6.896
4/5/04
L14.1

Bitonic Sorting on Butterfly (Slides 40 – 53)

$O(\lg^2 n)$ time

Routing perms by sorting

- Label each packet with dest. ID
- Sort ID's
- Packet with dest i ends up at processor i.

Works on any network capable of sorting.
But, what if some procs have nothing to send?
— Routing subpermutations.

Thm. Routing is no harder than sorting (to within $O(1)$)

Pf.
1. Give "empty" messages a label $= -1$
2. Each real message has label of dest.
3. Each proc i creates "dummy" msg with label i.
4. Sort the $2n$ msgs.

$\boxed{0}$	4	0	0
$\boxed{1}$	2	1	1
$\boxed{\cancel{2}}$	-1	2	2
$\boxed{3}$	5	3	3
$\boxed{\cancel{4}}$	-1	4	4
$\boxed{5}$	0	5	5

$\Longrightarrow$

$\boxed{\cancel{2}}$	-1
$\boxed{\cancel{4}}$	-1
0	0
$\boxed{5}$	0
1	1
2	2
$\boxed{1}$	2
3	3
4	4
$\boxed{0}$	4
5	5
$\boxed{3}$	5

5. Each real msg goes back to proc that sent adjacent dummy msg.

$\boxtimes$

Route n-perms on butterfly in $O(\lg^2 n)$ time.
In fact, can route N-subperms in $O(\lg N)$ expected time.

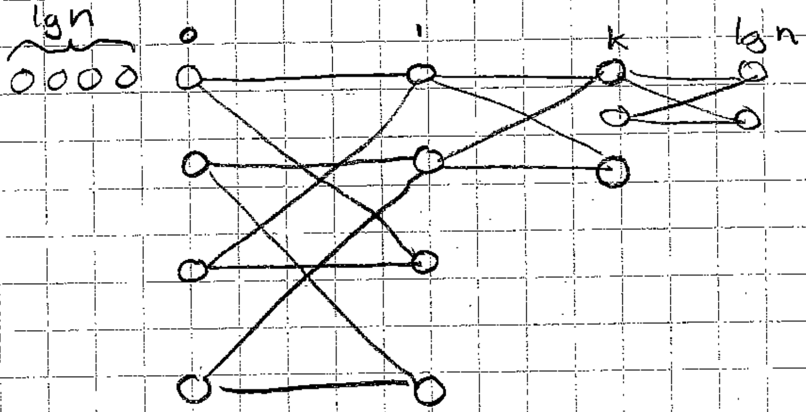
6.896
4/5/04
L14.2

<u>Theorem</u> Consider the N^N N-packet routing problems on an N-node ($n = \Theta(N/\lg N)$-input) butterfly. At least $N^N(1 - 1/N^{\Omega(1)})$ of these problems can be routed in $O(\lg N)$ time.

Proof. We'll do a congestion bound only, which leads to an $O(\lg^2 N)$-time result.

<u>Algorithm</u>

1. Route packet along row to output. $O(\lg n)$
2. Route to dest row using greedy
3. Route along row to dest node. $O(\lg n)$



Consider level-k node x during Phase 2.
packets that can reach x is $2^k \lg n$.
(tree in butterfly, Slide 27).

Prob. that given packet passes through node x
$\leq 2^{-k}$ (might not be able to reach x)

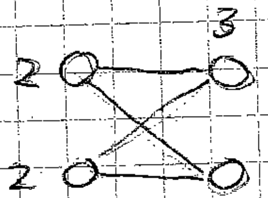
Consider any set of r specific packets.
Prob they <u>all</u> pass through node x
$\leq (2^{-k})^r = 2^{-kr}$ (independence)

Prob that $\geq r$ packets pass through node x

$$\leq \underbrace{\binom{2^k \lg n}{r}}_{\text{\# ways to choose } r \text{ packets}} \underbrace{2^{-kr}}_{\text{prob they all go through } x.}$$

Note: this overcounts. If $r+\Delta$ packets pass through x, this event is counted $\binom{r+\Delta}{r}$ times within the $\binom{2^k \lg n}{r}$ ways.

Eg.



01 10 00 00

00 00 01 10

$\binom{4}{3} = 4$

$$\leq \left(\frac{e 2^k \lg n}{r}\right)^r 2^{-kr} \qquad \boxed{\binom{a}{b} \leq \left(\frac{ea}{b}\right)^b}$$

$$= \left(\frac{e \lg n}{r}\right)^r$$

Choose $r = 2e\lg N$

$$\text{Prob} \leq \left(\frac{e \lg n}{2e \lg N}\right)^{2e\lg N}$$

$$\leq \left(\frac{1}{2}\right)^{2e\lg N}$$

$$= N^{-2e}$$

$$\leq 1/N^{5.4}$$

Prob that $\geq 2e \lg N$ packets go through <u>any</u> node

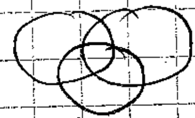
$\leq N \cdot (1/N^{5.4})$

↑ # packets

$= N^{-4.4}$

<u>Boole's Ineq.</u>

Prob of union $\leq \Sigma$



No indep. needed

$\therefore \geq N^N(1 - 1/N^{4.4})$ problems see $\leq 2e \lg N$ congestion

Thus, each level takes $O(\lg N)$ time $\times \lg N$ levels
$= O(\lg^2 N)$ time. $\boxtimes$ «Can show $O(\lg N)$ whp.»

<u>Corollary</u> $E[\text{routing time}] = O(\lg N)$.

Pf. $E[\ ] = \sum t \cdot \Pr\{\text{routing takes time } t\}$

$$\leq O(\lg N) \cdot (1 - 1/N^{4.4}) + O(N) \cdot \frac{1}{N^{4.4}}$$

$$= O(\lg N) \ \boxtimes$$

Still have bad routing problems.
Valiant: Use randomization to ensure no input prob can elicit w-c behavior (perm routing).

1. Route from source to rand intermediate dest.
2. Route from intermed. node to true dest.

Each accomplished in $O(\lg N)$ time whp.
Each is random routing prob.
No bad perm. Only unlucky choices for randomization.

Butterfly is $O(\lg N)$-universal for on-line simulations.