Lecture 6: Randomized routing in the hypercube CSE 525, Fall 2016
Instructor: James R. Lee

1 Chernoff bounds: Recap

In the last lecture, we saw the Chernoff bound for sums of independent indicator random variables. Let's recall: Suppose $X = X_1 + \cdots + X_n$, where $\{X_i\}$ are independent $\{0,1\}$ random variables. If $\mu = \mathbb{E}[X]$ and $\beta \geqslant 1$, we derived the bounds

$$\mathbb{P}[X \geqslant \beta\mu] \leqslant \left(\frac{e^{\beta-1}}{\beta^\beta}\right)^\mu \tag{1.1}$$
$$\mathbb{P}[X \leqslant \mu/\beta] \leqslant \left(\frac{e^{1/\beta-1}}{\beta^\beta}\right)^\mu .$$

Let us consider the bound (1.1) for a moment. If $\lambda \geqslant \mu$, then setting $\beta = \lambda/\mu$ in (1.1) yields

$$\mathbb{P}[X \geqslant \lambda] \leqslant \left(\frac{e\lambda}{\mu}\right)^\lambda .$$

In particular, for $\lambda \geqslant 6\mu$, we arrive at

$$\mathbb{P}[X \geqslant \lambda] \leqslant 2^{-\lambda} . \tag{1.2}$$

2 Routing in the hypercube

Consider the directed graph $G = (V, A)$ with $V = \{0,1\}^n$ and $A = \{(u,v) : \|u - v\|_1 = 1\}$. This is the usual n-dimensional hypercube with each undirected edge replaced by two arcs.

Given a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$, we consider a packet routing problem in which each node $v \in \{0,1\}^n$ needs to send a packet to the destination $\pi(v)$. Let us suppose that every directed arc $e = (u,v) \in A$ can process one packet per time step. Additional packets are queued at u waiting for their turn to cross the arc e. The particularly queuing discipline does not matter, but let's suppose it's FIFO for concreteness (with ties broken in some fixed way).

Once we have specified, for every $v \in \{0,1\}^n$, a directed path γ_v in G from v to $\pi(v)$, the protocol commences with every edge processing one packet per time step (if such a packet is waiting at the head of the edge ready to be routed).

2.1 Bit fixing

We would like to specify a simple mechanism for routing packets to their destination. Consider two nodes $a, b \in \{0,1\}^n$ and let γ_{ab} denote the path that starts at a, and then proceeds as follows: For each $i = 1, 2, \ldots, n$, if $a_i \neq b_i$, then flip the ith bit. Let us call this the *bit fixing route.*

It is not too difficult to come up with permutations π such that the bit fixing route takes more than $\Omega\left(\sqrt{2^n/n}\right)$ time steps in order to route all the packets to their destination. In fact, it is known that this lower bound continues to hold for any deterministic protocol where the route from an intermediate node x to a destination y depends only on x and y (and not on the node from which the packet originates). We will now present a randomized routing algorithm that finishes in $O(n)$ steps with probability close to 1.

2.2 The two phase randomized protocol

The routing proceeds in two phases.

- Phase I: Let $\varphi : \{0,1\}^n \to \{0,1\}^n$ denote a uniformly random map (*not* necessarily a permutation). First route each v to $\varphi(v)$ using the bit fixing route.
- Phase II: For each $v \in \{0,1\}^n$, route $\varphi(v)$ to $\pi(v)$ using the bit fixing route.

We will argue that Phase I finishes in $O(n)$ time steps with probability close to 1. The analysis will apply symmetrically to Phase II. We will assume that all packets are routed in Phase I before Phase II begins, but by being more careful, it is possible to do the analysis even when packets immediately enter Phase II asynchronously.

2.3 The analysis

For every $v \in \{0,1\}^n$, let γ_v denote the bit fixing route from v to $\varphi(v)$. jis the sequence of directed arcs encountered along the bit fixing route. Fix a node $u \in \{0,1\}^n$. If u reaches its destination at time t, let us define the *delay of* u to be $t-k$. Note that without any other packets getting in the way, u would reach its destination in k steps, so this represents the number of time steps in which u is waiting in a queue and not moving.

The following subtle lemma is key to the analysis.

Lemma 2.1 (Delay Lemma)**.** *Let* $S = \{v \in \{0,1\}^n : \gamma_u \text{ and } \gamma_v \text{ share at least one edge}\}$. *Then the delay of* u *is at most* $|S|$.

We will assume the correctness of Lemma 2.1 for now and finish the analysis of Phase I. Then we will address its proof.

For $v \in \{0,1\}^n$, define the indicator random variables

$$H_{uv} = \begin{cases} 1 & \gamma_u \text{ and } \gamma_v \text{ share an edge} \\ 0 & \text{otherwise.} \end{cases}$$

Note that

$$\text{delay of } u \leqslant |S| = \sum_v H_{uv}\,.$$

Moreover, observe that the random variables $\{H_{uv} : v \in \{0,1\}^n\}$ are mutually independent since the destinations $\varphi(v)$ are chosen uniformly and independently at random. Thus we are in position to apply a Chernoff bound, as long as we can calculate $\mathbb{E}[\sum_v H_{uv}]$.

Unfortunately, $\mathbb{E}[H_{uv}]$ is a bit complicated. In particular, it depends on $\|u-v\|_1$. Instead, let us argue as follows. For each $e \in A$, let $N(e)$ denote the number of paths $\{\gamma_v : v \in \{0,1\}^n\}$ that use the arc e. By symmetry (formally, by edge-transitivity of the hypercube), it holds that $\mathbb{E}[N(e)] = \mathbb{E}[N(e')]$ for all pairs of arcs $e, e' \in A$. This allows us to calculate $\mathbb{E}[N(e)]$ as follows: For any $e_0 \in A$,

$$|A| \cdot \mathbb{E}[N(e_0)] = \sum_{e \in A} \mathbb{E}[N(e)] = \sum_{v \in \{0,1\}^n} \mathbb{E}[\text{len}(\gamma_v)]\,,$$

where $\text{len}(\gamma_v)$ denotes the number of edges in the path γ_v. It is easy to see that $\mathbb{E}[\text{len}(\gamma_v)] = n/2$, hence we conclude that

$$\mathbb{E}[N(e_0)] = \frac{2^n n/2}{|A|} = \frac{2^n n/2}{2^n n} = \frac{1}{2}\,.$$

Thus for any fixed route $\gamma = (e_1, e_2, \ldots, e_k)$, we have

$$\mathbb{E}[\#\{v : \gamma \text{ and } \gamma_v \text{ share an edge}\}] \leqslant \mathbb{E}\left[\sum_{i=1}^{k} N(e_i)\right] = \frac{k}{2} \leqslant \frac{n}{2}.$$

In particular, this implies that

$$\mathbb{E}\left[\sum_{v} H_{uv}\right] \leqslant \frac{n}{2}.$$

Now setting $\lambda = 6 \cdot \frac{n}{2}$, we can apply (1.2) to conclude that

$$\mathbb{P}[\text{delay at } u \geqslant 3n] \leqslant 2^{-3n}.$$

By a union bound, we have

$$\mathbb{P}\left[\text{delay at } u < 3n \text{ for all } u \in \{0,1\}^n\right] \geqslant 1 - 2^{-3n}2^n = 1 - 2^{-2n}.$$

In particular, we conclude that with probability at least $1 - 2^{-2n}$, every node is routed within $4n$ time steps.

2.4 The Delay Lemma

We are left to prove Lemma 2.1. First, we need the following simple fact which is a consequence of the bit fixing route we use.

Observation 2.2. For any $u, v \in \{0,1\}^n$, if γ_u and γ_v meet and then diverge, they do not meet again.

Proof of Lemma 2.1. Fix $u \in \{0,1\}^n$ and let $\gamma_u = (e_1, e_2, \ldots, e_k)$.

For $t \geqslant 1$, let us define the quantity

$$\mathsf{lag}_t(v) = \begin{cases} t - j & v \text{ is waiting at the head of } e_j \text{ when time step } t \text{ begins} \\ 0 & \text{otherwise.} \end{cases}$$

If $\mathsf{lag}_t(v) = \ell > 0$ but $\mathsf{lag}_{t+1}(v) = 0$, let us say that v *is ejected with lag* ℓ. Note that from Observation 2.2, if v is ejected with lag ℓ at time t, then $\mathsf{lag}_{t'}(u) = 0$ for all $t' \geqslant t + 1$.

Let T be the time step in which u reaches its destination. Note that $\mathsf{lag}_1(u) = 0$ and if u reaches its destination in T time steps, then

$$\mathsf{lag}_T(u) = T - k = \text{delay of } u.$$

Thus we need to show that $\mathsf{lag}_T(u) \leqslant |S|$.

This will be done via a charging argument. We claim that if $\mathsf{lag}_T(u) \geqslant \ell$ for some $\ell \geqslant 1$, then some node $v \in S$ is ejected with lag ℓ. Therefore if $\mathsf{lag}_T(u) = L$, there must be nodes ejected with lag $\ell = 1, 2, \ldots, L$, meaning that $L \leqslant |S|$. (Again, by Observation 2.2, each node is ejected at most once.)

Fix $\ell \geqslant 1$. Let t be the last time step at which $\mathsf{lag}_t(v) = \ell$ for some $v \in \{0,1\}^n$. Such a time step must exist since $\mathsf{lag}_{T+1}(v) = 0$ for all v. Let $j = t - \ell$. Since $\mathsf{lag}_t(v) = \ell$, it must be that v is waiting at the head of e_j when time step t begins. Thus some packet moves across edge e_j in time step $t + 1$.

We may assume it is v. We claim that v is ejected with lag ℓ at time t. Indeed, if v is not ejected, then $\mathsf{lag}_{t+1}(v) = (t+1) - (j+1) = \ell$, contradicting our assumption that t is the last time at which some node has lag ℓ. ∎