Randomized Routing on Fat-Trees

Ronald I. Greenberg
Charles E. Leiserson

MIT
Goal
Universal network for a given physical area of circuitry

Problem
Valiant’s technique does not work directly.

Bisection width argument
\[ \text{time} = \Omega \left( \frac{n}{A^{1/2}} \right) \]
Fat-Trees

2 channels

Concentrator switch

Processor

n processors

capacity of channel \( \text{cap}(c) \) = no. of wires
Universality

Any bounded-degree network can be efficiently simulated by a fat-tree of comparable hardware cost.

efficiently = polylog time degradation
hardware cost = VLSI area

Leiserson[85]: Off-line simulations

Leiserson & Greenberg: Randomized on-line simulation
Proof of Universality

Assumption: At most $O(A)$ bits can pass through a surface of area $A$ in unit time.

Competing network of volume $V$

Communication bandwidth at $i$th level:

$O\left(\left(\frac{V}{2i}\right)^{2/3}\right)$
Universality Proof (continued)

Balance the decomposition tree.
Associate with FT with exponentially growing capacities.
Message set delivered in time $t$ by competition puts $O((t \log n) \cdot \text{cap}(c))$ messages on channel $c$. 
Load Factor

arbitrary network

message set $M$

\[
\text{load}(M,c) = \# \text{messages crossing } C
\]

\[
\text{cap}(c) = \text{bandwidth of } C
\]

\[
\lambda(M) = \max_C \frac{\text{load}(M,c)}{\text{cap}(c)}
\]

The load factor $\lambda(M)$ is a lower bound on the time to deliver $M$. 
A Communications Model

- synchronous
- bit-serial
- batched (delivery cycles)
- no buffering at switches

must drop message  can deliver both messages in one cycle

- successful deliveries acknowledged
- perfect switches \((\log n \text{ depth})\)
Routing Result

With high probability, e.g. $1 - O(1/n)$, a message set $M$ can be delivered in the following no. of delivery cycles:

- **load factor**: $0 \leq \lambda(M) \leq 1$
- **delivery cycles**:
  - $1 \leq \lambda(M) \leq \lg n \lg \lg n$: $O(\lg n \lg(\lambda(M)))$
  - $\lg n \lg \lg n \leq \lambda(M)$: $O(\lambda(M))$
  - (always $O(\lambda(M) + \lg n \lg \lg n)$)

Each delivery cycle takes $\Theta(\lg^2 n)$ time (gate delays).
Routing Algorithm
Must deal with polynomial congestion.

Randomize in the choice of messages to send in each delivery cycle.

- no randomization in basic paths taken
- switch randomization unnecessary
The Basic Idea

In each delivery cycle, choose a random subset of messages to try routing.

Idea: Suppose we know $\lambda(M)$. Send each message with probability $\approx \frac{1}{\lambda(M)}$.

Pr \{sending each message\} 
$\approx \frac{1}{\lambda(M)} \leq \frac{c}{m}$

Expected # messages sent $\leq c$. 
The Halving Lemma

**Lemma:** Let $U$ be the set of currently unrouted messages, and suppose $\lambda(U) \leq \lambda$ initially. Then $O(\max\{\lambda, \log n\})$ cycles suffice to yield $\lambda(U) \leq \frac{1}{2} \lambda$ with high probability.

**Idea of Proof:** In each cycle, send each message with an appropriate probability $p$. 
Routing with Known Load Factor

Halve the load factor \( \lg(\lambda(M)) \) times.

**CASE I:** \( \lambda(M) \geq \lg n \lg \lg n \)
\( O(\lambda(M)) \) cycles suffice

**CASE II:** \( \lambda(M) \leq \lg n \lg \lg n \)
\( O(\lg n \lg(\lambda(M))) \) cycles suffice
Routing with Unknown Load Factor

Make guesses in two phases.

**PHASE 1:** Square guesses. Work = $O(\log n \log \lambda)$.

**PHASE 2:** Double guesses. Work = $O(\lambda)$.

\[
\text{guessed } \lambda
\]

\[
\begin{align*}
1 & \quad 1 \\
2 & \quad \log n \\
4 & \quad 2 \log n \\
16 & \quad 4 \log n \\
256 & \quad 8 \log n \\
\vdots & \quad \vdots \\
\log n & \quad \log n \\
\log n \log n & \quad \log n \log n \\
2 \log n \log n & \quad 2 \log n \log n \\
4 \log n \log n & \quad 4 \log n \log n \\
\quad & \quad \lambda(M) \\
\end{align*}
\]

\[
\text{work}
\]

\[
\begin{align*}
1 & \quad 1 \\
\log n & \quad \log n \\
2 \log n & \quad 2 \log n \\
4 \log n & \quad 4 \log n \\
8 \log n & \quad 8 \log n \\
\vdots & \quad \vdots \\
\log n \log n & \quad \log n \log n \\
\log n \log n & \quad \log n \log n \\
2 \log n \log n & \quad 2 \log n \log n \\
4 \log n \log n & \quad 4 \log n \log n \\
\quad & \quad \lambda(M) \\
\end{align*}
\]
Other Results

- Good off-line schedules exist

- If the channel capacities are all \( \Omega(\lg n) \), then \( O(\lambda(M)) \) cycles suffice with high probability

- Improved fat-tree
Research Directions

1. Greedy strategy
2. Lower bounds
3. Different communication models
4. Different ways of handling congestion
5. Comparison with other routing schemes
6. Better universal networks