Lab2: Broadcast and Convergecast Communications

Due: Janurary 22th, 2009

The goal of this lab is to design a broadcast flood algorithm that produces a self-stabilizing tree, and then to use this tree to count the number of robots in the network.

1 Setup

Before you start, update your copy of the SwarmBotAPI SVN repository, unzip the lab 2 source, and download the new SwarmOS to the robot. Review the documentation's main file, index.html, located in the lab2\doc folder. There are a lot of new functions for querying neighbors and sharing data.

1.1 Neighbor Lists, Neighbor Ops, and Neighbor Variables

The robot's main API for interacting with neighboring robots is through *neighbor lists*. There is a large API for creating and modifying these lists. I will give a mini-lecture on this API during lab.

2 Broadcast Flood and Tree Construction

Design a broadcast flood algorithm. The algorithm should produce a spanning tree as it progresses. We want this algorithm to be self-stabilizing. In particular, the tree must be robust to the source robot moving to a different location, in which case the tree should rebuild. If the source robot is removed from the network entirely, the tree should maintain its most recent structure, then vanish after a short time, *i.e.* the depth of other robots should not increase to infinity if the source is removed.

Use the lights to indicate if a robot has received the broadcast flood message, and its depth in the tree.

3 Convergecast

Design a convergecast algorithm to count the total number of robots in your mini-swarm. Test your algorithm with a network with a depth of at least 4. Print the output to the console so you can see how well it is working.

4 Check-off and Write-up

4.1 Check-off (Complete as a group)

1. Demonstrate broadcast flood. Use the lights to indicate when robots have received a message from the source.
2. Demonstrate tree construction. Use the lights to indicate the number of hops from the source.

3. Demonstrate tree self-stabilization 1: Move the source. Use the lights to show the tree has been rebuilt.
4. Demonstrate tree self-stabilization 2: Remove the source. The other robots should maintain their previous depth, *i.e.* their depth should not increase without bounds. The robots should clear the tree after a short delay.
5. Demonstrate convergecast counting. Use the robot console to display the results of your counting algorithm.

4.2 Write-up (Complete individually)

Make a network with a diameter of at least four. Plot the number of robots counted by your convergecast sum vs. time over 30 seconds on a static network. Then plot the number of robots counted by your convergecast sum vs. time over 30 seconds in a dynamic network. Make the network dynamic by moving the source robot to a new location every 5 seconds. Make a **one page** document with a plot of the two data set and a brief explanation of the sources of error. Expound on the utility of minimal spanning trees and convergecast in dynamic network topologies.