Fat-Trees

By Siddharth Singh

Overview

- Most boolean hypercube based networks require a large volume (**$n^{3/2}$**) for packaging.
- Fat-trees can be packaged in **$\Omega(n\log n)$** volume and **$O(n^{3/2})$** volume, with minimal sacrifice in the communication capacity of the network at lower volumes.
- The authors prove that a fat-tree can simulate any arbitrary routing network while incurring atmost polylogarithmic more cost.
- All of these results are shown on a theoretical model of a fat-tree, which might not be necessarily how one implements it in practice.

Introduction

- Fat trees are a class of “universal” routing networks.
- Processors at the leaf nodes.
- Switches at the internal nodes.
- Bandwidth increases when going up.

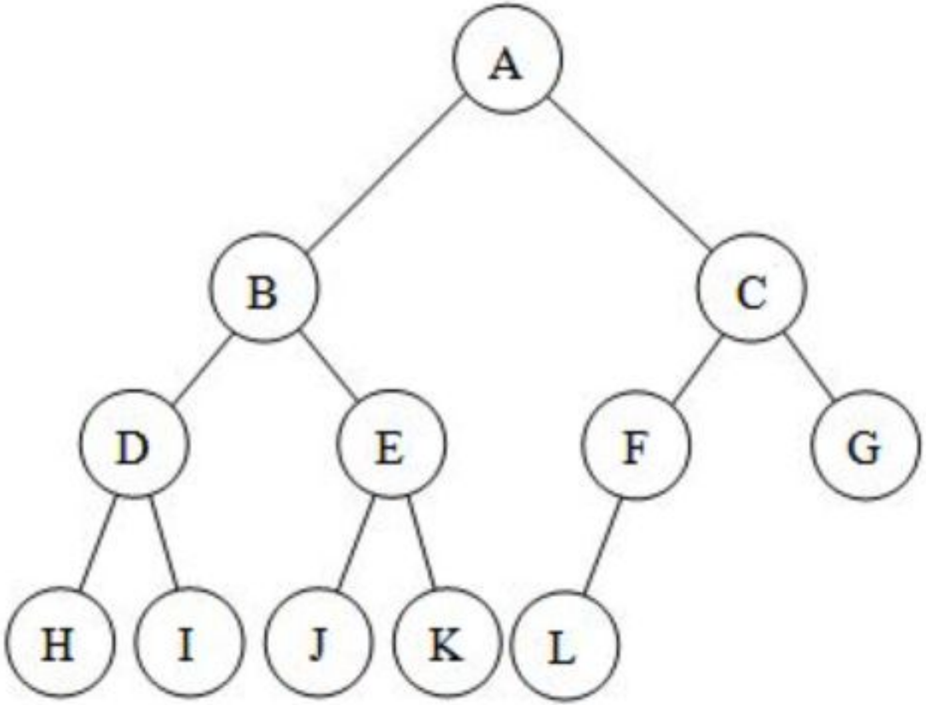


Figure 1: In this fat-tree [A-G] are switches [H-L] are processors

Terminology

- 1 edge = 2 channels (**c**) between parent and child.
- **cap(c)** = number of wires in the channel aka capacity.
- Switches at the internal nodes.
- **P** - Set of n processors

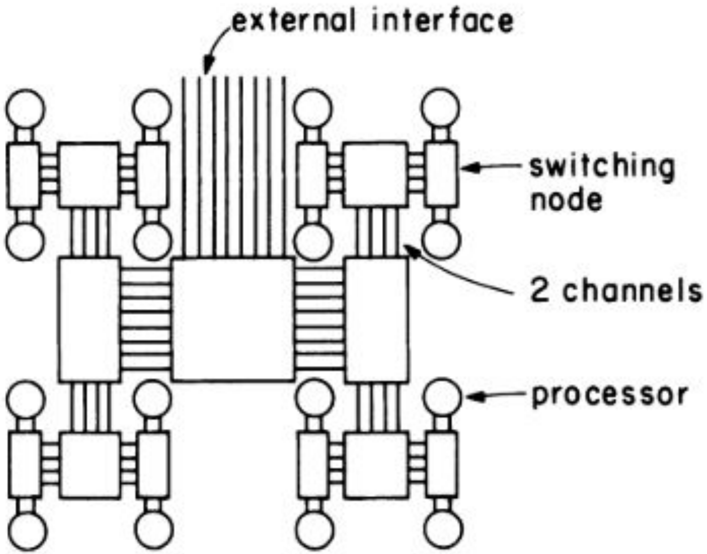


Fig. 1. The organization of a fat-tree. Processors are located at the leaves, and the internal nodes contain concentrator switches. The capacities of channels increase as we go up the tree.

Figure 2: Organisation of a fat-tree

Routing in Fat Tree

- At each node an incoming message has 2 paths - therefore 1 bit to make decision.
- Address of **2log(n)** bits. Why? (Go up and go down)
- Routing is **synchronous** and **bit serial**. Therefore routing time = **O(logn)**
- M = 1 => wire is active
- Switch uses first bit of address to make routing decision and drops it.

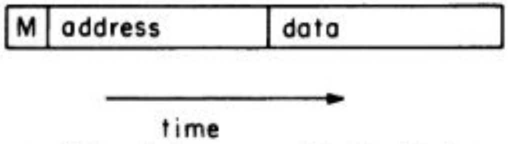
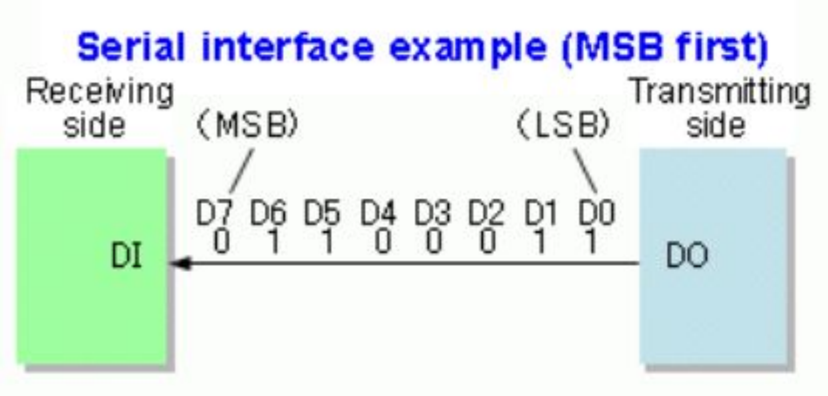


Fig. 2. The format of bit-serial messages. The first bit that a switch sees is the *M* bit, which indicates whether an input wire actually contains a message. The address bits arrive bit-serially in subsequent time steps, and the message contents are last.

What is synchronous and bit serial routing?

- Message bits are sent one by one through a wire, one bit per clock cycle.
- Thus number of wires in a channel = number of messages that can be transmitted in parallel = **cap(c)**



Congestion

- Example :- incoming channels c1 and c2 are full and all the c1+c2 messages are to be routed through c3.

- Given c1+c2>c3 (very likely as capacities increase when we go up)

- Here, c3 will not be able to transmit all the messages

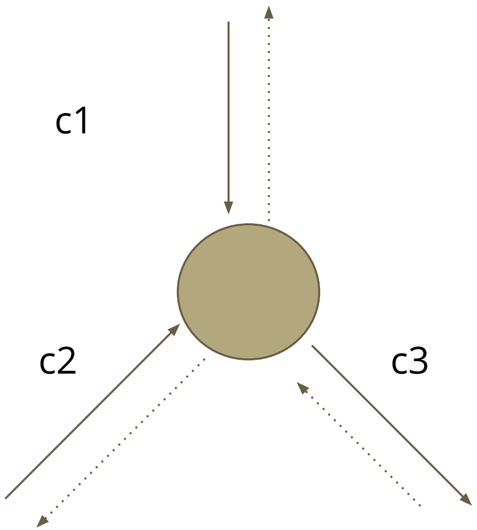


Figure 3 : A fat-tree node with capacities (c1,c2,c3). Dotted channels are not considered in this example.

Message Routing with Congestion

- Interestingly, the logarithmic guarantee still holds even when the network is congested.
- In Section 3 of the paper, the authors present an offline scheduling algorithm for Fat-trees that makes this possible.
- But first, some more terminology.

Terminology

1. Message set (**M**) - A set of messages that are concurrently transmitted through the fat-tree.
2. **load(M,c)** - The total number of messages in **M** that will pass through c.
3. One cycle message - if **load(M,c) <= cap(c)** $\forall$**c** i.e. **M** is transmitted without congestion.
4. Load factor of a channel **ƛ(M,c) = load(M,c)/cap(c)**
5. Load factor of the fat-tree **ƛ(M) =** $\mathbf{max}_c$ **ƛ(M,c)**

Offline scheduling for fat-trees

- Break **M** into a set of **d** one-cycle message sets **(M_1, M_2 .. M_d).** Then transmit each set without congestion.
- A simple lower bound on **d** is **ƛ(M).**
- The paper proves an upper bound of **O(ƛ(M) logn).**
- For channels with reasonably large capacities, they prove that the upper bound converges to **O(ƛ(M))**
- Thus, the entire message-set can be transmitted in **O(ƛ(M) logn)** time in a fat-tree.

Reasonably large?

Corollary 2: Let FT be a fat-tree on n processors, let C be the set of channels in FT, and suppose that there is a constant $a > 1$ *such that* $cap(c) \geq a \lg n$ *for all* $c \in C$*. Then for any message set M, there is an off-line schedule* $M_1, M_2, \cdots, M_d$ *such that* $d = O((a/a - 1)\lambda(M))$.

At large values of **a**, **a/(a-1)** tends to 1.

Universal fat tree

- The paper introduces a construction of fat-tree for n processors which <u>can simulate any other routing network within polylogarithmic slowdown</u> of the same volume.
- Volume = literal volume of the network. Volume is used as a proxy for hardware cost/transmission speed.
- The entire analysis is very complicated and assumes a lot of familiarity with 2D and 3D VLSI models.

Channel capacities of a universal fat-tree

- Level of a node (**k**) = minimum distance from root.
- Root capacity (**w**) = the capacity of wires coming out of the root.

Definition: Let FT be a fat-tree on n processors with root capacity w where $n^{2/3} \le w \le n$. Then if each channel $c \in C$ at level k satisfies

$$\mathrm{cap}(c) = \min\left\{\left\lceil \frac{n}{2^k} \right\rceil, \left\lceil \frac{w}{2^{2k/3}} \right\rceil\right\},$$

we call FT a *universal fat-tree*.

Unpacking this definition

- For **k < 3 log (n/w)**, the second term is lesser.
- Thus nearer to the root, the capacity drops by $\sqrt[3]{4}$ when we go down.
- Beyond **3log (n/w) levels**, the capacity drops of exponentially.
- n as the upper bound of w makes sense as at max n messages can be sent out of the network by the n processors.

Definition: Let FT be a fat-tree on n processors with root capacity w where $n^{2/3} \leq w \leq n$. Then if each channel $c \in C$ at level k satisfies

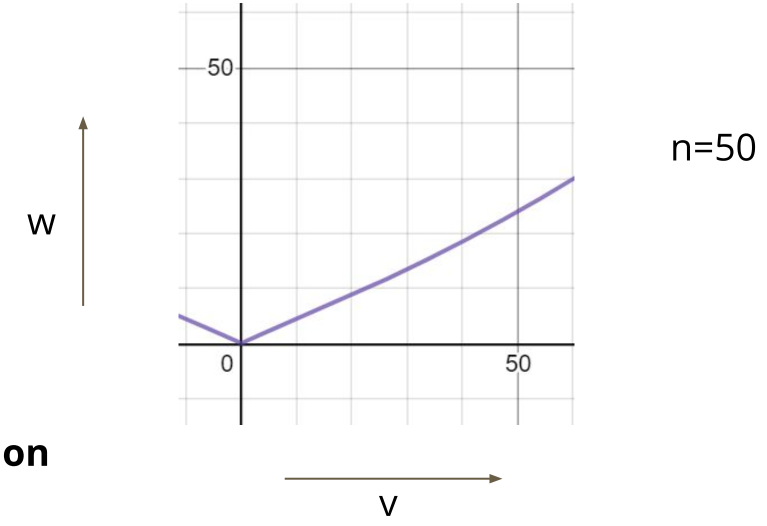
$$\operatorname{cap}(c) = \min\left\{\left\lceil \frac{n}{2^k} \right\rceil, \left\lceil \frac{w}{2^{2k/3}} \right\rceil\right\},$$

we call FT a *universal fat-tree*.

Hardware requirements for a universal fat-tree

- Volume is used as a proxy for hardware cost/transmission speed.
- The paper provides (without complete proof) a relation between the root capacity and the volume for a universal fat-tree

$$w = \theta\left(\frac{\left(v^{\frac{2}{3}}\right)}{\log\left(\frac{n}{v^{\frac{2}{3}}}\right)}\right)$$



Volume is thus an indirect measure of communication potential.

Proving universality of the universal fat-tree

- For n processors, consider a universal fat-tree routing network and an arbitrary routing network **R** of the same volume **v**.

- “If a message set **M** can be delivered by **R** in time **t**, then the fat-tree can deliver the same message set **M** in **$O(t\log^3 n)$.** The authors prove this result in the paper.

Proving universality of the universal fat-tree (cotd.)

- Reduces to O(**$t\log^2 n$**), for root capacities near to the upper bound.

- Reduces to O(**$t\log n$**), when using the routing algorithm discussed previously for reasonably large channel capacities.

Questions?