Dynamic Power Management for Power Optimization of Interconnection Networks Using On/Off Links

Vassos Soteriou and Li-Shiuan Peh
Department of Electrical Engineering
Princeton University, Princeton, NJ 08544
{soteriou, peh}@ee.princeton.edu

Abstract

Power consumption in interconnection networks has become an increasingly important architectural issue. The links which interconnect network node routers are a major consumer of power and will devour an ever-increasing portion of total available power as network bandwidth and operating frequencies upscale. In this paper we propose a dynamic power management policy where network links are turned off and switched back on depending on network utilization in a distributed fashion. We have devised a systematic approach based on the derivation of a connectivity graph that balances power and performance for a 2D mesh topology. This coupled with a deadlock-free, fully-adaptive routing algorithm guarantees packet delivery. Our approach realizes up to 37.5% reduction in overall network link power for an 8-ary 2-mesh topology with a moderate network latency increase.

1. Introduction

Interconnection network fabrics have been deployed and proposed for a broad range of communication systems – processors with on-chip routers for multiprocessor systems [1], terabit Internet routers [2], clusters [3], server blades [4], and on-chip networks [5]. This quest for high performance in terms of high throughput and low latency has sparked the design of high speed router cores and communication links with data rates on the order of several Gb/s to match bandwidth requirements. However this design trend has also resulted in increased power consumption, imposing tight constraints on interconnection network architecture and system scalability. Interconnection networks are a principal consumer of total available system power with the link circuitry consuming an increasingly significant portion of the allocated router power budget. The integrated router and links of the Alpha 21364 processor [1] consume 23W out of the total chip power of 125W, where 58% of the power is consumed in the link circuitry. Since off-chip high speed links' power consumption does not vary considerably with traffic fluctuations, this portion will become more significant when compared to average system power as operating frequencies and bandwidth requirements upscale.

Recently, there has been research focusing on the power consumption of interconnection networks. Researchers have developed power models for these networks and have characterized the power profile of network routers and links [6] [7] [8]. Power-efficient interconnection networks using dynamic voltage scaling (DVS) with links, where link frequency and voltage is dynamically adjusted in response to network utilization, have also been proposed [9].

In this paper, we investigate the potential of carrying DVS links to the extreme – dynamically turning links on/off in response to communication traffic variance. In interconnection networks, temporal and spatial communication pattern variance is high, giving rise to variable communication link utilization. Making use of path diversity, traffic can be redirected when specific links are turned off, so good performance can be maintained while optimizing power.

In comparison to DVS links, on/off links require simpler hardware, essentially fast timing circuits. On the other hand, DVS links, which require variable-frequency links [10] to be extended to support fast, online voltage and frequency changes, need sophisticated hardware mechanisms for ensuring correct link operation during scaling, thus incuring significant delay overhead and additional CMOS area. Clearly, the power savings realizable with an on/off link is also greater, as DVS links continue to consume power even while idle.

Network power optimization using on/off links presents tougher challenges than that using DVS links. With links off, path diversity is greatly reduced, and network disconnectivity can potentially arise. In addition, the network can easily deadlock if the routing protocol is not tailored to on/off links. These are not faced by DVS links, as the network remains always connected, just operating at different frequencies. While on/off links can be reactivated to bypass disconnectivity, large performance penalties need to be incurred if link on/off delay is long.

In this paper we consider all the above issues, and propose an efficient network power management policy using on/off links. First, a power-performance connectivity graph is derived for the network topology, identifying candidate on/off links (See Section 2). Next, a deadlock-free routing algorithm tailored for on/off links is discussed in Section 3, and architectural implementation details of our on/off decision mechanism in Section 4. The on/off link architecture and power model assumed is shown in Section 5, and sim-

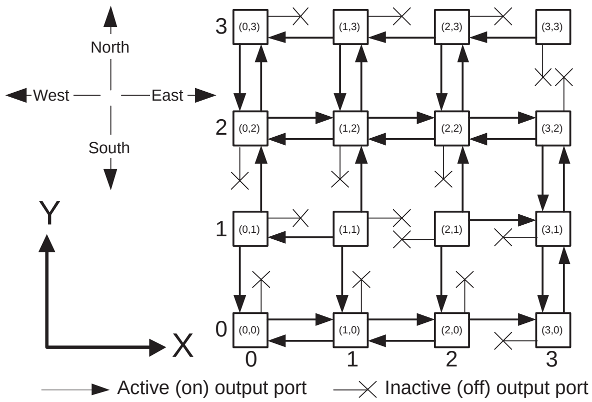

Figure 1. 4-ary 2-mesh topology with one randomly chosen off link per router. Severe disconnectivity results.

ulation results presented in Section 6. Section 7 concludes the paper.

2. Power-Performance Connectivity Graph

A fully random approach to on/off links is detrimental to performance - network connectivity suffers greatly and a core network router can be disconnected from the network with just $2n$ (n is the network dimension size of a k-ary n-cube topology with k nodes in each of the n dimensions) off links. Routers at the corners of the network can get disconnected with just n off links while border routers can get disconnected with $\frac{3n}{2}$ off links. Fig. 1 depicts a 4-ary 2-mesh topology in which just one link per router is allowed to turn off in a random manner. Even with this tight on/off link restriction corner router (3,3), and core routers (1,1) and (2,1) are directly and completely disconnected from the network. Though router (0,1) has an active incoming link from router (1,1), it cannot receive packets from any other router in the network. This is because router (1,1) cannot itself receive any packets, breaking the connection chain - router (0,1) is indirectly disconnected from the network. Only injected packets from router (1,1) can reach router (0,1) - clearly this is insufficient for full network connectivity.

Though links can potentially turn on, the transition overhead can introduce large delay penalties. What is more, considering a simplified distributed system with no global information sharing, it is almost impossible to achieve path diversity balance in the network under a random on/off link scheme. In Fig. 1 the route between routers (0,1) and (3,1) along row 1 is severely disconnected while just a row above, complete horizontal connectivity between routers (0,2) to (3,2) is enjoyed.

As a result, we propose to first derive a power-performance connectivity graph for on/off links. The graph dictates the candidate links for turning on/off, and is strictly coupled to the network topology so as to ensure packet delivery guarantee. A network connectivity graph for meshes is shown in Fig. 2 with *on/off link candidates* shown as switches and *all-time on* or *ever-on* links shown in solid

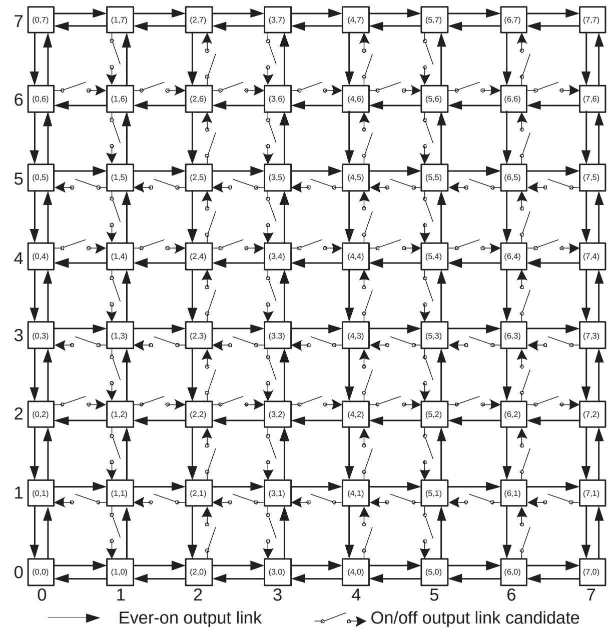

Figure 2. Power-performance connectivity graph of a 8-ary 2-mesh with ever-on and restricted on/off link candidates.

lines. Here, we allow at most 2 candidate on/off links out of a pool of 4 output links per router to ensure network connectivity. We have assumed unidirectional links. No border links are allowed to turn off for reasons explained in Section 3.

With such a power-performance connectivity graph, even when all candidate links are off, the network is still fully connected and every router can send messages to every other router in the network. In an 8-ary 2 mesh topology up to 84 links can turn off, with potential total power savings of 37.5% in all the links of the network. In an infinitely large 2D mesh topology, the power savings can theoretically reach 50%.

3. On/off Routing Algorithm

With on/off links, we require the use of a routing protocol that accounts for links in the off state and steers network traffic via operational links and routes. Several fault-tolerant routing protocols and theories for direct networks have been proposed [12] [13]. However, these approaches are *reactive* - they respond to link and node failures when they occur. Instead, our on/off link policy is *proactive*, exploiting the advance knowledge of link on/off state that is unavailable for fault-tolerant routing protocols, methodically steering traffic for best power-performance.

Armed with the power-performance connectivity graph, we now need to design a deadlock-free routing algorithm that delivers packets irrespective of the total number of on/off link candidates that are off during network operation. Using the Turn Model [14] and building upon the fault-

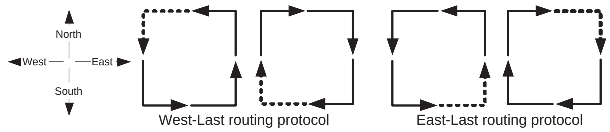

Figure 3. Turns permitted (shown with solid lines) and disallowed (shown with dashed lines) for the west-last and east-last routing protocols.

tolerant protocol idea proposed in [12], we developed a distributed, fully adaptive, non-minimal, deadlock-free routing algorithm. First, our algorithm splits the network into two separate virtual networks or virtual channel (VC) classes with no interaction between them. Every physical channel is split into two VC classes, where each class can have multiple VCs. Different routing protocols can be applied to the distinct classes. As a first study we used one VC per VC class (network). VC0 utilizes a nonminimal west-last (WL) and VC1 utilizes a nonminimal east-last (EL) routing protocol. The turns that these two routing protocols allow are shown in Fig. 3. The combined WLEL routing algorithm is deadlock-free as each algorithm forms an incomplete cycle [14].

To ensure deadlock freedom, no border links in the connectivity graph are allowed to turn off. This is justified with the following example - consider a packet at node (3,7) (Fig. 2.) which needs to go to router (3,6). If the south link at node (3,7) is off, the y-offset is nonzero and the packet is using the WL protocol, then the only choice left is to divert it to router (4,7) and then reroute it towards the south. The packet cannot travel to router (2,7) as the west direction is the last to be taken (y-offset must be reduced to zero first). At the same original node (3,7) if another packet needs to travel to south while the south link is off, the y-offset is nonzero and the packet travels westwards (EL), then the packet must be diverted to router (2,7) in the west and then traverse along the y-direction. Since both the east and west links of router (3,7) are needed to avoid deadlocks, both links should always remain on (ever-on). Similarly we can show that all border links of the remaining three sides of the topology should remain on. In the inner routers of our topology, up to 2 links can be turned off without incurring the above routing constraint, while guaranteeing full connectivity.

A packet uses the EL protocol when its destination node lies west of its source node and it uses the WL protocol when its destination node is at the east of its source node. A header flit of a packet carries 3 pieces of routing information: (1) the routing direction (eastwards or westwards which determines which of the two routing protocols to use, WL or EL which also identifies the VC number) which is determined by the subtraction of the destination and source routers x-coordinates; (2) the destination node coordinates for delivery and (3) the packet size and sequencing number.

Packets follow minimal routes as in XY or e-cube routing when no links are off. 180-degree turns are allowed only in the y-dimension and this usually occurs along the side borders of the network. The y-dimension has higher priority when the y-offset is nonzero, followed by the x-dimension when y-offset is zero and the x-offset nonzero. While routing eastwards (WL), a packet is allowed to route west at the most for one hop (routing west occurs when we are unable to minimize the x-offset to zero by routing via the E port and therefore mis-routing causes routing via the west port once); for the EL protocol, routing in the east can occur only once if we use similar arguments as in the WL case. When the initial x-offset is zero at packet injection, special routing cases occur. If the port which provides a progressive route in the y-dimension is available, this routing choice is taken. If this port is off, the packet is first routed for one x hop before it is progressively routed along the y dimension (this y port is guaranteed to be on due to the alternating on/off ports along the y-dimension of the connectivity graph).

A routing example is as follows: if a packet travels eastwards (WL), the current x and y-offsets are both greater than zero and the packet originates from the east (E) channel at the current router, the routing function returns N, E, S over virtual channel 0 (VC0). This is because the north (N) port has greater priority and should be traversed by the flit if it is active (on) - the y dimension is first covered (priority over the y-dimension) and then the x-dimension is covered as turns in the x-dimension experience more restrictions than in the y-dimension (i.e. travelling west is the last routing option taken for a packet using the WL protocol and no 180 turns are allowed in the x-dimension to avoid deadlocks). If the north port is unavailable (off) the packet routes east (to minimize the x-offset). If this output port is also off then the packet routes in the south dimension (routing south sends the packet further away from the destination so this routing option should be taken only when the other two options are exhausted). As we turn at most two links off per router, at least one of these ports should be on and thus it is impossible to experience deadlocks and packet drops.

4. On/Off Decision Mechanism

The decision to turn links on or off rests upon network statistics or counters that reflect current traffic. In this study, we use input buffer utilization as the metric in driving our on/off decision policy. Input buffer utilization, U_{buffer}, tracks the aggregate number of buffers in the current router:

$$U_{buffer} = \frac{\sum_{p=1}^{P}\sum_{t=1}^{t_{sw}}(F(t,p)/B)}{t_{sw} \times P}, 0 \leq U_{buffer} \leq 1 \quad (1)$$

where $F(t,p)$ is the number of input buffers that are occupied at time t for active input ports p, B is the input buffer size, t_{sw} is the sampling window size in clock cycles and P is the number of active input ports in a router. The input buffer utilization reflects the total amount of incoming traffic at the router. When U_{buffer} is relatively low, there is potential to turn a link off. When U_{buffer} is relatively high a link that has been turned off can be re-activated to avoid traffic build-up and performance degradation.

Simple threshold levels were chosen as link state transition deciders in our DPM on/off link policy.

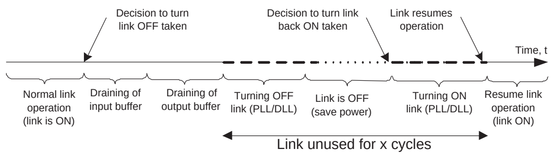

Figure 4. Timeline depicting link state switch stages.

$$0 \leq \alpha_{low} \ll \alpha_{high} \leq 1 \quad (2)$$

$$0 < \delta_{low} < \alpha_{low} \quad (3)$$

$$0 < \delta_{high} < \alpha_{high} \quad (4)$$

When U_{buffer} falls below the low threshold utilization ratio α_{low}, a link in a router can be switched off. A low threshold utilization change ratio defined as δ_{low} is then added to α_{low} in order to make the shutoff of an additional link in the same router tighter thus avoiding potential network congestion when network traffic increases abruptly. At the same time a high threshold utilization change ratio δ_{high} is subtracted from α_{high}, the high level utilization ratio, to increase the sensitivity to which a link responds to increased network traffic. When U_{buffer} surpasses α_{high} - δ_{high}, the link in the off state in the current router is turned back on. Links are randomly chosen from among the candidate links prescribed by the power-performance connectivity graph, one at a time. The threshold levels are adjusted according to the number of links which are on (or off) in a router, in order to fine-tune the on/off mechanism.

To ensure correct network operation, several steps need to be taken following an on/off decision. Fig. 4 illustrates the several operation stages. We have assumed virtual-channel (VC) flow control routers with input and output buffers. Flits from an input buffer traverse the crossbar switch on their way to an output buffer that has been acquired in the switch arbitration stage. Every output buffer is coupled to a single output link which connects the upstream (sender) router to a downstream (destination) router.

When a decision is taken to switch this link off, any remaining flits of a packet which has acquired this link are first drained out from the input buffer in order to avoid packet blocking and excess delays especially when a packet spans across several VC routers. If a packet blocks, resource arbitration constraints are imposed in downstream routers. This is because the packet holds both the VC of this router as well as that of the following router – no other packets can compete for these resources until they are freed after the link in the current router turns on and the acquired VCs are again freed for further arbitration. As the process of turning on a link can take a substantial span of time, severe performance penalties can be experienced.

Subsequent packet arrivals in all input buffers of the router are redirected to traverse active links by the on/off routing algorithm described in Section 3. Once the input buffer draining is complete (tail flit of this last packet departs from the input buffer), we then enter the output buffer drain stage. Once this output buffer is empty of flits, the process of turning off the link (sleep stage) is started and its duration depends on $t_{off}(link)$. Thereafter, the link enters the off stage during which power savings are incurred. This link enters the wake stage when at a later stage a decision is taken to switch it back to operation. This wake stage takes a time equal to $t_{on}(link)$ and once it is complete the link is fully operational. This link then becomes a valid candidate of the routing function again. Note that the link is unused starting from the beginning of the sleep stage until the end of the wake stage. During these two stages, we assume conservatively that the power consumption is the same as the normal operating power expenditure when the link is operational and in the on state.

When the link is de-activated, the output buffer coupled to this link at the upstream router is unused. This buffer can also be deactivated in order to save additional power until a decision is taken to switch the link on. Similarly, the input buffer at the downstream router can be shut down once it is emptied and while the incoming link is still off. Thus additional power savings can also be incurred here until the buffer begins to receive flits when the link is reactivated. This scheme can work in conjunction with proposed mechanisms for power-aware buffers in interconnection networks proposed in [15] for supplementary power savings when links are on and operational. However, in this study we only consider link power savings.

5. On/Off Link Architecture and Model

There are a few important characteristics which affect the functionality and operational efficiency of an on/off link. A link cannot immediately turn off due to the inherent physical timing limitations of reducing the circuit voltage level from some nonzero V_{cc} value to a zero level. Also time is needed to lock on (latch) to the clock when the link is switched back on to operation. The state transition delay, $t_{off}(link)$, can be expressed in terms of system clock cycles. The counterpart, $t_{on}(link)$ is the transition delay needed to turn a link back to operation.

Commercial chip-to-chip designs such as the Motorola MC92610 WarpLink 2.5Gb/s Quad SerDes Transceiver [11] have links that can be individually disabled in order to save power or they can be put into low power standby node. Current links have a startup time of approximately 10,000 clock cycles, but link designers are optimistic that this can be improved down to 100 clock cycles. These variables directly affect overall network performance- delaying link shutoff reduces the potential power savings, as power is also consumed in the link while transitioning between states. Delaying link re-activation potentially induces network traffic congestion.

In our experiments we assumed a simple link power model: that the power consumption of a link stays constant when it is on, since link worst-case and nominal power is close (the IBM InfiniBand 12X LPE TX consumes a nominal power of 0.26W and a worst case power of 0.3W while its RX takes up nominal 0.17W and worst case 0.2W); that a link takes zero power when shut off; and consumes the same power as when it is on during transitions from on to off states and vice versa.

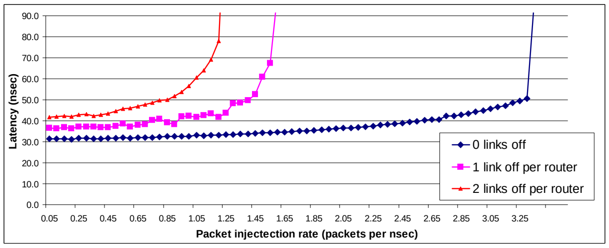

Figure 5. Latency-throughput of on/off network with worst-case traffic (uniform random).

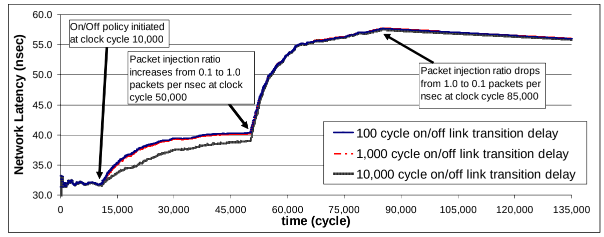

Figure 6. Network latency for various t_{off} and t_{on} link state transition delays as traffic injection rate varies across time.

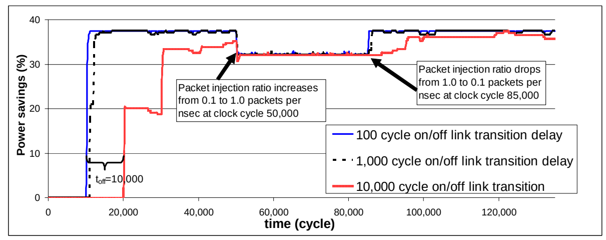

Figure 7. Network power savings for various t_{off} and t_{on} link state transition delays as traffic injection rate varies across time.

6. Experimental Results

6.1. Experimental Setup

We implemented an event-driven, flit-level interconnection network simulator with credit-based flow control for evaluating our proposed on/off policy by extending a network simulator publicly available at [16] . The simulator supports k-ary 2-mesh topologies consisting of 5-stage pipelined virtual-channel (VC) routers - routing, virtual channel arbitration, switch allocation, switch traversal and output link traversal. The simulator was implemented in C++ (7,500 lines of code) based on standard template libraries (STL), running under Red Hat v. 8.0 Linux OS.

We have assumed an 8-ary 2-mesh topology with the power-performance connectivity graph depicted in Fig. 2, consisting of 64 1GHz routers, with each input port having 2 virtual channels (one for each virtual network), 96 flit input buffers and 40 flit output buffers. Each flit is 32-bits wide and can be transported in 1 cycle over a 32Gb/s router channels. Each router consists of 8 unidirectional channels (four incoming and four outgoing for each direction). Injection and ejection queues at the network interface each housing an infinite number of flits were also assumed. Each simulation is run for 10 million cycles and the metrics which are considered are latency (averaged over all packets), throughput and power consumption. Latency spans the injection of the head flit of a packet until its tail flit is ejected from the destination router. The saturation throughput of the network is considered to be the point where the average latency of the packets is double the zero-load latency. Power savings is the ratio of the aggregate power savings across all links in the router, divided by the power consumption if all links are operational.

6.2. Performance of On/Off Link Policy

Workloads and traffic patterns directly impact on/off network performance and potential power savings. Workloads which exhibit no spatial or temporal variances such as uniform random traffic give worst case performance with an on/off link policy. This is because traffic flow is even throughout the network and across time - it is therefore virtually impossible to detect a varying network utilization which can trigger the on/off mechanism systematically for maximal power savings and minimal negative impact on performance. For heavy network traffic it is hard to turn off a specific link and for low network traffic almost all on/off candidate links will switch off, causing poor network performance.

As a first evaluation of our policy, and to investigate the potential of on/off links, we consider the worst-case workload: uniform random. This experiment basically explores the worst-case latency impact under maximal power savings of the power-performance connectivity graph. Packets consisting of five flits each were assumed with t_{sw} equal to 50 cycles. t_{off} and t_{on} were both set to 1,000 clock cycles while α_{low}, α_{high} were set at relatively low and high values respectively so as to force the links to stay and remain off.

Results are shown in Fig. 5. With one link off per router, the average increase in network latency is 29% and for two links off per router, this rises to 48.5%. Overall network link power savings for the first case is 21.4% and 37.4% for the second case.

To investigate the impact of on/off link transition delays on our connectivity graph, we run uniform random traffic at varying injection rates. A snapshot of the network power-latency behavior for three values of t_{on}=t_{off} at 100, 1000 and 10000 cycles is shown in Figs. 6 and 7. The simulation is first run without the on/off policy and at clock cycle 10,000 the on/off policy is triggered and links begin to shutoff. An initial injection low rate of 0.1 packets per $nsec$ (cycle) is assumed, which increases sharply to 1 packet per $nsec$ at cycle 50,000 and then drops back to 0.1 packets per $nsec$ at cycle 85,000. Power savings of the order of 35.9% are incurred for t_{off}=100 cycles, while the case of t_{off}=1000 cycles follows closely with 35.4% savings, a

1.59% relative drop. For t_{off}= 10,000 cycles the overall link power savings are 30.2%, a 15.97% relative drop to the t_{off}=100 cycles case. For t_{off}=10,000 the level of overall power savings is expected to drop when network traffic exhibits greater variance, underlining the importance of having links which can quickly toggle state. Fig. 6 shows the latency impact of increasing transition delays. Initially, with on/off links of t_{off}=10,000 cycles, delay increases at a slower rate as it takes longer to reach the network configuration where two links are off per node. Eventually, there is no significant difference between the links of different transition delays as the same number of links are turned on/off in all three cases, as dictated by the connectivity graph.

In summary, our preliminary results indicate that on/off links can realize significant power savings (up to 37.5% for a 8-ary 2-mesh) with moderate performance impact even with worst-case uniform random traffic. We will further explore the power-performance of on/off links with more realistic traffic patterns. Our findings also indicate the importance of fast on/off transition delays and we see our research providing insights that will motivate more efficient dynamically switchable link designs.

7. Conclusion

We proposed a DPM policy with on/off links for minimizing power in interconnection networks. We have devised a systematic approach that first derives a power-performance connectivity graph for a topology, coupled with a distributed deadlock-free routing algorithm that maximizes performance. On/off links are easier to implement as compared to prior power-aware link mechanisms such as DVS links, incurs minimal power overhead and avoids communication overhead in global information sharing. While it brings about tougher challenges due to the possibility of disconnectivity, we show that a systematic approach can lead to high power savings with moderate performance impact even with worst-case uniform random traffic. We will further explore on/off links with realistic communication traffic patterns, generalizing it to myriad network topologies, and designing more sophisticated policies that can further improve network power-performance.

Acknowledgments

The authors would like to thank Li Shang of Princeton for kindly providing a base network simulator to which our on/off link policies were added and for insightful technical discussions. We also wish to thank Ken Shepard and George Patounakis of Columbia University, Edward Lee of Velio Communications and Gu-Yeon Wei of Harvard University for technical discussions on link architectures and high-speed signaling. In addition, we thank Craig Stunkel of IBM T.J. Watson Research for providing architectural details and power numbers for the InfiniBand 8-port 12X switch.

References

[1] S. Mukherjee et al., *The Alpha 21364 network Architecture*, In Proc. Hot Interconnects 9, pp. 113–118, Aug. 2001.

[2] W. J. Dally, P. Carvey, and L. Dennison., *The Avici Terabit switch/router*, in Proc. Hot Interconnects 6, Aug. 1998.

[3] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su. *Myrinet - a Gigabit-per second local-area-network*. IEEE Micro, 15(1):29-36, Feb. 1995.

[4] InfiniBand Trade Alliance. *The InfiniBand architecture*. http:// www.infinibandta.org.

[5] W. J. Dally and B. Towles. Route packets, not wires: *On-chip interconnection networks*. In Proc. Design Automation Conference, pages 684–689, June 2001.

[6] C. Patel, S. Chai, S. Yalamanchili, and D. Schimmel. *Power-constrained design of multiprocessor interconnection networks*. In Proc. Intl. Conference on Computer Design, pages 408–416, Oct. 1997.

[7] Hang-Sheng Wang, Li-Shiuan Peh and Sharad Malik, *A Power Model for Routers: Modeling Alpha 21364 and InfiniBand Routers.*, In Proceedings of the 10th Symposium on High Performance Interconnects (Hot Interconnects), Stanford, CA, August 2002, pp. 21–27.

[8] Hang-Sheng Wang, Xinping Zhu, Li-Shiuan Peh and Sharad Malik, *Orion: A Power-Performance Simulator for Interconnection Networks.*, In Proceedings of the 35th International Symposium on Microarchitecture (MICRO), pages 294–305, November 2002.

[9] Li Shang, Li-Shiuan Peh and Niraj K. Jha, *Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks.*, In Proceedings of the 9th International Symposium on High-Performance Computer Architecture (HPCA), Anaheim, CA, pages 79-90, Feb. 2003.

[10] G.Wei, J. Kim, D. Liu, S. Sidiropoulos, and M. Horowitz, *A variable-frequency parallel I/O interface with adaptive power-supply regulation*, J. Solid-State Circuits, vol. 35, no. 11, pp. 1600–1610, Nov. 2000.

[11] http://www.motorola.com/

[12] J. Duato, *A Theory of Fault-Tolerant Routing in Wormhole Networks*, IEEE Transactions on Parallel and Distributed Systems, Vol. 8, NO. 8, August 1997.

[13] Patrick. T. Gaughan and Sudhakar Yalamanchili, *A Family of Fault-Tolerant Routing Protocols for Direct Multiprocessor Networks*, IEEE Trans. on Parallel and Distributed Systems, Vol. 6, NO. 5, May 1995.

[14] Christopher J. Glass and Lionel M. Ni, *The Turn Model for Adaptive Routing*, Proceedings of the 19th International Symposium on Computer Architecture, pp. 278–287 May 1992.

[15] Xuning Chen and Li-Shiuan Peh, *Leakage Power Modeling and Optimization in Interconnection Networks*, In Proc. International Symposium on Low Power Electronics and Design, Aug. 2003.

[16] http://www.ee.princeton.edu/∼lshang/popnet.html