A Review of Very Recent Research into Location Services for Position-Based Routing

Rich Goyette
Department of National Defence
goyette.rg@forces.gc.ca

Abstract

Position-based routing protocols use position information supplied through GPS, multilateration, etc. to assist mobile ad-hoc and sensor networks in the routing task while reducing control traffic and memory overhead when compared to traditional position-less protocols. However, position-based routing requires a *location service* to assist in the maintenance and dissemination of location information and the success of position-based protocols rests on the ability of the location service to be both effective and efficient. In this paper, we review four very recent developments in this area of research.

I. Introduction

Routing protocols that use node position information to assist with routing are known as *position-based* routing protocols ([S1] [C]). In such protocols, global routing decisions are made using local knowledge of neighbouring nodes and position information about the destination node [S1]. One of the main advantages of position-based routing protocols is that they appear to scale well with increasing network size since they do not need to maintain or communicate large routing tables. [MWH] [C]. Another advantage relates to the intelligent use of power. Ad-hoc networks of sensors are characterized by a battery limited power budget. Position information about neighbouring nodes can be used to help modulate the transmitted power and extend the useful lifetime of a sensor network [S2].

The chief disadvantage of position-based routing protocols is *precisely* that a source node S requires a reasonably accurate geographical location of the destination node D in order to make a routing decision. This requires that each node is both able to determine its own location and to share that information efficiently across the network.

Unfortunately, the maintenance and sharing of position information in a scalable, robust, and bandwidth efficient manner does not appear to be trivial - especially in high mobility networks. As pointed out in [S1]: "the problem of designing location update schemes to provide accurate destination information and enable routing in mobile ad hoc networks appears to be more difficult than routing itself…" Despite the challenge (or perhaps because of it), location services for position based routing remains an active area of research.

In the literature, protocols that are designed to handle location information have come to be known as *location services* [MWH]. A location service is composed of at least the following service primitives [PQ]:

i.) Location Registration
ii.) Location Update
iii.) Location Query

It is usually assumed that a node has an ability to determine its own location (normally assumed through the use of GPS) so this does not appear as a service primitive. However, position determination is a separate but active area of research (e.g. [HE][GMS][BGJ]) since GPS receivers are not necessarily appropriate in all circumstances.

In the rest of this paper, we review four very recent works (i.e. 2005) in location services. These works are described from the perspective of their location service primitives. Each description is drawn directly from the paper under review unless otherwise stated. A separate section is provided for a lighter review of some other recent research in the field.

II. Predictive Methods for Location Services in Mobile Ad Hoc Networks [LCN]

This work proposes a new location service called the Prediction Location Service (PLS). The objective is to use predictive methods to improve the location accuracy of other mobile nodes in the network. This work both extends and is compared to the Grid Location Service (GLS) [LJDKM], the Simple Location Service (SLS) [CBW], and the Legend Exchange and Augmentation Protocol (LEAP) [JC]. In all of these location services, each of the mobile nodes maintains a physical location table that contains information about other nodes in the network. In some cases, information in these tables is either not available or has expired and an update mechanism is engaged to obtain the most current information. In other cases, information is available and has not expired, yet it may be "stale" (in that it may not perfectly locate the requested node). In PLS, predictive methods are used to augment valid location information to produce a better estimate of a nodes actual location.

Like GLS and SLS, PLS incorporates a mechanism (flooding) for obtaining position information when such information is not found in a node's location table. When this information is present, it is combined with the desired node's last known velocity to predict its current state.

The service primitives for PLS and their mechanisms are briefly described next (from [LCN])

<u>Location Registration</u>: This phase is not explicitly described in [LCN]. However, it is assumed that flooding is used to compose the initial location table entries.

Location Update: Location updates are periodically and proactively sent to other nodes in the network via a "Location Packet" (LP). LPs are transmitted from a node to all its one-hop neighbors. Each LP contains the current position and velocity of the source node as well as a configurable number of recently updated entries from the source node's own location table. LPs are triggered when, after moving some distance d since the last LP was transmitted, the source node exceeds some constant threshold from its own predicted location. Since all other nodes will be using the same means to predict the source's location, this threshold forces the network to track accelerations in node movement. LPs are also triggered if the node moves some constant threshold from the location of its last update as well as after some constant time threshold has been exceeded.

Location Query: When a destination node location is queried from the source's local location table and it exists and has not expired, then the source node computes the predicted location of the destination node according to the following equation (equation (1) from [LCN]):

$$location = location_{record} + (v \times T)$$
$$T = \max(predictionFactor/speed, (t_{now} - t_{record}))$$

This equation represents the core mechanism of the paper. The variable $location_{record}$ is the record for the destination node being sought in the location table, v is the velocity computed during the last location update, and T is the greater ofPredictionFactor/speed or t_{now}-t_{record}. PredictionFactor is a constant inserted by the authors to prevent over-prediction and was determined iteratively for maximum optimization.

Experimentally, three simulation studies were performed to compare PLS against GLS, SLS, and LEAP. In all studies a simulated 802.11 MAC was used and the performance metrics were percentage of location requests answered, average location error, and overhead in bytes. A summary of the experimental studies is shown in Table 1.

	% Location Requests Answered	**Average Location Error**	**Overhead in Bytes**
Study 1: Increasing number of nodes from 20-150.	PLS, SLS, LEAP achieve close to %100	PLS superior for sparse nets, LEAP superior for dense nets.	PLS superior
Study 2: Increasing node speed for 50 nodes.	LEAP, PLS superior	PLS superior	PLS superior except for region around 5m/s
Study 3: Variable number and speed of nodes.	PLS, SLS, LEAP achieve close to %100	PLS superior for sparse nets, LEAP superior for dense nets.	PLS superior

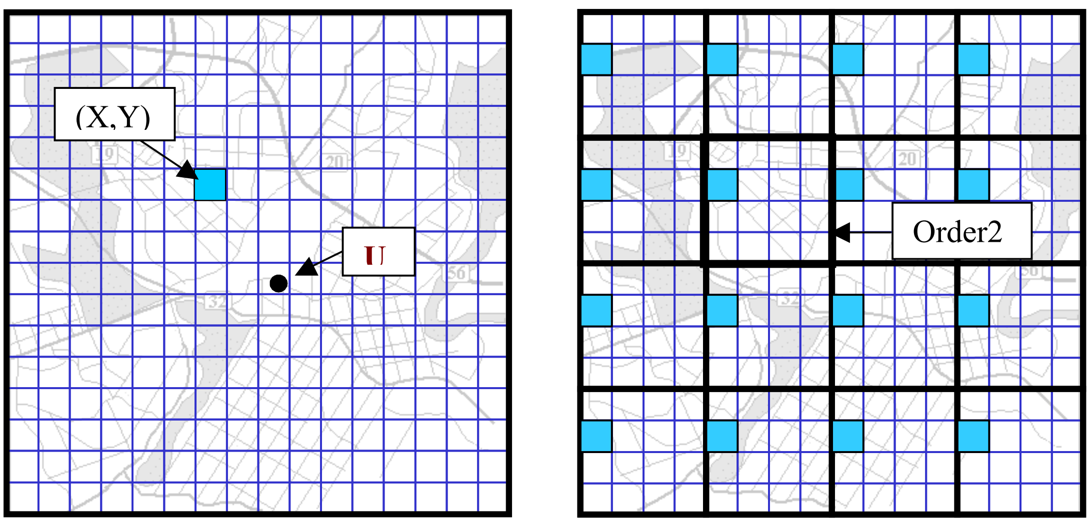
Table 1: Experimental Comparison of PLS to LEAP, GLS, SLS

As indicated by the results in Table 1, the authors conclude that PLS is generally superior to GLS, SLS and LEAP for the performance metrics of lower overhead and location error. In addition, PLS demonstrated fair robustness in the various dynamic study environments.

III. Adaptive Demand-driven Location Service [SPHL]

The authors of this work propose the Adaptive Demand-driven Location Service (ADLS) in an effort to scale the class of home region location services to large (metropolitan) area networks. In this algorithm, the overhead involved in creating and maintaining multiple home regions to address geographical (or logical) separation is off-set by allowing home regions to be created *on-demand* in order to service local requests for a particular node or set of nodes.

In this work, the performance of ADLS is compared to SLURP [WS] and SLALOM [CLPBZ], both of which are home-based location service algorithms often referenced in the literature.



a. b.

Figure 1: SLURP and SLALOM Geographical Division

As indicated in Figure 1a, the SLURP algorithm divides the geographical operating area of a network into multiple grid squares. The home region of all nodes in the network is then determined by performing a hash function on the unique node ID to produce an (x,y) coordinate. The grid square in which this coordinate resides is the home region for the node and all nodes in that square are responsible for maintaining position information for the node. An obvious scalability limit of SLURP is that location requests may have to traverse large distances in order to reach another node's home region. SLALOM is an approach that attempts to enhance the scalability of SLURP and to improve query performance by dividing the network area into Order1 and Order2 squares as shown in Figure 1b. Order2 squares are made up of k Order1 squares and each node's ID is hashed to produce a home region in every Order2 square. Although scalability and query performance is improved in SLALOM, there is an overhead cost in maintaining each of the home regions.

Like SLALOM, ADLS divides the geographical network region into Order1 and Order2 squares. However, like SLURP, only a single primary home region is set up and maintained for each node. The service primitives and their mechanisms for ADLS are briefly paraphrased below (from [SPHL]).

<u>Location Registration</u>: Location registration is not explicitly discussed in [SPHL] since the case of joining or leaving the network is not discussed. However, joining the network would involve simply determining the home region for the new node and then propagating an update to that square.

<u>Location Update</u>: In the simulations in [SPHL], position updates appear to be performed by each node after a certain, fixed timeout period rather than as implemented in SLURP and SLALOM. In those algorithms, an update to the home region is triggered when a node moves from one Order1 square to another (SLALOM also updates home regions defined as "near" during this operation). Additionally, like SLURP, the primary home region remains fixed and all location updates propagate there. In contrast, the "primary" home region in SLALOM is always the home region in the Order2 square in which the node currently resides. When a node crosses an Order2 boundary, all home regions are updated to the new "primary" home region in the current Order2 square and this contributes greatly to the overhead of the algorithm.

<u>Location Query</u>: The location query primitive is the area where ADLS seeks to improve over SLURP and SLALOM the most. When node A wishes to communicate with node B in Figure 2, it will (by default) query B's designated home region in its own Order2 square. If this is not B's primary home region and if the home region does not have current information on the location of B, then a node in the home region will forward the request to the primary home region on behalf of B. A node in B's primary home region will then forward the most recent position information to back to A (black lines in Figure 2).

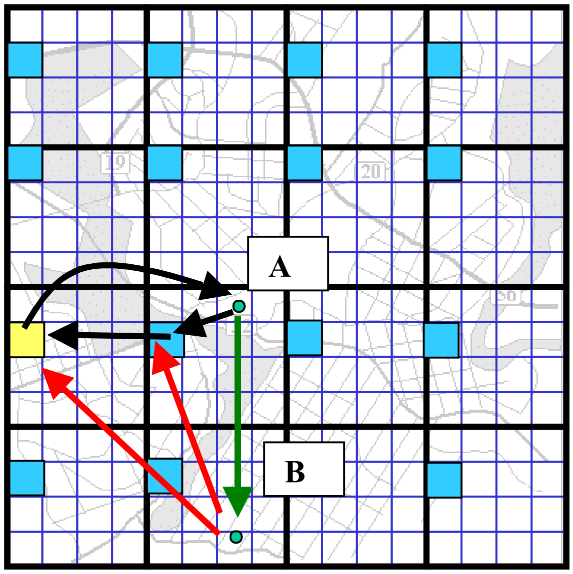


Figure XX: Location Query in ADLS

At this point, A will initiate communication with B (green line) whereupon B will realize that the request came from an Order2 square that does not contain its primary home region. B will then send an update packet to that home region as well as its primary home region (red lines).

All further references to B by any node in A's Order2 square will have immediate access to B's position information. However, the updates given to the home region in A's Order2 square are maintained only for a certain time period after which another reference to B's primary home region is required.

Although node mobility is considered in this work, it is not clear how node movement is handled by the update process. Secondary home region lifetimes are set to 25s while nodes can move with speeds of up to 20m/s. This represents substantial physical movement. It is assumed that nodes maintain a list of secondary home regions and update them with the same frequency as the primary home regions.

The simulated performance of ADLS indicates that it has about the same control overhead as SLURP which is significantly less than SLALOM. Location discovery latency for ADLS lay, as expected, between that of SLALOM (the least) and SLURP (the most). Finally, location update traffic for ADLS was found to be only marginally greater than SLURP and significantly less than SLALOM (which must maintain the home regions in all "near" Order2 squares).

The authors note that the simulations were constrained. Packet loss due to congestion was not modeled and the node density was maintained deliberately high since there were no recovery strategies simulated.

IV. Optimal Tradeoffs for Location-Based Routing in Large-Scale Ad hoc Networks [PS]

The aim of this work was to develop a location management scheme that simultaneously satisfies the requirements of scalability, bandwidth and power efficiency and supports the discovery and recovery of minimum cost routes. The authors contend that no work to date attempts to satisfy all three. Interestingly, this work does not seek to create a new algorithm or significantly extend an existing one. Instead, it combines salient features from a number of other algorithms that are successful against certain metrics into a hybrid approach.

The hybrid approach described is scalable; the memory requirement on each node is O(logN) and routing is localized. The approach provides power and bandwidth efficiency by providing a means of optimizing the location update frequency (a major portion of the paper is devoted to deriving the conditions of optimality for the frequency of location update and then validating these through simulation). By allowing each node to optimize its update frequency, bandwidth and power consumption are reduced. The algorithm provides a mechanism to find multi-path routes between a source and destination and a means of filtering out a route that meets energy efficiency or QoS constraints. This is achieved by a hybrid of elements from other algorithms. Finally, methods are provided for route recovery and to deal with certain instances of routing failure.

The hybrid approach elements are described below according to location management primitive.

<u>Location registration</u>: The hybrid approach has each node making use of a *beaconing* system to maintain awareness of its local neighbors. Together with the location update service, this beaconing system serves to provide the location registration mechanism. Each node maintains a position table of directly reachable neighbors (i.e. those neighbors within transmission range) through the exchange of beacon packets. A node will transmit a beacon packet containing its position and all neighbors will update their position table for that node. Beacon packet frequency is determined on a node-by-node basis. When a new node enters the network, its beacon packets will register it with its local neighbors and a location update will provide the other information required by the hybrid approach (discussed below). If a node does not transmit beacon packets for some threshold period of time, nodes having its position in their tables will remove it. Hence registration and deregistration is performed in a passive fashion.

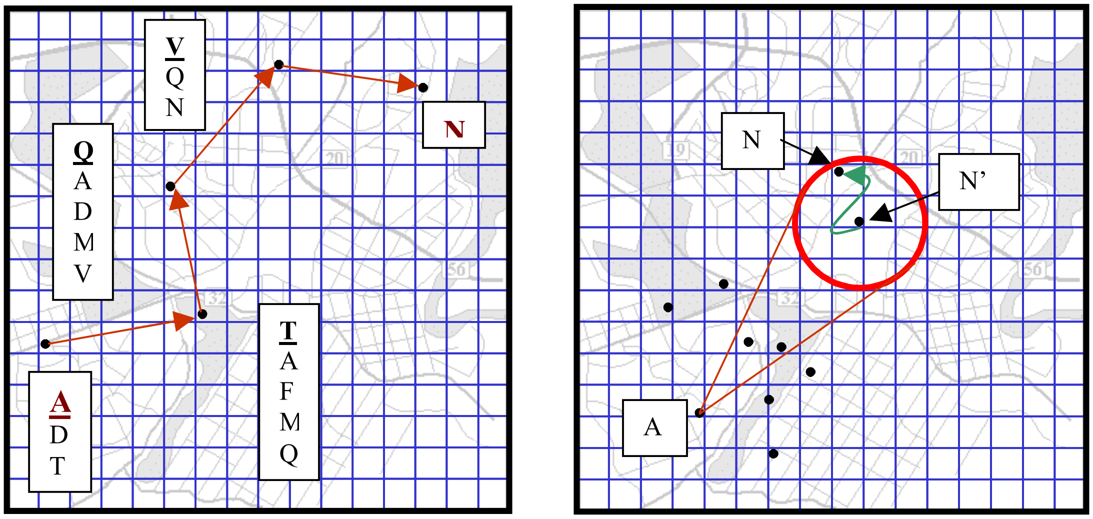
<u>Location update</u>: Location updates will occur when a certain threshold time has expired or when a node has displaced itself a certain distance since the last update that it performed. Location updates are performed by the transmission of Location INFOrmation (or LINFO) packets. LINFO packets contain a node's current position and speed (not velocity). A node first floods a LINFO packet to all of its known neighbors. LINFO packets destined for these neighbors may also contain a radius value. As each

neighbor receives the LINFO packet, they evaluate whether they are within the radius or not and update appropriately. Each node then re-broadcasts the LINFO packet to each of its neighbors. This mechanism, as described by the authors, allows a node to make use of the "distance effect" proposed as part of DREAM [BCS]. Because the LINFO packet is re-broadcasted, the radius value in the packet need not be constrained to the neighborhood of the local node.

LINFO packets are also constructed and sent to *location servers* for that node. The node maintains a list of other nodes it considers to be location servers. These servers are chosen based on how close their node IDs are to the node that has chosen them. This point is important later for efficient routing. LINFO packets are unicasted to each location server (using an established geographic based routing protocol) and each LINFO packet contains the unique node ID of the server. Servers that receive LINFO packets compare the timestamp of the packet to the entry contained in their database and will, if appropriate, update the speed and position of the entry.

<u>Location query</u>: Unlike other approaches, location query involves two phases. The first is, of course, for a node A to find the physical location of a presumably unknown node N. The second phase is then to determine an optimal route to node N.

a.) <u>Phase 1</u> (destination location): If node A has communicated with node N previously and the position information held by A for N is not considered stale, then Phase 2 (route discovery) is commenced immediately. If not, then the procedure illustrated in Figure 3a is performed. Node A determines node N's ID number (assume here that the number is its position number in the alphabet). It then searches its location server table for a location server with the smallest node ID that is (equal to) or greater than the node ID of N (in this case T). A then forwards a Location QeuRY (LQRY) packet to T using a geographic routing protocol. This process is repeated by each succeeding location server (T forwards to Q and Q forwards to V), until a location server having knowledge of N is reached at which point the LQRY packet is provided to N. N will respond to A with an updated position or, to save time, it will commence route discovery back to A.



a. b.

Figure 3 Destination Location and Route Discovery:

b.) Phase 2 (route discovery): This phase is similar in implementation to the scoped search performed in DREAM. Node N has responded to A with its current position and speed (or A had this information cached). Node A uses this information to compute a radius R_N in which N could have moved since the timestamp on its last known position given its last known speed (in Figure 3b, N moves from N'). A uses this information to compute an arc in which to send a Route REQuest (RREQ) packet. In Figure 3b, two of A's neighbors receive the RREQ packet and perform the same computation using local information about N (if they have it). RREQ packets will multiply in the direction of N and each packet will accumulate the node IDs through which it has passed. Therefore, it is likely that more than one RREQ will be received at N. Node N waits for a fixed period of time collecting RREQ packets and then uses the node list in each packet to determine the "best" route based on one or more routing or power metrics.

The authors then extend this process in an energy-aware manner by forcing each node receiving a RREQ packet to perform an additional step. Here, the source node selects other node(s) within the scope arc which are the furthest within range, and which still have sufficient power (measured by battery life) remaining. These are known as "border nodes." Between these nodes and the originating node, RREQ packets are transmitted within an ellipse that encloses the border and source nodes and then the border node selects the best "partial route" that maximizes the energy between the border and the source. This process is repeated by each border node until the destination is reached. The best energy aware route is the concatenation of all local partial routes.

A second major goal of this work was to derive the conditions of optimality for the frequency of location update that could be set by each node. The ability to change the location update frequency in a dynamic way at each node results in a reduction in control overhead and a minimization of bandwidth and energy utilization. The derivation is complex and much of it is beyond the scope of our understanding. The optimized frequency update depends on certain parameters that are time variable and the authors indicated that they are working towards a parameter estimation algorithm to solve this issue. However, the theoretical model for certain, fixed parameters was validated in simulation.

V. Scalable Location Services for Hierarchically Organized Mobile Ad hoc Networks

This work introduces a novel strategy to reduce the control overhead involved in the *update* primitive of location management services. The authors target only the update primitive because they feel that this type of control traffic dominates all other types. In their strategy, a node sends a location update to its associated home zone via a dominating set thereby limiting the constraining the control traffic without adversely affecting performance. In simulated performance, the strategy suggested in this work performs well against GLS [LJDKM], AODV[PRC] and DSR[JMH] - all three of which rely to various extents on network flooding for elements of their location management primitives.

The strategy in this work targets the use of clustering to form a dominating set. While clustering methods are hierarchical and scalable, it is noted in this work that they are not generally applied to mobile networks because the frequency at which clusters will need to be re-defined (due quite often to the smallest of changes in network mobility) causes unmanageable control overhead. Thus, the impetus of this work is to focus on stable clustering in the face of node mobility.

The clustering mechanism is based on the concept of a "virtual cluster." As shown in Figure Xxa, the static area to be covered by the scheme is divided up into a number of uniquely identified virtual clusters that are defined by a hash function. To determine a "home cluster" for a node, the hash function is used to map the node ID to a cluster ID. All nodes that happen to be physically located within the geographic bounds of a virtual cluster are then responsible for all other nodes that have elected that virtual cluster as a home region.

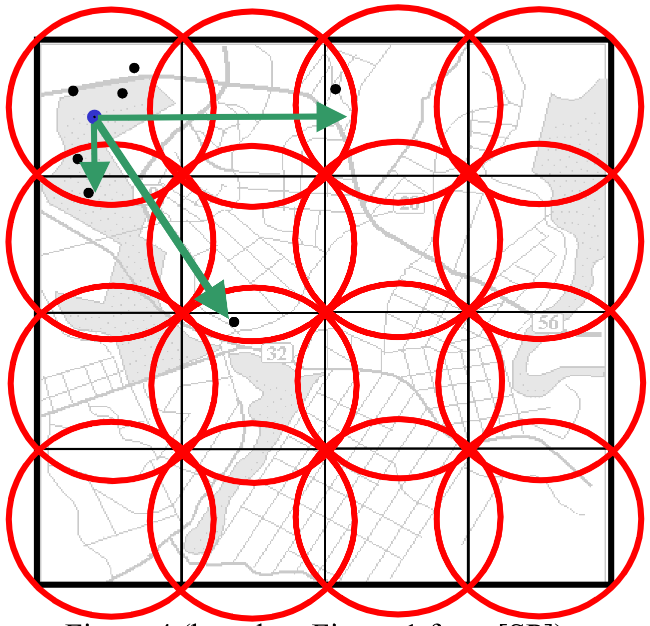


Figure 4 (based on Figure 1 from [SP])
Clusters and Cluster Head Update

For each cluster, a cluster head is elected (blue node in Figure 4). A node is elected a cluster head if it stays closer to the center of the virtual cluster longer than any other node and if it has moved slower or stayed at rest longer than any other node. In this way, cluster heads are chosen to maximize long term stability. If a cluster head determines that it is about to leave the virtual cluster that it represents, it locates a successor node (based on the criterion above) and then informs the cluster members via an update packet. Based on this architecture framework, the location management primitives of this approach are now briefly described.

<u>Location registration</u>: All nodes broadcast periodic HELLO packets and these are collected and tabulated by the cluster head representing the virtual cluster in which a node resides. It is assumed that de-registration occurs as a result of the cessation of HELLO broadcast and housekeeping within the cluster head node.

<u>Location update</u>: This is the primitive in which the maximum overhead reduction is achieved. All nodes report their positions to the cluster head of their current cluster. Cluster heads maintain a virtual communications backbone by unicasting HELLO messages among themselves. To perform a location update, a cluster head tabulates and groups all nodes that it knows about by their home cluster and then it sends these lists toward each respective cluster head. When the nodes in a virtual cluster receive a group update, they update their own location table (recall, each node is responsible to maintain information for which the current virtual cluster is a home cluster). They forward this updated table to their neighbors by way of a HELLO packet. Note that, for nodes moving faster than a certain threshold, they are responsible for sending update packets directly to their home region.

Location query: Location query is achieved by routing a location request in the direction of the virtual cluster ID that is determined using the node ID of the desired destination node and the hash function. Any node in the virtual cluster can respond to the location request.

VI Other Research

In this section, we discuss in a more succinct manner some other research performed in the field of location services. We start with Abraham et al [ADM] who have proposed a location service entitled Locality Aware Location Service (LLS). In this work, the authors use a spiral-flooding algorithm to locate nodes which are mapped into a hierarchical lattice. The authors claim this work to be the first location service whose "lookup and publish algorithms have worst case locality guarantees and average case locality awareness

VI Summary

In this review, four very recent papers in the field of location services for position-based routing were examined. The mechanisms of each work according to the three location service primitives (location query, location update, and location registration) were revealed in overview and a summary of the reported performance of the algorithms in each paper was provided. Predictive methods were used to anticipate the location of mobile nodes in a dynamic environment and reduce control overhead. A SLURP and SLALOM like approach saw the reduction in control overhead by populating secondary home regions with location information on-demand. In [PS] an algorithm that jointly optimizes scalability and bandwidth/power consumption is under development. Finally, scalable location services based on a clustering algorithm that is not brittle in the face of mobility was described.

References

Core References

[LCN] X. Luo, T. Camp, W. Navidi, Predictive methods for location services in mobile ad hoc networks, Proc. 19th IEEE Int. Parallel and Distributed Processing Symposium 2005 (IPDPS'05), pp. 246b-246b.

[PS] T. Park, and K. Shin, Optimal tradeoffs for location-based routing in large-scale ad hoc networks, IEEE/ACM Transactions on Networking, April 2005, vol. 13, no. 2, pp. 398-410.

[SP] S. Sivavakeesar, G. Pavlou, Scalable location services for hierarchically organized mobile ad hoc networks, Proc. MobiHoc'05, Urbana-Champaign, Illinois, May 25-27, pp. 217-228.

[SPHL] B.C. Seet, Y. Pan, W.J Hsu, C.T. Lau, Multi-home region location service for wireless ad hoc networks: an adaptive demand-driven approach, Proc. 2nd Annual Conf. On Wireless On-Demand Network Systems and Services (WONS'05), pp. 258-263.

Normal References

[BCS] S. Basagni, I. Chlamtac, V. Syrotiuk, A distance routing effect algorithm for mobility, Proc. 4th Annual ACM/IEEE Int. Conf. On Mobile Computing and Networking MOBICOM'98, pp. 76-84, 1998.

[BGJ] J. Bruck, J. Gao, A. Jiang, Localization and routing in sensor networks by local angle information, Proc. 6th ACM Intl. Symp. on Mobile ad-hoc Networking and Computing, May. 2005, pp. 181-192.

[C] T. Camp, Location information services in mobile ad hoc networks, Technical Report MCS-03-15, The Colorado School of Mines, October 2003.

[CBW] T. Camp, J. Boleng, L. Wilcox, Location information services in mobile ad hoc networks, IEEE International Conference on Communications, 2002 (ICC 2002), vol. 5, pp. 3318 – 3324.

[CLPBZ] C. Cheng, H. Lemberg, S. Philip, E. van den Berg, T. Zhang, SLALoM: a scalable location management scheme for large mobile ad-hoc networks, Proc. IEEE Wireless Communications and Networking Conference, 2002 (WCNC2002), vol. 2, pp. 574 – 578.

[GMS] S. Guha, R. Murty, E. Sirer, Sextant: A unified node and event localization framework using non-convex constraints, Proc. 6th ACM Intl. Symp. on Mobile ad-hoc Networking and Computing, May. 2005, pp. 205-216.

[HE] L. Hu, D. Evans, Localization for mobile sensor networks, Proc. 10th Intl. Conf. on Mobile Computing and Networking, Oct. 2004, pp. 45-57.

[LJDKM]J. Li, T. Jannotti, D. DeCouto, D. Karger, R. Morris, A scalable location service for geographic ad-hoc routing, Proc. ACM/IEEE MOBICOM, 2000, pp. 120-130.

[MWH] M. Mauve, J. Widmer, H. Hartenstein, A survey on position-based routing in mobile ad-hoc networks, *IEEE Network Magazine*, 15(6):30 - 39, November 2001.

[S1] I. Stojmenovic, Position-based routing in ad hoc networks, IEEE Communications Magazine, July 2002, pp. 2-8.

[S2] I. Stojmenovic, Location updates for efficient routing in ad hoc networks, in: Handbook on Wireless Networks and Mobile Computing, John Wiley & Sons, 2002, to appear.

[SQ] S. Philip, C. Qiao, ELF: efficient location forwarding in ad hoc networks, Proc. GLOBECOM '03 IEEE, vol. 2, 1-5 Dec. 2003, pp. 913 – 918.

[WS] S.C.M. Woo, S. Singh, Scalable routing protocol for ad hoc networks, Wireless Networks 7, pp. 513-529, Kluwer Academic Publishers, 2001.

Secondary References (discussed or referenced in primary work but not read)

[F] G. Finn, Routing and addressing problems in large metropolitan-scale internetworks, ISI Report, ISI/RR-87-180, Mar. 1987 (not included in document repository).

[JC] X. Jiang, T. Camp, An information dissemination protocol for an ad hoc network, Proc. 23rd IEEE ICPCC, 2004, pp. 337-345.

[JMH] D. Johnson, D. Maltz, Y. Hu, The dynamic source routing protocol for mobile ad hoc networks (DSR), IETF internet draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt.

[KV] Y.B. Ko and N. Vaida, Location-aided routing (LAR) in mobile ad hoc networks, ACM/IEEE MobiCom'98, Oct. 1998.

[PRC] C. Perkins, E. Belding-Royer, I. Chakeres, Ad hoc on demand distance vector (AODV) routing, IETF Internet draft, http://www.ietf.org/rfc/rfc3561.txt.