Thermal Control for Crossbar-based Input-Queued Switches

Andrea Bianco, Paolo Giaccone, Guido Masera, Marco Ricca
Dipartimento di Elettronica, Politecnico di Torino, Italy

Abstract—We consider an $N \times N$ input-queued switch based on a crossbar switching fabric implemented on a single chip. The thermal power produced by the crossbar chip grows as $NR^2$, where $R$ is the bit rate. Power dissipation is becoming more and more challenging, limiting the crossbar scalability for high performance switches.

We propose to exploit Dynamic Voltage and Frequency Scaling (DVFS) techniques, quite commonly used in integrated circuit design, to control packet transmissions through each crosspoint of the switching fabric. Our thermal control operates independently of the packet scheduler and it is based on short-term traffic measurements. We propose a family of control algorithms to reduce the thermal power dissipation in non-overloaded conditions.

Index Terms—Input queued switch, energy, thermal control.

I. INTRODUCTION

The aggregate bandwidth of high speed routers is growing fast, due to the increased traffic demand in the Internet. Usually, one or few switching fabrics are present in the core of the routers to switch all the data from the inputs to the outputs; each fabric is often implemented on a single integrated circuit. The hardware design of such fabrics is becoming more and more critical, because of the large pin count and the high bit rate. Indeed, if $f$ is the digital signal frequency, the power consumption of a single CMOS is proportional to $f^3$ [1].

Now in a $N \times N$ single-chip crossbar with $N^2$ crosspoint, each implemented through a combinatorial logic, we have $\theta(N^2)$ CMOSs (i.e., a fixed number for each crosspoint), and the total power consumption becomes proportional to $R^2N$, where $R$ is the bit rate and $N$ is the maximum number of data simultaneously flowing across the switching fabric.

Thermal dissipation is becoming a critical design issue, due to high integration level on a single chip, that implies very high power spatial density [2]. In integrated circuits, Dynamic Voltage and Frequency Scaling (DVFS) [1] is a classical technique used to control the power consumption.

DVFS is based on the idea of jointly varying the power supply voltage and the peak signal frequency. A vast literature on DVFS techniques is available, and many scenarios have been considered, such as focusing on a single CMOS, on a CMOS cascade, and on a complete CPU.

In this paper we propose to use DVFS for the thermal control of a single-chip crossbar, analyzing the tradeoff between throughput (i.e., performance) and power consumption (i.e., thermal energy to dissipate). The main idea is to exploit low load traffic conditions to extend packet duration by reducing bit voltage and frequency to control energy dissipation. Note that networks are typically provisioned for worst-case or peak-hour traffic. However, several measurements (see for example [3]) show that backbone utilization rarely exceeds 30%, thus suggesting that exploiting low traffic conditions can be a significant asset to reduce energy dissipation.

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

The paper is organized as follows. The system model is defined in Sec. II. Sec. III formalizes the optimal thermal control problem, describes its properties, and proposes a set of algorithms to solve it. Performance results in Sec. IV show the possible thermal gain of our approach.

The main contributions of the paper are the following:

- definition of the thermal control problem for the crossbar;
- definition of the optimal algorithm and of simpler approximated algorithms;
- performance evaluation of such algorithms.

II. PROBLEM DEFINITION

In a crossbar, the hardware combinatorial logic of a single crosspoint is basically an AND gate between the data signal and the control signal. We start by considering a single CMOS on which the combinatorial logic is based. Then, we examine the complete switching architecture to define the crossbar thermal control problem.

A. 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 a 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. Indeed, leakage currents are in general negligible or can be made negligible with a proper hardware design, whose discussion is out of the scope of this paper. The energy due to a bit transition (i.e., the switching activity) is a quadratic function of $V$ according to the well known formula $E_{\text{bit}} = 0.5CV^2$, where $C$ is the load capacitance. If we consider a 0-1 square wave signal with frequency $f$, the power consumption is $P = E_{\text{bit}}f \propto fV^2$; this value represents also the thermal energy to dissipate.
maximum allowed frequency is
\[ f_{\text{max}} \propto V \]  
(1)
due to the delay needed to switch from one logic state to another [4]. As a consequence, 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 current processing load [5].

We consider a CMOS device operating at a voltage \( V \) between a minimum \( V_{\text{min}} \) and maximum \( V_{\text{max}} \). In addition, within this range, we assume that bit transmission can occur at intermediate voltage levels. When operating at \( V < V_{\text{max}} \), thanks to (1), the signal peak frequency can be slowed down by a factor \( \alpha = V_{\text{max}}/V \) with respect to the maximum frequency allowed when using \( V_{\text{max}} \). Thus, \( \alpha \) represents an expansion factor of the bit duration with respect to the bit duration when using \( V_{\text{max}} \). Furthermore, \( V \) should be larger than \( V_{\text{min}} > 0 \), because of technological constraints that forbid to reduce too much the voltage level and of the impact of leakage currents, that would become not negligible anymore. Define \( \beta = V_{\text{min}}/V_{\text{max}} \). 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 [1]. By construction, \( 1 \leq \alpha \leq 1/\beta \).

B. Switching architecture

We consider an \( N \times N \) input queued (IQ) switch, with virtual output queueing (VOQ), i.e. one queue VOQ\(_{ij} \) for each input \( i \) and output \( j \) pair. This IQ architecture allows high scalability in terms of line rate and number of ports, and the VOQ scheme is theoretically optimal from the performance point of view. To avoid dealing with data content at this abstract level, we assume that a data packet of length \( P \) is transmitted using \( P \) signal transitions: i.e., each packet is composed by a sequence of alternating 0 and 1. The maximum line rate for each port is \( R_{\text{max}} \), measured in [bit/s]; this value can be only reached for \( V = V_{\text{max}} \). The switching fabric is a \( N \times N \) crossbar, with \( N^2 \) crosspoints and \( \theta(N^2) \) CMOSs. The crosspoint from input \( i \) to output \( j \) is denoted as XP\(_{ij} \).

III. CROSSBAR THERMAL CONTROL

In real switch implementation, a packet scheduler is responsible of selecting the set of packets to transfer simultaneously through the crossbar, satisfying the constraints that at most one packet is sent from each input and to each output. 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 50ns. Given such a strict timing constraint, packet schedulers are implemented directly in hardware. A large literature is available on the design of low complexity and high performance packet schedulers for input queued switches [6]–[8].

Differently from the packet scheduler, the thermal control operates at a larger time scale, related to the milliseconds thermal constants of the materials employed to build the chip. We propose a thermal control scheme decoupled from the packet scheduling decision, whose aim is to exploit DVFS at crosspoints to reduce the crossbar thermal power. Based on traffic measurements on the millisecond scale, the control sets the DVFS factor \( \alpha_{ij} \) for the combinatorial logic present at XP\(_{ij} \); each crosspoint is controlled independently. Note that, differently from the packet scheduler, the algorithm for thermal control can be implemented in software, due to the relaxed timing constraints.

Let \( \alpha = [\alpha_{ij}] \) be the \( N \times N \) matrix with all the DVFS factors. Note that setting \( \alpha_{ij} > 1 \) implies that the forwarding rate at XP\(_{ij} \) is reduced and the packet transmission time is increased by a factor \( \alpha_{ij} \). This has two main consequences: i) an additional queueing delay in VOQ\(_{ij} \), ii) the packet scheduler cannot serve any new packet from input \( i \) and to output \( j \) until XP\(_{ij} \) ends the packet transmission. This means that the packet scheduler should take into account thermal control expansion factor in packet scheduling. We disregard this issue in the paper, and we take an ideal fluid-based approach, looking only at rates of I/O flows, to understand which are the asymptotic potential benefits that can be obtained in terms of reduced power consumption.

A. Problem definition

The traffic load on each link is measured on a time window which duration is in the order of thermal constants (ms). Let \( r_{ij} \) be the average rate [bit/s] for the traffic flows enqueued at VOQ\(_{ij} \), and \( R = [r_{ij}] \) the corresponding \( N \times N \) traffic matrix. Let \( S = [s_{ij}] \) be the normalized traffic matrix obtained by setting \( s_{ij} = r_{ij}/R_{\text{max}} \), 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
\[ \rho_{\text{ave}}(S) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ij} \]

Definition 2: The load at input \( i \) and at output \( j \) is \( \rho_i^I(S) = \sum_{k=1}^{N} s_{ik} \) and \( \rho_j^O(S) = \sum_{k=1}^{N} s_{kj} \) respectively.

Definition 3: The maximum load of matrix \( S \) is defined as
\[ \rho_{\text{max}}(S) = \max \{ \max_i \{ \rho_i^I \}, \max_j \{ \rho_j^O \} \} \]
**Definition 4:** The traffic matrix \( S \) is said to be admissible iff \( \rho_{\text{max}}(S) \leq 1 \).

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

We now model the constraints related to the maximum time expansion allowed for the transmitted bits. Consider a generic time period \( T \), for which a flow rate is equal to \( r_{ij} \). The total duration of the bit transmissions is \( T_1 = r_{ij}T/r_{\text{max}} \) and the maximum bit expansion factor is \( T/T_1 = r_{\text{max}}/r_{ij} \), i.e. \( \alpha_{ij}r_{ij} \leq r_{\text{max}} \). At the same time, we have stricter constraints on the expansion factor imposed by the traffic load in \( T \) on input and output traffic relations:

\[
\sum_{i=1}^{N} \alpha_{ik}s_{ik} \leq r_{\text{max}} \quad \sum_{j=1}^{N} \alpha_{kj}s_{kj} \leq r_{\text{max}} \quad \forall k \in \{1, \ldots, N\}
\]

which can be normalized as

\[
\sum_{i=1}^{N} \alpha_{ik}s_{ik} \leq 1 \quad \sum_{j=1}^{N} \alpha_{kj}s_{kj} \leq 1 \quad (2)
\]

The power consumption of \( XP_{ij} \) is

\[
P_{ij} = r_{ij} \left( \frac{V_{\text{max}}}{\alpha_{ij}} \right)^2 = s_{ij}r_{\text{max}} \left( \frac{V_{\text{max}}}{\alpha_{ij}} \right)^2
\]

neglecting all constants of proportionality. The total crossbar power consumption is the sum of the power contributions of all crosspoints:

\[
P_{\text{tot}} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} = \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ij}r_{\text{max}}V_{\text{max}}^2
\]

Finally, the minimum thermal power problem (denoted as OPT-MTP) becomes: given a admissible \( S \), find \( \hat{\alpha} \) that minimizes the cost function \( f_P \):

\[
\min_{\alpha} f_P(\hat{\alpha}) = \min_{\alpha \in \mathbb{R}^+} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{s_{ij}}{\alpha_{ij}} \quad (3)
\]

such that

\[
\sum_{k=1}^{N} \alpha_{ik}s_{ik} \leq 1 \quad \forall i \quad (4)
\]

\[
\sum_{k=1}^{N} \alpha_{kj}s_{kj} \leq 1 \quad \forall j \quad (5)
\]

\[
\alpha_{ij} \in A \quad (6)
\]

where \( A \) is the set of all available voltage levels.

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

Following a standard methodology, we start to relax OPT-MTP to continuous variables; this defines the following problem, denoted as CONT-MTP: minimize (3) subject to (4) and (5); (6) is substituted by

\[
\alpha_{ij} \geq 1 \quad \forall i, j
\]

that corresponds to a DVFS scheme in which any voltage between 0 and \( V_{\text{max}} \) is allowed. Let \( \hat{\alpha}_{\text{OPT-MTP}} \) be the optimal solution of OPT-MTP. Let \( \hat{\alpha}_{\text{CONT-MTP}} \) be the optimal solution of CONT-MTP.

**Property 2:** \( f_P(\hat{\alpha}_{\text{CONT-MTP}}) \leq f_P(\hat{\alpha}_{\text{OPT-MTP}}) \)

i.e. \( \hat{\alpha}_{\text{CONT-MTP}} \) provides a lower bound on the thermal power cost.

**Theorem 1:** CONT-MTP is equivalent to

\[
\min_{\alpha} f_P(\hat{\alpha}) = \sum_{k=1}^{N} \sum_{i=1}^{N} \frac{s_{ik}}{\alpha_{ik}} = 1 \quad \forall i \quad (7)
\]

\[
\sum_{k=1}^{N} \sum_{j=1}^{N} \alpha_{kj}s_{kj} = 1 \quad \forall j \quad (8)
\]

\[
\alpha_{ij} \geq 1 \quad \forall i, j \quad (9)
\]

**Proof:** Let assume \( \hat{\alpha} = [\alpha_{ij}] \) to be the optimal solution. Let define \( s_{ij} = \alpha_{ij}s_{ij} \). By contradiction, assume that there exists \( i \) such that \( \sum_{k} s_{ik} < 1 \), i.e. \( i \)-th row 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. \( \sum_{k} s_{kj} < 1 \); hence, it is possible to increase \( s_{ij} \) to \( s'_{ij} \) while satisfying all the constraints (4)-(5); the new corresponding \( \alpha'_{ij} = s'_{ij}/s_{ij} \) is feasible and provides a lower cost function; this contradicts our assumption. In the second case, all the columns sum to one. Thus, \( \sum_{j} \sum_{k} s_{kj} = N \), implying that \( \sum_{i} \sum_{k} s_{ik} < N \), contradicting our assumption.

Note that exactly one of the constraints in (7)-(8) is linearly dependent from the other, and it can be omitted.

**Definition 5:** Given a non-negative matrix \( H \in \mathbb{R}^{N \times N} \), \( H \) is said to be \( \rho \)-double-stochastic if \( \rho_1^i = \rho \) for any \( i \) and \( \rho_1^O = \rho \) for any \( j \). In this case, \( \rho_{\text{ave}}(H) = \rho_{\text{max}}(H) = \rho \). 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 \( \rho \)-sub-stochastic if \( \rho_{\text{max}}(H) = \rho \). In this case, \( \rho_{\text{ave}}(H) = \rho_{\text{max}}(H) = \rho \).

Thanks to Theorem 1, CONT-MTP has the following explanation: given a \( \rho \)-sub-stochastic matrix \( S \), find a double-stochastic matrix \( \hat{S} = [\hat{s}_{ij}] \) such that the set of \( \alpha_{ij} = s_{ij}/s_{ij} \) minimizes (3). Hence, the problem consists of augmenting \( \hat{S} \) such that it becomes double-stochastic.

In the following specific case, we can analytically compute the optimal solution:

**Theorem 2:** Given a \( \rho \)-double-stochastic matrix \( S \), the optimal solution for CONT-MTP is \( \alpha_{ij} = 1/\rho \), for any \( i, j \). The corresponding power consumption is \( f_P(\hat{\alpha}_{\text{CONT-MTP}}) = N\rho^4 \).

The proof is based on the use of the Lagrange multipliers and on the Taylor’s Theorem for multivariate functions, and it is omitted here due to lack of space. Furthermore, we can get an important intuition from the above Theorem, that will drive the design of approximated algorithms for the CONT-MTP problem: In the optimal solution, all the \( \alpha_{ij} \) are expanded proportionally by the same factor.

When considering also \( V_{\text{min}} \), the expansion ratio is limited by: \( \alpha_{ij} \leq 1/\beta \). The optimal solution becomes \( \alpha_{ij} =
Note that, by construction, the set of $\alpha'_{ij}$ defines an admissible solution for Opt-MTP.

To solve Cont-MTP, we adopt two approaches:

1. a quasi-optimal algorithm (denoted as Opt), obtained by adopting the logarithmic barrier method for convex problems [9]. It provides an $\epsilon$-approximation of the optimal solution, where $\epsilon$ is an input parameter, with enough large number of iterations. Unfortunately, it converges quite slowly in our scenarios. Thus, we adopt it as a reference case for the optimal solution only.

2. a two-steps algorithm: we augment $S$ to a double stochastic $\hat{S}$ according to three algorithms: AUGM-1, AUGM-MAX or AUGM-SORT algorithm. Then, we compute $\alpha_{ij} = \hat{s}_{ij} / s_{ij}$.

The three algorithms to augment $S$ to a double-stochastic $\hat{S}$ are based on the MATRIX-INCREASE algorithm, described in the pseudo-code below.

\[ \begin{array}{l}
\text{Input: } N \times N \text{ matrix } S = [s_{ij}], [\rho^I_{ij}]_{i=1}^N, [\rho^O_{ij}]_{j=1}^N, \rho_T, \Omega^I, \Omega^O. \\
\text{Output: } N \times N \text{ matrix } \Delta = [\delta_{ij}]. \\
1. \delta_{ij} = 0 \text{ for any } 1 \leq i, j \leq N. \\
2. \Omega^I = \{(i,j): i \in \Omega^I, j \in \Omega^O\}. \\
3. \text{repeat until no choice is anymore available} \\
4. \text{choose any } (i,j) \in \Omega^I \text{ such that } \max\{\rho^I_{ij}, \rho^O_{ij}\} < \rho_T \\
5. \delta_{ij} = \min\{\rho^I_{ij} - \rho^O_{ij}, \rho^O_{ij} - \rho_T\} \\
6. \rho^I_{ij} = \rho^I_{ij} + \delta_{ij}, \quad \rho^O_{ij} = \rho^O_{ij} + \delta_{ij} \\
\end{array} \]

The MATRIX-INCREASE algorithms inputs are i) a sub-stochastic matrix $S$, whose corresponding row $\rho^I_{ij}$ and column $\rho^O_{ij}$ sums are pre-computed; ii) a target load value $\rho_T$ such that $\rho_T \leq \max_k \{\rho^I_{ik}\}$ and $\rho_T \leq \max_k \{\rho^O_{kj}\}$, and iii) a set of inputs $\Omega^I$ and a set of outputs $\Omega^O$. The algorithm returns a matrix $\Delta = [\delta_{ij}]$ with the largest possible elements such that:

(i) only the elements $\delta_{ij}$ corresponding to rows and columns present in both $\Omega^I$ and $\Omega^O$ may be $> 0$; (ii) the maximum row and column sum is $\rho_T$, i.e.

\[ \begin{array}{l}
\sum_{k=1}^N s_{ik} + \delta_{ik} \leq \rho_T \text{ for any } i \in \Omega^I \\
\sum_{k=1}^N s_{kj} + \delta_{kj} \leq \rho_T \text{ for any } j \in \Omega^O \\
\end{array} \]

The algorithm operates on a sub-matrix corresponding to the rows in $\Omega^I$ and the columns in $\Omega^O$. It chooses a sequence of elements in such sub-matrix for which both row and column sum to less than $\rho_T$. Then, each element in the sub-matrix is augmented as much as possible to reach $\rho_T$. Note that the maximum number of iterations in steps 3-6 is upper bounded by $2N$.

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

- AUGM-1: i) compute $\rho^I_{ij}$ and $\rho^O_{ij}$ for any $i$ and $j$; ii) run INCREASE-MATRIX on $S$, $\rho^I_{ij}$, $\rho^O_{ij}$, $\rho_T = 1$, $\Omega^I = \Omega^O = \{1, \ldots, N\}$; iii) compute $s_{ij} = s_{ij} + \delta_{ij}$. This algorithm
is a classical iterative algorithm (see Sec. II.A of [10]) to augment a sub-stochastic matrix to a double-stochastic one. The complexity is $O(N^2)$, due to steps i) and iii).

- **AUGM-MAX**: i) compute $\rho^l_i$ and $\rho^O_j$ for any $i$ and $j$; ii) compute $\rho_{\text{max}}(S)$ based on (i); iii) run **INCREASE-MATRIX** on $S$, $\rho^l_i$, $\rho^O_j$, $\rho_T = \rho_{\text{max}}(S)$, $\Omega^I = \Omega^O = \{1, \ldots, N\}$; iv) compute $\hat{s}_{ij} = s_{ij} + \delta_{ij} + (1 - \rho_{\text{max}}(S))/N$. The complexity is $O(N^2)$, due to steps i) and iv).

- **AUGM-SORT**: i) initialize $\hat{S}$ as $S$ by setting $\hat{s}_{ij} = s_{ij}$ for any $i$ and $j$; ii) compute $\rho^l_i$ and $\rho^O_j$ for any $i$ and $j$; iii) sort $\rho^l_i$ and $\rho^O_j$ in increasing order; the induced sequence of inputs is described by $\hat{I}(k)$ defined as the $k$-th input and by $\hat{O}(k)$ defined as the $k$-th output; iv) let $\Omega^I = \Omega^O = \emptyset$. Iterate, for $k$ from 1 to $N$, the following procedure: iv.a) $\Omega^I = \Omega^I \cup \hat{I}(k)$, i.e. compute the set of the $k$ inputs with the smallest row sums; iv.b) $\Omega^O = \Omega^O \cup \hat{O}(k)$, i.e. compute the set of the $k$ outputs with the smallest column sums; iv.c) run **INCREASE-MATRIX** on $S$, $\rho^l_i$, $\rho^O_j$, $\Omega^I$, $\Omega^O$ and $\rho_T = \max\{\rho^l_i(k), \rho^O_j(k)\}$, i.e. $\rho_T$ is the maximum load for first $k$ inputs and outputs of $S$; iv.d) update $\hat{s}_{ij} = s_{ij} + \delta_{ij}$ for any $i$ and $j$, and continue with a new iteration; v) compute $\hat{s}_{ij} = \hat{s}_{ij} + (1 - \rho_{\text{max}}(S))/N$. The complexity is $O(N^2)$. In iv) this complexity is achieved by optimizing the data structure to choose an $(i,j) \in \Omega^I \Omega^O$ in **INCREASE-MATRIX** and by initializing only once $\delta_{ij}$.

The intuition behind these algorithms is the following. AUGM-1 is a classical way to augment a matrix, but has the disadvantage that it increases a selected element to set the sum of the corresponding row or column equal to one. As such, a non uniform increase in matrix element is obtained. AUGM-MAX is a simple variant of AUGM-1 in which the matrix is augmented to reach exactly $\rho_{\text{max}}$ for all rows and columns. Then, all matrix element are proportionally augmented until the matrix becomes double stochastic. This approach is based on the intuition derived from Theorem 2. Unfortunately, if even a single row or column sums to 1, AUGM-MAX degenerates into AUGM-1. Finally, AUGM-SORT order rows and columns in the original matrix so as to define a set of sub-matrices of different size. Each $k \times k$ sub-matrix includes a sub-matrix of size $(k-1) \times (k-1)$ of smaller load. Elements in each sub-matrix are augmented to reach the maximum admissible value in the sub-matrix, starting from the sub-matrix of smallest size.

### IV. Performance Evaluation

#### A. Simulation settings

To compare the DVFS schemes, we define the relative power $\eta(\hat{\alpha})$ of a DVFS solution $\hat{\alpha}$, with respect to NoDVFS, as follows:

$$\eta(\hat{\alpha}) = \frac{f_p(\hat{\alpha})}{f_p(\text{NoDVFS})} = \frac{f_p(\hat{\alpha})}{N_{\text{ave}}(S)}$$

thanks to (11). Roughly speaking, $\eta(\hat{\alpha})$ is the power reduction factor (power gain) compared to NoDVFS. Since $\eta(\hat{\alpha}) \in [0, 1]$, the closer $\eta(\hat{\alpha})$ to zero, the larger the scheme gain with respect to NoDVFS.

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

#### B. Power consumption under $\rho$-double-stochastic matrices

According to Theorem 2, the optimal solution for CONT-MTP is expressed by (10). Fig. 2 shows the power consumption per port $f_p(\hat{\alpha})/N$ vs. the average load for CONT-MTP optimal solution and $\beta \in \{0.3, 0.5, 0.7\}$; NoDVFS shows a linear growing according to (11). For small loads, the DVFS scheme is really efficient, and the power reduction can be as high as 50%. For larger loads, the DVFS gain decreases, becoming negligible in highly loaded conditions, because bit expansion is not allowed due to the high traffic load.

Next we consider the OPT-MTP problem, with a finite set of voltage levels. Fig. 3 shows the power per port for $|\mathcal{A}| = 4$ for different values of $\beta$, assuming that all voltage values in $\mathcal{A}$ are equally distributed between $V_{\text{min}}$ and $V_{\text{max}}$. The simple solution to CONT-MTP, represented by the ideal curve $\rho_{\text{ave}}(S)^3$, approximates well the more computationally intensive solution to the OPT-MTP problem. Note that even with only 4 voltage levels, the power gain is very close to the power gain that could be obtained with infinite voltage levels, i.e., the practical solution is not far from the ideal one.
C. Power consumption under $\rho$-sub-stochastic matrices

For general traffic matrices, simple optimal solutions are not available. We consider the family of uniform traffic matrices generated as follows. Given $\rho_{\text{max}}(S) \in (0,1]$, generate $N^2$ random variables $u_{ij}$ uniformly distributed on the interval $(0,1]$. Then, compute $\rho_{\text{max}}(U)$ and derive each element of $S$ as

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

Using this construction, it can be shown that the corresponding average load follows the law $\rho_{\text{ave}}(S) = \rho_{\text{max}}(S)/(1 + \Theta(\sqrt{\log(N)/N}))$ for enough large $N$.

We compare the algorithms proposed in Sect. III-B for the CONT-MTP problem, since we have observed in the previous section that it is a good approximation of OPT-MTP even when few voltage levels are available. The solution to MISO-MTP is easily obtainable, but it provides potential power gain that could not be reached in practice, given that the constraints among different output are neglected. The OPT algorithm instead is a tight bound but its computational complexity is huge. We set $\beta = 0.3$ and we do not report results for other values of $\beta$, being qualitatively similar.

Fig. 4, 5 show the relative power vs. the average load, for $N = 16$ and $N = 256$, respectively. Note that the maximum average load in the abscissa is limited due to the procedure to compute $S$. Furthermore, we have already observed that the power gain for higher load is negligible.

For $\rho_{\text{ave}}(S)$ increasing, $\eta(\hat{\alpha}_{\text{MISO-MTP}})$ shows a quadratic growth, as expected by combining (10) with (12). More interestingly, also $\eta(\hat{\alpha}_{\text{AUGM-SORT}})$, $\eta(\hat{\alpha}_{\text{AUGM-MAX}})$ and $\eta(\hat{\alpha}_{\text{OPT}})$ show a similar growth. As a consequence, the algorithms AUGM-SORT, AUGM-MAX and OPT are close to the lower bound MISO-MTP. Looking $\eta(\hat{\alpha}_{\text{AUGM-1}})$, it is near to $\eta(\hat{\alpha}_{\text{NoDVFS}})$. Regardless of the considered load, AUGM-1 is worse, being close to NoDVFS, the solution in which DVFS is not adopted.

The above discussed results show the power efficiency of the DVFS approach for small sizes switching fabric. To understand the algorithm scalability, we consider larger switching fabric size. Results for the CONT-MTP problem are presented in Fig. 5 for $N = 256$. Power gain are quite similar to those observed for smaller size switching fabric. $\eta(\hat{\alpha}_{\text{AUGM-1}})$ overlaps $\eta(\hat{\alpha}_{\text{NoDVFS}})$ to confirm the inability of AUGM-1 to exploit the DVFS gain. Due to its slow convergence, especially at higher average load values, we do not present results for OPT in this scenario. Again AUGM-SORT and AUGM-MAX provide performance close to the lower bound. Thus, DVFS schemes can be efficiently used to reduce power consumption especially at low average load regardless of the switch size.

V. Conclusions

We discussed the potential power gains that DVFS techniques can provide when controlling a crossbar used as a switching fabric in an input-queued switch. We took an asymptotic approach, disregarding the details related to packet scheduling, and looking at flow rates. Performance results show that a significant power reduction can be obtained, especially at low loads. The proposed algorithms are computationally simple and obtain performance gain close to those of more complex, optimal algorithms.

REFERENCES