A Mobile Agent Based Mechanism to Discover Geographical Positions of Nodes in Ad Hoc Wireless Network

Romit RoyChoudhury
Dept. of Computer Science
Haldia Institute of Technology
West Bengal
INDIA

S. Bandyopadhyay*
PricewaterhouseCoopers Ltd.
Sector V, Saltlake
Calcutta 700 091
INDIA

Krishna Paul
CTS
Sector V, Saltlake
Calcutta 700091
INDIA

* e-mail : somprakash.bandyopadhyay@in.pwcglobal.com

Abstract

A mechanism to discover the geographical positions of mobile nodes in ad hoc wireless networks through the implementation of a mobile-multi-agent based framework is investigated in this work. In other words, we have developed an agent based information exchange and navigation protocol in order to make each node in the network aware of the positions of other nodes. Our primary aim is to collect position-related information from each node in the network and distribute them periodically (as updates) to other nodes through mobile agents. The notion of *stigmergic communication* has been used through the implementation of a shared information cache in each node. We have used the GPS (Global Positioning System) support at each node for the extraction of geographical co-ordinates, velocity and direction of movement of each node. We have also incorporated a predictive mechanism which helps each node to predict the instantaneous positions of other nodes, based on the information that agents had fetched in the near past. As a direct outcome of infiltrating geographical position information of all other nodes into each node, the foundations for designing efficient routing scheme and distributed network management gets automatically laid. We have presented the performance analysis of our protocol in support of the agent-based system, which proves to be effective, especially to the capacity of generating minimal congestion in the network.

1. Introduction

Ad hoc networks [1] are envisioned as a infrastructure-less networks where each node is a mobile routers, equipped with a wireless transceivers. A message transfer in an ad hoc network environment could either take place between two nodes that are within the transmission range of each other or between nodes that are indirectly connected via multiple hops through some intermediate nodes. This implies that the nodes, which act as intermediate nodes in the data transfer process, must be willing to participate in communication until successful message transfer has been accomplished. The failure of such an event would amount to messages getting lost or message transfer getting interrupted, thus generating unproductive congestion.

The dynamics of wireless ad hoc networks as a consequence of mobility and disconnection of mobile hosts pose a number of problems in designing proper routing schemes for effective communication between any source and destination [1,2,3]. The conventional proactive routing protocols that require to know the topology of the entire network is not suitable in such a highly dynamic environment, since the topology update information needs to be propagated frequently throughout the network. On the other hand, a demand-based, reactive route discovery procedure generates large volume of control traffic and the actual data transmission is delayed until the route is determined. Because of this long delay and excessive control traffic, pure reactive routing protocols may not be applicable to real-time communication. There are proposals to reduce the control traffic generated in reactive protocols. For example, Location-Aided Routing (LAR) Protocols [4] suggest an approach to decrease overhead of route discovery by utilizing location information for mobile hosts. Such location information may be obtained using the global positioning system (GPS). The LAR protocols use location information to reduce the search space for a desired route. Limiting the search space results in fewer route discovery messages. In a recent work, new MAC protocols using directional antennas has been proposed to improve the routing performance of Location-aided Routing [5]. However, if source node S does not know a previous location of destination node D, then node S cannot reasonably determine the expected zone. In this case, this algorithm reduces to the basic flooding algorithm.

In this paper we have assayed the issues of geographical-position-discovery through the implementation of a **mobile-multi-agent based framework**. In other words, we have developed an agent based information exchange and navigation protocol in order to make each node in the network aware of the positions of other nodes without consuming large portion of network capacity. This would eventually help us to implement an effective routing protocol without much overhead.

Mobile agents are an effective paradigm for distributed applications, and are particularly attractive in a dynamic network environment involving partially connected computing elements [6,7]. Intensive research on the "Insect-like Agent Systems" has been done over the last few years. The mobile agent systems have been popularly simulated in close resemblance to an ant colony [8]. Of particular interest is a technique for indirect inter-agent communication, called stigmergy, in which agents leave information in the cache (which other agent can use) of the nodes they have visited. Stigmergy serves as a robust mechanism for information sharing. Worker ants that leave pheromone trails when they venture outside their nest are using stigmergic communication. This notion has been used in [9,10].

Mobile agents or messengers that hop around in the network are a novel solution to the problem of position-discovery. The agents hop from node to node, collect information from these nodes, meet other agents in their journey, interact with both to collect updates of parts of the network that they have not visited or have visited a long time back**,** and gift these collected data sets to newly visited nodes and agents. A node therefore receives updated information about the other nodes in the network, from the agents visiting them at short regular intervals.

In this paper, our primary aim is to collect all position-related information from each node in ad hoc wireless network and distribute them periodically (as updates) to other nodes through mobile agents. The notion of stigmergic communication has been used through the implementation of a shared information cache in each node. It has been assumed that each node knows its position and velocity at any instant of time using GPS and this information is getting distributed to other nodes through agents.

2. Description of specific terms

2.1 Recency

A major aspect underlying the infiltration of topology information into mobile nodes is that the information carried must be recognized with a degree of correctness. Since the agent navigation is asynchronous and there is an obvious time gap between the procurement of information by an agent from one node and its delivery by the same agent to another node, it becomes imperative to introduce a concept of recency of information. For example, let us assume two agents A_1 and A_2 arrive at node n, both of them carrying information about node m which is multi-hop away from node n. In order to update the position-related information at node n about node m, there has to be a mechanism to find out who carries the most recent information about node m : agent A_1 or agent A_2 ?

To implement that, every node in the network has a counter that is initialized to 0. When an agent leaves a node after completing all its tasks at the node, it increments that counter by one. We term this counter as ***recency token***. An agent while leaving a node (after completion of all the necessary information exchange and calculations) stores the new value of the incremented recency token against the node's ID within its data structures and continues navigating. Thus at any point of time, the magnitude of the recency token of any node represents the number of times that node was visited by agents since the commencement of the network. This also implies that if two agents have a set of data concerning the same node, say node A, then the agent carrying the higher recency token value of node A has more current information about it.

2.2 Time-to-Migrate (TtM)

An agent visiting a node is not allowed to migrate immediately to another node. In other words, an agent will be forced to stay in a node for a pre-specified period of time, termed as **time-to-migrate (TtM)**, before migrating to another node. By controlling TtM, the network congestion due to agent traffic can be controlled. For example, if TtM = 100 msec., for a single-agent system, it implies that the wireless medium will see one agent in every 100 msec. In our simulation, it has been assumed that an agent would take 1 msec. to physically migrate from one node to another. So, in this example, the wireless medium would be free from agent traffic 99% of the time.

2.3 Average Topology Convergence

We have developed a metric **average topology convergence** to quantify the deviation of actual network topology with the network topology perceived by individual nodes at any instant of time. Let us assume that (x, y) is the actual co-ordinates of a particular node A at a certain instance of time. Let (x_1, y_1) be the co-ordinate of the same node A as per the information that node p carries about it. Information about the node location is said to converge iff the distance between the points (x, y) and (x_1, y_1) is less than a specified *tolerance*. Thus node-location-convergence for node A at node p, defined as $V^A_p = 1$ if distance between the points (x, y) and (x_1, y_1) is less than the tolerance; otherwise $V^A_p = 0$. Average node-location-convergence as perceived by node p is defined as :

$L_p = (\sum_{\text{for } A = 1 \text{ to } n} V^A_p) / n,$

where n is the number of nodes. Thus the average network topology convergence C is defined as the average of all the node-location-convergence values over each of the nodes.

In other words: $C = (\sum_{\text{for } p = 1 \text{ to } n} L_p) / n$.

Average topology convergence is a formal measure of the deviation of the actual topology information with that of the topology information that the nodes possess about the other nodes in the system. It could be said that when C equals unity, it indicates that any node in the system knows the location of any other node to a precision of specified *tolerance*.

3. Issues in Implementing Agent Paradigm

The topology traversing could well be performed using a single agent. However this strategy fails to perform well in conditions of low transmission range where clusters gets formed due to groups of nodes, moving to some spatially remote portion of the bounded region. Quite obviously, since the agent is going to be in only one of the clusters, the other clusters have no agents at all in them although the members belonging to those clusters may be connected to quite a number of other nodes. Thus, the deprived clusters can no more get topology maps and might even get misled by the old, and thus incorrect, information.

The above mentioned issues cause no serious concern in the case of a multi-agent system. It might be of interest to observe that convergence does not necessarily improve with the increase in number of agents in the system. It has been observed in our previous work [10] that the optimum number of agents should be *half the number of nodes in the system.*

Another issue is to control the agent TtM to optimise congestion. We look here into the congestion introduced in the system due to variation in TtM. Let us assume that agents would take t millisecond to physically migrate from one node to another. Let us assume that our bounded region of ad hoc operation is A sq.mt., our transmission range R, the agent population P and the Time to migrate T msec. In an average case where the topology is evenly distributed over the region A, the number of areas in which agents could migrate between nodes simultaneously, without mutual interference, equals $A / (\pi .R^2)$. Now since the nodes are distributed, the agents would also be found equally distributed in each of these areas in an average case. Thus in any area the number of agents would equal P_a where,

$P_a = P.(\pi .R^2) / A$

As each agent migrates at a time gap of T milliseconds and takes only t millisecond to do so, the medium will be free from agent traffic $[(T – t.P_a) *100 / T]$ % of the time. For example, if the bounded region of operation is 1500×1000 sq. m. and R is 400 m, and P and T are 15 and 100 milliseconds respectively for a 30 node network and t = 1msec., then $P_a = 5$ (approx.) and the medium would be free from agent traffic during 95 % of the time.

4. Discovering node-positions using agents

4.1 Navigation Algorithm

The navigation algorithm must ensure that all member nodes of the network update themselves uniformly, irrespective of their position and state in the network. We have designed an algorithm, called **least-visited-neighbor-first** algorithm, and implemented it in our simulation model to estimate its effectiveness.

In this algorithm the agents select destination nodes that has the least recency_token value, implying that the node has been visited less frequently. An agent, residing in node n at any instant of time (termed as current host node of the agent), does the following:

1. Update information cache of node n with the information available in its information cache.
2. Selects all the nodes that are neighbors of n.
3. Evaluates the minimum value of the recency token of these selected neighbors from the information cache. This is the least visited neighbor, as perceived by the agent's host node n at that instant of time.
4. If this neighbor of n has not been visited in the last 3 visits by other agents from node n, select this neighbor as next destination. This history information is stored in the nodes. Else, select the second least-visited neighbors, and so on. This will ensure that multiple agents from same host node do not choose the same destination consecutively.
5. After choosing the right destination, it updates its next_detination node id with the destination node's id, changes the history table of the host node n with this newly selected node id.
6. Increments the host node's recency token value and stores this value against the host node's id within its own information cache.
7. The agent resumes navigation.

4.2 Information Exchange Protocols in Node-Agent and Agent-Agent Interaction

Infiltration of partial network information into the nodes is an asynchronous process, as the agents visit the nodes asynchronously. Thus it becomes acutely necessary to develop strategies for information exchange (i.e. to accept only that information which is more recent than what the node / agent already possesses). It is a two-step process.

In step 1, the recency tokens of all the nodes stored in the information cache of the current host node and the recency tokens of all the nodes stored in the information cache of the agent is compared. If the recency token of any node, say X, in the host node's information cache happens to be less than that in the agent's information cache, then it is obvious that the agent is carrying more recent information about node X. So, the entire information about node X in the host node's information cache is overwritten by that in the information cache of the agent. This step is performed asynchronously for all the agents as they arrive at that host node. This step helps the node to acquire all the recent information that it can gather from the agents. The agents however have not yet updated their data structures from the host node's information cache.

When an agent is ready to migrate (i.e. after a waiting time defined earlier as TtM), the step 2 is performed. In step 2, the agent copies the entire information cache of the host node on its own information cache. This contains the most recent information since the data set contains a combination of all the recent information that could be collected from the visiting agents and those that were already present in the node. With these updated values, the agents select their destination on the basis of the navigation algorithm previously described.

5. Topology convergence through a predictive mechanism

The foremost characteristic of a dynamic environment is that information at each node about other nodes in the network is never absolute. Information gathered by agents and nodes regarding the location and velocity are constantly changing. The changing information is brought to the nodes only when the agents arrive with fresh information about some other nodes. However, during the span of time when a node does not get an agent visit, the former has to depend on the old information that it has.

In this work, we have defined a concept of predicting the locations of other nodes by a node using a simple algorithm running at each node. Each node uses the location and velocity of other nodes, as perceived by the node, and forecasts the current locations of other nodes at each time-tick, based on the direction of movement of those nodes and the magnitude of their velocity. In other words, each node, at each unit time, predicts the location of the other nodes in the system on the basis of the position and velocity information that it has about other nodes.

Thus, even though a node does not receive agents for a long time, or does not receive information about some other nodes over a long time span, it would still be able to predict the current location of the nodes in the system, provided that the node under prediction does not change its direction of motion or magnitude of velocity.

6. Performance Evaluation

6.1 Simulation Set-Up

The proposed protocols are evaluated on a simulated environment under a variety of conditions to estimate the topology convergence against time. In the simulation, the environment is assumed to be a closed area of 1500 x 1000 square meters in which mobile nodes are distributed randomly. We present simulations for networks with 30 mobile hosts, operating at a transmission range of 400 m. The bandwidth for transmitting data is assumed to be 1 agents / msec. In order to study the delay, throughput and other time-related parameters, every simulated action is associated with a simulated clock. The clock period (time-tick) is assumed to be one millisecond (simulated). Agent TtM is assumed to be 50 or 100 msec. The speed of movement of individual node ranges from 5 m/sec to 30 m/sec. Each node starts from a home location, selects a random location as its destination and moves with a uniform, predetermined velocity towards the destination. Once it reaches the destination, it waits there for a pre-specified amount of time, selects randomly another location and moves towards that. However, in the present study, we have assumed zero waiting time to analyze worst-case scenario.

6.2 Protocol Performances

In figure 1 and 2, we present the results of two mechanisms for discovering geographical distribution of nodes. The two mechanisms are:

1. The non-predictive mechanism, which has been used as the base line for evaluating performance of protocols. This implements just the basic mechanisms of stigmergic communication concepts and would be further referred to as the *basic* mechanism. As shown in figure 1, the performance of basic mechanism can be improved by reducing agent TtM. However, this would increase the agent-traffic in the network.
2. The predictive strategy, which employs the mechanism to forecast the instantaneous location of a node based on the available topology information. This will be termed as the *predictive* mechanism. Figure 2 compares the basic and predictive mechanism at agent TtM= 100 msec. With the incorporation of the predictive mechanism, the performance improves significantly.

7. Conclusion

In this study, we have designed a mobile, multi-agent-based protocol to make the nodes in the network position-aware from the context of geographical distribution of nodes**.** It has been assumed that each node knows its position and velocity at any instant of time using GPS and this information is getting distributed to other nodes through agents. We have assumed that agents do not get lost in transit nor suffer from any kind of errors in transmission and reception. We have not considered the situations where a set of nodes switch themselves off after creating agents. In this situation, agent population in the network compared to number of active nodes would increase which would degrade the performance. In such a situation, some agents need to be destroyed. We have also not considered the cases where nodes may vary their velocity while moving towards a particular direction. In such a case the prediction mechanism might falter. However, our preliminary results indicate that a predictive, multi-agent-based protocol could be quite successful in infiltrating GPS oriented topology information into the nodes of the system, without consuming much network capacity.

References

1. D. B. Johnson and D. Maltz, Dynamic source routing in ad hoc wireless networks, T. Imielinski and H. Korth, eds., *Mobile computing*, Kluwer Academic Publ. 1996.
2. Z. J. Haas, Milcom'97 Panel on Ad-hoc Networks, http://www.ee.cornell.edu/~haas/milcom_panel.htm
3. E. M. Royer and C-K Toh, "A Review of Current Routing Protocols for Ad hoc Wireless Networks", IEEE Personal Communication, April 1999, pp. 46-55.
4. Y-B.Ko and N.H.Vaidya, "Location-aided Routing in Mobile Ad hoc Networks", Proc. of the ACM/IEEE MOBICOM 98, Oct. 1998.
5. Y-B Ko, V. Shankarkumar, Nitin Vaidya, Medium Access Control Protocols using Directional antennas in Ad Hoc Networks , Proc. Of the IEEE INFOCOM 2000, Tel-Aviv, 26-30 March, 2000.
6. R.Gray, D.Kotz, S.Nog and G.Cybenko, Mobile agents for mobile computing, Technical Report PCS-TR96-285, Department of Computer Science, Dartmouth College, Hanover, NH 03755, May, 1996.
7. Steve Appeleby and S.Steward. Mobile software agents for control in Telecommunications networks.*BT Technology Journal*,12(2):104-113,April 1994.
8. Schoonderwoerd, R., et al.(1997), Ant-based load balancing in telecommunications networks. Adaptive Behavior,5(2):169207,1997. http://www.uk.hpl.hp.com/people/ruud/abc.html
9. G.Di Caro and M.Dorigo, Mobile agents for adaptive routing. In Proceedings of the 31st*Hawaii International Conference on Systems*, January 1988.
10. Romit RoyChoudhuri, S. Bandyopadhyay and Krishna Paul, A distributed mechanism for topology discovery in ad hoc networks using mobile agents. Accepted in the First Annual Workshop On Mobile Ad Hoc Networking & Computing (Mobihoc 2000) in conjunction with Mobicom 2000, August 11, 2000, Boston, Massachusetts, USA.

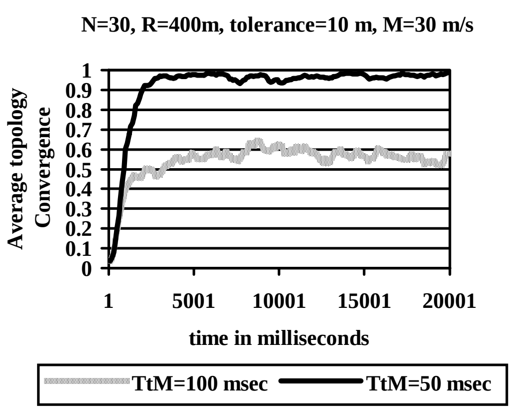


Fig. 1. Variation of average topology convergence against time with high mobility of 30 m/s and low tolerance of 10 m at different agent TtM.

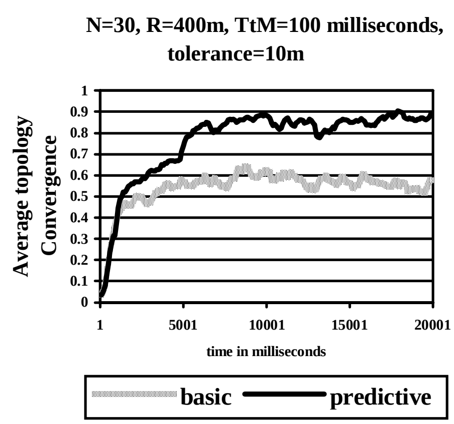
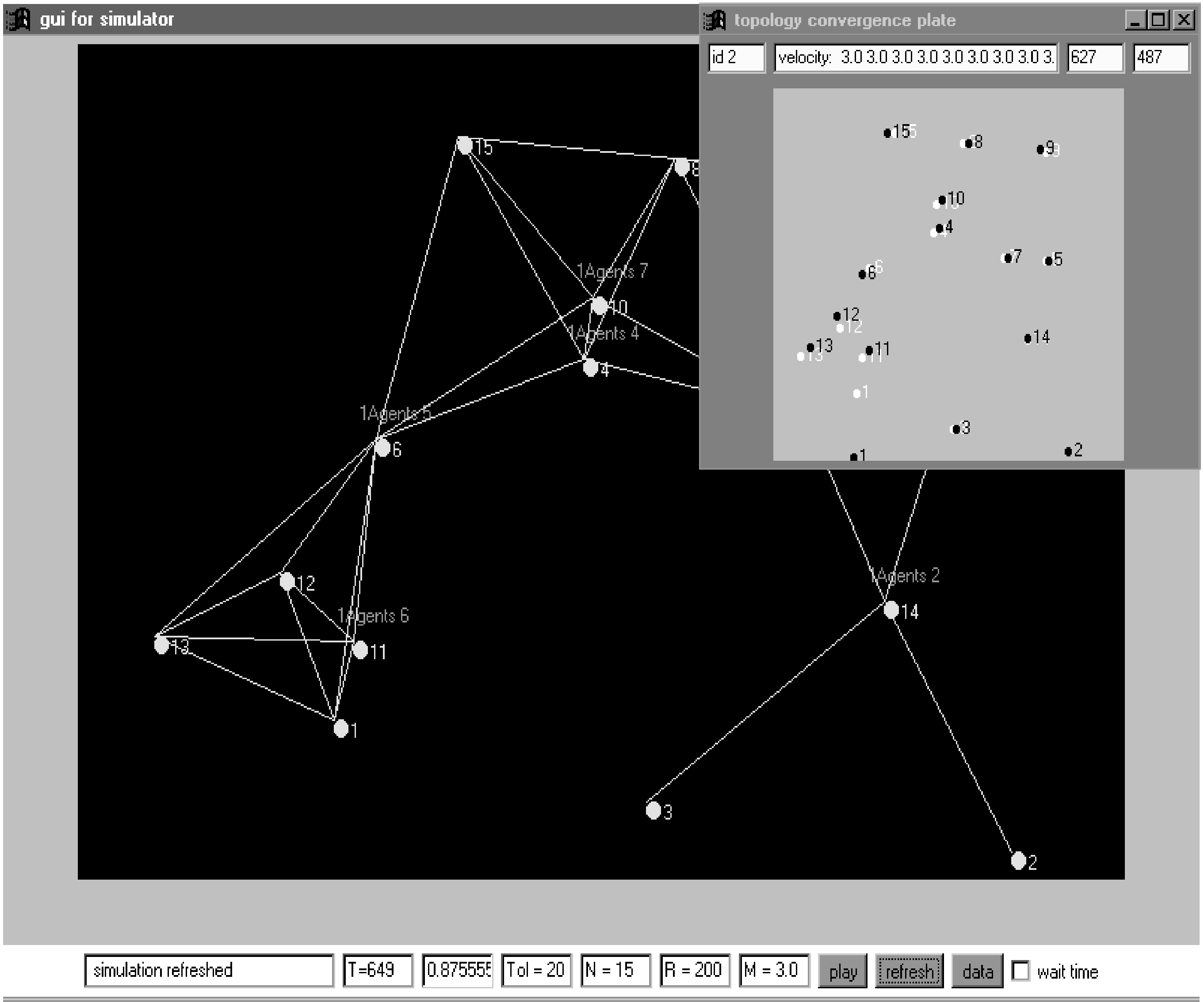


Fig. 2. Variation of average topology convergence against time with basic and predictive mechanism at agent TtM = 100 msec.

Appendix :



The simulator environment snapshot: the nodes move about with agents migrating between nodes. The inset window pops up if a mouse is clicked on any of the nodes. The window shows the information that the node possesses about the network topology. The inset window displays the nodes in the system in two distinct colors. The ones shown in white are the actual locations of the nodes and the ones in black are the locations that the node clicked on knows. The nodes whose locations are known accurately have the location in white overwritten by the dots in black.
The other parameters for the simulation are shown below in the text boxes. The instantaneous average topology convergence values are also displayed.