8.1 Symmetry Breaking

Consider a routing problem in the hypercube \(\{0, 1\}^n\). The input is a permutation
\[\pi : \{1, 2, ..., n\} \to \{1, 2, ..., n\}.\]

For every \(i\), the goal is to send one packet from \(i \to \pi(i)\).

First we consider a synchronous model. At every time step, we can send one packet along every edge (in each direction). For this, we need some kind of queueing strategy. The packets wait in a FIFO queue at each node. Our goal is to route all the packets to their destinations in the shortest amount of time possible.

We require that our routing strategies be oblivious: The \(i \to \pi(i)\) route cannot depend on \(\{\pi(j)\}_{i \neq j}\).

**Theorem 8.1** Any oblivious deterministic routing strategy on the hypercube requires worst case \(\Omega(\sqrt{N} n) = \Omega(\sqrt{2^n n})\) steps to route some permutation \(\pi\).

**Theorem 8.2 (Brebner-Valiant)** There is an oblivious random strategy that routes every \(i \to \pi(i)\) in \(O(n)\) steps with high probability, (failing with probability approximately \(2^{-n}\)).

We choose a routing strategy as follows:

- Phase 1: Every \(i \in \{0, 1\}^n\) chooses a uniformly random distributed destination \(\delta(i) \in \{0, 1\}^n\) and routes \(i \to \delta(i)\) using “bit fixing”.
- Phase 2: After everyone has reached their \(\delta(i)\), route from \(\delta(i) \to \pi(i)\) again using bit fixing.

By bit fixing, we mean that we send a packet from \(x\) to \(y\) through vertices having a progressively larger identical prefix. That is, in each step, for the next \(i \in \{1, 2, ..., n\}\) such that \(x_i \neq y_i\), we flip bit \(x_i\).

8.1.1 Analysis

It can be seen that the analyses of phases 1 and 2 are symmetric, so we only need to look at phase 1.

Let \(D(i)\) be the delay of the \(i\)th packet (number of time steps that \(i\) spends in a queue). Then we finish in time \(\leq n + \max_i D(i)\).
We will use a Chernoff bound to show that there exists a constant $c$ such that

$$Pr[D(i) > cn] \leq e^{-2n} \forall i \in \{0,1\}^n$$

This implies by a union bound that

$$Pr[\exists i \text{ s.t. } D(i) > cn] \leq e^{-2n} 2^n < 2^{-n}$$

**Definition 8.3** Let $P_i$ be the path from $i$ to $\delta(i) = \{e_1,e_2,...,e_k\}$ and let $S_i = \{j \neq i : P_i \cap P_j \neq \emptyset\}$. (This is the set of all $j$’s whose routes intersects with $i$’s path.)

**Claim 8.4** $D(i) \leq |S_i|$.

**Proof:**

**Lemma 8.5** By the nature of bit fixing, once two paths $P_i$ and $P_j$ separate, they never come back together.

A consequence of this is that we only have to wait for each packet that intersects our path once. The basic idea of the following analysis is that we “charge” some packet for each delay we experience. Each packet (as it turns out) will be charged at most once.

We charge each packet that made us wait at the point when he leaves my path.

**Definition 8.6** For $j \in S_i \cup \{i\}$, define the lag of packet $j$ (relative to $i$) at time $t$ to be $t-l$ if $j$ is waiting at $e_l$, where $e_l$ is the $l$th edge along path $P_i$.

The lag of packet $i$ moves nondecreasingly through $\{0,1,2...D(i)\}$. In particular, each step in which $i$ waits at any edge, its lag increases by 1.

There must be a time $t'$ at which some member of $S_i$ has lag $L$, but after $t'$ no member of $S_i$ has lag $L$ ever again. At time $t'$, some packet waiting at edge $t' - L$ (having lag $L$) will move, and since we know its lag does not remain at $L$, it must either reach its destination or diverge from $P_i$.

When packet $i$’s lag increases from $L$ to $L + 1$, there exists a corresponding time $t'$: the last step in which any packet has lag $L$. We may thus charge this delay (i.e. lag increase) of $i$ to time $t'$, or more correctly, to a packet (which we know must exist) that at time $t'$ both has lag $L$ and finishes or diverge from $P_i$. This ensures that we charge each time $i$ waits, and yet each vertex in $S_i$ gets charged only once, since each leaves $P_i$ or reaches its destination only once. Therefore, $D(i) \leq |S_i|$.

**Definition 8.7** Define

$$H_{ij} = \begin{cases} 1 & \text{if } P_i \cap P_j \neq \emptyset; \\ 0 & \text{otherwise.} \end{cases}$$

For an edge $e$, let

$$C(e) = |\{\text{paths that use } e\}|$$

We note that

$$|S_i| = \sum_{j \neq i} H_{ij} \leq \sum_{l=1}^k C(e_l) \ (k \leq n)$$
So
\[ E[C(e)] = E \left[ \frac{\text{total length of routes}}{\text{number of directed edges}} \right] \leq \frac{N \cdot n}{2N \cdot n} = \frac{1}{2} \]
where the first equality is due to the symmetry of every edge. Thus
\[ \mu = E[|S_i|] \leq \frac{n}{2}. \]

Therefore
\[ Pr[|S_i| > (1 + \delta)n/2] \leq Pr[|S_i| > (1 + \delta)\mu] \]

Applying a Chernoff bound,
\[ Pr[|S_i| > (1 + \delta) \cdot \mu] \leq \left( \frac{e^{\delta}}{(1 + \delta)^{1+\delta}} \right)^{\mu} \leq e^{-2n}, \]
for \( \delta \geq 5 \). Hence
\[ Pr[|S_i| > 3 \cdot n] \leq e^{-2n} \]

By a union bound,
\[ Pr[\exists i \text{ s.t. } D(i) > 3n] \leq e^{-2n} \cdot 2^n < 2^{-n} \]

This implies that the total time for phase 1 is \( \leq 4n \) with probability at least \( 1 - 2^{-n} \). Thus the overall time is \( O(n) \) with very high probability.