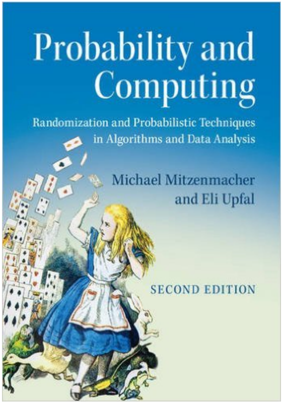
CS155/254: Probabilistic Methods in Computer Science

Chapter 4.2: Packet routing on an hypercube network



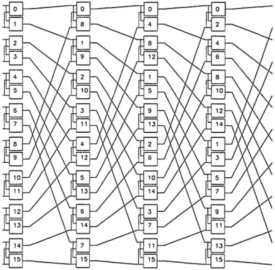
Packet Routing on Parallel Computer

Communication network:

Packet Routing on Parallel Computer

Communication network:

- nodes - processors, switching nodes;
- edges - communication links.

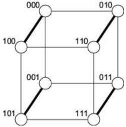


Model and Computational problem

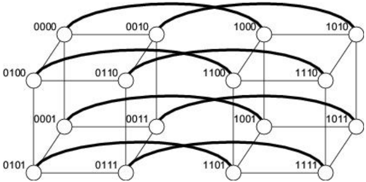
- An edge (v, w) corresponds to two directed edge, $v \to w$ and $w \to v$.
- Up to one packet can cross an edge per step, each packet can cross up to one edge per step.
- A permutation communication request: each node is the source and destination of exactly one packet.
- What is the time to route an arbitrary permutation on an N node network?

The n-cube

The 3-cube:



The 4-cube:



The n-cube

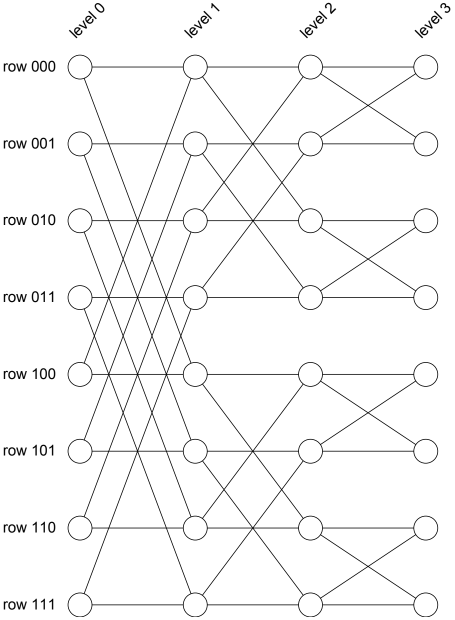
The n-cube:
$N = 2^n$ nodes: $0, 1, 2, \ldots, 2^n - 1$.

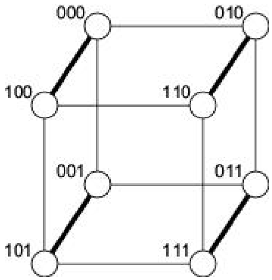
Let $\bar{x} = (x_1, ..., x_n)$ be the number of node x in binary.

Nodes x and y are connected by an edge iff their binary representations differ in exactly one bit.

Bit-wise routing: correct bit i in the i-th transition - route has length $\leq n$.

The Butterfly Network





The n-cube

The n-cube:
$N = 2^n$ nodes: $0, 1, 2, \ldots, 2^n - 1$.

Let $\bar{x} = (x_1, ..., x_n)$ be the number of node x in binary.

Nodes x and y are connected by an edge iff their binary representations differ in exactly one bit.

Bit-wise routing: correct bit i in the i-th transition - route has length $\leq n$.

Problem: Assume that a packet from $(x_1, \ldots, x_{n/2}, 0, 0, \ldots, 0)$ is routed to $(0, 0, \ldots, 0, x_1, \ldots, x_{n/2})$, for all possible assignments of $x_1, \ldots, x_{n/2}$.
We have $2^{n/2} = \sqrt{N}$ packets traversing node $(0, \ldots, 0)$.
There is an edge that is traversed by $\sqrt{N}/n$ packets.

Randomized Packet Routing Algorithm on the n-cube

Two phase routing algorithm:

1. Send packet to a randomly chosen destination.
2. Send packet from randomly chosen destination to real destination.

Path: Correct the bits, x_1 to x_n.
Queue policy: Any greedy queuing method - if a queue to an edge is not empty one packet traverse the edge.

Theorem

The two phase routing algorithm routes an arbitrary permutation on the n-cube in $O(\log N) = O(n)$ parallel steps with high probability.

Theorem

The two phase routing algorithm routes an arbitrary permutation on the n-cube in $O(\log N) = O(n)$ parallel steps with high probability.

- We focus first on phase 1. We bound the routing time of an arbitrary packet M.
- Let $e_1, ..., e_m$ be the $m \leq n$ edges traversed by packet M is phase 1.
- Let $X(e)$ be the total number of packets that traverse edge e at that phase.
- Let $T(M)$ be the number of steps till M finished phase 1.

Lemma

$$T(M) \leq \sum_{i=1}^{m} X(e_i).$$

- We call any path $P = (e_1, e_2, \ldots, e_m)$ of $m \leq n$ edges that follows the bit fixing algorithm a *possible packet path*.
- We denote the corresponding nodes $v_0, v_1, \ldots, v_m$, with $e_i = (v_{i-1}, v_i)$.
- For any possible packet path P, let $T(P) = \sum_{i=1}^{m} X(e_i)$.

- If phase I takes more than T steps then for some possible packet path P,

$$T(P) \geq T$$

- There are at most $2^n \cdot 2^n = 2^{2n}$ possible packet paths.
- Assume that e_k connects $(a_1, ..., a_i, ..., a_n)$ to $(a_1, .., \bar{a}_i, ..., a_n)$.
- Only packets that started in address

$$(*, ..., *, a_i, ..., a_n)$$

can traverse edge e_k, and only if their destination addresses are

$$(a_1, ..., a_{i-1}, \bar{a}_i, *, ..., *)$$

.

- There are no more than 2^{i-1} possible packets, each has probability 2^{-i} to traverse e_i.

- There are no more than 2^{i-1} possible packets, each has probability 2^{-i} to traverse e_i.
- $$\mathrm{E}[X(e_i)] \leq 2^{i-1} \cdot 2^{-i} = \frac{1}{2}.$$
- $$\mathrm{E}[T(P)] \leq \sum_{i=1}^{m} \mathrm{E}[X(e_i)] \leq \frac{1}{2} \cdot m \leq n.$$
- **Problem:** The $X(e_i)$'s are not independent.

- A packet is *active* with respect to possible packet path P if it ever use an edge of P.
- For $k = 1, \ldots, N$, let $H_k = 1$ if the packet starting at node k is active, and $H_k = 0$ otherwise.
- The H_k are independent, since each H_k depends only on the choice of the intermediate destination of the packet starting at node k, and these choices are independent for all packets.
- Let $H = \sum_{k=1}^{N} H_k$ be the total number of active packets.
- $$\mathrm{E}[H] \leq \mathrm{E}[T(P)] \leq n$$
- Since H is the sum of independent $0 - 1$ random variables we can apply the Chernoff bound

$$\Pr(H \geq 6n) \leq \Pr(H \geq 6\mathrm{E}[H]) \leq 2^{-6n}.$$

For a given possible packet path P,

$$\begin{aligned}
\Pr(T(P) \geq 30n) &\leq \Pr(T(P) \geq 30n \mid H \geq 6n)\Pr(H \geq 6n) \\
&+ \Pr(T(P) \geq 30n \mid H < 6n)\Pr(H < 6n) \\
&\leq \Pr(H \geq 6n) + \Pr(T(P) \geq 30n \mid H < 6n) \\
&\leq 2^{-6n} + \Pr(T(P) \geq 30n \mid H < 6n).
\end{aligned}$$

We use:

$$\begin{aligned}
\Pr(A) &= \Pr(A \mid B)\Pr(B) \\
&+ \Pr(A \mid \bar{B})\Pr(\bar{B}) \\
&\leq \Pr(B) + \Pr(A \mid \bar{B})
\end{aligned}$$

Lemma

If a packet leaves a path (of another packet) it cannot return to that path in the same phase.

Proof.

Leaving a path at the i-th transition implies different i-th bit, this bit cannot be changed again in that phase. $\square$

Lemma

The number of transitions that a packet takes on a given path is distributed $G\left(\frac{1}{2}\right)$.

Proof.

The packet has probability $1/2$ of leaving the path in each transition. $\square$

The probability that the active packets cross edges of P more than $30n$ times is less than the probability that a fair coin flipped $36n$ times comes up heads less than $6n$ times.
Letting Z be the number of heads in $36n$ fair coin flips, we now apply the Chernoff bound:

$$\begin{aligned} & \Pr(T(P) \geq 30n \mid H \leq 6n) \leq \Pr(Z \leq 6n) \\ \leq \quad & e^{-18n(2/3)^2/2} = e^{-4n} \leq 2^{-3n-1}. \end{aligned}$$

$$\begin{aligned} \Pr(T(P) \geq 30n) \quad & \leq \quad \Pr(H \geq 6n) + \Pr(T(P) \geq 30n \mid H \leq 6n) \\ & \leq \quad 2^{-6n} + 2^{-3n-1} \leq 2^{-3n} \end{aligned}$$

As there are at most 2^{2n} possible packet paths in the hypercube, the probability that there is *any* possible packet path for which $T(P) \geq 30n$ is bounded by

$$2^{2n} 2^{-3n} = 2^{-n} = O(N^{-1}).$$

- The proof of phase 2 is by symmetry:
- The proof of phase 1 argued about the number of packets crossing a given path, no “timing” considerations.
- The path from “one packet per node” to random locations is similar to random locations to “one packet per node” in reverse order.
- Thus, the distribution of the number of packets that crosses a path of a given packet is the same.

Oblivious Routing

Definition

A routing algorithm is **oblivious** if the path taken by one packet is independent of the source and destinations of any other packets in the system.

Theorem

Given an N-node network with maximum degree d the routing time of any deterministic oblivious routing scheme is

$$\Omega\left(\sqrt{\frac{N}{d^3}}\right).$$