \mathcal{N}(\mu N, \mu^2 N) \tag{2.10}$$

Rewriting T'_R as $T'_R = \max_{i=1,\ldots,N}\{B_i\}$, from Lemma 3

$$E[T'_R] \to \mu N + \mu\sqrt{N}\Gamma(N) \tag{2.11}$$

Since $T_R \geq T'_R$ (stochastically), the right side of (2.11) represents a lower bound on $E[T_R]$. Combining (2.9) and (2.11), the frame-expansion ratio S is upper bounded by:

$$E[S] \leq \frac{N(\log N + \gamma)}{N + \sqrt{N}\Gamma(N)}$$

□

Theorem 2. *Let $R = [r_{ij}]$ be a Uni-PS request matrix, being $E[r_{ij}] = \mu$. R is sustainable under Diag algorithm with a frame-expansion factor S whose average is:*

$$E[S] = 1 + \sqrt{\frac{1}{\mu}\left(1 - \frac{1}{N}\right)}\Gamma(N)$$

Proof. Note that, by construction, $T_R = \mu N$. We need to evaluate $E[F_R^{\mathcal{A}}]$ to compute $E[S]$. All the elements r_{ij} of the request matrix are identically distributed, even if not independent. Say A is the random variable corresponding to any r_{ij}.

Now A is obtained by the contribution of μN matchings, each of them including the element (i, j) with probability $1/N$. This is equivalent to state that:

$$A = \sum_{i=1}^{\mu N} H_i, \qquad \text{with } H_i = \begin{cases} 0 & \text{with probability } \dfrac{1}{N} \\ 1 & \text{with probability } 1 - \dfrac{1}{N} \end{cases}$$

Thanks to the Central Limit Theorem, A is normally distributed:

$$A \sim \mathcal{N}\left(\mu, \mu\left(1 - \frac{1}{N}\right)\right)$$

Define C as the maximum along a particular diagonal; C is the maximum of N i.i.d. random variables distributed as A.

By Lemma 3, $E[C] \to \mu + \sqrt{\mu(1 - 1/N)}\Gamma(N)$. Since $E[F_R^{\mathcal{A}}] = NE[C]$, we obtain:

$$E[S] = \frac{N\mu + N\sqrt{\mu(1 - 1/N)}\Gamma(N)}{N\mu}$$

which corresponds to the assert of the theorem. □

Theorem 3. *Let $R = [r_{ij}]$ be a Bim-PS request matrix, being $E[r_{ij}] = \mu(1 - p)$. R is sustainable under Diag algorithm with a frame-expansion factor S whose average can be upper bounded as:*

$$E[S] \leq \frac{\log(1 - p) + \log(N) + \gamma}{1 - p + \dfrac{\Gamma(N)}{\sqrt{N}}\sqrt{1 - p^2}} \tag{2.12}$$

Proof. It is possible to repeat exactly the same arguments as the proof of Theorem 1. To compute $E[F_R^{\mathcal{A}}]$, simply substitute $\log N$ with $\log N + \log(1 - p)$ (thanks to Lemma 1).

To compute $E[T'_R]$, it can be shown that B_i is normally distributed as $\mathcal{N}(\mu N(1 - p), N\mu^2(1 - p^2))$. The result immediately follows. □

Fig. 2.3 shows the average frame-expansion ratio obtained by simulating a large number of request matrices for different values N. We have investigated the different families of request matrices: Uni-AS, Bim-AS with $C_v = 2$ (large variance) and $C_v = 4$ (very large variance) and Uni-PS. The points (SIM) refer to the results obtained by simulation, and the curves (TEO) refer to the analytical curves of Theorems 1, 2 and 3. The graphs show that the bounds of Theorems 1 and 3 are quite tight, especially for large N, and the approximation of Theorem 2 is very accurate.

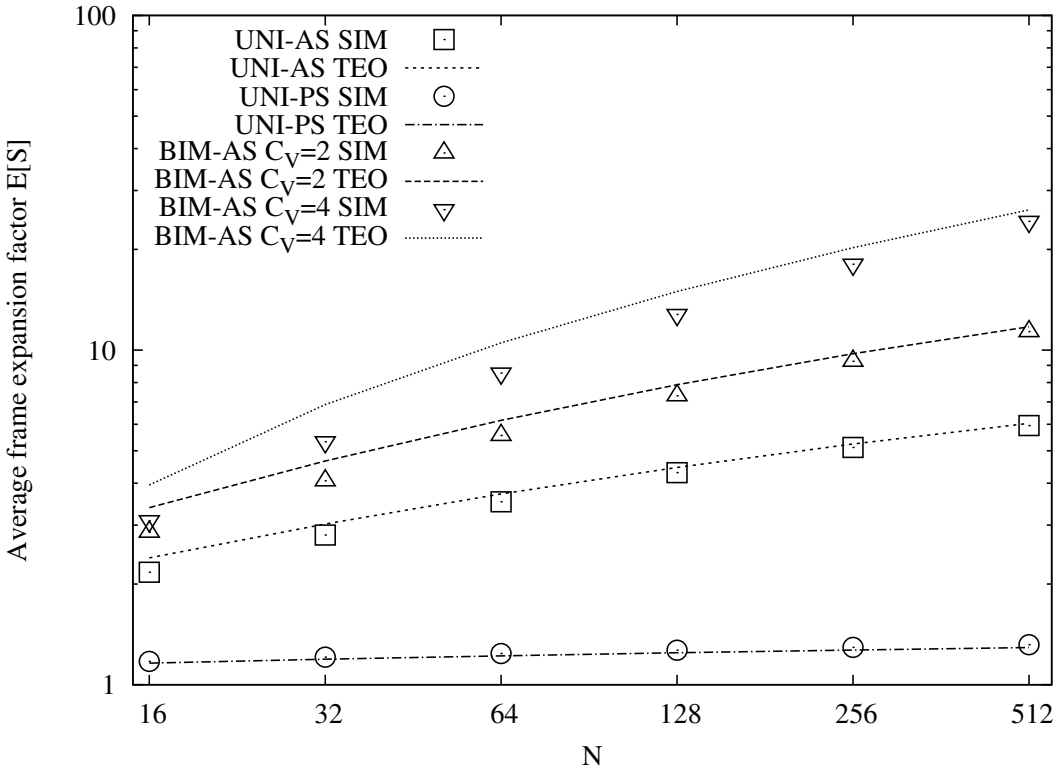

Figure 2.3. Analytical and simulated results for the average frame-expansion ratio and for different request matrices under the Diag algorithm.

2.3.3 Performance of the GExa algorithm

Let us evaluate the energy cost and the worst case throughput for the GExa algorithm. Since the service is exhaustive for each input-output pair, the energy cost is always equal to $2\hat{e}$ for each non-null r_{ij}, as for Diag. Hence, Table 2.2 is also valid for GExa and we can claim that GExa is optimal from the energy point of view.

Regarding the throughput, it can be easily shown that:

Theorem 4. *Let $R = [r_{ij}]$ be any request matrix. R is sustainable under the GExa algorithm with a frame-expansion factor $S \leq 2$.*

Proof. Observe that GExa decomposes R using a sequence of maximal matchings. From Theorem 2.2 in [7] or Theorem 4.2 in [8], if a matrix is decomposed by *any* sequence of maximal matchings, then the number of matchings needed is at most twice than the number obtained by BvN. Hence, $F_R^A \leq 2T_R$ and $S \leq 2$. □

As a consequence, the maximum sustainable load by GExa is always ≥ 0.5, independently of the switch size.

Table 2.3. Energy per packet for Uni-AS request matrices, with $N = 64$

Decomposition alg.	Worst Sort (WS)	No Sort (NS)	Best Sort (BS)
BvN	0.872	0.810	0.741
GMax	0.425	0.415	0.377
GMin	0.723	0.045	0.049
GExa	0.866	0.020	0.020
Diag	0.020	0.020	0.020

2.3.4 Effect of Frame Sorting

We first evaluate the effect of the algorithms sorting the frame. Table 2.3 reports the energy per packet obtained by combining a specific matching selection algorithm with a particular sorting algorithm, for Uni-AS request matrices with $\mu = 100$, in a $N \times N$ IQ switch, with $N = 64$. Very similar results were obtained for different switch size and different random request matrices. The Diag algorithm is not affected by the sorting and the energy per packet is coherent with the analytical values reported in Table 2.2.

Indeed, all the matchings are distinct and $2N^2\hat{e}$ is the total energy spent in a frame. Since the total number of packets is, on average $N^2\mu$, the average energy per packet is simply $2N^2\hat{e}/(N^2\mu) = 2\hat{e}/\mu$, independently from R, as shown in the table.

Recall that this value is the minimum energy achievable by any algorithm under the Uni-AS scenario, but it requires a large frame-expansion factor S, as shown later.

As a general comment, the beneficial effect of the frame-sorting algorithm on the energy minimization depends from the specific matching-selection algorithm. In general, we expect that best-sorting (BS) will outperform no-sorting (NS) which, in turn, will outperform worst-sorting (WS). This is not always true, as shown below.

For the BvN matching-selection algorithm, BS allows to reduce by 10% the energy cost with respect to NS, and 17% with respect to WS. In all cases, the absolute costs are the highest, and this is due to the specific algorithm adopted in BvN, based on computing a maximum size matching at each iteration, without considering the energy cost to change the matching.

When combined with GMax, BS reduces the energy cost similarly for the BvN case. In absolute terms, the cost are smaller than BvN, since the greedy algorithm based on the queue length induces some correlation between the matchings computed in subsequent iterations of the algorithm. This effect is highlighted in GMin.

Interestingly, in GMin, the effect of the frame-sorting is always negative. Indeed, the energy cost obtained by NS is natively very small, and BS increases the energy

cost. This is not surprising, since BS is an approximated algorithm to solve TSP and its solution is worse than the initial sequence offered by unsorted $\mathcal{U}_R$. Although not completely intuitive at a first glance, this effect is due to the particular metric used to compute the matching at each iteration. By subtracting the minimum weight matching M^k from R^k at iteration k, there is a high probability that the new minimum weight matching M^{k+1} shares some (at most, $N-1$) edge with M^k. This correlation induces an efficient "self-sorting" property, providing an energy efficiency comparable with, and in some situations even better, than the one achieved by BS sorting. On the contrary, when running GMax algorithm, M^k is a (almost) maximum weight matching; as such, there is a very low probability that M^{k+1} shares edges with M^k. This explain the lower energy cost of GMin with respect to GMax. As a conclusion, GMin is efficient in terms of energy-cost without any additional sorting algorithm.

Similarly to Diag, GExa-NS are GExa-BS are both optimal in terms of energy, since the matching order induced by GExa is already optimal. On the contrary, WS changes the order and (differently from Diag), the energy cost increases.

Since the above reported results hold qualitatively in many scenarios, we focus only on the following optimized combinations of frame scheduling algorithms in the next sections: BvN-BS, GMax-BS, GMin-NS, Diag-NS and GExa-NS. These algorithms have very different computational complexities and memory requirements; the sorting procedure itself requires to store the whole frame sequence to sort it. The ranking among the algorithms in terms of increasing complexity is: Diag-NS (less complex), GExa-NS, GMin-NS, GMax-BS and BvN-BS (more complex).

2.3.5 Energy and Throughput Tradeoff

Simulations have been run in a proprietary simulation environment written in C language. The parameter μ, related to the packet arrival rate, is set equal to 100; all simulations results are obtained as an average of 100 simulation runs, each run using a different randomly generated request matrix, to obtain statistically significant simulation results.

We mainly report the results for $N = 16$ (denoted with white shapes in the graphs) and $N = 128$ (denoted with black shapes in the graphs); however, similar results hold also for $N = 32$ and $N = 64$ scenarios.

In all the reported plots, each point corresponds to the average value; two bars around each point (one horizontal bar and one vertical bar) show the maximum and minimum values obtained considering all 100 runs. When the error-bars are not visible, the results of each run are almost identical to the average value, i.e., a small variance exists when changing the seed to generate the random matrices.

Fig. 2.4 shows the tradeoff between the maximum sustainable load and the energy per packet obtained by the different algorithms. To read the plot, suppose that the

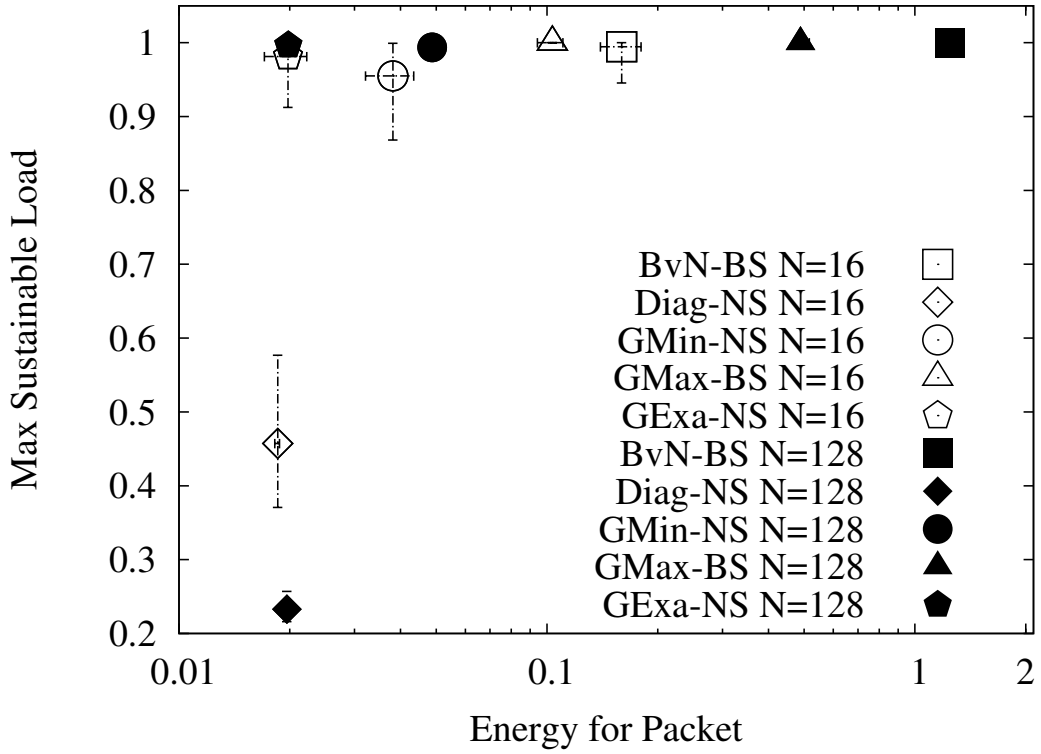

Figure 2.4. Throughput and energy tradeoff under Uni-AS traffic for $N = 16$ (white shapes) and $N = 128$ (black shapes).

switch designer is willing to obtain a minimum sustainable load and a maximum energy consumption per packet. These design constraints define a point (e', ρ') in the graph. All the algorithms whose performance are in the region to the left and above this point (i.e., with energy $\leq e'$ and maximum sustainable load $\geq \rho'$) satisfy the design constraint.

Only for a small switch size $N = 16$, the algorithms Diag-NS and GMin-NS show some variations in the maximum sustainable load; in all other cases, simulations results show a very small variability when changing the traffic matrix. As expected, the Diag-NS and GExa algorithms achieve the minimum energy per packet, whose value can be computed with the formulas in Table 2.2. However, due to the large frame-expansion factor required, Diag-NS cannot sustain large loads, as stated in Theorem 1.

The GMin-NS algorithm, despite its relative simplicity, achieves energy consumption levels only 2-3 times larger than Diag-NS, but with a throughput very close to the maximum throughput. GExa-NS achieves the best overall performance, with energy consumption as low as Diag-NS and almost maximum throughput. BvN-BS

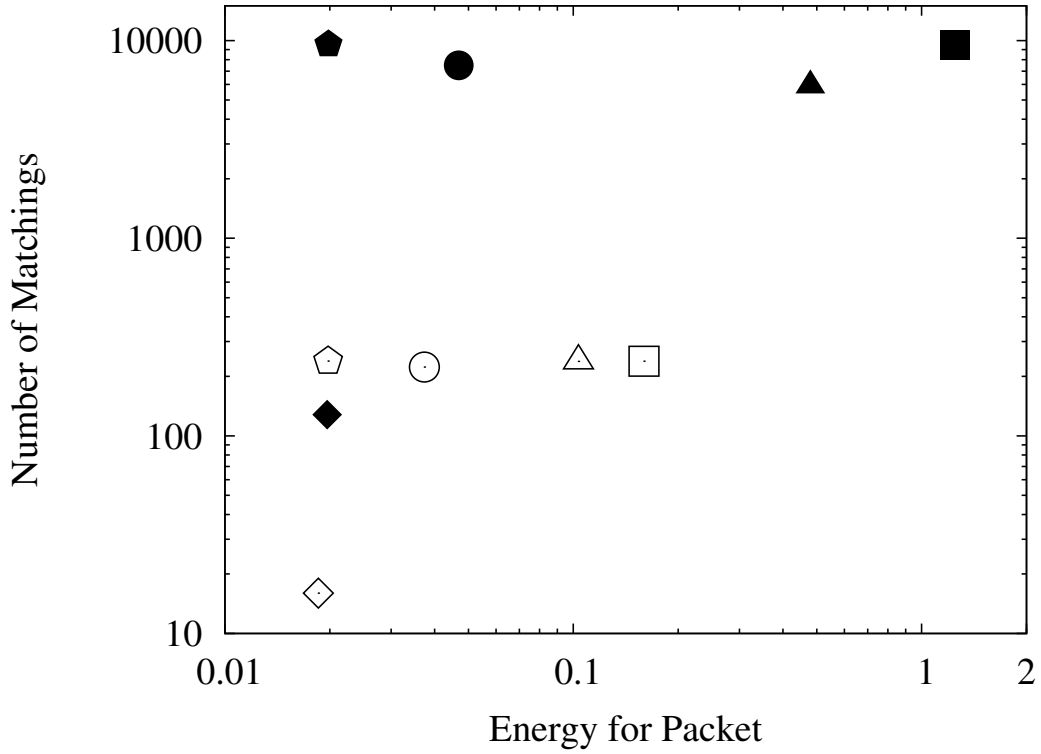

Figure 2.5. Tradeoff between the average number of matchings and energy consumption under Uni-AS traffic for $N = 16$ (white shapes) and $N = 128$ (black shapes).

and GMax-BS provide almost the same throughput, but at the expenses of large energy consumption, even after the frame sorting. Note that GMax-BS is more energy-efficient than BvN-BS, due to the metrics used to compute the matching, as already observed in Sec. 2.3.4. Finally, energy consumption per packet increases a lot for larger switch size, as expected.

Fig. 2.5 shows the number of distinct matchings computed by each algorithm, for the Uni-AS scenario. The algorithm Diag-NS, by construction, uses only N matchings. All other algorithms use a significantly larger number of matchings. For $N = 128$, GExa-NS uses the largest number of matchings, even if the energy consumption is minimum. Note that roughly $N^2\mu$ packets should be scheduled in a frame, and, to achieve the maximum frame load ($\rho = 1$), each single matching should roughly serve N packets. Hence, there are at most $N\mu = 12{,}800$ different matchings in the frame under Uni-AS. Even if the algorithm GExa-NS uses almost all the different matchings, the total energy cost is small because the energy cost between any pair of matchings is very small.

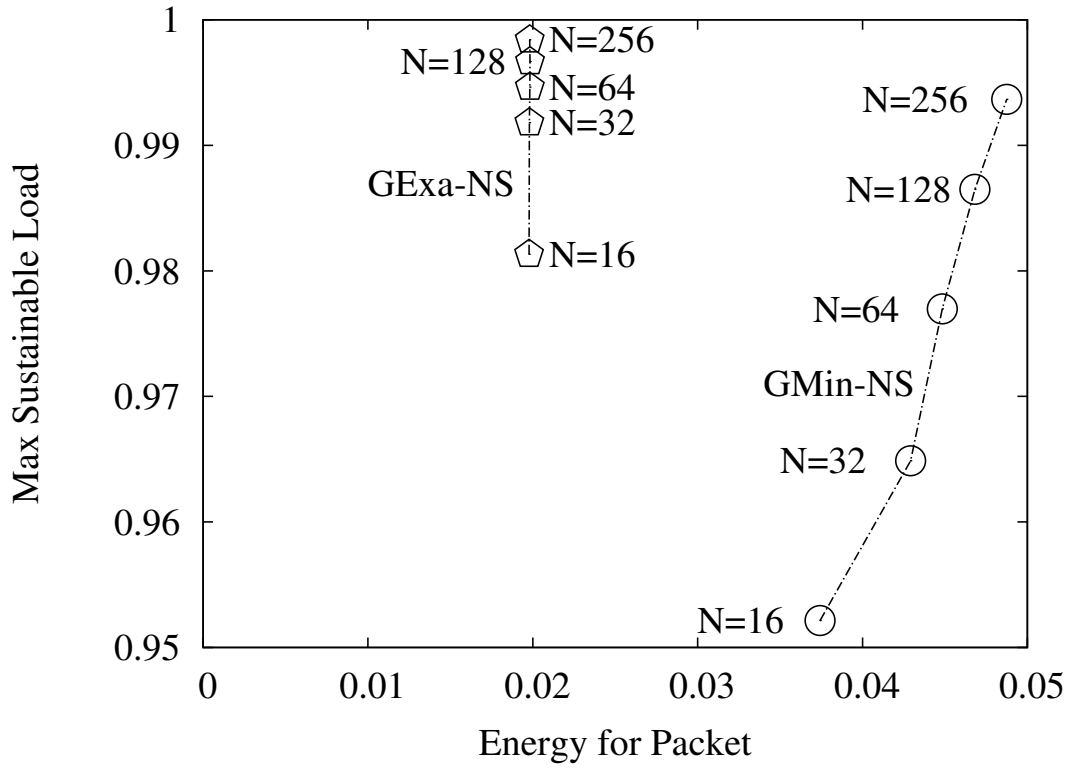

Figure 2.6. Throughput and energy tradeoff for GMin-NS and GExa-NS under Uni-AS traffic.

In Fig. 2.6 we focus on the energy-throughput tradeoff obtained by GExa-NS and GMin-NS by varying N. Regardless of the switch size, the maximum sustainable load is always significant for both algorithms, and the increase in the energy per packet as a function of N is marginal for GMin-NS and null for GExa-NS. Similar observations hold for the Bim-AS scenario, as reported in Fig. 2.7.

In the case of Uni-PS scenario, Fig. 2.8 shows that BvN-BS achieves the worst energy performance, even if the maximum sustainable load is always achieved. The best energy results are obtained by GExa-NS and Diag-NS, the latter providing higher throughput differently from the Uni-AS scenario. This is mainly due to the smaller variance of the values in the diagonal elements of the request matrix R: the maximum element on a diagonal is close to the average and Diag-NS shows better performance. BvN-BS appears to be very inefficient in terms of energy consumption, when compared with any other algorithm, especially when the switch size grows. GExa-NS and GMin-NS show the best tradeoff, since they achieve the minimum energy, close to Diag-NS, and almost the maximum throughput.

As a last scenario, we consider the Bid-PS scenario in Fig. 2.9, where GExa-NS

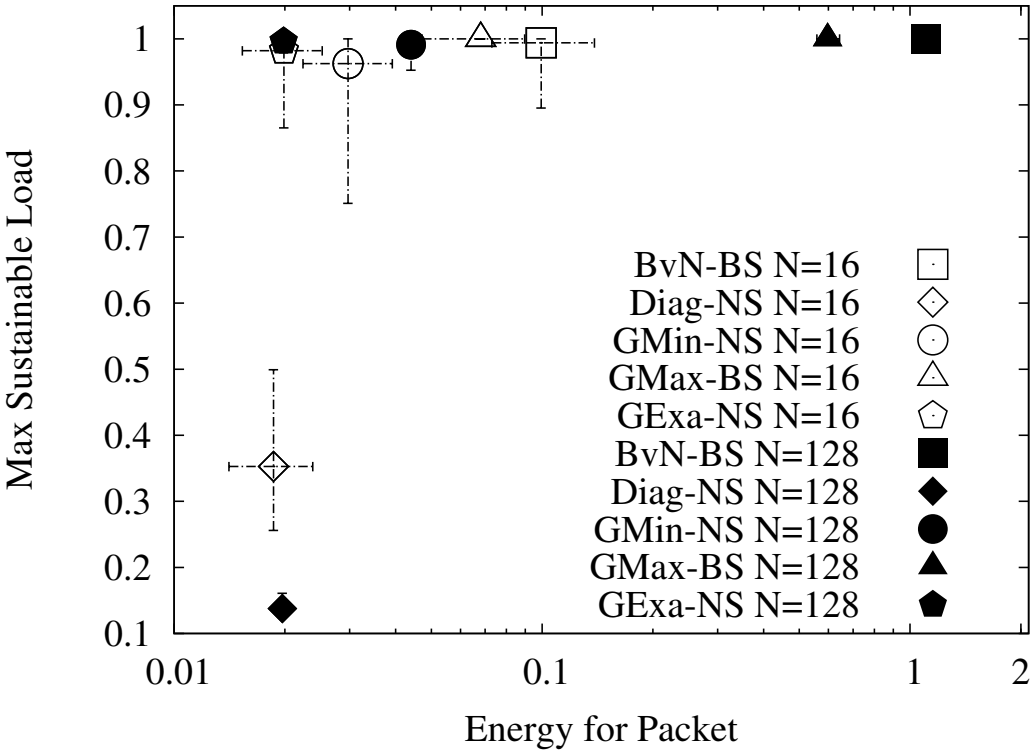

Figure 2.7. Throughput and energy tradeoff under Bim-AS traffic for $N = 16$ (white shapes) and $N = 128$ (black shapes).

and Diag-NS achieve the energies computed in Table 2.2. The two algorithms are not optimal anymore, since GMax-BS and GMin-NS find a frame with only two matchings (corresponding to D^1 and D^2) and with a total energy per frame equal to $(N-1)\hat{e}$, i.e. roughly $1/\mu$, half of the values achieved by GExa-NS and Diag-NS. This is the only traffic scenario in which we were able to show some energy impairment (bounded by a factor 2) for Diag-NS and GExa-NS algorithms.

Fig. 2.10 shows the results for Bid-AS scenario. All the considered algorithms achieve the maximum throughput (for large switch size) except for Diag-NS. Regarding the energy, Diag-NS and GExa-NS achieves the minimum values, as estimated in Table 2.2. GMin-NS behaves very similarly to GExa-NS.

2.4 Delay control through frame scheduling

In the previous performance evaluation, we have deliberately omitted to evaluate the delays, since they can be controlled by the frame scheduling in a complementary way to ours. Indeed, a generic frame scheduling approach can guarantee delay bounds;

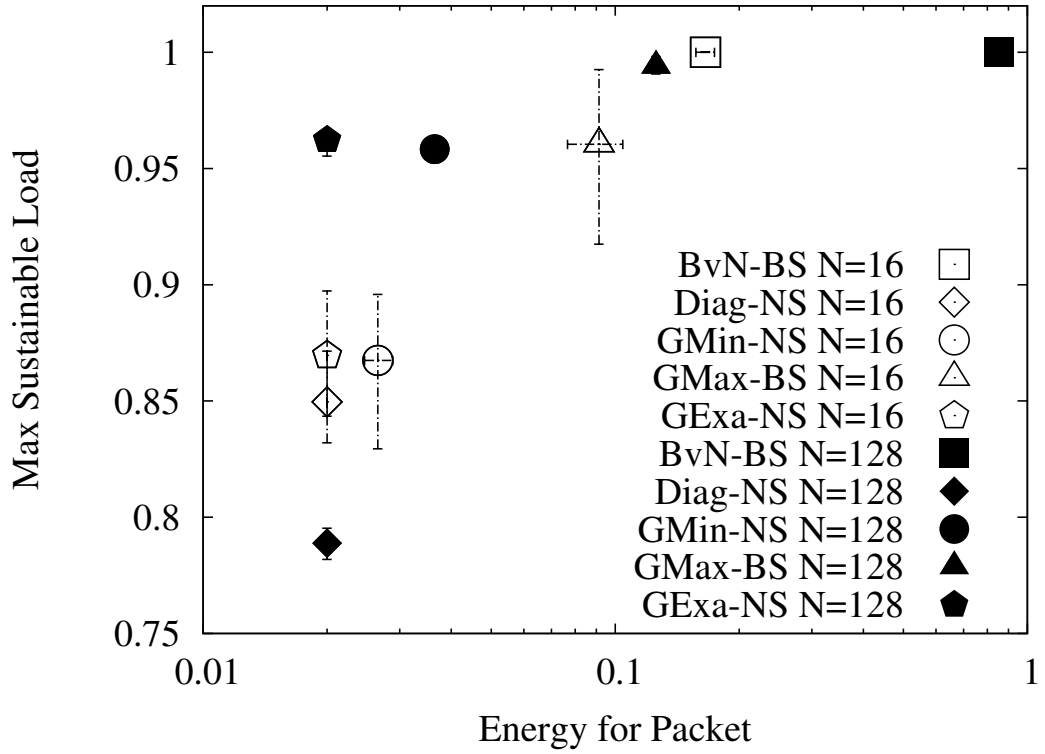

Figure 2.8. Throughput and energy tradeoff under Uni-PS scenario for $N = 16$ (white shapes) and $N = 128$ (black shapes).

we consider specifically the "Fair-Frame" approach proposed in [12], ensuring delays that grows logarithmically with N.

Let R_{new} be the request matrix corresponding to only the packets arrived during the current sampling period, i.e. during the interval $(n-1)T < t \leq nT$, for $n \in \mathbb{N}$; let R_{res} be the residual matrix with the packets arrived in earlier periods, during any timeslot $t \leq (n-1)T$, and not yet served at timeslot $t = nT$. By construction, the total request matrix at $t = nT$ is $R = R_{new} + R_{res}$.

Assume now that the input traffic is Poisson with rates λ_{ij} satisfying:

$$\sum_k \lambda_{ik} \leq \theta \qquad \sum_k \lambda_{kj} \leq \theta, \qquad \forall i, j$$

where θ is the normalized load. Sec. IV of [12] shows that, if the traffic is admissible ($\theta < 1$), it is possible to define a minimum sampling period T, function of θ, such that R is admissible with arbitrary high probability and $T_{R_{new}} \leq T$. Analogously, if the decomposition algorithm shows a frame-expansion ratio S, it is again possible to find a minimum T, function of θ/S, such that R is both admissible and sustainable

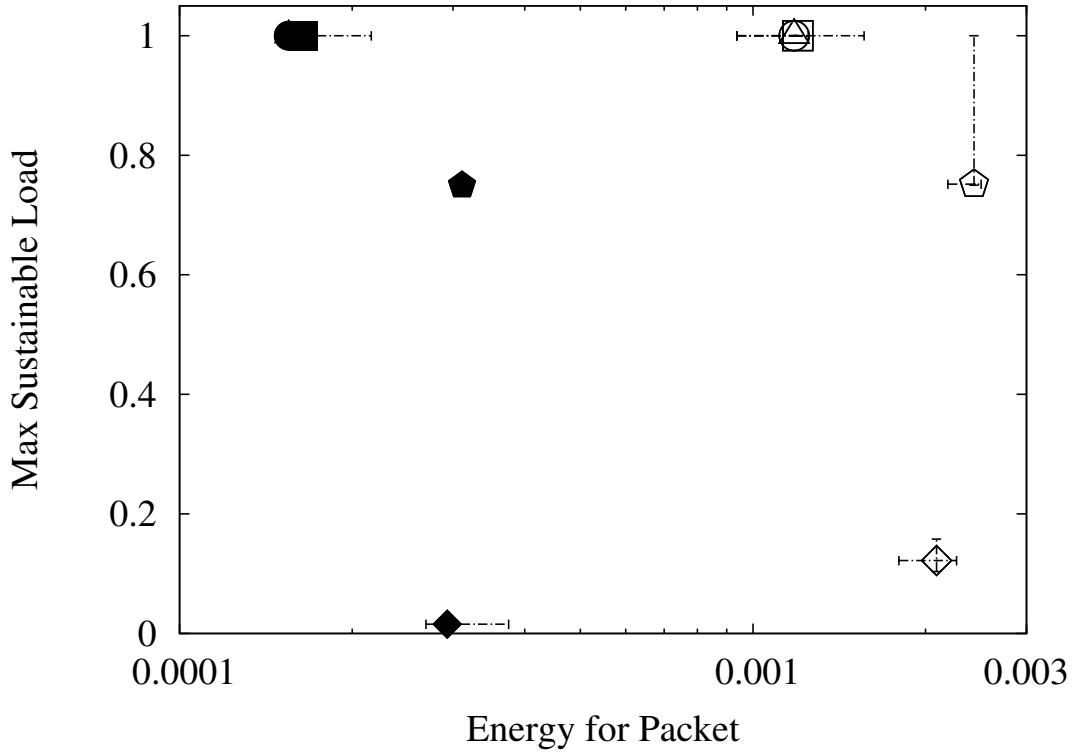

Figure 2.9. Throughput and energy tradeoff under Bid-PS scenario with $\alpha = 2/3$ for $N = 16$ (white shapes) and $N = 128$ (black shapes).

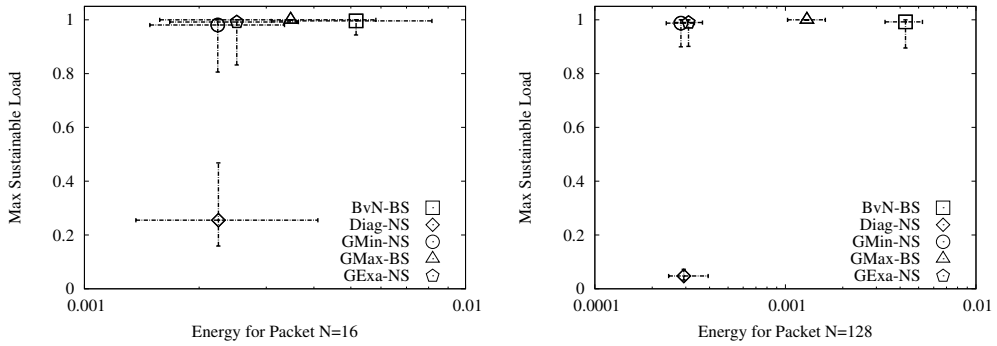

Figure 2.10. Throughput and energy tradeoff under Bid-AS scenario for $N = 16$ (left) and $N = 128$ (right).

with high probability. For simplicity, assume that $S = 1$. In the case $T_{R_{new}} > T$ (this occurs with low probability), a new request matrix R'_{new} is built by reducing some elements in R_{new} to obtain $T_{R'_{new}} = T$ and make R'_{new} admissible; the remaining packets, given by $R_{new} - R'_{new}$, will be served in future periods.

The main idea of "Fair-Frame" is to define the current frame $\mathcal{F}_R^A$ based on R'_{new}, without considering R_{old}. But, in the case a matching in $\mathcal{F}_R^A$ is not complete, the scheduler exploits the unused input/output ports to serve the packets in R_{res}.

This frame scheduling approach, applicable also for our energy-aware frame scheduling, allows to achieve an explicit delay bound for a given load θ using (26) of [12], i.e. to obtain delays $O(\log(N))$. Such delays are asymptotically much smaller than any frame scheduling policy oblivious of the VOQ lengths, for which delays are $O(N)$.

2.5 Related Work

In optical switches based on MEMS, tunable lasers and other technologies, a reconfiguration latency must be paid when the switching fabric changes configuration; a "blackout" period is experienced in packet transmissions, during which the whole switching fabric is not available to transfer packets.

In [13] the optimal frame scheduling to compute the fabric configurations was studied. The cost function minimized in [13] is similar to the one considered here. Indeed, the reconfiguration is a cost paid anytime the switching configuration changes. However, differently from our case, the cost in [13] is independent of the number of input-output connections that change inside the switching fabric: a single connection modification implies that the whole switching fabric becomes unavailable, thus introducing the cost of a complete reconfiguration. Hence, differently from ours, the scheduling policy is designed to minimize the number of matchings to serve all the packets in the request matrix. [13] showed that the optimal scheduling problem with reconfiguration latency belongs to the NP-complete class, and proposed two sub-optimal algorithms Min and Double. Min algorithm decomposes the request matrix into N matchings as our Diag, but the corresponding frame-expansion factor S grows as $S \approx 4 \log_2 N$. As an alternative, Double algorithm decomposes the request matrix with $2N$ matchings while keeping $S = 2$.

An input queued switch in which the power consumption depends on the speed at which packets are sent through the switching fabric is considered in [14]. A power quadratic cost with the transmission speed is assumed, and an on-line scheduling algorithm derived from the theory of dynamic programming is defined. At any time, the algorithm selects the packets to transfer and the switching fabric speedup, trying to achieve the best tradeoff between power consumption and delays. Similarly, [15] investigates on-line scheduling algorithms to stop temporary the packets transmissions across the switching fabric to reduce the power consumption and to keep a target delay/backlog performance. Finally, [16] combines the on-line approaches of [14, 15] in a more generic queueing system. However, the addressed problem is not the minimization of the energy due to reconfiguration costs.

Chapter 3

Power Control for Crossbar-based Input-Queued Switches

3.1 Problem Definition

We start by considering a single CMOS component, the basis of the combinatorial logic of a single crosspoint in the crossbar chip.

3.1.1 Energy model for a single CMOS gate

The energy consumption of a CMOS gate is strongly dependent on the supply voltage V and it can be modeled as the sum of a dynamic energy component (due to electrical signal switching activity needed to transfer sequence of 0s and 1s) and a static energy component (due to leakage currents). We consider only the dynamic energy component, while we neglect the latter contribution. Leakage currents tend to be proportional to occupied area and are normally controlled by means of circuit level techniques that are out of the scope of this work. The energy due to a bit transition (i.e., the switching activity) is a quadratic function of V according to the well known formula $E_{bit} = 0.5CV^2$, where C is the load capacitance. If we consider a 0-1 square wave signal with frequency f, the power consumption is

$$P = E_{bit} f \propto fV^2 \tag{3.1}$$

that represents also the thermal power to dissipate.

The allowed frequency is $f \propto V$ due to the delay needed to switch from one logic state to another [20]. Thereby, the power consumption for a CMOS operating at maximum frequency and voltage is proportional to f^3. DVFS techniques jointly reduce V and f to minimize power consumption, exploiting time periods in which the signal can be *"slowed down"* to a lower peak frequency.

This approach is actually implemented in commercial CPUs, where the processing speed changes with the instantaneous processing load [21].

We consider a CMOS device operating at voltage V, ranging between $V_{\min}$ and $V_{\max}$. Within this range, we assume that bit transmissions can occur at intermediate voltage levels. When operating at $V < V_{\max}$, since $f \propto V$, the signal frequency can be slowed down by a factor

$$\alpha = \frac{V_{\max}}{V}$$

with respect to the maximum frequency allowed when using $V_{\max}$. Thus, α is the *expansion factor* of the bit duration with respect to the bit duration when using $V_{\max}$.

Furthermore, V must be larger than $V_{\min} > 0$, because of technological constraints that forbid to reduce the voltage level too much and of the impact of leakage currents, that otherwise would become not negligible. Define $\beta = V_{\min}/V_{\max}$. By construction,

$$1 \leq \alpha \leq \frac{1}{\beta}$$

Depending on the technology, $\beta = 0.5$ for a classical DVFS scheme or $\beta = 0.3$ in the case of an "extreme" DVFS scheme, according to [17].

3.1.2 Switching architecture

We consider an $N \times N$ input queued (IQ) switch, with virtual output queueing (VOQ) as shown in Fig. 2.1.

In this chapter the switching fabric is an $N \times N$ crossbar chip, with N^2 crosspoints and $\Theta(N^2)$ CMOS components. The crosspoint connecting input i to output j is denoted as XP_{ij} and is fed by VOQ_{ij} traffic.

The scheduling decisions occur at a packet level, with a time granularity equal to the minimum packet duration. In the case of minimum Ethernet packet size and 10 Gbit/s line rates, a new scheduling decision must be taken every 50 ns. Given such a strict timing constraint, packet schedulers are often implemented directly in hardware, but off-chip, i.e., on a separate chip with respect to the crossbar chip.

We do not focus on any particular scheduler, although for simplicity the model assumptions hint at packet schedulers able to achieve 100% throughput under admissible traffic.

3.2 Crossbar power control

A vast literature is available on the design of low complexity and high performance packet schedulers for input queued switches [22].

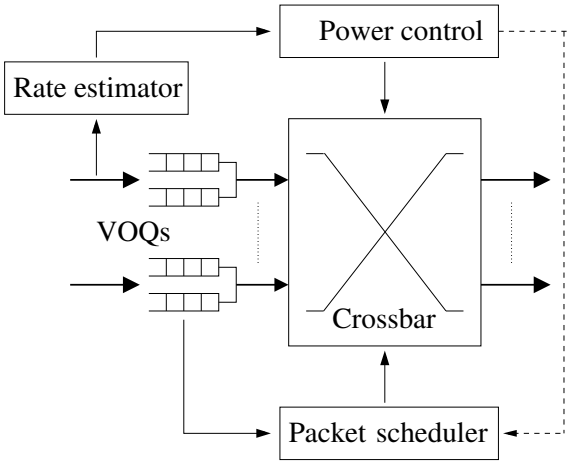

Figure 3.1. Power control scheme in an IQ switch

The aim of the power control block in Fig. 3.1 is to exploit DVFS at crosspoints to reduce the crossbar chip power consumption. Based on traffic measurements on the ms scale which provide rate estimations, the control determines the DVFS factor α_{ij} for the combinatorial logic at XP_{ij}, assuming that each crosspoint is controlled independently. Due to the relaxed timing constraints, the algorithm for power control is assumed to be implemented as a software component running on an off-chip processor. Since we focus on crossbar power consumption, we disregard the power contribution of the scheduler and of the power control block. However, the only additional power consumption introduced by our proposed DVFS is due to the power control block; this contribution is negligible with respect to the scheduler consumption due to comparable algorithmic complexity and much larger time scale.

Let $\alpha = [\alpha_{ij}]$ be the $N \times N$ matrix with the DVFS factors currently employed in the crossbar. Note that setting $\alpha_{ij} > 1$ implies that the forwarding rate at XP_{ij} is reduced and the packet transmission time is increased by the expansion factor α_{ij}. This has two main consequences:

1. an additional queueing delay in VOQ_{ij}

2. the packet scheduler cannot serve any new packet from input i and to output j until XP_{ij} ends the packet transmission

Thus, the packet scheduler should be slightly modified to take into account DVFS factors in packet scheduling. We disregard this issue in the analysis, and we take an ideal fluid-based approach, looking only at I/O flow rates, to evaluate the possible asymptotic benefits in terms of reduced power consumption. Note that extending packet duration might influence switch throughput and buffer size requirements.

However, the power control algorithms avoid switch overloading, by increasing packet duration only at low-medium input load. This translate in an internal load increase. In other words, the switch operates internally always in a high load regime, regardless of the real input load, but never in overload. As such, buffer requirements are not modified, because buffer size are designed for high load conditions, which are not modified by the power control scheme.

3.2.1 Input traffic

To avoid dealing with data content, we assume that a data packet of length L is transmitted using L signal transitions: i.e., each packet is composed by a sequence of alternating 0 and 1.

Denote the maximum line rate as $r_{\max}$, measured in [bit/s]: $r_{\max}$ is achievable only for $V = V_{\max}$. The traffic load on each link is measured on a time window whose duration T_w is in the order of ms. Let r_{ij} be the average arrival rate [bit/s] for the traffic flows enqueued at VOQ_{ij} during the current time window, and $R = [r_{ij}]$ the corresponding $N \times N$ traffic matrix. Let $S = [s_{ij}]$ be the normalized traffic matrix obtained by setting $s_{ij} = r_{ij}/r_{\max}$, with $s_{ij} \in [0,1]$. We assume that $s_{ij} > 0$ for any i and j.

Definition 1. *The* average load *of matrix S is defined as*

$$\rho_{ave}(S) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ij}$$

Definition 2. *The* average load *at input i and at output j is*

$$\rho_i^I(S) = \sum_{k=1}^{N} s_{ik} \qquad \textit{and} \qquad \rho_j^O(S) = \sum_{k=1}^{N} s_{kj}$$

respectively.

Definition 3. *The* maximum load *of matrix S is*

$$\rho_{\max}(S) = \max\{\max_k\{\rho_k^I(S)\}, \max_k\{\rho_k^O(S)\}\}.$$

Definition 4. *The traffic matrix S is said to be* admissible *if and only if*

$$\rho_{\max}(S) \leq 1$$

Obviously, $\rho_{\text{ave}}(S) \leq \rho_{\max}(S)$.

3.2.2 The minimum power control problem

To keep bounded queues and delays, and to avoid overload, we model the constraints related to the maximum time expansion allowed for the transmitted bits. During a measurement period, the total number of arrived bit is $T_w r_{ij}$, smaller than the maximum number of bits $T_w r_{\max}$ that can be transmitted at $V_{\max}$. Hence, the maximum allowed expansion factor for each bit is $r_{\max}/r_{ij}$, i.e. $\alpha_{ij} r_{ij} \leq r_{\max}$. At the same time, to avoid overload, it is necessary to limit the expansion at each input and output:

$$\sum_{k=1}^{N} \alpha_{ik} r_{ik} \leq r_{\max} \qquad \sum_{k=1}^{N} \alpha_{kj} r_{kj} \leq r_{\max} \qquad \forall i, j$$

which can be normalized as

$$\sum_{k=1}^{N} \alpha_{ik} s_{ik} \leq 1 \qquad \sum_{k=1}^{N} \alpha_{kj} s_{kj} \leq 1 \qquad \forall i, j \tag{3.2}$$

Similarly to (3.1), the power consumption of XP_{ij}, denoted as P_{ij}, is proportional to

$$P_{ij} \propto r_{ij} \left(\frac{V_{\max}}{\alpha_{ij}}\right)^2 = s_{ij} r_{\max} \left(\frac{V_{\max}}{\alpha_{ij}}\right)^2 \propto \frac{s_{ij}}{\alpha_{ij}^2}$$

The total crossbar power consumption is the sum of the power contributions of all crosspoints:

$$P_{tot} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \propto f_P(\alpha) = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{s_{ij}}{\alpha_{ij}^2} \tag{3.3}$$

where $f_P(\alpha)$ is a power cost factor.

Finally, the minimum power problem (denoted as Opt-MP) becomes: given an admissible S, find a feasible α minimizing f_P [1]:

$$\min_{\alpha} f_P(\alpha) = \min_{\{\alpha_{ij} \in \mathbb{R}^+\}_{i,j}} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{s_{ij}}{\alpha_{ij}^2} \tag{3.4}$$

$$\text{subject to} \begin{cases} \displaystyle\sum_{k=1}^{N} \alpha_{ik} s_{ik} \leq 1 & \forall i \quad (3.5) \\ \displaystyle\sum_{k=1}^{N} \alpha_{kj} s_{kj} \leq 1 & \forall j \quad (3.6) \\ \alpha_{ij} \in \mathcal{A} & \forall i, j \quad (3.7) \end{cases}$$

[1] More properly, we should say that f_P is a power cost factor.

where $\mathcal{A}$ is the set of all available voltage levels.

Property 1. OPT-MP *is an integer convex non-linear optimization problem.*

Continuous version of the problem

Following a standard methodology, we start to relax OPT-MP to continuous variables. This leads to the following problem, denoted as CONT-MP: minimize $f_P(\alpha)$ subject to (3.5) and (3.6); (3.7) is substituted by

$$\alpha_{ij} \geq 1 \qquad \forall i, j$$

corresponding to a DVFS scheme in which any voltage between 0 and $V_{\max}$ is allowed[2].

Let $\hat{\alpha}_{\text{OPT-MP}}$ be the optimal solution of OPT-MP. Let $\hat{\alpha}_{\text{CONT-MP}}$ be the optimal solution of CONT-MP. Since CONT-MP is a relaxed version of OPT-MP, $\hat{\alpha}_{\text{CONT-MP}}$ is a lower bound on the power cost

Property 2. $f_P(\hat{\alpha}_{\text{CONT-MP}}) \leq f_P(\hat{\alpha}_{\text{OPT-MP}})$.

Theorem 5. CONT-MP *is equivalent to*

$$\min_{\alpha} f_P(\alpha) \tag{3.8}$$

$$\textit{subject to} \quad \begin{cases} \displaystyle\sum_{k=1}^{N} \alpha_{ik} s_{ik} = 1 \qquad \forall i & (3.9) \\ \displaystyle\sum_{k=1}^{N} \alpha_{kj} s_{kj} = 1 \qquad \forall j & (3.10) \\ \alpha_{ij} \geq 1 \qquad \forall i, j & (3.11) \end{cases}$$

Proof. Assume $\hat{\alpha} = [\hat{\alpha}_{ij}]$ to be the optimal solution. Define $\hat{s}_{ij} = \hat{\alpha}_{ij} s_{ij}$. By contradiction, assume that there exists i such that $\sum_k \hat{s}_{ik} < 1$, i.e. the i-th row of $\hat{S} = [\hat{s}_{ij}]$ sums to less than one (the same argument holds for the case the column sums to less than one).

Now two cases can occur. In the first case, it exists also one column j that sums to less than one, i.e. $\sum_k \hat{s}_{kj} < 1$. Hence, it is possible to increase $\hat{s}_{ij}$ to $\hat{s}'_{ij}$ while satisfying constraints (3.5)-(3.6). The new corresponding $\alpha'_{ij} = \hat{s}'_{ij}/s_{ij}$ is feasible and provides a lower cost function; this contradicts our assumption. In the second case, all the columns sum to one and, summing over all the columns, we have $\sum_j \sum_k s_{kj} = N$, which contradicts the assumption $\sum_i \sum_k s_{ik} < N$. □

[2]The constraint on $V_{\min}$ will be discussed at the end of the section.

Note that one of the constraints in (3.9)-(3.10) is linearly dependent of the others and can be omitted.

Definition 5. *Given a non-negative matrix $H \in \mathbb{R}^{N\times N}$, H is said to be ρ-double-stochastic if $\rho_i^I(H) = \rho_j^O(H) = \rho$ for any i and j, i.e. $\rho_{ave}(H) = \rho_{\max}(H) = \rho$. A 1-double-stochastic matrix is usually called double-stochastic matrix.*

Definition 6. *Given a non-negative matrix $H \in \mathbb{R}^{N\times N}$, H is said to be ρ-sub-stochastic if $\rho_{\max}(H) = \rho$. In this case, $\rho_i^I \leq \rho$ for any i and $\rho_j^O \leq \rho$ for any j; furthermore, $\rho_{ave}(H) \leq \rho_{\max}(H) = \rho$ must hold.*

Thanks to Theorem 5, Cont-MP translates to: given a ρ-sub-stochastic matrix S, find a double-stochastic matrix $\hat{S} = [\hat{s}_{ij}]$ such that the set of $\alpha_{ij} = \hat{s}_{ij}/s_{ij}$ minimizes $f_P(\alpha)$. In other words, S is augmented to become double-stochastic.

The following Theorem provides an easily computable optimal solution:

Theorem 6. *Given a ρ-double-stochastic matrix S, the optimal solution $\hat{\alpha}$ for* Cont-MP *is*

$$\hat{\alpha}_{ij} = \frac{1}{\rho} \qquad \forall i,j$$

The corresponding power cost factor is

$$f_P(\hat{\alpha}_{\text{Cont-MP}}) = N\rho^3$$

Proof. The proof is based on the use of the Lagrange multipliers and on the Taylor's Theorem for multivariate functions. Denote $\otimes$ as the Hadamard product (i.e., element-by-element) of two matrices. Define $\hat{\alpha}$ as the optimal solution given by $\hat{\alpha}_{ij} = 1/\rho$ and define α, with $\alpha \neq \hat{\alpha}$, a generic feasible solution satisfying (3.9) and (3.10); $\alpha \otimes S$ and $\hat{\alpha} \otimes S$ are both double stochastic matrices. We can define matrix $\Delta = \alpha - \hat{\alpha}$ and assume that $\max_{i,j}\{\Delta_{ij}\} \leq \epsilon$ where $\epsilon > 0$. We can use Birkhoff-von Neumann Theorem [23] to claim that there exist a set of real numbers γ_k such that

$$\Delta \otimes S = \sum_k \gamma_k M^k \qquad \sum_k \gamma_k = 0 \tag{3.12}$$

where M^k is a permutation matrix. Equivalently,

$$\Delta_{ij} = \sum_k \gamma_k \frac{m_{ij}^k}{s_{ij}} \tag{3.13}$$

Consider for algebraic convenience consider the vectorization form of a matrix; the column vector form of matrix Δ is denoted by $\underline{\Delta}$. By classical Taylor's Theorem for multivariate functions,

$$f_P(\alpha) - f_P(\hat{\alpha}) = \underline{\Delta}^T \nabla f_P(\hat{\alpha}) + \frac{1}{2}\underline{\Delta}^T H(\underline{\eta})\underline{\Delta} \tag{3.14}$$

where $H(\underline{\eta})$ is the Hessian matrix computed in $\eta = (1-\gamma)\hat{\alpha} + \gamma\alpha = \hat{\alpha} + \gamma\Delta$, for some constant $\gamma \in [0,1]$. Equivalently,

$$\eta_{ij} = \hat{\alpha}_{ij} + \gamma\Delta_{ij} \tag{3.15}$$

We first show that the first term in the right hand side of (3.14) is null. Indeed, by (3.12) and (3.13):

$$\begin{aligned}
\underline{\Delta}^T \nabla f_P(\hat{\alpha}) &= \sum_{ij} \frac{-2s_{ij}}{\hat{\alpha}_{ij}^3} \sum_k \gamma_k \frac{m_{ij}^k}{s_{ij}} = \\
&= \sum_{ij} (-2\rho^3) \sum_k \gamma_k m_{ij}^k = \\
&= (-2\rho^3) \sum_k \gamma_k \sum_{i,j} m_{ij}^k = \\
&= (-2\rho^3) \sum_k \gamma_k N = 0
\end{aligned}$$

thanks again to (3.12).

Let us consider now the second term in the right hand side of (3.14). Observe that $H(\underline{\alpha})$ is a diagonal matrix, in which the element corresponding to (i,j) pair is equal to $6s_{ij}/\alpha_{ij}^4$. Hence, by (3.15):

$$\underline{\Delta}^T H(\underline{\eta})\underline{\Delta} = \sum_{i,j} \Delta_{ij}^2 \frac{6s_{ij}}{\eta_{ij}^4} = \sum_{i,j} \Delta_{ij}^2 \frac{6s_{ij}\rho^4}{(1+\gamma\rho\Delta_{ij})^4}$$

Let $\epsilon' = \min_{ij}\{\Delta_{ij} | \Delta_{ij} > 0\}$ and $s' = \min_{ij}\{s_{ij}\}$. Finally, we can claim

$$f_P(\alpha) - f_P(\hat{\alpha}) = \underline{\Delta}^T H(\underline{\eta})\underline{\Delta} \geq \frac{6\rho^4(\epsilon')^2 s'}{(1+\gamma\rho\epsilon)^4} > 0$$

that means that any $\alpha \neq \hat{\alpha}$ that satisfies (3.9) and (3.10) cannot be the optimal solution.

The minimum power cost factor is immediately obtained by computing $f_P(\hat{\alpha})$. □

In Sec. 3.4, we validate the cubic relation between power and load through the results of the actual hardware synthesis of a crossbar chip. Furthermore, we can get an important intuition from the above theorem, which will drive the design of approximated algorithms for the Cont-MP problem: *In the optimal solution, all the α_{ij} are expanded proportionally by the same factor.*

When considering also the constraint on $V_{\min}$, the expansion ratio is limited by $\alpha_{ij} \leq 1/\beta$. For ρ-double-stochastic matrices, the optimal solution becomes

$$\alpha_{ij} = \min\left(\frac{1}{\rho}, \frac{1}{\beta}\right), \qquad \forall i, j$$

and the corresponding optimal solution for CONT-MP becomes:

$$f_P(\hat{\alpha}_{\text{CONT-MP}}) = \begin{cases} N\rho\beta^2 & \text{if } \rho < \beta \\ N\rho^3 & \text{if } \rho \geq \beta \end{cases} \tag{3.16}$$

Thus, β is the value of "critical load" above which DVFS is not able to expand the bit duration due to the constraints imposed by the traffic load in (3.2).

Consider now a relaxed version of the CONT-MP problem, denoted as MISO-MP (Multiple-Inputs Single-Output), in which we remove the expansion constraints (3.9) on each input.

Theorem 7. *Given any admissible traffic matrix S, the optimal solution of* MISO-MP *is given by $\alpha_{ij} = 1/\rho_j^O(S)$. The corresponding power cost factor is:*

$$f_P(\hat{\alpha}_{\text{MISO-MP}}) = \sum_j (\rho_j^O(S))^3$$

Note that this results does not require S to be a double-stochastic matrix.

Proof. Define the Lagrange function as

$$\Lambda = \sum_{ij} \frac{s_{ij}}{\alpha_{ij}^2} + \sum_j \lambda_j \left(\sum_k s_{kj}\alpha_{kj} - 1 \right)$$

A necessary condition for the solution to be optimal is $\partial\Lambda/\partial\alpha_{ij} = 0$, which implies $-2s_{ij}\alpha_{ij}^{-3} + \lambda_j\alpha_{ij}s_{ij} = 0$. It should be $\alpha_{ij} = (2/\lambda_j)^{-4}$, i.e. for a fixed j, all the α_{ij} are constant. Thus (3.10) becomes $\alpha_{ij}\sum_k s_{kj} = 1$ and hence $\alpha_{ij} = 1/\rho_j^O(S)$. This satisfies also (3.11). By simple substitution, we get the corresponding power cost. □

Property 3. $f_P(\hat{\alpha}_{\text{MISO-MP}}) \leq f_P(\hat{\alpha}_{\text{CONT-MP}})$

i.e. MISO-MP provides a lower bound, simple to compute, for CONT-MP and OPT-MP under any admissible traffic matrix.

Power consumption without DVFS

A feasible, but not optimal, solution for Opt-MP is when no DVFS scheme is adopted, i.e. $\alpha_{ij} = 1$ for all i, j. We define this scheme as NoDVFS and the corresponding solution as $\hat{\alpha}_{\text{NoDVFS}}$. The power cost factor f_P under any admissible traffic matrix S can be obtained by setting $\alpha_{ij} = 1$ in (3.3):

$$f_P(\hat{\alpha}_{\text{NoDVFS}}) = \sum_{i=1}^{N}\sum_{j=1}^{N} s_{ij} = N\rho_{\text{ave}}(S) \tag{3.17}$$

denoting a linear relationship between the average load on S and the total power consumption.

Property 4. $f_P(\hat{\alpha}_{\text{Opt-MP}}) \leq f_P(\hat{\alpha}_{\text{NoDVFS}})$.

Thus $f_P(\hat{\alpha}_{\text{NoDVFS}})$ is a loose upper bound for Opt-MP. We define the *relative power* $\eta(\hat{\alpha})$ of a DVFS solution $\hat{\alpha}$, relative to NoDVFS, as:

$$\eta(\hat{\alpha}) = \frac{f_P(\hat{\alpha})}{f_P(\hat{\alpha}_{\text{NoDVFS}})} = \frac{f_P(\hat{\alpha})}{N\rho_{\text{ave}}(S)}. \tag{3.18}$$

Since $\eta(\hat{\alpha}) \in [0,1]$, the closer $\eta(\hat{\alpha})$ to zero, the larger the scheme gain with respect to NoDVFS.

For double-stochastic matrices, dividing (3.17) by (3.16):

Property 5. *Under ρ-double-stochastic matrices*

$$\eta(\hat{\alpha}_{\text{Cont-MP}}) = \begin{cases} \beta^2 & \textit{for } \rho < \beta \\ \rho^2 & \textit{for } \rho \geq \beta \end{cases}$$

In summary, the solution to the Cont-MP problem, which uses any voltage level between $V_{\min}$ and $V_{\max}$, provides a lower bound for the power of the Opt-MP problem. When the matrix is double-stochastic, the optimal solution to Cont-MP is trivial. Otherwise, a lower bound can be found with the solution of Miso-MP, trivial to compute.

3.2.3 Power control algorithms

To solve Opt-MP for any traffic matrix we propose to:

1. solve the corresponding Cont-MP problem
2. approximate each α_{ij} to the closest smaller value available in the set $\mathcal{A}$

In other words, if α_{ij} is the solution for CONT-MP, then use for OPT-MP:

$$\alpha'_{ij} = \max\{\alpha \in \mathcal{A} \mid \alpha \leq \alpha_{ij}\}$$

Note that, by construction, the set of α'_{ij} defines an admissible solution for OPT-MP.

To solve CONT-MP, we adopt a quasi-optimal algorithm based on the logarithmic barrier method for convex problems [24] which provides an ϵ-approximation of the optimal solution. Furthermore, we adopt a two-steps algorithm: we augment S to a double stochastic $\hat{S}$ according to one of algorithms among AUGM-1, AUGM-MAX or AUGM-SORT, described below. Then, we compute $\alpha_{ij} = \hat{s}_{ij}/s_{ij}$.

INCREASE-MATRIX Algorithm

Input: $N \times N$ matrix $S = [s_{ij}]$, $\{\rho_i^I\}_{i=1}^N$, $\{\rho_j^O\}_{j=1}^N$, ρ_T, Ω^I, Ω^O.

Output: $N \times N$ matrix $\Delta = [\delta_{ij}]$

1. $\delta_{ij} = 0$ for any $1 \leq i, j \leq N$
2. $\Omega^{IO} = \{(i,j) : i \in \Omega^I, j \in \Omega^O\}$
3. **repeat** until no choice is anymore available
4. **choose** any $(i,j) \in \Omega^{IO}$ such $\max\{\rho_i^I, \rho_j^O\} < \rho_T$
5. $\delta_{ij} = \min\{\rho_T - \rho_i^I, \rho_T - \rho_j^O\}$
6. $\rho_i^I = \rho_i^I + \delta_{ij}, \quad \rho_j^O = \rho_j^O + \delta_{ij}$

We now describe the INCREASE-MATRIX procedure, on which all the augmentation algorithms are based. The inputs of the procedure are

1. a sub-stochastic matrix S
2. the corresponding row ρ_i^I and column ρ_j^O sums
3. a target load value ρ_T such that

$$\max_k\{\rho_k^I, \rho_k^O\} \leq \rho \leq 1$$

4. a set of input ports Ω^I and output ports Ω^O

The algorithm returns a matrix $\Delta = [\delta_{ij}]$ with the largest possible elements such that:

1. only the elements δ_{ij} corresponding to rows and columns present in both Ω^I and Ω^O may be > 0
2. the maximum row and column sum is ρ_T, i.e.

$$\sum_{k=1}^{N} s_{ik} + \delta_{ik} \leq \rho_T \text{ for any } i \in \Omega^I$$
$$\sum_{k=1}^{N} s_{kj} + \delta_{kj} \leq \rho_T \text{ for any } j \in \Omega^O$$

The algorithm operates only on a sub-matrix restricted to the rows in Ω^I and the columns in Ω^O. It chooses a sequence of elements whose row and column sum to less than ρ_T. Then, each element in the sub-matrix is augmented to reach ρ_T without violating the constraints. Note that the maximum number of iterations in step 3 is upper bounded by $2N$.

Having defined INCREASE-MATRIX, we now describe the algorithms we propose to augment S to a double-stochastic $\hat{S}$:

- AUGM-1:
  1. compute ρ_i^I and ρ_j^O for any i and j;
  2. run INCREASE-MATRIX on S, ρ_i^I, ρ_j^O, $\rho_T = 1$, $\Omega^I = \Omega^O = \{1, \ldots, N\}$;
  3. compute $\hat{s}_{ij} = s_{ij} + \delta_{ij}$ for all i and j.

Note that AUGM-1 is a classical iterative algorithm (see Sec. II.A of [23]) to augment a sub-stochastic matrix to a double-stochastic one. The complexity is $O(N^2)$, due to steps 1) and 3).

- AUGM-MAX:
  1. compute ρ_i^I and ρ_j^O for any i and j;
  2. compute $\rho_{\max}(S)$;
  3. run INCREASE-MATRIX on S, ρ_i^I, ρ_j^O, $\rho_T = \rho_{\max}(S)$, $\Omega^I = \Omega^O = \{1, \ldots, N\}$;
  4. compute
$$\hat{s}_{ij} = s_{ij} + \delta_{ij} + \frac{1 - \rho_{\max}(S)}{N}$$
  .

The complexity of AUGM-MAX is $O(N^2)$, due to steps 1) and 4).

- AUGM-SORT:

1. compute ρ_i^I and ρ_j^O on S for any i and j;
2. sort ρ_i^I and ρ_j^O in increasing order. Let $i_{(k)}$ be the kth input and $j_{(k)}$ be the kth output in such increasing sequences;
3. initialize an auxiliary matrix $X^{(0)} = S$ and set $\Omega_0^I = \Omega_0^O = \emptyset$;
4. iterate, for k from 1 to N, the following steps:
   (a) $\Omega_k^I = \Omega_{k-1}^I \cup i_{(k)}$, i.e. the set of the inputs with the k smallest row sums;
   (b) $\Omega_k^O = \Omega_{k-1}^O \cup j_{(k)}$, i.e. the set of the outputs with the k smallest column sums;
   (c) run Increase-Matrix on $X^{(k-1)}$, ρ_i^I, ρ_j^O, Ω_k^I, Ω_k^O and $\rho_T^{(k)} = \max\{\rho_{i_{(k)}}^I, \rho_{j_{(k)}}^O\}$, i.e. $\rho_T^{(k)}$ is the maximum load for the first kth inputs and outputs of S;
   (d) $x_{ij}^{(k)} = x_{ij}^{(k-1)} + \delta_{ij}$ for any i, j, i.e. set $X^{(k)} = X^{(k-1)} + \Delta$;
   (e) eventually go to a) to start a new iteration;
5. compute

$$\hat{s}_{ij} = x_{ij}^{(N)} + \frac{1 - \rho_{\max}(X^{(N)})}{N}$$

The complexity of Augm-Sort is $O(N^2)$ by optimizing the data structure to choose an $(i, j) \in \Omega^{IO}$ in Increase-Matrix and by sorting only once ρ_i^I and ρ_j^O.

Theorem 6 suggests that the optimal way to increase the S is proportionally, at least for some families of traffic. Augm-1 is a classical way to augment a matrix. Instead, Augm-Max and Augm-Sort tend to augment the matrix more proportionally.

3.3 Performance evaluation

We first discuss the performance for ρ-double-stochastic matrices. Then, we move to ρ-sub-stochastic matrices.

3.3.1 Power consumption for double-stochastic matrices

According to Theorem 6, the optimal solution for Cont-MP is expressed by (3.16). Fig. 3.2 shows the *power consumption per port* $f_P(\hat{\alpha})/N$ vs. the average load, for the optimal solution of Cont-MP and $\beta \in \{0.3, 0.5, 0.7\}$. We show also the linear growth of NoDVFS, computed with (3.17).

For small loads, DVFS is very efficient, by reducing the power by a factor $1/\beta^2$ (see Property 5), equal to 11, 4 and 2, respectively, for each value of β. For larger loads, the DVFS power reduction decreases, becoming negligible in highly loaded conditions, because bit expansion is not allowed due to the high traffic load.

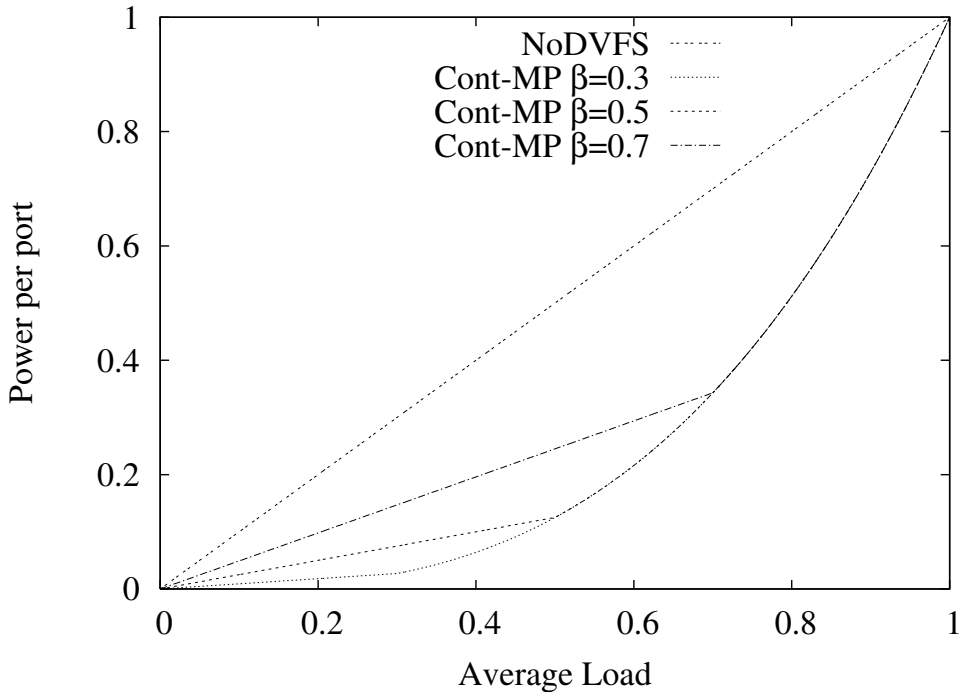

Figure 3.2. Optimal solution for continuous DVFS (Cont-MP), under any ρ-double-stochastic matrix.

We now consider the effect of a finite set $\mathcal{A}$ of voltage levels. Table 3.1 shows the worst-case (for any load) ratio between the consumption of Opt-MP with finite set of voltage levels and the consumption of Cont-Opt with continuous DVFS, as a function of the number of available voltage levels. The $|\mathcal{A}| - 2$ intermediate voltage levels between $V_{\min}$ and $V_{\max}$ have been numerically optimized to minimize such ratio. Note that very few intermediate levels (i.e., one for $\beta = 0.5$) are sufficient to observe differences lower than 10%. Hence, the simple solution to Cont-MP well approximates the solution to the Opt-MP problem. Finally, very few voltage levels are enough to exploit the potential benefits of DVFS.

3.3.2 Power consumption for sub-stochastic matrices

We consider the family of random traffic matrices generated as follows. Given $\rho \in (0,1]$, generate a matrix $U = [u_{ij}]$ of N^2 random variables, uniformly distributed on the interval (0,1]. Then, derive each element of S as

$$s_{ij} = \frac{u_{ij}\rho}{\rho_{\max}(U)}$$

Table 3.1. The power consumption ratio between DVFS with discrete voltage levels (Opt-MP) and continuous DVFS (Cont-MP), for double-stochastic matrices

$\lvert\mathcal{A}\rvert$	β	$\frac{\text{Voltage levels}}{V_{\max}}$	$\max_{0\le\rho\le1} \frac{f_P(\hat{\alpha}_{\text{Opt-MP}})}{f_P(\hat{\alpha}_{\text{Cont-MP}})}$
3	0.3	0.3,0.55,1	1.31
	0.5	0.5,0.71,1	1.09
	0.7	0.7,0.84,1	1.02
4	0.3	0.3,0.45,0.67,1	1.13
	0.5	0.5,0.63,0.79,1	1.04
	0.7	0.7,0.78,0.89,1	1.01
5	0.3	0.3,0.41,0.55,0.74,1	1.07
	0.5	0.5,0.60,0.71,0.84,1	1.02
	0.7	0.7,0.76,0.84,0.92,1	1.01

Using this construction, it can be shown that the corresponding average load

$$\rho_{\text{ave}}(S) \approx \frac{\rho}{1+\sqrt{0.67\frac{\log(N)}{N}}}$$

for large enough N.

We compare the algorithms proposed in Sect. 3.2.3 for continuous DVFS, because, as shown in the previous section, Cont-MP is a good approximation of Opt-MP even when few voltage levels are available. We show the optimal solution for Cont-MP only for smaller switch sizes ($N = 16$), due to computational constraints. We report also the solution for the lower bound provided by Miso-MP. Even if the results hold for $\beta = 0.3$, similar results were obtained for other values of β.

Figs. 3.3, 3.4 show the relative power (Eq. (3.18)), for different N. Note that, to ensure admissibility, the maximum average load in the abscissa is limited by construction to be always less than $1/(1+\sqrt{0.67\log(N)/N})$, i.e. 0.75 and 0.88 for $N = 16$ and $N = 256$ respectively.

When increasing $\rho_{\text{ave}}(S)$, the relative power of Miso-MP shows a quadratic growth, similarly to double-stochastic matrices for which Property 5 holds. The behavior is close to the optimal solution, justifying its use to approximate Cont-MP for large N. Even if not optimal, Augm-Sort and Augm-Max show performance close to the optimal. On the contrary, Augm-1 behaves the worst, only providing minor power reductions with respect to NoDVFS.

Similar results holds For $N = 256$ in Fig. 3.4. We were unable to obtain the

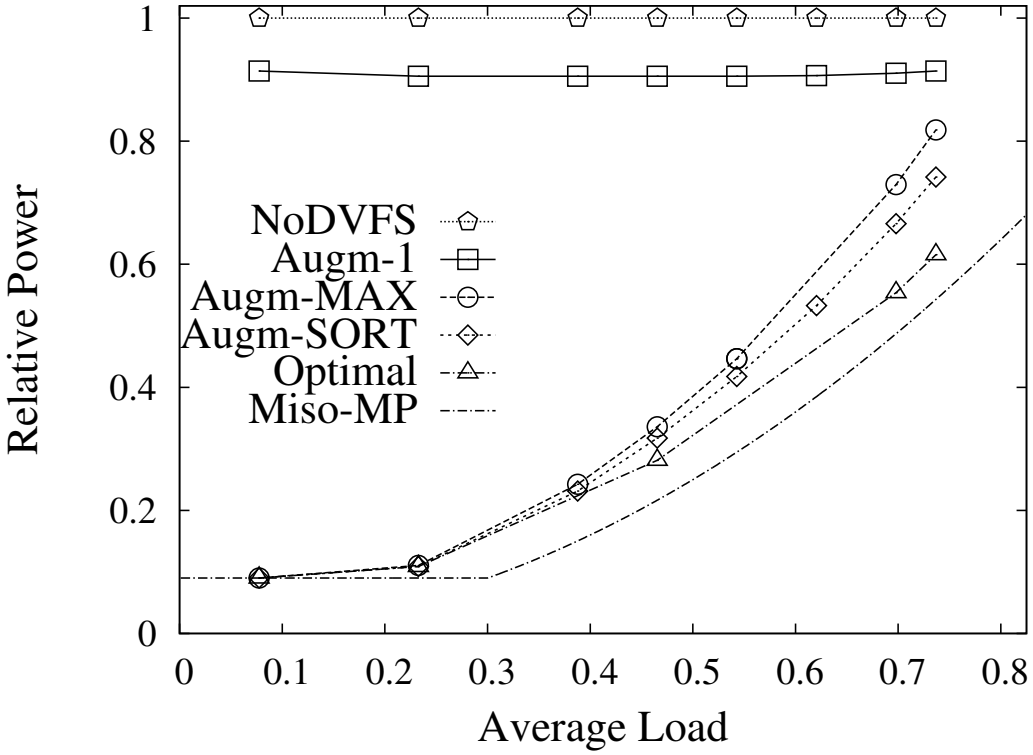

Figure 3.3. Relative power for $N = 16$ and $\beta = 0.3$, under sub-stochastic matrices

optimal solution in reasonable time. Augm-1 does not provide any benefit. Augm-Sort and Augm-Max provide performance close to the lower bound Miso-MP. Thus, these DVFS schemes appear to be the most efficient, especially at low average load, regardless of the switch size.

3.4 Hardware design and evaluation

To better explore the effects of DVFS on a real switch fabric, a 128×128 crossbar switch was adopted as a case study. To optimize crossbar scalability, instead of the classical X-Y architecture, we choose a mux-tree based pipelined architecture. Indeed, in classical X-Y based crossbar switches [25], any input–output connection is provided by horizontal and vertical wires spanning the whole area.

Hence, propagation delay along wires tends to grow rapidly with the number of input-output ports and soon becomes the limiting factor for throughput performance. Multiple bit slices can be used to cope with limited clock frequency, while reaching at the same time high line throughput. However, in this case, improved

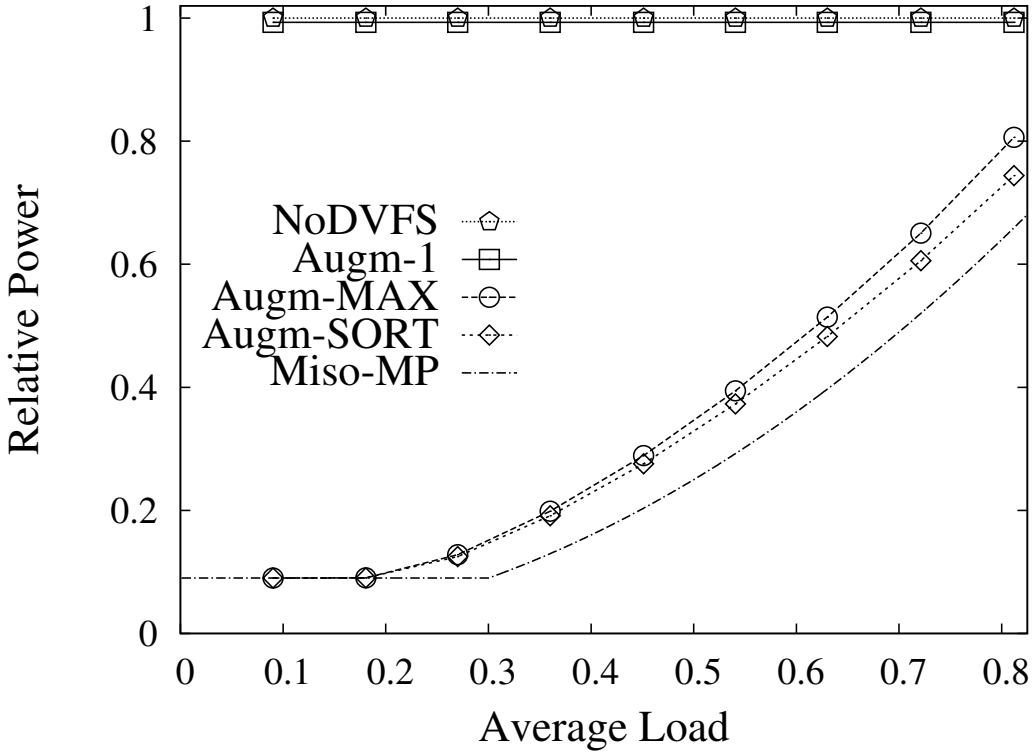

Figure 3.4. Relative power for $N = 256$ and $\beta = 0.3$, under sub-stochastic matrices.

performance comes at the cost of additional implementation complexity.

High data rates over a large switch, with more than one hundred input output ports, can be obtained at a lower implementation complexity with a mux-tree based pipelined architecture [25], shown in Fig. 3.5. Each output is connected through a *tree* of multiplexers that receive all input ports.

Two basic features of the tree organization can be exploited to improve speed:

1. the entire multiplexing operation can be split in several tree stages, with each

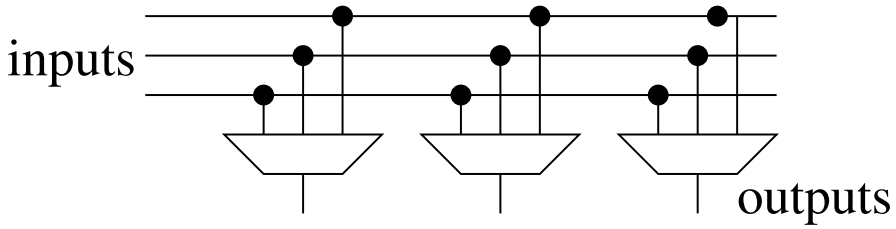

Figure 3.5. Mux-based 3×3 crossbar

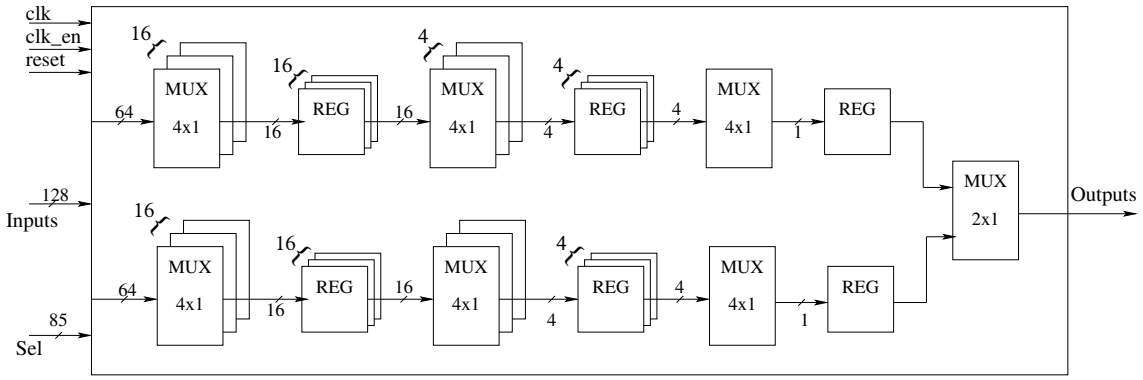

Figure 3.6. Architecture of a slice of the switch fabric

stage individually sized to match timing constants according to its load capacitance

2. pipeline registers can be inserted along the tree to cut critical path delays, thus achieving very high clock frequency

The mux–tree based pipelined switch of size 128×128 was modeled using VHDL language and synthesized to derive area occupation, achievable throughput and dissipated power. Fig. 3.6 shows the structure of a single slice of the crossbar fabric: each input port receives data serially and the 128 inputs are divided into two parts, where the upper (and the lower) portion deals with 64 inputs. Internal registers are used to provide pipelining. In the upper half of the fabric, 16 multiplexers and 4 multiplexers are contained in the first and second pipeline stages respectively. A 4×1 multiplexer is allocated in the third pipeline stage. The same structure is repeated in the lower half, and a 2×1 multiplexer is used for the final selection. Thus, the showed slice forms a 128×1 multiplexer with pipelining. To control the whole set of multiplexers, 85 select lines are required.

The complete fabric architecture consists of 128 slices equal to the one given in Fig.3.6. The same data inputs are applied to each slice and a total of $128 \times 85 = 10880$ select lines are used to control the switch. Destination conflicts are not allowed in the described architecture, and are prevented by a proper scheduling algorithm [22].

A further important property of the adopted switch fabric architecture is its modularity. This feature enables the possibility to adopt a hierarchical synthesis flow that simplifies the floorplan. Additionally, although this is not exploited in this work, the modular structure of the switch also allows for applying different choices of voltage and frequency scaling to individual slices. Assuming that a lower traffic is observed along paths associated with a specific slice, then voltage and frequency

scaling for this single slice would be beneficial to reduce power consumption and would allow at the same time for higher throughput across different slices.

The VHDL code of the fabric was written, debugged and simulated under Mentor Graphics Modelsim using randomly generated patterns of input data. Synthesis was performed using Synopsys Design Compiler on a 90 nm CMOS technology. The power analysis of the switch fabric was performed using Synopsys Power Compiler. We do not consider the power contribution due to the implementation of the power control algorithm or any other component because we focus on the crossbar chip. We restrict our analysis to the synthesis results and we do not consider the consumption due to the actual chip layout; hence, our power consumption results are relative. Derived power dissipation figures are based on the actual switching activities measured at circuit nodes during simulation of the fabric in the presence of different test patterns.

Thanks to the high level of applied pipelining, the maximum operating frequency of the designed crossbar, when the supply voltage is not scaled, is as high as 3.2 GHz, allowing to reach an aggregated bandwidth of 410 Gbps. To evaluate the potential of the described DVFS approach, the crossbar was synthesized with several values of supply voltage and frequency of the clock signal. Six scaling factors (i.e. $\{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, corresponding to $\alpha = \{2.50, 2.00, 1.67, 1.43, 1.25, 1.11\}$), were used to reduce supply voltage. In addition, the clock frequency, f_{CK}, was changed in the range between the maximum achievable value of 3.2 GHz down to 200 MHz, equally for all the ports. Hence, the corresponding traffic matrix S is ρ-double-stochastic with all $s_{ij} = \rho/N$ and $\rho = f_{CK}/(3.2\text{ GHz})$. The power consumption in the fabric is associated with the switching activity in the slice components and therefore to the average data throughput. For each selected value of f_{CK}, the maximum possible data rate has been assumed for input data. For example, with $f_{CK} = 1.2$ GHz, data are received at the rate of 1,200 Mbps per input port. The select lines which control the multiplexers are assumed to switch at a 1000 times lower rate. Note that power would also be consumed to change between voltage levels.

Furthermore, each transition to new values of supply voltage and f_{CK} introduces a latency, which may affect the global throughput. However, for simplicity reasons, latency and power overheads generated by these transitions are not considered in this study.

Switch fabric power consumption per port is reported in Fig. 3.7 for different voltage scaling factors and clock frequencies. The theoretical curve is $\rho_{\text{ave}}(S)^3$. As expected, power consumption scales linearly with f_{CK} and thus with input data rate, but the slope depends on the applied voltage scaling. Therefore different power reduction gains can be obtained at different input data rates. For example, if input data rate is 50% of the maximum allowed level, 75% of the dissipated power can be saved, from 4.2 mW with no applied DVFS to 1 mW with a voltage scaling

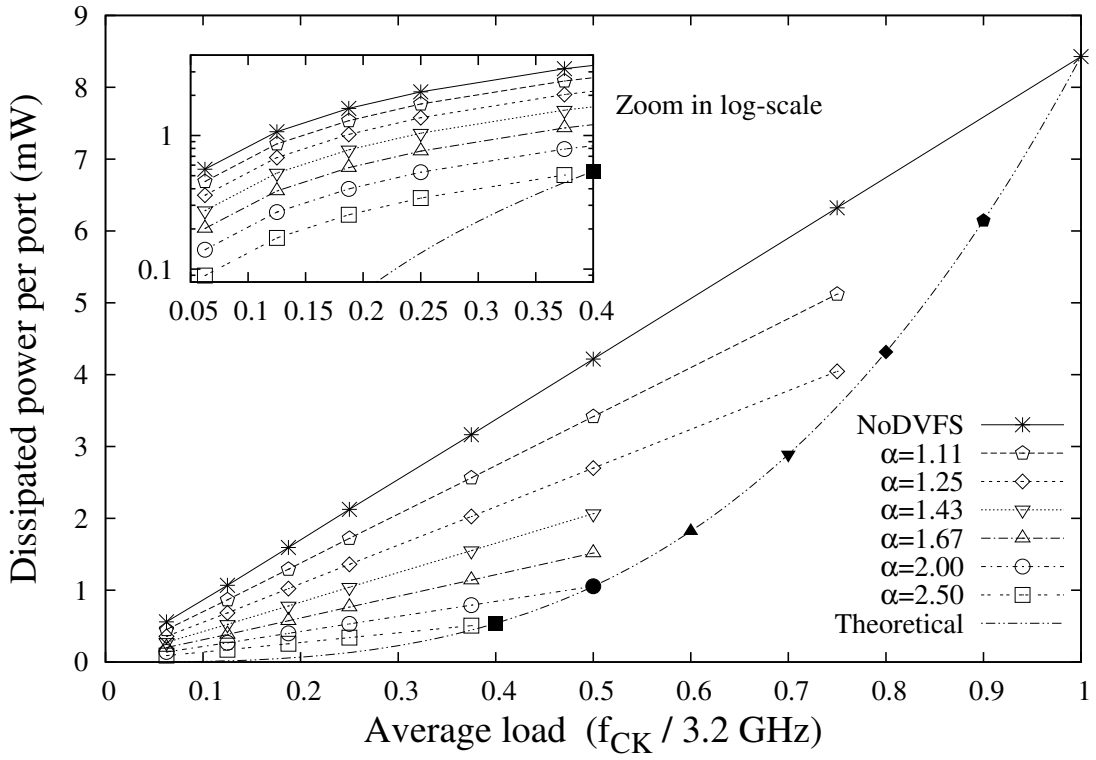

Figure 3.7. Power obtained by the VHDL synthesis, for a 128×128 crossbar with 410 Gbps bandwidth.

factor equal to 0.5. A lower reduction of dissipated power is possible when working at higher data rates: with input data at 75% of the maximum frequency, the dissipated power can be reduced by 51% from 6.3 mW to 3.1 mW.

Furthermore, the filled points on the theoretical curve for a specific load ρ are aligned with the linear interpolation of the powers obtained for a specific value of $\alpha = 1/\rho$. This means that the cubic dissipation model of Theorem 6, based on a single expansion factor for the whole crossbar, is accurate.

Chapter 4

Energy Profiling of Network Equipment for Rate Adaptation Technologies

4.1 Background

[1] In this section we summarize prior work on energy profiling that is relevant to rate adaptation and provide an overview of PTRA, BTRA, and DTRA techniques.

4.1.1 Energy profiling overview

We review energy profiling approaches from the literature, focusing on those that are suitable for inclusion in DTRA frameworks because they explicitly map system and traffic configurations onto power consumption levels.

Chabarek's linear model

In [34], Chabarek et al. construct energy profiles for two IP routers manufactured by Cisco Systems, namely the GSR 12008 core router and the 7507 edge router. The GSR 12008 chassis contains twelve slots, of which one accommodates the route processor card and two others are dedicated to switch fabric modules (10 Gbps

[1]Let me thanks all people met at Alcatel-Lucent Bell Laboratories, it was an extraordinary experience. In particular I would like to thank Steven Fortune and Thierry Klein for their support, their help and to all people who offer their contribution to the assembly of the experimental testbed in Murray Hill, NJ. Another thank is for the researchers of DOE-ESNet for their technical and logistical support before and during execution of the power measurements at the ANI Testbed facility in Brookhaven, NY.

capacity each). The remaining nine slots can be used for network adapter cards, each with capacity up to 4 Gbps. The 7507 has a seven-slot chassis, with one slot for the route processor card and the other six slots for adapter cards, each with capacity up to 1Gbps.

A first set of experiments yields for each system the power contribution of the chassis and of the different types of modules that can be installed in the adapter slots. No cables are attached to the network interfaces, so obviously no traffic flows through the system. The measured power consumption is the bare sum of the chassis and card contributions in idle state. A second set of experiments focuses on the GSR 12008, in a configuration that includes the route processor card, the switch fabric cards, one adapter card with four 1Gbps Ethernet (1GbE) interfaces (of which only three connected), and one adapter card with one OC-48 interface, also connected. Traffic flows from the three 1GbE interfaces to the OC-48 link. The experiments focus on the power consumption effects of different types of traffic (CBR versus bursty TCP traffic), of different packet sizes (100, 576, and 1500 bytes), of different routing table sizes, and of different routing functions.

Overall, the power contribution that derives from the presence of traffic is relatively small compared to the total. The differences observed across different traffic configurations are even smaller. The following equation defines the linear model adopted for construction of the energy profiles:

$$S = C_0 + \sum_{i=1}^{N_L} (L_{0,i} + L_{b,i}(\beta_i)) \tag{4.1}$$

where S is the total power consumption, C_0 is the power consumed by the chassis when idle, L_N is the number of line cards, possibly of different types, that are plugged into the chassis, $L_{0,i}$ is the power consumed by a line card i when idle, and, $L_{b,i}(\beta_i)$ is the additional power contribution of the same line card when traversed by traffic at bit-rate load β_i ($0 \leq \beta_i \leq 1$) where $\beta_i = 1$ when line card i is fully loaded). In (4.1) and in all linear model equations that follow, the chassis power includes contributions from the power modules, the cooling system, the switch fabric, and the control processor board, whether physically integrated or distributed in multiple modules. We remark that , $L_{b,i}(\ )$ is a generic, not necessarily linear function of the line card load β_i. The definition of β_i does not distinguish explicitly between input and output load (meaning "from the network to the line card" in the former case and "from the line card to the network" in the latter).

In [34], the model of (4.1) is primarily utilized in the optimization of network planning decisions that set the placement of network nodes and the allocation of line cards per node. Consistently with its application, the model does not attempt to single out the power contribution of individual network ports, especially those that are connected to a network cable but not enabled for handling traffic. To support

DTRA techniques it is desirable to enhance the model.

Mahadevan's Linear Model

In [35], Mahadevan et al. obtain energy profiles for seven systems of different sizes and capabilities, including an Ethernet hub, a wireless access point, three edge LAN switches, one core switch, and one edge router. The following equation defines the reference model for construction of the energy profiles:

$$S = C_0 + \sum_{i=1}^{N_L} L_{0,i} + \sum_{j=1}^{N_P} P_{b,j}(\beta_j) \tag{4.2}$$

where C_0,N_L and $L_{0,i}$ are the same as in (4.1), N_P is the total number of ports that are connected and enabled, $P_{b,j}$ is the power consumed by a port j when loaded at full bit rate, which depends on the type of the port and on its rate configuration, and β_j is the bit-rate load sustained by the port ($0 \leq \beta_j \leq 1$) where $\beta_j = 1$ when the port is fully loaded). Compared to (4.1), the linear model of (4.2) explicitly includes the power contribution of individual ports. However, the model assumes that a port consumes power only when there is traffic flowing through it. In absence of traffic, the power contribution of a port is null, whether the port is enabled or disabled. The measurements that we present show that this assumption can mislead the energy optimization procedures of DTRA frameworks. The authors of [35] indeed address this issue in [28], where they apply a slightly modified equation with the port load term β_j of (4.2) removed:

$$S = C_0 + \sum_{i=1}^{N_L} L_{0,i} + \sum_{j=1}^{N_P} P_j \tag{4.3}$$

where P_j is the power consumed by port j when enabled, irrespective of the traffic that flows through it. Although not explicitly stated in [28], we can safely assume that P_j in 4.3 is measured under the same traffic conditions as $P_{b,j}$ in 4.2, that is, when port j is fully loaded. The conversion of the load term into a fixed contribution most likely derives from the practical observation that, especially in Ethernet switches, power consumption shows very little sensitivity to the traffic volume handled by a port.

The simplification fits well the scope of the work presented in [28], which focuses exclusively on two Ethernet switches for data center applications, but may not be appropriate for systems like IP routers, which present a heavier energy overhead per packet header. Nevertheless, the attribution of a fixed power offset P_j to each active port j does ensure that 4.3 is more accurate than 4.2 in the context of DTRA operations, because it captures the power consumption effects of switching network links on and off.

Tamm's linear model

In [36], Tamm et al. deliver a comprehensive study of the distribution of power consumption among the functional components of a large set of Alcatel-Lucent network systems, including optical switches, Ethernet switches, and IP routers. The results help identify key hotspots for energy savings in systems designs and offer high-level indications of the benefits that may derive from the introduction of BTRA/PTRA-capable hardware in circuit packs. However, all power measures are obtained from typical ratings of common hardware components and not directly from experimental measurements. Also, the power distribution models presented do not provide a direct mapping of system and traffic configurations onto power consumption levels, falling short of supplying DTRA algorithms with the type of information that they require for running their network-wide energy optimizations.

4.1.2 Rate adaptation overview

We start our rate adaptation review with PTRA because it is one of the building blocks of the most advanced DTRA frameworks and its description covers most of the essential concepts of BTRA.

Packet-Timescale Rate Adaptation (PTRA)

PTRA techniques target the design of individual hardware components in the data path of network systems. They provide those components with multiple operating states, each state being characterized by a traffic processing rate (expressed in bits per second or packets per second depending on the function of the component) and a corresponding power consumption level. The goal is to minimize the energy spent by the hardware component to sustain the traffic workload that it receives from the data path. State-setting decisions occur at the micro/millisecond timescale, in response to fluctuations in traffic arrival rates and packet queue occupancies. Nedevschi et al. studied in [32] the application of sleep-state exploitation (SSE) and rate-scaling (RS) techniques to the links of a network. With SSE, a link alternates between only a full capacity state (at full power) and a low-power sleep state. With RS, a link can choose from an extended set of operating states that lie along a convex curve in the power-rate plane. In [33], Francini and Stiliadis refined the specification of the two techniques by embedding robust constraints on the packet delay degradation that they induce and formalized a new hybrid rate adaptation (HRA) scheme that combines the best properties of the two approaches. The IEEE 802.3az standard [39], approved in 2010, is an important example of PTRA instantiation in network elements. A fundamental property of PTRA, not always fully appreciated, is that a mandate to keep the state transition time well within the sub-millisecond

range guarantees that the technology is virtually transparent to the operation of the network. If the state transitions took longer to execute, the technology would simply not be suitable for widespread deployment in packet networks. Therefore, provided that the state transition time mandate is satisfied, network links and nodes are never seen missing by the rest of the network, even when most of their hardware components are in their low-power sleep states. Likewise, PTRA is never directly the cause of packet losses or of disruptive degradations in the performance of network protocols and applications.

Bit-Timescale Rate Adaptation (BTRA)

Compared to PTRA, BTRA techniques trade lower energy savings for simplicity and faster state transitions. To enter the sleep state of a PTRA scheme, a component must gate its clock signal and also adjust the power supply distribution network so that leakage currents are minimized (e.g., by resetting the supply voltage). The latter step and the reverse one needed to bring the component back into operation dominate the quantification of the state transition time, so that it spans over multiple packet transmission times. To prevent back-to-back transitions from excessively diluting the amount of data processed per unit of energy consumed, PTRA state setting policies must ensure that a minimum hold time is enforced in between consecutive transitions. This minimum hold time is the main contributor to the packet delay degradation introduced by a data path component with PTRA capabilities.

BTRA techniques retain the clock signal gating of PTRA but drop the adjustment of the power supply distribution network. State transitions can now occur in one clock cycle: the increase of packet delay is practically erased but the energy savings are also drastically reduced.

Demand-Timescale Rate Adaptation (DTRA)

An excellent example for the illustration of the goals and mechanics of DTRA techniques can be found in [28]. The paper uses simulations to estimate the energy savings that can be obtained in the Ethernet switching infrastructure of a data center by turning off unused switches, disabling unused ports, and adapting link capacities. The input to the simulation experiments is a 5-day trace of traffic demands averaged over 10-minute periods. A first round of tests produces ideal results under the assumption that a centralized power controller knows ahead of time the evolution of the traffic demands. More realistic results are subsequently obtained with predictors based on real-time traffic measurements. Load prediction errors translate into link overload conditions with higher queueing delays and packet losses, or simply wasted energy.

The portion of the network topology that is subject to DTRA control consists

of a set of 1-redundant trees with two tiers of switches. The algorithm that assigns processing jobs to the servers at the leaves of the trees is designed to minimize the overall energy consumption of the two tiers of Ethernet switches. The energy profiles of the switches conform to the model of (4.3) and provide the foundation of the job assignment algorithm. Using the best of the three job assignment algorithms studied in the paper, energy savings up to 75% can be obtained within the two switched tiers if evident performance impairments are accepted with respect to queueing delay and service availability. With a more conservative scheme that avoids any degradation of data center performance the maximum savings amount to 20%. In [29], Antonakopoulos et al. apply power-aware routing to variously meshed topologies for IP autonomous system (AS) networks. Compared to the tiered switching networks of [28], the AS networks present different hop counts for alternate paths between endpoints. As a consequence, the energy benefits of any diversion from the basic shortest-path routing are partially reduced by the associated increase in the average number of hops per end-to-end path. The reference model for power consumption only focuses on network ports and excludes contributions from the chassis and line cards:

$$S = \sum_{j=1}^{N_P} \left(P_{0,j+P_{b,j}(\beta_j)} \right) \tag{4.4}$$

As opposed to P_j in (4.3), the value of $P_{0,j}$ in (4.4) is obtained when port j is enabled but idle. The paper evaluates the joint effects of DTRA (instantiated as power-aware routing) and PTRA (only applied to network links, not entire nodes), concluding that power-aware routing is most beneficial when PTRA is scarcely deployed, as is the case in commercial equipment available today. In the case where PTRA is completely absent but individual links can be turned on and off, the energy savings in one sample topology range between 25% (at 90% of the maximum load) and 50% (at 10% load).

In [30], Rossi et al. evaluate the energy-saving benefits of power-aware routing in an experimental core network where the continuous transit of packets forces individual nodes to remain powered on without interruption. A mixed integer program that handles binary and continuous variables controls the distribution of traffic to the network links, creating the opportunity for switching off unused links and for saving additional energy with PTRA in partially utilized links and nodes. The reference linear model combines results from the literature [40], [41], [42], [43]:

$$S = C_0 + C_b\beta_C + \sum_{j=1}^{N_P} \left(P_{0,j+P_{b,j}(\beta_j)} \right) \tag{4.5}$$

where C_0 is the power consumed by the chassis when idle, C_b is the additional power consumed by the chassis when fully utilized, and β_C is the chassis bit-rate load.

The paper avoids a parametric analysis of the optimal solution by instantiating specific values for every parameter of the linear model. Compared to (4.4), equation (4.5) adds a chassis contribution that quantitatively dominates over the port terms, with substantial impact on the energy saving metrics network-wide. However, it does not include terms for the explicit contributions of individual line cards, which appear instead in the models of (4.1), (4.2), and (4.3). In absence of PTRA, DTRA saves only 0.2% of the overall energy, despite a 34% reduction in the energy consumed by the network links. With ideal PTRA, which scales power linearly with the load in the links and chassis of every node, PTRA alone saves 96% of the total energy, while DTRA only adds an extra 0.1%. One last point to remark is that the model of (4.5) rightly takes into account the full-duplex nature of network links and ports. As a consequence, a network port is fully loaded when its traffic load is 100% in both the input direction (from the network to the port) and the output direction (from the port to the network). Accordingly, the port load variable β_j ranges between 0 and 2.

4.2 SUT

We obtain energy profiles for the following five systems under test (SUT's), manufactured by multiple vendors:

- ES1 Ethernet switch in fixed system configuration with integrated control and switch module (no slots for plug-in cards), twenty-four 1GbE Ethernet ports (SFP), two 10GbE Ethernet ports (SFP), and AC power supply. The switch supports VLAN and MPLS tunneling for E-Line, E-LAN, and VPLS applications [44], [45]

- ES2 Ethernet switch with twenty-four integrated 1GbE ports (RJ-45), four of which are dual-mode ports that also offer the alternative of loading an SFP module, two 10GbE Ethernet ports (SFP), and AC power supply. The aggregate capacity and functional capabilities of ES2 are the same as those of ES1. One important difference that is worth noting is that ES2 has twentyfour integrated 1GbE ports, whereas all 1GbE ports of ES1 are SFP-ready

- IR1 Edge/aggregation router in fixed system configuration with integrated control and switch module, twenty 1GbE Ethernet ports (SFP), six 10GbE ports (SFP), and AC power supply.

- IR2 Aggregation router in fixed system configuration with integrated control and switch module, six 10/100Mbps Ethernet ports (RJ-45), two 1GbE ports (SFP), and DC power supply

- IR3 Aggregation router in modular system configuration with 8-slot chassis. In the IR3 instance available for our experiments, the chassis is populated with one fan card, two control and switch module (CSM) cards, and two 8-port Ethernet adapter cards (EAC's). Each EAC includes six 10/100Mbps Ethernet ports (RJ-45) and two 1GbE ports (SFP). IR3 also works with a DC power supply

Please note that Ethernet (RJ-45) identifies an integrated BASE-TX Ethernet port. Ethernet (SFP) identifies an Ethernet port that accommodates a small form-factor pluggable (SFP) transceiver. The SFP itself can be of different types depending on the type of cable connector that it supports: 1000BASE-LX and 1000BASE-SX SFP's support optics cables, 1000BASE-TX SFP's support copper cables with RJ-45 connectors, and 10GBASE-LW/LR SFP's support 10 Gbps optics cables. 10GBASE-LW/LR modules are commonly referred to as XFP's, but let us call them SFP's for simplicity of notation.

4.3 Testbed

We list the definitions and conventions that we follow in the presentation of our results and describe the equipment that makes up our experimental testbed, highlighting the constraints that it imposes on the execution of the power measurements.

Definition 7. *We refer to an SFP-ready SUT port as loaded if it has an SFP module attached; otherwise we call it an empty port.*

Definition 8. *The* average load *of matrix S is defined as We refer to a loaded port or to an integrated RJ-45 port as connected if a network cable connects the port to a peering interface on the same system or on a traffic generator/sink, and as disconnected otherwise.*

Definition 9. *We refer to a network port as enabled if it is configured for operation at a set rate, and as disabled otherwise.*

In general, a port can be switched between the enabled and disabled states when it is empty, loaded but disconnected, and connected. However, we are only interested in the distinction between the enabled and disabled states in the particular case where the port is connected, because this is the kind of state transition that is controlled by DTRA techniques.

Definition 10. *We refer to a connected port as input-busy if it receives traffic from the attached network cable and as input-idle otherwise. We refer to a connected port as output-busy if it has traffic ready for transmission over the attached network cable and as output-idle otherwise.*

Definition 11. *We refer to an input-busy (output-busy) port as input-saturated (output-saturated) when it receives (transmits) traffic at a rate that approaches its nominal capacity.*

Definition 12. *We refer to an input-busy (output-busy) port as input-overflowing (output-overflowing) when it receives traffic in excess of its capacity.*

4.3.1 Testbed equipment

Figure 4.1 shows a schematic drawing of the laboratory testbed where we execute the power measurement. The testbed includes the items listed in the following subsections.

Power meter station

The power meter station (PM in Fig. 4.1) consists of the power meter proper and of the auxiliary data logging software that runs on a connected laptop. The power meter is an Extech Instruments 380801 true RMS single-phase power analyzer, placed between the power supply (whether AC or DC) and the SUT.

The meter's resolution is 0.1W for readings up to 200W and 1W for readings between 200W and 2 kW. The data logging laptop acquires power samples at 1 s intervals over the serial port of the power meter. We obtain the 48V DC power supply for IR2 and IR3 from a Xantrex Technology XKW 1kW module (DC). We do not include the power consumption of the DC power supply module in our power measurements. With ES1, ES2, and IR1 the power supply path bypasses the DC module.

Traffic endpoints

For the generation and termination of test traffic we use two desktop computers with 1GbE network cards (PC TE in Fig. 4.1) and a Spirent SmartBit SMB-200

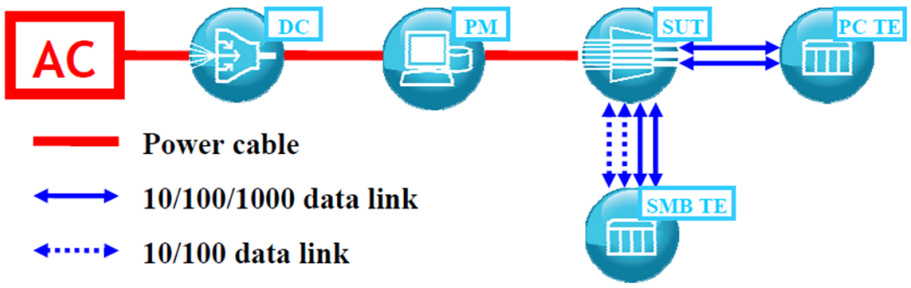

Figure 4.1. Experimental testbed for power measurements.

with 1GbE and 10/100Mbps Ethernet interfaces (SMB TE). Each computer runs an instance of the Linux OS (Ubuntu Release 10.10) and is equipped with a 3.0 GHz dual-core processor, 6 GB DRAM, 1TB hard drive, and one 1GbE network port (RJ-45). We rely on the iperf utility for configuration and operation of the traffic sources and sinks on the two PC's. The Spirent SmartBit SMB- 200 chassis (firmware version 6.7, umbrella SmartBit release 10.51) hosts two SmartMetrics 10/100Mbps Ethernet SmartCards (RJ-45) and two GX-1405B 1000BASE-SX Ethernet SmartCards (optics). We use the two PC's for traffic exchanges with RJ-45 SUT ports at rates up to 1Gbps. We use the 10/100Mbps Ethernet ports of the SMB-200 for exchanges with RJ-45 SUT ports at rates up to 100 Mbps. The two 1GbE ports on the SMB-200 can be used exclusively for exchanges with SUT ports loaded with 1000BASE-SX modules.

Network connectors and cables

We can rely on two 10GBASE-LW/LR SFP modules for loading the 10GbE ports on ES1, ES2, and IR1. For the SFP-ready 1GbE ports on ES1, IR1, IR2, and IR3 we have two 1000BASE-SX SFP modules and twenty-four 1000BASE-TX SFP modules. Notice that, while we have cables for all interfaces, we do not have matching ports on the traffic endpoints for the 10GBASE-LW/ LR modules.

4.4 A new model for energy profiles

In this section we define the linear model that we use for construction of the energy profiles of the SUT's of our testbed. We start by explaining how we isolate the contributions of bitrate and packet-rate loads to the power consumption of a generic system component (chassis, line card, or port). We then illustrate the model that we consider ideal for the construction of a complete, unconstrained energy profile. We provide insight about the reasons for inclusion in the ideal model of the various terms that compose it and evaluate the opportunity of removing some of those terms in the specific context of our energy profiling exercise, due to the characteristics of the SUT's and of the auxiliary equipment in our testbed. We integrate the conclusions of these discussions in the simplified model that we present at the end of the section.

4.4.1 Isolation of traffic contribution

In the data path of a network system we can identify hardware devices whose power consumption depends mostly on the bit rate of the sustained traffic (e.g., transceivers, switch fabric modules) and others for which it depends mostly on the packet rate (e.g., packet processors, traffic managers). A power meter that only

captures fluctuations of the current absorbed by the system cannot directly identify the power consumed by each device, but can detect the effects of varying bit and packet rates on the overall power consumption.

We should therefore include independent terms for the bit rate β and the packet rate ρ in the ideal expression of the power consumed by each controllable system component (chassis, line card, and port). However, since β and ρ are not independent of one another ($\rho = \beta/\sigma$, where σ is the average packet size measured in bits), the term for packet-rate sensitivity cannot be a function of the packet rate ρ.

Instead, the term is a linear function of the average packet size σ, such that the packet rate contribution is null when the average packet size is maximum (i.e., the packet rate is minimum for the given bit rate), and maximum when the packet size is minimum (the packet rate is maximum for the given bit rate). The following equation expresses our first-order approximation X_t^d of the contribution of each traffic direction d to the power consumed by a generic system component x (chassis, line card, or port):

$$X_t^d(\beta, \sigma) = X_b^d \beta_x^d \left(1 + \frac{\sigma_{max,x} - \sigma_x}{\sigma_{max,x} - \sigma_{min,x}}\right) \tag{4.6}$$

In (4.6), X_b^d is the bit-rate sensitivity of component x in direction d (X becomes C when x is the chassis, L when x is a line card and P when x is a port), X_r^d is the *packet-size sensitivity* for the same component and direction, β_r^d is the *sustained bitrateload* ($0 \leq \beta_r^d \leq 1$), $\sigma_{max,x}$ is the maximum size of a data packet in the component (e.g. $\sigma_{max,x} = 1518B$ when x is an Ethernet port), and $\sigma_{min,x}$ is the minimum size ($\sigma_{min,x} = 64B$ when x is an Ethernet port).

4.4.2 The complete linear model

The linear model that we consider ideal for application in rate adaptation contexts is one that captures the power contributions of all system components whose state can be controlled by external action, whether by network signaling, by system management interface, or by physically plugging or unplugging hardware. These system components include the chassis, the line cards (when present), and the network ports with respective accessories (e.g., SFP modules in our set of SUT's). For every component, there should be one term that expresses the fixed cost of keeping it powered on and one that is sensitive to traffic.

The contributions of bit rate and packet rate should be distinguished in parts of the system where the packet size is variable. The following equation synthesizes the above requirements:

$$S = C(\beta_C) + \sum_{i=1}^{N_L} L(\beta_i^{in}, \beta_i^{out}, \sigma_i) + \sum_{i=1}^{N_P} P(\beta_j^{in}, \beta_j^{out}, \sigma_j) \tag{4.7}$$

where

$$C(\beta_C) = C_0 + C_b\beta_C \tag{4.8}$$

is the power consumed by the chassis, inclusive of a fixed term 0 C and a variable term that depends on the aggregate traffic load sustained by the switch fabric;

$$L(\beta_i^{in}, \beta_i^{out}, \sigma_i) = L_{0,i} + L_i^{in}\beta_i^{in}\left(1 + L_{r,i}^{in}q(\sigma)\right) + L_i^{out}\beta_i^{out}\left(1 + L_{r,i}^{out}q(\sigma)\right) \tag{4.9}$$

is the power consumed by line card i , inclusive of a fixed term 0, i L and variable terms that depend on input and output loads;

$$q(\sigma) = \frac{\sigma_{max,x} - \sigma_x}{\sigma_{max,x} - \sigma_{min,x}}$$

is the *packet-size load*, completely independent of the bit-rate loads β_i^{in} and β_i^{out} and

$$P(\beta_i^{in}, \beta_i^{out}, \sigma_i) = P_{0,i} + L_i^{in}\beta_i^{in}\left(1 + P_{r,i}^{in}q(\sigma)\right) + P_i^{out}\beta_i^{out}\left(1 + P_{r,i}^{out}q(\sigma)\right) \tag{4.10}$$

is the power consumed by port j, inclusive of a fixed term $P_{0,j}$ and variable terms that depend on the bit-rate and packet-size loads in the input and output directions of the port.

4.4.3 Discussion of the new model

In this section we illustrate in detail the terms of equations (4.8), (4.9) and (4.10). We refine their definitions where required by the engineering and measurement constraints of our testbed.

Chassis power

It is fair to assume that the chassis power consumption C is sensitive to the traffic load, especially if the switch fabric exhibits some degree of modularity with rate adaptation capabilities within each module. In (4.8) we have no distinct terms for bit-rate and packet-rate contributions because packets typically cross the switch fabric after being segmented into fixed sized data units with either standard or proprietary formats, setting a constant ratio between the two rates. Also there is no distinction between input and output traffic because the amount of packets that enter and exit the central module through the switch fabric interfaces is always the same. The same is not true in individual line cards and network ports, where it is possible to have an unbalance between input and output traffic.

We remark that the type of power meter that we use in our testbed and the absence of rate adaptation capabilities in the switch-fabric hardware of current-generation network systems make the extraction of the bit-rate sensitivity C_b of the

chassis practically impossible. In theory there could be ways to isolate the variable terms in the power contributions of the chassis, line cards, and network ports by arranging the same amount of traffic over different combinations for the numbers of busy line cards and ports. However, the sensitivity to traffic shown by the power consumption in the SUT's of our testbed is so low that it is often masked by the measurement error of the power meter (between 0.05W and 0.5W). We expect the issue to get gradually solved in the future, as rate adaptation becomes more pervasive and critical resources for more accurate measurements become more affordable. For the time being, we consider it acceptable to reduce the chassis power term of (4.8) to the fixed component alone: $C(\beta_C) = C_0$.

Line card power

The line card, when present, is the place where packets that are associated with multiple ports undergo the format conversion from network to switch fabric and vice versa. It is easy to find a qualitative justification for every term that appears in (4.9). The switch fabric adapter is one example of a line card device where the power contribution of the sustained bit rate clearly dominates over its packet-rate counterpart. Packet rate dominance over bit rate can be expected instead in the packet processor.

However, the aggregate nature of the measurements produced by our power meter compromises our ability to discern the traffic-sensitive power contributions of a line card from those of its ports. As a consequence, we decide to concentrate all traffic-sensitive terms at the port level, identifying the line card power with its fixed term: $L(\beta_i^{in}, \beta_i^{out}, \sigma_i) = L_{0,i}$.

Port power

Due to the simplifications of the two previous subsections, the network port remains the only configurable component of the system where we can retain traffic-sensitive contributions to power consumption. Even the port power model is not exempt from trimming. In fact, because of measurement inaccuracies that are induced by the limited availability of traffic endpoints in our testbed, we cannot differentiate between the values of input and output load parameters.

We must resort instead to unified traffic sensitivity parameters $P_{b,j}$ and $P_{r,j}$, and accordingly to unified load variables β_j and $q(\sigma)$ The value of β_j ranges between 0 (when packet traffic is completely absent) and 1 (when port j sustains 100% bit-rate load simultaneously in both directions).

Preliminary measurements on idle systems show us that the fixed power contribution $P_{0,j}$ of a port j must be split into two distinct terms: the *fixed hardware port*

Table 4.1. Fixed port power terms for SFP-ready ports in ES1 (TX and SX ports set at 1Gbps , LW/LR ports at 10 Gbps).

	TX [W]	SX [W]	LW/LR [W]
$P_{0,j}^{(h)}$	0.308	0.5	1.2
$P_{0,j}^{(s)}$	1.091	0.3	1.8

power $P_{0,j}^{(h)}$ and the *fixed software port power* $P_{0,j}^{(s)}$

$$P_j = P_{0,j}^{(h)} + P_{0,j}^{(s)} + P_{b,j}\beta_j(1 + P_{r,j}q(\sigma))$$

The fixed hardware port power captures the power contribution of port j when it is loaded with an SFP, whether or not the port is enabled for operation. The term $P_{0,j}^{(h)}$ obviously disappears in the case of integrated BASE-TX ports. The isolation of $P_{0,j}^{(h)}$ is important because it offers the network operator the option to save energy by unplugging the SFP's of ports that remain disabled for extended periods of time. It also offers system vendors an incentive to add to their designs provisions for controlling this power contribution (and the associated energy waste in the case of disabled ports) via software.

The fixed software port power $P_{0,j}^{(s)}$ is the added contribution of a port that is enabled for operation, before it starts handling traffic. Table 4.1 lists values for the two terms measured on ES1 with BASE-TX and BASE-SX SFP's (configured at 1Gbps), and with BASE-LW/ LR SFP's (set at 10 Gbps).

The switching of individual ports between the enabled and disabled states is one of the primary knobs that DTRA techniques have available for saving energy. Setting the operating rate of an enabled port to a maximum of 10 Mbps, 100 Mbps, or 1Gbps (and 10 Gbps in the case of 10GbE ports) is another dimension of dynamic configuration that DTRA techniques can explore, because each rate generally presents a different value of $P_{0,j}^{(s)}$. In the example of Table 4.1, the measured values of $P_{0,j}^{(s)}$ for a BASE-TX SFP are 0.238W, 0.338W, and 1.091W when the configured rate of operation is 10 Mbps, 100 Mbps, and 1Gbps.

Simplified linear model

The following equation synthesizes the linear model that results from the simplifications of the previous subsections:

$$S = C_0 + \sum_{i=1}^{N_L} L_{0,1} + \sum_{j=1}^{N_P} P_{0,j}^{(h)} + P_{0,j}^{(s)} + P_{b,j}\beta_j(1 + P_{r,j}q(\sigma)) \tag{4.11}$$

We emphasize that the model of Eq. (4.11) derives entirely from simplifications of the model laid out in equations (4.7), (4.8), (4.9) and (4.10). As the engineering and measurement limitations that warrant the simplifications fade out over time, we expect all the terms of the complete model to gradually reappear in Eq. (4.11).

4.5 System with DC power supply

The routers IR2 and IR3 of our testbed receive power from a dedicated DC module that also dissipates its own power D . We decide to exclude D from the estimation of the linear parameters of IR2 and IR3 because the values of the parameters should be intrinsic of the two systems and independent of the specific DC module used in the measurements.

This is particularly true with the DC module of our testbed (Xantrex Technology XKW 1kW), which operates out of its high efficiency region when it supplies less than 70W to a single load, as in most of our experiments with IR2 and IR3. If we factored D in our measurements, we would likely overestimate the values of the linear parameters compared to rack-based multi-load applications where the efficiency of the DC module is much higher. To guarantee that the power contribution of the DC module is not ignored entirely, we include it when we estimate the total power that IR2 and IR3 consume under given system and traffic configurations. However, instead of adding the measured power consumption of our DC module, we add the nominal power consumption of a generic DC module with 90% efficiency [46]. The total power consumed by the SUT and the DC module is therefore $A = S + D = 1.11S$, where S is defined in Eq. (4.11).

4.6 Measurement methodology

Our power measurement experiments aim at obtaining accurate estimates for all the parameters in Eq. (4.11): C_0, L_0, $P_0^{(h)}$, $P_0^{(s)}$, $P_{b,j}$ and P_r.

We omit the line card and port indices in the symbols of the profile parameters unless we refer to specific instances of those system components.

To optimize the accuracy of the estimates in spite of the relatively coarse resolution of the power meter (0.1W) we must maximize in each experiment the number of target system components that operate in identical conditions, compatibly with the equipment availability limitations of our testbed. The limitations are particularly important when we measure the port parameters, especially those that require the presence of traffic (the bit-rate sensitivity P_b and the packet-size sensitivity P_r), because we can only rely on three pairs of traffic generator ports, each pair being of a different kind (two 1000BASE-TX ports on the PC's, two 10/100BASE-TX ports

on the SMB-200, and two 1000BASE-SX ports also on the SMB-200). Additional measurement constraints are imposed by the limited availability of BASE-SX and BASE-LW/LR SFP's (only two per type), and by the lack of BASE-LW/LR traffic generator ports.

Based on the above constraints, all measurements with BASE-TX SFP's involve up to twenty-four ports, whereas the measurements with BASE-SX and BASE-LW/RW SFP's never involve more than two ports. In BASE-TX and BASE-SX measurements with packet traffic the system is fed by traffic generator ports of the same kind. In BASE-LW/LR measurements, the traffic arrives from BASE-TX or BASE-SX ports, which also contribute to the power measurements: we must subtract the estimated contribution of the traffic generator ports from the total system power before we can estimate the power consumed by the target ports. Only two traffic generator ports are available for measurements that involve BASE-TX integrated ports and BASE-TX SFP's. To ensure that every BASE-TX port in the system actually contributes to the measured power, we join with loop-back cables all pairs that we can form with ports that are not directly attached to a generator. Within each loop-back pair, one port transmits traffic to the loop-back cable and the other port receives it. We enable the spanning tree protocol (STP) to prevent the volume of the injected broadcast traffic from exploding. With STP enabled, each receiving port internally forwards traffic to only one transmitting port. Since every port handles saturation-level traffic in one direction and no traffic in the other, we use $\beta = 0.5$ to derive the value of P_b from the power meter readings.

For traffic measurements with the 10GbE SFP's, instead, we keep STP disabled: the replication of broadcast packets that occurs at each receiving port is strictly necessary for expansion of the traffic volume from the 1Gbps rate supplied by the two generators to the 10 Gbps rate that each 10GBASE-LW/RW can sustain. Since every port that is enabled receives and transmits traffic at full capacity, we use $\beta = 1.0$ in the estimation of the bit-rate sensitivity.

4.7 Experimental results

In this section we present results from our experiments. We focus on data that gauge the compatibility of existing equipment with DTRA techniques and underscore the need for PTRA support in future system designs. Table 4.2 lists for each SUT the sum of the port capacities (possibly larger than the actual switching capacity) and the estimated values for five of the six parameters that make up the linear model of (11), in the specific case where the SUT is loaded with BASE-TX ports enabled for operation at 1Gbps. The missing parameter is the (port) packet-size sensitivity, whose values are practically impossible to distinguish from zero in the system configurations used in the experiments of Table 4.2. We observe non-negligible values

Table 4.2. Parameters of linear model (1GbE BASE-TX ports configured for operation at 1Gbps)

SUT	Chassis, idle $C_0[W]$	Line card, idle $L_0[W]$	Port, fixed hardware $P_0^{(h)}[W]$	Port, fixed software $P_0^{(s)}[W]$	Port, bit-rate sensitivity $P_b[W]$
ES1 44*Gbps*	32.4	N/A	0.3	1.1	0.3
ES2 44*Gbps*	35.0	N/A	N/A	1.0	0.1
IR1 80*Gbps*	216	N/A	0.2	1.0	1.1
IR2 2.6*Gbps*	40.4	N/A	0.2	1.0	0.8
IR3 15.6*Gbps*	54.8	14.5	0.2	1.0	0.8

Table 4.3. Port parameters (10/100BASE-TX ports in IR2 and IR3 configured for operation at 100 Mbps)

SUT	Port, fixed software $P_0^{(s)}[W]$	Port, bit-rate sensitivity $P_b[W]$	Port, packet-size sensitivity $P_r[W]$
IR1	0.3	0.1	10
IR2	0.3	0.1	9

of the parameter only in the case of integrated 10/100BASE-TX ports in IR2 and IR3 (see Table 4.3 for the values measured with ports configured at 100 Mbps).

Table 4.4 lists the parameters of 1GbE ports loaded with BASE-SX SFP's (ES2 is missing because its 1GbE ports are integrated, IR1 because the SMB-200 traffic

Table 4.4. Port parameters (1GbE BASE-SX ports configured for operation at 1Gbps)

SUT	Port, fixed hardware $P_0^{(h)}[W]$	Port, fixed software $P_0^{(s)}[W]$	Port, bit-rate sensitivity $P_b[W]$
ES1	0.5	0.3	0.3
IR2	0.5	0.1	0.8
IR3	0.5	0.2	0.7

Table 4.5. Port parameters (10GbE BASE-LR/LW ports configured for operation at 10 Gbps)

SUT	Port, fixed hardware $P_0^{(h)}[W]$	Port, fixed software $P_0^{(s)}[W]$	Port, bit-rate sensitivity $P_b[W]$
ES1	1.2	1.8	1.6
ES2	0.9	2.0	0.5
IR3	0.2	1.0	2.9

generator could not be moved to the facility where our instance of the system was located). Table 4.5 provides the same information for 10GbE ports loaded with BASE-LR/LW SFP's (10GbE ports are only available in ES1, ES2, and IR1). The results in Table 4.2 indicate that the fixed software port power $P_0^{(s)}$ is by far the dominant port power term in the two Ethernet switches. The traffic-sensitive terms gain relevance in the IP routers, consistently with the increased variety and intensity of the packet processing functions in those systems. Table 4.5 shows similar trends for the 10GbE ports, although the traffic-sensitive terms are generally heavier than with 1GbE ports. We note in Tables 4.2 and 4.4 the quantitative inversion between fixed hardware power $P_0^{(h)}$ and fixed software power $P_0^{(s)}$ when we replace BASE-TX SFP's with BASE-SX SFP's in the 1GbE ports of ES1, IR2, and IR3. Table 4.2 also shows that the idle chassis power is much higher in IR1 than in all other SUT's. This is because IR1 is the only system in the set that combines high aggregate switching capacity (80 Gbps) with the complex packet processing functions of a router in a non-modular architecture.

We define the margin for saving energy with DTRA techniques as the entire portion of the total energy consumption of a system that is associated with components that DTRA can control. This is clearly a hard upper bound on the amount of energy that DTRA can save. Network topology and traffic demands determine the tightness of the bound in practical applications. To quantify the DTRA margin, we must look at the relative weights of the line-card and port terms within the overall power consumption of each system. Figure 4.2 shows the breakdown of the total power consumption for the five SUT's when all ports are enabled and fully loaded. In ES1, ES2, and IR1 we configure the 1GbE ports as in Table 4.2 (BASE-TX at 1Gbps) and the 10GbE ports as in Table 4.5. In IR2 and IR3 the configurations are those of Table 4.2 (BASE-TX SFP's at 1Gbps) and Table 4.3 (integrated BASE-TX at 100 Mbps). We normalize the power levels in each column to the total power consumption of the respective system. We obtain the contributions of the port power terms by multiplying the maximum number of ports configurable for each type by

the respective per-port values. With IR2 and IR3 the sum of the port capacities in this maximum configuration exceeds by far the actual switching capacity of the system, with the effect of producing overestimated values for the traffic sensitive power contributions.

We would obtain more accurate estimates if we could rely on a larger number of traffic generator ports to pair with the system ports in the power measurement experiments (up to 48 ports with IR3). Still, even if overestimated and maximized by the assumption of minimum-length Ethernet frames (64 B), the traffic-sensitive shares of the total power remain marginal in IR2 and IR3, causing no qualitative impact on the interpretation of the results. In Figure 4.2, the traffic-sensitive terms range between 5% and 21% across the five systems. If we also consider that the two highest values, in IR2 and IR3, are certainly overestimated, the maximum traffic power share is likely well below 15%. We can comfortably conclude that current designs are far from exhibiting the type of rate-proportional power consumption behavior that rate adaptation techniques aim at establishing at the system level. While the indication is disappointing in terms of overall energy efficiency, in light of the results in [29] and [30] it signals that DTRA techniques have a clear window of opportunity in the short term for bringing along important energy savings through relatively simple signaling extensions and software modifications applied to existing hardware platforms.

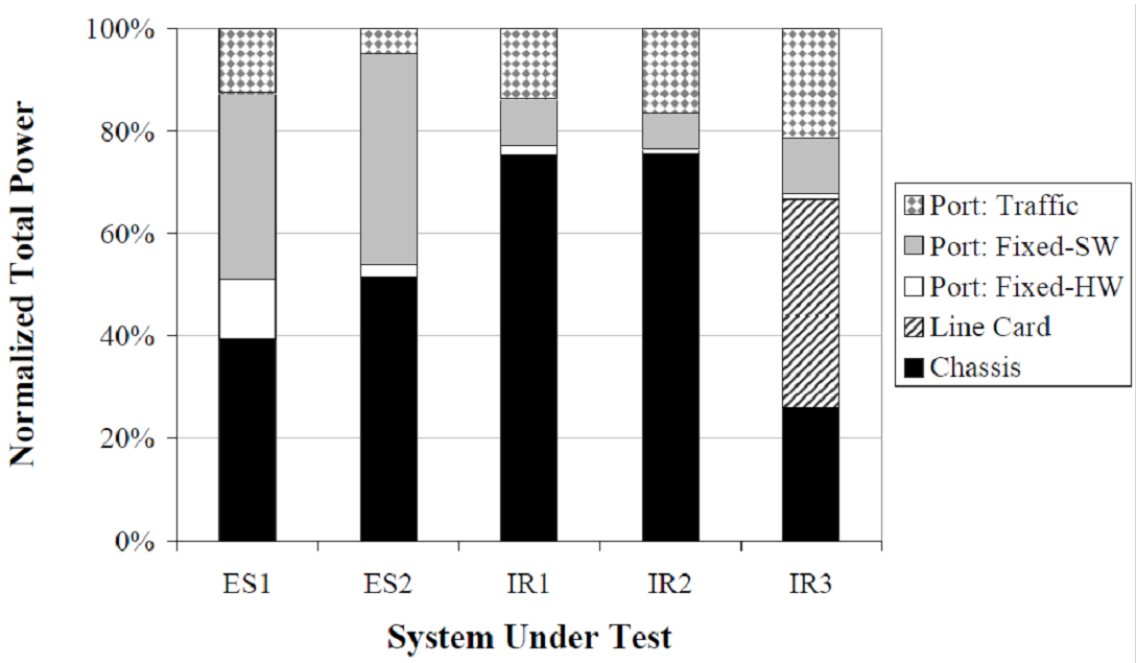

Figure 4.2. Estimated breakdown of system power when all ports in the system are fully loaded.

If we compute the DTRA margin as the sum of the fixed-software and traffic-sensitive port power terms, that is without including the fixed-hardware port power and the line card power, we see that it ranges between 46% and 49% in the two Ethernet switches and between clearly lower values (23% and 32%) in the three IP routers. The potential for DTRA savings increases substantially in modular systems where individual line cards can be switched on and off (+41% in IR3), and even more if DTRA can control the operating state of the entire chassis. However, in network applications that are not necessarily unusual, such as those addressed in [30], it may be likely that an entire system, or even just individual line cards, can never be switched off. To ensure that the energy savings remain consistently large irrespective of the network topology and application, PTRA capabilities must be pervasively deployed in future generations of hardware platforms. Design challenges and performance properties are well understood for PTRA techniques in linear data-path devices with one input and one output [33]. The same is not true for devices with multiple inputs and outputs like the switch fabric, which typically resides in the system chassis.

The challenge for those devices is to achieve direct proportionality between power consumption and aggregate switching throughput irrespective of the traffic load distribution across interfaces. Since the chassis contribution to the total power is always large (between 39% and 76% in the SUT's of our testbed), future research efforts should direct their aim at the identification of viable PTRA solutions for multi-interface devices.

Chapter 5

Conclusions

The motivations of this PhD thesis are based on the claim that the power consumption related to network elements has became a relevant issue, in particular looking the side of scalability and high performances. As practical example it is enough to think about high speed routers and how much the aggregate bandwidth is growing fast due to the increasing of the traffic demand. To achieve this target the price to pay is in term of power consumption.

In particular we have considered an $N \times N$ input-queued switch with a crossbar-based switching fabric implemented on a single chip. Thus, at increasing bit rate, power dissipation is becoming more and more challenging, limiting the crossbar scalability for high performance switches. The time is slotted and, in each timeslot, a centralized scheduler determines a switching fabric configuration to transfer packets.

As a first step of we have considered an optical switching fabric and we focused on the energy consumption needed to change its configuration, assuming that the energy is proportional to the number of changes between consecutive timeslots. We have addressed the computation of a minimum-energy frame to schedule a set of packets in a request matrix. We propose a family of algorithms that decouple the computation into two different phases: matching selection and frame sorting.

Throughout a theoretical and simulation study, we were able to investigate the throughput-energy tradeoff achieved by the different algorithms. We observed that the energy consumption per packet may vary by almost two order of magnitude depending on the algorithm and traffic matrix.

We showed that one specific algorithm achieves the best tradeoff between energy, throughput and complexity. In particular, we propose the GExa-NS algorithm, that provides always the best compromise between sustainable load (very close to the maximum) and energy consumption (very close to the minimum possible). Furthermore, its complexity is much lower that the other similar algorithms (GMax-BS and BvN-BS), since the matching selection induces already an energy-efficient ordering

among the matchings. The BvN-BS algorithm, even if optimal algorithm for energy-oblivious frame decomposition, is the least efficient in terms of energy consumption and furthermore it requires a high computational complexity.

After this first analysis we focused on the electrical crossbar switching fabric. The power consumption produced by the crossbar chip and due to the data transfer grows as NR^3, where R is the maximum bit rate. We discussed the potential power gains exploiting Dynamic Voltage and Frequency Scaling (DVFS) techniques to control packet transmissions through each crosspoint of a crossbar used as the switching fabric in an input-queued switch. We took an idealized approach, disregarding the details related to packet scheduling, looking at flow rates.

Our power control operates independently of the packet scheduler and exploits the knowledge of a traffic matrix obtained by on-line measurements. We proposed a family of control algorithms to reduce the power consumption. The proposed algorithms are computationally simple and obtain performance gain close to those of more complex, optimal algorithms and they are particularly efficient in non-overloaded conditions.

The actual potential of the proposed approach is bore out by performance results. They were validated through a real hardware synthesis, a real design case synthesized on a 90 nm CMOS technology, that show that a significant power reduction can be obtained, especially at low loads.

Note that this approach to the problem is compatible with complementary policies that minimize the other power components.

In the last part of the thesis, it has been focused on the energy characterization of existing network elements available for commerce. The definition of accurate energy profiles is a critical step in the process of planning for the short-term (i.e. software) and long-term (i.e. hardware) design upgrades that can enable better energy efficiency in future generations of network systems.

Learning from the limitations of existing models for energy profiling, we have introduced a new linear model that suits well the requirements for supporting new frameworks with capabilities of rate adaptation technologies at multiple timescales. These technologies aim at establishing a direct, possibly linear, relationship between the power consumption of a packet network and the traffic load that the network sustains. As a consequence, the deployment of accurate energy profiles for sample commercial equipment represents a first step in this direction.

A set of extensive power measurement experiments was run to compute the energy profiles of five network systems from multiple vendors, namely Ethernet switches and IP routers for enterprise and access applications. The first result found was that existing linear models for mapping system and traffic configurations onto power consumption levels are not adequate to drive effectively the operation of rate adaptation frameworks, as a consequence we enhanced those models with essential revisions.

This model indicates that a "current generation" network equipment requires straightforward signaling extensions and system management software upgrades to achieve appreciable energy savings thanks to the deployment of demand-timescale rate adaptation techniques that remotely control the operating state of individual system components (ports, line cards, and chassis) based on network-wide power minimization goals. The reason is that our model supplies information at the right granularity needed for control of the system components that tangibly contribute to the power consumption of individual systems and entire networks.

The model also shows that energy savings at a much larger scale can only be attained with a new generation of hardware platforms for network systems, through a pervasive deployment of packet timescale rate adaptation techniques. The future generations must support low-power sleep states for unutilized system components and packet-timescale rate adaptation methods in order to establish true proportionality between energy consumption and traffic workload.

Future research efforts should be particularly directed at the identification of viable PTRA techniques for data-path hardware components with multiple input and output interfaces. We have applied the model to the results of power measurement experiments conducted on five commercial network systems using a testbed with limited auxiliary resources, learning that the availability of a large number of traffic endpoints is instrumental to the accuracy of estimation of the traffic-sensitive terms of the model. We plan to keep running experiments on new network gear while expanding the traffic generation capabilities of the adopted testbed.

Bibliography

[1] Industrial Technologies Program., "Routing Telecom and Data Centers toward Efficient Energy Use," May 2009.

[2] J.A. Laitner, K. Ehrhardt-Martinez, "Information and Communication Technologies: The Power of Productivity," *Environmental Quality Management*, 2009, Part I: 18(2):4766, Part II: 18(3):1935.

[3] Fettweis G.P.and Zimmermann E., "ICT energy consumption - trends and challenges," in *Proceedings of the 11th International Symposium on Wireless Personal Multimedia Communications,* Lapland, Finland, September 2008.

[4] R. S. Tucker, "Green optical communicationspart ii: Energy limitations in networks," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. PP, no. 99, pp. 1 14, 2010.

[5] M. C. Wu, O. Solgaard, and J. E. Ford, "Optical mems for lightwave communication," *Lightwave Technology, Journal of*, vol. 24, no. 12, pp. 44334454, Dec. 2006.

[6] C. Chen et al., "Efficient fpgas using nanoelectromechanical relays," in *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays (FPGA10)*, New York, NY, USA, 2010, pp. 273282.

[7] T. Weller and B. Hajek, "Scheduling nonuniform traffic in a packet-switching system with small propagation delay," *Networking, IEEE/ACM Transactions on*, vol. 5, no. 6, pp. 813823, 1997.

[8] H. Attiya, D. Hay, and I. Keslassy, "Packet-mode emulation of output-queued switches," in *SPAA*, 2006, pp. 138147.

[9] V. V. Vazirani, *Approximation Algorithms.* Springer, March 2004.

[10] D. Aldous, *Probability Approximations Via The Poisson Clumping Heuristic.* Springer, 1989.

[11] S. Kotz and S. Nadarajah, *Extreme value distributions: theory and applications.* Imperial College Press, 2000.

[12] M. Neely, E. Modiano, and Y.-S. Cheng, "Logarithmic delay for N×N packet switches under the crossbar constraint," *Networking, IEEE/ACM Transactions on*, vol. 15, no. 3, pp. 657 668, Jun. 2007.

[13] B. Towles and W. Dally, "Guaranteed scheduling for switches with configuration

overhead," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 5, pp. 835847, Oct. 2003.

[14] L. Mastroleon, D. ONeill, B. Yolken, and N. Bambos, "Power aware management of packet switches," *High-Performance Interconnects, 2007. HOTI 2007.* 15th Annual IEEE Symposium on, pp. 4753, Aug. 2007.

[15] B. Yolken and N. Bambos, "Power management of packet switches via differentiated delay targets," *Communications, 2008. ICC 08. IEEE International Conference on*, pp. 354359, May 2008.

[16] B. Yolken, D. Tsamis, and N. Bambos, "Target-based power control for queueing systems with applications to packet switches," *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 16, Dec. 2008.

[17] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Transactions on VLSI Systems*, vol. 13, no. 11, pp. 1239–1252, Nov. 2005.

[18] F. Hameed, M. Faruque, and J. Henkel, "Dynamic thermal management in 3d multi-core architecture through run-time adaptation," in *IEEE Design, Automation & Test in Europe (DATE)*, 2011.

[19] `https://research.sprintlabs.com/packstat/packetoverview.php`.

[20] M. Flynn and P. Hung, "Microprocessor design issues: thoughts on the road ahead," *IEEE Micro*, vol. 25, no. 3, pp. 16–31, May 2005.

[21] T. Kolpe, A. Zhai, and S. Sapatnekar, "Enabling improved power management in multicore processors through clustered dvfs," in *IEEE Design, Automation & Test in Europe (DATE)*, 2011.

[22] H. J. Chao and B. Liu, *High Performance Switches and Routers.* Wiley-IEEE Press, 2007.

[23] C.-S. Chang, W.-J. Chen, and H.-Y. Huang, "Birkhoff-von neumann input buffered crossbar switches," in *IEEE INFOCOM*, vol. 3, March 2000, pp. 1614–1623.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[25] Ting Wu, Chi-Ying Tsui, and Mounir Hamdi, "A 2Gb/s 256 x 256 CMOS crossbar switch fabric core design using pipelined MUX," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Phoenix-Scottsdale, AZ, May 2002.

[26] Bolla, R., Bruschi, R., Lombardo, C., and Suino, D., "Evaluating the energy awareness of future Internet devices, " *In Proceedings of 2011 IEEE Conference on High Performance Switching and Routing*. Cartagena, Spain, July 2011.

[27] Barroso, L.A. and Holze, U., "The case for energy proportional computing," *IEEE Computer* 40, 12 (December 2007), 3337.

[28] Mahadevan, P., Sharma, P., Banerjee, S., and Ranganathan,P. "Energy aware network operations," *In Proceedings of 12th IEEE Global Internet Symposium*,

Rio de Janeiro, Brazil, April 2009.

[29] Antonakopoulos, S., Fortune, S., and Zhang, L., "Power-aware routing with rate-adaptive network elements", *In Proceedings of 3rd International Workshop on Green Communications*, Miami, FL, December 2010.

[30] Rossi, D., Bianzino, A.P., Rougier, J.-L., Chaudet, C., and Larroca, F., "Energy-aware routing: A reality check, " *In Proceedings of 3rd International Workshop on Green Communications* Miami, FL, December 2010.

[31] Benini, L., Siegel, P., and De Micheli, G., "Saving power by synthesizing gated clocks for sequential circuits," *IEEE Design and Test of Computers* 11, 4 (1994), 32-41.

[32] Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., and Wetherall, D., "Reducing network energy consumption via sleeping and rate adaptation," *In Proceedings of 5th USENIX Symposium on Networked Systems Design and Implementation,* San Francisco, CA, 2008, 323336.

[33] Francini, A. and Stiliadis, D., "Performance bounds of rate-adaptation schemes for energy-efficient routers," *In Proceedings of 2010 IEEE Workshop on High-Performance Switching and Routing,* Dallas, TX, July 2010.

[34] Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., and Wright, S., "Power awareness in network design and routing," *In Proceedings of IEEE INFOCOM 2008,* Phoenix, AZ, April 2008.

[35] Mahadevan, P., Sharma, P., Banerjee, S., and Ranganathan, P., "A power benchmarking framework for network devices," *In Proceedings of the 8th International IFIP-TC6 Networking Conference,* Aachen, Germany, May 2009.

[36] Tamm, O., Hermsmeyer, C., and Rush, A.M., "Eco-sustainable system and network architectures for future transport networks," *Bell Labs Technical Journal* 14, 4 February 2010, 311328.

[37] ECR Initiative, "Network and Telecom EquipmentEnergy and Performance Assessment, Draft 3.0.1," December 2010.

[38] Alliance for Telecommunications Industry Solutions (ATIS), "Energy Efficiency for Telecommunication Equipment: Methodology for Measurement and Reporting - General Requirements," *ATIS-0600015.2009,* February 2009.

[39] IEEE P802.3az Energy Efficient Ethernet Task Force, "Media Access Control Parameters, Physical Layers, and Management Parameters for Energy-Efficient Ethernet," *IEEE 802.3az Standard Amendment*, October 2010. Available: http://standards.ieee.org/getieee802/download/802.3az-2010.pdf

[40] Gunaratne, C., Christensen, K., and Nordman, B., "Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections and scaling of link speed," *International Journal of Network Management,* 15, 5, September 2005, 297310.

[41] Gunaratne, C., Christensen, K., and Suen, S.W., "Ethernet adaptive link rate (ALR): Analysis of a buffer threshold policy," *In Proc. of IEEE GLOBECOM*

2006, San Francisco, CA, November 2006.

[42] Tucker, R., Baliga, J., Ayre, R., Hinton, K., and Sorin, W.,"Energy consumption in IP networks. In Proc. of ECOC 2008," Brussels, Belgium, September 2008.

[43] Hays, R., Wertheimer, A., and Mann, E., "Active/Idle Toggling with Low-Power Idle," *IEEE 802.3az Task Force Group Meeting,* January 2008.

[44] Metro Ethernet Forum., "Metro Ethernet Services Definitions Phase 2," *Technical Specification MEF 6.1,* April 2008.

[45] Lasserre, M. and Kompella, V., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling," *IETF RFC 4762,* January 2007.

[46] Rasmussen, N. and Spitaels, J., "A Quantitative Comparison of High Efficiency AC vs. DC Power Distribution for Data Centers," *Schneider Electric white paper,* WP-127 v2, April 2011.