

## Lecture 9 — Sep. 28, 2017

*Prof. Eric Price**Scribe: Changyong Hu***NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS**

## 1 Overview

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

In this lecture we moved from hashing to routing, and we are going to use randomization to improve routing.

## 2 Introduction

Suppose we have a network modeled by a directed graph with  $N$  nodes and they can communicate with each other by sending messages through edges in the graph. Consider the communication model that each node has a message for some nodes, all communication among nodes proceeds in a sequence of synchronous round. In each round, each edge can only pass one message or packet. So sending  $k$  messages over an edge takes  $k$  units of 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.

So we discuss permutation routing in this class:

## 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:

An oblivious routing algorithm for the permutation routing problem satisfies the following property: the route followed by  $v_i$  depends on  $d(i)$  alone, and not on  $d(j)$  for any  $j \neq i$ .  $v_i$  is the packet from

node  $i$ ,  $d(i)$  is the destination of packet from node  $i$ .

$$\begin{aligned} \text{Path } P_i \text{ for } i \rightarrow \Pi(i), \quad P_i \text{ is independent of } \Pi(-i) \\ P_i = (e_1, \dots, e_l) \\ e_j \text{ is an edge: } l \text{ is length} \end{aligned}$$

### 3.1 Hypercube Graph

$N$  vertices:  $N = 2^n$ , each vertex is indexed by  $n$  bit string.  
 edge  $i$  to  $j$  exists if  $i$  and  $j$  have exactly one bit difference.  
 Diameter  $n \implies$  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.

Can we get  $O(n)$  time to resolve any permutation?

message that is targeted at node  $j$ , currently most significant bit  $b$  of  $i \oplus j$ , send  $i \rightarrow i \oplus 2^b$   
 Bit fixing routing:  
 fix bits left to right

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

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

$$\begin{array}{ccc} \underbrace{\phantom{X}}_{\lfloor n/2 \rfloor} & 0 & \underbrace{\phantom{00\dots 0}}_{\lfloor n/2 \rfloor} \\ X & & 00\dots 0 \\ 00\dots 0 & 1 & X \end{array}$$

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

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

So for all  $X$ , the messengers will cross the same edge  $e$  from  $00\cdots 0$  to  $\overbrace{00\cdots 0}^{\lfloor n/2 \rfloor} \overbrace{100\cdots 0}^{\lfloor n/2 \rfloor}$ . Pick  $X$  to have exact  $n/4$  1s in it,  $\binom{n/2}{n/4} \approx \sqrt{N/n}$

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

- (I) Bitfixing is good for random assignment
- (II) Reduce arbitrary case to random case

Define  $L(e)$ : # paths that use edge  $e$ . The expectation for  $L(e)$  is the same for any  $e$ .

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

Time for a path  $p = e_1, e_2, \dots, e_l \leq \sum_{i=1}^l L(e_i) \leq n \cdot \max_e L(e)$ .

Let  $L(e) = \sum_{i \in [N]} H_{ie}$ , where  $H_{ie}$  stands for event  $e \in P_i$ .  $H_{ie}$  are independent, bounded in  $[0, 1]$ . Now we can apply chernoff bound and then union bound.

$$Pr[L(e) > t] < 2^{-t} \quad \forall t \geq 6$$

Using union bound, we have  $\max_e L(e) \leq O(N)$  w.h.p, but we need a constant bound for  $\max_e L(e)$ .

**Lemma 1.** *Let  $T_i$  be the time of path  $i$ , we have  $T_i \leq n + |S_i|$ .*

*Proof.* Let the path be  $\rho_i = (e_1, e_2, \dots, e_k)$ . For a fixed  $i$ ,  $S$  is set of other paths that intersects  $i$ 's path. If a packet is ready to follow edge  $e_j$  at time  $t$ , we define its *lag* at time  $t$  to be  $t - j$ . The *lag* of  $v_i$  is initially zero, and the delay incurred by  $v_i$  is its *lag* when it traverses  $e_k$ . We will show that each step at which the *lag* of  $v_i$  increased by one can be charged to a distinct member of  $S$ . When the *lag* of  $v_i$  increases from  $l$  to  $l + 1$ , there must be at least one packet from  $S$  that wishes to traverse the same edge as  $v_i$  at that time step. Let  $t'$  be the last time step at which any packet in  $S$  has *lag*  $l$ . Thus there is a packet  $v$  ready to follow edge  $e_{j'}$  at  $t'$ , such that  $t' - j' = l$ . We know that some packet of  $S$  leaves  $\rho_i$  at  $t'$ . Since  $v$  is ready to follow  $e_{j'}$  at  $t'$ , some packet  $\omega$  in  $S$  follows  $e_{j'}$  at  $t'$ . Now  $\omega$  leaves  $\rho_i$  at  $t'$ ; if not, some packet will follow  $e_{j'+1}$  at step  $t' + 1$  with *lag* still at  $l$ , violating the maximality of  $t'$ . We charge to  $\omega$  the increase in the *lag* of  $v_i$  from  $l$  to  $l + 1$ ; since  $\omega$  leaves  $\rho_i$ , it will never be charged again because paths can't rejoin. Thus, each member of  $S$  whose path intersects  $\rho_i$  is charged for at most one delay, establishing the lemma.  $\square$

This lemma claims for path  $i$ , each intersecting path delays it for at most one time step.

Given  $\Pi$  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

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

Note that this  $L'(e)$  denotes the # of path that go through  $e$  except for  $P_i$ .

Therefore  $\forall e, \mathbb{E}[L'(e)] \leq 1$ .

Bounding  $|S_i|$ :

$$\begin{aligned}
Pr[|S_i| \geq t] &\leq 2^{-t}, \forall t \geq 2en. \\
\implies Pr[|S_i| \geq 6n] &\leq \frac{1}{N^6}
\end{aligned}$$

By Lemma. ?? we have

$$\begin{aligned}
Pr[\max_i S \leq 6n] &\geq 1 - \frac{1}{N^5} \\
\implies Pr[\max_i T \leq 7n] &\geq 1 - \frac{1}{N^4}
\end{aligned}$$

The problem here is we don't get random  $\Pi$  anymore, we will retrieve the randomness by randomly picking a mid-state. Note that the routing process is reversible.

- Given  $\Pi$  adversarial, for each node  $i$  choose  $\sigma(i) \in [N]$  i.i.d;
- Send  $i \rightarrow \sigma(i)$ , wait until  $7n$  rounds are complete;
- Send  $\sigma(i) \rightarrow \Pi(i)$ , wait until another  $7n$  rounds are complete;

Thus the whole process success with prob  $\geq 1 - \frac{2}{N^4}$ .

## References

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