University of Tokyo: Advanced Algorithms **Summer 2010**

Lecture 10 — 24 June

Lecturer: Christian Sommer *Scribe: Valentina Baljak*

10.1 Permutation Routing, Part 2/2

In the previous lecture, we discussed the *permutation routing* problem on the d–dimensional hypercube ($n = 2^d$ nodes): there is a packet at each node i to be delivered to a node $\pi(i)$, where π is a permutation. The routing schemes considered are *oblivious*, meaning that each route only depends on the source and the target; in particular, routes are not allowed to have any dependency on the permutation π. We proved a lower bound on the number of time steps that the *bit-fixing* algorithm requires in the worst case. Furthermore, it can be shown that any deterministic oblivious scheme requires at least $\Omega(\sqrt{n}/d)$ rounds to route a worst-case permutation.

Using a simple randomized scheme, $O(d)$ rounds are sufficient to round any permutation. Recall how the randomized scheme works: for each node i we choose a random intermediate node $\sigma(i)$, where we first send the packet to, and then we forward the packet from there to the final target.

In this lecture, we further discuss the use of random bits. In particular, we treat randomness as a resource and we ask: *how many random bits are necessary* to route all the $n = 2^d$ messages of a permutation routing problem in $O(d)$ rounds?

The most general deterministic oblivious algorithm specifies a route for each pair of nodes, or n^2 routes in total. A randomized algorithm can be seen as a distribution over deterministic algorithms. More specifically, an algorithm that uses r random bits specifies a distribution over a set of deterministic algorithms $\{A_1, A_2, \ldots, A_{2^r}\}$. In this set, there exists at least one deterministic algorithm A_j that is used with probability at least 2^{-r}. Therefore, the expected number of steps in a worst-case scenario of any random algorithm with r random bits is at least $\Omega\left(2^{-r}\frac{\sqrt{n}}{d}\right)$.

Note that if $r \geq d/2$ then the value in the above-stated lower bound decreases to $o(1)$ and becomes meaningless. The algorithm we analyzed in the previous lecture used a random permutation σ — choosing a random permutation among n elements in the straightforward way however uses $\Theta(\log(n!))$ bits. Is there an algorithm that uses only $O(d)$ random bits *and* routes all messages in at most $O(d)$ time steps?

Theorem 10.1. *For every d there exists a randomized oblivious scheme for permutation routing on a hypercube with $n = 2^d$ nodes that uses $3d$ random bits and runs in time at* *most 15d with high probability.*

We use the *probabilistic method* [Erd63, AS00] to prove the mere existence of such a scheme (see also [MR95, Chapter 5.4]). Unfortunately, we cannot give a concrete description and implementation of the scheme. Using the probabilistic method we can argue that a set $\mathcal{U}$ contains an element with property $p(\cdot)$ as follows: if an element chosen at random (under *any* distribution) from $\mathcal{U}$ satisfies p with positive probability, then at least one element of the set $\mathcal{U}$ must satisfy p.

Proof: Recall that the randomized scheme discussed in the previous lecture works by choosing a random permutation σ among the $n!$ permutations of n elements. Since the scheme routes all the messages within $14d$ time steps with probability at least $1 - 1/n$, at most a $(1/n)$–fraction of all the $n!$ permutations is "bad," meaning that the resulting scheme may use more than $14d$ steps. In that scheme, we choose a random permutation σ from a set of $n!$ permutations, using $\log(n!)$ random bits. The proof strategy for the new scheme is to show the existence of a set of only $O(n^3)$ permutations σ_j with the property that at most an $O(1/n)$–fraction of all the permutations σ_j is "bad" for any routing demand π.

Let us choose a set S of n^3 permutations σ_j among n elements at random. Now, after having chosen S, we fix any routing demand (a permutation) π_i. For demand π_i, each of the permutations σ_j is "bad" with probability at most $1/n$. Let x_{ij} be the indicator random variable that is 1 if σ_j is "bad" for π_i and 0 otherwise. We now give a bound on the probability that the number of "bad" permutations is more than $2n^2$. Formally, we wish to bound $\mathsf{Pr}[\sum_j x_{ij} \geq 2n^2]$. Note that $\mathsf{E}[\sum_j x_{ij}] \leq n^2$, and that the x_{ij} are pairwise independent. Using a Chernoff bound, we obtain $\mathsf{Pr}[\sum_j x_{ij} \geq 2n^2] \leq \exp\{-n^2/4\}$.

Let us now define another indicator variable B_i that is 1 if for a demand π_i more than $2n^2$ permutations σ_j are "bad." What is the probability that for a random set S less than $2n^2$ permutations σ_j are bad for *all* demands π_i? The goal is to show that this probability is *strictly less than 1* and then, using the probabilistic method, to argue that there *exists* a set S among whose elements σ_j for any demand π_i at most an $O(1/n)$–fraction is "bad."

We sum over all demands π_i to obtain

$$\mathsf{Pr}\left[\bigcup_{i=1}^{n!} S \text{ is ``bad'' for } \pi_i\right] \leq \sum_{i=1}^{n!} \mathsf{Pr}\left[B_i\right] \leq n^n \cdot \exp\{-n^2/4\} < 1.$$

Since a random set S of n^3 permutations is "bad" for at least one π_i with probability strictly less than 1, there *exists* a set $\bar{S}$ that is "good" for all π_i. Consequently, there exists a routing scheme with randomness $O(d)$ and success probability $1 - O(1/n)$. □

Bibliography

[AS00] Noga Alon and Joel Spencer. *The Probabilistic Method (second edition)*. John Wiley Interscience Series in Discrete Mathematics and Optimization, 2000.

[Erd63] Paul Erdős. On a combinatorial problem, I. *Nordisk Matematisk Tidsskrift*, 11:5–10, 1963.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.