SpectralFly: Ramanujan Graphs as Flexible and Efficient Interconnection Networks

Stephen Young*§, Sinan Aksoy*§, Jesun Firoz*§, Roberto Gioiosa*, Tobias Hagge*, Mark Kempton†, Juan Escobedo*, Mark Raugas*
*Pacific Northwest National Laboratory, †Brigham Young University
*{{first name}.{last name}}@pnnl.gov, †mkempton@mathematics.byu.edu

***Abstract*—In recent years, graph theoretic considerations have become increasingly important in the design of HPC interconnection topologies. One approach is to seek optimal or near-optimal families of graphs with respect to a particular graph theoretic property, such as diameter. In this work, we consider topologies which optimize the spectral gap. We study a novel HPC topology, SpectralFly, designed around the Ramanujan graph construction of Lubotzky, Phillips, and Sarnak (LPS). We show combinatorial properties, such as diameter, bisection bandwidth, average path length, and resilience to link failure, of SpectralFly topologies are better than, or comparable to, similarly constrained DragonFly, SlimFly, and BundleFly topologies. Additionally, we simulate the performance of SpectralFly on a representative sample of micro-benchmarks using the Structure Simulation Toolkit Macroscale Element Library simulator and study cost-minimizing layouts, demonstrating considerable benefit of the SpectralFly topology.**

***Index Terms*—Graph theory, spectral gap, spectral expansion, interconnection networks, Ramanujan graphs**

I. Introduction

In recent years, a deluge of different interconnection networks have been proposed to address the critical role of communication in modern HPC systems [1]–[7]. To efficiently and robustly enable communication, many of these topologies are designed to exhibit a plethora of beneficial structural properties. An ideal network will have low endpoint-to-endpoint latency, resiliency to link failures, high bisection bandwidth to avoid congestion – all while maintaining low-system cost. Researchers have employed the language of graph theory to formalize and quantify such properties. In this way, a number of graph statistics – such as diameter, average distance, vertex and edge-connectivity, and dense bipartitions – are well-known to be critically important to the performance of the computing system. However, constructing a graph topology simultaneously optimizing these criteria is challenging.

One approach is to focus on finding families of graphs that are extremal with respect to a particular property, with the hope that optimization of that property guarantees acceptably good, if not optimal, behavior with respect to the others. For example, the SlimFly topology [1] was proposed with the claim "it's *ALL* about the diameter." Specifically, the authors argued that graphs which minimize the diameter while simultaneously maximizing the number of vertices for a given radix also exhibit other good properties, such as resilience to link failure and high bisection bandwidth. However, how to construct such topologies (and whether they even exist) remains a challenging and ongoing topic in mathematics [8]. Indeed, the SlimFly topology is only made possible by a sophisticated algebraic construction by McKay, Miller and Širáň [9], which has nearly-optimal size with respect to degree-diameter condition. SlimFly is far from the only proposed topology to take an extremal approach; the well-known DragonFly [4] and associated variants aim to maximize performance while minimizing system cost, utilizing a group of high-radix routers as a virtual router to increase the network's effective radix. And more recently, a related topology called BundleFly [2] expands and adapts the SlimFly for use with multicore fiber systems.

In this work, we propose that utilizing a graph construction which optimizes the *spectral gap* – the difference between the largest two eigenvalues of the adjacency matrix – yields a broad family of flexible, balanced, low-cost, and congestion-avoiding topologies. We call this family of topologies SpectralFly, as they are examples of Ramanujan graphs which have optimal spectral gap. As we explain, the spectral gap is a highly nuanced and far-reaching indicator of graph structure, controlling diameter, average distance, fault tolerance, neighborhood expansion, and bisection bandwidth, among others. In comparison to SlimFly, we show SpectralFly makes marginal sacrifices in terms of diameter and average shortest path length, while offering comparable or sometimes significantly better properties, particularly in the case of bisection bandwidth and related properties involving bottlenecks. While no single topology can be optimal in every regard, our results show SpectralFly is extremely competitive for many key structural factors, making it a good fit for a variety of workloads.

Our work is organized as follows: in Section II, we provide a brief overview of graph eigenvalues, the spectral gap, and the Ramanujan property, establishing the importance of the spectral gap as a consideration in HPC interconnection network design. In Section III, we introduce the particular family of Ramanujan graphs we utilize, known as LPS graphs, providing the necessary definitions and examples, and highlighting some key LPS graph properties. Then, in Section IV, we study structural properties of SpectralFly [1] in comparison with SlimFly, BundleFly and DragonFly, across 5 classes of topology sizes which range from roughly 100 vertices

§Young, Aksoy, and Firoz contributed equally to this work.

[1]Forthcoming code for SpectralFly will, pending institutional approval, be posted at https://github.com/pnnl/SpectralFly.

1530-2075/22/$31.00 ©2022 IEEE
DOI 10.1109/IPDPS53621.2022.00105

to almost 10K vertices. Additionally, we also examine the resilience of these properties under varying levels of edge failures. Finally, in Sections V-VI, we validate the utility of the structural advantages of SpectralFly by performing simulations and experiments using the Structural Simulation Toolkit Macroscale Element Library (SST/macro). We define our routing algorithms in Section V, and evaluate several micro-benchmarks with different topologies in Section VI.

Throughout, we use standard graph theory terminology and consider only undirected graphs $G = (V, E)$, where V is a set of elements called *vertices* and E is a set of unordered pairs of elements of V called *edges*. In the context of interconnection networks, vertices represent routers, and edges represent bidirectional links. The degree of a vertex is the number of edges to which it belongs; we call a graph k-regular if each vertex has degree k and sometimes refer to k as the radix of the graph.

II. Eigenvalues, expansion, and the Ramanujan property

Graph eigenvalues capture a plethora of network properties critical to the design and function of interconnection networks. Diameter, bisection bandwidth, fault tolerance, average shortest path length, and other structural properties are controlled by eigenvalues; see [10], [11] for a survey. Here, we highlight graph theoretic results establishing these connections in order to explain why Ramanujan graphs possess superior structural properties for interconnection network design.

Many such properties are controlled by a *single* eigenvalue: if G is a k-regular graph with adjacency matrix A, this eigenvalue, denoted $\lambda(G)$, is the largest magnitude eigenvalue of A not equal to $\pm k$. The difference between the two largest adjacency eigenvalues is sometimes called the "spectral gap". In case of k-regular graphs, this is the difference between the second largest eigenvalue and k, as k is always the largest eigenvalue. As we will soon explain, Ramanujan graphs "optimize" the $\lambda(G)$ and hence have large spectral gap.

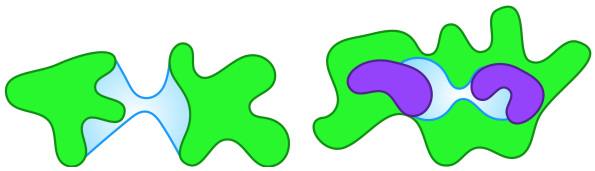
Perhaps the most important property for our purposes here, $\lambda(G)$ controls the expansion properties of the graph. Loosely speaking, expansion means every "not too large" set of vertices has a "not too small" set of neighbors. The vertex isoperimetric number of a graph, $h(G)$, is one way of formalizing this:

$$h(G) = \min_{\substack{X \subseteq V(G) \\ 2|X| \leq |V(G)|}} \frac{|\partial X|}{|X|},$$

where ∂X denotes the neighbors of X that are not in X. Thus, larger values of $h(G)$ suggest better expansion properties.

The vertex isopermetric number, as well as related variants, are closely linked to $\lambda(G)$. In particular, Tanner [12] proved a lower bound on $h(G)$ for k-regular graphs in terms of this eigenvalue, while Alon and Milman [13] gave an upper bound. Such results suggest it is natural to measure expansion directly in terms of eigenvalues themselves. We will concern ourselves with this notion of expansion, called *spectral expansion*.

As smaller values of $\lambda(G)$ mean better expansion properties, it is natural to ask: what is the theoretical minimum of $\lambda(G)$? The Alon-Boppana theorem [14] answers this question, stating that for a k-regular graph with second largest (in magnitude) adjacency eigenvalue λ and diameter D, we have $\lambda(G) \geq 2\sqrt{k-1}\,(1 - 2/D) - 2/D$. Consequently, if $(G_i)_{i=1}^{\infty}$ is a family of connected, k-regular, n-vertex graphs with $n \to \infty$ as $i \to \infty$, then, $\liminf_{i \to \infty} \lambda(G_i) \geq 2\sqrt{k-1}$. We call a graph Ramanujan if it achieves this theoretical minimum, i.e., is an optimal spectral expander.



(a) Low Bisection Bandwidth (b) Low Discrepancy

Fig. 1: Structures forbidden by high bisection bandwidth and discrepancy. Bisection bandwidth only concerns edges crossing a bipartition (blue shadow), while discrepancy also requires any two subsets (in purple), are bottleneck-free.

Definition 1. A k-regular graph G is called Ramanujan if $\lambda(G) \leq 2\sqrt{k-1}$, where $\lambda(G)$ denotes the largest magnitude adjacency eigenvalue of G not equal to $\pm k$.

As a consequence of their optimal spectral expansion, Ramanujan graphs possess a plethora of beneficial structural properties discussed in [10]. In particular, the Ramanujan property guarantees at least nearly optimal bisection bandwidth.

While we emphasize this near-optimality of bisection bandwidth in this work, we note the Ramanujan property guarantees something much stronger: it bounds the number of edges between *any* collection of vertices, not just bisections. This stronger property is called the discrepancy inequality [15]. Simply put, this means large spectral gap implies two arbitrary subsets of the network are bottleneck-free; see Fig. 1 for a cartoon of substructures forbidden by the discrepancy property.

While we will not explicitly design the experiments of Section VI to emphasize the impact of the discrepancy property, in practice, this is likely to be an important property for the practical usage of the systems. In particular, as the discrepancy property assures that given an arbitrary collection of vertices involved in a computation the bisection bandwidth on the topology restricted to those vertices is still high, we expect systems designed around Ramanujan graph topologies will be less susceptible to performance degradation based on job schedule and inter-job contention as illustrated in [16]. Additionally, we note that the discrepancy property will likely mitigate the benefit of routing strategies such as Valiant that attempt to homogenize traffic across the network. In particular, as high discrepancy networks are optimally bottleneck-free, this minimizes the advantage of homogenizing network traffic.

a) Related work in HPC: As evidenced by the relationships between eigenvalues and other structural properties highlighted above, it is unsurprising some HPC topologies consider spectral expansion *implicitly* in their network design. The well-known randomized Jellyfish topology has strong, albeit not optimal, spectral expansion properties. However,

random k-regular graphs are "sub Ramanujan" as shown by Friedman's proof [17] of Alon's second eigenvalue conjecture [14]; hence SpectralFly has superior spectral expansion over JellyFish. Further, as discussed in [7], the unstructuredness of randomized constructions such as Jellyfish makes "them hard to reason about (predict, diagnose), build (e.g., in terms of wiring complexity), and so poses serious, arguably insurmountable, obstacles to their adoption in practice."

Next, we briefly mention other work *explicitly* considering notions of spectral expansion or Ramanujan graphs in an HPC setting. In [10], the authors survey a wide swath of supercomputing topologies and derive either asymptotic bounds or exact expressions for their spectral gap, which shows many supercomputing topologies are far from Ramanujan. In this work, we aim to realize the theoretical potential suggested in [10] through SpectralFly, which has optimal spectral gap. So called $(\alpha, \beta, n, d)-$expanders are utilized to construct "multi-butterfly" networks [18], and later, "metabutterfly" networks [19], which aim to mitigate wiring complexity. Valadarsky et al [20] propose "Xpander", a general construction in the context of datacenter design. Xpander is based on the theory of graph lifts [21] which, via derandomization procedures, may generate deterministic almost-Ramanujan graphs. Theoretical work by Marcus, Spielman and Srivastava [22] suggests it may be possible to explicitly generate Ramanujan graphs utilizing k-lifts via sophisticated interlacing polynomial techniques.

III. SpectralFly Topology Construction

Constructing explicit families of Ramanujan graphs is an ongoing topic of research. The first constructions were by Lubotzky, Phillips and Sarnak [23], and independently, by Marguilis [24]. In 2013 and 2015, Marcus, Spielman and Srivastava [22], [25] gave new constructions of bipartite Ramanujan graphs. For more on these constructions, see [10].

Here, we focus on the construction by Lubotzky, Phillips and Sarnak, which we refer to as *LPS graphs*. These are the graph topologies underlying a SpectralFly network. Hence, when studying graph-theoretic properties, we use the terms "SpectralFly" and "LPS" interchangeably. When interpreted as a network, a vertex of an LPS graph corresponds to a router, and edges between vertices correspond to bidirectional links. While fully-realized SpectralFly networks must also specify endpoint concentration (see Section VI), in this section we focus on the core LPS topology formed by the routers.

We utilize an extension of the original LPS graphs provided by Morgenstern [26]. LPS graphs are examples of graphs encoding algebraic group structure, called Cayley graphs.

Definition 2 (Cayley Graph)**.** The Cayley graph, $\mathrm{Cay}(\mathcal{G}, S)$, of a group $\mathcal{G}$ and symmetric is a graph on vertex set $V = \mathcal{G}$ and edge set $E = \{\{u, v\} : u^{-1}v \in S\}$.

An LPS graph is a particular Cayley graph where both the group and the generating set S depend on number-theoretic properties of two input values, p and q, as defined below.

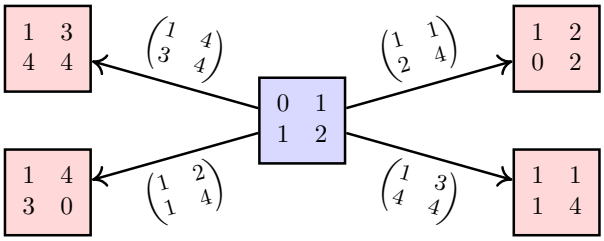


Fig. 2: Neighborhood of a vertex in LPS$(3, 5)$. Vertices are from $\mathrm{PGL}(2, \mathbb{F}_5)$ labeled by a representative matrix from the coset; edges $\{u, v\}$ are labeled by a generating element $u^{-1}v$.

Definition 3 (LPS Graphs)**.** The LPS graph LPS(p, q) is a Cayley graph defined for distinct odd primes p, q. To define the generator set, let x, y be solutions to $x^2 + y^2 + 1 = 0 \pmod q$, and D be the set of solutions $(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ to $\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = p$ which satisfy

- $\alpha_0 > 0$ is odd, if $p \equiv 1 \pmod 4$
- $\alpha_0 > 0$ is even, or $\alpha_0 = 0$ and $\alpha_1 > 0$, if $p \equiv 3 \pmod 4$.

The generating set S of LPS(p, q) consists of all matrices

$$\begin{bmatrix} \alpha_0 + \alpha_1 x + \alpha_3 y & -\alpha_1 y + \alpha_2 + \alpha_3 x \\ -\alpha_1 y - \alpha_2 + \alpha_3 x & \alpha_0 - \alpha_1 x - \alpha_3 y \end{bmatrix},$$

where $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) \in D$. The group G of LPS(p, q) is

$$G = \begin{cases} \mathrm{PSL}(2, \mathbb{F}_q) & \text{if } \left(\frac{p}{q}\right) = 1 \\ \mathrm{PGL}(2, \mathbb{F}_q) & \text{if } \left(\frac{p}{q}\right) = -1 \end{cases},$$

where $\left(\frac{p}{q}\right)$ is the Legendre symbol, and PSL and PGL are the projective special and linear groups, respectively. If $q > 2\sqrt{p}$, then LPS(p, q) is a $(p + 1)$-regular Ramanujan graph.

For those unfamiliar with algebraic graph theory or number theory, a full understanding of the details within the proceeding definition is unnecessary for the purposes of this work (see [27]). Nonetheless, we include this definition as a self-contained description of the LPS topology. Similarly for completeness and to help garner understanding, we briefly illustrate how to generate LPS graphs in practice with an example below, and include a visualization in Figure 3. For a full tutorial on LPS graph generation, see [28].

Example 1. Let $(p, q) = (3, 5)$. These are valid inputs for an LPS graph because p, q are distinct, odd primes and $5 > 2\sqrt{3}$. Since $x^2 \not\equiv 3 \pmod 5$ for any x, the Legendre symbol $\left(\frac{3}{5}\right) = -1$ and hence the group is $\mathrm{PGL}(2, \mathbb{F}_5)$ where $\mathbb{F}_5 = \{0, 1, \ldots, 4\}$. The elements of $\mathrm{PGL}(2, \mathbb{F}_5)$ are cosets of 2×2 matrices with elements in $\mathbb{F}_5$ and nonzero determinant such that A, B are in the same coset if $A = xB$ for some nonzero x. For example, the element

$$v = \left\{ \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 2 & 4 \end{bmatrix}, \begin{bmatrix} 0 & 3 \\ 3 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 4 \\ 4 & 3 \end{bmatrix} \right\}$$

represents a *single* element of $\mathrm{PGL}(2, \mathbb{F}_5)$, and hence a single vertex of the graph LPS$(3, 5)$.

Next, we construct the generating set S. As $p \equiv 3 \pmod 4$, we are interested in solutions of $\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 3$ where either $\alpha_0 > 0$ and is even, or $\alpha_0 = 0$ and $\alpha_1 > 0$. There are

no solutions of the former type; solutions of latter type are:

$$(0,1,1,1),(0,1,-1,-1),(0,1,-1,1),(0,1,1,-1).$$

Finally, using $(x,y)=(0,2)$ as a solution to $x^2+y^2+1=0 \pmod 5$, the elements of the generating set S may be constructed. For example, the coset for the generator $s \in S$ corresponding to the solution $(0,1,1,1)$ is

$$\left\{\begin{bmatrix}1 & 2\\ 1 & 4\end{bmatrix},\begin{bmatrix}2 & 4\\ 2 & 3\end{bmatrix},\begin{bmatrix}3 & 1\\ 3 & 2\end{bmatrix},\begin{bmatrix}4 & 3\\ 4 & 1\end{bmatrix}\right\}.$$

$\mathrm{LPS}(3,5)$ is then constructed by creating edges between $u, v \in \mathrm{PGL}(2,\mathbb{F}_5)$ whenever $us = v$ for $s \in S$. Figure 2 illustrates the neighborhood of a certain vertex in $\mathrm{LPS}(3,5)$, labeling edges by the associated element $s \in S$.

There are several reasons for selecting LPS graphs amongst currently proposed Ramanujan constructions. First, LPS graphs are flexible in terms of feasible sizes and radix values. Figure 4 (left) plots radix and vertex counts for all possible LPS graphs generated with inputs $p, q < 300$. While, like almost any structured family of topologies, some radix and vertex count combinations are infeasible, the absence of large gaps in the plot suggests the high likelihood of finding an LPS graph "acceptably close" to any given desired radix and vertex count combination. As discussed further in Section IV, this flexibility stands in contrast to many competing graph topologies. LPS graphs afford users the ability to generate *arbitrarily large* graphs for a given radix, whereas the sizes of other topologies can only be increased via the radix.

Secondly, in addition to exhibiting the Ramanujan property, LPS graphs possess other desirable characteristics. For example, since LPS graphs are Cayley graphs, they are vertex-transitive. Informally, this means every vertex has an identical local environment, i.e. the graph "looks the same" from every vertex. Thus, the 6 hop neighborhood of a vertex in $\mathrm{LPS}(3,17)$ seen in Figure 3 has an identical structure for all vertices. Consequently, vertex-transitivity enables simplifications which benefit the computational cost and design of routing protocols. Their algebraic structure also affords other benefits, such as optimal edge-connectivity (a key consideration for network resiliency) as well as efficient algorithms by which topologies on tens of millions of vertices may be easily generated [28].

In addition to possessing these properties by virtue of being a Cayley graph, LPS graphs are also widely studied. Over the past several decades, researchers have bounded or characterized the diameter, path length behavior, and fault tolerance of LPS graph, making them attractive options for supercomputing topologies, as argued in [10]. One such key property for interconnection networks is bisection bandwidth. Figure 4 (upper right) presents the normalized bisection bandwidth of LPS graphs for various sized topologies on radix values between $k = 4$ and 98, divided by $nk/2$ to ensure a size-agnostic comparison. We observe larger normalized bisection bandwidth values are achieved for larger radix graphs, with diminishing returns. In contrast to some other topologies we survey, the bisection bandwidth doen't decay as LPS graph size increases per fixed radix, which is a consequence of the Ramanujan property. Furthermore, larger normalized bisection bandwidth values are feasible for larger radix networks.

Topology	Routers	Router Radix	Diam.	Dist.	Girth	μ_1
$\mathrm{LPS}(11,7)$	168	12	3	2.39	3	0.50
$\mathrm{SF}(7)$	98	11	2	1.89	3	0.62
$\mathrm{BF}(13,3)$	234	11	3	2.56	3	0.27
$\mathrm{DF}(12)$	156	12	3	2.70	3	0.08
$\mathrm{LPS}(23,11)$	660	24	3	2.35	3	0.65
$\mathrm{SF}(17)$	578	25	2	1.96	3	0.64
$\mathrm{BF}(37,3)$	666	23	3	2.61	3	0.13
$\mathrm{DF}(24)$	600	24	3	2.84	3	0.04
$\mathrm{LPS}(53,17)$	2448	54	3	2.32	3	0.74
$\mathrm{SF}(37)$	2738	55	2	1.98	3	0.65
$\mathrm{BF}(97,4)$	3104	54	3	2.76	3	0.07
$\mathrm{DF}(53)$	2862	53	3	2.93	3	0.02
$\mathrm{LPS}(71,17)$	4896	72	4	2.61	4	0.77
$\mathrm{SF}(47)$	4418	71	2	1.98	3	0.66
$\mathrm{BF}(137,4)$	4384	74	3	2.76	3	0.05
$\mathrm{DF}(69)$	4830	69	3	2.94	3	0.01
$\mathrm{LPS}(89,19)$	6840	90	4	2.61	4	0.80
$\mathrm{SF}(59)$	6962	89	2	1.99	3	0.66
$\mathrm{BF}(157,5)$	7850	85	3	2.82	3	0.06
$\mathrm{DF}(85)$	7310	85	3	2.95	3	0.01

TABLE I: Basic structural properties

IV. Structural Property Comparison

In order to understand the trade-offs between costs, diameter, and bisection bandwidth we compare the combinatorial properties of four topologies representing extreme points at or near the design space Pareto frontier. Specifically, we consider the DragonFly (optimizing cost and diameter), SlimFly (optimizing diameter and size), BundleFly (optimizing diameter and cost), and LPS/SpectralFly (optimizing spectral gap).

Since random graph constructions, such as the aforementioned JellyFish, have sub-optimal spectral gap [17], and also face serious challenges to adoption in practice due to their unstructuredness, we limit our comparison to *deterministic* topologies. Furthermore, we've selected topologies capable of being scaled to beyond tens of thousands of vertices, and which are flexible enough to generate instances with similar size, radix and link counts to other topologies, in order to ensure a fair comparison. Satisfying these criteria, the topologies we consider are defined as follows:

- $\mathrm{LPS}(p,q)$: The topology underlying SpectralFly, LPS graphs [23] are described in Definition 3. The radix is $p+1$ and the number of vertices is $\left(3-\left(\frac{p}{q}\right)\right)\left(q^3-q/4\right)$.
- SlimFly, $\mathrm{SF}(q)$: Studied in [1], SlimFly topologies are based on the MMS graph construction by McKay, Miller and Širáň [9]. For a description of the MMS graph construction, see [29]. The number of vertices is $2q^2$ and radix is $\frac{3q-\delta}{2}$, where $q = 4k+\delta$ for $\delta \in \{-1,0,1\}$.
- BundleFly, $\mathrm{BF}(p,s)$: a multi-star product of an MMS graph with parameter s and Paley graph with parameter p – see [2]. The number of vertices is $2ps^2$ and the radix is $\frac{p-1}{2}+\frac{3s-\delta}{2}$ where $s = 4k+\delta$ for $\delta \in \{-1,0,1\}$.
- DragonFly, $\mathrm{DF}(a)$: while there are many DragonFly variants (see [10], [30] for specifications), we consider the "canonical" DragonFly topology consisting of $a+1$ fully connected groups, each on a vertices. The number of vertices is $a(a+1)$ and the radix is a.

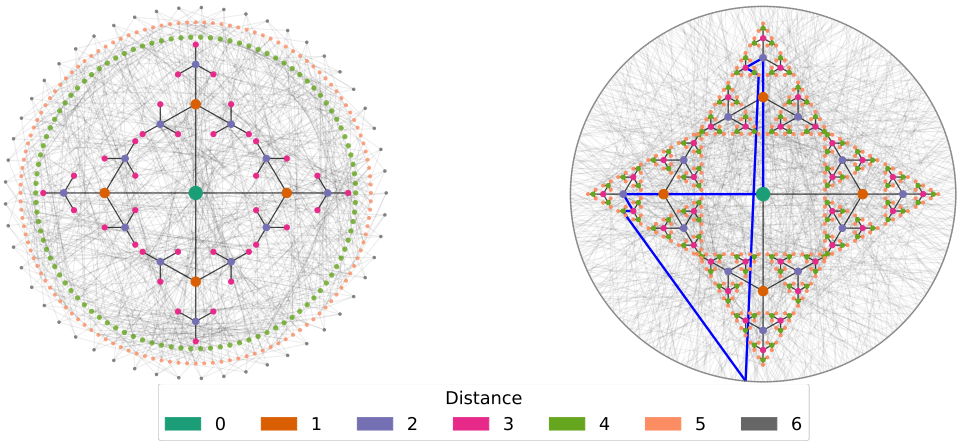


Fig. 3: Visualization of SpectralFly topology instances: the entire LPS$(3,7)$ graph (left) and the 6-hop neighborhood of a vertex in LPS$(3,17)$ (right). Since LPS graphs are vertex transitive, the k-hop neighborhood of every vertex has the same structure. Furthermore, the local neighborhood surrounding a vertex is a tree of variable depth depending on the inputs p, q. For instance, a shortest length cycle in LPS$(3,17)$ is highlighted as blue, and utilizes vertices at distance 6 from the center vertex.

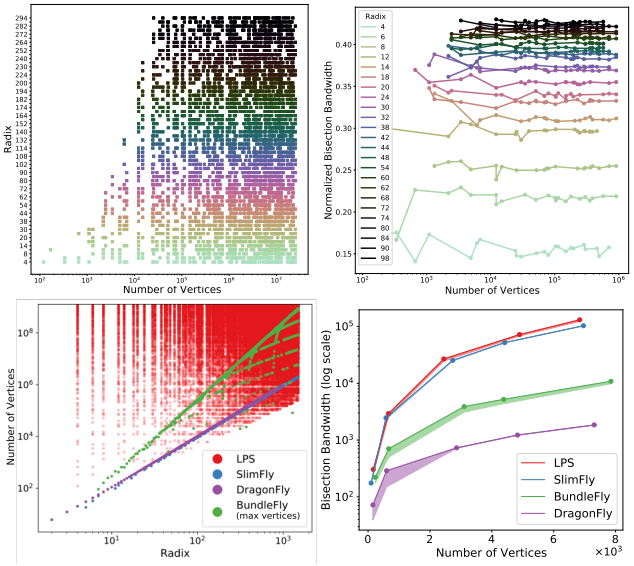


Fig. 4: Possible number of vertices and radix of LPS for $p, q < 300$ (upper left), normalized bisection bandwidth of LPS for $p, q < 100$ (upper right), feasible topology sizes per radix (lower left), raw bisection bandwidth comparison (lower right).

While it would be interesting to explore properties of Xpander topologies here, applying complicated interlacing polynomial approaches for their construction and the need to calculate the set of all shortest paths for every pair of routers makes such an evaluation impractical at scales of interest.

We consider 5 size classes for each topology, ranging from 100 vertices to 7K vertices. For each size class, we conduct a parameter search to select the topology with closest radix and number of vertices relative to the others in that class. Table I shows the 4 topologies within each size class have very close radix, and fairly close node counts ensuring a fair comparison of the performance based on the *structural properties* of these networks in our subsequent experiments. The topologies also have similar *girth* (length of the shortest cycle), with larger LPS topologies being the sole examples of girth 4 topologies.

a) Feasible topology sizes per radix: The LPS construction accommodates a variety of radix and node size combinations. In general, Ramanujan graphs of any size are possible; however the smallest possible LPS graph is on 120 vertices. Fig. 4 (lower left) plots possible vertex count and radix combinations. For SlimFly and canonical DragonFly, a large, low-radix topology is impossible, as the radix uniquely determines the topology size. BundleFly allows multiple possible vertex sizes per radix, but the choice of radix constrains the possible vertex sizes. The green points plot the maximum possible number of vertices per each feasible BundleFly radix. Some of maxima drop off sharply for certain radix values, suggesting the range of possible sizes may be unstable.

b) Diameter and average path lengths: As summarized in Table I. SlimFly always has diameter 2, while BundleFly and DragonFly have diameter 3. In contrast, the diameter of LPS graphs depends on the topology size; numerical experiments from [31] suggest this diameter is asymptotic to $(4/3)\log_5(n)$. LPS has the second smallest average shortest path length (i.e. distance) across all size classes, in spite of sometimes having the largest diameter (for the fourth and fifth size classes). This gap between diameter and average distance suggests "most" pairs of vertices in LPS graphs may be closer in distance than the diameter. This is also apparent in Figure 3's visualization of LPS$(3,7)$, where relatively fewer vertices appear at distance equal to the diameter from the center vertex. Indeed, recent work by Sardari [31] proved for any k-regular Ramanujan graph, only a tiny fraction of all pairs of vertices have distance greater than $(1+\varepsilon)\log_{k-1}(n)$. Furthermore, for each vertex x, the number of vertices at distance greater than this exponentially decays, being less than $n^{1-\varepsilon}$.

c) Normalized Laplacian spectral gap, μ_1: To enable cross-size comparison, we compute the normalized Laplacian spectral gap, μ_1, related to the second largest adjacency eigenvalue λ by $\mu_1 = {}^{k-\lambda}/_{k}$, where k is the radix. Whereas smaller values of λ ensure better spectral expansion, this is associated

with *larger* values of μ_1. Compared with SlimFly and LPS, Table I shows BundleFly and DragonFly with smaller values of μ_1, which decay for larger sized topologies. As proven in [10], the second normalized Laplacian eigenvalue of the SlimFly topology SF(q) is $\frac{2}{3+\delta/q} \sim \frac{2}{3}$. Since LPS graphs are Ramanujan, they have μ_1 at least as large as $\frac{k-2\sqrt{k-1}}{k}$. Thus an LPS graph with radix $k \geq 35$ is guaranteed to have larger μ_1 than *any* SlimFly topology. LPS graphs with smaller radix values may still have larger μ_1 (as seen in the second size class in Table I) or smaller μ_1 (as seen in the first size class).

d) Bisection bandwidth: We use METIS [32] to approximate bisection bandwidth, establishing an upper bound given by the points in Fig. 4 (lower right). We also compute a lower bound from [33], $\text{BW}(G) \geq \frac{\lambda_1 kn}{4}$, where k is the radix and n is the number of vertices. The exact bisection bandwidth lies between these points, represented by the shaded regions. Recall we are considering the router topology, without regard to a specific concentration. While one can further analyze bisection bandwidth under a particular concentration level, the relative orderings we observe here also hold whenever chosen concentration levels are equal, and so we omit this design choice for clarity and simplicity. As seen in Fig. 4 in log-scale, as the size of SlimFly topologies increase, the gap between its normalized bisection bandwidth and that of a similar radix LPS widens further. SpectralFly has up to a 39% increase in bisection bandwidth over SlimFly. This can be confirmed analytically: applying bounds from [10], the normalized bisection bandwidth of SlimFly is asymptotically $1/3$. LPS graphs have normalized bisection bandwidth at least $\frac{k-2\sqrt{k-1}}{2k}$, guaranteeing an LPS graph with $k \geq 36$ has larger normalized bandwidth than *any* SlimFly. We emphasize this is a *lower bound*; the normalized bisection bandwidth of LPS graphs computed by METIS exceeds $1/3$ around radix 18.

A. Structural Properties Under Link Failures

We also examine how these structural properties vary under link failures of varying magnitudes.For each topology, we delete k proportion of its edges, chosen randomly. Our results are averaged over sufficiently many trials.[2] We run these experiments for "small", 600 vertices, instances of each topology, as well as intermediate sized topologies on 5K vertices. Figure 5 presents the results for diameter, mean distance, and bisection bandwidth. Note these measures are only well-defined for *connected* topologies; however, all four topologies remain consistently connected for small (left column) and medium (right column) sizes under random link failures until 60% and 80%, respectively, of edges are deleted. Thus, we only consider edge deletion proportions up until this disconnection threshold. With regard to diameter, SlimFly has the smallest value of 2 of the topologies surveyed. However, at 10% edge failure, this diameter increases to 4, while LPS topologies exhibit slightly smaller diameter. This suggests SlimFly diameter is more *fragile* than that of LPS, congruent with our prior observation

[2]For each topology, proportion k, and structural property measured, we increase the number of trials x in powers of 10 until the coefficient of variation of sample means across 10 batches of x trials is less than 10%.

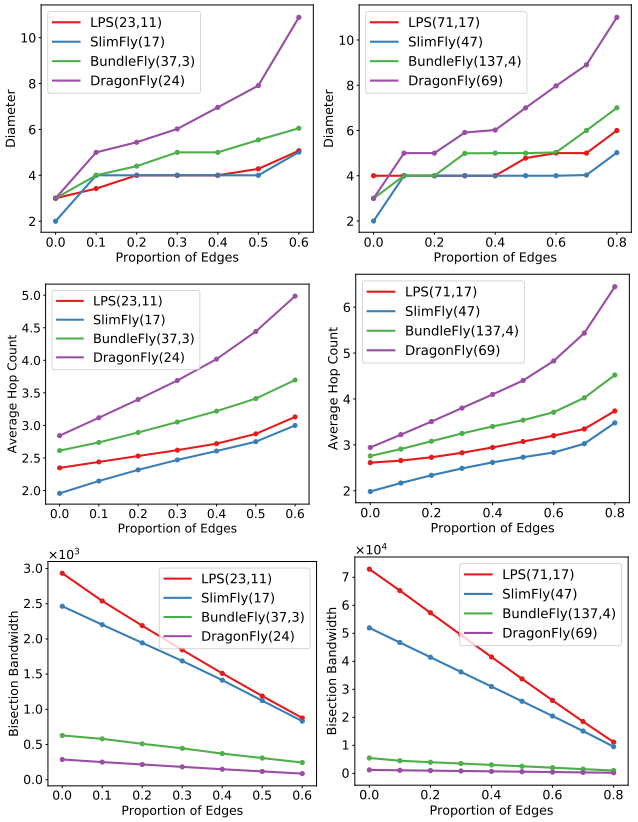


Fig. 5: Structural properties under edge failures for comparable LPS, SlimFly, BundleFly and DragonFly topologies on about 600 vertices (left column) and 7K vertices (right)

that while nearly every pair of vertices in SlimFly is separated by a 2 hop distance, only very few pairs of vertices in an LPS graph achieve the diameter [31]. While LPS maintains a slight edge over SlimFly for 10% edge failures, for 20-50% edge failures they have comparable diameter, and for $> 50\%$ SlimFly has slightly smaller diameter.

Lastly, for mean distance and bisection bandwidth, LPS and SlimFly perform the best. SlimFly has the smallest mean distance across all edge failure rates, with the gap between LPS narrowing slightly as a higher proportion of edges fail. For bisection bandwidth, LPS retains its larger bandwidth over SlimFly; this gap narrows significantly beyond 20% failure.

In summary, LPS and SlimFly are consistently more resilient under random edge failures than BundleFly and DragonFly with regard to diameter, average distance, and bisection bandwidth. For diameter, LPS and SlimFly are comparable, with LPS having slightly better diameter for 10% edge failures and worse for 50% and above. For average distance and bisection bandwidth, SlimFly retains lower hop count while LPS retains superior bisection bandwidth.

V. Routing Algorithms

We consider 3 types of routing strategies for SpectralFly: shortest path routing (minimal), Valiant routing, and Universal Global Adaptive (UGAL) routing. In minimal routing, given a source-destination pair (s, d), a packet is forwarded along the

routers on the shortest path from s to d. In theory, minimal routing will minimize the overall latency of communication thereby outperforming other routing schemes when the underlying network has no congestion. However, in a congested network, shortest paths may not be the best choice for routing. This is especially true when the betweenness centrality scores of a set of vertices (routers) in a graph (topology) are quite high, meaning these set of vertices will be on the shortest paths for many vertices in the graph. Consequently these vertices will become the bottlenecks in a highly-saturated network.

Congestion concerns have prompted alternative routing schemes which improve performance on various topologies. One such alternative to shortest path routing is Valiant routing [34] which proceeds in two phases: given a source-destination pair (s, d), a random intermediate router i is chosen. The packet is then routed from s to i along a shortest path. Once the packet arrives at i, the second phase forwards the packet from i to d by following a shortest path.

However, Valiant routing ignores the current state of routers, such as queue length. To ameliorate this, the UGAL family of routing protocols selects dynamically between the minimal path or a Valiant-style paths based on the current state of the system. For example, in the UGAL-L variant, each router only maintains information about the queue lengths of the local outports. Using this information at the source, a packet is either forwarded to a random intermediate node first or follows a minimal path based on the queue sizes of the local random outport and minimal outport and total hopcounts from the source to the destination for these two possible routes.

A. Deadlock Avoidance

Due to limited resources on each router (buffer count, size, etc.), cyclic dependencies can arise in the resource dependency graph, where messages may try to flow from one router to the next but also messages from the next router may try to flow in the reverse direction. As the buffers fill up and traffic from each router blocks each other in a cycle, this ultimately results in a deadlock. Such deadlocks can be avoided primarily in three ways: (1) by creating an acyclic routing scheme; (2) by using virtual channels (VC) and changing the virtual channel to route a packet on each network hop (by incrementing the virtual channel on each network hop, deadlock-free routing can be guaranteed); (3) by running a cycle-detection algorithm on the routing graph beforehand. Each time a cycle is detected, a new virtual channel is added to one of the routing edges, continuing until there are no more cycles. We've chosen the second approach to avoid deadlock based on virtual channels, since it does not require any preprocessing of the topology graph. We set the number of virtual channels to be equal to the diameter of SpectralFly, $d+1$ for the shortest path routing and $(2d + 1)$ for Valiant routing.

VI. Simulation Results

In this section, we report our simulation results on evaluating SpectralFly, SlimFly, BundleFly and DragonFly topologies with different workloads exhibiting interesting communication patterns that are prevalent in many HPC applications.

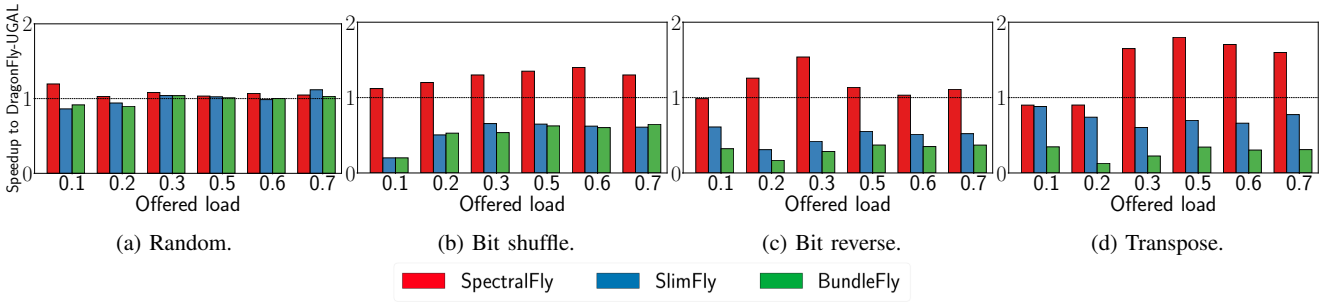
A. Simulation Software

We conduct our experiments in the Structural Simulation Toolkit (SST) Macroscale Element Library (SST/macro) simulator [35]. Our simulation approach performs online simulation, which involves skeletonization of an application during the compilation step so that part of the application involving communication (such as communication API calls, MPI_alltoall etc.) can be intercepted by the simulator during runtime. The simulator replaces these calls with various built-in network component model implementations. The application can then run inside the simulator without any significant change. The user can provide necessary hardware parameter values (for routers, NICS, topologies, routing schemes etc.) to the simulator for running the application with different hardware configurations. We have used the *Simulator Network for Adaptive Priority Packet Routing (SNAPPR)* network model in SST/macro to evaluate different topologies. SNAPPR implements coarse-grained cycle-based simulation to simulate priority queue-based QoS. In addition, it can also restrict injection rate of messages for congestion control. For a detailed discussion about available network models in SST/macro, we refer to the SST/macro user manual [35].

B. Configuration and Simulation Setup

We evaluate the performance of different micro-benchmarks by considering SpectralFly, DragonFly, SlimFly and BundleFly topologies. We conducted our experiments with ~$8.7k$ network endpoints and with 32-port routers. To generate the SpectralFly topology with ~$8.7k$ network endpoints, we set $(p, q) = (23, 13)$ to generate a graph with 1092 routers, and a concentration of 8 endpoints per router. For the DragonFly topology configuration, the number of groups used is 69 (g), with 16 routers per group (a), each router connected to 8 endpoints (p), and 8 global links (h) per router. This conforms to the recommended balance to support full global bandwidth for Dragonfly with radix-k switches ($p = k/4, h = k/4, a = k/2$). The global links in the DragonFly topology are arranged in a circulant manner [36], [37], since this arrangement provides better bisection bandwidth than the absolute arrangement. For the SlimFly topology, q is set to 27, with each router connected to 8 endpoints. Finally, for the BundleFly topology, the graph is constructed with $p = s = 9$, and each router has a concentration of 6 endpoints.

In the case of under-subscription (for example, when running microbenchmarks with $2^{13} = 8192$ ranks out of ~$8.7k$ available ranks), the physical nodes allocated to the job are chosen randomly. Each MPI rank is then sequentially allocated to nodes based on the standard ordering for the topology. For the SpectralFly topology we use the essentially unstructured ordering resulting from the Elzinga construction [28]. We report our experimental results with various routing strategies. Valiant routing demonstrates similar performance trend. The router buffer size has been set to 64KB (Other buffer sizes have also been tested but the results are not reported here



(a) Random. (b) Bit shuffle. (c) Bit reverse. (d) Transpose.

Fig. 6: Performance comparison across topologies, traffic patterns, and offered load conditions under UGAL-L routing.

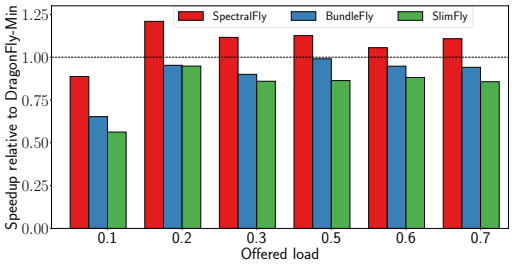


Fig. 7: Performance across topologies, and offered load conditions with random micro-benchmark, under minimal routing.

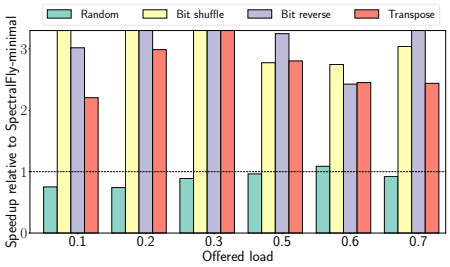


Fig. 8: Evaluation of Valiant routing for the SpectralFly topology with micro-benchmarks.

due to space constraint). For simulation, the number of virtual channels has been set to the diameter of the graph plus one.

C. Experimental results

1) Micro-benchmarks to assess performance under congestion: We consider standard traffic pattern micro-benchmarks to evaluate the performance of different topologies under various network capacities (offered load). These include random, bit shuffle, transpose, and bit reverse traffic patterns. In each case, a source node communicates with a destination node that is determined by a specific permutation of the bit representation of the source. Random traffic patterns can be found in many irregular and graph applications. The shuffle traffic pattern (obtained by rotating left 1 bit of the source) can be found in Fast Fourier Transform (FFT) and sorting applications. Matrix transpose is a basic linear-algebraic operation.

We consider a total of 8192 endpoints for these experiments. For each traffic pattern ran on a topology, we collect the maximum time taken across all the messages under a particular offered load. The results are reported in Figure 6. Here, on the x-axis we plot the offered load i.e. how much of the network is saturated when running the micro-benchmarks. To simulate network congestion, we inject messages with varying delays by simulating a Poisson process. We report the speedup relative to the execution with DragonFly Each topology was run with the UGAL-L routing. As can be observed from the figure, for all the micro-benchmarks SpectralFly performs the best. The better performance of SpectralFly can be attributed to the superior bisection bandwidth and available path diversity of the SpectralFly topology. At or beyond 70% of the network capacity, the network becomes saturated. Between BundleFly and SlimFly, BundleFly exhibits better performance (except with bit shuffle traffic). These experiments demonstrate that, because of stronger discrepancy and spectral properties, SpectralFly is robust to accommodate diverse traffic patterns under varying degrees of network congestion.

2) Evaluation of different routing schemes: Besides evaluating different topologies with the UGAL routing, we also consider minimal and Valiant routing for evaluation. Figure 7 presents the performance of the random benchmark with minimal routing. With random micro-benchmark, SpectralFly demonstrates better performance, compared to other topologies. Bit shuffle and transpose exhibit similar patterns. Next, in order to evaluate the difference between minimal and Valiant routing for SpectralFly, we ran the four micro-benchmarks under varying network loads and report the results in Figure 8. The execution time is normalized w.r.t. the execution time of minimal (shortest-path) routing scheme on SpectralFly.

We see significant improvement for all offered loads with the bit shuffle, bit reverse, and transpose traffic patterns, while the random pattern has a significant decrease in performance (except at 60% offered load). This suggests the increase in path diversity by applying Valiant routing to a structured communication pattern better exploits the discrepancy property of the LPS graphs. Moreover, there is already significant path diversity in minimal routing for the random micro-benchmark, and so the addition of Valiant routing provides a minimal increase in path diversity while doubling the expected length of the routing paths. This suggests SpectralFly performs best when traffic is unstructured, either due to the logical communication pattern or from the choice of routing algorithm.

D. Evaluation of Topologies under Real-World Traffic Patterns

1) Patterns Considered: For evaluating different topologies with real-world traffic patterns (under both minimal and UGAL routings), we consider communication motifs from the Ember Communication Pattern Library [38]:

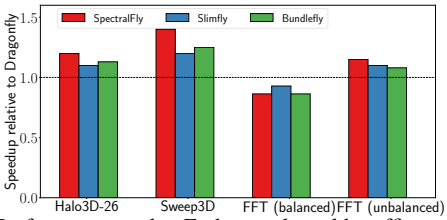


Fig. 9: Performance under Ember real-world traffic patterns with minimal routing, reported as speedup w.r.t. the DragonFly topology.

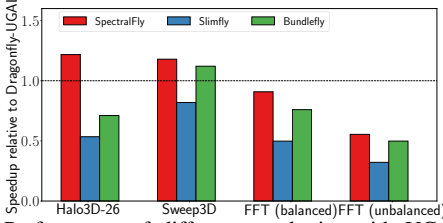


Fig. 10: Performance of different topologies with UGAL routing w.r.t. DragonFly UGAL routing for real-world traffic patterns.

(i) *Nearest neighbor communication pattern – Halo3D-26:* Nearest neighbor communication pattern, found in stencil workloads, is captured by the Halo3D-26 motif, where each MPI rank communicates with 6 of it's nearest neighbors as well as 20 of it's diagonal neighbors, a total of 26 neighbors. (ii) *Wavefront communication pattern – Sweep3D:* Wavefront communication pattern is prevalent in particle transport physics simulation, parallel iterative solvers and triangular solvers [39] that generally stresses network latency and has substantial dependency levels. One representative motif for the wavefront communication pattern is the ASCI Sweep3D [39]. Here, a 3D data domain is decomposed over a 2D array of MPI processes and repeated sweep along the diagonal is performed. (iii) *Subcommunicator all-to-all communication pattern–FFT:* In this communication pattern, found in Multi-dimensional FFT, a 3D domain is decomposed along the X and Y dimensions and subcommunicators are formed along the 1D line in both of the X and Y dimensions. One MPI process is assigned to each of the 3D grid points and communicates with all the subcommunicators along the X and Y dimensions.

2) Performance Results: The performance of each of the Ember motifs on different topologies are reported in Figure 9 (with minimal routing) and fig. 10 (with UGAL routing). As can be observed from Figure 9, for both the Halo3D-26 and the Sweep3D traffic patterns, the SpectralFly configuration outperforms the other topologies with a speedup of ~$1.2\times$ and ~$1.4\times$ respectively, over the DragonFly topology (with minimal routing). This indicates that, for SpectralFly, with communication patterns with relatively low per-node communication, the robust discrepancy property and the reduction in the average hop-count is sufficient to ameliorate any penalty accruing as a result of longer maximum hop-count. In contrast to this, we see that for the balanced FFT motif, DragonFly slightly outperforms the other topologies. As the communication pattern for FFT involves all-to-all communication along a 2D-plane within a 3D-arrangement of ranks, we suspect that relative improvement is a result of the partial alignment of these 2D all-to-all communication with the group

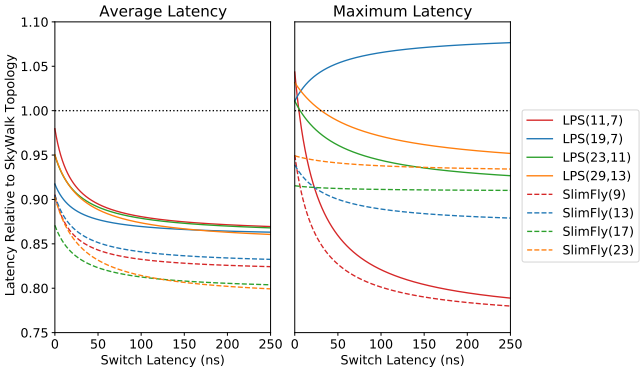


Fig. 11: Ratio of maximum and average latency between SpectralFly/SlimFly and SkyWalk as a function of the switch latency.

structure. Specifically, if multiple nodes from the same all-to-all communication land in the same group, there is an outsized decrease in the communication pressure on the global links. In particular, we note that the stronger group structure of DragonFly (even as compared to BundleFly and SlimFly) leads to the best performance on the balanced FFT motif. We also note that, because of the lack of large all-to-all clusters in Halo3D-26 and Sweep3D, there is as much marginal benefit to alignment with the group structure. Finally, for the unbalanced FFT traffic pattern, the SpectralFly configuration outperforms all other topologies. While the other topologies with strong group structure will again benefit from multiple elements from the 2D all-to-all aligning with the group, the increased sizes of the all-to-all groups will necessitate significantly more between-group traffic on global links, degrading overall performance. In contrast, SpectralFly handles the increased all-to-all communication pressure better.

We also evaluate the Ember benchmarks on different topologies with UGAL routing. Figure 10 shows SpectralFly outperforms other topologies for halo3D-26 and Sweep3D motifs. However, for both FFT motifs, DragonFly with UGAL routing performs better. For the FFT motif, SpectralFly performs better than SlimFly and BundleFly (achieving 90% of the execution efficiency w.r.t DragonFly for the balanced FFT motif). This suggests discrepancy properties of LPS graphs ensure performance of SpectralFly either better-than or competitive-with topologies which excel under the UGAL routing scheme.

VII. Beyond Structure

As noted in Section IV, network parameters for each topology (Table I), were chosen to facilitate a comparison of interconnects based on their fundamental *structural* properties. In Sections IV and VI, the structure of SpectralFly is superior to, or comparable with, that of DragonFly, SlimFly, and BundleFly across a variety of metrics. However, in practice, the trade-off between topology cost and performance is an important factor in the overall design. Since the competing topologies have a similar number of routers and connections per router, the total amount of wiring needed to build the topologies will be a primary driver of any cost differences. In addition to the direct costs of wire length there is an additional, indirect cost as longer wires often necessitate higher energy optical connections. As an added benefit, the analysis of wire

Topology	Routers	Router Radix	Average Wire Length (m)		Max. Wire Length (m)		Electrical Links	Optical Links	Bisection Bandwidth	Total Power (W)	Power/Bandwidth (mW per Gb/s)
LPS$(11,7)$	168	12	8.02	(10.29)	19.8	(21.21)	249	758	304	928	30.5
SF(9)	162	13	8.68		21.6		151	902	369	1028	27.9
LPS$(19,7)$	336	20	10.43	(13.94)	28.6	(31.05)	432	2928	1080	3276	30.3
SF(13)	338	19	10.89		27.8		315	2896	1105	3155	28.6
LPS$(23,11)$	660	24	14.35	(17.27)	39.8	(41.07)	531	7389	2928	7845	26.8
SF(17)	578	25	13.05		36.2		558	6667	2465	7138	29.0
LPS$(29,13)$	1092	30	17.32	(21.09)	50.8	(52.10)	831	15549	6150	16292	26.5
SF(23)	1058	35	16.00		47.4		1257	17258	6095	18336	30.1

TABLE II: Wire length and energy efficiency statistics for the heuristic embedding of comparable SpectralFly and SlimFly topologies. For mean and maximum wire length, we include in parenthesis the mean 20 instantiations of the SkyWalk topology in the same machine room.

lengths allows us to evaluate the end-to-end and typical latency of SpectralFly as compared to physical latency minimizing topologies, such as SkyWalk [40].

First, we compare the average and maximum wire length necessary to implement a SpectralFly topology to the similarly-sized SlimFly topology. SlimFly was chosen as point of comparison because the bisection bandwidth (see Figure 4) is most structurally comparable to the SpectralFly topology. To ensure an equitable comparison, we assume each topology is implemented equal concentration with rectilinear physical wiring. Following the methodology in [40], we assume an $x \times y$ grid of cabinets where intra-cabinent wires are all 2 meters while the inter-cabinet wires have length of $4 + 2\,|x_i - x_j| + 0.6\,|y_i - y_j|$, which includes a 2 meter overhead at each end of the link. Assuming a roughly square room, we fix $y = \lceil \sqrt{2c/0.6} \rceil$ and $x = \lceil c/y \rceil$ where c is the minimum number of cabinets need for the topology if, similar to the Summit supercomputer, each cabinet contains two routers.

This allows us to restrict our attention to the wiring between the routers. Thus the question of minimal average wire length is an instance of the Quadratic Assignment Problem (QAP), which is $\mathcal{NP}$-complete. To find a heuristically minimal layout, we apply an expectation minimization approach combined with a greedy refinement process which outperforms the the standard Fast Approximate QAP algorithm on these instances [41]. In order to take advantage of short lengths of intra-cabinent links, for each topology we fix as a maximum matching of the underlying topology and enforce that the matching edges are within a cabinet. Table II provides a summary of the results of this layout approach. As we can see the maximum and average wire lengths of SpectralFly and SlimFly topologies are within ~10% of each other across all sizes, with SpectralFly performing better on smaller topologies. To provide additional context, we compare the layout with the SkyWalk topology which was designed to minimize end-to-end latency in the case of ultra-low latency routers/switches. For each machine room, we report (in parenthesis) the average behavior over 20 instantiations of the SkyWalk topology in the same machine room. As we can see the SkyWalk topology typically requires between ~20-30% longer lines overall with a maximum wirelength ~3% longer. This indicates that despite the underlying expansion of the SpectralFly and SlimFly necessitating longer wires, with care the overall wire lengths can be made comparable to other modern topologies.

To translate the wire lengths to an estimate of the power usage, we update the methodology of [42] to modern hardware (i.e., the Mellanox SB7800 InfiniBand EDR 100Gb/s Switch) and assume each port connected to an electrical link uses ~3.76 W while ports with optical links use 25% more power at ~4.72 W. Using METIS to approximate bisection bandwidth, we quantify the trade-off between overall power expenditure versus communication performance (see Table II). As is the case with other metrics, the difference in energy cost per unit of bandwidth is ~5-10%, with the notable exception of the $(29, 13)$-SpectralFy being 15% more efficient than the similarly sized SlimFly. This efficiency gain is a consequence of SpectralFly's better expansion properties yielding slightly better bisection bandwidth while requiring ~15% fewer links.

The wire lengths allows the evaluation of end-to-end latency and clock cycle times implicit in SpectralFly and SlimFly. Following [40], we assume a cable delay of $5\ ns/m$ uniform switch latencies. Figure 11 provides a comparison of both SlimFly and SpectralFly with the latency minimizing SkyWalk topology. Except for LPS$(19, 7)$, we have that both topologies typically have lower end-to-end latency (and hence clock-cycle time) than the SkyWalk topology, as well as singificantly lower average latency. While the average and end-to-end latency of SpectralFly is slightly larger (~5-10%) than SlimFly, necssitating a longt clock-cycle time, the overall performance benefits illustrated in Section VI are sufficient to make up for this difference. Further, we believe that applying a more sophisticated multi-objective minimization approach to the layout problem will further close the gap in latencies between these two topologies.

VIII. Conclusion

The design of interconnection networks is increasingly informed by graph theoretic considerations. While researchers have established a long list of desirable criteria, such as low-diameter and average distance, high fault tolerance, and high bisection bandwidth, developing topologies exhibiting all these properties requires sophisticated methods. To this end, we've proposed SpectralFly, a class of topologies with optimal spectral gap based on the LPS graph algebraic construction.

Exploring first the design space of LPS graphs, we showed this construction permits a large range of topology sizes and radix values, including arbitrarily large topologies per fixed radix. We then highlighted, both via experiments and analytically, structural properties for which SpectralFly excelled in comparison to competing topologies. In particular, for bottleneck measures such as normalized bisection bandwidth, SpectralFly outperformed other topologies, which have decaying or tightly bounded bandwidth. The concession for these

properties is slightly larger diameter; however, we showed the average distance between nodes in an LPS topology is typically smaller than DragonFly and BundleFly, and only marginally larger than SlimFly. Furthermore, experiments suggest these LPS graph properties remain relatively robust under edge failures. Lastly, in order to experimentally validate the potential of SpectralFly suggested by its structural properties, we conducted simulations using the SST/macro simulator. SpectralFly outperformed other network topologies under a diverse range of communication patterns found in traditional HPC workloads. Further, we demonstrated that the cost of implementing a SpectralFly topology is on-par with, if not better than, the SlimFly topology (which is the only considered topology which has comparable bandwidth).

Acknowledgement. We would like to thank Jeremiah Wilke for very helpful technical exchanges regarding SST/macro. This work was supported by the High Performance Data Analytics program at PNNL. Information Release PNNL-SA-160551.

REFERENCES

[1] M. Besta and T. Hoefler, "Slim fly: A cost effective low-diameter network topology," in *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2014, pp. 348–359.

[2] F. Lei, D. Dong, X. Liao, and J. Duato, "Bundlefly: a low-diameter topology for multicore fiber," in *Proceedings of the 34th ACM International Conference on Supercomputing*, 2020, pp. 1–11.

[3] C. Hawkins, B. A. Small, D. S. Wills, and K. Bergman, "The data vortex, an all optical path multicomputer interconnection network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 3, pp. 409–420, 2007.

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

[5] A. Shpiner, Z. Haramaty, S. Eliad, V. Zdornov, B. Gafni, and E. Zahavi, "Dragonfly+: Low cost topology for scaling datacenters," in *2017 IEEE 3rd International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era (HiPINEB)*, 2017, pp. 1–8.

[6] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, 2012, pp. 225–238.

[7] A. Valadarsky, M. Dinitz, and M. Schapira, "Xpander: Unveiling the secrets of high-performance datacenters," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, 2015, pp. 1–7.

[8] M. Miller and J. Sirán, "Moore graphs and beyond: A survey of the degree/diameter problem," *The electronic journal of combinatorics*, pp. DS14–May, 2012.

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

[10] S. G. Aksoy, P. Bruillard, S. J. Young, and M. Raugas, "Ramanujan graphs and the spectral gap of supercomputing topologies," *The Journal of Supercomputing*, pp. 1–37, 2020.

[11] B. Mohar, "Eigenvalues, diameter, and mean distance in graphs," *Graphs and combinatorics*, vol. 7, no. 1, pp. 53–64, 1991.

[12] R. M. Tanner, "Explicit concentrators from generalized n-gons," *SIAM Journal on Algebraic Discrete Methods*, vol. 5, no. 3, pp. 287–293, 1984.

[13] N. Alon and V. Milman, "λ_1, isoperimetric inequalities for graphs, and superconcentrators," *Journal of Combinatorial Theory, Series B*, vol. 38, no. 1, pp. 73–88, feb 1985.

[14] N. Alon, "Eigenvalues and expanders," *Combinatorica*, vol. 6, no. 2, pp. 83–96, jun 1986.

[15] F. R. K. Chung, *Spectral graph theory*. Conference Board of the Mathematical Sciences, 1997, vol. 92.

[16] A. Bhatele, N. Jain, Y. Livnat, V. Pascucci, and P. Bremer, "Analyzing network health and congestion in dragonfly-based supercomputers," in *2016 IEEE International Parallel and Distributed Processing Symposium, IPDPS*, 2016, pp. 93–102.

[17] J. Friedman, "A proof of Alon's second eigenvalue conjecture," in *Proc. of the thirty-fifth ACM Symposium on Theory of Computing*, 2003.

[18] E. Upfal, "An o(log n) deterministic packet-routing scheme," *Journal of the ACM*, vol. 39, no. 1, pp. 55–70, jan 1992.

[19] E. A. Brewer, F. T. Chong, and T. Leighton, "Scalable expanders," in *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing - STOC '94*. ACM Press, 1994.

[20] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira, "Xpander: Towards optimal-performance datacenters," in *Proceedings of the 12th International on Conference on emerging Networking Experiments and Technologies - CoNEXT '16*. ACM Press, 2016.

[21] Y. Bilu and N. Linial, "Lifts, discrepancy and nearly optimal spectral gap," *Combinatorica*, vol. 26, no. 5, pp. 495–519, oct 2006.

[22] A. Marcus, D. A. Spielman, and N. Srivastava, "Interlacing families i: Bipartite ramanujan graphs of all degrees," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, oct 2013.

[23] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[24] G. A. Margulis, "Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators," *Problemy peredachi informatsii*, vol. 24, no. 1, pp. 51–60, 1988.

[25] A. W. Marcus, D. A. Spielman, and N. Srivastava, "Interlacing families IV: Bipartite ramanujan graphs of all sizes," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, oct 2015.

[26] M. Morgenstern, "Existence and explicit constructions of $q + 1$ regular ramanujan graphs for every prime power q," *Journal of Combinatorial Theory, Series B*, vol. 62, no. 1, pp. 44–62, sep 1994.

[27] G. Davidoff, P. Sarnak, and A. Valette, *Elementary number theory, group theory and Ramanujan graphs*. Cambridge univ. press, 2003, no. 55.

[28] R. Elzinga, "Producing the graphs of Lubotzky, Phillips and Sarnak in Matlab," 2010.

[29] P. R. Hafner, "Geometric realisation of the graphs of McKay–Miller–Širáň," *Journal of Combinatorial Theory, Series B*, vol. 90, no. 2, pp. 223–232, 2004.

[30] M. Y. Teh, J. J. Wilke, K. Bergman, and S. Rumley, "Design space exploration of the dragonfly topology," in *International Conference on High Performance Computing*. Springer, 2017, pp. 57–74.

[31] N. T. Sardari, "Diameter of ramanujan graphs and random cayley graphs," *Combinatorica*, vol. 39, no. 2, pp. 427–446, 2019.

[32] G. Karypis and V. Kumar, "Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices," 1997.

[33] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

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

[35] "Structural Simulation Toolkit (SST) Macroscale Element Library," https://github.com/sstsimulator/sst-macro, accessed: 2021-01-15.

[36] E. Hastings, D. Rincon-Cruz, M. Spehlmann, S. Meyers, A. Xu, D. P. Bunde, and V. J. Leung, "Comparing global link arrangements for dragonfly networks," in *2015 IEEE International Conference on Cluster Computing*. IEEE, 2015, pp. 361–370.

[37] F. Kaplan, O. Tuncer, V. J. Leung, S. K. Hemmert, and A. K. Coskun, "Unveiling the interplay between global link arrangements and network management algorithms on dragonfly networks," in *2017 IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Comput.(CCGRID)*, pp. 325–334.

[38] "Ember Communication Pattern Library," https://github.com/sstsimulator/ember, accessed: 2022-01-01.

[39] A. Hoisie, O. Lubeck, and H. Wasserman, "Performance analysis of wavefront algorithms on very-large scale distributed systems," in *Workshop on wide area networks and high performance computing*. Springer, 1999, pp. 171–187.

[40] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova, "Skywalk: A topology for hpc networks with low-delay switches," in *2014 IEEE 28th International Parallel and Distributed Processing Symposium*. IEEE, 2014, pp. 263–272.

[41] J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe, "Fast approximate quadratic programming for graph matching," *PLOS ONE*, vol. 10, no. 4, pp. 1–17, 04 2015.

[42] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *Proceedings of the 37th Annual International Symposium on Computer Architecture*, 2010, pp. 338–347.