The Journal of Supercomputing (2024) 80:18062–18080
https://doi.org/10.1007/s11227-024-06146-1

Ant Mill: an adversarial traffic pattern for low-diameter direct networks

Cristóbal Camarero[1] · Carmen Martínez[1] · Ramón Beivide[1,2]

Accepted: 14 April 2024 / Published online: 10 May 2024

Abstract

Since today's HPC and data center systems can comprise hundreds of thousands of servers and beyond, it is crucial to equip them with a network that provides high performance. New topologies proposed to achieve such performance need to be evaluated under different traffic conditions, aiming to closely replicate real-world scenarios. While most optimizations should be guided by common traffic patterns, it is essential to ensure that no pathological traffic pattern can compromise the entire system. Determining synthetic adversarial traffic patterns for a network typically relies on a thorough understanding of its topology and routing. In this paper, we address the problem of identifying a generic adversarial traffic pattern for low-diameter direct interconnection networks. We first focus on Random Regular Graphs (RRGs), which represent a typical case for these networks. Moreover, RRGs have been proposed as topologies for interconnection networks due to their superior scalability and expandability, among other advantages. We introduce Ant Mill, an adversarial traffic pattern for RRGs when using routes of minimal length. Secondly, we demonstrate that the Ant Mill traffic pattern is also adversarial in other low-diameter direct interconnection networks such as Slimfly, Dragonfly, and Projective networks. Ant Mill is thoroughly motivated and evaluated, enabling future studies of low-diameter direct interconnection networks to leverage its findings.

Keywords Data center · Interconnection network · Random regular graphs · Adversarial traffic pattern

✉ Cristóbal Camarero
cristobal.camarero@unican.es

Carmen Martínez
carmen.martinez@unican.es

Ramón Beivide
ramon.beivide@unican.es

[1] Department Computer Science and Electronics, Universidad de Cantabria, Santander, Spain

[2] Barcelona Supercomputing Center, Barcelona, Spain

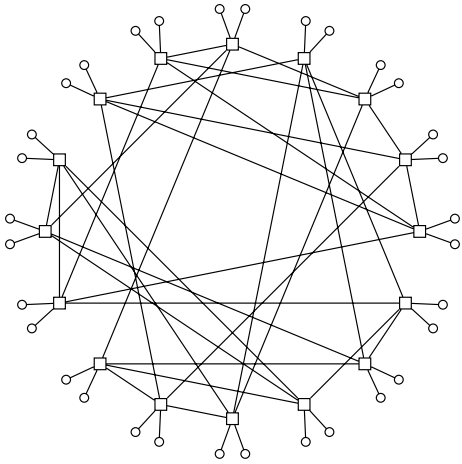

Fig. 1 A direct radix-6 Random Regular Network with 16 switches and 32 servers

1 Introduction

Contemporary supercomputers and data centers can consist of hundreds of thousands of servers, highlighting the critical importance of the interconnection network. Currently, a variety of topologies are deployed in the interconnection networks of large-scale systems. Notably, the Top 500 HPC ranking [27] showcases systems employing diverse architectures, including indirect Folded-Clos networks, as well as direct networks such as Dragonflies and Flattened Butterflies [14, 23, 24]. The rapid pace of technological advancement necessitates periodic consideration of novel topological designs. In previous years, toroidal topologies, exemplified by the BlueGene family [19], dominated the landscape of direct interconnection networks. Despite the continued use of TOFU in systems such as the Fujitsu Fugaku [2], which employs a 5D torus architecture similar to BlueGene's, there has been a noticeable shift toward high-degree networks in contemporary systems.

Among high-degree networks, we find that networks based on Random Regular Graphs (RRGs) are particularly attractive for our study. These networks have been proposed both for data centers [35] and for supercomputers [25]. In Fig. 1, a network based on a RRG can be seen. In the figure, squares represent switches, and circles represent servers. It is noteworthy that every switch is randomly interconnected to the same number of switches (the degree of the graph) and the same number of servers.

It is well-established that data center applications and user behavior are in a constant state of flux [26]. This renders it inappropriate to design a system focused exclusively on a few workloads; instead, it must be designed to cope with new scenarios. Hence, it is critical to characterize those traffic patterns that strain the network, known as *adverse traffic patterns*. Typically, adverse traffic patterns can be identified after a thorough analysis of the topological structure of the interconnection network, which complicates the determination of such traffic patterns. Therefore, in this paper, we identify a traffic pattern that constitutes an adverse situation for low-diameter topologies.

An RRG may contain any substructure, rendering it impossible to achieve a complete understanding of the topology, thus constituting an inherent. In [21], longest matchings are identified as an adverse traffic pattern in RRGs under ideal routing assumptions. In many networks, such matchings correspond to data permutations between switches where the destination is at distance equal to the diameter of the network. Contrary to this, our findings reveal the existence of permutation-based traffics that are much more adverse than longest matchings.

We have termed this adverse traffic pattern as *Ant Mill* due to its resemblance to a phenomenon observed in nature, occasionally performed by ants [32]. In this phenomenon, a group of army ants becomes separated from the main group by losing track of pheromone trails, and they begin to follow each other, forming a continuously rotating circle, eventually succumbing to exhaustion. Similarly, in this new traffic pattern, packets within the network follow each other in a cycle, leading to a complete degradation of throughput if left unaddressed. Importantly, while theoretical results may consider longest matchings as the worst-case scenario, our traffic pattern demonstrates that communication between closer switches can present an even more challenging situation to overcome in a random network. As will be demonstrated, this traffic pattern reduces throughput by at least 88% compared to that offered by a uniform traffic pattern when minimal routes are utilized in all the RRGs evaluated. Although Ant Mill is conceptualized as a theoretical construct to present a general adverse case, it can manifest in day-to-day communication patterns within interconnection networks, as will be discussed later.

The *Ant Mill* traffic pattern not only facilitates the generation of adverse scenarios in networks based on random graphs but also establishes a general criterion for consistently achieving comparable outcomes in other types of networks such as Dragonfly [24], Slimfly [5], and Projective networks [13]. All these networks employ low-diameter direct topologies designed to interconnect a substantial number of servers.

The paper is organized as follows: In Sect. 2, essential technical background on graphs and networks is provided. Section 3 reviews some adversarial traffic patterns for self-containment. Section 4 defines the Ant Mill traffic pattern, which, as will be described, is based on a Hamiltonian cycle embedded in the topology of the interconnection network. Section 5 explains the construction of the Hamiltonian cycle. Section 6 discusses the results of the experimentation. Finally, in Sect. 7, a brief discussion concludes the paper.

2 Background

In this section, we establish some concepts necessary for the development of the rest of the paper. First, a network comprises servers that are attached to switches, which are interconnected. The associated *topology* dictates this interconnection. The topology is mathematically modeled by a *graph* $G(V, E)$, where the set of vertices V represents the switches and the set of edges E represents the links connecting any pair of switches. We will assume that each switch has the same radix (number of ports), which is split into the servers connected to it and its adjacent switches. The

number of adjacent switches defines the *degree d* of the associated graph. The *routing mechanism* determines the path that is used to communicate any pair of servers.

The *distance* between two switches x_i and x_j is the distance in the graph, written $D(x_i, x_j)$, and defined as the number of edges in a minimum path between them. The *radius* and *diameter* of the graph are, respectively, the minimum and maximum values of the eccentricity, which is the longest distance from a switch to any other switch. The diameter of a graph G is written as $D(G)$, and its average distance is represented by $\bar{D}$.

There are various topologies employed in computer systems, each exhibiting distinct distance properties. Simple topologies like meshes and tori have diameters that increase with their size. Specifically, adding a new row to a mesh increases the diameter by 1. While tori are still utilized in some supercomputers [2], there is a prevailing trend aimed at reducing the cost and latency of modern systems by minimizing the diameter. *Low-diameter topologies* are structured to accommodate an increasing number of servers by augmenting the degree rather than the diameter. In practice, the router radix can be predetermined to establish the maximum size to which the system can expand. In some instances, routers can be replaced with others featuring a greater radix, although this process more closely resembles network migration than a simple upgrade. The least practicable diameter, which is 2, has garnered attention with several proposals available in the literature, see for example [5, 9, 13, 22, 38]. For diameter 3, the Dragonfly topology is notable, famously employed in the Frontier Supercomputer [4]. Practicable instances of Fat-Trees and Random Regular Graphs (RRGs) are also low-diameter networks.

In network communications, a traffic pattern is defined by the potential destinations of each server. We focus on traffic patterns where all servers generate and consume equal amounts of load, referred to as *admissible traffic*. In the traffic patterns defined later, no server sends traffic to another attached to the same switch. Collectively, the traffic pattern, topology, and routing establish the network's performance in a simplified sense. We take maximum throughput as primary performance metric, representing the maximum total load effectively accepted, normalized to the ideal total capacity of the servers. Within this framework, we theoretically introduce the concept of an adverse traffic pattern.

When dealing with adverse traffic patterns, a common approach is to replace the use of minimal routes with Valiant routes [39], which provide half of the throughput provided by minimal routing in uniform patterns. Essentially, a Valiant route involves routing from the source to a randomly chosen intermediate point, followed by a minimal route from that intermediate to the destination. In many topologies, including those termed low-diameter networks, maximum throughput is achieved with uniform traffic patterns and minimal routing. Consequently, in these topologies, Valiant routing yields at least half of the optimal throughput [15]. Any traffic pattern for which minimal routing falls short of this threshold would benefit from employing Valiant routing, thus motivating the following definition.

Definition 1 A traffic pattern P is called *adverse* for a topology T if the maximum throughput obtained by minimal routing is θ, and $2\theta < \theta_U$, where θ_U is the maximum throughput for the uniform traffic pattern obtained by minimal routing.

Recognizing these adverse patterns is crucial in the developing of routing algorithms, as Valiant routing serves as a clear baseline for comparison. In Sect. 4, we introduce Ant Mill, an adverse traffic pattern that can be systematically constructed for many topologies. Moreover, even for routings that employ a large diversity of paths, Ant Mill continues to be an important challenge for latency.

3 Related work

While tori are beyond the scope of the networks considered in this paper, the *tornado* traffic pattern [33, 36] serves as a classical reference of adversarial traffic pattern in such networks. The tornado pattern was designed as a worst-case for which minimal routing provides only a quarter of the throughput compared to a uniform pattern.

In [24], Dragonflies were introduced along with an adverse traffic pattern. This pattern is based on the servers in a group sending messages to servers in the following group, causing minimal routes to utilize the single link between the two groups. Subsequently, in [16], this idea was improved to obtain an adverse pattern that additionally presented problems for a particular Valiant scheme.

In [5], Slimfly was defined as a diameter-2 topology, and an adverse traffic pattern was also provided. Let us explain it here by considering a pair of adjacent switches x and y. The objective is to provide maximum load over the link x–y. This is achieved by selecting p servers among the neighbors of x that are at a distance 2 from y. These servers generate traffic toward the servers at y. Similarly, the p servers at x generate traffic toward some neighbors of y that are at a distance 2 from x. Since enough paths of length 2 in the Slimfly are unique, an adversarial selection makes $2p$ flows go through the x–y link.

Later, in [13], it was observed that the same concept as in [5] can be replicated in Projective networks, which are also of diameter 2. Additionally, the authors give the idea that in Generalized Moore Graphs [31] of diameter $D(G)$, the paths of length $D(G) - 1$ are unique, allowing for more adversarial patterns. It is important to note that in the previous disquisition, the behavior of the network at a global level is not taken into account, as the perspective of a unique link is considered. In the following, we provide an admissible traffic pattern in which all the used links are congested in this manner.

Adverse traffic patterns for topologies based on RRGs have been hardly explored. In [34], sending traffic to the diameter distance is regarded as the worst-case traffic pattern, when assuming ideal routing. In this paper, we adopt a more realistic perspective and identify an adversarial traffic pattern for RRGs, primarily focusing on minimal routing while also considering other practicable and implementable routing mechanisms. It is worth noting that our findings reveal significantly different outcomes for combinations of routings and traffic patterns.

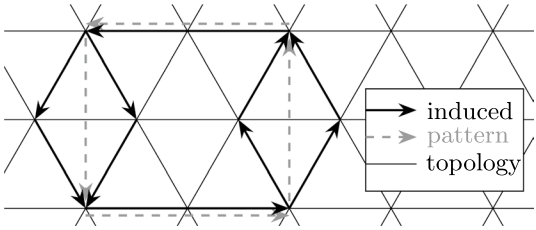

Fig. 2 Subgraph of a 6-regular mesh induced by minimal routing of a pattern

4 Ant Mill traffic pattern

In this section, we explore the definition of an adverse class of traffic patterns for low-diameter direct interconnection networks when using minimal routing.

Any traffic pattern induces a subgraph in the topology, which consists of the switches that communicate and the links used for that communication. To illustrate this concept, let us refer to Fig. 2 as an example, depicting a portion of a 6-regular mesh. In this figure, the dashed arrows represent the traffic pattern, indicating that all communications occur between switches at distance 2 from each other. Solid arrows highlight the minimal path employed for this traffic pattern. These minimal paths are unique in the case of horizontal arrows and non-unique in the vertical ones.

Hence, to define an adverse traffic pattern, the following two principles are employed:

1. Minimize the number of links in the directed subgraph induced by the traffic pattern, thus limiting the use of the remaining links in the topology.
2. Maximize the communication distances to increase the load over the induced subgraph.

To minimize the number of links in the induced subgraph, each server in a switch must communicate exclusively with servers in a single different switch. Since the traffic is supposed to be admissible, all switches must be included, and therefore, the average out-degree of the induced subgraph must be reduced as much as possible. Indeed, it can be reduced to exactly 1 by taking any degree-2 subgraph—a collection of cycles, also known as a 2-factor—and choosing an orientation for the links in each cycle. Then, if each server sends traffic to a server in the switch immediately next in the cycle, these cycles form the subgraph induced by shortest paths, leaving the rest of network unused, as required by the first principle.

It is worth noting that any permutation which maps a switch to a neighbor switch would achieve a similar effect. However, considering the second principle, communications within the cycle must be arranged so that that the distances in the induced subgraph are maximized without introducing additional paths. This effectively multiplies the load by such distance without increasing the number of links used by minimal routing. To maintain simplicity, we build a unique directed cycle going through all switches, and this is a *Hamiltonian cycle*, where every server sends traffic to a server at distance λ in the cycle direction. An example of such Hamiltonian cycle over a RRG is illustrated in Fig. 3, decorated with ants going around the cycle.

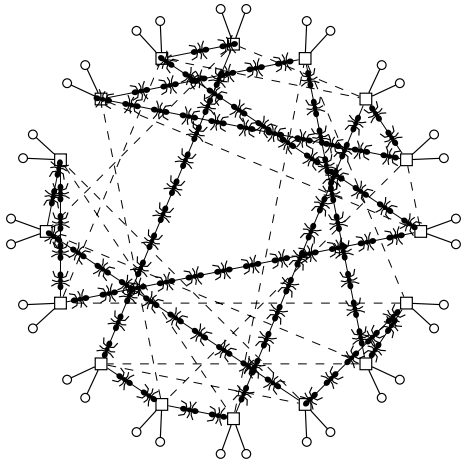

Fig. 3 A Hamiltonian cycle over a radix-6 Random Regular Network with 16 routers and 32 servers. The cycle is decorated with ants, which follow the Ant Mill pattern

The next definition establishes the communication pattern that fulfills these aforementioned principles.

Definition 2 Let us consider a topology T modeled by a graph $G(V, E)$. Let $H = x_0, x_1, \ldots, x_{n-1}$, where $n = |V|$, be a Hamiltonian cycle embedded in G. Then, the (H, λ)-*Ant-mill traffic pattern* establishes that every server attached to switch x_i sends traffic to a server attached to switch $x_{i+\lambda}$, with all index operations performed modulo the number of switches n.

Remark 1 Assume a Hamiltonian cycle $H = x_0, x_1, \ldots, x_{n-1}$, and let δ be the greatest integer such that for all $0 \leq i < n$, $D(x_i, x_{i+\delta}) = \delta$, and there is a unique shortest path from x_i to $x_{i+\delta}$. The induced subgraph by the (H, λ)-Ant-mill traffic pattern, with $\lambda \leq \delta$, is the H cycle itself. Under this traffic pattern, all traffic in the Hamiltonian cycle is sent from vertex x_i to $x_{i+\lambda}$ for each i. Then, when using minimal routing, λ^{-1} is an upper bound of the amount of traffic that can be injected in every switch. And if there are p servers generating such traffic per each switch, then each server can inject traffic into the network with an average rate of at most $p^{-1}\lambda^{-1}$. Furthermore, if the number of servers per switch, p, has been chosen to reach full throughput under uniform traffic, as $d/\bar{D}$, with $\bar{D}$ being the associated average hop count, then the slowdown of λ-Ant-mill relative to uniform is $\frac{d\lambda}{\bar{D}}$. Hence, λ-Ant-mill is an adverse traffic pattern when $\delta \geq \lambda > \frac{2\bar{D}}{d}$, with the lower limit being a scenario where the slowdown is just 2.

It may seem challenging to base a traffic pattern on finding Hamiltonian cycles, which is known to be a NP-complete problem [17]. However, it is known that almost all d-regular graphs are Hamiltonian for fixed $d \geq 3$ [30], and for all cases of our interest, it can be obtained very quickly. Additionally, as seen in Remark 1, it is important whether the Hamiltonian cycle contains unique shortest paths. Therefore, we will assume that the (H, λ)-Ant-mill pattern uses Hamiltonian cycles containing unique shortest paths up to distance δ, with δ being the maximum for the

considered topology. When considering a Hamiltonian cycle without this assumption, we denote it by $(\hat{H}, \lambda)$-Ant-mill by differentiating the Hamiltonian with a "hat." Although finding the Hamiltonian cycle with uniqueness is harder, it can still be done quickly enough, as discussed in Sect. 5.

4.1 Scope of application

The motivation behind considering ill-behaving traffic patterns is to avoid designing the system for just a few workloads. However, scenarios where the Ant Mill pattern or a similar pattern could actually occur are not too far-fetched. For instance, when several servers are connected to each switch, simply directing each switch to send traffic to a neighboring switch creates the simplest Ant Mill pattern. Additionally, a subset of collective operations, such as implementing all-to-all communication by sending traffic from the i-th switch to the $(i + k)$-th switch in the k-th step, can generate Ant Mill traffic. Moreover, random patterns may contain large enough sequences of switches with the properties of the Ant Mill pattern. While such occurrences may be rare, they should not compromise the entire system if they do happen. Cycles also arise in ring All-reduce, which is gaining notoriety as part of the training of some deep neural networks [37]. Finally, in certain structured topologies, it may appear more frequently, and indeed, some patterns in the literature specific to certain topologies can be viewed as particular cases of the Ant Mill traffic pattern. Several examples of such cases conclude this section.

The concepts presented can be applied to various direct topologies. For instance, in a Moore graph [28], it is guaranteed that all shortest paths are unique. Thus, for any cycle[1], we have $\delta = D(G)$. More generally, if the topology has minimum cycle length, or *girth*, g, then the uniqueness is guaranteed for some $\delta \geq \lfloor \frac{g-1}{2} \rfloor$. Nevertheless, large girth is not necessary for the uniqueness. If the number of cycles of length g is small, it may be possible to achieve greater δ by building a Hamiltonian cycle that avoids the edges in those short cycles. For instance, demi-projective networks [13] and Slimfly have a girth of 3 but for a few cycles, which means their structure is very similar to having a girth of 5, allowing for Hamiltonian cycles with $\delta = 2$. However, having an even integer as the girth does not provide an advantage for δ. For example, projective networks [13] have girth $g = 6$ but $\delta = 2$ similar to a topology with a girth of 5. In RRGs, we will manage δ values by a computational approach in the following section.

An outstanding low-diameter direct network is the HyperX [1], a Hamming graph-based network also known as Flattened Butterfly [23]. The uniqueness needed by the Ant Mill is only obtained for $\delta = 1$, and the throughput it provides with minimal routing is the same as any other permutation of the switches. In other words, any of these permutations are adverse, and their main differences lie in which alternative routes should be taken. Therefore, HyperX will not be considered in the evaluation section, as it would not yield any new insight into the problem.

[1] The Petersen graph is not Hamiltonian, but each of its cycles has this uniqueness up to $\delta = 2$. Ant Mill could be approximated either by skipping one vertex or by a decomposition into two 5-cycles.

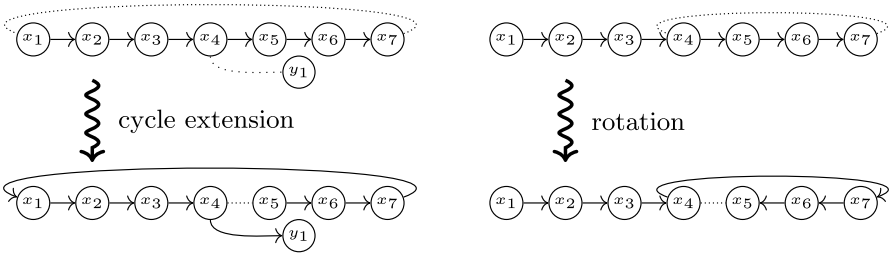

Fig. 4 Cycle extension and rotation to build Hamiltonian cycles

The Dragonfly is a hierarchical network with fully connected groups. In its largest form, the groups also form a complete graph, meaning there is a *global* link for every pair of groups. This naturally provides almost unique shortest routes of the form local–global–local. However, a cycle alternating local and global links only has $\delta = 2$, as each global–local–global path inside the cycle would have an alternative local–global–local (or shortest) outside the cycle. A worse pattern for the Dragonfly is the ADV-k from [16], where each node in group k sends traffic to a node in the group $g + k$. This pattern can be seen as a cycle going through each group once. Then, the nodes outside the cycle are made to send traffic to the same destination group as the one used by other sources in their group. The resulting pattern is indeed much more adverse than Ant Mill for minimal routings, but both cause such a great degradation of performance that alternative routing is required.

The idea of Ant Mill may be applied to topologies of higher diameter. Let us consider tori, which are Hamiltonian. Any Hamiltonian cycle in a torus must have at least as many zigzags as its lesser side. The presence of these zigzags imply $\delta = 1$, as for two hops, there are both one XY path and one YX path. However, their small number could mean it acts in practice more alike the $\delta =$ side. Indeed, if we allow the use of 2-factors other than Hamiltonian cycles, we find that decomposing the torus into parallel cycles is equivalent to the tornado traffic pattern, which is a worst-case traffic pattern for tori [36].

5 Hamiltonian cycle search

As discussed in the previous section, Ant Mill traffic pattern requires the construction of a Hamiltonian cycle that satisfies certain properties. In this section, we provide an algorithm for building such a cycle. For further information on Hamiltonian cycles, readers can refer to [7].

Let $x_1, x_2, \ldots, x_l$ be distinct vertices forming a path. This path is to be transformed by certain operations until we achieve the cycle. Such basic operations [7] are the following, and they are ilustrated in Fig. 4:

1. If there is some neighbor w of x_l outside the path, we can simply extend the path by adding w, resulting in $x_1, x_2, \ldots, x_l, w$.

2. If x_1 is neighbor of x_l and that connected component contains more than l vertices, then there is some vertex y_r outside the path. Let $x_i, y_1, y_2, \ldots, y_r$ be the shortest path from the closer x_i to y_r. We obtain a longer path by replacing the link joining x_i and x_{i+1} with the path to y_r. The resulting path is $x_{i+1}, x_{i+2}, \ldots, x_{i-1}, x_i, y_1, y_2, \ldots, y_r$. This operation is called a *cycle extension*.
3. If x_i is a neighbor of x_l other than $i \neq l-1$, then we can change direction at x_i to create another path of the same length l. This results in the path $x_1, x_2, \ldots, x_{i-1}, x_i, x_l, x_{l-1}, \ldots, x_{i+2}, x_{i+1}$. This is called a *rotation* or *simple transform* and enables further transformations.

The procedure consists on applying Operation 1 at every opportunity and Operation 3 otherwise. Operation 2 can be applied in the same way as Operation 1, but the scarcity of opportunities make it unnecessary in practice.

The algorithm in [7] was proposed for random irregular graphs, with a stated complexity of $O(n^{4+\epsilon})$ for any $\epsilon > 0$, where n denotes the number of vertices in the graph. However, for regular graphs with a degree $d \geq 4$, this algorithm finds a cycle much more quickly. It is important to note that, for any path, an endpoint can be modified in $d-1$ different ways, considering the three operations outlined in the algorithm. Applying the operations randomly can be likened to a random walk, with the cycle being completed when all vertices are visited. The time to complete this process is known as the *cover time*, which is known to be $\Theta(n \log n)$ in expander graphs (and hence in RRGs) [10]. With a $O(n)$ cost per rotation, the total number of operations is $O(n^2 \log n)$.

To build a Hamiltonian cycle with unique shortest paths up to δ, we proceed in the same way, but applying only operations that maintain this uniqueness. This reduces the number of candidates potentially to zero, which makes necessary to allow backtracking. Nevertheless, a simple depth-first search provides good results. Note that, for extend by a node w, it suffices to check whether $D(x_{l+1-\delta}, w) = \delta$ and the uniqueness of the shortest path. In the case of rotations, it is necessary to enforce δ pairs: $D(x_{i-j}, x_{l+j+1-\delta}) = \delta$ for $0 \leq j < \delta$, along with the associated uniqueness. In the graphs simulated in the following section, a Hamiltonian cycle with the given δ can be quickly be found using this approach.

6 Evaluation

In this section, we conduct an exhaustive evaluation of the Ant Mill traffic pattern using several topologies and routings. The first subsection details the experimental methodology, the second presents results for various RRGs, and the last one discusses empirical results for various low-diameter direct topologies.

6.1 Experimental setup

We simulate both RRGs and other low-diameter direct topologies. The simulated topologies and their parameters are detailed in Table 1. Specifically, RRGs

Table 1 Topologies included in the simulations. The topology in bold has been employed for a deep analysis of multiple features

Topology	Switches	Servers	d	Radius	Diameter	δ
RRG$k = 2.2$	242	4598	36	2	3	1
RRG$k = 2.5$	353	4589	28	3	3	1
RRG$k = 3.2$	780	5460	18	3	4	2
RRG$k = 3.7$	**1224**	**6120**	**14**	**4**	**4**	**2**
Dragonfly	876	5256	36	3	3	2
Slimfly	338	3042	19	2	2	2
Projective	614	4912	18	3	3	2
Demi-projective	381	3810	19–20	2	2	2

The RRG of 1224 switches is in bold to indicate it was used for the most detailed evaluations, which are shown in Fig. 5

constructed from various relations between the degree d and the number of switches n may exhibit different values of the maximum δ for which a Hamiltonian cycle can be constructed, ensuring unique paths up to distance δ. The distance properties of RRGs, including δ, are directly associated with the value k that satisfies the equation $d^k = 2n \ln n$ [6, 8]. Our simulations encompass a spectrum of RRG topologies ranging from exponent $k = 2.2$ to $k = 3.7$, as detailed in Table 1. This table also presents the maximum δ for which a Hamiltonian cycle with unique paths up to distance δ has been constructed, along with values of the radius and diameter. It is important to note that the *radius* represents the smallest eccentricity, *i.e.,* radius $= \min_y \max_x \{D(x, y)\}$. Clearly, the radius serves as an upper bound on δ, although it is never attained in the considered RRGs. Each simulated RRG is a medoid with respect to the space (link use, average distance) obtained from a sample of 100 independent RRGs. This ensures that the utilized instances closely resemble the majority of other instances. Additionally, various configurations of Dragonfly, Slimfly, and Projective networks are simulated.

The experiments are conducted using the CAMINOS simulator [12]. CAMINOS is an event-driven, phit-level simulator implemented in the Rust language and freely available for use. We employ a simple switch model implemented in the simulator, configured with buffers at both inputs and outputs to prevent deadlock, and employ the virtual channel policy based on [18]. The configuration follows standard practices: synthetic traffic generated following a Bernoulli process with destinations determined by the traffic pattern, virtual cut-through utilized for flow control, packets consisting of 16 phits, and a capacity for four packets in the input buffer and two packets in the output buffers. The metrics to be measured include throughput, average latency, and the Jain index [20]. The Jain index is a measure of fairness that is calculated as $\frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \sum_{i=1}^{N} x_i^2}$, where x_i is the load generated by server i and N is the total number of servers.

For each topology, it has been built a Hamiltonian cycle H with unique paths up to distance δ (the value in Table 1) and another Hamiltonian cycle $\hat{H}$ without the requirement of unique paths. Then, the traffic patterns simulated are (H, λ)

-Ant-mill and $(\hat{H}, \lambda)$-Ant-mill, where λ fulfills $1 \leq \lambda \leq \delta$. These traffic patterns are compared against the following:

- **Uniform traffic pattern**: Each source selects a new random target for each new communication.
- **Random server permutation**: A randomly selected permutation π of the servers is generated and used for the entire simulation. Whenever server x initiates a new communication, its target is server $\pi(x)$.
- **Switch permutation toward distance** λ: A randomly selected permutation of the switches is created, with each destination being at distance λ from its source. This is carried out for each possible value, $1 \leq \lambda \leq$ radius.

In the case of Dragonfly, as mentioned earlier, a well-known adverse traffic pattern denoted as ADV-h is also simulated [16]. In this traffic pattern, each packet from a server in group g has its destination set as a randomly selected server in group $g + h$, where h represents the number of global links per switch.

Regarding routing algorithms, since Ant Mill has been designed to stress minimal routing, simulations are primarily focused on this routing. Consistent with our definition of adverse traffic patterns, we include the Valiant scheme as a classical baseline. As a competitive adaptive routing strategy, we employ Polarized routing [11]. For RRGs, we also incorporate K-shortest paths routing (8-KSP) [40] as a simpler mechanism.

In the Valiant routing algorithm, for each communication, a random intermediate switch is chosen. Subsequently, communication is initiated minimally from the source server to the intermediate switch and then again minimally from the intermediate switch to the destination server. In certain special cases, the first subroute may include the destination, and the communication is completed at that point.

Polarized routing has demonstrated good performance across various topologies, indicating that issues caused by pathological traffic such as Ant Mill can be largely mitigated by employing a suitable routing algorithm. In Polarized routing, each switch determines the next hop to be taken based on a function of the distances to the source and destination, as well as the occupancy of the queues. Priority is given to the shortest routes, while many other routes are considered when they are underutilized.

In 8-KSP, a collection of eight routes among the shortest ones is selected for each pair of switches. This set of eight routes may consist solely of minimal routes or include a few longer routes if there are fewer than eight routes of minimal length available. Each communication will then utilize a randomly selected route from the pool of routes chosen for that particular source and destination pair. There exist several routing strategies derived from KSP, such as LLSK [41], KSP-adaptive [3], and KSP-UGAL [29]. However, results from these routing strategies are not included here, as their resulting performance is lower than that of Polarized routing and does not offer any new insights.

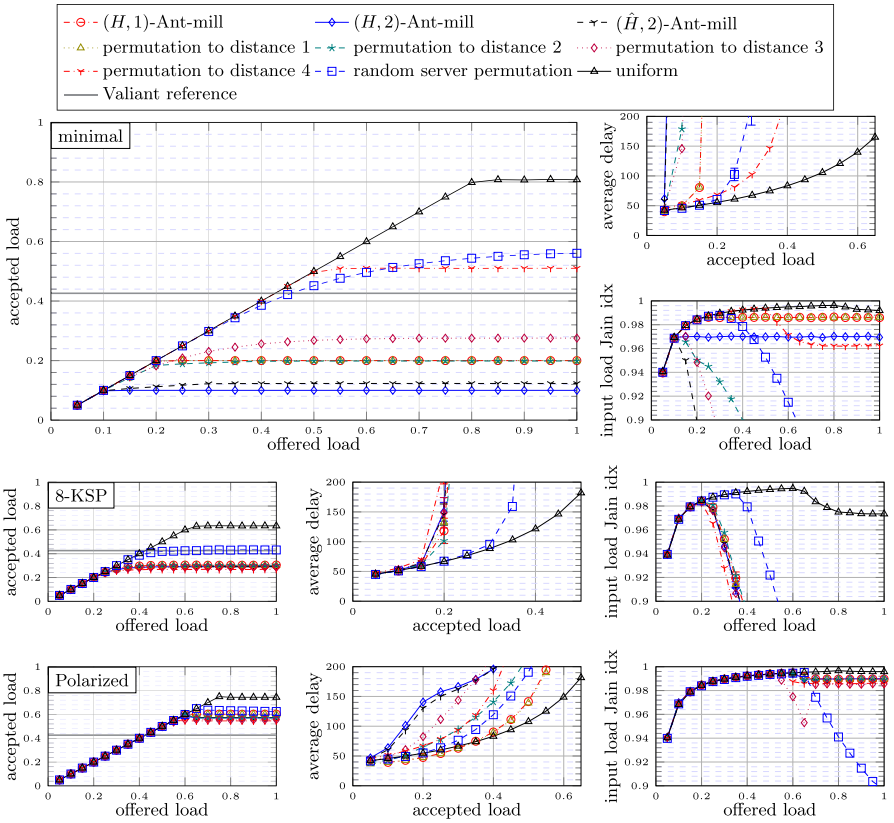

Fig. 5 Results for the RRG with 1224 routers for minimal and 8-KSP routings. $\hat{H}$ only included when results are visibly different. Maximum throughput when using Valiant routing included as reference

6.2 Experimental results for RRGs

Let us initially analyze common characteristics of the entire spectrum of simulated RRGs. We utilize the RRG with 1224 switches as a representative example. In Fig. 5, the main frame displays throughput against network load for various traffic patterns under minimal routing; the two smaller frames to its right illustrate the associated latency and fairness metrics. The second row, comprising three smaller frames, examines the performance under 8-KSP routing. The third and final row presents the results obtained using Polarized routing. Additionally, in all experiments, we include Valiant throughput as a reference, indicated by a horizontal line.

The uniform traffic pattern yields higher global throughput. As observed, (H, λ) -Ant-mill, with $\lambda = \delta$, exhibits significantly lower throughput compared to the other traffic patterns, specifically 88% less than uniform traffic, demonstrating its highly adversarial nature. The second-lowest throughput is recorded by $(\hat{H}, \delta)$-Ant-mill, indicating a Hamiltonian cycle without path-uniqueness constraints. However, this

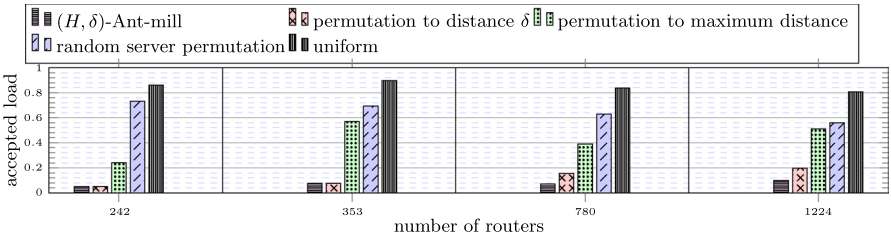

Fig. 6 Maximum throughput for minimal routing on RRGs with different characteristics

throughput improvement is accompanied by a considerable degradation in fairness. Moreover, when $\lambda < \delta$, as illustrated in this example with $\lambda = 1$, there is no discernible difference between H and $\hat{H}$. Regarding permutations, selecting destinations at distance δ yields the poorest throughput. In the presented topology, it closely competes with the permutation at distance $\delta - 1$, although differences are more pronounced in other scenarios. When communications to immediate neighbors are considered, there is no distinction between employing a permutation or an Ant Mill pattern with $\lambda = 1$. Notably, a permutation to the maximum distance achieves the highest throughput among all considered permutations, despite being established as a worst-case traffic scenario with ideal routing [21]. It is worth noting that this longest matching throughput is similar to that obtained by a random server permutation, which is evidently less adversarial than any switch permutation.

When examining the results for 8-KSP (the three frames in the middle row), it can be observed that this routing categorizes all the analyzed traffic patterns into three distinct types: uniform (yielding the highest throughput but 22% lower than minimal routing), random server permutation occupying the middle position, and the final category comprising all permutations of switches.

In the bottom row, the results for Polarized routing are presented. As observed, the uniform traffic pattern yields the highest throughput, with a 7.8% reduction compared to minimal routing. For all other traffic patterns, Polarized routing outperforms the other routing algorithms in terms of throughput. However, while throughput remains relatively constant for non-uniform patterns, latencies exhibit notable differences. Most prominently, the average latency of the $(H, 2)$-Ant-mill and $(\hat{H}, 2)$-Ant-mill are the first to rise, as packets traversing the shortest path experience higher latency. Thus, Ant Mill not only reduces throughput but also poses challenges for latency. In the subsequent experiments discussed, latency graphs are not presented as the focus has been on throughput. However, similar scenarios to the one described here are encountered.

Figure 6 shows the maximum accepted load for all random graphs listed in Table 1, focusing solely on minimal routing. We wish to emphasize that the overall behavior observed in the experiments of Fig. 5 is consistent across these graphs as well. Figure 6 shows clearly that (H, δ)-Ant-mill emerges as the most adverse traffic pattern, irrespective of the RRG under consideration. Ant Mill results in a minimum of an 8.1× slowdown compared to the uniform traffic pattern across all the examined networks. It is noteworthy that in comparison with a

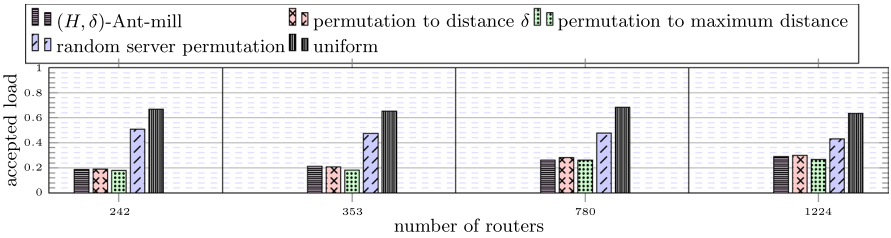

Fig. 7 Maximum throughput for KSP routing on RRGs with different characteristics

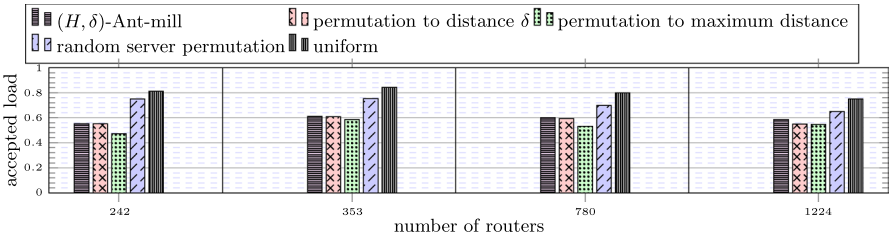

Fig. 8 Maximum throughput for Polarized routing on RRGs with different characteristics

random server permutation, a scenario common in many networks, the observed slowdown ranges from 5.6× to 13.9×.

Similarly, in Fig. 7, the maximum accepted load for 8-KSP routing is depicted. In this case, it can be observed that both Ant Mill and router permutations constitute an adverse situation.

Finally, in Fig. 8, similar results are presented for Polarized routing. Once more, Polarized routing effectively mitigates the adversarial situation, and none of the traffic patterns under consideration can be deemed adversarial.

6.3 Experimental results for low-degree direct networks

In this subsection, we examine similar experiments but for other low-diameter direct networks, namely Dragonfly, Slimfly, and Projective networks. Initially, Fig. 9 illustrates the maximum throughput achieved for minimal routing. As it can be observed, in both Slimfly and demi-projective networks, we have $\delta =$ radius, so the maximum distance for permutations is δ, and those two bars coincide. The (Levi) projective network, with a radius of 3, exhibits $d = 18$ shortest paths to any destinations at a distance of 3, which suffices to yield good performance. Conversely, for destinations at a distance of $\delta = 2$, there exists only one shortest path. Finally, for the Dragonfly network, we also showcase the specific ADV+h adverse traffic pattern, which notably yields the lowest throughput on the Dragonfly. This pattern, akin to Ant Mill in terms of link usage, places a non-uniform stress on those links by directing an overwhelming load to a few global links. Regardless,

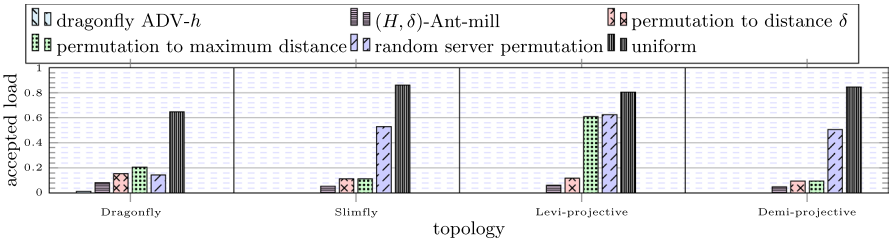

Fig. 9 Maximum throughput for routing minimally on various low-diameter direct networks. In the Dragonfly network, the shortest routing is the standard hierarchical variant that uses at most one global link

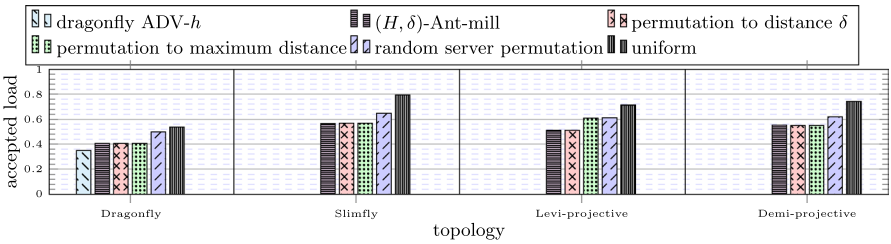

Fig. 10 Maximum throughput for Polarized routing on various low-diameter direct networks

Ant Mill remains a severe adverse traffic pattern for all evaluated topologies, effectively representing a generic adverse traffic pattern for each of them.

Finally, in Fig. 10, similar results are considered but for Polarized routing.

7 Conclusions

It is possible to identify adversarial traffic patterns for particular networks by an in-depth examination of their topological structure, which proves to be exceptionally challenging for random networks. This underscores the necessity for a systematic approach to constructing adverse traffic patterns. It is a misleading intuition to consider communications to the longest possible distance as adverse situations. In fact, our findings, utilizing practicable routing schemes, demonstrate that communications to neighbors can pose greater management difficulties. This insight has led to the definition of a new traffic pattern, Ant Mill, which exhibits even greater adversarial characteristics than typical permutation-based traffic patterns.

In Ant Mill, communications are established at a certain distance λ within a Hamiltonian cycle embedded in the network. We have demonstrated that if this cycle is constructed using unique shortest paths, it yields the most adverse traffic pattern in RRGs. Furthermore, the principles underlying the definition of Ant Mill directly apply to low-diameter direct networks, as we have shown, thereby enabling us to assert that Ant Mill constitutes a general adverse traffic pattern for this class of networks.

Author contributions All authors contributed to the main document text and its whole review. C. Camarero prepared figures and carried out software work.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. C. Camarero is under Ramón y Cajal contract RYC2021-033959-I from Spain's Ministerio de Ciencia e Innovación with funding from the Mecanismo de Recuperación y Resiliencia de la Unión Europea. The three authors participate in these projects: PLANIFICADORES Y REDES PARA DATA CENTERS SOSTENIBLES project TED2021-131176B-I00 with funding from MCIN/AEI /10.13039/501100011033 and Unión Europea/NextGenerationEU/PRTR; REDES DE INTERCONEXIÓN, ACELERADORES HARDWARE Y OPTIMIZACIÓN DE APLICACIONES, project PID2019-105660RB-C22 with funding from MCIN/AEI /10.13039/501100011033; and ARQUITECTURA Y PROGRAMACIÓN DE COMPUTADORES ESCALABLES DE ALTO RENDIMIENTO Y BAJO CONSUMO III-UC (TEAMMATES UC) project PID2022-136454NB-C21 with funding from MICIU/AEI /10.13039/501100011033 and FEDER, UE. R. Beivide is supported by The Barcelona Supercomputing Center (BSC) under contract CONSER02023011NG. Some simulations have been performed in the supercomputer Altamira Supercomputer at the Institute of Physics of Cantabria (IFCA-CSIC), member of the Spanish Supercomputing Network, with the support of the Santander Supercomputacion support group at the University of Cantabria.

Data and materials availability No data have been generated.

Declarations

Conflict of interest The authors do not have any conflict of interest.

Ethical approval Not applicable.

References

1. Ahn JH, Binkert N, Al D, McLaren M, Schreiber RS (2009) HyperX: topology, routing, and packaging of efficient large-scale networks. In: Proceedings of the conference on High Performance Computing Networking, Storage and Analysis, SC'09, New York, NY, USA, ACM. pp 1–11
2. Ajima Y, Kawashima T, Okamoto T, Shida N, Hirai K, Shimizu T, Hiramoto S, Ikeda Y, Yoshikawa T, Uchida K, Inoue T (2018) The Tofu interconnect D. In: 2018 IEEE International Conference on Cluster Computing (CLUSTER), pp 646–654
3. ALzaid Z, Bhowmik S, Yuan X (2021) Multi-path routing in the Jellyfish network. In: 2021 IEEE international parallel and distributed processing symposium workshops (IPDPSW), pp 832–841
4. Atchley S, Zimmer C, Lange JR, Bernholdt DE, Melesse Vergara VG, Beck T, Brim MJ, Budiardja R, Chandrasekaran S, Eisenbach M, Evans T, Ezell M, Frontiere N, Georgiadou A, Glenski J, Grete P, Hamilton S, Holmen J, Huebl A, Jacobson D, Joubert W, McMahon K, Merzari E, Moore SG, Myers A, Nichols S, Oral S, Papatheodore T, Perez D, Rogers DM, Schneider E, Vay J-L, Yeung PK

(2023) Frontier: exploring exascale the system architecture of the first exascale supercomputer. In: SC23: International Conference for High Performance Computing, Networking, Storage and Analysis, pp 1–16
5. Besta M, Hoefler T (2014) Slim Fly: a cost effective low-diameter network topology. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC'14, Piscataway, NJ, USA. IEEE Press. pp 348–359
6. Bollobás B (2001) Random Graphs. Cambridge studies in advanced mathematics, 2nd edn
7. Bollobas B, Fenner TI, Frieze AM (1987) An algorithm for finding Hamilton paths and cycles in random graphs. Combinatorica 7(4):327–341
8. Bollobás B, Fernandez de la Vega W (1982) The diameter of random regular graphs. Combinatorica 2(2):125–134
9. Brahme D, Bhardwaj O, Chaudhary V (2013) SymSig: a low latency interconnection topology for HPC clusters. In: 20th International Conference on High Performance Computing (HiPC), pp 462–471
10. Broder AZ, Karlin AR (1989) Bounds on the cover time. J Theor Probab 2(1):101–120
11. Camarero C, Martinez C, Beivide R (August 2021) Polarized routing: an efficient and versatile algorithm for large direct networks. In: 2021 IEEE symposium on high-performance interconnects (HOTI), Los Alamitos, CA, USA. IEEE Computer Society, pp 52–59
12. Camarero Cristóbal. CAMINOS. https://crates.io/crates/caminos
13. Camarero C, Martínez C, Vallejo E, Beivide R (2017) Projective networks: topologies for large parallel computer systems. IEEE Trans Parallel Distrib Syst 28(7):2003–2016
14. Clos C (1953) A study of non-blocking switching networks. Bell Syst Tech J 32(2):406–424
15. Dally W, Towles B (2003) Principles and practices of interconnection networks. Morgan Kaufmann Publishers Inc., San Francisco
16. García M, Vallejo E, Beivide R, Odriozola M, Camarero C, Valero M, Rodríguez G, Labarta J, Minkenberg C (2012) On-the-fly adaptive routing in high-radix hierarchical networks. In: The 41st International Conference on Parallel Processing (ICPP), pp 279–288
17. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-Completeness (series of books in the mathematical sciences). W. H. Freeman, 1st edn
18. Günther KD (1981) Prevention of deadlocks in packet-switched data transport systems. IEEE Trans Commun 29(4):512–524
19. Staff IBG (2008) Overview of the IBM Blue Gene/P project. IBM J Res Dev 52(1/2):199–220
20. Jain R, Chiu DM, Hawe W (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. In: DEC Research Report TR-301
21. Jyothi SA, Singla A, Godfrey PB, Kolla A (2016) Measuring and understanding throughput of network topologies. In: SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, pp 761–772
22. Kathareios G, Minkenberg C, Prisacari B, Rodriguez G, Hoefler T (2015) Cost-effective diameter-two topologies: analysis and evaluation. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC'15, New York, NY, USA. ACM. pp 1–11
23. Kim J, Dally WJ, Abts D (2007) Flattened butterfly: a cost-efficient topology for high-radix networks. In: Proceedings of the 34th annual international symposium on Computer architecture, ISCA '07, New York, NY, USA. ACM. pp 126–137
24. Kim J, Dally WJ, Scott S, Abts D (2008) Technology-driven, highly-scalable dragonfly topology. In: Proceedings of the 35th annual international symposium on computer architecture. IEEE Computer Society, pp 77–88
25. Koibuchi M, Matsutani H, Amano H, Hsu DF, Casanova H (2012) A case for random shortcut topologies for HPC interconnects. In: Proceedings of the 39th annual international symposium on computer architecture, ISCA'12, Washington, DC, USA. IEEE Computer Society, pp 177–188
26. Mellette WM, Das R, Guo Y, McGuinness R, Snoeren AC, Porter G (2020) Expanding across time to deliver bandwidth efficiency and low latency. In: 17th USENIX symposium on networked systems design and implementation (NSDI 20), pp 1–18
27. Meuer H, Strohmaier E, Dongarra J, Simon H, Meuer M (2023) Top500 supercomputer sites. http://www.top500.org/lists/2023/11/
28. Miller M, Širáň J (2013) Moore graphs and beyond: a survey of the degree/diameter problem (2nd ed). Electron J Comb 5

29. Mollah Md A, Faizian P, Rahman Md S, Yuan X, Pakin S, Lang M (2018) A comparative study of topology design approaches for HPC interconnects. In: 2018 18th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID), pp 392–401
30. Robinson RW, Wormald NC (1994) Almost all regular graphs are Hamiltonian. Random Struct Algorithms 5(2):363–374
31. Michael S (2004) Vertex-symmetric generalized Moore graphs. Discrete Appl Math 138(1–2):195–202
32. Schneirla TC (1944) A unique case of circular milling in ants, considered in relation to trail following and the general problem of orientation. Am Mus Novitates, pp 1–26
33. Singh A, Dally WJ, Towles B, Gupta AK (2002) Locality-preserving randomized oblivious routing on torus networks. In: Proceedings of the fourteenth annual ACM symposium on parallel algorithms and architectures, SPAA'02, New York, NY, USA. Association for Computing Machinery, pp 9–13
34. Singla A, Godfrey PB, Kolla A (2014) High throughput data center topology design. In: 11th USENIX symposium on networked systems design and implementation (NSDI 14), Seattle, WA. USENIX Association, pp 29–41
35. Singla A, Hong C-Y, Popa L, Godfrey PB (2012) Jellyfish: networking data centers randomly. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI'12, Berkeley, CA, USA. USENIX Association, pp 17–17
36. Towles B, Dally WJ (2002) Worst-case traffic for oblivious routing functions. In: Proceedings of the fourteenth annual ACM symposium on parallel algorithms and architectures, SPAA'02, New York, NY, USA. Association for Computing Machinery, pp 1–8
37. Ueno Y, Yokota R (2019) Exhaustive study of hierarchical all reduce patterns for large messages between GPUs. In: 2019 19th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID), pp 430–439
38. Valerio M, Moser LE, Melliar-Smith PM (1994) Recursively scalable fat-trees as interconnection networks. In: IEEE 13th Annual International Phoenix Conference on Computers and Communications, pp 40–46
39. Valiant LG, Brebner GJ (1981) Universal schemes for parallel communication. In: Proceedings of the thirteenth annual ACM symposium on theory of computing, STOC'81, New York, NY, USA. ACM. pp 263–277
40. Yen JY (1971) Finding the k shortest loopless paths in a network. Manag Sci 17(11):712–716
41. Yuan X, Mahapatra S, Nienaber W, Pakin S, Lang M (2013) A new routing scheme for jellyfish and its performance with HPC workloads. In: SC'13: proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp 1–11