

Using Swarm Intelligence to Broadcast Messages in Highly Mobile Ad Hoc Networks

Michael Kirkpatrick
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556
(260) 486-5518
mkirkpatrick@acm.org

Gregory Madey
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556
(574) 631-8752
gmadey@cse.nd.edu

ABSTRACT

This paper explores the potential for the efficient broadcasting of messages in highly mobile multi-hop ad hoc networks by treating the nodes in the network as autonomous agents for which only a simple set of rules governing the passing of messages from one node to another is defined.

The general problem presented by highly mobile systems is one of coordinating communication between nodes in settings in which the details of their connections to each other – the portions of the network that are within radio range at any given time – can change. Nature provides us with examples – such as birds flying in formation – of mobile, independently operating agents that seemingly work together to perform tasks in a highly efficient manner without complex communication networks and without global knowledge of the locations of individuals. This paper applies this paradigm to the problem of message broadcast in ad hoc networks formed by highly mobile agents. A "biologically inspired" heuristic for broadcasting is introduced. This heuristic has low computational overhead making it especially suitable for "power-limited" systems.

General Terms

Performance, Design, Experimentation

Keywords

Ad Hoc, Mobile, Network, Swarm

1. INTRODUCTION

With the proliferation of portable computing and communication devices such as laptop and hand-held computers and cellular telephones has come an increasing need for communication networks that can support devices that move around as

they are working. This has been addressed in part by radio based local networks within buildings and by stationary antennas in telephone cells. Each of these types of methods, however, relies on a portion of the network being fixed in place. For example, the "on air" network hub connecting laptops moving within an office is installed in a fixed location, both physically and in terms of its logical connection to the larger network.

In such a setting a computer wishing to communicate with another must pass a message to the hub which may then send it directly to the intended destination or may first route it through other parts of the stationary network. This will be done even if the sending and receiving computers are physically within inches of each other. Work has been done to develop communication protocols that exploit the fact that computers in radio range of a central hub may also be in radio range of each other and to allow direct communication to occur in such cases [1,2].

In some situations, such as when all of the nodes of the network are mobile, it may even be desirable to eliminate the reliance on fixed network elements entirely. Examples of such networks include robots exploring the surface of a distant planet, soldiers moving on a battlefield, robots searching for avalanche victims [3], or the elimination of fixed towers in cellular telephone systems.

The general problem presented by these mobile systems is one of coordinating communication between them in settings in which the details of their connections to each other – the portions of the network that are within radio range at any given time – can change. Inouye, et al. [2], describe algorithms that test for connectedness and route messages accordingly, adapting to

changes in the configuration of the system. A common feature of such algorithms is an attempt to establish and maintain some type of global knowledge about the current connections either within the system as a whole or on a more localized sub-system, called an *ad hoc network*.

Nature provides us with examples of mobile, independently operating agents that seemingly work together to perform tasks in a highly efficient manner without complex communication networks and without global knowledge of the locations of individuals. One such example can be found in a flock of birds flying in formation. No individual bird is aware of the positions of all of the other birds. No specific bird directs the movement of the flock. Instead, each bird takes its cue to turn in one direction or another from those immediately surrounding it. Yet, when viewed as a whole the flock appears to be moving in an intelligent, directed manner [4]. This method of groups performing tasks effectively by using only a small set of rules for individual behavior – called *swarm intelligence* – has shown promise when applied to routing problems such as those encountered in transmitting messages through telephone and computer networks [5].

This paper considers the possibilities for applying similar techniques to the problem of communicating in ad hoc mobile networks. Specifically, we are interested in exploring the potential for providing reasonably efficient and reliable communication between nodes in an entirely mobile network – one without any necessarily stationary parts – by defining only a simple set of rules governing the passing of messages from one node to another. In such a network, messages may be passed between two nodes within radio range of each other. A message intended for a third node not in range of either of them may be passed from the first node to the second and held by the second until it either encounters the intended recipient or is able to pass it on to yet another node to carry.

As a starting point for studying this problem, a simulation was developed which models a number of agents moving independently and at random within a defined space. When two agents encounter (come within range) of each other, if one of them has a message and the other does not, the message is passed between them so that both of them have it. So two simple rules govern the entire operation of the agents:

1. Move according to a (possibly random) schedule independent of other agents;
2. When encountering another agent, pass a message to or receive a message from that agent, as appropriate.

A third rule, which determines whether an agent retains a message indefinitely, "times it out," replaces it with a newer message, or drops it upon an acknowledgement of receipt, can also be defined to control the retention of a message by an agent.

2. LITERATURE REVIEW

Networks in a mobile environment are characterized primarily by the fact that the connections between nodes can change. A common view of such a network is of nodes which are connected to each other and/or to some stationary network by a wireless medium such as radio [6, 7], infrared, and so on. At any given time in such a system, a given node will be within range of some subset of the nodes in the network. The number of nodes in this subset can range from zero (in which case the node is isolated) to $n - 1$, where n is the total number of nodes in the system. Viewing this connectivity from the system's perspective, rather than from that of a node, gives a picture of a partially connected graph made up of connected subgraphs, or groups [7]. These subgraphs, referred to in the literature as ad hoc networks, allow communication to take place for a while but do not guarantee the existence of a link at arbitrarily selected times.

Most communication in these networks tends to be either broadcast (flooding) or point-to-point [6]. We are interested, then, in efficiently transmitting messages from one node to another and work has been done exploring ways of determining the connections available at any given time. Perkins and Bhagwat [8] describe a method in which each node maintains a routing table of nodes it can reach and periodically broadcasts this table (or otherwise makes it available) to its neighbors. Johansson, et al. [9], compare the efficiency of maintaining such tables proactively, as done by Perkins and Bhagwat, to building them as needed and, in so doing, saving the overhead of maintaining tables with links to nodes which are no longer in communication. They also use the past history of routes in the ad hoc network to make

predictions of the stability of paths and attempt to use more stable paths. Chandy and Lamport [10] describe a method of taking “snapshots” of a distributed system showing its state at a given point in time and Murphy, et al. [11], adapt this technique for use in delivering messages in an ad hoc network.

A good summary of issues relating to data communications in a mobile setting is given by Perkins [12]. Of particular interest, he says that the overall bandwidth can be improved by increasing the number of base stations in a system in which stationary base stations are used to facilitate communication (as, for example, in a cellular telephone network). Perkins [12] also states that the retransmission of messages can be done more efficiently at the data link layer than at higher levels because the time required is shorter. This can be of great importance in considering what activities can reasonably be expected to be accomplished while a connection exists.

Adler and Scheideler [6] consider power-controlled ad-hoc networks. In such a system, the ability of a given node to connect with another is not entirely a function of the chance that the nodes are positioned in such a way that they can communicate (either directly or through other nodes in the ad hoc network). Instead, a node may be able to boost its signal, allowing it to communicate farther. Conversely, reducing the strength of its transmission may permit it to “whisper” to a specific node with a reduced likelihood of being “overheard” by other nodes. They say that the problem of routing messages in an ad-hoc setting is, in general, NP-hard. However, if one relaxes the quality of service requirements and considers it acceptable if a message can be delivered with probability p , where p is very close, but not equal to, 1 then the problem becomes manageable. Similarly, the “Swarm Intelligence” approach to broadcasting messages in highly mobile environments described in this paper also attempts to deliver messages with high probability, but without necessarily guaranteeing delivery.

Work has also been done studying the dynamics of passing messages between vehicles traveling on highways [13, 14, 15]. Applications include delivering messages between occupants of vehicles in different cars, in a manner resembling cellular telephone service – or, depending on speed, perhaps more like e-mail – but without

stationary cell towers. This is also being investigated for use in allowing the vehicles themselves to communicate safety related information. For example, a braking car could notify the cars behind it that it was slowing, giving more reaction time than that provided by the driver simply seeing the brake lights of the car immediately ahead.

One of the more significant problems common to message delivery in mobile settings, in general, and in highly mobile settings, in particular, is the overhead of maintaining routing information which, in addition to storage and bandwidth considerations, consumes power which is likely to be of limited supply in a mobile system. So any effort to reduce overhead may pay off in terms of power available to the robot, say, to conduct its assigned tasks. The nature of highly mobile environments is such that this information is often not available “naturally.” Das, et al. [16] offer an algorithm for multicasting in a highly mobile network which reduces overhead and increases multicast efficiency by assigning specific forwarding tasks to certain nodes and Chen and Liestman [17] explore an approach which uses weakly connected dominating sets to cluster nodes, simplifying the ad hoc networks.

Much of this work has been concerned with point-to-point or multicast scenarios and has involved determination of the ad hoc networks existing at any given time. When a message is to be broadcast we are interested in delivering it to all other agents efficiently in a timely manner. Research has been done, as well, into this particular problem. Williams and Camp [18] provide a good summary of twelve broadcast protocols with comparisons between them. And Peng and Lu [19] describe ways of reducing redundancy of message transmissions in mobile ad hoc networks. For example, they suggest delaying the relaying of messages until an agent has had a chance to move out of the area in which it received the message. Hass, et al. [20] suggest a probability-based approach to determining whether or not to relay a message when given an opportunity to do so.

Finally, Obraczka, et al. [21, 22] discuss a special case in which the agents are very highly mobile, or “fast moving.” This last scenario – and the broadcasting of messages through a process of “flooding” – is most closely related to the work we report in this paper.

3. SIMULATION ENVIRONMENT

For this research, mobile network nodes were modeled as agents in a Swarm [23, 24] simulation. Swarm is an open source software tool developed at the Santa Fe Institute that provides support for agent based model simulations written in Objective C or Java. Agents operate according to relatively simple sets of rules and do so with little or no global knowledge of the system. The rules defining behavior instruct each agent how and when to move and how to interact with other agents. Global knowledge, if any, is limited to static information such as the total number of agents in the system, the dimensions of the environment, and so on. Dynamically, a given agent is aware of only its own state and its immediate surroundings.

Of primary importance to the concept of agent based modeling is the idea that the agents are autonomous. According to Chantemargue, et al. [25], autonomy is believed to guarantee and to be a necessary condition for adaptability and self-organization. We are interested here in defining an environment that enables us to observe the emergence of self-organization to facilitate message routing in lieu of a central controller.

Agents can be modeled to have a variety of attributes. Several are listed by Thangiah, et al [26]. Those that are relevant to the work described in this paper are

- Can communicate with other agents;
- Can sense and react to changes in the environment;
- Are capable of long periods of unattended operation;
- There is no central authority governing an agent's behavior.

Key among these for our purposes is the last point: we are especially interested in the agents' communication capabilities without reliance on central or regional control.

For this work the environment is simulated as a rectangular (usually square) grid of cells. A cell is either empty or occupied by an agent and, overall, the grid is populated by placing agents in some given percentage of the total number of grid locations. This percentage is called the *density* of agents in the grid. The dimensions of the grid and the density of agents are selected at

the beginning of each run. An agent is selected to be the originator of the message. In some runs a destination agent is selected as well. In others the message, is sent to all agents. As the simulation runs, individual agents move randomly, in any of eight possible directions (corresponding to compass points N, NE, E, etc.). Only one agent can occupy a given grid location at a time so if moving in the selected direction would result in a collision, the agent simply sits still for one time step and attempts to move again (in a randomly selected direction) at the next step. If an agent reaches an edge of the grid its choice of moves is limited to those grid locations surrounding it in the grid. In other words, it does not wrap around to the other side. This method of dealing with the edges of the models was chosen because applications considered for this work are to problems such as robots mapping a region around a landing site on another planet or searching for a lost person in a remote area. In such situations the area of interest is less than the entire planet, so it is not reasonable to consider that moving far enough east, say, would put the agent in the west.

The simulations run can be categorized into four types. These types are described by the following cases:

- Case 1: Messages are passed only to neighbors (that is, agents in any of the eight locations surrounding the agent carrying the message) at each time step;
- Case 2: Messages are passed throughout the ad hoc networks at each step;
- Case 3: Some agents move in a fixed direction (reversing at grid edges);
- Case 4: Agents purge the message from memory after some period of time and, once having carried it, do not accept it a second time.

Except for the fixed direction agents in Case 3, the agents move randomly¹ at each step and pass messages to other agents with which they have contact – either direct or networked, depending on the case. (In the simulation “direct contact” means physically touching on a side or diagonal. In actual applications this could simply mean that they are within, say, radio range of each other.)

¹ The random number generator built into the Java compiler is used for this purpose.

Initial parameters can be set to provide variations in these simulations. For example, the frequency with which an agent will accept a message can be adjusted. That is, instead of accepting delivery of a message every time an agent encounters another agent wishing to pass it a message, it may only accept one every other time or every fourth time, or so on. Also the length of time that an agent will retain a message can be adjusted from one time step to infinity.

The simulator described here was written in Java and uses the Java Swarm libraries. Simulations were run using the Cygwin UNIX emulator from Red Hat, Inc. running under Microsoft Windows 2000 Professional. The results of these runs are described in Section 4, "Simulations" and are discussed further in Section 5, "Analysis of Results." The simulator provides both, output data describing the run, for example the number of steps required to pass a message to all agents in the simulation and a graphical display of the movements and interactions of the agents. This feature can be extremely helpful in showing trends that would not otherwise be obvious and which could be discovered only by extensive analysis of the numerical output. The simulator also displays a run-time graph of the saturation (that is, the percentage of agents who have received the message). These displays are shown in Figures 1 and 2.



Figure 1: Animation of agents' motion and message propagation; gray cells represent mobile agents which have received message; white cells represent agents which have not.

Figure 1 shows an animation representing the movement of agents and the passing of a

message between them. Figure 2 shows a graph of the message saturation throughout the run.

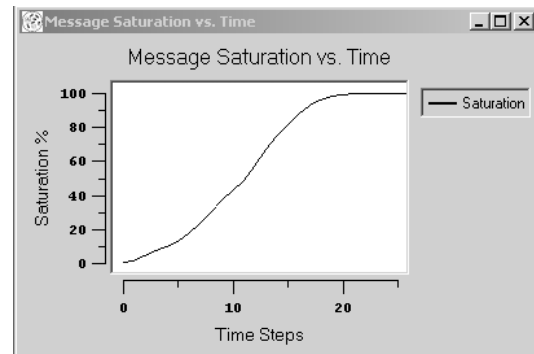


Figure 2: Run-time graph showing message saturation at end of run.

4. SIMULATIONS

This work focuses primarily on the problem of broadcasting messages in a highly mobile ad hoc network. As noted in Section 2, above, Obraczka, et al. [21,22] make a further distinction of networks in which the agents (nodes) are "fast-moving." The simulation results reported here are expressed in terms of "time steps" which represent the time required for an agent to move from one location to another. It is assumed that communication can take place between two agents within range of each other in less than a single time step. This would obviously be the case if messages were sent by radio. However, the use of time steps in this way enables the study of the message passing independent of the speed of the agents in the system. We assume, without loss of generality, that all agents have the same maximum speed, since slower maximum speeds could be modeled in individual agents by skipping movement for a number of time steps proportional to their slower speeds.

Several simulations, as described in section 3, above, were run and those results are given here. An analysis is provided in the next section. These runs were made using a 50 x 50 grid for a total of 2500 locations and these locations were populated at random to a given density. Several runs were made for each density setting, using different starting points. In each run the number of time steps required for the message to reach all of the agents was recorded. Also recorded

were the number of steps needed for the message to reach 90% of the agents and the average number of steps for the message to reach another agent. This last measurement was made by counting the number of steps required for the message to be passed from the selected origin to every other agent. The average of these step counts was then calculated.

Using several randomly selected starting points provides data for a large number of scenarios involving passing a message from one agent to another. The results of the runs for each density were then averaged (mean) over ten runs and are summarized in the tables that follow, starting with Case 1 data shown in Table 1 and graphed in Figure 3.

When graphed as in Figure 3, it is apparent that the number of time steps required to pass the message drops dramatically as the density is increased from, say, 10% to 20% or 30% but changes much more slowly after that.

Table 1: (Case 1) Message passed only to neighbors at each time step

| Density | All | Avg | 90% |
|---------|-----|-----|-----|
| 10% | 109 | 51 | 77 |
| 20% | 55 | 23 | 37 |
| 30% | 29 | 12 | 22 |
| 40% | 23 | 9 | 16 |
| 50% | 17 | 7 | 12 |

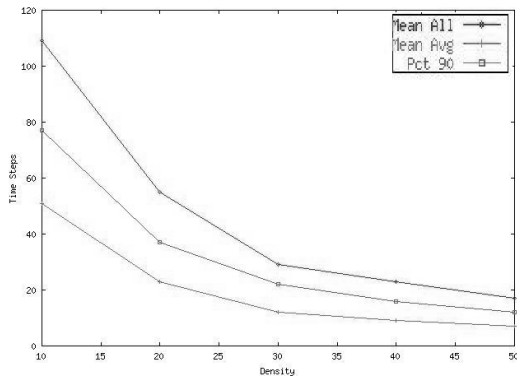


Figure 3: (Case 1) Message passed only to neighbors at each time step

In any given time step, an agent will be connected to (within range of) zero or more other agents. Those agents, in turn, may be connected to still other agents, and so on. This “collection” of connected agents comprises an ad hoc

network within the system. Case 2 exploits the existence of these networks by passing messages throughout the networks before advancing to the next time step. This is justified by the realization that radio communication, for example, can take place much more quickly than the agents themselves move. As before, a 50 x 50 grid was used and the density was varied from 10% to 50%. The results of these runs are summarized in Table 2 and Figure 4.

In Case 3 simulations, most of the agents moved as in Cases 1 and 2 but some percent of them was selected to move back and forth across the grid in an otherwise fixed direction. What this means, for example, is that an agent may start moving north and will continue to do so until the north boundary of the grid is reached. It will then reverse direction and move south until that boundary is reached then reverse again, and so on. Other “fixed direction” agents would start by moving south, east, or west and reverse direction in a similar manner when boundaries are reached. This behavior might correspond to specialized agents in a transportation system, search and rescue systems, or survey systems. The results of these runs are summarized in Table 3.

Table 2: (Case 2) Message passed throughout existing networks at each time step

| Density | All | Avg | 90% |
|---------|-----|-----|-----|
| 10% | 88 | 40 | 65 |
| 20% | 35 | 16 | 27 |
| 30% | 16 | 6 | 11 |
| 40% | 6 | 1 | 1 |
| 50% | 5 | 1 | 1 |

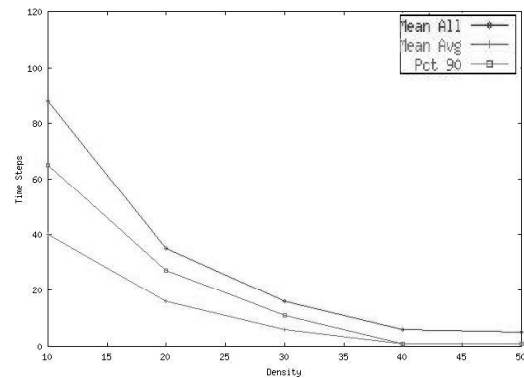


Figure 4: (Case 2) Message passed throughout existing networks at each time step

Table 3: (Case 3) Some agents moving in a fixed direction

| Density | Fixed | All | Avg | 90% |
|---------|-------|-----|-----|-----|
| 10% | 10% | 109 | 47 | 75 |
| | 20% | 94 | 39 | 61 |
| | 30% | 108 | 46 | 74 |
| 20% | 10% | 52 | 25 | 39 |
| | 20% | 57 | 24 | 39 |
| | 30% | 63 | 24 | 38 |
| 30% | 10% | 31 | 12 | 21 |
| | 20% | 31 | 12 | 19 |
| | 30% | 32 | 14 | 23 |

In Case 4 simulations the agents held a message for only a given number of time steps after acquiring it and then purged it from memory. In this case, agents that had once carried the message refused to accept it after that from other agents. Here we were interested in the percentage of the agents who received the message before it was lost in the system. Results from these runs are summarized in Table 4.

Table 4: (Case 4) Message retained for 10 time steps

| Density | %Delivered |
|---------|------------|
| 10% | 27 |
| 15% | 96 |
| 20% | 100 |

If the density was 20% or more, all – or nearly all – of the agents received the message before it was lost. In fact, if the density was as low as around 14%, it could reasonably be expected that about 90% of the agents would receive the message. Other simulations showed that around 70% of the agents would still get the message when the retention time was cut to half of that, or five steps. The reason for this can be seen from watching the display during the runs. As shown in Figure 5, when the density is sufficiently high, the message carrying agents form an advancing wall ahead of the area in which agents have had but lost the message.

As stated above these simulations all involved a test region defined by a 50 x 50 grid. While this provides a basis for comparing results of one simulation with those of another, it also raises the question of whether or not the shape of the region can affect the results. For example, would the number of steps needed to saturate two regions having the same area but different

dimensions be approximately the same or would they differ significantly? If the latter is the case, what is the correlation between the dimensions and the required number of steps? This question was addressed for Case 2 in the next set of simulation runs.

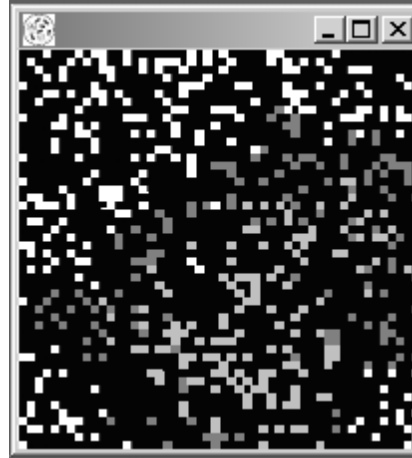


Figure 5: (Case 4) Dark gray agents form “advancing wall” across grid

These simulations compared a number of regions having areas equal, or nearly equal, to 2500. Starting with a long narrow region, the ratio of width to length was increased until the region was a square. The shapes used are described in Table 5.

Table 5: (Case 2) Different dimensions tested

| Length | Width | W / L |
|--------|-------|-------|
| 250 | 10 | 0.04 |
| 167 | 15 | 0.09 |
| 125 | 20 | 0.16 |
| 100 | 25 | 0.25 |
| 81 | 31 | 0.38 |
| 61 | 41 | 0.67 |
| 50 | 50 | 1.00 |

Simulations were run for densities of 10, 20, 30, and 40 percent for each of the regions listed. The number of time steps taken by a message originating near the center of the region to saturate the area was recorded for each run. The results of these runs are summarized in Figure 6.

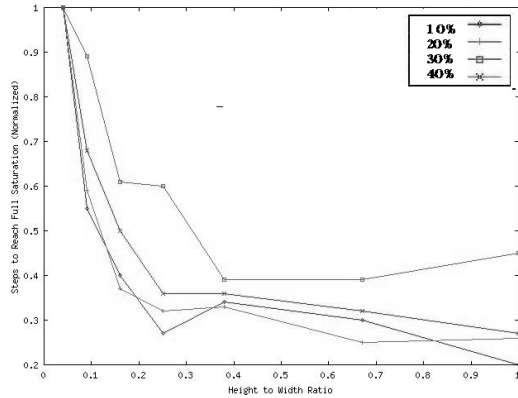


Figure 6: (Case 2) Time steps (normalized) to saturate regions of different dimensions.

The dimensions of the regions reported in Figure 6 are given in terms of the ratio of width to length as shown in Table 5. Since the number of time steps required to saturate an area can differ by an order or two of magnitude, depending on the agent density, the step counts for each are normalized to the interval [0.0, 1.0]. These counts for each density are shown plotted for each shape.

In order to examine this method of message passing further and to validate the results obtained, a second simulation was built using StarLogo – an agent-based modeling tool developed at MIT and designed especially for studying the local interactions of individuals in systems exhibiting self-organizing behavior [4, 27].

Table 6: (Case 4 replicated in StarLogo) Saturation percentage for varying density/retention settings

| Density | Time Steps Retained | | | |
|---------|---------------------|-----|-----|-----|
| | 10 | 15 | 20 | 25 |
| 3% | 2 | 5 | 5 | 8 |
| 5% | 5 | 10 | 20 | 19 |
| 8% | --- | --- | 58 | 86 |
| 10% | 24 | 60 | 93 | 98 |
| 13% | --- | --- | 99 | 100 |
| 15% | 82 | 97 | 99 | 100 |
| 18% | --- | 99 | --- | --- |
| 20% | 98 | 100 | 100 | 100 |
| 25% | 100 | 100 | 100 | 100 |
| 30% | 100 | 100 | 100 | 100 |

The Swarm and StarLogo simulations, although developed on different tools by different programmers, yield almost identical results (compare Table 4 with the first column of Table 6), thus increasing our confidence in the simulations. Additionally, more runs were made comparing the effects of varying the densities and retention times for Case 4 simulations. These results are summarized in Table 6 and graphed, along with data from runs with shorter and longer message retention times in Figure 7.

5. ANALYSIS OF RESULTS

There were four categories of simulations, described above as Cases 1 – 4, run for this study. In some of them a message, once acquired by an agent, was held indefinitely by that agent. In others, the agent purged the message from its memory after some fixed number of time steps. In most of the simulations the agents moved in a random direction at each step but in one set some percentage of the agents was selected to move in a fixed direction. Of primary interest was the speed with which a message can be dispersed throughout the agents in the modeled “world.”

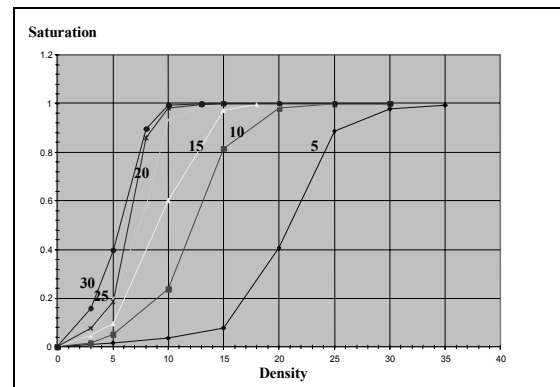


Figure 7: (Case 4) Saturation at different densities for retention times of 5, 10, 15, 20, 25, and 30 steps.

It is important to understand the meaning of time steps in the context of the models used. An agent can move one space in the grid in a single time step. One way of viewing this is as a speed limit – or definition of maximum speed capability – for the agents. This implies that the network cannot be reconfigured more frequently than once per time step. Another view is to see a time step as an interval during which messages

can be passed from one agent to another or, in some cases, among all agents comprising an ad hoc network. This view is roughly equivalent to the idea of “hops” which relates to the number of intermediate nodes a message visits on a network before finally arriving at its intended destination.

The first of these views raises the question of what is meant by saying that an agent moves from one grid position to another. In other words, what does the grid represent? At least two interpretations are possible. One is that they depict physical areas such as square miles or square yards. Another is that they refer to the ranges of whatever communication devices the agents are using. This latter view relaxes the implication that all agents move with the same speed but introduces the idea that an agent’s physical speed is proportional to the range of its radio – a somewhat unrealistic notion. Rather than choose an interpretation here these features of the model are mentioned in order to provide a basis for interpretation of the results. Other, more flexible, models are discussed in the Conclusions and Future Work sections that follow.

In a realistic “real world” application, the speed of messages between agents or throughout an ad hoc network is likely to be very much greater than the speed at which the agents themselves travel. This means that the time required to pass a message from one agent to another is dominated by the time spent physically moving. Accordingly, the simulation results are expressed in terms of time steps.

Recall that there are two defining characteristics of the agents in these simulations. The first is that they are autonomous, that is that the actions of one agent are independent of those of other agents except when they are logically close enough to each other to pass messages or when they are physically close enough that they may bump into each other (or that they must take action to avoid a collision). The other is that there is no global knowledge of the state of the system available to the mobile agents. Because of this it is difficult to compare results from these simulations directly to those describing more “typical” networking environments in which, for example, the nodes connected at any given time may be known.

For this reason, an initial set of simulations, Case 1 reported in Table 1 in Section 4, above, was

run in which messages are simply passed from an agent to its neighbors at each time step. In other words, the ad hoc networks were not exploited. The intention of this is to provide a basis for comparison of subsequent results. However, some interesting observations can be made about this data itself.

The measurements taken for this case were the time required (in time steps) for a message to reach all² of the agents on the grid; the time needed to reach ninety percent of the agents; and the average time to reach all of the agents. As can be seen in the run-time graph from a typical run with 20% density, shown in Figure 8, most of the agents receive the message noticeably sooner than the last one receives it. As a result, the average time for all agents to receive the message is only about 40% of that for all agents. Likewise, 90% receive the message (that is, 90% saturation is achieved) in about 70% of the time required for full saturation.

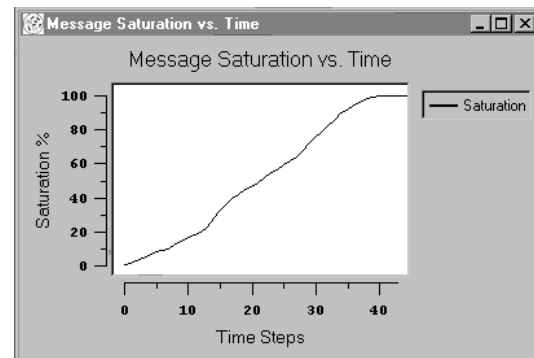


Figure 8: Typical³ run-time graph at 20% density

Often the time taken to pass a message from one node in a network to another is reported in terms of “hops,” which refers to the number of times the message is passed to an intermediate node. The notion of time steps as used here is similar to that of hops except that time steps represent

² It should be noted that the autonomy of agents and the absence of centralized knowledge of the system’s state make it impossible to guarantee delivery of a message to all agents.

³ Differences between the values depicted here and those reported in Table 1 are due to the fact that Table 1 data represents an averaging of data collected from several runs whereas Figure 7 shows a specific run.

the number of times that a message could have been passed somewhere in the system, but not necessarily the number of times that it was. The distinction is that agents holding a message will carry it with them, possibly passing it on in later steps. In a sense, it could be said that an agent moving from one location to another represents two distinct nodes in the network through which the message is ultimately delivered – one node while in its first position and another while in its second. This view provides a means of comparison to hop-based routing statistics. It should be kept in mind, however, that some of these “hops” are used not to pass the message in the conventional sense, but to position agents so that the message can be passed.

For Case 2 simulation runs reported above (Section 4, Table 2) the existence of ad hoc networks in the system was exploited. In these runs the message was passed to all agents that were connected to any agent having the message during a single time step. That is, if any agent in an ad hoc network held the message at the beginning of the time step, it was passed to all other agents in that network before the end of the time step. (This may have required several steps in the simulation but the time step counter was not advanced and no movement of agents was permitted.)

Comparing Tables 1 and 2 shows a reduction in time steps from 109 to 88 to reach all agents in a ten percent dense grid when the ad hoc networks are used. This is a reduction of 19%. At twenty percent density, the improvement is 36%. As the density increases, the improvement seen from exploiting the networks increases as well. At fifty percent density, it reaches 70%. This is to be expected as it reflects the increasing likelihood that an agent will be touching (that is, be within range of) other agents as the grid becomes more densely populated.

Case 3 simulations examined the effects of restricting the movements of some number of agents to a fixed direction (or, actually, back and forth in an otherwise fixed direction). The data in Table 3 show that, as implemented, this did not have much effect on the efficiency of message dispersal. It is believed that this is because the fixed motion is still a subset of the potential random motions and that with all else being equal the difference is not large enough to greatly affect the outcome.

Case 4 simulations considered agents that did not retain the message indefinitely but purged it after some number of steps and, having purged it, refused to accept it again. As seen in the Table 4 data, at twenty percent density all agents would receive the message before it was lost in the system even when the message was retained for only ten time steps. With a density as low as fifteen percent, 96% of the agents could be expected to receive the message before it was lost.

Next, Case 2 was revisited and simulations were run for regions having different dimensions. All areas were rectangular and varied from a long, narrow region to a square. In each of the other sets of tests reported above the region studied was a 50 x 50 grid. These dimensions were used for the square region examined here. The other regions studied were defined in such a way that the area remained as close as possible to 2500. In each run the message originated near the center of the region.

As can be seen from the graph in Figure 6, the message in the long, narrow region (width to length ratio of 0.04) took a large number of steps to saturate the region relative to those required for a square region ($W/L = 1.00$). This was true regardless of the agent density.

Also, regardless of the density, as the width/length ratio increased, the number of steps required for saturation dropped quickly and at around 0.25 became almost flat. This suggests that messages can be more efficiently disseminated throughout regions for which the length is at most four times the width.

Since the passing of messages in the systems being studied relies on the movement of the agents, this observation can be explained by the fact that the more narrow a region is the more restricted the movement of agents in the region⁴.

Finally, the Case 4 saturation data listed in Table 6 show a strong correlation with the data from the Swarm simulations, which serves to validate the approach and the earlier results. Additionally, these runs show that an inverse relationship exists between the density and the number of time steps required to reach saturation. That is, if a message is retained for a

⁴ If this is not readily apparent, consider the extreme case of a region having a width of one.

longer period, saturation can still be reached in a system with a lower density. This is not surprising but the data also show that at all density/retention combinations studied a threshold is clearly present above which full saturation – or nearly full saturation – can be expected and below which the message will be lost from the system before reaching all agents. Specifically, on average, the message reached 20 percent of the agents at settings below the threshold while, just above it, 80 percent or more received it.

6. CONCLUSIONS

This paper shows the potential for the efficient broadcast of messages in highly mobile ad hoc networks by treating the nodes in the network as autonomous agents. It is recognized that because the agents move on their own without any centralized control or even centralized knowledge of the system state it is not possible to guarantee that all agents will receive a message. It is possible, in other words, that an agent could move in such a way that it is isolated from message carrying agents at all times. Accordingly, it is reasonable – even appropriate – to consider message saturation levels of less than 100% which would nevertheless be acceptable.

It has been shown here that even at relatively low densities a message will be delivered to all (or nearly all) agents. Further, if the ad hoc networks are exploited, a significant improvement in the dispersal speed can be realized. It was also shown that if the agents purge messages after some period of time and refuse to reacquire them, the message is still efficiently dispersed even when the retention time is short. This indicates that messages can be routed efficiently in highly mobile ad hoc networks in which the nodes in the network are defined as autonomous agents and where no protocol is defined for passing messages except that defined in general for individual agents.

An important consideration in any mobile message-passing scheme is keeping the redundant transmission of messages to a minimum. Ideally, of course, agents would only transmit (or relay) a given message until all agents in the system have received it. However, with no globally aware controlling entity, it is not possible to know when that total saturation point has been reached. Agents can instead set a

limit on the number of time steps for which a message will be passed on. The Case 4 analysis shows that this number can be relatively low and still provide full, or nearly full, saturation.

Finally, it has been shown that the shape of the region can affect the efficiency of message saturation. This awareness can possibly be used to advantage when designing systems relying on this method of message delivery.

7. FUTURE WORK

The studies reported here involved passing a single message throughout a system of nodes. Each node, then, either had the message or it did not. This is good as a means of establishing a basis for further research but more ultimately useful results would consider systems in which several messages were being passed at any given time. In some of these simulations the message was purged after a period of time. A more realistic scenario is that an agent's buffer would become full and the agent would have to either refuse to accept new messages or purge older messages. This could be expressed in a model that supports several concurrent messages.

This work focused on broadcast messages. A natural extension is to refine these techniques to provide point-to-point or multicast message routing. For this, the ability to return an acknowledgment would be helpful.

In one of the simulations reported here some of the agents were selected to move in a “fixed” direction. This did not produce results significantly different from those obtained when all agents moved at random. It would be interesting to introduce “super agents” with a greater communication range and compare results with these new agents moving at random and moving in predefined patterns including fixed direction lines but also circles, squares, and other patterns.

The effects of shape can be further studied. One question to consider is how the results reported here scale. That is, whether or not all square regions, say, behave similarly or whether the relative size of the area is a factor. Also, regions with non-rectangular shapes could be studied.

Finally, it can be observed that if an agent, *A*, is within communication range of another agent, *B*,

and some set S of other agents, at time t , then the probability that B will be within range of the members of S (who will also have received the message from A) at time $t + 1$ is greater than at later time steps. It is reasonable, then, to expect that redundant message transmission could be further reduced by not only limiting the number of time steps for which an agent will relay a message, but by delaying the start of that relay period. The scalable broadcast algorithm presented by Peng and Lu [19] makes use of this and future research can examine the effect of such a delay in the context of the work presented here.

8. REFERENCES

- [1] Jim Binkley and William Trost, "Authenticated Ad Hoc Routing at the Link Layer for Mobile Systems Wireless Networks", 7, pp 139-145 (2001). Kluwer Academic Publishers.
- [2] Jon Inouye, Jim Binkley, and Jonathan Walpole, "Dynamic Network Reconfiguration Support for Mobile Computers", Proceedings of The Third Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'97), pp. 13--22 (1997).
- [3] Neal Singer "Avalanche Victims Found Four Times Faster With New Sandia-Developed 'Swarm' Algorithm Technique", Sandia Lab News, Vol. 52, No. 2 (January 28, 2000).
- [4] Resnick, Mitchel, *Turtles, Termites, and Traffic Jams – Explorations in Massively Parallel Microworlds*, pp. 3, 34-35, The MIT Press (1994).
- [5] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz, "Swarm Intelligence, From Natural to Artificial Systems," Chapter 2, Oxford University Press (1999).
- [6] Micah Adler and Christian Scheideler, "Efficient Communication Strategies for Ad-Hoc Wireless Networks," Proceeding of The Tenth Annual ACM Symposium on Parallel Algorithms and Architectures, 1998, pp. 259 – 268.
- [7] Yu-Liang Chang and Ching-Chi Hsu, "Routing in Wireless/Mobile Ad-hoc Networks via Dynamic Group Construction," *Mobile Networks and Applications* 5, (2000) , pp. 27 – 37.
- [8] Charles Perkins and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," SIGCOMM 94, (1994) pp.234 – 244.
- [9] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, Mikael Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks," Mobicom '99, (1999), pp. 195 – 206.
- [10] K. Mani Chandy and Leslie Lamport, "Distributed Snapshots: Determining Global States in Distributed Systems," *ACM Transactions on Computer Systems*, Vol. 3, No. 1, (February 1995), pp. 63 – 75.
- [11] Amy Murphy, Gruia-Catalin Roman, and George Varghese, "An Algorithm for Message Delivery to Mobile Units," PODC 97, (1997), p. 292.
- [12] Charles Perkins, "Mobile Networking in the Internet," *Mobile Networks and Applications* 3, (1998), pp.319 – 334.
- [13] Hannes Hartenstein, Bernd Bochow, Andre Ebner, Matthias Lott, Markus Radimirsch, and Dieter Vollmer, "Position-Aware Ad Hoc Wireless Networks for Inter-Vehicle Communications: the Fleetnet Project," *MobiHoc 2001*, pp. 259-262.
- [14] Zong Chen, H T Kung, Dario Vlah, "Ad Hoc Relay Wireless Networks over Moving Vehicles on Highways," *MobiHoc 2001*, pp. 247-250.
- [15] Linda Briesemeister and Gunter Hommel, "Role Based Multicast in Highly Mobile but Sparsely Connected Ad Hoc Networks," *MobiHoc 2000*, pp. 45-50.
- [16] Subir Kumar Das, B. S. Manoj, C. Siva Ram Murthy, "A Dynamic Core Based Multicast Routing Protocol for Ad Hoc Wireless Networks," *MobiHoc 2002*, pp. 24-34.

[17] Yuanzhu Peter Chen and Arthur L. Liestman, "Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks," *MobiHoc 2002*, pp. 165-172.

[18] Brad Williams and Tracy Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *MobiHoc 2002*, pp. 194 – 205.

[19] Wei Peng and Xi-Cheng Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," *MobiHoc 2000*, pp. 129-130.

[20] Zygmunt J. Hass, Joseph Y. Halpern, and Li Li, "Gossip-Based Ad Hoc Routing," *Proceedings of the 21st IEEE INFOCOM*, (2002).

[21] Katia Obraczka, Kumar Viswanath, and Gene Tsudik, "Flooding Reliable Multicast in Multi-Hop Ad Hoc Networks," *Wireless Networks 7*, pp. 627-634, (2001).

[22] Christopher Ho, Katia Obraczka, Kumar Viswanath, and Gene Tsudik, "Flooding Reliable Multicast in Multi-Hop Ad Hoc Networks," *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 64-71, (1999).

[23] Swarm Development Group,
www.swarm.org

[24] N. Minar, R. Burkhart, C. Langton, and M. Askenzi, *The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations*. pp. 1-11, (1996).

[25] Fabrice, Chantemargue, Thierry Dagaëff, and Beat Hirsbrunner. "Emergence-based Cooperation in a Multi-Agent System," *Proceedings of the Second European Conference Cognitive Science (ECCS'97) Manchester U.K.* . April 9-11. . 91-96.

[26] Sam R. Thangiah, Olena Shmygelska, and William Mennell, "An Agent Architecture for Vehicle Routing Problems," *ACM Symposium on Applied Computing 2001*, pp. 517-521.

[27] StarLogo,
<http://education.mit.edu/starlogo/>