PolarFly: A Cost-Effective and Flexible Low-Diameter Topology

Kartik Lahotia*, Maciej Besta†, Laura Monroe‡, Kelly Isham§, Patrick Iff†, Torsten Hoefler†, and Fabrizio Petrini*
* Intel Labs, Santa Clara, CA, 95054, USA
{kartik.lakhotia, fabrizio.petrini}@intel.com
† Scalable Parallel Computing Laboratory, ETH Zürich, 8092 Zürich, Switzerland
{maciej.besta, patrick.iff, torsten.hoefler}@inf.ethz.ch
‡ High Performance Computing Division, Los Alamos National Laboratory, Los Alamos, NM, 87545, USA
lmonroe@lanl.gov
§ Colgate University, Hamilton, NY, 13346, USA
kisham@colgate.edu

arXiv:2208.01695v3 [cs.NI] 14 Oct 2022

***Abstract*—In this paper we present PolarFly, a diameter-2 network topology based on the Erdős-Rényi family of polarity graphs from finite geometry. This is a highly scalable low-diameter topology that asymptotically reaches the Moore bound on the number of nodes for a given network degree and diameter.**

PolarFly achieves high Moore bound efficiency even for the moderate radixes commonly seen in current and near-future routers, reaching more than 96% of the theoretical peak. It also offers more feasible router degrees than the state-of-the-art solutions, greatly adding to the selection of scalable diameter-2 networks. PolarFly enjoys many other topological properties highly relevant in practice, such as a modular design and expandability that allow incremental growth in network size without rewiring the whole network. Our evaluation shows that PolarFly outperforms competitive networks in terms of scalability, cost and performance for various traffic patterns.

I. Introduction

Traditional demand for scalable networks comes from government labs and research institutions to perform large scientific simulations. For example Fukagu [1], the largest supercomputer in the world at the time of this writing, connects $158,976$ processing nodes in a single system. The immediate forerunners in the Top500 list [2], Sierra [3] and Summit [4]. use Infiniband configurations with, respectively, $4,474$ and $4,608$ processing nodes. Another notable example of large scale network is BlueGeneQ [5], with $98,304$ network endpoints. Meta recently announced the AI Research SuperCluster (RSC) [6], which is expected to be the fastest AI supercomputer in the world in later 2022 with almost $10,000$ nodes. RSC will help Meta's AI researchers build better AI models that can learn from trillions of examples, work across hundreds of different languages to analyze text, images and video together, and develop new augmented reality tools. Microsoft, Google and Amazon also have expressed strong interest in simulating large AI models with analogous network scale [7], [8].

A common requirement of academic, governmental and industrial High-Performance Computing (HPC) data centers, is increased efficiency and reduced network cost, which is typically proportional to the number of network links. For this reason low-diameter networks, and in particular diameter 2 and 3 topologies, have seen growing interest in the scientific community over the last few years [9]–[12]. Low diameter networks are also considered an essential ingredient to tackle one of the major issues in data centers, tail latency [13].

The emergence of high-radix optical IO modules is a technological amplifier of low diameter networks. These modules increase the network by squeezing many connections per unit of space or "shoreline", leading to higher system connectivity [14]–[24]. In addition, tremendous efforts are being made to bring silicon photonic connections directly into the chip, using various integration methods [14], [15], [25], commonly known as co-packaging.

Co-packaged photonics not only reduces power consumption and improves performance, but also impacts the overall network design. With co-packaged modules, each element of the chipset (compute, acceleration, memory and storage) becomes a first-class citizen with a direct low-latency interface to the network. To maximize application performance in this novel system design, the onus is on network fabric to provide high-bandwidth, low-latency communication and extreme scalability. This further increases the need for scalable low-diameter networks.

A. State of Art Diameter 2 Topologies: Slim Fly

Given the availability of high-radix routers, it is desirable to maximize the number of nodes that can be supported on a network of a given diameter. Slim Fly [26] was the first topology analyzed in the networking community that explicitly optimized its structure towards the Moore bound [27], an upper

This work was supported by Triad National Security, LLC, operator of the Los Alamos National Laboratory under Contract No.89233218CNA000001 with the U.S. Department of Energy, and by LANL's Ultrascale Systems Research Center at the New Mexico Consortium (Contract No. DE-FC02-06ER25750). The United States Government retains and the publisher, by accepting this work for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce this work, or allow others to do so for United States Government purposes. This paper has been assigned the LANL identification number LA-UR-22-23079, Version 2.

bound on the number of vertices in a graph with given diameter and for a given maximal degree. By fixing its diameter to 2, Slim Fly reduces construction cost and power consumption, while ensuring low latency and high bandwidth.

However, Slim Fly has several issues with respect to practical layout and deployment. The number of feasible configurations/topological constructions is limited, and it is not competitive with commercially available products. And there are no results in the literature to address network re-configurability and expansion. This is a matter of great importance in real-world scenarios, where data centers need to increase the size of a compute center gradually over time without being forced to rewire and re-layout the whole interconnect.

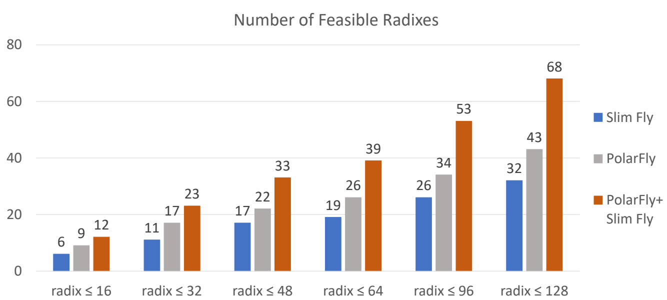


Fig. 1: Design space of feasible degrees (network radixes) for PolarFly and Slim Fly. Asymptotically, there are 50% more PolarFly feasible degrees than what Slim Fly offers.

B. Contributions

To address the above, we present PolarFly: a diameter-2 network topology that is asymptotically optimal in terms of the Moore bound.

- PolarFly comes with a *larger number of feasible designs*, in contrast to Slim Fly [26], as shown in Figure 1. Asymptotically, PolarFly offers 50% more configurations that can be constructed, without much overlap with the design space of Slim Fly. This is especially *beneficial for co-packaged systems*, where the network is integrated in the same chip with the computational engines and the network radix is fixed.
- PolarFly can be constructed from routers having radix at or near powers of 2. In particular, PolarFly supports radixes $32, 48, 62$ and 128, *making the network design more practical for HW implementations*.
- PolarFly has a surprisingly elegant construction that allows *modularity and expansion*. Specifically, the network can be decomposed into groups of nodes, all but one of which is a fan-out of triangles, to which new node groups may be added incrementally.
- PolarFly is asymptotically optimal in terms of the Moore bound, while Slim Fly asymptotically achieves only $8/9$ of the bound. Hence, PolarFly *asymptotically supports as many routers as possible* for the router degree in a diameter-2 system, and offers *reductions in construction costs* of up to 20%.
- PolarFly exhibits *high bisection bandwidth*, empirically approaching an optimal 50% of edges in the cut-set as q gets large. PolarFly also shows *high resilience upon link breakage*, with the diameter experimentally staying at 4 even after 55% link breakage.

Our evaluation shows that PolarFly is performance- and cost-competitive with other network topologies, including Slim Fly, Dragonfly, and Fat Tree. Thanks to its compact layout, PolarFly achieves very high saturation under random traffic with low latency, for both minimal and non-minimal adaptive routing. Moreover, under adversarial permutation patterns, PolarFly saturates between 50% and 66%, outperforming Slim Fly and Dragonfly, and approaching the non-blocking Fat Tree.

II. Background

A. Network Model

We model an interconnection network as an undirected graph $G = (V, E)$; V is the set of nodes and E is the set of links ($|V| = N$). We consider only direct networks with co-packaged modules, so there is no notion of endpoints attached to routers: each node in the network serves both as a router/switch and as a compute endpoint. There are N such nodes in total, and k channels from each node to other node (*radix*). The diameter D denotes the maximum length of shortest paths between any pair of nodes.

B. The Degree-Diameter Problem and Network Design

The degree of a network is determined by current technology, and the diameter is chosen according to the system requirements. Based on this, one would like to maximize the number of nodes in such a network. This is the degree-diameter problem: find the maximum number $n(D, k)$ (or N) of vertices in a graph given maximal degree k and diameter D.

The degree-diameter problem is a major open problem in graph theory. For a comprehensive survey, see [28]. Bounds exist for this problem, but few optimal graphs have been identified. Loz, Pérez-Rosés and Pineda-Villavicencio give two tables [27] with the largest known graphs and bounds as of 2010 for a given degree and diameter. Only a few of the graphs are known to be optimal.

1) The Moore Bound

The Moore bound [29] is the most general upper bound on the number of vertices n for a graph with maximum degree k and diameter D, and is given by

$$N \leq 1 + k \cdot \sum_{i=0}^{D-1} (k-1)^i. \qquad (1)$$

Few graphs of any diameter and degree actually meet the Moore bound; in fact, few even come close. Hoffman and Singleton [29], Bannai and Ito [30], and Damerell [31] have identified all of the graphs that meet the bound.

The Erdős-Rényi polarity graphs were introduced by Erdős and Rényi in [32] and by Brown in [33]. They have diameter 2 and asymptotically approach the Moore bound, which is $N \leq 1+k^2$ for graphs of diameter 2. They also have properties useful for network design, which we exploit for PolarFly.

III. Feasibility Analysis of Candidate Topologies

There are many available topologies. However, not all are suitable for use in a data center, especially in a co-packaged setting. In this section, we investigate representative networks and show that PolarFly meets the data center needs best of all.

We consider Slim Fly [26] (a variant with $D = 2$), Dragonfly [34] (the "balanced" variant with $D = 3$), and HyperX (Hamming graph) [35] that generalizes Flattened Butterflies [36] with $D = 2$. We also use established three-stage Fat Trees [37]. Finally, for completeness, we also consider two Fat Tree variants, Orthogonal Fat Trees [38] and Multi-Layer Full Meshes [38]. In the following, we identify the criteria for a topology to be a suitable candidate for a data center.

Directness. *Direct* networks can be constructed using only one type of co-packaged chip that integrates the compute, routing hardware, and communication ports in the same package. In contrast, indirect networks such as fat trees, require design, fabrication, and deployment of additional chiplet(s) for the switches, which significantly increases their overall cost.

Flexibility. A *flexible* network provides many feasible configurations that could be constructed using available equipment while delivering high performance. This means that one must be able to build networks using switches with feasible radix.

Low Diameter. Upcoming distributed shared-memory systems such as PIUMA [39], and future disaggregated memory systems [40], heavily rely on low-latency remote accesses for performance scalability. This can only be delivered by *scalable* networks with small diameter, **ideally two**, or networks with average path length of two. In case of direct networks, low-diameter topologies also support higher ingestion bandwidth.

Modularity. In a *modular* network, the nodes can be decomposed into smaller units that could be, e.g., racks, blades, or chassis. This feature facilitates manufacturing, deployment and cabling. Most of the considered networks satisfy this requirement. For example, plain Fat Trees consist of pods, while Dragonflies have a group-based structure.

Expandability. A network is *expandable* if its size can be incrementally increased by adding a basic unit, such as a rack, by using empty ports in an under-provisioned network. This need is usually related to budgetary issues – the budget-limited purchased system is smaller than the optimal system, and may only be extended to a larger size later. Incremental growth may be preferred over complete rewiring into a new topology, as the latter is much more disruptive, expensive and time-consuming. While some of the considered networks, like Dragonfly, do enable incremental growth, it is not known how to increase the size of the most competitive target Slim Fly.

We now analyze the considered networks and show whether they satisfy the above criteria. A summary of the analysis is illustrated in Table I. All networks are at least partially modular and flexible. Most networks have diameter two. Only PolarFly satisfies all the criteria almost fully.

Topology	Direct	Modular	Expandable	Flexible	Diameter-2
Fat tree	✖	■	■	■	✖
Dragonfly	◧	■	■	◧	✖
HyperX	◧	■	■	◧	■
OFT	✖	◧	✖	■	■
MLFM	✖	■	✖	◧	■
Slim Fly	■	■	◧	◧	■
PolarFly	■	■	◧	■	■

TABLE I: Feasibility. "■": full support, "◧": partial support, "✖": no support.

IV. PolarFly Topology

In this section, we discuss in detail the graph underlying the PolarFly layout. This mathematical description is used in the construction and for the exploitation of the graph properties.

The topology of PolarFly is an Erdős-Rényi (ER) polarity graph, also known as a Brown graph, constructed using the relationship of points and lines in finite geometry. These were discovered independently by Erdős and Rényi [32] and Brown [33]. There is a great deal of mathematical structure to these graphs, and they have been studied in depth, both in the original papers [32], [33] and other references, e.g. [41]–[43].

ER graphs have several useful features for network design:

- **Low diameter.** They have diameter 2, giving a short path between any two nodes.
- **Scalability.** At the same time, they asymptotically reach the Moore bound, surpassing the scalability of all other diameter-2 topologies. We compare to other diameter-2 topologies that are direct, as needed for co-packaging, and show the comparison in Figure 2.
- **Flexibility.** They cover a wide range of degrees, having degree $k = q + 1$ for every prime power q. While not completely general, they meet or come very close to the radixes of many current and near-term high-radix routers. For example, for $q = 31, 47, 61$ and 127, ER_q may be applied to systems with routers of radix 32, 48, 64 and 128, with all router ports used at radix 32, 48 and 128.

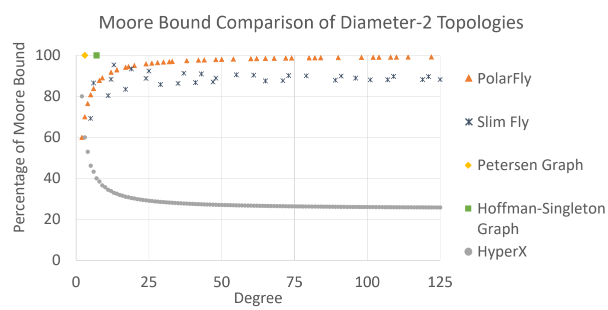


Fig. 2: Scalability of direct diameter-2 topologies in terms of the optimal Moore bound. Petersen (10 vertices) and Hoffman-Singleton (50 vertices) are the only known graphs to achieve the Moore Bound.

A. Some Background on Finite Fields and Their Arithmetic

The construction of ER polarity graphs is based upon the arithmetic operations of finite fields. A field is a set having addition and multiplication, where every element has an additive inverse, and every non-zero element has a multiplicative inverse. The construction of ER polarity graphs depends especially upon the existence of multiplicative inverses.

The set of integers modulo a prime p is an example of a finite field. Finite fields $\mathbb{F}_q$ of order q exist for all prime powers q, and for no other integers. Finite fields are also called Galois extension fields. They are fundamental to many areas in mathematics and computer science, and are discussed in detail in [44]–[46] and in many other references.

It is important to note that addition and multiplication operations in $\mathbb{F}_q$ are quite different from those in $\mathbb{R}$:

- If $q = p$ is a prime, then addition and multiplication are just modular arithmetic over p.
- If $q = p^m$ is a prime power, with $m > 1$, then addition and multiplication in $\mathbb{F}_q$ are derived from modular arithmetic over an irreducible degree-m polynomial over $\mathbb{F}_p$. For further details on arithmetic in the finite fields $\mathbb{F}_q$, with q not prime, see any of the references listed above.

We will use primes $q = p$ for the examples in this paper for simplicity, so the addition and multiplication in the dot products is just modular arithmetic over p. The same graph construction will hold for any prime power $q = p^m$, using the associated arithmetic of $\mathbb{F}_q$ for the dot products.

B. Geometric Intuition

Erdős-Rényi polarity graphs express the orthogonality (or perpendicularity) between vectors, or equivalently, lines passing through the origin. The *dot product* is a convenient way of expressing this; two length-n vectors v and w are orthogonal when $v \cdot w = \sum_{i=1}^{n} v_i w_i = 0$.

Note that multiples of a vector retain the same orthogonality relationships as the original vector. So for our purposes, we may consider all multiples of a vector to be the same, and simply choose one as representative of all its multiples.

ER polarity graphs are defined by vertices that represent length-3 vectors over a field, and edges that exist between two vertices if the vectors they represent are orthogonal. This graph has diameter 2.

As an example, consider ordinary Euclidean 3-dimensional space. The existence of 2-hop paths between any two vectors in the corresponding graph depends upon this fact: any pair of (non-multiple) vectors has a vector to which both are orthogonal: their cross-product. The 2-hop path linking them passes through the orthogonal vector. This is shown in Figure 3.

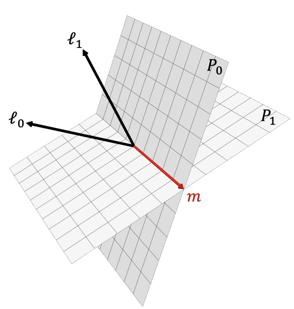


Fig. 3: Let ℓ_0 and ℓ_1 be arbitrary lines in Euclidean 3-space passing through the origin. The line ℓ_0 is perpendicular to the plane P_0, and the line ℓ_1 is perpendicular to the plane P_1. The two planes intersect in a line m which is perpendicular to both ℓ_0 and ℓ_1. A graph including ℓ_0, ℓ_1 and m as vertices has edges (ℓ_0, m) and (m, ℓ_1), so there is a 2-hop path from ℓ_0 to ℓ_1 passing through m. This construction may be generalized to $\mathbb{F}_3^3$, using the dot product to represent perpendicularity.

Euclidean 3-space obviously has infinitely many lines and planes passing through the origin. However, a similar construction may be used to obtain a finite space of dimension 3 over the finite field $\mathbb{F}_q$. The geometric relationships are similar to those discussed above in the Euclidean case. Orthogonality is again expressed by the dot product, using the addition and multiplication from $\mathbb{F}_q$. So the construction over $\mathbb{F}_q$ also gives rise to a graph of diameter 2, but this time the graph is finite.

Because $\mathbb{F}_q$ is finite with modular arithmetic, some non-zero vectors in $\mathbb{F}_q^3$ have the interesting property that they are orthogonal to themselves, which never happens in Euclidean space. For example, consider $\mathbb{F}_3^3$, where the arithmetic operations are modular addition and multiplication $\bmod 3$. The vector $[1, 1, 1]$ is self-orthogonal, since $[1, 1, 1] \cdot [1, 1, 1] = 1 + 1 + 1 = 0 \mod 3$.

C. Construction With Dot Products over $\mathbb{F}_q$

ER polarity graphs are easily constructed using the set of non-zero left-normalized vectors $[x, y, z] \in \mathbb{F}_q^3$ as vertices. These are vectors in which the first non-zero entry is 1.

For example, in $\mathbb{F}_3^3$, $[1, 0, 2]$ and $[0, 1, 0]$ are vectors in the set under consideration, but $[0, 2, 1]$ would not be, since its first non-zero entry is 2. Instead, $[0, 2, 1]$ is multiplied by the multiplicative inverse of 2 (mod 3), giving its left-normalized representation: $2 \cdot [0, 2, 1] = [0, 1, 2]$. The existence of multiplicative inverses in the field $\mathbb{F}_q$ assures us that each non-zero vector can be represented as a left-normalized vector.

ER_q is then constructed as follows:

- The vertices are the left-normalized vectors in $\mathbb{F}_q^3$.
- The edges are pairs (v, w) of vertices that are orthogonal to each other, as per the dot product; in other words, using the addition and multiplication of $\mathbb{F}_q$, the dot product of v and w is 0.

Self-orthogonal vertices are distinguished from the others and are called *quadric*. All other vertices are called *non-quadrics*. Quadrics may be considered to have a self-loop, and play a special role in the construction of PolarFly.

We show an example of this construction in Figure 4. There is a great deal of structure in the ER_q graph, some of which can be seen there and in the graph layout in Figure 6. We exploit this in the construction of an efficient network.

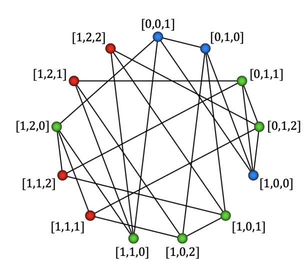


Fig. 4: The dot-product construction of ER_3. The left-normalized vectors of $\mathbb{F}_3^3$ are the vertices, arranged lexicographically, and clockwise starting at the top. The base field is $\mathbb{F}_3$. 3 is a prime rather than a prime power, so the operations are addition and multiplication mod 3. Edges exist between vertices v and w when the dot product $v \cdot w$ is 0, using arithmetic from $\mathbb{F}_3$. For example, the vertex $[1, 1, 1]$ is adjacent to $[0, 1, 2]$, since the dot product $([1, 1, 1] \cdot [0, 1, 2]) = 0 + 1 + 2 \equiv 0 \mod 3$. The self-adjacent quadrics (W) for which the dot products $w \cdot w$ are 0 are red. Vertices adjacent to quadrics (V_1) are green, and vertices not adjacent to quadrics (V_2) are blue.

D. Minimal paths, intermediate points and routing

Since an ER graph has diameter 2, the minimal path between two vertices is either 1 or 2 hops. A path is of length 1 if and only if the vertices are orthogonal (so their dot product is 0).

There is only one minimal path of length 1 or 2 between vertices. For reasons of efficiency, table-based routing is the best method for finding paths. However, the unique intermediate vertex on a 2-hop path may also be found by solving a pair of linear equations representing the dot-product construction.

Intuitively, the two vertices are represented by two distinct lines, which are orthogonal to a single line (or single vertex in projective space), as seen in Figure 3. So the problem of finding the intermediate vertex in a minimal 2-hop path reduces to finding that unique orthogonal line.

The intermediate vertex v between vertices s and d is orthogonal to both s and d, so $s \cdot v = d \cdot v = 0$. Thus v is found by solving this system of equations, via an augmented matrix:

$$\left[\begin{array}{ccc|c} s_0 & s_1 & s_2 & 0 \\ d_0 & d_1 & d_2 & 0 \end{array}\right]$$

Since s and d are not multiples of each other, and since all vectors in ER_q must be left-normalized, there will be a unique solution v to this system of equations, which may be found by Gaussian elimination or otherwise.

For example, in ER_3, the vectors $(0, 0, 1)$ and $(1, 2, 2)$ are not orthogonal, (since their dot product is not 0) so the minimal path has length 2. The left-normalized solution in $\mathbb{F}_3^3$ to the resulting augmented matrix

$$\left[\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 0 \end{array}\right]$$

is $(1, 1, 0)$. Figure 4 confirms that $(1, 1, 0)$ is indeed intermediate on the unique length-2 path between $(0, 0, 1)$ and $(1, 2, 2)$.

Another, perhaps simpler way to obtain the vertex v orthogonal to both s and d is to compute the cross-product of s and d, as discussed in [43]:

$$s \times d = (s_2 d_3 - d_3 s_2,\ s_3 d_1 - d_3 s_1,\ s_1 d_2 - s_2 d_1) \qquad (2)$$

Since multiples of a vector represent the same vertex, the coordinates obtained from cross-product can be left-normalized to obtain v. For example, in ER_3, the intermediate vertex between the vectors $(0, 0, 1)$ and $(1, 2, 2)$ is given by:

$$(-2, 1, 0) = (1, 1, 0)$$

as vectors modulo 3.

Because vectors in ER_q are left-normalized, this method is quite efficient, in the worst case needing only two multiplies and three adds in $\mathbb{F}_q$ to compute the cross-product, then at most another two multiplies for the left-normalization.

E. Formal Construction

1) A Bipartite Graph From Finite Geometry

The projective plane $\mathrm{PG}(2, q)$ is a geometric structure that arises from projecting lines and planes in three-dimensional space over $\mathbb{F}_q$ to points and lines respectively. As discussed above, projective points can be thought of as the left-normalized vectors in $\mathbb{F}_q^3$. In more formal language, each point $[a]$ in $\mathrm{PG}(2, q)$ is an equivalence class of 3-tuples where $(a_1, a_2, a_3) \sim (b_1, b_2, b_3)$ if and only if these tuples are multiples of each other. Projective lines $(b_1 : b_2 : b_3)$ contain all points $[x]$ so that $b_1x_1 + b_2x_2 + b_3x_3 = 0$ in $\mathbb{F}_q$. By counting such vectors, we see there are $q^2 + q + 1$ points in $\mathrm{PG}(2, q)$. Dually, there are $q^2 + q + 1$ lines in $\mathrm{PG}(2, q)$ as we will see.

We will construct ER_q graphs for any prime power q using properties of points and lines in $\mathrm{PG}(2, q)$. To begin, we first build a bipartite graph $B(q)$. The vertex set of $B(q)$ is $U \cup V$ where U is the set of points in $\mathrm{PG}(2, q)$ and V is the set of lines in $\mathrm{PG}(2, q)$. There is an edge between $v \in U$ and $w \in V$ if and only if the point v lies on the line w. In $\mathrm{PG}(2, q)$, each point lies on $q + 1$ lines, and each line contains $q + 1$ points. The graph $B(q)$ has $2(q^2+q+1)$ vertices and degree $q+1$. The graph $B(q)$ has diameter 3, which will be reduced to diameter 2 by a polarity construction in the next section.

2) Decreasing the Diameter Using a Polarity Map

For each point $[a]$ in $\mathrm{PG}(2, q)$, the *dual* $[a]^\perp$ is the line in $\mathrm{PG}(2, q)$ which contains all points (x_1, x_2, x_3) so that $a_1x_1 + a_2x_2 + a_3x_3 = 0$. Since the dual map is a bijection, this shows that $\mathrm{PG}(2, q)$ contains $q^2 + q + 1$ lines. The dual of a line can be defined symmetrically. Notice that $([a]^\perp)^\perp = [a]$. Clearly, $[x]$ lies on line $[a]^\perp$ if and only if $[a]$ lies on $[x]^\perp$. Such a bijection is also known as *polarity*.

In order to decrease the diameter of $B(q)$, we use this polarity: take $B(q)$ and glue the vertices $v \in U$ and $w \in V$ together if and only if $[w] = [v]^\perp$. That is, combine the point $[v] \in U$ and the line $[w] \in V$ together if they are duals of each other. Define ER_q to be the graph formed by applying this gluing process to $B(q)$. ER_q has q^2+q+1 vertices since we glued pairs of vertices in $B(q)$ together. The degree is still $q + 1$, since every line passing through point $[v]$ is glued to a point on $[v]^\perp$, but now the diameter is reduced to 2.

This construction is quite general: if a polarity map exists on a bipartite graph with $N = 2n$ vertices, maximum degree k, and diameter D, it can be used to construct another graph with n vertices, maximum degree k, and diameter $D - 1$.

3) Quadric Vertices

In a finite geometry, the point $[a]$ may lie on its own dual, the line $[a]^\perp$. When this occurs, i.e., when $a_1^2 + a_2^2 + a_3^2 = 0$, the vertex $[a]$ is called a *quadric vertex*. In ER_q, there is a loop at vertex $[a]$ since $[a]$ and $[a]^\perp$ are glued together. These quadric vertices are the same as the self-orthogonal vectors discussed at the end of Section IV-B, and will be discussed further in Section IV-F in terms of the layout of the network.

F. Structural Properties of ER polarity graphs

We make heavy use of the structure of ER graphs in the design of the network discussed in this paper.

ER graphs ER_q have $N = q^2 + q + 1$ vertices, degree $k = q + 1$, and diameter $D = 2$. The vertex set of ER_q can be divided into three disjoint subsets [43]:

- $W(q) \rightarrow$ set of $q + 1$ quadric vertices.

- $V_1(q) \rightarrow$ set of $\frac{q(q+1)}{2}$ vertices adjacent to $W(q)$.
- $V_2(q) \rightarrow$ set of $\frac{q(q-1)}{2}$ vertices not adjacent to $W(q)$.

The following properties used in the construction and analysis of the network were presented by Bachratý and Širáň in [41].

Property 1. *[41] For every odd prime power q, ER_q has the following properties:*

1) *No two vertices in $W(q)$ are directly connected. Every vertex in $W(q)$ is adjacent to exactly q vertices in $V_1(q)$.*
2) *Every vertex in $V_1(q)$ is adjacent to exactly 2 vertices in $W(q)$, and $\frac{q-1}{2}$ vertices each in $V_1(q)$ and $V_2(q)$.*
3) *Every vertex in $V_2(q)$ is adjacent to exactly $\frac{q+1}{2}$ vertices each in $V_1(q)$ and $V_2(q)$.*
4) *There is exactly one path of length two between every vertex pair (considering the self-loop of the self-adjacent quadrics as an edge).*
5) *As a corollary, the edges incident with quadric vertices do not participate in any triangle. Any edge incident with two non-quadric vertices participates in exactly one triangle.*

V. PolarFly Layout

Network layouts need a modular topology decomposable into smaller units for easy and cost-effective deployment. Since ER graphs are derived from polarity quotient graphs of finite projective planes, such structures are not trivially available, in contrast to topologies derived from multiple generating sets, such as Slim Fly [26] or Bundlefly [47], in which modular units can be readily obtained from individual generators.

Instead, we use the connectivity of quadrics to other vertices to obtain a **modular** and **generalized** layout for PolarFly. Property 1 from Section IV-F tells us that every quadric $v \in W(q)$ is connected to exactly q vertices in $V_1(q)$. We use these q vertices to construct q clusters, plus the quadrics themselves as the $(q+1)^{st}$ cluster.

More formally, Algorithm 1 assigns the vertices in ER_q into $q{+}1$ clusters, which corresponds to assigning nodes to racks in PolarFly . We use the terms *racks* and *clusters* interchangeably.

For brevity, we only discuss ER_q for odd q (even radix) as even prime powers $q = 2^i$ are sparse in the set of all prime powers. The layout for even q is similarly modular, and is derived using an analogue to Property 1 for even q [41].

Algorithm 1 PolarFly layout

$ER_q \leftarrow$ ER Graph of max degree $q+1$
1: Initialize empty clusters (racks) $C_0, C_1 \ldots, C_q$
2: Add all quadrics $W(q)$ to C_0
3: Select an arbitrary quadric $v \in W(q)$
4: **for each** vertex u adjacent to v
5: Add u to an empty cluster C_i
6: Add all non-quadric neighbors of u to C_i

Figure 5 is a diagram of the layout, and Figure 6 shows an example of the ER_q graph structure supporting this layout.

Proposition V.1. *Algorithm 1 adds every vertex in ER_q to exactly one cluster.*

Proof. Each vertex is at most 2 hops away from the quadric v selected in Algorithm 1 line 3. The algorithm assigns clusters to $W(q)$ and all non-quadrics at shortest distance ≤ 2 from v. Thus, every vertex is added to at least one cluster. By Property 1.5, the vertices adjacent to v are independent and have no common neighbor aside from v. Hence, non-quadric vertices are added to at most one cluster. The quadrics are added to exactly one cluster, C_0. □

A. Intra-rack Layout

1) The quadrics cluster

The layout of the quadrics cluster is quite simple: there are no edges within C_0 by Property 1.1. This is seen in ER_7 in Figure 6(a), and also in ER_3 in Figure 4.

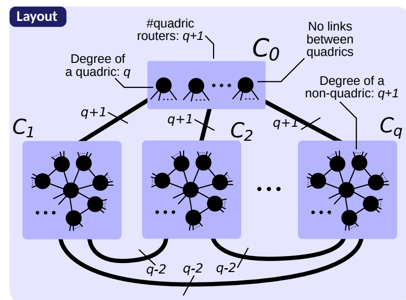


Fig. 5: The layout.

2) Non-quadrics clusters

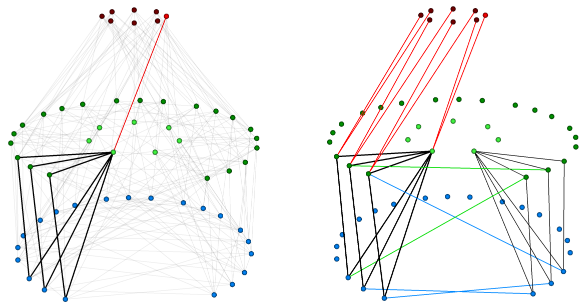
The layout of a non-quadrics cluster is also rather simple: the edges of a non-quadrics cluster form a fan made up of $\frac{q-1}{2}$ triangles. The fan has a center vertex, and centers of all fans have a common quadric neighbor. The triangles share only the center, and are otherwise disjoint, giving a fan-blade appearance. This is easily seen in ER_7 in Figure 6, and may be traced out in ER_3 in Figure 4.

Proposition V.2. *The vertices of a cluster form $\frac{q-1}{2}$ triangles, all having the center of the cluster as a common vertex.*

Proof. Let v be the quadric chosen as a starter. Then every vertex c_i adjacent to v is selected as the center of cluster C_i. By Property 1.2, c_i has $q-1$ non-quadric neighbors, and with c_i, these make up the q elements of the non-quadric cluster C_i. So there is an edge between c_i and any other vertex in C_i.

If u is one of the non-center vertices, Property 1.5 tells us that the edge (c_i, u) is contained in exactly one triangle in C_i. This shows that C_i consists of exactly $\frac{q-1}{2}$ edge-disjoint triangles, each of which contains c_i. □

This fan structure gives PolarFly a modular layout with isomorphic structures induced by all q non-quadric clusters. In physical terms, this means that the q^2 non-quadric nodes can be deployed as q identical copies of the same rack. Moreover, the *fan layout for PolarFly is generalizable to any odd prime power q*, unlike Slim Fly where generator sets and thus the intra-rack layout can change significantly with router radix.



(a) A non-quadric cluster. An arbitrary quadric (in red) generates the center vertices (in bright green). For each center vertex c, all non-quadric neighbors of c are added to the cluster. Edges of one such cluster are rendered in black. This figure illustrates Section V-A.

(b) Inter-cluster connectivity from the cluster shown in (a). Edges to the quadrics are shown in red, edges to the $V_1(q)$ vertices in another cluster in green, and edges to the $V_2(q)$ vertices in the other cluster in blue. This figure illustrates Propositions V.3 and V.4.

Fig. 6: Layout for PolarFly with $q = 7$. The quadrics $W(7)$ (quadrics) are in red (top layer), $V_1(7)$ are in green (middle layer) and $V_2(7)$ are in blue (bottom layer). The left figure shows the $\frac{q-1}{2} = 3$ fan-blades (triangles) emanating from the center c (light green) of a non-quadric cluster. The right figure shows the $q-2 = 5$ edges between two non-quadric clusters, and $q+1 = 8$ edges between a non-quadric cluster and quadrics.

B. Inter-rack Layout

There are two types of racks in PolarFly : one quadric rack C_0 and q non-quadric racks $\{C_1, ...C_q\}$. In this section, we describe the connectivity between two racks. For ease of notation, we use $V_1(q, C_i)$ and $V_2(q, C_i)$ to denote vertices of C_i in $V_1(q)$ and $V_2(q)$ subsets, respectively (Section IV-F).

1) Connections between C_i and the quadrics

Proposition V.3. *For every cluster C_i, with $i > 0$:*

1) *Each vertex in $V_1(q, C_i)$ is adjacent to exactly two vertices in C_0.*
2) *There are exactly $q + 1$ links between C_i and C_0.*
3) *Each quadric in C_0 is adjacent to exactly one vertex in C_i.*

Proof. Proposition V.3.1 follows directly from Property 1.2. C_i has $\frac{q+1}{2}$ vertices from $V_1(q)$: the center vertex and its $\frac{q-1}{2}$ neighbors in $V_1(q)$. Hence, there are $q+1$ edges between C_i and C_0. Since there is an average of 1 link to C_i per quadric, to prove Proposition V.3.3, it is sufficient to show that no quadric is adjacent to more than one vertex in C_i. Let $v \in W(q)$ be adjacent to multiple vertices in C_i. Either v lies in a triangle or there are multiple 2-hop paths between v and the center of C_i. This contradicts Property 1.5. $\square$

2) Connections between the non-quadric clusters

Proposition V.4. *Let C_i and C_j be two clusters with $0 < i, j \leq q$ and $i \neq j$. Let c_i denote the center of cluster C_i. Then:*

1) *Every vertex in $V_2(q, C_i)$ is adjacent to exactly one vertex in C_j.*
2) *There are $q - 2$ independent edges between C_i and C_j.*
3) *There exists a vertex $u' \in V_1(q, C_i) \backslash \{c_i\}$ such that every vertex in $V_1(q, C_i) \backslash \{u', c_i\}$ is adjacent to exactly one vertex in C_j, and u' is not adjacent to C_j.*

Proof. For a vertex $w \in C_i$ such that $w \neq c_i$, let $S(w)$ denote the set of edges between w and all clusters other than C_i and C_0. If $w \in V_1(q, C_i)$, then $|S(w)| = q-3$, and if $w \in V_2(q, C_i)$, then $|S(w)| = q - 1$. Every edge in $S(w)$ must connect w to a unique cluster, otherwise there will be multiple 2-hop paths from w to the center of a cluster. This proves Proposition V.4.1.

We give a brief sketch of the proof for Proposition V.4.2 for brevity. Adding $|S(w)|$ for all vertices $w \in C_i$, we see that there are $(q-1)(q-2)$ edges between C_i and other non-quadric clusters. If Proposition V.4.2 is false, then there exists a non-quadric vertex at least 3-hops away from the center c_i. This is a contradiction since ER_q has diameter 2.

From Propositions V.4.1 and V.4.2, we see that there are $(q-3)/2$ independent edges between $V_1(q, C_i)$ and C_j. None of these edges are incident to c_i. Further, $|V_1(q, C_i) \setminus \{c_i\}| = \frac{q-1}{2}$. Hence, there must be exactly one $u' \in V_1(q, C_i) \backslash \{c_i\}$ which is not adjacent to C_j. Vertex u' can be identified easily as it shares a common quadric neighbor with center c_j. $\square$

Propositions V.3 and V.4 show that there are $q - 2$ links between all $\frac{q(q-1)}{2}$ pairs of non-quadric racks, and $q + 1$ links between quadric rack and each of the q non-quadric racks. Thus, PolarFly exhibits a nearly balanced all-to-all connectivity between racks. Moreover, links between any pair of racks can be *bundled* into cost-effective solutions such as a single multi-core fiber [47].

Using ER_7 as an example, Proposition V.3 may be seen in the red edges of Figure 6(b), and Proposition V.4 may be seen in the green and blue edges of Figure 6(b).

C. Triangles and Other Polygons in PolarFly

PolarFly has many triangles, and no quadrangles. The absence of quadrangles is because each pair of non-quadric vertices has exactly one path of length 2, by Property 1.4. If PolarFly had a quadrangle, there would then be vertices with two paths of length 2 between them.

We discuss the abundance of triangles and the implications of this throughout the rest of this section. This becomes important when we analyze the path diversity of PolarFly in the next section.

1) The Number and Types of Triangles in PolarFly

Triangles are of two kinds: those entirely internal to a non-quadric cluster, and those linking three distinct non-quadric clusters together.

Proposition V.5. *There are $\binom{q+1}{3}$ triangles in ER_q.*

Proof. By Property 1.5, edges not incident to a quadric participate in one triangle, and edges incident to a quadric participate in no triangles. There are $\frac{q(q+1)^2}{2}$ total edges, and $q(q+1)$ edges incident to a quadric. Thus

$$\frac{(q+1)q(q-1)}{2}$$

edges are not incident to a quadric. This implies that there are

$$\frac{(q+1)q(q-1)}{6} = \binom{q+1}{3}$$

triangles. $\square$

Proposition V.6. *The triangles of ER_q either join three distinct non-quadric clusters or are internal to a single non-quadric cluster. In particular,*

(a) $\binom{q}{3}$ *triangles join non-quadric clusters.*

(b) $\binom{q}{2}$ *triangles are internal to non-quadric clusters.*

Proof. There are a total of $\binom{q}{2}(q-2)$ inter-cluster edges across non-quadric clusters. From Property 1.4, each such edge participates in exactly one triangle. So there are

$$\frac{q(q-1)(q-2)}{2 \cdot 3} = \binom{q}{3}$$

such inter-cluster triangles. Each of these must be made up of three distinct clusters: if not, then the edge (a, b) internal to a cluster will also be on a triangle entirely internal to that cluster, thus there will be two length-2 paths between a and b, which cannot occur.

There are $\frac{q-1}{2}$ triangles internal to a non-quadric cluster, by Proposition V.2, and there are q such clusters, so there are

$$q \cdot \frac{q-1}{2} = \binom{q}{2}$$

triangles internal to non-quadric clusters.

Finally,

$$\binom{q}{3} + \binom{q}{2} = \binom{q+1}{3},$$

so this accounts for all triangles in ER_q, by Proposition V.5. ☐

2) Intra-cluster triangles

The $\frac{q-1}{2}$ internal triangles in a non-quadric cluster share the cluster center as a triangle vertex, giving the edges of the cluster the form of a triangle fan-out. They pairwise share no other vertices.

If $q \equiv 1 \mod 4$, the vertices of each internal triangle consist of the center and either two vertices from V_1, or two vertices from V_2.

If $q \equiv 3 \mod 4$, the vertices of each internal triangle consist of the center, an element of V_1 and an element of V_2. This may be seen in Figures 6b and 13.

3) Inter-cluster Triangles and a Block Design on Clusters

The remaining triangles are all inter-cluster triangles. In this section, we prove the following Theorem V.7, which says that every non-quadric cluster triplet is joined by one triangle.

This gives rise to a $3-(q, 3, 1)$ block design, where the q non-quadric clusters are the points, triangles joining cluster triplets are the blocks, and each set of 3 points appears in 1 block. Block designs are well known combinatorial structures that express symmetries in terms of the points and blocks [48], and are therefore of interest in constructing networks.

Theorem V.7. *Every triplet of non-quadric clusters under any cluster layout is connected by exactly one triangle.*

The theorem is a consequence of the symmetry between all length-2 paths in ER_q that have a quadric as the intermediate node. This symmetry was first shown in [43], and is restated below as Theorem V.8.

To prove Theorem V.7, it suffices to show that all non-quadric triplets are joined by at most one triangle. The theorem then follows from an application of the pigeonhole principle.

To do this, we show a corollary of Theorem V.8 that expresses a symmetry on non-quadric cluster triplets. A technical lemma exhibits an example of such triplets that are joined by at most one triangle. The symmetry in the corollary then implies that this is true of all non-quadric triplets in ER_q.

Theorem V.8. *[43, Corollary 5] Let (s_0, w_0, d_0) and (s_1, w_1, d_1) be paths of length 2 in ER_q, where s_i and $d_i \in V_1$ and $w_i \in W$, the quadrics cluster. Then there exists some automorphism θ of ER_q such that $\theta(w_0) = w_1$, $\theta(s_0) = s_1$ and $\theta(d_0) = d_1$.*

Corollary V.9. *If there exists some non-quadric cluster X such that every non-quadric cluster triplet that includes X is joined by at most one triangle, then every non-quadric cluster triplet is joined by at most one triangle.*

Proof. Let w be the starter quadric, and let X be a cluster meeting the above condition. Let x_c be the center of X.

We assume that there exists some triplet (D, E, F) of distinct non-quadric clusters joined by more than one triangle, and show a contradiction.

Let d_c, e_c and f_c be the centers of D, E, and F respectively. By assumption, the triplet (D, E, F) is joined by two distinct triangles with vertices (d_0, e_0, f_0) and (d_1, e_1, f_1).

Let $Y \neq X$ be an arbitrary cluster with center y_c. By Theorem V.8, there is an automorphism θ of ER_q so that $\theta(w) = w$, $\theta(d_c) = x_c$, and $\theta(e_c) = y_c$. Any automorphism of ER_q preserves edges, so $\theta(f_c)$ is connected to $\theta(w) = w$, making $\theta(f_c)$ the center of some cluster Z. Again, θ preserves edges, so two distinct triangles $(\theta(d_0), \theta(e_0), \theta(f_0))$ and $(\theta(d_1), \theta(e_1), \theta(f_1))$ link X, Y and Z.

By the condition on X, (X, Y, Z) cannot be a triplet, so Z must be one of X or Y. But then the triangle $(\theta(d_0), \theta(e_0), \theta(f_0))$ joins exactly two non-quadric clusters. This contradicts Proposition V.6. ☐

Lemma V.10. *Let X be a non-quadric cluster whose center x has form $(1, x_1, x_2)$ as a point in $\mathbb{P}^2(\mathbb{F}_q)$. Then any triple of distinct non-quadric clusters that includes X is joined by at most one triangle.*

Proof. Let X be as stated, and let $Y, Z \neq X$ be distinct clusters such that the triple (X, Y, Z) is joined by the triangle (a, b, c) with $a \in X$, $b \in Y$ and $c \in Z$. Write each point r as $r = (r_0, r_1, r_2)$, a point in $\mathbb{P}^2(\mathbb{F}_q)$. Given centers x, y, and z of X, Y, and Z, we will write down a system of equations to count how many possible triples (a, b, c) can exist.

By construction, the points a, b and c are connected to the respective centers x, y and z, and to each other in ER_q. This implies that

$$a \cdot x = b \cdot y = c \cdot z = a \cdot b = b \cdot c = c \cdot a = 0$$

in $\mathbb{F}_q$. Because $a \cdot x = 0$ and $x = (1, x_1, x_2)$,

$$a_0 = -(a_1 x_1 + a_2 x_2). \tag{3}$$

Because $a \cdot b = b \cdot y = 0$ and $a \cdot c = c \cdot z = 0$, (a, b, y) and (a, c, z) are two-hop paths. The cross-product derivation of the intermediate vertex of a two-hop path given in (2) from Section IV-D then shows that

$$b = a \times y \quad \text{and} \quad c = a \times z. \tag{4}$$

Using $b \cdot c = 0$, the above cross-products may be substituted for b and c, giving:

$$(a \times y) \cdot (a \times z) = 0$$

Substituting (3) into this equation gives

$$r_{11} a_1^2 + r_{12} a_1 a_2 + r_{22} a_2^2 = 0 \tag{5}$$

where r_{11}, r_{12} and r_{22} are constants in y_i, z_i for $i \in \{0, 1, 2\}$. Notice that (4) implies we can determine the entries of the points b and c completely once we determine a. Thus we have reduced our problem to understanding the number of solutions to (5) in terms of a_1 and a_2.

Either a_2 is 0 or a_2 is invertible. If $a_2 = 0$, then (5) implies $a_1 = 0$, and (3) implies $a_0 = 0$. This is impossible since $(0, 0, 0) \notin \mathbb{P}^2(\mathbb{F}_q)$. So a_2 must be invertible.

Without loss of generality, we may then solve for a vertex of the form $a' = (a'_0, a'_1, 1)$, since a' may then be multiplied by a_2 and then left-normalized to give $a \in \mathbb{P}^2(\mathbb{F}_q)$.

In that case, (5) reduces to a quadratic polynomial in a'_1, which has at most two solutions. However, notice that $a = b = c = w$, the starter quadric, satisfies all of the equations. In this case, a, b, and c do not form a triangle. This implies there is at most one valid solution to (5) and at most one vertex triplet (a, b, c) that form a triangle between the clusters X, Y and Z. $\square$

We are now ready to prove the main theorem of this section.

Proof of Theorem V.7. Let $w = (w_0, w_1, w_2)$ be the quadric vertex that generates the cluster layout. At least one of w_1 and w_2 is non-zero, since w is non-zero and self-orthogonal.

There is thus at least one vector x having form $(1, x_1, x_2)$ that is orthogonal to w (as can be calculated using the dot product on w), and this vector x is the center of some non-quadric cluster X. By Lemma V.10, any non-quadric cluster triplet that includes X is joined by at most one triangle.

Corollary V.9 then implies that every non-quadric cluster triplet is joined by at most one triangle. By Proposition V.6(a), the number of inter-cluster triangles is $\binom{q}{3}$, which is the same as the number of non-quadric cluster triplets. The theorem immediately follows by the pigeonhole principle. $\square$

4) Distribution of inter-cluster triangles

We can further classify the triangles in ER_q. The distribution of the inter-cluster triangles is as shown in Table II. The table follows from a simple combinatorial argument.

We note that every element of V_1 serves as a center in the two layouts induced by its adjacent quadrics. Since the choice of a particular layout does not affect adjacency at all, triangles remain the same in all layouts, in terms of participating vertices. Any triangle with a participating center is entirely internal to the center's cluster. So if $q \equiv 1 \mod 4$, any triangle holding an element of V_1 must be of the form (v_1, v_1, v_1) or (v_1, v_2, v_2). Likewise, if $q \equiv 3 \mod 4$, any triangle holding an element of V_1 must be of the form (v_1, v_1, v_2). We also know that the total number of inter-cluster triangles for any q is $\binom{q}{3}$.

	(v_1, v_1, v_1)	(v_1, v_1, v_2)	(v_1, v_2, v_2)	(v_2, v_2, v_2)
$q \equiv 1 \mod 4$	$\frac{q(q-1)(q-5)}{24}$	0	$\frac{q(q-1)^2}{8}$	0
$q \equiv 3 \mod 4$	0	$\frac{q(q-1)(q-3)}{8}$	0	$\frac{(q+1)q(q-1)}{24}$

TABLE II: Distribution of inter-cluster triangles, of different forms. The variable v_1 indicates a vertex in V_1, and the variable v_2 indicates a vertex in V_2.

Clusters are either internal to a cluster, or entirely inter-cluster, as in the proof of Proposition V.6. This also implies that if a triangle is internal to a cluster with a center c in a given layout, it will be entirely inter-cluster for any layout in which c is not a center, in other words, layouts induced by any of the $q - 1$ quadrics not adjacent to c.

We assume some particular layout starting with an arbitrary starting quadric, and calculate the inter-cluster triangles for that layout. The triangles of course remain the same for any particular layout.

Case 1: $q \equiv 1 \mod 4$. First, we calculate the number of inter-cluster triangles of the form (v_1, v_1, v_1). Choose one of the q non-quadric clusters at random. There are $\frac{(q-1)}{2}$ non-center V_1 elements in the cluster. We choose one of these and call it w. Looking at a layout in which w is a center, w participates in $\frac{q-1}{2}$ triangles of the form (v_1, v_1, v_1), of which exactly two are internal to the cluster, so the other $\frac{q-5}{2}$ triangles are entirely inter-cluster. Considering all the non-center V_1 elements in all the clusters, we get

$$\frac{q(q-1)(q-5)}{4}$$

triangles, but each is counted six times. So there are

$$\frac{q(q-1)(q-5)}{24}$$

triangles in total of the form (v_1, v_1, v_1). We also have that w participates in $\frac{q-1}{2}$ triangles of the form (v_1, v_2, v_2). None of these is internal to the cluster, since all internal triangles of that form are (c, v_2, v_2), where c is the center. Considering all the non-center V_1 elements in all the clusters, we get

$$\frac{q(q-1)^2}{4},$$

but each is counted twice. So there are

$$\frac{q(q-1)^2}{8}$$

triangles in total of the form (v_1, v_2, v_2). Since

$$\frac{q(q-1)^2}{8} + \frac{q(q-1)(q-5)}{24} = \binom{q}{3},$$

we have accounted for all of the inter-cluster triangles for $q \equiv 1 \mod 4$.

Case 2: $q \equiv 3 \mod 4$. First, we calculate the number of inter-cluster triangles of the form (v_1, v_1, v_2). Choose one of the q non-quadric clusters at random. There are $\frac{q-1}{2}$ non-center V_1 elements in the cluster. We choose one of these and call it w. We see by considering a layout in which w is a center that w_0 participates in $(\frac{q-1}{2}$ triangles of the form (v_1, v_1, v_2), of which exactly 1 is internal to the cluster, so the other $\frac{q-3}{2}$ triangles are entirely inter-cluster. Considering all the non-center V_1 elements in all the clusters, we get $\frac{q(q-1)(q-3)}{4}$ triangles, but each is counted twice. So there are

$$\frac{q(q-1)(q-3)}{8}$$

triangles in total of the form (v_1, v_1, v_2). We know that when $q \equiv 3 \mod 4$, there are no triangles of the form (v_1, v_2, v_2) nor of the form (v_1, v_1, v_1), so the remaining triangles must be of the form (v_2, v_2, v_2), and there are

$$\binom{q}{3} - \frac{q(q-1)(q-3)}{8} = \frac{(q+1)q(q-1)(q-1)}{24}$$

of these.

As a corollary of Table II, we have Table III, giving the possible types of the intermediate vertex of an alternative 2-hop path between two adjacent vertices. Such a 2-hop path always exists if neither of the vertices are quadric.

		$\mathbf{v_1}$	$\mathbf{v_2}$
$q \equiv 1 \mod 4$	$\mathbf{v_1}$	v_1	v_2
	$\mathbf{v_2}$	v_2	v_1
$q \equiv 3 \mod 4$	$\mathbf{v_1}$	v_2	v_1
	$\mathbf{v_2}$	v_1	v_2

TABLE III: Types of the intermediate vertices in a 2-hop path between two adjacent non-quadric vertices.

VI. EXPANDABILITY

Expandability is crucial in budget-driven scenarios, as discussed in Section III. In an underprovisioned expandable network, unused ports on the nodes can be used to incrementally connect additional nodes to the network. In this section, we show that PolarFly affords **incremental expansion** and present two methods to accomplish this.

Importantly, these methods do not require rewiring of the existing links. They offer a trade-off across different parameters, as summarized in Table IV. These methods are based on *cluster replication* in PolarFly, which is defined as follows:

Definition VI.1. *Given a graph $G(V, E)$, replication of a vertex cluster $C \subseteq V$ creates a new graph $G'(V \cup C', E')$. For every vertex $v \in C$, there exists a replica $v' \in C'$ such that in the graph G':*

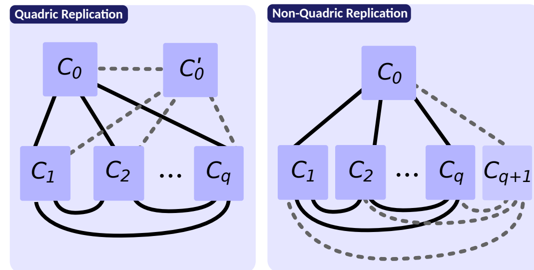


Fig. 7: Expansion methods.

- *For every intra-cluster edge $(v, w) \in E$ between two vertices $\{v, w\} \in C$, the corresponding replicas $\{v', w'\} \in C'$ are also adjacent, i.e. $(v', w') \in E'$.*
- *For every inter-cluster edge $(v, w) \in E$ where $v \in C$ and $w \notin C$, the replica v' is adjacent to w, i.e. $(v', w) \in E'$.*

Physically, replication is achieved by simply adding an additional rack of nodes, which has similar intra-rack layout and connectivity to rest of the clusters as its original counterpart. Hence, cluster replication methods (sec.VI-A and VI-B) allow *modular expansion without rewiring any of the existing links*.

A. Replicating the Quadric Cluster

One way to expand PolarFly is to replicate quadric cluster C_0 as per Def.VI.1, until the desired scale is reached. After replication, to increase the network radix of quadrics, we directly connect every quadric $v \in C_0$ and all of its replicas with each other. It can be shown that every replication of C_0:

1) Increases the number of vertices by $q+1$, while preserving the diameter $D = 2$.
2) Increases the degree of quadrics $W(q)$ (and their replicas) and vertices in $V_1(q)$ by 1 and 2, respectively.
3) Creates $q+1$ edges between the replicated cluster C_0' and all other clusters.

With this method, using n additional ports per node, the size of PolarFly can be increased by $\frac{n(q+1)}{2}$, while keeping the diameter $D = 2$. However, new links are only added between quadric nodes and $V_1(q)$. Hence, a large number of quadric replications can result in a non-uniform degree distribution.

Method	Scalability	Degree Distribution	Diameter	Average Shortest Path Length	Rewiring
Replicate Quadrics	$\frac{q+1}{2}$	Non-uniform	2	< 2	None
Replicate Non-Quadrics	$\approx q$	Uniform	3	< 2	None

TABLE IV: Characteristics of Expansion Methods. **Scalability** refers to the increase in number of nodes per unit increase in the maximum network radix.

B. Replicating Non-Quadric Clusters

In this method, we expand PolarFly by replicating non-quadric clusters $C_{i|i>0}$ in a round-robin order, as per Def.VI.1. The replica of C_i is labeled C_{q+i}, as shown in Figure 7.

Replicating a non-quadric cluster does not add edges to existing center vertices, which can lead to a non-uniform degree distribution. To mitigate this, we note that for every non-quadric cluster C_j where $i \neq j$ (and replica C_{q+j} if it exists), there is exactly one vertex in C_i with no edges to

C_j (C_{q+j}, respectively). We connect the replica of this vertex with the centers of C_j (C_{q+j}, respectively). It can be shown that for any $n \leq q$, n such replications of non-quadric clusters:

1) Increase the number of vertices by qn, which is approximately $2\times$ compared to quadric replication.
2) Increase the maximum degree by $n + 1$.
3) Increase diameter to 3 – for every vertex $u \in C_{i|i>0}$ (replica $u' \in C_{i+q}$), there are at most $q - 1$ vertices, all in replica C_{i+q} (C_i, respectively), that are at a shortest distance of 3 from u (u', respectively).

With non-quadric repliction, new links are distributed across all vertices, providing a near-uniform degree distribution. While the diameter of topology increases to 3, the average shortest path length is still clearly less than 2.

VII. Routing

To facilitate the adoption of PF, we rely on established schemes and show in the evaluation (Section VIII) that they deliver high performance. However, to show the highest PF potential, we also develop a new adaptive protocol suited specifically for PF. Note that under co-packaged setting, nodes and routers are the same entity in direct networks.

A. Minimal Static Routing

With minimal static routing, a packet is routed from its source router R_s over the minimal path to its destination router R_t.

B. Valiant Routing

Let R_s and R_t denote the source and destination routers, respectively. For each packet, the Valiant routing scheme [49] selects a random router R_r such that $R_r \neq R_s$ and $R_r \neq R_t$. Then, it routes the packet from R_s to R_r and R_r to R_t along the corresponding shortest paths. This avoids potential hot spots in the network, but reduces the available bandwidth.

The general Valiant design selects *some* intermediate router. For PolarFly, we use a variant which we call *Compact Valiant*, where R_r is chosen from the neighborhood of R_s. The path length for any packet in Compact Valiant is at most 3-hops, as opposed to 4-hops in general Valiant. This reduces the amount of bandwidth wasted on links due to intermediate traffic.

However, the 3-hop route selection will be disadvantageous if the shortest path between R_r and R_t goes through the source router R_s. In this scenario, the random neighbor selection would result in the packets bouncing back to the source router. Fortunately, in PolarFly, this situation is easily avoided as it occurs only when R_s and R_t are adjacent. Hence, we use Compact Valiant only when R_s and R_t are not adjacent.

C. Adaptive Routing

In adaptive routing, the router into which a packet is first injected decides whether this packet should be routed over a minimal path or over a Valiant path. This decision is made on the basis of occupancy of *local* output buffers used in the respective paths, as well as the lengths of the considered paths. This routing algorithm is called Universal Globally-Adaptive Load-balancing (UGAL) [50].

For PolarFly, we also explore a UGAL variant which we call UGAL$_{\text{PF}}$. To achieve high bandwidth, UGAL$_{\text{PF}}$ reduces the average hops per packet by using:

- Compact Valiant described in Section VII-B, and
- Adaptation threshold – Valiant path is chosen over min-path only when fractional occupancy of the output buffer towards min-path is greater than a threshold ($\frac{2}{3}$ in our case).

Thus, UGAL$_{\text{PF}}$ offers a trade-off between adaptability of UGAL and low hop count of minimal static routing.

VIII. Performance Analysis

We now evaluate the latency and throughput of PolarFly.

A. Methodology and Comparison Targets

We compare PolarFly to Slim Fly [26] (as the most competitive diameter-2 network), Dragonfly [34] (as a popular recent choice when developing interconnects), Jellyfish [51] (as a random expander network) and 3-level fat tree [37] (as the most widespread existing interconnect baseline). Except fat tree, all topologies are direct. As numerous past works illustrate, networks such as torus, hypercube or Flattened Butterfly are less competitive in latency and bandwidth [26], [34], [52].

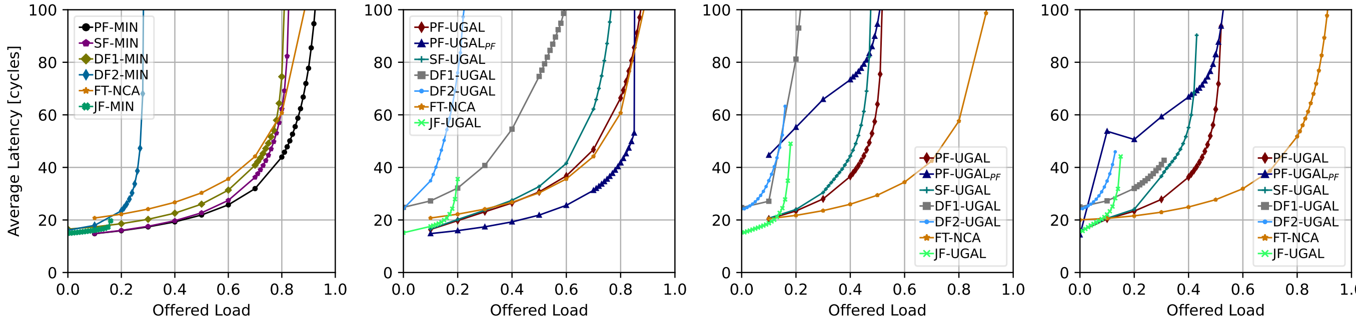
We use two variants of Dragonfly – (a) balanced Dragonfly (DF1), and (b) Dragonfly with radix and scale almost equivalent to PolarFly (DF2). Configurations of the baseline topologies are given in Table V.

Network	Parameters	Number of Routers	Network Radix
PolarFly (PF)	q=31, p=16	993	32
Slim Fly (SF)	q=23, p=18	1058	35
Balanced Dragonfly (DF1)	a=12, h=6, p=6	876	17
Equivalent Dragonfly (DF2)	a=6, h=27, p=10	978	978
Jellyfish (JF)	–	993	32
Fat Tree (FT)	n=3, k=18	972	36

TABLE V: Configuration of topologies used for simulations.

Following traffic patterns are simulated to effectively analyze the network behavior:

1) *Uniform* random traffic – for each packet, the source selects a destination uniformly at random (representing graph processing and distributed-memory graph algorithms, sparse linear algebra solvers, and adaptive mesh refinement methods [26], [53]–[58]).
2) *Tornado* traffic – endpoints on every router i send all traffic halfway across to endpoints on router $i + \frac{N}{2}$ modulo N.
3) *Random permutation* traffic – a fixed permutation mapping of source to destination is chosen uniformly at random from the set of all permutations. In PolarFly, Tornado and Random permutation traffic are adversarial for min-path routing because there is only one shortest path between any pair of routers.
4) Finally, two special permutation traffic patterns *Perm1Hop* and *Perm2Hop* are chosen to analyze UGAL$_{\text{PF}}$. In Perm1Hop, every router communicates with a 1-hop neighbor – the min-path length is 1-hop and valiant path length in UGAL$_{\text{PF}}$ is 4-hops. In Perm2Hop, every router communicates with a 2-hop neighbor – the min-path length is 2-hop and valiant path length in UGAL$_{\text{PF}}$ is 3-hops.



(a) Uniform traffic with min-path routing. (b) Uniform traffic with adaptive routing. (c) Random Permutation Traffic. (d) Tornado Permutation Traffic.

Fig. 8: Performance analysis (comparison with other topologies).

We use the established BookSim simulator [59] to conduct cycle-accurate simulations. Each router along with all of its endpoints in BookSim represents a single co-packaged node. To mimic co-packaged setting under permutation traffic, we enforce that all endpoints of a router send data to endpoints of only one other router. In other words, permutations are computed between routers, and not endpoints.

Packets of size 4 flits each are injected with a Bernoulli process. We use input-queued routers with 128 flit buffers per port and 4 virtual channels. In all simulations, we use a warm-up phase where no measurements are taken, to ensure that the simulator first reaches a steady-state.

B. Discussion of Results – Comparison against Baselines

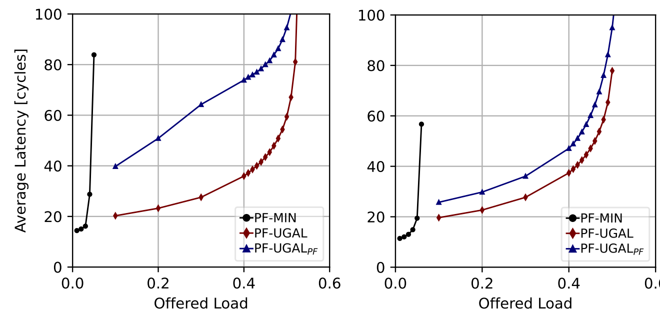
Figure 8 compares the performance of PolarFly (PF) and the topologies shown in Table V. The labels follow the scheme <*network*>-<*routing*>. The offered load in Figure 8 is normalized to the maximum capacity of each network.

For Permutation traffic, min-path routing in direct networks can achieve at most $\frac{1}{p}$ of peak throughput, because all p endpoints of a source router access the same path to the destination router. Hence, we only compare adaptive routing performance under permutation patterns in these topologies.

In general, we observe that PolarFly offers superior performance – for all traffic patterns, it outperforms all competitive direct topologies. Its advantages over Jellyfish and Dragonfly in terms of lower latency, are a direct consequence of its low diameter. Its benefits over Slim Fly in terms of higher saturation bandwidth, are due to careful design of routing protocols that exploit PolarFly structure to ensure that the routing decisions are as good as possible. Amongst the Dragonflies, the balanced DF1 outperforms DF2 whose throughput is bottlenecked by the traffic volume on intra-group links.

For the Uniform traffic, the adaptive routing based on Compact Valiant (UGAL$_{\text{PF}}$) exhibits latency and saturation throughput comparable to that of min-path routing, while significantly outperforming other adaptive algorithms and baseline topologies. Remarkably, the maximum throughput sustained by PolarFly for uniform traffic is comparable to the fat tree, with considerable reduction in latency.

For Random and Tornado Permutation traffic patterns, PolarFly is able to sustain up to 50% of the full injection bandwidth, using adaptive algorithms UGAL and UGAL$_{\text{PF}}$. The performance of these patterns is similar to Perm2Hop traffic shown in figure 9a, as min-path for most packets is 2-hops long. The total buffer space in the min-path is higher compared to Perm1Hop (Figure 9b, all 1-hop min-paths), rendering UGAL$_{\text{PF}}$ slower to adapt to congestion. Hence, UGAL$_{\text{PF}}$ has considerably higher latency than UGAL for Tornado, Random and Perm2Hop permutation patterns. UGAL has relatively higher entropy in terms of path selection, resulting in smaller queues inside routers and lower latency. Figure 9 also provides detailed insight into adversarial nature of permutation patterns for min-path routing in PolarFly. It can only withstand 5% of the full injection bandwidth, compared to almost 50% with adaptive routing.



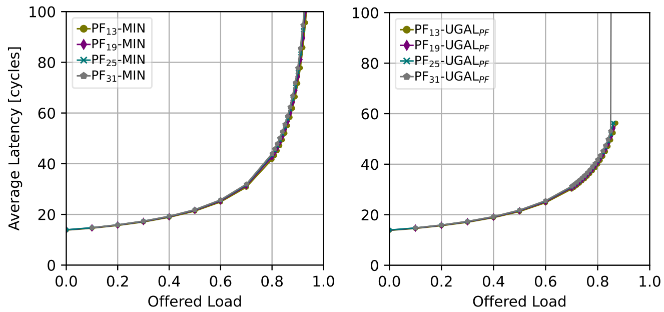
(a) Perm2Hop Permutation Traffic (b) Perm1Hop Permutation Traffic

Fig. 9: Performance of adaptive routing in PolarFly under Permutation Traffic

C. Discussion of Results - PolarFly Size

Next, we investigate the impact of PolarFly size on the performance by (a) varying q (radix), and (b) expanding network incrementally using the methods described in Section VI. We analyze balanced variants of PolarFly topology under uniform traffic i.e. the ratio of number of endpoints to network radix is maintained to $1:2$ in all experiments.

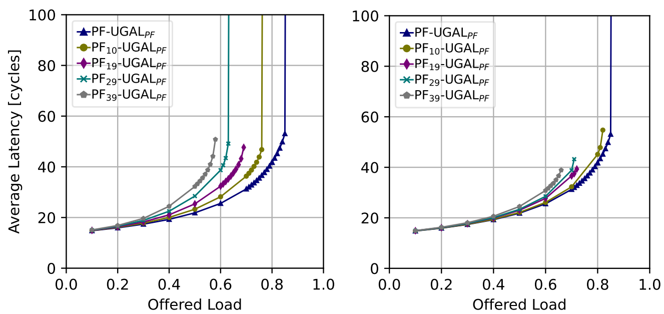
Figure 10 shows the latency and throughput for PolarFly for $q = 13, 19, 25$ and 31, which corresponds to $183, 381, 651$ and 993 routers, respectively. The labels follow the scheme <*network*>$_q$-<*routing*>. All PolarFlies provide similar saturation bandwidth and latency for both min-path and UGAL$_{\text{PF}}$ routing. This shows that PolarFly performance is stable with



(a) Min-path Routing (b) Adaptive Routing

Fig. 10: Performance Comparison of Polarfly of different sizes under uniform traffic

respect to the size of the network.



(a) Replicating the quadrics cluster. (b) Replicating non-quadrics clusters.

Fig. 11: Performance analysis of incrementally expanded PolarFly.

Figure 11 shows the latency and throughput of Polarfly incrementally expanded by adding 3, 6, 9 and 12 clusters by quadric or non-quadric cluster replication, which corresponds to approximately 10%, 19%, 29% and 39% increase in network size, respectively. The labels of incrementally expanded networks follow the scheme $<$*network*$>_{<\textit{size increase in percent}>}$-$<$*routing*$>$. We observe that 39% incremental growth in size using quadric replication results in a 31% drop in throughput. Comparatively, non-quadric replication creates only 19% drop in throughput for an equivalent increase in network size, thanks to its near-uniform degree distribution. Moreover, after the first replication, subsequent non-quadric replications have little impact on maximum throughput – 73% of peak bandwidth with 10% incremental growth vs 67% of peak bandwidth with 39% incremental growth.

IX. Structural Analysis

We compare bisection bandwidth and link failure resilience of PolarFly, against the topologies given in Table V.

A. Bisection bandwidth

Figure 12 shows the bisection bandwidth of compared topologies in terms of the fraction of edges in the bisection cut-set computed by METIS [60]. Fat Trees provide optimal bisection bandwidth with 50% edges lying in the cut-set. PolarFly closely approximates the optimal ratio, reaching it asymptotically. For network radix $\geq$ 18, PolarFly has more than 40% links crossing the bisection, even surpassing random expander networks such as Jellyfish [51]. This is not suprising since PolarFly topology expands extremely well, enforcing an almost Moore Bound spanning tree view from each vertex, whereas Jellyfish relies on random distribution of links and only achieves 50% of links in expectation for a random bisection. PolarFly has significantly higher bisection bandwidth compared to deterministic topologies SlimFly and Dragonfly, that have only 33% and 17% links in bisection.

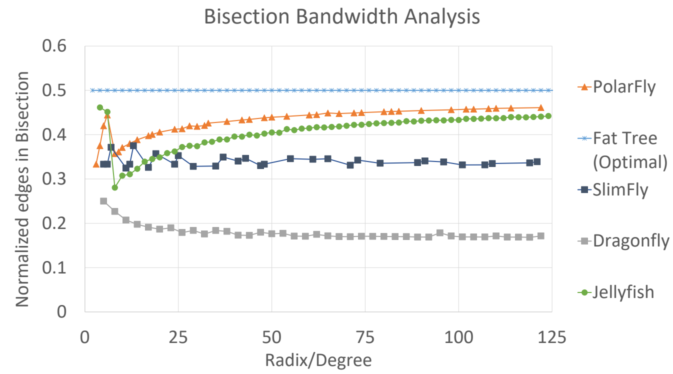


Fig. 12: Bisection bandwidth of different topologies shown by the number of links in the cut normalized to total links in the topology.

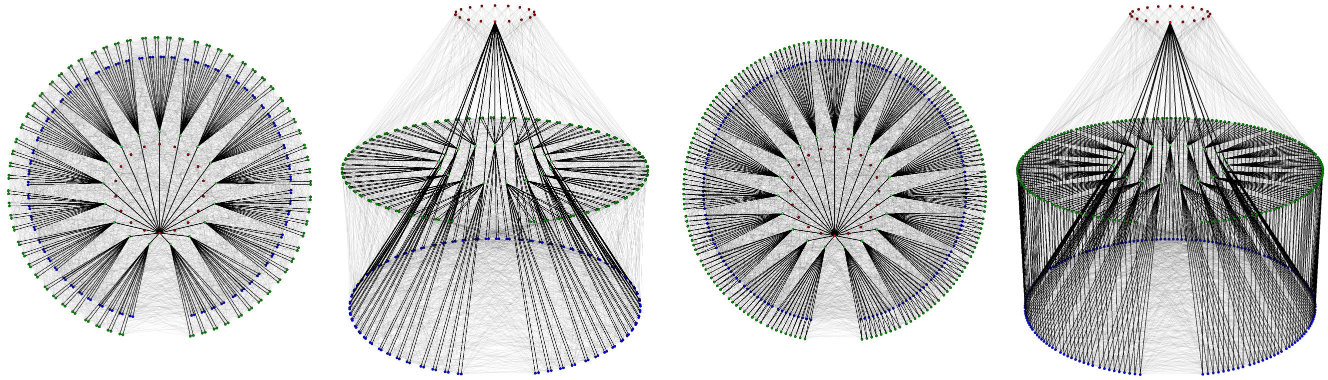
B. Fault Tolerance and Path Diversity

On the topology configurations given in Table V, we simulate 100 random link failures until network disconnection, and compute the median disconnection ratio.[1] We then randomly select a run with median disconnection ratio, and report its variation in network diameter and average shortest path length in Figure 14. We also analyze path diversity in PolarFly in Table VI to better understand its behavior under link failures.

Path length	Conditions	Number of paths
1	v, w adjacent	1
2	v, w adjacent and one of v, w quadric all other cases	0 1
3	v, w adjacent v, w not adjacent, x not quadric v, w not adjacent, x quadric	0 $q-1$ q
4	v, w adjacent and neither of v, w quadric v, w adjacent and one of v, w quadric v, w not adjacent and both of v, w quadric v, w not adjacent, $v, w \in V_1$, x not quadric v, w not adjacent, v quadric, $w \in V_1$ v, w not adjacent, $v, w \in V_1$, x quadric v, w not adjacent, $v \in V_1$, $w \in V_2$ v, w not adjacent, v quadric, $w \in V_2$ v, w not adjacent, $v \in V_2$, $w \in V_2$	$(q-1)^2$ q^2-q q^2-q q^2-4 q^2-3 q^2-2 q^2-2 q^2-1 q^2

TABLE VI: Path diversity in ER_q for small lengths: the number of paths of given lengths between arbitrary vertices v and w, and the conditions under which such paths exist. The vertex x is the unique intermediate vertex between v and w (if it exists). There are many different cases for path length 4; however, all are $\mathcal{O}(q^2)$. They are listed from smallest number of paths to largest.

[1]Mean and Standard Deviation statistics cannot be used because if any run disconnects at a particular failure ratio, its diameter becomes infinite.



(a) Overhead view of ER_{17}. (b) Side view of ER_{17}. (c) Overhead view of ER_{19}. (d) Side view of ER_{19}.

Fig. 13: Graphs for ER_{17} and ER_{19} are compared. The edges from the starting quadric to the clusters and the cluster edges are rendered in black. All other edges are rendered in light grey. Quadrics are in red, the centers are in light green, V_1 vertices are in green in the top layer, and V_2 vertices are in blue in the bottom layer. The triangle fan-outs internal to the clusters may be compared for $q = 17$ and $q = 19$: $17 \equiv 1 \mod 4$, and the triangle pairing of V_1 vertices with each other, and V_2 vertices with each other may be seen in subfigures 13a and 13b, with no vertical edges within a cluster joining the V_1 upper layer with the V_2 lower layer. Likewise, $19 \equiv 3 \mod 4$, so the triangle pairing of V_1 vertices with V_2 vertices are seen in subfigures 13c and 13d, in this case with vertical edges within a cluster joining the V_1 upper layer with the V_2 lower layer.

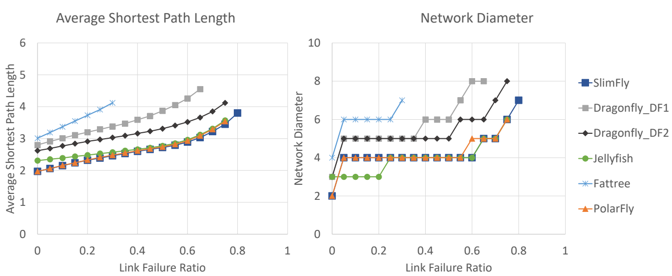


Fig. 14: Resilience properties of topologies as a function of the fraction of failed links.

Jellyfish, being a random expander, is highly resilient to link failures – random failures in Jellyfish result in just another random graph. PolarFly and SlimFly exhibit similar resilience, with higher disconnection ratio than both Fat tree and balanced Dragonfly DF1. They are both expanders and have comparable resilience to Jellyfish. However, being diameter-2 networks with close to Moore bound scalability, the diameter of PolarFly and SlimFly increases more rapidly than Jellyfish. Compared to PolarFly, SlimFly has slightly more redundancy in minimal paths, resulting in marginally higher disconnection ratio, even though it reduces scalability.

If a single link fails, the diameter of PolarFly increases to 3, or 4 if the link is from a quadric. Table VI shows that there are no 2- or 3-hop paths between quadrics and the adjacent vertices, which intuitively explains why PolarFly diameter increases to 4 with only 5% link failure, as in Figure 14. However, PolarFly has a great deal of path diversity for path length 4, so its diameter stays at 4 even when 55% links fail.

If a node x fails, PolarFly diameter would increase from 2 to 3, as the 2-hop minimal paths between neighbors of x would be lost. However, for any neighbor v of x, the neighbors of v have 1-hop or 2-hop paths to other neighbors of x, that do not pass through x. Hence, despite x failing, v can still reach other nodes within 3-hops.

X. Cost Analysis

We now analyze the cost of the network topologies under iso-injection bandwidth constraints. We fill focus on a specific case which is reflective of the latest technological developments: co-packaged Optical IO (OIO) [16]. The primary cost indicator is the total number of optical IO ports: each port requires an OIO module, a laser, a connector and cables. Technological constraints limit to the number of OIO modules that can be co-packaged in a die due to shoreline limitations: the state of the art is 4 to 6 OIO modules per die, with 8 links per module. We consider configurations with approximately 1,024 nodes, with each topology having the same injection bandwidth. Given that not all the constructions have exactly that number of nodes, we normalize the number of links to a network configuration with 1,024 nodes. In addition, we also consider the achievable injection performance and we normalize the achievable performance, under two distinct scenarios: uniform and permutation traffic. While most networks reach comparable saturation points with uniform traffic, typically around 90%, fat trees are almost insensitive to the type of permutation while direct topologies must resort to some type of misrouting, bringing their saturation points down to approximately 50%. Both Polarfly and Slim Fly use 4 OIO modules with 32 links per node, while Dragonfly 6 OIO modules with 48 links. Fat trees use switches with 4 OIO modules and 32 links, and each of the 1,024 nodes has 2 OIOs with 16 injection links. Figure 15 shows the relative costs to PolarFly under the two traffic patterns. Slim Fly has a slight cost increase of about 20%, reflective of the lower fraction of Moore's bound, while the Dragonfly is a diameter 3 network, so the ratio injection bandwidth to overall bandwidth is 1:3 vs 1:2 for PolarFly and Slim Fly. Due to packaging limitations, fat tree switches can only connect two input nodes with 16 links each, resulting in a rather deep 10-level construction of 512 switches per level, and 256 switches in the top level. PolarFly compares very favorably to fat trees with a 5.19X cost reduction under uniform traffic and 2.68X under permutation traffic.

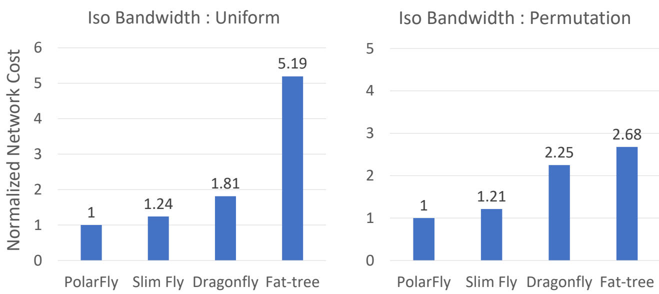


Fig. 15: Cost per node under different topologies normalized to 1,024 nodes.

XI. Related Work

Network topologies considered in this paper are described in detail in Section II, more details are also provided in a recent survey [61]. Early works into novel topologies with diameter lower than that of 3-stage Fat trees [37] include Flattened Butterfly [52] and its generalization called HyperX [35], and the Dragonfly topology [34], [62]. These designs mainly aimed at facilitating the physical layout of networks. Lowering the diameter of a network in order to reduce cost and power consumption while maintaining high performance have been introduced in the Slim Fly class of interconnects [26], [63]. Since then, several other designs followed, including Xpander [9], Megafly [64], Bundlefly [47] or Galaxyfly [65]. However, they do not focus on diameter-2 and thus none of them improves upon key properties such as latency, cost, or power consumption. PolarFly extends this line of work by exploiting a family of graphs that is asymptotically optimal with respect to the Moore Bound, allowing close to optimal scalability. It simultaneously offers superior cost, power consumption, and performance. Moreover, it specifically targets the recent developments into copackaged optics, something not addressed so far in the literature for scalable network design.

Routing in low-diameter networks has also been a subject of research, especially in recent years. For example, the FatPaths [66] routing architecture, enables adaptive multipathing in data-center and HPC clusters in low-diameter networks, focusing on Slim Fly. However, none of these works is particularly well suited for the unique structure of PolarFly in which some routers form intra-connected clusters while a single cluster of quadric routers forms an independent set. We address this with a novel adaptive UGAL routing protocol suited for PolarFly.

The mathematical foundations of the Erdős-Rényi polarity graphs (ER) are embedded in projective geometry and were laid down in mid-20$^{\text{th}}$ century. Singer [67] first formulated perfect difference sets – a numerical structure that encodes the incidence between lines and points of projective planes. Erdős and Rényi [32] discovered the polarity quotient graph of this incidence structure, which forms the basis of PolarFly. Indpendently, Brown [33] also constructed the same graph using orthogonality relationship of points in projective planes.

Building on these foundations, some prior works have proposed an ER graph topology for high performance interconnection networks. Parhami et al. [68] use perfect difference sets to construct the bipartite network of same degree, diameter and order as the incidence graph described in section IV-E1. Brahme et al. [69] rediscover the ER graphs by defining a symmetric adjacency equation on the perfect difference sets. They also compare the performance of certain communication primitives on this topology and the Clos network. Camarero et al. [70] use the polarity-map based construction of ER graphs (section IV-E2) and compare the cost of conventional networks with various topologies.

To the best of our knowledge, PolarFly is the first work to comprehensively analyze networking properties of ER graphs, covering several aspects beyond the prior attempts [69], [70], including a comparison of feasible radixes, performance for various traffic patterns, bisection width, resilience, and network cost under a co-packaged and iso-bandwidth setting. We also develop a novel modular layout, incremental expansion strategies, and routing schemes to exploit non-minimal path diversity, all of which utilize new mathematical properties of the ER graphs that are presented for the first time in this paper. In this way, our work extends the feasibility of ER graphs as network topologies, well beyond the existing literature.

XII. Conclusion

In this paper, we propose PolarFly, a diameter-2 network that asymptotically reaches the Moore upper bound on the number of nodes for a given degree and diameter. PolarFly improves upon Slim Fly, being more performant, scalable and cost-effective by up to 10%. Importantly, PolarFly is flexible (it offers a wide range of feasible designs using manufacturable routers), modular (its structure can be decomposed into groups), and expandable (one can incrementally increase its size without much performance loss). We expect that PolarFly will become the enabler for more energy-efficient interconnects in the next-generation era of co-packaged devices.

Acknowledgment

The authors would like to thank Guillermo Pineda-Villavicencio for many insightful discussions on the nature of graphs approaching the Moore bound.

References

[1] J. Dongarra, "Report on the Fujitsu Fugaku System," University of Tennessee, Knoxville, Tech. Rep. ICL-UT-20-06, June 2020.

[2] H. Meuer, E. Strohmaier, J. Dongarra, H. Simon, and M. Meuer, "Top 500: The List," https://top500.org/lists/top500/, Top500.org, 2021.

[3] "Sierra," https://hpc.llnl.gov/hardware/compute-platforms/sierra, Lawrence Livermore National Laboratory, 2018.

[4] C. Zimmer, S. Atchley, R. Pankajakshan, B. E. Smith, I. Karlin, M. L. Leininger, A. Bertsch, B. S. Ryujin, J. Burmark, A. Walker-Loud *et al.*, "An Evaluation of the CORAL Interconnects," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019, pp. 1–18.

[5] D. Chen, N. Eisley, P. Heidelberger, S. Kumar, A. Mamidala, F. Petrini, R. Senger, Y. Sugawara, R. Walkup, B. Steinmacher-Burow *et al.*, "Looking under the hood of the ibm blue gene/q network," in *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE, 2012, pp. 1–12.

[6] "Introducing Meta's Next-Gen AI Supercomputer," https://about.fb.com/news/2022/01/introducing-metas-next-gen-ai-supercomputer/, Meta, 2022.
[7] S. Rajbhandari, O. Ruwase, J. Rasley, S. Smith, and Y. He, "Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14.
[8] B. J. Gutiérrez, N. McNeal, C. Washington, Y. Chen, L. Li, H. Sun, and Y. Su, "Thinking about gpt-3 in-context learning for biomedical ie? think again," *arXiv preprint arXiv:2203.08410*, 2022.
[9] 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*, 2016, pp. 205–219.
[10] S. Aksoy, S. Young, J. Firoz, R. Gioiosa, M. Raugas, and J. Escobedo, "Spectralfly: Ramanujan graphs as flexible and efficient interconnection networks," *arXiv preprint arXiv:2104.11725*, 2021.
[11] 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.
[12] 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.
[13] J. Dean and L. A. Barroso, "The Tail at Scale," *Communications of the ACM*, vol. 56, pp. 74–80, 2013. [Online]. Available: http://cacm.acm.org/magazines/2013/2/160173-the-tail-at-scale/fulltext
[14] N. Abrams, Q. Cheng, M. Glick, M. A. Jezzini, P. E. Morrissey, P. O'Brien, and K. Bergman, "Silicon Photonic 2.5 D Multi-Chip Module Transceiver for High-Performance Data Centers," *IEEE/OSA Journal of Lightwave Technology*, January 2020.
[15] S. Daudlin, A. Rizzo, N. C. Abrams, S. Lee, D. Khilwani, V. Murthy, J. Robinson, T. Collier, A. Molnar, and K. Bergman, "3D-Integrated Multichip Module Transceiver for Terabit-Scale DWDM Interconnects," in *Optical Fiber Communications, OFC 2021*, June 2021.
[16] M. Wade, E. Anderson, S. Ardalan, P. Bhargava, S. Buchbinder, M. Davenport, J. Fini, H. Lu, C. Li, and R. Meade, "TeraPHY: a Chiplet Technology for Low-Power, High-Bandwidth In-Package optical I/O," *IEEE Micro*, vol. 40, no. 2, pp. 63–71, 2020.
[17] G. Keeler, "ERI Programs Panel - Phase II Overview," dARPA ERI Summit 2019.
[18] Avicena Tech, https://avicena.tech/.
[19] Teramount, https://teramount.com/.
[20] Lightmatter, https://lightmatter.co/.
[21] DustPhotonics, https://dustphotonics.com/.
[22] Celestial AI, https://www.celestial.ai/.
[23] Axalume, https://www.axalume.com/.
[24] P. Maniotis, L. Schares, B. G. Lee, M. A. Taubenblatt, and D. M. Kuchta, "Scaling hpc networks with co-packaged optics," in *Optical Fiber Communication Conference*. Optical Society of America, 2020, pp. T3K–7.
[25] R. Mahajan, R. Sankman, N. Patel, D.-W. Kim, K. Aygun, Z. Qian, Y. Mekonnen, I. Salama, S. Sharan, D. Iyengar, and D. Mallik, "Embedded Multi-Die Interconnect Bridge (EMIB) – A High Density, High Bandwidth Packaging Interconnect," in *2016 IEEE 66th Electronic Components and Technology Conference*, May 2016.
[26] 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.
[27] E. Loz, H. Pérez-Rosés, and G. Pineda-Villavicencio. (2010) The degree-diameter problem for general graphs. [Online]. Available: http://www.combinatoricswiki.org/wiki/The_Degree_Diameter_Problem_for_General_Graphs,CombinatoricsWiki
[28] M. Miller and J. Širáň, "Moore graphs and beyond: A survey of the degree/diameter problem," *Electronic Journal of Combinatorics, Dynamic survey*, vol. 14, pp. 1–61, 12 2005.
[29] A. J. Hoffman and R. R. Singleton, "On Moore graphs with diameters 2 and 3," *IBM Journal of Research and Development*, vol. 4, no. 5, pp. 497–504, 1960.
[30] E. Bannai and T. Ito, "On finite Moore graphs," *Journal of the Faculty of Science, the University of Tokyo. Sect. 1 A, Mathematics*, vol. 20, pp. 191–208, 1973.
[31] R. M. Damerell, "On Moore graphs," *Proc. Camb. Phil. Soc.*, vol. 74, pp. 227–236, 1973.
[32] P. Erdős and A. Rényi, "On a problem in the theory of graphs," *Publ. Math. Inst. Hungar. Acad. Sci.*, vol. 7A, pp. 623–641, 1962.
[33] W. G. Brown, "On graphs that do not contain a Thomsen graph," *Canadian Mathematical Bulletin*, vol. 9, no. 3, p. 281–285, 1966.
[34] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable Dragonfly topology," in *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ser. ISCA '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 77–88.
[35] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, "Hyperx: topology, routing, and packaging of efficient large-scale networks," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, 2009, pp. 1–11.
[36] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *Proceedings of the 34th annual international symposium on Computer architecture*, 2007, pp. 126–137.
[37] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Trans. Comput.*, vol. 34, no. 10, pp. 892–901, Oct. 1985.
[38] G. Kathareios, C. Minkenberg, B. Prisacari, G. Rodriguez, and T. Hoefler, "Cost-effective diameter-two topologies: Analysis and evaluation," in *SC'15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2015, pp. 1–11.
[39] S. Aananthakrishnan, N. K. Ahmed, V. Cave, M. Cintra, Y. Demir, K. D. Bois, S. Eyerman, J. B. Fryman, I. Ganev, W. Heirman *et al.*, "Piuma: programmable integrated unified memory architecture," *arXiv preprint arXiv:2010.06277*, 2020.
[40] Z. Guo, Y. Shan, X. Luo, Y. Huang, and Y. Zhang, "Clio: a hardware-software co-designed disaggregated memory system," in *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2022, pp. 417–433.
[41] M. Bachratý and J. Širáň, "Polarity graphs revisited," *Ars Mathematica Contemporanea*, vol. 8, pp. 55–67, 01 2015.
[42] M. Miller and J. Širán, "Moore graphs and beyond: A survey of the degree/diameter problem," *Electronic Journal of Combinatorics*, vol. 61, pp. 1–63, 2005.
[43] T. D. Parsons, "Graphs from projective planes," *Aequat. Math.*, vol. 14, p. 167–189, 02 1976.
[44] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*. Boston, MA: Springer, 1987.
[45] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, 2nd ed. Cambridge University Press, 1994.
[46] J. A. Gallian, *Contemporary Abstract Algebra*. New York, NY: Chapman and Hall / CRC, 2020.
[47] F. Lei, D. Dong, X.-K. Liao, and J. Duato, "Bundlefly: a low-diameter topology for multicore fiber," in *Proceedings of the 2020 International Conference on Supercomputing*, 06 2020, pp. 1–11.
[48] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, 2nd ed. Cambridge University Press, 2001.
[49] L. Valiant, "A scheme for fast parallel communication," *SIAM journal on computing*, vol. 11, no. 2, pp. 350–361, 1982.
[50] A. Singh, "Load-Balanced Routing in Interconnection Networks," Ph.D. dissertation, Stanford University, 2005.
[51] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: networking data centers randomly," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 17–17.
[52] J. Kim, W. J. Dally, and D. Abts, "Flattened Butterfly: A cost-efficient topology for high-radix networks," in *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ser. ISCA '07. New York, NY, USA: ACM, 2007, pp. 126–137.
[53] X. Yuan, S. Mahapatra, W. Nienaber, S. Pakin, and M. Lang, "A new routing scheme for Jellyfish and its performance with HPC workloads," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2013, pp. 1–11.
[54] M. Besta, M. Podstawski, L. Groner, E. Solomonik, and T. Hoefler, "To push or to pull: On reducing communication and synchronization in graph computations," in *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing*. ACM, 2017, pp. 93–104.
[55] M. Besta, R. Kanakagiri, H. Mustafa, M. Karasikov, G. Rätsch, T. Hoefler, and E. Solomonik, "Communication-efficient jaccard similarity for high-performance distributed genome comparisons," in *2020 IEEE*

International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2020, pp. 1122–1132.

[56] S. Sakr, A. Bonifati, H. Voigt, A. Iosup, K. Ammar, R. Angles, W. Aref, M. Arenas, M. Besta, P. A. Boncz *et al.*, "The future is big graphs: a community view on graph processing systems," *Communications of the ACM*, vol. 64, no. 9, pp. 62–71, 2021.

[57] L. Gianinazzi, P. Kalvoda, A. De Palma, M. Besta, and T. Hoefler, "Communication-avoiding parallel minimum cuts and connected components," *ACM SIGPLAN Notices*, vol. 53, no. 1, pp. 219–232, 2018.

[58] M. Besta and T. Hoefler, "Accelerating irregular computations with hardware transactional memory and active messages," in *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing*, 2015, pp. 161–172.

[59] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. E. Shaw, J. Kim, and W. J. Dally, "A detailed and flexible cycle-accurate network-on-chip simulator," in *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 86–96.

[60] G. Karypis and V. Kumar, "MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0," http://www.cs.umn.edu/~metis, University of Minnesota, Minneapolis, MN, 2009.

[61] M. Besta, J. Domke, M. Schneider, M. Konieczny, S. Di Girolamo, T. Schneider, A. Singla, and T. Hoefler, "High-performance routing with multipathing and path diversity in ethernet and hpc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 4, pp. 943–959, 2020.

[62] B. Arimilli *et al.*, "The PERCS High-Performance Interconnect," in *Proceedings of the 2010 18th IEEE Symposium on High Performance Interconnects*, ser. HOTI '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 75–82.

[63] M. Besta, S. M. Hassan, S. Yalamanchili, R. Ausavarungnirun, O. Mutlu, and T. Hoefler, "Slim noc: A low-diameter on-chip network topology for high energy efficiency and scalability," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 43–55, 2018.

[64] M. Flajslik, E. Borch, and M. A. Parker, "Megafly: A topology for exascale systems," in *International Conference on High Performance Computing*. Springer, 2018, pp. 289–310.

[65] F. Lei, D. Dong, X. Liao, X. Su, and C. Li, "Galaxyfly: A novel family of flexible-radix low-diameter topologies for large-scales interconnection networks," in *Proceedings of the 2016 International Conference on Supercomputing*, 2016, pp. 1–12.

[66] M. Besta, M. Schneider, M. Konieczny, K. Cynk, E. Henriksson, S. Di Girolamo, A. Singla, and T. Hoefler, "Fatpaths: Routing in supercomputers and data centers when shortest paths fall short," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020, pp. 1–18.

[67] J. Singer, "A theorem in finite projective geometry and some applications to number theory," *Transactions of the American Mathematical Society*, vol. 43, no. 3, pp. 377–385, 1938.

[68] B. Parhami and M. Rakov, "Perfect difference networks and related interconnection structures for parallel and distributed systems," *IEEE transactions on parallel and distributed systems*, vol. 16, no. 8, pp. 714–724, 2005.

[69] D. Brahme, O. Bhardwaj, and V. Chaudhary, "Symsig: A low latency interconnection topology for hpc clusters," in *20th Annual International Conference on High Performance Computing*. IEEE, 2013, pp. 462–471.

[70] C. Camarero, C. Martínez, E. Vallejo, and R. Beivide, "Projective networks: Topologies for large parallel computer systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 2003–2016, 2017.