Creating An Object-Oriented Wireless Network Simulator

Skyler Nesheim Luong Hoang
Math & Computer Science Department
Drake University

Abstract

Wireless routing protocols are currently among the most actively researched areas in computer science. There are hundreds of such protocols already established and even more under development. Wireless network simulators however have not received as much attention. Currently, there are only a handful of these simulators which are powerful tools for analyzing routing protocols. But, altering their code can be tremendously cumbersome due to the great level of detail they were developed. Some of these systems were designed using the object-oriented programming languages but failed to take into account the various strengths of using such languages, such as inheritance and polymorphism furthermore making them less extensible.

As we have researched wireless routing at Drake University, we have found the need to develop our own wireless simulator as a test environment for our protocols. The goal of this simulator was to improve the ability to add new and sometimes very different wireless protocols to the system and remove the overhead and complexity in the layers of wireless communication. We felt to develop such protocols and test features of interest to us we should not be required to have an in-depth knowledge of the networking layers.

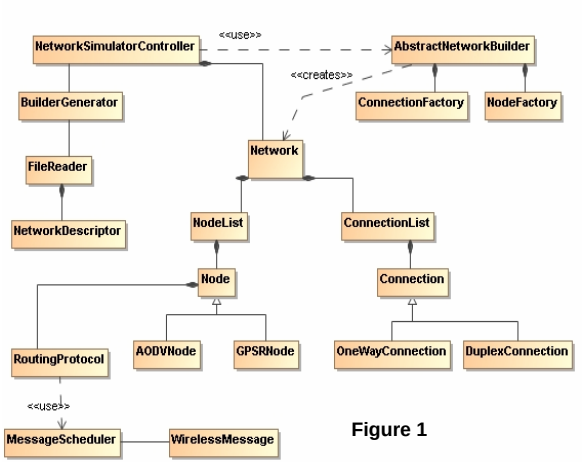

Figure 1

Screenshot of a Sample Network Activity

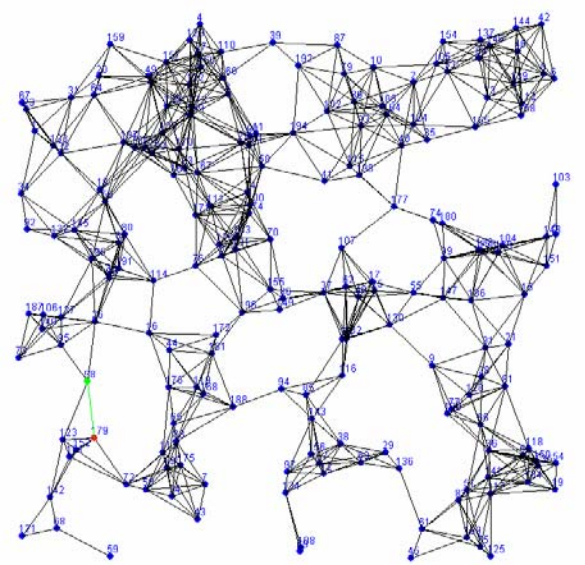

Figure 2

- Figure 2 is a sample of a GPSR network created randomly by our software.
- The nodes represent computers in a real network and are colored blue.
- Each edge connects two nodes and represents an available connection between two computers.
- In this picture, an edge colored green indicates a message is being sent from the red node to the green node.

- Figure 1 is a UML class diagram which represents a small portion of our software system.
- Before actually programming our project we attempted to carefully consider all possible aspects of our system.
- We chose to use object oriented design patterns which enhanced the maintainability and extendibility of our network simulator.

GPSR Protocol

GPSR allows network nodes to use their positions to make message forwarding decisions. A message is forwarded from one node to its neighbor closest to the destination. If there is no such neighbor, the right-hand rule is applied. Under this rule, when a packet encounters a dead-end, it will be forwarded to the node which is first in counter-clockwise order. This rule however, fails when a graph has crossing links. Therefore, we introduce an RNG graph algorithm to remove any overlapping edges in a graph. To do so, we use the following rule: an edge exists between two nodes only if the distance between them is less than or equal to the maximal distance between each of the two and every other node in the network. (See Figure 3.)

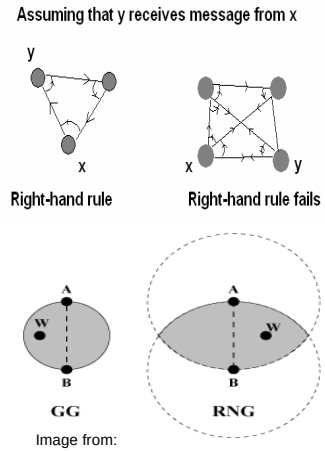

Image from:
http://www.usenix.org/events/nsdi05/tech/full_papers/kim/kim_html/index.html

Figure 3

AODV Protocol

Unlike in a GPSR network, AODV nodes do not know their geographic location. The only information an AODV node maintains about the network is a list of its one hop neighbors and a list of discovered paths in a routing table. In a true AODV system this list is preserved by periodically sending hello messages to every node in broadcast range. A node sends out route request (RREQ) to find routes to other nodes; route reply (RREP) to confirm a route; and route error (RERR) to report a broken link. (See Figure 4.)

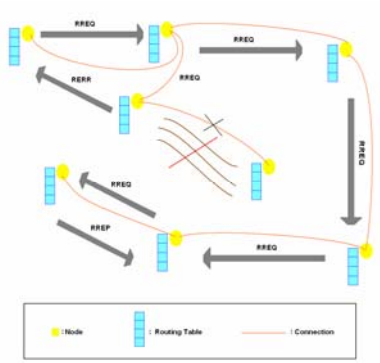

Figure 4

AODV/KSPR Protocol

We are currently researching the AODV / k-SPR routing protocol. A k-SPR set is a special type of k-dominating set of nodes, meaning that every node is within k hops of one of these special nodes. These special nodes are called “routers,” and they are used to help control network traffic. To date, Drake students have implemented three such router selection protocols: k-SPR-I, k-SPR-C, and k-SPR-G. The latter is based on the Greedy Algorithm, and results in a comparatively small set of router nodes. (See Figure 5.)

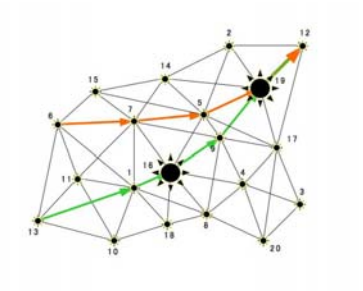

Figure 5