Cost-Effective Diameter-Two Topologies: Analysis and Evaluation

Georgios Kathareios, Cyriel Minkenberg
Bogdan Prisacari, German Rodriguez
IBM Research – Zurich
Säumerstrasse 4, 8803 Rüschlikon, Switzerland
{ios,sil,bpr,rod}@zurich.ibm.com

Torsten Hoefler
ETH Zurich
Universitaetsstrasse 6, 8092 Zürich, Switzerland
htor@inf.ethz.ch

ABSTRACT

HPC network topology design is currently shifting from high-performance, higher-cost Fat-Trees to more cost-effective architectures. Three diameter-two designs, the Slim Fly, Multi-Layer Full-Mesh, and Two-Level Orthogonal Fat-Tree excel in this, exhibiting a cost per endpoint of only 2 links and 3 router ports with lower end-to-end latency and higher scalability than traditional networks of the same total cost. However, other than for the Slim Fly, there is currently no clear understanding of the performance and routing of these emerging topologies. For each network, we discuss minimal, indirect random, and adaptive routing algorithms along with deadlock-avoidance mechanisms. Using these, we evaluate the performance of a series of representative workloads, from global uniform and worst-case traffic to the all-to-all and near-neighbor exchange patterns prevalent in HPC applications. We show that while all three topologies have similar performance, OFTs scale to twice as many endpoints at the same cost as the others.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.4 [**Performance of Systems**]

General Terms

Performance, Theory, Experimentation

Keywords

Diameter-two networks, Adaptive routing, All-to-all, Nearest neighbor, Global and adversarial traffic, Slim Fly, Multi-Layer Full-Mesh, Orthogonal Fat-Tree

1. INTRODUCTION / RELATED WORK

One of the most popular interconnection network designs, used in both the High Performance Computing and datacenter space, is the Fat-Tree [13,18,19]. Full bisection Fat-Trees ensure that any permutation traffic can traverse the network at maximum bandwidth and can attain close to ideal behavior for any communication pattern (with a properly chosen routing strategy) in practice. At small scale, Fat-Trees require only two levels of switches and as such are very cost effective. When moving to larger scales, they require increasingly more resources as the number of levels increases, both in terms of routers and network cables. Typical cost reduction measures (such as slimming for example) generally have the drawback of impacting performance across many traffic patterns, particularly the more global ones [20]. In practice, the network does not need to be optimal for *every* pattern, but global communication (such as uniform random global traffic, or all-to-all exchanges) is central to many applications. Therefore, several alternatives have been proposed that are more cost effective than a three level Fat-Tree while maintaining close to ideal performance for a subset of communication patterns, particularly global uniform traffic. The most widely deployed of these has been the Dragonfly topology [11], employed for example in the IBM PERCS systems [3] and in Cray Cascade systems [8] (e.g., Piz Daint and Shaheen II [1]).

However, most of these alternative topologies are still not a match for the two level Fat-Tree, neither in terms of cost nor in terms of latency/diameter. Two classes of topologies, one direct and one indirect, are exceptions to this rule, sharing the same cost and diameter metrics with the two-level Fat-Tree but having better, approaching maximal, scalability, making them ideal candidates for current HPC and datacenter interconnects. The direct topology is the recently proposed Slim Fly (SF) [4]. The indirect topology class is one that we introduce in this paper and that we call Stacked Single-Path Trees (SSPT). The Multi-Layer Full-Mesh (MLFM) [9] as well as the two-level Orthogonal Fat-Trees (OFT) [22, 23] are examples of members of this class.

While the individual topologies have already been described in the literature, the SSPT class has not. Also, other than for the Slim Fly there have been no concrete proposals as to how to perform routing and deadlock avoidance, nor have there been comparative performance studies between the three options. In this paper:

- We introduce and analyze the SSPT topology class.
- We compare the diameter-two topologies in terms of cost, scalability, and bandwidth limitations.
- We propose load oblivious and adaptive deadlock-free routing strategies for the MLFM and OFT topologies

SC '15, November 15 - 20, 2015, Austin, TX, USA

ACM 978-1-4503-3723-6/15/11:.$15.00.
DOI: http://dx.doi.org/10.1145/2807591.2807652

and discuss these in comparison with respective strategies for the SF.

- We evaluate the performance of the three topologies under the proposed routing and deadlock avoidance strategies through simulations, for several representative communication patterns: on the one hand global uniform and adversarial worst-case synthetic traffic and on the other hand two of the most representative traffic patterns in the HPC space: all-to-all and nearest-neighbor exchanges.

2. DIAMETER-TWO TOPOLOGIES

In this section we present the description of the topologies and perform an analysis of their main characteristics.

		Symbol	Explanation
General		N	Number of end-nodes
		R	Number of routers
		r	Router radix
		p	Number of end-nodes attached to routers (Applies only to routers with end-nodes for the OFT and MLFM)
		N_p	Number of total router ports in the topology
		N_l	Number of total links in the topology
Direct	SF	q	Prime power that defines N, R, and r
		r'	Network radix
		(i,j,k)	Router in the j-th row and k-th column of the i-th subgraph
Indirect	MLFM	l	Number of layers
		h	Network radix of local routers
		R_g, R_l	Number of global, local routers
	OFT	k	Network radix of routers with end-nodes
		(i,j)	j-th router of the i-th level
		R_L	Number of routers per level

Table 1: Symbols used in the paper.

2.1 Direct Topologies

2.1.1 Two-Dimensional HyperX

The n-dimensional HyperX (also known as Generalized Hypercube) [2, 5] is the direct topology resulting from the Cartesian product of n fully-connected graphs. The longest path between a pair of routers in such a topology has length n, corresponding to one movement in each of the n dimensions. The two-dimensional HyperX is thus a direct diameter-two topology. It is defined by three main parameters: the sizes of the two fully-connected graphs in the Cartesian product and the number p of end-nodes connected to each router. Typical configurations use the same size for the two fully-connected graphs while p is chosen such that the network is balanced, that is, it is able to sustain full injection bandwidth for uniform traffic. Given a router with radix r, this translates into an equal number of ports $r/3$ being used to i) connect to routers in the same fully-connected graph in the first dimension; ii) connect to routers in the same fully-connected graph in the second dimension; and iii) connect to end-nodes. The number of routers in each fully-connected graph is thus $\frac{r}{3}+1$ leading to a total number of routers of $R=(\frac{r}{3}+1)^2$ and a total number of end-nodes of $N=pR=\frac{r}{3}\cdot(\frac{r}{3}+1)^2$. The network has a per end-point cost of 3 router ports and 2 links.

2.1.2 Diameter-Two Slim Fly

The SF [4] is a direct topology created by arranging the routers in a McKay, Miller, and Širán (MMS) graph [14]. MMS graphs approximate the Moore Bound [15], the maximum number of nodes a graph can have for a given node degree and diameter. This effectively means that the SF is among the largest direct diameter 2 networks possible, reaching approximately 88% of the Moore Bound [4].

The creation of the SF topology begins with the selection of a prime power q of the form $q=4w+\delta$, $w\in\mathbb{N}$, $\delta\in\{-1,0,1\}$. From that, we calculate ξ, a primitive element of the Galois Field F_q and create two *generator sets* X and X', as follows (all calculations are performed over the field):

- $X=\{1,\xi^2,\cdots,\xi^{q-3}\}$, $X'=\{\xi,\xi^3,\cdots,\xi^{q-2}\}$, if $\delta=1$
- $X=\{1,\xi^2,\xi^4,\cdots,\xi^{2w-2},\xi^{2w-1},\xi^{2w+1},\cdots,\xi^{4w-3}\}$, $X'=\{\xi,\xi^3,\xi^5,\cdots,\xi^{2w-1},\xi^{2w},\cdots,\xi^{4w-4},\xi^{4w-2}\}$, if $\delta=-1$
- $X=\{1,\xi^2,\cdots,\xi^{q-2}\}$, $X'=\{\xi,\xi^3,\cdots,\xi^{q-1}\}$, if $\delta=0$

The topology consists of $R=2q^2$ routers, arranged in two subgraphs, each with q^2 routers in q rows and columns (Fig. 1a). Each router has two kinds of network connections: $2w$ connections to routers of the same column in its subgraph, and q connections directed to the other subgraph, one in each of its columns, leading to a network radix $r'=\frac{3q-\delta}{2}$. Specifically:

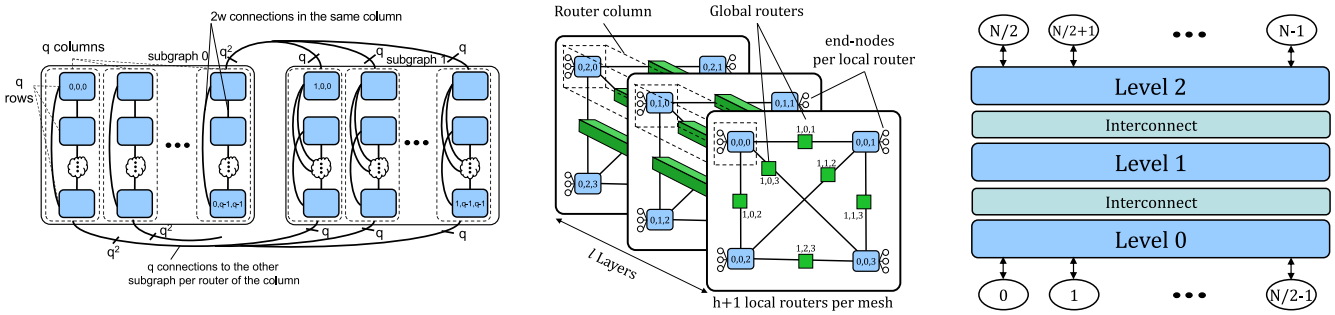
$$(0,x,y) \text{ connects to } (0,x,y') \text{ if } y-y'\in X$$

$$(1,m,c) \text{ connects to } (1,c,c') \text{ if } c-c'\in X'$$

$$(0,x,y) \text{ connects to } (1,m,c) \text{ if } y=mx+c$$

According to Besta and Hoefler [4], the number p of end-nodes connected to each router is selected to be equal to half the network radix: $p=\lceil\frac{r'}{2}\rceil$. This value ensures full global bandwidth for the topology (i.e., given a uniform communication pattern, end-nodes are theoretically able to sustain full bandwidth injection). Here, we argue that using the ceiling function, while enabling higher scalability and better cost per endpoint, slightly overestimates the number of nodes that can be attached to a router, leading to lower performance. As such, we also consider configurations with $p=\lfloor\frac{r'}{2}\rfloor$ and show in Section 4 the performance implications of this choice.

The Slim Fly comprises $R=2q^2$ routers. These routers have a network radix of $r'=q+2w=\frac{3q-\delta}{2}$ and $p=\frac{r'}{2}\approx\frac{3q-\delta}{4}$ attached end-nodes, amounting to a router radix of $r=\frac{3}{4}(3q-\delta)$. The number of end-nodes is $N=pR\approx\frac{q^2(3q-\delta)}{2}$. The total number of router ports is $N_p=rR\approx\frac{3}{2}q^2(3q-\delta)$, and the total number of links is $N_l=N+\frac{r'R}{2}\approx q^2(3q-\delta)$, resulting in a cost of approximately 3 ports and 2 links per end-node. As practical values of q are relatively small, the choice of rounding up or down when determining p has a non-negligible impact on the cost metrics. As an example, for $q=13$, selecting $p=10$ results in a cost of 2.9 ports and 1.95 links per endpoint, while selecting $p=9$ results in 3.11 ports and 2.05 links per endpoint.

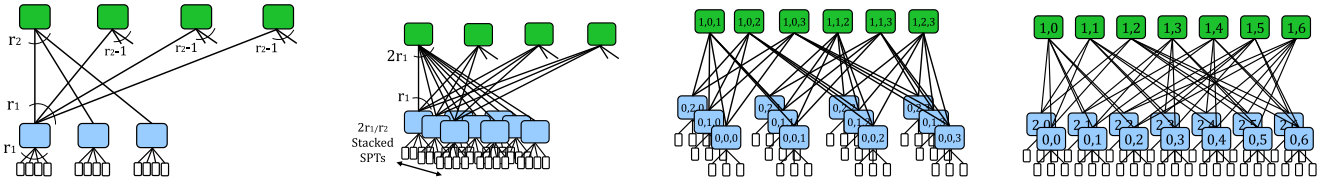


(a) Diameter-Two Slim Fly (End-nodes are omitted for clarity).

(b) 3,3,3 MLFM.

(c) Two-level OFT.

Figure 1: System view of diameter-two cost-effective topologies



(a) SPT.

(b) SSPT.

(c) 3,3,3-MLFM.

(d) 3-OFT.

Figure 2: Tree view of diameter-two cost-effective topologies

2.2 Indirect Topologies

2.2.1 *Two Level Fat-Trees*

A two level Fat-Tree is an indirect topology built from two levels of routers. The endpoints are connected to the routers on the first level while the second level is used to provide an increase in bandwidth/path diversity. A full bisection two level Fat-Tree must additionally obey the constraint that every level-one router have as many links connecting it to level-two routers as endpoints. This effectively translates into the network theoretically providing sufficient bandwidth to accommodate any permutation traffic pattern at full injection. Given a configuration where every router has a fixed even router radix r, the number of nodes connected to every level-one router is then $p = r/2$, the total number of routers is $3r/2$ and the total number of endpoints is $r^2/2$. The cost of the network is 3 router ports and 2 links per endpoint.

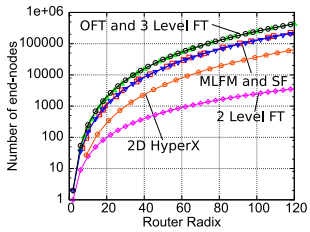
2.2.2 *Stacked Single-Path Trees*

While the full bisection two-level Fat-Tree simultaneously offers a low cost, a low diameter, and high throughput for any permutation pattern (due to high path diversity), its use in practice is limited by its low scalability under a fixed router radix. In this section we will show that we can sacrifice one of the defining characteristics of this design, the high path diversity, and still preserve the other two advantages, while also providing high throughput for random uniform traffic. To define the structure of *Single-Path Trees* (SPT) (Fig. 2a), we start as in the Fat-Tree case with two levels of routers, where nodes will be connected only to routers in the first layer and every router-to-router link has an endpoint in each of the levels. Unlike the Fat-Tree though, we propose to interconnect the two layers in such a way that i) exactly a single minimal path exists between any pair of level-one routers, and ii) a minimal number of level-two routers are used. Given a router-to-router radix of r_1 for the first level routers and of r_2 for the second level routers, such a design scales to $R_1 = 1 + r_1 \cdot (r_2 - 1)$ first level routers (requiring $R_2 = R_1 \cdot r_1/r_2$ second level routers). To achieve maximum performance for random uniform traffic, the number of nodes connected to each level-one router is limited to $p = r_1$. The scalability of an SPT is thus given by $N = r_1^2 \cdot (r_2 - 1) + r_1$. In terms of cost, an SPT then requires R_1 routers with $2r_1$ ports each and R_2 routers with r_2 ports each, for a total of $3R_1r_1$ ports, or 3 ports per end-point. Similarly, the number of links per endpoint is equal to 2.

Determining level-one to level-two interconnection patterns with the constraints characteristic of SPTs is not straightforward and as such we might be limited in practice to certain combinations of (r_1, r_2) values. In particular, precise procedures are known to build such interconnection patterns for the $r_2 = r_1$ case when $r_1 - 1$ is prime as well as for any value of r_1 when $r_2 = 2$. However, when building interconnection networks in practice, it is often desirable that individual routers be identical, i.e., have the same radix. An interconnection pattern might not be readily available for the $(r_1, r_2 = 2r_1)$ case corresponding to this goal for an SPT. One might however be available for the (r_1, r_2) case, r_2 being some divisor of $2r_1$. In that case, one way of achieving the goal of building the network from individual routers is a procedure that we will call stacking SPTs and leading to *Stacked Single-Path Tree* (SSPT) topologies (Fig. 2b). The procedure consists of logically instantiating $2r_1/r_2$ identical $SPT(r_1, r_2)$ topologies and then "merging" each $2r_1/r_2$-tuple of corresponding radix-r_2 level-two routers together to form a single $2r_1$ radix physical router. For every intra-SPT pair of endpoints, the SPT properties are trivially preserved. For every inter-SPT pair

i	j, s.t. $(1,j)$ and $(0,i)$ are connected			
0	9	10	11	12
1	9	0	1	2
2	9	3	4	5
3	9	6	7	8
4	10	0	3	6
5	10	1	4	7
6	10	2	5	8
7	11	0	4	8
8	11	1	5	6
9	11	2	3	7
10	12	0	5	7
11	12	1	3	8
12	12	2	4	6

Table 2: 4-ML3B.



	Topology	Diam.	Scale	$\frac{N_l}{N}$	$\frac{N_p}{N}$
Direct	2D HyperX	2	$\approx r^3/27$	2	3
	Slim Fly (SF)	2	$\approx r^3/8$	2*	3*
Indirect	2-lvl Fat-Tree	2	$r^2/2$	2	3
	3-lvl Fat-Tree	4	$r^3/4$	3	5
	MLFM	2	$\approx r^3/8$	2	3
	OFT	2	$\approx r^3/4$	2	3

* See Section 2.1.2

Figure 3: Comparison of the scale and cost (links and ports per end-node) of various low diameter topologies.

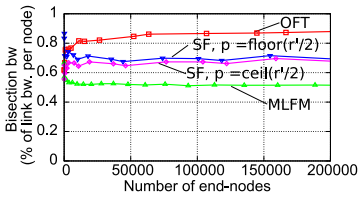


Figure 4: Approximating bisection bandwidth.

of endpoints, the diameter property is preserved (a shortest path always consists of an upward SPT traversal in the source SPT and a downward SPT traversal in the destination SPT), while the single path property is preserved for all pairs excepting those containing corresponding nodes in two different SPTs. Indeed, for the latter, the path diversity is equal to r_1. The scale of the resulting network is: $N = \left(r_1^2(r_2-1)+r_1\right)\cdot\frac{2r_1}{r_2} = \frac{r^3}{4}\left(\frac{r_2-1}{r_2}\right)+\frac{r^2}{2r_2}$, where $r = 2r_1$ is the router radix of the resulting topology. As each SPT that we stack has a network cost of 3 ports and 2 links per end-node, the SSPT has the same cost per end-node.

In the following, we will present two particular instances of SSPTs, the Multi-Layer Full-Mesh, obtained for $r_2 = 2$ and the two level Orthogonal Fat-Tree, obtained for $r_2 = r_1$ when $r_1 - 1$ is prime.

2.2.3 Multi-Layer Full-Mesh

The *MLFM* [9], as the name suggests, is initially conceived as stacked layers of full-mesh networks. The stacking procedure is performed by considering each layer as a normal full-mesh, and replacing the direct link between any pair of local routers (LRs) with two links to one global router (GR) (Fig. 1b). By connecting the respective pairs of all layers to the same GRs, each LR can reach any other through its GR neighbors.

The LRs of each full mesh are the routers of the lower level of the SPT, with end-nodes attached to each of them. The GRs, with $r_2 = 2$ connections on each SPT, are the routers of the upper level of the SPT, and stacking them leads to the creation of the SSPT (Fig. 2c).

In more detail, the *(h,l,p)-MLFM* contains l layers with $h+1$ LRs each, and p endpoints attached to each LR. The number of GRs used to perform the stacking is $R_g = \frac{h(h+1)}{2}$, and their radix is $r_g = 2l$. The radix of LRs is $r_l = h + p$. Following the description of SSPTs that can be realized with a single fixed router radix, we will only consider the case where $h = l = p$, where all routers of the MLFM have the same radix $r = 2h$ and refer from now on to the *h-MLFM*.

The h-MLFM contains h layers of $h+1$ LRs each, along with $\frac{h(h+1)}{2}$ GRs, amounting to a router count of $R = \frac{3}{2}h(h+1)$. Each LR is connected to h end-nodes, and thus the whole topology contains $N = h^3 + h^2$ end-nodes. Being an SSPT, the MLFM has a cost of 3 router ports and 2 links per end-node.

2.2.4 Two-Level Orthogonal Fat-Trees

Building SSPTs with $r_1 = r_2 = k$ requires stacking only 2 SPTs, leading to the *Two-Level k-OFT* [22, 23], a three-level indirect network (Fig. 1c). The lower layers of each of the two SPTs (which we will call $L0$ and $L2$) will consist of $R_0 = R_2 = 1 + k(k-1)$ routers each. The common upper layer of the two SPTs (which we will call $L1$) will also have the same number of switches $R_1 = R_0 = R_2 = R_L$ (due to $r_1 = r_2$ within the individual SPTs).

The interconnection pattern of the SPT that creates the k-OFT is called the *Maximal Leaves Basic Building Block of degree k* (k-ML3B) of the topology [23]. The tabular representation of the k-ML3B is a $R_L \times k$ table, whose i-th row contains all values j for which $(0,i)$ connects to $(1,j)$. An algorithm to create this representation of the k-ML3B has been described for the case where k equals a prime plus one [22]. This algorithm involves filling the table using a set of slightly modified Mutually Orthogonal Latin Squares (MOLS) [7]:

1. Fill the first row with the numbers in the range $[R_L - k, R_L - 1]$ in that order.

2. Fill the remaining empty cells of the first column with $k-1$ instances of $R_L - k$, $k-1$ instances of $R_L - k + 1$, ..., $k-1$ instances of $R_L - 1$.

3. At this point, a $k(k-1) \times (k-1)$ area of the tabular representation is unfilled. This subset is divided in k squares, each of size $(k-1) \times (k-1)$. The first is filled with the numbers 0 to $(k-1)^2 - 1$, ordered from left to right and from top to bottom. The second is filled with the transpose of the first. Each remaining square is filled with each of the $k-2$ MOLS of size $(k-1) \times (k-1)$ with elements in the range $[0, k-2]$. Finally, the i-th column of each of these $k-2$ squares is increased by $(i-1)\cdot(k-1), \forall i \in [1, k-1]$.

Table 2 shows the result of the above algorithm for the tabular representation of the 4-ML3B.

Each router of the $L0$ and $L2$ levels connects to $p = k$ end-nodes (as explained in Section 2.2.2). This results in a topology with $N = 2kR_L = 2k^3 - 2k^2 + 2k$ end-nodes, built with $2k$-radix routers. The number of routers needed is $R = 3R_L = 3k^2 - 3k + 3$. As all SSPTs, the OFT also has a cost of 3 router ports and 2 links per end-node.

2.3 Analysis

2.3.1 Scalability

Fig. 3 shows the scalability of the considered diameter-two topologies, as well as that of the three-level Fat-Tree as a reference for comparison. The table in the same figure contrasts the scale with the cost of each of these topologies.

Asymptotically, all topologies scale to a number of nodes that is proportional to the cube of the router radix with the exception of the two-level Fat-Tree which scales proportionally to only the square of the router radix. However, the exact end-node count is significantly different from topology to topology: i) among the direct topologies, a SF will be able to accommodate $\approx 27/8$ more end-nodes than a 2D HyperX build with the same-sized router; ii) among the indirect topologies, both OFT and MLFM offer significantly higher scalability than the same diameter FT, with the OFT scaling to twice as large a network than the MLFM (thus achieving a number of end-points similar to the much more costly 3-level FT).

As an example, using a radix-64 router design, the OFT can support approximately 63.5K nodes, while the MLFM and SF support around 36K and 33.7K, respectively. Thus, they are both good candidates for the interconnection network of the largest of today's datacenters or that of exascale HPC systems (e.g. a system comprising 40TFlop nodes – i.e., CORAL nodes [17], 2017 time frame – would require an interconnect that scales to $25,000$ nodes to reach peak exaflop performance).

Comparing between the best direct topology option (SF) and the best indirect topology option (OFT) is more problematic, as the choice of one over the other is highly dependent on the technology envisioned for the design. Particularly, direct topologies typically benefit from having the router integrated close to the compute chips, whereas in the case of indirect topologies routers are discrete. Thus, the choice between the two topologies is subject to a cost-scalability tradeoff.

2.3.2 Upper bound for bisection bandwidth

Due to the irregular nature of the OFT and the Slim Fly, an analytical calculation of their bisection bandwidth is not easy. However, we can approximate it for the topologies under discussion using a graph partitioning tool [10]. The approximate results (Fig. 4) suggest that the OFT benefits from the higher bisection bandwidth among the three topologies, offering $\approx 0.89b$ per end-node ($\approx 0.81b$ for small scale networks), b being the link bandwidth. Conversely, the approximate bisection bandwidth for the SF is $\approx 0.71b$ per node when $p = \lfloor \frac{r'}{2} \rfloor$ and $\approx 0.67b$ per node when $p = \lceil \frac{r'}{2} \rceil$. Finally, the MLFM seems to have the lowest bisection bandwidth of the three, being limited to $\approx 0.5b$.

2.3.3 Diversity of shortest paths

Compared to the two-level Fat-Tree, the considered diameter-two topologies (both direct and indirect) trade diversity of shortest paths for higher scalability. Nonetheless, in all of them there exist (source,destination) router pairs between which more than one shortest path exists. These pairs can potentially be leveraged by optimized routing and/or mapping policies to improve the network's performance under adversarial traffic patterns.

For the SF there is no path diversity between routers that are directly connected. However, some pairs of non-directly connected routers share more than one neighbor and thus there exists some path diversity between them. Such pairs are scarce, thus system-wide path diversity is relatively low. For instance, for $q = 23$, the average number of minimal paths between pairs of non-directly connected routers is approximately 1.1, with the maximum path diversity being 8.

The MLFM also exhibits path diversity that is irregularly distributed across the network. A pair of LRs that belong to the same router column, that is, having the same relative index in their respective layer (Fig. 1b), have h minimal routes between them. Any other pair of routers however have strictly one minimal path between them.

The OFT (and SPTs/SSPTs in general) are designed to provide high scalability through the reduction of diversity of minimal paths between routers connected to end-points. In general, only one minimal route exists between any pair of such routers, with the only exception of pairs of counterpart routers in different stacked layers. Due to the symmetry, routers $(0, i)$ and $(2, i)$ connect to the same $L1$ routers and as result there are k minimal paths between them.

3. ROUTING

We discuss routing of packets from a source node directly connected to router R_s, to a destination node directly connected to a different router R_d.

3.1 Oblivious Minimal Routing

The SF is the only one of the three topologies without constant length minimal paths. Being a direct topology, R_s and R_d can be either directly connected, in which case the path has a length of 1, or connected through a common neighbor, in which case there exists a 2-hop minimal path.

Minimal routing for the MLFM consists exclusively of 2-hop paths. The local source router R_s sends the routed packet to a global router, to be forwarded to R_d. If the communicating router pair belongs to the same column (Fig. 1b), any output port of the source leads to a global router that is connected to the destination. However, if this is not the case, there is again only one global router that is a common neighbor of both R_s and R_d.

Similarly, in minimal routing for the OFT, a packet from R_s traverses an $L1$ router to reach R_d, on a strictly 2 hop path. In the case where R_s and R_d are a symmetric pair, any $L1$ router connected to R_s can be used as the intermediate hop, since the same $L1$ routers are connected to R_d. In the opposite case, only one $L1$ router connects the source-destination pair and therefore the intermediate hop is their single common neighbor.

3.2 Oblivious Indirect Random Routing

Valiant's algorithm [24] can be used to load balance adversarial traffic patterns where minimal routing underperforms. A packet from R_s is first minimally routed to a uniformly randomly selected intermediate router R_i (other than the source and destination routers) and from there minimally routed to its final destination R_d. In the case of SF, any router in the topology is eligible to become an intermediate router. This means that indirect routes have a length of 2, 3 or 4 hops.

In the case of the OFT and the MLFM however, if any router in the topology is eligible to become R_i, indirect paths would have lengths of 2, 4 or 6 hops. However, this is not desirable, since short paths will result in inadequate load balancing and long paths will result in higher latency. Hence, for these two topologies, the intermediate destination is chosen among the routers that are directly connected to end-nodes ($L0$ or $L2$ routers for the OFT, local routers for the MLFM), restricting the indirect path length to 4 hops.

3.3 Adaptive Routing

For adaptive routing on the topologies under discussion, we explore variants of the Universal Globally-Adaptive Load-balanced (UGAL) algorithm [21], which has already been successfully used in various other topologies [4, 11]. The UGAL algorithm selects between minimal and indirect random routing on a per-packet basis, based on the channel load at the moment of the packet's injection, as conceived by the network's buffers' occupancy level. The global variant of the algorithm requires knowledge of the buffers' state for the whole topology at the point of injection, which is hard to implement in practice. Here, we only consider the local variant of UGAL, in which each router has access exclusively to information about the state of its own buffers.

In general, the generic UGAL algorithm works as follows: When a packet is injected to the network a number of possible paths is selected and each one of them is assigned a cost. The minimal path is assigned a cost C_M equal to the occupancy of the first output port of the path : $C_M = q_M$[1]. Additionally, n_I indirect routes are randomly selected, and a cost $C_I^j, j \in \{1..n_I\}$ is assigned to each one of them, calculated as follows:

$$C_I^j = c \cdot q_I^j$$

where c denotes the penalty of the selection of an indirect path over a minimal one, and q_I^j is the occupancy of the first output port of the particular path. Finally, the path with the minimum cost is selected for the routing of the packet.

The generic UGAL algorithm has the drawback that it allows packets to be routed indirectly even when the occupancy of the minimal path is low, because some indirect path starts with an empty or lower occupancy first buffer (when $q_I^j = 0$, the value of c doesn't matter). Because an indirect path is twice as long as a minimal one, we expect the packet to have an increased latency. Adding to this problem is the fact that there is a good chance that the occupancy of the first output buffer does not always accurately reflect the congestion on its links. Therefore, in our experiments, in addition to the generic UGAL, we use also a modified version of the algorithm, in which packets are routed minimally when $q_M < T$ and adaptively in the opposite case, with T being a threshold in the buffer occupancy, expressed as a percentage of the buffer size.

SF adaptive routing.

For the SF we use the generic algorithm (SF-A), but we base the cost calculation to the one of the original UGAL algorithm [21], similarly to Besta and Hoefler [4]. In this calculation, the cost of an indirect path is analogous to the ratio of the length of the indirect path (L_I^j) to the length of the minimal path (L_M). Thus, we have:

$$c = \frac{L_I^j}{L_M} \cdot c_{SF}$$

where c_{SF} is a constant, selected to balance the ratio between minimal and indirect routes. The same calculation is also used for adaptive routing with a threshold (SF-ATh).

[1]In the rare cases where multiple minimal paths exist, we can either select one of them at random, or select the minimal path with the lowest cost.

MLFM and OFT adaptive routing.

For both MLFM and OFT we use the generic UGAL algorithm (MLFM-A and OFT-A, respectively) and the version of UGAL with a threshold (MLFM-ATh and OFT-ATh) with a constant value for c.

3.4 Deadlock Freedom and Avoidance

The deadlock avoidance scheme proposed by Besta and Hoefler [4] for the SF utilizes 2 VCs in the case of minimal routing, and 4 in the case of indirect routing to effectively avoid deadlocks without restricting turns.

The OFT and MLFM topologies require less VCs than that. Both are inherently deadlock-free when minimal routing is utilized. In both cases, all uni-directional links belong in one of two groups. In the case of the OFT these groups can be characterized as *towards* or *away* from an $L1$ router, and in the case of the MLFM, they can be characterized as *towards* or *away* from a global router. In both topologies, a minimal route comprises a *towards* link followed by an *away* link, and therefore, since an order can be imposed on the classes of links so that they are always allocated in ascending order, this kind of routing entails no risk of routing deadlock [6].

On the contrary, with indirect routing the risk is present, since the routes are now of the form: *towards*, *away*, *towards*, *away*, thus forming cycles on the channel dependency graph (CDG). These cycles can be easily avoided by using 2 Virtual Channels (VCs) at each port. The first VC is used when a packet is moving towards the intermediate destination (first *towards*, *away* pair) and the second VC is used when the packet is moving away from it (second pair). This VC allocation scheme results in two virtual networks, each of which has the same cycle-free CDG as minimal routing, thus avoiding any deadlock.

4. EXPERIMENTAL RESULTS

In this section we present simulation-based performance results for the three topologies, using the routing and deadlock avoidance approaches we introduced.

4.1 Framework, parameters and metrics

The results presented in this section were obtained using a simulation framework [16] that is able to accurately model generic and custom networks at a flit level. The switch architecture chosen was that of a virtual-channel capable, input-output-buffered switch with 100 KB of buffer space per port per direction and a switch traversal latency of 100 ns. The links had a bandwidth of 100 Gbps and a latency of 50 ns. Credit based flow control was used and messages consisted of 256 byte packets.

We benchmarked several traffic patterns, both synthetic (global uniform traffic and adversarial permutation traffic) and representative of real-world applications (nearest neighbor and all-to-all communication patterns). For the synthetic traffic patterns, messages were generated continuously at link rate for the entire duration of the simulation, while for the realistic patterns, the total amount of data exchanged between any communicating pair was 512 KB for nearest neighbor and 7.5 KB (30 packets) for all-to-all. For each traffic pattern, the assignment of processes to nodes was performed contiguously with a single process per node.

For the synthetic traffic experiments, the system was simulated for 200 microseconds with a 20 microseconds warm-

up. For the realistic communication patterns, the system was simulated for the entire duration of the exchange.

The topology configurations that were used are the following:

- SF with $q = 13$, $p = \lfloor \frac{r'}{2} \rfloor = 9$, $N = 3042$, $R = 338$, $r = 28$
- SF with $q = 13$, $p = \lceil \frac{r'}{2} \rceil = 10$, $N = 3380$, $R = 338$, $r = 29$
- MLFM with $h = 15$, $N = 3600$, $R = 360$, $r = 30$
- OFT with $k = 12$, $N = 3192$, $R = 399$, $r = 24$

These were selected to approximate the number of nodes in CORAL Summit [17], a 150 PetaFlops system to be deployed in the 2017 time frame in a collaboration between IBM, NVIDIA and Mellanox.

4.2 Worst-Case Traffic

What constitutes an adversarial or worst-case (WC) workload varies from topology to topology. We consider patterns that are not end-node limited, meaning that a node does not generate a higher load than its link to the network can accommodate, nor does it receive more. In other words, patterns for which the bottleneck is in the network itself, not in the interface from the nodes to the network.

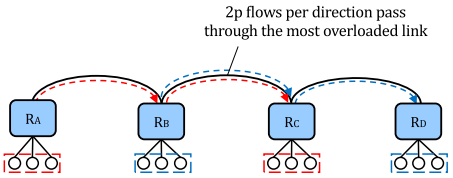
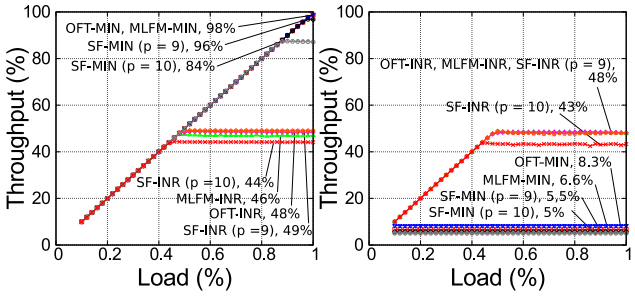


Figure 5: Worst-Case traffic for the SF is encountered when all routers of the topology communicate in pairs of distance 2, with overlapping routes.

The WC traffic pattern under minimal routing in SF is encountered when all routers communicate in pairs with a distance of 2, and pairs of the routes of this communication partially overlap. Fig. 5 shows an example of such a pair of overlapping routes. All end-nodes of R_A send their traffic to the end-nodes of R_C and all nodes from R_B send to all nodes of R_D. Thus, the link that connects R_B and R_C becomes overloaded, with $2p$ flows passing per direction, resulting to $\frac{1}{2p}$ of the maximum throughput. Arranging all traffic in the network in such pairs for our experiments is easily achieved with a greedy assignment.

The WC traffic pattern under minimal routing for the MLFM occurs when end-nodes belonging to pairs of routers connected by a single minimal path communicate exclusively across that path. Specifically, this occurs when the end-nodes connected to a router R_s send all their traffic to the endpoints of a router R_d which does not belong to same column as R_s, thus overloading the single minimal path between R_a and R_d with h flows. A particular case of this pattern is the shift traffic pattern with a shift value of h, which we use in our experiments.

The WC traffic pattern for the OFT under minimal routing is very similar to the one for the MLFM. Once again, it occurs when all end-nodes of some $L0$ or $L2$ router R_s communicate exclusively with all end-nodes of some other router R_d of the same levels, which is not the symmetrical equivalent of R_s. In this case, each link of the common path used is oversubscribed with k flows, resulting in a throughput equal to only $\frac{1}{k}$ of the link bandwidth. Once more, we use in our experiments a particular case of this pattern that is the shift traffic pattern with an offset of k.



(a) Uniform random traffic. (b) Worst case traffic.

Figure 6: Throughput, and throughput saturation points for oblivious (minimal, MIN and indirect random, INR) routing, under uniform random and worst-case traffic.

4.3 Synthetic traffic

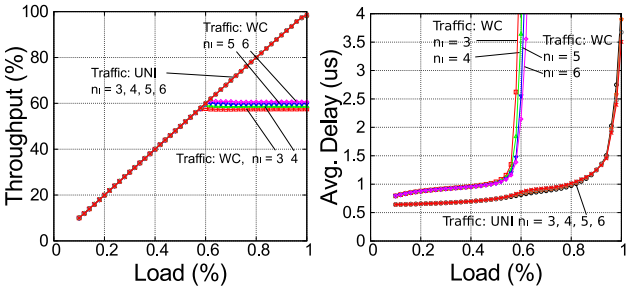
Our synthetic traffic experiments were conducted under global uniform and worst-case adversarial permutation traffic (as discussed in Section 4.2), in order to explore the limits of each topology and each routing strategy.

4.3.1 *Oblivious routing*

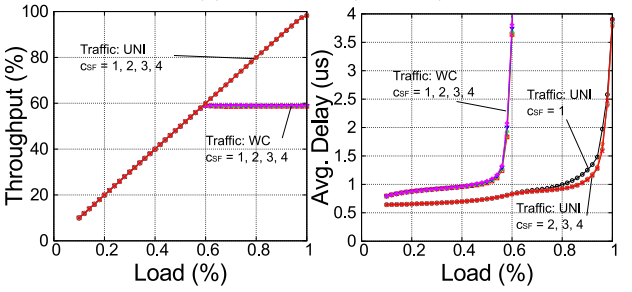
Fig. 6 shows the throughput achieved by the three topologies using oblivious routing, either minimal or indirect random. Under minimal routing, all three topologies are able to support almost full bandwidth in uniform traffic, up to approximately 96 to 98% of the total load. Between the two versions of SF, the one with higher p saturates faster, at approximately 87% of the injection rate (in accordance with the results presented by Besta and Hoefler [4]). Still, this routing strategy severely underperforms on WC traffic. As all three topologies have a low degree of minimal path diversity, they saturate at load levels as low as 5% (SF), 6.6% (MLFM), and 8.3% (OFT), equal to $\frac{1}{2p}$, $\frac{1}{h}$, and $\frac{1}{k}$ respectively, as calculated in Section 4.2. Indirect random routing helps alleviate the problem by load-balancing the network equally. However, because it effectively doubles the length of all routes, the throughput saturation point in both kinds of traffic becomes equal to half the saturation point in uniform traffic, and the average packet latency increases accordingly.

4.3.2 *Adaptive routing*

Fig. 7 shows throughput and packet delay results for the SF-A routing strategy when different values of c_{SF} and n_I are used. SF-A manages to match the performance of minimal routing under uniform random traffic, and exceeds indirect random routing for the worst-case, by adaptively selecting both minimal and indirect paths. Overall, varying the number of indirect routes considered at each packet injection (n_I) affects slightly the throughput saturation point under worst-case traffic, with higher numbers providing better results, as more available routes are available. On the other hand, c_{SF} affects the average delay under high loads

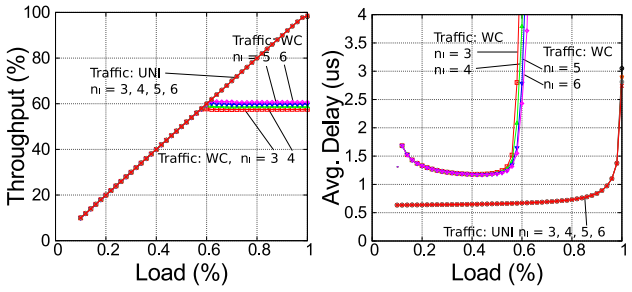


(a) Varying n_I ($c_{SF} = 1$).

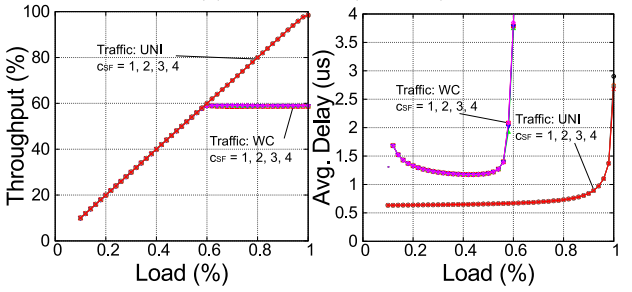


(b) Varying c_{SF} ($n_I = 4$).

Figure 7: SF-A routing on the SF with $p = \lfloor \frac{r'}{2} \rfloor$, with various values for c_{SF} and c, under uniform random (UNI) and worst case (WC) traffic.



(a) Varying n_I ($c_{SF} = 1$).



(b) Varying c_{SF} ($n_I = 4$).

Figure 8: SF-ATh routing on the SF with $p = \lfloor \frac{r'}{2} \rfloor$, with various values for c_{SF} and c, under uniform random (UNI) and worst case (WC) traffic ($T = 10\%$).

of uniform traffic. A low value of the paremeter results in indirect paths being selected easily and their increased length negatively affects the packet delay.

The drawback of the generic UGAL algorithm is apparent under uniform random traffic in the increase in latency for higher injection loads (higher than 50%). As the load increases, even with a very low occupancy on the minimal buffers, the algorithm turns to indirect routes, effectively increasing the average packet delay. The SF-ATh routing (Fig. 8) manages to alleviate this problem, keeping the delay in low levels by selecting only minimal routes. However, the threshold has a negative effect in adversarial traffic, increasing the latency for low loads, as a result of $T = 10\%$ of the packets in each port being routed minimally.

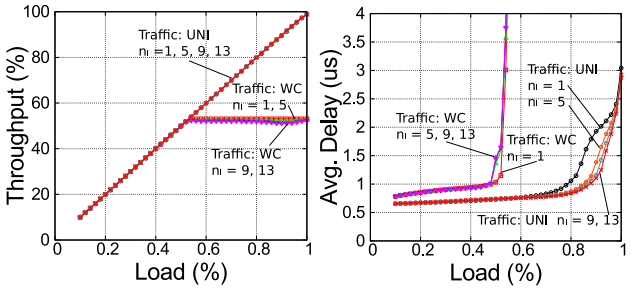
Compared to the SF-A, the effects of the MLFM-A routing algorithm on its respective topology are more sensitive to the parameters c and n_I (Fig. 9). Although the throughput achieves the levels of minimal and indirect random routing, the delay depends heavily on both parameters: higher values for c and n_I provide lower latency under uniform random traffic, meaning that the algorithm requires a variety of choices for indirect paths to work effectively, but needs to select them with strict criteria. On the contrary, lower values for c and n_I appear best in the worst-case.

With carefully selected parameters, MLFM-A does not exhibit the symptom of the the generic UGAL algorithm (higher delay in higher loads of uniform random traffic). Thus, the MLFM-Ath algorithm (Fig. 11) does not provide any significant benefit under uniform random traffic, other than parameter-independence. Nevertheless, the effect of increased delay for low-load worst-case traffic is once again apparent, similarly to the SF case.

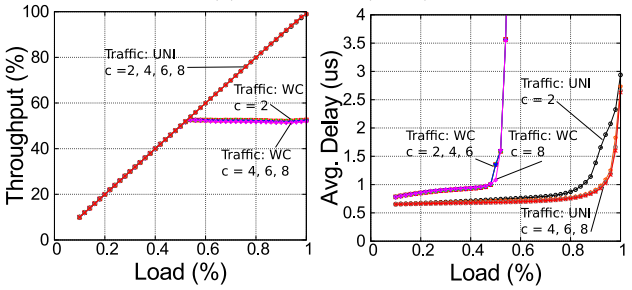
Fig. 10 shows throughput and packet delay results for the OFT-A routing strategy when different values of c and n_I are used. Contrary to MLFM-A, OFT-A offers the lowest delay under uniform random traffic when the selection of indirect paths is constricted (low values of n_I and high values of c). Nevertheless, the performance under worst-case traffic appears mostly independent of the routing algorithm parameters. Once again, adaptive routing with a threshold (OFT-ATh, Fig. 12) manages to lower the delay in uniform random traffic by trading off a higher delay in low load levels of worst-case traffic.

4.4 Exchange patterns

Synthetic traffic patterns, as informative as they are about a topology's limitations, are rarely encountered in real-life applications. As such, we also experimented with two prevalent data exchange patterns, the *All-to-All* (A2A) and the *Nearest-Neighbor* (NN) exchange. In the former, each process sends one message to each other process, meaning that N^2 messages are exchanged in total (a node is assigned a single process). For the latter, the processes are arranged in a 3D Torus (the largest one that fits in each topology), and each process sends one message to each of its 6 neighbors. The tori used have the following sizes: $12 \times 14 \times 19$ (OFT), $15 \times 16 \times 15$ (MLFM), $13 \times 13 \times 18$ (SF, $p = 9$) and $13 \times 13 \times 20$ (SF, $p = 10$). We chose a contiguous mapping, in which processes consecutive in dimension order in the application domain are mapped to consecutive end-nodes in the network. The order of the end-nodes in the network is derived from the morphology of each topology: the nodes are ordered consecutively, first at the intra-router level, then at the intra-column (SF, Fig. 1a)/intra-layer(MLFM,OFT) level and finally at the subgraph(SF)/inter-layer(MLFM,OFT) level. The routing strategies compared for each topology are the MIN, INR, and the adaptive configuration that showed the best performance under synthetic traffic.

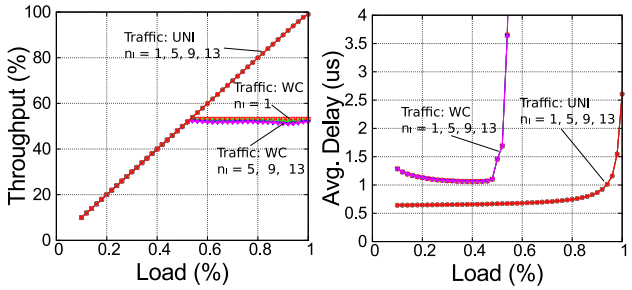


(a) Varying n_I ($c = 2$).

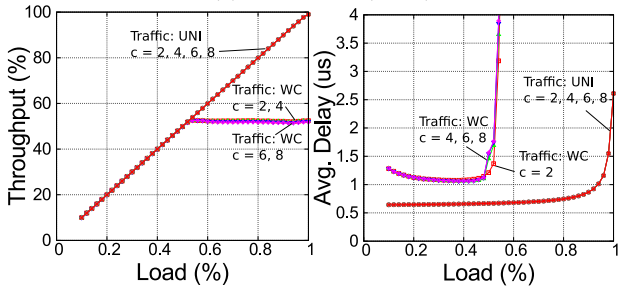


(b) Varying c ($n_I = 5$).

Figure 9: MLFM-A routing, with various values for c and n_I, under uniform random (UNI) and worst case (WC) traffic.

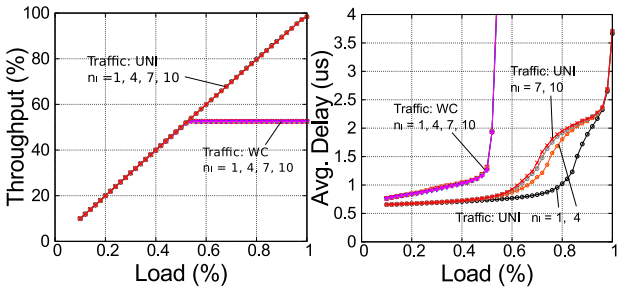


(a) Varying n_I ($c = 2$).

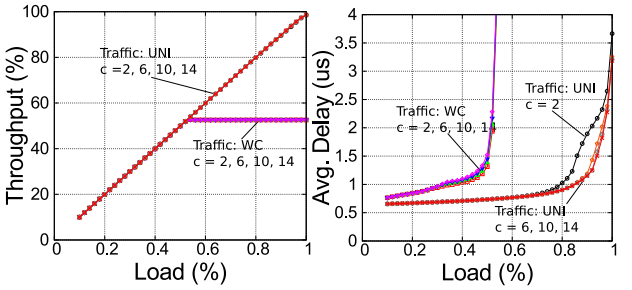


(b) Varying c ($n_I = 5$).

Figure 11: MLFM-ATh routing (various c, n_I) under uniform random (UNI) and worst case (WC) traffic ($T = 10\%$).

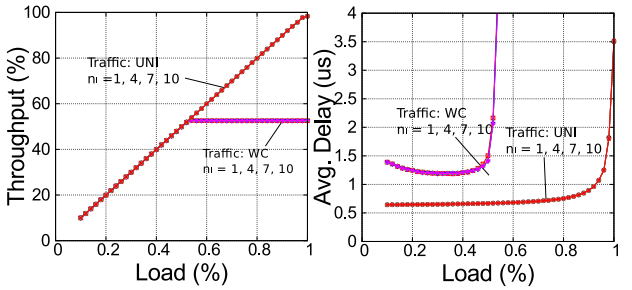


(a) Varying n_I ($c = 2$).

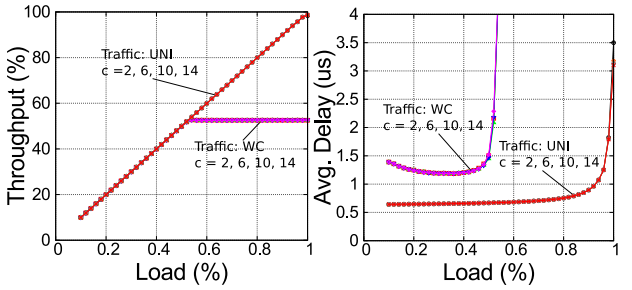


(b) Varying c ($n_I = 1$).

Figure 10: OFT-A routing, with various values for c and n_I, under uniform random (UNI) and worst case (WC) traffic.



(a) Varying n_I ($c = 2$).



(b) Varying c ($n_I = 1$).

Figure 12: OFT-ATh routing (various c, n_I) under uniform random (UNI) and worst case (WC) traffic ($T = 10\%$).

Fig. 13 shows the effective throughput achieved when performing a single A2A exchange for each topology. The exchange is performed in a manner similar to that described by Kumar et al. [12]. We calculate the effective throughput of the exchange by dividing the total amount of data exchanged by the completion time (the time interval between the moment when the first message is injected in the network and the moment when the last message in the network reaches its destination). The result is normalized per end-node and expressed as a percentage of the maximum injection bandwidth.

The performance of each of the three topologies is similar, with the SF ($p = 9$), MLFM and OFT exhibiting an effective throughput close to 100% for both minimal and adaptive routing strategies and half of that for indirect random routing, similar to what we observed before for uniform

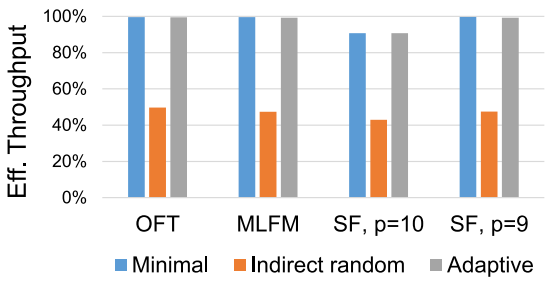


Figure 13: Effective throughput for one all-to-all exchange, with different routing strategies.

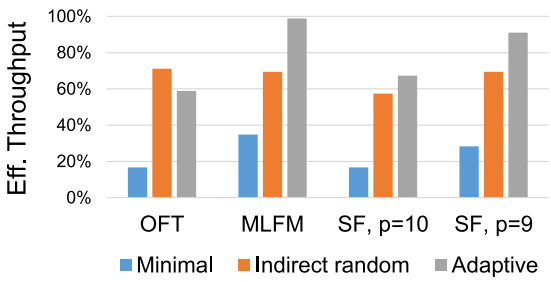


Figure 14: Effective throughput for one nearest-neighbor exchange, with different routing strategies.

traffic. The main difference between these results and those obtained for the synthetic pattern is that here we measure the performance of a fixed load (as opposed to steady state throughput) and thus we would have also captured negative tail effects should they have occurred. The fact that the effective throughput is almost identical to the steady state throughput is a strong indicator that such tail effects are negligible.

Fig. 14 shows the respective results for the NN exchange. In this exchange, minimal routing has very low performance for all topologies, as only a few routes are available to accommodate the traffic from a large number of processes. Indirect random routing achieves load balancing across the network, leading to higher effective throughput. The close to 70% effective throughput obtained is explained by the X exchanges (which stay within the first router) achieving 100% throughput while the Y and Z exchanges achieve the 50% expected of indirect random routing.

The adaptive routing schemes are generally able to improve on the performance for indirect random routing, with the exception of the OFT. For the MLFM in particular, adaptive routing achieves close to 100% effective throughput. The contiguous mapping readily maps the 3 dimensions of the 3D Torus to the three dimensional structure of the topology: the X exchanges take place inside a single router, the Y exchanges take place inside a single layer, and the Z exchanges take place across a router column. The adaptive algorithm decides to route minimally, indirectly, and minimally, respectively, achieving full bandwidth. The same effect is not witnessed on the OFT, as the Torus that would fit the topology in the same way would be the highly impractical: $12 \times 133 \times 2$ one. For the SF, fitting the Torus exactly on the topology would not be a trivial task given its complex structure, but nevertheless, the contiguous mapping is sufficient to allow the adaptive routing to outperform indirect random routing by $\approx 20\%$.

5. CONCLUSIONS

In this paper we survey the options for cost-effective diameter-two network topologies. We consider both direct topologies, in particular the Slim Fly, and indirect topologies, where we introduce a more scalable alternative to the traditional two-level Fat-Tree, the family of Stacked Single-Path Trees of which existing designs such as the Multi-Layer Full-Mesh and the two-level Orthogonal Fat-Tree are particular members. For the latter two, we introduce mechanisms for routing and deadlock avoidance.

Through theoretical analysis of the characteristics of these networks, we showed that all three exhibit several useful properties: a per-endpoint cost of only two links and three router ports (an improvement of more than 30% over the three-level Fat-Tree design), a low end-to-end network latency (at most three switch traversals for any minimal path), full global bandwidth, and reasonable scale (33K-64K end-nodes using radix-64 routers).

Furthermore, we identified for each topology adversarial traffic patterns for which performance is the lowest possible and presented mechanisms to mitigate this effect, mainly in the shape of indirect routing approaches. Finally, through detailed simulations, we showed that close to ideal levels of performance can be attained for each design for both best-case (global uniform) and worst-case (adversarial) traffic as well as good performance for traffic patterns that are representative of real world applications (nearest neighbor and all-to-all exchanges). Instrumental in achieving these performance levels was the use of adaptive routing mechanisms. We showed that with proper tuning, such mechanisms are able to handle a wide range of communication behavior by seamlessly switching from minimal to indirect routing, as needed.

All in all, we have shown that both the diameter-two Slim Fly (as the best overall direct topology) and the two level Orthogonal Fat-Tree (as the best overall indirect topology) i) exhibit good if not ideal performance across a wide range of traffic patterns, ii) have a degree of scalability that is compatible with the requirements for future datacenter and HPC interconnects, and iii) achieve this at a very low cost, especially compared to current options such as three-level Fat-Trees or Dragonflies. The choice between the two hinges mainly on the tradeoff between, on the one side, the lower cost characteristic of direct topologies (where the routers can be integrated close to the nodes), and on the other side, the factor of two higher scalability that can be achieved with OFT (surprisingly allowing it to accommodate as many end-nodes as a three level Fat-Tree at the cost of a two-level Fat-Tree).

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA) and the Army Research Laboratory (ARL) under contract W911NF-12-2-0051. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

6. REFERENCES

[1] Top 500 list. `http://www.top500.org/list/2015/06/`, June 2015. Accessed: 2015-07-27.

[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 the International Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 41:1–41:11, New York, NY, USA, 2009. ACM.

[3] B. Arimilli, R. Arimilli, V. Chung, S. Clark, W. Denzel, B. Drerup, T. Hoefler, J. Joyner, J. Lewis, J. Li, N. Ni, and R. Rajamony. The PERCS High-Performance Interconnect. In *Proceedings of the 2010 18th IEEE Symposium on High Performance Interconnects*, HOTI '10, pages 75–82, Washington, DC, USA, 2010. IEEE Computer Society.

[4] M. Besta and T. Hoefler. Slim Fly: A Cost Effective Low-Diameter Network Topology. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC14*, pages 348–359. IEEE, Nov. 2014.

[5] L. N. Bhuyan and D. P. Agrawal. Generalized hypercube and hyperbus structures for a computer network. *IEEE Trans. Comput.*, 33(4):323–333, Apr. 1984.

[6] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[7] J. Dénes and A. Keedwell. *Latin squares and their applications*. Academic Press, 1974.

[8] 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 the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, pages 103:1–103:9, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.

[9] Fujitsu Laboratories Ltd. Fujitsu Laboratories Develops Technology to Reduce Network Switches in Cluster Supercomputers by 40%. `https://www.fujitsu.com/global/about/resources/news/press-releases/2014/0715-02.html`, 2014. Accessed: 2015-04-16.

[10] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[11] J. Kim, W. J. Dally, S. Scott, and D. Abts. Technology-Driven, Highly-Scalable Dragonfly Topology. *SIGARCH Comput. Archit. News*, 36(3):77–88, June 2008.

[12] S. Kumar, A. Mamidala, P. Heidelberger, D. Chen, and D. Faraj. Optimization of MPI collective operations on the IBM Blue Gene/Q supercomputer. *Int. J. High Perform. Comput. Appl.*, 28(4):450–464, Nov. 2014.

[13] C. Leiserson et al. The network architecture of the Connection Machine CM-5. In *Proc. of the Fourth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 272–285, San Diego, CA, USA, June 1992.

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

[15] M. Miller and J. Siran. Moore graphs and beyond: A survey of the degree/diameter problem. *Electronic Journal of Combinatorics*, (DS14), 2005.

[16] C. Minkenberg, W. Denzel, G. Rodriguez, and R. Birke. End-to-end modeling and simulation of high-performance computing systems. *Springer Proceedings in Physics: Use Cases of Discrete Event Simulation: Appliance and Research*, page 201, 2012.

[17] NVIDIA. Summit and Sierra Supercomputers: An Inside Look at the U.S. Department of Energy's New Pre-Exascale Systems. `http://www.teratec.eu/actu/calcul/Nvidia_Coral_White_Paper_Final_3_1.pdf`, Nov. 2014.

[18] S. R. Öhring, M. Ibel, S. K. Das, and M. J. Kumar. On generalized fat trees. In *Proceedings of the* 9^{th} *International Parallel Processing Symposium*, page 37, Washington, DC, USA, 1995. IEEE Computer Society.

[19] F. Petrini and M. Vanneschi. k-ary n-trees: High performance networks for massively parallel architectures. In *Proceedings of the 11th International Parallel Processing Symposium*, pages 87–93. IEEE, 1997.

[20] B. Prisacari, G. Rodriguez, C. Minkenberg, and T. Hoefler. Bandwidth-optimal all-to-all exchanges in fat tree networks. In *Proceedings of the 27th International ACM Conference on Supercomputing*, ICS '13, pages 139–148, New York, NY, USA, 2013. ACM.

[21] A. Singh. *Load-balanced routing in interconnection networks*. PhD thesis, Stanford University, 2005.

[22] M. Valerio, L. Moser, and P. Melliar-Smith. Using fat-trees to maximize the number of processors in a massively parallel computer. In *Proceedings of the 1993 International Conference on Parallel and Distributed Systems*, pages 128–134, 1993.

[23] M. Valerio, L. Moser, and P. Melliar-Smith. Recursively scalable fat-trees as interconnection networks. In *Phoenix Conference on Computers and Communications*, volume 13, pages 40–46, 1994.

[24] L. G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11(2):350–361, 1982.