CS 388R: Randomized Algorithms, Fall 2023 September 25, 2023

Lecture 10: Routing

Prof. Eric Price *Scribe: Sam Ziegelbein, Pranav Venkatesh*

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

1 Routing Setup

Suppose we want to pass messages between nodes in a network. Let's say that each node i wants to send a message to some node $\pi(i)$. We adopt a synchronous model of the network, i.e. at each time-step, at most one message can pass along any given edge. Our goal is then to minimize the total amount of time T it takes for *all* messages to reach their destinations. In particular, we'd like to produce an *oblivious* routing, that is the $i \to \pi(i)$ route depends only on i and $\pi(i)$ and not on the rest of the routing.

For this lecture, we'll consider the special case of permutation routing on the hypercube. Permutation routing means that π is guaranteed to be a permutation, so no two nodes will send a message to the same destination. The hypercube graph consists of $N = 2^n$ nodes, each of which is labeled by n bits. There is an edge between nodes x and y if and only if their labels differ by exactly one bit. In particular, this means each node has degree n.

1.1 Some Lower Bounds

Observe that the diameter of the graph is a lower bound on the time needed for routing, since a message sent between vertices at distance d requires at least d time to traverse any path between them. The hypercube graph has diameter $\Theta(n)$, so T is $\Omega(n)$. In this lecture we will show a matching (randomized) upper bound. That is, for any permutation π on the nodes of the hypercube, we will show how to randomly construct a routing requiring only $O(n)$ time with high probability.

Note also that we can try to obtain a somewhat different lower bound as follows: there are N total nodes, and for each node i, there is a path from i to $\pi(i)$ of length at most n. Thus in the worst case, there is $\Theta(Nn)$ "work" to do, i.e. there are $\Theta(Nn)$ total edge traversals by packets. As there are $\Theta(Nn)$ total edges, we can do at most $\Theta(Nn)$ work in a single time-step. Thus we get that T is $\Omega(1)$, a completely useless lower bound.

2 The Bit Fixing Algorithm

One natural idea is bit fixing. To determine a route between nodes i and $\pi(i)$, we find the first bit that differs between the two and then "fix" it by routing the message along the edge corresponding to flipping that bit. Repeating this procedure as needed gives an entire route from i to $\pi(i)$, the length of which is clearly minimal by construction.

Once we've established a route for each message, then at each time-step we can advance each message through the next edge on its route. The only issue arises when multiple messages try to use the same edge at any given time-step, in which case we select one arbitrarily to advance and force the others to wait.

2.1 A Bad Case

Unfortunately, there exists a permutation π such that the bit fixing algorithm is slow. In particular, it suffices to find a permutation that causes a single edge to be traversed by many routes, since if an edge is traversed δ times, then we must have $T \geq \delta$.

Consider the permutation consisting of the following send-receive pairs

$$\overbrace{X}^{\frac{n}{2}-1 \text{ bits}}\ 00\ \overbrace{00\cdots0}^{\frac{n}{2}-1 \text{ bits}} \to \overbrace{00\cdots0}^{\frac{n}{2}-1 \text{ bits}}\ 01\ \overbrace{X}^{\frac{n}{2}-1 \text{ bits}}$$

where X has exactly $\frac{n}{4}$ ones. The permutation can be arbitrary on the remaining nodes not of this form. Observe that there are $\binom{\frac{n}{2}-1}{\frac{n}{4}} \approx \frac{2^{n/2}}{\sqrt{n}} = \sqrt{\frac{N}{n}}$ such pairs of this form, all of which cross the edge

$$\overbrace{00\cdots0}^{\frac{n}{2}-1 \text{ bits}}\ 00\ \overbrace{00\cdots0}^{\frac{n}{2}-1 \text{ bits}} \to \overbrace{00\cdots0}^{\frac{n}{2}-1 \text{ bits}}\ 01\ \overbrace{00\cdots0}^{\frac{n}{2}-1 \text{ bits}}$$

Thus in the worst case, T is $\Omega\left(\sqrt{\frac{N}{n}}\right)$, significantly worse than the $\Omega(n)$ lower bound we showed earlier. In fact, it can be shown that a similarly bad lower bound exists for any deterministic, oblivious algorithm.

3 A Randomized Algorithm

One natural next idea would be to use an approach similar to the deterministic bit fixing algorithm, but to randomize the order in which bits are fixed. Unfortunately, this still requires $2^{\Omega(n)}$ time in the worst case, which will be shown on the homework.

An important observation is that while the time taken in the worst case is quite bad, it's possible that the algorithm performs better *on average*. Our new idea is to first send each message i to a uniformly random vertex x_i, and then to send x_i to $\pi(i)$, while still using bit fixing to determine the route $i \to x_i \to \pi(i)$.

3.1 A Loose Analysis

To analyze the behavior of this approach, suppose we sample uniformly random intermediate vertices $x_1, \ldots, x_N$ *with replacement*. Let $L(e)$ denote the number of $i \to x_i$ paths using a particular edge e, where we treat the edge e as directed. By symmetry, we have that $\mathbb{E}[L(e)]$ is the same for all edges e, so

$$\mathbb{E}\left[\sum_{e\in E} L(e)\right] = \sum_{e\in E} \mathbb{E}[L(e)] = (Nn)\,\mathbb{E}[L(e)]$$

We also have that

$$\mathbb{E}\left[\sum_{e\in E} L(e)\right] = \mathbb{E}[\text{total length of all paths}] = N \cdot \mathbb{E}[\text{length of } i \to x_i \text{ path}] = N \cdot \frac{n}{2}$$

Using the two equations above, we see that $(Nn)\,\mathbb{E}[L(e)] = \frac{Nn}{2}$, so $\mathbb{E}[L(e)] = \frac{1}{2}$. Thus the average load on a given edge is small in expectation. Next we'd like to show that this load is also small with high probability.

We define additional indicator variables $H_{i,e}$, where $H_{i,e} = 1$ if the $i \to x_i$ path uses edge e. Observe that $H_{i,e}$ is independent of $H_{j,e'}$ if $i \neq j$. It follows that $L(e) = \sum_{i\in[n]} H_{i,e}$. We have from above that $\mathbb{E}[L(e)] = \frac{1}{2}$, so applying a Chernoff bound we have

$$\mathbb{P}\left[L(e) \geq (1+\epsilon)\frac{1}{2}\right] \leq \exp\left(\frac{\epsilon^2}{2+\epsilon} \cdot \frac{1}{2}\right)$$

Plugging in $\epsilon = 10n$, we get that

$$\mathbb{P}[L(e) \geq 6n] \leq \mathbb{P}\left[L(e) \geq (1+10n)\frac{1}{2}\right] \leq \exp\left(\frac{100n^2}{2+10n} \cdot \frac{1}{2}\right) \approx e^{-5n} < e^{-4n} < N^{-4}$$

so with probability at least $1 - N^{-4}$, $L(e) \leq 6n$.

In fact, there is a slightly nicer version of a Chernoff bound that says if X is the sum of independent $\{0,1\}$ random variables, then $\mathbb{P}[X \geq t] \leq 2^{-t}$ for $t > 2e\,\mathbb{E}[X]$. Applying that bound here gives us that

$$\mathbb{P}[L(e) \geq 5n] \leq 2^{-5n} = N^{-5}$$

Union bounding over all the edges gives us that

$$\mathbb{P}[\max_e L(e) \geq 5n] \leq (Nn) \cdot N^{-5} < N^{-3}$$

so with probability at least $1 - N^{-3}$, the wait time is at most $5n$ for *every* edge. Thus any single path is delayed for at most $5n$ time steps on each of the at most n edges it traverses, giving an upper bound of $5n^2$ time for the routing of every path.

Notice that this approach requires only $O(n^2)$ time, which is an exponential improvement over the deterministic version. However, it still seems as though the analysis is a bit loose. In particular, we wouldn't expect to spend $\Omega(n)$ time waiting at most edges along a given path, since once we wait on a message once and it gets "ahead", we shouldn't have to wait on it again. In the next section we'll see how to formalize this analysis to show the above approach actually does achieve $O(n)$ time.

3.2 Tighter Analysis

We may improve the average case to $O(n)$ by instead considering the number of paths that intersect with a path as opposed to the load on each path. Let us define the set of paths which intersect some path P_i as

$$S_i = \{j \mid P_j \cap P_i \neq \emptyset\}.$$

We briefly touched on the following lemma in class. Refer to the 2019 iteration of this course for a more detailed proof. The aforementioned proof analyzes the relationship between "lags" and messages on a common path.

Lemma 1. *Let T_i be the time for packet i to traverse path P_i. We have $T_i \leq n + |S_i|$.*

We may use the preceding lemma to bound $|S_i|$. Suppose path $P_i = \{e_1, e_2, ..., e_l\}$.

$$\begin{aligned}
\mathbb{E}[|S_i|] &\leq \mathbb{E}[\text{total \# of edges on paths intersecting } P_i] \\
&\leq \mathbb{E}\left[\sum_{i=1}^{l} L(e_i)\right] \\
&= \frac{l}{2} \leq \frac{n}{2}
\end{aligned}$$

Using a form of the Chernoff bound, we may obtain

$$\mathbb{P}[|S_i| \geq t] \leq 2^{-t}, \forall t \geq en.$$
$$\mathbb{P}[|S_i| \geq 4n] \leq \frac{1}{N^4}$$

Applying Lemma 1, we have

$$\mathbb{P}[T_i \geq 5n] \leq \frac{1}{N^4}$$

and after taking a union bound

$$\mathbb{P}[\max T_i \geq 5n] \leq \frac{1}{N^3}$$

Therefore, we have shown routing a message may be done in $O(n)$ with high probability.