Solution of Problem Set 1

September 21, 2005

Question 1(3.9):

There are two algorithm which can work for this problem as the following.

(a) **Algorithm:** All the nodes start being active. In each phase $p, p = 0, 1, \ldots$, each active node sends a message with its own UID to its successor. These messages are intended to travel distance 2^p. While a m message is received by a process j, if j's own UID is larger than m's carrying UID, j simply discards m, otherwise, j becomes passive and relays m to its successor. When a node receives its own UID, it declares itself the leader.

Sketch of Proof: It's easy to see that the node with the largest UID will never become passive and receive its own UID after $\lceil \log_2 n \rceil$ phases since no other nodes can discard its messages. At the same time, it is impossible for any other node with smaller UID to receive their own UIDs, because at least the node with largest UID will discard any message from the other nodes. So finally only the node with the largest UID can receive its own UID and declare itself the leader.

Next we analyze the message complexity of the algorithm. In phase p, we divide n nodes into $n/2^p$ disjoint groups of size 2^p, in each of which nodes are consecutive in the ring. Each group $g_i^p, i = 0, 1, \ldots, n/2^p$ in phase p contains two subgroups g_{2i}^{p-1} and g_{2i+1}^{p-1} in phase $p-1$. From our algorithm, in each phase, all the nodes within the same group except the node of the local largest UID either become passive or their messages are discarded without leaving the group. For group g_i^p in phase p, consider the internal messages which never leave the group and outgoing messages respectively. From the above discussion, we know that only one message can travel from subgroup g_{2i}^{p-1} to g_{2i+1}^{p-1} and if excluding this message, all the nodes in these two subgroup send or relay only one message. So the number of internal messages for group g_i^p

is no more than $3 * 2^{p-1}$. For the outgoing messages, it's easy to see that at most two messages can leave group g_i^p, one of which is from subgroup g_{2i}^{p-1} and the other from g_{2i+1}^{p-1}, so the relaying number of these two messages is no more than $2 * 2^p$. There are totally $n/2^p$ groups, so the total messages sent in phase p are $n/2^p(3 * 2^{p-1} + 2 * 2^p) = O(n)$. The algorithm will end within $O(\log n)$ phase, so the message complexity is $O(n \log n)$.

(b) Please refer to **The Peterson Leader-Election Algorithm** on Page 482 in Lynch book.

Question 2(4.3):

In the worst case, the message complexity of the *OptFloodMax* algorithm is $\Omega(n^3)$. There are two classes of digraphs which need $\Omega(n^3)$ messages.

(a) Consider such a bidirectional graph: $n/2$ nodes form a clique, and the other $n/2$ nodes form a line in a strictly increasing order of their UIDs. The UIDs of the nodes in the line are strictly larger than the ones of all the nodes in the clique. And the node with the local smallest UID of the line is connected to one node in the clique. It is easy to see that it will take $n/2$ rounds for the largest UID to propagate to the clique. And in each round, each node in the clique receives a new larger UID propagated down from the line and has to broadcast to its neighbors, which cost $n/2 * n/2$ messages totally. So the total message complexity is $n/2 * n/2 * n/2 = \Omega(n^3)$.

(b) Without loss of generality, assume the UIDs of n nodes are in a strictly increasing order, that is $id_i < id_j$ for $i < j$. There is a directional edge from node i to node j if $i < j$ or $i = j+1$. It is easy to see that it will take n rounds for the largest UID id_n to propagate to the whole graph. For node i, it sends out its own id to its $n-i+1$ neighbors in the first round and receive new larger new id in the first $n-i$ rounds and then send this new id to its neighbors in the following rounds (node 1 has only $n-1$ neighbors), so it sends out $(n-i+1)^2$ messages totally(node 1 sends out $n(n-1)$ messages). So the total message complexity is $n^2+(n-1)^2+\cdots+2^2+1-n = 1/6n(n+1)(2n+1)-n = \Omega(n^3)$

Question 3(4.11):

(a) Assume there is a special node acting as an initiator. It initiates the construction of a BFS tree which costs $O(diam) = O(n)$ time and $O(E)$ messages. After the construction, the initiator, also the root of the tree, invokes a convergecast operation which aggregates the node numbers in any

subtree and then broadcasts the total node number to the whole tree. The broadcast and convergecast cost $O(depthofthetree) = O(n)$ time and $O(n)$ messages. If there is no such a special node, we can apply a leader-election algorithm first to elect a leader and let this leader act as the initiator.

(b) Assume every node has a global unique identifier. Building a BSF tree in a directed graph is more costly than in a undirected graph, since children cannot report to their parents directly. Instead, children have to broadcast their report messages to the entire network adding a n term to the message comlexity in undirected graphs. So the message complexity will be $O(En)$, and the time complexity is still $O(n)$ since the broadcasting and reporting can be executed in parallel. Because every report message is broadcast into the whole network, the root can receive all these messages and is able to rebuild the whole tree construction by itself, and of course the root can know the network size and inform the rest network.