Problem Set 3, Part a

Due: Wednesday, April 5, 2006
Problem sets will be collected in class. Please hand in each problem on a separate page, with your name on it.

Reading

Local infrastructure Chockler *et. al*: Consensus and collision detector
Broadcast Kowalski, Pelc: Deterministic broadcasting paper (skim)
Bar-Yehuda *et. al*: Time complexity of broadcast
Bar-Yehuda *et. al*: Efficient emulation of single-hop radio network

Reading for next week

Broadcast Kowalski, Pelc: Deterministic broadcasting paper. Read the algorithm for radius-2 networks (but not the one for $o(\log \log n)$ networks. Try to understand the main ideas of the lower bound.
Kushelevits, Mansour: Lower bound for broadcast in radio networks
Point-to-point routing Johnson, Maltz: Dynamic Source Routing (DSR)
Perkins, Royer: Ad hoc on-demend distance-vector routing (AODV)
Hu, Johnson: Caching strategies for on-demand routing protocols
Chen, Murphy: Enabling disconnected transitive communication

Problems

1. Consider Algorithm 1 in the Chockler at al. paper.
   (a) Expand on the correctness proof sketch given in the paper, filling in more steps. Be sure to take into account considerations involving failures and active/passive advice.
   (b) Suppose Algorithm 1 is run with a 0-complete (instead of complete or majority-complete), eventually accurate collision detector. Describe an execution that causes two nodes to decide on two different values.
2. Again suppose we are in the setting of the Chockler et al. paper, with a 0-complete eventually accurate collision detector. The leader election problem requires exactly one node to output "leader", and every other node to output "not leader".
   (a) Is this problem solvable with a 0-complete collision detector? If yes, describe an algorithm, if no, provide a brief discussion of why not.
   (b) Does your answer change if the nodes are assumed to have unique identifiers?
3. The adversary in the Hitting Game construction, in Section 3.3 of the first Bar-Yehuda et al. paper, constructs a set S that can fool a given sequence $M_1, \ldots, M_t$ of queries, where $t = n/2$, thereby preventing the sequence from causing the Explorer to "win" the game.
   (a) Can you construct a longer sequence of queries, for t slightly larger than $n/2$, such that the given adversary does not prevent a win? (If you can't do this for all n, try it for some specific value of n.)

(b) Can you construct a longer sequence of queries, again for t slightly larger than $n/2$, such that *no* adversary can prevent a win—that is, a short winning strategy for the Hitting Game?

4. Consider the collision detection algorithm presented in Section 2.3 of the Bar-Yehuda et. al paper on emulating a single hop network. Sketch a proof that it achieves the stated success probability within the stated time bound.