

# Improving Location Services with Prediction\*

Jeremy Powell and Tracy Camp  
Dept. of Math. and Computer Sciences  
Colorado School of Mines  
Golden, CO 80401  
jepowell@mines.edu, tcamp@mines.edu

**Abstract**—We present the Predictive - Legend Exchange and Augmentation Protocol (P-LEAP), a predictive location service. P-LEAP improves upon the LEAP location service, which has been shown to reduce location error in mobile networks, by incorporating a method of predicting the topology of the network. P-LEAP is designed to create and transmit a data structure that traverses an ad hoc network. The data structure updates all nodes in the network with previous location information stored on the nodes; this information is then used to predict the current location of the nodes in the network. With our new predictive protocol, at a fixed number of packets sent, the location error can be reduced by as much as 68 percent, with an average of 52 percent, compared to previous protocols. Also, with our new predictive protocol, at a fixed location error, the number of packets sent can be reduced by as much as 73 percent, with an average of 45 percent, compared to previous protocols.

**Index Terms**—mobile networks, location services, prediction

## I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is created from numerous wireless mobile nodes which interact with each other by transmitting and receiving data. These networks are created on the fly without existing framework or structure. In these networks, three major problems exist, the first being contention among nodes for communication along the single wireless channel. The second problem relates to data congestion due to the low amount of bandwidth available compared to wired networks. The final complication concerns dynamic topology in which nodes have the ability to leave, enter, and move about the network. All of these problems increase the difficulty of transmitting data in the network. This paper discusses and provides a solution to the last concern, i.e., the dynamic topology in a mobile network.

One difficulty that arises with a MANET is how a node locates another node when both are mobile. As a location service for a MANET, it is important to provide the most reliable spatial information regarding in-

dividual nodes. In customary protocols, this would mean that each node would need to update its neighbors of its location frequently, increasing both the data that flows through the network and the amount of contention in the network. Additionally, the nodes would use a lot of energy to transmit and receive packets.

Our paper addresses this problem of locating mobile nodes. More specifically, our paper first describes a location service that has been proven to lower the average location error in an MANET, i.e., the LEAP with LRV protocol [3], [4], [8]. This paper then describes our modification to the LEAP protocol which adds prediction. Finally, the results of numerous simulations of both LEAP and our predictive protocol are compared and the results show that our predictive protocol performs better than LEAP by lowering both the location error and the amount of packets/bytes sent.

## II. LEAP

### A. LEAP Overview

The Legend Exchange and Augmentation Protocol (LEAP) [3], [4] consists of a data structure called a legend, which travels throughout a MANET updating each node with information regarding the structure of the mobile network. The legend contains an entry for every node in the network. In our application of LEAP, each entry in the legend is comprised of five items for each node, which can be seen in Figure 1. These include each node's ID, the node's last known x and y locations, a timestamp on the location information, and a boolean value that states if the node has been visited by the legend. In this article, we refer to the legend as a Global Location Table (GLT) because our LEAP application is a location service. Each node also has its own version of the GLT, called the Local Location Table (LLT). With the LLT stored locally on the nodes, each node in the MANET is able to locate other nodes.

When the MANET is first created, each node broadcasts hello packets to its neighbors; neighbors then update their LLTs accordingly with the transmitting

\*This work supported in part by NSF Grant #ANI-0240558. Research group's URL is <http://toilers.mines.edu>.

ID	Location	Time	Visited
A	(1,4)	5	1
B	(2,3)	6	1
C	(1,2)	3	0
D	(inf, inf)	0	0
E	(inf, inf)	0	0

Fig. 1. Example of the legend in the LEAP location service protocol.

node's ID, the node's location, and the time the location information was transmitted. In our simulations, we initially allow the nodes to communicate with neighbors and populate their LLTs before the GLT is created by the node with the lowest ID. Once created, the GLT is sent to the closest neighbor. When the neighbor receives the legend, it compares each corresponding entry in the GLT with each entry in the node's LLT; that is, the entry in the GLT for node A is compared with the entry in the LLT for node A. The entry with the oldest timestamp for the node's location information will be replaced with the entry that has the youngest timestamp for the node's location information. For example, if the LLT has a timestamp of 3 for node A, and the GLT has a timestamp of 8 for node A, the LLT will copy node A's entry from the GLT to the LLT, i.e., the GLT's information for node A is more current than the LLT's information regarding node A. Next the GLT's node B entry is compared with the LLT's node B entry in the same manner. When the node has compared each corresponding entry, the node's LLT and the GLT will contain the same information for all nodes. The GLT is then ready to be passed to another node. The following section discusses how the GLT traverses the MANET.

### B. LEAP Traversal Methods

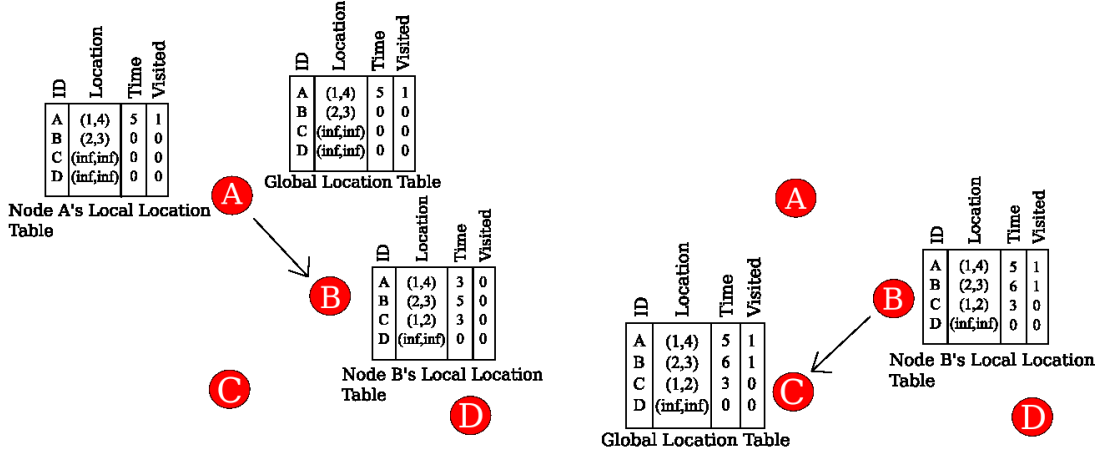
How the legend is passed around the mobile network is an interesting question. In [3], [4], three traversal methods for LEAP were evaluated. The first traversal method evaluated was the Location Aided Routing (LAR) method. LAR sends the GLT to the closest unvisited node with the number of transmission attempts less than three [8]. When a node receives the GLT, it updates both the GLT and its LLT as described in Section II-A. Thus, in addition to the GLT fields shown in Fig. 1, the number of attempts is also stored for each node. When all the nodes in the network have been visited, or have had at least three attempts at being visited, the node that has the GLT resets both the visited

bits and the number of attempts for each node, and begins the cycle again by sending the GLT to the closest unvisited node.

The second traversal method evaluated was the Trace Back (TB) traversal method. The TB method traverses the nodes in the network in a depth-first search manner [3], [4]. With this method, a node receiving the GLT records the ID of the node which transmitted the GLT, defined as the traceback node. The node with the GLT then updates both the GLT and its LLT as described in Section II-A. The node with the GLT then sends the GLT to the node's closest unvisited neighbor, which repeats these steps. If all the neighbors have been visited, the node with the GLT sends the GLT to its traceback node. The traceback node, now with the GLT, tries to send the GLT to its closest unvisited neighbor. Again, if the node with the GLT does not have any unvisited neighbors, it sends the GLT to its traceback node. If the traceback node is no longer a neighbor, or if all the nodes in the network have been visited, the node with the GLT resets its traceback node ID and each visited bit in the GLT, waits for a user specified time, then begins the traversal again by sending the GLT to its closest unvisited neighbor.

The last traversal method evaluated was the Least Recently Visited (LRV) traversal method. The LRV method is based on the idea that the neighboring node with the oldest location information should be updated by the GLT [3], [4]. The node with the GLT sends the GLT to the closest neighbor that has not been visited. If all neighboring nodes have been visited, the transmitting node compares the timestamp of its neighboring nodes' location information timestamps. The GLT is then sent to the node that has the least recent location information stored in the GLT, i.e., the node with the oldest timestamp. After 25 hops, the GLT is paused for a user defined period and then the process is repeated. Note that no values are reset in the LRV method.

An example of the LRV method, which is similar to an example in [3], [4], is shown in Fig. 2. Initially, each node broadcasts a packet which informs neighboring nodes of each other's locations. The node that currently has the GLT (node A) searches its LLT for its neighbor which has the least recent timestamp in the GLT. Node B is selected because LRV chooses the node with the lowest ID value if a tie breaker is needed (e.g., two nodes have the same timestamp). Node A then transmits the GLT to node B, which updates entries in both the GLT and LLT as described in Section II-A. Node B then searches its LLT for the node with the oldest timestamp. Node C is selected because node D is not in B's transmission range. Node B sends the updated GLT to node C, and the process is repeated.



(a) At time = 5, the closest, least recently visited node is node B, so the legend is sent to node B.

(b) At time = 6, node B updates its local location table with information regarding Node A. Node B also updates the global legend with information regarding node B and C. The legend is sent to node C, the closest, least recently visited node.

Fig. 2. An example of the LRV traversal method. In both (a) and (b), the legend is being transferred to a node which the legend has not yet visited.

The LEAP-LRV protocol has been shown to be more desirable than the LAR and TB traversal methods because of its low location error compared to these other methods [3], [4]; however, problems still exist with this protocol. For example, consider what happens if the information in a node's LLT has not been recently updated, and the node tries to transmit data to another node using the old, possibly useless, location information. In this case, it is unlikely that the data will be received by the destination node, if at all. Also, both the transmitting and receiving node will have wasted some of their limited supply of energy from this transaction. To reduce the severity of this problem, nodes could decrease the time spent between receiving and sending the GLT; however, this would lead to increased energy usage and increased data transmitted over the network. A second option is to predict where a node is, in hopes of lowering location error, while maintaining low data transmissions. The following sections discuss our prediction additions to the LEAP-LRV protocol.

### III. P-LEAP

Most location services use previously stored location information as the estimated location of a mobile node. At high speeds and long pauses in data transfer, this method becomes unreliable because a node may no longer have valid location information for another node. As shown in Fig. 3, nodes A and C have the most current location information for node B. At some

later time, pictured in Fig. 4, node B has moved, or is in the process of moving; however, nodes A and C have information stating that node B is at its previous location. To reduce the occurrence of this problem, node B could update the network nodes with its current location as it moves through the network, but this would increase both the amount of packets being passed and the amount of energy used to process these packets. A second solution is to have nodes A and C predict where node B is at a given time.

As stated in Section II-B, the LEAP protocol with the LRV traversal method has been shown to have the lowest average location error compared to other location service protocols; however, in Section IV we show the location error is even lower when we add prediction to LEAP. We call our implementation the Predictive - Legend Exchange and Augmentation Protocol (P-LEAP) using the LRV traversal method. The prediction calculation within P-LEAP is similar to the prediction calculation within the Predictive Location Service (PLS) [14]. Specifically, the GLT for P-LEAP is similar to that of LEAP-LRV, with the addition of three fields: the x velocity, the y velocity, and the speed of every node. P-LEAP works exactly the same way as LEAP-LRV, but rather than using the stored data as the estimated location of a node, it uses a set of prediction equations to predict where a given node currently is.

Similar to LEAP-LRV, a node will periodically update its neighbors with its current location, velocity, and speed. If a second node wants to transmit to the first node, the second node will estimate the first

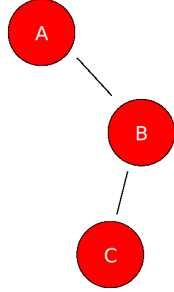


Fig. 3. Nodes A and C have current location information for node B.

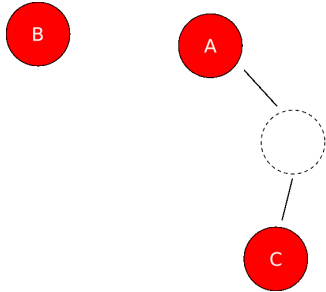


Fig. 4. Node B has moved, but nodes A and C continue to believe that node B is at its previous location.

node's location with the equation:

$$estimateLocation = storedLocation + velocity * t \quad (1)$$

where  $t$  is the time that has elapsed since the node's location information was updated. Our prediction protocol only works if the amount of time that has passed since the node was last updated is reasonable. If a long time has passed since the location information on a node was stored, the node could have reached its destination and gone off in a different direction, i.e., P-LEAP cannot predict where the node is. P-LEAP is not an omniscient location service; P-LEAP can only estimate a node's current location within some maximum time. We use the Prediction Location Factor [14] to adjust for this issue.

Together, the speed and this value called the Predic-

ID	Location	x Velocity	y Velocity	Speed	Time	Visited
A	(1,4)	2	3	10	5	1
B	(2,3)	3	5	10.1	6	1
C	(1,2)	1	-1	9.9	3	0
D	(inf, inf)	0	0	0	0	0
E	(inf, inf)	0	0	0	0	0

Fig. 5. Example of the legend (or GLT) for the P-LEAP protocol.

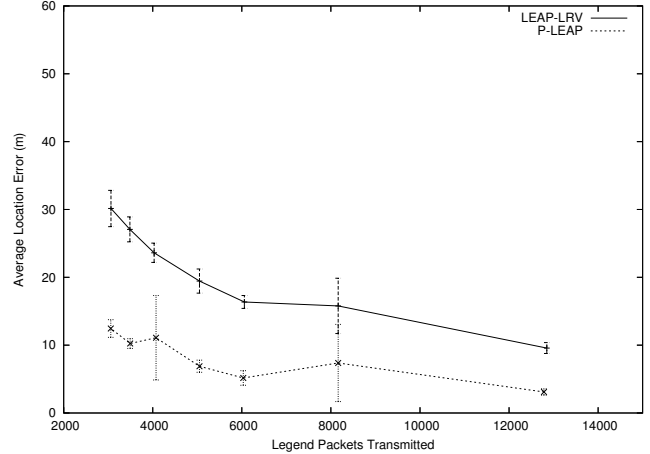


Fig. 6. Location Error vs. Packets Transmitted at 2 m/s. We note that we set the y-axis to be 0 m to 60 m for easier viewing of the results.

tion Location Factor calculate the values used to predict each node's current location. The prediction factor is described by the author of PLS as a way to prevent over prediction of a node's location [14]. To determine if the node's information is indeed valid, the change in time since the previous update is compared to the ratio of the Prediction Location Factor over the speed of the node. If the change in time is greater than the ratio, i.e., if too much time has passed since the last location update, the ratio value is used to predict the location of the node. If the change in time is less than the ratio, the time difference is used. This logic is shown by computing  $t$  using equation 2 and then using the value of  $t$  in equation 1.

$$t = \min(predictionFactor/speed, deltaTime) \quad (2)$$

As mentioned, P-LEAP adds three fields to LEAP-LRV's GLT and LLT. Fig. 5 shows the fields contained in P-LEAP's GLT, which include each node's: ID, most recently updated location, timestamp for location information, visited bit, speed, and x and y velocities. (We note that a z value for each node can be easily added.) The LEAP-LRV packet contains 8 bytes of data for each node's entry; e.g., a legend containing information on 100 mobile nodes would be 800 bytes. If we assume the three new fields in P-LEAP's GLT are one byte each, then the GLT has 11 bytes of data for each node, or 1100 bytes for a packet containing 100 nodes.

#### IV. LEAP vs. P-LEAP

##### A. Simulation Environment

The LEAP-LRV and P-LEAP location service protocols were implemented and compared with the network simulator NS-2 [16]. The size of the simulation

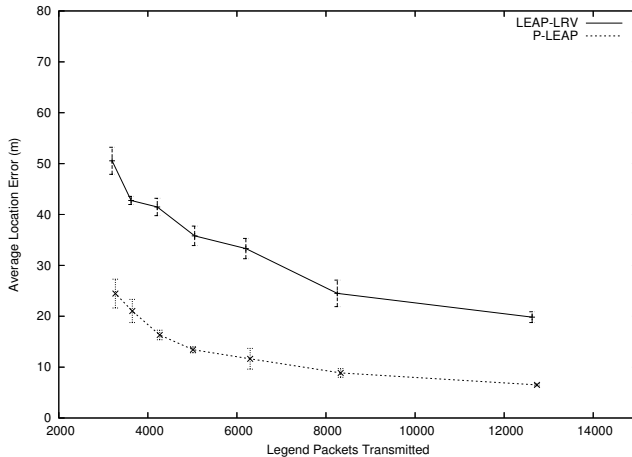


Fig. 7. Location Error vs. Packets Transmitted at 5 m/s. We note that we set the y-axis to be 0 m to 80 m for easier viewing of the results.

area was determined by a set of equations provided in [11], which helps ensure that the protocols are rigorously tested. We execute our simulations in a 447.5 by 895 meter area with 100 nodes, and each node has a transmission range of 100 meters. This simulation area is the smallest area with a 1x2 aspect ratio (which is the ratio of the shorter side of the simulation area to the longer side of the simulation area) that meets the standards specified for rigorous MANET evaluation (defined in [11]). Specifically, in our simulation scenario, the average amount of network partitioning is less than 5% and the average shortest-path hop count is more than 4 hops. (See [11] for details.) The nodes move according to the Random Waypoint Mobility Model, implemented from a file created by the mobgen-ss program [15]. Nodes are given a random destination and speed; when that destination is reached, nodes wait for a specified amount of time, then are given a new destination and speed. Simulations are conducted with nodes moving at 2, 5, 10, 15, 20 m/s  $\pm 10\%$ . Node pause time, the time spent after reaching a destination, is 10 seconds  $\pm 10\%$ . The GLT's pause time, the length of time the GLT waits before being transmitted to the next node, ranges from 1 to 7 seconds; we note that the lower the GLT's pause time is, the more GLT packets are transmitted in the simulation.

Each node in the network transmits a "hello" packet once per "hello" packet interval. This interval is set to  $10m/\bar{V}$ , where  $\bar{V}$  is the average node speed (2, 5, 10, 15, 20 m/s). In other words, nodes transmit "hello" packets more frequently when the average node speed is higher, and less frequently when the average node speed is lower. The total simulation duration is 500 seconds; the first 100 seconds of the simulation are

TABLE I  
SIMULATION DETAILS.

Input Parameters	
Number of Nodes	100
Simulation Area Size	447.5 m x 895 m
Transmission Range	100 m
Simulation Duration	500 s, legend traversing 100-500 s
"Hello" Packet Interval	10 m / Average Node Speed
Mobility Model	
Mobility Model	Random Waypoint [15]
Mobility Speed	2, 5, 10, 15, 20 m/s $\pm 10\%$
Pause Time	10 s $\pm 10\%$
Simulator	
Simulator Used	NS-2 (version 2.1b7a)
Medium Access Protocol	IEEE 802.11
Number of Trials	10
Confidence Interval	95%

used to allow nodes to share "hello" packets and for any initialization to stabilize. The legend then begins traversing the network for 400 seconds. The simulation details are summarized in Table I.

We measure performance in terms of average location error in a given simulation. We define location error as the absolute distance between where a node is estimated to be (via our prediction method) and the node's actual position. We measure overhead in terms of legend packets transmitted, which includes all GLT packets transmitted. We do not count "hello" packets in this overhead for two reasons. First, nodes can usually obtain neighbor knowledge by listening promiscuously to their neighbors' communication, or via a lower layer protocol. Second, the number of "hello" packets transmitted in both LEAP and P-LEAP for a given speed is the same.

### B. Simulation Results

Our simulation results show that nodes using P-LEAP and moving at 2, 5, 10, 15, and 20 m/s produce lower location errors than LEAP-LRV. On average, the location error for P-LEAP is approximately 50% less than LEAP-LRV (or 10 to 40 meters), as seen in Fig. 6 through Fig. 10. At node speed of 2 m/s and with approximately 5000 packets transmitted (a high GLT pause time), P-LEAP outperforms LEAP-LRV by reducing the location error more than 10 meters (see Fig. 6). At a higher speed, 20 m/s and with approximately 5000 packets transmitted, P-LEAP outperforms LEAP-LRV by reducing the location error more than 30 meters (see Fig. 10).

All our results show that P-LEAP consistently improves upon the performance of LEAP-LRV; by in-

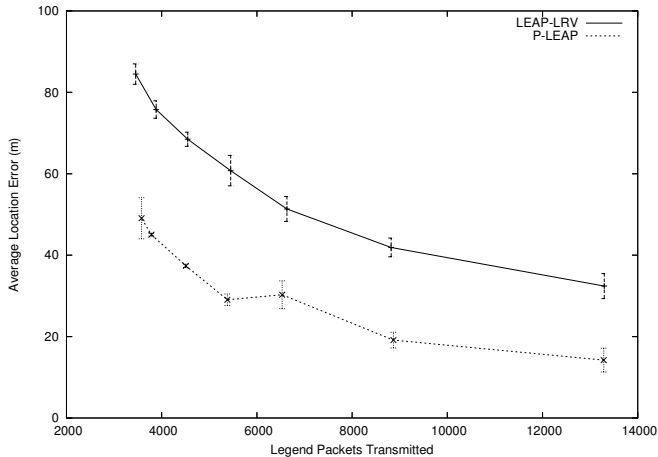


Fig. 8. Location Error vs. Packets Transmitted at 10 m/s. We note that we set the y-axis to be 0 m to 100 m for easier viewing of the results.

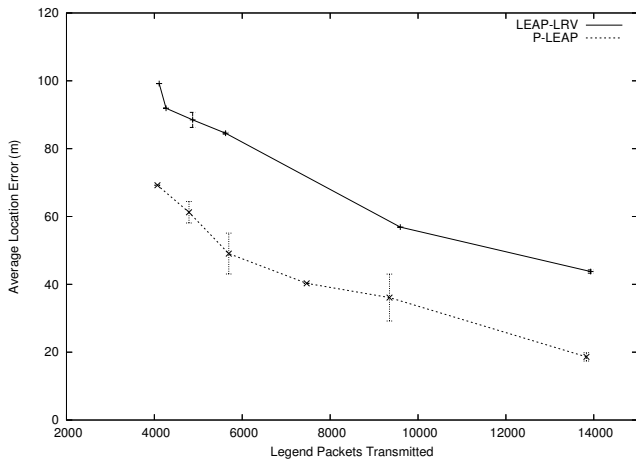


Fig. 9. Location Error vs. Packets Transmitted at 15 m/s. We note that we set the y-axis to be 0 m to 120 m for easier viewing of the results.

cluding prediction, P-LEAP can significantly decrease the location error in a MANET. Specifically, on average, P-LEAP offers 52% less location error than LEAP-LRV. At low node speeds, such as 2 and 5 m/s, P-LEAP offers approximately 60%, with the maximum being 68.5%, less location error than LEAP-LRV. At high node speeds, such as 20 m/s, P-LEAP offers approximately 30% less location error than LEAP-LRV. In all our simulation trials, P-LEAP offers 21.9% less location error in the worst case.

If a certain error is acceptable as a maximum error in the location information, say 25 meters at 5 m/s, Fig. 7 shows that approximately 8000 packets would need to be transmitted with the LEAP-LRV protocol. With the same maximum error, approximately 3000 packets would need to be transmitted with our P-LEAP protocol (i.e., a savings of approximately 62.5 percent). If we

assume each LRV packet has 8 bytes of data and each P-LEAP packet has 11 bytes of data, LEAP-LRV and P-LEAP send approximately 64,000 and 33,000 bytes of data, respectively, in this example. That is, P-LEAP would send approximately half of the bytes sent by LRV. If 55 meters of location error is acceptable when node speed is 20 m/s, then LEAP-LRV would need to transmit approximately 14,000 packets while P-LEAP would need to transmit approximately 7700 packets (i.e., a savings of approximately 45 percent), as shown in Fig. 10. Assuming the number of bytes previously discussed, LEAP-LRV would transmit approximately 112,000 bytes while P-LEAP would transmit approximately 84,700 bytes in this example (i.e., a savings of approximately 25 percent).

## V. RELATED WORK

For a review of both proactive and reactive location services, we direct interested readers to [5]. The use of prediction in mobile networks was first proposed in [13]. In this paper, the authors propose a mobile tracking scheme that exploits the predictability of user mobility patterns in wireless PCS networks. As far as we are aware, only two location prediction techniques for MANETs have been previously proposed: Predictive Location Service (PLS) [14] and the Dead Reckoning Method (DRM) [1], [10]. We note that these two predictive location services for MANETs were developed concurrently, and that they are similar in terms of how each node predicts other nodes' locations.

One difference between PLS and DRM is how location information traverses the network. In PLS, a node's location table is periodically sent to neighbors, similar to the Internet standard Routing Information Protocol (RIP) [7]. In DRM, on the other hand, if a node updates its own location models, it must flood the network with this newly calculated information. A second difference concerns when a node determines to transmit new location information. In PLS, how often a location update packet is sent to neighbors depends on the average speed of a node. Nodes using DRM periodically predict their own location using stored location information. If the node's predicted location differs from the node's actual location by a given threshold, then the node transmits updated information to other nodes. Lastly, unlike PLS, the location service of DRM is included with a complete location-based routing protocol similar to the Location-Aided Routing (LAR) protocol [9] and DREAM [2].

Lastly, in this paper, we only compare P-LEAP to LEAP. We note, however, that LEAP was compared to three other location services in [8], i.e., the Grid Location Service [12], the Simple Location Service [6],

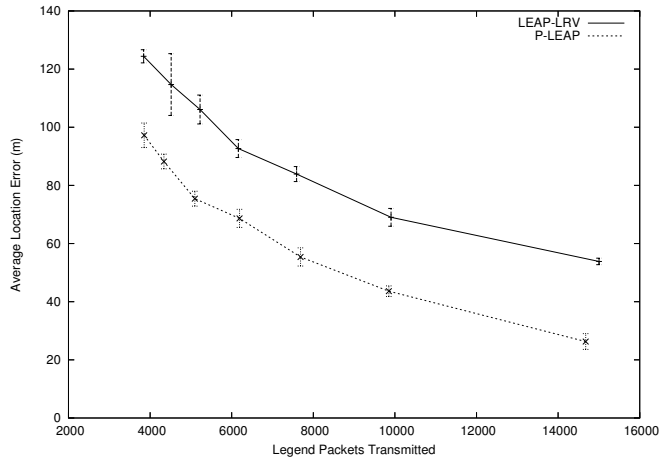


Fig. 10. Location Error vs. Packets Transmitted at 20 m/s. We note that we set the y-axis to be 0 m to 140 m for easier viewing of the results.

and the Reactive Location Service [6]. The simulation results within that paper illustrate that LEAP consistently outperforms these three other location services. Because P-LEAP outperforms LEAP, we expect P-LEAP would outperform these three other location services as well.

## VI. CONCLUSIONS

In this paper, we present the Predictive - Legend Exchange and Augmentation Protocol (P-LEAP), which is a predictive location service. By including the velocity and speed of each node in the location information shared within the network, nodes in a MANET can predict other node's location. We have shown that P-LEAP, compared to LEAP, reduces the amount of location error by approximately 50% at a fixed number of packets sent. A lower location error in the estimated locations of destinations means that data packets transmitted will more likely be received correctly. Also, we have shown that P-LEAP, compared to LEAP, reduces the average number of packets transmitted by approximately 45% at a fixed location error. By lowering the number of control packets transmitted in the network, each node will retain more energy, which can then be used for a more useful purpose.

## REFERENCES

- [1] A. Agarwal and S.R. Das. Dead reckoning in mobile ad hoc networks, Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), pp. 1838-1843, March 2003.
- [2] S. Basagni, I. Chlamtac, V.R. Syrotyuk, and B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp. 76-84, 1998.
- [3] N. Bauer, M. Colagrosso, and T. Camp, An agile approach to distributed information dissemination in mobile ad hoc networks, Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 131-141, 2005.
- [4] N. Bauer, M. Colagrosso, and T. Camp, Efficient implementations of all-to-all broadcasting in mobile ad hoc networks, Pervasive and Mobile Computing, vol. 1, no. 3, pp. 311-342, September 2005.
- [5] T. Camp, Location information services in mobile ad hoc networks, Chapter 14 in Handbook of Algorithms for Mobile and Wireless Networking and Computing, (editor A. Boukerche), pp. 317-339, 2005.
- [6] T. Camp, J. Boleng, and L. Wilcox, Location information services in mobile ad hoc networks, Proceedings of the IEEE International Conference on Communications (ICC '02), pp. 3318-3324, 2002.
- [7] C. Hedrick, Routing Information Protocol, Request for Comments 1058, June 1988.
- [8] X. Jiang and T. Camp, An information dissemination protocol for an ad hoc network, Proceedings of the 23rd IEEE International Conference on Performance, Computing, and Communications Conference (IPCCC), pp. 337-345, 2004.
- [9] Y. Ko and N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp. 66-75, 1998.
- [10] V. Kumar and S.R. Das. Performance of dead reckoning-based location service for mobile ad hoc networks. Wireless Communications and Mobile Computing Journal, vol. 4, no. 2, pp. 189-202, March 2004.
- [11] S. Kurkowski, T. Camp, and W. Navidi. Two Standards for Rigorous MANET Routing Protocol Evaluation. Submitted 2006.
- [12] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, A scalable location service for geographic ad-hoc routing, Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp. 120-130, 2000.
- [13] B. Liang and Z. J. Haas, Predictive distance based mobility management for pcs networks, Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 1377-1384, 1999.
- [14] X. Luo, T. Camp, and W. Navidi, Predictive methods for location services in mobile ad hoc networks, Proceedings of the 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), pp. 246-252, 2005.
- [15] W. Navidi, T. Camp, and N. Bauer. Improving the accuracy of random waypoint simulations through steady-state initialization, Proceedings of the 15th International Conference on Modeling and Simulation (MS), pp. 319-326, March 2004.
- [16] The VINT Project. The network simulator ns-2. <http://www.isi.edu/nsnam/ns/>. Page accessed on February 5th, 2006.