CSc72010

Homework 2

Due Thursday February 24, 2005

Answers

Assume an undirected connected network graph, with processes using the learning bridge algorithm.

Problem 1.
All messages reach their destinations.

The intuition:
Every message travels through the graph via broadcasts from all nodes that see copies of the message, for 0 or more rounds. If at round r, a copy of the message arrives at a node i with mac-address-table$_i$[m.dst] non-null, then there is a path all the way to the destination defined by the mac-address-table entries. If it never sees a non-null mac-address-table entry for the destination, nodes keep broadcasting the message until it gets to the destination.

Proof:
We first prove two invariant assertions. The first states that all processes broadcast the message until some round when a copy arrives at a process with a non-null mac-address-table entry for the destination.

Invariant Assertion 1: If j sends a message to k at the beginning of round 1, then at the end of round r, either some copy of the message has arrived at a process i with distance < r from j and mac-address-table$_i$[k] non-null, or every process at distance r from j has seen a copy of the message.

Proof of Invariant Assertion 1:
By induction on the number of rounds:
Round 1: Either j itself has a non-null mac-address entry for k or it broadcasts the message to all its neighbors. Thus at the end of round 1, all nodes at distance 1 from j have seen a copy of the message.
Induction hypothesis: Assume the invariant is true for round $r<r_0$.
Round r_0: Let i be a process at distance r_0 from j. Then there is a process i' that is a neighbor of i and is distance r_0-1 from j. By the induction hypothesis, either there is a process h with distance < r_0-1 from j and mac-address-table$_h$[k] non-null or process i' has seen a copy of the message. If the first case holds, the invariant is true. If the second case holds, either i' has a non-null mac-address-table entry for k or it broadcasts the message to all its neighbors, including i, and the invariant is true for this case also.
□

The second invariant states that once a copy arrives at a process with the mac-address-table entry for the destination non-null, it will follow mac-address-table entries all the way to the destination.

Invariant Assertion 2: If mac-address-table$_k$[j] $\neq$ null then there is a path (i_o, i_1), (i_1, i_2), …, (i_{n-1}, i_n) from k to j such that $k=i_0$, $j=i_n$, and mac-address-table$_{i_h}$[j]$=i_{h+1}$.

Proof of Invariant Assertion 2:
The proof is by induction on the number of changes that have been made, collectively, to the mac-address-tables in all processes in the network.
If no changes have been made, the invariant assertion holds trivially.
Assume that the invariant assertion holds whenever $C<C_0$ changes have been made.
At change C_0: Suppose that at change C_0, mac-address-table$_k$[j] changes from null to non-null. This means that a message m with m.src = j has just arrived at k. Let's suppose that i is the neighbor that forwarded m to k. Then mac-address-table$_k$[j]=i, and either i=j or at some previous round, mac-address-table$_i$[j] was set to non-null. (If mac-address-table$_i$[j] was null at the beginning of the previous round, it was set to a non-null value after receiving the message with m.src=j.)

In the first case (i=j), we have a path from k to j defined by the mac-address-table at k. In the second case, by the induction hypothesis, there is a path from i to j defined by the mac-address-tables, and so we can add the link from k to i (defined by k's mac-address-table) to the path from i, constructing a path from k to j.
□

The result follows easily. The message is either broadcast by all processes at all rounds or arrives at a process with non-null mac-address-table at some round. In the first case (setting r to the diameter of the network graph in invariant assertion 2) it reaches all other processes in a connected graph. In the second case, after it arrives at a process with a non-null mac-address-table entry for the destination, it follows the path defined by the mac-address-table entries.

Problem 2.
If the network graph is a tree, no message cycles forever.

Proof:
The intuition is that at the end of round r, every copy of the message has traveled on a path of length r. This is clear because the message has been forwarded from one node to another at each round (we could do the induction, but it's not worth the trouble).

This almost establishes that the message can't cycle forever, since all simple paths (paths with no repeated nodes or edges) have finite length in a tree. However, we will also need to show that each message travels only on simple paths in a tree (note that in a graph with cycles, the algorithm does not guarantee that messages travel on simple paths).

In a tree, if there is a repeated edge in a path, then there must be a pair of adjacent edges (j,k), (k,j) in the path. Otherwise, the path would define a cycle in the tree. But the algorithm guarantees that there is no pair of adjacent edges (j,k), (k,j) in any path, because no message is forwarded back along the link from which it arrived.
□