Dynamic Power Saving in Fat-Tree Interconnection Networks Using On/Off Links[*]

Marina Alonso[1], Salvador Coll[2], Juan-Miguel Martínez[1],
Vicente Santonja[1], Pedro López[1], José Duato[1]

[1]Universidad Politécnica de Valencia
Dept. of Computer Engineering
Camino de Vera s/n
46022 Valencia, SPAIN
malonso@disca.upv.es

[2]Universidad Politécnica de Valencia
Dept. of Electronic Engineering
Camino de Vera s/n
46022 Valencia, SPAIN
scoll@eln.upv.es

Abstract

Current trends in high-performance parallel computers show that fat-tree interconnection networks are one of the most popular topologies. The particular characteristics of this topology, that provide multiple alternative paths for each source/destination pair, make it an excellent candidate for applying power consumption reduction techniques. Such techniques are being increasingly applied in computer systems and the interconnection network is not an exception, since its contribution to the system power budget is not negligible. In this paper, we present a mechanism that dynamically switches on and off network links as a function of traffic. The mechanism is designed to guarantee network connectivity, according to the underlying routing algorithm. In this way, the default routing algorithm can be used regardless of the power saving actions taken, thus simplifying router design. Our simulation results show that significant network power consumption reductions can be obtained at no cost. Latency remains the same although the number of operating network links is dynamically adjusted.

1 Introduction and Motivation

Nowadays a growing number of high performance computing systems are clusters. This is the chosen architecture of 360 out of 500 systems listed in the November 2005 edition of the Top500 Supercomputers sites [1]. This represents 72% of the computers in the list (one year before, only 58.8% of the systems in the list were clusters). A significant fraction of these clusters use fat-tree interconnection networks due to its high bisection bandwidth and ease of application mapping for arbitrary communication topologies [9]. But most applications have communication topology requirements that are far less than the total connectivity provided by fat-trees. Vetter and Mueller show that applications that scale most efficiently to large numbers of processors use point-to-point communications patterns where the average number of distinct destinations is relatively small [16]. This provides strong evidence that many application communication topologies exercise a small fraction of the resources provided by fat-trees [10]. Moreover, traffic in an interconnection network exhibits high spatial and temporal variance, giving rise to inactivity periods at several links in the network [14]. Thus, there is a chance to reduce power consumption by dynamically switching on/off links based on their traffic while running a set of applications. These power reduction techniques are relevant if we take into account that the interconnection network consumes an important part of a cluster total power consumption. For example, the routers and links in a Mellanox server blade, consume almost the same power as that of a processor (15W), and is about 37% of the total power budget [7].

Several power reduction techniques have been proposed for interconnection networks. Most of them are based on Dynamic Voltage Scaling (DVS). DVS was originally proposed, and now is widely deployed, for microprocessors. When applied to networks, this approach allows DVS links to work in a discrete range of frequencies and supply voltages, which leads to different levels of power consumption in response to their traffic utilization. For example, Soteriou, Eisley and Peh propose a DVS model in which links can take ten discrete frequency levels and corresponding voltage levels [12]. The history-based DVS policy pro-

[*]This work was supported by Spanish MCYT under Grant TIC2003-08154-C06-01.

1-4244-0054-6/06/$20.00

poses to use past network utilization to predict future traffic, therefore tuning dynamically link frequency and voltage to reduce network power consumption [11]. Chen et al. use DVS policies in optoelectronic links [4]. Stine and Carter compare DVS with the use of adaptive routing in non DVS links, showing that, as long as the network provides enough bandwidth to meet the needs of the application, an adaptively-routed network can improve latency with the same power consumption [15]. The drawback of DVS is that it requires a sophisticated hardware mechanism to ensure correct link operation during scaling. Moreover, DVS links continue to consume power even while idle.

Other techniques are based on the use of on/off links that are selectively switched on and off according to their utilization [7, 13, 14]. In order to avoid deadlocks, adaptive routing algorithms must be used. Kim et al. also investigate hybrid techniques based on both DVS and on/off links. The idea is to shut down DVS links when traffic drops to very low levels [7].

Following a different approach, the DALW (Dynamically Adjusting Link Width) algorithm adjusts link width according to network load [2]. As it never completely switches off links, the same routing algorithm can be used regardless of the power consumption level, which simplifies the router design. Its main drawback is that average packet latency is increased for low network loads due to the bandwidth reduction derived from thinner links. A different approach is proposed for interconnection networks that use regular topologies based on high-degree switches [3]. The power saving mechanism is based on aggregating the links into trunk links and dynamically turning on and off network links as a function of their traffic. In this case the latency penalty for low loads is significantly lower than in DALW since link bandwidth is not reduced.

In this paper, we present a new method to reduce power consumption in fat-trees based on the use of on/off links.

The rest of the paper is organized as follows. Section 2 briefly introduces the formal aspects of k-ary n-trees, a particular class of fat-trees, together with its routing basis. The proposed power saving mechanism is described in Section 3. A simulation-based evaluation is presented in Section 4 and, finally, some conclusions are drawn in Section 5.

2 Fat-trees

A k-ary n-tree is composed of $N = k^n$ processing nodes and $S = nk^{n-1}$ $k \times k$ switches. Processing nodes are identified by $\langle p_0, p_1, \ldots p_{n-1}\rangle$ where $p_i \in \{0, 1, \ldots, k-1\}$ for $0 \leq i \leq n-1$, and each switch is identified by $\langle w_0, w_1, \ldots w_{n-2}, l\rangle$, where $w_i \in \{0, 1, \ldots, k-1\}$ for $0 \leq i \leq n-2$ and $l \in \{0, 1, \ldots, n-1\}$ is the level of the switch (0 is the root level).

- Two given switches, $\langle w_0, w_1, \ldots w_{n-2}, l\rangle$ and $\langle w'_0, w'_1, \ldots w'_{n-2}, l'\rangle$ are connected if and only if $l' = l + 1$ and $w_i = w'_i$ for all $i \neq l$. The link connecting both switches is labeled with w'_l on the level l switch and with $k + w_l$ on the l' switch.

  Thus, each switch has $2k$ outgoing links: k of these connected to the level $l + 1$ switches or processing nodes (down links) and the remaining k to the level $l - 1$ switches (up links).

- There is a link between the switch in the bottom level $\langle w_0, w_1, \ldots w_{n-2}, n-1\rangle$ and the processing node $\langle p_0, p_1, \ldots p_{n-1}\rangle$ if and only if $w_i = p_i$ $\forall i \in \{0 \ldots n-2\}$.

The labeling scheme shown in the previous definitions makes the k-ary n-tree a delta network: any path starting from a level 0 switch and leading to a given node $\langle p_0, p_1, \ldots, p_{n-1}\rangle$ traverses the same sequence of edge labels: $p_0, p_1, \ldots, p_{n-1}$ [6]. An example of such labeling is shown in Figure 1, for a quaternary fat-tree of dimension 3 (64-node network), that is, a 4-ary 3-tree.

Minimal routing between any pair of processing nodes can be accomplished by sending the message to one of the nearest common ancestor switches and from there to the destination. Hence, each message experiences two routing phases: an ascending phase, from the processing node to a nearest common ancestor, followed by a descending phase to the destination node. While the descending phase is necessarily deterministic, since there is a single path from a nearest common ancestor switch to the destination, there could be alternative routes to reach a nearest common ancestor. The availability of alternative routes makes it possible to randomly choose ascending links or even implementing an adaptive algorithm that makes a decision according to the local state of the switch, avoiding congested links. The latter option is used by the routing algorithm used by the Quadrics interconnection network, one of the most popular commercial fat-tree based networks [8].

3 Description of the Power Saving Mechanism

The basic idea behind the proposed mechanism is dynamically switching links on/off as a function of the required network throughput. In our model, we consider bidirectional links that can be turned on/off in a given direction, either ascending or descending. For doing so, every switch in the network measures outgoing traffic and controls the status of the outgoing links depending on traffic variations. A subset of network links, which is defined as the Minimal Tree, cannot be switched off in order to maintain the network connectivity. As stated below, this bounds the achievable level of power saving.

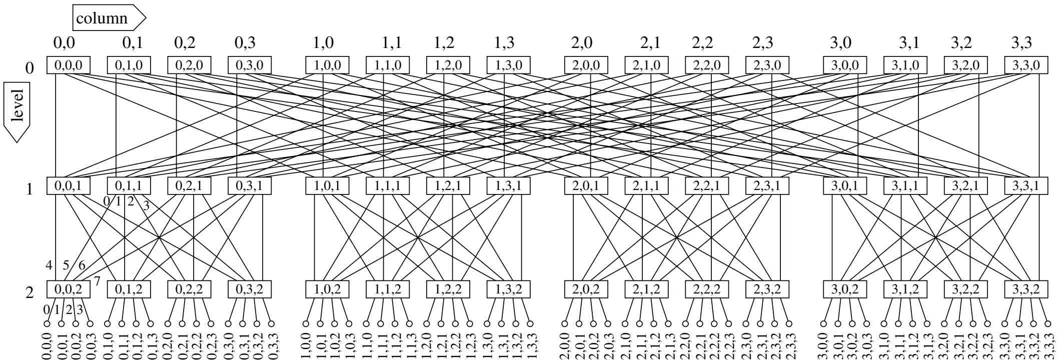


Figure 1. 4-ary 3-tree node, switch and edge labels.

Given a k-ary n-tree, the Minimal Tree (MT) is the subset of the k-ary n-tree composed of all the processing nodes, a subset of the communication switches, and the edges between them, as defined in Section 2.

A switch $\langle w_0, w_1, \ldots w_{n-2}, l\rangle$ belongs to the MT if one of the following properties holds:

1. $l < n-1$ and $w_i = 0 \ \ \forall i \in \{l, \ldots, n-2\}$.
2. $l = n-1$ (all the switches in the level $n-1$ belong to the MT).

The Minimal Tree of a quaternary fat tree of dimension 3 (4-ary 3-tree) is shown in Figure 2.

The MT is composed of the switches that cannot be turned off, as they provide the minimum paths needed to maintain all the processing nodes connected. Within these switches, all the down links, and the up link with index k also belong to the Minimal Tree. Thus, there are $k+1$ links in the MT per switch in the MT (except the root switch: only its k down links are in the MT). The links that connect the $N = k^n$ processing nodes with the switches also belong to the MT.

The number of switches in the MT is:

$$|MT_S| = k^{n-1} + k^{n-2} + \ldots + k + 1 = \frac{1-k^n}{1-k}$$

The number of (unidirectional) links in the MT is:

$$|MT_L| = (k+1)|MT_S| - 1 + k^n = 2k|MT_S|$$

Considering only the link power consumption, the minimum relative power consumption is given by the ratio between the number of links in the minimal tree and the total number or links in the fat-tree:

$$p_{\min} = \frac{|MT_L|}{2kS} = \frac{|MT_S|}{S}$$

For a 4-ary 3-tree, $|MT_S| = 21$, $|MT_L| = 168$, and $p_{\min} = 0.4375$

Another key issue of the proposed mechanism is that it allows the same routing algorithm to be used. In this way the implementation of the network switches is greatly simplified.

The power saving mechanism controls the network links state according to the following general rules:

- Up links: the utilization of the up links of a given switch is used to decide whether to turn on or turn off up links. This decision propagates upward to guarantee at least one path to level 0 switches (this is required to provide a path to every destination), and downward to guarantee descending routes to processors (for the same reason).
- Down links: these links are turned on/off all at once. Down links at a given switch are turned off when that switch cannot receive descending traffic, that is, when all its input links (from upper or lower levels) have been turned off. Those links will be turned on again when an input link is turned on.
- A switch can be moved to a standby state in the case that all the incoming links, both from the upper and lower levels, are inactive. In this case, the only functionality needed is detecting the activation of an incoming link. In this paper we do not consider the additional power reduction that can be obtained by moving the switches to the standby state.

Taking into account the above general considerations the behavior of a given switch will be conditioned by its level in the network.

Level $n-1$ switches. All the switches in this level belong to the Minimal Tree. For these switches, the utilization of each up link is measured. Periodically, the average utilization of all the up links in a switch, u_{up},

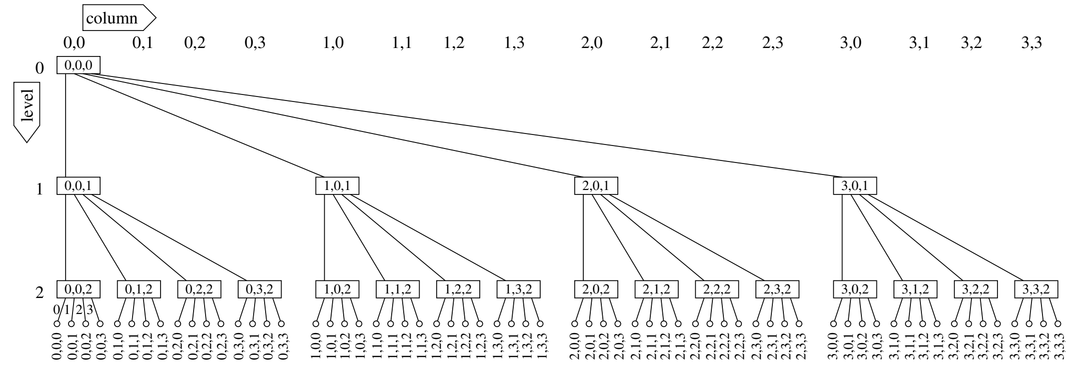


Figure 2. 4-ary 3-tree Minimal Tree.

is obtained. Two thresholds are defined: U_{off} is the turn off threshold, and U_{on} is the turn on threshold. If $u_{\text{up}} < U_{\text{off}}$ one of the up links is turned off. The first link to be turned off is $2k-1$, then $2k-2$, and so on. The link k cannot be turned off, as it belongs to the MT and provides the minimum connectivity with the upper level. When the load is low, this disconnection sequence concentrates the traffic in few switches at the upper level, favoring the power saving strategy.

When $u_{\text{up}} > U_{\text{on}}$, an inactive up link is turned on. The selected link is the one with lower index.

Down links in level $n-1$ cannot be turned off as they belong to the MT and provide connectivity to the processing nodes.

Level 1 to $n-2$ switches. If a switch belongs to the MT, then it performs exactly as described for the switches in level $n-1$. All the switches that do not belong to the MT use the following power saving mechanism.

When a switch detects that an input ascending link (connected to a down port at this level) has been turned off (initiates its transition from on to off), it turns off an up link. Specifically, when it detects that the link attached to port i is being turned off, it turns off the up link $i+k$. As it will receive less traffic from the lower level, it needs less bandwidth to the upper level. This policy achieves a balance between the incoming traffic from lower levels and the outgoing traffic to upper levels. If all the incoming links from the lower level are turned off, all the outgoing links to the upper level can be disconnected, as the switch will not receive ascending traffic.

When the switch detects that an incoming link from the lower level is being turned on (it begins its transition from off to on), the switch immediately initiates the turning on of the corresponding up link. Note that both processes can take place simultaneously. Moreover, the connection of the up link could trigger more reconnections in the upper levels.

Down links need a different approach as they do not have the redundancy of the up links. A message can not freely choose which down link to use in order to go to the lower level, instead it must use the appropriate down port (namely, a message addressed to a processing node $\langle p_0, p_1, \ldots, p_{n-1}\rangle$ must use the p_l port in the level l). While a switch has active incoming links from the upper or lower levels, all the down links must be in the on state, as they provide the necessary connectivity for the descendant messages.

When all the incoming links of the switch are turned off, additional power saving could be achieved by moving to the standby state and turning off different switch components (e.g., buffers). As soon as one incoming link initiates its activation process, the switch returns to the active state and all its down links are turned on to provide descending paths to the processor nodes. In the particular case the switch is activated by an ascending incoming link, in addition to all its down links one up link is turned on too. This is necessary to guarantee ascending messages can reach all the possible destinations.

Level 0 switches. Switches in this level do not have up links. Only one switch in this level, $\langle 0, 0, \ldots, 0\rangle$, belongs to the Minimal Tree. This is the root of the Minimal Tree. There is no need for any power saving mechanism in this switch because its output links will be always active. For the other switches in this level the following policy applies. When all the incoming links from the level 1 are turned off, all the down links can be disconnected, as the switch will not receive any traffic. In this case, the switches can be moved to a standby state, providing an additional reduction in power consumption.

The behavior of the system greatly relies on the pair of thresholds (U_{on} and U_{off}) used by the power saving mechanism. The effect of these two parameters can be analyzed in terms of two complementary aspects:

- Mechanism aggressiveness: this is controlled by the average value of the thresholds, U_{avg} ($U_{\text{avg}} = \frac{U_{\text{on}}+U_{\text{off}}}{2}$). High average thresholds provide an aggressive policy, since the mechanism keeps links disconnected even with high loads. On the other hand, if $U_{\text{avg}}$is low, a conservative policy is applied since power savings are only tried for low loads.

- Mechanism responsiveness: the hysteresis band, defined by the difference $U_{\text{on}} - U_{\text{off}}$, controls the mechanism responsiveness against traffic variations. Higher hysteresis bands will require higher traffic variations for the mechanism to be activated, while lower traffic variations will not modify the system status, and vice versa.

According to the above considerations, our proposal can be tuned to meet different requirements by adjusting aggressiveness and responsiveness. However, a number of limitations apply to the set of possible threshold values. First of all, U_{on} must be higher than U_{off}. In addition, it has to be guaranteed that $U_{\text{off}} > 0$. Finally, an additional limitation that depends on the particular system configuration must be taken into account. This constraint is conditioned by the fraction of up link bandwidth available for each switch state. We have performed an in-depth analysis of this effect for high-degree switches [3]. The conclusions we obtained can be easily translated to fat-tree networks. Every time an ascending link is turned off the load of the remaining active links increases. The highest increase in load is produced when moving from two ascending active links to one, when the load is multiplied by two. In order to avoid cyclic state transitions, U_{on} must be higher than or equal to $2 \cdot U_{\text{off}}$. In the worst case, the increase on average up link utilization when turning off up links will not be high enough to make the links go back to the previous state.

The use of constant thresholds, simplifies the implementation of the mechanism, as there is only necessary to compare with a unique pair of thresholds. Notice that this requires to measure the switch utilization (in the ascending direction) considering the actual bandwidth of the on links.

4 Performance Evaluation

In this section, we evaluate the power saving strategy proposed in the paper. Using simulation, we study the existing trade-off between power reduction and latency. We analyze the influence of the most important design parameters, namely thresholds U_{on} and U_{off}. The metrics used in this study are the average latency of a message (measured from generation to delivery time) and the relative power consumption of the links as compared with the default system. Although the switches can be moved to the standby state we do not measure the impact on power consumption as we focused exclusively on the links.

4.1 Network model

Our simulator models a wormhole switching network at the flit level [5]. The network is composed of two types of nodes, switch nodes and processor nodes. The switches contain a routing control unit, a crossbar and as many physical links as indicated by the network arity. Physical channels are split into three virtual channels. Each virtual channel has an associated buffer with capacity for four flits. Every switch implements the power saving mechanism presented in this paper. The results presented in this paper have been obtained for quaternary fat-trees of dimension 3 (64 nodes).

4.2 Traffic model

The network load is defined by several parameters: the message generation rate at each node, the message size, and the destination of each message.

The experiments reported in this section are used to analyze the static and dynamic behavior of the proposed power saving mechanism. First, the network performance is evaluated by using a constant message generation rate. In this case, the full range of traffic, from low load to saturation is evaluated. After that, experiments aimed at studying the dynamic behavior of the network are presented. These tests use variable generation rates during the simulation. In all the experiments, a synthetic workload based on the uniform distribution is used. All the nodes in the network have the same behavior: the message inter arrival time is generated according to the workload type, the message length is fixed to 16 flits and, the destination node for each message is chosen among all the nodes in the network (except the source node) with the same probability.

4.3 Parameters of the proposed mechanism

As explained in Section 3, at a given time the operational level of a link depends on its utilization. The dynamics of the model is driven by the off threshold U_{off} and the on threshold U_{on}. In the following figures, we explore the design space by selecting different values of U_{off} and U_{on} in order to achieve different goals of responsiveness and aggressiveness for the power saving mechanism.

On the other hand, a link cannot be instantaneously turned on, but it requires a time $T_{\rm on}$. Turning off a link also needs some time $T_{\rm off}$ to decrease the circuit voltage level to zero. When a link is turned off, we assume that it becomes immediately unavailable but it continues consuming power until $T_{\rm off}$ cycles have elapsed. Similarly, when a link is turned on, the new link is available to messages after $T_{\rm on}$ cycles, but power consumption increases at once. Based on the values reported by other authors, we have used $T_{\rm on} = T_{\rm off} = 1000$ clock cycles [7]. The state of the network is periodically checked to decide if it is necessary to turn on or off any link. We use a period greater than $T_{\rm on}$ and $T_{\rm off}$ in order to allow network stabilization after the changes. Specifically, the check period used is 2000 clock cycles.

4.4 Performance and power consumption results

Several experiments have been conducted to analyze the network behavior when the power saving mechanism works with different average thresholds and, hence, different power saving policies. Average thresholds of 0.5, 0.6, and 0.7 have been tested with hysteresis bands of 0.3, 0.4, and 0.5. These test points, whose results are shown on Figure 4, provide system configurations with different aggressiveness and responsiveness properties. The labels on the graphs indicate the thresholds in the format $U_{\rm off} - U_{\rm on}$. As expected, higher power savings can be obtained when the policy is more aggressive, that is, the traffic necessary to get the network to the full connected state (highest power consumption) increases with higher u_{avg}, as seen in Figure 4 (c). In addition, when keeping the average threshold constant, the narrower the hysteresis band, the higher the power saving; this is due to the increase in the off threshold when reducing the hysteresis band, for a constant average, that makes the switch to disconnect links with higher loads. The minimum power consumption achieved for the reported experiments is 50% of the nominal value which is consistent with the theoretical limit defined in Section 3 that, for a 4-ary 3-tree is 43.75%.

An important result is that there is no latency penalty since the resulting latency when applying the power saving mechanism exactly matches the nominal latency, when no power saving mechanism is applied. For the experiments we have performed, no performance penalty have been measured. Only the graph with $u_{avg} = 0.5$ is shown since the latency is not affected by the mechanism (Figure 3). This is due to the fact that the alternative paths fat-tree networks provide can be switched off for low network loads while the Minimal Tree links still provide connectivity at full bandwidth. When the load increases links are rapidly reconnected, providing additional bandwidth. As a result, depending on the traffic conditions of the network, links are turned off and turned on reducing power consumption with no impact on network latency.

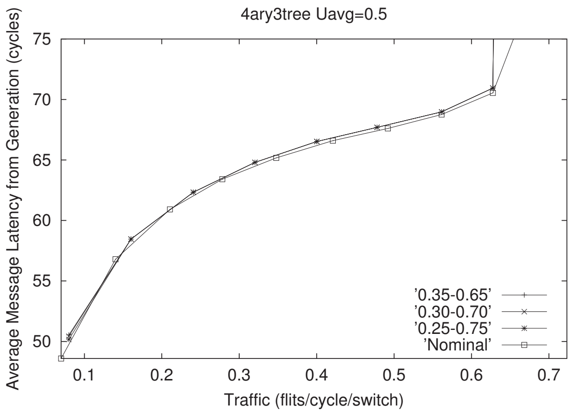
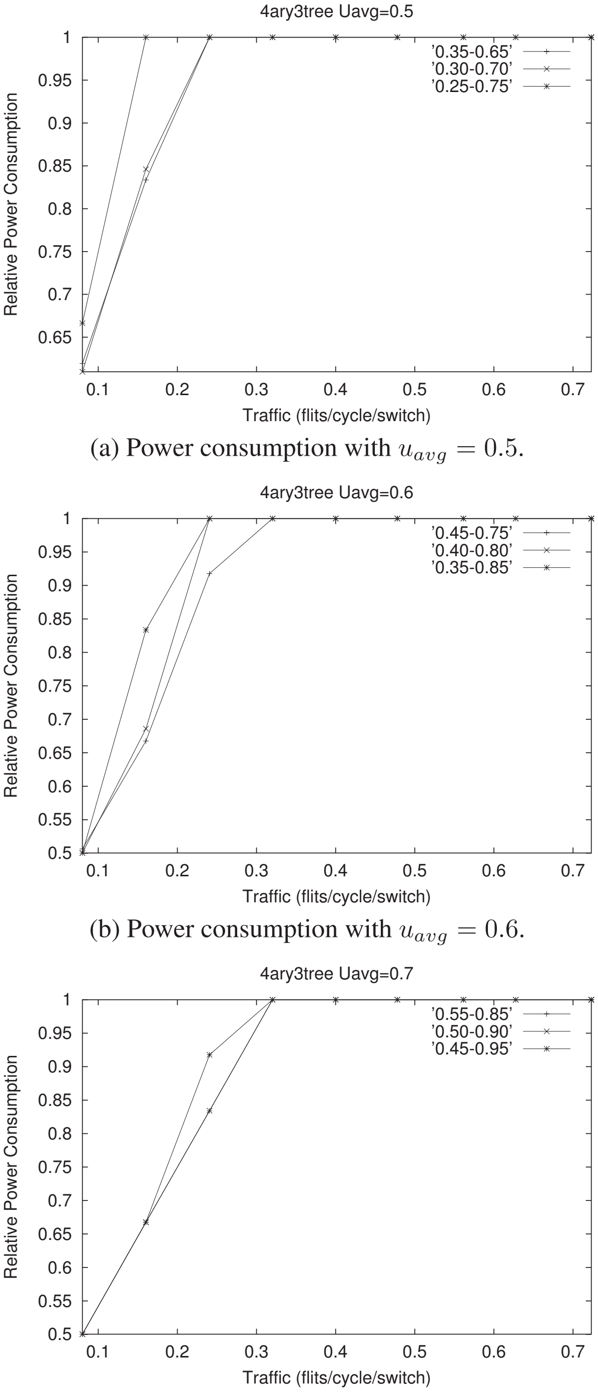


Figure 3. Latency with $u_{avg} = 0.5$.

In order to analyze the dynamic behavior of the proposed mechanism, simulations using a two-level load have been run. The simulation is initiated with low traffic. After a period of time, the load grows smoothly following a constant increase rate, up to a value seven times greater than the initial load. This high load remains during 60000 cycles. Then, the input traffic decreases again with constant rate down to the previous low load value. In this way, this test can illustrate the system behavior under rising and falling load slopes. The results are shown in Figure 5.

Initially all the links in the network are switched on. As the initial traffic is low, the power saving mechanism begins to turn off links. This disconnection proceeds until switch utilization balances with the power saving thresholds. At this moment, the consumption of the network is reduced to a 67% of its nominal value (see Figure 5 (c), while latency remains constant (Figure 5 (b). As the traffic grows, the power consumption increases due to the reconnection of links that have been previously turned off. Eventually, all links become active. When the traffic reaches its highest value and all links are fully operational, the latency becomes stable in the same value achieved by the network when no power reduction technique is in use (Figure 3). The downward slope of the traffic is followed by a decrease in latency and power consumption. As can be seen from the results the latency is insensitive to the power saving mechanism.

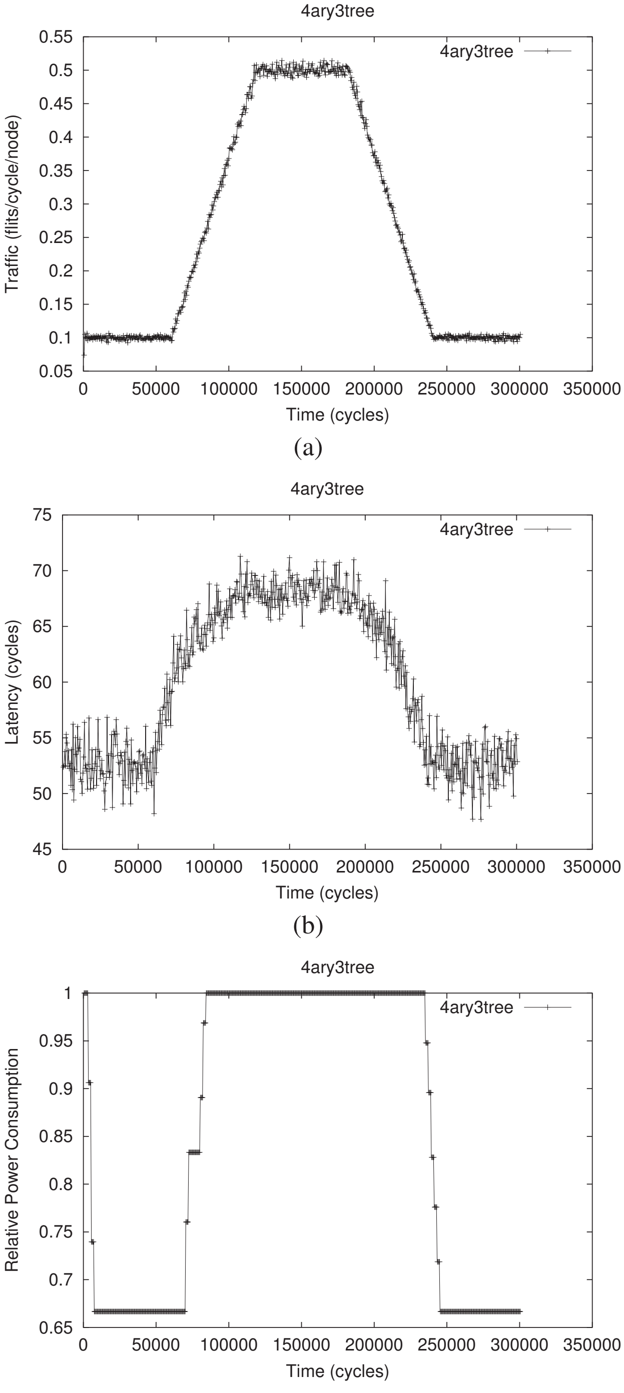


(a) Power consumption with $u_{avg} = 0.5$.

(b) Power consumption with $u_{avg} = 0.6$.

(c) Power consumption with $u_{avg} = 0.7$.

Figure 4. Results with constant u_{avg} and different hysteresis ratios.



(a)

(b)

(c)

Figure 5. Results with variable injection rates and $U_{\text{off}} = 0.3$ and $U_{\text{on}} = 0.65$.

5 Conclusions

In this paper, we presented a new mechanism to reduce power consumption in interconnection networks that use fat-tree topologies based on constant arity switches, k-ary n-trees. One important contribution of our mechanism is its simple implementation and the fact that the underlying routing algorithm does not need to be changed.

Our results, obtained by simulation on a 4-ary 3-tree network show that significant power savings can be obtained with no performance penalty. This is a very powerful result since any reduction in power consumption will be obtained at no performance cost. The path diversity provided by fat-tree networks makes it possible to turn off some links when the network load is low while alternative links provide their full bandwidth.

References

[1] http://www.top500.org.

[2] M. Alonso, J.-M. Martínez, V. Santonja, and P. López. Reducing power consumption in interconnection networks by dynamically adjusting link width. In *EuroPar 2004 Conference (Europar'2004)*. Springer-Verlag, 2004.

[3] M. Alonso, J.-M. Martínez, V. Santonja, P. López, and J. Duato. Power saving in regular interconnection networks built with high-degree switches. In *Proceedings of the 2005 International Parallel and Distributed Processing Symposium (IPDPS 2005)*, Denver, Colorado, USA, 2005.

[4] X. Chen, L.-S. Peh, G.-Y. Wei, Y.-K. Huang, and P. Prucnal. Exploring the design space of power-aware opto-electronic network systems. In *Proc. of the 11th Int. Symposium on High-Performance Computer Architecture (HPCA-11)*, pages 120–131, Feb 2005.

[5] J. Duato. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. 4(12):1320–1331, Dec. 1993.

[6] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: an Engineering Approach*. Morgan Kaufmann, August 2002.

[7] E. J. Kim et al. Energy optimization techniques in cluster interconnects. In *Proc. of the Int. Symposium on Low Power Electronics and Design (ISLPED'03)*, pages 459–464, Aug 2003.

[8] F. Petrini, W. chun Feng, A. Hoisie, S. Coll, and E. Frachtenberg. The Quadrics Network: High Performance Clustering Technology. *IEEE Micro*, 22(1):46–57, January-February 2002.

[9] F. Petrini and M. Vanneschi. Performance analysis of wormhole routed k-ary n-trees. *Int. Journal on Foundations of Computer Science*, 9(2):157–177, June 1998.

[10] J. Shalf, S. Kamil, L. Oliker, and D. Skinner. Analyzing ultrascale application communication requirements for a reconfigurable hybrid interconnect. In *Super Computing*, 2005.

[11] L. Shang, L.-S. Peh, and N. K. Jha. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proceedings of the 9th Int. Symposium on High-Performance Computer Architecture (HPCA-9)*, pages 79–90, Anaheim, CA, January 2003.

[12] V. Soteriou, N. Eisley, and L.-S. Peh. Software-directed power-aware interconnection networks. In *Proceedings of the Int. Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, San Francisco, September 2005.

[13] V. Soteriou and L.-S. Peh. Dynamic power management for power optimization of interconnection networks using on/off links. In *Proceedings of the 11th Symposium on High Performance Interconnects (Hot Interconnects)*, Stanford, CA, August 2003.

[14] V. Soteriou and L.-S. Peh. Design-space exploration of power-aware on/off interconnection networks. In *Proceedings of the 22nd Int. Conference on Computer Design (ICCD'04)*, pages 510–517, San Jose, October 2004.

[15] J. M. Stine and N. P. Carter. Comparing adaptive routing and dynamic voltage scaling for link power reduction. *Computer Architecture Letters*, June 2004.

[16] J. Vetter and F. Mueller. Communication characteristics of large-scale scientific applications for contemporary cluster architectures. In *Proc. Int. Parallel and Distributed Processing Symposium (IPDPS)*, 2002.