Galaxyfly: A Novel Family of Flexible-Radix Low-Diameter Topologies for Large-Scales Interconnection Networks

Fei Lei, Dezun Dong*, Xiangke Liao, Xing Su and Cunlu Li
National Laboratory for Parallel and Distributed Processing
Collaborative Innovation Center of High Performance Computing
College of Computer, National University of Defense Technology
Changsha 410073, China
leifei,dong,xkliao,xingsu,cunluli@nudt.edu.cn

ABSTRACT

Interconnection network plays an essential role in the architecture of large-scale high performance computing (HPC) systems. In the paper, we construct a novel family of low-diameter topologies, Galaxyfly, using techniques of algebraic graphs over finite fields. Galaxyfly is guaranteed to retain a small constant diameter while achieving a flexible tradeoff between network scale and bisection bandwidth. Galaxyfly lowers the demands for high radix of network routers and is able to utilize routers with merely moderate radix to build exascale interconnection networks. We present effective congestion-aware routing algorithms for Galaxyfly by exploring its algebraic property. We conduct extensive simulations and analysis to evaluate the performance, cost and power consumption of Galaxyfly against state-of-the-art topologies. The results show that our design achieves better performance than most existing topologies under various routing algorithms and traffic patterns, and is cost-effective to deploy for exascale HPC systems.

CCS Concepts

•**Computer systems organization** → **Architectures;** *Parallel architectures;* Interconnection architectures;

Keywords

high performance computing, interconnection, topology, flexible-radix, low-diameter

1. INTRODUCTION

Large-scale supercomputers currently have tens of thousands of compute nodes, e.g. Tianhe-2 system and IBM Blue Gene/Q network, and strive to progress towards larger size. Exascale system would require hundreds of thousands of interconnected processors. Interconnection network, being essential to the scalability of high performance computing (HPC) systems, impacts 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 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 being subjected to strict constraints, e.g. 20 MW power envelope for exascale systems. Building cost and power consumption of network can attain 33% [19] and 50% [1] of the whole system, respectively. Last but not least, flexibility is emerging to be a demanding characteristic of efficient topology. According to an application-driven perspective on HPC system designs [27], interconnect systems are expected to be adaptable to applications of varied scales and communication requirements through static deployment or dynamic reconfiguration.

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. Dragonfly [20], HyperX [2], Skywalk [8], Slim Fly [4]. Those 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 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 currently are the largest building block available and will be used by Argonne, Cray and Intel to build a DOE's 180 petaflops HPC system, Aurora, in 2018. Chip physical resources and power consumption are the two key limits to boosting the radix [5]. In electronic router chips, increasing radix while maintaining or increasing per-port bandwidth is limited by the wiring complexity of switching crossbar, high bandwidth density at chip edge [5] and high speed SerDes I/O. Many high-radix router models are proposed to solve the challenges, like 64-radix tile-based YARC switch [29],

*corresponding author

ICS '16, June 01-03, 2016, Istanbul, Turkey

DOI: http://dx.doi.org/10.1145/2925426.2926275

64-radix 2D Swizzle-switch [28] and 64-radix 3D Hi-Rise switch [18], 136-port SCOC [6]. However, there is still a long way for them to be COTS routers. For example, the bandwidth per port of SCOC router is 25Gbps [6], much less than that of current COTS routers, i.e., 48-port Intel Omni-Path switch and 36-port Mellanox EDR InfiniBand switch. Furthermore, 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. It becomes a great challenge to build a high-radix router that increases the radix and port bandwidth simultaneously. Silicon nanophoronics technology is also expected to solve these problems. Altera and Avago have made an optical FPGA with the concept of embedded parallel optical modules[14]. However, it is still difficult to accommodate and integrate much optical modules in one chip. The growth of radix will assume bigger power budget for a router chip. For example, one directional SerDes link transmitting a bit costs approximately 20pJ in Tianhe-2 system, and SerDes consumes 79% the overall power budget for a router chip[30]. In addition, floorplan and thermal constraints for physical package design will radically impact on the number of high-speed SerDes transceivers and router radix density [25]. Hence, interconnection design for exascale systems and beyond may have to face the issue of scaling down of router radix.

Recently, most of existing work is focusing on high-radix topologies, such as Fat Tree, Flattened Butterfly [19], Dragonfly [20], HyperX [2], Skywalk [8], Slim Fly [4]. High-radix topologies can attain lower diameter than low-radix topologies like 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 consuming a large amount of cables and routers. Dragonfly [20] not only achieves low diameter but also combines the advantages of long optical channels and short electrical channels for reducing the cost. But it is still intractable to build an exascale network of 100K size by using Dragonfly and the newest COTS high-radix routers, i.e., 48-port Intel Omni-Path switch [23]. Indeed, the scalability issue of Dragonfly has limits its deployment in real HPC systems. For instance, Cray Cascade system [7] employs a variation of Dragonfly, that is, the intra-group is 2-D Flattened Butterfly [19] instead of fully connected graph to improve the scalability of Dragonfly. HyperX [2] is a flexible topology via adjusting various parameters and every dimension is fully connected thus 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 lead to reducing the growth of network scale and increasing the network costs. Random topology [22, 21, 3] provides various design spaces and satisfies kinds of requirements like incremental expansion. But routing algorithm and link congestion affect the network performance. Slim Fly explores a fancy graph theory that aims to be close to Moore bound. The goal of Slim Fly is to construct the largest scale, given the parameters of degree and diameter. But the scalability of Slim Fly is limited by its low diameter and router radix. Once the growth of router radix stagnate, we have to reexamine the design trade-off in high-radix topology between node degree and network diameter, as well as between performance and scalability.

This paper aims at constructing exascale (and even beyond) interconnection networks using routers with moderately high radixes. We make the initial attempt to build 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. The further goal is to improve the flexibility of topology, in the means of that the topology can adjust scale and bisection bandwidth such that it can be deployed in systems of different scales, from petascale to exascale systems and beyond. In this paper, we introduce a novel family of low-diameter topologies, Galaxyfly, utilizing techniques of algebraic graphs over finite fields. Galaxyfly lowers the demands for high radix of network routers to expand flexibly with different configurations. And it retains a small constant diameter while achieving a flexible tradeoff between network scale and bisection bandwidth. Furthermore, by exploring the algebraic property 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 can make it have the potential to support different traffic characteristics, i.e., using fully connection to absorb the dominating local traffics, and using redundant global links to adaptively serve global traffics. 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 and bisection bandwidth.
- Taking advantage of Galaxy graph, we design minimal routing algorithm and non-minimal adaptive routing algorithm for Galaxyfly and compare different configurations with other classical topologies in various workloads.
- Compared to other classical topologies, Galaxyfly can achieve improvements in both cost model and energy model owing to the flexible layout.

The remainder of the paper is organized as follows. We describe Galaxy graph and Galaxyfly in details in Section 2 and identify the characteristics of Galaxyfly in Section 3. Section 4 demonsrates 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 layout and consumption of Galaxyfly in practice. Related work is discussed in Section 7 and followed by conclusions.

2. GALAXYFLY TOPOLOGY

In this section, we present a flexible and configurable topology, Galaxy graph, which is derived from algebraic graphs

n	Number of clusters in Galaxy graph
q	Number of supernodes in a cluster in Galaxy graph
δ	q *mod* 4

Table 1: Parameters of Galaxy graph

and whose diameter is guaranteed to be at most 2. Based on Galaxy graph, we design a novel family of low-diameter topologies, Galaxyfly, which lowers the demands for router radix to expand flexibly and achieves a flexible tradeoff between network scale and bisection bandwidth.

2.1 Galaxy Graph

Our purpose is to use fixed router radix to construct larger networks and relax the requirement of the number of routers and diameter, which is partially a dual problem of classical Moore graph. Similar to Slim Fly [4], we adopt graph techniques introduced by McKay-Miller-Siran in [24], to construct the basic blocks of 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. It can be proved that the diameter of Galaxy graph is at most 2 through mathematical induction, which is similar to that of McKay-Miller-Siran in [24]. In the remainder of this subsection, we will show the construction of Galaxy graph in a step-by-step way.

Galaxy graph is a size nq graph $G(V, E)$ and is equally divided into n clusters, $V(G) = V_0 \cup V_1 \cup \cdots \cup V_{n-1}$, with q nodes in each cluster. The parameters used in constructing Galaxy graph are shown in Table 1.

Before constructing Galaxy graph, it is necessary to introduce some basis of algebraic graphs over finite field. A few concepts, lemma and definitions are presented.

A prime power q generates a finite field $\mathbb{F}_q$. We can find a primitive element of $\mathbb{F}_q$, ξ, which belongs to $\mathbb{F}_q$ and generates $\mathbb{F}_q$ in forms of $\{\xi^t mod\ q | t \in \mathbb{N}\}$. Two generator sets, X and X', can be constructed from ξ as shown in equation (1) and (2)

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

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

After introducing the basic concepts above, we give the definition of generator graph and a lemma on it. Note all arithmetic operations are performed in a modulo q manner.

DEFINITION 2.1. *Let $\mathbb{F}_q$ be a finite field with primitive element ξ, and X be a generator set constructed from ξ, a **generator graph** of X is a q node graph with nodes labeled $0, 1, \cdots, q-1$, and there exists an edge between node i and j if and only if $i - j \in X$.*

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.[11]*

Algorithm 1 Computing the Coordinate Matrix

Input: Generator sets X, X'
Output: Coordinate matrix of cluster C_k, M_k
1: $M_k \leftarrow \emptyset$
2: **for** i in $0, \cdots, n-1$ **do**
3: **if** $i \geq k$ **then**
4: $col_i^k \leftarrow [0, 1, \cdots, q-1]$
5: **else**
6: $col_i^k \leftarrow \tilde{H}_{X \to X'}([0, 1, \cdots, q-1], X, X')$
7: **end if**
8: $M_k \leftarrow M_k \cup col_i^k$
9: **end for**

The isomorphic function [13] is also used in the construction of Galaxy graph.

DEFINITION 2.2. *The **isomorphic function** $H_{X \to X'}$ is a function defined on $\{0, 1, \cdots, q-1\}$ such that $H_{X \to X'}(i) = j$ if and only if node i in G corresponds to node j in G'. The **generalized isomorphic function** $\tilde{H}_{X \to X'}$ is a function defined on $\{0, 1, \cdots, q-1\}^q$ such that $\tilde{H}_{X \to X'}(\vec{u}) = \vec{v}$ if and only if $H_{X \to X'}(u_i) = v_i$, $\forall i \in \{0, 1, \cdots, q-1\}$.*

Each cluster C_k in Galaxy graph is $G(V_k)$, $0 \leq k < n$. The ith node v_i^k of the kth cluster C_k is represented by coordinate vector $(a_{i,0}^k a_{i,1}^k \cdots a_{i,n-1}^k)$, $0 \leq i < q$, $0 \leq k < n$. We associate q nodes of each cluster C_k in Galaxy graph with a coordinate matrix M_k of size $q \times n$. The coordinate matrix M_k is computed using Algorithm 1. Each column $col_i^k = [a_{0i}^k, a_{1i}^k, \cdots, a_{q-1,i}^k]^T$ is a permutation of $\{0, 1, \cdots, q-1\}$, $0 \leq i < n$. The input is the generator sets X and X', and the output is the coordinate matrix M_k of the kth cluster C_k. First, an uninitialized matrix is created (line 1), then the first k columns is given the value $\tilde{H}_{X \to X'}([0, 1, \cdots, q-1]^T)$ (line 5-6), and the other columns are give the value $[0, 1, \cdots, q-1]^T$ (line 3-4).

$$M_k = \begin{bmatrix} a_{0,0}^k & a_{0,1}^k & \cdots & a_{0,n-1}^k \\ a_{1,0}^k & a_{1,1}^k & \cdots & a_{1,n-1}^k \\ \vdots & \vdots & \ddots & \vdots \\ a_{q-1,0}^k & a_{q-1,1}^k & \cdots & a_{q-1,n-1}^k \end{bmatrix}$$

Using the coordinate matrices, two algorithms are used to create edges in Galaxy graph, one for intra-cluster edges and the other for inter-cluster edges. Algorithm 2 shows how intra-cluster edges E_k in $G(V_k)$ are created, $0 \leq k < n$. We take the kth column col_k^k of M_k, and perform subtraction operation for each two distinct elements in col_k^k (line 1-2), and if the result falls into generator set X, we create an edge connecting the two nodes (line 4).

Algorithm 3 shows how inter-cluster edges $E_{i,j}$ between $G(V_i)$ and $G(V_j)$ are created, $0 \leq i, j < n$. We take the jth column col_j^i of M_i and the ith column col_i^j of M_j (line 3). Elements of the two columns are compared in all-to-all manner (line 1-2), and an edge is created if an equal value pair is encountered. By the computation process shown in Algorithm 1, col_i^j and col_j^i are permutations of each other, so exactly q edges are created between each two distinct clusters C_i and C_j.

Algorithm 2 Create Intra-cluster Edges

Input: M_k, X
Output: Edges of $G(V_k)$, E_k,
1: $E_k \leftarrow \emptyset$
2: **for** i in $0, \cdots, q-1$ **do**
3: **for** j in $i+1, \cdots, q-1$ **do**
4: **if** $M_k[j][k] - M_k[i][k] \in X$ **then**
5: $E_k \leftarrow E_k \cup \{(v_j^k, v_i^k\}$
6: **end if**
7: **end for**
8: **end for**

Algorithm 3 Create Inter-cluster Edges

Input: M_i, M_j
Output: Edges between $G(V_i)$ and $G(V_j)$, $E_{i,j}$
1: $E_{i,j} \leftarrow \emptyset$
2: **for** r in $0, 1, \cdots, q-1$ **do**
3: **for** s in $0, 1, \cdots, q-1$ **do**
4: **if** $M_i[r][j] = M_j[s][i]$ **then**
5: $E_{i,j} \leftarrow E_{i,j} \cup \{(v_r^i, v_s^j)\}$
6: **end if**
7: **end for**
8: **end for**

Finally, we give a concrete example by step-by-step constructing a Galaxy graph with $n = 3$ and $q = 5$.

1. The construction of Galaxy graph begins with choosing $n = 3$, $q = 5$. $q = 5$ is a prime power such that $q = 4\ell + \delta$, where $\delta = 1 \in \{-1, 0, 1\}$ and $\ell = 1 \in \mathbb{N}$

2. This prime power q generates a finite field $\mathbb{F}_q = \{0, 1, 2, 3, 4\}$

3. Then we find a primitive element $\xi = 2$ of $\mathbb{F}_q$

4. We can construct generator sets $X = \{1, 4\}$ and $X' = \{2, 3\}$ from ξ

5. The generator sets X and X' are used to compute the coordinate matrix M_k of each cluster C_k, $0 \le k < n$. Take C_1 as example, its coordinate matrix is

$$M_1 = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 1 \\ 4 & 2 & 2 \\ 1 & 3 & 3 \\ 3 & 4 & 4 \end{bmatrix}$$

6. For each cluster C_k, $0 \le k < n$, create intra-cluster edges E_k using Algorithm 2. Intra-cluster edges of C_1 is shown in the bottom-left of Figure 1

7. For any two distinct cluster C_i, C_j, $0 \le i, j < n$, $i \ne j$, create inter-cluster edges $E_{i,j}$ using Algorithm 3. Inter-cluster edges between C_0 and C_2 is shown in the right part of Figure 1

8. Edge set of Galaxy graph is obtained by unioning all E_k created in step 6 and all $E_{i,j}$ created in step 7. Galaxy graph with $n = 3$, $q = 5$ is shown in Figure 1

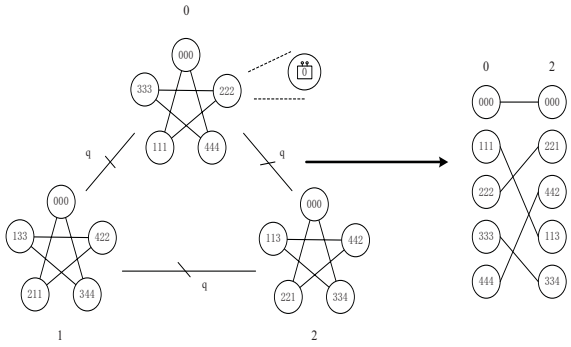


Figure 1: Galaxy graph ($n = 3, q = 5$)

N	Number of terminals in the network
r	Router radix
r'	The radix of each supernode
a	Number of routers in a supernode
p	Number of terminals attached to a router
h	Number of links to other supernodes attached to a router
h_g	Number of links to other clusters attached to a supernode in Galaxy graph
h_s	Number of links to other supernodes attached to a supernode in a cluster
n	Number of clusters in Galaxy graph
q	Number of supernodes in a cluster in Galaxy graph
δ	$q \ mod \ 4$

Table 2: Parameters of Galaxyfly

2.2 Topology Specification

Galaxyfly is a family of low-diameter topologies and consists of Galaxy graph and fully connected graph to meet the scalability requirement using cost-effective and radix-constrained routers. Galaxyfly is also a flexible hierarchical structure, which can well match traffic characteristics of HPC applications, being adaptive to dominating local traffics and reasonable global uniform traffic through adjusting scale and bisection bandwidth. We describe some parameters used in Galaxyfly in Table 2.

In Galaxyfly, nq supernodes constitute Galaxy graph as the top level and a routers are fully connected inside each supernode as the lower level. Each router $(a_{i,0}^k a_{i,1}^k \cdots a_{i,n-1}^k, t)$, $0 \le t \le a-1$ is attached to p terminals, $a-1$ links to other routers in the same supernode $(a_{i,0}^k a_{i,1}^k \cdots a_{i,n-1}^k)$, and h links to other supernodes. The radix of each router is $r = p + a + h - 1$.

Each supernode is composed of a routers, ap connections attaching to terminals and ah links jointing other supernodes as a virtual router with radix $r' = a(p+h)$ in Galaxy graph. The links of each supernode can be partitioned into $h_g = n - 1$ links connecting to other clusters and $h_s = (q - \delta)/2$ links connecting to other supernodes in the same cluster, where $ah \ge h_g + h_s$. Generally, links in the supernodes are local links, global links are connecting to two supernodes. The concept of supernode and cluster are analogous to the supernode in Dragonfly [20] and the subgraph

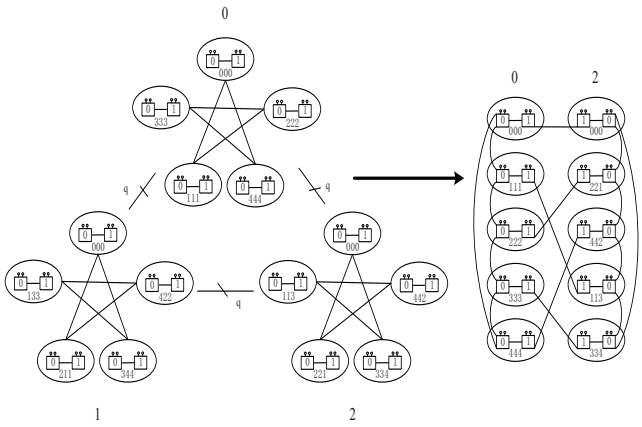


Figure 2: Galaxyfly$(n = 3, q = 5, a = 2, h = 2)$

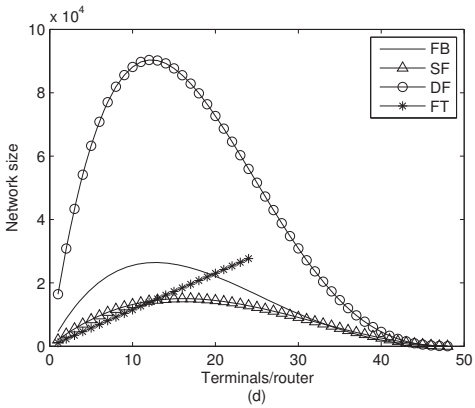


Figure 3: The scalability of different topologies with 48-port router

in Slim Fly [4].

If the value of parameter a is changed, Galaxyfly will be flat with $a = 1$ and hierarchical with $a > 1$. Dragonfly is a special case of Galaxyfly with $q = 1$. Although Dragonfly itself can support various scales through adjusting the values of parameters a, p, g, the maximal scale and balanced configuration of Dragonfly is achieved at the same time only when the parameters a, p, g satisfy $a = 2p$ and $h = p$. Slim Fly is a flat topology compared with Galaxyfly with $a > 1$ and Dragonfly. Galaxyfly with diameter 2 is a flat topology, which can afford higher performance than Slim Fly due to abundant links. Galaxyfly can support more various scales with the same router radix than Slim Fly and Dragonfly.

Figure 2 shows that Galaxyfly$(n = 3, q = 5, a = 2, h = 2)$ is diameter 4. There are 2 routers in each supernode. Each circle represents a supernode. In Figure 2, supernode (111) in the 0th cluster is connected to supernode (113) in the 2th cluster because of $a_{1,2} = a_{3,0}$ and in the same 1th cluster supernode (133) is connected to supernode (422) because of $a_{3,1} - a_{2,1} \in X$. Moreover, $a = 2$ routers in each supernode are divided to provide h_g and h_s links. In order to clarify, we denote routers $(a^k_{i,0}a^k_{i,1}a^k_{i,2}, 0)$ to connect supernodes in the same cluster and routers $(a^k_{i,0}a^k_{i,1}a^k_{i,2}, 1)$ to connect supernodes in different clusters.

3. GALAXYFLY STRUCTURE ANALYSIS

We now analyze the structure of Galaxyfly in accordance with common properties: flexibility, network diameter and bisection bandwidth. We compare our proposed Galaxyfly(GF) with Dragonfly(DF) [20], Flattened Butterfly(FB) [19], Fat Tree(FT) and Slim Fly(SF) [4]. For convenience, in following text we will refer to these topologies using abbreviations.

3.1 Flexibility

Flexibility means that the topology can be deployed in systems of different scales, from petascale to exascale systems and beyond. It is emerging to be a demanding characteristic of efficient topologies. We discuss flexibility of GF in this section through two ways. One is constructing different network scales using the same router radix. The other is constructing the same scale network using different router radix.

Figure 4 shows that GF in different configurations can build networks of various sizes from less than 1K to more than 10000K with 48-port routers via varying the number of terminals for each router. Compared to other topologies, in Figure 3, DF is the most scalable topology. DF itself can support various scales through adjusting the values of parameters a, p, g. The maximal scale and balanced configuration of DF is achieved at the same time only when the parameters a, p, g satisfy $a = 2p$ and $h = p$. FB and FT are 3-dimension and 3-level respectively. Given the degree and diameter, SF can achieve the largest scale. However, SF has the worst flexibility that it builds up the smallest range of network scales. GF is able to cover a wide range of scale requirements and achieve larger scale with the same router radix.

GF with configuration $d = 3, a > 1, q = 1$ is DF in Figure 4c. Hence, for the sake of fairness, we use the best configuration of DF with $a = 2h$ when compared to GF. The results show that GF can support larger scales than DF with its best configuration. Furthermore, GF can be configure to be of lower diameter than that of DF in small systems. Analysis with different configurations show that with the scale of GF growing larger, the diameter increases to an upper bound 5. Therefore, the construction of GF is a tradeoff between network size and network diameter.

The diameter of GF is determined by q and a. If $a = 1$, the diameter of GF is 2. If $q = 1$ and $a > 1$, the diameter is 3. If $q > 1$ and $a = 2$, the diameter is 4. If $a > 2$, the diameter is at most 5. GF is distinguished from other topologies that it has no fixed diameter.

Flexibility also indicates that a topology can be used to build up the same scale systems with various radixes. Figure 5 illustrates that the $N = 100K$ network can be constructed by balanced configurations with varied router radix and diameter. We change the terminals of each router and observe the variation of router radix. With the diameter increasing, the range of router radix is wider and the minimal radix is lower. However, the diameter is not the unique parameter affects. For instance, the minimal radix of $d = 3, a = 2h$ is lower than that of $d = 4, a = 2$ in Figure 5c and Figure 5e. The balanced configuration has at least $N/2$ bidirectional links across any bisection. More detailed analysis will be discussed in the bisection bandwidth sections. $n = q$ is set default in Figure 4d, Figure 4e, Figure 5e and Figure 5f.

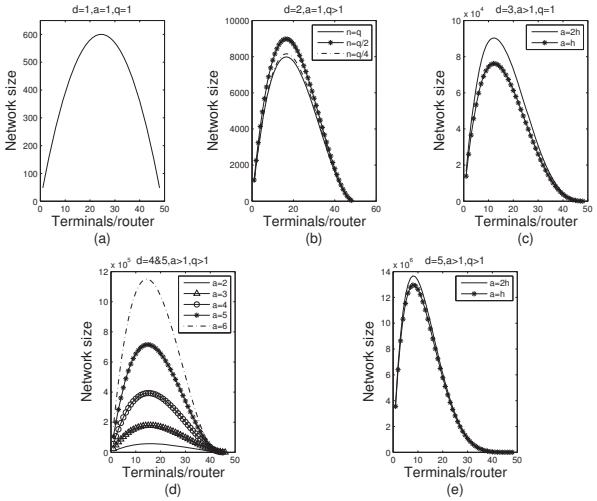


Figure 4: The scalability of Galaxyfly in different configuration with 48-port router (a) $d = 1$ **(b)** $d = 2$ **(c)** $d = 3$ **(d)** $d = 4\&5$ **(e)** $d = 5$

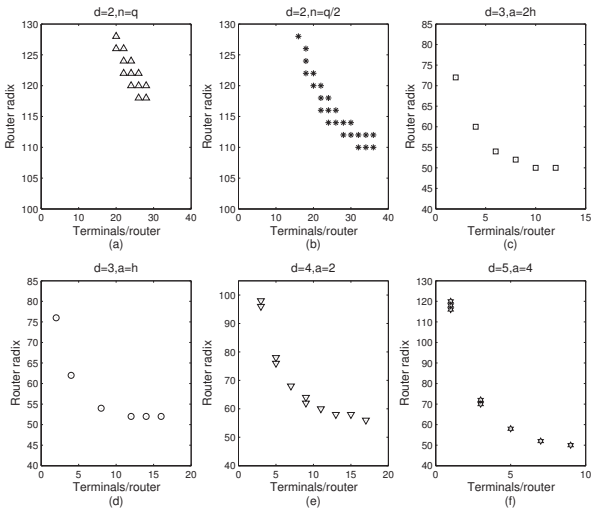


Figure 5: The $N = 100K$ **Galaxyfly in different balanced configuration with different radix (a)** $d = 2, n = q$ **(b)** $d = 2, n = q/2$ **(c)** $d = 3, a = 2h$ **(d)** $d = 3, a = h$ **(e)** $d = 4, a = 2, n = q$ **(f)** $d = 5, a = 4, n = q$

3.2 Bisection Bandwidth

Bisection bandwidth is a traditional comparison metric between topologies. In order to fairly compare our design with those existing topologies, we utilize it to measure the performance of topologies. We analyse the bisection bandwidth of GF with two parts, one is Galaxy graph for the top level networks, the other is the fully connected graph for supernodes.

For fully connected graph, the bisection bandwidth is obviously $a^2/4$. If supernode is nonblocking, it needs to have $ap/2$ bidirectional links.

The bisection bandwidth of Galaxy graph is considered by cutting inter-cluster or intra-cluster in half. If $B_m = l_m nq/4$ is the smallest link bisection among all the bisections of Galaxy graph, then it determines the bisection bandwidth. l_m is the smallest number of links for inter-cluster or intra-cluster in Galaxy graph. We assume that the link bandwidth is 1.

Without blocking in Galaxy graph, there are at least $N/2$ bidirectional links across any bisection. Hence, we utilize the ratio $\alpha = 2B_m/N = l_m/2ap$ to evaluate the bisection bandwidth. Figure 6 presents that we can construct proper GFs satisfying bisection bandwidth requirements with 48-port routers. In order to have a balanced network under uniform random traffic pattern, the bisection bandwidth ratio α must be no less than $1/2$. Figure 6a shows that the GF($d = 2, n = q/2$) can attain the maximal size with $\alpha = 1/2$. Figure 6b shows that by varying configurations, GF($d = 3, a > 1, q = 1$) can cover a wide range of network sizes from 20K to 83K with $\alpha = 1/2$. Figure 6c and 6d show that by varying size a of supernodes GF($n = q, q > 1, a > 1$) can satisfy different requirements of bisection bandwidth ratio and network size.

Figure 7 shows that GF with different configurations can achieve performance of other topologies. Suppose that we want to construct a balanced $N = 50K$ system with $\alpha \geq 0.5$, we can choose GF($d = 5, a = 2$) or GF($d = 3$). DF may also be used to build $N = 50K$ systems with the cost of more redundant links and routers, which means $\alpha \geq 2$. It is impossible to use SF and FB to achieve the scale $N = 50K$. Besides, with longer diameter, GF can achieve larger scale than other topologies given the same bisection bandwidth ratio. In Figure 7, GF($d = 5, a = 3$) can construct 130K network with $\alpha = 0.5$ and diameter 5. Other topologies cannot achieve such large scale unless higher router radix are utilized. In a word, GF makes better compromise between network size and performance.

3.3 Resiliency

Resiliency of a topology is reflected by the average path length and diameter. We analyse the resiliency of GF via simulating random link failures in GFs. Different ratios are used to remove links from the topology until the network is disconnected. Figure 8a and Figure 8b show that average path length and diameter vary with the link failure ratios. The average path length and diameter increase with link failure ratio. In case $N = 1K$, the average path length aggrandizes from 3.29 to 4.59 and the diameter changes from 4 to 8 with 60% links removed. GF with larger size is more resilient for the reason of high link density. For example, the link failure ratio of $N = 2K$ can be up to 70% compared to 60% of $N = 1K$. To route in the irregular topology with link failures, we can employ topology-agnostic deterministic routing algorithms. GF consists of Galaxy graph and fully connected graph. Each supernode in GF is a fully connected graph and fully connected graphs provide abundant links, so the resilience of GF should be comparable to other topologies. Due to the constrained space, detailed comparison with other topologies are not presented.

4. ROUTING

We now discuss minimal and non-minimal adaptive routing algorithm for GF .

4.1 Minimal Routing

The minimal routing algorithm for GF will be demonstrated in two distinct situations, $a = 1$ and $a \neq 1$.

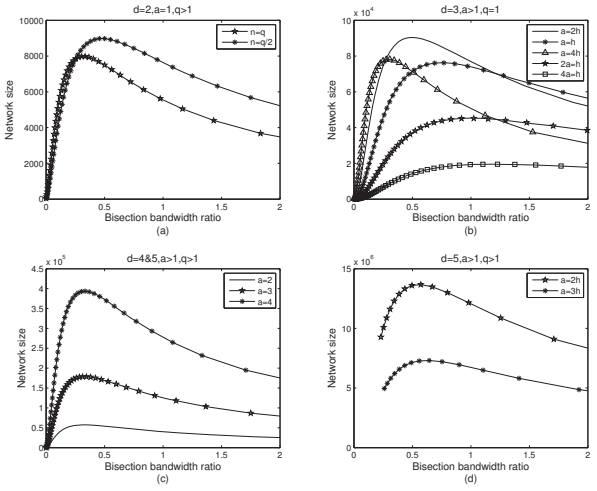


Figure 6: The bisection bandwidth of Galaxyfly in different configurations with 48-port routers (a) $d = 2$ **(b)** $d = 3$ **(c)** $d = 4\&5$ **(d)** $d = 5$

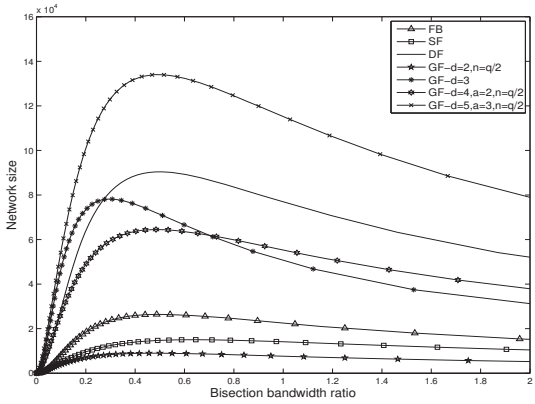


Figure 7: The bisection bandwidth of different topologies compared with GF in different configurations with 48-port routers

In the case $a = 1$, 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).

When the size of supernode is $a \neq 1$, a packet is routed. Firstly, the connection of source supernode and destination supernode is considered. One is connected directly (if G_s is connected to G_d or G_s and G_d are the same supernodes). The other is routed using two hops (if distance between G_s and G_d is two links via intermediate supernode G_i). And then it is considered that the packet is routed in the source supernode, the destination supernode and the intermediate supernode respectively.

The routing is discussed in detail:

Step 1: If $G_s \neq G_d$ and there is no links from G_s to G_d, it will find the 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.

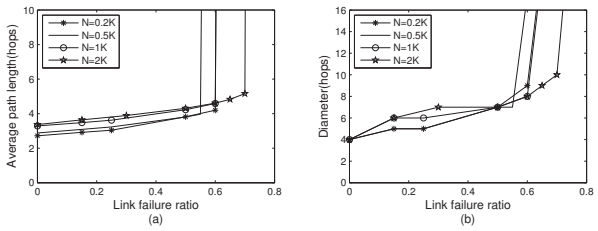


Figure 8: (a) Average path length and (b) Diameter with various link failure ratio in different sizes of GFs

Step 2: If there is no direct link from R_x to the supernode 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 minimal routing for GF. According to the locations of G_s and G_d, the minimal routing can be adjusted to be a balanced routing. In case G_s, G_d ($G_s \neq G_d$) are in the same cluster and not connected directly, there must exist more than one common neighbours of G_s and G_d. Then there are different paths from G_s to G_d and we can randomly pick one of the paths.

Figure 9 shows that the additional virtual channels (VC) are used to prevent deadlock in GF with different configurations. We partition GF into supernodes and promise deadlock-free routing algorithm for both inter- and intra-supernode routing only with enough VCs. The method is similar to the scheme of DF [20] and SF [4]. In Figure 9c, a packet comes across at most three intra-supernode links and two inter-supernode links. Therefore, we need at least 3 VCs for both intra-supernode links and inter-supernode links to avoid cycles in minimal routing algorithm.

4.2 Non-minimal Adaptive Routing

In GF, various parameter configurations make the network diameter varies from 2 to 5. Therefore, in order to control the hops between two terminals and reduce the number of VCs, non-minimal adaptive routing employs a constraint on selecting a random path of at most 5, 6 or 7 hops. We design three routing algorithms, non-minimal adaptive random link routing algorithm (NAR), non-minimal adaptive local link routing algorithm (NAL) and 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) algorithm.

The routing path of NAR is determined by queue size, though two supernodes is connected directly. The non-minimal path of NAL is determined by queue size of local links and hop distance. The non-minimal path of NAG depends on the queue size of global links. Thus, NAL is at most 6 hops and NAG is at most 7 hops. To avoid deadlocks with NAR, NAL and NAG, we need to respectively utilize 3, 4, 4 VCs for local links and 2, 2, 3 VCs for global links. Besides, we also can employ escape networks or turn models to break channel dependency with less VCs.

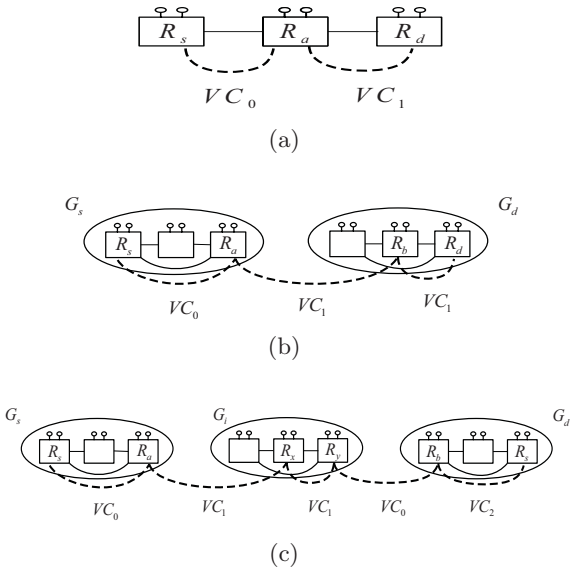


Figure 9: Minimal routing algorithm of GF(a) diameter = 2 (b) diameter = 3 (c) diameter = 5

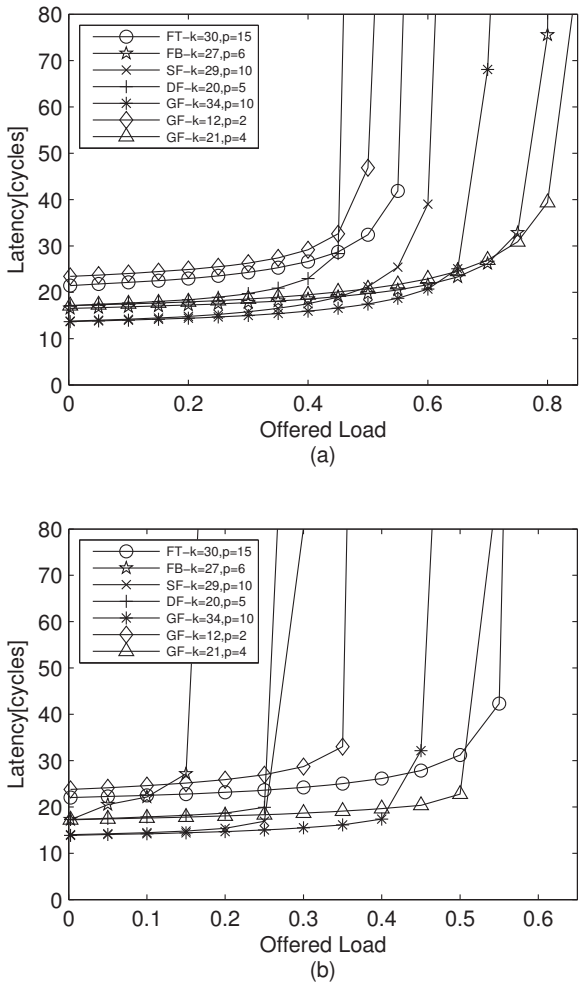


Figure 10: Performance comparison of GF, FT, FB, SF and DF. (a) Uniform random traffic pattern (b) Worst-case traffic pattern

5. PERFORMANCE

We simulate GF and other topologies with a cycle-accurate network simulator Booksim [26]. Packets are injected with a Bernoulli process. In order to avoid the influence of wormhole routing and virtual cut-through routing, messages are single flit packets. We assume 7 cycles delay in each input-queued router consisting of credit processing 2 cycles, routing computing 2 cycles, switch allocation, VC allocation and crossbar processing respectively 1 cycle. Input and output speedup of a crossbar is 1, and the speed of internal routers is 2 times of the channel transmission rate. Channels are set 1 cycle transmission delay. The buffer of each port has 256 flits and the policy is that each VC has 5 flits in privacy and others are shared by all VCs. We mainly evaluate uniform random and worst-case traffic pattern to test various HPC workloads.

In Figure 10a and Figure 10b, we show the comparison of some classical topologies with full global bandwidth. The full global bandwidth has at least $N/2$ bidirectional links across any bisection and the bisection bandwidth ratio α is more than 0.5. FT is one of the most common 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. We choose three configurations of GF with different diameters and bisection bandwidth ratio $\alpha \geq 0.5$. The size of networks in simulation is $N \approx 3K$ and there is at most 10% difference in size between various topologies. Router radix k and terminals p of each router for various topologies are presented as the legends of Figure 10a and 10b. The routing algorithms are Adaptive Nearest Common Ancestor (ANCA) routing for FT, minimal and non-minimal adaptive routing algorithms by local queue size for DF [20], FB [19], SF [4] and GF.

In uniform random traffic pattern, the source and destination of each packet are selected randomly. Figure 10a presents that in uniform random traffic pattern GF-$k = 21, p = 4$ with diameter 3 gets the best performance among all topologies and is slightly higher than FB-$k = 27, p = 6$. GF-$k = 21, p = 4$ and FB-$k = 27, p = 6$ are both diameter 3 topologies, but GF-$k = 21, p = 4$ uses lower router radix and provides more effective global bisection bandwidth with the bisection bandwidth ratio $\alpha = 1$. FB-$k = 27, p = 6$ is better than GF-$k = 34, p = 10$. However, the number of routers in FB-$k = 27, p = 6$ is nearly 60% more than that of GF-$k = 34, p = 10$ and FB-$k = 27, p = 6$ has 40% more links than GF-$k = 34, p = 10$. GF-$k = 34, p = 10$ has the same diameter as SF-$k = 29, p = 10$, but GF-$k = 34, p = 10$ has lower latency than SF-$k = 29, p = 10$ for SF-$k = 29, p = 10$ has 16% more links. Although the diameter of GF-$k = 12, p = 2$ is 5, the performance is little higher than DF-$k = 20, p = 5$ on account of more global bandwidth with the bisection bandwidth ratio $\alpha \geq 0.9$. Due to the diameter of FT-$k = 30, p = 15$ and GF-$k = 12, p = 2$, the zero load latency of them are higher than other topologies and configurations.

The saturation point for FT-$k = 30, p = 15$ is much lower than that of SF-$k = 29, p = 10$, GF-$k = 34, p = 10$ and GF-

$k = 21, p = 4$. There are three reasons. First, the diameter of FT-$k = 30, p = 15$ is larger than SF-$k = 29, p = 10$ and GF. The long-hop routing in FT-$k = 30, p = 15$ increases head-of-line (HOL) blocking and impacts the performance. Second, GF and SF-$k = 29, p = 10$ are both in balanced configurations. The lower diameter of SF-$k = 29, p = 10$ and GF mitigates the HOL in multistage switches and improves the performance. Third, the saturation point of injection rate of GF-$k = 21, p = 4 (a = 10, h = 8, p = 4, q = 1)$ is higher than that of FT-$k = 30, p = 15$ thanks to GF's abundant links.

In worst-case traffic pattern, the source and destination are selected by the longest path in the network. Some links are congested due to the high frequency usage. Packets are transmitted through the highest level routers in FT. In DF, packets are generated between two groups, resulting in the congestion of global links [20]. For SF, communication only occurs between two terminals in the same subgraph but different groups [4]. The worst-case pattern of FB is similar to the tornado pattern of $k - ary\ n - cube$ network. In GF, we arrange any two terminals in different clusters to communicate.

Figure 10b shows the results of worst-cast traffic pattern in different networks. FT saturates at 55% offered load thanks to the abundant links between levels and the balanced links density. FB has the lowest saturation point among all, which is caused by a lack of global links. The saturation point for GFs with three different configurations are better than SF and DF due to more global links between clusters. GF-$k = 12, p = 2$ has the lowest router radix among the three configurations, however, due to the diameter 5 and less links the performance is the lowest. Although GF-$k = 34, p = 10$ has the shortest diameter among the three configurations , the performance is still lower than GF-$k = 21, p = 4$ on account of the bisection bandwidth ratio $\alpha \approx 0.5$. Detail analyses with the different configurations of GFs will be shown in the following.

Figure 11a and Figure 11b indicate different configurations of GFs with $N \approx 3K$. We use minimal routing and NAR within 5 hops in simulation.

As expected, GF($n = 15, q = 29, a = 1, p = 7$) with diameter 2 has the best performance both in uniform random and worst-case traffic pattern thanks to the low diameter and bisection bandwidth ratio $\alpha = 1$. The saturation point for GF($n = 11, q = 29, a = 1, p = 10$) is lower than that for GF($n = 15, q = 29, a = 1, p = 7$) on account of less terminals per router and higher link density among inter-cluster network. GF($n = 81, q = 1, a = 10, p = 4$) is also better than GF($n = 11, q = 29, a = 1, p = 10$) because of the bisection bandwidth ratio $\alpha = 1$. GF($n = 81, q = 1, a = 10, p = 4$) and GF($n = 15, q = 29, a = 1, p = 7$) both have bisection bandwidth ratio $\alpha = 1$. The diameter 2 of GF($n = 15, q = 29, a = 1, p = 7$) is shorter than the diameter 3 of GF($n = 81, q = 1, a = 10, p = 4$). Therefore, the saturation point and latency for GF($n = 15, q = 29, a = 1, p = 7$) are better.

GF($n = 56, q = 1, a = 11, p = 5$) with $\alpha = 1/2$ is DF. It has similar saturation point to GF($n = 11, q = 37, a = 4, p = 2$), and its router radix is approximately 2 times of GF($n = 11, q = 37, a = 4, p = 2$). GF($n = 15, q = 37, a = 2, p = 3$) and GF($n = 11, q = 29, a = 2, p = 5$) with diameter 4 have lower performance than GF($n = 11, q = 37, a = 4, p = 2$) and GF($n = 15, q = 29, a = 4, p = 2$) with diameter 5 in both traffic patterns. The reason is that the former two topologies has fewer intra-supernode links.

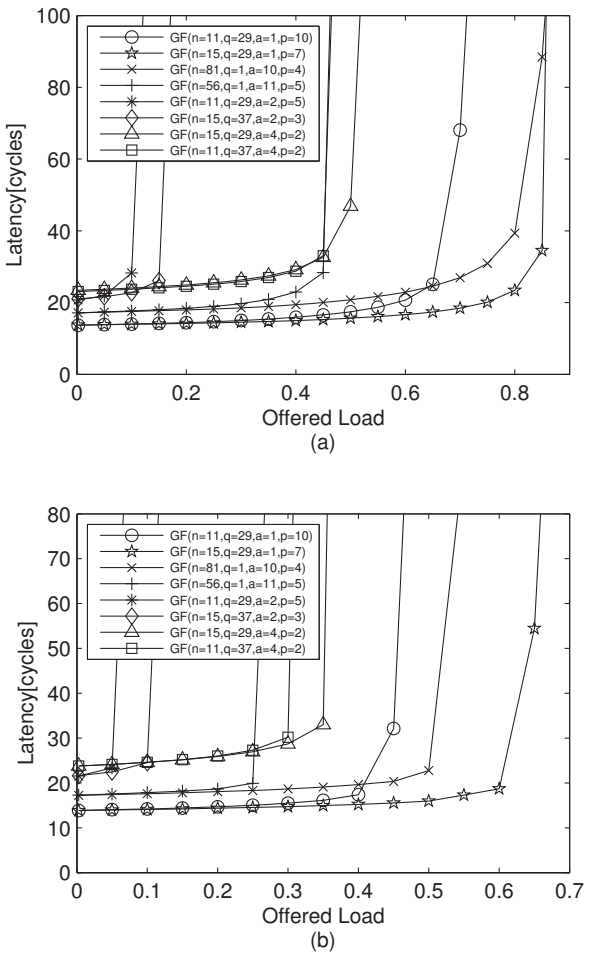


Figure 11: Performance comparison of GF with different parameter configurations. (a) Uniform random traffic pattern (b)Worst-case traffic pattern

The router radix of GF($n = 11, q = 37, a = 4, p = 2$) and GF($n = 15, q = 29, a = 4, p = 2$) are both 12. The bisection bandwidth ratio of GF($n = 11, q = 37, a = 4, p = 2$) and GF($n = 15, q = 29, a = 4, p = 2$) are $\alpha \approx 0.6$ and $\alpha \approx 0.9$ respectively. Thus, the saturation point for GF($n = 15, q = 29, a = 4, p = 2$) is slightly higher than GF($n = 11, q = 37, a = 4, p = 2$).

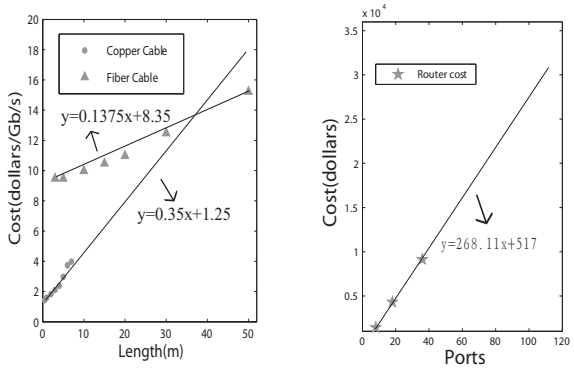
GF can construct the same scale network with various configurations. However, choosing a proper configuration of GF is a tradeoff among network size, network diameter, bisection bandwidth and router radix.

6. COST AND POWER COMPARISON

We now proceed to analyse the cost and power comparison of Galaxyfly with other topologies.

6.1 Physical Layout

We consider network topologies arranged by sets of compute nodes, routers and cables for the following physical layout.



(a) (b)

Figure 12: The tend of cables and routers cost (a) Cable cost model (b) Router cost model

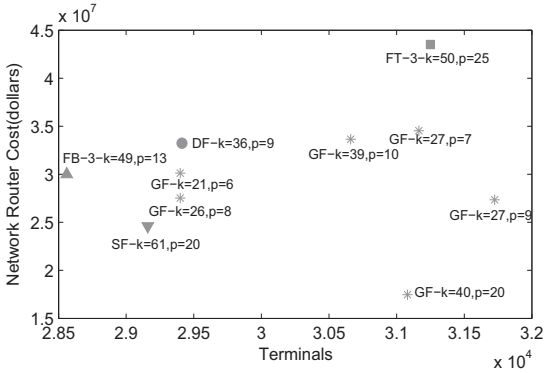


Figure 13: Network router cost with terminals $N = 30K$

One central and practical concern for low-diameter network is to enclose it in cabinets with minimal cabling costs. A good deployment from network topology to physical layout largely improves the system wiring. We now introduce a physical layout of GF.

GF is a flexible hierarchical network, which can be partitioned into some modules. We divide the routers and their attached terminals into cabinets with an equal number of cables connecting the cabinets. GF is easy to be divide into modules by the number of clusters n and the size of clusters q as a 2-D grid pattern. If a is too small, then an entire cluster with q supernodes can be arranged into a cabinet, otherwise, a cabinet contains one supernode or more. Therefore, we can exploit the proper design to limit the cost by parameters n, q, a, p of GF.

6.2 Cost Model

We now introduce a cost model (similar to the model used in [4] and [19]) that the overall network costs is maninly constituted by the cost of routers and interconnection cables. We assume that router together with terminals are packaged in cabinets of size $1\times1\times2$ meters. Intra-cabinet cables always use electrical cables (backplane resources) while inter-cabinet use optical cables. In order to compute cable length, we conservatively assume that intra-cabinet cables are 1.5m long on average and length of inter-cabinet cables are estimated using the Manhatten distance [4] in a 3-D 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[20], Regular Random topology(RR) [22], FB-3 (3-D FB) [19], FT-3 (3-level FT) and SF [4]. On account of restricted space, we have to omit the layout of those topologies.

To estimate the costs of these topologies, we use the price of cables and routers [15, 16] as Figure 12a and Figure 12b. The size N of comparable networks cannot be identical for each topology in their balanced configurations. The difference of network size is controlled in $1-6\%$.

We compare the network cost of different topologies with network size $N = 30K$. Figure 13 presents the network router cost of various topologies. Firstly, GF has various balanced configuration and the router cost of DF, FB-3 and FT-3 are almost 20%, 9% and 58% more expensive than GF-$k = 26, p = 8$. Secondly, although the SF is approximately 10% more cost-effective than GF-$k = 26, p = 8$, the router radix of SF is 2.3 times that of GF. Besides, we can relax the requirement of bisection bandwidth in order to reduce cost. GF-$k = 40, p = 20$ has lower router radix than SF and is much cheaper than SF. Finally, GF can achieve the reduction of router cost owing to lower router radix and lower bisection bandwidth.

In the analysis of network cable cost, we propose two methods to deploy routers, cabinets and cables for different topologies. The first method partitions the network according to the approximate size of group in DF. The other one is determined by the approximate size of cluster in GF. GF-$k = 26, p = 8$ is compared with other topologies under the above configurations. The analysis of network cable cost by the first method is presented as Figure 14. The cabinets layout of DF, SF and GF are packaged as $13 \times 13 + 3$, $13\times12+6$ and $12\times14+7$ Grid. The results of GF get closed to SF. Though the electric cable length of SF is shorter than GF, the fiber cable length of SF is longer than GF. FT-3 is the most expensive one among the topologies because of a large number of fiber cables. RR occupies the second place with randomly connected cables. FB-3 is 43% and 39% cheaper than FT-3 and RR. SF and DF are approximately 23% and 7% cheaper than FB-3, respectively. Among these topologies, GF is the most cost-effective, which is about 24% cheaper than FB-3.

If we employ the second method to deploy the network, the cabinets layout of DF, SF and GF are packaged as $7 \times 6 + 1$, $5 \times 5 + 2$ and 5×5 Grid and the cost is showed in Figure 15. With various layouts, the network cable cost of different topologies present the identical trend. GF is the most cost-effective one among the topologies. Increasing the size of cabinet can reduce the network cable cost on account of cost-effective backplane resources.

6.3 Energy Model

Network energy consumption occupies approximately 50% of the usage of the whole system [1]. We employ the model [1] that per port with four lanes(four SerDes) consumes about 2.8 watts and network interface controller of each compute node is fully used with 10 watts. We compare GF with

Topology	DF	FB-3	FT-3	SF	GF-k=26,p=8	GF-k=27,p=9	GF-k=40,p=20
Terminals	29412	28561	31250	29160	29400	31725	31080
Routers	3268	2197	3125	1458	3675	3525	1554
Radix	36	49	50	61	26	27	40
Power per terminal (W)	21.2	20.6	24	18.5	19.1	18.4	15.6

Table 3: The energy model of GF and other topologies

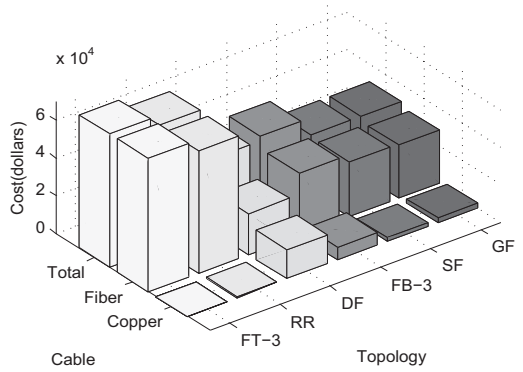


Figure 14: Network cable cost with terminals $N = 30K$

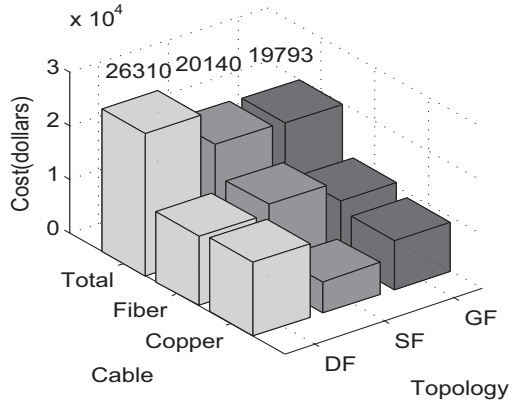


Figure 15: Network cable cost with the approximate size of cluster in GF per cabinet

other topologies in the energy model in Table 3. Although the number of routers in SF is smaller than GF-$k = 27, p = 9$ and GF-$k = 40, p = 20$, the power consumption of SF is still more than those owing to more radixes in SF.

7. RELATED WORK

In this paper, some topologies are summarized and compared with Galaxyfly, such as FT, FB [19], DF [20], SF [4] and so on. FT is popular among HPC systems and data center systems. Although FT provide abundant path diversity and high bandwidth, the physical cost and power consumption are bottlenecks in usage. FB and DF take advantage of high radix routers to achieve higher bandwidth and lower cost and power consumption. SF is a approximately optimal network with fixed router radix and diameter 2. However, once router radix is fixed, the scale of networks are limited by these topologies above. Regular Random graph [22] support arbitrary scale networks with moderately high radixes, but the performance of network and physical cost are undetermined. Besides, it is impossible for COTS routers to provide enough space for routing table used in regular random graphs.

Apart from HPC systems, some topologies in data center systems are varied and worthy of usage in HPC systems. Dcell [10], which is a hierarchically recursive network with fully connected graph in each level, is a scalable and fault-tolerant network structure for data centers. Bcube [9] is a server-centric hierarchical network for modular data center, in which the server is not only a terminal but also transmits messages like a mini router. Via adding free-space optical links [17] or wireless links [12], network can be reconfigured for different applications.

8. CONCLUSION

High-radix interconnection network is the current and upcoming solution for larger-scale HPC systems. However, the development of high-radix router 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, based on Galaxy graph, we have proposed a novel family of flexible-radix low-diameter topology Galaxyfly, which not only enhances the flexibility of topology but also works out the construction of HPC systems from petascale to exascale systems and beyond with moderately high radixes. Furthermore, we construct Galaxyfly which makes a compromise in network performance, physical layout complexity, cost and power consumption.

9. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful comments and professors Longjiang Qu, Li Dai, Yue Zhou and Yunwen Liu for their introduction of graph theory. Besides, we would thank professor Kefei Wang and all teachers in our lab for their instructions and discussions. Finally, we would thank Wenxiang Yang, Ji Wu, Jingyue Zhao and He Huang for their help. The work was partially supported by 863 Program under Grant No. 2015AA01A301, NSFC under Grants No. 61272482, No. 61303066 and FANEDD under Grant No. 201450.

10. REFERENCES

[1] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu. Energy proportional datacenter networks. In *Proceedings of International Symposium on Computer Architecture (ISCA)*, pages 338–347, 2010.

[2] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber. Hyperx: Topology, routing, and packaging of efficient large-scale networks. In

Proceedings of International Conference for High Performance Computing, Networking, Storage, and Analysis (SC), pages 41:1–41:11, 2009.

[3] S. Ankit, H. Chi-Yao, P. Lucian, and G. P. Brighten. Jellyfish: Networking data centers randomly. In *Proceedings of Symposium on Network System Design and Implementation (NSDI)*, pages 225–238, 2012.

[4] M. Besta and T. Hoefler. Slim fly: A cost effective low-diameter network topology. In *Proceedings of International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)*, pages 348–359, 2014.

[5] N. Binkert, A. Davis, N. P. Jouppi, M. McLaren, N. Muralimanohar, R. Schreiber, and J. H. Ahn. The role of optics in future high radix switch design. *ACM Sigarch Computer Architecture News*, 39(3):437–448, June 2011.

[6] N. Chrysos, C. Minkenberg, M. Rudquist, C. Basso, and B. Vanderpool. Scoc: High-radix switches made of bufferless clos networks. In *Proceedings of High-Performance Computer Architecture (HPCA)*, pages 402–414, 2015.

[7] G. Faanes, A. Bataineh, D. Roweth, T. Court, E. Froese, B. Alverson, T. Johnson, J. Kopnick, M. Higgins, and J. Reinhard. Cray cascade: A scalable hpc system based on a dragonfly network. In *Proceedings of International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)*, pages 103:1–103:9, 2012.

[8] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova. Skywalk: a topology for hpc networks with low-delay switc. In *Proceedings of International Parallel and Distributed Processing Symposium (IPDPS)*, pages 263–272, 2014.

[9] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, and et al. Bcube: A high performance, server-centric network architecture for modular data centers. In *Proceedings of International Conference on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, pages 63–74, 2009.

[10] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: A scalable and fault-tolerant network structure for data centers. *ACM SIGCOMM Computer Communication Review*, 38(4):75–86, Aug. 2008.

[11] P. R. Hafner. Geometric realisation of the graphs of mckay- miller- siran. *Journal of Combinatorial Theory*, 90(2):223–232, 2004.

[12] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer. Firefly: A reconfigurable wireless data center fabric using free-space optics. In *Proceedings of International Conference on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, pages 319–330, 2014.

[13] https://en.wikipedia.org/wiki/Isomorphism.

[14] http://www.avagotech.com/optical_fpga#.

[15] http://www.colfaxdirect.com.

[16] http://www.mellanoxstore.com.

[17] F. Ikki, K. Michihiro, O. Tomoya, M. Hiroki, and C. Henri. Augmenting low-latency hpc network with free-space optical links. In *Proceedings of High-Performance Computer Architecture (HPCA)*, pages 390 – 401, 2015.

[18] S. Jeloka, R. Das, R. G. Dreslinski, T. Mudge, and D. Blaauw. Hi-rise: A high-radix switch for 3d integration with single-cycle arbitration. In *Proceedings of IEEE Computer Society Technical Committee on Microprogramming and Microarchitecture (MICRO)*, pages 471–483, 2014.

[19] J. Kim, W. J. Dally, and D. Abts. Flattened butterfly: A cost-efficient topology for high-radix networks. In *Proceedings of International Symposium on Computer Architecture (ISCA)*, pages 126–137, 2007.

[20] J. Kim, W. J. Dally, S. Scott, and D. Abts. Technology-driven, highly-scalable dragonfly topology. In *Proceedings of International Symposium on Computer Architecture (ISCA)*, pages 77–88, 2008.

[21] M. Koibuchi, I. Fujiwara, H. Matsutani, and H. Casanova. Layout-conscious random topologies for hpc off-chip interconnects. In *Proceedings of High-Performance Computer Architecture (HPCA)*, pages 484–495, 2013.

[22] M. Koibuchi, H. Matsutani, H. Amano, D. F. Hsu, and H. Casanova. A case for random shortcut topologies for hpc interconnects. *ACM Sigarch Computer Architecture News*, 40(3):177–188, 2012.

[23] S. B. Mark, D. Mark, H. Ram, K. James, L. Tom, R. Todd, D. U. Keith, and R. C. Zak. Intel omni-path architecture: Enabling scalable, high performance fabrics. In *Proceedings of Symposium on High-Performance Interconnects (HOTI)*, pages 402–414, 2015.

[24] B. D. McKay, M. Miller, and J. Siran. A note on large graphs of diameter two and given maximum degree. *Journal of Combinatorial Theory*, 74(1):110 – 118, 1998.

[25] T. P. Morgan. The road to 200g networks starts with the transceiver. http://www.nextplatform.com.

[26] J. Nan, U. B. Daniel, M. George, B. James, T. Brian, K. John, and J. D. William. A detailed and flexible cycle-accurate network-on-chip simulator. In *Proceedings of International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 86 – 96, 2013.

[27] A. Putnam, A. Caulfield, E. Chung, D. Chiou, K. Constantinides, and et al. A reconfigurable fabric for accelerating large-scale datacenter services. In *Proceedings of International Symposium on Computer Architecture (ISCA)*, pages 13–24, 2014.

[28] S. Satpathy, K. Sewell, T. Manville, Y.-P. Chen, R. Dreslinski, D. Sylvester, T. Mudge, and D. Blaauw. A 4.5Tb/s 3.4Tb/s/W 64×64 switch fabric with self-updating least-recently-granted priority and quality-of-service arbitration in 45nm CMOS. In *Proceedings of International Solid-State Circuits Conference (ISSCC)*, pages 478–480, 2012.

[29] S. Scott, D. Abts, J. Kim, and W. J. Dally. The blackwidow high-radix clos network. In *Proceedings of International Symposium on Computer Architecture (ISCA)*, pages 16–28, 2006.

[30] L. Xiangke, P. Zhengbin, W. Kefei, L. Yutong, X. Min, X. Jun, D. Dezun, and S. Guang. High performance interconnect network for tianhe system. *Journal of Computer Science & Technology*, 30(2):259–272, 2015.