Effects of Congestion Management on Energy Saving Techniques in Interconnection Networks

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

Pedro Yebenes
Dept. of Computing Systems
University of Castilla-La Mancha
Albacete, Spain
pedroyebenes@gmail.com

Jesús Escudero-Sahuquillo
Dept. of Computing Systems
University of Castilla-La Mancha
Albacete, Spain
jesus.escudero@uclm.es

Pedro Javier Garcia
Dept. of Computing Systems
University of Castilla-La Mancha
Albacete, Spain
pedrojavier.garcia@uclm.es

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

***Abstract*—In post-Dennard scaling energy becomes more and more important. While most components in data-center and supercomputer become increasingly energy-proportional, this trend seems to pass on interconnection networks. Although previous studies have shown huge potential for saving energy in interconnects, the associated performance decrease seems to be obstructive. An increase of execution time can be caused by a decreased bandwidth as well as by transition times which links need to reconfigure and are not able to transmit data. This leads to more contention on the network than usually interconnects have to deal with.**

Congestion management is used in similar situations to limit the impact of these contentions only to single links and avoiding them to congest the entire network. Therefore, we propose combining energy saving policies and congestion management queueing schemes in order to maintain performance while saving energy. For synthetic hotspot traffic, which we use to stress the network, this combination shows promising results for multiple topologies. In 3D torus, k-ary n-tree, and dragonfly this combination provides a more than 50% lower latency and increases energy efficiency by more than 50% compared to the baseline. Although both techniques aim for fundamental different goals, none of the investigated configurations seems to suffer any disadvantages from their combination.

***Index Terms*—Energy efficiency, congestion management, energy saving policies, interconnection networks**

I. Introduction

Today's scientific applications and workloads, as well as data center applications, require for more and more computing performance. In the Post-Dennard era, this performance increase is often achieved by scaling out parallel processors. The growing number of processors results in a rising demand for communication, not only inside certain nodes but also between different nodes and even cabinets via interconnection networks. Although the number of cores increases constantly, these systems have to operate within strict power budgets due to technical, ecological, and economical reasons. Interconnection networks are usually very inefficient regarding energy, as they consume a constant amount of power independent of their load. However, they are expected to contribute substantially to the overall energy consumption of the system [1] [2] in the near future, which emphasizes the need for more energy efficient interconnects.

In recent work we have analyzed interconnection networks regarding energy proportionality. Various exascale workloads tend to minimize communication overhead. Even if equipped with fast interconnects, systems usually achieve best performance when data can be processed locally. Therefore, great effort has been made by software engineers to avoid unnecessary communication and synchronization. In [3] we have shown that in a 3D torus, links are idling more than 93% and 97% of the total execution time while running a Graph500 and NAMDstmv workload, respectively. These idle periods can be used to reduce link width or to switch them completely off in order to reduce energy consumption. Even rather simple approaches, such as an on/off or high/low policy, have enabled link energy savings of more than 85% for direct, indirect and hierarchical networks [4]. Link power is especially important since the underlying serialization technology consumes about 80% of the total switch power for small switches [3].

Energy saving policies in interconnection networks are often trading power consumption for execution time due to additional idling when links change power states [5]. However, a successful deployment of energy saving techniques in real systems has to limit the loss in performance. While our initial experiments, in general, show promising results in terms of large energy savings within reasonable increase of execution time, there are some configurations which result in a significant increase of execution time. Although these configurations are rare exceptions, they still can influence other workloads running on the same system and, thereby, impair the overall performance. Hence, for a successful deployment of such energy saving policies in real products, it is crucial to provide mechanisms to diminish these negative effects.

When bursts of data arrive at a switch, reduced bandwidth in links can easily lead to filling or even overflowing buffers. These situations are similar to contentions or congestions in networks operating close to saturation. While congestion management schemes are widely used to address such congestions, there has been no studies about the influence of congestion management and energy saving strategies on each other. Although congestion management suggests to be a good addition to energy saving in interconnects, both are independent management techniques, with partly contrary goals and possibly conflicting methods.

In this work, we are combining different state-of-the-art queueing schemes for congestion management with energy saving policies for an initial understanding of how both mechanisms influence each other. We use synthetic traffic in order to investigate this combination when the network is stressed.

In particular, we attempt to answer the following guiding questions:

- How do congestion management queueing schemes and energy saving policies influence each other?
- How does this combination affect performance and energy efficiency of interconnection networks?
- How does the underlying network topology influence this combination?

The remainder of this work is structured as follows: First, in Section II, we provide a short overview about general methods of both, congestion management and energy saving. In Section III, we introduce the settings for our experiments. The results of these experiments are provided in Section IV and discussed more detailed in Section V. Then, a brief overview about related work is shown in Section VI. Section VII concludes this work.

II. Background

The focus of this work is on the effects of combining energy saving policies in interconnects with different congestion management schemes. The first one aims for reducing power and energy consumption while usually accepting slight increases of execution times, and congestion management focuses on improving performance in situations similar to the ones caused by energy saving policies in the network. This section provides a short overview about both techniques.

A. Power Saving in Interconnects

Data movements tend to be one of the largest bottlenecks in high performance computing. This lead to two different trends: first, HPC system designers often over-provision the interconnection network in their systems in order to avoid this bottleneck. Second, software engineers try to leverage locality and reduce communication overhead of applications. Additionally, many scientific-technical workloads are iterative, resulting in a rather bursty communication pattern, in which short but intense communication phases are followed by long idling times.

As the result is a rather low network utilization, the idea of energy saving in interconnects is based on adjusting the power consumption to the current utilization, similar to other components,such as processors.

Recent works [3] [6] have shown that serial links are the main contributor to power consumption of interconnects with a relative share of more than 80%. Hence, adjusting link bandwidth according to the current utilization is the most promising approach. In contrast to most other components and their use of CMOS logic, serial links are mostly based on Current-Mode Logic (CML). In terms of power consumption, CML depends on current and, therefore, operates fundamentally different to CMOS logic, which consumes power only when switching. As a result, link-width scaling provides much higher benefits regarding energy saving compared to frequency scaling. Since today's links are already composed of multiple parallel lanes, link width can be scaled by switching these parallel lanes on and off. Consequently, the number of possible discrete power states is determined by the number of parallel lanes inside a given link.

If the processor is not able to overlap communication with other computations, exchanging data over the network is crucial for the overall performance. In order to minimize this overhead, it is important to transfer data as fast possible when it is moved to the network. Therefore, two different power states are often the best approach for link-width scaling without further insights about the applications running on the system: a fast one, which is used to transfer data and one which reduces power as much as possible when a link is idling.

In our recent research, two different policies prove themselves to be suitable [5]: the first one is a rather simple one with two different states: on and off. If a link is idling for a predefined period it is switched off and switched on again when a new message needs to be transmitted. Contrary to most other components, switching back to a higher power state comes with a performance penalty since links require re-training for clock locking and word alignment.

The second policy switches between a high power state in which all lanes are switched on and a low power state in which only one lane remains switched on. Although keeping single lanes active even when idling reduces the maximum amount of energy that can be saved, this approach aims for better performance. Especially for small messages this is an improvement. In contrast to the On/Off policy, links remain in the low power state when a small message arrives. Although the transmitting data on only one lane takes longer, it remains still faster than a link reconfiguration while switching on. When a link is in the low power state and the buffer reach a certain level at which it is faster to wait the transition time and send at the highest speed, the link switches to the high power state.

B. Congestion Management

Congestion consists of an intense traffic that clogs paths within the network [7]. This slows down traffic and degrades the network performance. The origin of congestion is contention, which occurs when several packet flows simultaneously request access to the same output port in a switch. Moreover, congestion also occurs when a destination node is not able to remove packets from the network at the speed they are received. In these cases and assuming lossless networks, i.e., packet discarding is not allowed, any packet stored in a switch or network adapter port remains blocked in a buffer until it is chosen to cross. These blocked packets delay the advance of other packets in the same buffer. If this situation persists in time, the buffers build up and finally the flow-control backpressure propagates this congestion to other switches. Eventually, congestion may spread throughout the network reaching the source nodes, increasing packet latency and degrading network performance.

In a congestion situation, not only the flows contributing to congestion (hot flows) are affected by the traffic jam. The flows not contributing to congestion (cold flows) end up advancing at the same speed as the hot ones because both share the same buffers. This situation is a particular case of the Head-of-line (HoL) blocking effect. HoL blocking occurs when a packet, which requests a busy port is blocked. This prevents other packets stored behind it in the same buffer from advancing, even if these packets are requesting free ports [8]. Therefore, in a congestion situation hot flows may produce HoL blocking to cold flows if a hot packet is blocked at the head of a queue containing cold packets.

Currently, there exist two main approaches to deal with congestion. The first one is *injection throttling* [9], which is included by the InfiniBand specification [10]. When switches detect congestion, they inform the source nodes contributing to congestion to reduce their injection rates. Once congestion is removed, its derived problems are removed too, such as the HoL blocking. However, this technique does not scale with network size, as notifications may be too slow. Therefore, there are situations in which the source nodes are warned to throttle the injection, but the congestion information is obsolete [11] or the congestion have become irreversible.

The second approach is known as queueing schemes. They prevent HoL blocking by allocating packet flows to different queues or virtual channels (VCs) [8]. There are two different families following this idea. On the one hand, some techniques explicitly identify hot flows and isolate them in dynamically-allocated VCs, such as the mechanism described for ATLAS [12], the RECN mechanism [13], or EcoCC [11]. However, they require additional and expensive resources which are not supported by current commercial interconnects. On the other hand, other techniques allocate packets from different flows to VCs according to a static mapping policy, independently of the traffic conditions, the topology or the routing algorithm. Proposals such as VOQnet [14], VOQsw [15] or DBBM [16] follow this idea. Although most of these techniques use resources available in commercial networks, they only prevent HoL blocking partially or they are not feasible in large networks (e.g. VOQnet). By contrast, other solutions are specially designed to be aware of the topology and the routing algorithm, so that HoL blocking is reduced more efficiently and/or by using fewer resources. For instance, queueing schemes such as OBQA [17] and vFTree [18] have been devised for fat-tree topologies [19] using the DESTRO [20] and D-MOD-K routing algorithms [21], respectively. IODET [22] considers the torus topology [19] and its dimension order routing algorithm. BBQ [23] is designed for the KNS topology [24] with the Hybrid-DOR routing algorithm. H2LQ [25] is tailored to Dragonfly topology using its minimal routing [26]. SF2LQ [27] is intended for Slim Fly networks with its minimal routing [28].

III. Methodology

In this section, we provide a short overview about the experimental set up and the used queueing schemes for congestion management and energy saving policies. Although both techniques can be tuned with multiple parameters, evaluations in this work are performed with default parameters. We leave further optimizations for future studies.

A. Network Simulator and Traffic Pattern

The different queuing schemes for congestion management, as well as link-level energy saving policies, are rather new concepts and not available in hardware or commercial solutions. In order to evaluate their interactions we are using the OMNeT++-based, cycle-accurate network simulator SAURON [29], which was extended with power and energy features [3].

While the employed energy saving policies are topology independent, multiple queuing schemes are using certain topology-dependent features. Hence, we select three different, widely-used topologies with similar size:

1) A 3D Torus composed of 1056 (12x11x8) nodes with dimension order routing.
2) A k-ary n-tree with 5 stages and switch radix of 8, resulting in 1024 nodes. DESTRO routing [20] is the according routing algorithm for this topology.
3) The dragonfly topology with design parameters and routing similar to [26]. The design for the experiments consists of 33 groups with switches with a radix of 15, resulting in 1056 nodes.

Radices of the switches used in these topologies differ only within one magnitude. Therefore, we assume core power for all these switches to be approximately the same. The switch core includes essential parts, such as crossbar, routing unit, and arbiter. It is consuming constantly the same amount of power, not influenced by energy saving policies.

Energy saving in interconnects utilize idling periods, most often due to computation periods, in the communication pattern in order to reduce bandwidth and thereby power consumption. The best way to show possible energy saving capabilities is using traces of real HPC applications. These traces provide realistic insights about how much the interconnect is actually utilized, and how energy saving affects the overall execution time. However, congestion management shows its impact best when the network is utilized heavily. Therefore, we decided to use synthetic hotspot traffic in order to stress the network enough to evaluate the impact of different queuing schemes: in our experiments, 1/4 of all nodes, which are randomly selected, send messages to a single "hot spot" node. The remaining nodes generate random traffic on the network. For our experiments we used a fixed load of 40%, which equals a 40% utilization of the input bandwidth.

B. Energy Saving

Energy saving in interconnects require a policy that determines when a link should switch between two power states. For our experiments, we select two topology-independent policies: The first one has an On and an Off state. Switching links completely off after idling for a certain period t_{down} enables more energy saving possibilities but increases the risk of higher performance losses since links have to reconfigure for every small message sent. The time a link is not able to send data while performing this reconfiguration is referred to $t_{transsiton}$. In order to overcome this issue, we also select a policy with a High and Low power state. Both policies are explained more detailed in [5].

Note, that synthetic traffic pattern are rather disadvantageous for these policies. The increased load on the network leads to high utilizations on the path to the hot spot node, while the random traffic effect the remaining links to switch on and off permanently. In order to get deeper insights about the interaction of energy saving policies and congestion management, we decided to configure the policies for rather aggressive power saving. All important parameters regarding energy saving are based on real hardware evalutaion and are shown in Table I.

Parameter	Value
$t_{transition}$	$10\mu s$
t_{down}	$11.1\mu s$[1]
BW_{link}	$96\frac{Gb}{s}$
$\#_{lanes}$	12

TABLE I: Network parameters.

C. Congestion Management

Congestion management and energy saving are techniques with fundamentally different approaches. In order to get insights about how they influence each other, we are testing multiple queueing schemes. While some of them are tailored to a respective topology, some others are topology-agnostic. In the following, we provide a short overview about the used queueing schemes:

VOQsw: **Virtual Output Queues at switch level** [15] can reduce HoL-blocking produced by a congestion tree in all topologies. Each input port has one virtual queue for every output port in which packets according to their requested output port are stored.

DBBM: **Destination Based Buffer Management** [16] is also applicable for all topologies. In this scheme, packets are with destination D are mapped to D modulo number of VOQs per port. For our experiments we use four virtual output queues.

IODET: **In-Order DETerministic routing** [22] is a queueing scheme tailored to direct topologies. It assigns packets to VCs according to the dimension for the next hop.

Flow2SL: This technique [30] is a topology- and routing-aware queuing scheme, specially tailored to fat-trees using deterministic routing. Flow2SL defines groups of end nodes and maps flows that have the same source group and the same destination group to the same queue (or VC), while flows that have the same source group but are addressed to different groups are mapped to different queues.

H2LQ: **Hierarchical Two-Level Queuing** [25] is designed for dragonfly networks, in which it reduces HoL-blocking and guarantees deadlock freedom. It splits traffic flows up in a standard virtual network (SVN) and an escape virtual network (EVN). The mapping to different VCs in the SVN depends on the destination, while the VCs of the EVN are used to prevent deadlocks. We selected six VCs for the SVN and two VCs for the EVN in our simulations.

IV. Results

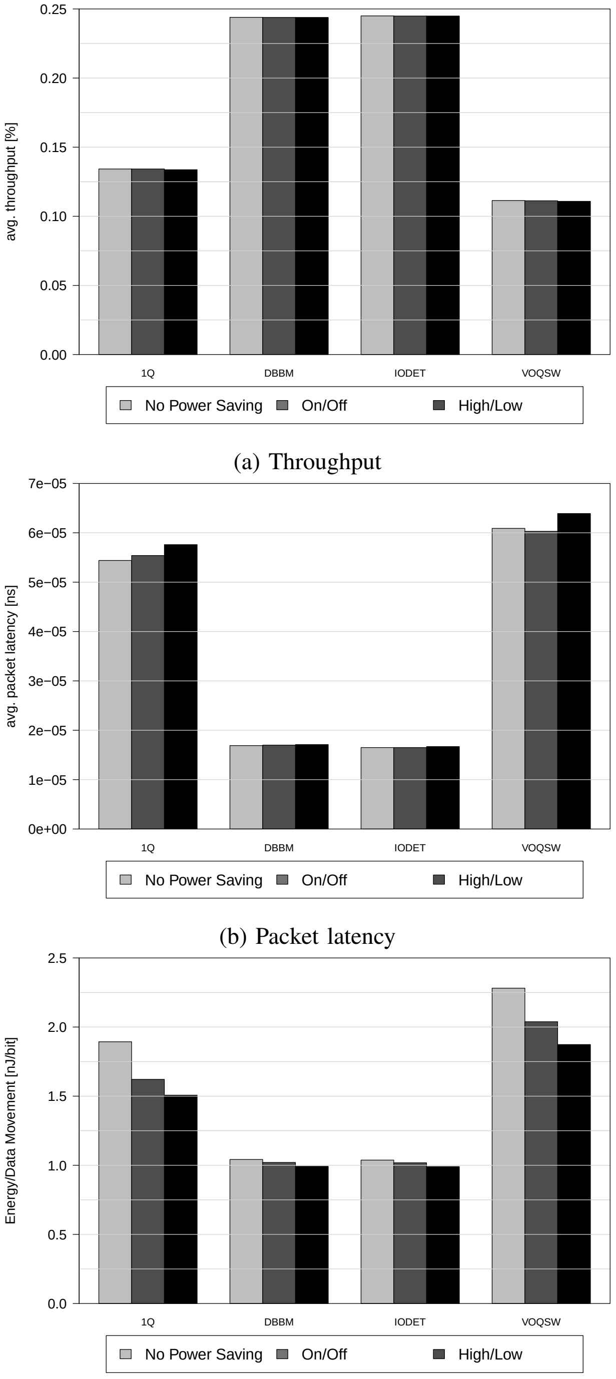
In the following, we evaluate the effects of combining congestion management with energy saving policies. Note that while the used hot spot traffic pattern is a common scenario for congestion management evaluations, it rather unsuitable for energy saving which basically tries to exploit network idling periods. Hence, we expect congestion management effects to dominate compared to energy saving policies. For all topology we select 1q, which means no congestion management and serves as a baseline, two topology-independent schemes (VOQsw and DBBM) and one scheme tailored especially to the respective topology.

In order to evaluate the effects on congestion management capabilities we are looking at throughput and packet latency. In contrast to recent work [4], [5], we use a different metric here to evaluate the energy saving policies. The implementation of synthetic traffic in the simulator requires to define a fixed simulation time. Although there are no changes in the execution time due to performance penalties in terms of transition time during link reconfiguration, there are differences in the amount of data that can be transferred in this time. Therefore, we introduce the metric energy/data to evaluate the energy saving policies. All simulations are performed with synthetic hotspot traffic, generating 40% load.

A. 3D Torus

The results regarding packet throughput, latency, and energy/data for a 3D torus topology are depicted in Figure 1. For these simulations, IODET is the topology-specific queueing scheme that we use. In this topology, VOQsw provides the worst results in performance and energy efficiency, respectively. These results are caused by the poor fit of VOQsw with the employed xyz-dimension-order routing algorithm, which routes messages always first in the x-dimension. VOQsw also divides flows in virtual channels according to the respective dimension, which deteriorates the situation since buffer size is effectively reduced to only $1/\#ports$.

[1] This equals a ρ-parameter of 90%. For further information see [5]



(a) Throughput

(b) Packet latency

(c) Energy/Data

Fig. 1: Results for 3D torus

As expected, across all configurations, especially regarding performance, there are few differences in between the different energy saving policies. Although on/off is little better regarding latency and throughput, high/low provides better results regarding energy efficiency. This seems to be caused by the random nature of synthetic traffic. Because on/off can only switch links on completely, even if only for a small message, there are more links in the network that are in ”on” state, providing high bandwidth but also high power consumption. Additionally, congestion management seems to effect energy efficiency even more than the energy saving policies. However, this is likely caused by the unfavorable experimental setup for the energy saving polices. It is also notable that for DBBM and IODET, which both provide the best performance results, there are no significant differences between the different energy saving policies. This suggests that the right combination of queueing scheme and energy saving policy can improve performance even in adverse conditions.

B. *K-ary N-tree*

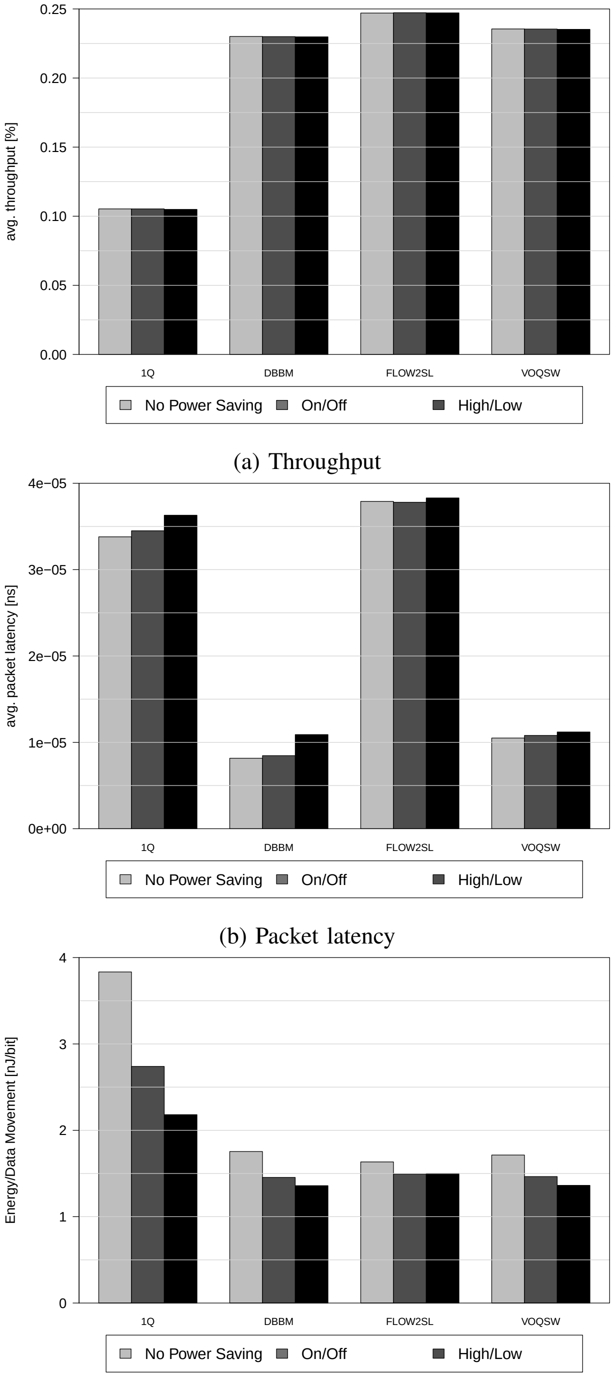
For experiments with the k-ary n-tree, depicted in Figure 2, we select Flow2SL as topology-specific queueing scheme. Note, that the configuration in these experiments is built with many small switches (radix = 8) in five stages. While all queueing schemes show good results in terms of throughput, the latency of the Flow2SL is worse than without congestion management. The reason of this variation comes from the fact that the Flow2SL mapping policy smartly balances flows among available queues or VCs in the upwards stages of the fat-tree. It maps all the flows addressed to the same group together in the same VC, although the are addressed to different destinations. Therefore, the mapping in the downward stages introduces delays as congested flows may share queues with not congested ones, if both of them are addressed to the same group. Again, on/off provides better performance, but high/low shows better results in terms of energy efficiency. In general, a 3D torus provide better energy efficiency, while a k-ary n-tree results in a lower packet latency.

C. *Dragonfly*

Figure 3 depicts the results for a dragonfly topology; in this case H2LQ queueing is used as topology-specific scheme. Overall, all combinations show the expected results. Congestion management is increasing link utilization to the effect that there are no significant differences in between different energy saving policies any more. This effect is also bolstered by the design of this topology, with few but highly utilized links between different groups. Still, even though energy saving policies do not improve energy efficiency significantly, they also not worsen the performance. Dragonfly, in general, is the most efficient topology in all studied metrics.

V. Discussion

Our results show that energy saving policies for interconnection networks in combination with advanced congestion management, such as DBBM, H2LQ, FLOW2SL, or IODET has no significant negative effect on performance and energy efficiency. Although we observe a small increase in latency in some configurations, there is a combination of an energy saving policy and a queueing scheme for every topology in which there is no significant increase of packet latency. Despite a traffic pattern that is rather obstructive for energy saving policies, in most cases they show little improvements regarding energy efficiency. Additionally, even in other configurations in

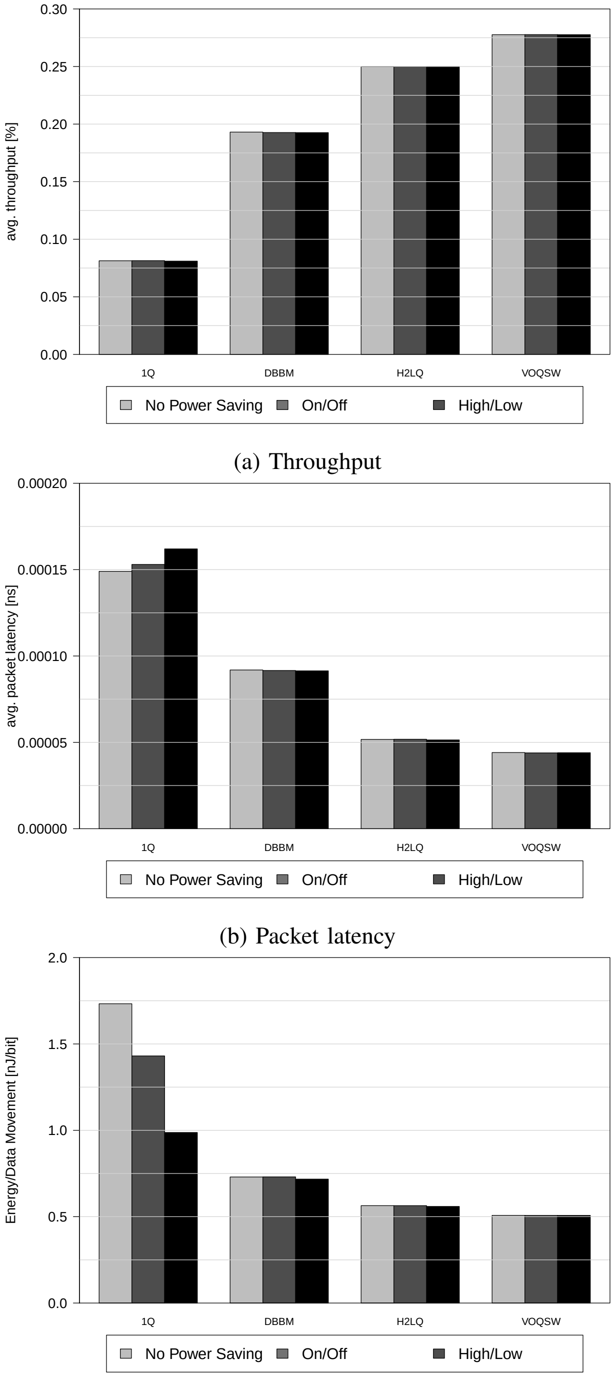


(a) Throughput

(b) Packet latency

(c) Energy/Data

Fig. 2: Results for k-ary n-tree



(a) Throughput

(b) Packet latency

(c) Energy/Data

Fig. 3: Results for Dragonfly

which they do not impact energy efficiency, they have also no impact on performance.

What is surprising, is the behavior of the high/low policy in combination with all queueing schemes, as these two policies are principally working against each other: while high/low aims for filling up buffers before sending aggregated data at high speed, congestion management tries to avoid filling buffers. Despite this, high/low provides the best energy efficiency in all configurations.

Overall, these results suggest that advanced queueing schemes are able to maximize utilization of congested links. Especially in a dragonfly network, queueing schemes are able to increase the average link utilization from 11.2% up to 28.1%. This is reflected in the differences for energy saving policies: the increased link utilization causes a decreasing potential for energy savings since links are significantly less idling, resulting in almost no power state changes. But note that synthetic traffic is a rather heavy network workload. Thus, while this initial work demonstrates that both policies are compatible, for a comprehensive and deep understanding more realistic workloads based on application traces should be used, and how the combination of policies impacts the buffer space that is required to ensure stable throughput.

Also, a more detailed understanding of the impact on energy efficiency is necessary. This can be achieved by a more extensive parameter tuning, possibly taking dynamic conditions into account, such as link utilization. Compared to hardware properties ($42.7\frac{nJ}{bit}$ at 100% network utilization), measured efficiency is worse by one order of magnitude in the best case. Of course, in a not completely energy-proportional network, the efficiency is determined by hardware and the actual network utilization. Therefore, we believe further studies on actual data movement costs in terms of energy under realistic conditions to be useful in order to evaluate sustained energy efficiency.

VI. Related Work

Although, best to our knowledge, the combination of congestion management and energy saving policies in interconnection networks has not been analyzed yet, there are studies on both topics individually. Since related work regarding congestion management is already discussed in the Section II, here we focus on energy saving in interconnection networks.

Hendry [31] introduces Asynchronous Circuit Programming (ACP). ACP provides a high-level interface for the programmer to interact with the underlying hardware. Although this approach provides benefits, such as close to optimal performance through accurate knowledge of the communication, it also shifts responsibility from hardware architecture to the programmer.

Saravanan and Cerpenter [32] evaluate the effects of Energy-Efficient Ethernet (EEE) interacting with HPC workloads. Additionally, the authors introduce a new technique which ensures a continuous clock locking and word alignment. In [33], both authors introduce the EEE-based power saving technique PerfBound. PerfBound allows to pre-define the tolerated performance degradation. This is achieved by multiple sleep states implemented in EEE and selecting the right stall-timer. In contrast to our work, PerfBound focuses on special EEE features, while our studies are technology-independent.

Kim et al. [34] propose traffic consolidation for energy-proportional high-radix networks (TCEP), a management software which centralizes traffic to fewer links in order to switch low-utilized links off. While this requires a special routing algorithm and a global network management with centralized decision making, our policies use local information that are available in each link port.

Principally, different techniques including ACP, continuous clock locking for improved transition time, traffic consolidation and technology-dependent features can be combined. In particular, link-level power saving is naturally compatible with high-level power saving methods, and we note that the combination with congestion management is a similar concept of combining low-level methods with higher level ones.

Andujar et al. [35] a new power aware routing algorithm for fat-tree and torus networks. This routing algorithm is an addition to energy saving polices. It takes into account which links are in a low power state by utilizing other links instead and thereby reducing impacts on performance. We believe this to be a extension which increases the usability of power states in hardware, such as EEE.

VII. Conclusion

Although energy saving in interconnection networks itself shows promising results decreasing energy consumption, in some configurations this comes at a price of a significant increase of execution time. Congestion management, however, is used in similar situations to improve performance and reduce dependencies inside one queue, thereby isolating congested traffic. Here we show that, indeed, energy saving policies benefit from advanced queueing schemes even in unfavorable circumstances, such as a synthetic traffic pattern. Even though in some combinations the benefit is small, energy efficiency can be improved by employing energy saving policies.

For all evaluated topologies there is at least one combination of a queueing scheme and an energy saving policy in which they provide significant improvements regarding throughput, latency and energy efficiency. While throughput increases by more than 100% for k-ary n-tree and dragonfly, latency and energy/data can be reduced by more than 50% for all studied topologies. Even in configurations in which the benefits of energy saving policies are rather neglectable, the combination with congestion management ensures that there is no negative impact on performance.

Further studies should focus on this effect in realistic scenarios using traces of real HPC applications. Although these application utilize the network much less than our used synthetic traffic pattern, the energy saving policies could also benefit from a higher utilization of single links while avoiding others. Furthermore, real hardware implementation of these queueing schemes increase energy consumption by adding more logic and increasing buffer size. For a complete picture, this should also be taken into account.

Another interesting question is the scalability of these techniques, as the number of nodes in supercomputing systems is growing constantly. The impact of this growth in size on energy saving policies, congestion management and their combination remains to be investigated.

VIII. Acknowledgements

We like to thank the Carl-Zeiss-Foundation for providing funding for this work. Additionally, we want to express our thanks to the HiPEAC network, which supported this work with a HiPEAC travel grant that enabled an on-site collaboration.

This work has been also supported by the Spanish MINECO and European Commission (FEDER funds) under the project TIN2015-66972- C5-2-R (MINECO/FEDER), and by Junta de

Comunidades de Castilla-La Mancha under the project SBPLY/17/180501/000498. Jesus Escudero-Sahuquillo is funded by UCLM and the European Commission (FSE funds), with a contract for accessing the Spanish System of Science, Technology and Innovation, for the implementation of the UCLM research program (UCLM resolution date: 31/07/2014)

References

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

[2] "The international technology roadmap for semiconductors 2.0 - executive report," 2015. http://www.semiconductors.org/clientuploads/Research_Technology/ITRS/2015/0_2015%20ITRS%202.0%20Executive%20Report%20(1).pdf.

[3] F. Zahn, P. Yebenes, S. Lammel, P. J. Garcia, and H. Fröning, "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, March 2016.

[4] F. Zahn, A. Schäffer, and H. Fröning, "Evaluating energy-saving strategies on torus, k-ary n-tree, and dragonfly," in *2018 IEEE 4th International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era (HiPINEB)*, pp. 16–23, Feb 2018.

[5] F. Zahn, S. Lammel, and H. Fröning, "On link width scaling for energyproportional direct interconnection networks," *Concurrency and Computation: Practice and Experience*, vol. 0, no. 0, p. e4439, 2019.

[6] L. Shang, L.-S. Peh, and N. K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *The Ninth International Symposium on High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings.*, pp. 91–102, Feb 2003.

[7] P. J. Garcia, "Congestion management," in *Encyclopedia of Parallel Computing*, pp. 378–386, 2011.

[8] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[9] E. G. Gran, M. Eimot, S. Reinemo, T. Skeie, O. Lysne, L. P. Huse, and G. Shainer in *24th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2010, Atlanta, Georgia, USA, 19-23 April 2010 - Conference Proceedings*.

[10] "Infiniband architecture specification volumen 1," standard, InfiniBand Trade Association, 2015.

[11] J. Escudero-Sahuquillo, E. G. Gran, P. J. García, J. Flich, T. Skeie, O. Lysne, F. J. Quiles, and J. Duato, "Efficient and cost-effective hybrid congestion control for HPC interconnection networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 107–119, 2015.

[12] M. Katevenis, D. N. Serpanos, and G. Dimitriadis, "ATLAS I: a single-chip, gigabit ATM switch with HIC/HS links arid multi-lane back-pressure," *Microprocessors and Microsystems - Embedded Hardware Design*, vol. 21, no. 7-8, pp. 481–490, 1998.

[13] P. J. García, F. J. Quiles, J. Flich, J. Duato, I. Johnson, and F. Naven, "Efficient, scalable congestion management for interconnection networks," *IEEE Micro*, vol. 26, no. 5, pp. 52–66, 2006.

[14] W. Dally, P. Carvey, and L. Dennison, "The avici terabit switch/router architecture.," in *Proceedings of Hot Interconnects Symposium VI*, pp. 41–50, 1998.

[15] M. E. Gómez, J. Flich, A. Robles, P. López, and J. Duato, "Voqsw: A methodology to reduce hol blocking in infiniband networks," in *IPDPS*, 2003.

[16] T. Nachiondo, J. Flich, and J. Duato, "Buffer management strategies to reduce hol blocking," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 739–753, June 2010.

[17] J. Escudero-Sahuquillo, P. J. García, F. J. Quiles, J. Flich, and J. Duato, "OBQA: smart and cost-efficient queue scheme for head-of-line blocking elimination in fat-trees," *J. Parallel Distrib. Comput.*, vol. 71, no. 11, pp. 1460–1472, 2011.

[18] W. L. Guay, B. Bogdanski, S. Reinemo, O. Lysne, and T. Skeie, "vftree - A fat-tree routing algorithm using virtual lanes to alleviate congestion," in *25th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2011, Anchorage, Alaska, USA, 16-20 May, 2011 - Conference Proceedings*, pp. 197–208, 2011.

[19] J. Duato, S. Yalamanchili, and L. M. Ni, *Interconnection Networks: An Engineering Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2003.

[20] C. G. Requene, *Low-Memory Techniques for Routing and Fault-Tolerance on the Fat-Tree Topology*. PhD thesis, Universidad Politécnica de Valencia, 2010. https://riunet.upv.es/bitstream/handle/10251/8856/tesisUPV3368.pdf.

[21] E. Zahavi, G. Johnson, D. J. Kerbyson, and M. Lang, "Optimized infiniband fat-tree routing for shift all-to-all communication patterns," *Concurrency and Computation: Practice and Experience*, vol. 22, pp. 217–231, Feb. 2010.

[22] R. Penaranda, C. Gomez, M. E. Gomez, P. Lopez, and J. Duato, "Iodet: A hol-blocking-aware deterministic routing algorithm for direct topologies," in *Proceedings of the 2012 IEEE 18th International Conference on Parallel and Distributed Systems*, ICPADS '12, (Washington, DC, USA), pp. 702–703, IEEE Computer Society, 2012.

[23] P. Yebenes Segura, J. Escudero-Sahuquillo, C. Gomez Requena, P. J. Garcia, F. J. Quiles, and J. Duato, "Bbq: A straightforward queuing scheme to reduce hol-blocking in high-performance hybrid networks," in *Euro-Par 2013 Parallel Processing* (F. Wolf, B. Mohr, and D. an Mey, eds.), (Berlin, Heidelberg), pp. 699–712, Springer Berlin Heidelberg, 2013.

[24] R. Peñaranda, C. G. Requena, M. E. Gómez, P. López, and J. Duato, "The k-ary n-direct s-indirect family of topologies for large-scale interconnection networks," *The Journal of Supercomputing*, vol. 72, pp. 1035–1062, Mar. 2016.

[25] P. Yébenes, J. Escudero-Sahuquillo, P. J. García, and F. J. Quiles, "Efficient queuing schemes for hol-blocking reduction in dragonfly topologies with minimal-path routing," in *2015 IEEE International Conference on Cluster Computing*, pp. 817–824, Sept 2015.

[26] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *International Symposium on Computer Architecture*, pp. 77–88, June 2008.

[27] P. Yébenes, J. Escudero-Sahuquillo, P. J. García, F. J. Quiles, and T. Hoefler, "An effective queuing scheme to provide slim fly topologies with hol blocking reduction and deadlock freedom for minimal-path routing," in *3rd IEEE International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era, HiPINEB@HPCA 2017, Austin, TX, USA, February 5, 2017*, pp. 25–32, 2017.

[28] M. Besta and T. Hoefler, "Slim fly: A cost effective low-diameter network topology," in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2014, New Orleans, LA, USA, November 16-21, 2014*, pp. 348–359, 2014.

[29] P. Yebenes, J. Escudero-Sahuquillo, P. J. Garcia, and F. J. Quiles, "Towards modeling interconnection networks of exascale systems with omnet++," in *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pp. 203–207, Feb 2013.

[30] J. Escudero-Sahuquillo, P. J. Garcia, F. J. Quiles, S.-A. Reinemo, T. Skeie, O. Lysne, and J. Duato, "A new proposal to deal with congestion in infiniband-based fat-trees," *J. Parallel Distrib. Comput.*, vol. 74, pp. 1802–1819, Jan. 2014.

[31] G. Hendry, "Decreasing network power with on-off links informed by scientific applications," in *IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum*, pp. 868–875, May 2013.

[32] K. P. Saravanan, P. M. Carpenter, and A. Ramirez, "Power/performance evaluation of energy efficient ethernet (eee) for high performance computing," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 205–214, April 2013.

[33] K. P. Saravanan and P. M. Carpenter, "Perfbound: Conserving energy with bounded overheads in on/off-based hpc interconnects," *IEEE Transactions on Computers*, vol. 67, pp. 960–974, July 2018.

[34] G. Kim, H. Choi, and J. Kim, "Tcep: Traffic consolidation for energy-proportional high-radix networks," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 712–725, June 2018.

[35] F. J. Andújar, S. Coll, M. Alonso, P. López, and J.-M. Martínez, "Powar: Power-aware routing in hpc networks with on/off links," *ACM Trans. Archit. Code Optim.*, vol. 15, pp. 61:1–61:22, Jan. 2019.