A Comparative Study of Mobility Prediction Schemes for GLS Location Service

Shshank Sharma, Venugopal Alatzeth, Gurpreet Grewal, Saurabh Pradhan, Ahmed Helmy
Department of Electrical Engineering
University of Southern California
Los Angeles, USA
{shshanks, alatzeth, ggrewal, sapradha, helmy}@usc.edu

Abstract— **Performance of geographic routing suffers from mobility-induced location errors. Location errors can also occur due to infrequent and/or lost updates to location servers, especially when the nodes are highly mobile. In GLS the location update frequency to the higher order location servers is very low. A query to a location server fails when a node moves far away from its previous location rendering the cached location in the location servers invalid. In this work we present a detailed study on the impact of node velocity on query failure rates and reasons in GLS, across different mobility models. Correct and efficient mobility prediction by the location serves themselves can improve the query success rate in the Grid/GLS framework. We present a comparative study to investigate the performance of various prediction schemes with GLS over a rich set of mobility models.**

Keywords- Mobile ad hoc networks, geographic routing, mobility prediction,GLS, prediction, location prediction.

I. INTRODUCTION

Geographic routing has been proposed as a solution for routing in mobile ad hoc networks. It eliminates the need for route set up and maintenance in mobile networks by making use of the geographic locations of the mobile nodes. For geographic routing to work, the location information of the destination has to be known. Various location services have been suggested for this purpose. The Grid Location Service (GLS) [1] is a distributed location service in which each node in the network maintains a part of the overall location database. It uses the concept of a grid wherein all the nodes are aware of the complete grid topology and each node knows only a cell-worth of other nodes. Therefore for the purpose of reaching any node in GLS, the location of the each node is reduced from an actual geographic position to the grid location.

In GLS, a node is called the location server of another node if it stores the mapping of the node's ID to its location. Three main functions of GLS are location server selection, location query request, and location server update. If node A wants to find the location of node B, then A will send a request to the least node greater than or equal (in circular ID space) to B for which A has location information (using greedy forwarding). The node receiving the request forwards the query in the same way until the query reaches one of the location servers that know the location of B, and this location server will forward the query to B itself. B responds directly to the requestor (A) since the request contains A's location.

A. Helmy was supported by NSF Career, Intel and Pratt&Whitney Institute for Collaborative Engineering (PWICE)

The rate at which nodes update their location servers falls off with increasing distance from the location server, thereby minimizing update traffic. However, this also means that the location update frequency to the higher order location servers is low. A query to a location server fails when a node moves far away from its previous location rendering the cached location in the location servers invalid. Location errors can occur due to infrequent and/or lost updates to location servers, especially when the nodes are highly mobile.

We performed an extensive study of the reasons for query failure in GLS for various mobility models [2], namely the Random Way-Point (RWP), Reference Point Group Mobility (RPGM), Freeway (FW), and Manhattan (MH), across a range of node speeds. We observed an increase in the query failure rates with node velocity due to location errors.

In order to study potential improvements in GLS query success rate by incorporating prediction in the location servers themselves, we implemented three prediction schemes, namely linear velocity prediction (LVP), weighted velocity prediction (WVP) and recent history based Order (1) Markov prediction (MHP). We performed a comparative study to investigate GLS performance for the three prediction schemes.

The remainder of the paper is organized as follows. Section II explains the related work covering various prediction schemes. Section III explains the environment for our simulations and analysis of the GLS query failure reasons. The implementation of the prediction schemes is presented in Section IV, and the simulation results are discussed in Section V. Finally, our conclusions from this study and planned future work are listed in Section VI.

II. RELATED WORK

A. Prediction Schemes

The improvements suggested in [3], namely NLP and DLP, advocate location prediction by forwarding node. However, this is more useful when routing packets hop-by-hop. We address the issue at the query point (location server) even before a destination location is given to start the routing. Problems may arise if the location information maintained in the location server itself is in error. The situation further deteriorates in presence of node mobility. To solve this problem we suggest that the location server employ a

0-7803-8521-7/04/$20.00 © 2004 IEEE

prediction scheme to proactively provide more accurate location information. This will provide better improvements if the location server queried by the source is far from the destination node in terms of distance, as making a correct prediction the routing would choose paths better to reach the correct destination location. For example, a second or third order square location server in GLS or a server in a far-off region is more likely to have incorrect location information, hence would result in a better accuracy with prediction. Various prediction schemes have been suggested in [4], [5], [6], [7] and [8]. [4] discusses predicting random movement of nodes using random mobility model. [5] explains the cause and effect relationship between group mobility and network partitioning, and suggests methods to predict network partitioning. [6] compares various unicast and multicast routing protocol performance with and without prediction of link and route expiration times. [7] suggests a method to capture history of node movements using efficient path updates. [8] implements a prediction scheme using movement patterns in a grid setting. To the best of our knowledge no prediction scheme has been suggested for location servers in GLS. In this work, we investigate the query failure rates along with associated reasons (across different node speeds, node densities and mobility models) in detail to determine the major factors contributing to query failures. We then analyze the applicability of three prediction schemes to GLS, over a rich set of mobility models.

III. GLS Analysis

A. Simulation Background

All of our simulations are based on the GLS code ported to ns-2.1b8 [9]. They were performed using CBR traffic and a setting called *test queries only*, resulting in queries being sent out instead of data. The area of simulation used was 1000m x 1000m (unless specified otherwise). We used the IMPORTANT [2] mobility tool to generate scenarios and simulate node movement for the Freeway, Manhattan and RPGM mobility models. The packet types associated with location information are: *Hello, Location Update, Location Query, Query Response, Location Notification and Forwarding Pointer Update* [9]. Of these we mainly focus on the location query and response packets since we are interested in investigating GLS query failure rates. The important reasons for dropped queries are: *RLOOP, NRTE, NOSRVF* and *TTL* [9]. GLS avoids routing loops by limiting the number of geographic route hops from location server to the destination to four. When the number of hops is exceeded the query packet is dropped with RLOOP as reason. Thus RLOOP hides location errors and prediction may be helpful in alleviating this. When a geographic void is encountered, packet is dropped with reason code NRTE. When no location server is found through GLS logic (the number closest in id space), query is dropped with NOSRVF as reason.

B. Simulation Results and Analysis for Query Drops

This section presents the simulation results for GLS query failures and failure reasons (RLOOP, NRTE etc.) for different mobility models and node speeds.

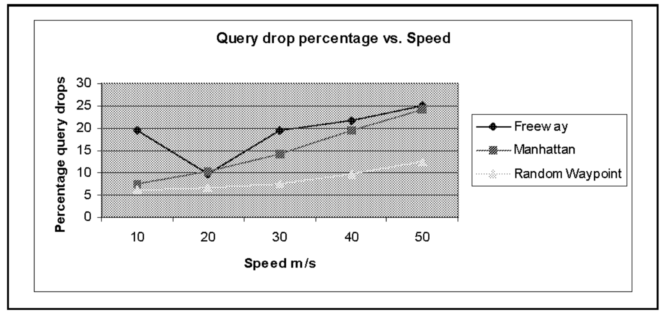

Figure 1. Query Failure Rate for 100 nodes in 1000m x 1000m area

RWP: Simulations for RWP were done in a 1000m x 1000m area with a grid size of 250m and update distance (d) 100m. Each simulation was run for 300s and results averaged across 5 runs. As the number of nodes or node density (because simulation area remains the same) increases the number of query drops reduce from a high of 75% (100 nodes – here the nodes are basically disconnected, due to low density) to a low of 20% (400 nodes), node speed being 30 m/s in these cases. This trend observed due to reduction in the number of NRTE (due to reduction in the number of voids with increasing node density). As node speeds increase from 10 m/s to 50 m/s (Fig. 1) the query drops increase from a low of 8% to a high of 40% primarily due to mobility induced errors (like RLOOP). Fig.2 shows the distribution of the reasons for the drop. (The percentage of drops is the drops for that reason per 100 drops). With an increase in speed from 10m/s to 50 m/s, failures due to RLOOP increase from a low of 0.37% to a high of 26.9%. NOSRVF (which are an inherent problem with GLS) stay constant at 31%. Node mobility seems to decrease the number of voids in this case, with the query failures due to NRTE reducing from 55.2% (at 10 m/s) to 22.2% (at 50 m/s). Thus, we can see that at low speeds NRTE dominates, at medium to high speeds, we see the drops due to RLOOP being significant.

Manhattan: Simulation area is 1000m x 1000m with grid size 250m. Simulation runs were for 300s each with

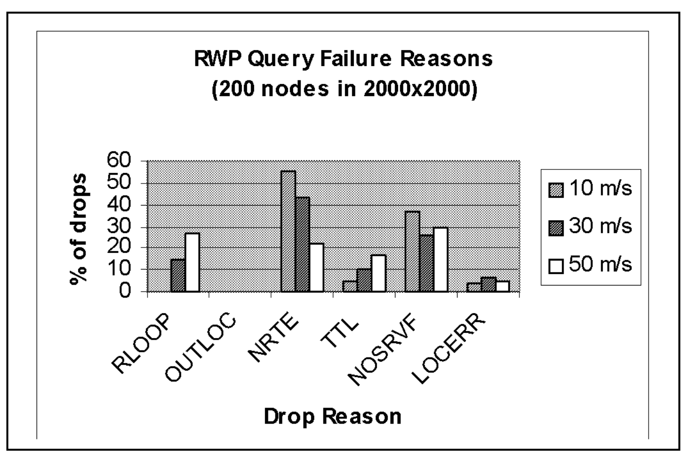

Figure 2. RWP Query Failure Reasons vs Speed

update distance being 100m. Query failures increase from a low of 3% to a high of 23% when max velocity increases from 10 m/s to 50 m/s (Fig. 1). NOSRVF with a contribution of 55-

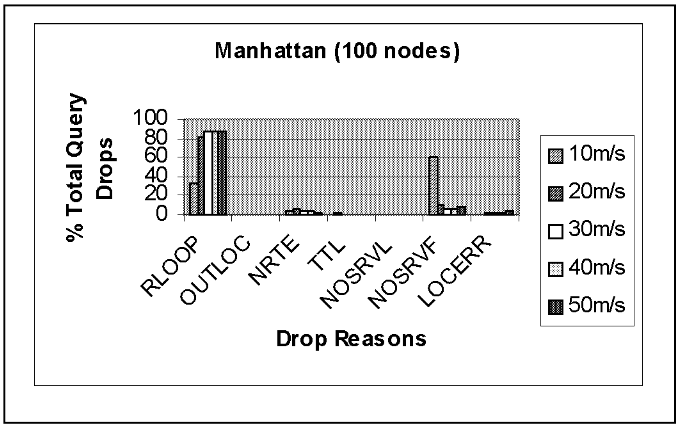

Figure 3. Manhattan Query drop reasons, 100 nodes in 1000m x1000m area

60% to total query loss rate dominates at low speeds. Nearly 80% of failures are due to RLOOP at speeds greater than 20 m/s (Fig. 3).

Freeway: Simulation parameters are same as in Manhattan. In case of Freeway mobility model as node speeds increase the query drops stay steady at 15-20%. Main reasons for query drops in this case are NOSRVF, NRTE and TTL. Our analysis of query drops and our observations from the visual tool (nam) make us conclude that GRID GLS may not be suitable for models with geographic restrictions that do not resemble a grid-like structure. We found that in this case nodes were unable to locate location servers due to non-uniform node distribution in grid area (which is the basic requirement for GLS to function). In a freeway based mobility model, the nodes are separated by linear or columnar geographic restrictions. The columnar structure of the freeway node distribution causes forwarding based on the GLS logic to be very costly in terms of hops and result in TTL expiry drops.

RPGM: Simulation area is 1000m x 1000m and we ran simulations for 300s with 4 groups of 15 nodes each (Total of 60 nodes). Update distance was kept constant at 100m. From our query drop analysis for the RPGM model we observed that NRTE dominates at lower speed (less than 30 m/s) and TTL dominates at high speeds (greater than 40 m/s). We observed an interesting trend that query failure rates increase from 57% to a high of 67% at max velocity of 40 m/s; thereafter it drops off to lower values with increasing speeds. We attribute this to an increased connectivity with higher mobility. Another observation we make here is that is all simulation runs for RPGM most of the communication was within the group and the number of queries generates were far too less than generated for other models. Hence we studied the RPGM model for inter-group connections with the same group constitution as mentioned earlier. The connection pattern used connected at a random time during the simulation one node from one group to a node in another group. The inter group request for queries suffers from NRTE and NOSRVF drops. These are primarily due to network partitioning.

IV. PREDICTION SCHEME IMPLEMENTATION

Velocity Based: We implemented two different velocity based prediction schemes as described below:

a) Linear Velocity Prediction (LVP): This implementation uses a simple prediction approach. Each node maintains two history samples for every node for which it is a location server. Each history sample consists of an (x,y,t) triplet (where t is the time at which the update was received for location (x,y)). Node speed is calculated using the distance-time formula. Further, the location server predicts the current location of the node using the time elapsed since the last location update and the calculated speed.

b) Weighted Velocity Prediction (WVP): In this scheme we take running weighted average (similar to TCP RTT calculations) of node velocity for doing prediction. Whenever the location server gets a location update for any node, it will calculate and store the nodes velocity using the following equation:

$$V_{av} = \alpha\, V_{av} + (1-\alpha)\, V_{rec} \qquad (1)$$

Where V_{av} is the weighted average velocity, V_{rec} is the most recent velocity and α is the filter gain constant.

The location server uses V_{av} to predict the node's current location. Lower the α value, higher is the contribution of current velocity to the weighted average. At $\alpha = 0$, WVP becomes the same as LVP. So LVP can be said to be a special case of WVP.

Both these schemes don't take into account the change in direction, which if incorporated is expected to provide an improvement in prediction accuracy.

History Based: The *O(1) Markov Recent History (MHP)* based prediction, as described in [10] is implemented as the history-based prediction scheme. Each node maintains a defined parameter (NUM_SAMPLES) worth of recent history samples (GLS grid numbers) for every node for which it is a location server. In order to predict the next grid for a particular node, the algorithm searches backwards in the list to find the preceding occurrence of the most-recent grid number. The grid number following this grid in the history is returned as the predicted value. In case there is no previous occurrence of the current grid, no attempt for prediction is made, and the value as per the default GLS behavior is returned. The scheme incurs no communication overhead, while the storage requirement at location server will increase as we increase NUM_SAMPLES. (One history sample for a single node takes 12 bytes in the current implementation).

A. *Investigated Parameter Space*

Our evaluation framework investigates the effect of the following parameters on prediction accuracy and GLS protocol performance: node speed, node density, mobility models, simulation area, location update frequency and grid size.

B. Evaluation Metrics

In our framework the following metrics are used for evaluating the benefit of incorporating prediction mechanisms into location servers:

Control packet overhead: This represents the change in terms of the number of bytes due to any additional communication the prediction scheme incurs.

Prediction accuracy: For velocity-based prediction it is the absolute difference between the predicted and actual location of the node and is measured in meters. For history-based prediction we measure the prediction accuracy in terms of grid numbers (number of times the predicted grid number is same as the actual grid number).

Query Success Rate: It is the percentage of queries sent out to location servers that are successfully resolved.

Storage Requirement: Increase in storage requirement at location servers for prediction scheme implementation.

V. SIMULATION RESULTS

All simulations were run for a 1000m x 1000m simulation area, with 100 nodes. Grid size is 250m and the update distance 100m.

A. Velocity Prediction

RWP: The prediction accuracy, as compared to the accuracy of GLS for the RWP mobility model, reduces as speed increases. This can be attributed to the facts that the nodes change direction in smaller intervals of time and velocity predictions do not take change of direction into account. Even though updates are sent at a greater rate, the need for prediction is based more on the connection pattern and thus may not be fresh enough. Prediction accuracy for the velocity based mechanism is better than the value GLS would have provided as the current location about 70% of the time for 10m/s and drops to about 64% at 40m/s. GLS with prediction tracks a node, better than original GLS by greater than 100m for about 15% of the time and by greater than 250m about 2% of the time. We observe that, if velocity prediction does give an incorrect location, it is generally very close to the value that the GLS system would have given without prediction, indicating that there has been no update received from the destination node and hence the error. The other observation is that the error

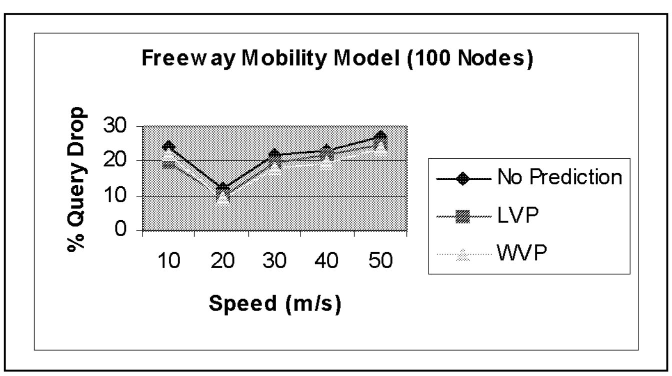

Figure 4. Freeway GLS query performance comparison

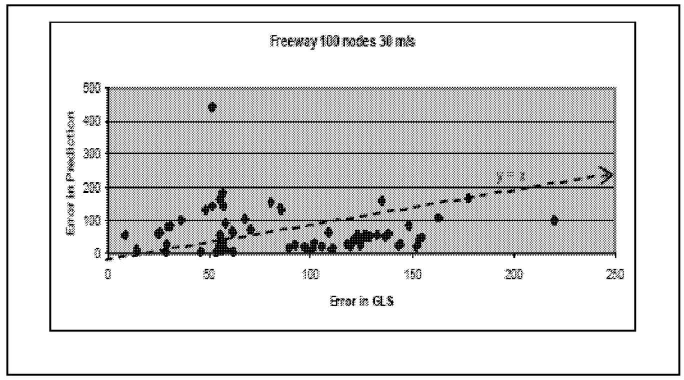

Figure 5. Freeway LVP accuracy at 30 m/s

in the prediction scheme is very large when RWP randomly inserts nodes once they reach simulation area boundary.

Freeway: We ran simulations at varying node speeds from (10 ms – 40 m/s), under three scenarios, original GLS, GLS with LVP and GLS with WVP. Fig. 4 draws a comparison between the three different scenarios and GLS with WVP having 5% better Query Failure Rate. LVP predicts a more accurate location 68- 72% of times (Fig. 5) compared original GLS although it leads to only a 2% improvement in Query Drops. We observe that the relative ratio of the drop reason components remains more or less the same, though the TTL and RLOOP drops do decrease location errors are reduced.

Manhattan: For Manhattan mobility model, LVP predicts a more accurate location for around 57- 59% of times than normal GLS. A comparison between the three different scenarios (original GLS, LVP and WVP) shows that WVP performs the best (though the improvement is not as visible as Freeway Model). The relative ratio of the drop reason components remains more or less the same, though the TTL and RLOOP drops decrease.

We observe that the velocity based prediction schemes perform better with Freeway mobility model as compared to Manhattan. This is because Freeway has higher geographic restrictions as compared to Manhattan (where node can move left or right at an intersection). Thus velocity based prediction (without taking into consideration the direction of movement) works better for Freeway. Performance for Manhattan is expected to improve if we consider the knowledge of map as well as the probabilities of node changing direction at any intersection. WVP works better than LVP in both cases because it filters out possible transient velocity changes by taking the weighted running average. The error in velocity prediction is large when the node is randomly inserted back in to the simulation area upon reaching the simulation boundary.

Both WVP and LVP schemes have no communication overhead and with the current implementation the storage requirement in location server increases by 12 bytes for each node (for which it serves as the location server) in case of LVP, whereas for WVP is just 4 bytes (as we store just weighted average velocity instead of (x,y,t) triplets). LVP being a special case of WVP, can also have its storage requirement reduced to 4 bytes (per node) if we run WVP with $\alpha = 0$. Thus for WVP if the location server stores locations for say N nodes, the storage requirement is N*4 bytes.

B. *History Based Prediction*

In addition to using the standard mobility models (including Random-Way Point (RWP), Manhattan (MH), and Freeway (FW)), we were also interested in looking at the performance for a movement scenario containing temporal patterns, since history-based schemes are intuitively expected to perform better in such scenarios. We created a simplistic pattern-based scenario for ns and evaluated both the original and modified GLS performance for the same.

Based on the analysis of GLS performance with and without history-based prediction (specifically MHP), we observed that there is no significant change in the query failure rates when MHP is used in GLS. The prediction made by the algorithm resembles very closely the choices that the original GLS implementation makes as regards to the location grid of a node. We investigated the reasons for the same. Intuitively, the update frequency has a significant bearing on the history formed at the location server. The accuracy of the predictor closely mirrors the rate at which the location updates come in from the node to the location server. In this respect, both the predictor and the default GLS implementation uses similar information in making decisions about node locations, hence the similarity in behavior. The method of predicting node locations in terms of grid numbers is coarse. This amounts to predicting the position of a node to be somewhere within a 250 m x250m square. This seems to effect scarce, if any, improvement in GLS query success rates. Further, it is noticeable that the aforementioned prediction scheme is by nature more suited to cases where the node movement shows a temporal pattern.

VI. CONCLUSION

In this paper we have a presented an extensive analysis of GLS query success rate and query drop reasons across various mobility models and node speeds. Location error due to infrequent and lost location updates to location servers was identified as one of the major factors leading to query failures. We proposed that location servers themselves employ location prediction mechanisms so as to alleviate query failures due to location errors. An evaluation framework to objectively compare and analyze the performance of the above prediction schemes with GLS across the various mobility models was presented. Our framework investigates various dimensions of the parameter space to provide insight to the performance of the prediction schemes and to the conditions under which such performance is desirable. We implemented two velocity based prediction schemes and a third prediction scheme which exploits patterns in the history of node movement. Prediction accuracy as well as its impact on GLS performance was analyzed. Prediction accuracy depends on mobility models and the prediction scheme being used. Node velocity was identified as the major factor impacting the accuracy of a particular prediction scheme for a specific mobility model.

In future we intend to enhance the velocity-based prediction schemes by take into consideration the direction of node movement. This can be combined with the knowledge of map available at each location server. Probabilities can be used to determine the direction of node movement at each intersection of streets in Manhattan model. A street map can also be used to validate the predicted location. In case of weighted average velocity prediction scheme impact of varying α on GLS performance for different mobility models and node speeds would be an interesting study in itself. It would also be interesting to evaluate history-based prediction for more sophisticated movement models having varying levels of temporal patterns. The current implementation of history-based approach can be extended to one that is frequency based and higher order Markov models with confidence thresholds.

REFERENCES

[1] J. Li, J. Jannotti, D. Couto, D. Karger, R. Morris, A Scalable Location Service for Geographic Ad Hoc Routing (GLS/Grid), *ACM Mobicom* 2000.

[2] F. Bai, N. Sadagopan, and A. Helmy. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *IEEE INFOCOM*, 2003..

[3] D. Son, A. Helmy, B. Krishnamachari, The Effect of Mobility-induced Location Errors on Geographic Routing in Ad Hoc Networks: Analysis and Improvement using Mobility Prediction, *IEEE WCNC*, March 2004..

[4] R. Chellapa, A. Jennings, N. Shenoy, A comparative study of mobility prediction in fixed wireless networks and mobile ad hoc networks. *IEEE ICC* 2003.

[5] K.H. Wang, L. Baochun, Group mobility and partition prediction in wireless ad-hoc networks Communications., *IEEE ICC* 2002.

[6] W. Su, S-J. Lee, M. Gerla, .Mobility prediction in wireless networks, *21st Century Military Communications Conference Proceedings MILCOM* 2000.

[7] A. Bhattacharya; S. K. Das. LeZi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks. *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking MobiCom* 1999

[8] I-R Chen; N. Verma, Simulation study of a class of autonomous host-centric mobility prediction algorithms for wireless cellular and ad hoc networks,*Simulation Symposium*,2003

[9] M. Käsemann, H. Hartenstein, H. Füßler, M. Mauve, A Simulation Study of a Location Service for Position-Based Routing in Mobile Ad Hoc Networks. Technical Report, Reihe Informatik, 7/2002, Universität Mannheim, June 2002

[10] L. Song, D. Kotz, R. Jain, X. He, Evaluating location predictors with extensive Wi-Fi mobility data. In Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), March, 2004.