

Universal Routing in Distributed Networks*

Kevin F. Chen, Edwin H.-M. Sha
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75083, USA
{fxc015200, edsha}@utdallas.edu

Bin Xiao
Department of Computing
Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
csbxiao@comp.polyu.edu.hk

Abstract

We show that universal routing can be achieved with low overhead in distributed networks. The validity of our results rests on a new network called the fat-stack. We show that from a routing perspective the fat-stack is efficient and is suitable for use as a baseline distributed network. We prove that the fat-stack is universal by routing efficiency. A requirement for the fat-stack to be universal is that link capacities double up the levels of the network. We use methods developed in the areas of VLSI and processor interconnect for much of our analysis. Our universality proof shows that a fat-stack of area $\Theta(A)$ can simulate any competing network of area A with $O(\log^{\frac{3}{2}} A)$ overhead independently of wire delay. The universality result implies that the fat-stack of a given size is nearly the best routing network of that size. The fat-stack is also the minimal universal network for an $O(\log^{\frac{3}{2}} A)$ overhead in terms of number of links.

1. Introduction

An efficient network should move traffic speedily for the computing task and require no excessive hardware to build. We show that the fat-stack is such an efficient network by showing that it is *universal* in terms of routing efficiency, i.e. it can simulate any other network with an overhead of no more than (some power of) the logarithm of the area A of the hardware containing the network. This universality result implies that the fat-stack performs much better than or as well as most, if not all, of known networks.

The choice of the term “fat-stack” stems from the observation that the network is a construct of identical atomic subnetwork units stacked up and tapering upwards. The fat-stack is relatively simple in structure, which makes it scalable to closely represent a distributed network. We consider two variants of the fat-stack in this paper. The *general fat-*

stack (GFS), the focus of this paper, has only one upward link from a subnetwork and the top level node is omitted. The *augmented fat-stack* (AFS) has as many upward links as the number of nodes in the subnetwork.

Universal routing networks have been studied in the past in the context of interconnection networks [1–4]. To apply the tenets acquired there to large-scale distributed networks, it is crucial for the network to be scalable. We propose the fat-stack because its structure is amenable to scaling. The GFS is also the minimal universal network for the same asymptotic overhead. The GFS, AFS, and fat-pyramid all incur an $O(\log^{\frac{3}{2}} A)$ overhead under nonunit wire delay assumption despite expectant variation in their absolute efficiencies. The GFS uses the minimal number of nodes and links. In a previous paper [5], we have reported the results of the AFS as an efficient interconnection network where processor nodes are packed differently.

Routing schemes have direct impact on the universality of a network. The universality of the fat-stack relies on routing capability in terms of a linear combination of congestion and distance that a packet travels. This capability applies to offline routing. But online routing should have the same efficiency in the unit wire delay case due to analysis of packet routing on a “leveled network” [6, 7]. Throughout this paper, we say that network A can simulate network B with overhead μ if, for any t , the routing performed by B in time t can be performed by A in time μt .

We demonstrate that universal routing is achievable in distributed networks requiring only minimal connectivity. We provide a crucial benchmark to evaluate the performance of practical distributed networks. The GFS serves as a model network with proven efficiency and scalability for building new distributed systems.

2. Network model

The fat-stack is a hierarchical network, consisting of tiers or levels. Each level has one or more subnetworks. Each

*Work supported in part by TI University Program, NSF EIA-0103709, Texas ARP 009741-0028-2001 and NSF CCR-0309461.

subnetwork is a ring of n nodes. A graphical representation of a GFS is shown in Figure 1. A fat-stack can have arbitrary levels of rings.

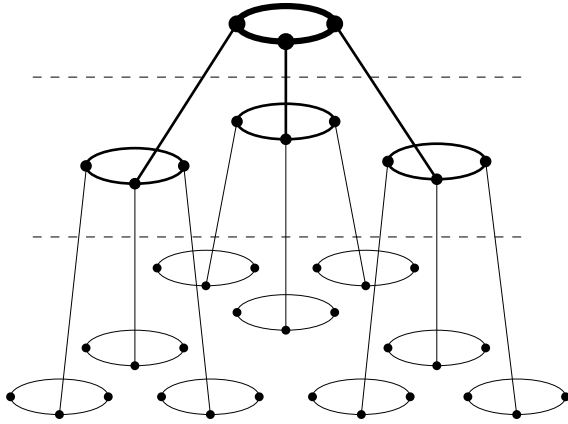


Figure 1. A fat-stack topology that has three nodes in a subnetwork. Each subnetwork connects to its upper level via a node by a single link. Dashed lines represent tier boundaries. Link capacities double upwards.

Figure 2 shows the hardware layout of a fat-tree, a fat-pyramid, and an AFS. The figure is adapted from Figure 1 of [3]. In these structures, each edge represents a wire of one unit of capacity. To increase capacity, extra edges and nodes of the same capacity are created. In analyzing distributed networks, it is often necessary to combine these separate edges and nodes to form a singular structure. In a singular network graph such as Figure 1, an edge will represent multiple units of capacities and corresponds to a channel consisting of a certain number of wires in the context of interconnection networks. The channel capacities of the networks in Figure 2 double at increasing levels of the underlying 4-ary tree inasmuch as all 4 child nodes connect to one parent node with a double redundancy at each level.

We employ an abstract model of packet routing and operation. The basic mode of operation assumed of the fat-stack is the usual distributed random-access machine (DRAM) model [8]. In a DRAM, all memory is located at the processors and its access is made by messages routed through the routing interconnection network. Indivisible packets will be used for routing analysis and they can be perceived as the basic constituents of data traffic and as a convenient scalar quantity for mathematical analysis. Large messages can be fragmented into packets.

3. Universality of the fat-stack

In this section, we prove the universality of the GFS. We first set forth the necessary postulates for the proofs. We

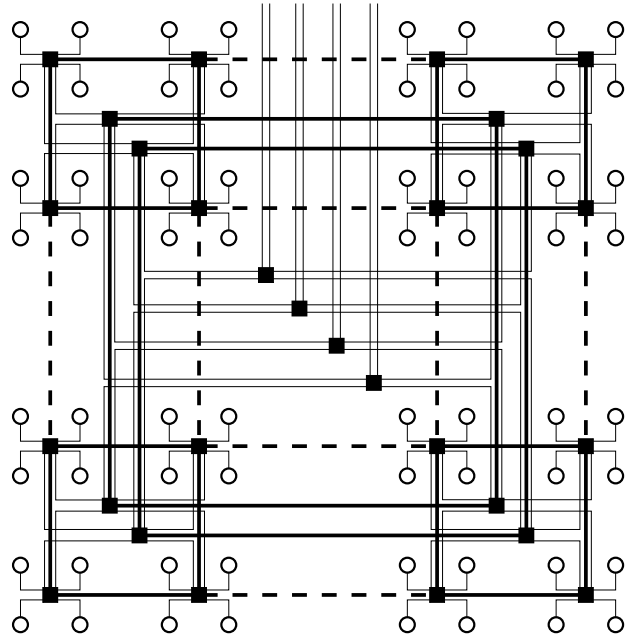


Figure 2. A fat-tree, a fat-pyramid, and an AFS in one layout. Processors are represented by circles; the squares are switches. The fat-tree links are represented by thin lines, the fat-pyramid mesh links by thick solid and dashed lines, and the AFS ring links by the thick solid lines.

then prove the universality of the GFS under both unit and nonunit wire delay conditions.

Universality proofs for the fat-tree and the fat-pyramid are parameterized by hardware area or volume. A geometric bisecting method is used to decompose the competing network to match with the structure of the base network [1, 3]. We shall adopt this method in our proofs. This method has its basis in the theories and constructions of VLSI graph layouts [9–11]. The competing network can be in terms of a cube or an area in a two dimensional design space. Extension of computation analysis from two dimensions to three dimensions has been shown to be straightforward [12, 13]. The decomposition is to recursively cut the cube or area into two pieces in the direction of the shorter edges until each piece contains either zero or one processors. It has been proven that a balanced decomposition tree can always be obtained, in which the number of processors on either side of a given node (cut) is equal to within one [1]. In view of the decomposition tree, a valid assumption is that the number of packets that can enter or leave an area (equivalently a subtree) in unit time is proportional to the perimeter of the area.

In addition, our proofs will be based on the general but powerful routing results obtained by Leighton, Maggs, and

Rao [6, 7, 14, 15]. The results pertain to offline algorithms that work under unit wire delay condition. Unit wire delay denotes that it takes unit time for a packet to move through a wire. This assumption implies that a packet traverses a distance of at most one during a single routing step or one unit time, and that at most one packet can pass through a wire during one routing step. In later analysis, wires will be considered as *transmission lines* that pipeline bits (packets) and have delay (speed) variations. The following lemma will suffice to prove the universality of the fat-stack. In the lemma, the term *congestion* refers to the maximum number of packets that must traverse a single edge in one direction, and *dilation* refers to the maximum number of edges that must be traversed by a single packet.

Lemma 3.1 ([14]). *For any set of packets with edge-simple paths having congestion c and dilation d , there is a schedule of length $O(c+d)$ requiring a maximum queue size of $O(1)$.*

It has been shown that a fat-tree can *efficiently* (i.e. in no more than polylogarithmic slowdown) simulate any network of comparable volume or area under the unit wire delay assumption [1, 2, 7, 16]. With the provisions of the bisection method, the routing results as described above, and an H-tree processor packing, it is proved in [3] that the fat-tree can simulate any network with an $O(\log A)$ overhead under unit wire delay condition, and that the fat-pyramid can simulate any network with an $O(\log A)$ overhead regardless of wire delays, provided that the base network and the competing network are of the same area A . With similar postulates, we showed in [5] that the AFS is universal with an $O(\log A)$ overhead under both unit and nonunit wire delay conditions and is in fact the minimal network for that efficiency.

We also want to make a modification of the processor packing used in [3, 5] that each bottom level node is an H-tree layout of $\log_2 A$ processors each of area $\Theta(\log A)$. In the subsequent analysis, each leaf node is thought to be of a single processor of area $\Theta(\log A)$. Also, there is a local ring for nodes in a subnetwork at the bottom level.

Using single processor leaf nodes introduces a complication on the number of processors (N) that can be packed in area $\Theta(A)$. Referring to Figure 2, we can determine N by solving the following recursion for the side length $S(N)$:

$$S(N) = 2S(N/4) + O(\sqrt{N}), \text{ and } S(1) = \Theta(\sqrt{\log A}).$$

The solution is

$$S(N) = \Theta(\sqrt{N}(\sqrt{\log A} + \log N)).$$

Since $\log N \leq \log A$ (because $N \leq A/\log A$ necessarily),

taking approximation gives

$$\begin{aligned} A &= \Theta(N(\sqrt{\log A} + \log N)^2) \\ &\leq N(\sqrt{\log A} + \log A)^2 \\ &\leq N(2\log A)^2 \\ &= 4N\log^2 A. \end{aligned}$$

Therefore, $N \geq A/(4\log^2 A)$, or $N = \Omega(A/\log^2 A)$.

One stark difference between the GFS and the AFS is that the AFS has a top level node and all subtending nodes have a link to an upper level node. To show that the GFS is universal, we retain the same framework as of the AFS and we examine the impact of the absence of a top node and extra links on the area and the number of packets on a wire in the GFS which is now directly the base network. For a fixed N , we will retain a $\Theta(A)$ area for the GFS by imagining the existence of a top node and links but treating them as only “stuffing” dummy constituents in that they do not route packets. We can apply the same equal-area concept to address that there is now only one link (the joint link) connecting a subnetwork to its upper level.

We can now focus on the routing overloading on the GFS from a competing network to arrive at the following theorem.

Theorem 3.1. *A GFS F of area $\Theta(A)$ with capacity doubling up the basis network tree can simulate any network of area A with $O(\log A)$ overhead under unit wire delay assumption.*

Proof. The framework of the GFS is kept the same as the fat-tree except it includes a dummy node and some dummy links that do not route. Matching the packets out of a perimeter of $O(\sqrt{A}/2^{\frac{l}{2}})$ of a piece of the competing network R is done in the same fashion, i.e. to a subtree of $N/2^l$ processors in F . The link capacity (i.e. the number of wires) up a subtree s_0 that has an upward link is still at least $\sqrt{N/2^l}$ as it is fixed as doubling upwards and $N = \Omega(A/\log^2 A)$ as derived in the preceding recursion solution. Thus for that subtree the link load is $O(\log A)$. Now each other subtree in the same subnetwork as s_0 routes its load $O(\log A)$ to the s_0 link through the mesh (ring) links. The s_0 link capacity, i.e. the network congestion c , is now $O(n \log A)$. (Three notes are in order. First, we could exert flow control on the subnetwork, but that would be equivalent to extending the schedule length over the unit time constraint now being considered. Second, we could increase the capacity gradient to get a smaller c , but then d would still limit the overhead to be only as good as $O(\log A)$. Third, if any second link was added from a subtree node to the upper common node, then the load of $O(n \log A)$ could be amortized, i.e. increased connectivity will reduce congestion.) Since a packet has to traverse $O(n)$ extra (ring) links, the network dilation d is now $O(n \log A)$.

This factor of n also occurs in overloading at the top level subnetwork of F . It is certainly reasonable to assume that n is fixed in all three cases. Then, by Lemma 3.1, the simulation overhead of F is $O(\log A)$. \square

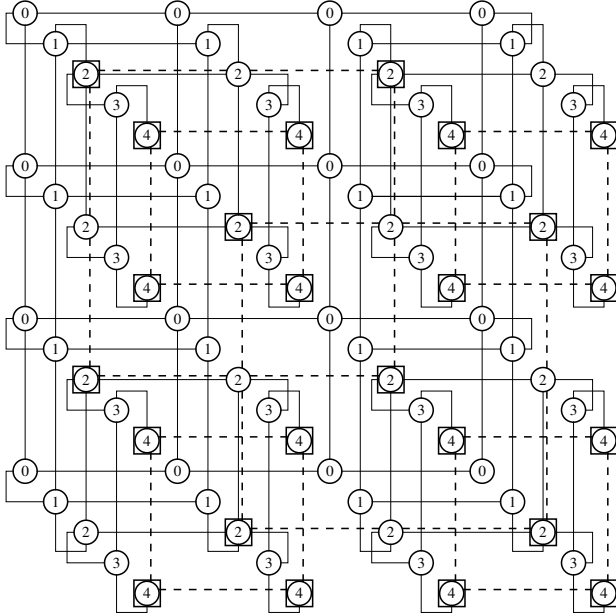


Figure 3. Regular layout of the GFS. Nodes corresponding to the nodes of the tree of meshes are represented by circles; the squares are switches. The fat-tree edges are routed through the edges of the tree of meshes shown as solid lines. The ring links are shown as dashed lines.

We now prove the universality of the GFS under nonunit wire delay assumption. We follow a similar procedure as in the fat-pyramid proof to create a regular layout which is shown in Figure 3 and is adapted from Figure 3 of [3]. The regular layout of the fat-pyramid is produced by embedding the base butterfly fat-tree into the graph of the tree of meshes, performing a “fold-and-squash” transformation, and adding the mesh connections of the fat-pyramid [3].

The layout now obtained differs from the fat-pyramid layout in the following ways: (1) level 0 will have no switch nodes; (2) level 4 switch nodes are locally connected by a ring; (3) the four processors subtending each of the level 4 switch form a ring subnetwork and have only one node (i.e. the joint node) connected to the level 4 node via a link; and (4) there is only one node in a local level 4 ring that is connected to a level 2 switch. Note that to accommodate capacity splitting, each level 4 ring must have two nodes connecting to two level 2 switches. Also note that we retained the framework of the tree of meshes and added ring

links. In the succeeding analysis, we should imagine some of the tree links are “dummy links” which affect only the n factor.

We shall need to define a linear wire delay condition in order to prove our universality theorem. Let $w(\delta)$ denote the wire delay function of wire length δ . Function w should be nondecreasing and satisfy the following condition:

Definition 3.1. A function w is said to satisfy the linear delay condition if there exists a constant c such that

$$\frac{w(\delta x)}{w(\delta)} \leq x^\gamma$$

for all $x \geq 0$, $\delta \geq c$, and $0 < \gamma \leq 1$.

Similar to the result of [3], the above condition can be met by most functions $w(\delta)$ likely to be of interest in the context of wire delay such as those in the form of $c\delta^q \log_2^k \delta$ for constants $c, q (\leq 1)$, and $k (\leq 0)$.

Theorem 3.2. Using transmission lines, a GFS F of area $\Theta(A)$ with capacity doubling up the basis network tree can simulate any network of area A with $O(\log^{\frac{3}{2}} A)$ overhead under nonunit wire delay assumption.

Proof. Let δ be the maximum physical distance that a message of a message set S travels in the competing network. The number of fat-stack edges which a message traverses is at most $2 \log_2 \delta$, plus $O(n) \cdot 2 \log_2 \delta$ ring edges at each of the $\log_2 \delta$ levels since on a subnetwork a packet traverses $O(n)$ edges to get to the joint node. Since any link connected to a switch h levels up is of length $O(2^h \sqrt{\log A})$ and each ring edge is of length $O(\sqrt{\log A})$, the routing path connecting processors at (horizontal) distance δ in the competing network is of length $O(\delta \sqrt{\log A})$. (Now each leaf node is a single processor of area $\Theta(\log A)$ instead of an H-tree block of area $\Theta(\log A) \times \Theta(\log A)$.) Therefore, each of the $2 \log_2 \delta$ fat-stack edges should contain at most $w(\delta \sqrt{\log A})$ real and imaginary switches. (Imaginary switches are auxiliary switches thought placed on each wire in number equal to the delay for that wire. Their inclusion enables us to use the unit wire delay result.) The total distance that a packet travels (hence the dilation) in F is $O(w(\delta \sqrt{\log A}) \log \delta)$.

Now let T be the time required to deliver S . We have $T \geq w(\delta)$. Also, the congestion caused by S in F is $O(T \log A)$ by the congestion argument in the proof of Theorem 3.1. The routing overhead μ can be obtained based on

Lemma 3.1 as follows:

$$\begin{aligned}
\mu &\leq O\left(\frac{T \log A + w(\delta \sqrt{\log A}) \log \delta}{T}\right) \\
&\leq O\left(\frac{T \log A}{T}\right) + O\left(\frac{w(\delta \sqrt{\log A}) \log \delta}{w(\delta)}\right) \\
&\leq O(\log A) + O((\sqrt{\log A})^\gamma \log \delta) \\
&\leq O(\log A) + O(\sqrt{\log A} \log \delta) \\
&\leq O(\log^{\frac{3}{2}} A),
\end{aligned}$$

where the third line follows from the linear delay condition (Definition 3.1). Note that when $\sqrt{\log A} < 1$, $(\sqrt{\log A})^\gamma < 1$, and then we can obtain $\mu \leq O(\log A)$. \square

Corollary 3.3. *Using transmission lines, an AFS and a fat-pyramid F of area $\Theta(A)$ with capacity doubling up the basis network tree each can simulate any network of area A with $O(\log^{\frac{3}{2}} A)$ overhead under nonunit wire delay assumption.*

Proof. For the same δ as in the proof of Theorem 3.2, the number of fat-tree edges which a message traverses is still at most $2 \log_2 \delta$, but the number of ring edges it traverses is now just 2 for either network. (The absence of n in the number of ring edges shows that these two networks can be far more efficient than the GFS.) Henceforth, the proof is the same as that for Theorem 3.2. No improvement on the asymptotic overhead is possible. \square

4. Conclusion

We have shown that universal routing can be achieved in distributed networks of minimal connectivity with low overhead. The structures of the networks must contain a GFS for these results to be applicable. Most practical distributed networks do contain a GFS-like structure though with much more links than the GFS. A network that contains a GFS with substantially higher connectivity than the GFS can be as efficient as any known network such as a hypercube. The regular layout used to prove the universality of the GFS imposes a restriction on the wire lengths (Figure 3). Workable scaling can be extension by multiples which will still abide by the area limitation. Precise scaling would be worth further study.

References

- [1] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Transactions on Computers*, vol. C-34, no. 10, pp. 892–901, Oct. 1985.
- [2] R. I. Greenberg and C. E. Leiserson, "Randomized routing on fat-trees," in *Randomness and Computation*, ser. Advances in Computing Research, S. Micali, Ed. JAI Press, 1989, vol. 5, pp. 345–374.
- [3] R. I. Greenberg, "The fat-pyramid and universal parallel computation independent of wire delay," *IEEE Transactions on Computers*, vol. 43, no. 12, pp. 1358–1364, Dec. 1994.
- [4] V. Strumpfen and A. Krishnamurthy, "A collision model for randomized routing in fat-tree networks," Technical Memo MIT-LCS-TM-629, Laboratory for Computer Science, Massachusetts Institute of Technology, 15 July 2002.
- [5] K. F. Chen and E. H.-M. Sha, "The fat-stack and universal routing in interconnection networks," in *Proceedings of the ISCA 17th International Conference on Parallel and Distributed Computing Systems*, San Francisco, CA, Sept. 2004, pp. 321–326.
- [6] T. Leighton, B. Maggs, and S. Rao, "Universal packet routing algorithms," in *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 1988, pp. 256–269.
- [7] F. T. Leighton, B. M. Maggs, A. G. Ranade, and S. B. Rao, "Randomized routing and sorting on fixed-connection networks," *Journal of Algorithms*, vol. 17, no. 1, pp. 157–205, 1994.
- [8] C. E. Leiserson and B. M. Maggs, "Communication-efficient parallel algorithms for distributed random-access machines," *Algorithmica*, vol. 3, pp. 53–77, 1988.
- [9] F. T. Leighton, "A layout strategy for VLSI which is provably good," in *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, May 1982, pp. 85–98.
- [10] S. N. Bhatt and F. T. Leighton, "A framework for solving VLSI graph layout problems," *Journal of Computer and System Sciences*, vol. 28, pp. 300–343, Apr. 1984.
- [11] S. N. Bhatt and C. E. Leiserson, "How to assemble tree machines," in *VLSI Theory*, ser. Advances in Computing Research, F. P. Preparata, Ed. JAI Press, 1984, vol. 2, pp. 95–114.
- [12] R. I. Greenberg, "Efficient interconnection schemes for VLSI and parallel computation," Ph.D. dissertation, Massachusetts Institute of Technology, Aug. 1989.
- [13] R. I. Greenberg and C. E. Leiserson, "A compact layout for the three-dimensional tree of meshes," *Applied Mathematics Letters*, vol. 1, no. 2, pp. 171–176, 1988, also see erratum in vol. 1, no. 3, p. 315.
- [14] F. T. Leighton, B. M. Maggs, and S. B. Rao, "Packet routing and job-shop scheduling in $O(\text{congestion} + \text{dilation})$ steps," *Combinatorica*, vol. 14, no. 2, pp. 167–186, 1994.
- [15] T. Leighton, B. Maggs, and A. W. Richa, "Fast algorithms for finding $O(\text{congestion} + \text{dilation})$ packet routing schedules," *Combinatorica*, vol. 19, no. 3, pp. 375–401, 1999.
- [16] P. Bay and G. Bilardi, "Deterministic on-line routing on area-universal networks," *Journal of the ACM*, vol. 42, no. 3, pp. 614–640, May 1995.