Permutation routing in the hypercube

Scribe: Muhan Li

Introduction

Suppose there is a network N, and a list of sending node pairs:

$$[s_1 \rightarrow t_1, s_2 \rightarrow t_2, \ldots, s_n \rightarrow t_n],\ s.t.\ s_1 \ldots s_n \in N, t_1 \ldots t_n \in N$$

And we view the routing process as a continuous sequence of unified time windows, which satisfies:

1. In each time window, an edge (information channel) $e \in N$ can forward 1 message.
2. If there are multiple messages sharing the same edge in their routes, only 1 message will pass through e, while other messages will be queued on senders for 1 unit of time.

Routing in hypercube topology

Suppose there is a d dimension hyper cube, with $n = 2^d$ nodes, each node has a binary index $i \in \{0,1\}^d$.

In such an environment, we usually use the **bit fixing** strategy to find the shortest route from some node n^i to some node n^j (note we use superscripts here to represent index):

> In order to send packet from n^i to n^j, each time send to next node by correcting one bit (usually from left to right) of current encoding (initial value = i) to a bit value in j.
>
> Eg:
>
> let $i = 01000$, let $j = 00101$, the shortest route is $01000 \rightarrow 00000 \rightarrow 00100 \rightarrow 00101$

There are several interesting observations we can make if we use the bit fixing strategy to create all routes:

1. Nodes n^i and n^j are connected if only 1 bit of their encoding is different.
2. If no collision happens, the longest one of all (shortest) paths has size d, because by using the bit fixing strategy there are only d bits to flip to send message from n^i to n^j.

Suppose routing is oblivious (each node chooses a path on its own), there are two types of strategy we can use to create routes for all routing pairs:

1. Deterministic strategy, directly apply the bit fixing strategy without considering collisions, delay $\sim\ O(\sqrt{n}) = 2^{d/2}$
2. Randomized scheduling, also known as the Valiant-Brebner algorithm:

   > For routing $n^i \rightarrow n^j$, first choose a random intermediate destination n_k, then establish paths $n^i \rightarrow n_k$ and $n_k \rightarrow n^j$ using the bit fixing strategy.

   Which performs much better and only has delay $\sim O(d)$.

Deterministic delay proof

(Problem 1 in homework 2, the basic idea is first proving that there are $C\sqrt{n}$ pairs sharing some edge $a \to b$, then conclude that this edge will cause $O(\sqrt{n})$ delay.)

Lemma 1

Let $d = 2k + 1$, let binary sequences of length k $P, Q \in \{0, 1\}^k$, and define two adjacent nodes in the hypercube as $a = (P, 0, Q), b = (P, 1, Q)$, there exists at least 2^k pairs of source and target (s_i, t_i) in the hypercube such that all paths (s_i, t_i) routed according to the bit fixing strategy contain the edge $a \to b$.

Lemma 1 proof

let $P^* = \{0, 1\}^k - P, Q^* = \{0, 1\}^k - S$, then $|P^*| = |Q^*| = 2^k - 1$, we will then define the source set and the target set as:

$$\mathcal{S} = \{s_j | s_j = (p_j, 0, Q), p_j \in_u P^*, j = 0, \ldots, 2^{k/2}\}$$
$$\mathcal{T} = \{t_k | t_k = (P, 1, q_k), q_k \in_u Q^*, k = 0, \ldots, 2^{k/2}\}$$

Where $\in_u$ means uniquely select, since $|P^*| = |Q^*| = 2^k - 1$ and $2^k - 1 > 2^{k/2}$ when $k \geq 1$, we can always select such two sets.

There are $2^{k/2} * 2^{k/2} = 2^k$ unique combinations of source and targets if we select a source node with index from $\mathcal{S}$ and a target node with index from $\mathcal{T}$.

Finally, according to the bit fixing strategy, the routing policy will correct each bit of the source node to the target node from left to right, therefore after correcting the first k bits, reached node index will always be a, and then correcting the $k + 1$ bit will let us reach b. So routes of all 2^k (s_i, t_i) pairs will always contain the edge $a \to b$.

With lemma 1, we can make the following claim:

Theorem

There exists a case where the regular bit fixing strategy has a maximum packet traveling time of $\Omega(\sqrt{n})$, and therefore this strategy is inferior to randomized routing, which has a maximum packet traveling time of $O(d)$.

Theorem proof

With the routing case defined in the proof of lemma \ref{lemma-1.1}, all 2^k messages have to go through edge $a \to b$, and since an edge can pass 1 message in 1 time unit, the total routing time will be at least $2^k = C\sqrt{n}$, where C is a constant factor, in this case, the maximum delay of the bit fixing strategy is equal to the total length of the routing time, $\Omega(\sqrt{n})$.

Randomized delay proof

Theorem

W.h.p. the maximum delay $\max_i delay(s_i, t_i) \leq O(d)$.

Theorem proof

We need to prove $P\{delay(s_i, t_i) \geq cd\} \ll 1 \quad (e.g.: \frac{1}{n})$ for all routes, where c is a constant value.

Select and fix a random route $s_i \to t_i$, denote the route between $s_i \to t_i$ by R_i.

For other routes R_j, R_i and R_j only interfere if they intersect or share at least one edge, and cannot share more than 1 edge (because one packet is going to be delayed by one time unit, and there is no shortcut, because between and point A and B, all paths have the same length, this is determined by the bit difference number between A and B).

Therefore:

$$delay(R_i) \leq \#\ \{R_i, R_j\ interfere\}$$

Let X_j represent event $\mathbb{I}\{R_i \text{ and } R_j \text{ interfere}\}$ for all other routes, we would like to know $\mathbb{E}[\sum X_j | R_i]$ and use the Chernoff bound to prove our target probability $P\{\sum X_i > cd | R_i\}$ is extremely small for some constant c. In order to analyze the total number of interfering R_js, we can enumerate all edges $e \in R_i$ and count the number of R_j which contain e, then sum these numbers up (multiply expectation with d since there are d edges).

Since e could be represented as:

$$(P, 1, S) \to (P, 0, S)$$

when the route $s_j \to t_j$ goes over e, v_j has prefix P, s_j has suffix S, since:

$$\begin{gathered}\#\{s_j \text{ with suffix } S\}: 2^{d-|S|} = 2^{|P|+1} \\ P\{t_j \text{ that has prefix } P\} = 2^{d-|P|}/2^d\$\end{gathered}$$

Therefore:

$$\begin{aligned}&\mathbb{E}[\#j\ s.t.\ e \in R_j | R_i] \leq 2 \\ &\mathbb{E}[\#R_j \text{ interfere}] \leq 2d \Rightarrow \mathbb{E}[\sum X_j] \leq 2d\end{aligned}$$

With above observations, we can prove that:

$$\begin{aligned}&P\{\sum X_j > 10d\} \\ &\text{by chernoff bound:} \\ &\leq e^{-u} * (\frac{eu}{10d})^{10d}, s.t. u = \mathbb{E}[\sum X_j] \leq 2d \\ &\leq e^{-2d}(\frac{2e}{10})^{10d} \\ &\leq (\underbrace{e^{-2} * (\frac{2e}{10})^{10}}_{\ll 1})^d \\ &\ll 1\end{aligned}$$