Def n-dimensional hypercube is a <u>directed</u> graph with

- $N = 2^n$ nodes, each indexed by n-Bit integer.
- containg an edge (x, y) iff

  x and y different in only one bit.

How many edges!

$Nn. = N \log N$

<u>oblivious routing algorithm.</u>

Path from s to t depends only on s and t.

<u>permutation routing</u>

Each node is the source of one packet

— — — destination of — — — .

Eg. on a complete graph, this problem can be solved in one step.

<u>Hypercube</u> : N nodes. $N \log N$ edges

<u>Bit-Fixing Routing Algorithm for Hypercube</u>

1. Let x be the current node, and y be the destination.

2. Find smallest i such that $x_i \neq y_i$

3. Traverse the edge $(x_1 \cdots x_n, x_1 \cdots x_{i-1} \underline{\underline{y_i}} x_{i+1} \cdots x_n)$

<u>Bad example for conjection</u>

n even

<u>Transpose permutation</u>

From each x. send a packet to

$$(\underline{x_{\frac{n}{2}+1}, \cdots x_n}, \underline{x_1 \cdots x_{\frac{n}{2}}})$$

Claim Bit-fixing algorithm will take $\Omega(\sqrt{N})$ steps. on transpose permutation.

Proof: Consider nodes with $X_1 = 1$ and $x_{\frac{n}{2}+1} \cdots = x_n = 0$.

$\boxed{1 \quad\quad | \; 0000 \cdots 0}$

Their packets will use edge

$(0^n \to \cdots \boxed{0 \cdots 0 \,|\, \underline{1}\, 0000})$

stude 1

$(0^n, \; 0^{\frac{n}{2}} 1 0^{\frac{n}{2}-1})$ $\qquad n = \ln N$

$2^{\frac{n}{2}-1}$ nodes. $= \dfrac{2^{\frac{n}{2}}}{2} = \dfrac{\sqrt{N}}{2}$ $\boxtimes$

Randomized Routing Algorithm (RRA)

a. For each packet going from x to y. pick a uniformly random node z

1. Use bit fixing to route the packet from x to z
2. Use bit fixing to route from z to ~~x~~ y.

Thm For every permutation RRA. takes $O(\log N)$ steps. w.p. ~~$1-O(\frac{1}{N})$~~ $\geq 1-\frac{1}{N}$

Pf Idea. 1 Analyze Phase 1 (Phase 2 is symmetric)

Idea 2 ~~Each bit~~ In intermediate destination. each bit z_i is 0/1 uniform i.i.d

Idea 3 # step in Phase 1 $\le$ "# bits to fix" + delay (waiting time in queues)

$$\underbrace{\text{"\# bits to fix"}}_{\frac{n}{2}\ \le n}$$

<u>claim</u> consider each route in Phase 1. as a directed path from the source to the destination. Once two packets separate, they never reunite.

Q: Does it imply that two packets can not wait in more than one queue together. $\boxed{NO}$

<u>Lemma</u>: Let $P_i = (e_1, \ldots e_k)$ be the path of packet i. Let $\underline{S}$ be the set of packets (other than i) whose routes path through at least one edge of P_i. Then the delay of packet i is at most $\underline{|S|}$

☆<u>Lemma</u> Consider any packet i
It fails to reach its destination in Phase 1 ~~in~~ in $(4 \log N)$ steps with prob $\le \frac{1}{N^2}$

<u>Pf</u> Let X be the number of packets other than i that use ~~$\cdot$~~ path P_i

$$X_j = \begin{cases} 1 & \text{if packet } j \text{ uses } P_i \\ 0 & \text{o/w.} \end{cases}$$ ← independent but hard to analyze.

$$X = \sum_{j \neq i} X_j$$ <u>we want $E[X]$</u>

For any edge $\underline{\underline{e}}$. let $\underline{\underline{Y_e}}$ = # routes that pass through edge e.

$P_i = (e_1 \cdots e_k)$.

Then $X \leq Y_{e_1} + Y_{e_2} + \cdots + Y_{e_k}$.