Power Utilization Techniques with Links of Interconnection Networks

Yanjing Li Milton Lei Jie Zhang Duc Pham

Abstract

Power consumption in interconnection networks has become a major concern in the community, with links of interconnection networks taking up a major fraction of the total power usage in the network. In this paper, we have surveyed three popular techniques for utilizing link power: Dynamic Voltage Scaling, On/Off Links, and Variable Width Links. From a thorough analysis of these techniques, we have concluded that there is a fundamental tradeoff between power and performance, but under certain network conditions, the performance penalty can be hidden. We have also investigated a new promising technique to utilize power usage in the links of a network.

1. Introduction

Power consumption has become a major concern in interconnection networks. In the past, interconnection networks are mainly used in high-end multiprocessors, which are used solely for computation-intensive applications. At that time, low latency and high throughput are the ultimate goals, constrained by packaging cost. Much research effort has been made in order to increase interconnection network performance. As a variety of communication systems emerge, interconnection network has made its appearance in on-chip networks, server blades, clusters, and terabit internet routers. These applications demand the design of interconnection networks to be power-aware and power-constrained. This demand needs to be urgently addressed. For example, the integrated router and links of the Alpha 21364 microprocessor consume about 20% of the total power (23W of a total of 125W) [9]. Designers of a Mellanox server blade estimate that the IBM InfiniBand 8-port 12X switch consumes about 37.5% of the 40W power budget [8].

Distribution of the power consumption in most interconnection networks[1] suggests that a majority of power is taken up by the links, compared to buffers and router components and logic. For example, the link circuitry of Alpha 21364 consumes 58% of the network power. In the IBM InfiniBand 8-port 12X switch, links take up to 65% (20W) of the total network power.

As the power budget for a system becomes tighter, power consumption in a network becomes more constrained. Even if power budget was abundant, it is still important to efficiently manage power usage, since the probability of failures increases with temperature, which can be raised by a greater power consumption. As a result, researchers have developed power models and profiles for interconnection networks, and both circuit level and architectural level techniques to utilize power usage.

In this paper, we focus on architectural techniques for utilizing power usage in links of an interconnection network. As links are a major consumer of power in most

[1] The major power consumer in some interconnection networks is not the links. For example, in on-chip networks, link circuitries are much simpler and thus consume a smaller portion of network power [6].

interconnection network, an effective method to reduce power consumption in links will in turn reduce the total network power substantially. However, as we will analyze in depth later, developing such effective techniques is a challenging task, with different constraints and tradeoffs between network performance (latency and throughput), design complexity, and hardware overhead.

This paper is organized as the following. In Section 2, we will present a survey and critique of three popular techniques in this area: Dynamic Voltage Scaling, On/Off Links, and Variable Width Links. In Section 3, we will summarize the similarities and differences of these three techniques. A new technique that augments the Variable Width Links method, combined with algorithms developed for Dynamic Voltage Scaling, will be investigated in Section 4, followed by conclusions in Section 5.

2. Existing Methods

2.1 Dynamic Voltage Scaling (DVS)

2.1.1 Description and Results of Technique

Dynamic voltage scaling (DVS) is a popular power saving technique originally proposed for microprocessors. The basic idea of DVS is to exploit the different requirements for frequency and voltage for different workloads. Workload variation also exists in interconnection networks. This is the motivation to adopt DVS techniques for potential power savings in interconnection networks [1] [4].

A crucial aspect of DVS techniques is the transition policy, which dictates when to scale and how much to scale. Channel bandwidth decreases linearly with link frequency, which can be seen from the equation b=w*f, where b, w, and f are the channel bandwidth, the channel width, and the link frequency, respectively. Reduced channel bandwidth due to reduced link operating frequency can cause degradation in both network latency and throughput. As depicted in Figure 1, the graph of network latency vs. input traffic rate shifts up and to the left as a result of frequency down scaling.

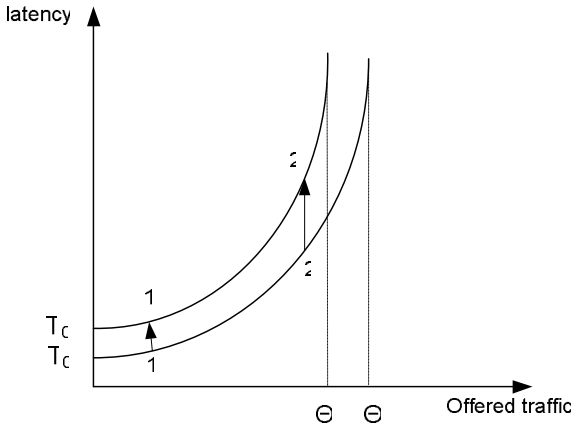

Figure 1: Latency and throughput penalty as a result of voltage scaling

A good transition policy will well balance the tradeoff between power savings and performance penalty. A distributed history-based DVS policy is proposed in [1]. To minimize performance impact, this policy first categorizes the existing traffic into two classes: lightly loaded and congested. When the network is lightly loaded, latency is more

sensitive to latency penalty introduced by frequency/voltage down-scaling, as illustrated by transition 1→1' in Figure 1. Therefore, voltage scaling should be conservative to minimize performance penalty. On the other hand, when the network is congested, the flit transmission across a router is constrained by the availability of the downstream buffers, not by link speed. Consequently, a more aggressive voltage scaling can be applied to save more power, as shown by transition 2→2' in Figure 1.

The distributed history-based DVS technique works as the following. Each port tracks its workload history, predicts its future workload, and dynamically adapts its frequency and voltage to the minimum required by the predicted future workload. Link utilization (1) and input buffer utilization (2) are then used to predict future workload using an exponential weighted relation (3). Two sets of thresholds, [TL_low, TL_high] and [TH_low, TH_high], are used in link utilization as criteria for scaling in light traffic conditions and congested network conditions, respectively. B_congested is used in buffer utilization to determine whether the network is predicted to be congested.

$$LU = \frac{\sum_{t=1}^{H} A(t)}{N}, \quad 0 \le LU \le 1 \qquad \text{…….. (1)}$$

$$\text{where } A(t) = \begin{cases} 1 & \textit{if traffic passes the link in cycle } t \\ 0 & \textit{if no traffic passes the link in cycle } t \end{cases}$$

$$BU = \frac{\sum_{t=1}^{H} (F(t)/B)}{H}, 0 \le BU \le 1 \qquad \text{…….. (2)}$$

where F(t) is the number of input buffers that are occupied at time t, and B is the input buffer size.

$$Par_{predict} = \frac{weight \times Par_{current} + Par_{past}}{weight + 1} \qquad \text{……. (3)}$$

where $Par_{predict}$ is the predicted indicator, $Par_{current}$ the indicator for current history period, and Par_{past} the indicator for past history period.

To evaluation the proposed history-based DVS policy, gate-level netlists combined with real-delay timing simulation is performed. Self-similar traffic pattern is injected into the network. On average, power reduction of 4.6X is obtained. This power saving comes at a price. The average network latency is increased by 15.2% and the average throughput is reduced by 2.5%. Sensitivity studies of policy performance to threshold settings are also conducted as part of policy evaluation.

2.1.2 Critique of the DVS technique

The distributed history-based DVS technique offers significant power savings (4.6X on average), which is a substantial improvement on the previously reported 2X power saving using a different policy [4]. Another advantage of this technique is that the DVS policy is based on information local to a node. As a result, hardware overhead is lower than other policies that require global network information. Also, since scaling is only applied locally, performance impact is minimal, as shown by the reported 2.5% reduction in the

network throughput. Using self-similar traffic, which has been empirically found to represent actual network traffic [10][11][12], provides further support for the effectiveness of the DVS policy. The disadvantage of DVS is that it requires complex hardware components such as the adaptive power-supply regulators and frequency synthesizers. These hardware components lead to higher cost and limit the applicability of the DVS technique.

Several issues need to be further addressed in [1]. First, prediction of future workload using link utilization and input buffer utilization needs to be further investigated. Second, DVS with its history-based policy can be tested under a more realistic environment in future experiments. Third, it will also be interesting to show the area overhead incurred from DVS. Last but not the least, effects of frequency and voltage levels can also be explored as there may be an optimal frequency and voltage granularity.

2.2 Dynamic Power Management Using On/Off Links (DPM)

2.2.1 Description and Results of Technique

The idea of using DVS links to optimize interconnection network power can be extended to turning off the link totally. This is the concept proposed in [2]. In [2], links are dynamically turned on/off depending on link utilization variations. Link utilization is evaluated via input buffer utilization, which is calculated following Equation (2). When link utilization on a router falls below the "Off" threshold, a link on that router is randomly selected and turned off. The remaining flits of a packet on the selected link are drained before the link is switched off. When link utilization on the router goes above the "On" threshold, an off link on the router is randomly selected and turned back on. Switching a link back on can take a significant amount of time, depending on the link technology. Thus, severe performance degradation can be observed. However, the off links in the method presented reduce path diversity and can possibly leave part of the network disconnected. Therefore, a power-performance connectivity graph is used to select candidates for on/off links. The ultimate purpose of the graph is to ensure that the entire network is still connected together when all on/off link candidates are turned off. Also, to avoid deadlocks, the employed routing algorithms, either proactive or reactive, must account for links in the off state and redirect traffic through links in the on state. As a result, routing is non-minimal. The hop count component of packet latency increases and is a function of the currently operational links.

To evaluate the performance of DPM with on/off links, Soteriou and Peh used an 8-ary 2-mesh topology in their network simulation [2]. For network connectivity, the topology imposes a maximum of 2 outgoing links per inner router as candidates for on/off links. Outer routers do not have any outgoing on/off link candidates. A 2 virtual channel class system is employed with one class using non-minimal east-last routing and the other non-minimal west-last routing. Uniform random traffic is used. On/off link transition latency (t_{sw}) is set to 1000 cycles. In this scenario, average latency increases by 48.5% when two links per router are off. A 34% power saving is observed under such situation. The impact of t_{sw} on power saving is studied by varying injection rates. With t_{sw} = 100 cycles, power savings of 35.9% is observed, while with t_{sw} = 1000 cycles, it slides down to 35.4%. When t_{sw} = 10,000 cycles, power savings drops significantly to 30.2%. Please note that due to the varying injection rate in this second experiment, the power saving number is different from the first experiment.

2.3.2 Critique of the DPM approach

In contrast to DVS links, the on/off links do not consume power when they are off. They also require simpler hardware to implement and can operate at higher speed. It is, thus, easier and less costly to implement. However, the need to keep the entire network connected limits the theoretical power savings of the scheme. Although the deadlock free routing in the simulation is fairly simple, a deadlock avoidance routing in a real interconnection network for the proposed method in [2] may render the system more complex. In fact, both the power-performance connectivity graph and deadlock avoidance routing are network type specific. Such specificity makes the adoption of the DPM approach more difficult. A methodology to systematically generate candidates for on/off links and routing algorithms given a network topology is needed. Power savings in this method is highly dependent on the on/off transition latency. Current on/off transition latency is about 10,000 cycles and the projected figure is about 100 cycles. The improvement from 10,000 to 100 cycles may lead to more link hardware overhead and, thus, may demand more power, and can offset the benefit of the on/off link transition latency improvement to power savings. The reader may notice that the power and performance numbers are less impressive than those stated in the DVS paper. However, please note that the experiment in [2] uses a network simulator and the experiment in [1] uses gate-level netlist. Also, the experiments used different traffic patterns for evaluation. Thus, no direct comparison should be made. In addition, uniform random traffic does not reflect realistic network traffic. As a result, the latency increase and power saving numbers stated will likely not be observed in a real world environment. Realistic network traffic should be used for evaluation purpose. Study should also be done on the effect of on/off threshold on power savings.

2.3 Variable Link Width Method (DAWL)

2.3.1 Description and Results of Technique

Unlike the DPM approach, which turns a link off to reduce the power, DAWL dynamically adjusts the bandwidth of a link by narrowing or increasing the width of links to utilize power consumption [3]. For simplicity and productivity, the adjusted width is always chosen to be double or half of the original width. The algorithm monitors the network traffic by increasing a counter every time a phit is transferred through a link (one counter per link). The counter value indicates the link utilization. When the link utilization drops below a threshold value U-off, the link width will be halved and when the link utilization rises above a threshold value U-on, the width will be doubled. The implementation of this mechanism is shown in the Figure 2.

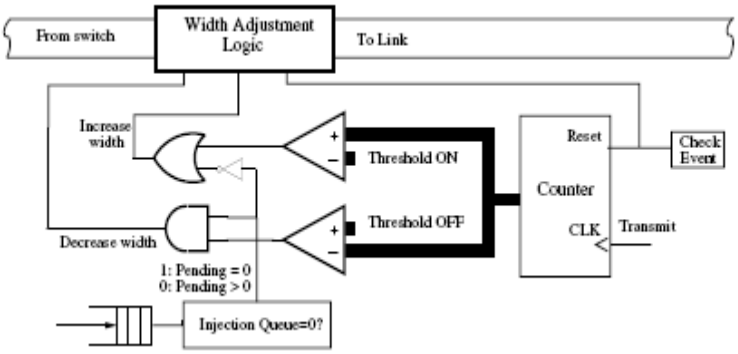

Figure 2: DAWL Hardware model implementation

Performance evaluation is done on a 32-ary 2-cube torus (1K nodes) and an 8-ary 3-cube torus (512 nodes), with minimal adaptive routing and wormhole flow control [3]. When traffic load is light, most of the link utilization is around 25%, resulting in a 4x deduction in power (26% of the original power consumption), with a 3x increase in latency. With a higher traffic load, a 2x power deduction is achieved (50% of the original power), with a 2x increase in latency. When the network is highly congested, the power saving approaches to zero and the latency matches the original network value.

2.3.2 Critique of the DAWL approach

There are several advantages of the DAWL approach compared to DPM. First, unlike in the DPM approach where routing is restricted when some links are off, we can always use the same routing algorithm. As a result, there is also no need to create a new connectivity graph and a new complex on/off algorithms to avoid deadlock as in DPM, which simplifies hardware design. Second, the minimal hop count remains unchanged which also makes the latency and power estimation easier. Third, DAWL has a granular power saving compared to DPM. Last but not the least, it offers a lower switching overhead than DPM. When compared to DVS, is also has the advantage that hardware design is simpler. In short, DAWL achieves a satisfactory power reduction with a simple design. The great disadvantage of DAWL is the latency penalty. Specifically, the 3-fold latency penalty may not be tolerable by many applications.

Several issues remain unaddressed by the paper. First, the claim that the HW implementation for DAWL is simple is very relative and subjective. For example, if we want to make the approach more applicable, we need to have a finer granularity in bandwidth adjustment for a better balance of performance and power. The paper suggests a selection function to allocate higher priority packets to wider links, which implies more complicated hardware to prioritize packets and perform packet switching. As another example, a minimal adaptive routing may not be adequate any more. Some long paths might have a lower latency because many links in these paths have high bandwidth, while some short paths may impose a high latency due to low bandwidth of the links. As a result, global information is necessary in order to estimate the effective minimal routing for lowest latency, which will in turn increases hardware complexity. Second, the simulation is performed based on a linear traffic model with uniform distribution. The results are not well justified with irregular or bursty traffic.

3. **Summary of Critiques for Existing Methods**

All three methods are related to one another. First, DPM can be considered as an extreme case of either DVS or DALW. In the extreme case of DVS, consider reducing the frequency of a link to zero, then DVS becomes DPM. Similarly, in the extreme case of DALW, consider reducing the width of a channel to 0, and then DALW is equivalent to DPM.

Second, all three methods sacrifice performance for power. They all decrease the throughput and increase the latency of the network by reducing widths or frequency of a link. DPM also affects the average minimal hop counter H_{min} in the network since some packets will need to resort to non-minimal routing due to the off links. The tradeoff between power and performance is a fundamental one, inherent in the physics of the network. However, as we can see in DVS, when the network is heavily congested, the latency required by a link becomes less critical if the downstream buffer space is mostly packed. In this case, the power reduction becomes "free" since the performance penalty is hidden.

On the other hand, each method has its own strong points and weak points. Table 1 summarizes these three methods with respect to different metrics, which are all critical characteristics for measuring the quality of the techniques.

Technique/Metric	DVS	DPM	DALW
Power Reduction	Limited by voltage/frequency scaling granularity	Limited by topology since links that can be turned off while keeping a connected network is limited	Limited by available channel width and width reduction granularity
Latency Penalty	Low penalty for lightly loaded network; Hidden (~0 penalty) for congested network	Heavy penalty for lightly loaded network; not applicable to congested network	Heavy penalty for lightly to modestly loaded network; not applicable to congested network
Throughput Penalty	Throughput Reduced	Throughput Reduced	Throughput Reduced
Design Complexity	High to provide voltage/frequency adjustments	Complex; needs new routing algorithms, deadlock considerations	Low with naïve implementation
Hardware Overhead	High to support frequency and voltage adjustments	Low	Low
Fault Tolerance Impact	Minimal impact	Reduce path diversity	Minimal impact

Table 1 : Summary of Methods

4. Proposed method

We have noticed that in order to utilize power usage in links in interconnection networks, there are mainly two methods: reducing frequency/voltage, and reducing link width. From a physics standpoint, given a network, these are the only techniques that can be used to reduce power consumption by links. Furthermore, there are mainly two policies determining when these techniques should be invoked: when network is lightly loaded and when network is congested. Both policies are well justified. Under light traffic, we do not need to provide as high throughput and while under congestion, we do not need to provide as low latency. However, it does not make sense if we invoke power adjustments when the traffic is heavy but is still below the network throughput limit, since both latency and throughput are critical. With these considerations, there are four possible configurations, as shown in the four entries of Table 2. Table 2 also summarizes the existing techniques with respect to these four configurations.

An interesting observation is that there is not yet a technique comprising "reducing width method" and "network congested policy". We therefore have investigated the feasibility of the method lying in this quadrant. Our conclusion is that this technique is promising. First, it improves DALW by achieving a better power reduction without sacrificing much performance penalty when the network is congested. Second, we expect that it has comparable results to DVS, while having a simpler design and less hardware overhead. We will present our arguments rigorously below.

Method/Policy	Network lightly loaded	Network Congested
Reduce frequency/voltage	**DVS, DPM**	**DVS**
Reduce link width	**DALW, DPM**	**Proposed Method**

Table 2: Techniques with respect to methods and policies

4.1 Method and Implementation Outline

This method is essentially a combination of DVS and DALW: it augments the DALW approach, incorporating the DVS history-based policy. To determine when the method should be invoked, we first sample link and buffer utilization within a predefined history window as described in Equations (1) and (2). When the link utilization is low while the buffer utilization is high, i.e. greater than a threshold B_congested, it is an indication that the network is congested. Second, we use an exponential weighted average utilization, combining current and past utilization policy, to predict the future link utilization, as shown in Equation (3). Similar to DVS, we also have two sets of thresholds to accommodate different network traffic loads. [TL_high, TL_low] are used when the network is lightly loaded, when width adjustment is conservatively carried out to minimize the impact on performance. [TH_high, TH_low] are used when the network is highly congested, when aggressive link width adjustment can be applied with link delay hidden in the congested network. When the predicted link utilization is below the lower bound for the corresponding traffic load, we double the link width if it is not already at its full bandwidth. On the other hand, if the predicted link utilization is above the upper

bound, we decrease the link width to half, if it is not already at its minimal bandwidth, which is the bandwidth of one signal in our case, to make sure a link is never completely off. This algorithm is shown below.

Algorithm 1

```
while (Signal width adjustment) do
LU_predicted = (W * LU_current + LU_past)/(W+1);
LU_past = LU_predicted;
BU_predicted = (W * BU_current + BU_past)/(W+1);
BU_past = BU_predicted;
if (BU_predicted < B_congested) then
        T_low = TL_low;
        T_high = TL_high;
else
        T_low = TH_low;
        T_high = TH_high;
end if
if (LU_predicted < T_low) then
        newWidth = Width/2;
else if (LU_predicted > T_high) then
        newWidth = Width * 2;
else
        newWidth = Width;
end if
end while
```

To implement this method, we simply make use of the hardware framework proposed by DALW, shown in Figure 2. This framework allows us to dynamically adjust link width based on certain criteria signaled to the network. However, the width adjustment logic will be replaced by the history-based policy described above.

4.2 Expected Results

We expect the results obtained by the proposed method to be very similar to that obtained by DVS. Intuitively, they impose a similar impact on channel bandwidth. Given that the channel bandwidth b = f*w, where f is the frequency of a signal and w is the total number of the signals, reducing the frequency by half and reducing the width by half will have the same effect on b: reducing it by half.

First consider the case when the network is lightly loaded. We rely on the fact that since the input injection rate of the network is low, a smaller fraction of the link width (i.e., smaller bandwidth of each channel) is enough to handle the amount of traffic without incurring congestion. However, the latency of each packet will be affected. Since the bandwidth is smaller, the serialization latency Ts = L/b will be higher due to a smaller b, where L is the packet length. This effect will be negligible for short packets. However, for long packets, this may induce substantial overhead that is not tolerable by an application.

Next we consider the case when the network is congested. In this case, the downstream buffer of a link will be mostly occupied. Getting the packet faster through the link will not help. In this case, the link width can be reduced more aggressively to save power, since performance penalty can be hidden.

From the above analysis, if we perform the same experiment as DVS (shown in section 2.1.1), and plot the power reduction vs. traffic injection rate and latency vs. traffic injection rate for our proposed method, the shape of these graphs are expected to be similar to those of DVS. However, the absolute numbers will be different, since the proposed method and DVS differ by the granularity of bandwidth reduction. At large link width, width reduction by half gives a significantly coarser adjustment in power and performance reduction than DVS. However, as link width becomes narrower, the power and performance reduction adjustment of our proposed method becomes finer. In the latter case, the difference in power savings and performance reduction of both methods reduces.

5. Conclusions

In this paper, we have explored many aspects of link power reduction techniques of an interconnection network. We have a thorough analysis of three popular techniques and investigated a new technique. The essence of these techniques is sacrificing performance for power, which represents an inherent tradeoff. However, with an accurate estimation of traffic conditions in a network, performance penalty can be hidden. Therefore, developing a good estimation function is the key to these techniques. Currently all methods make predictions based on linked utilization and/or buffer utilization. More sophisticated models are possible, and can motivate future research directions. Furthermore, formal optimization techniques can also be incorporated to select the granularity of power reduction to balance the tradeoff between power and performance.

References

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

[2] V. Soteriou, and L. Peh. Dynamic Power Management for Power Optimization of Interconnection Network Using On/Off Links. In *Proc. of the 11th Symposium on High Performance Interconnects (Hot Interconnects),* Stanford, CA, August 2003.

[3] M. Alonso, J.M. Martinez, V. Santonja, and P. Lopez. Reducing Power Consumption in Interconnection Networks by Dynamically Adjusting Link Width. In *10th International Euro-Par Conference*, Pisa, Italy, pages 882-890 August 2004.

[4] E.J. Kim, et al. Energy Optimization Techniques in Cluster Interconnects, in *Int. Symp. on Low Power Electronics and Design*, Aug. 2003.

[5] J. Kim and M. Horowitz, Adaptive supply serial links with sub-1V operation and per-pin clock recovery, in *Int. Solid-State Circuits Conf.*, Feb. 2002.

[6] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik. Orion: A Power-Performance Simulator for Interconnection Networks, in *Proc. MICRO*, November 2002.

[7] H.-S. Wang, L.-S. Peh, and S. Malik. A power model for routers: Modeling Alpha 21364 and InfiniBand routers, in *Proc. Hot Interconnects 10*, 2002.

[8] Mellanox Technologies performance, price, power, volume metric (PPPV). `http://www.mellanox.com/products/shared/PPPV.pdf.`

[9] S.S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb. The Alpha 21364 network architecture, IEEE Micro, 22(1):26-35, Jan/Feb. 2002.

[10] V. Paxson, and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226-244, June 1995.

[11] W. Willinger, M.S. Taqqu, R Sherman, and D.V. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. In *Proc. ACM SIGCOMM*, pages 100-113, Sept. 1997.

[12] G. Varatkar, and R. Marculesu. Traffic analysis for on-chip network design of multimedia applications. In *Proc. Design Automation Conference*, pages 795-800, June 2002.