

DISTRIBUTED ALGORITHM

Assignment 02

Group 11

Xugang ZHOU	352032
Fangzhou YANG	352040
Yuwen CHEN	352038

November 6, 2013

1 Echo

The basic idea of improvement of the Echo-Algorithm by sending taboo nodes is to avoid the visit of nodes which are known to be visited by other explorers.

- **a. Ring with n nodes**

In this structure, the number of edge and the number of nodes are both n , so the needed number of messages for a normal Echo-Algorithm is $2n$. For Echo-Algorithm with taboo nodes, it only needs n messages, because the message goes only forwards.

- **b. Binary X-tree of height h**

In this structure, the number of nodes $n = 2^{h+1} - 1$, the number of edges $e = 2n - h - 2 = 2^{h+2} - h - 4$, so the needed number of messages for a normal Echo-Algorithm is $2 * n = 2^{h+2}$.

For Echo-Algorithm with taboo nodes, because of the taboo nodes, the messages will not be delivered between sons of a same parent, therefore the number of needed messages is $2 * (n - (2^h - 1)) = 2^{h+1}$

2 Election

- **Worst-Case**

The worst-case message complexity is

$$nk - \frac{k(k-1)}{2} + n$$

where $nk - \frac{k(k-1)}{2}$ election messages and n notification messages are sent. In our test configuration, we constructed a unidirectional ring with 8 nodes. Nodes 2,3,4 have been selected to initiate the election, which means $k = 3$. The initiators are aligned in descending order and initiate election in ascending order.

After the election algorithm terminated, the sent message number is 29, which match exactly the theoretical message complexity of worst cases.

- **Best-Case**

The best-case message complexity is $k - 1 + 2n$, where $k - 1 + n$ election messages and n notification messages are sent.

In our test configuration, we constructed a unidirectional ring with 8 nodes. Nodes 2,3,4 have been selected to initiate the election, which means $k = 3$. The initiators are aligned in ascending order and initiate election in ascending order.

After the election algorithm terminated, the sent message number is 18, which match exactly the theoretical message complexity of best cases.

- **Average-Case**

The average-case message complexity is $nlnk + n$, including $nlnk$ election messages and n notification messages.

To increase the test accuracy, we have constructed rings with different n and k , and make 16 runs for each $n - k$ pair to get the average value. The results is shown by the following table, where the first value of each cell

is the average message number, and the second value is the theoretical complexity:

	n = 8	n = 128
k = 3	22.89/16	385.18/256
k = 7	28.47/24	452.25/384

From the results we can tell that for one fixed n , the practical average complexity becomes more closer to the theoretical one when the value of k increases. A similar situation can be found when we fixed the value of k and increases n .

3 Election

1. The Chang and Roberts Algorithm won't work properly if the assumption of unique IDs does not hold. Considering a typical scenario in a three-node ring $0 \rightarrow 1_a \rightarrow 1_b \rightarrow 0$, where 1_a and 1_b have the same ID. If 1_a and 1_b start the initiate procedure at the same time, 1_b will receive a message with ID 1 from 1_a , which makes it believe that it's the winner. However, since node 0 will forward the message from 1_b to 1_a according to the algorithm, 1_a will also mark itself as winner. This ends up in two winners that cannot be distinguished by other nodes.
2. In some special cases the algorithm can still produce the right answer. Two possibilities are enumerated below.
 - (a) Nodes with duplicate ID don't participate in the election (i.e. not initiator).
Since the election only happens among initiators, the ID of non-initiator nodes will not be considered. In such kind of scenario the ID of each initiator is still unique, and thus won't affect the election result.
 - (b) The largest initiator ID is unique among all initiators.
Assume that the initiator with largest ID is n_w . Considering two nodes n_1 and n'_1 with the same ID also participate in the election. The conflict described above will arise so that both of them would think they have won the game and set the local winner record M_p to their own ID. However, as they received a message from n_w , they will still update their M_p to n_w 's ID and forward it to next node, which means the algorithm continues running. The fake notifications produced by n_1 and n'_1 will also be overwritten by the notification sent from n_w , so that every node know finally that the winner is n_w .