Dynamic Voltage and Frequency Scaling Control for Crossbars in Input-Queued Switches

Dynamic Voltage and Frequency Scaling Control for Crossbars in Input-Queued Switches

Andrea Bianco, Paolo Giaccone, Marco Ricca
Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino, Italy

***Abstract*—The power consumption in chips, in general, and in crossbars switching fabrics, in particular, grows with the maximum sustainable throughput. Due to the fast increasing traffic demands, the performance scalability of crossbars is severely limited by the capability of cooling the hardware devices. Hence, reducing the power consumption is an important design question to improve the crossbar switching performance.**

We propose to leverage Dynamic Voltage and Frequency Scaling (DVFS) hardware technique for the switching fabric. The main idea is to exploit temporary underloaded conditions to decrease the crossbar transmission rate while preserving maximum throughput. Differently from previous works, we consider a scenario in which the arrival rates are unknown in advance. Our proposed architecture is based on a power controller which runs periodically and independently of the packet scheduler, and whose decisions are based on the real time estimation of the arrival rates. We discuss the performance tradeoff in terms of throughput, delays and power, and show the relevant performance gain due to the use of DVFS in controlling the crossbar.

I. Introduction

Internet devices (e.g. high speed core routers) are designed to run at fully utilization, following a classical worst-case design approach. However, given the highly-variable Internet traffic features, the real device utilization is around 30-50% [1]. As a practical consequence, a network element, even if often underutilized, still consumes the maximum power. Power consumption can critically limit the performance due to the significant heating of the hardware components, in particular for the switching fabrics implemented on chip. Indeed, the power consumption increases more than linearly with respect to the aggregate bandwidth [2], [3], thus a sufficient chip cooling is becoming more and more difficult to achieve. Reducing this thermal dissipation "bottleneck", or reducing the on-chip power consumption, represents a challenging issue to solve. Many solutions have been investigated in the past, but they targeted mainly data processing elements (e.g. CPUs) and not data switching elements.

Power consumption of an integrated CMOS-based switching fabric is due to two main contributions, named static and dynamic. We consider only the dynamic power, due to the gates activity when transferring digital signals. We neglect the static power, due to leakage currents, since it tends to be proportional to the occupied area and can be controlled by means of circuit-level techniques that are complementary to the scheme considered in our work.

The dynamic power can be reduced through techniques like Dynamic Voltage and Frequency Scaling (DVFS) [4], [5], in which the power decreases by jointly lowering the power supply voltage and decreasing the clock frequency. The price to pay are temporal overhead, due to the need to change voltage/frequency, and increased latency in data transfer.

We aim at reducing the dynamic power consumption in a CMOS-based switching fabric. In a classical architecture, data packets are always transferred at the maximum speed. The approach we propose exploits DVFS in temporary underload conditions, by decreasing the packet transmission rate across the whole switching fabric. Indeed, to simplify the hardware design we assume that a single voltage is supplied to the whole chip. We wish to achieve the best tradeoff between performance (in terms of throughput and delay) and power consumption by an on-line power control policy, which adapts the DVFS scheme to the actual traffic pattern. We propose to estimate the incoming packets arrival rate and to periodically set the minimum voltage level and clock frequency to guarantee high throughput and bounded delays. The temporal overhead induced by the DVFS scheme is controlled by keeping large enough voltage/frequency update periods.

The reminder of the paper is organized as follows. Sec. II introduces the system model. Sec. III discusses some relevant previous work. Finally, Sec. IV describes the proposed power-aware switching architecture, whose performance are investigated in Sec. V. Conclusions are drawn in Sec. VI.

II. System model

We consider a $N \times N$ *Input Queued* (IQ) switch where queues are organized in a *Virtual Output Queuing* (VOQ) structure, i.e. one FIFO queue able to store at most B packets for each input-output pair, as shown in Fig. 1. We assume as a reference scenario fixed-size packet transmission to permit synchronous transfers across the switching fabric. Thus, time is slotted and one timeslot corresponds to the transmission time of one packets. In the case of variable-size data units (e.g. IP

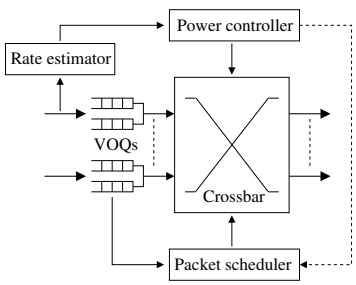

Fig. 1. Architecture of the on-line power-aware control for IQ switches

packets), they are chopped into fixed-size packets, according to a standard scheme [6].

The packets arrived at timeslot t are described by a $N \times N$ *arrival matrix* $A(t) = [a_{ij}(t)]$ where i is the input and j is the output, with $1 \leq i, j \leq N$. It holds $a_{ij}(t) = 1$ when a new packet arrives, otherwise $a_{ij}(t) = 0$. We assume a stationary arrival process and define the *rate matrix* as $\Lambda = [\lambda_{ij}]$, where $\lambda_{ij} \triangleq \mathbb{E}[a_{ij}(t)]$; by construction $0 \leq \lambda_{ij} \leq 1$. Given a traffic matrix Λ, its *maximum load* is computed as the maximum row/column sum as:

$$\rho_{\max}(\Lambda) = \max\left(\max_{k=1,\ldots,N}\left(\sum_{i=1}^{N}\lambda_{ik}, \sum_{j=1}^{N}\lambda_{kj}\right)\right)$$

Its *average load* is computed as

$$\theta(\Lambda) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_{ij}$$

Traffic Λ is said to be *admissible* iff $\rho_{\max}(\Lambda) < 1$. Indeed, an output queued switch would be able to transfer such traffic with bounded average delays and would achieve the maximum throughput.

A. Power consumption in a crossbar

Referring again to Fig. 1, we assume an electronic crossbar chip as switching fabric: every input port is connected through a *crosspoint* to every output port; in total, N^2 crosspoints are present. We consider the on-chip dynamic power consumption due to the CMOS gate activity to transfer a stream of bit. As also shown in [7] and validated in [2], [3], such power is known to be proportional to λV^2 where V is the operating voltage and λ is the bit rate (i.e. the traffic).

In existing designs, the gates corresponding to each *crosspoint* run always at the maximum available operating voltage $V_{\max}$, independently of its actual load. In such a scenario, the dynamic power is proportional to the number of packet transferred over time, i.e. the arrival rate if losses are not experienced. Instead, we propose to apply DVFS with a reduced operating voltage $V = V_{\max}/\alpha$ equal for the whole switching fabric, where $\alpha \geq 1$ can be seen either as the voltage reduction factor, or the bit *expansion factor*. Indeed, when decreasing the voltage, the clock frequency (in this case, the service bitrate) must be proportionally reduced to permit correct bit reception. The expansion factor is bounded by a maximum value $\alpha_{\max}$, because of some minimum voltage threshold that depends on the adopted hardware technology. In practical cases, $\alpha_{\max} \in [2, 3]$, as discussed in [8]. At the same time, because of the lower clock frequency, the packet transmission time is increased by a factor α and the offered load for a generic VOQ becomes $\alpha\lambda_{ij}$. To avoid overloading the input/output ports, it must be:

$$\rho_{\max}(\alpha\Lambda) < 1 \qquad \Rightarrow \qquad \rho_{\max}(\Lambda) < 1/\alpha \tag{1}$$

This relation provides a constraint to the maximum value of α. Note that (1) provides a necessary condition to achieve 100% throughput which is stricter than the traffic admissibility condition.

As shown in [7], given a static scenario in which arrival rates are known a-priori, the power consumption of each crosspoint is proportional to λ_{ij}/α^2, if no losses are experienced. Thus, the power consumption of the whole switching fabric can be obtained by adding the contributions of all crosspoints:

$$P_{tot} = \frac{1}{\alpha^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_{ij} \tag{2}$$

Differently from [7], we consider a dynamic scenario in which the arrival rates are a-priori unknown and the maximum throughput varies with time due to the DVFS adaptation of the chip to the arrival pattern. We assume to change the voltage periodically, every T_{up} seconds. Whenever the voltage and the frequency is changed, some reset time τ is experienced, during which the crossbar is not able to forward packets. Assuming on-chip voltage regulators, voltage transitions can occur in the order of tens of nanosec, several orders of magnitude faster than off-chip regulators [5]. This time can be compensated by setting large enough T_{up}, e.g., at least to some microseconds. Note that $T_{up} = 10\ \mu s$ corresponds to 200 packets of 64 bytes transferred at 10 Gbit/s, or 20 packets transferred at 1 Gbit/s. We will consider in this paper always situations in which $T_{up} \gg \tau$, to guarantee negligible reset times. If some packets are still in transmission through the crossbar when the voltage transition is expected to occur, the transition is delayed until the end of the current transmissions, to guarantee that each packet transfer is not interrupted.

We define a sequence of epochs, indexed by n. During each epoch, whose (approximated) duration is T_{up}, the voltage is kept constant. Let $\alpha_n \in [1, \alpha_{\max}]$ be the expansion factor during epoch n and $\mu_{ij}(n)$ be the corresponding throughput (measured, for example, in terms of packet/s) for a specific crosspoint. Since the energy consumption of a crosspoint in an epoch is proportional to $\mu_{ij}(n)T_{up}/\alpha_n^2$, similarly to (2) the average power consumption, after h epochs, is equal to:

$$P_{tot}(h) = \frac{1}{h}\sum_{n=1}^{h}\frac{1}{\alpha_n^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mu_{ij}(n) \tag{3}$$

The power control problem is defined for a generic arrival process as:

$$\min \lim_{h\to\infty} P_{tot}(h) \tag{4}$$

where the minimum is evaluated across all the possible sequences $\{\alpha_n\}_{n=1}^{h}$, with $\alpha_n \in [1, \alpha_{\max}]$, and subject to the non-overload conditions derived from (1):

$$\rho_{\max}(\Lambda) < 1/\mathbb{E}[\alpha_n]$$

During each epoch, the packet scheduler, shown in Fig. 1, chooses the packets to transfer from the VOQs across the switching fabric, while satisfying the crossbar constraints that at most one packet can be transferred from each input and to each output. This requires to solve a standard matching problem, for which a wide set of hardware solutions are known from the literature [9]. Thanks to the fact that all the crosspoints work synchronously at the same frequency, the packet scheduler can be clocked to the crossbar internal frequency, running once every packet transmission, oblivious of the power control. This fact permits a simple integration

between the packet scheduler and the power control modules, because there is no direct dependency among the decisions of the two modules.

III. Related works on DVFS in IQ switches

A very large literature is available on DVFS, e.g. [10]–[12]. We consider here only the works that have been specifically tailored to our scenario.

In [7] we investigated an ideal version of the power control problem for IQ packet switches, exploiting DVFS at crosspoint level, i.e. the voltage of each crosspoint can be controlled; this is different from the current work, since here we assume that a single voltage is applied to all the crosspoints simultaneously. A family of power control algorithms were proposed in order to compute the N^2 voltages, one for each crosspoint. The traffic matrix was assumed to be known and a fluid model was used to described the constant-bit-rate sources. In the current work, instead, we consider a stochastic arrival process in which arrival rates are unknown and the power control reacts to the actual packet arrivals. Here, we consider the same power model as [7] since it was validated through a real hardware synthesis. Furthermore, [7] showed that, under a large class of traffic scenarios, just one common voltage is enough to achieve nearly optimal power performance; thus, the implementation complexity can be reduced thanks to a single voltage regulator. Motivated by such encouraging result, here we assume just one common voltage for the whole crossbar. Finally, due to the fluid nature of the sources, [7] investigates only the power and throughout tradeoff, without taking into account the delays, which are instead the main performance metrics considered in this work.

To better understand the achievable tradeoff between delays and power, in [13] we have recently analyzed a single queue system in which the server modifies its service rate to minimize the power consumption, while achieving maximum throughput. We have investigated the performance achieved by a family of power policies that are targeting either (i) minimum power or (ii) a fixed average queue size or (iii) a fixed utilization of the queue. We showed that in all these cases the average delays show a non-monotonic behavior with respect to the offered load. To limit the possible negative effects of such behavior on the congestion control mechanisms at network level, we proposed some control schemes to achieve monotonic delays. In the current work we adapt the policy targeting a fixed queue utilization to the set of VOQs. Interestingly, the same non-monotonic behavior of the delays shown in [13] for a single queue is observed in our more complex scenario consisting of a network of interacting queues.

Similarly to our work, [14] studies the delay-power tradeoff, for an IQ switch with VOQ, achievable by an optimal dynamic power control policy. Packet scheduling and power control decisions are integrated into a single scheduler module, which computes both the packets to transfer (i.e. the crossbar matching) and the corresponding transmission rate. The transmission rate is assumed to be equal for all the packets under transmission as in our scenario with one single voltage/frequency for the whole crossbar. The proposed scheduler by [14] must solve a complex optimization problem at each timeslot and this may unfeasible at high speed. Instead, in our work we rely on a standard packet scheduler (eventually already implemented on a chip) and on a power control which is easy to implement in hardware, and whose decisions, even if not provably optimal, will be shown to be efficient in terms of power/delay performance. Furthermore, the transmission rate in [14] varies packet-by-packet, and this fact can introduce some non-negligible overhead due to the reset time. For example, for a 64 byte packet arriving on a 10 Gbit/s link, the voltage must vary every 50 ns, which is compatible with current on-chip voltage regulators but not with the reset time which is around tens of ns [5]. Instead, in our scenario we decouple the time scale of the packet scheduling decisions ($\approx$ ns) with the one of the voltage/frequency variations ($\approx \mu$s).

IV. On-line power control

To exploit DVFS capabilities in the switching fabric, we propose the complete architecture represented in Fig. 1. Based on the most recent estimation of the arrival rates, at the beginning of epoch n the power controller computes the new expansion factor α_n, that drives the voltage regulator during the current epoch. In the following, we describe in details each module implemented in the architecture.

A. Estimator of the arrival rates

Arrival rates are estimated with an exponential moving average, based on arrived packets at each VOQ during timeslot t, according to the classical relation:

$$\hat{\lambda}_{ij}(t) = \beta\hat{\lambda}_{ij}(t-1) + (1-\beta)a_{ij}(t)$$

Here $\beta \in (0,1)$ is the averaging parameter, whose *rate estimation window* W can be computed as $W = \log(1-\zeta)/\log(\beta)$, where ζ is the filter threshold. In the following, we set $\zeta = 0.99$. Note that, when the arrival process is also ergodic, $\hat{\lambda}_{ij}(t) \to \lambda_{ij}$ for $t \to \infty$ and $\beta \to 1$. Let $\hat{\Lambda}_n$ be the matrix with the current estimated rates at the beginning of epoch n.

B. Power controller

During each epoch, the power controller (PC) selects the *expansion factor* α for all crosspoints, based on the estimated rates. The main idea is to choose the largest possible value of α compatible with the non-overload conditions, based on the estimated arrivals during the last rate estimation window (i.e. approximatively the last W slots). More formally, during epoch n the expansion factor is evaluated as the maximum $\alpha_n \in [1, \alpha_{\max}]$ that guarantees

$$\alpha_n \rho_{\max}(\hat{\Lambda}_n) \leq 1 \tag{5}$$

This relation corresponds to run the switch in an operating point for which the throughput is maximum but delays would grow unbounded (in the case of infinite queue). To keep finite delays, we introduce a control parameter $\rho_v \in (0,1)$ named *virtual load*, which corresponds to a "safety margin" to avoid overloading the VOQ. So the original policy is modified to guarantee

$$\alpha_n \rho_{\max}(\hat{\Lambda}_n) \leq \rho_v \tag{6}$$

instead of (5). By reducing ρ_v, we get smaller delays. This new defined policy, denoted as PC-ρ_v, is the extension of the "fixed-utilization" policy proposed in [13] for a single queue.

To satisfy (6), PC-ρ_v must choose $\alpha_n = \rho_v / \rho_{\max}(\hat{\Lambda}_n)$. The following pseudo code reports the final algorithm and highlight three operational regimes.

POWER CONTROL Algorithm

Input: $\hat{\Lambda}_n$, ρ_v. **Output**: α_n.

1. Compute $\gamma = \rho_{\max}(\hat{\Lambda})$
2. Compute expansion factor

$$\alpha_n = \begin{cases} 1 & \text{if } \gamma > \rho_v \quad \text{(high load)} \\ \dfrac{\rho_v}{\gamma} & \text{if } \dfrac{\rho_v}{\alpha_{\max}} \leq \hat{\rho}_n < \rho_v \quad \text{(medium load)} \\ \alpha_{\max} & \text{if } \gamma < \dfrac{\rho_v}{\alpha_{\max}} \quad \text{(low load)} \end{cases}$$

Under high load, DVFS is not active and the crossbar runs at the maximum speed. For medium load, the optimal value of α_n is chosen to target exactly ρ_v as maximum utilization factor among the VOQs; thus, when decreasing the load, α_n is increased until it reaches its maximum value $\alpha_{\max}$.

Now two main issues must be discussed. First, we must investigate the effect of W on PC. Second, we must understand whether the scheme is robust also for non-uniform traffic patterns. We will devote the following section to discuss such issues.

V. PERFORMANCE ANALYSIS

We developed a discrete-time simulator, written in C language, to assess the performance of our proposed power-aware IQ switch. The considered performance metrics will be the average packet delay, the average power per port and the maximum queue occupancy, as a function of the offered average load, denoted as θ. Since the power control runs independently of the packet scheduler, we have chosen the well known iSLIP as the reference packet scheduler [15]. Note that iSLIP is an iterative algorithm to compute a maximal size matching. It is amenable to efficient hardware implementation due to its intrinsic parallel behavior.

A. Scenario description

We report the results obtained only for $N = 16$. Similar results, not reported for space limitations, were obtained for smaller switch sizes ($N = 8$) and larger ones ($N = 32, 64$). Following a standard methodology, packets arrivals have been generated with a Bernoulli i.i.d. process, according to a given traffic matrix Λ. To highlight the potential gain due to DVFS, we have chosen the extreme case in which $\alpha_{\max} = 3$, i.e. the voltage (and frequency) can be reduced at most to one third of the nominal voltage (and frequency). Coherently with the discussion in Sec. II, we have chosen $T_{up} = 200$ timeslots as the voltage and frequency update period.

Our proposed PC-ρ_v is simulated with the online rate estimation described in Sec. IV-A. We consider different values of the rate estimation window W to highlight the interaction with the power control. In particular, we will always choose $W > T_{up}$, because we expect to change the voltage/frequency on a time scale which is smaller than the rate estimation window. Note that the case in which $W \ll T_{up}$ is not relevant, because it would capture traffic transient behaviors that cannot be exploited by the power controller, which is running at a much larger scale time. Furthermore, if W is too small, the rate estimation process would be unstable. Thus, we considered $W \in \{200, 10^3, 10^4, 10^5\}$ timeslots.

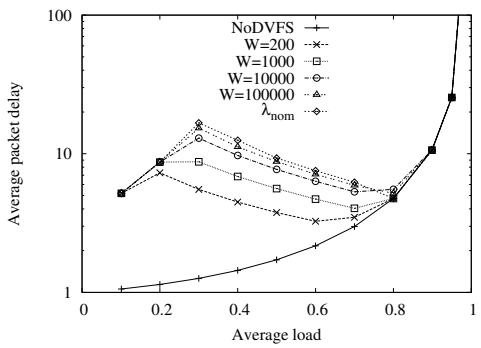

Fig. 2. Delay under uniform traffic for NoDVFS and PC-0.8.

We will compare our proposed PC-ρ_v with the case in which DVFS is not exploited. The latter case, denoted as *NoDVFS*, corresponds to $\alpha_n = 1$, $\forall n$. By construction, it provides a lower bound for PC-ρ_v on the achievable delays and an upper bound on the power consumption. In the following simulations, we will show only the results for PC-0.8, because the chosen value for the virtual load offers a reasonable compromise between delays and power. We consider also, as a reference, the situation in which the rates are known in advance, as it was assumed in [7]. Given Λ, PC chooses a fixed value for α_n, $\forall n$. This case will be denoted as λ_{nom} ("nominal rate"). It is a useful reference because it provides a strict lower bound on the power for DVFS, and permits to understand the effect of the rate estimator on PC.

B. Uniform traffic

We start considering uniformly distributed traffic, i.e. $\lambda_{ij} = \theta/N$ for all i and j. Fig. 2 shows the average packet delay vs the average load. One curve refers to NoDVFS, all the others to PC-0.8 for different rate estimation schemes. As previously noted, under NoDVFS scheme the crossbar runs always at the highest clock frequency, thus providing minimum packet delays. By construction, the corresponding curve is the same as the one achieved by the standard iSLIP [15] scheduler. For low load ($\theta < 1/\alpha_{\max} \approx 0.33$), $\alpha_n = 3$ for most of the time, and the minimum delay is always lower bounded by 3 timeslots. Interestingly, the delays show a non-monotonic behavior, because for $\theta > 0.33$ the delays decrease again. This interesting fact has been previously investigated in [13], which showed that it occurs for a large family of power control algorithms. The main motivation can be understood qualitatively by considering a single queue fed by a stationary arrival process at rate λ and assuming unbounded α (i.e. $\alpha_{\max} \to \infty$). If λ is known, the power control, to optimally solve (4), sets $\alpha_n = 1/\lambda$, $\forall n$. Since α_n is also the service time of a packet, then for $\lambda \to 0$ the delay would go to infinity. In the case of a finite maximum value $\alpha_{\max}$, the delays remain bounded, but still reach a local maximum when θ approaches $1/\alpha_{\max}$.

Coming back to Fig. 2, under medium load ($\theta > 0.33$) the delays decrease until, for high load ($\theta > \rho_v$), the queues become congested and the delays grow again. Recall that, for high load, PC sets $\alpha_n = 1$, $\forall n$.

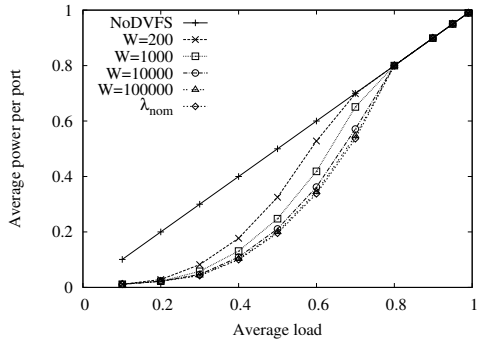

Fig. 3. Power consumption under uniform traffic for NoDVFS and PC-0.8.

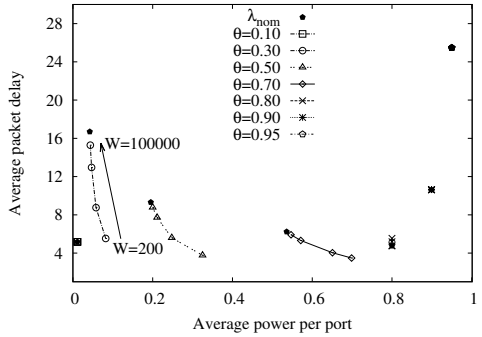

Fig. 4. Delay and power tradeoff under uniform traffic achieved by PC-0.8, for different

Under medium load, W plays an important role. Since the Bernoulli process is ergodic, by increasing W the estimated rates becomes closer to the nominal rate (i.e. $\hat{\Lambda}_n \to \Lambda$) and the delays go asymptotically to λ_{nom}. Furthermore, when W is large, due to the law of large numbers, $\rho_{\max}(\hat{\Lambda})$ in PC tends to the actual average load: $\rho_{\max}(\hat{\Lambda}) \approx \theta$. On the contrary, for small W, the averaging effect is less evident. Thus, $\hat{\Lambda}_n$ is a worse estimation of Λ. Furthermore, $\rho_{\max}(\hat{\Lambda})$ tends to overestimate the actual load: $\rho_{\max}(\hat{\Lambda}_n) \gg \theta(\hat{\Lambda}_n)$. Indeed, as an extreme example, consider the case in which $W = 1$:

$$Pr(\rho_{\max}(\hat{\Lambda}_n) \geq 1) \geq 1 - (1 - \theta/N)^N \approx 1 - e^{-\theta}$$

With such probability, PC selects $\alpha_n = 1$, i.e. DVFS is not exploited: For example, with $\theta = 0.5$, the probability is > 0.39. This explains why small W tend to approach the NoDVFS case.

Fig. 3 compares the power consumption among the different schemes. As expected, the power for NoDVFS linearly grows with the load, coherently with (2) when $\alpha = 1$. The best policy in terms of power is PC-0.8 running on the actual arrival rates (λ_{nom}). In this case, the power grows as a cubic function of the load, highlighting a remarkable power reduction with respect to NoDVFS, especially at low load. As observed for the delay, for large W, PC performance tends to λ_{nom}, whereas for small W PC approaches NoDVFS.

To better understand the actual tradeoffs between power and delays, in Fig. 4 we show the delays, one curve for each load θ, as a function of the achievable power. The points represents different values of W, and the additional point

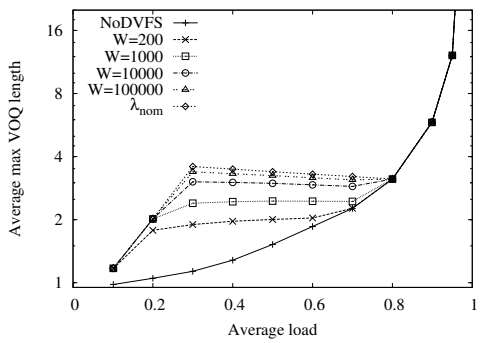

Fig. 5. Average maximum VOQ occupancy under uniform traffic for NoDVFS and PC-0.8

above each curve refers to the corresponding case λ_{nom}. Note that for medium load (i.e. $\theta \in \{0.3, 0.5, 0.7\}$), a different tradeoff is achieved for each value of W. All the corresponding points are Pareto optimal: given the same θ, for each operating point it is not possible to find a "better" point, i.e. with lower power and delay. Depending on the desirable tradeoff, a proper value for W can be chosen. Instead, for low load and high load, all the points of each curve degenerate into a single point, because the control always chooses the same α_n, independently of the rate estimation method.

As clarified by the previous figures, in our proposed power-aware system, lower power is exchanged with increased delays, and higher queue occupancies. Because of the possible mismatch between the most recent rate estimation and the actual arrival rates during the current epoch, some queues could grow "without control". Since occupancies are typically large, this may lead to buffer overflows and throughput reduction. To address this important issues, we have plotted the average length of the maximum queue across all the VOQs in Fig. 5. Interestingly, at medium load and independently of W, the (average) maximum VOQ length remains very small, only 2-3 packets larger than NoDVFS. Furthermore, the queue occupancy appears to be constant with respect to the load. This is not surprising, and it has also been observed in [13] for the so called "fixed utilization" (FU) policy. Intuitively, since delays are approximatively proportional to the service time α, which is proportional to $1/\theta$, by the well known Little's law, the average queue occupancy tends to be a constant, being proportional to $\theta\alpha = 1$.

C. Bidiagonal traffic

To evaluate the robustness of our proposed power-aware system, we have considered also a non-uniform scenario, defined as follows, for any $i \in \{1, \ldots, N\}$: $\lambda_{ii} = 2\theta/3$ and $\lambda_{i|i+1|_N} = \theta/3$, where $|\cdot|_N$ denotes modulus-N operation[1]; this scenario is defined as *bidiagonal traffic* because only the values on the first two diagonals of Λ are non-null. This traffic is considered "critical" to schedule, because only provably optimal (but unfeasible) packet schedulers are able to achieve the maximum throughput. Indeed, the standard iSLIP is known to achieve around 80% of throughput in this scenario.

[1]More precisely, to be able to operate on $x \in [1, N]$, we define $|x|_N = ((x-1) \mod N) + 1$

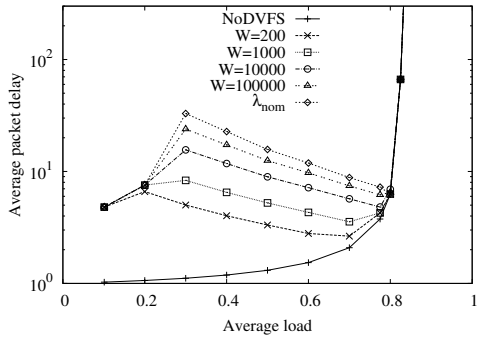

Fig. 6. Average delay under bidiagonal traffic for NoDVFS and PC-0.8

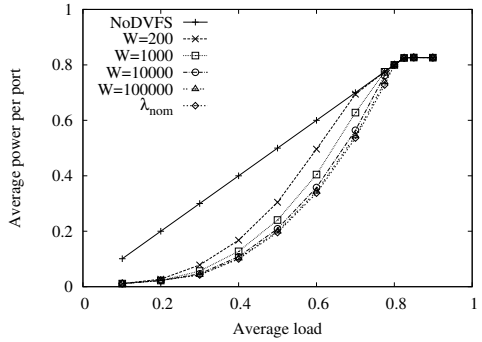

Fig. 7. Power consumption under bidiagonal traffic for NoDVFS and PC-0.8

Fig. 6 shows the average delays as a function of the offered load for NoDVFS and PC-0.8, under different rate estimation schemes. We can observe exactly the same qualitatively behavior of Fig. 2, showing the robustness of the proposed approach. The only difference is that, when $\theta > 0.8$, the packet scheduler is not able to cope with the specific arrival pattern (even without DVFS), and queue overflows occur and delays asymptotically grow.

Also the power consumption, shown in Fig. 7, exhibits the same qualitative behavior as in the case of uniform traffic. Notably, the power reduction of PC with respect to NoDVFS is remarkable. When the offered load becomes too high for iSLIP ($\theta > 0.8$), the power remains the same because the throughput is constant, due to the experienced losses.

VI. Conclusions

We considered an input-queued switch whose power consumption can be controlled through Dynamic and Voltage Frequency Scaling (DVFS) applied to the whole crossbar. The voltage and frequency of the crossbar are periodically changed to vary the transmission rate of the whole switching fabric, and are adapted to the current traffic conditions by a power controller (PC).

The intuitive idea is to reduce the transmission rate when the offered load is small, while preserving the maximum throughput. We define the offered load based on the maximum row and column sum of the estimated rate matrix. We introduce as the main control parameter the virtual load, which is the targeted maximum utilization factor at any input or output port, to control delays. We show that a remarkable power reduction is experienced when using our power control, with respect to the case in which DVFS is not applied. The power gain is traded with larger delays, which show a non-monotonic behavior with respect to the load. The desired tradeoff between power and delays can be achieved by setting a proper value of the rate estimation window. Furthermore, the overall power-aware architecture appears to be robust for different stationary arrival patterns.

From the implementation point of view, the computation required in the power-control module is low. Furthermore, this module runs independently of the packet scheduler, thus it can be easily integrated in any pre-existing input-queue switch architecture. As such, the proposed power-aware architecture offers an interesting compromise between implementation complexity and performance.

References

[1] J. Guichard, F. L. Faucheur, and J.-P. Vasseur, *Definitive MPLS Network Designs*. Cisco Press, 2005.

[2] T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Design Automation Conference*, 2002, pp. 524–529.

[3] H.-S. Wang, L.-S. Peh, and S. Malik, "A power model for routers: modeling Alpha 21364 and InfiniBand routers," *IEEE Micro*, vol. 23, no. 1, pp. 26–35, 2003.

[4] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Trans. on Very Large Scale Integrated Systems*, vol. 8, no. 3, pp. 299–316, June 2000.

[5] W. Kim, M. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2008, pp. 123–134.

[6] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "Packet-mode scheduling in input-queued cell-based switches," *IEEE/ACM Trans. on Networking*, vol. 10, pp. 666–678, October 2002.

[7] A. Bianco, P. Giaccone, G. Masera, and M. Ricca, "Power control for crossbar-based input-queued switches," *IEEE Trans. on Computers*, vol. 62, no. 1, pp. 74–82, 2013.

[8] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 11, pp. 1239–1252, Nov. 2005.

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

[10] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.

[11] R. Marculescu, U. Ogras, L.-S. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in noc design: System, microarchitecture, and circuit perspectives," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 3–21, 2009.

[12] L. Shang, L.-S. Peh, and N. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *High-Performance Computer Architecture (HPCA)*, 2003, pp. 91–102.

[13] A. Bianco, P. Giaccone, M. Casu, and M. Ricca, "Joint delay and power control in single-server queueing systems," in *IEEE OnlineGreenComm*, Oct. 2013.

[14] L. Mastroleon, D. C. O'Neill, B. Yolken, and N. Bambos, "Power and delay aware management of packet switches," *IEEE Trans. on Computers*, vol. 61, no. 12, pp. 1789–1799, 2012.

[15] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. on Communications*, pp. 1260–302, 1999.