Exploring the Galaxyfly Family to Build Flexible-Scale Interconnection Networks

Fei Lei, Dezun Dong, and Xiangke Liao

Abstract—Interconnection networks play an essential role in the architecture of high-performance computing (HPC) systems. In this article, we explore the Galaxyfly family to build flexible-scale interconnection networks. Galaxyfly is guaranteed to retain a small constant diameter while achieving a flexible tradeoff between network scale and bisection bandwidth. Galaxyfly not only supports small-scale interconnection networks with smaller diameter but also lowers the demands for high-radix routers and is able to utilize routers with moderate radix to build exascale interconnection networks. We analyze the constructible configuration of Galaxyfly and evaluate the properties of Galaxyfly. We conduct extensive simulations and analysis to evaluate the performance, cost, and power consumption of Galaxyfly on physical layout against state-of-the-art topologies. The results show that our design achieves better performance than most existing topologies under typical HPC workloads, and is cost-effective to deploy for exascale HPC systems.

Index Terms—High-performance computing, interconnection, topology, flexible-radix, low-diameter

1 Introduction

Large-scale supercomputers currently have tens of thousands of compute nodes, for example, Sunway Taihulight system and Tianhe-2 system, and are progressing towards larger size. Exascale systems require hundreds of thousands of interconnected processors. Interconnection networks, being essential to the scalability of high-performance computing (HPC) systems, impact the overall goals of system design. One of the most important design issues of interconnection networks is topology, which establishes performance bounds, i.e., end-to-end latency and bisection bandwidth and determines the cost of the network.

There are several metrics that have to be taken into account when designing an efficient topology for next-generation large-scale supercomputers. First, high bandwidth and low latency are indispensable, which has always been the primary target for optimizing network topologies. Second, an efficient topology has to be both cost and power effective. Power consumption and total cost of ownership are subjected to strict constraints, e.g., 20–30 MW power envelope for exascale systems [1]. The building cost and power consumption of a network can attain 33 percent [2] and 50 percent [3] of the whole system cost, respectively. Last but not least, flexibility is emerging as a required characteristic of efficient topology. According to an application-driven perspective on HPC system designs [4], interconnection networks are expected to be adaptable to applications of varied scales and communication requirements through static deployment or dynamic reconfiguration.

• *The authors are with the College of Computer, National University of Defense Technology, Changsha 410073, China. E-mail: {leifei, dong, xkliao}@nudt.edu.cn.*

Manuscript received 4 Sept. 2020; revised 27 May 2021; accepted 7 July 2021. Date of publication 29 July 2021; date of current version 15 Oct. 2021.
(Corresponding author: Dezun Dong.)
Recommended for acceptance by J. Zhai.
Digital Object Identifier no. 10.1109/TPDS.2021.3100783

Most recent research focuses on designing low-diameter networks, as well as optimizing cost and other factors. Some excellent topologies have been proposed to offer low diameters, e.g., Flattened Butterfly [2], Dragonfly [5], HyperX [6], Skywalk [7], and Slim Fly [8]. These topologies, however, still face great challenges when scaling to interconnect exascale systems or beyond. The main constraint lies in that their scalability excessively depends on the port number (radix) of the building blocks (routers). We argue that it is difficult for high-performance routers, especially commercial-off-the-shelf (COTS) routers in the current or near future, to increase their radix and sufficiently meet the requirements of existing high-radix topologies in exascale computing and beyond. 48-radix routers [9] are the commercial building blocks currently available and will be used by Argonne, Cray, and Intel to build a 180 petaflops HPC system, Aurora, in 2018. Chip physical resources and power consumption are the two key limits to boosting the radix [10]. In electronic router chips, increasing radix while maintaining or increasing per-port bandwidth is limited by the wiring complexity of the switching crossbar, high bandwidth density at the chip edge [10] and high speed SerDes I/O. Furthermore, the bytes-to-flops ratio, which refers to the amount of each node's network communication with respect to its floating-point capability, is an important metric for a well-balanced interconnect. The link bandwidth of one compute node may have to increase proportionally with the computation capability. The race is on to develop 200 Gbps technologies for the requirements of today's most advanced networks [11]. The radix of the newest 200G Mellanox HDR InfiniBand switch is only 40-port [12]. It is a great challenge to build a high-radix router that increases both the radix and port bandwidth simultaneously. Silicon nanophotonics technology is expected to solve these problems. Altera and Avago have made an optical FPGA based on the concept of embedded parallel optical modules [13]. However, it is still difficult to accommodate and integrate a sufficient number of optical modules in one chip. The growth of radix will increase the power budget of router chips. For example, one SerDes link transmitting a bit costs approximately 20 pJ in Tianhe-2 system, and SerDes consumes 79 percent of the overall power budget of a router chip [14]. In addition, floorplan and thermal constraints on physical package design will radically impact the number of high-speed SerDes transceivers and the router radix density [11]. Hence, interconnection design for exascale systems and beyond may have to face the issue of the scaling down of router radix.

Most of the existing work focuses on high-radix topologies, such as Fat Tree, Flattened Butterfly [2], Dragonfly [5], HyperX [6], Skywalk [7], and Slim Fly [8]. High-radix topologies can attain lower diameter than low-radix topologies such as Torus and Mesh. Fat Tree is a classical high-radix and indirect network, which can be expanded by adding levels with lower radix routers but consumes a large amount of cables and routers. Dragonfly [5] not only achieves low diameter but also combines the advantages of long optical channels and short electrical channels to reduce the cost. However, it is still intractable to build an exascale network of 100K size using Dragonfly and the newest COTS high-radix routers. Indeed, the scalability issue of Dragonfly has limited its deployment in real HPC systems. For instance, Cray Cascade system [15] employs a variation of Dragonfly, that is, the intra-group is 2-D Flattened Butterfly [2] instead of a fully connected graph to improve the scalability of Dragonfly. HyperX [6] is a flexible topology via adjusting various parameters and every dimension is fully connected thus making it suitable for optical interconnection. Flattened Butterfly and Hypercube are both special cases of HyperX. A common challenge among HyperX topologies is that fully connected channels reduce the growth of network scale and increase the network costs. Random topology [16], [17], [18] provides various design spaces and satisfies requirements like incremental expansion. However, the routing algorithm and the link congestion affect the network performance. Slim Fly explores an elaborate graph theory that aims to be close to Moore bound. The goal of Slim Fly is to achieve the largest scale given the degree and diameter. However, the scalability of Slim Fly is limited by its low diameter and router radix. Once the growth of router radix stagnates, we have to reexamine the design tradeoff in high-radix topology between node degree and network diameter, as well as between performance and scalability.

This paper aims at constructing flexible-scale (especially exascale and even beyond) interconnection networks using routers with moderately high radixes. We make an initial attempt at building a family of flexible-radix low-diameter topologies, following a new tradeoff that has not previously been explored sufficiently. The preliminary goal is to build the interconnection by utilizing state-of-the-art COTS routers. A further goal is to improve the flexibility of the topology, so that the topology can be deployed in systems of different scales, from petascale to exascale systems and beyond. In this paper, we explore the Galaxyfly family and it lowers the demand for high-radix routers to expand flexibly with different configurations. It also retains a small constant diameter while achieving a flexible tradeoff between network scale and bisection bandwidth. Furthermore, by exploring the algebraic properties of Galaxyfly, we develop congestion-aware routing algorithms. In addition, Galaxyfly is a flexible hierarchical structure consisting of two building blocks, Galaxy graph and fully connected graph, which can make the most of optical chip-to-chip and backplane interconnect technology and is easy to layout in a cost-effective way. The flexible hierarchy of Galaxyfly also endows it the potential to support different traffic characteristics, i.e., using full connection to absorb the dominating local traffic, and using redundant global links to adaptively serve global traffic. Dragonfly and fully connected graph are both special cases of Galaxyfly.

In summary, the main contributions of this works are as follows:

- We define a novel flexible topology, Galaxy graph, whose diameter is at most 2.
- We design and analyze a novel family of flexible-radix low-diameter topologis, Galaxyfly, based on Galaxy graph and fully connected graph. In addition, we compare Galaxyfly with other classical topologies in flexibility, constructible configuration, shortest paths and resiliency.
- Taking advantage of Galaxy graph, we design a minimal routing algorithm and several non-minimal adaptive routing algorithms for Galaxyfly and compare different configurations with other classical topologies on physical layout under various workloads.
- Compared to other classical topologies, Galaxyfly can achieve improvements in both building cost and power consumption owing to the flexible layout.

The remainder of the paper is organized as follows: We describe Galaxy graph and Galaxyfly in detail in Section 2 and identify the characteristics of Galaxyfly in Section 3. Section 4 demonstrates routing algorithms for Galaxyfly. Evaluation of Galaxyfly via cycle-accurate simulation and comparison to previously proposed topologies are provided in Section 5. Section 6 shows the layout and power consumption of Galaxyfly in practice. Related work is discussed in Section 7 and followed by conclusions.

2 Galaxyfly Topology

In this section, we first present a flexible and configurable graph, Galaxy graph, which is derived from algebraic graphs and whose diameter is guaranteed to be at most 2. Then based on Galaxy graph, we design a novel family of low-diameter topologies, Galaxyfly, which lowers the demands for router radix to scale flexibly and achieves a flexible tradeoff between network scale and bisection bandwidth.

2.1 Galaxy Graph

To construct flexible-scale networks with given number of routers and router radix, we design Galaxy graph, whose diameter is at most 2. Similar to Slim Fly [8], we adopt graph techniques introduced by McKay *et al.* in [19], to construct the basic blocks of the diameter-2 Galaxy graph. It is worth noting that those diameter-2 graph techniques close to Moore bound are orthogonal to the design of Galaxy graph [19].

Galaxy graph is a size $n \times q$ graph $G(V, E)$ and is equally divided into n clusters, $V = V_0 \cup V_1 \cup \cdots \cup V_{n-1}$, with q nodes in each cluster. q is a prime power that can be written as $q = 4\ell + \delta$, in which $\ell \in \mathbb{N}$ and $\delta \in \{-1, 0, 1\}$. There are two kinds of edges in Galaxy graph, (1) intra-cluster edges connecting nodes in the same cluster and (2) inter-cluster edges connecting nodes in two different clusters. In the

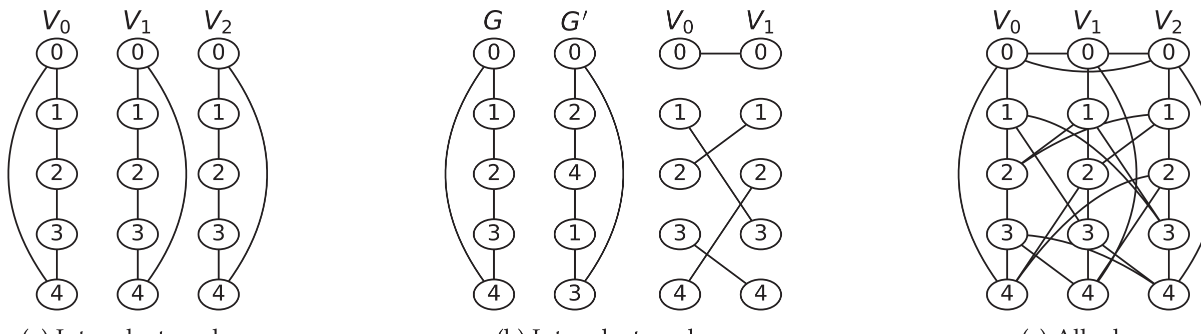


Fig. 1. Galaxy graph with $n = 3$ and $q = 5$.

remainder of this subsection, we first show the construction of Galaxy graph in a step-by-step manner, with the size configuration $n = 3$ and $q = 5$ as a running example. Then we prove that the diameter of Galaxy graph is at most 2.

2.1.1 Intra-Cluster Edges

The construction of intra-cluster edges in Galaxy graph requires some basic knowledge on algebraic graphs over finite fields. We only present basic results necessary to understand the construction process, additional theoretical details of algebraic graphs can be found in [19], [20].

There are q nodes in a cluster. q generates a finite field $\mathbb{F}_q = \{x_0, x_1, \ldots, x_{q-1}\}$. And $\mathbb{F}_q^+$ denotes the additive group of $\mathbb{F}_q$. There exists a primitive element $\xi \in \mathbb{F}_q$, which generates $\mathbb{F}_q$ in the form of $\mathbb{F}_q = \{\xi^t mod\ q | t \in \mathbb{N}\} \cup \{0\}$.

Given the finite field $\mathbb{F}_q$ generated by q, and its primitive element ξ, two *generator set*s, X and X', can be constructed as shown in Equations (1) and (2). Note that all arithmetic operations are performed in a modulo q manner. It is easy to prove that $|X| = |X'| = (q - \delta)/2$ and $X \cup X' = \mathbb{F}_q - \{0\}$ [20].

$$X = \begin{cases} \{1, \xi^2, \ldots, \xi^{q-3}\} & q = 4\ell + 1 \\ \{1, \xi^2, \xi^4, \ldots, \xi^{2\ell-2}, \\ \xi^{2\ell-1}, \xi^{2\ell+1}, \ldots, \xi^{4\ell-3}\} & q = 4\ell - 1 \\ \{1, \xi^2, \ldots, \xi^{4l-2}\} & q = 4\ell \end{cases} \quad (1)$$

$$X' = \begin{cases} \{\xi, \xi^3, \ldots, \xi^{q-2}\} & q = 4\ell + 1 \\ \{\xi, \xi^3, \xi^5, \ldots, \xi^{2\ell-1}, \\ \xi^{2\ell}, \ldots, \xi^{4\ell-4}, \xi^{4\ell-2}\} & q = 4\ell - 1 \\ \{\xi, \xi^3, \ldots, \xi^{4l-1}\} & q = 4\ell \end{cases}. \quad (2)$$

Definition 2.1. *Let $\mathbb{F}_q$ be a finite field with primitive element ξ, and X, X' are generator sets constructed from ξ. A* generator graph *of $\hat{X} \in \{X, X'\}$ is a q node graph with nodes labeled $\{x_0, x_1, \ldots, x_{q-1}\}$, and there exists an edge between node x_i and x_j if $x_i - x_j \in \hat{X}$.*

For each cluster, we first label the q nodes with $\{x_0, x_1, \ldots, x_{q-1}\}$, and then build the intra-cluster edges in the cluster in the same way as the generator graph of X. That is, without considering inter-cluster edges, each cluster is a generator graph of X.

Example. In a $n = 3$, $q = 5$ Galaxy graph, $q = 5$ is a prime power such that $q = 4\ell + \delta$, $\delta = 1 \in \{-1, 0, 1\}$ and $\ell = 1 \in \mathbb{N}$. q generates a finite field $\mathbb{F}_q = \{0, 1, 2, 3, 4\}$, and $\xi = 2$ is a primitive element of $\mathbb{F}_q$. As in Equations (1) and (2), two generator sets $X = \{1, 4\}$ and $X' = \{2, 3\}$ are constructed from ξ. Following Definition 2.1, we can build intra-cluster edges for all three clusters V_0, V_1 and V_2 in the same way as the generator graph of X. The intra-cluster edges are shown in Fig. 1a.

2.1.2 Inter-Cluster Edges

The construction of inter-cluster edges is determined by an isomorphic function H. We first introduce a lemma on the generator graphs, then give the definition of the isomorphic function H and a lemma on it.

lemma 2.1. *Let $\mathbb{F}_q$ be a finite field with primitive element ξ, and X, X' are generator sets constructed from ξ. If G, G' are generator graphs of X and X', respectively, then G and G' are isomorphic graphs. [20]*

Based on Lemma 2.1, the isomorphic function H is defined as follows:

Definition 2.2. *An* isomorphic function *H is a one-to-one function defined on $\mathbb{F}_q = \{x_0, x_1, \ldots, x_{q-1}\}$ such that for any $\forall x_i \in \mathbb{F}_q$, x_i in G and $H(x_i)$ in G' are corresponding nodes in terms of isomorphism.*

From Definitions 2.1 and 2.2, we have the following lemma on the isomorphic function H:

lemma 2.2. *For $x_i, x_j \in \mathbb{F}_q$, $x_i - x_j \in X$ if and only if $H(x_i) - H(x_j) \in X'$.*

As G and G' are isomorphic graphs, there exists at least one isomorphic function H on $\mathbb{F}_q$. Given an isomorphic function H, the inter-cluster edges between any two clusters V_s and V_t $(0 \le s < t < n)$ are constructed as follows: for each node x_i $(i = 0, 1, \ldots, q-1)$ in V_t, connect it to node $H(x_i)$ in V_s.

Example. Fig. 1b (left) shows the structure of generator graphs G and G' constructed from $X = \{1, 4\}$ and $X' = \{2, 3\}$, respectively. An isomorphic function H on $\mathbb{F}_q$ can be found such that $H(0) = 0$, $H(1) = 2$, $H(2) = 4$, $H(3) = 1$, $H(4) = 3$. With this isomorphic function H, the inter-cluster edges between V_0, V_1 and V_2 can be built. Fig. 1b (right) shows the inter-cluster edges between clusters V_0 and V_1. The full topology of Galaxy graph with $n = 3$ and $q = 5$ is shown in Fig. 1c.

TABLE 1
Parameters of Galaxyfly

Name	Description
n	# of clusters
q	# of supernodes per cluster
a	# of routers per supernode
p	# of links per router to terminals
h	# of links per router to other supernodes

2.1.3 Proof of the Diameter-2 Property

Theorem 2.1. *The diameter of Galaxy graph G is at most 2.*

Proof. For $q = 1$, Galaxy graph is a fully connected graph with diameter 1.

For $q > 1$, consider any two nodes n_i and n_j in Galaxy graph, there are two cases to be discussed: (1) n_i and n_j are in the same cluster and (2) n_i and n_j are in different clusters. Suppose n_i and n_j are labeled with x_i and x_j ($x_i, x_j \in \mathbb{F}_q$), respectively.

(1) n_i and n_j are in the same cluster. Let $m = x_i - x_j$. If $m \in X$ then there is an edge between n_i and n_j and the shortest path between n_i and n_j is 1. If $m \notin X$, n_i and n_j are not directly connected. Both n_i and n_j has $(q-\delta)/2$ edges to other nodes in the same cluster, resulting in a total of $q-\delta$ intra-cluster edges from n_i and n_j connecting to the other $q-2$ nodes. As $q-\delta > q-2$ is always true for any $\delta \in \{-1,0,1\}$, n_i and n_j must has a common adjacent node, and the shortest path between n_i and n_j is 2.

(2) n_i and n_j are in different clusters. Without losing generality, we assume that $n_i \in V_s$, $n_j \in V_t$ and $s < t$. Let $m = x_i - H(x_j)$. As $m \in \mathbb{F}_q$ and $\mathbb{F}_q = X \cup X' \cup \{0\}$, There are three cases to be considered: (a) $m = 0$ then there is an edge between n_i and n_j and the shortest path between n_i and n_j is 1. (b) $m \in X$ then node with label $H(x_j)$ in V_s is a common adjacent node of n_i and n_j, the shortest path between n_i and n_j is 2. (c) $m \in X'$ then by Lemma 2.2, $H^{-1}(x_i) - x_j \in X$. That is, node with label $H^{-1}(x_i)$ in V_t is a common adjacent node of n_i and n_j, the shortest path between n_i and n_j is 2.

Therefore, the diameter of Galaxy graph is at most 2. □

2.2 Galaxyfly

The Galaxyfly topology is built by replacing each node in Galaxy graph with a supernode which is a fully connected graph consisting of a routers. Galaxyfly is defined by 5 parameters (n,q,a,p,h), denoted as GF(n,q,a,p,h). As described in Table 1, GF(n,q,a,p,h) contains n clusters, each cluster contains q supernode, each supernode contains a routers, resulting in a total of $N_r = nqa$ routers. Each router connects to p terminals, so the network size is $N = nqap$.

For each router, p, $a-1$ and h ports are used to link terminals, routers in the same supernode and routers in other supernodes, respectively. Let r be the router radix, then the configuration of GF(n,q,q,p,h) should meet the constraint

$$r \geq p + (a-1) + h.$$

Each supernode has ah links to other supernodes. Recall that each node in Galaxy graph has $n-1+(q-\delta)/2$ edges, then we got another constraint for GF(n,q,a,p,h)

$$ah \geq n - 1 + (q-\delta)/2.$$

The concepts of supernode and cluster are analogous to the supernode in Dragonfly [5] and the subgroup in Slim Fly [8]. The hierarchical structure empowers Galaxyfly to well match the traffic characteristics of various HPC applications. Through adjusting the (n,q,a,p,h) parameters, network scale and bisection bandwidth can be customized to handle a big range of traffic patterns, from dominating local traffic to reasonable global uniform traffic.

Example. Galaxyfly is a family of low-diameter topologies and Table 2 lists some example configurations with different diameters. The diameter upper bound of GF(n,q,a,p,h) is 5. Generally, the h is equal to $n-1+(q-\delta)/2a$. We use GF(n,q,a,p) instead of GF(n,q,a,p,h).

TABLE 2
Galaxyfly Examples

Configuration	Diameter	Description
GF$(n=1, q=1, a>1, p, h)$	1	Fully connected graph
GF$(n>1, q=1, a=1, p, h)$	1	Fully connected graph
GF$(n>1, q>1, a=1, p, h)$	2	Galaxy graph
GF$(n>1, q=1, a>1, p, h)$	3	Dragonfly
GF$(n>1, q>1, a>1, p, h)$	5	General configurations

3 Galaxyfly Structure Analysis

In this section we analyze the structure of Galaxyfly in accordance with common properties: flexibility, bisection bandwidth, shortest paths, and resiliency. We compare our proposed Galaxyfly (GF) with Dragonfly (DF) [5], Flattened Butterfly (FB) [2], Fat Tree (FT), and Slim Fly (SF) [8]. For convenience, in the following text we will refer to these topologies using the abbreviations defined above.

3.1 Flexibility

Flexibility means the ability that the topology can be deployed in systems of different scales, e.g., from petascale to exascale and beyond. It is emerging as a demanded characteristic of efficient topologies. We discuss the flexibility of GF in this section in two aspects. (1) constructing networks of different scales with specified router radix, and (2) constructing networks of specified scale with different router radix.

By varying the (n,q,a,p) parameters, GF can be used to build networks of different scales. Using 48-port routers, Fig. 2 shows how network scale (N) varies with number of terminals per router (p) under different configurations. We choose configurations listed in Table 2. Configurations in Figs. 2a, 2b and 2c are diameters 1, 2 and 3, respectively. The diameter 5 configuration is splited into Figs. 2d and 2e. From Fig. 2 we see GF can scale from less than 1K to more than 10000K, and larger diameter configuration supports larger network scale.

Fig. 3 compares GF with other topologies using 48-radix routers. The selected FB and FT are 3-dimensional and 3-level, respectively. While DF can be configured with parameters a and p to support different network scales, we choose the configuration $a = 2p$ under which the topology is most balanced and the scalability is maximized. For GF, configuration

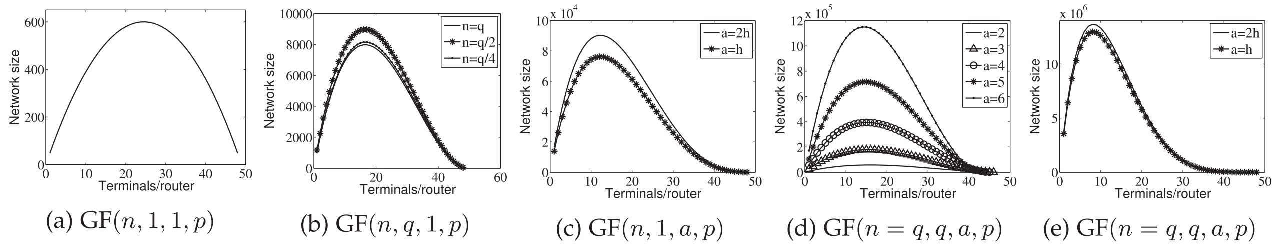
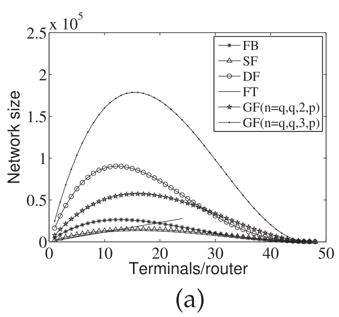


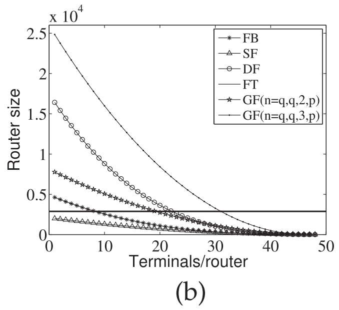
Fig. 2. The scalability of Galaxyfly in different configurations with 48-port routers.

$GF(n>1,q=n,a=2,p)$ and $GF(n>1,q=n,a=3,p)$ are chosen. With constrained router radix, although the router size of GFs are more than other topologies with the increasing terminals per router as shown in Fig. 3b, GF$(n>1,q=n,a=3,p)$ can build the largest networks as shown in Fig. 3a. There is a trade-off between router radix and router size. Given the degree and diameter, SF can achieve the largest scale. SF with constrained router radix has the worst flexibility so it builds up the smallest range of network scales. Once the router radix is fixed, the router size of 3-level FT is constant. Although FT with constrained router radix can be expanded to 100K by adding levels, it will cost a large amount of network equipments and expand exponentially. The scalabilities of FB and FT are lower than those of GF and DF. DF is the most scalable topology apart from GF, but it still cannot scale to 100K. GF can be constructed the maximal size 2458K with 48-radix routers. Besides, GF can be constructed the 100K network with the least 20-radix routers.

GF$(n,1,a,p)$ is DF in Fig. 2c. Hence, for the sake of fairness, we use the best configuration of DF with $a=2h$ compared to GF. The results show that GF can support larger scales than DF with its best configuration. Furthermore, GF can be configured to be of lower diameter than DF in small systems as shown in Fig. 2b. Figs. 2b and 2e show how a of different values affect the network size and the maximum size of the network becomes larger with increasing a. Analyses with different configurations show that with the scale of GF growing larger, the diameter increases to an upper bound of 5.



(a)



(b)

Fig. 3. The scalability of different topologies with 48-port routers.

Flexibility also indicates that a topology can be used to build up the same scale systems with various radixes. In other words, different COTS routers can be used to construct the required scale. There is no restriction of routers. Especially for GF, the same size network can be constructed with different configurations using various range of router radix (r).

3.2 Constructible Configuration

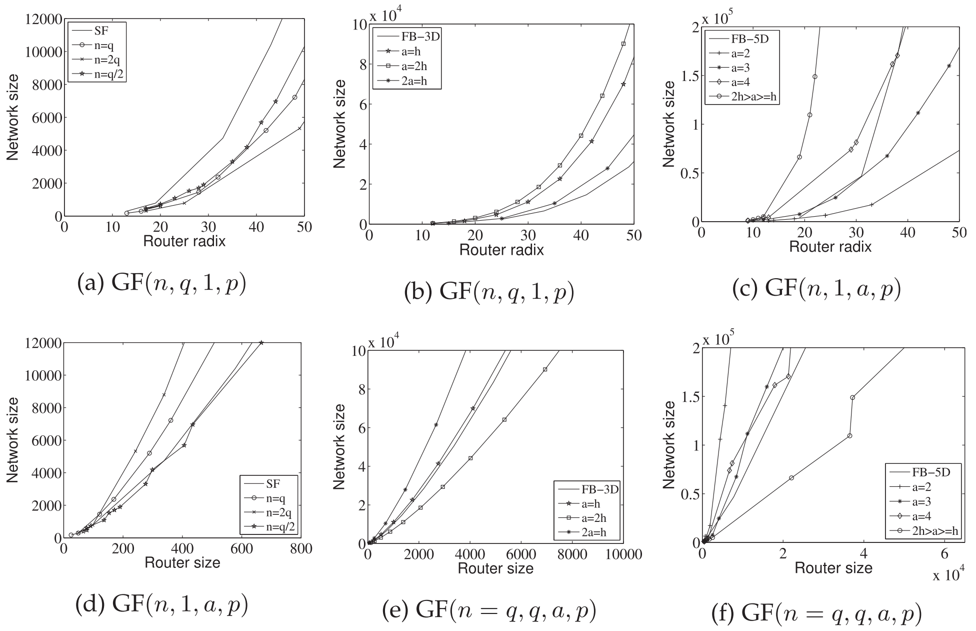
We consider a network with an constructible configuration that would ensure the performance of the network.

The constructible configuration should meet the router radix, network size, performance requirements and cost requirements. GF is a direct topology. Each router radix should follow the bound $a+h+p-1 \leq r$ and the size of network N is constrained by the bound $nqap \geq N$. With a given r and N, these bounds provide the range of possible configurations.

Bisection bandwidth is the minimum bandwidth along all bisections and is a traditional comparison metric for topologies. In order to fairly compare our design with those existing topologies, we utilize it to measure the performance of topologies. Owing to the irregular nature of GF and SF, an analytical calculation of their bisection bandwidth is difficult to represent using a formula. However, we can approximate it using the graph partitioning tool METIS [21]. Without blocking in each topology, there are at least $N/2$ bidirectional links across any bisection. In the uniform random traffic pattern, it is considered that the balanced configuration is $N/4$ bidirectional links across any bisection. We assume that the bandwidth of each link is 1. The minimal cut C_{min} is the bisection bandwidth. Thus, the ratio $\beta = (C_{min}/(N/2))$ describes the relative bisection bandwidth of the topologies. Generally, we require the minimum relative bandwidth $B \leq \beta$ for network constructions.

Our aim is to find constructible configurations that meet the constraints above. Fig. 4 shows the design space of some GFs and other topologies with the range of router radix $r \leq 48$ (the state-of-the-art high-radix routers) and $\beta \approx 0.5$. With different diameters, we can construct GF to attain different network sizes. There are five parameters n, a, q, h, and p to fix on the design space. Once n, q, and a are determined, the h is constant. Besides, if q, a, and h or n, a, and h are determined, n or q are constant.

Figs. 4a, 4b, and 4c describe the trend of N with increasing r. Correspondingly, Figs. 4d, 4e, and 4f show the increase in N with the increase in N_r. We choose different proportions between n and q for GF$(n,q,1,p)$ to analyze the trend in network size in Fig. 4a. With the same r, we can construct a larger size network in SF. However, at the same



(a) GF$(n, q, 1, p)$ (b) GF$(n, q, 1, p)$ (c) GF$(n, 1, a, p)$

(d) GF$(n, 1, a, p)$ (e) GF$(n = q, q, a, p)$ (f) GF$(n = q, q, a, p)$

Fig. 4. The global bandwidth ratio 0.5 of different topologies compared with Galaxyfly in different configurations. (a)–(c) The design space of router radix and network size. (d)–(f) The design space of router size and network size.

scale, we can utilize fewer routers constructing GF$(n = q, q, 1, p)$ and GF$(n = 2q, q, 1, p)$ than those for SF. GF$(n, 1, a, p)$ is the feasible DF. Then GF$(n, 1, a, p)$ with $a = 2h$ $(n = a^2/2 + 1)$ is the balanced DF and it is also the largest network size among other configurations of GF$(n, 1, a, p)$ and the 3-dimensional FB in Fig. 4b. With 48-radix routers, the largest size still cannot reach 100K. The 5-dimensional FB, GF$(n = q, q, 4, p)$, and GF$(n = q, q, 2h > a \geq h, p)$ with 48-radix routers can attain 100K nodes in Figs. 4c and 4f. GF$(n = q, q, 2h > a \geq h, p)$ costs more routers than other configurations and the 5-dimensional FB. However, the router radix of the GF$(n = q, q, 2h > a \geq h, p)$ is the lowest among them and it is only half of the 5-dimensional FB's size of 200K.

We also analyze the trend in different β constraints. With the higher β constraint, more routers are required to attain a specific scale. We use the same parameters a, h, and p to construct DF and GF and the size of GF can reach approximately $a^2/4$ times that of DF with $a = 2h = 2p$. However, the bisection bandwidth of GF is a little lower than that of DF. Therefore, if it lowers the demand for bisection bandwidth, we can construct larger-scale GF with the same radix routers. In Table 3, we give some instances of GF and DF with the same parameters a, h, and p. GF is a flexible topology and we can find some configurations that satisfy the network constraints. It is not only a flexible tradeoff between bisection bandwidth and network size but also a tradeoff between router size and router radix for constructing GF.

Bisection bandwidth is a theoretical metric and an upper bound of routing algorithm between two parts. Given different traffic patterns and various routing algorithms, the effective bisection bandwidth is lower than the theoretical bisection bandwidth [22]. Besides, in the fully connected graph as the local unit of GF, the bisection bandwidth is obviously $a^2/4$. If the supernode is nonblocking, it requires $ap/2$ bidirectional links under uniform random traffic pattern. If we design an constructible configuration of GF, the configuration of supernode should be taken into consideration.

In conclusion, there are at least 3 steps to select a constructible configuration. First, depending on the requirement of real systems and the restrict of network equipment, we fix the network size and router radix. Second, we take the range of diameter by the design space of network size and router radix. Third, with the requirement of bandwidth, we assure the diameter of constructible configuration and adjust the parameters of configuration. We also can take the cost, cable length, power consumption, packaging complexity and so on into consideration for reducing the design space and selecting a proper configuration.

3.3 Shortest Paths

The number of shortest paths is a significant metric for the load balance of a network. Table 4 shows the shortest paths of GF in different configurations and SF. SP is the ratio between the number of node pairs whose shortest paths are

TABLE 3
The Bisection Bandwidth of GF and DF With the Same Parameters a, h, and p

Topology	GF(n, q, a, p)	N_r	Bisection Bandwidth	β
DF	(9,1,4,2)	36	20	0.56
	(19,1,6,3)	114	94	0.55
	(33,1,8,4)	264	275	0.52
	(55,1,10,5)	550	701	0.51
	(73,1,12,6)	876	1,332	0.51
GF	(5,7,4,2)	140	44	0.31
	(9,19,6,3)	1,026	560	0.36
	(17,31,8,4)	4,216	2,350	0.28
	(30,53,10,5)	15,900	14,026	0.35
	(37,73,12,6)	32,412	36,676	0.38

TABLE 4
The Shortest Paths of Different Topologies

Topology	k'	N_r	SP	Intra-cluster	Inter-cluster
SF	25	578	0.01	0.01	0
	35	1,058	0.05	0.01	0.04
GF(n,q,1,p)	31	496	0.14	0.03	0.11
	47	992	0.13	0.01	0.12
GF(n,1,a,p)	14	510	0.17	0	0.17
	17	876	0.15	0	0.15
GF(n,q,a,p)	8	510	0.43	0.03	0.41
	8	966	0.51	0.06	0.44

more than 1 and the total number of node pairs ($N_r \times (N_r - 1)$). Intra-cluster SP is the ratio between intra-cluster node pairs and the total number of node pairs. In contrast, inter-cluster SP is for the inter-cluster node pairs whose shortest paths are more than 1. For SF, although it can use a given router radix to construct larger diameter-2 networks, the number of shortest paths is lower than other topologies in Table 4. gf$(n, q, 1, p)$ is also a diameter-2 network, its SP is higher than SF. The number of links and the router radix in GF$(n, q, 1, p)$ are more than those in SF. GF$(n, 1, a, p)$ is DF and its SP is approximate that of GF$(n, q, 1, p)$. GF(n, q, a, p) presents the best SP and uses the least radix among the topologies, but the diameter is 5. Most of the intra-cluster SP is lower than the inter-cluster SP because of the size of q. Therefore, improving the load balance of a network is to take advantage of the inter-cluster shortest paths.

3.4 Resiliency

We analyze the resiliency via simulating random link failures and the results are reflected by the average path length and diameter. Different ratios are used to remove links from the topology until the network is disconnected. Fig. 5 shows how the average path length and diameter of GFs with different configurations and others topologies increase as the link failure ratio increases. For GF, DF, and SF, the resilience level increases as Nr increases. Figs. 5a and 5b illustrate the results of GF$(n, q, 1, p)$ and SF with Nr = 0.2K–2K. Figs. 5c and 5d illustrate the results of GF$(n, q, a! = 1, p)$ and DF with Nr = 0.2K–2K. In the case of Nr = 2K, the link failure ratio can increase by up to 90 and 86 percent for GF$(n, q, 1, p)$ and SF, respectively. GF$(n, q, a! = 1, p)$ and DF can only remove up to 60 and 65 percent links with Nr = 2K, respectively. GF$(n, q, 1, p)$ and SF are superior to GF$(n, q, a! = 1, p)$ and DF at the same scale because of the shorter diameter and more links.

4 Routing

We now discuss minimal and non-minimal adaptive routing algorithms of packets from a source node directly connected to router R_s to a destination node directly connected to a different router R_d for GF .

4.1 Minimal Routing

The minimal routing algorithm (MIN) for GF will be demonstrated in two distinct situations: GF$(n, q, 1, p)$ and GF$(n, q, a \neq 1, p)$.

In the case of GF$(n, q, 1, p)$, GF is a graph of diameter 2. So a packet is transmitted either directly (R_s is connected to R_d) or routed by two hops (R_s is connected to R_d by some intermediate node R_i).

MIN for GF$(n, q, a \neq 1, p)$ is determined by the connection of source supernode G_s and destination supernode G_d. Its length is at most 5 hops.

The routing is discussed in the following text:

Step 1: If $G_s \neq G_d$ and there are no links from G_s to G_d, it will find a common neighbour G_i of G_s and G_d as the intermediate supernode. Then the packet will arrive at the router R_x in G_i via R_a and R_a in G_s. Turn to Step 2.

If $G_s \neq G_d$ and there is a link from G_s to G_d, it will employ the link at R_s or the other router R_a in the same supernode G_s. Then the packet will arrive at R_b in G_d . Turn to Step 3.

Step 2: If there is no direct link from R_x to G_d, then through the other router R_y in the same supernode G_i the packet will arrive at R_b in G_d. Turn to Step 3.

Step 3: If $R_b \neq R_d$, it will employ the local link from R_b to R_d in the same supernode G_d, then the packet will arrive at R_d.

There is a minor optimization in MIN for GF. According to the locations of G_s and G_d, MIN can be adjusted to be a balanced routing for intra-cluster routing by the number of shortest paths as shown in Table 4. If G_s and G_d are in the same cluster and not connected directly ($G_s \neq G_d$), there must exist at least one common neighbour of G_s and G_d because supernodes in the same cluster with $q \ mod \ 4 = 1$ are connected as a strong regular graph. For instance, when $q = 5$, there is one common neighbour between two unconnected supernodes in the same cluster. If $q = 13$, there is three common neighbours between two unconnected supernodes in the same cluster. Then, there are various paths between G_s to G_d and we can randomly pick one of the paths. However, the multiple shortest paths between G_s and G_d may not be the shortest paths between R_s and R_d because of the different inter-supernode connection. We should localize the inter-supernode connections of R_s and R_d and the common neighbours. Besides, if we exploit the routing tables for routing packets, we can improving the load balance by taking advantage of multiple inter-cluster shortest paths as shown in Table 4.

4.2 Non-Minimal Adaptive Routing

For a non-minimal adaptive routing algorithm on the topologies, we only explore the buffer state of the local router to select between the minimal routing algorithm and the non-minimal routing algorithm. For GF, various configurations make the network diameter vary from 2 to 5. Therefore, in order to control the hops between two terminals and reduce the number of virtual channels (VC), non-minimal adaptive routing employs a constraint of selecting a random path of at most five, six, or seven hops. We design three routing algorithms: a non-minimal adaptive random link routing algorithm (NAR), a non-minimal adaptive local link routing algorithm (NAL), and a non-minimal adaptive global link routing algorithm (NAG), which are based on the characters of GF, the valiant random routing, and Universal Globally-Adaptive Load-balanced (UGAL) algorithms [5], [8], [23], [24].

NAR is a non-minimal adaptive routing algorithm. In order to constrain the path length, NAR may select non-minimal paths only if two terminals are in two connected supernodes in the same cluster. There may be some common

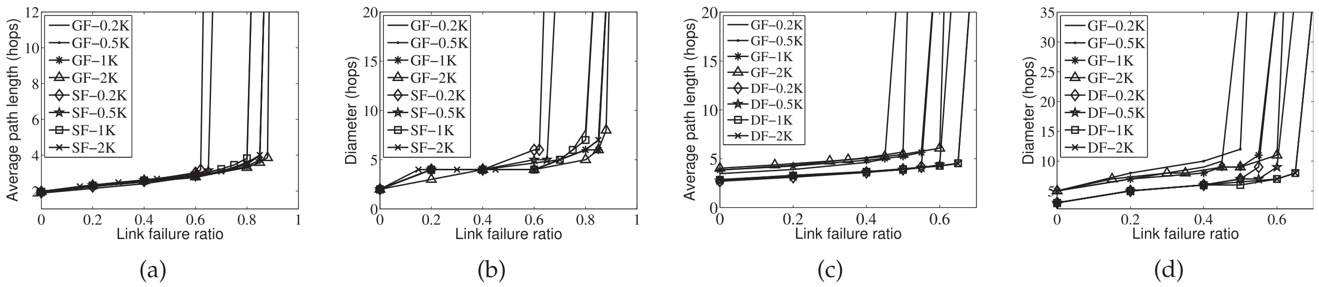


Fig. 5. With various link failure ratios in the GF$(n, q, 1, p)$ and SF. (a) Average path length. (b) Diameter. With various link failure ratios in the GF$(n, q, a \neq 1, p)$ and DF. (c) Average path length. (d) Diameter.

neighbours depending on the size of q between the two connected supernodes. NAL allows misrouting of one hop in a supernode. NAG allows misrouting to another supernode. Therefore, the constraints of NAL and NAG are six and seven hops. The longest path of NAG is through three hops between supernodes and four hops in supernodes. The selection between the non-minimal and minimal paths in NAL and NAR takes place on the source router. It is based on the occupancy of the buffer queue Q_{min} in the minimal path and Q_{nonmin} in the non-minimal path, as well as the thresholds Th_{min} and Th_{nonmin} at the current router. If $Q_{nonmin}<Th_{nonmin}$ and $Q_{min}>Th_{min}$, then the packet will be routed by the non-minimal path. The values for Th_{nonmin} and Th_{min} are calculated empirically. We set $Th_{nonmin} = 0.5 \times Q_{min}$ and $Th_{min} = 0.8\times\ buffer\ size$.

4.3 Deadlock Freedom and Avoidance

The deadlock avoidance scheme of GF is similar to the schemes of DF [5] and SF [8]. The distance-based scheme uses as many VCs per link as hops in the longest path allowed in the network. Packets are injected at the first VC with index 0, and for each hop the index of the next used VC is incremented in one unit [5], [25], [26]. The last VC never blocks because packets will be ejected. Thus, packets either advance to a higher VC in the next hop or are ejected in the current router. In GF, we use the scheme in ascending order of VCs. Owing to the partition of supernodes, we can reduce the number of VCs for each local port and each global port. We show the VCs of the deadlock-free minimal routing algorithm for GFs with different configurations in Fig. 6. For instance, in Fig. 6c, packets route across at most three local links and two global links. Therefore, it requires at least three VCs for local links and two VCs for global links to avoid cycles in MIN.

Similarly, to avoid deadlocks in NAR, NAL, and NAG, they require three, four, four VCs for local links and two, two, three VCs for global links, respectively. Besides, we can also employ rewiring, escape networks or turn models to break channel dependency for lowering the number of VCs [26], [27], [28], [29], [30]. The three methods can reduce the VCs to zero, one or two, respectively.

5 Performance

In this section we evaluate the performance of GF and other topologies on a cycle-accurate network simulator Booksim [31] using packets injected with a Bernoulli process. First, we evaluate different GFs, packet sizes, and routing algorithms under various traffic patterns. Second, we compare GFs with different high-radix networks. Finally, we take the layout into consideration and analyze blended traffic patterns specially for hierarchical high-radix networks.

Routers. We deploy traditional input-queued (IQ) routers with the entire router pipeline. Routers are based on Virtual Cut-through (VCT) switching. We assume that credit processing takes two cycles, routing computing takes two cycles, switch allocation, VC allocation, and crossbar processing each requires one cycle in routers. Input and output speedup of a crossbar is one, and the speed of internal routers is twice the channel transmission rate.

Network. By default, channels are set to one cycle transmission delay and the packet size is 1 flit. The buffer of each port has 256 flits. The buffer management policy is that each VC has 5 private flits and other flits are shared by all VCs. We compare topologies with full global bandwidth under the uniform random traffic pattern. The full global bandwidth has at least $N/4$ bidirectional links across any bisection and the bisection bandwidth ratio β is more than 0.5. The size of networks in the simulation is $N \approx 3K$ and there is at most 10 percent difference in size between various topologies. Owing to space constraints and for clarity of plots, we compare GFs with different configurations, as shown in Table 5. The configurations of other topologies are also shown in Table 5. FT is one of the most commonly used topologies in HPC systems and data center systems. FB and DF are representations of high-radix interconnection networks. SF is the newest optimal low latency interconnection network.

Routing Algorithms. We compare the following routing algorithms: Adaptive Nearest Common Ancestor (ANCA) routing for FT, and minimal and non-minimal adaptive routing algorithms by local queue size for DF [5], FB [2], SF [8], and GF.

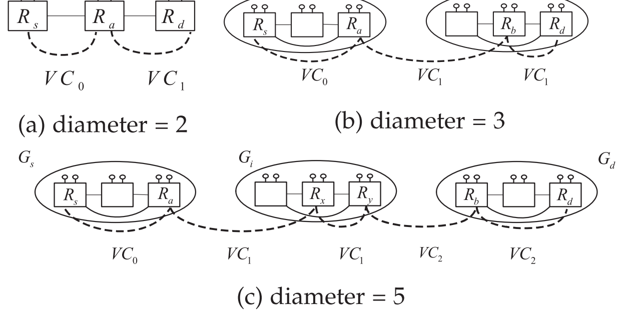


Fig. 6. Minimal routing algorithm of GF.

TABLE 5
The Configurations of GFs, FT, FB, SF, and DF

Topology	r	p	Diameter	N_r	β
GF(11,29,1,10)	34	10	2	319	0.60
GF(15,29,1,7)	35	7	2	435	1.12
GF(56,1,11,5)	20	5	3	616	0.51
GF(81,1,10,4)	21	4	3	810	1.01
GF(11,29,2,5)	18	5	4	638	0.59
GF(15,37,2,3)	20	3	4	1,110	1.13
GF(11,37,4,2)	12	2	5	1,628	0.62
GF(15,29,4,2)	12	2	5	1,740	0.68
FT	30	15	4	675	1
FB	27	6	3	512	1.33
SF	29	10	2	338	0.65
DF	20	5	3	616	0.50

Traffic Pattern. We consider several traffic patterns that are representative for typical HPC workloads. We test uniform random traffic for graph computations, sparse linear algebra solvers, and adaptive mesh refinement methods. In the uniform random traffic pattern, the source and destination of each packet are selected randomly. We also test the worst-case traffic pattern in which the source and destination are selected by the longest path in the network and some links are congested owing to high usage frequency. In FT, Packets are transmitted through the highest level routers. In DF, packets are generated between two different groups, resulting in the congestion of global links [5]. For SF, communication only occurs between two terminals in the same subgraph but different subgroups [8]. The worst-case pattern of FB is similar to the tornado pattern of the k-ary n-cube network. In GF, we arrange any two terminals in different clusters to communicate. In addition, we build a blended traffic pattern to test mixed workloads. In HPC workloads, it is important to take the communication between different node types into consideration [32], [33]. The blended traffic pattern consists of 90 percent of uniform random traffic for computing nodes and 10 percent hotspot traffic for computing nodes and I/O nodes. The computing nodes send packets to the fixed range of I/O nodes. Finally, we evaluate our topologies with a benchmark, GPCNeT [34], which proves the real-world network utilization seen on congested systems.

5.1 Different Parameters

5.1.1 Galaxyflies With Different Configurations

In Table 5, we show GFs with different configurations for the requirement of $N \approx 3K$. We classify the configurations of GFs by diameter and compare the performance of them in Fig. 7. With the same diameter, higher bisection bandwidth gives higher performance. Whichever traffic pattern it runs, uniform random traffic or worst-case traffic, GF (15,29,1,7), GF(81,1,10,4), GF(15,37,2,3), and GF(15,29,4,2) are better than GF(11,29,1,10), GF(56,1,11,5), GF(11,29,2,5), and GF(11,37,4,2). For example, in Figs. 8a and 7e, GF (15,29,1,7) outperforms GF(11,29,1,10) by about 21.5 and 55.6 percent thanks to 36.3 percent more routers and 79.5 percent more links between bisections. The shorter diameter and higher bisection bandwidth of GFs do not result in higher performance. GF(15,37,2,3) has $\beta > 1$. However, the configuration of the supernode in GF(15,37,2,3) affects the network performance. The saturation point of GF(15,37,2,3) is at an injection rate of 0.15 under uniform random traffic, only reaching 33 percent of that of GF(11,37,4,2).

5.1.2 Different Packet Sizes

We also analyze how the packet size affects the performance of GFs. In Figs. 10c and 10d, we present the results under different traffic patterns. Smaller sizes result in slightly lower latency, while bigger sizes lead to a higher saturation point, especially for GFs of lower diameter.

5.1.3 Different Routing Algorithms

For larger-scale supercomputers, traffic patterns are more mixed and complex. We introduce a blended traffic pattern to evaluate the performance in Fig. 11. Fig. 11b displays the performance of four routing algorithms in GF(15,29,4,2) under blended traffic (other GFs show similar performance) and NAG presents better performance than the other routing algorithms. NAG for GFs with different configuration can achieve about 25, 260, and 100 percent higher saturation points than MIN in Fig. 11c.

5.1.4 GPCNet

GPCNet is a topology agnostic benchmark which captures the complex workloads anticipated on multi-tenant HPC networks. It combines multiple configurable congestion patterns concurrently to simulate multiple jobs and employs concurrent canary jobs for measuring the impact of congestion. Fig. 8 shows the performance of GFs. GF(15,29,1,7) presents the best performance than other configurations and SF because of its shorter diameter and higher bandwidth. However, GF(15,29,4,2) and GF(11,37,4,2) with diameter-5 present no worse performance than diameter-3 GFs. GF(15,29,4,2) even attains 50 percent higher saturation points. It is caused by the higher router size of GF(15,29,4,2) and GF(11,37,4,2) than that of diameter-3 GFs. Besides, GF (15,29,4,2) and GF(11,37,4,2) provide more shortest paths than diameter-3 GFs. GF(15,37,2,3) and GF(11,29,2,5) present the worst performance because of the value of a.

5.2 Comparison With Other Topologies

5.2.1 Comparison With Fat Tree

We next detail the performance of GF and FT. We show that GF(15,29,1,7) can support the same level of terminals as FT at a better level of performance with only 64 percent of the routers in Figs. 9a and 9d. There are two reasons for the performance benefits of GF(15,29,1,7). One is the shorter diameter. The diameter of GF(15,29,1,7) is 2 and it mitigates the head-of-line (HOL) blocking. The long-hop routing in FT increases HOL blocking and influences network performance. The other is higher global links density. Each router in GF(15,29,1,7) provides 80 percent ports to interconnect other routers in the same cluster or other clusters. We also compare other GFs to FT. Although the number of routers is higher than that of FT, the router radix of GF(81,1,10,4) and GF(15,29,4,2) are lower than FT's. The router radix of GF (81,1,10,4) is 21 and it is 70 percent of FT's. In addition, GF

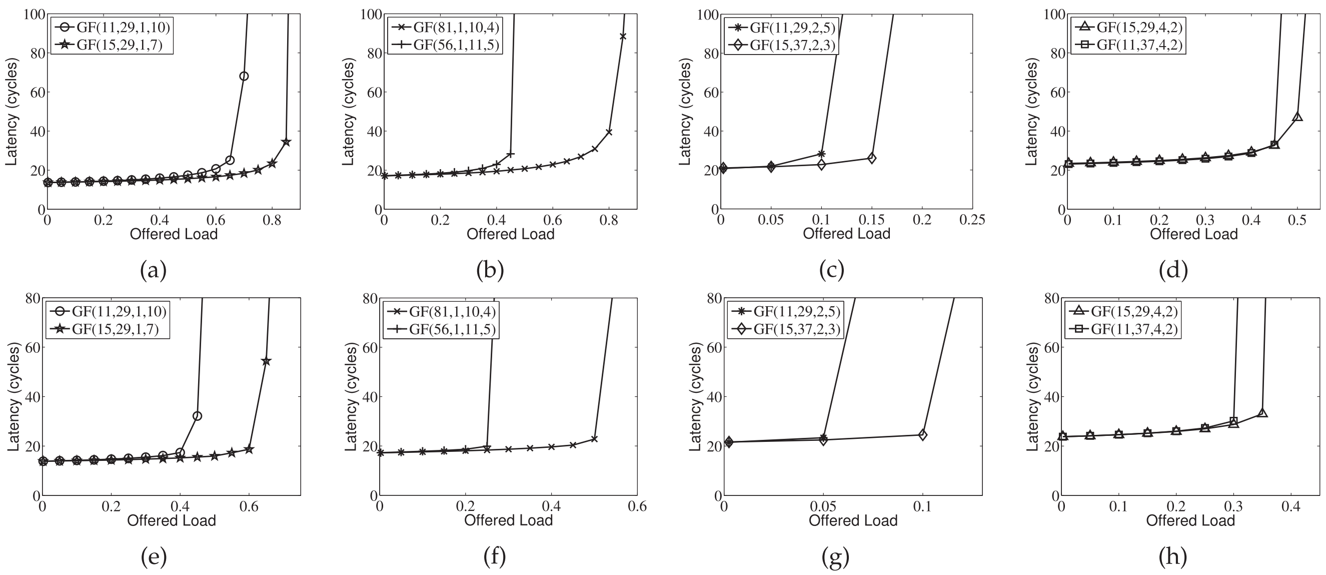


Fig. 7. Performance comparison of GFs in different parameter configurations. (a)–(d) Uniform random traffic pattern. (e)–(h) Worst-case traffic pattern.

(81,1,10,4) is diameter 3, which outperforms FT at zero load latency. GF(15,29,4,2) is diameter 5, which gets little higher than FT at zero load latency. However, the router radix of GF(15,29,4,2) is 12 and it is only 40 percent of FT′s.

5.2.2 Comparison With Flattened Butterfly

We also compare GF with FB. FB is a high-radix k-ary n-cube topology. Each dimension is fully connected. Compared to FT, FB can economize routers and links to decrease the cost. We use 3-dimensional FB to evaluate the performance in Figs. 9b and 9e. FB is similar to GF(81,1,10,4) under uniform random traffic. They have the same diameter. Although FB provides 38 percent more global links than GF (81,1,10,4) (the first dimension in FB and the supernodes in GF(81,1,10,4) are regarded as local groups), the distribution of global links in FB is concentrated in each dimension, not the whole global network. Besides, FB presents 62.5 and 70 percent lower saturation point than GF(15,29,4,2) and GF (81,1,10,4), respectively, under worst-case traffic.

5.2.3 Comparison With Slim Fly and Dragonfly

SF is a recent proposal for attaining lower latency in HPC systems. SF also leverages techniques of algebraic graphs over finite fields, like GF. However, the low diameter of SF imposes an extremely strict condition on the relationship between the radix to other routers k' and the number of routers N_r, $k' \geq \Omega(\sqrt{N_r})$. This imposes a strict limit on the scalability of Slim Fly. Comparatively, GF can be constructed by arbitrarily large-scale systems even with a fixed router radix. GF(n,q,1,p) is also a diameter-2 topology. Compared to SF, GF(n,q,1,p) with higher-radix routers provides similar bisection bandwidth at the same routers size. For example, in Table 5, GF(11,29,1,10) and SF attain the approximate β but each router in GF(11,29,1,10) requires more radixes. However, it is difficult for SF to provide the flexibility as GF. GF(15,29,1,7) is another diameter-2 configuration with higher β and higher radix. In Figs. 9c and 9f, we present the performance of SF and different GFs at the same level of terminals. GF(15,29,1,7) outperforms SF by about 42 and 140 percent higher saturation points under uniform random traffic and worst-case traffic. The better level of performance is caused by 28 percent more routers and 89 percent more global links. GF(11,29,1,10) also outperforms SF by about 8 and 60 percent under above traffics because more links of each router are connected to other routers. GF(81,1,10,4) is diameter 3. The zero load latency of GF(81,1,10,4) is slightly higher than SF. However, the saturation point of GF(81,1,10,4) is 33 and 100 percent higher than SF under the two traffic patterns. In addition, the router radix of GF(81,1,10,4) is 21, only about 75 percent of SF′s.

DF is GF(56,1,11,5). Not only is GF more flexible than DF but it also exhibits better performance than DF. GF(81,1,10,4)

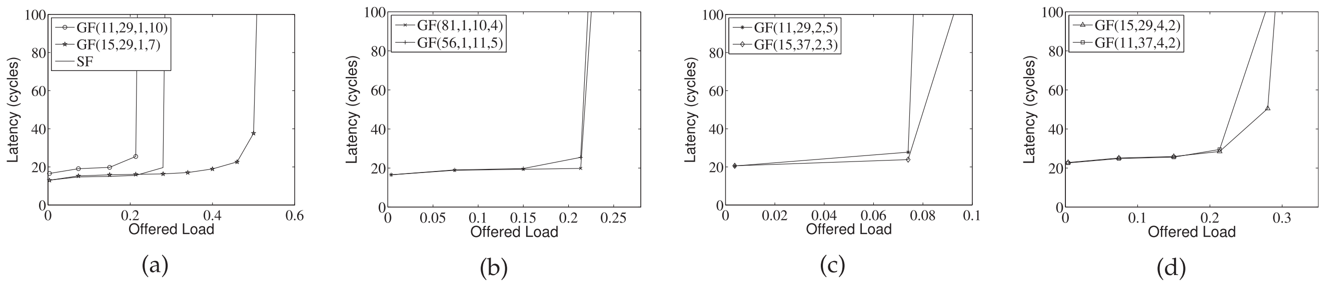


Fig. 8. Performance comparison of GFs in different parameter configurations with GPCNeT benchmark.

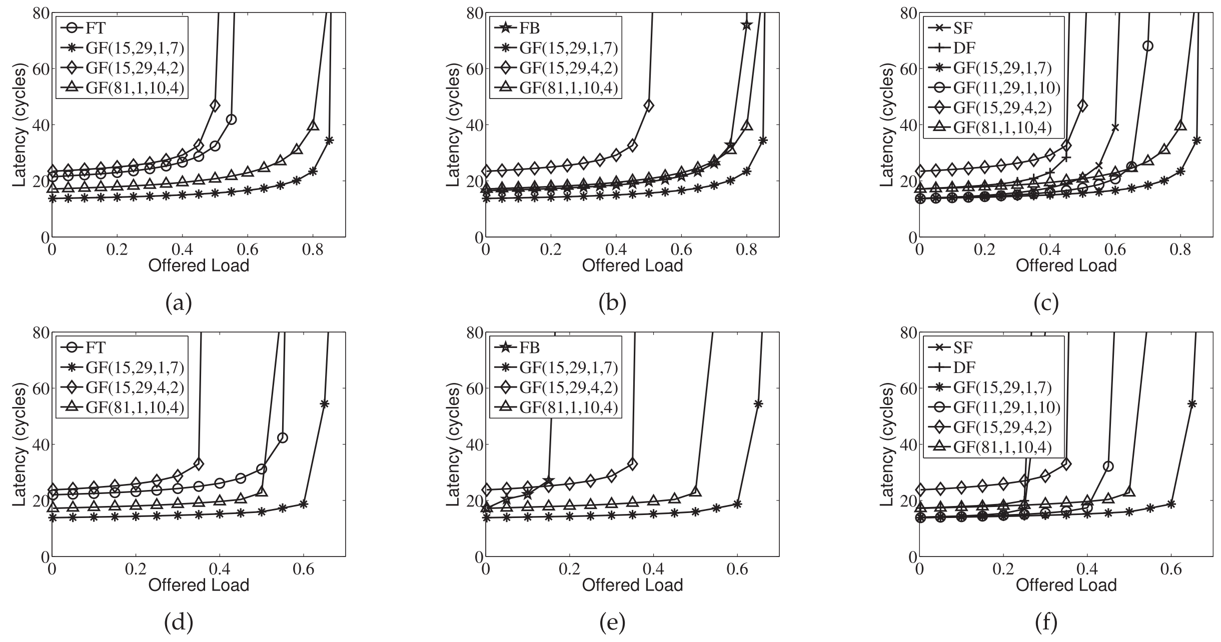


Fig. 9. Performance comparison of GFs, FT, FB, SF, and DF. (a)–(c) Uniform random traffic pattern. (d)–(f) Worst-case traffic pattern.

uses 5 percent more ports of each router than DF, however, it presents 100 percent better performance under whichever traffic pattern. Although the zero load latency of GF(15,29,4,2) is higher than DF, GF(15,29,4,2) leverages 40 percent smaller ports of each router than DF and it exhibits similar performance under uniform random traffic. The results of GF (15,29,4,2) present better performance than those of DF thanks to more global links between any two clusters in GF(15,29,4,2) under worst-case traffic.

5.3 Physical Layout

We take different link latencies into consideration, so as to compare GFs effectively with other topologies in terms of physical layout.

GF is a flexible hierarchical network, which consists of several clusters. Each cluster contains the same number of supernodes and is connected to each other cluster via the same number of cables. The module structure of GF has important implications: If the size of the supernode is too small, placing all supernodes in a cluster in close proximity (the same cabinet) enables bundling cables between every two clusters. Otherwise, placing one or more supernodes in close proximity (the same cabinet) enables bundling cables between every two cabinets. Manufacturing fiber in bundles can reduce fiber costs considerably (by nearly 40 percent) and expedite deployment of fabric [35].

SF and DF also can be considered as hierarchical networks. Their layouts are similar to that of GF. We set each cabinet containing 200–360 terminals. The connections inside the cabinet are using electric wires as local links with about 50 cycles latency. The average latency of global links between cabinets is set to 100 cycles owing to the rate of fiber cable and the frequency. The link latency contains the latency crossing a router.

We deploy SF and DF in 10 and 14 cabinets, respectively. The deployments of GF(15,29,4,2), GF(15,29,1,7), and GF (81,1,10,4) are 15, 15, and 9 cabinets. Figs. 10a and 10b present the performance of GFs, SF, and DF on a fixed layout. When comparing the results with different link latency to the results without different link latency, we find that the diameter influences the performance more than those without different link latency. The zero load latency of GF(15,29,4,2) is higher than other topologies because of the longer diameter. Compared to GF(15,29,1,7), the saturation point of GF(81,1,10,4) is the same (or better) because it provides 59.6 percent more global links between cabinets than GF(15,29,1,7). GF(15,29,1,7) and GF (81,1,10,4) outperform SF and DF in both traffic patterns. We also introduce a blended traffic pattern to evaluate the performance of GF, SF, and DF in Fig. 11. GF(15,29,1,7) not only attains the highest saturation point but also gets the lowest latency, as shown in Fig. 11a. At injection rates of around 0.25 and 0.35, the latencies of DF and GF(81,1,10,4) sharply increase because of the congestion of minimal paths.

6 Cost and Power Comparison

We now proceed to analyze two important aspects of building networks with Galaxyfly and other topologies: (1) router and cabling costs; and (2) power consumption.

We consider network topologies arranged by sets of compute nodes, routers, and cables as a 2-dimensional grid pattern. One central and practical concern for low-diameter networks is to enclose them in cabinets with minimal cabling costs. A good deployment from network topology to physical layout largely improves the system wiring.

It is easy for GF to be partitioned into some modules. We divide the routers and their attached terminals into cabinets with an approximately equal number of cables connecting the cabinets. GF can be divided into modules by clusters or supernodes. Therefore, we can exploit the proper design to limit the cost by parameters n, q, a, and p of GF.

6.1 Cost Model

We now introduce a cost model (similar to the model used in [8] and [2]) in which the overall network costs mainly

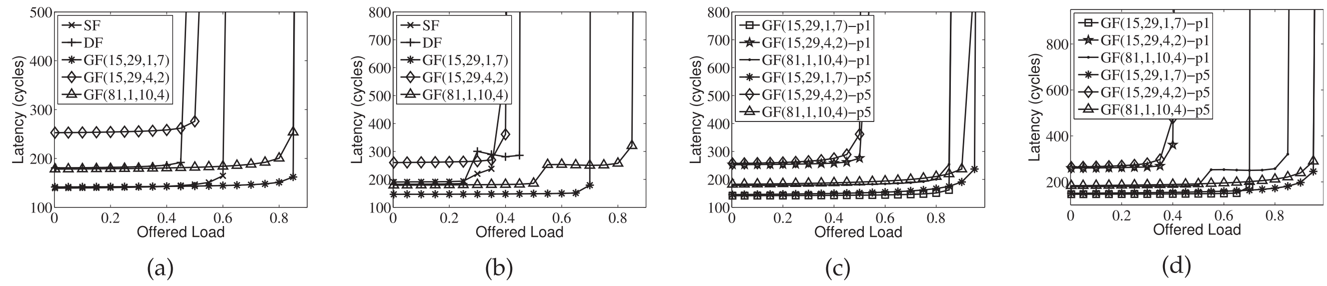


Fig. 10. Performance comparison of GFs, SF, and DF on fixed layout in uniform random (a) and worst-case (b) traffic patterns, and comparison of GF with different packet sizes in uniform random (c) and worst-case (d) traffic patterns.

constitute the cost of routers and interconnection cables. We assume that routers together with terminals are packaged in cabinets of size $1 \times 1 \times 2$ meters. Intra-cabinet cables always use electrical cables (backplane resources) while inter-cabinet cables use optical cables. In order to compute cable length, we conservatively assume that intra-cabinet cables are about 1–1.5m long on average and the lengths of inter-cabinet cables are estimated using the Manhatten distance [8] in a 3-dimensional cube. Given the number of cabinets c for balancing the cable length, we assume the layout of cabinets as a $x \times y + z$ grid. In addition, each inter-cabinet cable will be added with 2m overhead for each cabinet.

We compare GF with DF [5], Regular Random topology (RR) [16], FB (3-dimensional) [2], FT (3-level), and SF [8]. On account of restricted space, we have to omit the layouts of those topologies.

To estimate the costs of these topologies, we use the price of cables and routers [36], [37] as shown in Figs. 12a and 12b. The size $N \approx 30K$ of comparable topologies in their balanced configurations cannot be identical. The difference of the network size is controlled within 1–3 percent.

Table 6 presents the router cost of various topologies. The R/T and PpT in Table 6 represent respectively the ratio of the routers cost to the terminals cost and the power of per terminal. First, the router radix of GF(25,49,1,24) is the highest, but GF(25,49,1,24) is the most cost-effective topology among all and the router costs of DF, FT, FB, and SF are almost 36.8, 68.4, 27.1, and 2.2 percent more expensive than that of GF(25,49,1,24). The diameter of GF(25,49,1,24) is 2 and it can attain lower latency. Second, although GF (25,49,8,3) is the least cost-effective topology among all, the router radix of GF(25,49,8,3) is only 44, 25.8, 32.7, and 26.2 percent of DF′s, FT′s, FB′s, and SF′s. Besides, we can adjust the parameters of GF to meet the requirements of router cost and router radix. By adjusting the size of the supernode a in GF, GF(25,49,3,8), GF(25,49,4,6), and GF(25,49,6,4) attain lower router cost than GF(25,49,8,3) and the router radix of them are lower than those of other topologies.

In the analysis of cable cost, we propose a partition to deploy routers, cabinets, and cables for different topologies. Each cabinet contains approximately the same number of terminals as a group in DF. The results of cable cost using the method are presented in Fig. 12c. The layouts of DF, FT, FB, SF, RR, and five GFs are packaged as $13 \times 13 + 3$, $9 \times 16 + 12$, 13×13, $13 \times 12 + 6$, $13 \times 13 + 3$, and $13 \times 13 + 7$ grids. The layout of FT is different from other topologies owing to the indirect connections between some of the routers and terminals. Leaf routers and terminals are packaged in $7 \times 16 + 12$ cabinets and we have to take the connection between leaf routers and terminals into consideration. The fiber cost of FT is the highest owing to a large number of links between levels. The router radix and the layout of RR are the same as those of DF. A large number of inter-cabinet links in RR leads to a higher cable cost than those of other direct topologies. The configurations of five GFs are described in Table 6. The five GFs are packaged using the same layout and the fiber costs are equal. The difference between the five GFs is the cost of intra-cabinet connections. GF(25,49,1,24) is the most cost-effective among the five, even among all topologies, and its cable cost is 5 percent lower than SF′s.

We also can partition the network by approximately the same size of a cluster in GF. With various layouts, the cable costs of different topologies present an identical trend. GF is the most cost-effective one in a proper configuration among the topologies.

6.2 Energy Model

Network energy consumption takes up approximately 50 percent of the usage of the whole system [3]. We employ a model [3] that one port with four lanes (four SerDes)

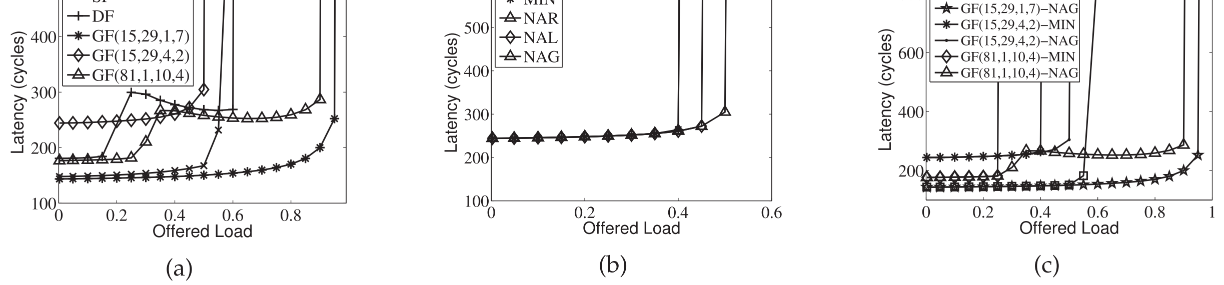


Fig. 11. Blended traffic pattern. (a) Performance comparison of GFs, SF, and DF. (b) Different routing algorithms in the GF(15,29,4,2). (c) Performance comparison of MIN and NAG in GFs.

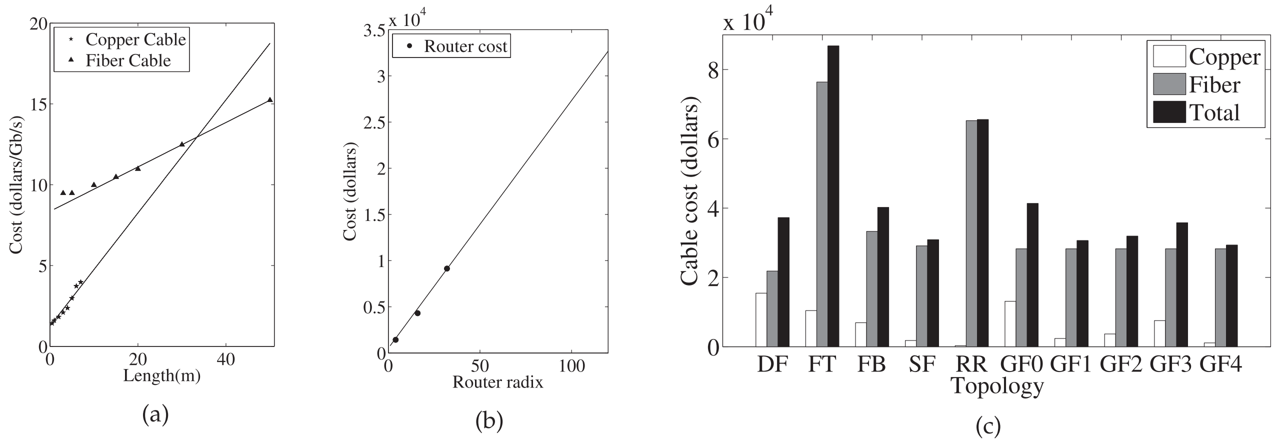


Fig. 12. Cost Model. (a) The trend of cable cost. (b) The trend of router cost model. (c) Network cable cost with $N \approx 30K$.

consumes about 2.8 watts and the network interface controller of each compute node consumes 10 watts under full use. We compare GF with other topologies in the energy model in Table 6. GF(25,49,1,24) is the lowest one among all the topologies owing to the smaller number of routers.

7 Related Work

In this paper, some topologies are summarized and compared with Galaxyfly, such as Fat Tree, Flattened Butterfly [2], Dragonfly [5], and Slim Fly [8]. Fat Tree is an indirect network and is popular among HPC systems and data center systems. Although Fat Tree can be expanded by adding levels with lower radix routers and provide abundant path diversity and high bandwidth, physical cost and power consumption are bottlenecks to its usage. Some topologies have been proposed to try to solve the bottlenecks, such as a new n-dimensional hybrid topology [38]. Flattened Butterfly and Dragonfly take advantage of high-radix routers to achieve higher bandwidth with lower cost and power consumption. Slim Fly is an approximately optimal network with fixed router radix and diameter of 2. However, once the router radix is fixed, the scale of Flattened Butterfly, Dragonfly, and Slim Fly are limited and it is restricted to constructing an exascale network of 100K size by using these topologies and the current COTS high-radix routers. Regular Random graph [16] supports arbitrary scale networks with moderately high radixes, but the network performance and physical cost are undetermined. Besides, it is impossible for COTS routers to provide enough space for a routing table used in regular random graph. Dragonfly+ [39] is proposed to extend conventional Dragonfly by Clos-like topology inside the group and supports similar or better performance. With the same router radix, the size of Dragonfly+ can attain approximately constant times of Dragonfly. Given the same network size, although the router radix in Dragonfly+ can be cut down to approximately three quarters of that in Dragonfly, the number of routers is nearly doubled as shown in Fig. 13.

Apart from HPC systems, some topologies in data center systems are varied and worthy of usage in HPC systems.

Xpander [40] achieves the same level of performance as random networks. Moreover, Xpander has inherently better structure and order, which makes wiring Xpander manageable. However, the connections in Xpander are uncertain, which leads to the similar routing table problems as random networks. Dcell [41] and Bcube [42] are both hierarchical networks. Dcell is a recursively defined structure and scales doubly exponentially as the node degree increases. Bcube is a server-centric network structure and the number of server ports increases as the network scales. Therefore, Dcell and Bcube cannot preferably support flexible scalability. Some configurable networks have been proposed for the requirement of applications and scalability, via reconfiguring the optical circuit switch states [43], adding free-space optical links [44] or wireless links [45]. However, the cost of equipment and surroundings is higher than those for other networks.

In order to lower the complexity of real construction, many researchers focus on the problems of wiring, placement, rewiring and so on. A unified framework of three-layer topology model is proposed for lowering the complexity of topology design [46]. FatClique combines the hierarchical structure in Clos topology with the edge expansion in expander graphs to lower the complexity of life cycle management [47]. The minimal rewiring of Clos topology achieves fine-grained expansions [48]. The three work are

TABLE 6
The Cost and Energy Model of GFs and Other Topologies

Topology	DF	FT	FB	SF	GF (25, 49 ,8,3)	GF (25,49 ,3,8)	GF (25,49 ,4,6)	GF (25,49 ,6,4)	GF (25,49 ,1,24)
Terminals	29,412	29,791	28,561	29,160	29,400	29,400	29,400	29,400	29,400
Routers	3,268	2,883	2,197	1,458	9,800	3,675	4,900	7,350	1,225
Radix	36	62	49	61	16	26	21	17	72
R/T (dollars)	1,130	1,391	1,050	844	1,602	936	1,025	1,269	826
PpT(W)	21	24	21	19	25	19	20	22	18

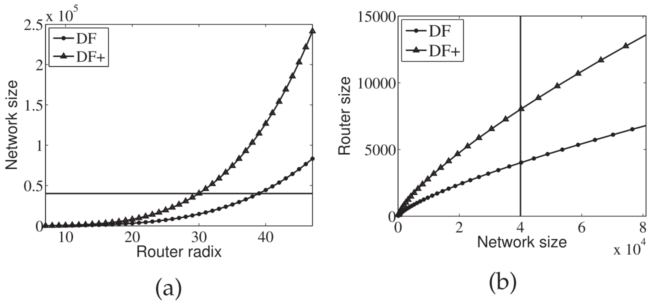


Fig. 13. The scalability of Dragonfly and Dragonfly+.

based on the indirect Clos-like topologies and there are more network equipments than the direct topologies.

8 Conclusion

High-radix interconnection network is the current and upcoming solution for larger-scale HPC systems. However, the development of high-radix routers is constrained by the state-of-the-art technology, such as the complexity of crossbar arbitration, the area of SerDes, and power consumption. In this paper, we have proposed to explore the Galaxyfly family to build flexible-scale low-diameter interconnection networks. Galaxyfly, which not only enhances the flexibility of the topology but also works out the construction of HPC systems from petascale to exascale and beyond with moderately high radixes. Furthermore, we construct Galaxyfly which makes a compromise between network performance, physical layout complexity, cost, and power consumption.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and Xing Su for his modification to the graphs, Ke Wu and Changhong Wang for their help to the simulation. Besides, we gratefully acknowledge members of Tianhe interconnect group at NUDT for many inspiring conversations. This work was supported in part by National Key R&D Program of China under Grant 2018YFB0204300 and Excellent Youth Foundation of Hunan Province (Dezun Dong).

References

[1] 2017. [Online]. Available: https://www.nextplatform.com/wp-content/uploads/2017/07/Leading-the-charge-to-exascale-computing.pdf
[2] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: A cost-efficient topology for high-radix networks," in *Proc. 34th Annu. Int. Symp. Comput. Architect.*, 2007, pp. 126–137.
[3] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *Proc. 37th Annu. Int. Symp. Comput. Architect.*, 2010, pp. 338–347.
[4] A. Putnam *et al.*, "A reconfigurable fabric for accelerating large-scale datacenter services," in *Proc. ACM/IEEE 41st Int. Symp. Comput. Architect.*, 2014, pp. 13–24.
[5] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Proc. Int. Symp. Comput. Architect.*, 2008, pp. 77–88.
[6] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, "HyperX: Topology, routing, and packaging of efficient large-scale networks," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2009, pp. 41:1–41:11.
[7] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova, "Skywalk: A topology for HPC networks with low-delay switches," in *Proc. IEEE 28th Int. Parallel Distrib. Process. Symp.*, 2014, pp. 263–272.
[8] M. Besta and T. Hoefler, "Slim fly: A cost effective low-diameter network topology," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2014, pp. 348–359.
[9] S. B. Mark *et al.*, "Intel omni-path architecture: Enabling scalable, high performance fabrics," in *Proc. IEEE 23rd Annu. Symp. High-Perform. Interconnects*, 2015, pp. 402–414.
[10] N. Binkert *et al.*, "The role of optics in future high radix switch design," *ACM SIGARCH Comput. Architecture News*, vol. 39, no. 3, pp. 437–448, 2011.
[11] T. P. Morgan, "The road to 200g networks starts with the transceiver," 2016. [Online]. Available: http://www.nextplatform.com
[12] 2020. [Online]. Available: https://www.mellanox.com/sites/default/files/doc-2020/pb-quantum-hdr-switch-silicon.pdf
[13] 2012. [Online]. Available: http://www.avagotech.com/optical_fpga#
[14] L. Xiangke *et al.*, "High performance interconnect network for tianhe system," *J. Comput. Sci. Technol.*, vol. 30, no. 2, pp. 259–272, 2015.
[15] G. Faanes, A. Bataineh *et al.*, "Cray cascade: A scalable HPC system based on a dragonfly network," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2012, pp. 103:1–103:9.
[16] M. Koibuchi *et al.*, "A case for random shortcut topologies for HPC interconnects," *ACM SIGARCH Comput. Architecture News*, vol. 40, no. 3, pp. 177–188, 2012.
[17] M. Koibuchi, I. Fujiwara, H. Matsutani, and H. Casanova, "Layout-conscious random topologies for HPC off-chip interconnects," in *Proc. IEEE 19th Int. Symp. High-Perform. Comput. Architect.*, 2013, pp. 484–495.
[18] S. Ankit *et al.*, "Jellyfish: Networking data centers randomly," in *Proc. 9th USENIX Symp. Netw. Syst. Des. Implementation*, 2012, pp. 225–238.
[19] B. D. McKay, M. Miller, and J. Siran, "A note on large graphs of diameter two and given maximum degree," *J. Combinatorial Theory*, vol. 74, no. 1, pp. 110–118, 1998.
[20] P. R. Hafner, "Geometric realisation of the graphs of McKay–Miller–Širáň," *J. Combinatorial Theory*, vol. 90, no. 2, pp. 223–232, 2004.
[21] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998.
[22] T. Hoefler, T. Schneider, and A. Lumsdaine, "Multistage switches are not crossbars: Effects of static routing in high-performance networks," in *Proc. IEEE Int. Conf. Cluster Comput.*, 2008, pp. 116–125.
[23] A. Singh, "Load-balanced routing in interconnection networks," PhD dissertation, Dept. Electr. Eng., Stanford Univ., Stanford, CA, 2005.
[24] G. Kathareios, C. Minkenberg, B. Prisacari, G. Rodriguez, and T. Hoefler, "Cost-effective diameter-two topologies: Analysis and evaluation," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2015, pp. 36:1–36:11.
[25] K. Gunther, "Prevention of deadlocks in packet-switched data transport systems," *IEEE Trans. Commun.*, vol. COM-29, no. 4, pp. 512–524, Apr. 1981.
[26] M. García, E. Vallejo, R. Beivide, M. Odriozola, and M. Valero, "Efficient routing mechanisms for dragonfly networks," in *Proc. 42nd Int. Conf. Parallel Process.*, 2013, pp. 582–592.
[27] D. Xiang and X. Liu, "Deadlock-free broadcast routing in dragonfly networks without virtual channels," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2520–2532, Sep. 2016.
[28] D. Xiang, B. Li, and Y. Fu, "Fault-tolerant adaptive routing in dragonfly networks," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 259–271, Mar./Apr. 2019.
[29] M. Garcia *et al.*, "On-the-fly adaptive routing in high-radix hierarchical networks," in *Proc. 41st Int. Conf. Parallel Process.*, 2012, pp. 279–288.
[30] M. Garcia, E. Vallejo, R. Beivide, M. Valero and G. Rodríguez, "OFAR-CM: Efficient dragonfly networks with simple congestion management," in *Proc. IEEE 21st Annu. Symp. High-Perform. Interconnects*, 2013, pp. 55–62.
[31] J. Nan *et al.*, "A detailed and flexible cycle-accurate network-on-chip simulator," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2013, pp. 86–96.
[32] M. Mubarak *et al.*, "Quantifying I/O and communication traffic interference on dragonfly networks equipped with burst buffers," in *Proc. IEEE Int. Conf. Cluster Comput.*, 2017, pp. 204–215.

[33] J. Gliksberg, J.-N. Quintin, and P. J. Garcia, "Node-type-based load-balancing routing for parallel generalized fat-trees," in *Proc. Int. Workshop High-Perform. Interconnection Netw. Exascale Big-Data Era*, 2018, pp. 9–15.

[34] S. Chunduri *et al.*, "GPCNeT: Designing a benchmark suite for inducing and measuring contention in HPC networks," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2019, Art. no. 42.

[35] A. Singh *et al.*, "Jupiter rising: A decade of clos topologies and centralized control in Google's datacenter network," *Commun. ACM*, vol. 59, no. 9, pp. 88–97, 2016.

[36] 2018. [Online]. Available: https://www.colfaxdirect.com/store/pc/viewCategories.asp?idCategory=2

[37] 2018. [Online]. Available: https://www.colfaxdirect.com/store/pc/viewCategories.asp?idCategory=7

[38] R. Peñaranda, C. Gómez, M. E. Gómez, P. López, and J. Duato, "A new family of hybrid topologies for large-scale interconnection networks," in *Proc. IEEE 11th Int. Symp. Netw. Comput. Appl.*, 2012, pp. 220–227.

[39] A. Shpiner, Z. Haramaty, S. Eliad, V. Zdornov, B. Gafni, and E. Zahavi, "Dragonfly+: Low cost topology for scaling datacenters," in *Proc. IEEE 3rd Int. Workshop High-Perform. Interconnection Netw. Exascale Big-Data Era*, 2017, pp. 1–8.

[40] A. Valadarsky *et al.*, "Xpander: Towards optimal-performance datacenters," in *Proc. 12th Int. Conf. Emerg. Netw. EXperiments Technol.*, 2016, pp. 205–219.

[41] C. Guo *et al.*, "DCell: A scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 75–86, 2008.

[42] C. Guo *et al.*, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. Int. Conf. Appl. Technol. Architectures Protocols Comput. Commun.*, 2009, pp. 63–74.

[43] M. Y. Teh, Z. Wu, and K. Bergman, "Flexspander: Augmenting expander networks in high-performance systems with optical bandwidth steering," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 12, no. 4, pp. B44–B54, 2020.

[44] I. Fujiwara, M. Koibuchi, T. Ozaki, H. Matsutani, and H. Casanova, "Augmenting low-latency HPC network with free-space optical links," in *Proc. IEEE 21st Int. Symp. High-Perform. Comput. Architecture*, 2015, pp. 390–401.

[45] N. Hamedazimi *et al.*, "FireFly: A reconfigurable wireless data center fabric using free-space optics," in *Proc. Int. Conf. Appl. Technol. Architectures Protocols Comput. Commun.*, 2014, pp. 319–330.

[46] Y. Chang, X. Huang, L. Deng, Z. Shao, and J. Zhang, "Systematic topology design for large-scale networks: A unified framework," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 347–356.

[47] M. Zhang *et al.*, "Understanding lifecycle management complexity of datacenter topologies," in *Proc. 16th USENIX Symp. Netw. Syst. Des. Implementation*, 2019, pp. 235–254.

[48] S. Zhao *et al.*, "Minimal rewiring: Efficient live expansion for clos data center networks," in *Proc. 16th USENIX Symp. Netw. Syst. Des. Implementation*, 2019, pp. 221–234.

Fei Lei received the BS degree from Tongji University, China, in 2011, and the MS and PhD degrees from the National University of Defense Technology (NUDT), China, in 2013 and 2018, respectively. Her research interests include high-performance networking and architecture for parallel and distributed systems.

Dezun Dong received the BS, MS, and PhD degrees from the National University of Defense Technology (NUDT), Changsha, in 2002, 2004, and 2010, respectively. He is currently a professor with the College of Computer, NUDT, where he leads the research group of high-performance network and architecture (HiNA) and serves as the deputy director designer of Tianhe supercomputer. His research interests include high-performance network and architecture for supercomputer, datacenter and deep learning systems. He has published more than 70 peer-reviewed papers in reputed international journals and conferences.

Xiangke Liao received the BS degree from Tsinghua University, China, in 1985, and the MS degree from the National University of Defense Technology (NUDT), China, in 1988, both in computer science. He is currently a professor with the College of Computer, NUDT. His research interests include high-performance computing systems, operating systems, and parallel and distributed computing. He is the principle investigator and chief designer of Tianhe-2 supercomputer.