Location Based Dissemination Protocol For Mobile Adhoc Network - LDP

LOCATION BASED DISSEMINATION PROTOCOL FOR MOBILE ADHOC NETWORKS - LDP

Abstract

Location services are used in mobile ad hoc networks either to locate the geographic position of a given node in the network or for locating a data item. One of the main usages of position location services is in location based routing algorithms. In particular, geographic routing protocols can route messages more efficiently to their destinations based on the destination node's geographical position, which is provided by a location service, a predictive model-based mobility tracking method, called dead reckoning, is developed for mobile ad hoc networks. It disseminates both location and movement models of mobile nodes in the network so that every node is able to predict or track the movement of every other node with a very low overhead. The basic technique is optimized to use "distance effect," where distant nodes maintain less accurate tracking information to save overheads. In this paper we proposed a new protocol using dissemination with dead reckoning methodology, GLOMOSIM – (A Global mobile simulator) is developed for evaluating the location based services.

Index Terms – mobile ad hoc networks, location-based protocols, Location information service, performance evaluation, dead reckoning.

1. INTRODUCTION

A mobile ad hoc network [10] is an autonomous system of mobile, wireless nodes. There is no static infrastructure such as base station. If two nodes are not within radio range, all message communication between them must pass through one or more intermediate nodes. Thus, in such a network, each node functions not only as a host but also as a router. As the nodes are mobile, the network topology is dynamic, leading to frequent and unpredictable connectivity changes. It is critical to route packets to destinations effectively without generating excessive overhead. This presents a challenging issue for routing protocol design since the protocol must adapt to frequently changing network topologies in a way that is transparent to the end user. Knowledge of geographic locations of the nodes can aid in routing protocol design. Several location based routing protocols have been proposed (see, for example, [7, 8, 6]) in recent literature. These protocols utilize available location information of other nodes for low-overhead routing. However, for these protocols to be effective, this location information must be efficiently disseminated and/or updated in the proactively announce their locations to other nodes. While these schemes are simpler, they are usually inefficient as they are dependent on some form of periodic flooding.

2. RELATED WORK

There is a growing body of work that uses location information for routing in ad hoc networks. Early work started with the LAR and DREAM protocols. LAR [7] works as an add-on to any on-demand flood based protocol such as DSR [5] or AODV [10], and limits the route request flood within a zone where the destination is most likely to be found based on its past location and speed information. The location dissemination here is passive. DREAM [1], on the other hand, disseminates location information actively (periodically) and creates a location database locally. It then floods data packets within a cone radiating from the source, where the destination is guaranteed to be found within the broad end of the cone. The design of this cone follows from the available location information. (Note here that the DREAM routing protocol was not used in our study above; only the DREAM location service component was used, as our goal was to compare location service protocols with routing as an application.) Several techniques were proposed for geographic forwarding where availability of location information is assumed and this information is directed used for packet forwarding without having to explicitly establish a route. A very simple forwarding scheme simply forwards the packet to the neighbor closest to the destination[3]. This is the technique we used. As mentioned before, some techniques are needed if this greedy method reaches a dead end, i.e., there is no neighbor closer to destination than the current node itself. Karp and Kung in their GPSR technique [6], and Bose et. al. [2] independently, present efficient techniques to recover from such situations. Another solution is provided in [4]. In [9] authors propose GLS, a location database service. GLS uses geographic hierarchy to partition the network space onto grids.

Each node then maintains its current location in a number of location servers distributed throughout the network. Queries and registrations are sent to these servers using geographic forwarding. Care is taken such that location queries are sent to servers closer to the queries

4. DESIGN OF THE LDP

Our project design involves the designing of the following modules,

1. Position Estimation of Mobile Terminals.
2. Enhanced Location Dissemination.
3. Boundary Buffering Module.

4.1 Position Estimation of Mobile Terminals:

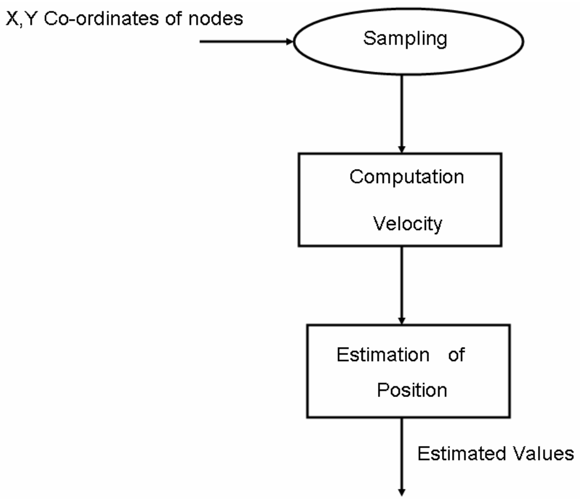


Figure 1.Position Estimation of MT

In this module, we use a very simple dead reckoning model which as a simple first order model that models a moving node's velocity (speed and direction). Complex models and use of path planning applications are possible, but not investigated here. We assume that each node in the network is aware of its location. Most commonly the node will be able to learn its location using an on-board GPS (Global Positioning System) receiver. Other methods such as radio-location [6], beacons from an available fixed infrastructure [5], or GPS-less positioning systems such as [10] are also possible. The accuracy of the location information will affect performance, though we will not discuss it here. In the dead reckoning model we have used, each node constructs a movement model for itself by periodically sampling its location estimates. It then computes its velocity components v_x

and v_y along the X and Y axes from two successive location samples (x_1,y_1) and (x_2,y_2) taken at times t_1 and t_2.

Thus,

$$V_x = \frac{x_2-x_1}{t_2-t_1}$$

and

$$V_y = \frac{y_2-y_1}{t_2-t_1}$$

After the first calculation of its velocity components, the node floods the network with this information using a *DRM update* packet. This packet contains the node's id, its location coordinates and the calculated velocity components. The location and movement models together form the dead-reckoning model. Each node maintains a DRM table. Whenever it receives a DRM update from another node it adds or updates an entry for that node's model in its *DRM table.* The model entry for each node in the table has a timestamp denoting when it was last updated. Thus, each node now has a location and movement model for every other node in the network. It uses this to estimate the location (x_{est},y_{est})) of the nodes at the current time as per the following formula:

$$x_{est} = x_{mod} + (v_{xmod} *(t_{cur} – t_{mod}))$$
$$y_{est} = y_{mod} + (v_{ymod} *(t_{cur} – t_{mod})),$$

where, (x_{mod} , y_{mod}) are the (x,y) coordinates and (v_{xmod} , v_{ymod}) are the velocity components of the node in the model table, t_{mod} = time at which model was updated and t_{cur} = current time.

Note that we have a tacit assumption here that t_{mod} actually reflects the time when the model is computed at the originating node. It is possible to include this computation timestamp in the DRM update, and use this time as t_{mod} instead of the update time. In that case, however, t_{mod} and t_{cur} would be timestamps taken on different nodes. So, for this scheme to work, the nodes must have access to synchronized clocks on all nodes. Synchronized clocks are not unrealistic as they can be obtained from GPS fixes. But even if such synchronization is not available, we expect that error introduced would be negligible unless the mobile nodes move at an unrealistic fast speed. The reasoning for this is as follows.

(i) Assuming that t_{mod} is the computation time, "loosely" synchronized clocks will work fine so long as the time-scale of the clock error is much smaller than the time-scale of a node's movement to significant distances. A significant distance here is a distance larger than the radio range.(ii) If no clock synchronization is available at all, t_{mod} should reflect the time when the DRM update is recorded. In this case, the speed at which update messages propagate in the network must be much faster than the speed of the mobile nodes.

After the initial distribution of its dead-reckoning model, each node continues to periodically sample its location (x_{cur},y_{cur}) and also computes its predicted location as per the above model it advertised. It calculates the deviation of its current location from its predicted location by simply computing the Euclidean distance

$$d = \sqrt{(x_{cur} - x_{est})^2 + (y_{cur} - y_{est})^2}.$$

If this distance d exceeds a predetermined threshold (called the *dead reckoning threshold*) the node recalculates the DRM and disseminates it again in the network.

Otherwise, no further dissemination takes place. The threshold essentially determines the allowable error in the location estimates. Too small a threshold results in too many updates adding to the load in the network. Too large a threshold results in inaccurate and stale location information persisting the network for a long time. It is expected that the application (e.g.,routing) that utilizes the location information is able to choose an appropriate threshold based on the desired level of accuracy.

4.2 Enhanced Location Dissemination Module

The basic technique can be upgraded for efficiency in large networks. The idea is to disseminate more accurate location information to nearby nodes and progressively less accurate information to far away nodes. This reduces control overhead. Most applications (such as routing) will tolerate less accurate location information for far away nodes (sometimes called "distance effect" [2]). A layering technique is proposed to achieve this.

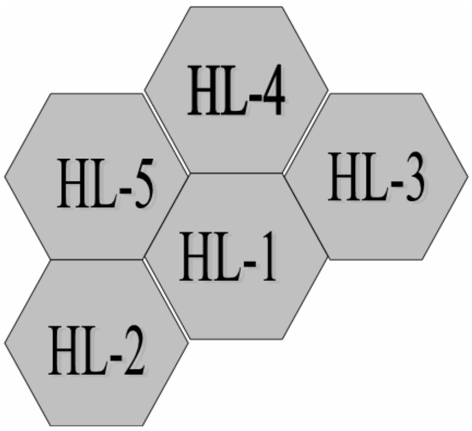


Figure 2.Adjacent Hexagonal Layers

Layering is not absolute; it is relative to a specific node . Each node has its own layer structure. The nodes in the network can reside in different layers with respect to different other nodes.

The thresholds of different layers are chosen such that the "angular error" threshold (maximum allowable angular location error) remains constant for all layers. From simple trigonometry, this requires that the threshold used in layer HL-i is i*threshold for the first layer (HL-1), assuming that the layer widths are the same. The threshold for the innermost layer is the only input parameter to the algorithm.

To implement the layering technique, each mobile node must maintain state information regarding the DRM updates propagated to nodes in different layers. Note that not all layers get all updates. It must also tag each update with a layer number before which an update flood must be stopped. This sounds straightforward, but this technique presents a problem. If the mobile node determines that nodes in a layer HL-i require an update, there is no way to send the update only to nodes in layer HL-i. All nodes in layers HL-1,…,HLi-1 must also receive and propagate the update. This is an unnecessary overhead.

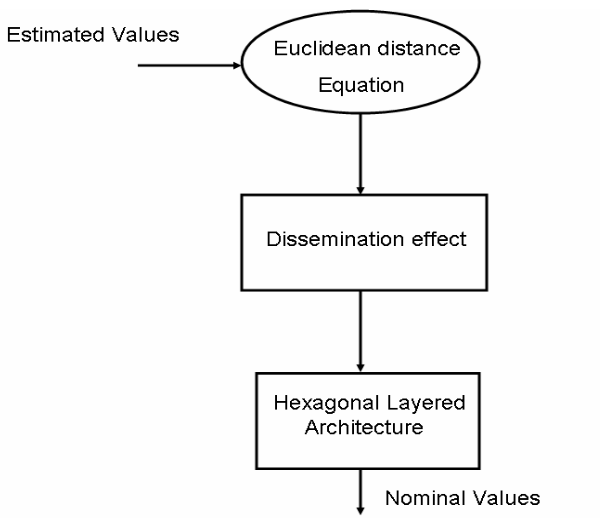


Figure 3: Enhanced Location Dissemination Module

4.3 Boundary buffering Module:

Assume that the mobile node is moving in a straight line in the direction shown. Assume also that a network wide update has been sent at time t=0,when the node is at A and the model used in this update predicts that the node is stationary. Assume that the DRM threshold for the hexagonal layer HL-1is s. When the node moves past B at time t=*t1*an update must be sent to all nodes in layer HL-1.Assume that this update also models the mobile node as stationary. When the node moves past at t=*t2* another update must be sent to nodes in layers HL-1and HL-2. Assume that enough samples have been gathered by this time for the movement model to be correctly predicted. Thus, when the node reaches D at t=*t3*,the update needs to be sent to nodes in layer HL-3 only. Nodes in HL-1and HL-2 already have the correct model of the node's movement. Now, how do we send updates to layerHL-3 nodes directly skipping over nodes in HL-1and HL-2? So nodes in HL-1 and HL-2 2 will get updates as well! These updates are redundant, as those nodes already have the correct model of the mobile node's movement.

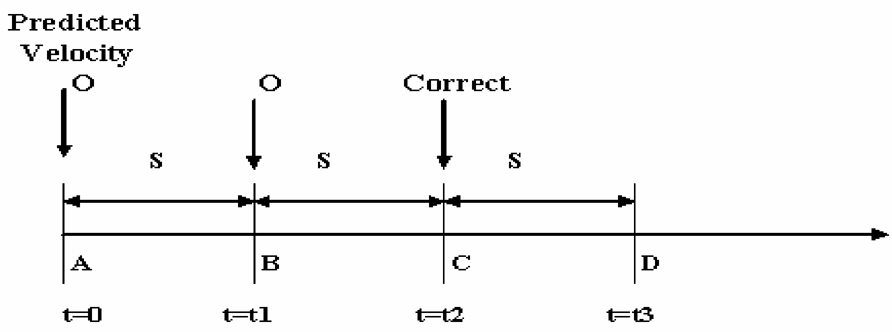


Figure 4:Example of Buffering Technique

To get around this additional overhead, we maintain the state information in the "boundary" nodes between two layers instead of at the mobile node in question. The idea here is to "buffer" a DRM update in a boundary node that "stops" that update,[1] and then propagate that update from the boundary nodes towards outer layers, if this update is needed in future by the outer layers. As a rule, a boundary node (say, P) always remembers the DRM update from a mobile node (say,N) that P last propagated. We refer to this by the term "old DRM," while the latest DRM update from N that P stops and buffers (i.e., does not propagate) is called "current DRM" or "buffered update." The old DRM serves as a reference to the DRM propagated to the topmost layer nodes. If, at any time, the deviation of the mobile node's location as per the old and current DRMs is more than the next outer layer's threshold, the buffered DRM update (same as the current DRM) is propagated to the next outer layer. Note that only one current DRM and one old DRM need to be maintained on each node for every other node in the network.

In the above example, at t=*t3* the nodes at the boundary of HL-2 and HL-3 realize that the current model that they previously buffered must now be propagated to HL-3.This is because at t=*t3* the old model predicts the mobile node to be at A whereas the current model predicts it to be at D, a deviation of 3*s.

5. SIMULATION OF LDP

5.1 GloMoSim

Global Mobile Information System Simulator (GloMoSim) is a scalable simulation environment for large wireless and wire line communication networks. GloMoSim uses a parallel discrete-event simulation capability provided by Parsec. GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking, and traditional Internet protocols. The following table lists the GloMoSim models currently available at each of the major layers:

Layer	**Model**
Physical(Radio Propagation)	Free space, two-Ray
Data Link(MAC)	CSMA,MACA,TSMA,802.11
Network(Routing)	Bellman-Ford,FSR,OSPF,DSR,WRP,LAR,AODV
Transport	TCP,UDP
Application	Telnet,FTP

Table 1. GloMoSim layer for wireless Environment.

The *node aggregation technique* is introduced into GloMoSim to give significant benefits to the simulation performance. Initializing each node as a separate entity inherently limits the the scalability because the memory requirements increase dramatically for a model with large number of nodes. With node aggregation, a single entity can simulate several network nodes in the system. Node aggregation technique implies that the number of nodes in the system can be increased while maintaining the same number of entities in the simulation. In GloMoSim, each entity represents a geographical area of the simulation. Hence the network nodes which a particular entity represents are determined by the physical position of the nodes .

5.2 Input Formats of LDP Routing

- From config file of GloMoSim:
- TERRAIN-DIMENSIONS (900, 1000)
- NUMBER-OF-NODES 20
- NODE-PLACEMENT FILE
  - NODE-PLACEMENT-FILE ./nodes. Input
- MOBILITY TRACE
  - MOBILITY-TRACE-FILE ./mobility. in
- RADIO-BANDWIDTH 50000
- ROUTING-PROTOCOL SO2P
- APP-CONFIG-FILE ./cbr.in

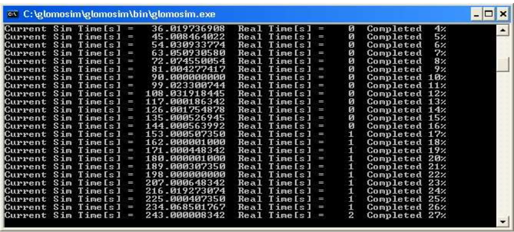
From cbr.in file of GloMoSim

- CBR 0 6 0 512 3S 40S 170S
- CBR 12 19 0 512 3S 80S 200S
- CBR 24 30 0 512 2S 0S 150S

From mobility. in file of GloMoSim

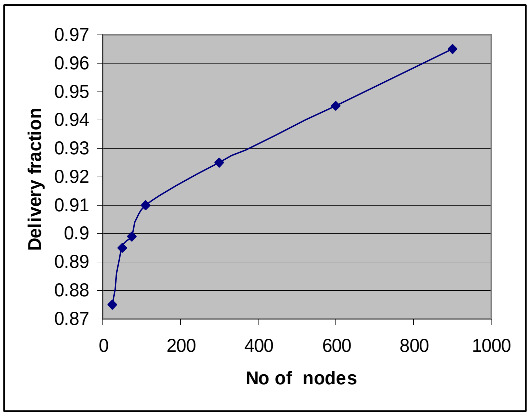
- 2 500MS (350, 50, 0)
- 15 500MS (510, 585, 0)
- 28 500MS (525, 735, 0)

5.5 Simulation of LDP



5.6 Observation Results

The following performance metrics are evaluated: (i) *Packet delivery fraction*: ratio of the packets delivered to the destination to those generated by the CBR sources. (ii) *Routing load*: the number of control packets transmitted per node per sec to maintain and propagate location information. We plat a chart for this two consideration, this is in figure 5.



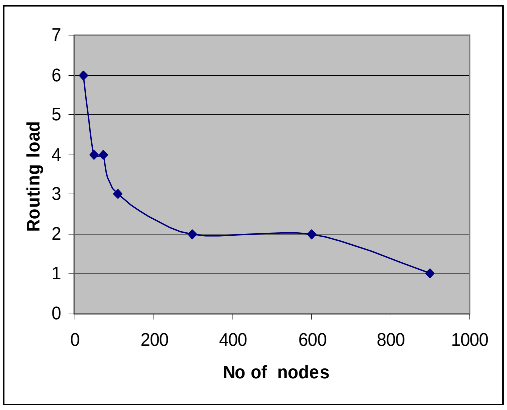


Figure 5: DRM performance with Delivery fraction, Routing load .

Number of nodes	Delivery fraction
25	0.875
50	0.895
75	0.899
110	0.910
300	0.925
600	0.945
900	0.965

Number of nodes	Routing Load
100	6
200	4
300	4
600	3
900	2

Table 2. LDP performance of delivery fractions and routing load.

Conclusion:

We have developed a location dissemination protocol called dead reckoning that uses predictive models to track node locations. Significant deviation from the predicted model triggers location updates. Better prediction keeps control overhead lower. The basic protocol has been optimized to maintain a more accurate location information on nodes closer to a mobile node relative to farther nodes by using a layering and buffering technique. In future we will evaluate the performance of this protocol against three other significant dissemination protocols. We have chosen geographic routing as an application for performance evaluation.

References:

www.technicalpapers.co.nr