Fine-Grained Power Control for Combined Input-Crosspoint Queued Switches

Yu Xia, Ting Wang, Zhiyang Su and Mounir Hamdi
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
emails: rainsia@gmail.com, {twangah, zsuab, hamdi}@cse.ust.hk

***Abstract*—Reducing the power consumption of packet switches is becoming increasingly significant to future networks. However, previous research all focused on reducing power in crossbar-based switches, which is either complex or not effective, especially in some extreme cases. This paper proposes to leverage the dynamic voltage and frequency scaling (DVFS) technique in the buffered crossbar-based switches, which is more flexible and simple. The basic idea is to decrease the working frequencies of the crosspoint buffers while still preserving the maximum throughput and the satisfactory delay. Traffic estimators are used at the input and output ports to estimate the traffic arrival rates, based on which the power controller can adjust the working frequencies of the crosspoint buffers at a fine-grained level. Simulation results show that the scheme is effective.**

I. Introduction

Power consumption of IT infrastructure is becoming an increasingly serious problem. It is reported that the annual electricity consumed by networking devices in the U.S. is 6.06 Terra Watt hours, or 1 billion US dollars, in 2003 [1], and is quickly increasing. Hence, reducing the unnecessary power consumption in network devices has its economic, environmental and marketing significance.

Power management techniques were proposed to match the power consumption to the actual system workloads. When workloads are low, some components of the system are slowed down or put into sleep mode or even turned off. Turning a device off and on incurs a significant overhead, thus, the frequency scaling approach is usually preferred. Such techniques have been successfully applied to microprocessors and hard disk drives [2–4] in the servers. However, only limited efforts were devoted to the core of networks, i.e., packet switches/routers, whose power consumption is rapidly increasing, given the ever-growing data rate and traffic demands.

Most of the switches are designed to run at fully utilization to support the worst-case traffic conditions. Even when a switch is lightly loaded, it still runs at the highest frequency and consumes the maximum power. However, Internet traffic varies largely. The real utilization is only around 30∼50% [5]. If the switches/routers are still running at the maximum rate all the time, then a large amount of unnecessary power will be wasted. Even worse, the scalability and the stability of the switches/routers are restricted due to the significant heat generated, especially from the switch fabric chips. Moreover, as the traffic rate or the size of a switch increases, the cooling system is becoming more and more difficult to design, which in turn limits the scalability and performance of the switches. Thus, it is critical to reduce this thermal dissipation "bottleneck", and to reduce the on-chip power consumption of the switch fabric.

Dynamic voltage and frequency scaling (DVFS) [6] is a well-known technique to limit the power consumption of electronic devices. Its idea is to jointly lower the power-supply voltage and the peak signal frequency of a device when the offered workload is low. However, most of the existing works all try to apply the DVFS scheme to the crossbar-based input-queued (IQ) switches. In this work, we propose to exploit the DVFS technique for the *buffered crossbar* used in combined input-crosspoint queued (CICQ) switches to reduce the unnecessary power consumption when the traffic load is low. Compared with the IQ switches or other types of switches, the CICQ switch has some exclusive advantages when using the DVFS scheme. First, the CICQ switches possess the close-to-optimal delay performance with just very simple scheduling algorithms [7]. Thus, even we reduce the switching frequency; they can still provide better delay performance than IQ switches do. The performance can be further improved by using larger crosspoint buffers (CPBs) if necessary. Second, the scheduling is distributed and no central controller is required. Thus, the implementation complexity of the scheduler and the power controller is lower. Finally, the frequency scaling of each port is independent. Thus, the ports can adjust the frequencies individually, when the traffic is unbalanced among the ports. Moreover, the input and output frequency scaling is also independent. Therefore, the CPBs can work in different clock domains to allow the low-complexity fine-grain power control. These two features make the fine-grain DVFS scheme possible. However, the crossbar and buffered crossbar use different devices internally, thus, their power consumption is not directly comparable. And we are not comparing the power consumption of these two types of switches in this paper; instead, we show that the power control in the CICQ switches are finer-grained, simpler and more effective than in IQ switches.

There are static and dynamic power consumptions in a buffered crossbar. Reducing the static power consumption can be done at the circuit level, which is out of the scope of

This research is partially supported by HKUST Research Grants Council (RGC) under the grant number 613113.

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

this paper. In this work, we try to reduce the *dynamic power consumption* by adjusting the working frequencies of the CPBs in a buffered crossbar, according to the estimated traffic loads. This power reduction is at the cost of the increased packet delay, but without sacrificing throughput. However, with the appropriate control, we maintain the delay performance at a satisfactory level.

The rest of this paper is organized as follows. Section II briefly summarizes the related work on the power control of packet switches. Section III introduces the system model for the power control of the CICQ switch. Section IV proposes the fine-grain power control. In Section V, we evaluate the performance of the proposed power control through simulations. Section VI draws conclusions.

II. Related Works

Several previous efforts have been devoted to applying the DVFS scheme to IQ switches. Reference [8] assumes *linear* power and convex delay costs, and then formulates the scheduling problem in terms of a constrained, convex program. In [9], the authors extend the linear model to the *quadratic* energy and backlog costs and then employ the theory of linear-quadratic controls to derive optimal service levels for power controls in a switch. However, the methods in these two references are only suitable for specific power and delay cost models. Reference [10] takes cost input data in a tabular form, so that no specific cost model is assumed. However, the scheduler should calculate the transmission rate in a packet basis by solving a complex dynamic programming problem, which is infeasible for high-speed or large-scale switches. Moreover, the packet-level voltage and rate adjustment incurs a non-negligible overhead due to the reset time. Reference [11] considers an ideal switch model where the traffic matrix is known in advance, and uses the dynamic programming to calculate the optimal frequencies of the crosspoints to minimize the power consumption while still retaining the maximum throughput. However, the work is not practical and does not take delay into consideration. In [12], the authors improve the previous work to consider delay performance. To reduce the complexity, the authors use a periodic chip-level single voltage/frequency adjustment. However, that adjustment is not optimal when the traffic is unbalanced among the different ports.

The problem of the existing works is that they all concentrated on applying the DVFS scheme to the IQ switches. The strict constraints on IQ scheduling make the DVFS scheme either very complex, if each corsspoint should use a different voltage control, or not effective on power optimization, if only a single voltage/frequency is used for the whole crossbar chip. In this paper, we present the work that applies the DVFS scheme to CICQ switches, which is based on the buffered crossbar. By taking advantage of the flexible features of the buffered crossbar, the DVFS scheme can be made finer-grained, simpler and more effective.

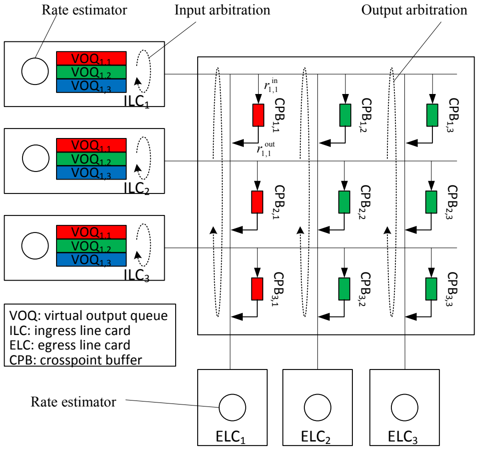

Fig. 1. The architecture for power control in a CICQ switch

III. System model

A. The switch architecture

We consider a packet switch that handles only fixed-size packets internally. The variable-size packets are first segmented into fixed-size cells at the input ports, and then reassembled at the output ports. An $N \times N$ CICQ switch uses a buffered crossbar with N input ports and N output ports as its switch fabric. The buffered crossbar is a variant of the crossbar by adding a small crosspoint buffer (CPB) at each crosspoint, which can store at most B cells. Most of the arrived cells are stored in the buffers in ingress line cards (ILCs). The input buffers are organized into virtual output queues (VOQs), each temporarily storing the cells from the input port to a distinct output port. Shown in Fig. 1 is a 3×3 CICQ switch, which has 9 CPBs. The scheduling in the CICQ switch can be divided into the input arbitration and the output arbitration. In each scheduling cycle, the input arbiter chooses a VOQ whose corresponding CPB is not full; while the output arbiter chooses a non-empty CPB. A scheduling cycle is one time slot, if the working frequency is not scaled.

Let $a_{i,j}(n)$ be the cumulative number of packets that have arrived from input port i and destined for the output port j during time slots [0, n], with $a_{i,j}(0) = 0$. Suppose that the arrival process $\{a_{i,j}(\cdot), i,j = 1,2,\cdots,N\}$ obeys the strong law of large numbers (SLLN), i.e., with probability 1, $\lim_{n\to\infty} \frac{a_{i,j}(n)}{n} = \lambda_{i,j}$, where $i,j = 1,2,\cdots,N$. $\lambda_{i,j}$ is then called the arrival rate at VOQ$_{i,j}$, and the matrix $\Lambda = [\lambda_{i,j}]_{N\times N}$ is usually named the arrival rate matrix or traffic matrix. Traffic matrix Λ is said to be *admissible*, if it satisfies

$$\lambda_{i,\Sigma} = \sum_{j=1}^{N} \lambda_{i,j} \le 1, \forall i; \lambda_{\Sigma,j} = \sum_{i=1}^{N} \lambda_{i,j} \le 1, \forall j.$$

An exclusive feature of the buffered crossbar is that the input arbitration and output arbitration can work with different rates. For example, in Fig. 1, the rate of writing packets into CPB$_{1,1}$ is $r_{1,1}^{\text{in}}$, while the rate of reading packets from CPB$_{1,1}$ is $r_{1,1}^{\text{out}}$. Furthermore, each input and output port can work on its own frequency according to its traffic load. This feature is vitally important, which allows us to have a fine-grain control of the voltage and frequency of CPBs to optimize the power consumption.

B. The power consumption in a buffered crossbar

It is shown in [11, 13] that the on-chip dynamic power consumption when transferring packets (actually bit streams) across the CMOS gates at the crosspoint (i, j) of a switch fabric is proportional to $r_{i,j}V_{i,j}^2$, where $V_{i,j}$ is the operating voltage supplied to crosspoint (i, j) and the $r_{i,j}$ is the packet transferring speed. However, in buffered crossbars, a packet should cross two levels of gates in the fabric; one is for writing the packet into the CPB, and the other is for reading the packet out. Thus, the dynamic power consumption of the crosspoint (i, j) in a buffered crossbar can be divided into two parts: input power $P_{i,j}^{\text{in}} \propto r_{i,j}^{\text{in}}{V_{i,j}^{\text{in}}}^2$ and output power $P_{i,j}^{\text{out}} \propto r_{i,j}^{\text{out}}{V_{i,j}^{\text{out}}}^2$.

It has been stated clearly in [14] that due to the delay needed to switch from one logic state to another, the allowed operating rate $r_{i,j}$ is proportional to the supplied operating voltage $V_{i,j}$. However, in most of the current switch designs, the COMS gates of the crosspoints always run at their maximum rate, $r_{\max}$, which is often normalized to one packet per time slot. This also requires the maximum operating voltage, $V_{\max}$. With the DVFS technique, we can jointly reduce $V_{i,j}$ and $r_{i,j}$, when the offered traffic load through the crosspoint (i, j) is low, to save the unnecessary power. Reference [11] defined an expansion factor, $\alpha_{i,j}$, to describe the voltage reduction, i.e., $\alpha_{i,j} = \frac{V_{\max}}{V_{i,j}}$. Since $r_{i,j} \propto V_{i,j}$, the packet transmission speed through crosspoint (i, j) is also slowed down by a factor of $\alpha_{i,j}$, i.e., $r_{i,j} = \frac{r_{\max}}{\alpha_{i,j}}$. In other words, the packet transmission time through crosspoint (i, j) is prolonged by a factor of $\alpha_{i,j}$, or the traffic load through the crosspoint is increased from $\lambda_{i,j}$ to $\alpha_{i,j}\lambda_{i,j}$ However, constrained by the technology, the operating voltage $V_{i,j}$ could not be too low. In the extreme cases, the maximum expansion factor could be $\alpha_{\max} = 3$ [6]. Thus, the DVFS scheme is constrained by $1 \leq \alpha_{i,j} \leq 3$.

For a buffered crossbar-based switch, under some traffic load Λ, to avoid overload, it is necessary to limit the expansion factors at the crosspoints so that

$$\sum_{k=1}^{N} \alpha_{i,k}^{\text{in}}\lambda_{i,k} \leq 1, \forall i; \sum_{k=1}^{N} \alpha_{k,j}^{\text{out}}\lambda_{k,j} \leq 1, \forall j.$$

The power consumptions of CPB$_{i,j}$ of a buffered crossbar are

$$P_{i,j}^{\text{in}} \propto \lambda_{i,j}{V_{i,j}^{\text{in}}}^2 = \lambda_{i,j}\left(\frac{V_{\max}}{\alpha_{i,j}^{\text{in}}}\right)^2 \propto \frac{\lambda_{i,j}}{{\alpha_{i,j}^{\text{in}}}^2},$$

$$P_{i,j}^{\text{out}} \propto \lambda_{i,j}V_{i,j}^{\text{out}2} = \lambda_{i,j}\left(\frac{V_{\max}}{\alpha_{i,j}^{\text{out}}}\right)^2 \propto \frac{\lambda_{i,j}}{\alpha_{i,j}^{\text{out}\,2}}.$$

Then, the total power consumption of the buffered crossbar

$$P^{\text{in}} = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{i,j}^{\text{in}} \propto f^{\text{in}}(\alpha^{\text{in}}) = \sum_{i=1}^{N}\sum_{i=1}^{N} \frac{\lambda_{i,j}}{{\alpha_{i,j}^{\text{in}}}^2},$$

$$P^{\text{out}} = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{i,j}^{\text{out}} \propto f^{\text{out}}(\alpha^{\text{out}}) = \sum_{i=1}^{N}\sum_{i=1}^{N} \frac{\lambda_{i,j}}{\alpha_{i,j}^{\text{out}\,2}}.$$

Since the input arbitration and output arbitration are independent in a buffered crossbar, the problem of minimizing the power consumption becomes that given the traffic Λ, find feasible $\alpha^{\text{in}} = [\alpha_{i,j}^{\text{in}}]_{N\times N}$ and $\alpha^{\text{out}} = [\alpha_{i,j}^{\text{out}}]_{N\times N}$ that can minimize $f^{\text{in}}(\alpha^{\text{in}})$ and $f^{\text{out}}(\alpha^{\text{out}})$, respectively, i.e.,

$$\min_{\alpha_{i,j}^{\text{in}}} \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\lambda_{i,j}}{\alpha_{i,j}^{in\,2}}, \text{ subject to}$$

$$\begin{cases} \sum_{j=1}^{N} \alpha_{i,j}^{\text{in}}\lambda_{i,j} \leq 1, \forall i = 1, 2, \cdots, N; \\ 1 \leq \alpha_{i,j}^{\text{in}} \leq 3; \end{cases}$$

$$\min_{\alpha_{i,j}^{\text{out}}} \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\lambda_{i,j}}{\alpha_{i,j}^{out\,2}}, \text{ subject to}$$

$$\begin{cases} \sum_{i=1}^{N} \alpha_{i,j}^{\text{out}}\lambda_{i,j} \leq 1, \forall j = 1, 2, \cdots, N; \\ 1 \leq \alpha_{i,j}^{\text{out}} \leq 3. \end{cases}$$

IV. Fine-grain power control for CICQ switches

The basic idea of DVFS is to reduce the packet transmission rate of each crosspoint when the traffic load is low. However, the changing of the voltage and transmission rate takes time. When the transition is going on, the crosspoint will experience a temporary reset time t_{reset}, during which no packet can be transmitted. If we consider the on-chip voltage regulator, then the temporary reset time can be controlled in about several tens of nanoseconds [3]. Given the 10 Gbps line rate and 64 bytes packet size, each time slot is about 51.2 ns. If we adjust the voltage every time slot, then the voltage transition overhead will be too large. The problem can be solved by enlarging the voltage adjustment period, w, to $w \gg t_{reset}$ For example, when $w = 1000$ time slots, the adjustment period will be about 50 μs, which makes the reset time negligible. Furthermore, we only adjust the voltage between packet transmissions in order not to interrupt the transmitting packets. Thanks to the independent feature of the buffered crossbar, which makes the fine-grain adjustment easy to implement. We estimate traffic rate periodically, and adjust the voltage and transmission rates of the CPBs according to the estimated traffic rates in the previous period.

The power consumption of the buffered crossbar is measured during each voltage adjustment period, w. We first measure the actual rates of writing packets into CPBs and reading packets out of CPBs during the k-th period as $\hat{r}_{i,j}^{\text{in}}(k)$ and $\hat{r}_{i,j}^{\text{out}}(k)$, respectively. Then, the power consumed in the period k is calculated as

$$P(k) = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\hat{r}_{i,j}^{\text{in}}(k)}{{\alpha_{i,j}^{\text{in}}}^2(k)} + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\hat{r}_{i,j}^{\text{out}}(k)}{\alpha_{i,j}^{\text{out}\,2}(k)},$$

The average power consumption after m periods is

$$\bar{P}(m) = \frac{1}{m}\sum_{k=1}^{m} P(k).$$

Although we could adjust the voltage/frequency of each crosspoint individually, it is complex to implement and a large number of regulators are required. Instead, to simplify the voltage adjustment, we use a port-level adjustment. To adjust the expansion factor of each input port and each output port independently, we measure the arrival rate to each input port and to each output port during period k as $\hat{\rho}_{i,\Sigma}(k)$ and $\hat{\rho}_{\Sigma,j}(k)$, respectively. However, these rates are measured during a single period. If the traffic rate varies frequently, then the instant rates will be inaccurate. We use the widely applied exponential weighted moving average (EWMA) [15] to estimate the smooth average traffic rates to input ports and output ports:

$$\hat{\lambda}_{i,\Sigma}(k) = \beta\hat{\rho}_{i,\Sigma}(k) + (1-\beta)\hat{\lambda}_{i,\Sigma}(k-1);$$

$$\hat{\lambda}_{\Sigma,j}(k) = \beta\hat{\rho}_{\Sigma,j}(k) + (1-\beta)\hat{\lambda}_{\Sigma,j}(k-1),$$

where $0 < \beta < 1$ is usually called the *weight factor* and should be small enough to filer the noises introduced by abrupt traffic changes during only a few periods. These estimated arrival rates are then used to adjust the voltages and the transmission rates of the crosspoints in the next period.

We choose an appropriate $\alpha_i^{\text{in}}(k)$ for each input port and $\alpha_j^{\text{out}}(k)$ for each output port according to the estimated traffic rates. Then, all the CPBs connected to input port i share the same expansion factor $\alpha_i^{\text{in}}(k)$ during period k when writing packets into the CPBs. Similarly, all the CPBs connected to output port j share the same expansion factor $\alpha_j^{\text{out}}(k)$ during period k when reading packets out of the CPBs.

It is shown that the expansion factor for a single queue, $\alpha = \frac{1}{\lambda}$ can achieve the maximum throughput [16], where λ is the arrival rate to the queue and α is the expansion factor. This result is verified in [11] for IQ switches. However, if we always choose such an expansion factor, then the queue length will grow infinitely, leading to large delay. In [16], the authors proposed a "fixed utilization" for a single queue system to control the backlog. In this paper, we adopt a similar idea to set a "safety margin" for the expansion factor, i.e., $\alpha = \frac{s}{\lambda}$, where s is the *safety margin factor*. To ensure that $\alpha \le \alpha_{\max}$, we set a minimum threshold, $\lambda_{\text{min}} < \frac{s}{\alpha_{\max}}$. When the measured load $\hat{\lambda} < \lambda_{\text{min}}$, we always choose $\alpha_{\max}$. On the other hand, when the traffic is high, the expansion factor might be less than 1, i.e., $\frac{s}{\lambda} < 1$, which this is not possible in reality. In this case, we set a maximum threshold, $\lambda_{\max} \le s$. When $\hat{\lambda} > \lambda_{\max}$,we set $\alpha = 1$. And the constant safety margin DVFS algorithm is shown in Alg. 1 and Alg. 2 for the input ports and output ports, respectively.

Algorithm 1 DVFS for input transmission in period k

```
for each input port $i$:
  if duration $w$ has passed since $(k-1)$-th period
  && the current packet has finished transmission:
    Estimate the traffic arrival rate to input port $i$, $\hat{\lambda}_{i,\Sigma}(k)$.
    if $\hat{\lambda}_{i,\Sigma}(k) < \lambda_{\text{min}}$:
      $\alpha^{\text{in}}_{i,j}(k) = \alpha_{\max}, \forall j = 1, 2, \cdots, N$;
    else if $\lambda_{\text{min}} \le \hat{\lambda}_{i,\Sigma}(k) < \lambda_{\max}$:
      $\alpha^{\text{in}}_{i,j}(k) = \frac{s}{\hat{\lambda}_{i,\Sigma}(k)}, \forall j = 1, 2, \cdots, N$;
    else
      $\alpha^{\text{in}}_{i,j}(k) = 1, \forall j = 1, 2, \cdots, N$.
  Adjust the input transmission rate and voltage of the CPBs
  connected to input port $i$ during period $k$ according to the
  $\alpha^{\text{in}}_{i,j}(k)$.
```

Algorithm 2 DVFS for output transmission in period k

```
for each output port $j$:
  if duration $w$ has passed since $(k-1)$-th period
  && the current packet has finished transmission:
    Estimate the traffic arrival rate to output port $j$, $\hat{\lambda}_{\Sigma,j}(k)$.
    if $\hat{\lambda}_{\Sigma,j}(k) < \lambda_{\text{min}}$:
      $\alpha^{\text{out}}_{i,j}(k) = \alpha_{\max}, \forall i = 1, 2, \cdots, N$;
    else if $\lambda_{\text{min}} \le \hat{\lambda}_{\Sigma,j}(k) < \lambda_{\max}$:
      $\alpha^{\text{out}}_{i,j}(k) = \frac{s}{\hat{\lambda}_{\Sigma(k),j}}, \forall i = 1, 2, \cdots, N$;
    else
      $\alpha^{\text{out}}_{i,j}(k) = 1, \forall i = 1, 2, \cdots, N$.
    Adjust the output transmission rate and voltage of the
  CPBs connected to output port $j$ during period $k$ according
  to the $\alpha^{\text{out}}_{i,j}(k)$.
```

Through simulations, as shown in Section V, we found that when the offered load is between λ_{min} and $\lambda_{\max}$, the average delay decreases as the traffic load increases. However, if the DVFS is not used, the average delay should increase as the load increases. That means that the power-saving scheme is not efficient; otherwise, we could have saved more power by sacrificing a little bit more delay performance. Thus, we fix the expansion factor adjustment equation a little. Instead of using a constant safety factor, s, we adopt a *dynamic safety factor*, which increases proportionally as the traffic load increases:

$$s(k) = g + (1-g)\frac{\hat{\lambda}(k) - \lambda_{\text{min}}}{\lambda_{\max} - \lambda_{\text{min}}},$$

where $g \le \lambda_{\max}$ is the *initial safety factor*. The final safety factor, $s(k)$, is now increasing slowly as the load increases, and the amount it increased is proportional to the difference between the current estimated load and the minimum threshold. This makes the rate and voltage adjustment smoother. In the next section, we will show that the delay between λ_{min} and $\lambda_{\max}$ now stays at almost the same level while the offered load increases, which is reasonable for real uses; however, it leads to higher power efficiency.

V. Performance Evaluation

In this section, we evaluate the effectiveness of the DVFS scheme in a CICQ switch using our switch simulation platform built on network simulator 3 (NS-3) [17]. The DVFS scheme can be used with any proposed CICQ scheduling algorithm. In the simulations, we choose the LQF-RR algorithm [18] because of its simplicity and good performance. Note that even with the simplest round-robin algorithms [7], the CICQ switch still has better delay performance than the IQ switch; however, due to the limited space, will only show the results with the LQF-RR algorithm.

We evaluate the stability of the DVFS scheme under different traffic patterns, and also the power-saving performance under different scenarios. In the simulations, we make the switch size $N = 32$ and the crosspoint buffer size $B = 1$, if not explicitly mentioned. The measurement and adjustment period is $w = 1000$ time slots and the weight factor is $\beta = 0.15$.

In the first simulation, we evaluate the delay performance of CICQ switches using constant safety margin DVFS, and compare the performance to that of an IQ switch with the same DVFS scheme. We set the constant safety factor $s = 0.8$. According to the constraints, we set the minimum threshold $\lambda_{\text{min}} = 0.25$, to make sure that $\alpha_{i,j}(k) \le 3$, and the maximum threshold $\lambda_{\max} = 0.8$, to make sure that $\alpha_{i,j}(k) \ge 1$. The simulation results are shown in Fig. 2. As we can see, the CICQ switch, which uses the buffered crossbar, has better delay

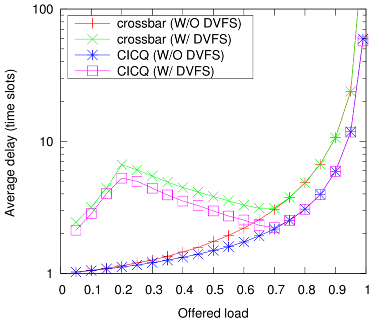

Fig. 2. Delay of the IQ and CICQ switches using constant factor DVFS

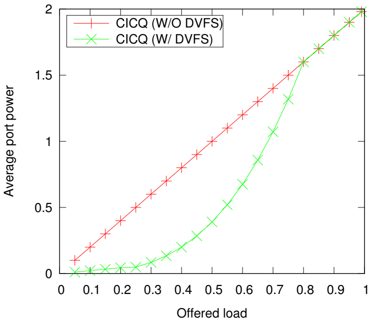

Fig. 3. Power consumption of the CICQ switch using constant factor DVFS

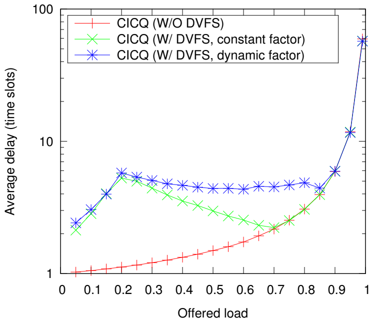

Fig. 4. Delay of the CICQ switch using dynamic factor DVFS

performance than the crossbar-based IQ switch in both cases with and without DVFS. Fig. 3 shows the normalized average port power consumption of the CICQ switch. Since each CPB will be written once and read once, which all consume power, the maximum average port power consumption, which is a sum of the input power and the output power, is normalized to 2 instead of 1. The power consumption of the IQ switch is not shown because it uses a different model and is not directly comparable. As the figure shows, when DVFS is not used, the power consumption is proportional to the traffic rate; however, when DVFS is used, the reduction in power consumption is obvious when the traffic load is lower than 0.8, which is the maximum threshold. When the traffic load is higher than 0.8, the switch uses the maximum transmission rate to guarantee the best delay performance, thus, no power is saved.

As we have described, when using the constant safety margin, the delay between λ_{min} and λ_{max} is decreasing as the load increases, which is not necessary and also not power efficient. We proposed dynamic safety margin to adjust the expansion factor, according to the traffic load, to save more power. For the dynamic safety margin scheme, we set the initial safety factor $g = 0.8$. The minimum threshold is the same as in the constant case $\lambda_{\text{min}} = 0.25$, but the maximum threshold is $\lambda_{\text{max}} = 0.95$. Note that λ_{max} is higher than the constant case without affecting λ_{min}. The delay performance of constant safety margin and dynamic safety margin DVFS is shown in Fig. 4. When the load is between λ_{min} and λ_{max}, the delay performance of the dynamic scheme is almost stable at the same level, which is reasonable in reality. As shown in Fig. 5, the power consumption of the dynamic scheme is much lower than the constant scheme, especially when the load is high.

One drawback of the IQ-based DVFS scheme is that the adjustment of the voltage/frequency for individual crosspoint is complex. Thus, most of the practical schemes adopt the chip-level single voltage adjustment [12], which covers all the crosspoints. As a result, in the case that only a few ports are heavily loaded, they will affect the rate adjustment of other lightly loaded ports. In this simulation, we compare the power consumption of the chip-level and the port-level DVFS in CICQ switches. We always feed one of the ports with load 75%, and let the offered loads to other ports vary from 0$\sim$99%. The simulation results are shown in Fig. 6. As we can see, with the chip-level DVFS, all the underloaded crosspoints have to run at the same rate as the heavily loaded crosspoints, thus, the power consumption is still large, which wasted a lot of unnecessary power. However, when the port-level DVFS is used, the ports can have a finer-grain control of voltages and rates of crosspoints individually, according to their estimated loads. At the same time, the CPB can work in two different clock domains when writing and reading packets. As a result, the power consumption is greatly controlled, and the effectiveness is obvious.

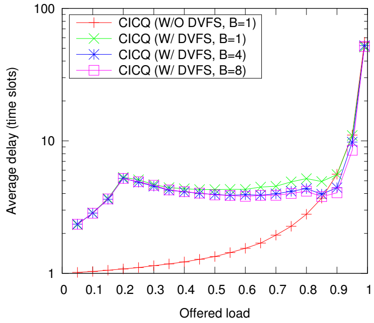

Fig. 8. Delay of the CICQ switch under hotspot traffic

In the above simulations, we only use the uniform traffic distribution, which is ideal. In the next simulations, we use two widely used nonuniform traffic, diagonal and hotspot [19] to evaluate the stability and performance of the dynamic DVFS scheme.

Fig. 7 shows the delay performance of the CICQ switch with and without the dynamic DVFS under diagonal traffic. Similar to the uniform traffic case, the delay increase as the traffic load increases. When the load is between 0.2 and 0.95, the delay is well controlled by the dynamic safety margin. When the load is higher than 0.95, the DVFS is not used. In this simulation, we also evaluate the delay performance using different CPB sizes. As the results show, the delay performance can be improved when larger CPB sizes are used. Fig. 8 shows the case under hotspot traffic. We still have the similar results as in the diagonal case. The dynamic DVFS scheme still has good control of the delay performance, and the switch is stable under all offered loads. When we increase the CPB size, the delay can be improved, but not so obvious as in the previous case. The power consumption in these two cases is very close to the uniform case, thus, we will not list the results here due

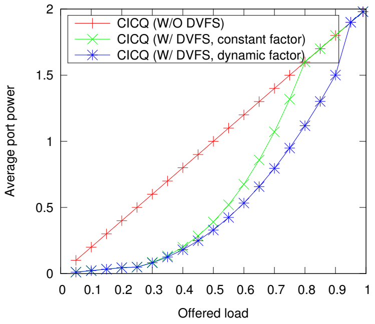

Fig. 5. Power consumption of the CICQ switch using dynamic factor DVFS

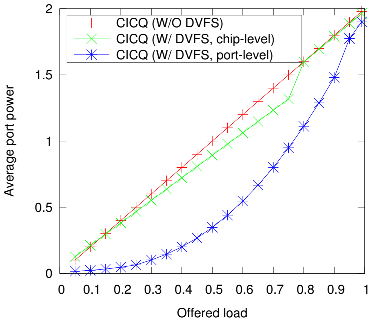

Fig. 6. Power consumption of the CICQ switch when loads are not balanced

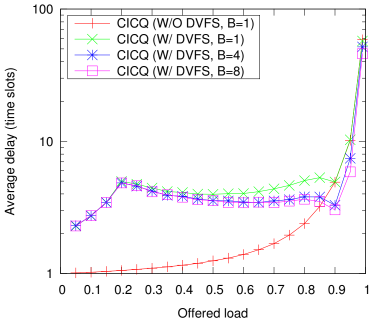

Fig. 7. Delay of the CICQ switch under diagonal traffic

to the limited space.

VI. Conclusion

In this paper, we proposed the dynamic voltage and frequency scaling (DVFS) scheme for the combined input-crosspoint queued (CICQ) packet switch, which uses the buffered crossbar as its switch fabric, to reduce the dynamic power consumption, when the traffic load is low. This reduction in power consumption is at the cost of the increased delay. As far as we know, this is the first work to apply DVFS in the buffered crossbar. The advantages of using buffered crossbar are as follows: First, it has better delay performance than crossbars, and its delay performance can be improved by using larger crosspoint buffers (CPBs). Second, the scheduling can be distributed. Finally, each port can independently adjust their own voltage and rate without incurring high computing complexity.

The proposed DVFS scheme does not need to know the traffic matrix in advance; instead, it estimates the traffic rate periodically. The port-level DVFS is used to simplify the implementation. Each port adjusts the voltages and working frequencies of the CPBs connected to it. To make sure that the throughput of the CICQ switch using DVFS is maximized, the expansion factor of the voltage and frequency is calculated as the reciprocal of the estimated traffic rate. To further guarantee the delay performance and still maintain the power efficiency, we proposed a dynamic safety margin DVFS scheme, which uses a dynamic safety factor to control the delay and power consumption. Finally, we evaluate the delay and power consumption of the CICQ switch with the DVFS scheme. The results show that the dynamic DVFS scheme can effectively reduce the power consumption in the buffered crossbar while still providing satisfactory delay performance. The simulation results also showed that when the traffic loads are not balanced among the ports, the buffered crossbar allows the port-level control to save more power than the chip-level control, which is usually used in crossbar-based switches. The scheme is also shown to be stable under different nonuniform traffic loads.

References

[1] M. Gupta and S. Singh, “Greening of the internet,” in *Proc. ACM SIGCOM*, 2003, pp. 19–26.

[2] “Power and cooling in the data center,” AMD, Tech. Rep. 34146A-PC-WP-en, 2005.

[3] L. Benini, A. Bogliolo, and G. De Micheli, “A survey of design techniques for system-level dynamic power management,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 299–316, 2000.

[4] A. Wierman, L. L. Andrew, and A. Tang, “Power-aware speed scaling in processor sharing systems,” in *Proc. IEEE INFOCOM*, 2009, pp. 2007–2015.

[5] J. Guichard, F. L. Faucheur, and J.-P. Vasseur, *Definitive MPLS network designs*. Cisco Press, 2005.

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

[7] R. Rojas-Cessa, E. Oki, and H. J. Chao, “On the combined input-crosspoint buffered switch with round-robin arbitration,” *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1945–1951, 2005.

[8] N. Bambos and D. O Neill, “Power management of packet switch architectures with speed modes,” in *Proc. Allerton Conf. Comm., Control and Computing*, vol. 41, no. 1, 2003, pp. 571–580.

[9] L. Mastroleon, D. C. O’Neill, B. Yolken, and N. Bambos, “Power and delay aware management of packet switches,” *IEEE Transactions on Computers*, vol. 61, no. 12, pp. 1789–1799, 2012.

[10] M. Valdez-Vivas, N. Bambos, and D. O’Neill, “Delay-sensitive power management for packet switches,” in *Proc. IEEE ICC*, 2013, pp. 4443–4448.

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

[12] A. Bianco, P. Giaccone, and M. Ricca, “Dynamic voltage and frequency scaling control for crossbars in input-queued switches,” in *Proc. IEEE ICC*, 2014.

[13] H.-S. Wang, L.-S. Peh, and S. Malik, “A power model for routers: Modeling alpha 21364 and infiniband routers,” in *Proc. IEEE High Performance Interconnects*, 2002, pp. 21–27.

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

[15] S. Floyd and V. Jacobson, “Link-sharing and resource management models for packet networks,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 4, pp. 365–386, 1995.

[16] A. Bianco, M. R. Casu, P. Giaccone, and M. Ricca, “Joint delay and power control in single-server queueing systems,” *IEEE OnlineGreenComm*, 2013.

[17] Y. Xia, H. Zeng, and Z. Shen, “Design and implementation of switch module for ns-3,” in *Proc. ICST ValutTools*, 2009, p. 3.

[18] T. Javidi, R. Magill, and T. Hrabik, “A high-throughput scheduling algorithm for a buffered crossbar switch fabric,” in *Proc. IEEE ICC*, vol. 5, 2001, pp. 1586–1591.

[19] Y. Shen, S. S. Panwar, and H. J. Chao, “Design and performance analysis of a practical load-balanced switch,” *IEEE Transactions on Communications*, vol. 57, no. 8, pp. 2420–2429, 2009.