Computing Small Dominating Sets in Ad Hoc Networks

Project for ECE 3656, Spring 2002
Completion is expected by May 29th 2002

1 Problem description

The input to the problem is an ad hoc network with vertex set V in which nodes can be either *white* or *red*. Each white node u has an associated *covering requirement* r_u. A *Politburo* is a set of red nodes such that every white node u is neighboring to at least r_u red nodes. The aim is to find a Politburo which is as small as possible. *It is assumed that a feasible solution always exists.*

2 Algorithm description

Before describing the algorithm we introduce some notation.

The algorithm consists of a *sequence* of *basic steps*, described below, each taking 2 rounds of communication with the immediate (i.e., one-hop) neighbors. The Politburo is computed incrementally, starting from the empty set and (probably) adding new nodes at the end of every basic step. The set so computed will be denoted as P. When white a node u already has r_u red neighbors in P it is called a *dominated* node, and non-dominated otherwise. The *degree* of a red node X, denoted as $deg(X)$, is the number of non-dominated, white neighbors it has. In the sequel, we shall use lowercase letters to refer to white nodes and uppercase letters to refer to red nodes. The notation

$$u \in X$$

means that u is a white, non-dominated node and that it is a neighbor of the red node X. Finally, given a node u, define

$$value(u) := \min_{X \ni u} \frac{1}{deg(X)}.$$

We now come to the description of the algorithm, which consists in repeating a simple *basic step*, until all nodes are dominated. Although this is a global condition, rather than actually checking it, it is enough that nodes simply drop out of the algorithm according to the following rules: A white node quits the execution of the algorithm when it is covered; a red node quits the execution of the algorithm as soon as either all its white neighbors are covered, or it enters the Politburo. During each basic step a random total ordering of a certain set A is computed. Again, strictly speaking, this needs global synchronization but the problem can be easily circumvented if nodes select a number uniformly at random from a large enough interval. In practice, the interval $1, \ldots, |A|^2$ will suffice. In this fashion the probability that two nodes are assigned the same place in the ordering will be exceedingly small. In case of an unlikely tie, it can be broken arbitrarily.
The basic step is made up of the five substeps listed below.

1. Each white node u computes $value(u)$.

2. Each red node X is a *candidate* if

$$\sum_{u \in X} value(u) \geq \frac{1}{2}. \quad (1)$$

3. Let $\mathcal{T}$ be a randomly generated total ordering of the candidates, and let r_u be the covering requirement of node u. Each white node votes for the first r_u red neighbors in the random total ordering $\mathcal{T}$.

4. Each red candidate X checks if at least a 1/128-Th fraction of its non-dominated, white neighbors have voted for it. That is, X checks if

$$\sum_{u \in X, u \text{ voted for } X} value(u) \geq \frac{1}{128}.$$

   If yes, X enters the Politburo P.

5. Each white node updates the value r_u in the obvious fashion, i.e., by decrementing it by 1 for each red neighbor that has entered the Politburo.

3 Assignment

You are supposed to write a program that simulates the construction of the Politburo.

This means that you will have to place n nodes in the plane (or in 3D space, if you prefer: This would give you extra points) randomly and uniformly. As customary in ad hoc networks, two nodes are neighbors if their (Euclidean) distance is less than the smaller of their transmission radii. The resulting graph is called a *topology*. You should only consider *connected* topologies (this will affect the size of the geographic area you are disseminating your nodes on).

The sides on the geographic area and the transmission radius of a node should be expressed in meters. Also, consider to choose the radius of a node between the following two: 250m (IEEE 802.11) and 10m (Bluetooth). You can consider only one of these, but if you consider two is better.

Some of the n nodes are white and some are red. Given n, the percentage of white and red nodes has to be computed (by you) so that a feasible solution to the problem always exists. This means that every white node u must have at least r_u red neighbors. (Those who can give a mathematical solution to the problem of determining this percentage need to do no simulation.)

You are supposed to generate a number of topologies large enough to guarantee the following statistical accuracy of the result: 95% confidence interval and 5% precision.

Here are the input to your algorithm (that can of course be centralized, i.e., implemented with no message exchange):

- n, the number of nodes. You have to simulate the algorithm for n at least = 100, 200, 300, 400 and 500. Getting up to 1000 will give you extra points. The more, the better.

- r_u, the coverage requirement of each white node. For each white node u consider r_u the same and equal to 1, 2 and 3. (This of course should be done for each different value of n). For yet some more extra point, r_u could be randomly and uniformly chosen between 1, 2 and 3.

For each experiment, the results sought are:

- The average number of red nodes, and the average number of white nodes.
- The average number of red nodes neighbors of a white node (called the *red degree* of a white node).
- The average number of (red) nodes in P and its percentage with respect to the number of red nodes.
- The average number of steps needed to construct P.

You should provide a brief paper with graphs showing the obtained results.
Further readings can be found in [1], [2] and [3].

References

[1] S. Rajagopalan and V. V. Vazirani. Primal-dual RNC approximation algorithms for set cover and covering integer programs. *SIAM Journal on Computing*, 28(2):525–540, 1998.

[2] L. Gonick and W. Smith. *The Cartoon Guide to Statistics.* HarperPerennial, 1993.

[3] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for experimental Design, Measurement, Simulation, and Modeling.* John Wiley and Sons, New York, NY, 1991.