To: Dr. Luiz DaSilva and Dr. Allen MacKenzie
From: Richie Clark, Branden Izumi, Kafu Chau
Subject: MANIAC Abstract
Date: November 21st, 2007

Introduction

We implemented several solutions to maximize the amount of packets we receive that were meant for our node, while consuming the least amount of energy. We want to participate in the network, but we lose points for every packet we forward. Due to the limited feedback from the network we cannot rely on any one node. It is difficult to assume whether a node is dead, or whether it is actively participating in the network. Therefore we want to implement strategies with the least reliability on other nodes.

Description

Our first method is used to tell whether a node may be uncooperative. By saving a list of sent packets and the nodes we forwarded them to, it is possible to cross check any received packet with that list. If a packet has already been received, it is likely the node the packet was sent to is being uncooperative. Although this solution does not help determine if a node is dead, it will allow us to figure out if a node is forwarding packets in an unproductive manner.

However, it is possible that after some time, a previously uncooperative node might become cooperative, either due to parameter changes or node movement. To account for this, after a specified amount of time, the probability that a node is uncooperative is cut in half, thus creating more opportunity for improvement.

A team received points based upon the number of packets received by their team nodes. Therefore, in an effort to accumulate the most points, our nodes will always forward "friendly" packets. In addition, to help ensure that "friendly" packets reach the other team node, we want to forward those packets to the one-hop neighbor that has shown to be the least uncooperative (we say "least uncooperative" since we can't determine that the node is definitely being cooperative). The most reliable forward is determined by the probability computed in the above method.

Another method is to only forward packets whose destination is a node that is at least two-hops away. This is a preferred method to a simple "probability" approach because it ensures that while "our" node is participating, the packet will have to at least be sent by one more person before reaching its destination. This means that the competition will be sending more packets, while we send less since all the one-hop neighbors are being ignored.

A third method is to keep track of how many packets are sent out for each possible destination, and if needed, limiting whether or not a destination is forwarded to. While it is unknown who else is forwarding to that destination, it is useful to make sure that one destination is not taking up a majority of your forwarding time.

A team receives points based upon the number of packets received that are meant for that node. Therefore we chose to forward “friendly” packets in an attempt to accumulate the most points. We send this packet to the most reliable forwarder, established in method two.