CS 388R: Randomized Algorithms Fall 2014

Lecture 12 — Oct. 12, 2015

Prof. Eric Price *Scribe: Qi Lei, Jiong Zhang*

1 Overview

In the last lecture we discussed Perfect Hashing and Bloom Filters.

In this lecture we removed from hashing to routing, and we are going to use randomization to prove routing.

2 Introduction

Suppose we have some network, and can communicate to each other.
For an arbitrary graph, what a reasonable model for a conversation to happen here?
Suppose we have

- synchronous messages, and
- each edge can transmit one message per time step
- Sending k messages over a link takes k time.

What is a proper problem to analyze?

- If one node wants to communicate with other node, one just broadcasts to everyone.
- What makes it hard is if multiple nodes want to send messages. However, if every node want to communicate with one same node, the bottleneck would happen there so it's not interesting either.

That's why we want to discuss permutation routing:

3 Permutation Routing

Each node i wants to send a message to $\Pi(i) \in [N]$, $\Pi(i)$ is a permutation.
Question: how long does it take to route all messages?
Ideally: oblivious routing:

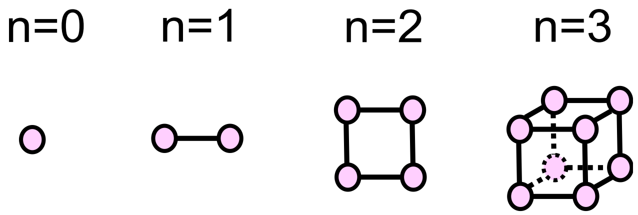

Figure 1: "hypercubes" for $n = 0, 1, 2, 3$

$$\text{Path for } i \to \Pi(i), \qquad P \text{ is independent of } \Pi(-i)$$
$$P_i = (e_1, \cdots, e_l)$$
$$e_j \text{ is an edge: } l \text{ is length}$$

3.1 Hypercube Graph

$N = 2^n$: each vertex is indexed by n bit string.
edge i to j exists if i and j have exactly one bit difference.
Diameter $n \Longrightarrow$ ideally $O(n)$ time to route.

$$\begin{aligned} \mathbb{E}[\text{distance between } i \text{ and } \Pi(i)] &= \frac{n}{2} \\ \mathbb{E}[\text{total number of messages passed}] &\geq \frac{Nn}{2} \end{aligned}$$

can hope for N messages/round
so hope for $O(n)$ rounds.

Bit fixing routing(an naive thing to do):
fix bits left to right

$$\begin{array}{ll} \Pi(0101101) = & 1101110 \\ \downarrow \quad 1101101 & \\ \downarrow \quad 1101111 & \\ \downarrow \quad 1101110 & \end{array}$$

However, $\exists$ permutations such that bit fixing takes $\Omega(\sqrt{N})$ time steps.

$$\begin{array}{ccc} \overbrace{X}^{\lfloor n/2 \rfloor} & 0 & \overbrace{00\cdots0}^{\lfloor n/2 \rfloor} \\ 00\cdots0 & 1 & X \end{array}$$

So after weight(X) times of bit fixing, where weight$(X) = \sum_{i=1}^{n} X_i$, all the messages come to the same position

$$\begin{array}{lll} \overbrace{00\cdots0}^{\lfloor n/2\rfloor} & 0 & \overbrace{00\cdots0}^{\lfloor n/2\rfloor}, \text{ next step for every node goes to} \\ 00\cdots0 & 1 & 00\cdots0, \text{and final destination is} \\ 00\cdots0 & 1 & \underbrace{X}_{\lfloor n/2\rfloor} \end{array}$$

So for all X, the messengers will cross the same edge e from $00\cdots0$ to $\overbrace{00\cdots0}^{\lfloor n/2\rfloor}1\overbrace{00\cdots0}^{\lfloor n/2\rfloor}$. There are $2^{\lfloor n/2\rfloor}$ paths cross e, causing $\sqrt{N}$ time of queueing.
Claim: for all deterministic oblivious algorithms, $\exists\Pi$ s.t. it takes $\Omega(\sqrt{\frac{N}{n}})$ times.

3.2 Randomized routing algorithm: $O(n)$ time

Theorem 1. *Part I: average case: bit fixing takes time $O(n)$ with $1-\frac{1}{N^c}$ prob for random $\Pi \in [N]^N$.*

Theorem 2. *Part II: $\forall n$, one can get $O(n)$ with probability $1-\frac{1}{N^c}$ for some routing algorithm.*

Proof of Part I:
Suppose permutation $\Pi \in [N]^N$ uniformly at random.
Define $L(e)$: load of edge e, i.e. # path using e.

$$\begin{array}{rl} \because & \mathbb{E}[L(e)] \text{ are equal for different } e. \\ \text{and} & \mathbb{E}[\text{total length of all paths}] = \dfrac{Nn}{2} \\ \therefore & \mathbb{E}[L(e)] = \dfrac{\mathbb{E}[\text{total length}]}{\#\text{ edges}} = \dfrac{Nn/2}{Nn} = \dfrac{1}{2}. \end{array}$$

- Concentration bound for load $L(e)$:
  Time for a path $p = e_1, e_2, \cdots, e_l \leq \sum_{i=1}^{l} L(e_i)$. $\mathbb{E}[T(i)] \leq \frac{n}{2}$.
  $\mathbb{E}[L(e)] = \frac{1}{2}$.
  $L(e) = \sum_{i\in[N]} H_{ie}$, where H_{ie} stands for event $e \in P_i$. H_{ie} are independent, bounded in $[0,1]$.

  1. Chernoff:
     $Pr[L(e) > \frac{1}{2} + t] < e^{-\frac{2t^2}{N}}$
     This is a terrible bound as t at least need to be comparable to $\sqrt{N}$ for a small probability. This is because Chernoff bound is not tight enough for event with small probability.

  2. Bernstein:

     Lemma 3. *Let $X = \sum_i X_i, X_i \in [0,1]$, independent. $Pr[X \geq t] \leq 2^{-t}, \forall t \geq 2e\mathbb{E}[X]$.*

Proof of Lemma 3:

$$\begin{aligned}
& \mathbb{E}[X_i] = p_i, Var(X_i) = \mathbb{E}[X_i^2] - \mathbb{E}[X_i]^2 \leq \mathbb{E}[X_i \cdot 1] - p_i^2 = p_i(1-p_i) \leq p_i, \\
\implies & X_i \text{ is } (\theta(p_i), \theta(1)) \text{ subgamma (in terms of } (\sigma^2, B)). \\
\implies & X \text{ is } (\theta(\sum_i p_i), \theta(1)) = (\theta(\mathbb{E}[X]), \theta(1)) \text{ subgamma.} \\
& Pr[X \geq \mu + t] \leq \max\{e^{-\frac{t^2}{\theta(\mathbb{E}[X])}}, e^{-\theta(t)}\}, \mu = \mathbb{E}[X] \\
& \therefore \text{ for } t > c\mu, Pr[X \geq t] \leq e^{-\theta(t)}
\end{aligned}$$

Continuing Proof of Part I:

$$\begin{aligned}
& Pr[L(e) \geq t] \leq 2^{-t}, \forall t \geq e, \\
\implies & Pr[L(e) \geq 3n] \leq \frac{1}{N^3} \\
\implies & Pr[\max L(e) \geq 3n] \leq \frac{n}{N^2} < \frac{1}{N}
\end{aligned}$$

Also, noticed that $l \leq n$, we have $\sum_{i=1}^{l} L(e_i) \leq 3n^2$ with probability at least $1 - \frac{n}{N^3}$, thus

$$Pr[\text{total time} \leq 3n^2] \geq 1 - \frac{n}{N^2}$$

which we are not satisfied with. So try another way. Instead of bound load, we bound colliding packets instead. Define

$$S_i = \{j | P_j \cup P_i \neq \emptyset\}$$

Lemma 4. *Let T_i be the time of path i,we have $T_i \leq n + |S_i|$.*

This lemma claims for path i,each intersecting path delays it for at most one time step.
Given Π random, for fixed P_i we know that $Pr[P_i \cap P_j \neq \emptyset]$ is independent of $Pr[P_i \cap P_k \neq \emptyset]$. Also

$$\begin{aligned}
\mathbb{E}[|S_i|] & \leq \mathbb{E}[\#(P_j, e \in P_j \text{ s.t. } e \in P_i)] \\
& \leq \mathbb{E}[\sum_{i=1}^{l} L'(e_i)] \\
& = \frac{l}{2} \leq \frac{n}{2}
\end{aligned}$$

Note that this $L'(e)$ denotes the # of path that go through e except for P_i.
Therefore $\forall e, \mathbb{E}[L'(e)] \leq \frac{1}{2}$.

Bounding $|S_i|$:

$$\begin{aligned} Pr[|S_i| \geq t] &\leq 2^{-t}, \forall t \geq en. \\ \Longrightarrow Pr[|S_i| \geq 4n] &\leq \frac{1}{N^4} \end{aligned}$$

By Lemma. 4 we have

$$\begin{aligned} Pr[T_i \leq 5n] &\geq 1 - \frac{1}{N^4} \\ \Longrightarrow Pr[\max T_i \leq 5n] &\geq 1 - \frac{1}{N^3} \end{aligned}$$

□ of part I.

Proof of Part II:
The problem here is we don't get random Π anymore, we will retrieve the randomness by randomly picking a mid-state. Note that the routing process is reversible.

- Given Π adversarial, for each node i choose $\sigma(i) \in [N]$ i.i.d;
- Send $i \to \sigma(i)$, wait until 5n rounds are complete;
- Send $\sigma(i) \to \Pi(i)$, wait until another 5n rounds are complete;

Thus the whole process success with prob $\geq 1 - \frac{2}{N^3}$.

References

[MR] Rajeev Motwani, Prabhakar Raghavan Randomized Algorithms. *Cambridge University Press*, 0-521-47465-5, 1995.