A Case for Uni-Directional Network Topologies in Large-Scale Clusters

Michihiro Koibuchi*
*National Institute of Informatics, Japan
koibuchi@nii.ac.jp

Tomohiro Totoki†,
Hiroki Matsutani†,
Hideharu Amano†
† Keio University, Japan
blackbus@am.ics.keio.ac.jp

Fabien Chaix‡
‡ Institute of Computer Science,
Foundation for Research and
Technology - Hellas, Greece
fabien.chaix@gmail.com

Ikki Fujiwara§
§ National Institute of Information and Communications Technology, Japan
ikki@nict.go.jp

Henri Casanova¶
¶University of Hawai‘i at Manoa
henric@hawaii.edu

***Abstract*—Designing low-latency network topologies of switches is a key objective for next-generation large-scale clusters. Low latency is preconditioned on low hop counts, but existing network topologies have hop counts much larger than theoretical lower bounds. To alleviate this problem, we propose building network topologies based on uni-directional graphs that are known to have hop counts close to theoretical lower bounds. A practical difficulty with uni-directional topologies is switch-by-switch flow control, which we resolve by using hot-potato routing. Cycle-accurate network simulation experiments for various traffic patterns on uni-directional topologies show that hot-potato routing achieves performance comparable to that of conventional deadlock-free routing. Similar experiments are used to compare several uni-directional topologies to bi-directional topologies, showing that the former achieve significantly lower latency and higher throughput. We quantify end-to-end application performance for parallel application benchmarks via discrete-even simulation, showing that uni-directional topologies can lead to large application performance improvements over their bi-directional counterparts. Finally, we discuss practical issues for uni-directional topologies such as cabling complexity and cost, power consumption, and soft-error tolerance. Our results make a compelling case for considering uni-directional topologies for upcoming large-scale clusters.**

***Index Terms*—HPC clusters, interconnection networks, uni-directional network topologies, hot-potato routing**

I. Introduction

A goal for upcoming high performance computing (HPC) clusters with possibly millions of cores is to achieve low network latency, e.g., 1μs across system, as well as high bisection bandwidth [1]. Switch delays, e.g., around 100 or 200 nsec for recent InfiniBand switches, are large compared to the typical 5ns/m cable delay. To achieve low latency, a topology of switches must thus have low diameter and low average shortest path length (ASPL), both measured in numbers of switch hops.

Defined by graph theoreticians, the degree diameter problem (DDP) consists in finding the largest graph for given degree and diameter constraints. Given a graph with degree d and diameter k, the number of vertices in the graph is at most $1+d\sum_{i=0}^{k-1}(d-1)^i$ and $1+\sum_{i=0}^{k} d^i$ for bi- and uni-directional graphs, respectively, which is called the Moore bound. Not that for uni-directional graphs d is the in-degree of a vertex, which is also equal to its out-degree. Similar bounds are also known for given degree and ASPL constraints. Researchers have attempted to find solutions to the DDP with numbers of vertices that approach the Moore bound. This problem has been studied both for bi- and uni-directional graphs, and although only a few graphs that achieve the Moore bound are known [2], best known solutions for d and k values are publicly available [3].

Interestingly, the best-known uni-directional graphs typically achieve lower hop counts than the best-known bi-directional graphs. This is because: (a) Best-known uni-directional solutions achieve up to 99% of the Moore bound, while best-known bi-directional solutions often only reach about 10% of the bound; and (b) The Moore bound for uni-directional graphs is larger than that for bi-directional graphs for the same d and k values. For instance, Figure 1 shows the size of the largest known uni- and bi-directional DDP solutions for degree d=8. For large diameters, the uni-directional solution is more than 10 times larger than the bi-directional solution.

In addition to hop count, a key metric for comparing bi- and uni-directional graphs is bisection, and here again we find that uni-directional graphs outperform bi-directional graphs. For instance, consider bi- and uni-directional random graphs with 1024 vertices and degree d=8. For each graph, we estimate the bisection as the minimum number of edges between any two same-size subgraphs using the METIS graph partitioning tool [4]. We find that the bi-directional, resp. uni-directional, graph has bisection 1016, resp. 2655. For the uni-directional graph in- and out-degrees are counted separately, which means that, everything else being equal, a uni-directional graph would achieve a twofold improvement over its bi-directional counterpart. However, the improvement is still significantly larger than twofold. In general, we find that best known uni-directional DDP solutions lead to lower hop count and higher bisection than their bi-directional counterparts.

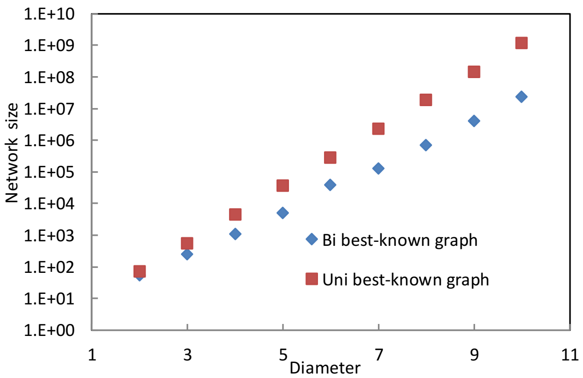

Figure 1. Diameter vs. network size for best known bi- and uni-directional DDP solutions.

Given the above, in this work we propose to use uni-directional topologies for interconnection networks in HPC clusters. To the best of our knowledge, all current clusters use bi-directional topologies. Among the relevant practical concerns discussed in this paper, a key concern is that of switch-by-switch flow control, which is necessary to avoid switch buffer overflow. Stop-and-wait or credit-based flow control is usually supported between two switches in conventional bi-directional interconnection networks. However, such flow control techniques could be problematic in our case because they assume some high-speed feedback wire on each link, which would not be available in uni-directional networks. To remove this high-speed feedback wire requirement we propose to use hot-potato routing, which does not need switch-by-switch flow control and yet can avoid buffer overflow. The main idea of hot-potato routing is that the number of input ports is equal to the number of output ports, and that an incoming packet can always be routed to a free output port [5]. Overall, this work makes the following contributions:

- We identify three previously proposed classes of uni-directional graphs, quantify their hop count and bisection properties, and propose to use such graphs as the basis for uni-directional network topologies (Section III-A);
- We propose the use of hot-potato routing for uni-directional network topologies so as to circumvent the problem of flow control with uni-directional links (but at the possible cost of increasing hop counts) and demonstrate via cycle-accurate simulation that this routing scheme compares favorably to a conventional minimal routing approach (Section III-B);
- Using discrete-event simulation, we evaluate end-to-end application performance of our three considered uni-directional topologies for ten parallel application benchmarks (Section III-C);
- We provide qualitative and quantitative comparisons between uni- and bi-directional interconnection networks in terms of closeness to the Moore bound, end-to-end-application performance, cabling, cost, power consumption, and soft error tolerance (Section IV);

Section II reviews background information and related work. Section V concludes with a summary of our findings.

II. Background and Related Work

A. Network Topologies

The most common topologies for HPC clusters are either low-degree Tori, high-degree Fat-Tree topologies, or high-degree Dragonfly topologies. As a point of reference, these topologies account for 4, 3, and 3 of the top 10 systems on the November 2017 Top500 list [6], respectively. All these topologies are deterministic, with regular structures that are amenable to custom routing schemes.

Since random graphs are known to have low diameters [7], [8], several authors have proposed low-degree random network topologies to achieve low hop counts [9], [10]. These random topologies can be generated for arbitrary network sizes, which is useful since supercomputers are increasingly being designed for ranges of system sizes (recently proposed cable-geometric and floorplan designs can be combined to deploy network topologies with arbitrary numbers of switches [11], [12]). Also, due to their lack of structure, random topologies can be easily expanded, which is useful because many systems grow in size year after year based on evolving resource demands [9].

Related to this work, the SlimFly high-radix topology [13] is based on good bi-directional DDP solutions with low diameter, i.e., MMS graphs [14] with diameter $k = 2$. As a result, a SlimFly topology can only be constructed for particular numbers of switches, e.g., n=98, 242, 338, 578, 722, 1058, 1682, 1922, 2738 and 5618, for particular degrees d=11, 17, 19, 25, 29, 35, 43, 47, 55 and 79.

B. Routing

Many standard network topologies are amenable to custom routing schemes, most of which ensure paths with minimal hop counts. For example, in k-ary n-cubes, one option is dimension-order routing, by which a packet is routed along each of the n dimensions in a pre-determined sequence.

In the case of unstructured network topologies, such as fully random topologies or the uni-directional topologies we propose in this work, topology-agnostic routing, such as up*/down* routing, is usually required [15]. In some cases, minimal deadlock-free paths can be implemented with the use of virtual channels. In this case, overlapping paths on a link can generate network contention, meaning that while a packet is going through a link, another packet may be waiting for this link to be available. Deflection routing, a.k.a. “hot-potato routing”, is a routing scheme that can avoid packet contention by enforcing that all incoming packets be transferred to different output ports on a switch. The drawback is that, as network traffic load increases, so does the probability that packets will follow non-minimal paths. Nevertheless, an advantage of hot-potato routing is that is obviates the need for stop-and-wait flow control between switches, because buffer overflow never occurs. This property is key for making uni-directional interconnection networks feasible in practice.

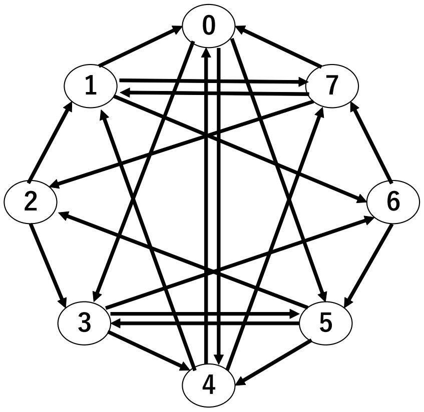

Figure 2. An example Imase graph for n=8, a=b=1, d=3.

C. *Uni-directional Interconnection Networks*

Although uni-directional Multistage Interconnection Networks (MINs) have been proposed for parallel computers (e.g., the Omega network), to the best of our knowledge all current production systems use bi-directional links. In this work we make a case for "reviving" interest in uni-directional interconnects.

Most large-scale clusters (HPC, datacenters) currently use optical links due to steadily decreasing costs and increasing length. An optical Ethernet cable (e.g., 10GBASE-LR, 10GBASE-SR, 40GBASE-LR4) consists of two opposite links; sender and receiver have separate light sources and separate transmission lines. One possible way to implement a uni-directional topology from currently available commodity components is to use such bi-directional optical links as uni-directional links by physically unpacking the bundle of the two opposite links. But it is then no longer possible to implement link-level error correction by sending requests for data retransmission. However, given current and expected technological advances, this should not be an issue for upcoming systems with FEC (see Section IV-D for a discussion). It is also no longer possible to use standard flow control techniques that required a feedback wire (e.g., stop-and-wait, credit). In this work we propose to use hot-potato routing as a solution to this problem (see Section III-B).

In this work we consider uni-directional topologies of switches and assume that hosts are connected to switches via bi-directional links. In other words, the only uni-directional links are inter-switch links.

III. Uni-directional Interconnection Networks

A. *Network Topologies*

We consider three classes of uni-directional graphs that can be generated for arbitrary network sizes and have low hop counts. Imase et al. [16], [17] have proposed non-random, uni-directional graphs with diameter at most one hop larger than the Moore bound. These graphs, which we simply call *Imase*, are constructed by connecting vertex i to vertex $j = ((ai + b)d + \alpha) \mod n$, $\alpha = 0, 1, \ldots, d-1$, where d is the degree, n is the number of vertices, and parameters a and b are user-defined but constrained by d and n. In our evaluations we find that values of a and b have little impact on hop counts, and so we use $a = b = 1$. Figure 2 shows an example Imase graph for $n = 8, a = b = 1, d = 3$ (we omit loop-back edges). We also consider MDDP (Modified DDP solutions) graphs, which are generated by using known DDP solutions as starting points and employing simple randomized heuristics to add/remove vertices to obtain graphs with arbitrary number of vertices [18]. Finally, we consider fully random graphs, which we call *Random*. These graphs are generated by inserting edges between vertex pairs picked using a uniform distribution, but enforcing that all vertices have the same degree.

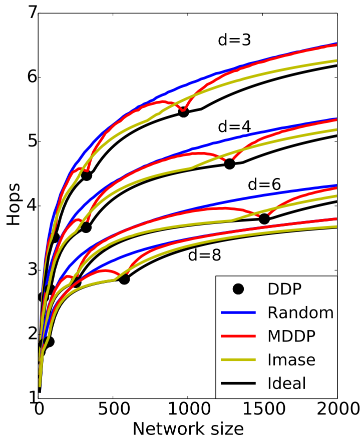

Figure 3. ASPL vs. graph size for uni-directional best known DDP solutions (DDP), Random graphs, MDDP graphs, Imase graphs, and Moore bounds (Ideal), for low-degree scenarios ($3 \leq d \leq 8$).

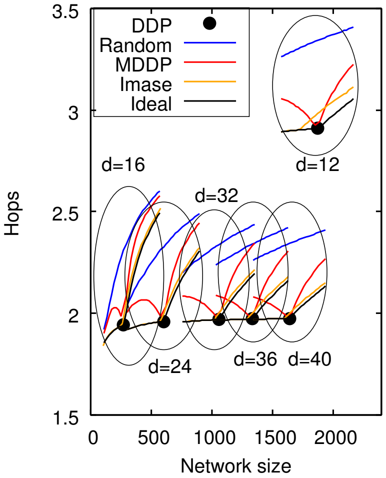

Figure 4. ASPL vs. graph size for uni-directional best known DDP solutions (DDP), Random graphs, MDDP graphs, Imase graphs, and Moore bounds (Ideal), for high-degree scenarios ($12 \leq d \leq 40$).

MDDP and Imase graphs have diameter and ASPL close to theoretical lower bounds. Figure 3, resp. Figure 4, shows ASPL vs. graph size for uni-directional Imase, MDDP, and purely random graphs, for low-degree, resp. high-degree, scenarios. Best known DDP solutions are also shown on both figures as black dots. The Moore bound is shown as a black curve (Ideal). These results show that Imase graphs generally lead to low hop counts, close to the Moore bound. Imase graphs have lower hop count than Random graphs across the board. They are only beaten by MDDP graphs for relatively low-degree scenarios for network sizes around that of best known DDP solutions. But for other network sizes, MDDP graphs have significantly higher hop counts, on part with that of Random graphs.

In spite of good hop count results, Imase graphs generally lead to lower bisection. Computing the minimum number of edges between any two same-size subgraphs is an NP-complete problem, and would thus requires exponential time to compute. However, using METIS [4], a graph partitioning tool, we can estimate the bisection tractably for graphs with thousands of vertices. For instance, for 1,024-vertex graphs, METIS provides bisection estimates at 1598, 2498, and 2655, for Imase, MDDP, and Random graphs, respectively. Results for other graph sizes lead to similar observations. Overall, Imase graphs achieve significantly lower bisection that Random and MDDP graphs, among which Random graphs have a marginal advantage over MDDP graphs. One might argue that MDDP graphs provide a good compromise between hop count (better than or equivalent to Random, often worse than Imase) and bisection (better than Imase, worse than Random).

B. Routing

1) Base routing schemes: MDDP and Random topologies are unstructured, and therefore require the use of topology-agnostic routing schemes [15]. By contrast, a custom routing scheme is known for Imase topologies. For completeness, we briefly describe this scheme hereafter in the simplest case: $a = 1$, $b = 0$. In this case, recall that vertex i is connected to vertex j with the condition

$$j = id + \alpha \pmod n, \;\; 0 \leqq \alpha \leqq d-1 \;,$$

where n is the number of vertices and d the degree. Let $J(i,l)$ be the set of vertices that can be reached from vertex i in l hops. $J(i,1)$ is trivially:

$$J(i,1) = \{j \,|\, j \equiv id + \alpha_1 \pmod n, \;\; 0 \leqq \alpha_1 \leqq d-1\}.$$

By induction, $J(i,j)$ is expressed as

$$J(i,l) = \{j \,|\, j \equiv id^l + \alpha_1 d^{l-1} + \cdots + \alpha_{l-1} d + \alpha_l \pmod n, \\ 0 \leqq \alpha_1, \alpha_2, \cdots, \alpha_l \leqq d-1\}$$

Given an integer k, we can express $j - id^k$ as a polynomial function of d as follows

$$j - id^k \equiv \alpha_1 d^{k-1} + \cdots + \alpha_{k-1} d + \alpha_k \pmod n, \\ 0 \leqq \alpha_1, \alpha_2, \cdots, \alpha_k \leqq d-1.$$

Consider now a packet that must be routed from vertex i to vertex j. It turns out that by selecting the α_l-th output link for the l-th hop, where the α_l's are the coefficients of the above polynomial, the packet is routed from vertex i to vertex j in k hops. To find the minimal path, one simply searches for the lowest $k \geq 1$ that satisfies $0 \leqq j - id^k \pmod n \leqq d^k - 1$.

2) Hot-potato routing: Regardless of the routing scheme (topology-agnostic or custom), enforcing that packets take minimal paths requires additional feedback links to implement switch-by-switch flow control so as to avoid buffer overflow. As discussed previously, this is not possible with purely uni-directional topologies. To address this problem, we propose the use of the hot-potato routing scheme. In this scheme, each packet has to move constantly, so when the output link along a minimal path is not available, then the packet is "deflected", i.e., forwarded to another link. Although packets may then take non-minimal paths, thus increasing latency, no flow control is needed since each packet moves at each clock cycle, meaning that the input channel of the next-hop switch is always guaranteed to be empty. Furthermore, again since packets move at every clock cycle, deadlock-freedom is guaranteed even with cyclic dependencies [5].

In our proposed scheme, even though hot-potato routing is used, priority is given to shortest paths (determined from custom routing or topology-agnostic schemes) when available. If not available, then alternate paths are used. Note that topologies typically include multiple minimal paths between vertex pairs. For instance, for the Imase topology, when n is not a multiple of d, multiple shortest paths between node pairs exist and can be identified. In this case, hot-potato routing may often still lead to minimal path routing. We propose that a hop counter be stored in the header flit of each packet, so that output links along known minimal paths can be selected with higher priority for packets with higher hop counts. This scheme has the desirable side-effect of avoiding livelock.

3) Cycle-accurate network simulation: We use a flit-level network simulator written in C++ [10] to evaluate the network latency and throughput of our two proposed routing schemes on uni-directional MDDP topologies of degree $d = 6$ and $d = 8$. The number of switches is set to 256, and the number of hosts is set to 1,024. Each simulated switch is configured to use virtual cut-through switching. A header flit transfer requires over 100ns, which includes routing, virtual-channel allocation, switch allocation, and flit transfer from an input channel to an output channel through a crossbar. The flit injection delay and link delay together are set to 20ns. We present results for both minimal routing and hot-potato routing. For hot-potato routing, we use a simple implementation that relies on packet injection limitation. More precisely, each host limits the injection of packets to each switch so that we maintain the non-blocking behavior of incoming input packets from neighboring switches. We use four virtual channels in all our experiments so as to allow for a fair comparison of the routing schemes.

We simulate three synthetic traffic patterns that define source-and-destination pairs: *random uniform*, *matrix transpose* and *bit reversal*. These traffic patterns are commonly used for measuring the performance of large-scale interconnection networks [19]. Each switch is connected to four hosts, and the hosts inject packets into the network independently of each other. In each synthetic traffic the packet size is set to

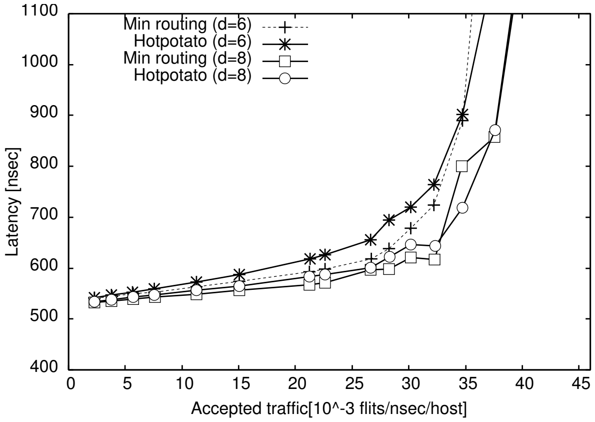

Figure 5. Latency vs. accepted traffic for uni-directional MDDP network topologies with minimal routing and hot-potato routing (Uniform).

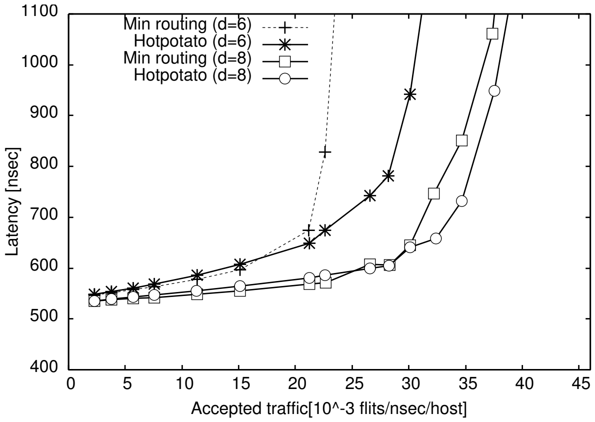

Figure 6. Latency vs. accepted traffic for uni-directional MDDP network topologies with minimal routing and hot-potato routing (Matrix Transpose).

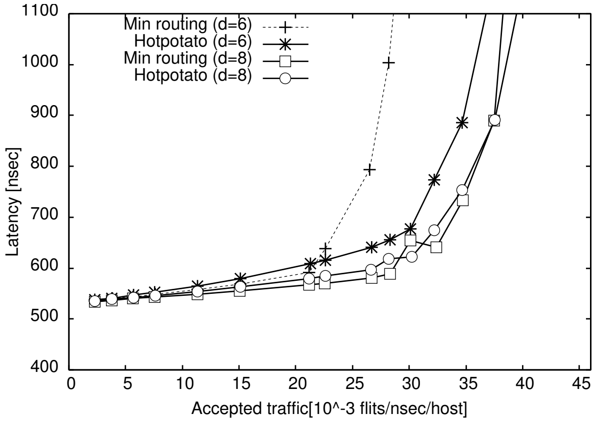

Figure 7. Latency vs. accepted traffic for uni-directional MDDP network topologies with minimal routing and hot-potato routing (Bit Reversal).

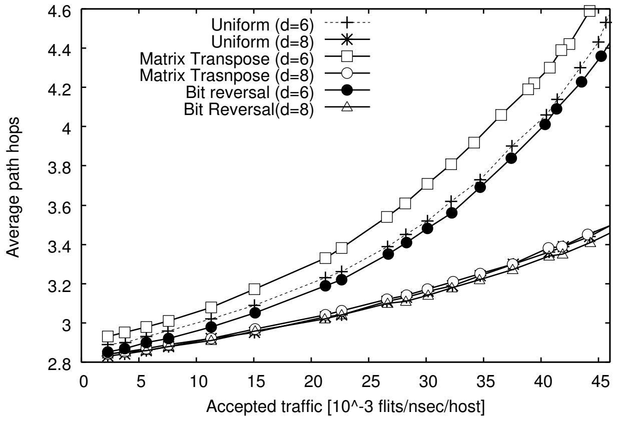

Figure 8. Average path length vs. accepted traffic for hot-potato routing on 256-switch uni-directional MDDP topologies.

33 flits (one of which is for the header). We pick relatively small packet sizes since we wish to study the performance of latency-sensitive traffic that consists of small messages [1]. Our results quantify two metrics: *latency* and *throughput*. The latency is the elapsed time (in nsec) between the generation of a packet at a source host and its delivery at a destination host. The throughput is the largest amount of traffic (in flit/nsec/host) accepted by the network before network saturation is reached.

Figures 5, 6 and 7 show latency vs. accepted traffic for the three traffic patterns for 256-switch MDDP topologies. (Cycle-accurate simulation is CPU- and RAM-intensive and we are able to simulate only up to 256 switches.) As expected, we find that accepted traffic increases with the degree. This is because the higher the degree the lower the hop counts (for instance, compare the $d = 6$ to the $d = 8$ curves). But the main observation from these results is that hot-potato routing provides comparable and often better throughput than minimal routing, leading in many cases to higher accepted traffic. This is perhaps counter-intuitive as one would expect for hot-potato routing to increase path hop counts at high load due the increased chance for packets to travel along paths that are not minimal. Recall that with minimal routing, instead, path length does not depend on load. To better explain this phenomenon, Figure 8 shows average path hop count vs. accepted traffic for hot-potato routing on the 256-switch topology. As expected, path length increases with traffic load, but this increase is not rapid. For instance, when the degree is $d = 6$, resp. $d = 8$, we find that a 10x increase in accepted traffic leads to less than a 1.65x, resp. 1.25x, increase in average path hop counts. As a result, hop-potato routing lowers congestion and increases path length only marginally, explaining that it compares well with minimal routing. Results not included here lead to similar conclusions for Imase and Random topologies. Overall, we conclude that hot-potato routing is a good choice for uni-directional topologies.

C. Parallel Application Performance Evaluation

While graph metrics (hop count and bisection) and network metrics (latency and throughput) are important for evaluating and comparing network topologies, the ultimate metric is application performance. We cannot tractably run cycle-accurate simulations of real parallel applications on even small-scale platforms due to the involved CPU and RAM demands. In this section, instead, we use discrete-event simulation to evaluate the performance of parallel application benchmarks on uni-

Table 1. ASPL (in hops) and bisection (in number of edges) for three parallel application performance case studies on uni-directional topologies (n: number of switches; d: switch degree).

	Case 1 (n=64, d=6)		Case 2 (n=256, d=6)		Case 3 (n=256, d=8)	
	ASPL	Bis.	ASPL	Bis.	ASPL	Bis.
Rand	2.39	112	3.18	467	2.80	657
MDDP	2.37	120	2.85	368	2.80	678
Imase	2.30	109	2.95	349	2.69	472

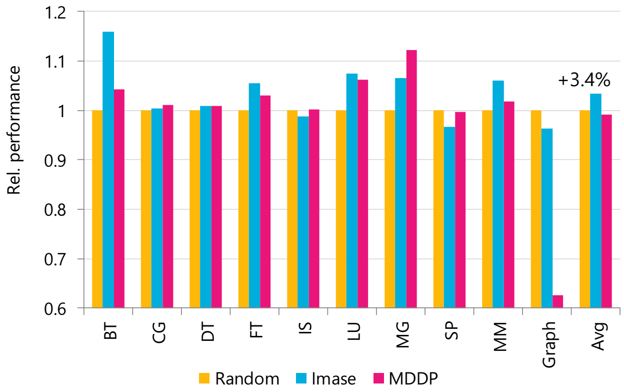

Figure 9. Case study 1 (uni-directional, 64 switches, d=6) application performance, relative to the Random topology.

directional interconnection networks. To this end we use the SIMGRID simulation framework (v3.12) [20] to simulate the execution of unmodified parallel applications that use the Message Passing Interface (MPI) [21]. Note that SIMGRID implements non-cycle-accurate network models, and thus does not capture "microscopic" network protocol effects such as flow-control behavior. Instead, its models capture the "macroscopic" behavior of network protocols. Nevertheless, model validation against cycle-accurate models has shown high accuracy for network traffic regimes that correspond to typical parallel applications running on HPC interconnects [22], [23]. The key advantage here is that SIMGRID's models are highly scalable and can thus be used for end-to-end simulation of large applications [21].

We consider three case studies, as described in Table 1. SIMGRID only supports static routing and thus cannot simulate dynamic routing algorithms such as hot-potato routing. As a result, we use minimal routing, computing shortest uni-directional paths using Dijkstra's algorithm (all topologies are connected). However, recall that the results in the previous section show that the performance of hot-potato routing is comparable to (or better than) that of minimal routing. We configure SIMGRID so that each switch has a 100 nsec delay, switches and hosts are interconnected together via links with 40 Gbps bandwidth, and each host computes at 100 GFlops. We also configure SIMGRID to utilize its built-in version of the MVAPICH2 implementation of MPI collective communications [24].

We simulate the execution of the MPI NAS Parallel Benchmarks version 4.3.1 [25] (Class B for BT, CG, DT, LU, MG and SP, and Class A for FT and IS benchmarks), the matrix multiplication example provided in the SIMGRID distribution (MM), and the Graph500 benchmark version 2.1.4 [26]. Figures 9-11 show performance results for the Random, Imase, and MDDP topologies, normalized to the performance achieved by the Random topology for each of the case studies in Table 1. For a few applications, the three topologies lead to similar performance, but in other cases (e.g., BT, Graph500) the choice of the network topology can have a large impact on application performance. The results are highly dependent on the scale of the network, on the degree of the topology, on the topology structure, and on the application's traffic pattern. The last group of bars in Figures 9-11 shows the average relative performance over all applications. On average, our three considered topologies lead to relatively similar performance. The Imase topologies leads to the best average performance for the first and third case study, and the MDDP topology leads to the best average performance (but very close to that of the Imase topology) for the second case study. Based on these results, one would thus conclude that the Imase topology is the best choice among the three classes of uni-directional topologies considered in this work.

It is beyond the scope of this work to perform detailed analyses of the performance behavior of these topologies for

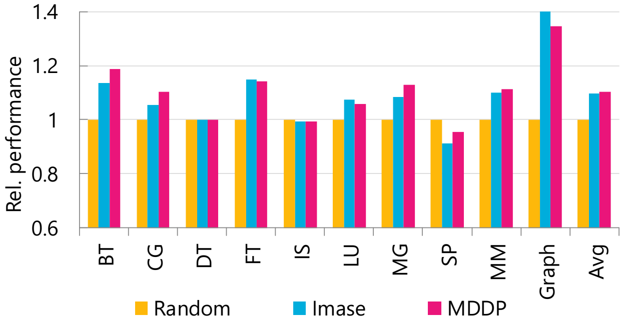

Figure 10. Case study 2 (uni-directional, 256 switches, d=6) results, relative to the Random topology.

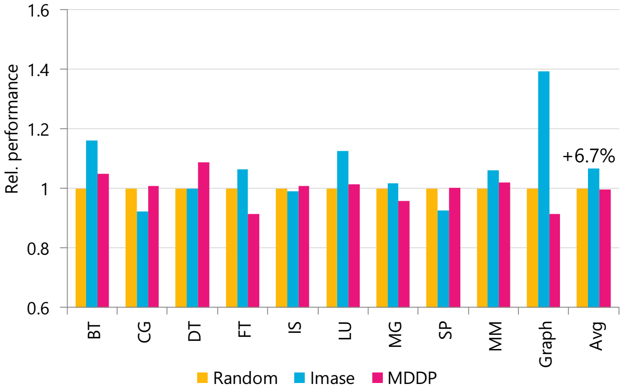

Figure 11. Case study 3 (uni-directional, 256 switches, d=8) results, relative to the Random topology.

each application, so as to determine in which regimes which topology class is best for which degree and scale. Instead, our main purpose is to compare uni-directional topologies to bi-directional topologies, thus making a case for the former although they are not currently being used in current systems. We perform this comparison in the next section.

IV. Uni- vs. Bi-directional Interconnection Networks

In this section we draw direct comparisons between bi- and uni-directional topologies, in terms of distance to the Moore bound, end-to-end parallel application performance, topology layout and cabling. We also discuss the issue of soft error handling on network links.

A. Distance to the Moore bound

In the uni-directional case, the best known DDP solutions are mostly Kautz graphs. Kautz graphs achieve a high percentage of the Moore bound, e.g., 98% for d = 8 and k = 4 with 4,608 vertices, 99% for d = 12 and k = 4 with 22,464 vertices, and 97% for d = 6 and k = 5 for 3,750 vertices (recall that d is the degree and k is the diameter). By contrast, best known solutions to the bi-directional version of the DDP typically have relatively small numbers of vertices, and are thus far from the Moore bound [27], [28]. For example, most best known solutions for $d > 7$ and $k > 7$ achieve less than 10% of the bound. One exception, in the case of high-radix networks with high degree, is presented in [3] for $k = 2$ with bi-directional graphs close to the Moore bound (e.g., 90%). This graphs form the basis for the SlimFly topology [13]. This exception notwithstanding, in general uni-directional graphs are much closer to theoretical bounds than their bi-directional counterparts, and more so as graph size increases. This observation provided the initial motivation for this work, as explained in Section I.

B. Application performance

As in Section III-C we use SimGrid to quantify end-to-end performance of parallel application benchmarks. We use the same simulation setting as that described in that section.

Our main purpose of this work it to draw direct comparisons between uni- and bi-directional topologies. The results in the previous section point to the Imase topology as the best among the uni-directional topology classes we have considered. Unfortunately, there is no clear “bi-directional equivalent” of the Imase topology that would allow a fair comparison. This is because Imase graphs were initially designed as uni-directional graphs. By contrast, the MDDP approach can be used to construct both uni- or bi-directional graphs [18]. This is because the heuristics it uses to add/remove vertices are essentially identical in the bi- and uni-directional case. The same is true for the Random topology, but results in the previous section show that it achieves poorer results than the MDDP topology. Given these considerations, in this section we compare uni- and bi-directional MDDP topologies, which is a fair comparison. We also present results that compare uni-directional Imase topologies to bi-directional MDDP topologies. This is more of an “apples and oranges” comparison, but we included it because the Imase topology leads to the best results in the previous section. Note that we do not include results for traditional bi-directional Torus topologies as results in [18] show it to be (expectedly) inferior to the bi-directional MDDP topology for most application benchmarks.

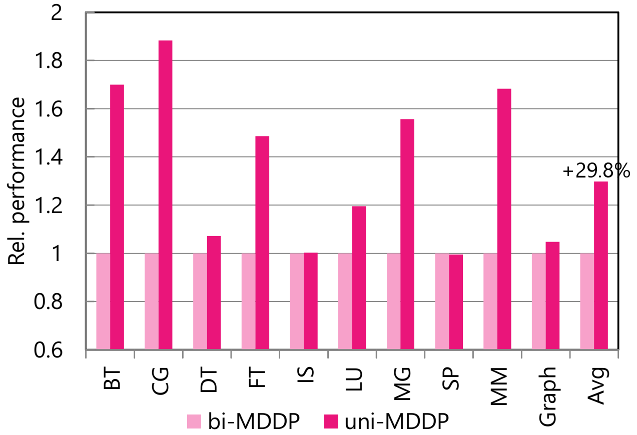

Figure 12. Relative performance of uni-directional and bi-directional MDDP topologies for each application benchmarks, and average over all benchmarks. ($n = 256$ switches, degree $d = 6$.)

Figure 12 shows performance results for each benchmark, relative to the performance achieved by the bi-directional topology, for uni- and bi-directional MDDP topologies for n=256 switches and degree $d = 6$. The main observation is that the uni-directional topology leads to equivalent or better (by up to 88% for the CG benchmark) performance than the bi-directional topology. Note that many of our benchmark applications send and receive data between pairs of processes. Since uni-directional topologies usually take different paths in each direction between two vertices, an interesting question was whether these topologies can be effective for such communication patterns. These results show that the answer to this question is the affirmative. The relative performance averaged over all benchmarks is given in the rightmost set of bars, showing that on average the uni-directional topology leads to almost 30% performance gain over the bi-directional topology.

Figure 13 shows similar results, but for the uni-directional Imase topology and the bi-directional MDDP topology, and for degree $d = 8$. Although for two of the benchmarks (DT, IS), the uni-directional topology leads to slightly lower performance than the bi-directional topology, as for the results in Figure 12 we find that the uni-directional topology typically outperforms the bi-directional topology (up to 60% for the FT benchmark). The average performance improvement (rightmost set of bars) is just above 15%.

We conclude that uni-directional topologies can lead to significant end-to-end application performance gains over their bi-directional counterparts.

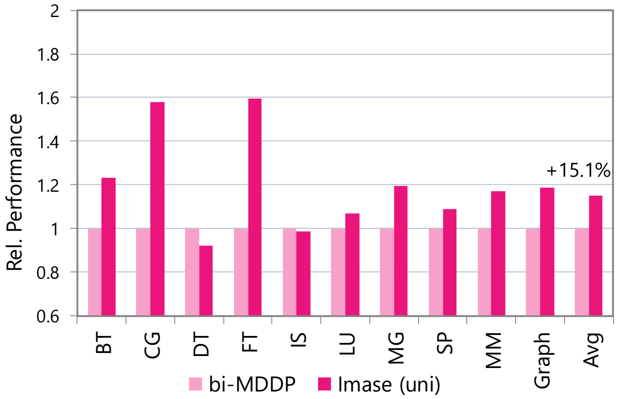

Figure 13. Relative performance of uni-directional Imase and bi-directional MDDP topologies for each application benchmarks, and average over all benchmarks. ($n = 256$ switches, degree $d = 8$.)

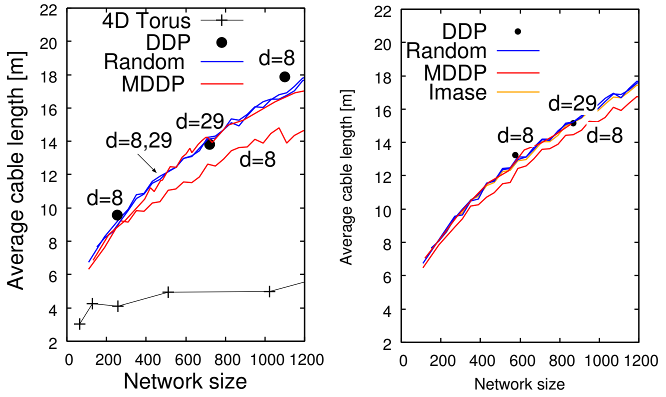

Figure 14. Average cable length vs. network size for 4-D Torus, Random, and MDDP bi-directional topologies (left) and Random, MDDP, and Imase uni-directional topologies (right) for degree $d = 8$ and $d = 29$. Best known DDP solutions are also shown for particular network sizes and degrees.

C. Layout and Cabling

Practical concerns for (large) network topologies include cabling complexity and cable length. The 1st-generation Earth Simulator supercomputer implemented a full-crossbar network using about 100,000 "fat" electric cables whose aggregate length reached 2,400km. Even though this is a well-known example of a successful production system that had high cabling complexity and high aggregate cable length, in general low complexity and length are desirable. Two other and related important practical concerns are cost and power consumption. In this section we compare bi- and uni-directional topologies qualitatively and quantitatively in terms of the above concerns. In all that follows we assume a typical deployment of the topology across cabinets in a machine room.

1) Cabling complexity: The wiring of uni-directional topologies would likely be more complex than that of bi-directional topologies since bi-directional topologies bundle two opposite links between a same source-destination pair to a single cable. Note that the inter-cabinet cabling medium is optical, meaning that cables are "thin", while the intra-cabinet cabling medium may be electric, meaning that cables are "fat" as their bandwidth becomes large. Thinner optical cables can enable higher cabling density and can relax cable bending constraints. This could mitigate the higher complexity of deploying uni-directional network topologies.

2) Cable length: We assume a physical floorplan sufficiently large to align all cabinets on a 2-D grid. Formally, assuming m cabinets, the number of cabinet rows is $q = \lceil \sqrt{m}\ \rceil$ and the number of cabinets per row is $p = \lceil m/q \rceil$. We assume that each cabinet is 0.6m wide and 2.1m deep including space for the aisle. The distance between the cabinets is computed using the Manhattan distance. We estimate cable length using the method in [13].

Figure 14 shows the average cable length of inter-switch cables for bi-directional topologies (4D Torus, Random, MDDP) and uni-directional topologies (Random, MDDP, Imase) for degree d=8 and d=29. The 4-D Torus bi-directional topology is included as a baseline reference. The figures also shows best known DDP solutions. Note that the best-known DDP solution for $d = 29$ in the bi-directional case is the exact MMS graph used by the SlimFly topology [13], and that the DDP solutions in the uni-directional case are all Kautz graphs. As expected, the average cable length increases with the network size. Considering bi-directional topologies (left-hand side of Figure 14), an expected observation is that the Torus topology has cable length several factors lower than that of all the other topologies. For $d = 29$, at a given network size, all topologies, including best known DDP solutions, have roughly the same average cable length, regardless of whether topologies are uni-directional or bi-directional. For $d = 8$, at a given network size, we find that the MDDP topology leads to lower average cable length than the Random topology and the best known DDP solutions. In the case of uni-directional topologies (right-hand side of Figure 14), we find that all topologies lead to very similar cable length, with again a small advantage to the MDDP topology in the $d = 8$ case.

The key purpose of examining the results in Figure 14 is to compare the bi-directional case to the uni-directional case. (Note that it is difficult to compare best known DDP solutions in both cases because they are for different network sizes). The broad observation is that the Random and MDDP uni-directional topologies lead to similar average cable length as their bi-directional counterparts. The only small variation is for the MDDP topology for $d = 8$, in which case the average cable length is higher in the uni-directional case. For instance, for network size 1,190, the average cable length of the uni-directional MDDP topology is 15% higher than that of the bi-directional MDDP topology.

We conclude that although cable length is an important concern for choosing a topology, this concern seems mostly orthogonal to the uni- or bi-directionality of the topology.

3) Network cost: The left-hand side of Figure 15 shows network cost vs. network size for bi-directional 4D Torus, Random, and MDDP topologies for degree $d = 8$ and $d = 29$. Best known DDP solutions are also included. As expected, the network cost is strongly impacted by switch degree. Instead,

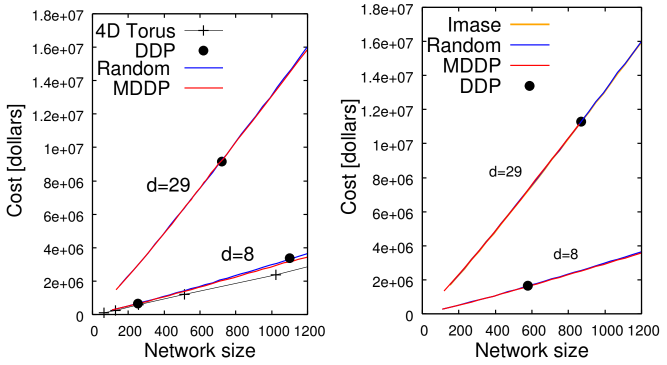

Figure 15. Average network cost vs. network size for 4-D Torus, Random, and MDDP bi-directional topologies (left) and Random, MDDP, and Imase uni-directional topologies (right) for degree $d = 8$ and $d = 29$. Best known DDP solutions are also shown for particular network sizes and degrees.

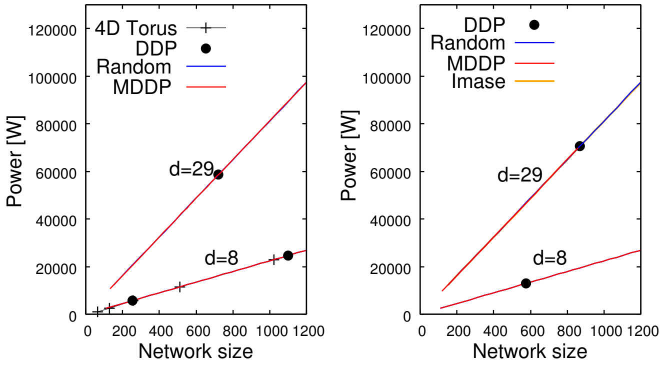

Figure 16. Average network power consumption vs. network size for 4-D Torus, Random, and MDDP bi-directional topologies (left) and Random, MDDP, and Imase uni-directional topologies (right) for degree $d = 8$ and $d = 29$. Best known DDP solutions are also shown for particular network sizes and degrees.

the cable length, upon which depends the medium used (electric or optical), is not a dominant cost factor. Although these cost estimates are obtained using the models in [13], we find that similar trends are obtained when using the alternate cost model described in [29].

The right-hand side of Figure 15 shows similar results for uni-directional Random, MDDP and Imase topologies. Best known DDP solutions are also included. Due to the lack of an available cost model for these topologies, we simply assume that the cost for one bi-directional link is the same that that for two uni-directional link. The results show little variations between topologies. Importantly, network costs are very similar to those seen in the left-hand side of Figure 14. Even though our assumption regarding the cost of uni-directional links may be optimistic, these results show that the cost of uni-directional networks is likely within acceptable bounds when compared to the cost of current bi-directional networks.

4) Network power consumption: The left-hand side of Figure 16 shows network power consumption vs. network size for bi-directional 4D Torus, Random, and MDDP topologies for degree $d = 8$ and $d = 29$, using the power model in [13]. Best known DDP solutions are also included. The right-hand side of Figure 14 shows results using the same model but for uni-directional Random, MDDP and Imase topologies. Overall, for a given degree, all considered topologies have similar power consumption, regardless of uni- or bi-directionality. Note that the popular Dragonfly topology would have a similar curve to that of the MDDP and 4-D Torus, for the same degrees and network sizes.

Given the quantitative results in this section so far, we conclude that uni-directional topologies would have cable length, cost and power consumption on par with that of of bi-directional topologies with similar scale and degree.

D. Soft Error Tolerance

A practical concern with uni-directional links is the tolerance of soft errors (i.e., bit flips). Existing bi-directional interconnection networks use link-level error detection, such as CRC (Cyclic Redundancy Check). If a bit flip is detected, then a request is sent back to the previous hop so that data can be re-transmitted. This approach is thus not feasible with purely uni-directional links, and instead costly end-to-end approaches would be needed.

We argue that this concern will soon disappear because raw bit error rate increases as link bandwidth increases. Consequently, in upcoming high-bandwidth interconnects the above error detection/correction mechanism would lead to large numbers of re-transmissions, which would then preclude low latency [30]. And indeed, emerging standards, such as the 200/400GbE standard, include the use of FEC (Forward Error Correction) at every port. Thus, it is expected that not only error detection, but also error correction, will be performed at the destination switch. This will remove the need for re-transmission requests, and thus remove the above concern with uni-directional links.

V. Conclusions

Designing low-latency network topologies of switches for upcoming large-scale cluster is a crucial but challenging objective. Given current and foreseeable switch delays, a way to achieve this objective is to design topologies with low hop counts. In this work we have proposed a radical departure from current cluster interconnect designs: uni-directional network topologies. Although uni-directional networks have been considered in the past, they have been abandoned, at least in the context of cluster interconnects; to the best of our knowledge, all production clusters today use bi-directional topologies. However, uni-directional graphs can be constructed that are much closer to theoretical hop count bounds than bi-directional graphs. Therefore, uni-directional topologies have the potential to provide a good solution to the problem of designing low hop count topologies.

In this work we have considered several classes of uni-directional graphs that can be used for creating uni-directional network topologies (Imase, Random, MDDP). Each class has some advantages and drawbacks in terms of network metrics (diameter, ASPL, bisection) and design (deterministic or random components). Discrete-event simulation results also show that these topologies lead to different, but often significant, end-to-end application performance behaviors across a range of parallel application benchmarks. Regardless of the particular underlying graph, a practical challenge is that of routing and, more specifically, congestion control. This is because, in purely uni-directional topologies, congestion control cannot be achieved via any kind of "backward" communication along a link. One of our key contributions is that we demonstrate that this challenge can be addressed via deflection, a.k.a., hot-potato, routing. Cycle-accurate simulation results show that hot-potato routing leads to similar, and in fact often better, network throughput than traditional minimal path routing.

We have drawn direct qualitative and quantitative comparisons between uni-directional topologies and their bi-directional counterparts (when available). With regards to end-to-end application performance, our discrete-event simulation results show that, when considering ten standard parallel application benchmarks, a uni-directional MDDP topology leads to almost 30% average performance improvement over its bi-directional counterpart. For particular application benchmarks, the performance improvement is above 80%, and we do not observe any performance loss for any of our benchmarks. We have also compared the uni-directional Imase topology to the bi-directional MDDP topology, and also found significant performance improvements for the uni-directional topology (15% on average, up to 60% for a single benchmark). We have then argued that uni-directional topologies are on part with, or within reasonable bounds, bi-directional topologies in terms of cabling (complexity, average cable length), cost and power consumption. Finally, the issue of soft-error tolerance, while problematic for uni-directional topologies if using current error-detection-retransmission approaches, will be resolved once FEC error correction is available in upcoming switches. In conclusion, while it is true that our proposed network design may not feasible using the off-the-shelf network equipment available today, we claim that the results in this work provide a compelling motivation for considering uni-directional interconnection networks for upcoming large-scale cluster platforms.

Acknowledgments

This work was partially supported by KAKEN 15K00144 and 16KK0009.

References

[1] K. Scott Hemmert and Jeffrey S. Vetter and Keren Bergman and Chita Das and Azita Emami and Curtis Janssen and Dhabaleswar K. Panda and Craig Stunkel and Keith Underwood and Sudhakar Yalamanchili, "Report on Institute for Advanced Architectures and Algorithms, Interconnection Networks Workshop 2008."

[2] M. Miller and J. Siran, "Moore graphs and beyond: A survey of the degree/diameter problem," *Electronic Journal of Combinatorics*, 2013, Dynamic Surveys.

[3] Combinatorics Wiki, "The Degree Diameter Problem for General Graphs," http://combinatoricswiki.org/wiki/The_Degree_Diameter_Problem_for_General_Graphs.

[4] "METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering." [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/metis/overview

[5] J.Duato, S.Yalamanchili, and L.Ni, *Interconnection Networks: an engineering approach*. Morgan Kaufmann, 2002.

[6] Top 500 Supercomputer Sites, `http://www.top500.org/`.

[7] B. Bollobás and F. R. K. Chung, "The Diameter of a Cycle Plus a Random Matching," *SIAM J. Discrete Math.*, vol. 1, no. 3, pp. 328–333, 1988.

[8] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[9] A. Singla, C.-Y. Hong, L. Popa, and P. B. h. Godfrey, "Jellyfish: Networking Data Centers Randomly," in *NSDI*, 2012, pp. 225–238.

[10] M. Koibuchi, H. Matsutani, H. Amano, D. F. Hsu, and H. Casanova, "A Case for Random Shortcut Topologies for HPC Interconnects," in *ISCA*, 2012, pp. 177–188.

[11] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-Driven, Highly-Scalable Dragonfly Topology," in *ISCA*, 2008, pp. 77–88.

[12] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova, "Skywalk: a Topology for HPC Networks with Low-delay Switches," in *IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, May 2014, pp. 263–272.

[13] M. Besta and T. Hoefler, "Slim Fly: A Cost Effective Low-diameter Network Topology," in *Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 348–359.

[14] B. McKay, M. Miller, and J. Širán, "A note on large graphs of diameter two and given maximum degree," *Journal of Combinatorial Theory*, vol. 74, no. 1, pp. 110–118, 1998.

[15] J. Flich, T. Skeie, A. Mejia, O. Lysne, P. Lopez, A. Robles, J. Duato, M. Koibuchi, T. Rokicki, and J. C. Sancho, "A Survey and Evaluation of Topology Agnostic Deterministic Routing Algorithms," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 405–425, 2012.

[16] M. Imase and M. Itoh, "Design to Minimize Diameter on Building-Block Network," *IEEE Trans. Computers*, vol. 30, no. 6, pp. 439–442, 1981.

[17] ——, "A Design for Directed Graphs with Minimum Diameter," *IEEE Trans. Computers*, vol. 32, no. 8, pp. 782–784, 1983.

[18] M. Koibuchi, I. Fujiwara, F. Chaix, and H. Casanova, "Towards ideal hop counts in interconnection networks with arbitrary size," in *Fourth International Symposium on Computing and Networking, CANDAR*, Nov. 2016, pp. 188–194.

[19] W. D. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.

[20] SimGrid: Versatile Simulation of Distributed Systems, http://simgrid.gforge.inria.fr/.

[21] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899–2917, 2014.

[22] P. Bedaride, A. Degomme, S. Genaud, A. Legrand, G. Markomanolis, M. Quinson, M. Stillwell, F. Suter, and B. Videau, "Toward Better Simulation of MPI Applications on Ethernet/TCP Networks," in *Proc. of the 4th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems*, 2013.

[23] P. Velho, L. Mello Schnorr, H. Casanova, and A. Legrand, "On the Validity of Flow-level TCP Network Models for Grid and Cloud Simulations," *ACM Trans. Model. Comput. Simul.*, vol. 23, no. 4, pp. 1–26, 2013.

[24] "MVAPICH: MPI over InfiniBand, 10GigE/iWARP and RoCE." [Online]. Available: http://mvapich.cse.ohio-state.edu/

[25] The NAS Parallel Benchmarks, http://www.nas.nasa.gov/Software/NPB/.

[26] Top 500 Sites, http://www.graph500.org/.

[27] F. Comellas, "The (Degree,Diameter) Problem for Graphs," http://maite71.upc.es/grup_de_grafs/grafs/taula_delta_d.html/.

[28] G. Exoo, "Large Regular Graphs of Given Degree and Diameter," http://isu.indstate.edu/ge/DD/index.html.

[29] J. Mudigonda, P. Yalagandula, and J. Mogul, "Taming the flying cable monster: A topology design and optimization framework for data-center networks," *USENIX ATC*, pp. 1–14, 2011. [Online]. Available: http://static.usenix.org/events/atc11/tech/final_files/Mudigonda.pdf

[30] D. Fujiki, K. Ishii, I. Fujiwara, H. Matsutani, H. Amano, H. Casanova, and M. Koibuchi, "High-Bandwidth Low-Latency Approximate Interconnection Networks," in *HPCA*, 2017, pp. 469–480.