Early Experiences with Saving Energy in Direct Interconnection Networks

Felix Zahn
Heidelberg University
Institute of Computer Engineering
Mannheim, Germany
felix.zahn@ziti.uni-heidelberg.de

Steffen Lammel
Heidelberg University
Institute of Computer Engineering
Mannheim, Germany
s.lammel@stud.uni-heidelberg.de

Holger Fröning
Heidelberg University
Institute of Computer Engineering
Mannheim, Germany
holger.froening@ziti.uni-heidelberg.de

***Abstract*—Energy is emerging to become one of the most crucial factors in design decisions for future large scale computing systems. Especially Exascale-installations will have to operate within hard power and energy constraints. Besides economical reasons, power consumption is also limited by a limited power distribution, cooling capabilities, and minimization of carbon footprints. While other components, such as processors, become more and more energy-proportional, interconnects are still highly energy-disproportional. Although interconnection networks are contributing only about 10-20% to the overall power consumption of High-Performance Computing (HPC) or Cloud systems, this fraction is likely to increase significantly in the near future. Therefore, power saving strategies are mandatory for improving energy efficiency and thereby performance within hard power constraints.**

In this work, we introduce a simple energy saving strategy, which switches links on and off, depending on the user's performance constraints. Therefore, we adapted an existing OMNeT++ network simulator by adding new energy features. This simulator allows us to run traces of real world applications, including LULESH, NAMD, and Graph500 with different configurations. We show that this policy enables possible energy savings of up to 39% in interconnection networks. Furthermore, we demonstrate the impact of hardware design parameters, such as transition time, on possible power saving strategies.

I. Introduction

In today's CMOS-based compute technology, power consumption is the dominating constraint. In Post-Dennard performance scaling, traditional techniques like frequency scaling and an improved amount of instructions per cycle (IPC) are no longer applicable. Instead, the key to performance is scaling the amount of operations per Watt. Due to data dependencies, data movement is an inherent part of scientific applications and besides optimizing the amount of floating point or integer operations per Watt, one has to consider the associated costs when moving the input and output operands. Another observation is that power consumption for data movements strongly depends on the distance. For short on-die links, power consumption depends linearly on the transmission line length, while for longer connections it quickly behaves super-linear due to effects like dielectric loss, skin effect, and more. In clustered systems, inter-node communication dominates power consumption.

In spite of these fundamental constraints, surprisingly little attention has yet been paid to optimize the power consumption of scalable interconnection networks. In particular, related work shows that such networks contribute about 20% to the system's total power consumption [1]. Note that this fraction is likely to increase as other components such as processors, memory, power distribution, and auxiliary equipment are becoming more energy-proportional and thus more power efficient [2]. A component is energy-proportional, if its energy consumption depends linearly on its utilization. Then, a power consumption of 100% at peak load will decrease linearly until it reaches zero Watt for an idling system.

Surprisingly, in recent work we have shown that the biggest contribution to power consumption in a network switch is not actually the switching and routing logic, instead the serial links contribute most [3]. This is mainly due to the fact that their transmission standard is based on a constant current, which also implies that their power consumption is independent on the actual switching rate. Also, embedded clocking and multiple parallel lanes require complex link training protocols that have a significant impact on the transition time from one configuration to another. In the following, thus, is being referred to as transition time. We anticipate this transition time to be key for an effective power saving technique, however, it seems complex to optimize the associated techniques. Instead, in this work we provide insights by studying how power savings vary with transition time, providing valuable insight on trade-offs regarding its optimization.

Applications' communication patterns are another crucial component of power savings, as we have to exploit idle periods and varying network loads to scale down network capacity. In recent work we have shown that typical high-performance computing applications have a surprisingly low overall utilization and long inactivity periods [3]. Also, we have shown how workloads differ and characterized typical link loads [4].

This work coninues our previous contributions by an analysis at network level, based on a network simulation that is extended by power consumption data. We add a rather simple but effective power saving policy [5] to the network links and demonstrate how the power saving potential varies with different technical and operative constraints. This includes transition time and maximum tolerated performance loss.

In particular, we make the following contributions:

- Introducing a simple power saving policy, which allows to balance performance loss and energy savings.
- A study on the impact of different transition times regarding power consumption and performance in interconnection networks.
- An evaluation of the power saving potential when introducing a power-saving strategy for three representative applications.
- A short discussion about the limits, strengths, and weaknesses of this power saving policy and an outlook on future research directions.

The remainder of this paper is structured as follows. In section II we provide a short overview about energy consumption in today's interconnection networks as well as the workloads we used for our experiments. We continue in section III with providing information about our methodology, our implemented power saving policy, and the setup used in our experiments. The results of our first experiments is provided in section IV. Finally, we conclude with a short discussion of our results and future directions, a brief review of related work, and a final summary.

II. Background

While processors and memory are becoming increasingly energy-proportional, other components such as interconnection networks are still highly energy-disproportional. This means they consume about the same amount of power, regardless of whether they are used or not. For example, in a fully energy-proportional system a 50% amount of utilization would consume 50% of thermal design power (TDP).

A. Energy-proportional networks

In previous work we analyzed an example TSMC-65nm network switch (EXTOLL Tourmalet). This analysis has shown that serial links are dominating overall power consumption, indicating that energy-proportional networks have to employ energy-proportional links. The scaling behavior for these links is shown in Fig. 1.

In contrast to other components such as processors and memory, the impact of frequency changes is almost neglectable for overall link power consumption, because links in today's interconnects are dominated by the current mode logic (CML) signalling standard. As its name suggests, it relies on a constant current, independent of transitioning or not. This means power is constantly consumed, not only when clock slope is switching. However, link encoding must ensure frequent transitions for the clock recovery to remain locked, which causes frequent transitioning signals, even when no data is transmitted.

As opposed to link frequency, link width has much greater impact on power consumption and appears to have an almost linear effect on power consumption. Thus, scaling link width seems to be a more suitable approach to implement energy-proportional links. Most of current interconnection network technologies are composed of multiple lanes, which are serial sub-links in parallel, and therefore provide the necessary

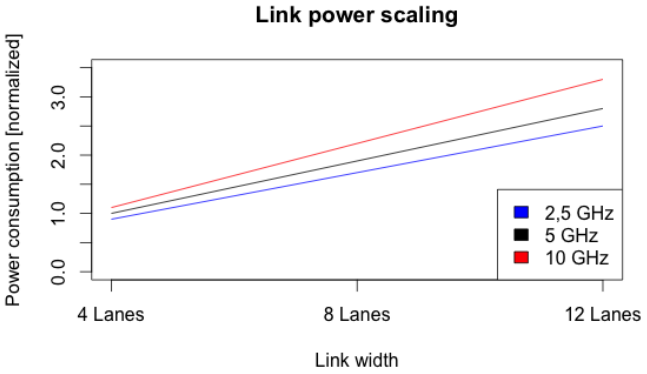


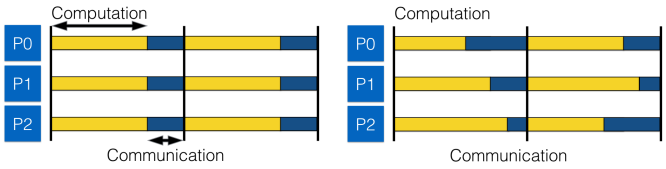
Fig. 1: Link power scaling.

conditions for link width scaling. The width of a link varies among technologies, for instance up to 32 for PCIe, typically 4 for Ethernet, Infiniband and Omni-Path, and 12 for EXTOLL.

An important technical restriction for a dynamic bandwidth adjustment is transition time. After each reconfiguration, including power-up, the different end points of a link have to perform a training, which may last from a couple of microseconds up to milliseconds. Additionally, link synchronization requires time-extensive locking of phase-locked loops (PLLs) or delay-locked loops (DLLs). It seems that these design parameters have received too little attention recently, and we believe that initial insights about their influence on different energy saving strategies are crucial for future design decisions.

B. Workloads

Due to their variety of communication patterns, we see traces of real-world HPC applications as more appropriate and representative than synthetic traffic. While a random synthetic traffic pattern will prevent most benefits of an energy saving strategy, traces draw a much more realistic picture of the energy saving potentials in today's HPC systems.



(a) Iterative/temporal pattern (b) Iterative/non-temporal pattern

Fig. 2: Classification of different communication pattern [5].

We used our communication characterization tool SONAR [4] to analyze a variety of workloads. Analog to the classifications the authors in [5] have introduced, we decided to use the following three different benchmarks, which we believe to be representative.

1) Iterative/Temporal: Workloads from this class feature an iterative communication pattern. Additionally, all processes perform similar computations and show recurrent duration of communication, as depicted in Fig. 2a.

One example for this kind of application is LULESH (Livermore Unstructured Lagrange Explicit Shock Hydrodynamics)[1]. It calculates hydrodynamic simulations and is a proxy application in the United States Department of Energy's co-design efforts for exascale computing. In this application, a stencil code is used to calculate the physical forces. Fig. 3 depicts the network activity map for one node running this workload, in which each discrete event is represented by one data point. The X axis shows the time of occurrence, the Y axis indicates the message size, and the color visualizes point-to-point (red, green) and collective messages (purple, blue). The regular communication pattern of this workload is well recognizable.

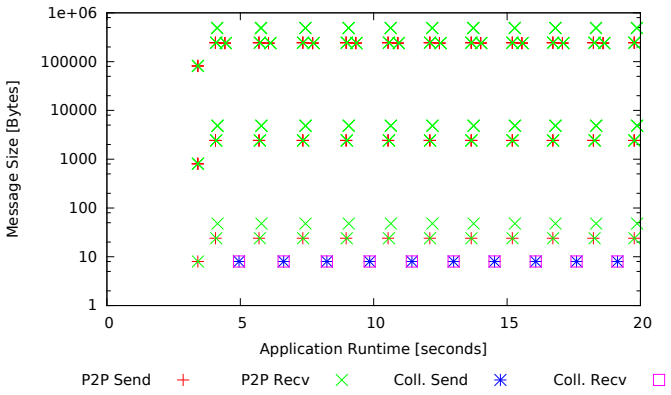


Fig. 3: Network activity map of LULESH.

2) Iterative/Non-Temporal: This class also includes applications with an iterative communication pattern. As opposed to the uniform behavior of all processes in the iterative/temporal class, the time spent in MPI calls in this kind of workloads can differ among processes and iterations. Fig. 2b shows the communication pattern for this class of applications.

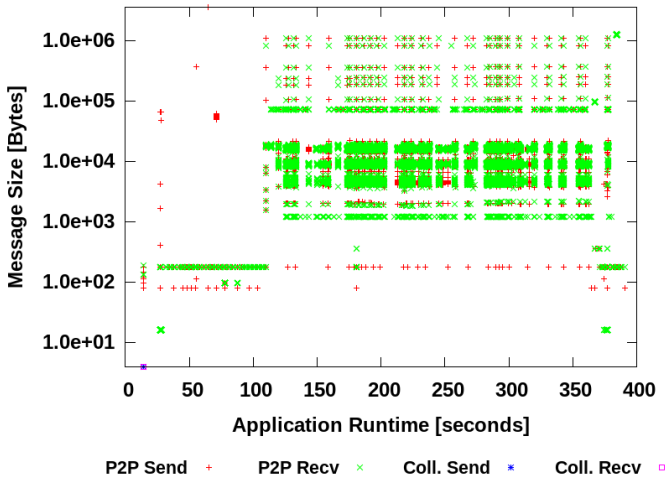


Fig. 4: Network activity map of NAMD stmv.

[1]https://codesign.llnl.gov/lulesh.php

We selected the NAMD (Nanoscale Molecular Dynamics program)[2] as representative for these group, which is a molecular-dynamic-based application, designed for the simulation of biomolecular systems [6]. The code is mainly a particle simulation, which is compute-bound in most cases. We used the common Satellite Tobacco Mosaic Virus (STMV) as input data, which comprises of about one million atoms. The network activity, shown in Fig 4, has a more irregular traffic pattern than LULESH, but discrete iteration steps are still clearly visible.

3) Non-Iterative: The last group of workloads classifies all workloads without an iterative behavior. The most common examples are graph algorithms.

For our analysis we used the Graph500[3] workload. It is a memory-bound benchmark, which employs a breadth-first search (BFS) graph traversal. Both, its memory access pattern and its communication pattern are highly irregular and dynamic, as Fig 5 depicts.

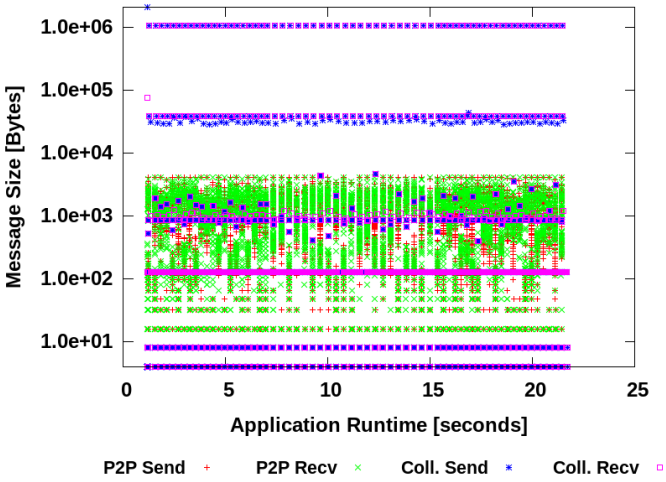


Fig. 5: Network activity map of Graph500.

This benchmark is also used to rank the 500 fastest supercomputers on the Graph500 list as an alternative to the TOP500 list.

III. Power-Aware Network Simulation

In this section we describe the settings and parameters that were used for our experiments. First, we want to show which power saving potential in current hardware already exists, and second we want to demonstrate the impact and potential of parameters such as transition time or the number of different power saving states on future designs.

A. Methodology

Since the impact of different design parameters cannot be tested with existing hardware, we use a network simulator for our experiments. An existing OMNeT++ based network

[2]http://www.ks.uiuc.edu/Research/namd
[3]http://www.graph500.org/

simulator [7] was extended by power-analysis features. The detailed implementation of this simulator is shown in our previous work [3].

In order to save energy in the network it is mandatory to introduce power saving states by reducing link width or frequency. As shown in section II, reducing link width is the most promising approach for CML-based high-speed serial links. For our first experiments and complexity reasons, we decided to limit our power saving strategy to two different power states. Unsurprisingly, our power analysis of the EXTOLL Tourmalet network switch has shown that the two most efficient power states in terms of bit/J are running links with full speed and switching them completely off.

B. Power Saving Policy

Every time a power state is changed in a link, the link needs a certain time to reconfigure and lock its DLLs or PLLs, which can take up to $10\mu s$ [8]. During this transition time no data can be sent over the link. Thus, a power saving strategy based on different power states always correlates with a loss in performance. The user has to find an appropriate trade-off between performance and power. Taking this performance loss into account, energy is the appropriate metric for optimization, since it is the product of power and time. We decided to implement a simple policy [5], which lets the user decide how much performance loss can be tolerated and which links can be switched off. The policy is based on two parameters: the transition time t_t and the relative maximum tolerated performance loss ρ. With these two parameters a time Δt can be calculated, which indicates when to change a power state. If a link idles for this amount of time it is switched off until a new packet arrives at this link. In the worst case, a packet arrives at the moment a link is switched off. Now, the link is switched on again and needs the transition time t_t to reconfigure before it is able to send the packet. This means the inactivity period Δt has to be ρ times larger than the transition time t_t to ensure a performance loss less or equal ρ in this worst case scenario. This means: $\Delta t = \frac{t_t}{\rho}$. For example, if a link needs $t_t = 100\mu s$ to reconfigure and a performance loss of 10% ($\rho = 0.1$) is tolerable, then $\Delta t = 1000\mu s$.

Note that the maximum performance loss in this policy is only from a node's perspective. If a message propagates through the network and multiple links are switched off, the actual performance loss can become much larger than actual tolerated. But since this policy is supposed to handle the worst case, we believe it to be appropriate for first investigations and experiments.

C. Setup

The following parameters for our simulator and benchmark applications were selected to run our experiments.

1) Traces: We used traces from three different scientific applications, which differ fundamentally in their communication pattern. These traces were generated with the VEF framework [9], which records communication events and determines computation phases based on logs.

For the LULESH traces we chose a problem size of 100, which results in one million elements per node. The number of iterations was set to 50. We run the NAMD application with the STMV molecule as input and 512 MPI tasks. The Graph500 traces were generated with a scale factor of 20, an edge factor of 16, and 512 MPI tasks.

2) Simulator: Since our simulator is currently limited to only one task per node, we use an 8x8x8 Torus with 512 nodes in total. A link between two nodes allows for a 12GB/s bandwidth in each direction. Our first experiments should provide insights about the energy saving potentials of a simple power saving strategy and the impact of certain hardware parameters, such as transition time. Therefore, we chose deterministic X-Y-Z dimension order routing as a rather simple routing algorithm in order to exclude other side effects caused by adaptive routing, for example.

IV. Results

With our experiments we want to gain first insights for the following problems:

- What is the trade off between performance loss and energy saving?
- What is the impact of transition time?
- What are the benefits of a simple power saving strategy?

A. Performance

As explained in section III, changing between different power states decreases performance due to transition times penalties. Our policy allows to cap the maximum performance loss that is tolerated while running an application. Fig. 6 depicts the progression of performance and network energy consumption for different configurations. For these experiments, we assumed a transition time of $10\mu s$. In this regard, each value of a maximum tolerated performance loss (ρ) on the x axis can be converted to a time Δt. If a link idles for a period of Δt, it is switched off until a new message arrives.

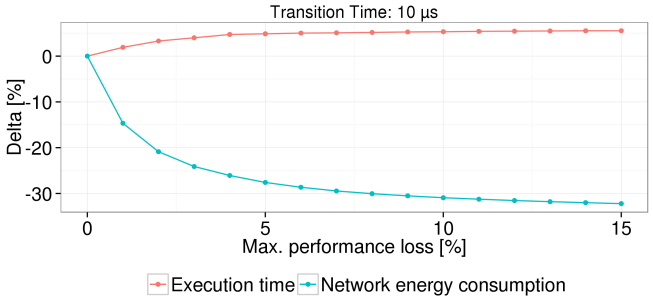


Fig. 6: Network energy vs. performance for LULESH.

This exemplary configuration indicates that actual performance losses are much smaller than the previously configured maximums. This gap between actual and maximum performance loss widens with a decreasing Δt. In contrast to actual performance, the benefits from our policy in terms of saved network energy reaches its saturation at a much smaller

Δt, respectively a larger performance loss. In addition, the saved energy outweighs the performance loss. For a maximum tolerated performance loss of 15%, which equals Δt= 66.7μs, there is an actual performance loss of 5%. However, both are small compared to more than 30% saved network energy.

B. Transition Time

When investigating energy savings in interconnection networks, the transition time becomes an important factor. A shorter transition time not only decreases the performance loss while switching between different power states, but also increases the amount of energy that can be saved.

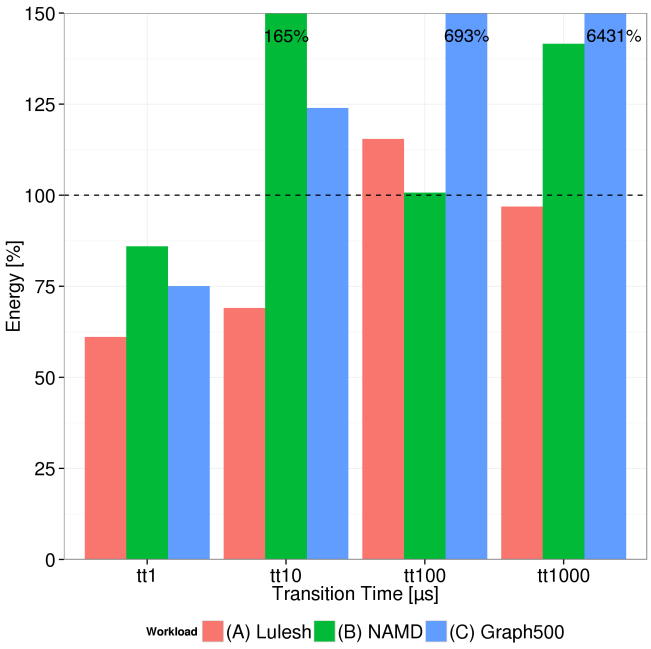


Fig. 7: Network energy for different transition times and workloads ($\Delta t = 100\mu s$).

Fig. 7 shows the effect on different transition times for different workloads on energy consumption. All values correspond to the energy consumption affiliated for a maximum performance loss of 10%. Especially for more irregular workloads the performance loss and energy consumption increases sharply. While a transition time of 10μs still enables energy saving possibilities, longer transition times seem to be inappropriate for our policy.

Note that the maximum performance loss our policy allows to tolerate is only per port. If a message propagates through the network the performance losses can sum up to a larger overall performance loss. Additionally, this policy produce some non-linear effects on power consumption, as the results of the NAMD benchmark indicate. This behavior is explained more detailed in the next section.

C. Power Savings

With our experiments, we want to focus on the detailed analysis of energy consumption for various parameters, such as transition times and workloads.

Fig. 8 shows the relative energy consumption of the LULESH benchmark, representing iterative/temporal applications. The bars indicate the energy consumption for different amounts of maximum tolerated performance loss and thereby different inactivity periods, after which links are switched off. Since these inactivity periods depend on the transition time, relative maximum performance loss is the more suitable metric for comparison reasons. Note that a larger percentage of performance loss means a shorter time of inactivity (Δt), after which power states are changed.

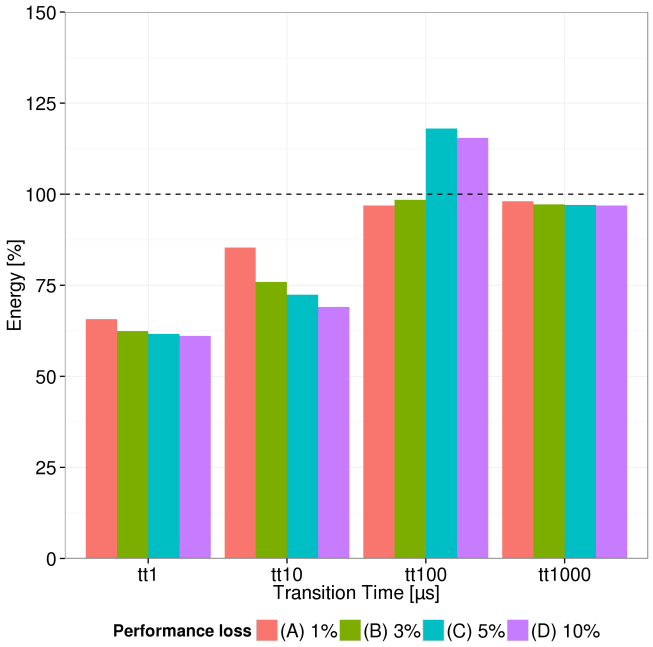


Fig. 8: LULESH: relative network energy consumption for different configurations.

These kind of applications show high potential for energy saving in the interconnection network, even with simple strategies. Nearly in every configuration energy could by saved with our policy. Only with a transition time of 100μs and a Δt less than 2000μs ($\rho \geq 5\%$) the energy consumption increases compared to the energy consumption of today's interconnects.

The fact that the power consumption decreases again for a transition time of 1000μs is caused by the also increased Δt ($10,000\mu s$). Because of this long time Δt only links, which are not utilized anyway, are switched off. This decreases the number of penalties for switching power states significantly and consequently the execution time. Additionally, a shorter execution time is also decreasing energy consumption since energy is a time dependent physical quantity.

All other configurations show the expected behavior: the shorter the transition time and the shorter Δt, the more energy is saved. The best result for LULESH is a saving of 39% for a transition time of 1μs.

The next group of workloads we investigate are iterative/non-temporal applications. Fig 9 depicts the results

for our exemplary NAMD STMV benchmark. It shows a similar trend as for LULESH, but the power saving potential is reduced. Again, the best configuration is a transition time of $1\mu s$ and 10% of maximum performance loss, which allows for energy savings of 13%. It is notable that even a transition time of $10\mu s$ only enables an energy saving potential of about 10%. Again, this emphasizes the need for shorter transition times in interconnection networks.

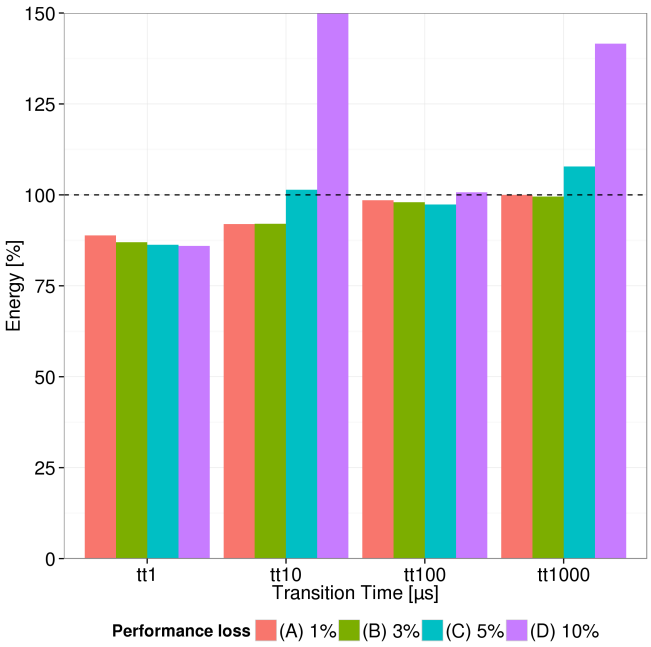


Fig. 9: NAMD: relative network energy consumption for different configurations.

Surprisingly, for $\rho = 10\%$ the energy consumption has its maximum at a transition time of $10\mu s$. Then it decreases for $100\mu s$, before it rises again at $1000\mu s$. These results can be explained by differences in the inactivity period Δt, after which links are switched off, and the time-dependency of energy. For $t_t = 10\mu s$ there is a rise of execution time from 0.35s to 0.72s, caused by more than 2.5 million transition time-penalties. Although, for $t_t = 100\mu s$ the transition time increases by one magnitude, there are only about 46,000 penalties over the network, which result in a execution time of 0.4s. Increasing the transition time by another magnitude, decreases the number of penalties only by a factor of about 2 to 18,000 and therefore, the execution time rises again to 0.56s.

The non-iterative applications class is represented by the Graph500 benchmark. The results for these experiments are shown in Fig. 10. For large numbers of maximum performance loss, the network consumes more energy than without any power saving features. This indicates that there are many small inactivity periods in the network, which allow only for power saving if the penalty of transition time is within the same magnitude as these inactivity periods. For this application up

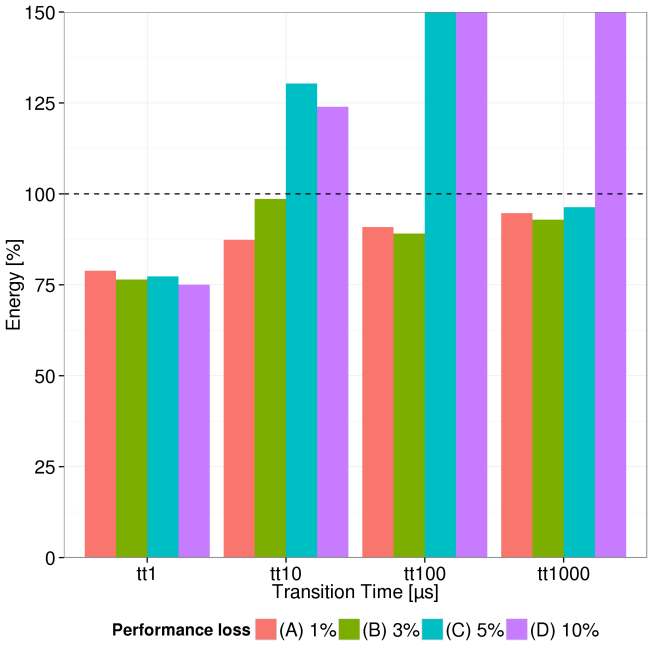


Fig. 10: Graph500: relative network energy consumption for different configurations.

to 24% of energy is potentially saved with the best parameter configuration.

Fig. 11 depicts the energy consumption for the same configuration as before, but as an average across all workloads. Since large systems are often used to run several different workloads simultaneously, this is a common scenario for most current cluster installations.

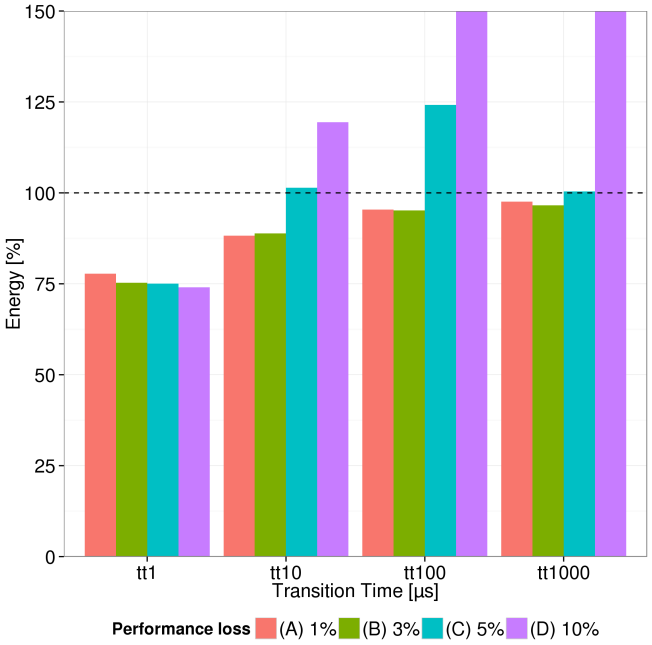


Fig. 11: Average network energy across all workloads.

As before, the combination of all three different kinds of workloads provide potential for power savings if they provide short transition times. For large transition times, the consumed energy rises sharply for an increasing ratio of maximum tolerated performance loss. This is dominated by the Graph500 benchmark, which shows a poor performance for these configurations and therefore a significant increase of energy consumption.

V. Discussion

We have implemented a simple energy saving strategy, which switches underutilized links off depending on the required performance. The results of our initial experiments have shown not only that there is a huge potential for energy saving in interconnection networks, but also that even a simple link policy allows for saving a large amount of energy. Depending on the communication pattern of a workload, 13%-39% of energy is saved with a transition time of $1\mu s$. Even a transition time of $10\mu s$ still enables energy savings of more than 30%. These promising results motivate for a further research in this area.

The main weakness of our power saving strategy we observed during our experiments is performance loss. Since this metric refers only to one port, actual loss of performance can sum up over the entire network and quickly exceed the maximum tolerated. This problem renders our policy inappropriate for an actual implementation.

One solution to this problem could be a combination of power saving features with advanced techniques, such as congestion management and adaptive routing. This approach seems promising, but also highly complex due to interactions among multiple, possibly conflicting, network management techniques.

Another approach is to allow more power states. Instead of switching links completely off, they could turn into a power state which consumes less power but is still able to transfer data. This decreases the amount of power that is saved, but also reduces the performance loss. A policy that enables switching between several power states also provides useful insights about the impact of lane respectively quad granularity.

Results show that energy savings are not necessarily linear depending on transition time. According to our policy, the inactivity period, which decides when to switch a link off, depends on the transition time. These mutual dependencies of energy, transition time, and performance loss require further observations, since we observe that longer transition times sometimes cause larger energy savings than for shorter transition times.

Also, in order to complete the picture of the impact of different design parameters, we plan to expand our experiments to indirect networks and other topologies, such as Dragonfly or Fat Tree. These topologies receive plenty of attention recently from academia and industry and allow to address network complexity while maintaining reasonable performance levels.

In the future we plan to design a network power model, which allows to predict power and energy of interconnection networks based on abstract metrics. However, a fundamental understanding of different power saving aspects, including hardware design parameters and communication metrics is mandatory to derive such a model.

VI. Related Work

Power modeling for interconnection networks is a rather new research area. An important motivation for analyzing and modeling network power is the work of [1], which argues for energy-proportional networks. The authors identify opportunities for optimization by determining the power consumption analytically. However, a detailed power model or other solutions are not provided in this work.

In the field of networks-on-chip several works exist, including [10], [11], [12], and [13]. The authors in these works are using power-gating in order to switch their routers on and off. Although they are addressing similar problems and techniques, there are fundamental differences between on- and off-chip networks.

Another very encouraging work is done by the authors of [14]. Although they are also looking at networks at other scale and integration by analyzing energy aspects and optimization for on-chip networks, their approach can be adapted for interconnection networks.

In [15], two different power saving approaches in interconnection networks with high-degree switches are introduced. The first one is dynamically switching links on and off. The second one is based on dynamically reducing network bandwidth. This reduction can be achieved by switching off a certain amount of parallel links per dimension. As opposed to our work, in which general network design parameters are addressed, they focus on specific hardware with multiple parallel links per dimension that can be configured independently. However, this work is highly inspiring our approach.

The authors of [5] are proposing an energy-efficient MPI runtime. As mentioned before, we adapted their algorithm off calculating a threshold for switching of links. Additionally, this work focuses on optimizing the MPI layer for a network with different power levers.

VII. Summary

Current interconnection networks are contributing about 20% to the overall power consumption of computing systems, such as HPC and cloud installations. Furthermore, interconnection networks seem to be neglected regarding energy savings so far. It is not surprising that today's interconnection networks, as opposed to other components, are highly energy-disproportional.

With this work, we provide first insights about energy-aware interconnection networks. We introduced a simple link policy, which switches between to different power states, depending on link utilization. We showed, that even this simple strategy allows for energy savings of up to 39%. In addition, our experiments yield first observations about the influence of the transition time on energy saving. Surprisingly, energy consumption shows a non-linear behavior for rising transition

times. We believe this to be an important design parameter, which should earn more attention in the future. However, this simple policy has crucial weaknesses. In our near future work, we want to improve this policy by either using more power states or advanced network techniques, including adaptive routing and congestion management.

VIII. Acknowledgments

We would like to thank Pedro Yebenes and Fran Andujar from the University of Castilla-La Mancha for providing their simulator and trace files and for their general support.

We also want to thank Alexander Matz and Benjamin Klenk for constructive discussions and their support in the course of this work.

Furthermore we highly acknowledge the funding by the Carl-Zeiss-Foundation we are receiving for this work.

References

[1] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *ACM SIGARCH Computer Architecture News*, vol. 38, pp. 338–347, ACM, 2010.

[2] L. A. Barroso and U. Hlzle, "The case for energy-proportional computing," *Computer*, vol. 40, pp. 33–37, Dec 2007.

[3] F. Zahn, P. Yebenes, S. Lammel, P. J. Garcia, H. Fröning, *et al.*, "Analyzing the energy (dis-) proportionality of scalable interconnection networks," in *2016 2nd IEEE International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era (HiPINEB)*, pp. 25–32, IEEE, 2016.

[4] S. Lammel, F. Zahn, and H. Fröning, "Sonar: Automated communication characterization for hpc applications," in *International Conference on High Performance Computing (ISC)*, pp. 98–114, Springer, 2016.

[5] A. Venkatesh, A. Vishnu, K. Hamidouche, N. Tallent, D. D. Panda, D. Kerbyson, and A. Hoisie, "A case for application-oblivious energy-efficient mpi runtime," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '15, (New York, NY, USA), pp. 29:1–29:12, ACM, 2015.

[6] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kal, and K. Schulten, "Scalable molecular dynamics with namd," *Journal of Computational Chemistry*, vol. 26, no. 16, pp. 1781–1802, 2005.

[7] T. Colombo, H. Frning, P. J. Garca, and W. Vandelli, "Modeling a large data-acquisition network in a simulation framework," in *2015 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 809–816, Sept 2015.

[8] T. Hoefler, "Software and hardware techniques for power-efficient hpc networking," *Computing in Science Engineering*, vol. 12, pp. 30–37, Nov 2010.

[9] F. J. Andújar, J. A. Villar, J. L. Sánchez, F. J. Alfaro, and J. Escudero-Sahuquillo, "VEF Traces: A Framework for Modelling MPI Traffic in Interconnection Network Simulators," in *Proceedings of 2015 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 841–848, Sep 2015.

[10] L. Chen, D. Zhu, M. Pedram, and T. M. Pinkston, "Power punch: Towards non-blocking power-gating of NoC routers," in *HPCA*, pp. 378–389, IEEE, 2015.

[11] L. Chen and T. Pinkston, "Nord: Node-router decoupling for effective power-gating of on-chip routers," in *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*, pp. 270–281, Dec 2012.

[12] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin, "Energy-efficient interconnect via router parking," in *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pp. 508–519, Feb 2013.

[13] R. Parikh, R. Das, and V. Bertacco, "Power-aware nocs through routing and topology reconfiguration," in *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pp. 1–6, June 2014.

[14] V. Soteriou and L.-S. Peh, "Exploring the design space of self-regulating power-aware on/off interconnection networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 393–408, Mar. 2007.

[15] M. Alonso, S. Coll, J.-M. Martínez, V. Santonja, P. López, and J. Duato, "Power saving in regular interconnection networks," *Parallel Computing*, vol. 36, pp. 696–712, Dec. 2010.