2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture

TCEP: Traffic Consolidation for Energy-Proportional High-Radix Networks

Gwangsun Kim
Arm Research
gwangsun.kim@arm.com

Hayoung Choi, John Kim
KAIST
{*hayoungc, jjk12*}*@kaist.ac.kr*

***Abstract*—High-radix topologies in large-scale networks provide low network diameter and high path diversity, but the idle power from high-speed links results in energy inefficiency, especially at low traffic load. In this work, we exploit the high path diversity and non-minimal adaptive routing in high-radix topologies to *consolidate* traffic to a smaller number of links to enable more network channels to be power-gated. In particular, we propose TCEP (Traffic Consolidation for Energy-Proportional high-radix networks), a distributed, proactive power management mechanism for large-scale networks that achieves energy-proportionality by proactively power-gating network channels through traffic consolidation. Instead of naively power-gating the least utilized link, TCEP differentiates links with the type of traffic (i.e., minimally vs. non-minimally routed traffic) on them since the performance impact of power-gating on minimal traffic is greater than non-minimal traffic. The performance degradation from the reduced number of channels is minimized by concentrating available links to a small number of routers, instead of distributing them across the network, to maximize path diversity. TCEP introduces a *shadow link* to quickly reactivate an inactive link and Power-Aware progressive Load-balanced (PAL) routing algorithm that incorporates the link power states in load-balancing the network. Our evaluations show that TCEP achieves significantly higher throughput across various traffic patterns while providing comparable energy savings for real workloads, compared to a prior approach proposed for the flattened butterfly topology.**

***Keywords*-High-radix networks; global adaptive routing; link power-gating**

I. Introduction

Interconnection networks are critical components of supercomputers and data centers as they can have a significant impact on overall system performance and cost [1]. Increasing router chip I/O bandwidth has resulted in high-radix networks [2], [3], [4], [5], [6] that provide low network diameter and high path diversity. However, high-speed links with Serializer/Deserializer (SerDes) consume a significant amount of power even when no data is being transferred to keep lane alignment [7]. Prior work [8] has shown that interconnection networks can take a significant fraction of total power in large-scale systems at low utilization, resulting in the network consuming ~50% of total power at 15% system utilization. Meanwhile, data center utilization varies widely as it is subject to daily and seasonal user demand [9]. A recent study [10] showed that in Facebook's data center network, the average link utilization is less than 1%, and 99% of all links typically have lower than 10% average utilization, as data center resource provisioning is driven by the peak demand [11]. HPC workloads also vary widely in communication intensity [12] and compute-intensive workloads can underutilize the network. In addition, as the compute efficiency improves through server consolidation and virtualization, the network also needs to perform "consolidation" to achieve energy-proportionality.

There has been a significant amount of work done on power-gating in interconnection networks. Different prior work [13], [14], [15], [16], [17], [18] have proposed power-gating channels and/or routers for on-chip networks. However, these approaches are not necessarily applicable to large-scale systems with different system constraints, including both the technology (e.g., the use of high-speed signaling with SerDes) or the interconnect architecture (e.g., high-radix topologies). Other work [19], [20], [13], [14], [21] explored power-gating in off-chip networks but focused on low-radix topologies. To the best of our knowledge, very few studies investigated power-gating of channels in *high*-radix networks. As high-radix topologies are becoming more widely used in large-scale networks [22], [23], this is one of the first work to exploit the opportunities to achieve power-efficiency in high-radix topologies through channel power-gating.

In this work, we propose a distributed, proactive power management mechanism that we refer to as *TCEP (Traffic Consolidation for Energy-Proportionality)* for high-radix networks. One of the critical components in high-radix networks is the global (non-minimal) adaptive routing [24], [25], [26], [27], [22] that exploits the high path diversity to distribute the traffic across different paths. However, since not all of the paths available are fully utilized at low traffic load, TCEP improves energy-efficiency by consolidating or merging different flows onto fewer links and power-gate other links.[1] Traffic consolidation is done through non-minimal routing; while it increases the hop count and the energy of a single packet, overall system energy efficiency is improved as the links can be power-gated to reduce idle power. One challenge in the design-space of power-gating is the connectivity of the topology and determining which links

[1]A low-radix topology might also have high path diversity. However, modifying a path can result in significant change to the topology and the routing. In comparison, given the high-path diversity in a high-radix topology with the same hop count, power-gating does not necessarily have the same impact.

2575-713X/18/$31.00 ©2018 IEEE
DOI 10.1109/ISCA.2018.00065

to power-gate. Because of the high connectivity in high-radix topologies, we show how concentrating active links to a small number of routers can minimize the loss of path diversity since the routers can function as "hubs".

An important aspect of power-gating is determining which link to power-gate [13]. A naive approach is making a local decision by choosing the link with the lowest utilization; however, this does not incorporate "global" impact from power-gating a particular link. We propose to approximate the global impact by differentiating the type of traffic (i.e., the fraction of minimally vs. non-minimally routed traffic) since re-routing minimal traffic will have more negative impact on overall performance as it increases not only packet latency but also the network bandwidth consumed. In comparison, re-routing non-minimally routed traffic through a different non-minimal path has no such impact.

However, the two key observations – concentrating links to fewer routers and minimizing re-routing of minimally routed flows – can lead to different links that should be power-gated. To reconcile the different choices at low complexity, we propose a novel algorithm that first identifies a set of links that can be turned off with minimal impact on performance and choosing the link that minimizes the re-routing of traffic among them.

Another challenge in power-gating network channels is that it is difficult to predict the future behavior of a workload. Once a link is power-gated, re-activating it incurs high latency penalty. Thus, in order to quickly respond to short-term network variations, we introduce a *shadow* link that is considered logically inactive but remains physically active for a given duration and can be logically re-activated instantly if needed. As links are power-gated, load-balancing the network channels through adaptive routing becomes more challenging as it needs to consider the link states and reduced path diversity in the network. Thus, we propose a *Power-Aware progressive Load-balanced (PAL) routing* that takes the link state into account in progressively making adaptive routing decisions to minimize the increase in latency and network bandwidth consumed while load-balancing available network links.

In particular, the contributions of this work include the following:

- We propose TCEP (Traffic Consolidation for Energy-Proportional high-radix networks) – distributed, proactive network link power-gating that exploits non-minimal routing to consolidate traffic and proactively power-gate links to improve energy efficiency.
- We show how concentrating active links to a small number of routers maximizes path diversity in high-radix networks. By considering the type of traffic (minimally vs. non-minimally routed traffic) on the link, TCEP reduces the performance impact from power-gating.
- We propose a *shadow* link that is logically inactive but physically active to quickly re-activate a link if it results in performance loss after it is deactivated.
- A novel routing algorithm is proposed that incorporates link power states in adaptively load-balancing traffic which we refer to as Power-Aware progressive Load-balanced (PAL) routing.
- We evaluate the benefits of TCEP and show that compared to SLaC, TCEP can provide significantly higher throughput for various traffic patterns (up to 7× for adversarial traffic patterns) while achieving similar energy savings as SLaC [28] for real workloads. For multiple workloads running simultaneously, TCEP is able to achieve approximately 3× improvement in energy efficiency compared to SLaC.

II. Background / Motivation

A. High-radix Topology and High-speed Links

High-radix routers provide a large number of narrow links, compared to low-radix routers that provide a small number of wide links [2]. High-radix topologies such as flattened butterfly (FBFLY) [3], HyperX [5], and Dragonfly [4] exploit the large number of narrow links to achieve low network diameter and high path diversity while reducing cost compared to other topologies such as fat-tree. A FBFLY can be obtained by combining or *flattening* the routers in the same row of a conventional butterfly network. Routers in each dimension are fully connected and multiple nodes are concentrated to each router. Dragonfly topology was proposed to reduce the number of expensive global links by grouping multiple high-radix routers as a *virtual*, very high-radix router. For the intra- and inter-group networks in a Dragonfly, FBFLY can be used to achieve low network diameter.

Modern off-chip network routers use high-speed links that consist of multiple lanes or differential pairs that operate at a higher frequency than the router to provide high bandwidth. Since the link width is narrower than router's internal datapath, data are serialized by the transmitter and deserialized by the receiver through SerDes (Serializer/Deserializer) units. To maintain signal integrity across the link, *idle* packets with a predetermined pattern need to be transmitted even when there are no data to send for the alignment of multiple serial lanes and proper clock and data recovery by the PLL (Phase-Locked Loop) unit. Thus, the SerDes represents a significant source of idle power consumption [8]. In this work, we reduce the idle power by power-gating links and reducing path diversity in the high-radix topology while minimizing the impact on overall performance.

B. Latency Sensitivity of HPC Workloads

Since link power-gating can increase network latency, its impact on system performance needs to be carefully considered. Figure 1 shows the normalized runtime of two communication-intensive workloads that we evaluated as the network latency, which includes the network interface (NIC),

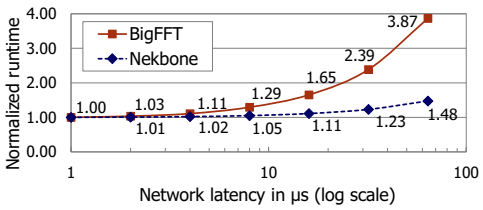


Figure 1. Sensitivity of workloads to network latency.

is varied.[2] The plot shows that even if the latency is doubled from 1 μs to 2 μs, the runtime increases by only 1-3% and further doubling the latency to 4 μs results in only 2% and 11% impact on the runtime for `Nekbone` and `BigFFT`, respectively. Tong et al. [29] analyzed the network behavior of HPC workloads and showed that communication-intensive workloads such as `BigFFT` become *load-imbalanced-bound* as a significant amount of time is spent on synchronization across nodes on low-latency networks such as Infiniband, which they assumed to have 1.3 μs latency. They also showed that increasing network latency by 8× had less than 5% impact on the runtime. Other prior work [30] similarly reported low sensitivity of `AMG` workload to network latency, and Hilgeman [31] also showed that the significant difference in latency between Infiniband and RoCE (~2.4× higher MPI latency at zero-load) had only 3-4% impact on runtime for `WRF` workload which spent 40-50% of its runtime on communication. Thus, non-minimal routing does not necessarily increase the overall runtime significantly even for communication-intensive workloads and power-gating network links can reduce total system energy.

C. Routing Table and Adaptive Routing

Routing computation logic in routers can be implemented with combinational logic or a look-up table [1]. While dedicated logic can provide lower latency, routing tables are commonly used in large-scale networks due to its flexibility. Routing tables can implement non-minimal adaptive routing by having multiple routing tables – some for minimal routes and others for non-minimal routes [32]. In global adaptive routing [24], [25], [26], a packet adapts based on network congestion between a minimal path and a non-minimal path determined from Valiant's routing algorithm [33]. Valiant's routing chooses a random intermediate node and routes first to the random node, before routing to the destination. Thus, it approximately doubles the hop count. There can be different implementations of the routing table, but in this work, we assume that each entry of the minimal routing table entry specifies the minimal output port for each destination similar to InfiniBand switches [34]. For the non-minimal routing table, we assume that each table entry represents a set of available output ports for non-minimal routes to the destination with a bit vector. We approximate the global adaptive routing such as UGAL [24] by randomly selecting one of the available output ports for each packet.

[2]The methodology is described in Section V.

III. Topology Connectivity

A. Overview

To the best of our knowledge, this is one of the first work to investigate power-gating of channels in high-radix networks. Prior work [13], [14] assumed low-radix networks and often required additional virtual channels (VCs) to avoid routing deadlock as they introduced additional non-minimal paths. In comparison, we exploit existing non-minimal paths in high-radix networks to minimize cost.

In this work, we propose TCEP for high-radix topologies by exploiting the high path diversity for link power-gating. In order to perform power management at low complexity, our approach divides a network into many *subnetworks* that are independently managed. A subnetwork consists of routers within the same dimension that are fully connected with each other. A 1D FBFLY consists of a single subnetwork while for a 2D FBFLY, each row or column of routers forms a separate subnetwork. Similarly, a "group" in a Dragonfly consists of one or multiple subnetworks depending on the intra-group network organization.

In order to maximize the opportunity to deactivate links, TCEP performs aggressive *traffic consolidation*, where multiple flows distributed to multiple links are consolidated to fewer links to deactivate other links. As a result, more links can be turned off compared to prior approaches [19], [8] that only deactivate links with very low utilization. For example, even a link with high utilization (e.g., 70%) can be deactivated by routing the traffic on the link through alternative (and non-minimal) paths. However, link deactivation needs to be carefully done to minimize its impact on performance. In the following subsections, we describe the observations and algorithms to achieve high energy-efficiency with minimal impact on performance.

B. Maintaining Connectivity

As links are power-gated, all nodes in the network need to remain connected to maintain connectivity. To ensure connectivity, we define a *root* network as a topology that consists of all nodes in the network with a subset of links to ensure connectivity within each subnetwork. Different topologies can be used for the connectivity within a subnetwork but we use star topology as the maximum hop count between any two nodes within the star topology is two – equivalent to a non-minimal route within a single dimension. The root network is defined to be always active, and thus, all other links can be turned on or off without affecting connectivity, simplifying power-gating decisions. Figure 2 shows examples of root networks for 1D and 2D FBFLY networks. In a 2D network, all row subnetworks remain connected to each other through column subnetworks, and vice versa. In this

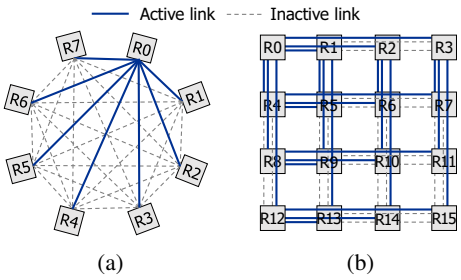


Figure 2. Root networks for a (a) 1D and (b) 2D FBFLY based on the star topology.

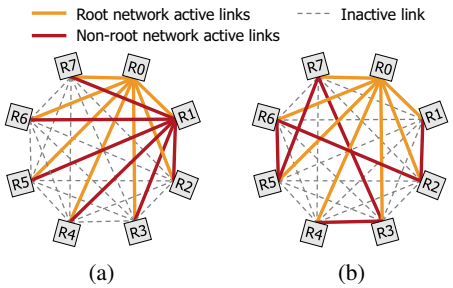


Figure 3. Path diversity comparison of (a) concentration and (b) arbitrary distribution of active links.

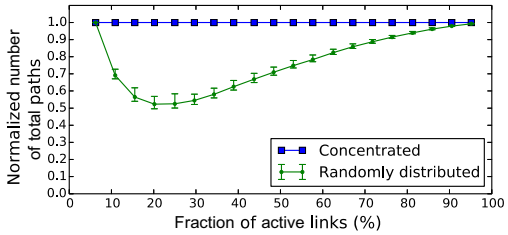


Figure 4. The total number of paths available, including minimal and non-minimal paths, with concentration and random distribution of active links to routers.

work, we define the *central hub* router as the router where links are concentrated in the subnetwork that forms the root network. In Figure 2(a), R0 is the central hub router whereas in Figure 2(b), R0, R1, R2, and R3 correspond to central hub router for each column subnetwork while R0 is also the central hub router for the top row subnetwork. While it is possible to further reduce the links in multi-dimensional root networks, there are trade-offs between additional power savings and routing complexity. For example, in Figure 2(b), all horizontal links in the second, third, and fourth rows can also be turned off while keeping the network connected. However, this requires significant changes in the routing algorithm compared to the baseline global adaptive routing while the additional power reduction it provides is marginal. Thus, we do not further reduce the number of active links s in the root network.

C. Maximizing Path Diversity

As links are deactivated, the choice of which links to deactivate impacts network path diversity in high-radix topologies. In this work, we observe that, ***as links are power-gated, concentrating the active links to a small number of routers minimizes the loss of path diversity compared to distributing them across many routers (Observation #1).*** Concentrating active links to a small number of routers make them function as "hubs" that provide paths for many source-destination pairs, resulting in a small-world network [35] and provide the multiplicative effect.

In this work, we define the *central hub* router as the router where links are concentrated in the root network. In Figure 3, R0 corresponds to the central hub router and both root links (or links that are part of the root network) are shown as well as six additional non-root links. In Figure 3(a), the non-root links are concentrated to R1 and results in at least two non-minimal paths between any source-destination pair by routing through either R0 or R1 as an intermediate router. However, if the active links are distributed across the routers (Figure 3(b)), path diversity for some source-destination pairs is reduced to one since the R0 becomes the only available intermediate router (e.g., the path between R2 and R3). As a result, the total number of paths is only 40 when active links are distributed, compared to 56 when concentrated. Figure 4 compares the total number of available paths for concentration vs random distribution of links across all source-destination pairs as the fraction of active links is increased. The results are shown for a 32-node network that is fully connected (i.e., 1D FBFLY) and we plot the results for 10,000 random samples – with the error bar indicating the maximum and the minimum number of paths. When only the root network is active (i.e., the leftmost data point), there is no difference between the two approaches, but as more links become active, link concentration provides up to 1.93× more paths than the random distribution. The gap becomes smaller as more links become active. We exploit this link concentration in the proposed TCEP to maximize path diversity as links are power-gated.

D. Minimizing the Impact of Power-gating

Another important criterion in choosing which link to turn off is its global impact caused by local power-gating decision. We identify how ***power-gating a link with higher utilization can result in a lower performance impact while improving energy-efficiency because of the global effect (Observation #2)***. If a link used by minimally routed traffic is power-gated, the traffic needs to be re-routed non-minimally and increases bandwidth consumption as well as latency. However, traffic that is already routed non-minimally can be re-routed through another non-minimal route and does not consume more bandwidth. An example in Figure 5 compares the impact of minimally and non-minimally routed traffic as

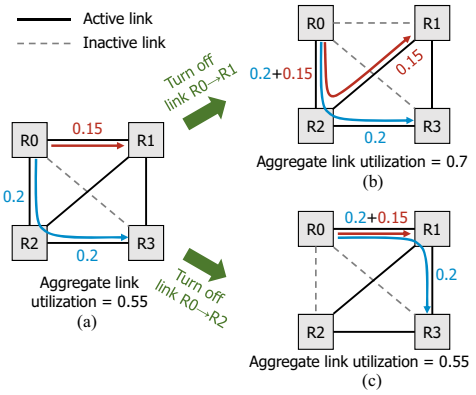


Figure 5. The impact of power-gating different links at R0. The initial state is shown in (a). Naively power-gating the least utilized link results in (b) while considering the type of traffic gives a better result in (c).

R0 chooses the link to power-gate between its two active links (one towards R1 and the other towards R2). Assume two traffic flows are being sent from R0 – minimally routed traffic to R1 and non-minimally routed traffic to R3, routed through R1. If the power-gating decision is made naively to choose the least utilized link, R0 will power-gate the link between R0 and R1 and re-route the flow on the link non-minimally through R2 (Figure 5 (b)) – resulting in an overall increase in link utilization from 0.55 to 0.7. In comparison, if the link between R0 and R2 is power-gated, the non-minimal traffic is still routed non-minimally but through R1 and results in the same aggregate link utilization. As a result, TCEP chooses the link with the least amount of minimally routed traffic in determining which link to power-gate.

IV. Network Power Management

A. Link Deactivation

While the aforementioned two observations in Section III-C and III-D can be used to guide power management, the challenge is that they often lead to different power-gating decisions. For example, a link that would have the least impact on path diversity after power-gating may be currently heavily utilized by minimally routed traffic. As a result, trade-offs between path diversity and the cost of link power-gating (i.e., increase in overall bandwidth usage due to re-routing of minimally routed traffic) need to be made. However, determining the optimal set of links to deactivate has high complexity since it requires global knowledge of the network traffic and network link state. Thus, we propose a simple link power-gating algorithm that chooses one link to deactivate at a time within each epoch while balancing the trade-offs at low complexity. The key idea is to partition the set of links for each router into *inner links* and *outer links* where the inner links remain active and have enough bandwidth to absorb the traffic from the outer links. Then, the inner links from all routers in the subnetwork will be

Algorithm 1 Pseudo-code of the link deactivation algorithm.

1: k: the number of links for a router
2: $Util_i$: utilization for link i
3: $MinTrafficUtil_i$: utilization by minimally routed traffic for link i
4: $InnerBudget$: inner links budget
5: $OuterUtil$: total utilization of outer links
6: $Boundary$: boundary between inner and outer links
7: L_{off}: the link to deactivate (no deactivation if -1)
8:
9: $InnerBudget \leftarrow Util_0$, $OuterUtil \leftarrow 0$, $L_{off} \leftarrow -1$
10: **for** $l = 1$ to $k-1$ **do** ▷ initially, all links but link 0 are outer links
11: $OuterUtil \leftarrow OuterUtil + Util_l$
12: **end for**
13:
14: **for** $l = 1$ to $k-1$ **do** ▷ Find the boundary b/w inner and outer links
15: $InnerBudget \leftarrow InnerBudget + (1 - Util_l)$
16: $OuterUtil \leftarrow OuterUtil - Util_l$
17: **if** $InnerBudget \geq OuterUtil$ **then**
18: $Boundary \leftarrow l + 1$
19: **break**
20: **end if**
21: **end for**
22:
23: **for** $l = Boundary$ to $k-1$ **do** ▷ Find the outer link with the least cost
24: **if** $MinTrafficUtil_l < MinTrafficUtil_{L_{off}}$ **then**
25: $L_{off} \leftarrow l$
26: **end if**
27: **end for**

concentrated to a small number of routers, resulting in high path diversity based on the Observation #1. In addition, since the outer links' traffic can be handled by the inner links, any outer link that has the least amount of minimally routed traffic can be turned off based on the Observation #2.

1) Inner Link Set: Without loss of generality, we assume all routers in a subnetwork are sorted in an ascending order, based on the router ID (RID). The first router in the RID list becomes the central hub router of the star topology formed within the subnetwork. The high-level algorithm used to partition links into inner and outer links at each router is summarized in Algorithm 1. The inner link set initially only contains the link towards the first router (or the most "inner") in the RID list. Additional links are added to the inner link set, one at a time, by incrementally considering the next link in the RID list until the sum of unused bandwidth from inner links (referred to as *inner links budget*) is greater than the sum of outer links' utilization (line 17-20). The outer links consists of all of the links that are not part of the inner link set. The outer links become candidates for power-gating since by definition, the inner links have the bandwidth available to handle the traffic from the outer links. An example of inner and outer links is shown in Figure 6. The first column in the table shows current link utilization and the second column shows the unused bandwidth of each link, where the two numbers in each row add up to 1. The first three links that are part of the inner links, have a budget of 1.9 while the outer links currently have total utilization of 1.2. Thus, if all outer links are turned off, their traffic can be handled by inner links and any outer link can be safely power-gated. However,

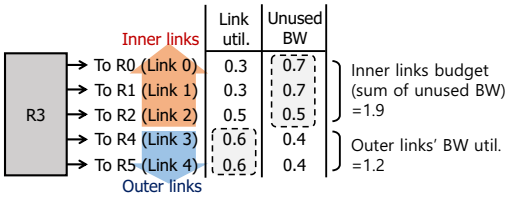


Figure 6. An example of inner and outer links. We assume R3 is fully connected to 5 other routers in the example.

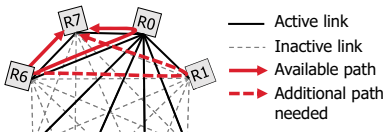


Figure 7. An example of an indirect activation request.

since there can be short-term variations in the traffic load, the unused bandwidth of a link is conservatively calculated by subtracting current utilization from a *high-water mark*, U_{hwm} ($0 < U_{hwm} < 1$), instead of 1. U_{hwm} indicates the desired upper limit of an inner link's utilization in a steady-state. If a link's current utilization is greater than U_{hwm}, the link's unused bandwidth is not added to the inner links budget. If all currently active links are highly utilized, there will not be any outer link and no link will be deactivated.

2) Deactivation Decision: After the links are partitioned into inner and outer links, the link with the least amount of minimally routed traffic among the outer links is chosen for deactivation (line 23-27), regardless of *non-minimally* routed traffic, to minimize the performance impact of power-gating. Even for adversarial traffic patterns, our mechanism can function effectively when combined with an adaptive routing algorithm [24], [25], [27] that utilizes the minimal path as much as possible. Instead of the links that have a significant amount of minimally routed traffic because of the traffic pattern, another link with smaller amount of minimally routed traffic will be deactivated. In off-chip networks, link power-gating needs to be done in the unit of a bi-directional link since the flow control is implemented across the links (e.g., send flits in one direction and receive buffer credits in the other direction). Thus, after the link to deactivate is chosen, the router sends a deactivation request across the link and the far-end router responds with either an ACK or a NACK to the deactivation request (Section IV-C).

3) Shadow Link: Deactivating a link can result in unpredicted performance degradation because of network utilization variation [36] – e.g., unexpected bursty behavior. However, reactivating a link has high cost because of the latency as well as energy cost. Thus, after a link deactivation request is acknowledged, the link is first switched to a *shadow* link state. A shadow link is considered logically inactive but physically active such that it can be immediately switched to a logically active state if needed to minimize performance impact. The routing table is updated to not use the shadow link to observe the impact of deactivation but if significant performance impact is observed, it is immediately re-activated to quickly recover from a suboptimal power-gating decision. Reactivation of a shadow link can be initiated by either router connected to the shadow link by sending a re-activation request which is always implicitly acknowledged (i.e., no ACK response will be returned). If reactivation does not occur during an epoch, the shadow link is physically deactivated after the packets already routed to use the link are transmitted. In this work, we allow only a single link to be physically turned on or off for a router within an epoch to avoid supply voltage shift [37]. Thus, only one link can be deactivated (i.e., transition into shadow state), and since it is physically turned off after an epoch, there is at most one shadow link for each router at any moment.

B. Link Activation

As traffic pattern changes, TCEP turns on inactive links to provide more network bandwidth and minimize performance degradation. With limited network bandwidth (and path diversity) from inactive links, one side-effect is that more traffic will be routed non-minimally since not all links in minimal paths are available in the network. Thus, if any of the links have utilization greater than U_{hwm} but the link is dominated by non-minimally routed traffic (i.e., more than half of the traffic on the link is non-minimally routed), the router activates an additional link. In order to activate the link that provides the highest benefit for a given traffic pattern, each router measures *virtual utilization* of all inactive links. We define virtual utilization as the potential utilization of an inactive link by minimally routed traffic if the link had been active for the last epoch. The inactive link with the highest virtual utilization will be activated. An activation request is generated and sent to the router at the other end of the link through any available (non-minimal) path. The virtual utilization of the link is embedded in the request such that the recipient can choose between multiple requests if necessary.

In addition, for adversarial traffic patterns, multiple routers need to coordinate to enable additional non-minimal paths. Since a non-minimal path includes a link between downstream routers, link activation needs to be requested to the downstream router *indirectly*. Figure 7 shows an example where R6 currently uses a minimal path and a single non-minimal path through R0. At higher traffic loads, R6 needs to enable another non-minimal path to R7 via R1, but it cannot activate the link R1-R7 since it does not belong to R6, which necessitates indirect activation. Thus, when utilization of an output link chosen for non-minimal routing is greater than U_{hwm}, an indirect activation request is sent to the router with the lowest ID that is not available as an intermediate router for current packet's destination. In the example, R6 sends an

indirect activation request to R1 such that the link between R1 and R7 is activated.

C. Handling Activation/Deactivation Requests

Since we assume a router can change only a single link's physical state in an epoch, the choice between multiple links requested for activation/deactivation needs to be carefully done. While a shadow link can be immediately activated as it only affects logical state, other requests are buffered and processed once in an epoch.

Activation requests are prioritized over deactivation requests to avoid performance degradation. Among multiple activation requests, the one with the highest virtual utilization is chosen as the link can be the most beneficial in handling the current traffic load. If there is no activation request, the router determines if it needs to activate any link as described in Section IV-B and generates the request.

If no link was activated and deactivation requests for outer links were received, the requested link with the least amount of minimally routed traffic will be deactivated. The deactivation is not allowed for an inner link since it is uncertain if other links can handle the link's traffic without turning on another link. If no deactivation request was received, the router executes Algorithm 1 to determine if any link can be power-gated. However, oscillation of the link state between active and inactive can happen if the re-routed traffic from a deactivated link is unevenly distributed among inner links due to the traffic pattern, causing the utilization of an inner link to be greater than U_{hwm}. In order to prevent such oscillation, the most recently activated link is not chosen for deactivation if any of the current router's inner links have high utilization (i.e., greater than $U_{hwm}/2$). If any of the requests was processed, an ACK is sent back to the request, and for other requests, NACKs are returned.

D. Asymmetric Activation/Deactivation Epoch

In order to quickly react to increasing traffic load as workload phase changes, it is desirable to use the shortest epoch possible for link activation. On the other hand, if link deactivation is done too frequently, links that should remain active for a longer period can be incorrectly turned off. Thus, we use two asymmetric epoch lengths for activation and deactivation. We use the physical link wake-up delay as the activation epoch length to quickly activate additional links needed, but use a deactivation epoch length that is multiple times longer than the activation epoch such that the network does not become susceptible to short-term traffic variations. Thus, separate link utilization counters are kept for both the long and short epochs.

E. Power-Aware Progressive Load-balanced (PAL) Routing

Global adaptive routing is critical to fully exploit the path diversity in high-radix topologies and is even more critical when some of the links are power-gated. Global adaptive routing consists of determining 1) a non-minimal path and 2) whether to route minimally or non-minimally. Commonly used approaches for selecting non-minimal routing (such as Valiant routing [33] in UGAL [24]) is not valid when links are power-gated since minimal routes to the random intermediate routers may not be available. In addition, the status of all global links is not known – thus, it is difficult to determine the hop count towards the intermediate router. For example, to route from any source router to a random intermediate router with UGAL, it takes at most two hops in the baseline 2D FBFLY, but when some links are power-gated, it can take up to four hops (e.g., R5 to R10 in Figure 8). Since the global link state information is not available, the hop count information needed for the adaptive routing decision is difficult to obtain. In addition, the use of shadow link needs to be carefully incorporated in the adaptive routing algorithm to observe the impact of the link deactivation but properly re-activate it as necessary.

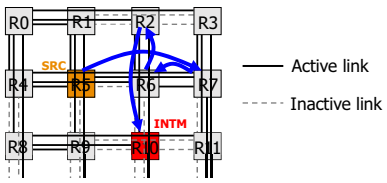


Figure 8. Problem of the baseline adaptive routing algorithms with link power-gating in a 2D FBFLY topology with UGAL. R5 is the source router and R10 is the random intermediate router.

Table I
ROUTING DECISION BASED ON THE OUTPUT PORT STATE.

MIN port	Non-MIN port credit	Routing decision
Active	Don't care	Adaptive routing based on credit count
Shadow	Available	Route non-minimally
Shadow	Not available	Reactivate the shadow link and route minimally
Inactive	Don't care	Route non-minimally

In our PAL routing, we progressively determine the intermediate router in each dimension iteratively while taking into account the link power state. At the source router, a non-minimal path towards the destination position in the current subnetwork dimension is randomly chosen among available, active paths. The adaptive decision between minimal and non-minimal path is summarized in Table I. If the output link for the minimal path is active and a non-minimal path is also available, the congestion metric (e.g., hop count, buffer occupancy [24]) is used to make the adaptive routing decision. However, if the minimal output port is in the shadow state, the router needs to avoid using the link to observe the impact of deactivating it. Thus, if there is any downstream credit in the non-minimal path, the non-minimal path is chosen. However, if the non-minimal path is fully congested with no credits available, the shadow link is changed to an active

link and is used for the packet. If the minimal output port is physically inactive, the non-minimal path is taken regardless of credit count until additional links are turned on.

When the packet finishes routing in the current dimension, the routing decision for the next dimension in the dimension order is made until the destination is reached. In comparison, prior work on indirect adaptive routing [25] does not re-evaluate the decision between minimal and non-minimal paths once a non-minimal path is chosen due to congestion. DAL [5] was also proposed to incrementally make the adaptive routing decision but requires additional VCs to avoid routing deadlock. In comparison, the proposed PAL routing only requires two VCs since dimension-order routing is used across multiple dimensions.

Updating the Routing Table: To update routing tables, any changes in the link state need to be broadcast within the subnetwork while other subnetworks are not affected. Each router updates not only its routing table but also its *link state table* that maintains the state of all links in the subnetwork for each dimension. When a link between two routers (R_x, R_y) in the subnetwork is (logically) deactivated, R_x is removed from the available intermediate routers towards R_y in current router's routing table, and vice versa. When current router's link is deactivated, the router at the other end of the link is removed from the available intermediate routers towards any routers in the routing table. The opposite changes are made to the routing table for an activated link within the subnetwork. Routing table updates do not occur frequently since a router can turn on or off only a single link in an epoch. The table update can occur at most $N_d \cdot k/2$ times at a router in an epoch, where N_d and k denote the number of dimensions and the number of routers per dimension, respectively. In addition, current in-flight packets are not affected by the delay in routing table updates since the packets can still use the shadow link as an exception or can be re-routed through the root network if it is already physically deactivated.

V. Methodology

We modified Booksim [38], a cycle-accurate interconnection network simulator, to model TCEP in detail, including the energy and delay cost of link power state transitions and control packets. We assumed a 512-node 2D FBFLY network unless otherwise mentioned.[3] For the baseline network without power-gating, we assumed 6 VCs per port, 32 flit entries per input VC buffer, and 10-cycle link latency. For TCEP, an additional VC was used for power management control packets. We provided sufficient router internal speedup such that the router microarchitecture does not become a bottleneck. For adaptive routing algorithms, we used the history window approach [27] to mitigate phantom congestion. For the baseline network, instead of the original UGAL routing, we used a modified UGAL (UGAL$_p$) that implements the progressive adaptive routing, similar to DAL (Dimensionally-Adaptive Load-balanced) routing [5] but dimension-order routing is used in traversing multiple dimensions. The network was warmed up to a steady state before measurements and we assumed single-flit packets for synthetic traffic patterns unless otherwise mentioned.

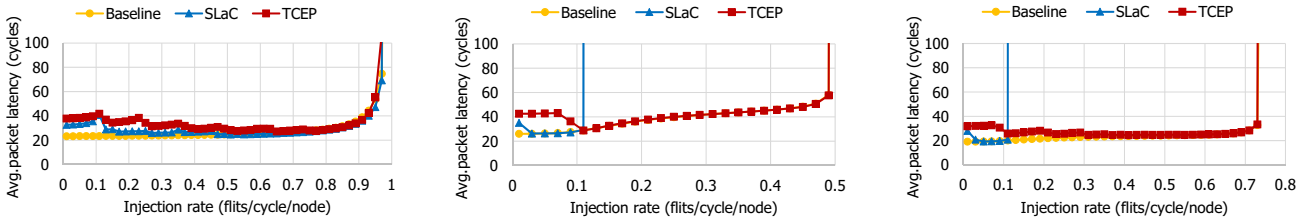
[3]In comparison, Cray Aries's intra-group network [22] is a 384-node 2D FBFLY. Further discussion on scalability is provided in Section VI-E.

Table II
HPC workloads used for evaluation.

Abbr.	Description
BigFFT	Large 3D FFT with 2D domain decomposition [42]
BoxMG	Multigrid solver based on BoxLib from combustion simulation [43]
HILO	Neutron transport evaluation and test suite [44]
FB	Fill boundary operation from PDE solver [45]
MG	Geometric multigrid v-cycle from elliptic solver [45]
NB	Nekbone: Poisson equation solver using conjugate gradient iteration with no preconditioner [46]

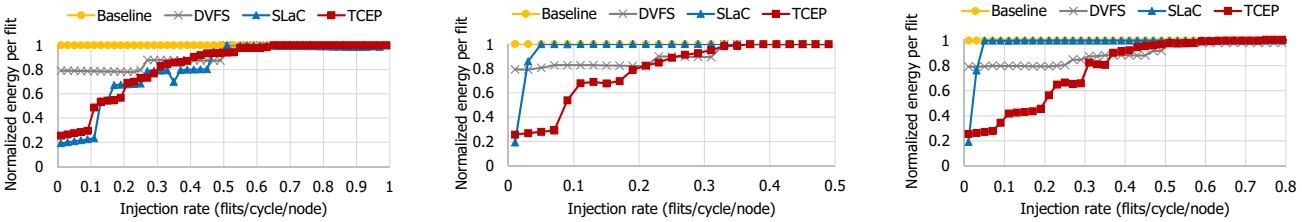
We also present results from SST/Macro simulator [30] that was integrated with Booksim. We assumed 1 GHz network frequency with node injection bandwidth of 15 GB/s and injection latency of 1 µs, and used the HPC workload traces in Table II. We assumed 48-bit flits and the maximum packet size of 14 flits similar to Cray Aries [32]. The latency sensitivity study in Section II-B assumed fixed network latency, but for other results, a cycle-accurate network model was used to faithfully evaluate the network. For power results, we report the total network link power as links dominate the power of off-chip routers [19], [36], [39] and significant resources in routers such as input and output buffers can be power-gated along with the links. We assumed 1 µs link wake-up delay [19], [40] and 1 µs was also used as the activation epoch for TCEP while the deactivation epoch was 10× longer unless otherwise mentioned. We assumed p_{real} = 31.25 pJ/bit and p_{idle} = 23.44 pJ/bit for link energy and the p_{real} to p_{idle} ratio was obtained from [8]. The values were calibrated to approximate the peak power of YARC router chip [41] for a radix-64 router such that full utilization of all ports results in ~100 W. We assumed $U_{hwm} = 0.75$ unless otherwise mentioned.

We compared TCEP to a recently proposed link power-gating mechanism called SLaC (Staged Laser Control) [28]. SLaC assumes a 2D FBFLY topology and the power-gating is done in the unit of a *stage*. A stage roughly corresponds to a row of routers in the network, and it consists of all links within the same row and all column links that connect the row with any other higher rows in the 2D FBFLY. Only stage 1 is initially active with SLaC, but as traffic load increases, additional stages are activated if input buffer utilization increases beyond a high threshold. If the router that triggered the activation of an additional stage later observes that buffer utilization is lower than a low threshold, the most recently activated stage is turned off. We used the low and high



(a) Uniform Random (b) Tornado (c) Bit Reverse

Figure 9. Latency-throughput curves of different power management mechanisms for different traffic patterns.



(a) Uniform Random (b) Tornado (c) Bit Reverse

Figure 10. Network energy with different power management mechanisms for different traffic patterns.

threshold values of 25% and 75% assumed in [28], and we favorably assumed that activation of a stage can be done in 100 cycles multiplied by the number of links in the stage.

We also compare the energy-efficiency of TCEP with an aggressive link DVFS (Dynamic Voltage and Frequency Scaling). The energy savings from DVFS was calculated based on the link utilization measured with the baseline network such that each link was considered to have operated at the lowest possible frequency that meets the link's throughput. This aggressive DVFS provides higher energy savings than prior work [8] that keeps the link utilization under target rate (e.g., 50% or 75%). We assumed three different data rates for DVFS – 1×, 2×, and 4× – similar to Infiniband and energy parameters were based on [8].

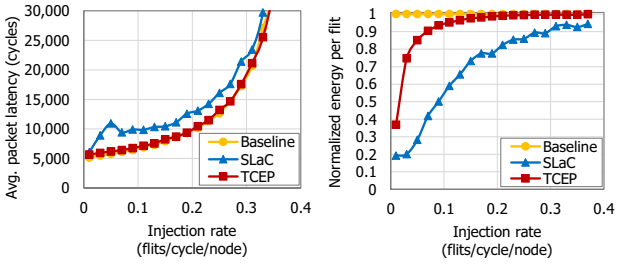
VI. Evaluation Results

A. Synthetic Traffic Result

Figure 9 shows the latency-throughput curves of the baseline and different power management mechanisms for three synthetic traffic patterns – uniform random (UR), tornado (TOR), and bit reverse (BITRV). TCEP and SLaC performed similarly for the benign pattern such as UR (Uniform Random) traffic pattern (Figure 9(a)) since they appropriately activated links as the load increased and achieved similar throughput as the baseline without any power-gating. At low traffic loads, they both kept the minimal number of links active and the average packet latency increased to 37.8 and 32.7 cycles with TCEP and SLaC, respectively, as the hop count increased by ~1.3 on average, while the baseline resulted in 23.3 cycles. As traffic load increases, both TCEP and SLaC activate more links and enable more minimal paths. At high traffic loads, all links eventually become active and the network performs identically to the baseline.

However, for TOR and BITREV traffic patterns in Figure 9(b,c), SLaC significantly underperformed and resulted in 78% and 85% lower throughput compared to the baseline. While SLaC does perform non-minimal routing based on link states, it does not support load-balancing of different active links and limits the network throughput. Fine-tuning the parameters of SLaC cannot recover the low throughput without load-balanced routing. Since different traffic patterns can be created depending on the task mapping [47], SLaC can potentially result in large performance degradation as it does not provide robust performance for adversarial traffic patterns. By contrast, TCEP achieved the same throughput as the baseline with minimal increase in packet latency since the PAL routing can properly load-balance the network even if path diversity is significantly reduced from power-gating. SLaC did achieve lower latency at very low loads for TOR and BITREV since SLaC quickly activated most of the stages but at the cost of reduced power savings.

Figure 10 shows network energy per flit normalized to that of the baseline for different traffic patterns. The step-wise increase in energy occurs since each router turns on an additional link as the utilization of active links becomes higher than U_{hwm}. For the UR (Figure 10(a)), SLaC similarly showed a step-wise increase in energy since it activates network links in the coarse-grained unit of stages and the energy benefits are similar from SLaC and TCEP. However, TCEP resulted in significantly lower energy compared to SLaC on adversarial traffic patterns (Figure 10(b,c)) as SLaC did not provide any energy savings for 5% or higher injection rate. As described earlier, all stages in SLaC become active



(a) (b)

Figure 11. Bursty traffic results showing (a) latency-throughput curves and (b) normalized energy comparison for uniform random traffic pattern.

at very low traffic loads and reduces the energy reduction benefits. The result also shows that potential energy reduction through DVFS is limited compared to TCEP since the energy consumption does not decrease in proportion to the decrease in data rate. Since changing data rate requires shorter delay than link activation or deactivation, DVFS is more suitable for addressing short-term traffic behavior while link power-gating can be more energy-efficient for long-term variation. Furthermore, it is also possible to combine TCEP with DVFS to further improve energy efficiency.

Figure 11 shows the performance and energy results with different power-gating mechanisms for a bursty uniform random traffic pattern. A very long packet size of 5,000 flits was used to generate the bursty traffic. Compared to the baseline, SLaC resulted in significantly higher latency at low traffic loads (by up to 1.81× at 0.05) since it did not sufficiently activate network links (Figure 11(a)). While its thresholds for power-gating can be adjusted to improve latency, it will reduce the energy savings for short packets shown in Figure 10. For long packets, since the latency is already high due to high serialization latency, the application can be more sensitive to such a significant increase in network latency measured in several µs. In comparison, TCEP had a very little impact on latency (only up to 1.1× at the injection rate of 0.01) as it provided enough bandwidth. While link power-gating can increase packet hop count, it only impacts head latency, which accounts for a very small fraction of packet latency for large packets. Figure 11(b) shows that SLaC can result in lower energy than TCEP but at the cost of its high impact on latency for bursty traffic.

Comparison to a Theoretical Lower Bound on Active Channels: For a simple 1D FBFLY, a *theoretical lower bound* on the number of active links can be calculated for the UR traffic pattern. Although we do not provide a detailed derivation due to space constraint, the following inequality asserts that the amount of traffic crossing the bisection should be less than or equal to the amount of bandwidth provided by active links.

$$N \times \frac{l}{2} \times \left(\frac{C_{on}}{C} + 2 \cdot \frac{C - C_{on}}{C} \right) \leq \frac{R^2}{2} \times \frac{C_{on}}{C},$$

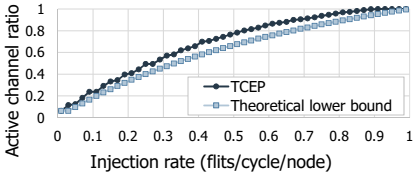


Figure 12. Comparison of the active link ratio between TCEP and a theoretical lower bound on the ratio.

where C and C_{on} denote the number of total and active channels, respectively, N is the number of nodes, R is the number of routers, and l is injection rate. Solving the inequality for C_{on} with a constraint that $C_{on} \geq R-1$ gives the lower bound on the number of active links and its ratio to the total link count is compared to that of TCEP with $U_{hwm} = 0.99$ for a 1024-node 1D FBFLY in Figure 12. As shown in the plot, TCEP closely follows the bound and the largest difference in the ratio was only 0.117 at the injection rate of 0.41. For adversarial traffic patterns, the traffic will be mostly routed non-minimally and the bandwidth usage (or the active channel count) will be nearly doubled for both TCEP and the theoretical bound.

B. Real Workload Result

Figure 13 shows the average packet latency with real workload traces normalized to that of the baseline network. The workloads are sorted in an ascending order of packet injection rate. The increase in the latency by SLaC was more pronounced with workloads with higher injection rate such as `BigFFT` and `NB`. For the two workloads, as bursty traffic increased buffer utilization across the network, SLaC activated all stages. However, as it was not able to properly load-balance different links, packet latency significantly increased by 4.5× for `BigFFT`, while TCEP resulted in 68% lower latency than SLaC. For `HILO`, both TCEP and SLaC operated at the minimal power state due to low traffic loads, but compared to TCEP, SLaC resulted in 15% higher average hop count since all routers except for those in the first stage do not have any active links within the same row. Thus, traffic between routers in the same row took more hops as the packets need to be routed through the first stage. In comparison, TCEP provides active links within all column and row subnetworks at the minimal power state through the root network and significantly reduces hop count compared to SLaC. For `BoxMG`, SLaC activated all stages for a significant portion of runtime while TCEP mostly stayed in the minimal power state. Thus, SLaC resulted in lower latency but more power consumption. Overall, SLaC significantly increased the geometric mean of packet latency across the evaluated workloads by 61% while TCEP had only 15% impact on latency. The number of control packets generated by TCEP only accounted for 0.34% on average, and 0.65% at most.

Figure 14 shows the network energy consumption normal-

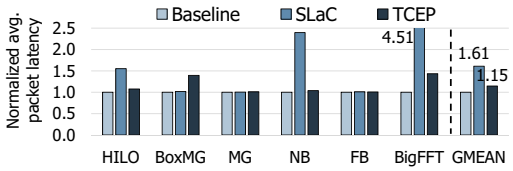


Figure 13. Average packet latency.

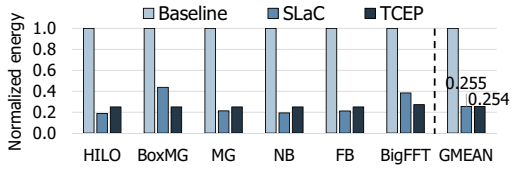
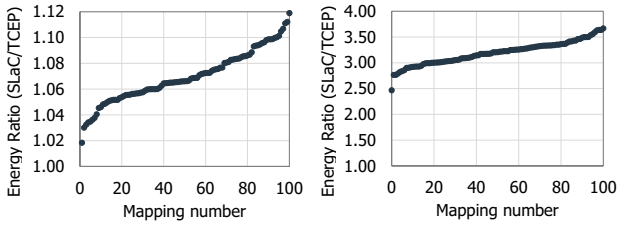


Figure 14. Total network energy.



(a) Uniform Random (b) Random Permutation

Figure 15. Energy results with two batch workloads running simultaneously with different random mappings for uniform random or random permutation traffic within each workload.

ized to that of the baseline network. Overall, both TCEP and SLaC significantly reduced energy compared to the baseline, but TCEP further reduced energy by 19% and 11% for `BoxMG` and `BigFFT`, respectively, than SLaC. Since SLaC uses a stage as the granularity of power management, it may not match the needs of the traffic pattern well and can increase power consumption unnecessarily. For example, if there is heavy traffic among routers in a single column, SLaC will activate all links in the network even if there is little traffic within other parts of the network. By contrast, TCEP only activates the links that can be well-utilized as power management is done within each subnetwork independently to better match the traffic pattern. For other workloads, SLaC results in additional energy reduction (approximately 5%) compared to TCEP since in the minimal power state, the number of active links is lower for SLaC.

In addition, TCEP did not show significant sensitivity to the epoch lengths. Increasing the activation epoch by 1.5× and 2× only increased the geometric mean of packet latency by 11% and 19%, respectively, while the energy was impacted by less than 0.2%. Among the evaluated workloads, `BigFFT` was the most sensitive, but even with 2× longer activation epoch, TCEP still resulted in 12% lower latency than that of SLaC shown in Figure 13, and its runtime increased by less than 0.5%. Varying the deactivation epoch length by -50% or +50% increased the geometric mean of latency by only 2.1% and 0.4%, respectively, and the impact on network energy was within 0.4%.

C. Multi-workload Scenario Result

Large-scale HPC systems are often shared by multiple users running different workloads [48] that share the same network. While different subnetworks in TCEP are managed independently, SLaC has limited flexibility as its stages are always turned on or off in a fixed order (e.g., beginning with the bottom stage to the top stage). Thus, if the last stage to be activated is most heavily used by some workload, all other stages need to be first turned on even if they are not used. To understand the impact of task mapping, Figure 15 shows the energy comparison of SLaC and TCEP for batch-mode synthetic traffic [1] that models two workloads running simultaneously. A 512-node network is randomly partitioned into two groups or two "jobs" and each node send traffic only within their group. Both a balanced UR traffic (Figure 15(a)) and an adversarial random permutation (RP) (Figure 15(b)) was used in the evaluation. The injection rate for the two groups are assumed to be 0.1 and 0.5 flits/cycle/node, and the batch sizes (or the amount of packets injected by the nodes within each group) are assumed to be 100,000 and 500,000 such that they finish ideally at the same time. Experiments are done with 100 random mappings and the results are sorted by the energy ratio between SLaC and TCEP. SLaC resulted in higher energy consumption compared to TCEP by up to 12% and 3.7× for the UR and RP traffic pattern, respectively. With SLaC, as the group with high injection rate creates congestion, all stages in the network are eventually triggered to be turned on. The disadvantage of SLaC is aggravated for the RP traffic pattern since different links activated are not efficiently used as its routing algorithm does not load-balance the links for adversarial traffic patterns. The execution runtime of TCEP is approximately similar to SLaC for UR while for RP, TCEP achieved 1.9× to 3.6× lower runtime than SLaC through load-balanced routing.

D. Hardware Overhead

TCEP can be implemented with low hardware overhead. For each link of a router, the utilization for each direction needs to be monitored for minimally and non-minimally routed traffic for the activation and deactivation epochs, resulting in 8 counters in addition to per-link virtual utilization. Thus, if utilization is represented with 16 bits, 144 bits per link is needed. Each router needs a buffer for control packets received, but with only one entry for each neighboring router since a router can send only one request per epoch. A request can be represented with 11 bits (8-bit router ID within the subnetwork and 3-bit control packet type). Thus, assuming a radix-64 router, the storage overhead for each router is only $(144+11)\times 64/8 \approx 1.2$ KB, which represents only ~0.7%

storage overhead compared to YARC [41]. Determining whether the minimal path is in shadow state or not can be done by simply comparing the value looked up from the minimal routing table with current shadow port number, as there is only up to one shadow link at any moment. While PAL routing algorithm revisits non-minimal routing decision in each dimension, the latency overhead is negligible considering the high per-hop latency of off-chip networks (e.g., ~100 ns [32]). The minimal and non-minimal routing table lookups can be done in parallel.

E. Scalability Discussion

TCEP can scale well to large networks due to its low complexity and overhead. The proposed power-gating algorithm has $O(p)$ complexity, where p is the number of ports in a router, since inner link set calculation requires a single sweep over the ports, and choosing the link with the least minimally routed traffic (for deactivation) or highest virtual utilization (for activation) is also $O(p)$. The storage overhead is less than 1.2 KB for a radix-64 router (Section VI-D) which can scale to a 10,648-node 2D FBFLY or even more nodes with Dragonfly [4] as Cray Aries system [22] scales up to 92,544 nodes with radix-48 routers. In Dragonfly networks, while TCEP can be used in the intra-group network, power-gating the inter-group network may not be appropriate as it can have a significant performance impact because a large number of nodes share the global links. However, to the best of our knowledge, no prior work studied power-gating of the inter-group network and it remains to be investigated. TCEP also incurs very small control packet overhead since each router can send only one request, one response (either ACK or NACK), and only $k-1$ link state broadcast packets per epoch, where k is the number of routers in a subnetwork. In addition, the PAL routing does not impact network scalability as it requires minimal changes to the routing table for UGAL.

VII. Related Work

A. Power-gating for Off-chip Networks

Prior work [21], [49] have proposed power-gating links when topologies included *trunking* where neighboring switches are connected with multiple independent links. However, such an approach is not applicable to a network with a single (bi-directional) link between neighboring routers. Power-gating for fat-tree topology has been proposed [20] but since all paths in a fat-tree are minimal, the challenges of power-gating are very different from high-radix topologies such as FBFLY where path diversity consists of both minimal and non-minimal paths. In addition, some techniques proposed in this work (e.g., shadow link, power-aware routing) can also be extended to fat-trees. As discussed earlier in Section V, the closest related work is SLaC [28] that was proposed to reduce laser cost for on-chip and off-chip nanophotonic networks through power-gating. We extend SLaC to large-scale networks for comparison but the limited flexibility of SLaC results in poor performance across diverse traffic patterns. Energy-Efficient Ethernet (EEE) [50] defines low power modes for Ethernet links but does not address how to make power-gating decisions.

B. Power-gating for Networks-on-Chip

While there has been significant work done on power-gating in NoCs, off-chip networks have different constraints compared to NoCs. NoRD [14] was proposed to decouple NoC routers from nodes such that packets can be injected even when the router is turned off by utilizing bypass paths in the NoC. Power Punch [15] exploits relatively quick wake-up latency to turn-on on-chip routers. However, off-chip network routers are not closely coupled with the compute nodes compared to NoC routers, and incur significantly longer latency to power-gate – thus, similar approaches cannot be adopted in off-chip networks. Soteriou and Peh [13] explored the design space of link power-gating in interconnection networks. The high-level issues they identified, including routing algorithms and which links to power-gate, are also applicable to this work; however, their design space focused on low-radix topology and their solutions are not necessarily applicable to high-radix topologies. Catnap [17] proposed power-gating in NoC with multiple parallel networks through channel slicing. While large-scale networks can have multiple networks (e.g., Cray BlackWidow [41]), the overhead of power-gating the entire slice of the network is too prohibitive for off-chip networks. MP3 [16] proposed power-gating for Clos topology for NoCs.

C. Orthogonal Approaches

Other power reduction techniques, such as DVFS of network links, can be combined with TCEP to further improve energy-efficiency. Kim et al. [19] leveraged both DVS and power-gating of links to save network power. Their algorithm finds a minimal set of links that satisfies network traffic to turn off other links, but power-gates links only at very low utilization and provides limited power reduction at medium-to-high loads. Shang et al. [36] proposed DVFS of links in off-chip networks and Chen et al. [51] extended it to opto-electric networks. Their power management policy used the history of input buffer utilization to predict future traffic and minimize performance degradation. Abts et al. [8] proposed a data center network based on FBFLY and a link DVFS scheme to improve network energy proportionality. They also provided high-level discussion of dynamic topologies where some links in the network are turned off to create a low-radix topology (e.g., mesh or torus) and reduce power, but the details of such mechanisms were not investigated in the work. Li et al. [40] proposed Thrifty Interconnection Network (TIN) and Network Power Shifting (NPS). The TIN changes the network power state before traffic is generated based on system event information. The NPS boosts processor performance when the network power is low to improve

performance while staying within the given system power budget. Their approaches can be combined with TCEP to improve performance.

D. Reliability and Wear-out

There has been many prior work on the reliability and interconnection networks [52], [53], [54]. The policy to concentrate active links to a small number of routers provides better robustness against link failures compared to arbitrarily distributing active links to all routers due to higher path diversity. For example, if one of the active links in Figure 3(a) fails, there is still at least one non-minimal path for any source-destination pairs. In comparison, if the link between R2 and R0 fails in Figure 3(b), there is no minimal path or two-hop non-minimal paths available between R2 and R3. If a failure occurs to the central hub router in the star topology of a subnetwork where the active links are concentrated, all paths through the router become unavailable; however, link failures have been shown to be more common than router failures [52] in large-scale networks. In order to prevent uneven wear-out among the routers, the central hub router can be periodically shifted to mitigate wear-out concerns.

VIII. Conclusion

In this work, we proposed TCEP, a distributed power management mechanism based on proactive traffic consolidation for energy-proportional high-radix networks. TCEP leverages two key observations that we exploit for power management. Our first key observation is that concentrating active links to few routers, instead of distributing the active links across different routers, maximizes path diversity while minimizing performance impact. Another key observation is that instead of naively turning off links with the lowest utilization, the type of traffic (i.e., minimally vs. non-minimally routed traffic) on the link needs to be considered in the power-gating decision. Since the two observations can lead to different power-gating decisions, we proposed a novel algorithm that balances the two criteria to maximize energy-efficiency at low complexity. In order to minimize performance loss due to short-term variation, we proposed *shadow* link that decouples the logical power state from physical power state to quickly respond to the variation as well as Power-Aware progressive Load-balanced (PAL) routing that adapts to link power state changes and reduced path diversity from power-gating. Our evaluations show that compared to SLaC, TCEP achieved significantly higher throughput across various traffic patterns (up to 7× for an adversarial traffic pattern) while providing comparable energy savings for real workloads. In addition, evaluation with multiple batch workloads running simultaneously showed that TCEP resulted in up to 3.7× lower energy consumption compared to SLaC.

Acknowledgment

This research was supported by National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (MSIP) (2015M3C4A7065647) and (2017R1A2B4011457).

References

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

[2] J. Kim *et al.*, "Microarchitecture of a high-radix router," in *Proceedings of ISCA'05*, pp. 420–431.

[3] J. Kim *et al.*, "Flattened butterfly: A cost-efficient topology for high-radix networks," in *Proceedings of ISCA'07*, pp. 126–137.

[4] J. Kim *et al.*, "Technology-driven, highly-scalable dragonfly topology," in *Proceedings of ISCA'08*, pp. 77–88.

[5] J. Ahn *et al.*, "Hyperx: Topology, routing, and packaging of efficient large-scale networks," in *Proceedings of SC'09*, pp. 41:1–41:11.

[6] M. Besta and T. Hoefler, "Slim fly: A cost effective low-diameter network topology," in *Proceedings of SC'14*, pp. 348–359.

[7] J. Moreira and H. Werkmann, *An Engineer's Guide to Automated Testing of High-Speed Interfaces*, 2nd ed. Norwood, MA, USA: Artech House, Inc., 2016.

[8] D. Abts *et al.*, "Energy proportional datacenter networks," in *Proceedings of ISCA'10*, pp. 338–347.

[9] D. Gmach *et al.*, "Workload analysis and demand prediction of enterprise data center applications," in *Proceedings of IISWC'07*, pp. 171–180.

[10] A. Roy *et al.*, "Inside the social network's (datacenter) network," in *Proceedings of SIGCOMM'15*, pp. 123–137.

[11] "America's data centers are wasting huge amounts of energy," Natural Resources Defense Council, August 2014.

[12] B. Klenk and H. Fröning, "An overview of mpi characteristics of exascale proxy applications," in *High Performance Computing*. Springer International Publishing, 2017, pp. 217–236.

[13] 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, no. 3, March 2007.

[14] L. Chen and T. M. Pinkston, "Nord: Node-router decoupling for effective power-gating of on-chip routers," in *Proceedings of MICRO'12*, pp. 270–281.

[15] L. Chen *et al.*, "Power punch: Towards non-blocking power-gating of noc routers," in *Proceedings of HPCA'15*, pp. 378–389.

[16] L. Chen *et al.*, "Mp3: Minimizing performance penalty for power-gating of clos network-on-chip," in *Proceedings of HPCA'14*, 2014, pp. 296–307.

[17] R. Das *et al.*, "Catnap: Energy proportional multiple network-on-chip," in *Proceedings of ISCA'13*, pp. 320–331.

[18] G. Kim *et al.*, "Flexibuffer: Reducing leakage power in on-chip network routers," in *Proceedings of DAC'11*, 2011, pp. 936–941.

[19] E. J. Kim *et al.*, "A holistic approach to designing energy-efficient cluster interconnects," *Computers, IEEE Transactions on*, vol. 54, no. 6, pp. 660–671, Jun 2005.

[20] M. Alonso *et al.*, "Dynamic power saving in fat-tree interconnection networks using on/off links," in *Proceedings of IPDPS'06*, April.

[21] M. Alonso *et al.*, "Power saving in regular interconnection networks built with high-degree switches," in *Proceedings of IPDPS'05*, pp. 5b–5b.

[22] G. Faanes *et al.*, "Cray cascade: a scalable hpc system based on a dragonfly network," in *Proceedings of SC'12*, November, Article 103.

[23] M. Xie *et al.*, "Tianhe-1a interconnect and message-passing services," *IEEE Micro*, vol. 32, no. 1, pp. 8–20, Jan. 2012.

[24] A. Singh, "Load-Balanced Routing in Interconnection Networks," Ph.D. dissertation, Stanford University, March 2005.

[25] N. Jiang *et al.*, "Indirect adaptive routing on large scale interconnection networks," in *Proceedings of ISCA'09*, Austin, TX, June, pp. 220–231.

[26] M. García *et al.*, "On-the-fly adaptive routing in high-radix hierarchical networks," in *Proceedings of ICPP'12*, Sept, pp. 279–288.

[27] J. Won *et al.*, "Overcoming far-end congestion in large-scale networks," in *Proceedings of HPCA'15*, pp. 415–427.

[28] Y. Demir and N. Hardavellas, "Slac: Stage laser control for a flattened butterfly network," in *Proceedings of HPCA'16*, pp. 321–332.

[29] Z. Tong *et al.*, "Fast classification of mpi applications using lamport's logical clocks," in *Proceedings of IPDPS'16*, pp. 618–627.

[30] H. Adalsteinsson *et al.*, "A simulator for large-scale parallel computer architectures," *Int. J. Distrib. Syst. Technol.*, vol. 1, no. 2, pp. 57–73, Apr. 2010.

[31] M. Hilgeman, "A technical look at ethernet vs. infiniband interconnects in hpc clusters," Dell white paper.

[32] B. Alverson *et al.*, "Cray xc series network," Cray White Paper, 2012.

[33] L. G. Valiant, "A scheme for fast parallel communication," *SIAM Journal on Computing*, vol. 11, no. 2, pp. 350–361, 1982.

[34] "InfiniBand Architecture Specification Release 1.3," InfiniBand Trade Association, March 2015.

[35] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, Jun. 1998.

[36] L. Shang *et al.*, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *Proceedings of HPCA'03*, pp. 91–102.

[37] "Hybrid Memory Cube Specification 2.1," Hybrid Memory Cube Consortium, 2014.

[38] N. Jiang *et al.*, "A detailed and flexible cycle-accurate network-on-chip simulator," in *Proceedings of ISPASS'13*, pp. 86–96.

[39] S. Derradji *et al.*, "The bxi interconnect architecture," in *Proceedings of HOTI'15*, pp. 18–25.

[40] J. Li *et al.*, "Power shifting in thrifty interconnection network," in *Proceedings of HPCA'11*, pp. 156–167.

[41] S. Scott *et al.*, "The blackwidow high-radix clos network," in *Proceedings of ISCA'06*, pp. 16–28.

[42] D. F. Richards *et al.*, "Beyond homogeneous decomposition: Scaling long-range forces on massively parallel systems," in *Proceedings of SC'09*, November, Article 60.

[43] "Exact mini apps," https://portal.nersc.gov/project/CAL/exact.htm.

[44] H. Dong *et al.*, "Quasi diffusion accelerated monte carlo," Los Alamos National Laboratory, 2011.

[45] J. Bell *et al.*, "Boxlib user guide," Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, 2013.

[46] "Characterization of the doe mini-apps," http://portal.nersc.gov/project/CAL/doe-miniapps.htm, National Energy Research Scientific Computing Center.

[47] B. Prisacari *et al.*, "Efficient task placement and routing of nearest neighbor exchanges in dragonfly networks," in *Proceedings of HPDC'14*, pp. 129–140.

[48] X.-J. Yang *et al.*, "The tianhe-1a supercomputer: Its hardware and software," *Journal of Computer Science and Technology*, vol. 26, no. 3, pp. 344–351, May 2011.

[49] J. Ahn *et al.*, "Dynamic Power Management of Off-Chip Links for Hybrid Memory Cubes," in *Design Automation Conference (DAC)*, 2014.

[50] K. P. Saravanan *et al.*, "Power/performance evaluation of energy efficient ethernet (eee) for high performance computing," in *Proceedings of ISPASS'13*, pp. 205–214.

[51] X. Chen *et al.*, "Exploring the design space of power-aware opto-electronic networked systems," in *Proceedings of HPCA'05*, pp. 120–131.

[52] P. Gill *et al.*, "Understanding network failures in data centers: Measurement, analysis, and implications," in *Proceedings of SIGCOMM'11*, pp. 350–361.

[53] P. Bodík *et al.*, "Surviving failures in bandwidth-constrained datacenters," in *Proceedings of SIGCOMM'12*, pp. 431–442.

[54] V. Liu *et al.*, "F10: A fault-tolerant engineered network," in *Proceedings of NSDI'13*, pp. 399–412.