Hyperelect - Leader election in directed hypercube network

Idea behind the algorithm

The algorithm will work over several stages. During each stage a *candidate* (called a *duellist*) will have a match against another duellist. One of them will win and proceed to the next stage, while the other will become *defeated*. After each stage only half of the duellists will enter the next stage. At the end, only one duellist will be standing. This duellist will become *leader* and notify the others.

Let us take a closer look on how to execute aforementioned idea. We will start by explaning how to pair the duellist, then how to perform a match and how to notify all nodes.

Let H_k be a k-dimensional hypercube and $H_{k:i}$ be a collection of i-dimensional hypercubes obtained from H_k by removing all connections between nodes in dimensions $> i$, e.g. $H_{3:1}$ will be collection of 4 segments (1-dimensional hypercubes). After stage i we want to have exactly one duellist left in each hypercube $H_{k:i}$. So after stage 2 we want to have only two duellist in H_3, one in each hypercube $H_{3:2}$. Let us also observe that hypercubes from collection $H_{k:i}$ will be nicely paired in colection $H_{k:i+1}$. So we start from $H_{k:0}$ where every node is a duellist, pair them up according to collection $H_{k:1}$ and have a match between them to determine the winner. We continue this process up to $H_{k:k}$. At the end, we are left with one duellist that will become the leader.

In order to perform a match, a *match* message has to get to the other duellist. We will do so in two steps. First, send a message to the other hypercube. Second, forward a message to a duellist. A duellist can perform the first step by itself as it has the connection to the other hypercube. In order to forward the message every node that was defeated will remember the shortest path (what dimensions to travel) to its opponent (that won). The message will be forwarded in that fashion until it reaches the duellist.

After electing a leader we need to notify all nodes. In k-dimensional hypercube we will perform that process over k rounds. In a round $i \in [1, k]$ every node that is a leader or a *follower* will notify their neighbor over connection in dimension $k - i + 1$ to become a follower. So after round i exactly one node in each hypercube $H_{k:k-i}$ will be a leader or a follower.

Complexity - number of messages

Let $N = 2^k$ be the number of nodes. Let $d(i)$ be the maximal length of the shortest path from node defeated in stage i to the winner. Clearly, $d(i) = i$.

Sending *match* message in stage i can cost up to

$$l(i) = 1 + \sum_{j=1}^{i-1} d(j) = 1 + \sum_{j=1}^{i-1} j = 1 + \frac{i(i-1)}{2}.$$

In stage i there will be $2 \cdot 2^{k-i} = 2^{k-i+1}$ *match* messages.

So in total the communication will cost us

$$\begin{aligned} M[Hyperelect] &\leq N - 1 + \sum_{i=1}^{k} 2^{k-i+1} \cdot l(i) = N - 1 + \sum_{i=1}^{k} 2^{k-i+1} + \sum_{i=1}^{k} 2^{k-i} i(i-1) \\ &= N - 1 + 6 \cdot 2^k - k^2 - 3k - 6 = 7N - (log\, N)^2 - 3\, log\, N - 7. \end{aligned}$$

Complexity - time

The time complexity can be determined using above definitions

$$T[Hyperelect] \leq k + \sum_{i=1}^{k} l(i) = k + \sum_{i=1}^{k} 1 + \sum_{i=1}^{k} \frac{i(i-1)}{2} = 2k + \frac{(k-1)k(k+1)}{6}$$
$$= O(log^3 N).$$

Useful resources

For more information take a look at:

- N. Santoro, Design and analysis of distributed algorithms, section 3.5 (Election in cube networks)
- P. Flocchini, B. Mans - Optimal elections in labeled hypercubes
- S. Robbins, K. A. Robbins - Choosing a leader on a hypercube